FIREBRAND

# Linux

## LPIC Level 1 Certification

# LPIC Courseware

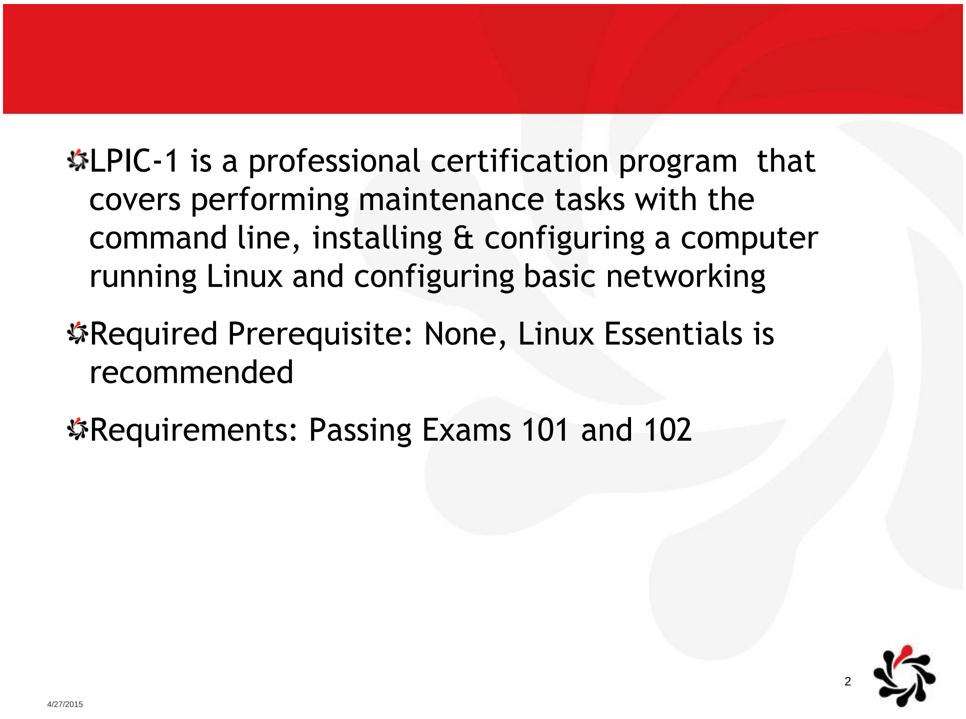
Version 2.0

www.firebrandtraining.com

FIREBRAND

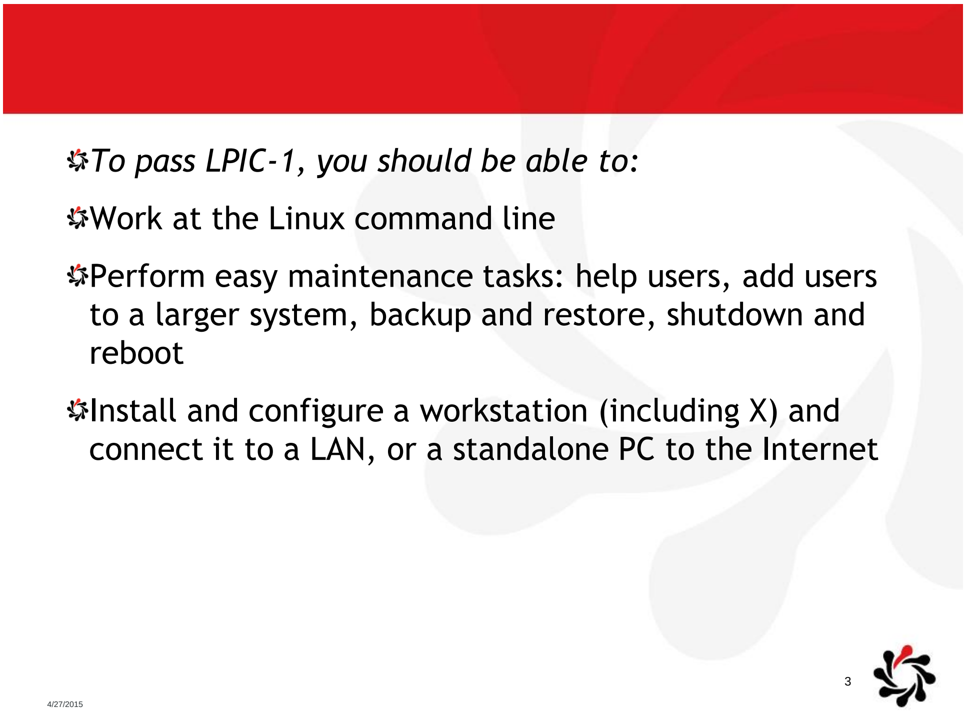
# Linux LPI Level 1

## Course Introduction



- LPIC-1 is a professional certification program that covers performing maintenance tasks with the command line, installing & configuring a computer running Linux and configuring basic networking
- Required Prerequisite: None, Linux Essentials is recommended
- Requirements: Passing Exams 101 and 102

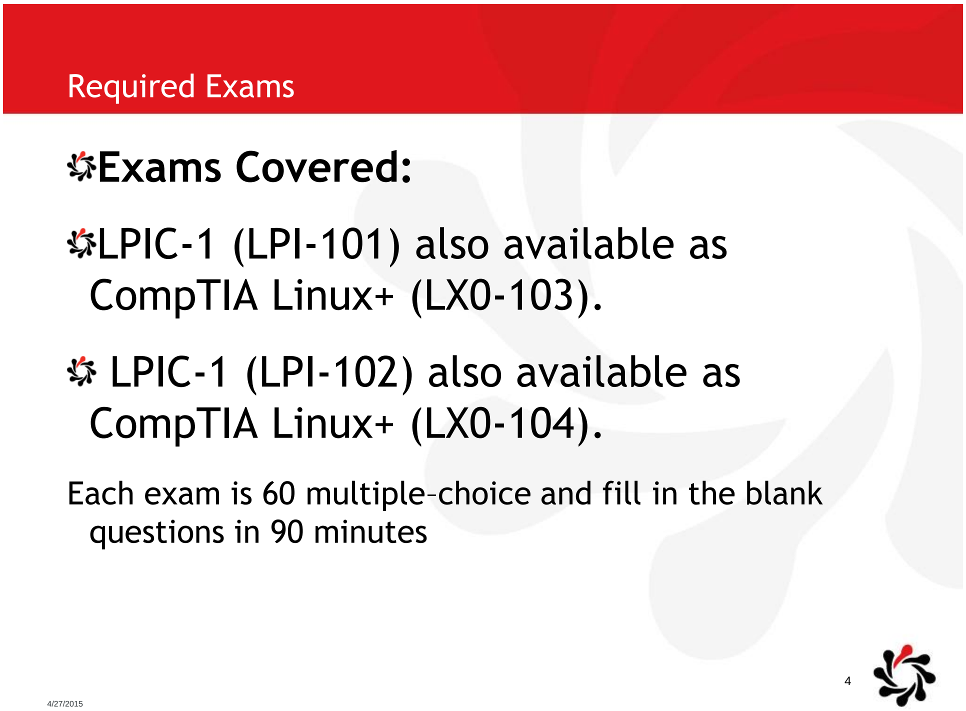
4/27/2015



- *To pass LPIC-1, you should be able to:*
- Work at the Linux command line
- Perform easy maintenance tasks: help users, add users to a larger system, backup and restore, shutdown and reboot
- Install and configure a workstation (including X) and connect it to a LAN, or a standalone PC to the Internet

3

4/27/2015



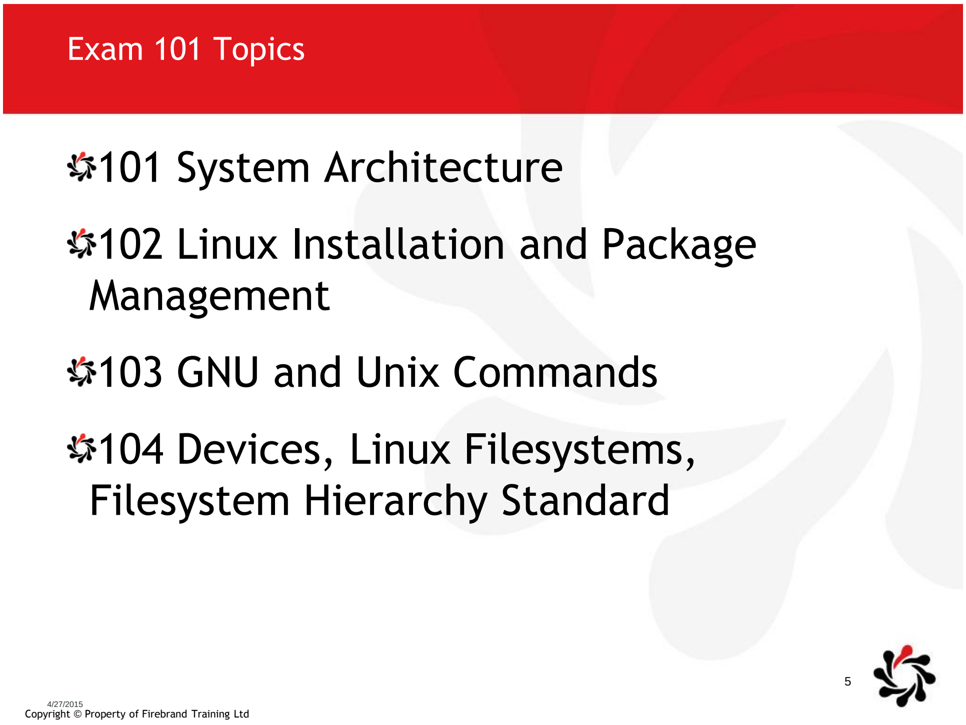
## Required Exams

- **Exams Covered:**
- LPIC-1 (LPI-101) also available as CompTIA Linux+ (LX0-103).
- LPIC-1 (LPI-102) also available as CompTIA Linux+ (LX0-104).

Each exam is 60 multiple-choice and fill in the blank questions in 90 minutes

4

4/27/2015

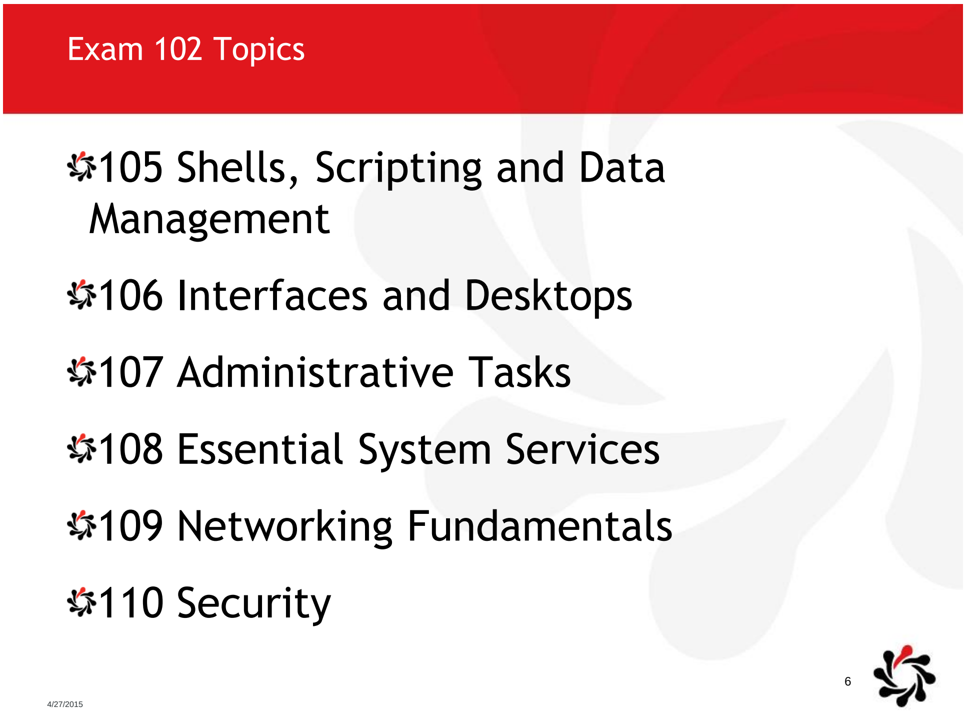


## Exam 101 Topics

- 101 System Architecture
- 102 Linux Installation and Package Management
- 103 GNU and Unix Commands
- 104 Devices, Linux Filesystems, Filesystem Hierarchy Standard

4/27/2015

5

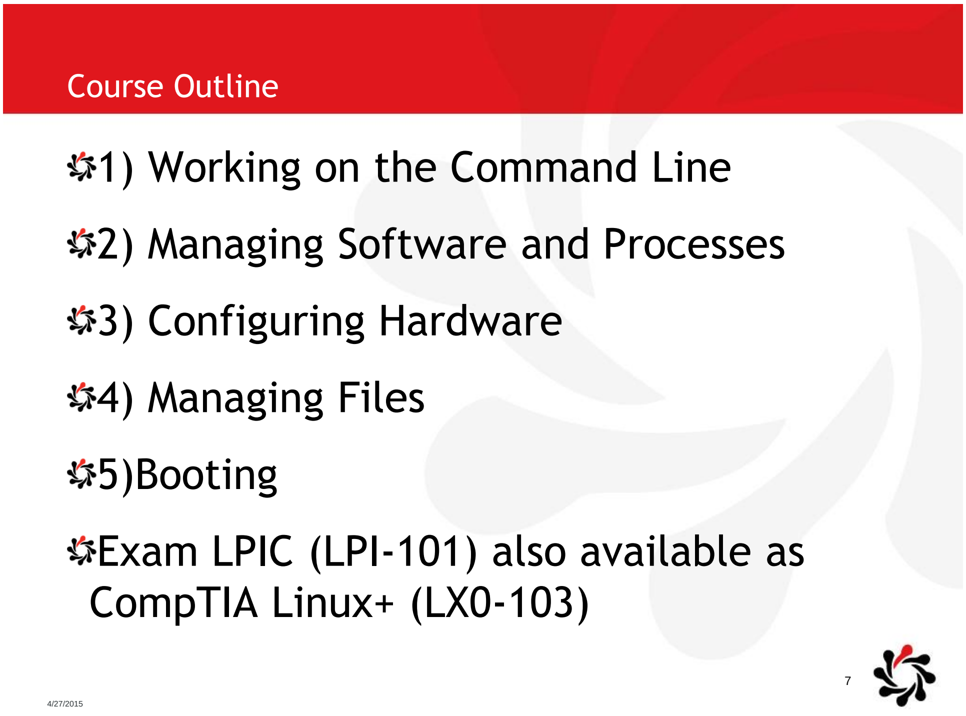


## Exam 102 Topics

- 105 Shells, Scripting and Data Management
- 106 Interfaces and Desktops
- 107 Administrative Tasks
- 108 Essential System Services
- 109 Networking Fundamentals
- 110 Security

4/27/2015

6

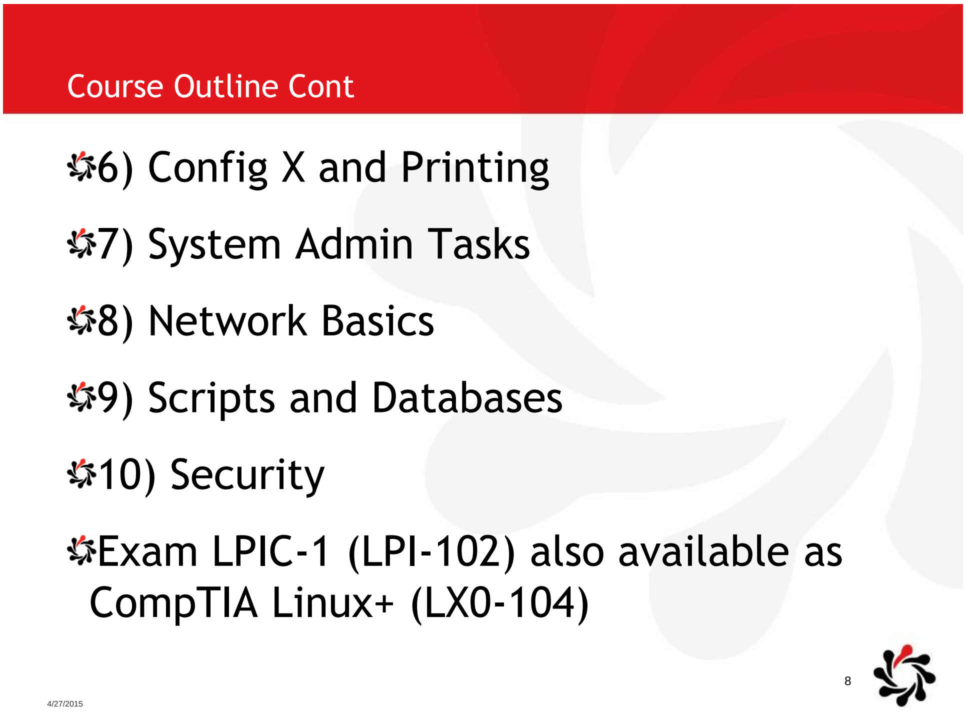


## Course Outline

- 1) Working on the Command Line
- 2) Managing Software and Processes
- 3) Configuring Hardware
- 4) Managing Files
- 5)Booting
- Exam LPIC (LPI-101) also available as CompTIA Linux+ (LX0-103)

7

4/27/2015

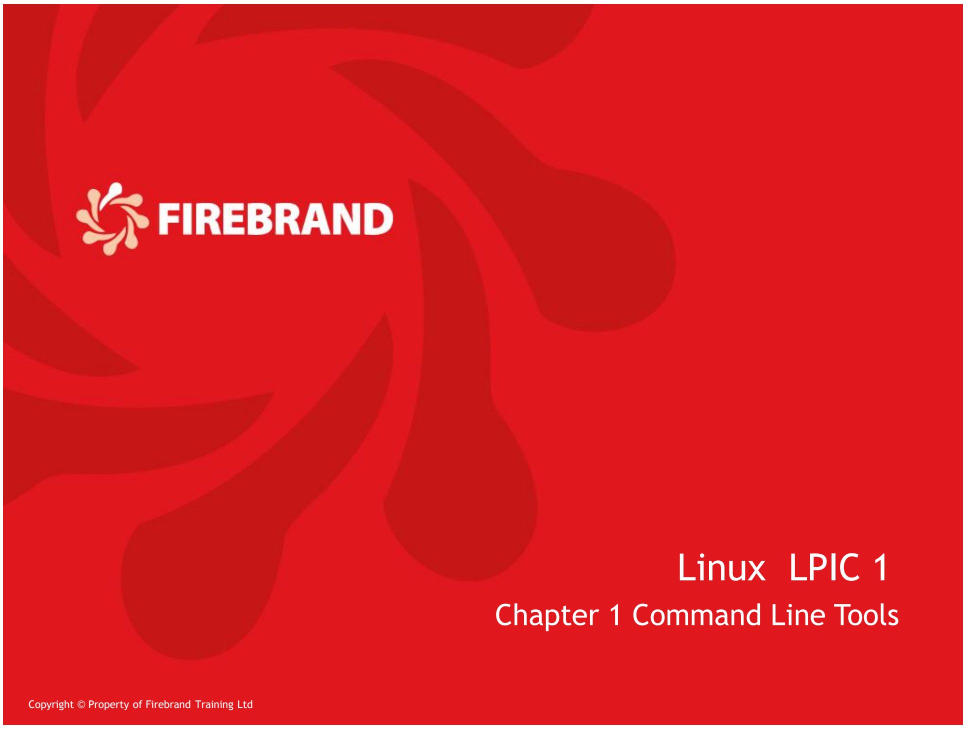


## Course Outline Cont

- 6) Config X and Printing
- 7) System Admin Tasks
- 8) Network Basics
- 9) Scripts and Databases
- 10) Security
- Exam LPIC-1 (LPI-102) also available as CompTIA Linux+ (LX0-104)

8

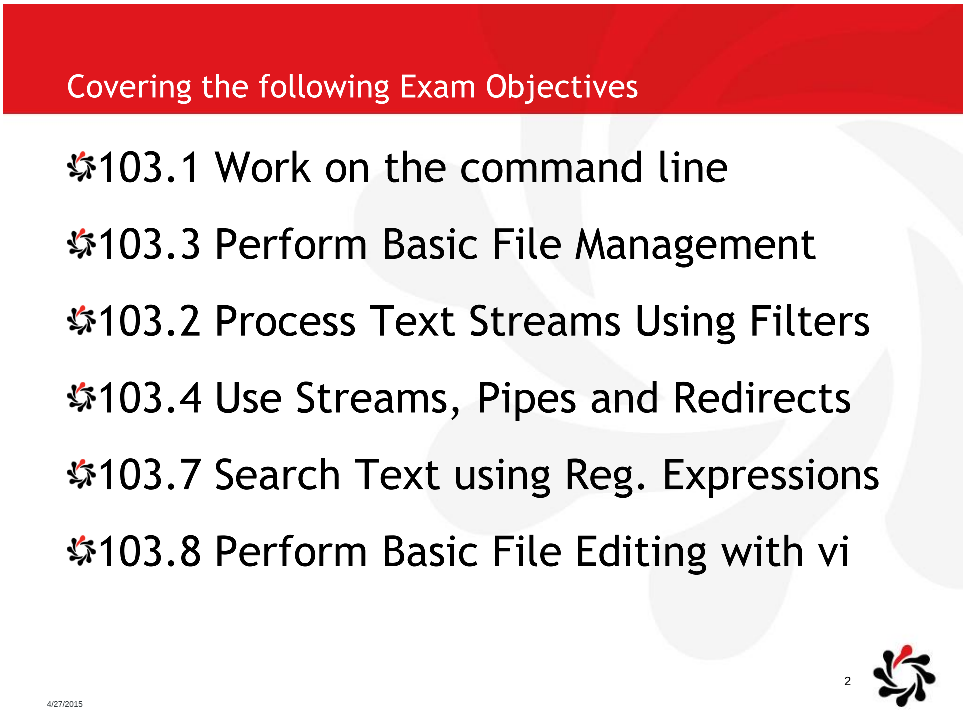
4/27/2015



FIREBRAND

# Linux LPIC 1

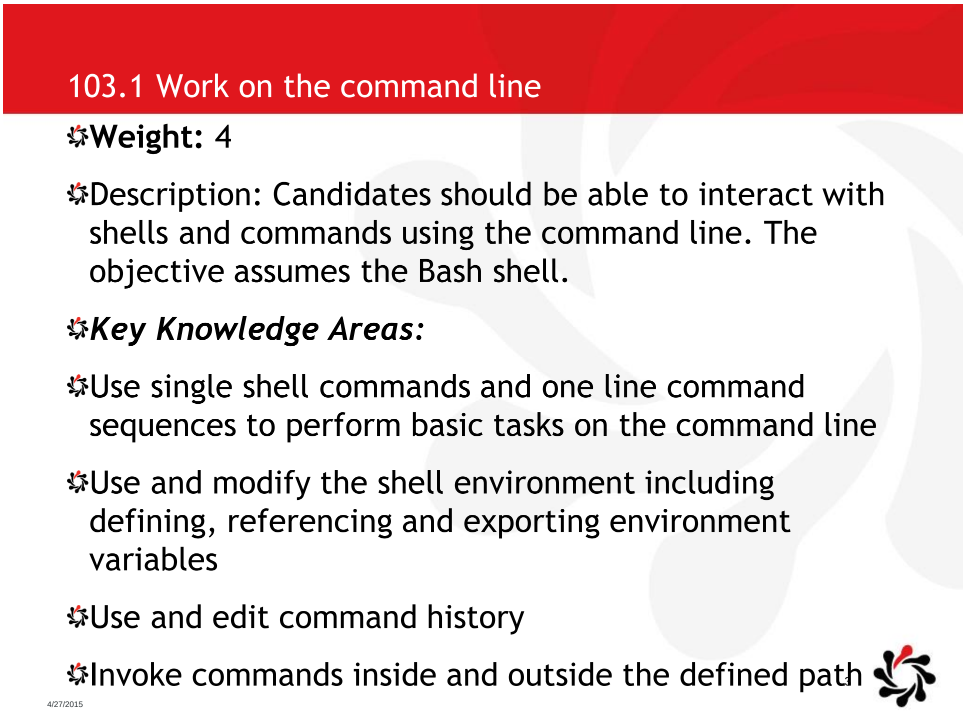
## Chapter 1 Command Line Tools



## Covering the following Exam Objectives

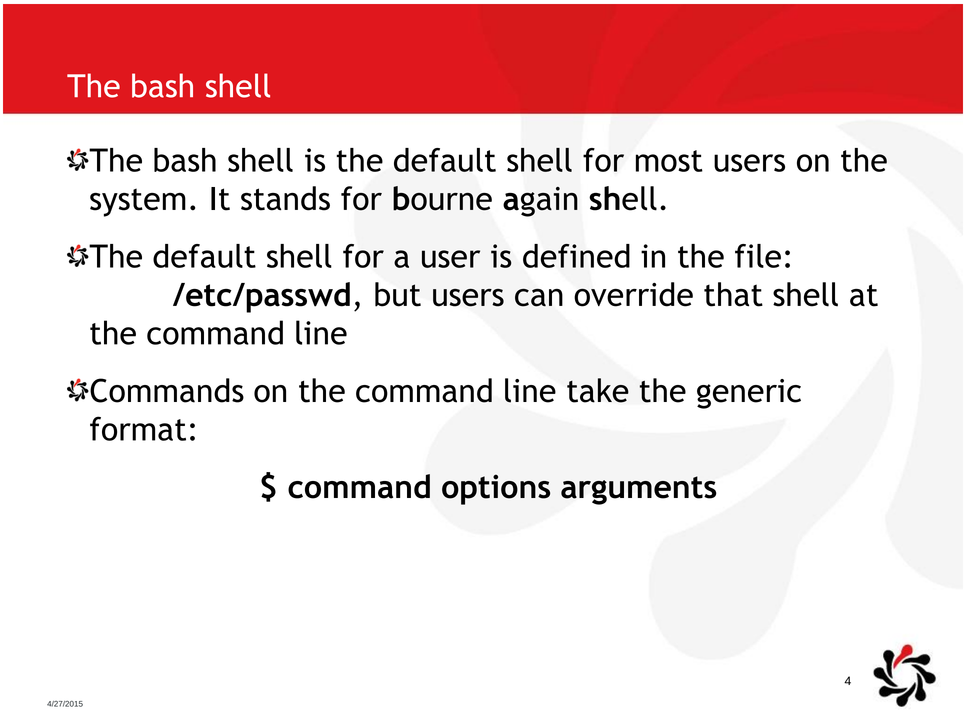
- 103.1 Work on the command line
- 103.3 Perform Basic File Management
- 103.2 Process Text Streams Using Filters
- 103.4 Use Streams, Pipes and Redirects
- 103.7 Search Text using Reg. Expressions
- 103.8 Perform Basic File Editing with vi

4/27/2015



## 103.1 Work on the command line

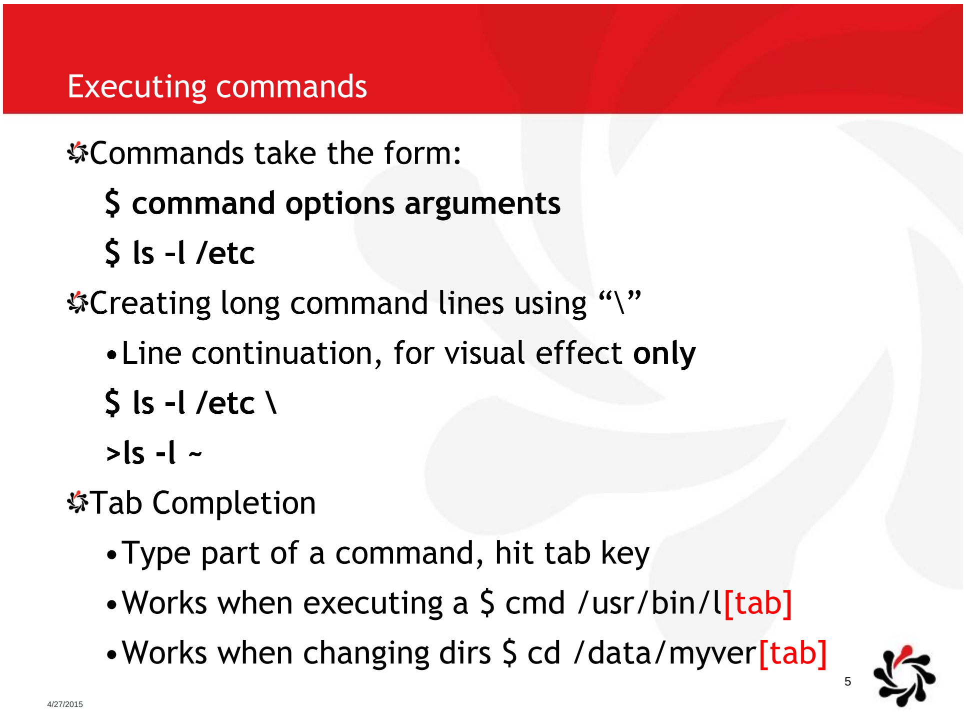
- **Weight:** 4
- Description: Candidates should be able to interact with shells and commands using the command line. The objective assumes the Bash shell.
- ***Key Knowledge Areas:***
- Use single shell commands and one line command sequences to perform basic tasks on the command line
- Use and modify the shell environment including defining, referencing and exporting environment variables
- Use and edit command history
- Invoke commands inside and outside the defined path



## The bash shell

- The bash shell is the default shell for most users on the system. It stands for **b**ourne **a**gain **sh**ell.
- The default shell for a user is defined in the file: **/etc/passwd**, but users can override that shell at the command line
- Commands on the command line take the generic format:

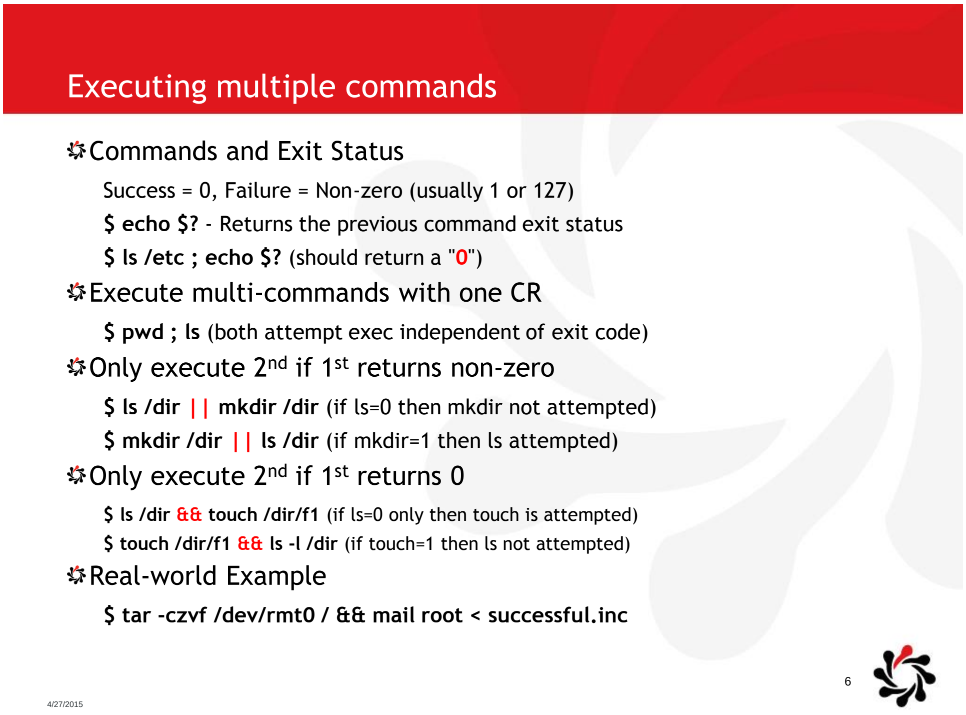
**$ command options arguments**



## Executing commands

- Commands take the form:
  - **$ command options arguments**
  - **$ ls -l /etc**
- Creating long command lines using “\”
  - Line continuation, for visual effect **only**
  - **$ ls -l /etc \**
  - **>ls -l ~**
- Tab Completion
  - Type part of a command, hit tab key
  - Works when executing a $ cmd /usr/bin/l[tab]
  - Works when changing dirs $ cd /data/myver[tab]

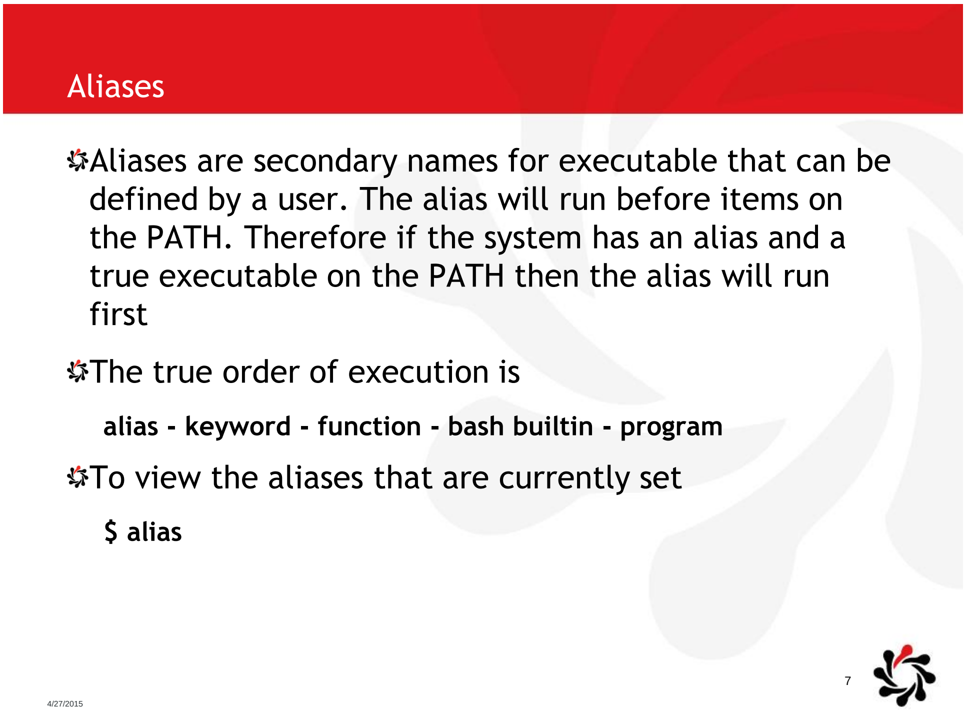
4/27/2015 5



## Executing multiple commands

- Commands and Exit Status
  - Success = 0, Failure = Non-zero (usually 1 or 127)
  - **$ echo $?** - Returns the previous command exit status
  - **$ ls /etc ; echo $?** (should return a "0")
- Execute multi-commands with one CR
  - **$ pwd ; ls** (both attempt exec independent of exit code)
- Only execute 2nd if 1st returns non-zero
  - **$ ls /dir || mkdir /dir** (if ls=0 then mkdir not attempted)
  - **$ mkdir /dir || ls /dir** (if mkdir=1 then ls attempted)
- Only execute 2nd if 1st returns 0
  - **$ ls /dir && touch /dir/f1** (if ls=0 only then touch is attempted)
  - **$ touch /dir/f1 && ls -l /dir** (if touch=1 then ls not attempted)
- Real-world Example
  - **$ tar -czvf /dev/rmt0 / && mail root < successful.inc**

4/27/2015 6



## Aliases

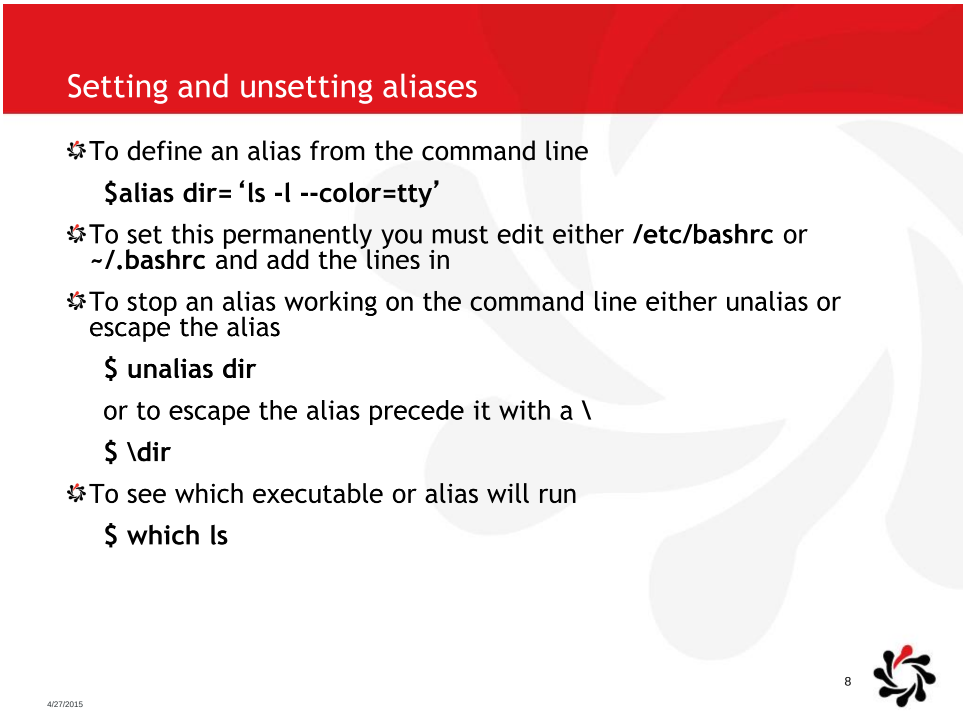
- Aliases are secondary names for executable that can be defined by a user. The alias will run before items on the PATH. Therefore if the system has an alias and a true executable on the PATH then the alias will run first
- The true order of execution is

  **alias - keyword - function - bash builtin - program**
- To view the aliases that are currently set

```
$ alias
```

4/27/2015

7



## Setting and unsetting aliases

- To define an alias from the command line

```
$alias dir=‘ls -l --color=tty’
```

- To set this permanently you must edit either **/etc/bashrc** or **~/.bashrc** and add the lines in
- To stop an alias working on the command line either unalias or escape the alias

```
$ unalias dir
```

  or to escape the alias precede it with a \

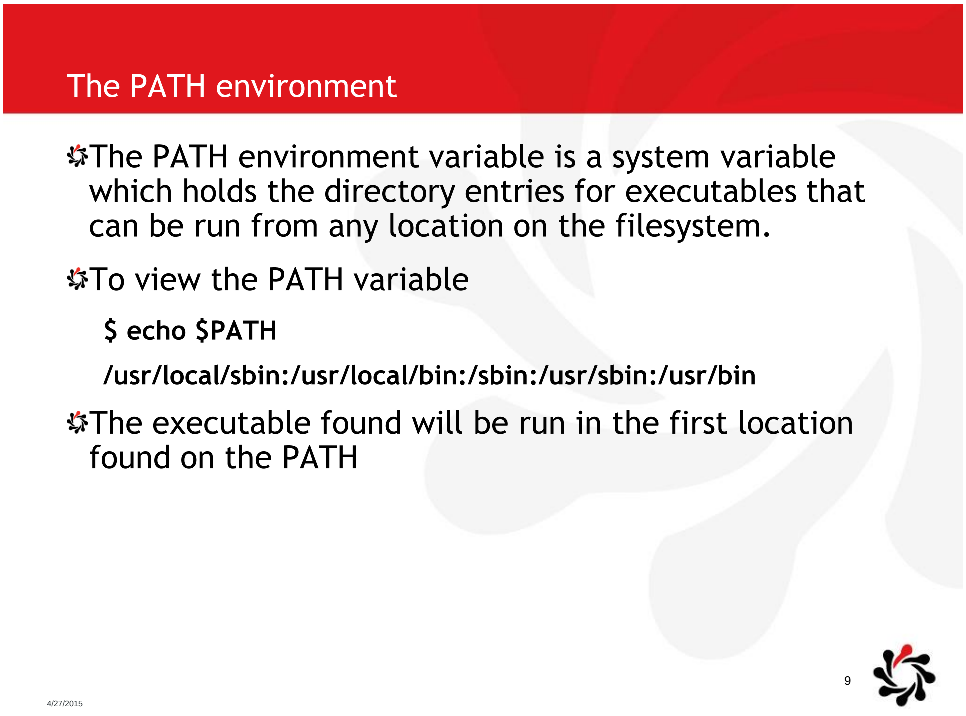
```
$ \dir
```

- To see which executable or alias will run

```
$ which ls
```

4/27/2015

8

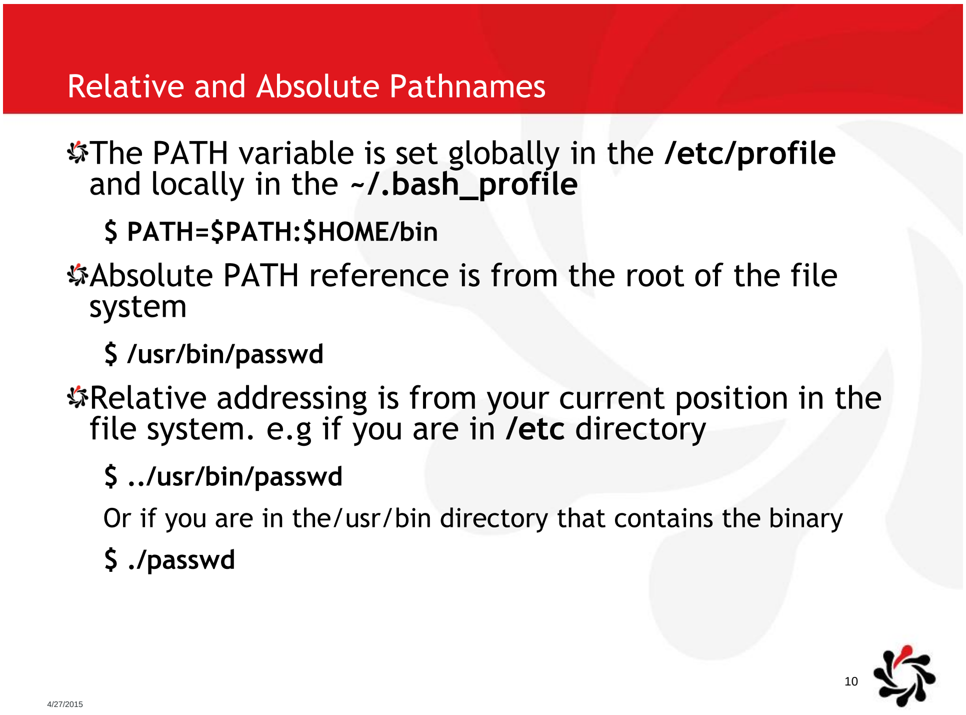


## The PATH environment

- The PATH environment variable is a system variable which holds the directory entries for executables that can be run from any location on the filesystem.
- To view the PATH variable

```
$ echo $PATH
/usr/local/sbin:/usr/local/bin:/sbin:/usr/sbin:/usr/bin
```

- The executable found will be run in the first location found on the PATH



## Relative and Absolute Pathnames

- The PATH variable is set globally in the **/etc/profile** and locally in the **~/.bash_profile**

```
$ PATH=$PATH:$HOME/bin
```

- Absolute PATH reference is from the root of the file system

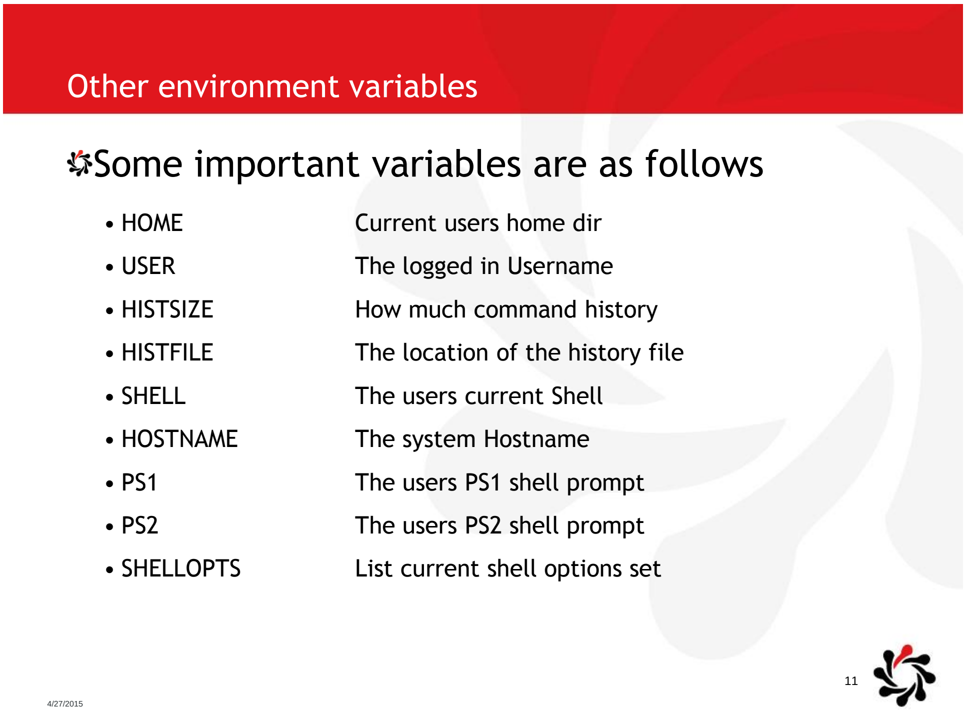
```
$ /usr/bin/passwd
```

- Relative addressing is from your current position in the file system. e.g if you are in **/etc** directory

```
$ ../usr/bin/passwd
```

Or if you are in the/usr/bin directory that contains the binary

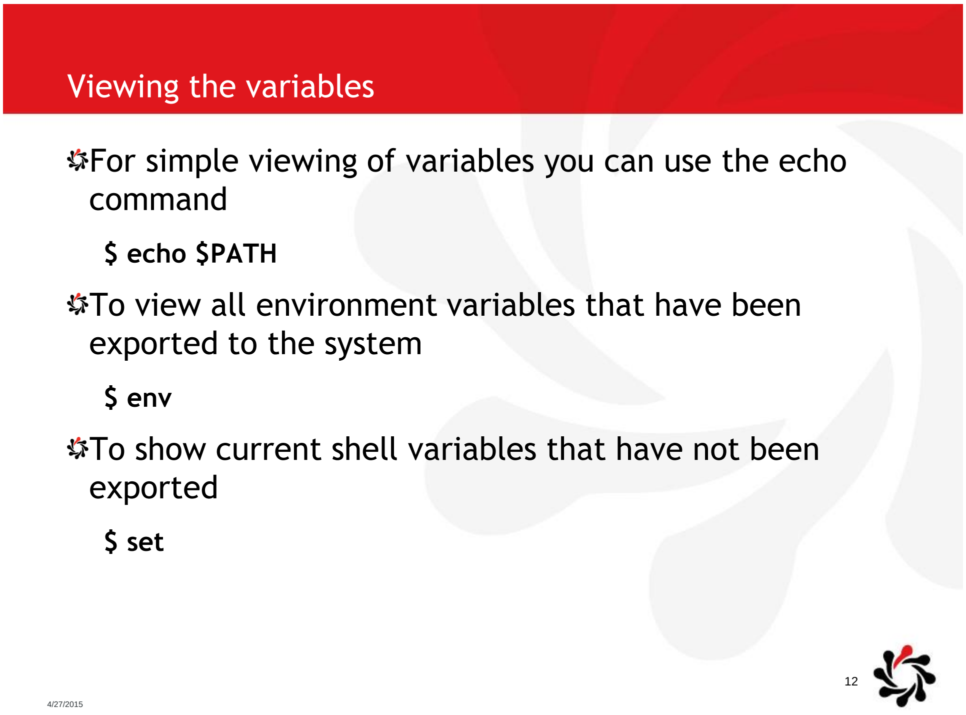
```
$ ./passwd
```



## Other environment variables

- Some important variables are as follows

| | |
|---|---|
| • HOME | Current users home dir |
| • USER | The logged in Username |
| • HISTSIZE | How much command history |
| • HISTFILE | The location of the history file |
| • SHELL | The users current Shell |
| • HOSTNAME | The system Hostname |
| • PS1 | The users PS1 shell prompt |
| • PS2 | The users PS2 shell prompt |
| • SHELLOPTS | List current shell options set |



## Viewing the variables

- For simple viewing of variables you can use the echo command

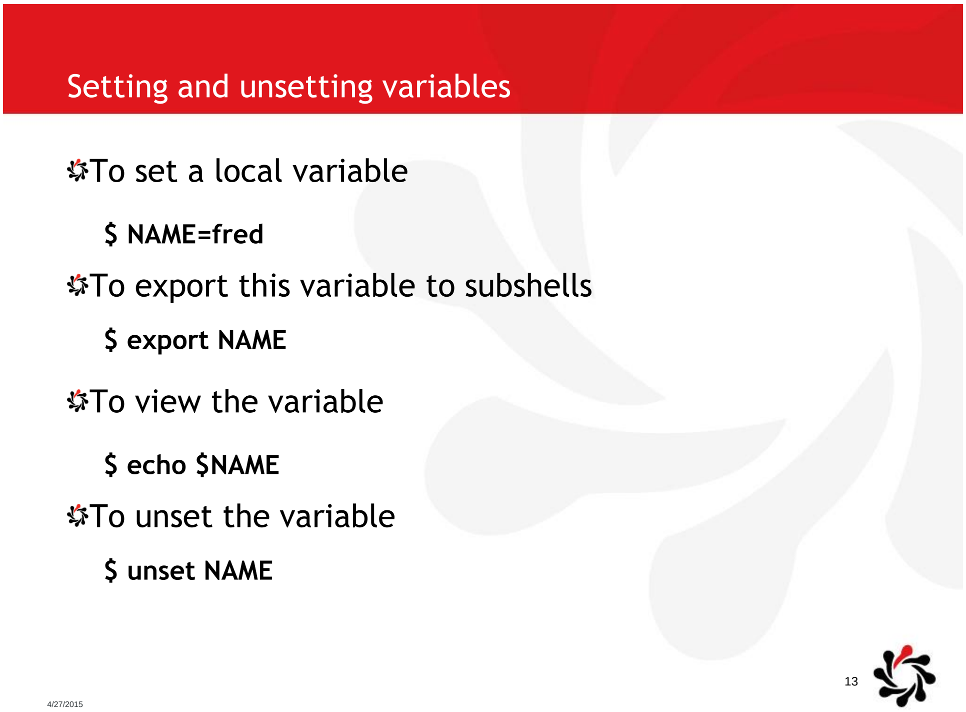
```
$ echo $PATH
```

- To view all environment variables that have been exported to the system

```
$ env
```

- To show current shell variables that have not been exported

```
$ set
```



## Setting and unsetting variables

- To set a local variable

```
$ NAME=fred
```

- To export this variable to subshells

```
$ export NAME
```

- To view the variable

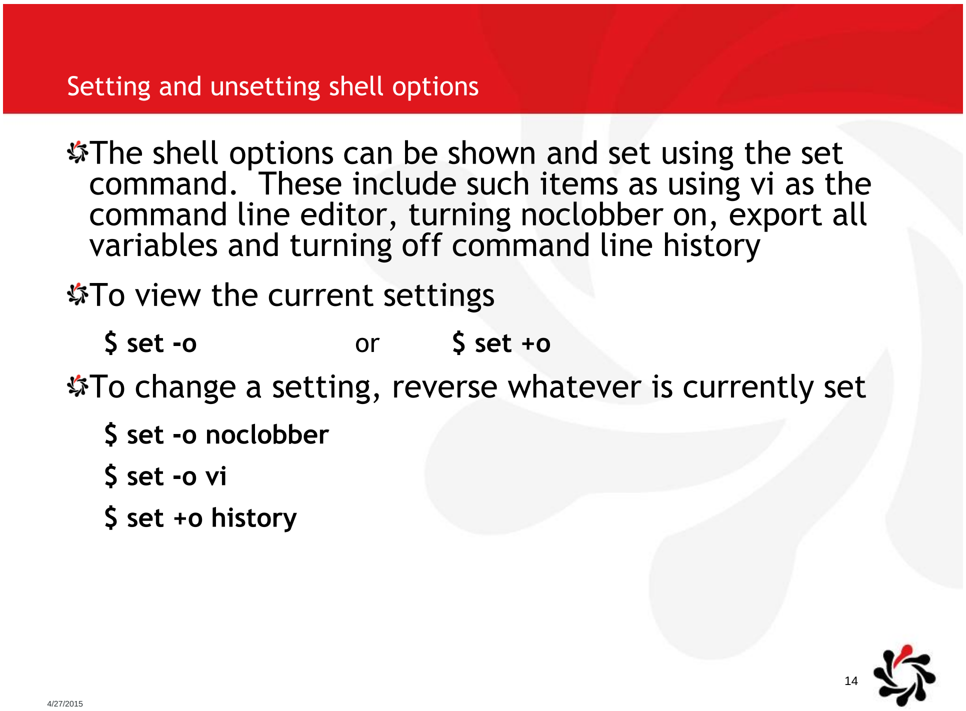
```
$ echo $NAME
```

- To unset the variable

```
$ unset NAME
```

13

4/27/2015



## Setting and unsetting shell options

- The shell options can be shown and set using the set command.  These include such items as using vi as the command line editor, turning noclobber on, export all variables and turning off command line history
- To view the current settings

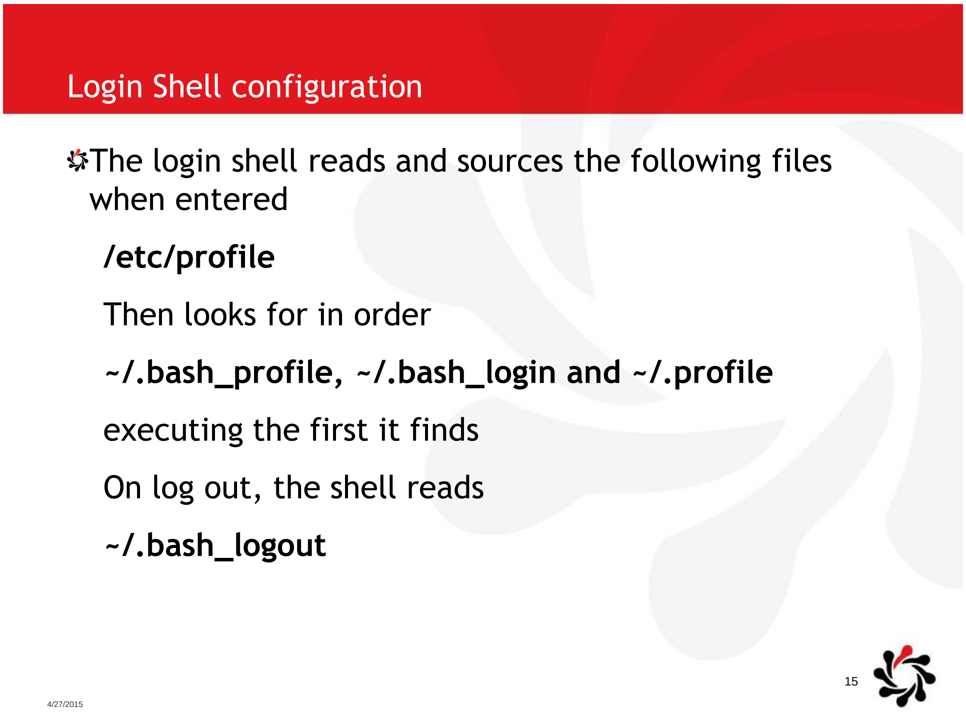
```
$ set -o                    or        $ set +o
```

- To change a setting, reverse whatever is currently set

```
$ set -o noclobber
$ set -o vi
$ set +o history
```

14

4/27/2015



## Login Shell configuration

- The login shell reads and sources the following files when entered

  **/etc/profile**

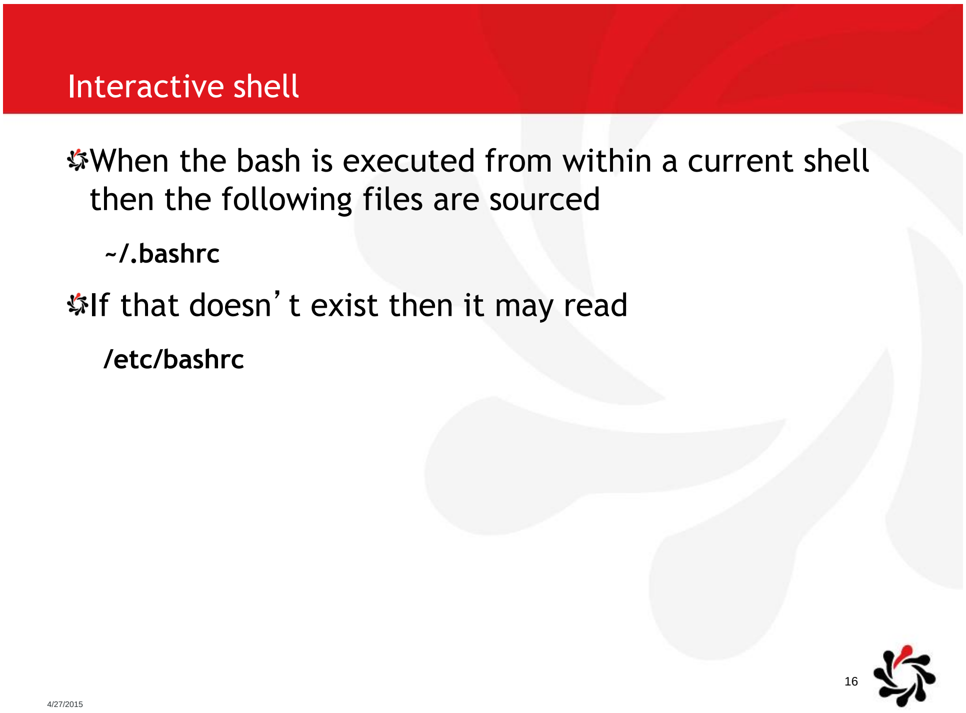
  Then looks for in order

  **~/.bash_profile, ~/.bash_login and ~/.profile**

  executing the first it finds

  On log out, the shell reads

  **~/.bash_logout**



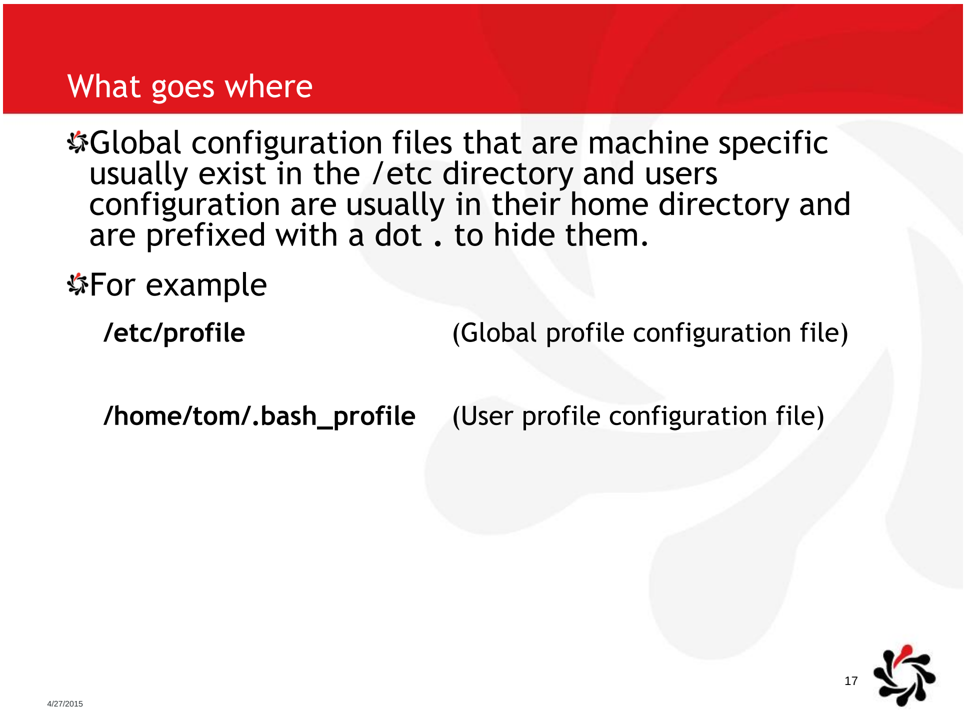
## Interactive shell

- When the bash is executed from within a current shell then the following files are sourced

  **~/.bashrc**

- If that doesn’t exist then it may read

  **/etc/bashrc**



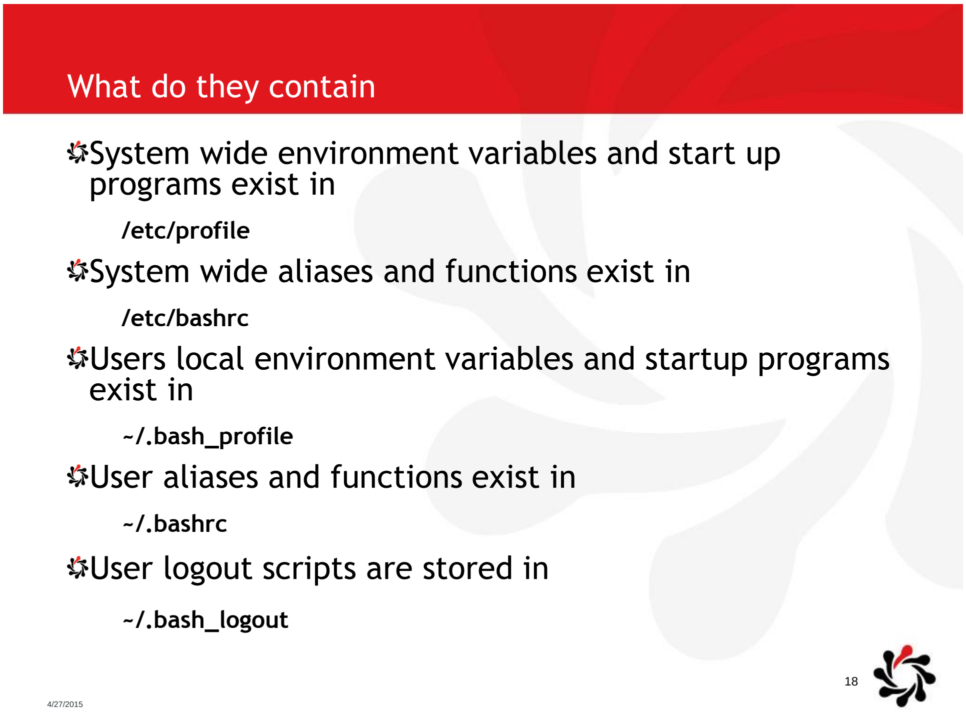
## What goes where

- Global configuration files that are machine specific usually exist in the /etc directory and users configuration are usually in their home directory and are prefixed with a dot **.** to hide them.
- For example

| | |
|---|---|
| **/etc/profile** | (Global profile configuration file) |
| **/home/tom/.bash_profile** | (User profile configuration file) |

4/27/2015

17



## What do they contain

- System wide environment variables and start up programs exist in

  **/etc/profile**

- System wide aliases and functions exist in

  **/etc/bashrc**

- Users local environment variables and startup programs exist in

  **~/.bash_profile**

- User aliases and functions exist in

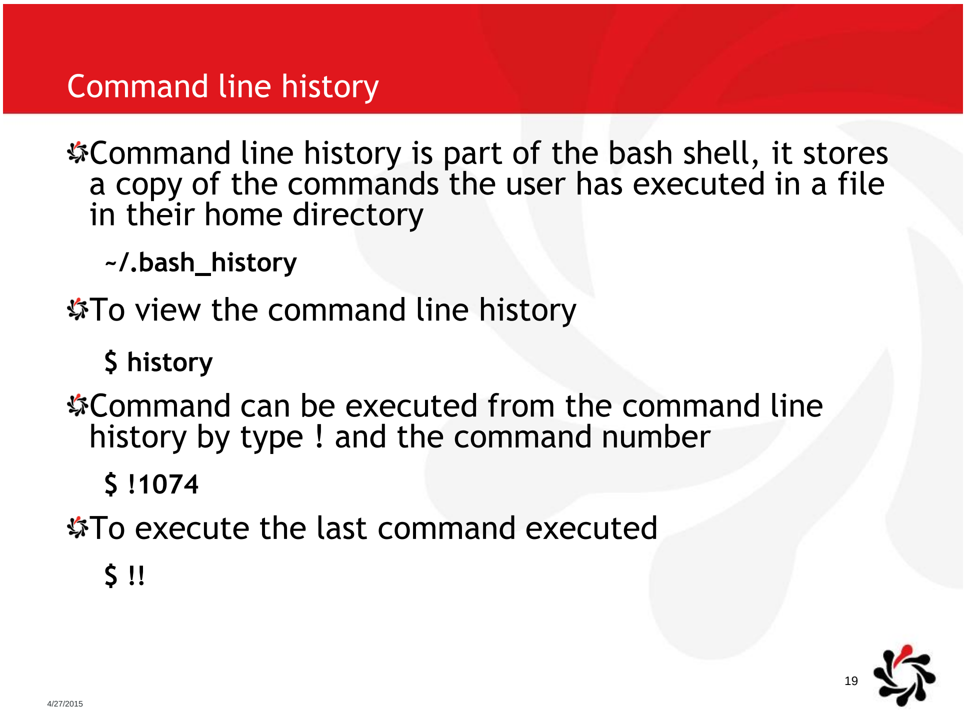
  **~/.bashrc**

- User logout scripts are stored in

  **~/.bash_logout**

4/27/2015

18



## Command line history

- Command line history is part of the bash shell, it stores a copy of the commands the user has executed in a file in their home directory

```
~/.bash_history
```

- To view the command line history

```
$ history
```

- Command can be executed from the command line history by type ! and the command number

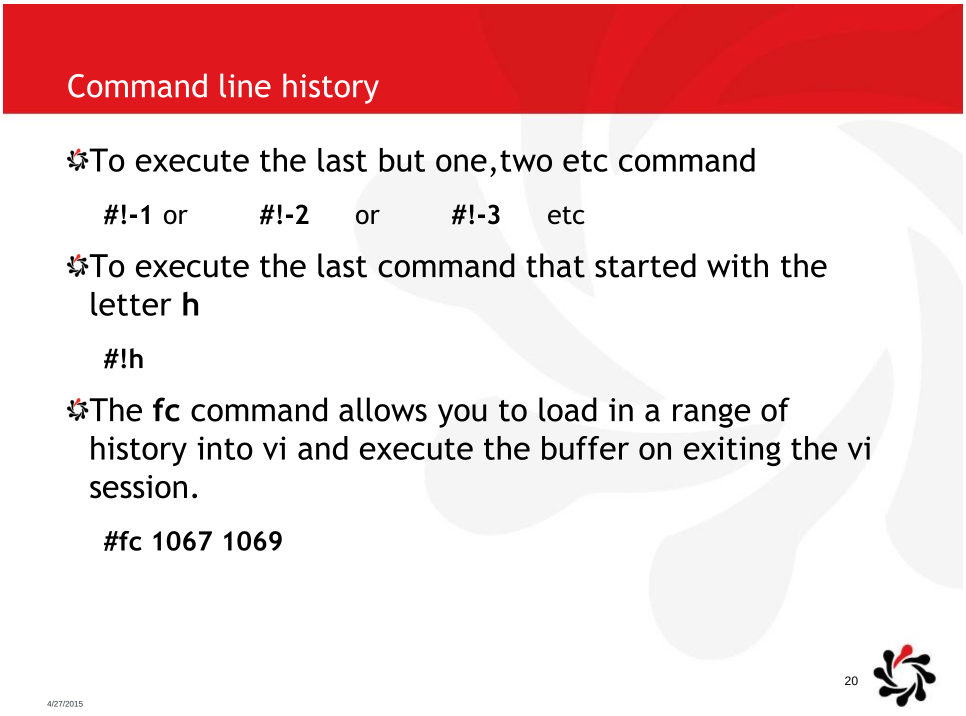
```
$ !1074
```

- To execute the last command executed

```
$ !!
```

19

4/27/2015



## Command line history

- To execute the last but one,two etc command

```
#!-1 or        #!-2     or        #!-3      etc
```

- To execute the last command that started with the letter **h**

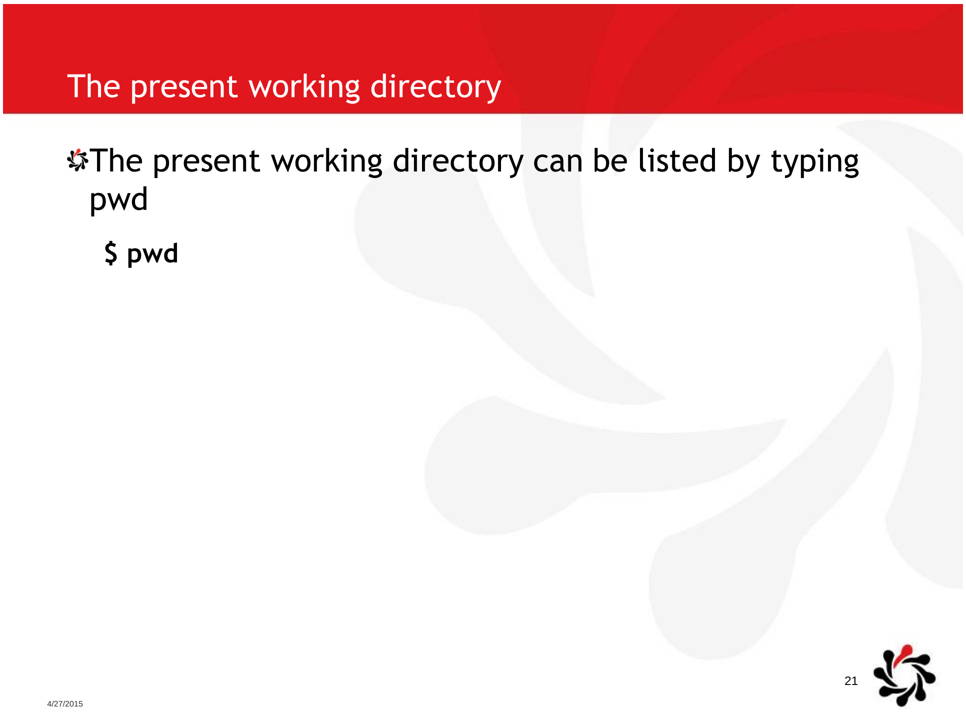
```
#!h
```

- The **fc** command allows you to load in a range of history into vi and execute the buffer on exiting the vi session.

```
#fc 1067 1069
```

20

4/27/2015



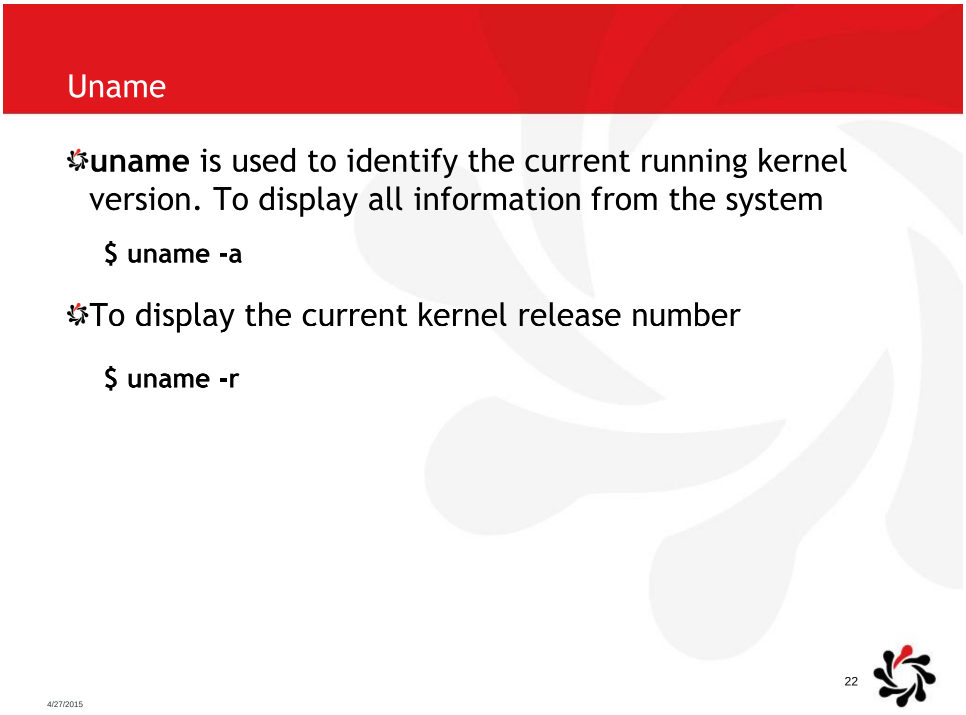
## The present working directory

- The present working directory can be listed by typing pwd

```
$ pwd
```

4/27/2015

21



## Uname

- **uname** is used to identify the current running kernel version. To display all information from the system

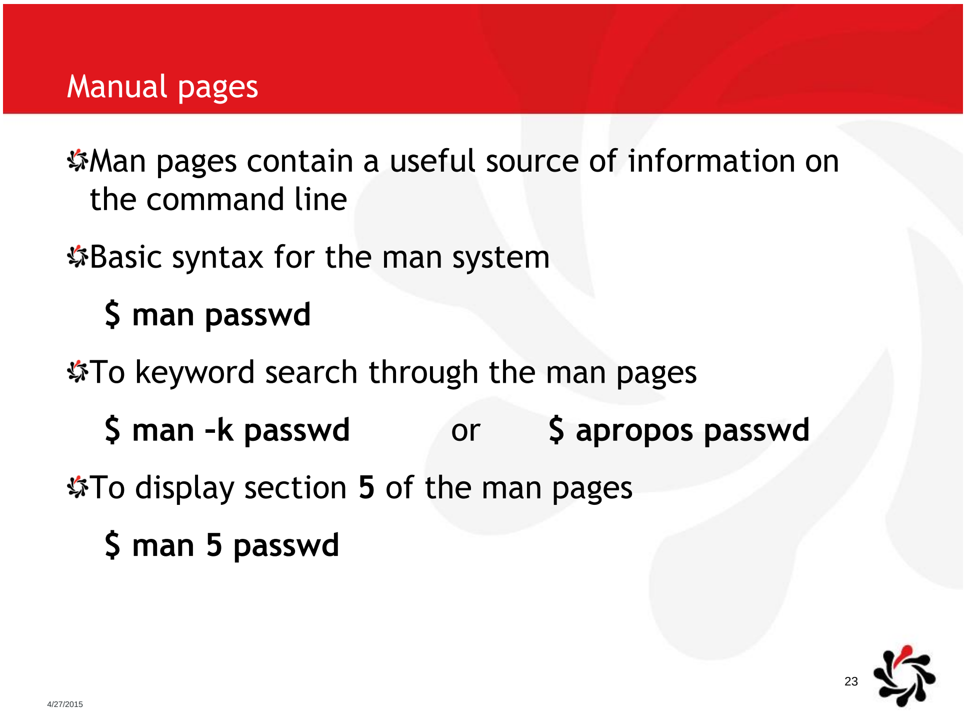
```
$ uname -a
```

- To display the current kernel release number

```
$ uname -r
```

4/27/2015

22



## Manual pages

- Man pages contain a useful source of information on the command line
- Basic syntax for the man system

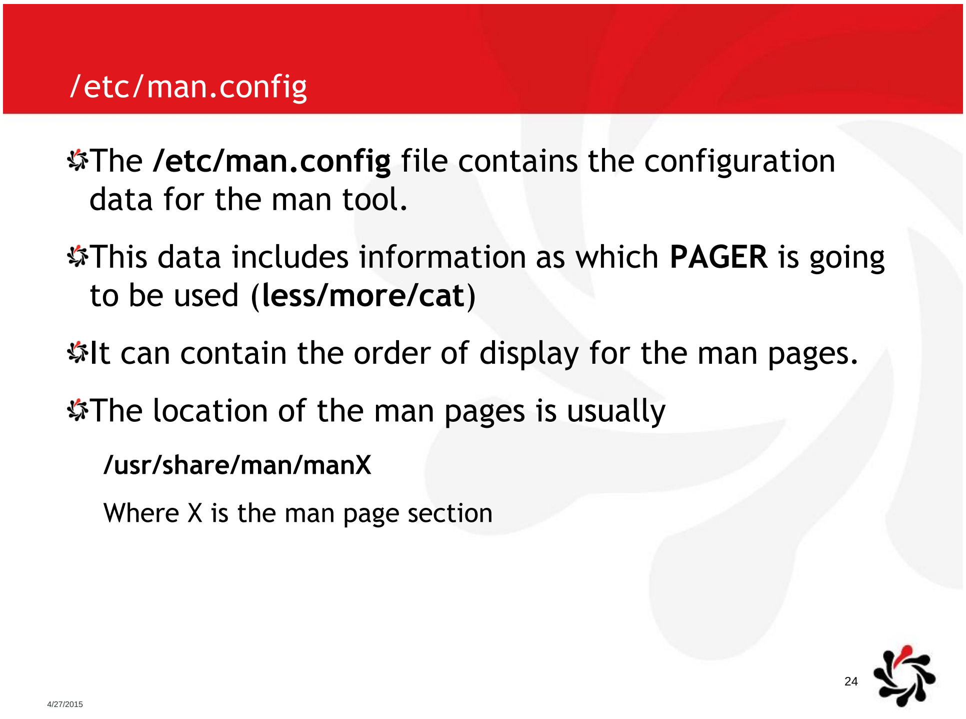
  **$ man passwd**
- To keyword search through the man pages

  **$ man -k passwd** or **$ apropos passwd**
- To display section **5** of the man pages

  **$ man 5 passwd**

4/27/2015

23



## /etc/man.config

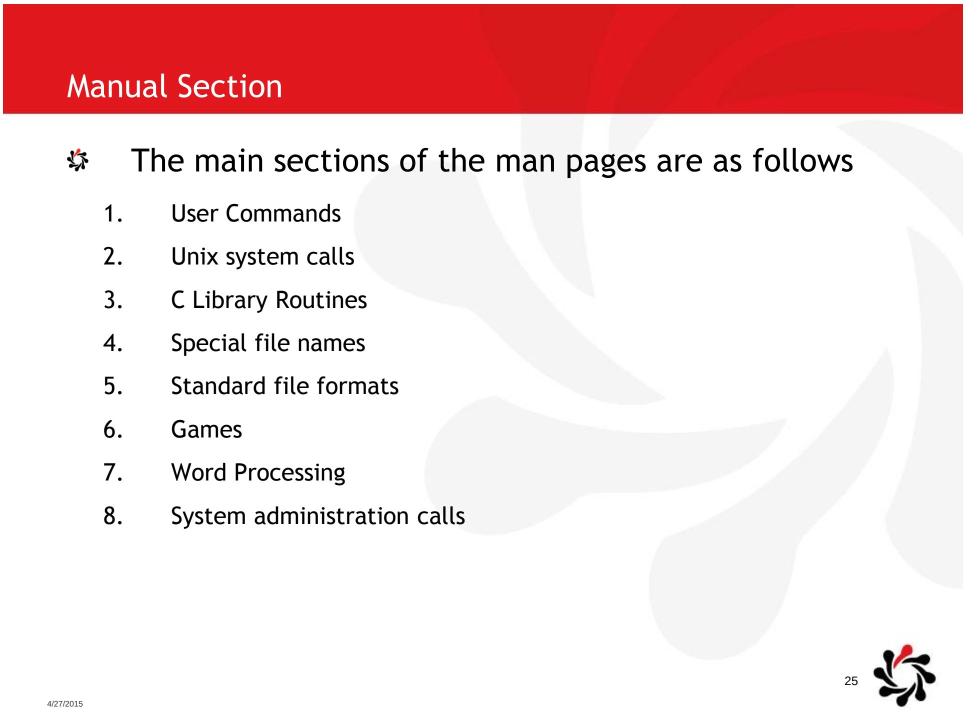
- The **/etc/man.config** file contains the configuration data for the man tool.
- This data includes information as which **PAGER** is going to be used (**less/more/cat**)
- It can contain the order of display for the man pages.
- The location of the man pages is usually

  **/usr/share/man/manX**

  Where X is the man page section

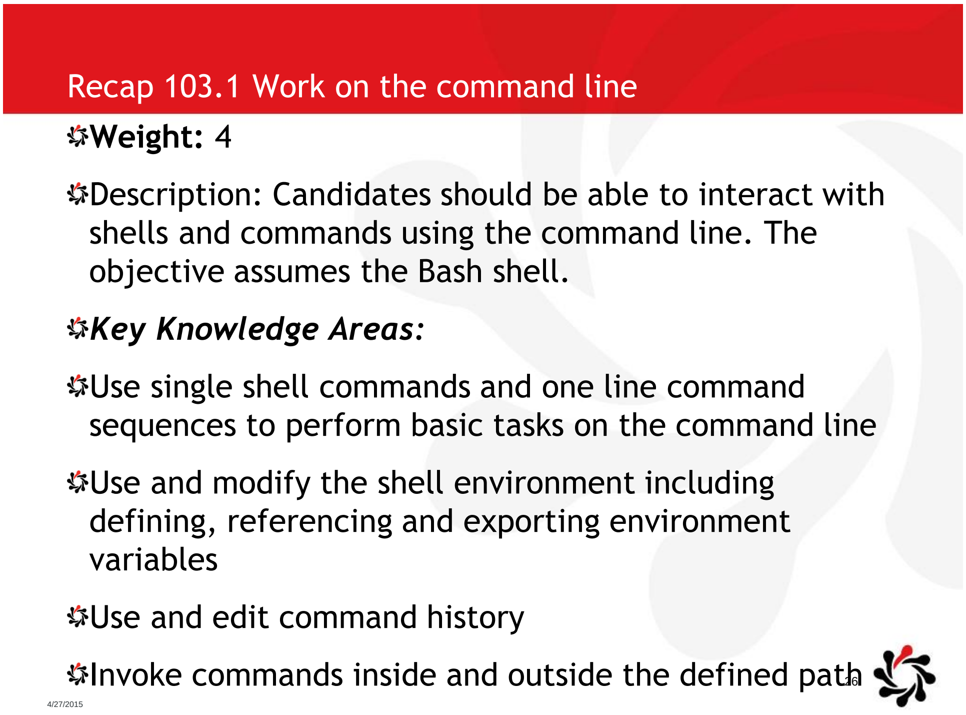
4/27/2015

24



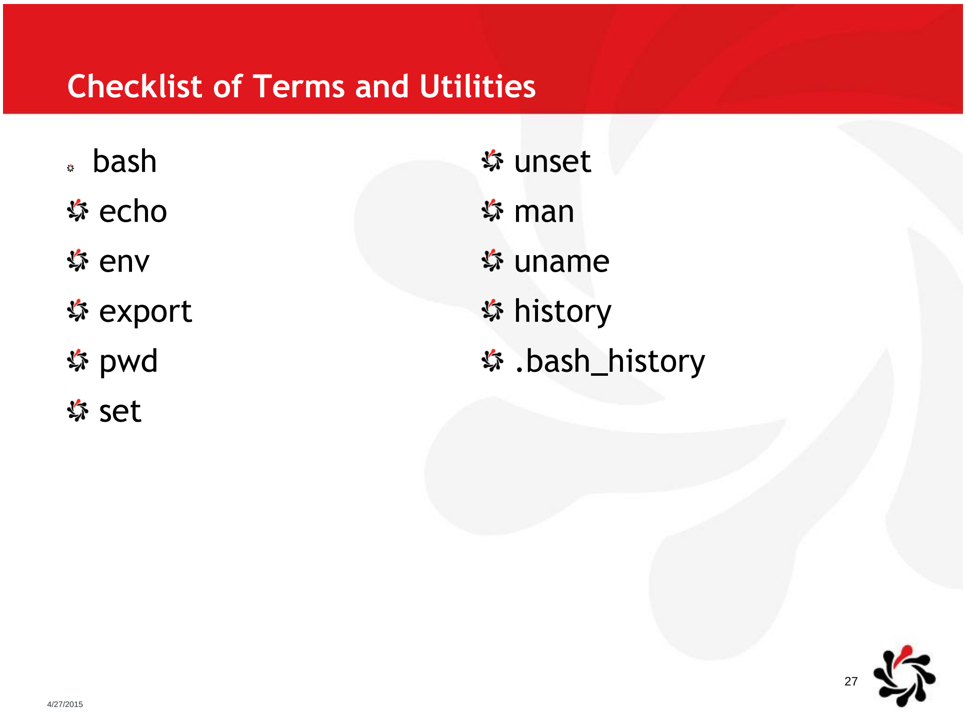
## Manual Section

- The main sections of the man pages are as follows
  1. User Commands
  2. Unix system calls
  3. C Library Routines
  4. Special file names
  5. Standard file formats
  6. Games
  7. Word Processing
  8. System administration calls



## Recap 103.1 Work on the command line

- **Weight:** 4
- Description: Candidates should be able to interact with shells and commands using the command line. The objective assumes the Bash shell.
- ***Key Knowledge Areas:***
- Use single shell commands and one line command sequences to perform basic tasks on the command line
- Use and modify the shell environment including defining, referencing and exporting environment variables
- Use and edit command history
- Invoke commands inside and outside the defined path

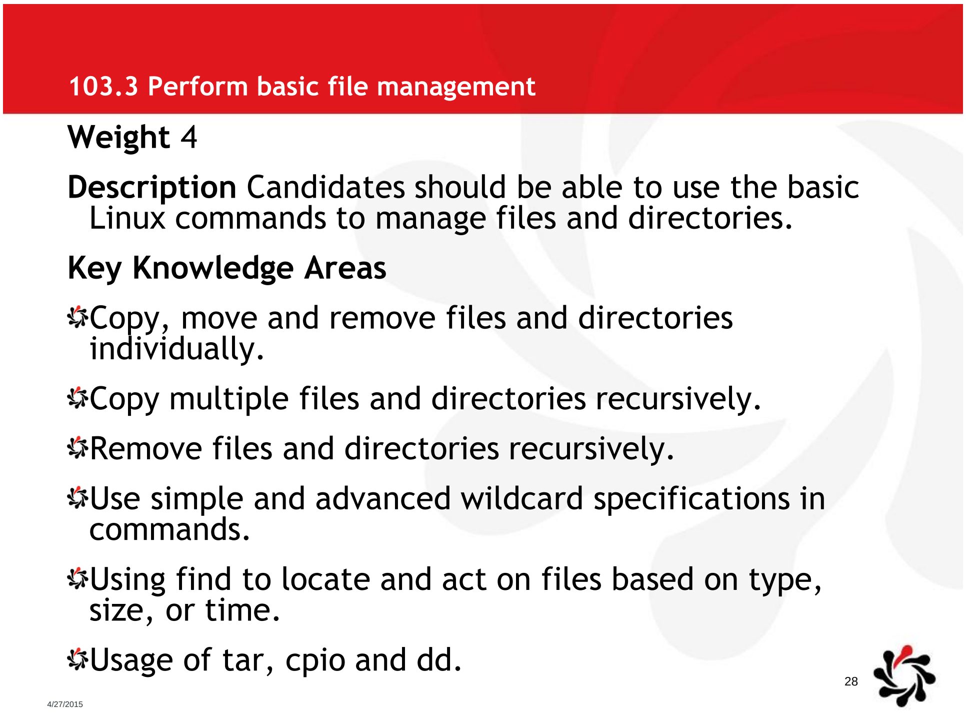


## Checklist of Terms and Utilities

- bash
- echo
- env
- export
- pwd
- set
- unset
- man
- uname
- history
- .bash_history

4/27/2015

27



## 103.3 Perform basic file management

**Weight** 4

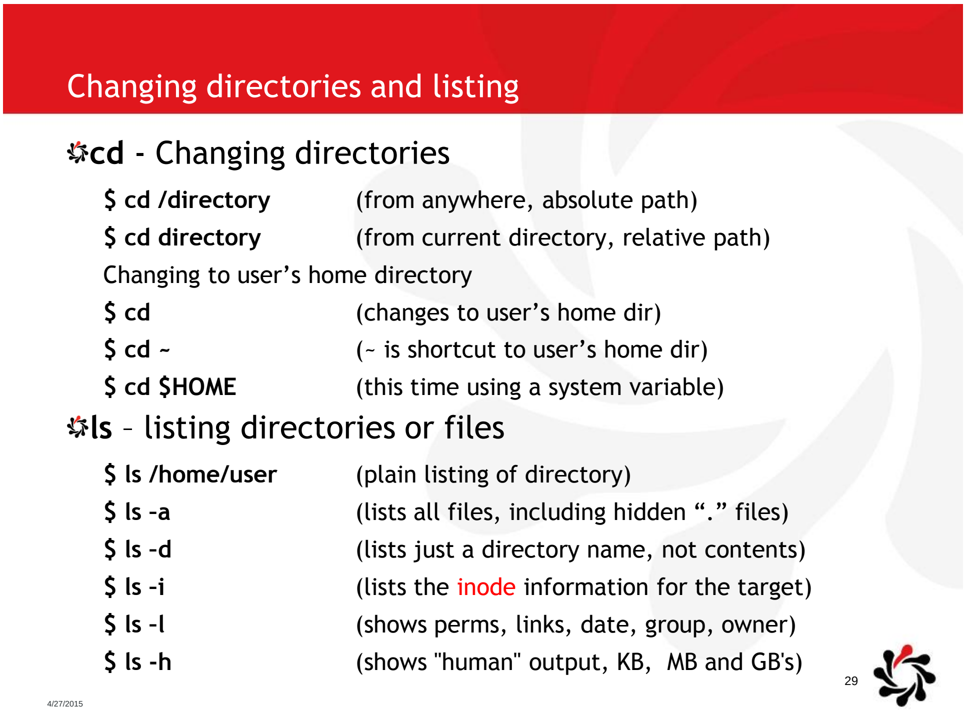
**Description** Candidates should be able to use the basic Linux commands to manage files and directories.

**Key Knowledge Areas**

- Copy, move and remove files and directories individually.
- Copy multiple files and directories recursively.
- Remove files and directories recursively.
- Use simple and advanced wildcard specifications in commands.
- Using find to locate and act on files based on type, size, or time.
- Usage of tar, cpio and dd.

4/27/2015

28



## Changing directories and listing

### **cd** - Changing directories

| | |
|---|---|
| **$ cd /directory** | (from anywhere, absolute path) |
| **$ cd directory** | (from current directory, relative path) |

Changing to user's home directory

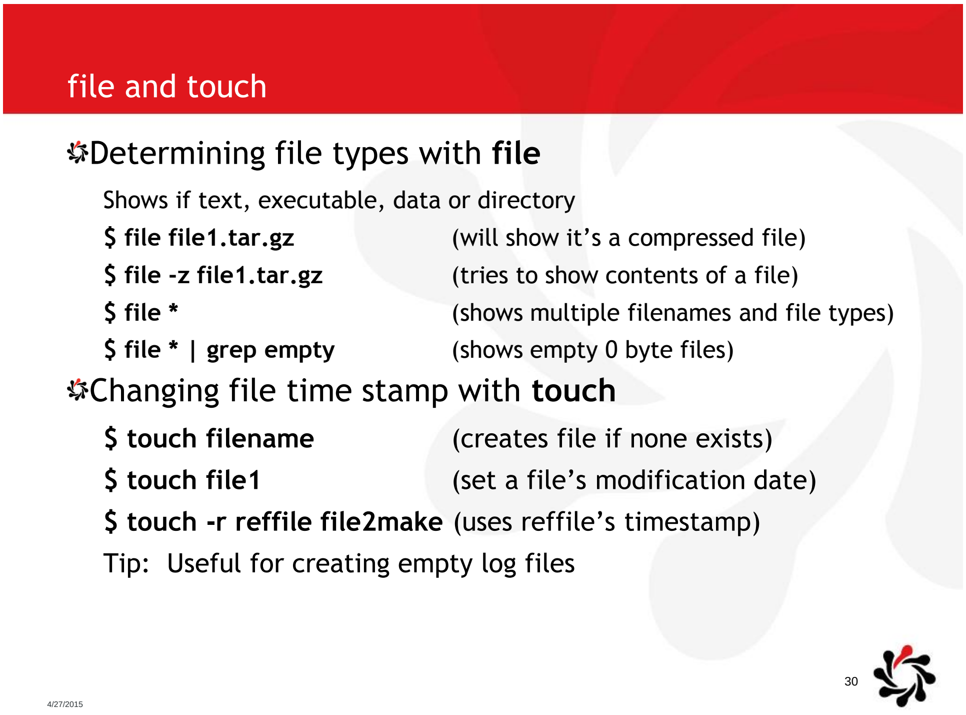
| | |
|---|---|
| $ cd | (changes to user's home dir) |
| $ cd ~ | (~ is shortcut to user's home dir) |
| $ cd $HOME | (this time using a system variable) |

### **ls** – listing directories or files

| | |
|---|---|
| $ ls /home/user | (plain listing of directory) |
| $ ls -a | (lists all files, including hidden "." files) |
| $ ls -d | (lists just a directory name, not contents) |
| $ ls -i | (lists the inode information for the target) |
| $ ls -l | (shows perms, links, date, group, owner) |
| $ ls -h | (shows "human" output, KB,  MB and GB's) |

4/27/2015

29



## file and touch

### Determining file types with **file**

Shows if text, executable, data or directory

| | |
|---|---|
| **$ file file1.tar.gz** | (will show it's a compressed file) |
| **$ file -z file1.tar.gz** | (tries to show contents of a file) |
| **$ file *** | (shows multiple filenames and file types) |
| **$ file * \| grep empty** | (shows empty 0 byte files) |

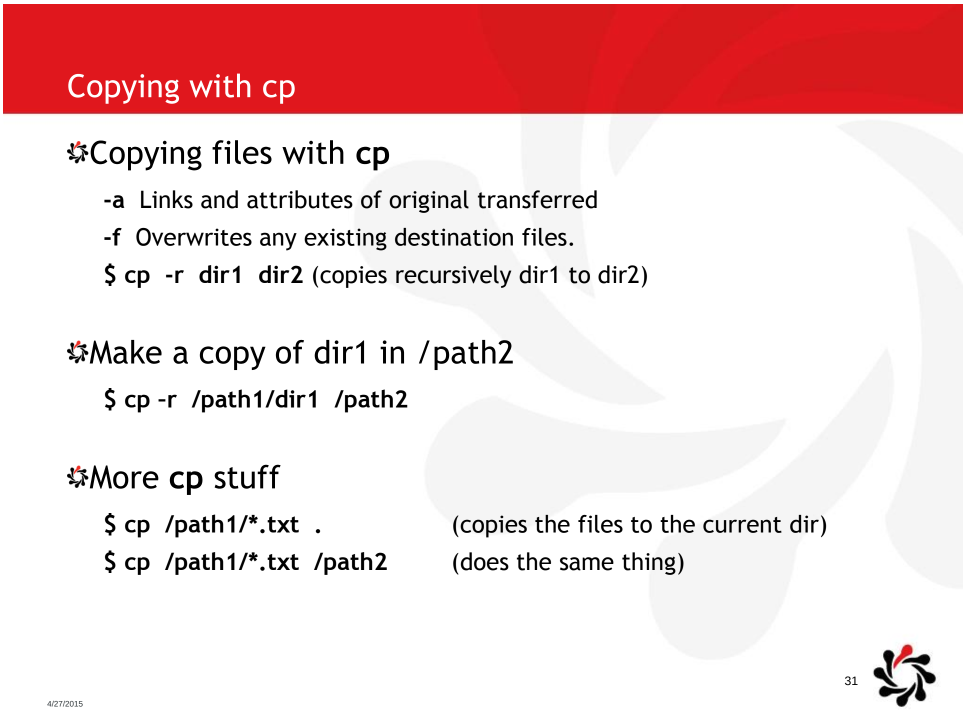
### Changing file time stamp with **touch**

| | |
|---|---|
| **$ touch filename** | (creates file if none exists) |
| **$ touch file1** | (set a file's modification date) |
| **$ touch -r reffile file2make** | (uses reffile's timestamp) |

Tip:  Useful for creating empty log files

4/27/2015

30



## Copying with cp

- Copying files with **cp**

  **-a** Links and attributes of original transferred

  **-f** Overwrites any existing destination files.

  **$ cp -r dir1 dir2** (copies recursively dir1 to dir2)

- Make a copy of dir1 in /path2

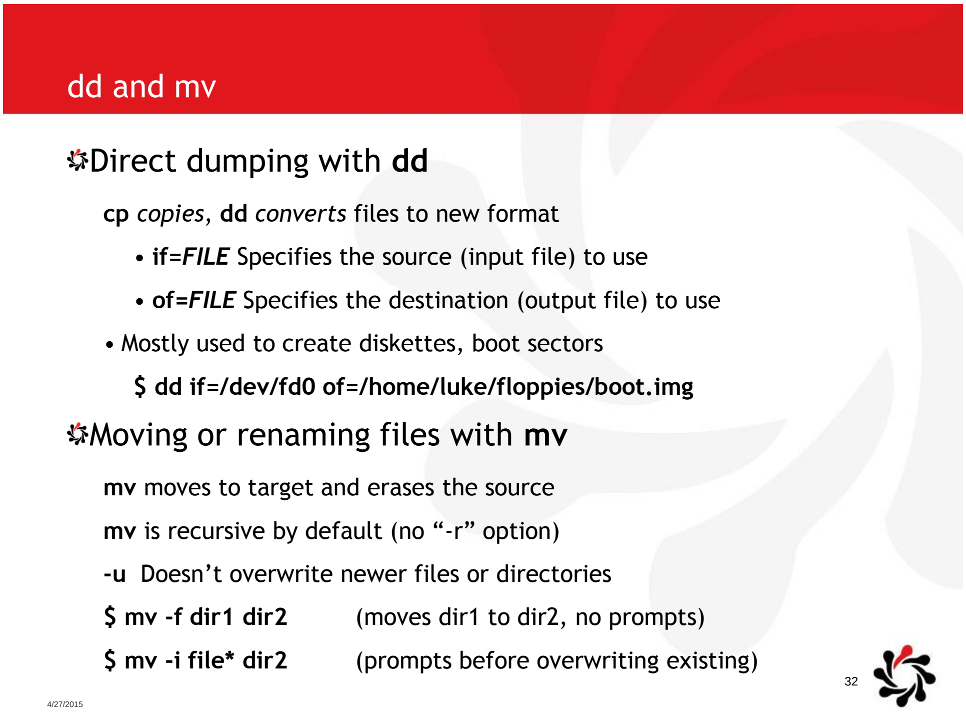
  ```
  $ cp -r  /path1/dir1  /path2
  ```

- More **cp** stuff

  ```
  $ cp  /path1/*.txt  .            (copies the files to the current dir)
  $ cp  /path1/*.txt  /path2       (does the same thing)
  ```

4/27/2015

31



## dd and mv

- Direct dumping with **dd**

  **cp** *copies*, **dd** *converts* files to new format

  - **if=*FILE*** Specifies the source (input file) to use
  - **of=*FILE*** Specifies the destination (output file) to use

  - Mostly used to create diskettes, boot sectors

    ```
    $ dd if=/dev/fd0 of=/home/luke/floppies/boot.img
    ```

- Moving or renaming files with **mv**

  **mv** moves to target and erases the source

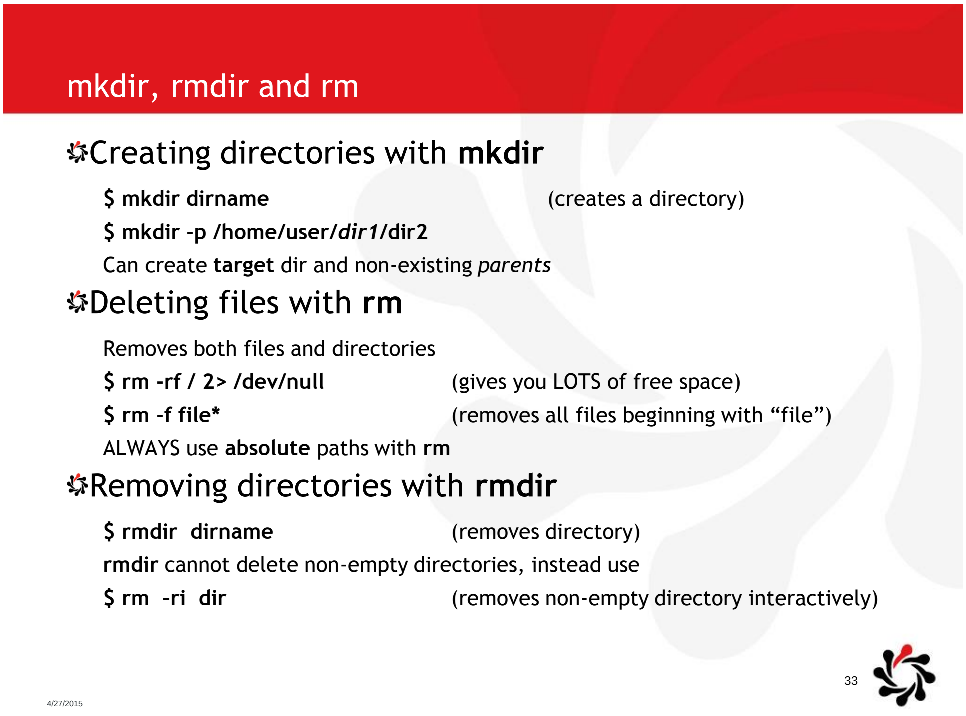
  **mv** is recursive by default (no “-r” option)

  **-u** Doesn’t overwrite newer files or directories

  ```
  $ mv -f dir1 dir2        (moves dir1 to dir2, no prompts)
  $ mv -i file* dir2       (prompts before overwriting existing)
  ```

4/27/2015

32



## mkdir, rmdir and rm

- Creating directories with **mkdir**

```
$ mkdir dirname                        (creates a directory)
$ mkdir -p /home/user/dir1/dir2
```

Can create **target** dir and non-existing *parents*

- Deleting files with **rm**

Removes both files and directories

```
$ rm -rf / 2> /dev/null        (gives you LOTS of free space)
$ rm -f file*                  (removes all files beginning with “file”)
```

ALWAYS use **absolute** paths with **rm**

- Removing directories with **rmdir**

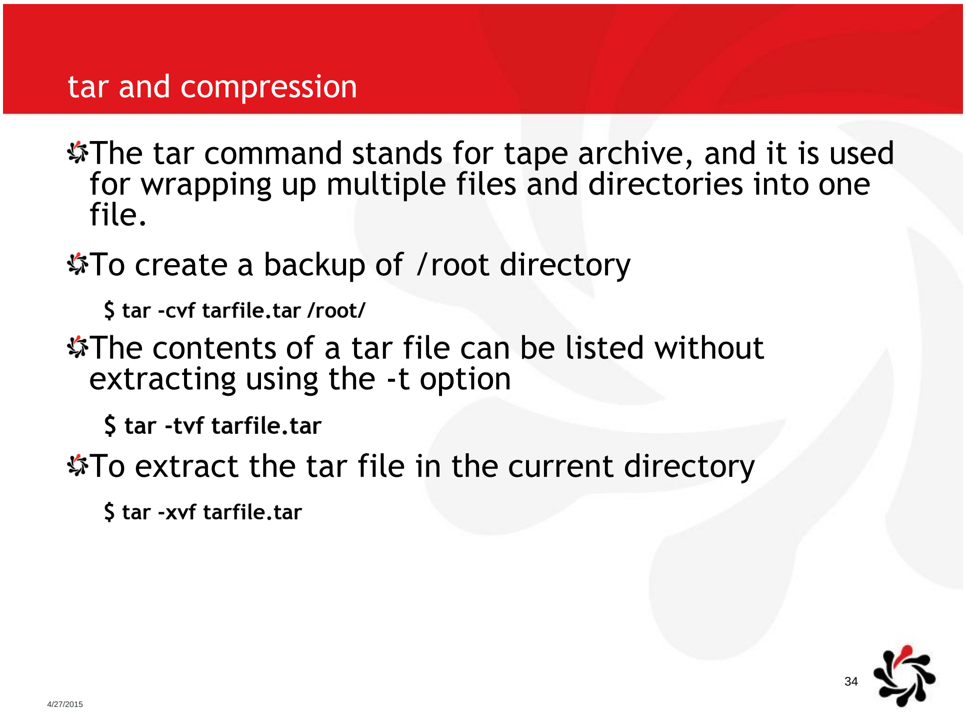
```
$ rmdir  dirname               (removes directory)
```

**rmdir** cannot delete non-empty directories, instead use

```
$ rm  -ri  dir                 (removes non-empty directory interactively)
```

33

4/27/2015



## tar and compression

- The tar command stands for tape archive, and it is used for wrapping up multiple files and directories into one file.
- To create a backup of /root directory

```
$ tar -cvf tarfile.tar /root/
```

- The contents of a tar file can be listed without extracting using the -t option

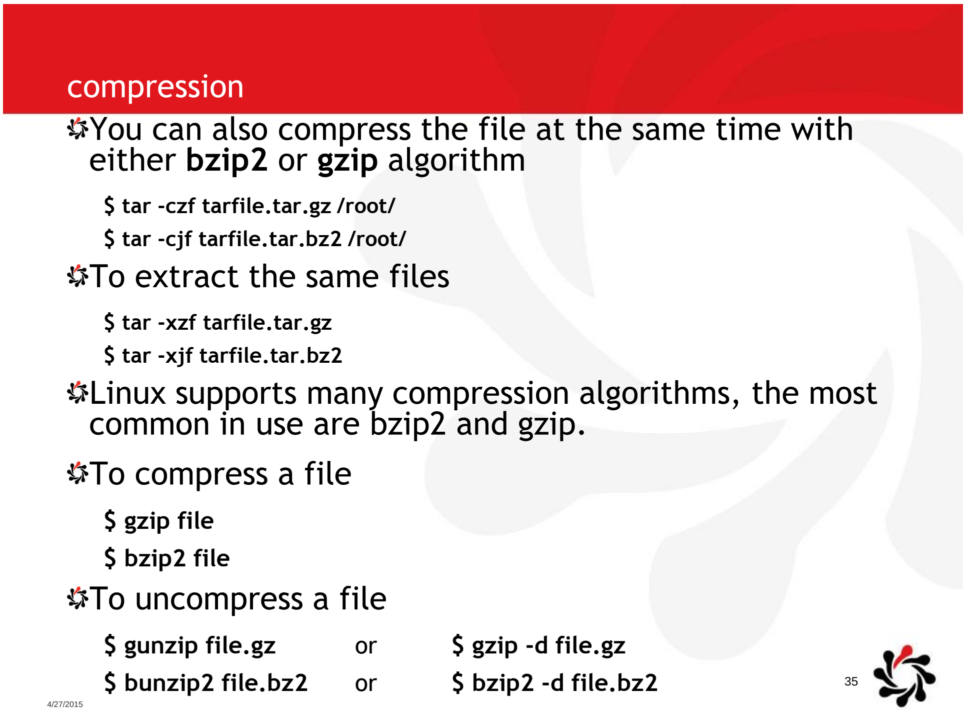
```
$ tar -tvf tarfile.tar
```

- To extract the tar file in the current directory

```
$ tar -xvf tarfile.tar
```

34

4/27/2015



## compression

- You can also compress the file at the same time with either **bzip2** or **gzip** algorithm

```
$ tar -czf tarfile.tar.gz /root/
$ tar -cjf tarfile.tar.bz2 /root/
```

- To extract the same files

```
$ tar -xzf tarfile.tar.gz
$ tar -xjf tarfile.tar.bz2
```

- Linux supports many compression algorithms, the most common in use are bzip2 and gzip.
- To compress a file

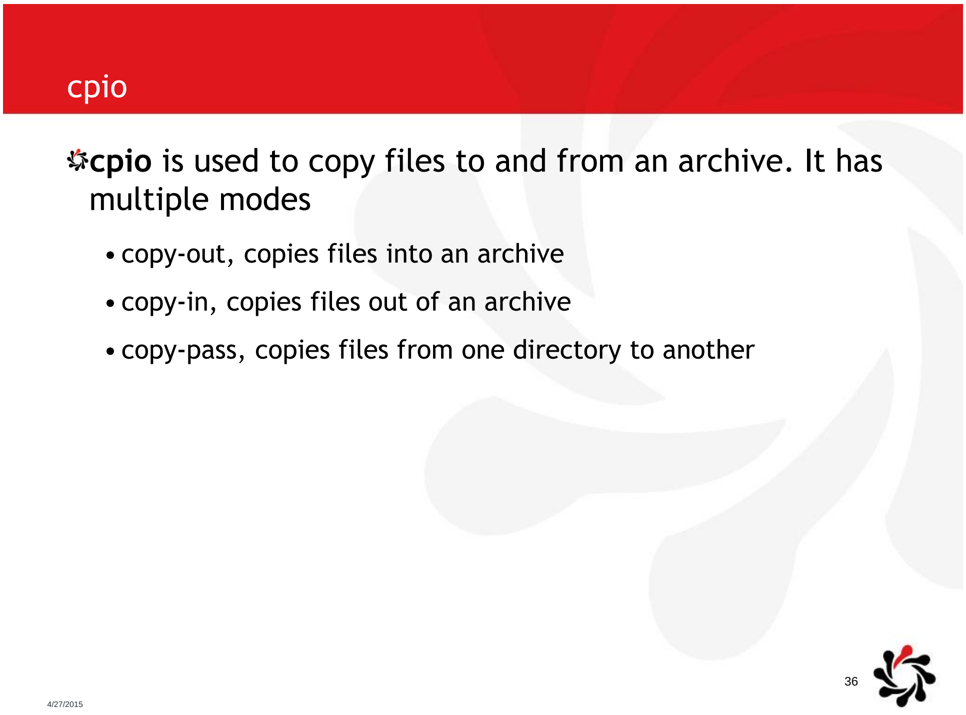
```
$ gzip file
$ bzip2 file
```

- To uncompress a file

```
$ gunzip file.gz       or      $ gzip -d file.gz
$ bunzip2 file.bz2     or      $ bzip2 -d file.bz2
```

35

4/27/2015

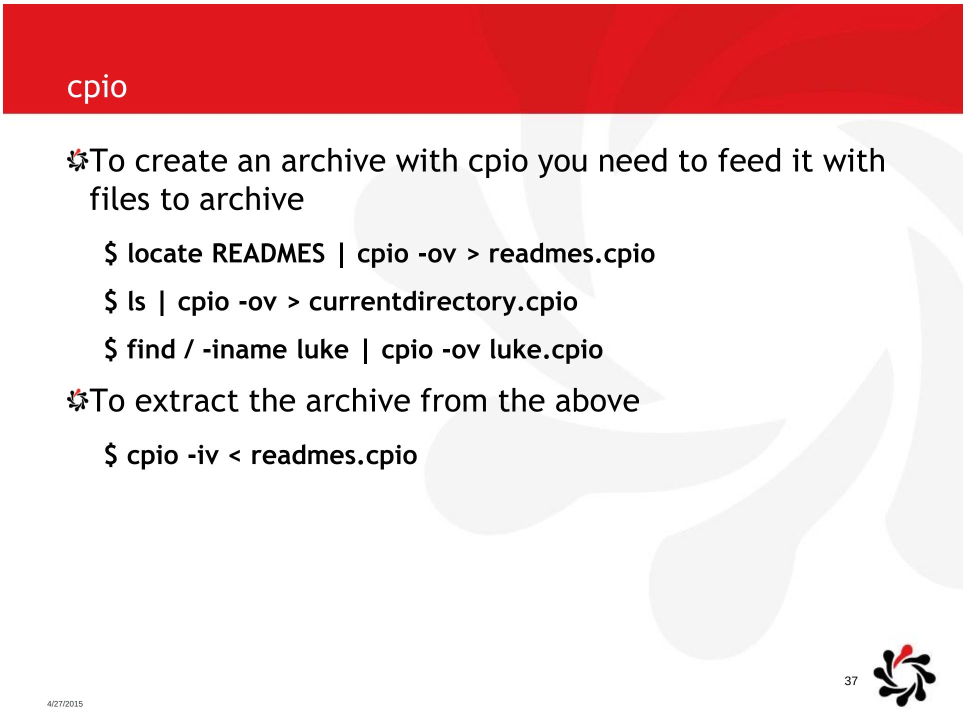


## cpio

- **cpio** is used to copy files to and from an archive. It has multiple modes
  - copy-out, copies files into an archive
  - copy-in, copies files out of an archive
  - copy-pass, copies files from one directory to another

36

4/27/2015



## cpio

- To create an archive with cpio you need to feed it with files to archive

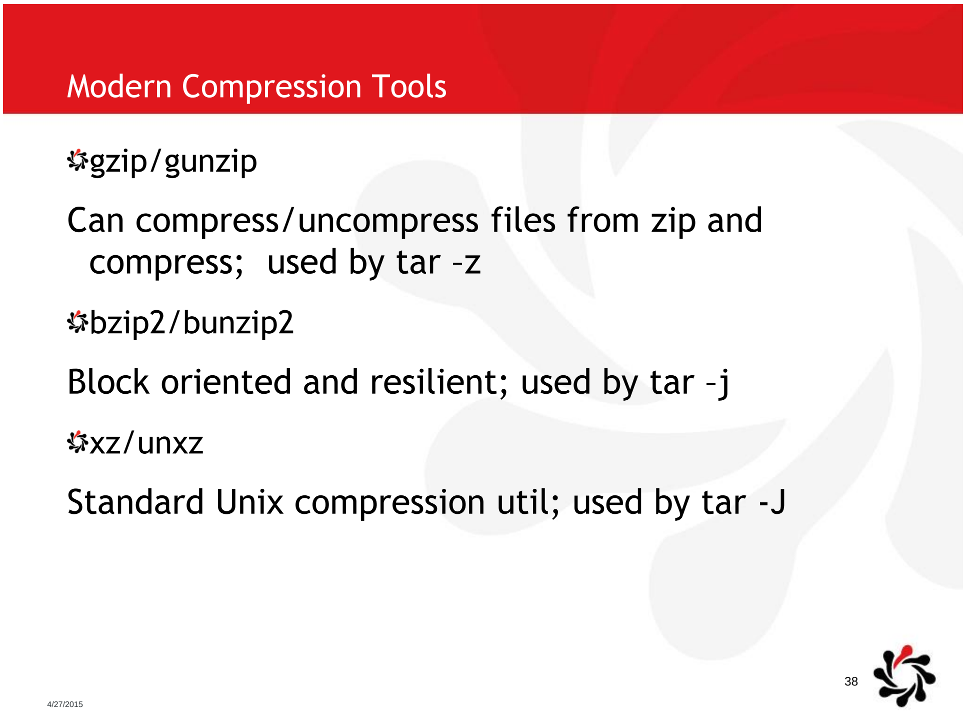
```
$ locate READMES | cpio -ov > readmes.cpio
$ ls | cpio -ov > currentdirectory.cpio
$ find / -iname luke | cpio -ov luke.cpio
```

- To extract the archive from the above

```
$ cpio -iv < readmes.cpio
```

4/27/2015

37



## Modern Compression Tools

- gzip/gunzip

Can compress/uncompress files from zip and compress;  used by tar –z

- bzip2/bunzip2

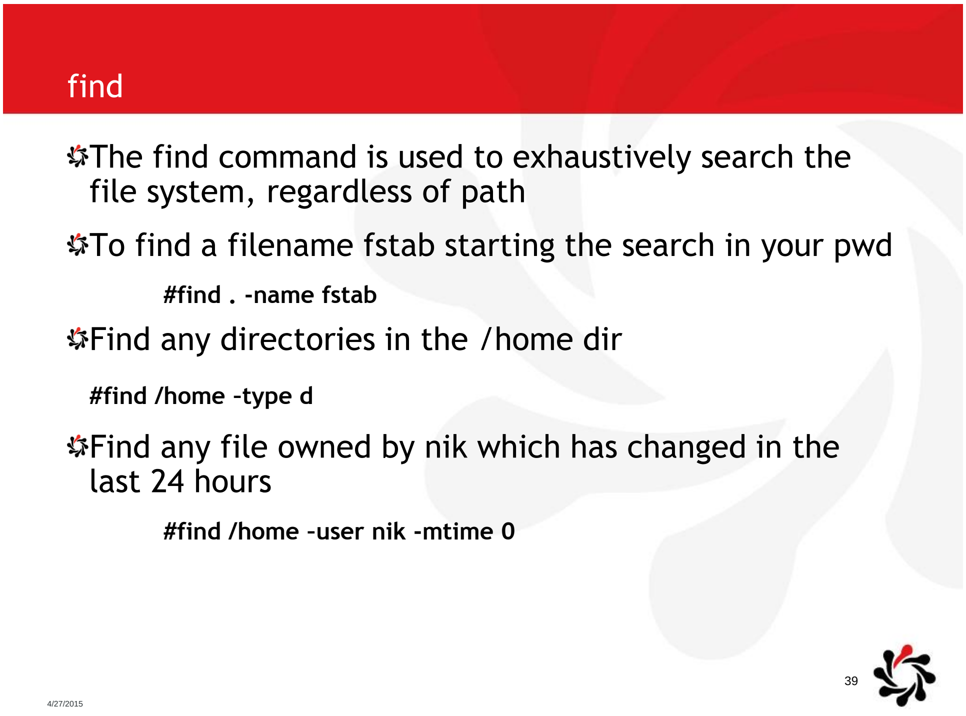
Block oriented and resilient; used by tar -j

- xz/unxz

Standard Unix compression util; used by tar -J

4/27/2015

38



## find

- The find command is used to exhaustively search the file system, regardless of path
- To find a filename fstab starting the search in your pwd

```
#find . -name fstab
```

- Find any directories in the /home dir

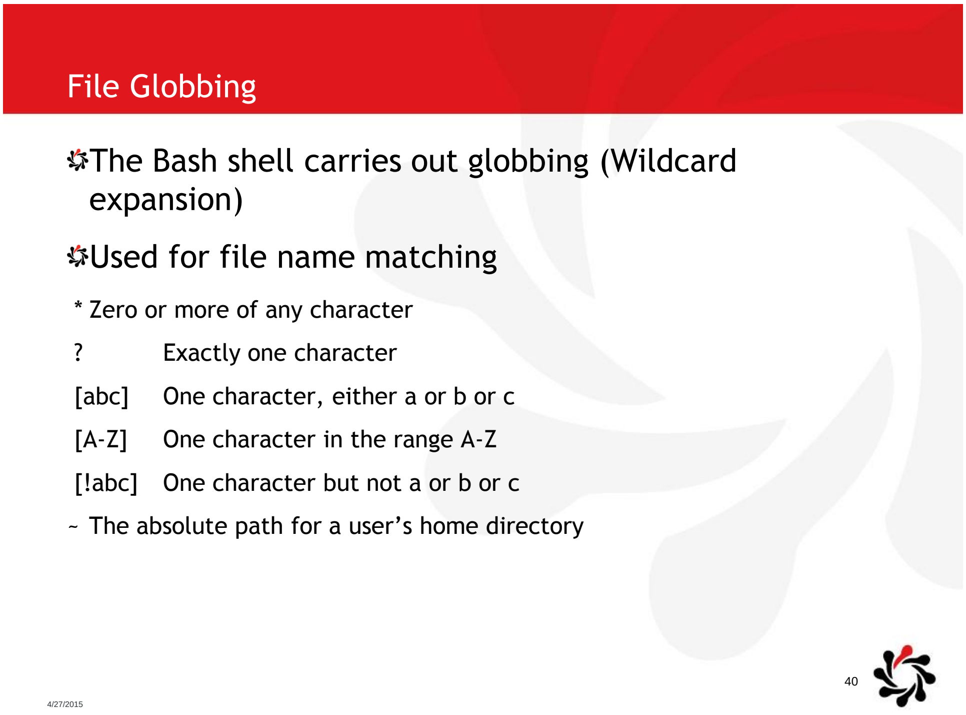
```
#find /home -type d
```

- Find any file owned by nik which has changed in the last 24 hours

```
#find /home -user nik -mtime 0
```

4/27/2015

39



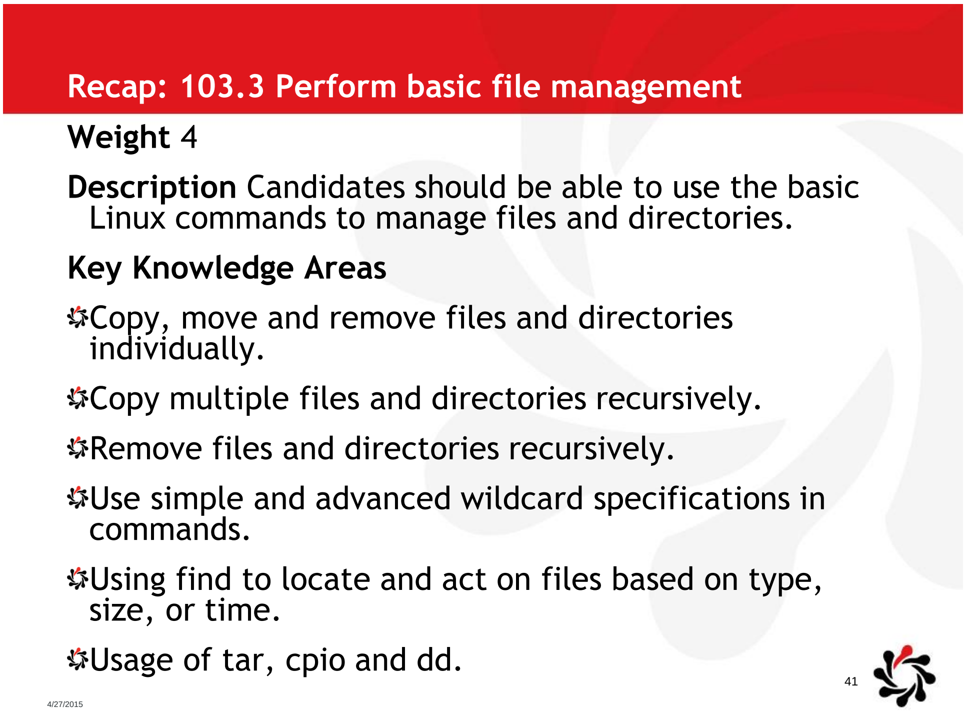
## File Globbing

- The Bash shell carries out globbing (Wildcard expansion)
- Used for file name matching

| | |
|---|---|
| * | Zero or more of any character |
| ? | Exactly one character |
| [abc] | One character, either a or b or c |
| [A-Z] | One character in the range A-Z |
| [!abc] | One character but not a or b or c |
| ~ | The absolute path for a user’s home directory |

4/27/2015

40



## Recap: 103.3 Perform basic file management

**Weight** 4

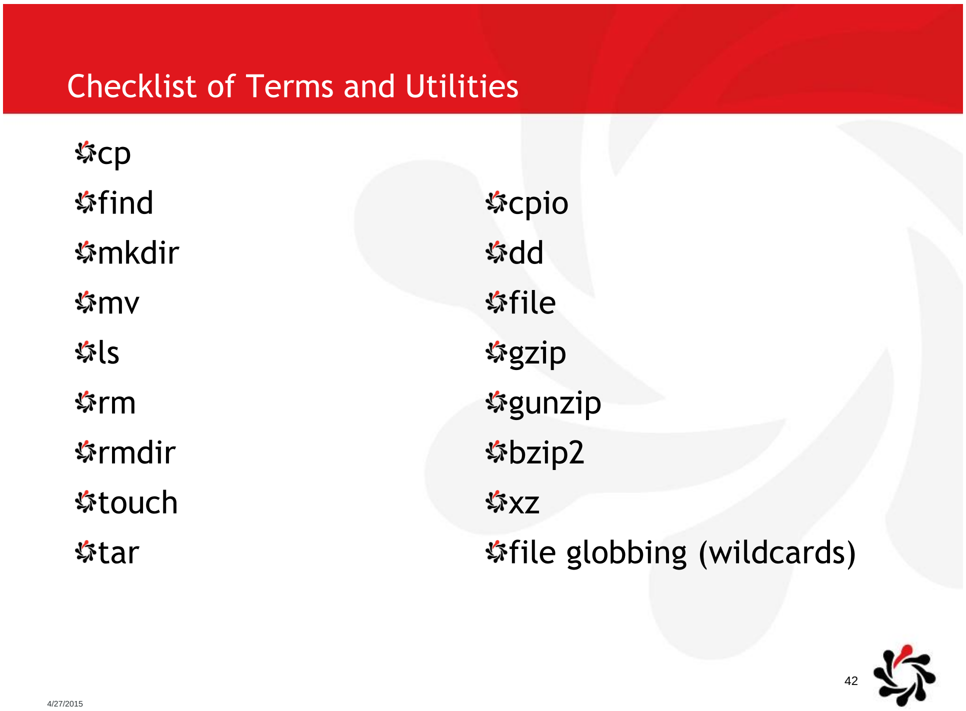
**Description** Candidates should be able to use the basic Linux commands to manage files and directories.

**Key Knowledge Areas**

- Copy, move and remove files and directories individually.
- Copy multiple files and directories recursively.
- Remove files and directories recursively.
- Use simple and advanced wildcard specifications in commands.
- Using find to locate and act on files based on type, size, or time.
- Usage of tar, cpio and dd.

4/27/2015

41

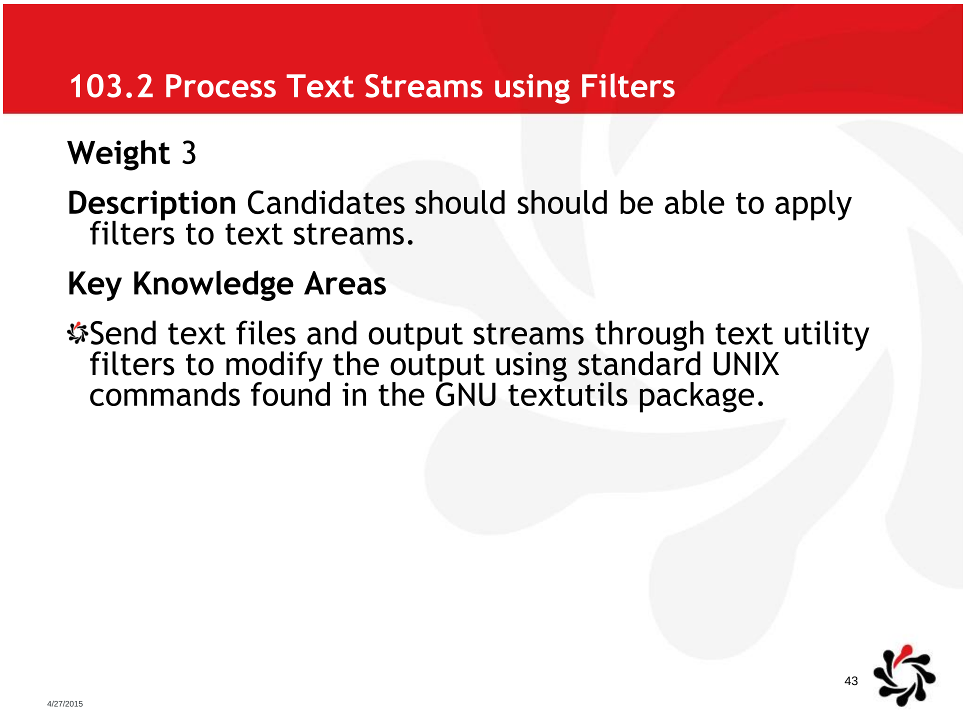


## Checklist of Terms and Utilities

- cp
- find
- mkdir
- mv
- ls
- rm
- rmdir
- touch
- tar
- cpio
- dd
- file
- gzip
- gunzip
- bzip2
- xz
- file globbing (wildcards)

4/27/2015

42



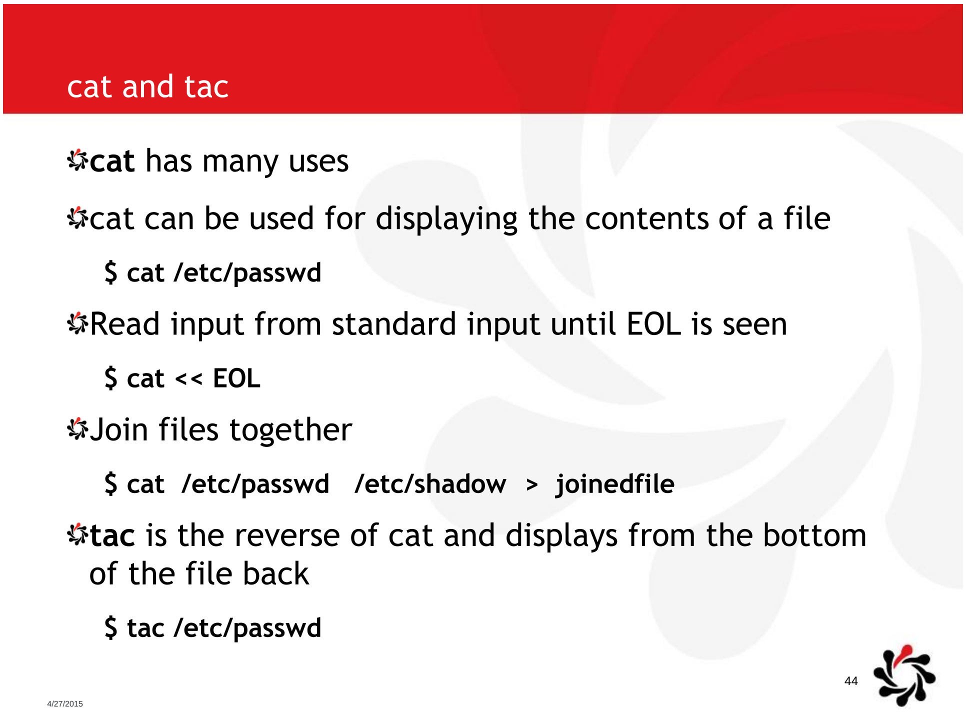
## 103.2 Process Text Streams using Filters

**Weight** 3

**Description** Candidates should should be able to apply filters to text streams.

**Key Knowledge Areas**

- Send text files and output streams through text utility filters to modify the output using standard UNIX commands found in the GNU textutils package.



## cat and tac

- **cat** has many uses
- cat can be used for displaying the contents of a file

```
$ cat /etc/passwd
```

- Read input from standard input until EOL is seen

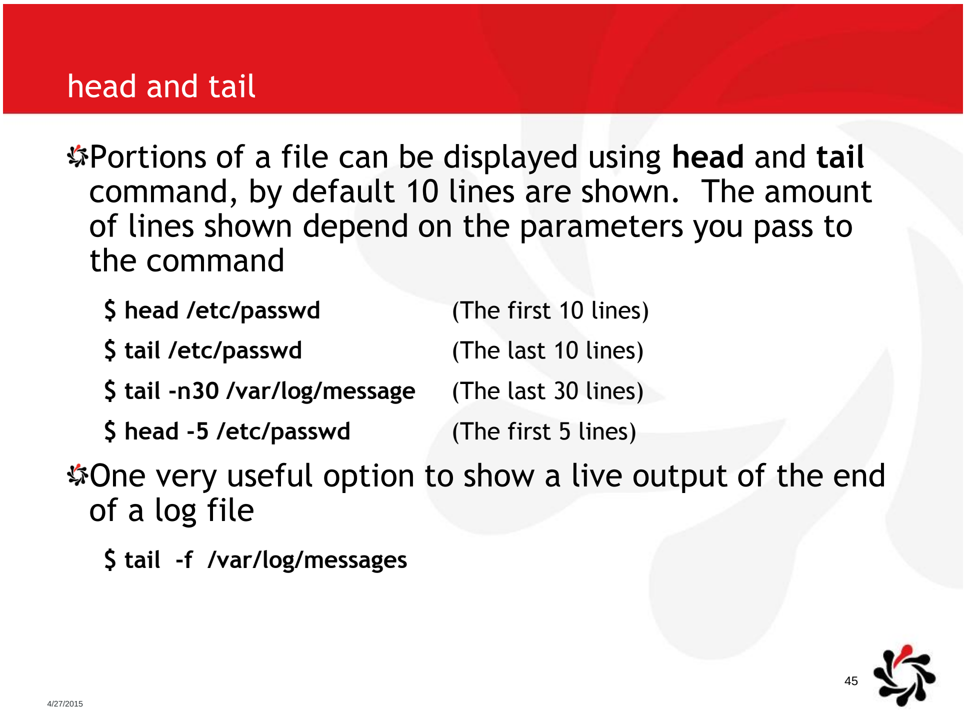
```
$ cat << EOL
```

- Join files together

```
$ cat  /etc/passwd   /etc/shadow  >  joinedfile
```

- **tac** is the reverse of cat and displays from the bottom of the file back

```
$ tac /etc/passwd
```



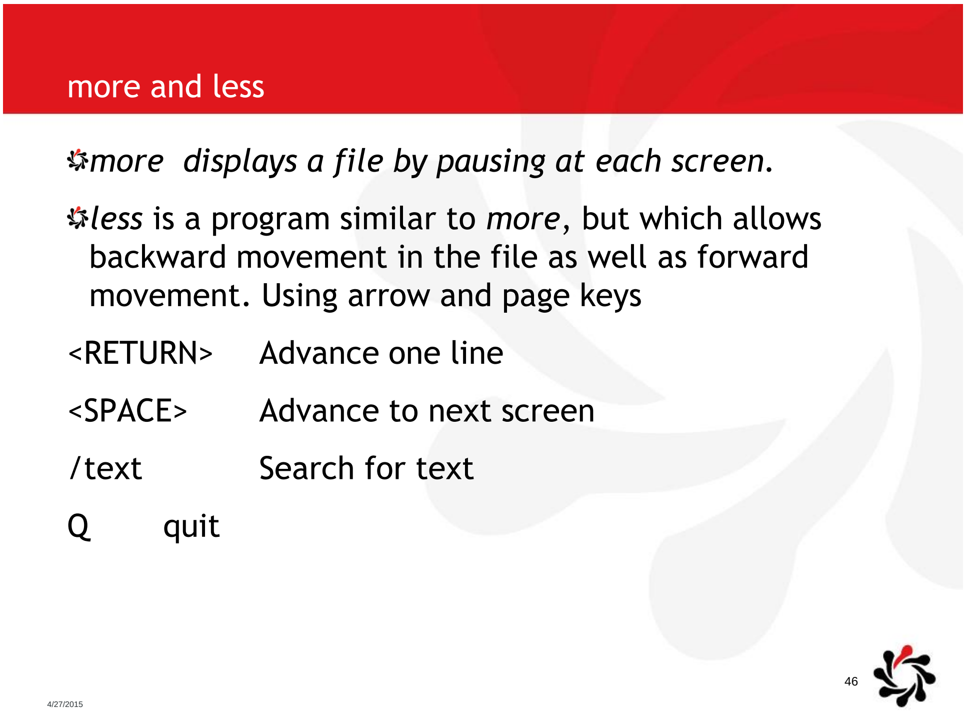
## head and tail

- Portions of a file can be displayed using **head** and **tail** command, by default 10 lines are shown. The amount of lines shown depend on the parameters you pass to the command

```
$ head /etc/passwd              (The first 10 lines)
$ tail /etc/passwd              (The last 10 lines)
$ tail -n30 /var/log/message    (The last 30 lines)
$ head -5 /etc/passwd           (The first 5 lines)
```

- One very useful option to show a live output of the end of a log file

```
$ tail  -f  /var/log/messages
```



## more and less

- *more  displays a file by pausing at each screen.*
- *less* is a program similar to *more*, but which allows backward movement in the file as well as forward movement. Using arrow and page keys

| | |
|---|---|
| <RETURN> | Advance one line |
| <SPACE> | Advance to next screen |
| /text | Search for text |
| Q | quit |

## cut and paste

- **cut** is used to extract portions of a file and display to std out. The following display characters 1 through 10 of the passwd file

```
$ cut -c1-10 /etc/passwd
```

- To specify a delimiter and work on fields, use the **-d** and the **-f** options

```
$ cut -d: -f 1,3,5 /etc/passwd
```

- The **paste** command joins files together by line matching line number

```
$ paste file1 file2
```

## join

- This powerful utility allows merging two files in a meaningful fashion, which essentially creates a simple version of a relational database.

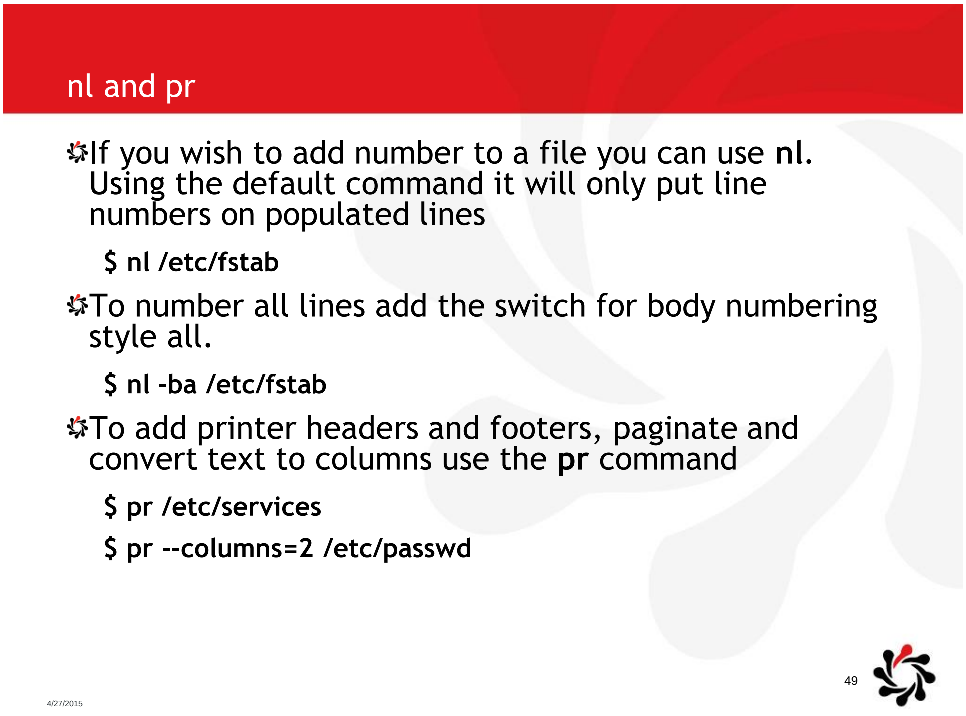
```
$ join file1 file2
```

- Use **-1** to specify the field in file 1 and **-2** to specify the field in file2

| File1 | |
|---|---|
| 100 | Shoes |
| 200 | Laces |
| 300 | Socks |

| File2 | |
|---|---|
| 100 | £40 |
| 200 | £1 |
| 300 | £2 |

| Output | | |
|---|---|---|
| 100 | Shoes | £40 |
| 200 | Laces | £1 |
| 300 | Socks | £2 |



## nl and pr

- If you wish to add number to a file you can use **nl**. Using the default command it will only put line numbers on populated lines

```
$ nl /etc/fstab
```

- To number all lines add the switch for body numbering style all.

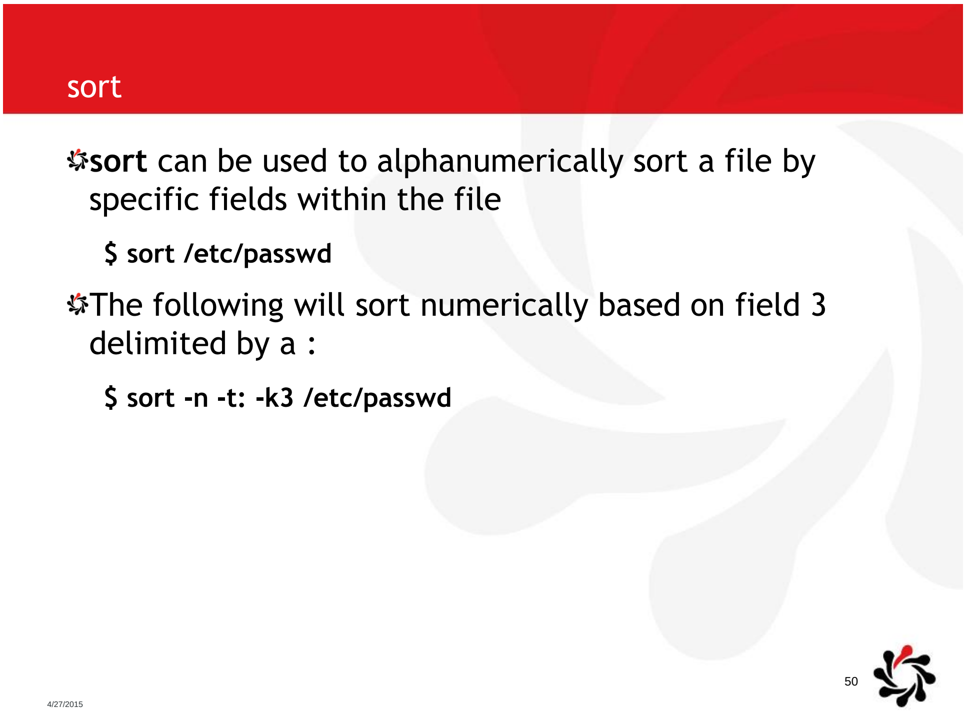
```
$ nl -ba /etc/fstab
```

- To add printer headers and footers, paginate and convert text to columns use the **pr** command

```
$ pr /etc/services
$ pr --columns=2 /etc/passwd
```

49

4/27/2015



## sort

- **sort** can be used to alphanumerically sort a file by specific fields within the file

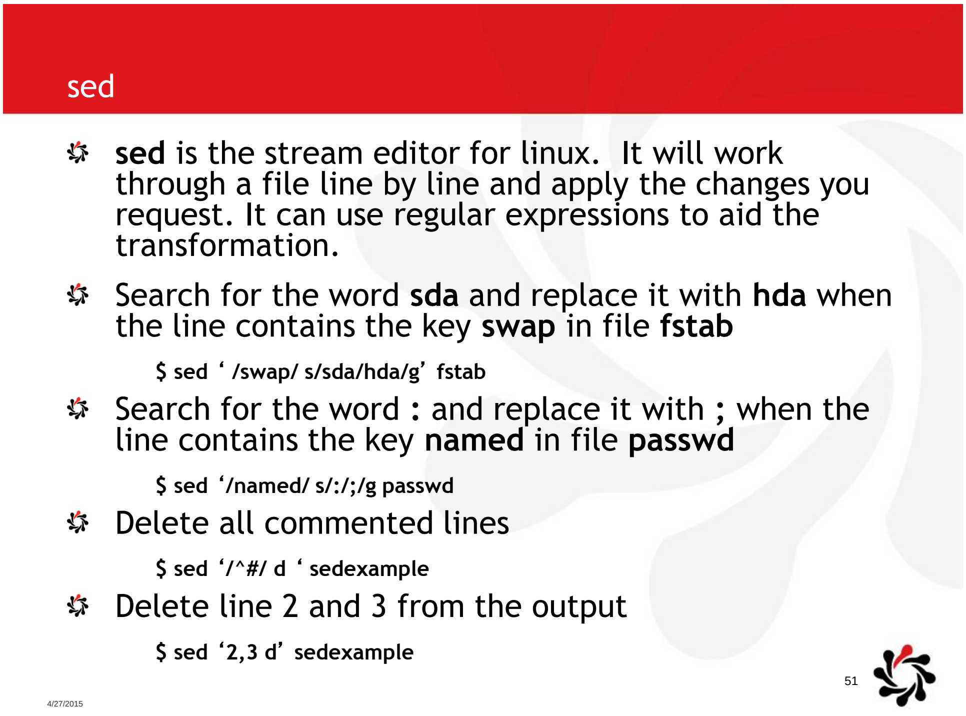
```
$ sort /etc/passwd
```

- The following will sort numerically based on field 3 delimited by a :

```
$ sort -n -t: -k3 /etc/passwd
```

50

4/27/2015



## sed

- **sed** is the stream editor for linux. It will work through a file line by line and apply the changes you request. It can use regular expressions to aid the transformation.
- Search for the word **sda** and replace it with **hda** when the line contains the key **swap** in file **fstab**

```
$ sed ‘ /swap/ s/sda/hda/g’ fstab
```

- Search for the word **:** and replace it with **;** when the line contains the key **named** in file **passwd**

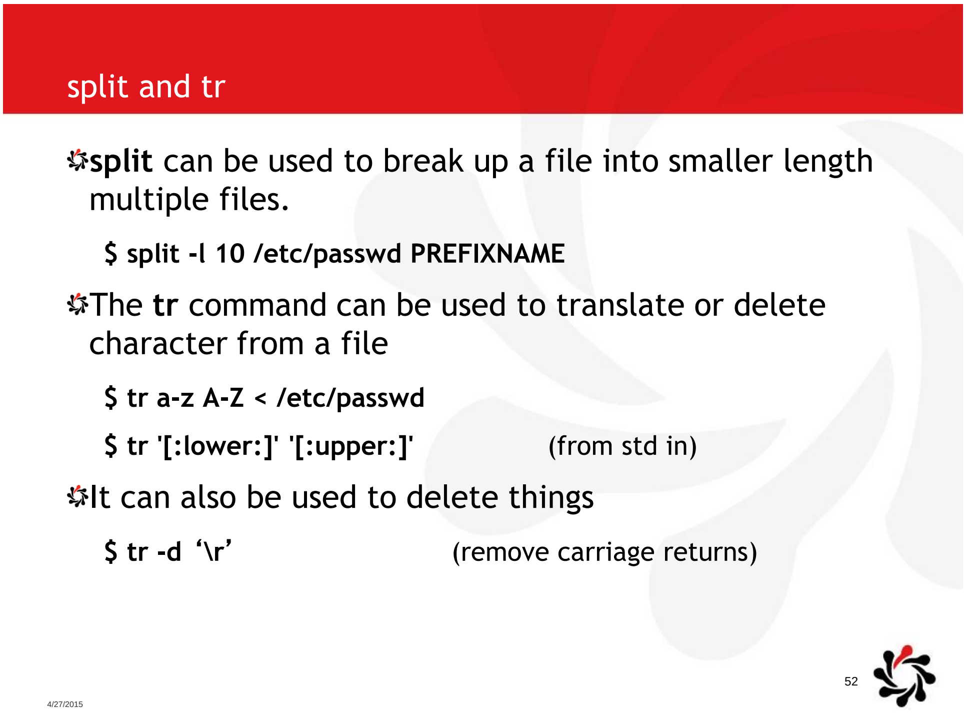
```
$ sed ‘/named/ s/:/;/g passwd
```

- Delete all commented lines

```
$ sed ‘/^#/ d ‘ sedexample
```

- Delete line 2 and 3 from the output

```
$ sed ‘2,3 d’ sedexample
```



## split and tr

- **split** can be used to break up a file into smaller length multiple files.

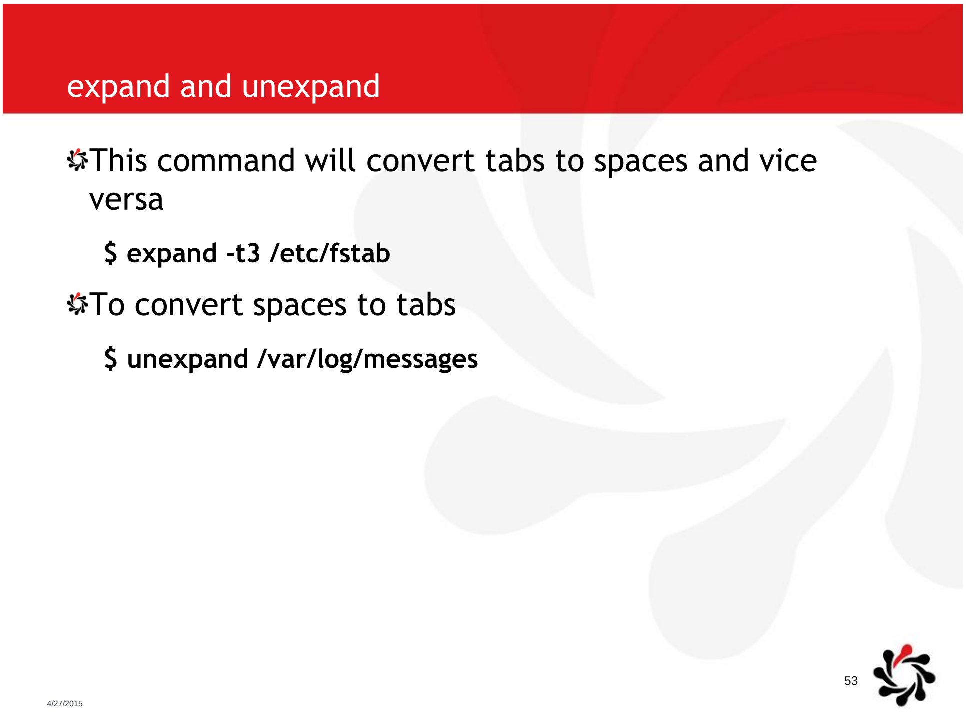
```
$ split -l 10 /etc/passwd PREFIXNAME
```

- The **tr** command can be used to translate or delete character from a file

```
$ tr a-z A-Z < /etc/passwd
$ tr '[:lower:]' '[:upper:]'               (from std in)
```

- It can also be used to delete things

```
$ tr -d ‘\r’                    (remove carriage returns)
```



## expand and unexpand

- This command will convert tabs to spaces and vice versa

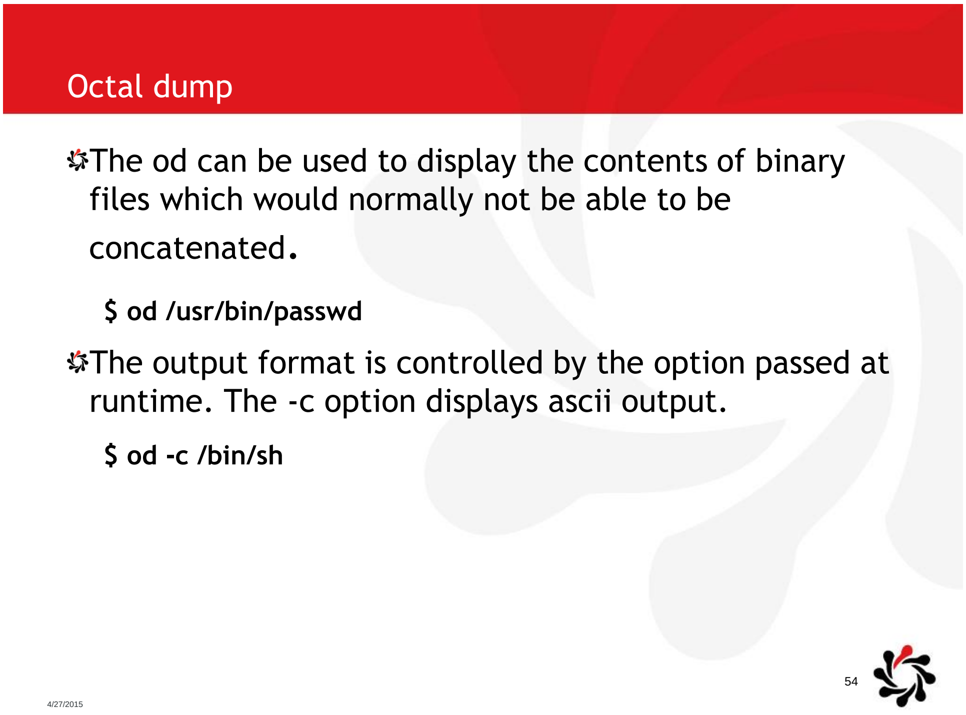
```
$ expand -t3 /etc/fstab
```

- To convert spaces to tabs

```
$ unexpand /var/log/messages
```

53

4/27/2015



## Octal dump

- The od can be used to display the contents of binary files which would normally not be able to be concatenated.

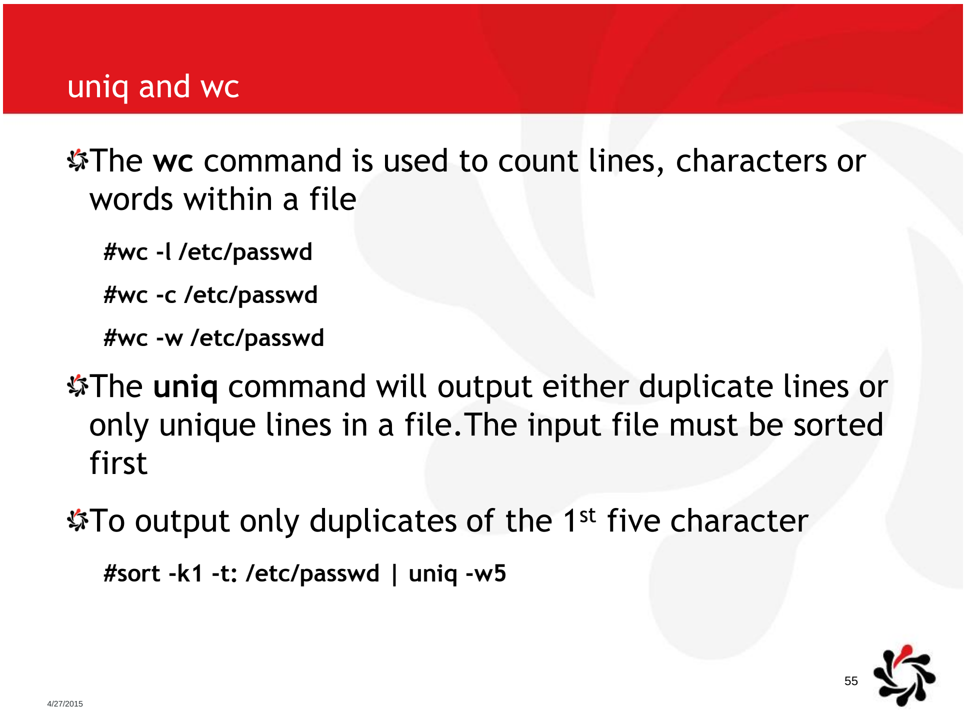
```
$ od /usr/bin/passwd
```

- The output format is controlled by the option passed at runtime. The -c option displays ascii output.

```
$ od -c /bin/sh
```

54

4/27/2015



## uniq and wc

- The **wc** command is used to count lines, characters or words within a file

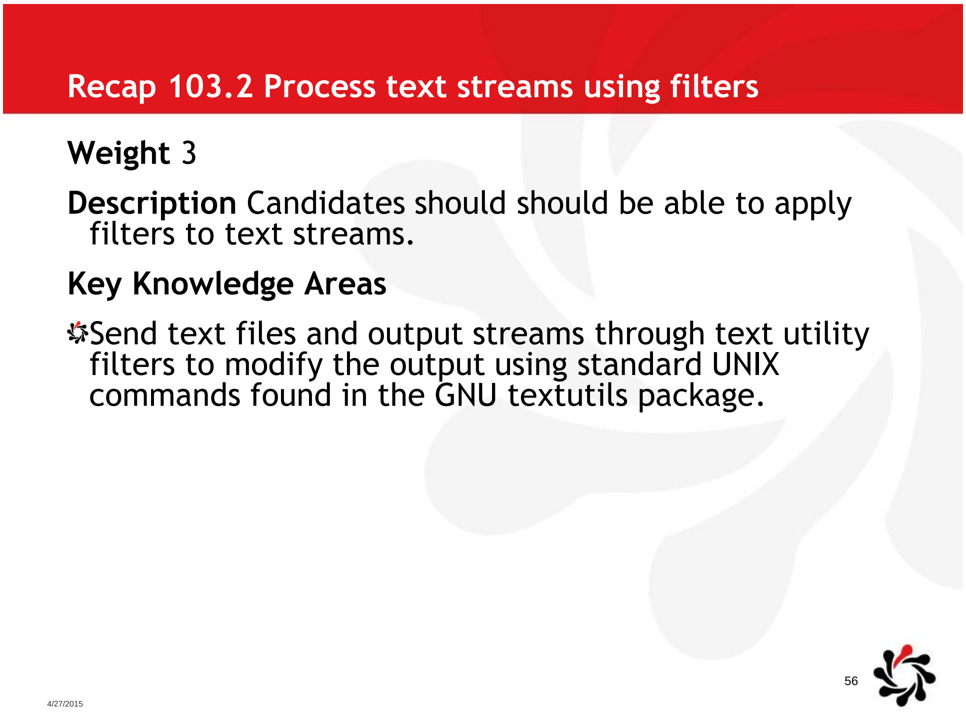
```
#wc -l /etc/passwd
#wc -c /etc/passwd
#wc -w /etc/passwd
```

- The **uniq** command will output either duplicate lines or only unique lines in a file.The input file must be sorted first
- To output only duplicates of the 1st five character

```
#sort -k1 -t: /etc/passwd | uniq -w5
```

55

4/27/2015



## Recap 103.2 Process text streams using filters

**Weight** 3

**Description** Candidates should should be able to apply filters to text streams.

**Key Knowledge Areas**

- Send text files and output streams through text utility filters to modify the output using standard UNIX commands found in the GNU textutils package.

56

4/27/2015

## Checklist of Terms and Utilities

- cat
- cut
- Expand
- head
- less
- od
- join
- nl
- paste
- pr
- sed
- sort
- split
- tail
- tr
- unexpand
- uniq
- wc

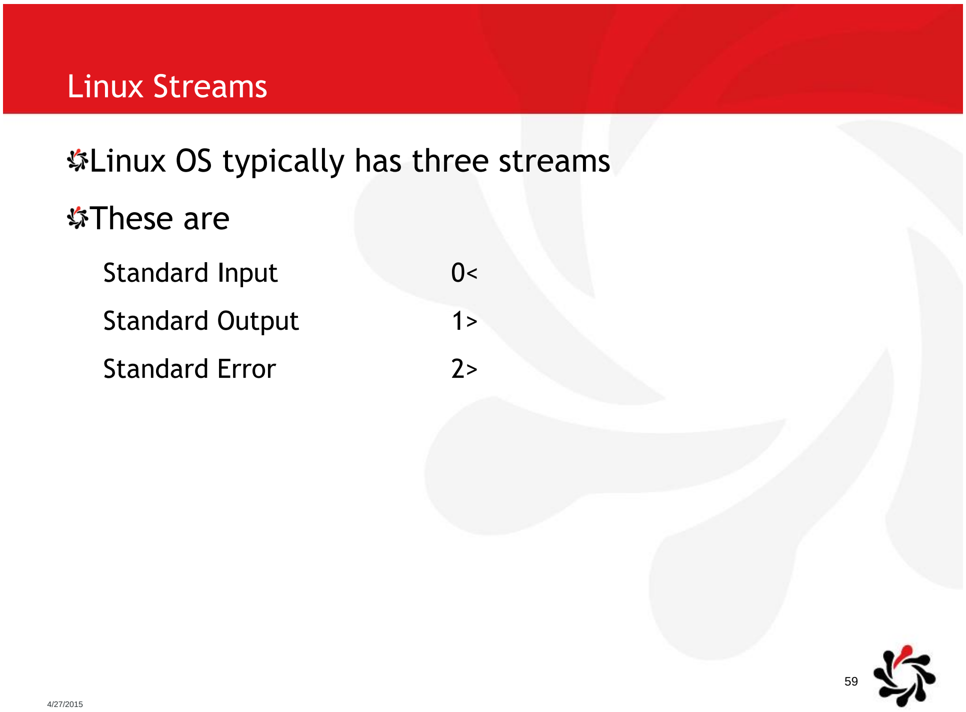
## 103.4 Use streams, pipes and redirects

**Weight** 4

**Description** Candidates should be able to redirect streams and connect them in order to efficiently process textual data. Tasks include redirecting standard input, standard output and standard error, piping the output of one command to the input of another command, using the output of one command as arguments to another command and sending output to both stdout and a file.

**Key Knowledge Areas**

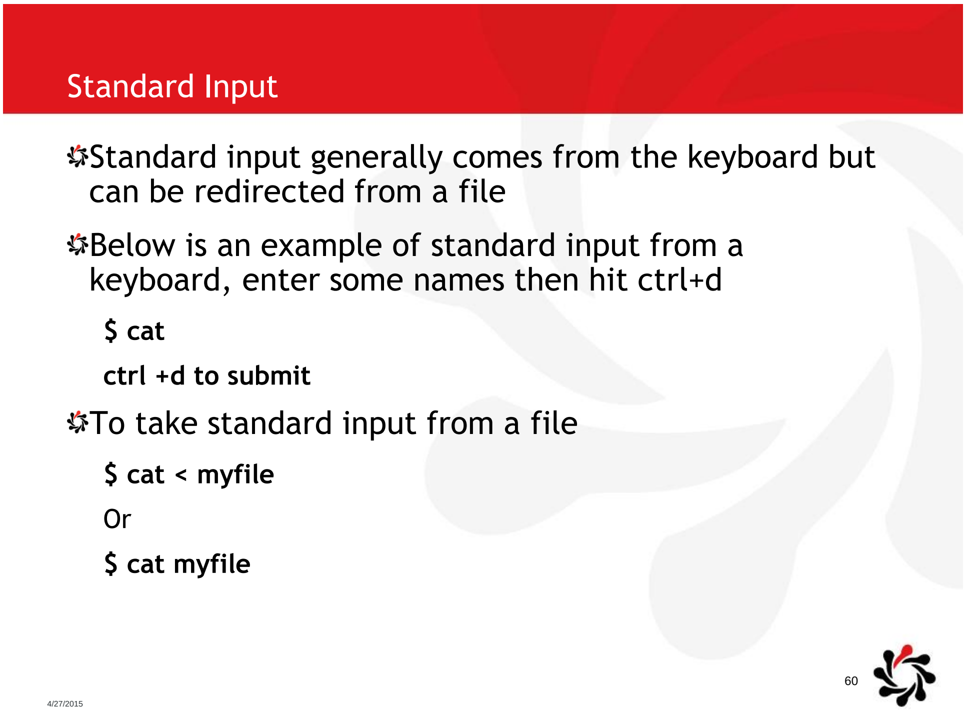
- Redirecting standard input, standard output and standard error.
- Pipe the output of one command to the input of another command.
- Use the output of one command as arguments to another command.
- Send output to both stdout and a file.



## Linux Streams

- Linux OS typically has three streams
- These are

| | |
|---|---|
| Standard Input | 0< |
| Standard Output | 1> |
| Standard Error | 2> |



## Standard Input

- Standard input generally comes from the keyboard but can be redirected from a file
- Below is an example of standard input from a keyboard, enter some names then hit ctrl+d

```
$ cat
ctrl +d to submit
```

- To take standard input from a file

```
$ cat < myfile
Or
$ cat myfile
```

## Standard Input

- To read standard input until a qualifier is seen use <<

```
$ cat << EOL
tom
Dick
harry
EOL
```

- The output will then display to standard output, which is the screen, but **NOT** the **EOL**
- To redirect this to a file

```
$ cat << EOL > myfile            (create the file)
$ cat << EOL >> myfile           (append to the file)
```

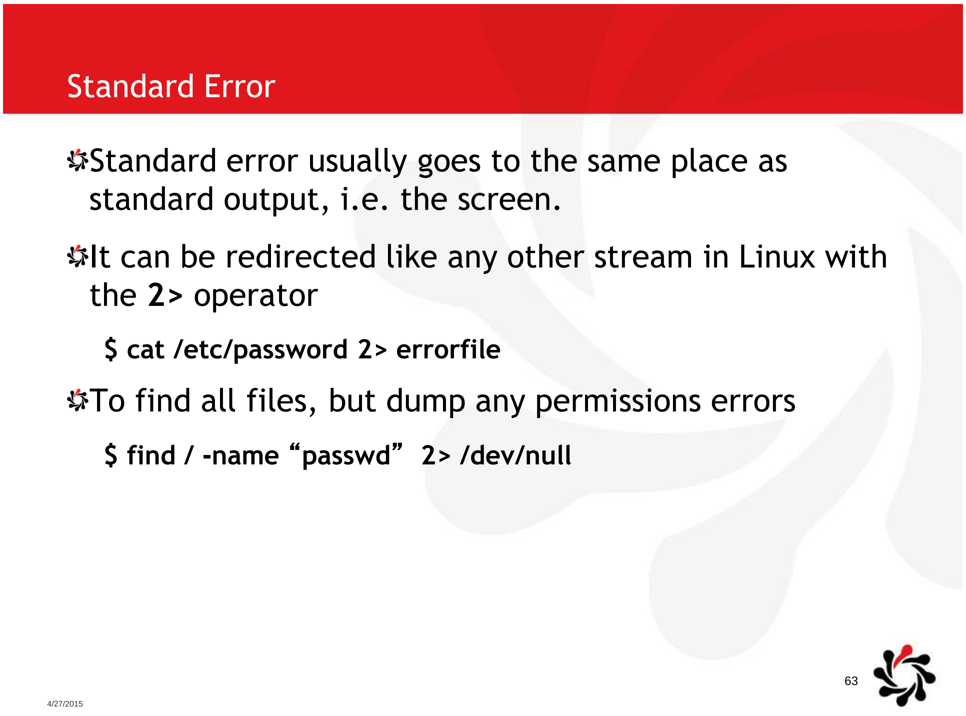
## Standard Output

- In the previous example using cat, the standard output went to the screen, which is the default
- This can be redirected out to a file

```
$ cat > myfile
```

- The above takes its input from the keyboard until ctrl+d is pressed and a new file called myfile will be created.
- Appending to the file use the >>

```
$ cat >> myfile
```



## Standard Error

- Standard error usually goes to the same place as standard output, i.e. the screen.
- It can be redirected like any other stream in Linux with the **2>** operator

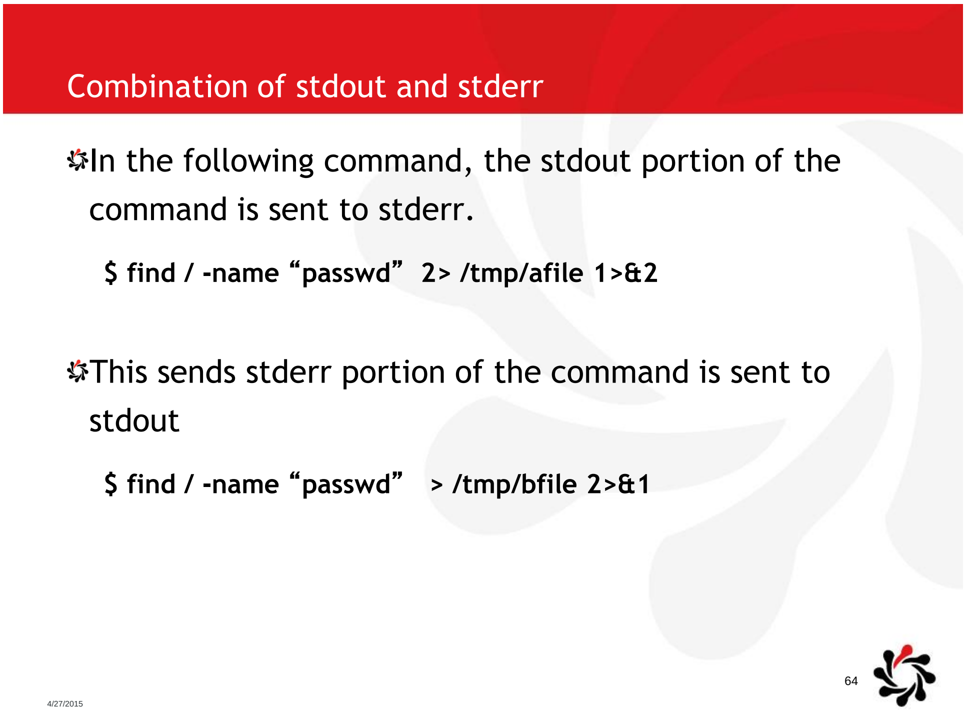
```
$ cat /etc/password 2> errorfile
```

- To find all files, but dump any permissions errors

```
$ find / -name “passwd”  2> /dev/null
```

4/27/2015

63



## Combination of stdout and stderr

- In the following command, the stdout portion of the command is sent to stderr.

```
$ find / -name “passwd”  2> /tmp/afile 1>&2
```

- This sends stderr portion of the command is sent to stdout

```
$ find / -name “passwd”   > /tmp/bfile 2>&1
```

4/27/2015

64

## Pipe

- The standard output of one command can be fed as standard input into another command using the pipe symbol **|**

```
$ ps -ef | grep -i bash | wc -l
$ who | grep -i root
```

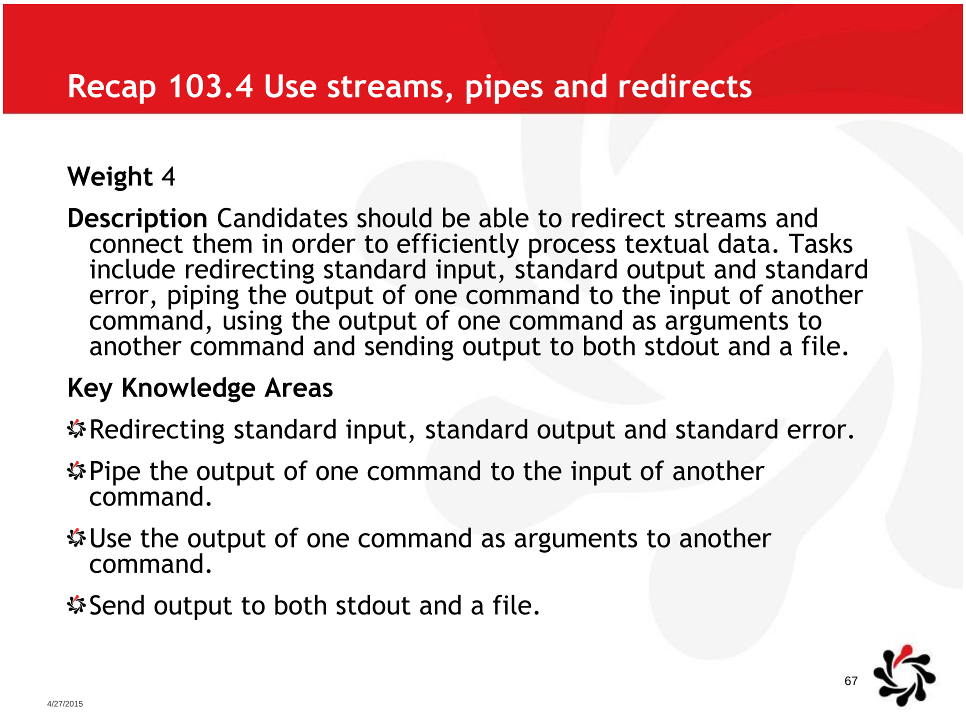
## The tee and xargs command

- The **tee** command allows you to create a duplicate stream from standard output of a command

```
$ sort names | tee sorted_out | nl > number_out
```

- The **xargs** command makes an argument out of standard output and feeds that as an argument into another command

```
$ locate README | xargs cat > all_the_readmes
```



## Recap 103.4 Use streams, pipes and redirects

**Weight** 4

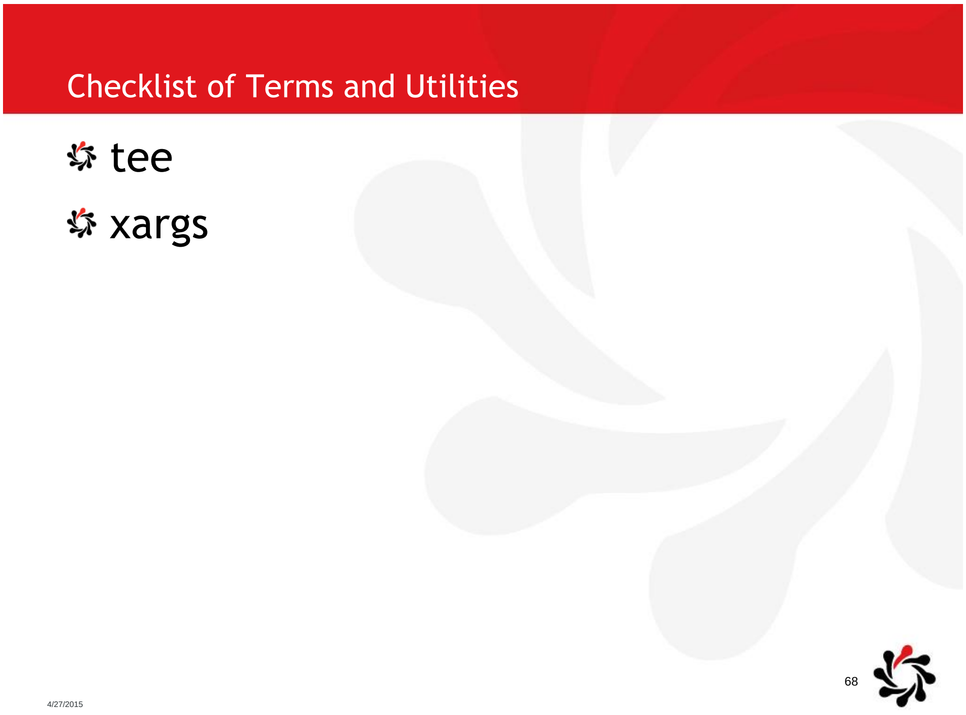
**Description** Candidates should be able to redirect streams and connect them in order to efficiently process textual data. Tasks include redirecting standard input, standard output and standard error, piping the output of one command to the input of another command, using the output of one command as arguments to another command and sending output to both stdout and a file.

**Key Knowledge Areas**

- Redirecting standard input, standard output and standard error.
- Pipe the output of one command to the input of another command.
- Use the output of one command as arguments to another command.
- Send output to both stdout and a file.

4/27/2015

67

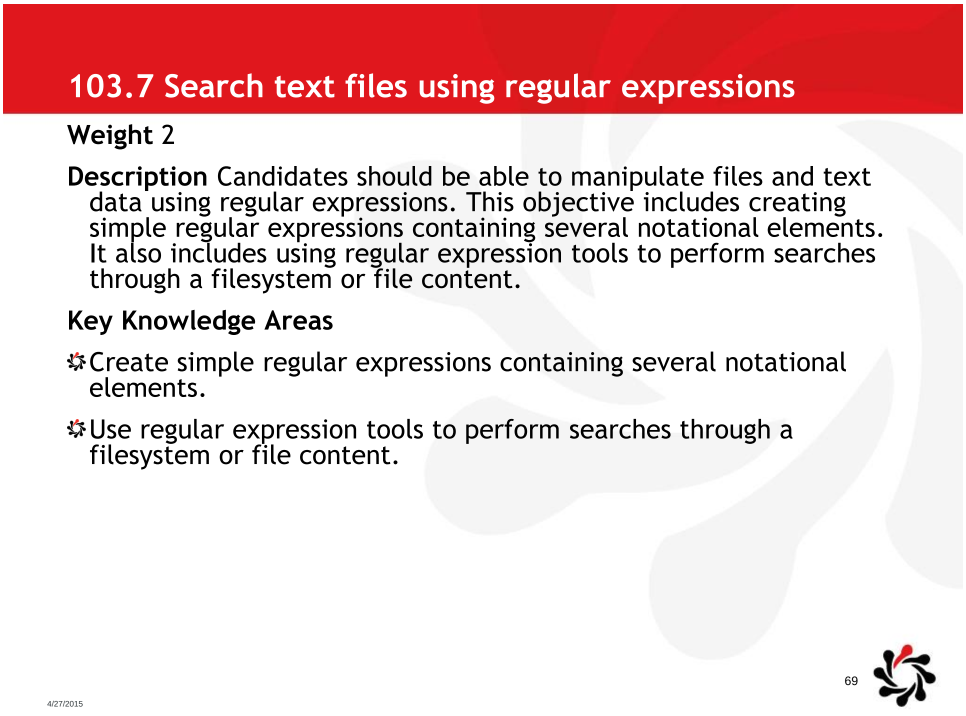


## Checklist of Terms and Utilities

- tee
- xargs

4/27/2015

68



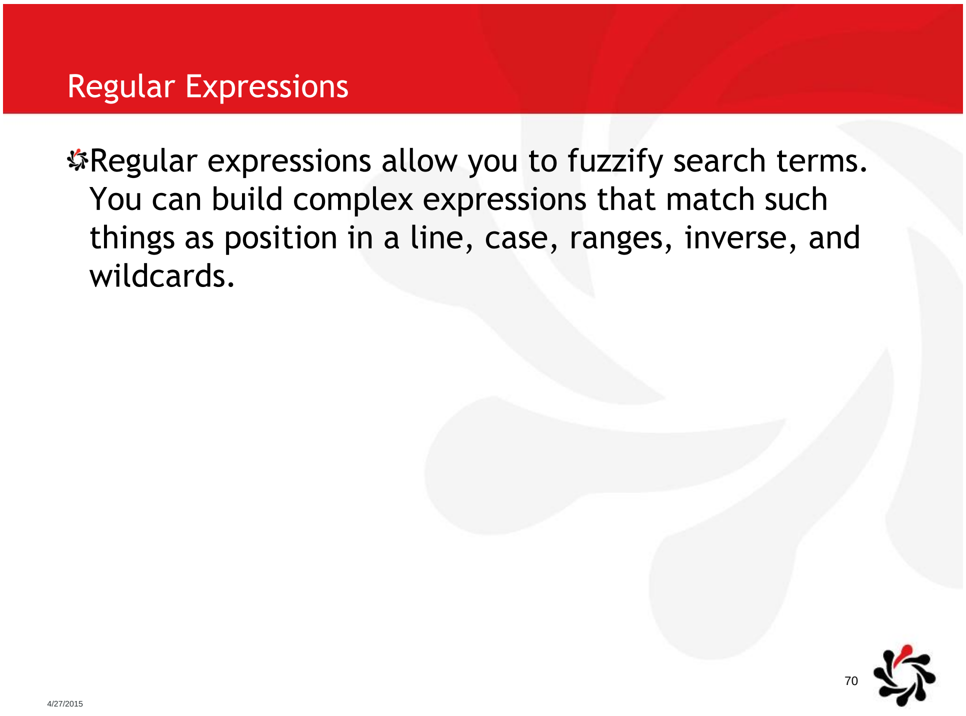
## 103.7 Search text files using regular expressions

**Weight** 2

**Description** Candidates should be able to manipulate files and text data using regular expressions. This objective includes creating simple regular expressions containing several notational elements. It also includes using regular expression tools to perform searches through a filesystem or file content.

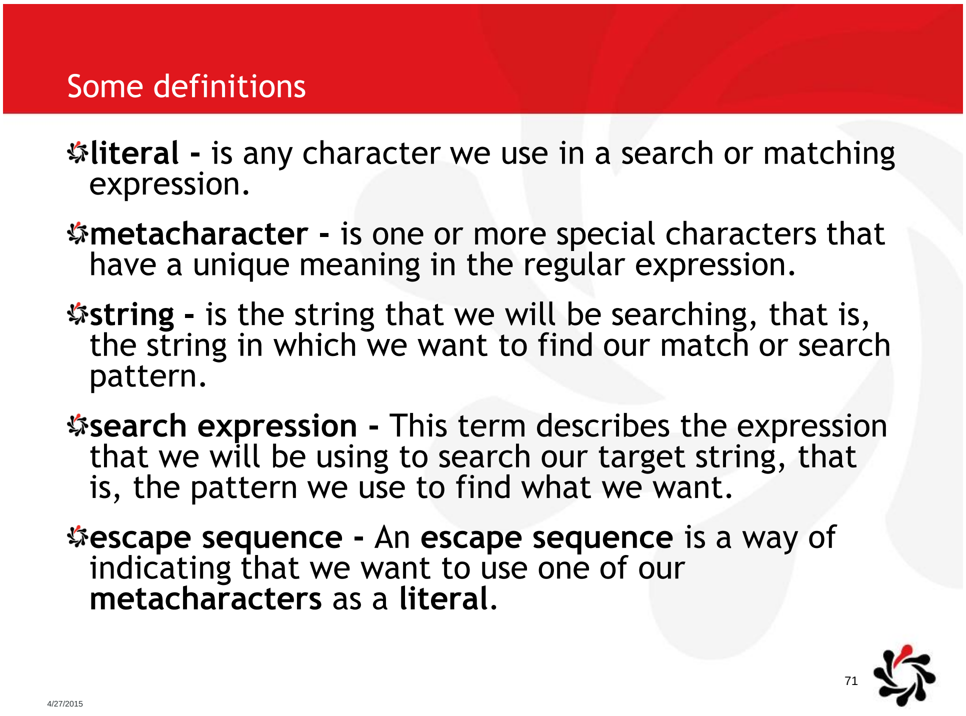
**Key Knowledge Areas**

- Create simple regular expressions containing several notational elements.
- Use regular expression tools to perform searches through a filesystem or file content.



## Regular Expressions

- Regular expressions allow you to fuzzify search terms. You can build complex expressions that match such things as position in a line, case, ranges, inverse, and wildcards.

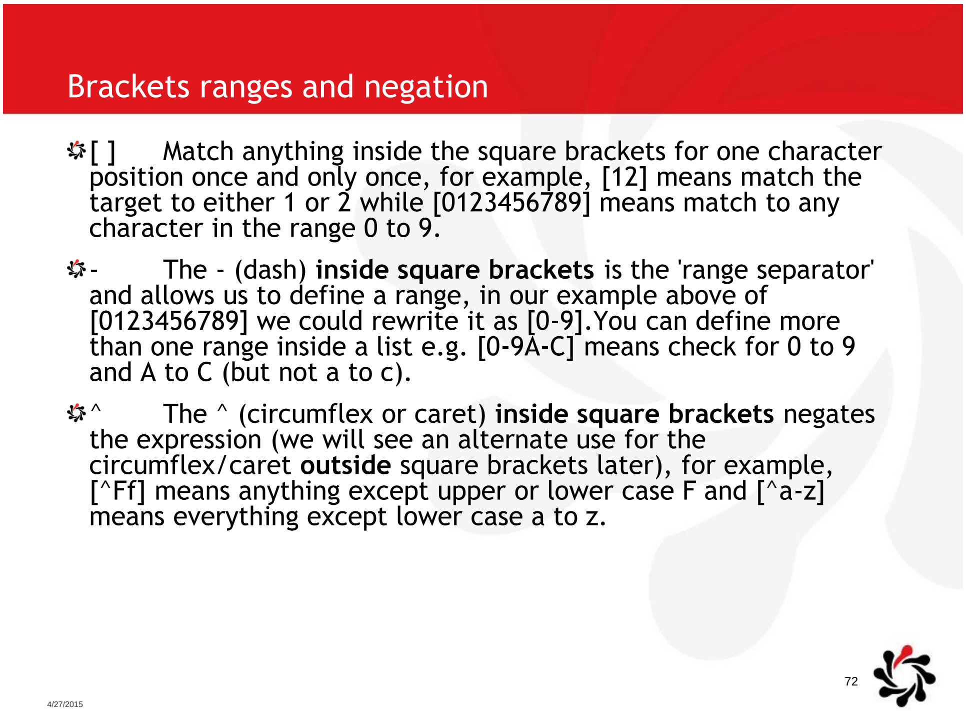


## Some definitions

- **literal -** is any character we use in a search or matching expression.
- **metacharacter -** is one or more special characters that have a unique meaning in the regular expression.
- **string -** is the string that we will be searching, that is, the string in which we want to find our match or search pattern.
- **search expression -** This term describes the expression that we will be using to search our target string, that is, the pattern we use to find what we want.
- **escape sequence -** An **escape sequence** is a way of indicating that we want to use one of our **metacharacters** as a **literal**.

71

4/27/2015

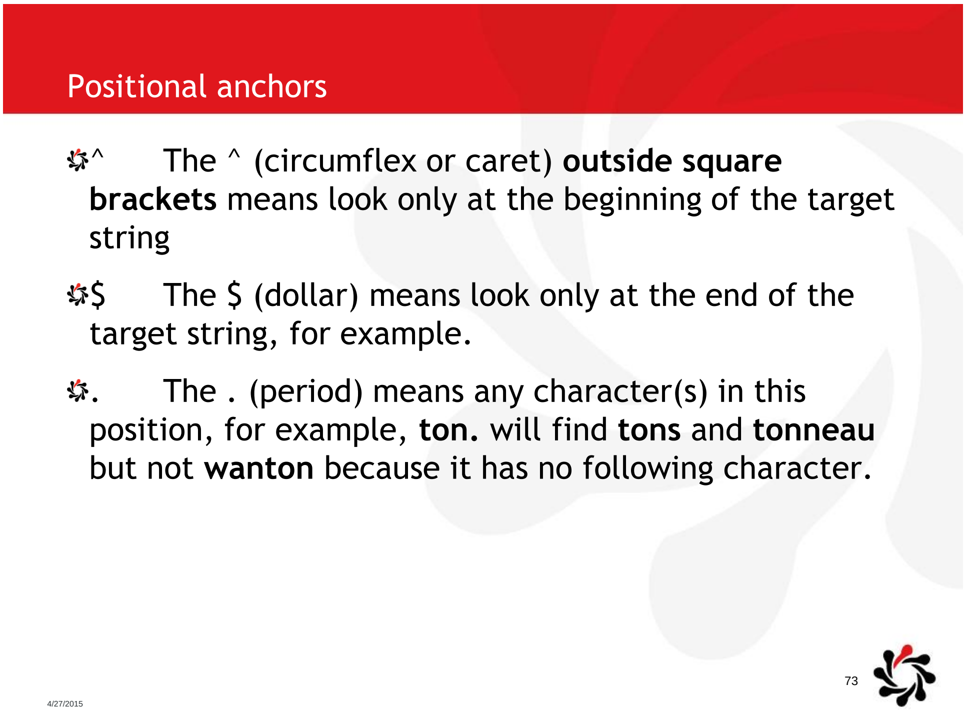


## Brackets ranges and negation

- [ ]	Match anything inside the square brackets for one character position once and only once, for example, [12] means match the target to either 1 or 2 while [0123456789] means match to any character in the range 0 to 9.
- -	The - (dash) **inside square brackets** is the 'range separator' and allows us to define a range, in our example above of [0123456789] we could rewrite it as [0-9].You can define more than one range inside a list e.g. [0-9A-C] means check for 0 to 9 and A to C (but not a to c).
- ^	The ^ (circumflex or caret) **inside square brackets** negates the expression (we will see an alternate use for the circumflex/caret **outside** square brackets later), for example, [^Ff] means anything except upper or lower case F and [^a-z] means everything except lower case a to z.

72

4/27/2015

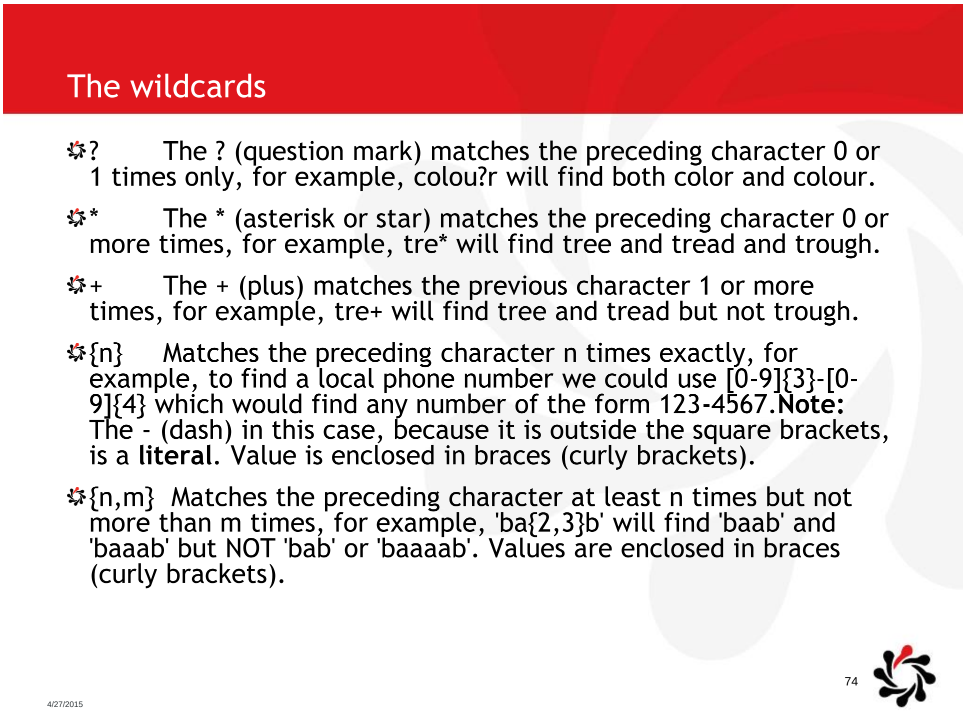


## Positional anchors

- ^ The ^ (circumflex or caret) **outside square brackets** means look only at the beginning of the target string
- $ The $ (dollar) means look only at the end of the target string, for example.
- . The . (period) means any character(s) in this position, for example, **ton.** will find **tons** and **tonneau** but not **wanton** because it has no following character.

4/27/2015

73

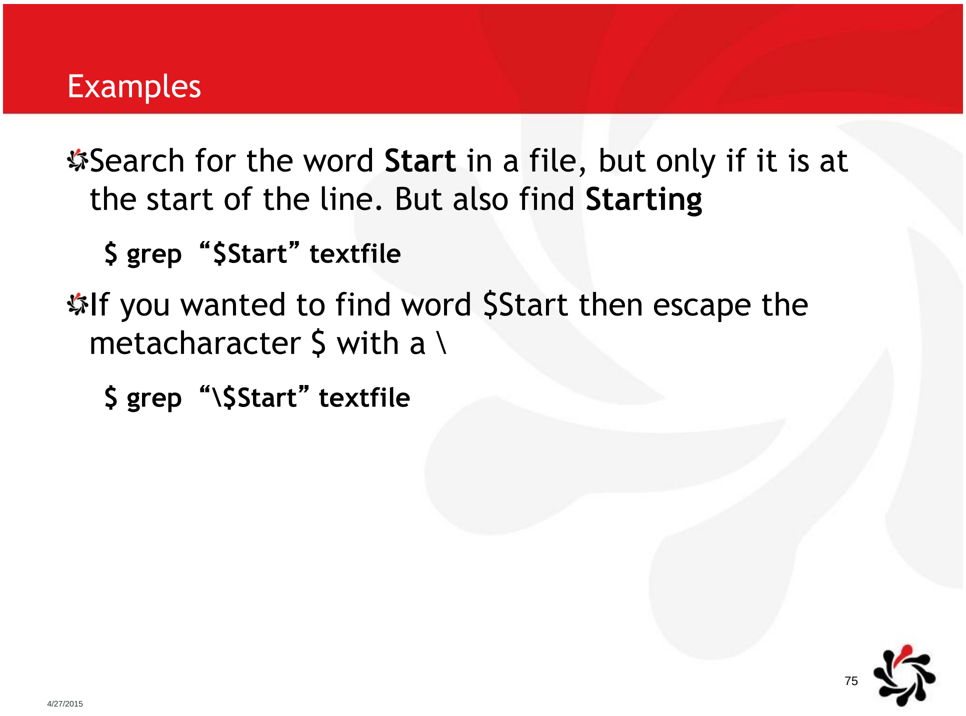


## The wildcards

- ? The ? (question mark) matches the preceding character 0 or 1 times only, for example, colou?r will find both color and colour.
- * The * (asterisk or star) matches the preceding character 0 or more times, for example, tre* will find tree and tread and trough.
- + The + (plus) matches the previous character 1 or more times, for example, tre+ will find tree and tread but not trough.
- {n} Matches the preceding character n times exactly, for example, to find a local phone number we could use [0-9]{3}-[0-9]{4} which would find any number of the form 123-4567.**Note:** The - (dash) in this case, because it is outside the square brackets, is a **literal**. Value is enclosed in braces (curly brackets).
- {n,m} Matches the preceding character at least n times but not more than m times, for example, 'ba{2,3}b' will find 'baab' and 'baaab' but NOT 'bab' or 'baaaab'. Values are enclosed in braces (curly brackets).

4/27/2015

74



## Examples

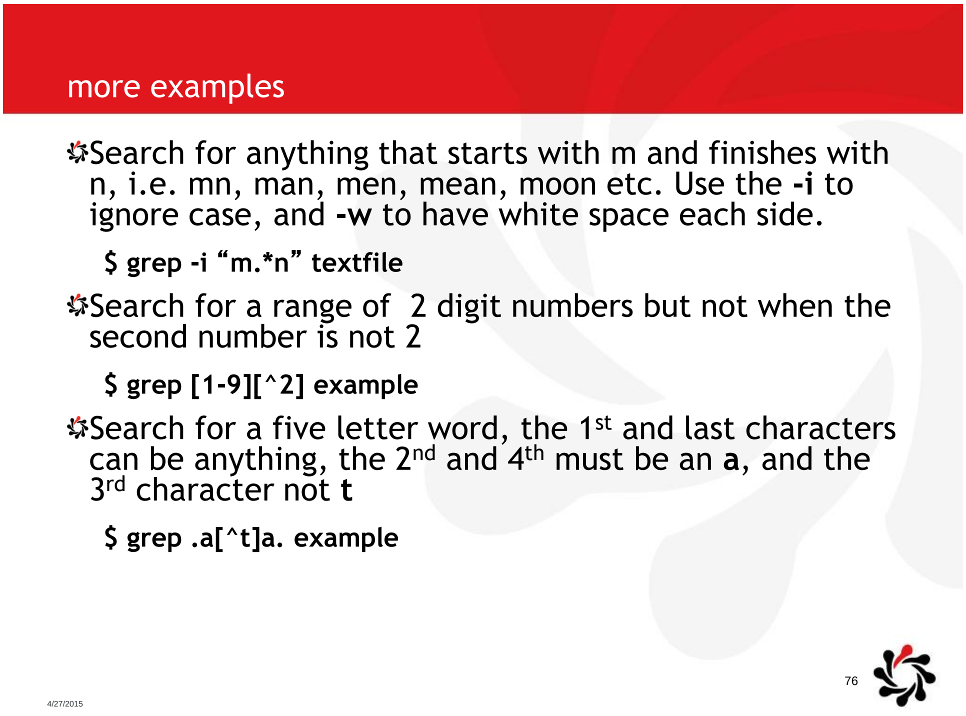
- Search for the word **Start** in a file, but only if it is at the start of the line. But also find **Starting**

```
$ grep  "$Start" textfile
```

- If you wanted to find word $Start then escape the metacharacter $ with a \

```
$ grep  "\$Start" textfile
```

4/27/2015 75



## more examples

- Search for anything that starts with m and finishes with n, i.e. mn, man, men, mean, moon etc. Use the **-i** to ignore case, and **-w** to have white space each side.

```
$ grep -i "m.*n" textfile
```

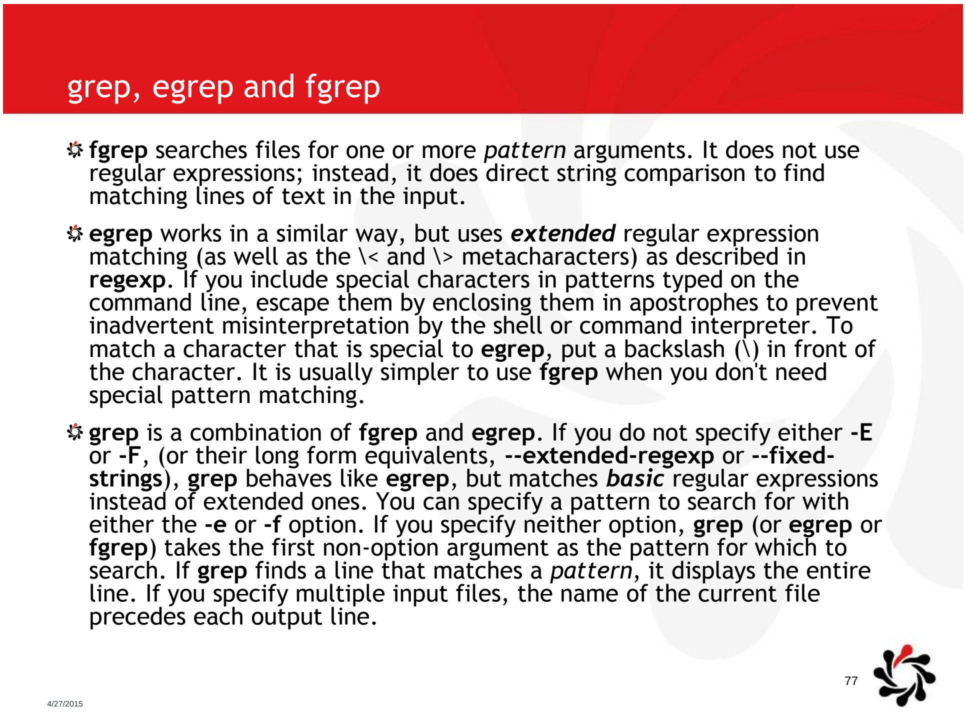
- Search for a range of  2 digit numbers but not when the second number is not 2

```
$ grep [1-9][^2] example
```

- Search for a five letter word, the 1st and last characters can be anything, the 2nd and 4th must be an **a**, and the 3rd character not **t**

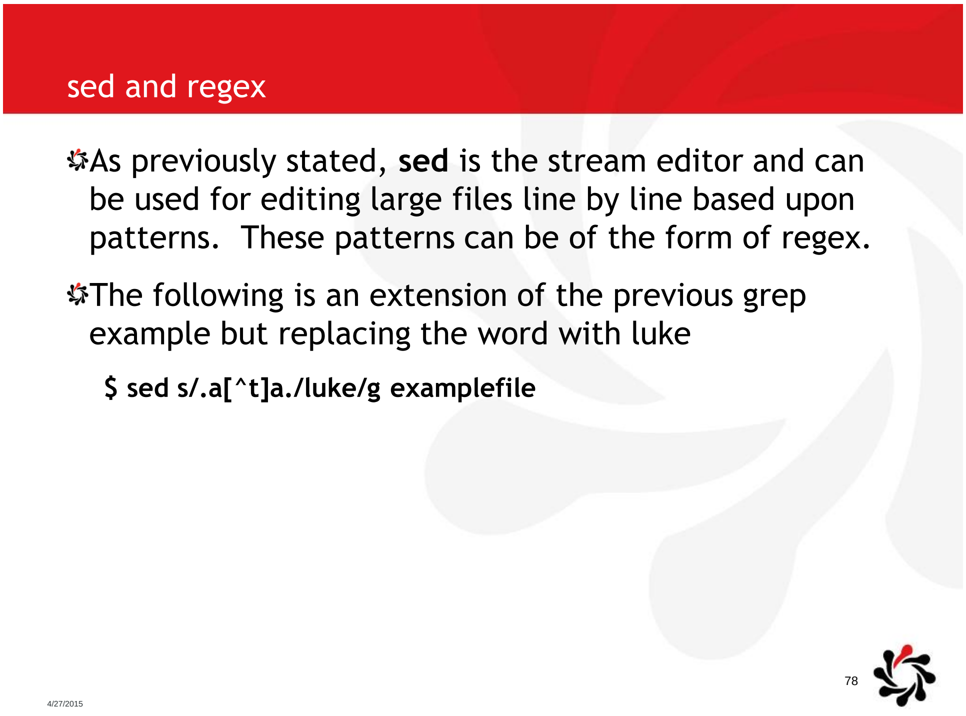
```
$ grep .a[^t]a. example
```

4/27/2015 76



## grep, egrep and fgrep

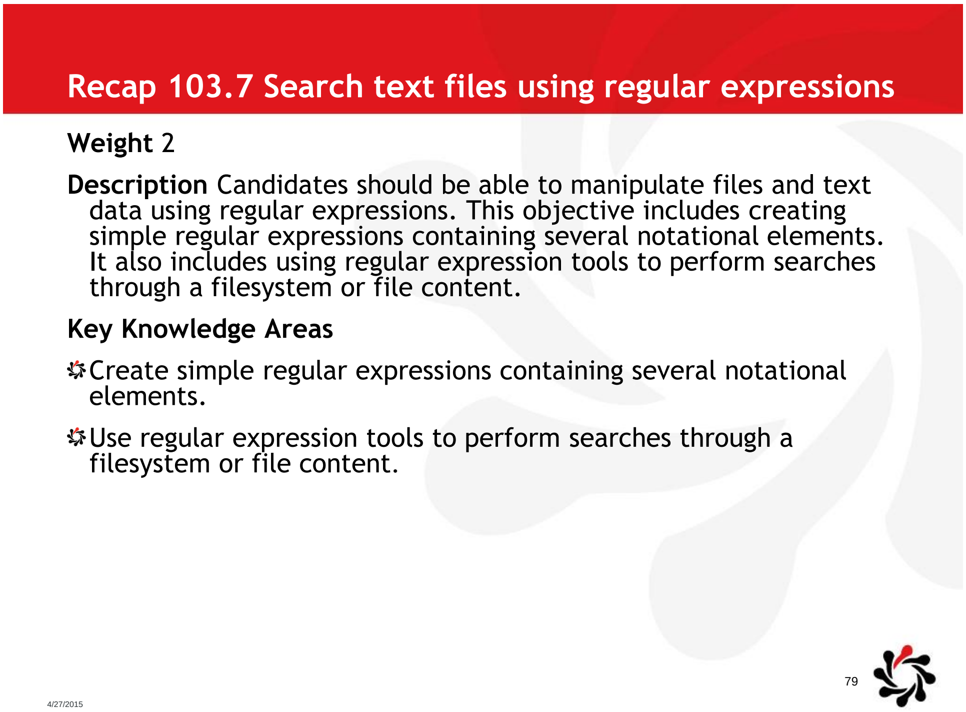
- **fgrep** searches files for one or more *pattern* arguments. It does not use regular expressions; instead, it does direct string comparison to find matching lines of text in the input.
- **egrep** works in a similar way, but uses ***extended*** regular expression matching (as well as the \< and \> metacharacters) as described in **regexp**. If you include special characters in patterns typed on the command line, escape them by enclosing them in apostrophes to prevent inadvertent misinterpretation by the shell or command interpreter. To match a character that is special to **egrep**, put a backslash (\) in front of the character. It is usually simpler to use **fgrep** when you don't need special pattern matching.
- **grep** is a combination of **fgrep** and **egrep**. If you do not specify either **-E** or **-F**, (or their long form equivalents, **--extended-regexp** or **--fixed-strings**), **grep** behaves like **egrep**, but matches ***basic*** regular expressions instead of extended ones. You can specify a pattern to search for with either the **-e** or **-f** option. If you specify neither option, **grep** (or **egrep** or **fgrep**) takes the first non-option argument as the pattern for which to search. If **grep** finds a line that matches a *pattern*, it displays the entire line. If you specify multiple input files, the name of the current file precedes each output line.



## sed and regex

- As previously stated, **sed** is the stream editor and can be used for editing large files line by line based upon patterns.  These patterns can be of the form of regex.
- The following is an extension of the previous grep example but replacing the word with luke

```
$ sed s/.a[^t]a./luke/g examplefile
```



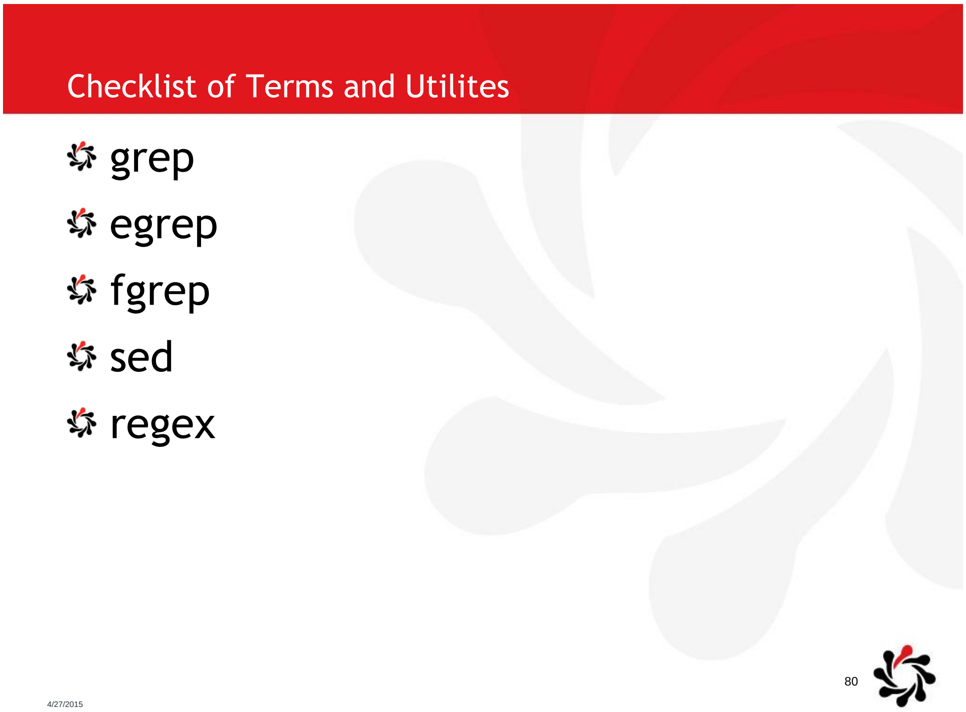
## Recap 103.7 Search text files using regular expressions

**Weight** 2

**Description** Candidates should be able to manipulate files and text data using regular expressions. This objective includes creating simple regular expressions containing several notational elements. It also includes using regular expression tools to perform searches through a filesystem or file content.

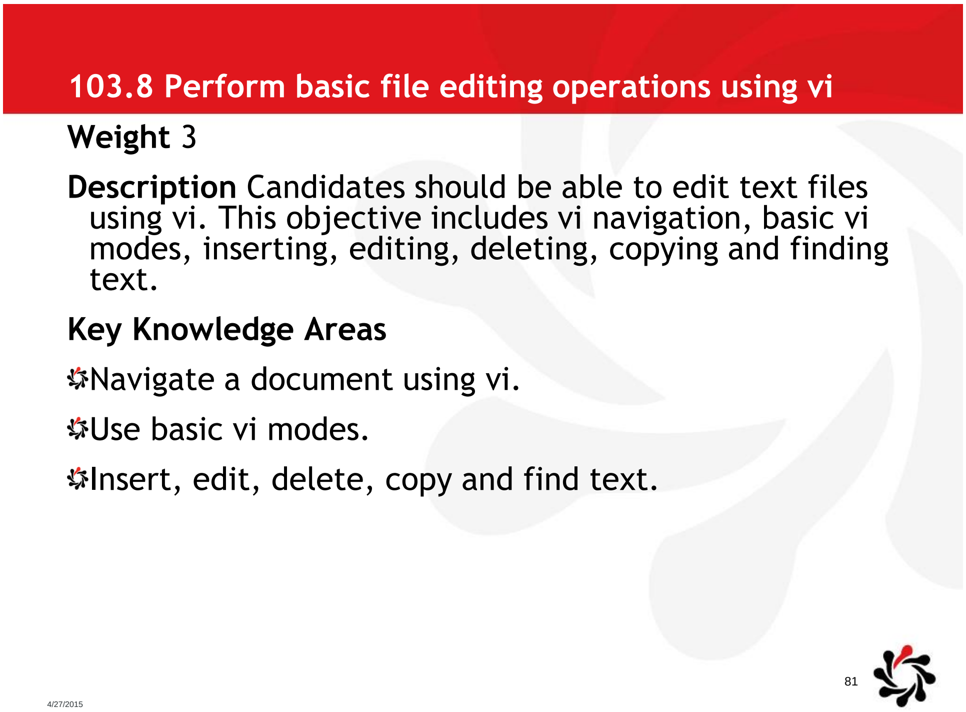
**Key Knowledge Areas**

- Create simple regular expressions containing several notational elements.
- Use regular expression tools to perform searches through a filesystem or file content.



## Checklist of Terms and Utilites

- grep
- egrep
- fgrep
- sed
- regex



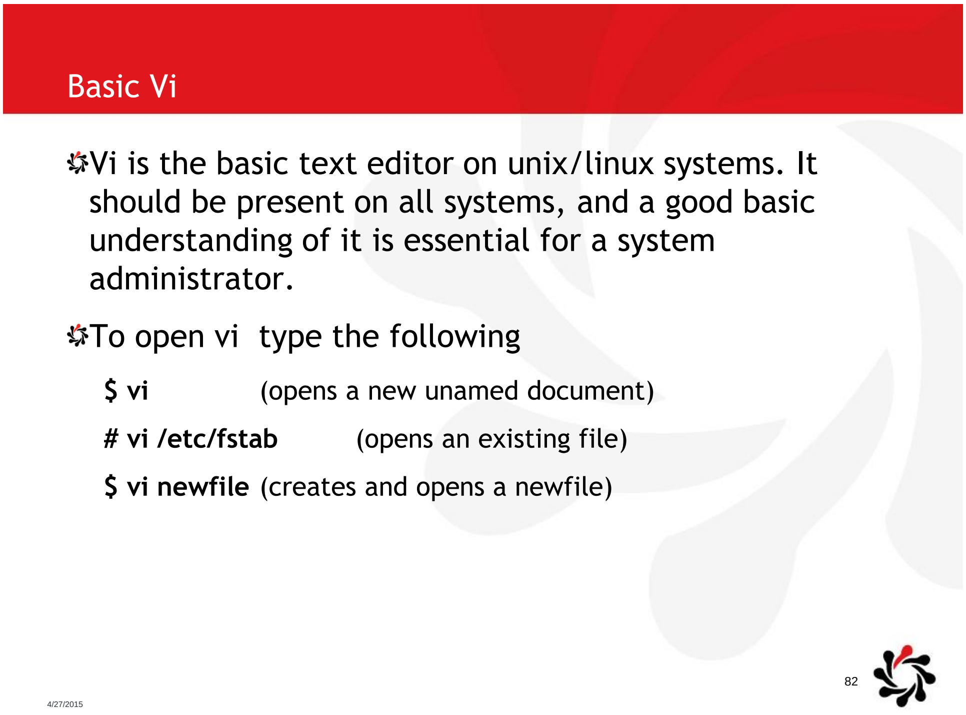
## 103.8 Perform basic file editing operations using vi

**Weight** 3

**Description** Candidates should be able to edit text files using vi. This objective includes vi navigation, basic vi modes, inserting, editing, deleting, copying and finding text.

**Key Knowledge Areas**

- Navigate a document using vi.
- Use basic vi modes.
- Insert, edit, delete, copy and find text.



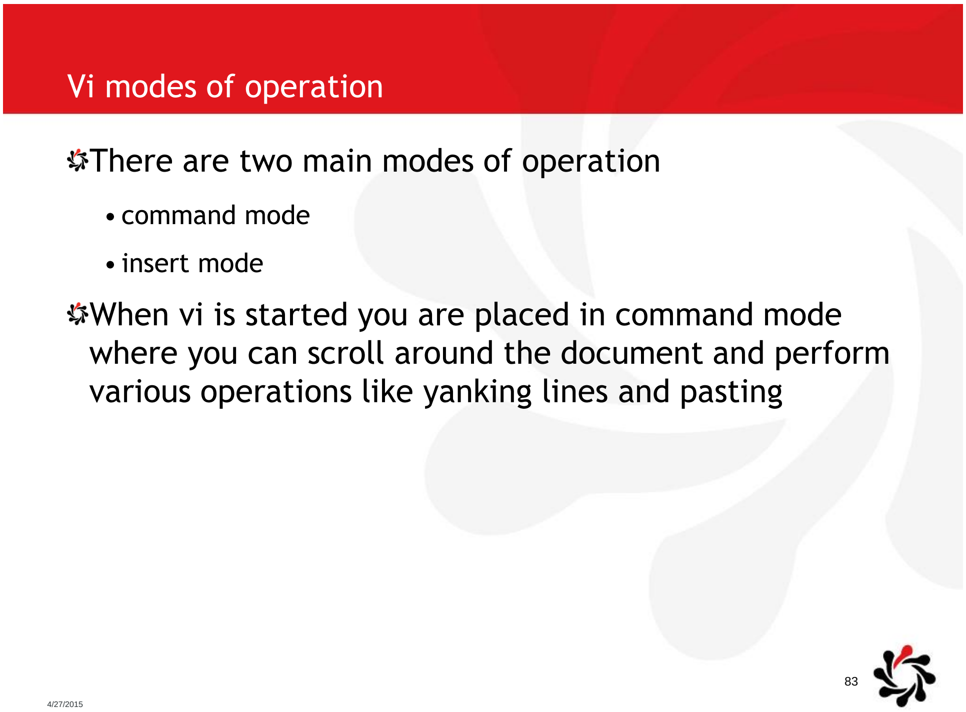
## Basic Vi

- Vi is the basic text editor on unix/linux systems. It should be present on all systems, and a good basic understanding of it is essential for a system administrator.
- To open vi  type the following

  **$ vi**  (opens a new unamed document)

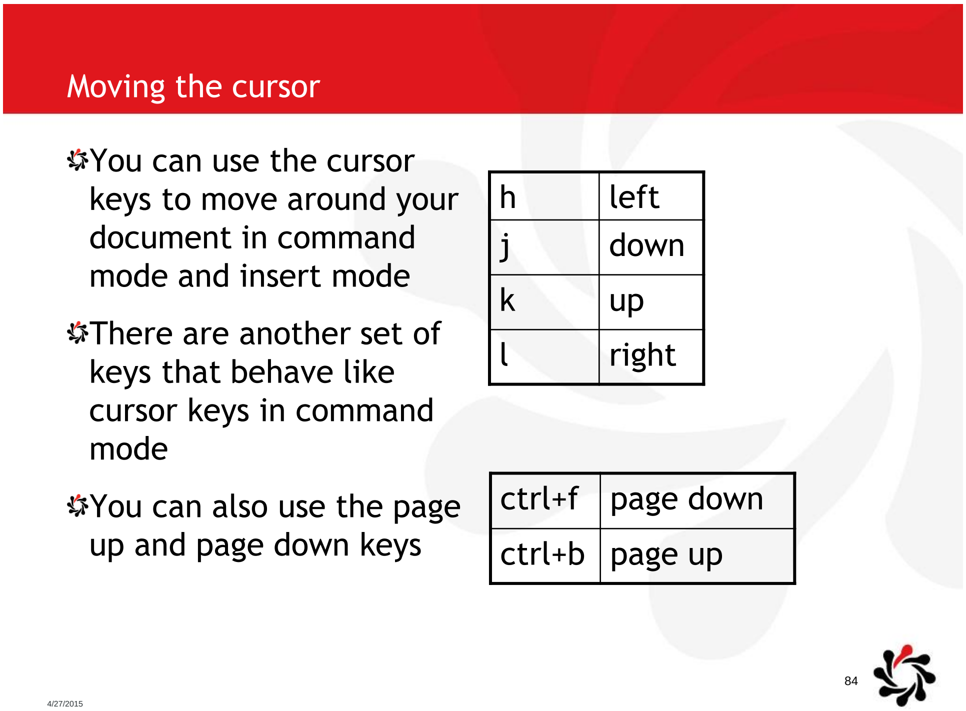
  **# vi /etc/fstab**  (opens an existing file)

  **$ vi newfile** (creates and opens a newfile)



## Vi modes of operation

- There are two main modes of operation
  - command mode
  - insert mode
- When vi is started you are placed in command mode where you can scroll around the document and perform various operations like yanking lines and pasting

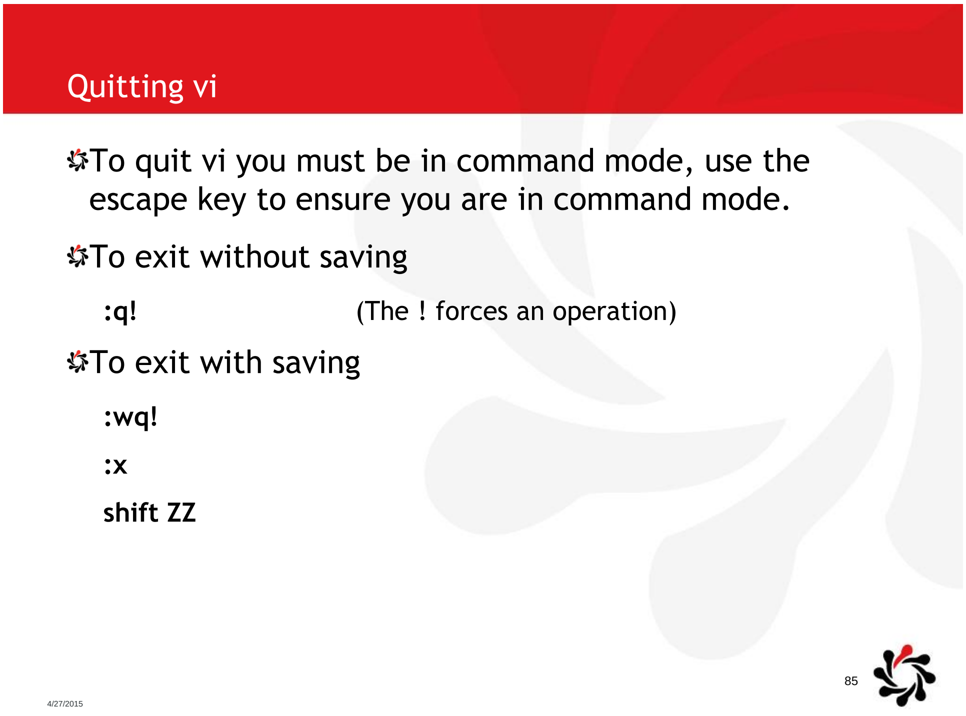


## Moving the cursor

- You can use the cursor keys to move around your document in command mode and insert mode
- There are another set of keys that behave like cursor keys in command mode
- You can also use the page up and page down keys

| | |
|---|---|
| h | left |
| j | down |
| k | up |
| l | right |

| | |
|---|---|
| ctrl+f | page down |
| ctrl+b | page up |



## Quitting vi

- To quit vi you must be in command mode, use the escape key to ensure you are in command mode.
- To exit without saving

  **:q!** (The ! forces an operation)
- To exit with saving

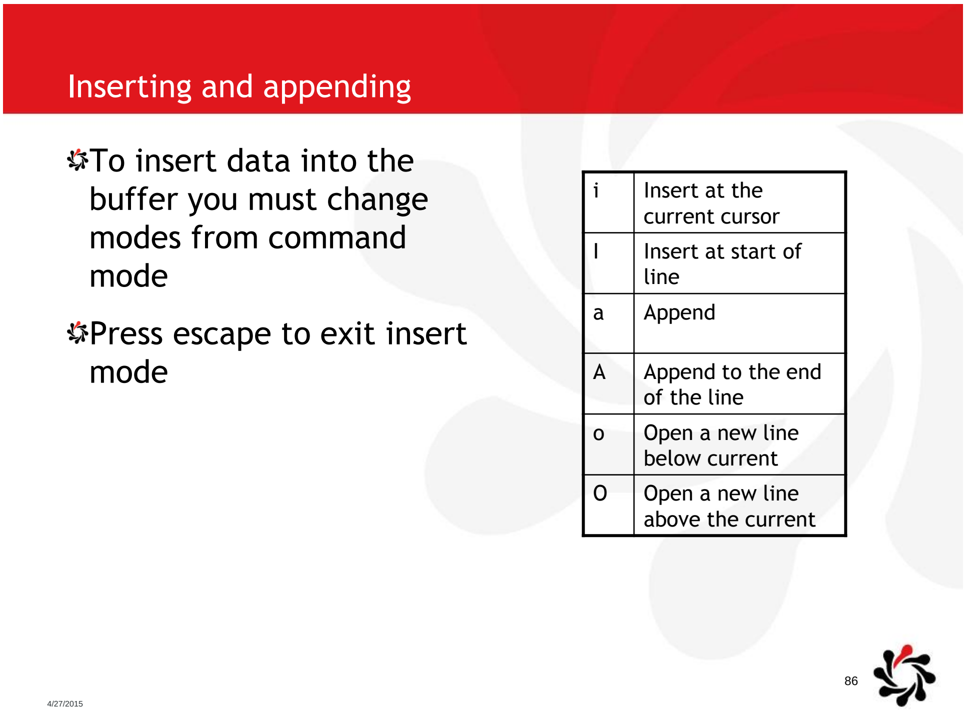
  **:wq!**

  **:x**

  **shift ZZ**

4/27/2015

85



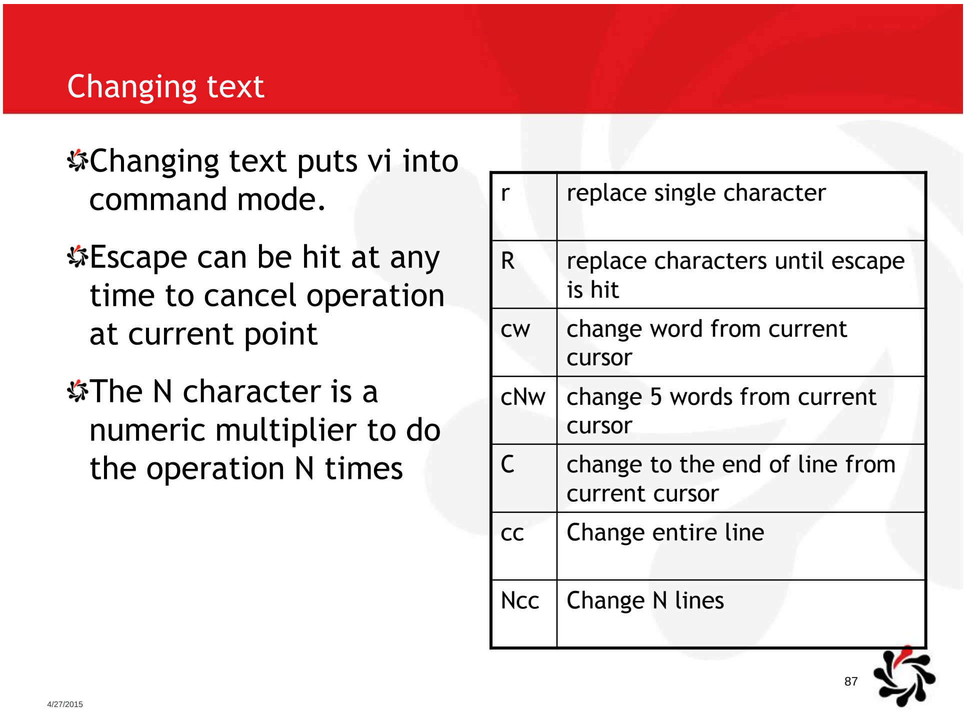
## Inserting and appending

- To insert data into the buffer you must change modes from command mode
- Press escape to exit insert mode

| | |
|---|---|
| i | Insert at the current cursor |
| I | Insert at start of line |
| a | Append |
| A | Append to the end of the line |
| o | Open a new line below current |
| O | Open a new line above the current |

4/27/2015

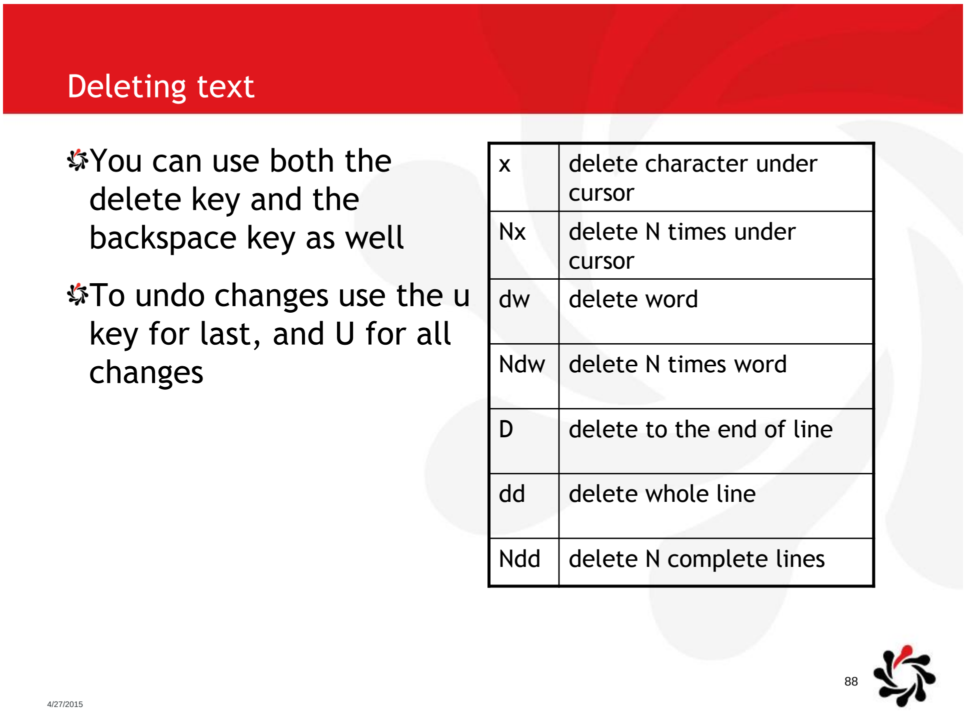
86



## Changing text

- Changing text puts vi into command mode.
- Escape can be hit at any time to cancel operation at current point
- The N character is a numeric multiplier to do the operation N times

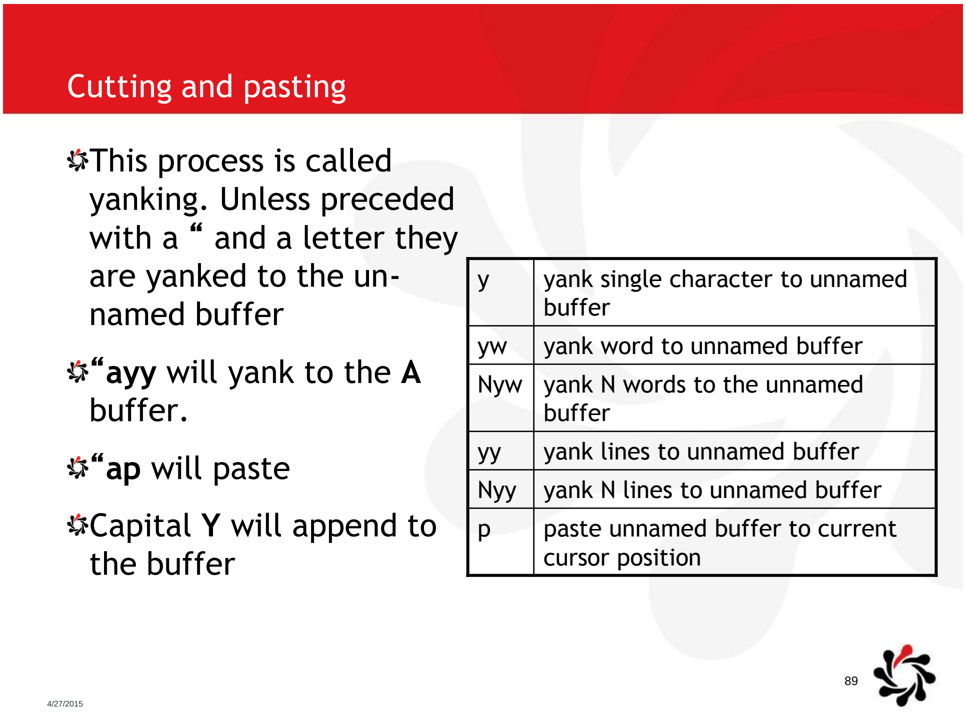
| | |
|---|---|
| r | replace single character |
| R | replace characters until escape is hit |
| cw | change word from current cursor |
| cNw | change 5 words from current cursor |
| C | change to the end of line from current cursor |
| cc | Change entire line |
| Ncc | Change N lines |



## Deleting text

- You can use both the delete key and the backspace key as well
- To undo changes use the u key for last, and U for all changes

| | |
|---|---|
| x | delete character under cursor |
| Nx | delete N times under cursor |
| dw | delete word |
| Ndw | delete N times word |
| D | delete to the end of line |
| dd | delete whole line |
| Ndd | delete N complete lines |



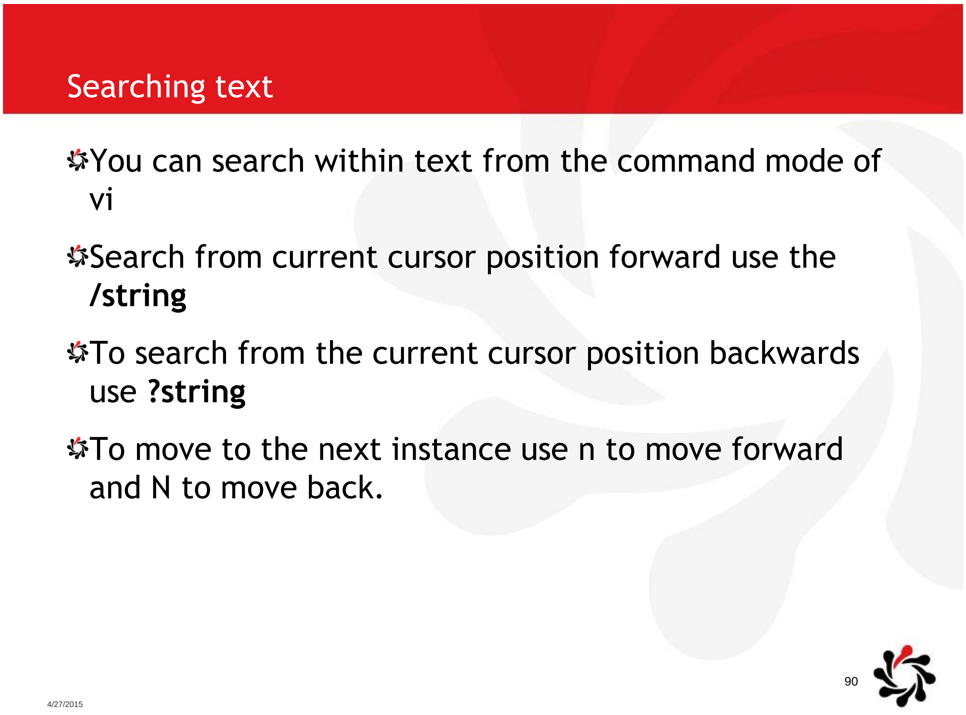
## Cutting and pasting

- This process is called yanking. Unless preceded with a “ and a letter they are yanked to the un-named buffer
- “**ayy** will yank to the **A** buffer.
- “**ap** will paste
- Capital **Y** will append to the buffer

| | |
|---|---|
| y | yank single character to unnamed buffer |
| yw | yank word to unnamed buffer |
| Nyw | yank N words to the unnamed buffer |
| yy | yank lines to unnamed buffer |
| Nyy | yank N lines to unnamed buffer |
| p | paste unnamed buffer to current cursor position |

4/27/2015

89

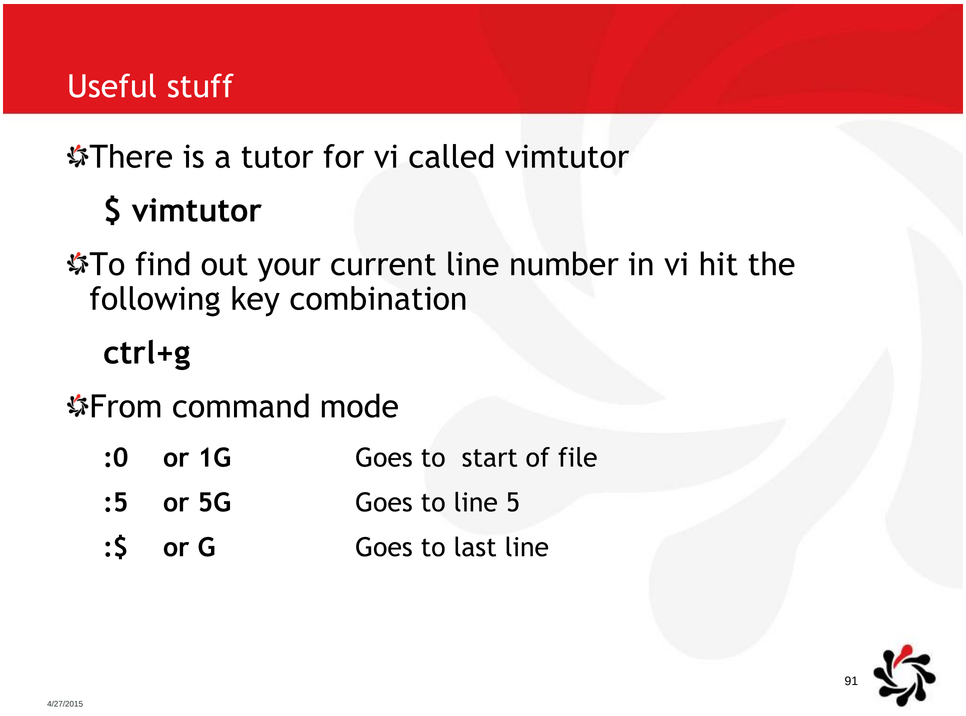


## Searching text

- You can search within text from the command mode of vi
- Search from current cursor position forward use the **/string**
- To search from the current cursor position backwards use **?string**
- To move to the next instance use n to move forward and N to move back.

4/27/2015

90



## Useful stuff

- There is a tutor for vi called vimtutor

  **$ vimtutor**

- To find out your current line number in vi hit the following key combination

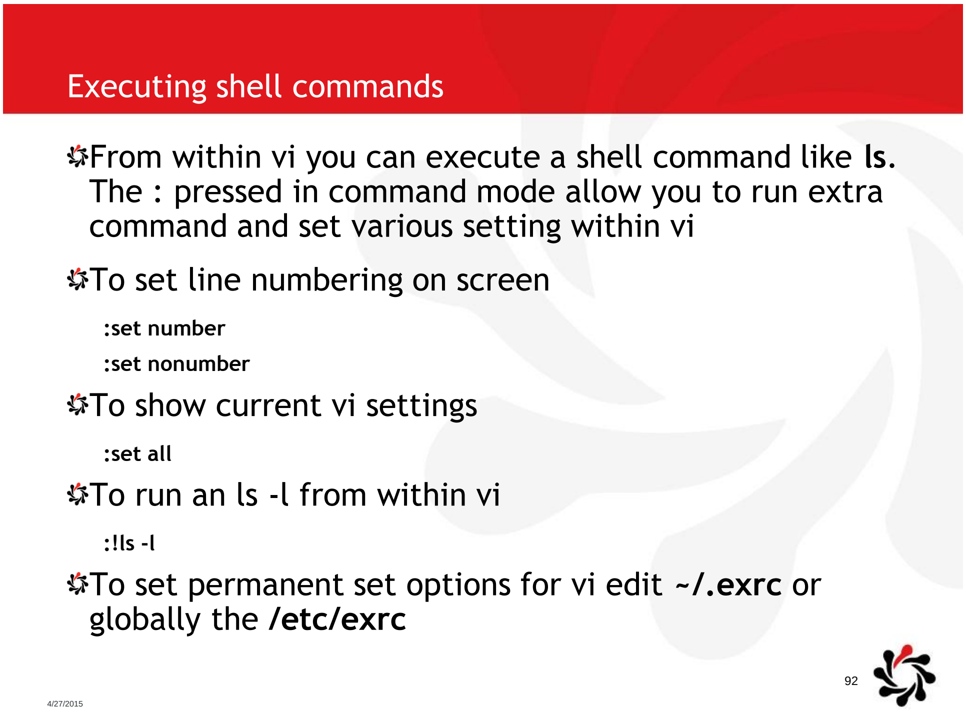
  **ctrl+g**

- From command mode

| | | |
|---|---|---|
| **:0** | **or 1G** | Goes to  start of file |
| **:5** | **or 5G** | Goes to line 5 |
| **:$** | **or G** | Goes to last line |

4/27/2015

91



## Executing shell commands

- From within vi you can execute a shell command like **ls**. The : pressed in command mode allow you to run extra command and set various setting within vi
- To set line numbering on screen

  **:set number**

  **:set nonumber**

- To show current vi settings

  **:set all**

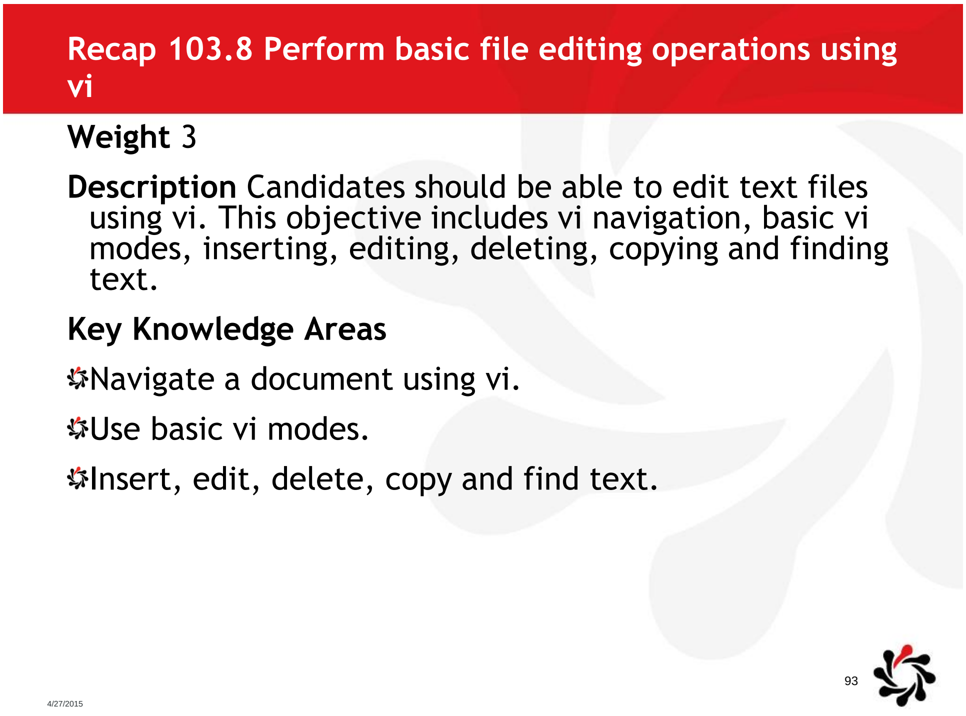
- To run an ls -l from within vi

  **:!ls -l**

- To set permanent set options for vi edit **~/.exrc** or globally the **/etc/exrc**

4/27/2015

92



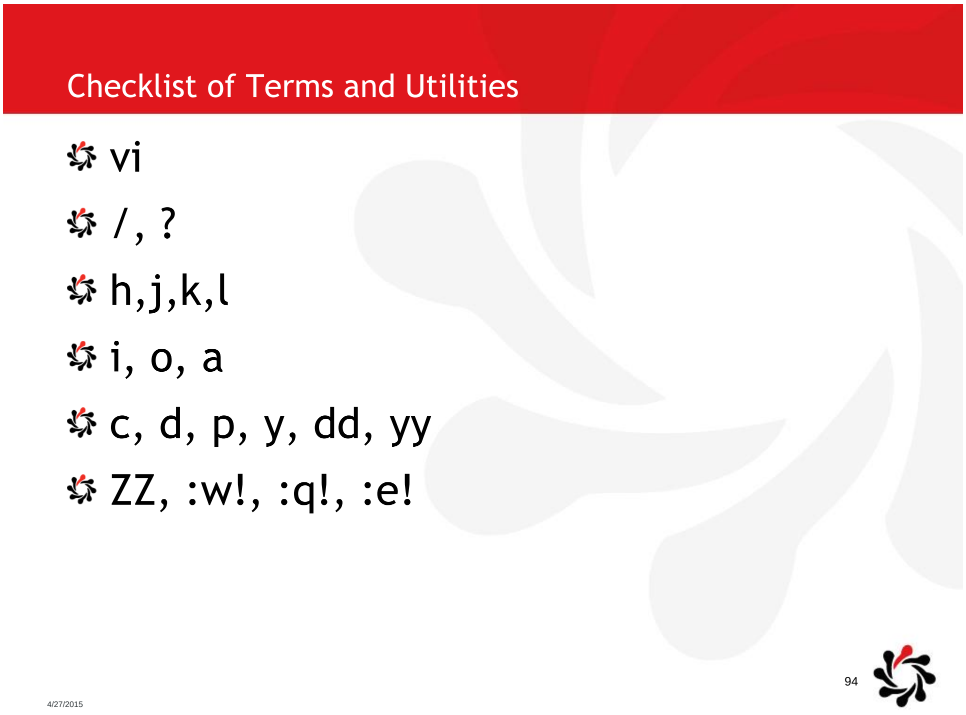
## Recap 103.8 Perform basic file editing operations using vi

**Weight** 3

**Description** Candidates should be able to edit text files using vi. This objective includes vi navigation, basic vi modes, inserting, editing, deleting, copying and finding text.

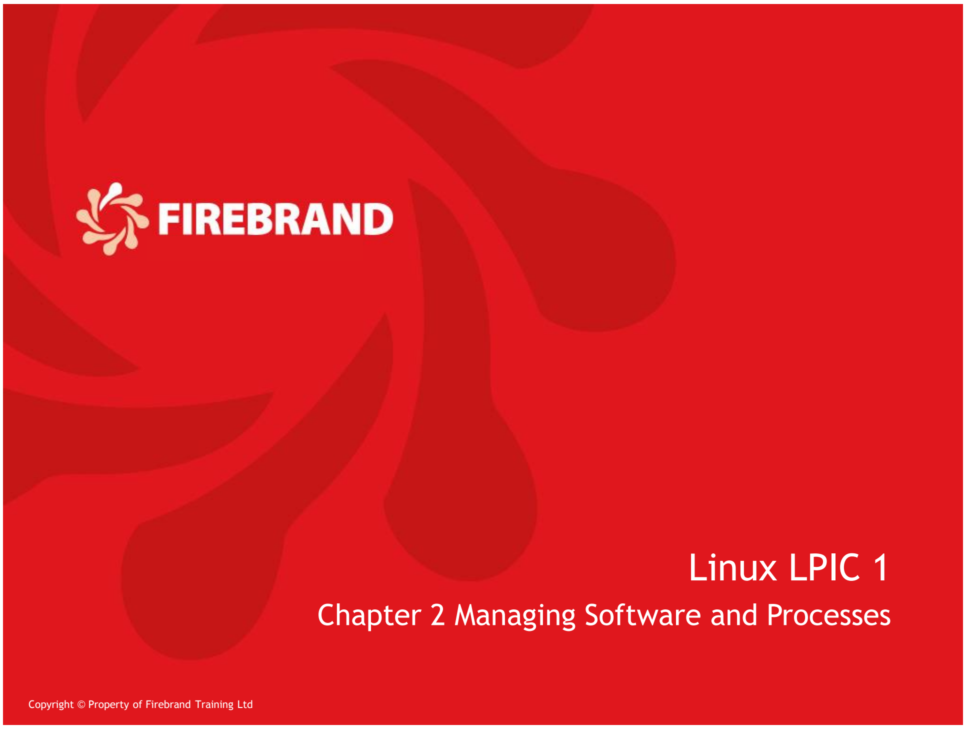
**Key Knowledge Areas**

- Navigate a document using vi.
- Use basic vi modes.
- Insert, edit, delete, copy and find text.



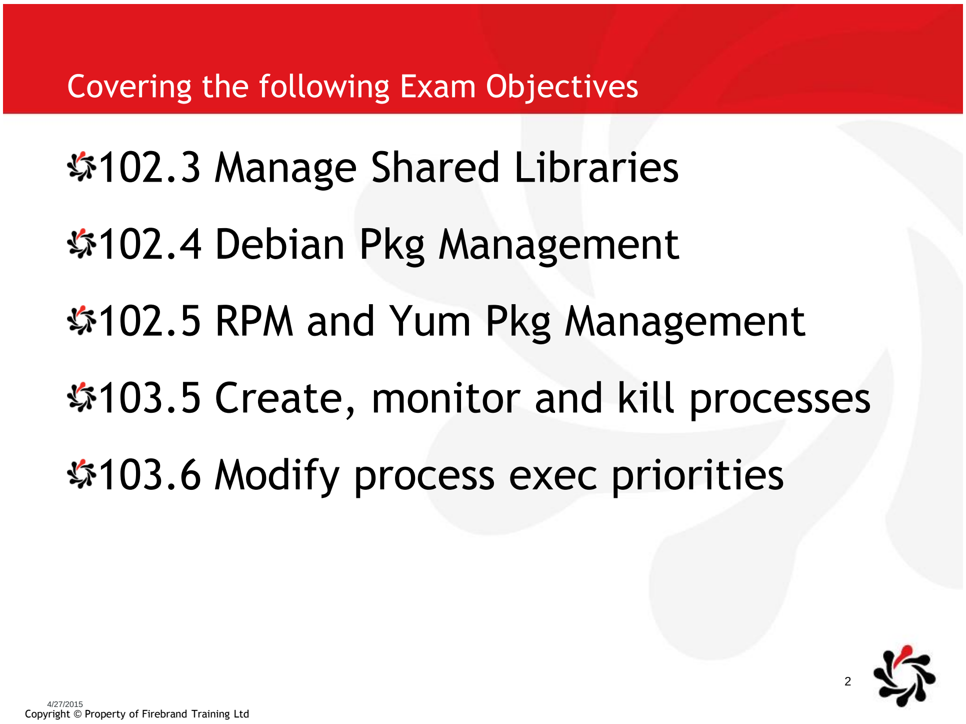
## Checklist of Terms and Utilities

- vi
- /, ?
- h,j,k,l
- i, o, a
- c, d, p, y, dd, yy
- ZZ, :w!, :q!, :e!



# Linux LPIC 1

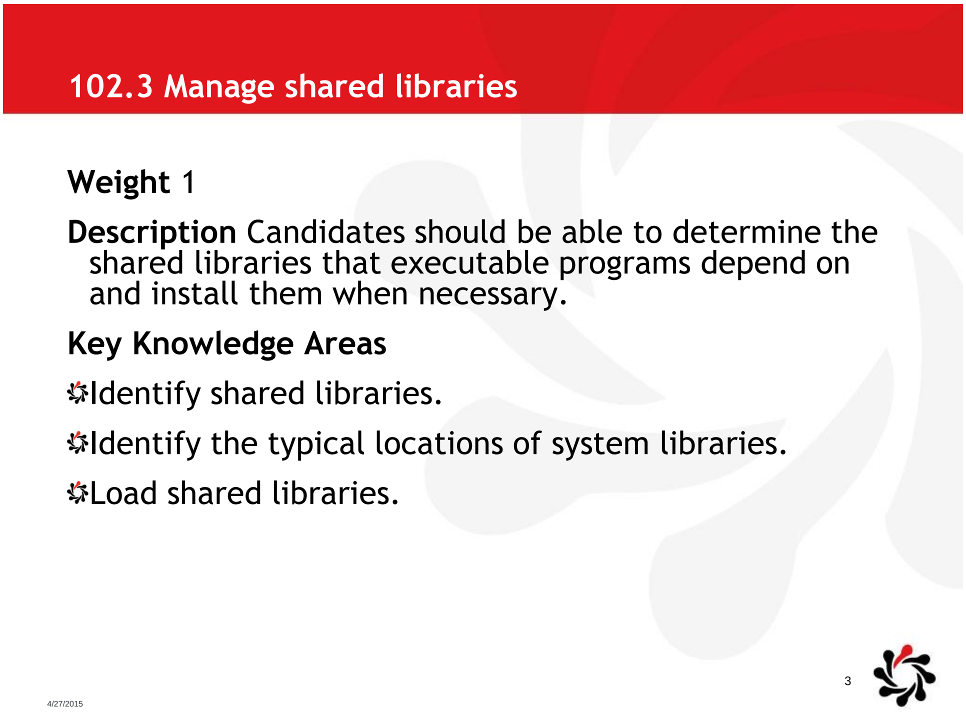
## Chapter 2 Managing Software and Processes



## Covering the following Exam Objectives

- 102.3 Manage Shared Libraries
- 102.4 Debian Pkg Management
- 102.5 RPM and Yum Pkg Management
- 103.5 Create, monitor and kill processes
- 103.6 Modify process exec priorities

4/27/2015



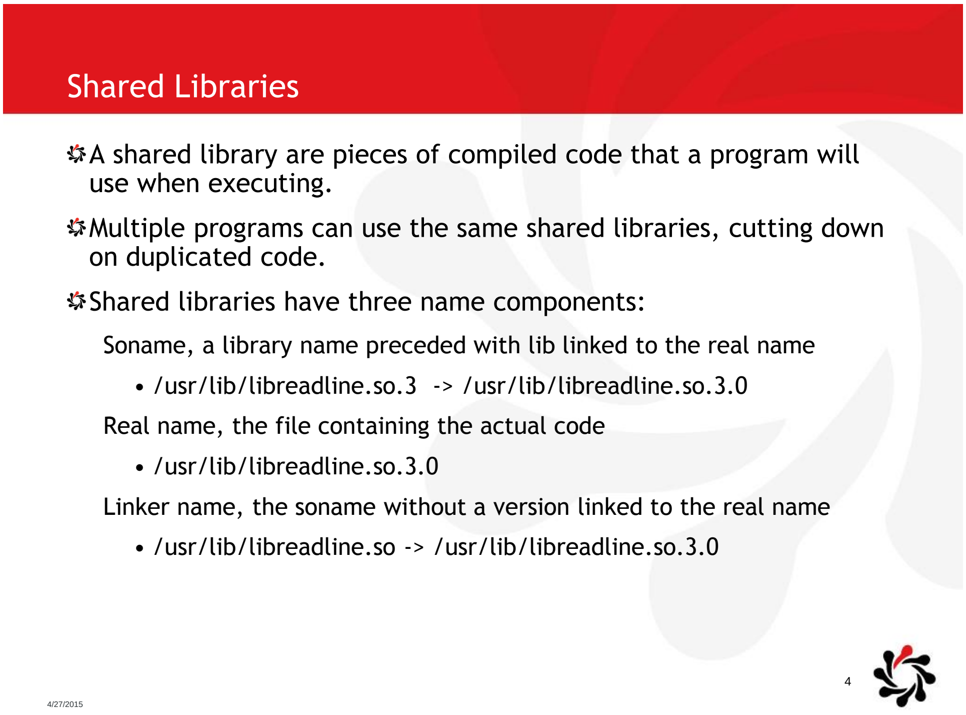
## 102.3 Manage shared libraries

**Weight** 1

**Description** Candidates should be able to determine the shared libraries that executable programs depend on and install them when necessary.

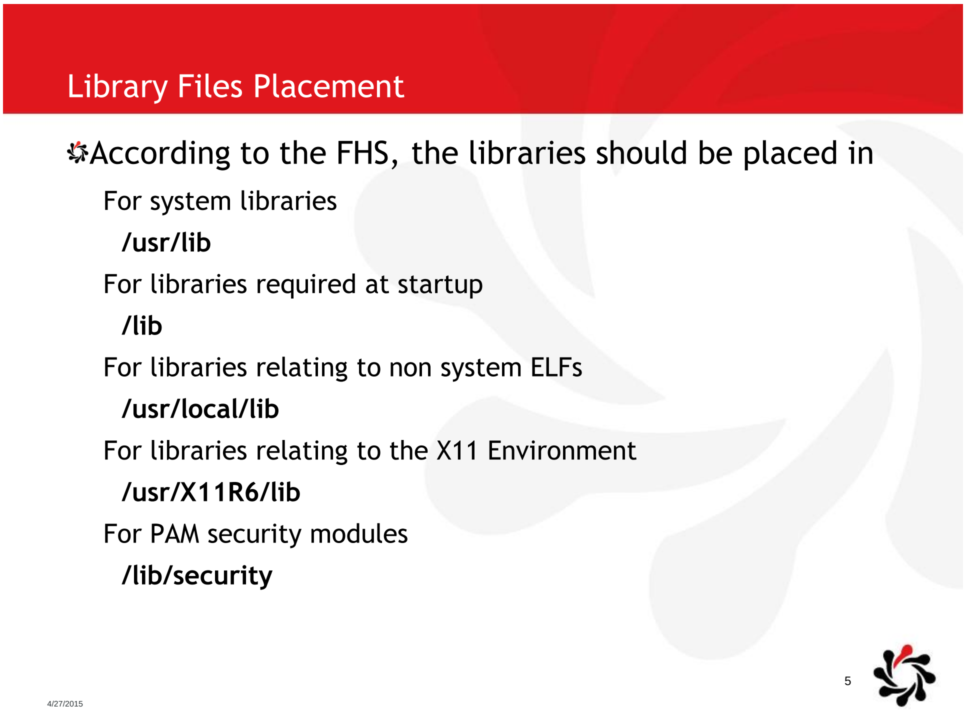
**Key Knowledge Areas**

- Identify shared libraries.
- Identify the typical locations of system libraries.
- Load shared libraries.



## Shared Libraries

- A shared library are pieces of compiled code that a program will use when executing.
- Multiple programs can use the same shared libraries, cutting down on duplicated code.
- Shared libraries have three name components:
  - Soname, a library name preceded with lib linked to the real name
    - /usr/lib/libreadline.so.3 -> /usr/lib/libreadline.so.3.0
  - Real name, the file containing the actual code
    - /usr/lib/libreadline.so.3.0
  - Linker name, the soname without a version linked to the real name
    - /usr/lib/libreadline.so -> /usr/lib/libreadline.so.3.0

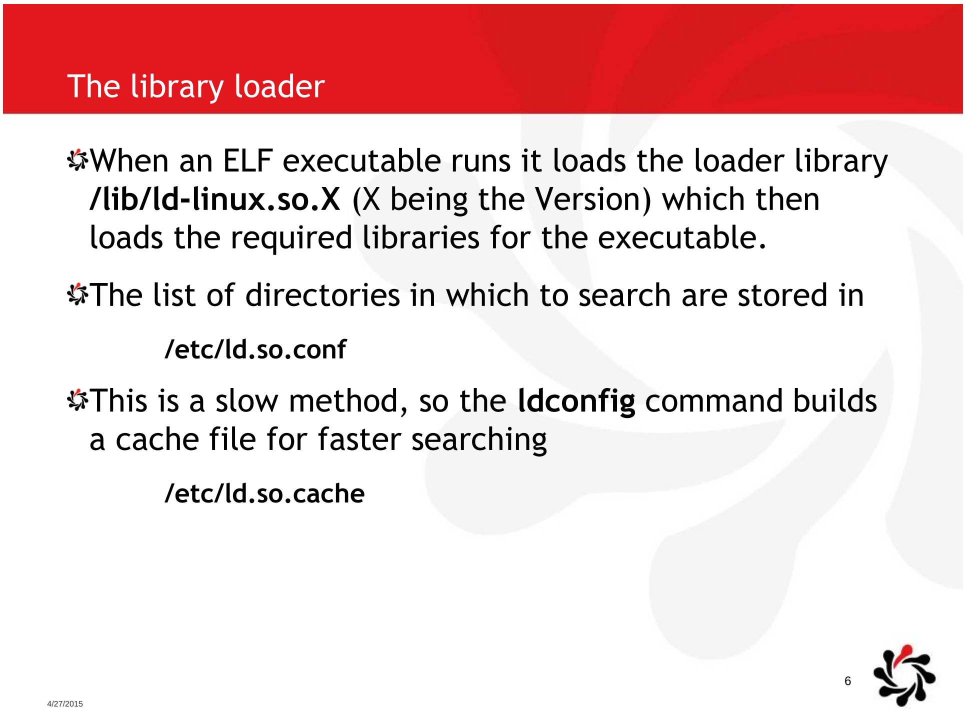


## Library Files Placement

- According to the FHS, the libraries should be placed in
  - For system libraries
    - **/usr/lib**
  - For libraries required at startup
    - **/lib**
  - For libraries relating to non system ELFs
    - **/usr/local/lib**
  - For libraries relating to the X11 Environment
    - **/usr/X11R6/lib**
  - For PAM security modules
    - **/lib/security**

4/27/2015

5

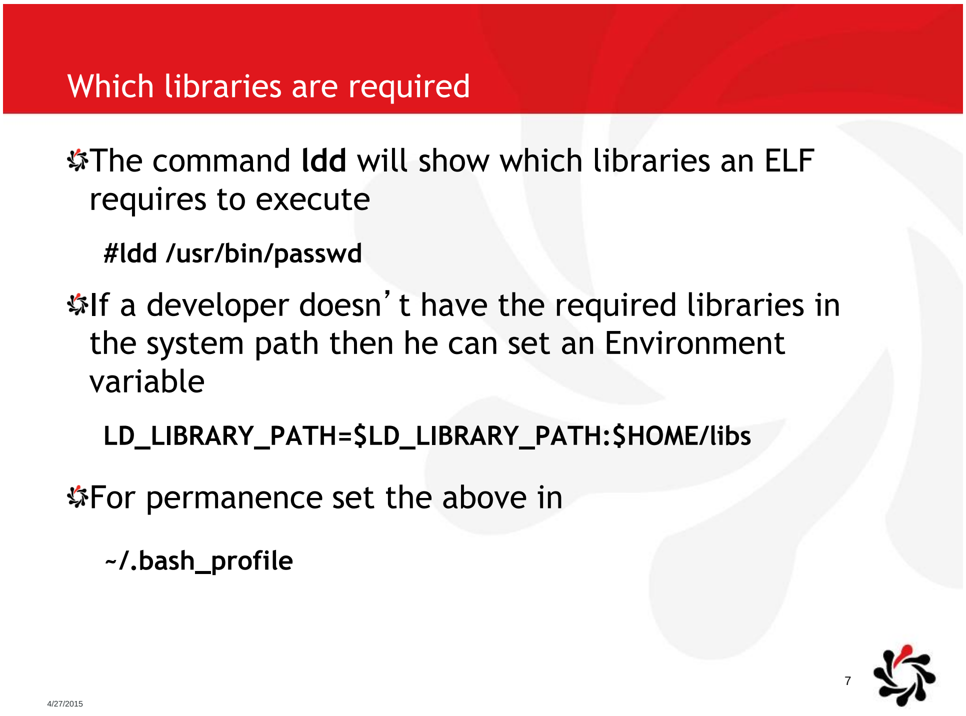


## The library loader

- When an ELF executable runs it loads the loader library **/lib/ld-linux.so.X** (X being the Version) which then loads the required libraries for the executable.
- The list of directories in which to search are stored in
  - **/etc/ld.so.conf**
- This is a slow method, so the **ldconfig** command builds a cache file for faster searching
  - **/etc/ld.so.cache**

4/27/2015

6



## Which libraries are required

- The command **ldd** will show which libraries an ELF requires to execute

```
#ldd /usr/bin/passwd
```

- If a developer doesn’t have the required libraries in the system path then he can set an Environment variable

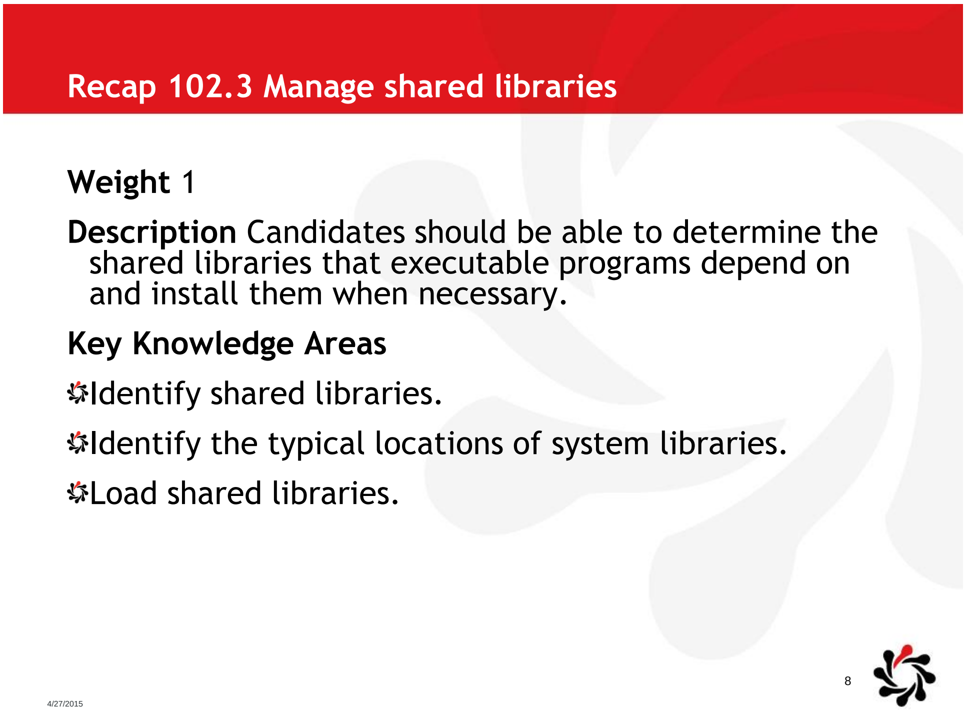
```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/libs
```

- For permanence set the above in

```
~/.bash_profile
```

7

4/27/2015



## Recap 102.3 Manage shared libraries

**Weight** 1

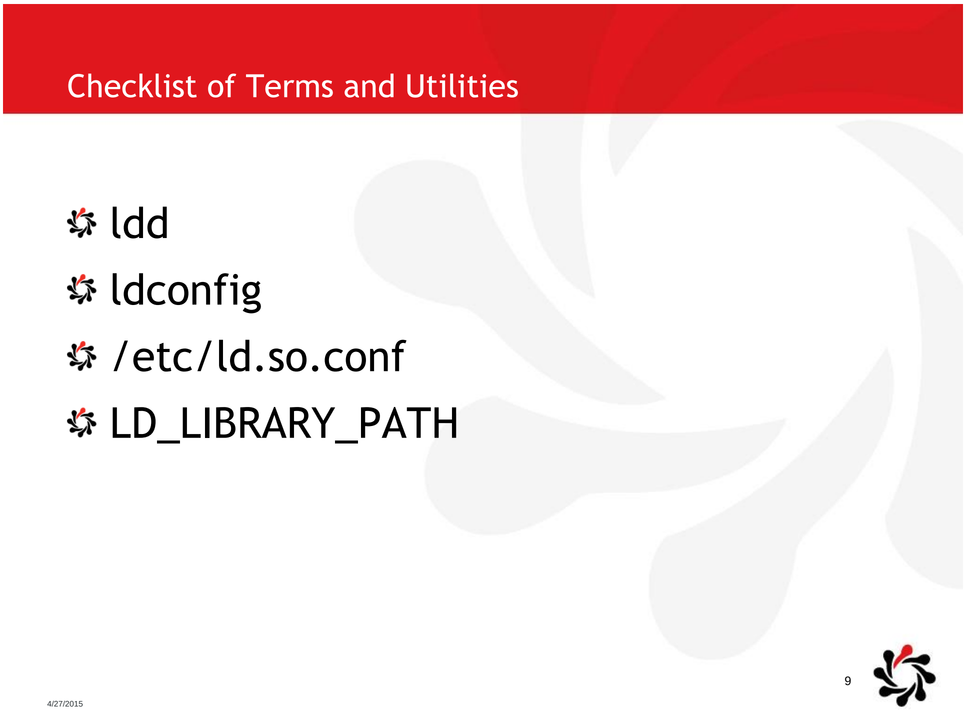
**Description** Candidates should be able to determine the shared libraries that executable programs depend on and install them when necessary.

**Key Knowledge Areas**

- Identify shared libraries.
- Identify the typical locations of system libraries.
- Load shared libraries.

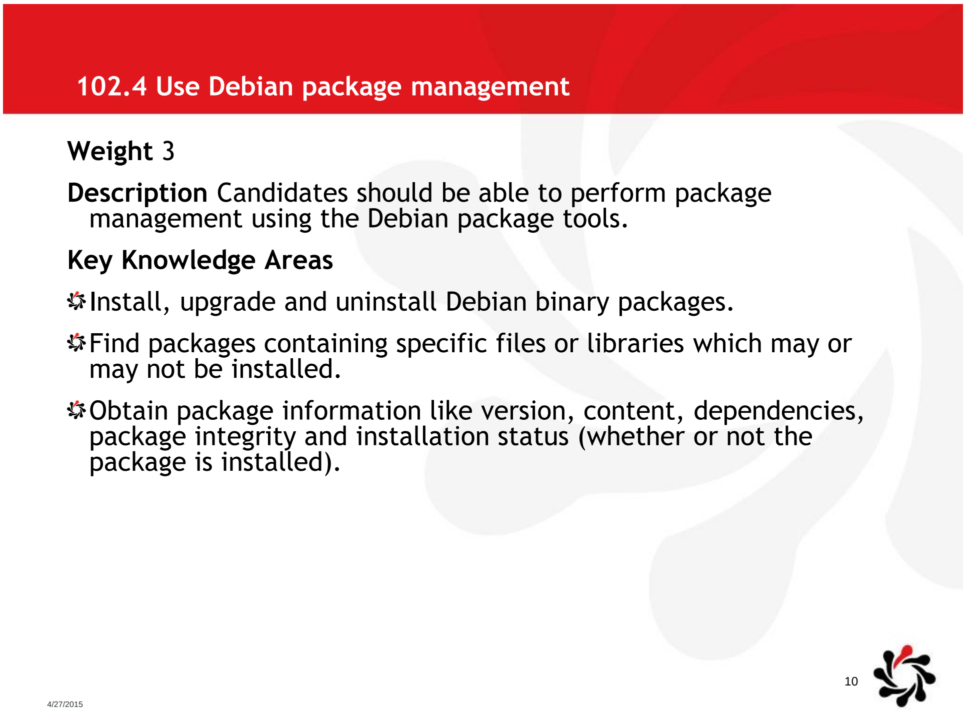
8

4/27/2015



## Checklist of Terms and Utilities

- ldd
- ldconfig
- /etc/ld.so.conf
- LD_LIBRARY_PATH



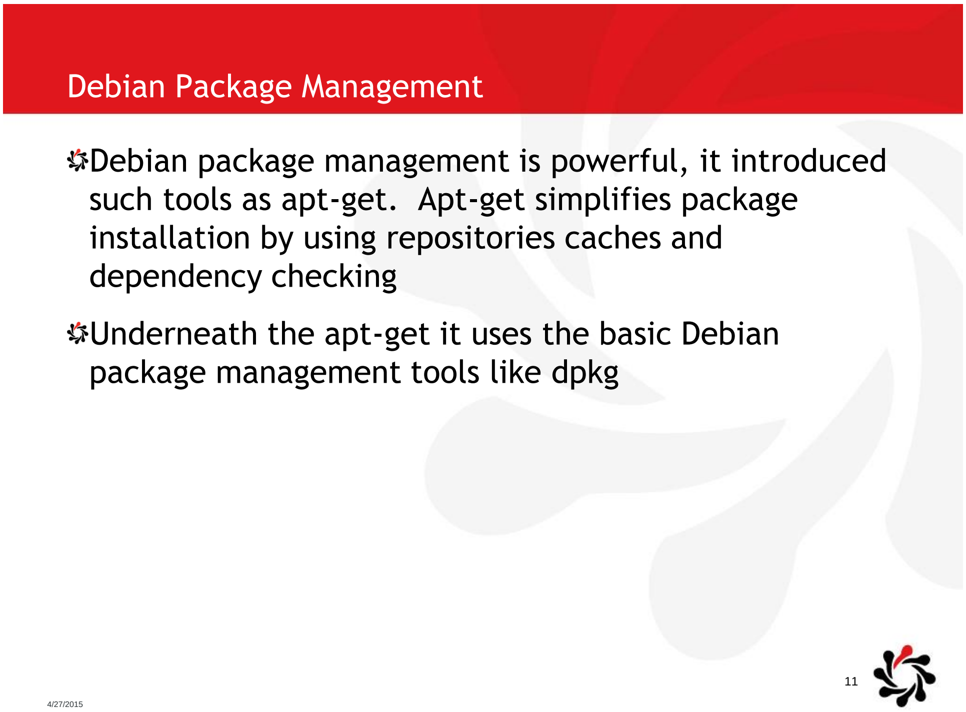
## 102.4 Use Debian package management

**Weight** 3

**Description** Candidates should be able to perform package management using the Debian package tools.

**Key Knowledge Areas**

- Install, upgrade and uninstall Debian binary packages.
- Find packages containing specific files or libraries which may or may not be installed.
- Obtain package information like version, content, dependencies, package integrity and installation status (whether or not the package is installed).

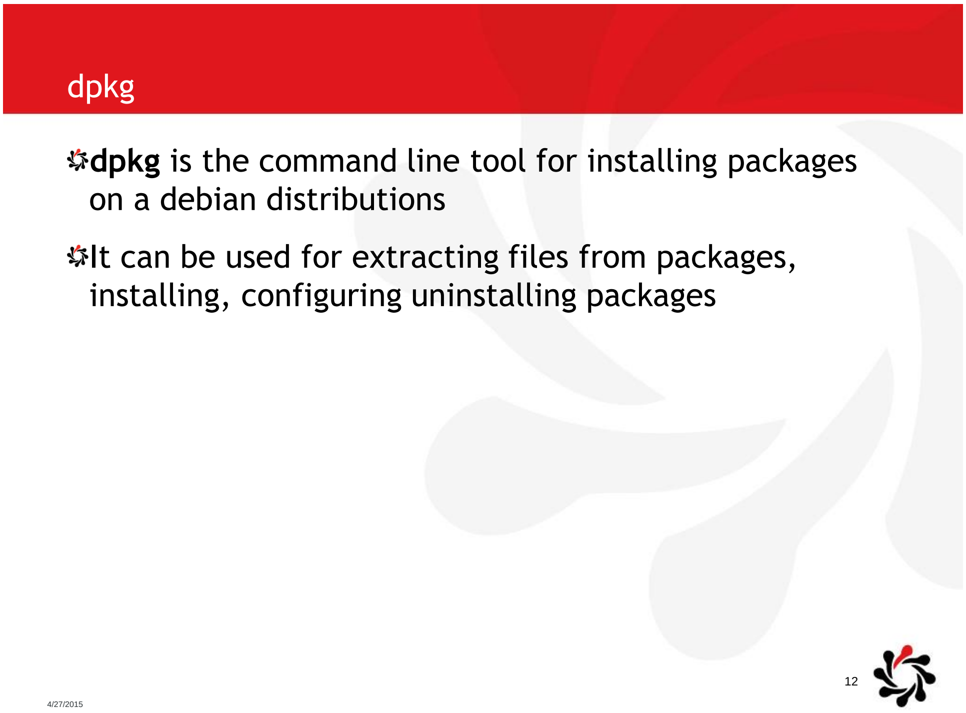


## Debian Package Management

- Debian package management is powerful, it introduced such tools as apt-get. Apt-get simplifies package installation by using repositories caches and dependency checking
- Underneath the apt-get it uses the basic Debian package management tools like dpkg

11

4/27/2015

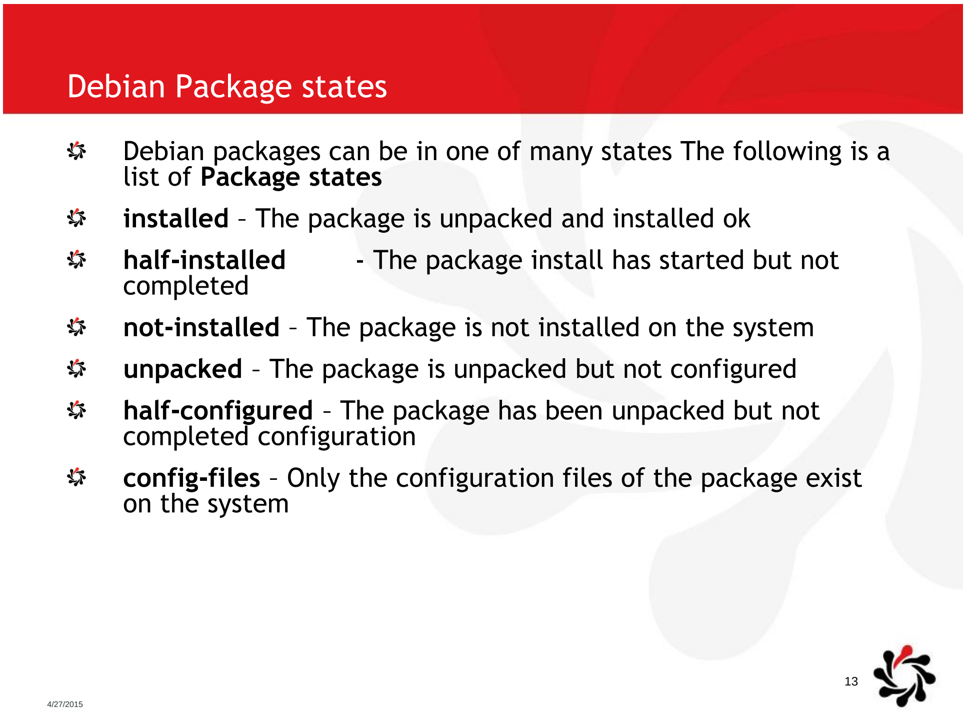


## dpkg

- **dpkg** is the command line tool for installing packages on a debian distributions
- It can be used for extracting files from packages, installing, configuring uninstalling packages

12

4/27/2015

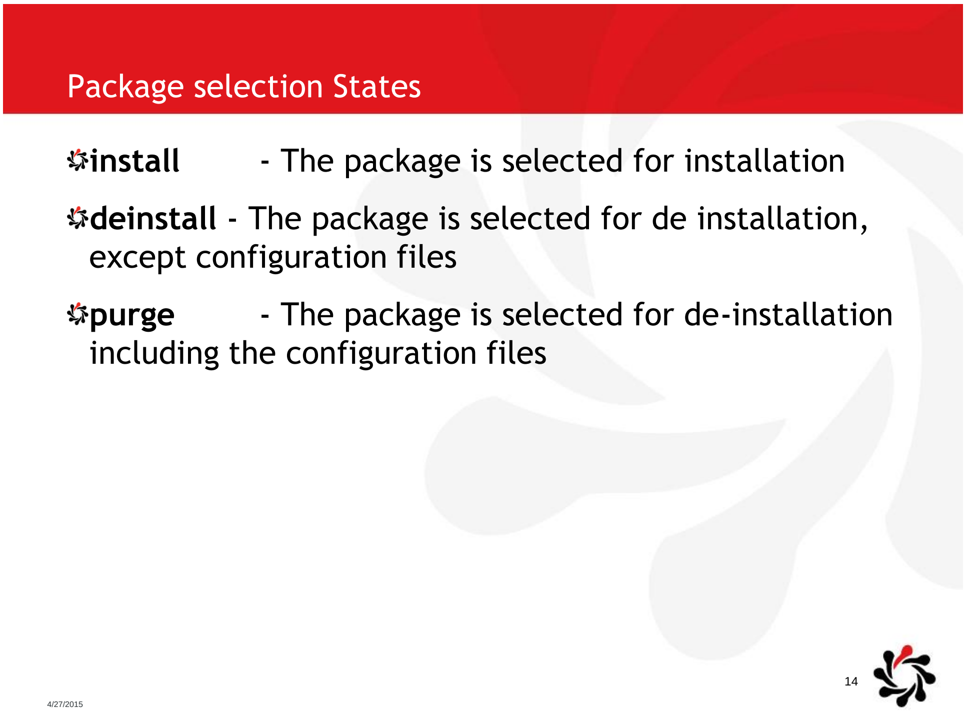


## Debian Package states

- Debian packages can be in one of many states The following is a list of **Package states**
- **installed** – The package is unpacked and installed ok
- **half-installed** - The package install has started but not completed
- **not-installed** – The package is not installed on the system
- **unpacked** – The package is unpacked but not configured
- **half-configured** – The package has been unpacked but not completed configuration
- **config-files** – Only the configuration files of the package exist on the system

4/27/2015

13

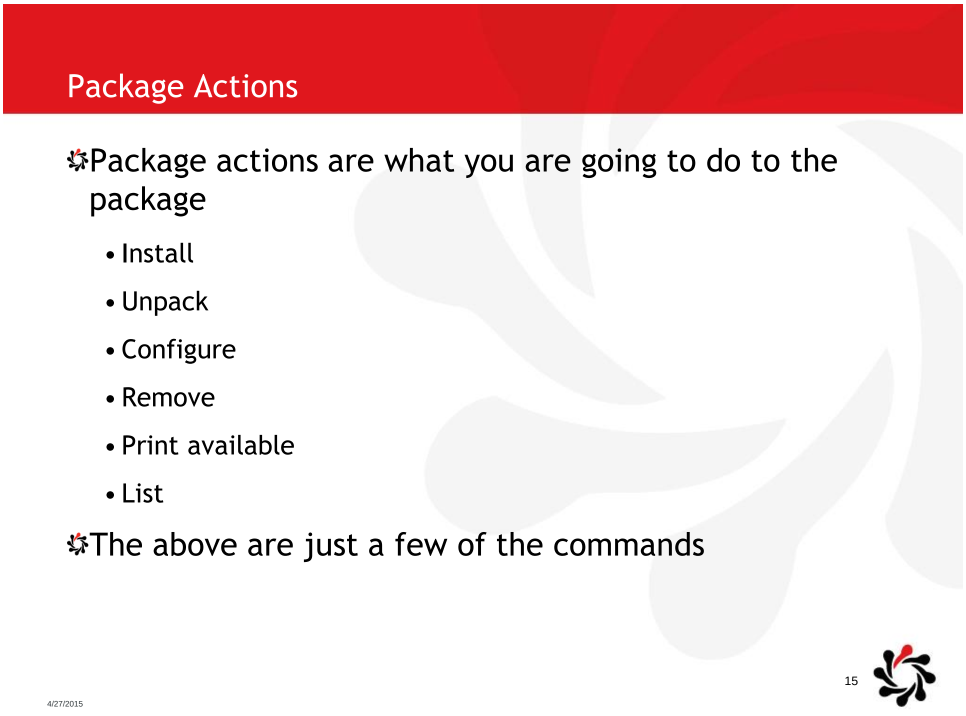


## Package selection States

- **install** - The package is selected for installation
- **deinstall** - The package is selected for de installation, except configuration files
- **purge** - The package is selected for de-installation including the configuration files

4/27/2015

14

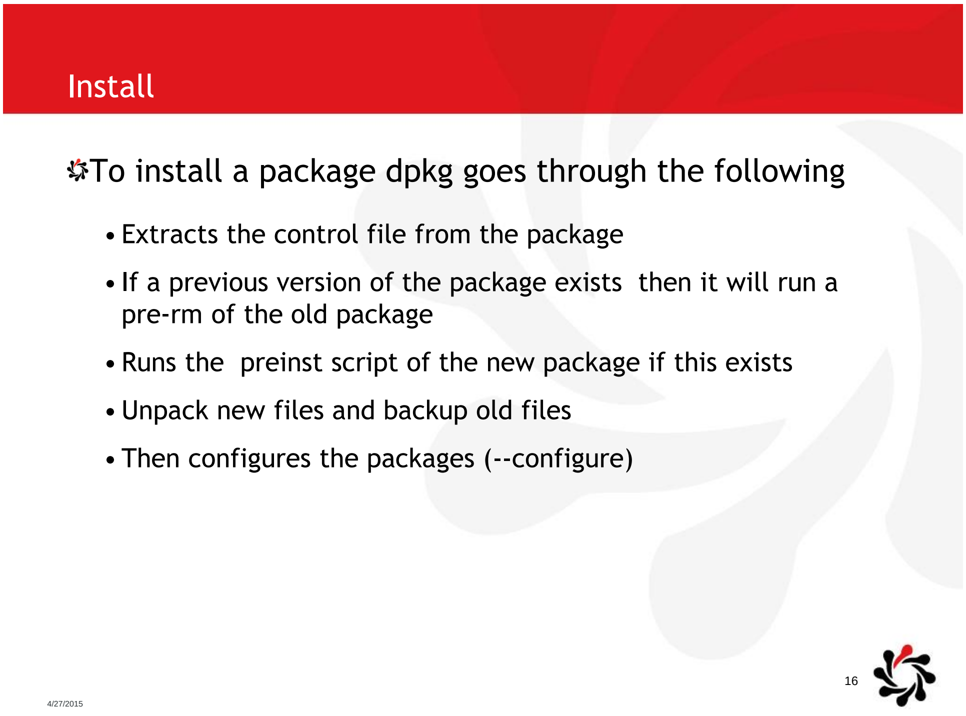


## Package Actions

- Package actions are what you are going to do to the package
  - Install
  - Unpack
  - Configure
  - Remove
  - Print available
  - List
- The above are just a few of the commands

4/27/2015

15

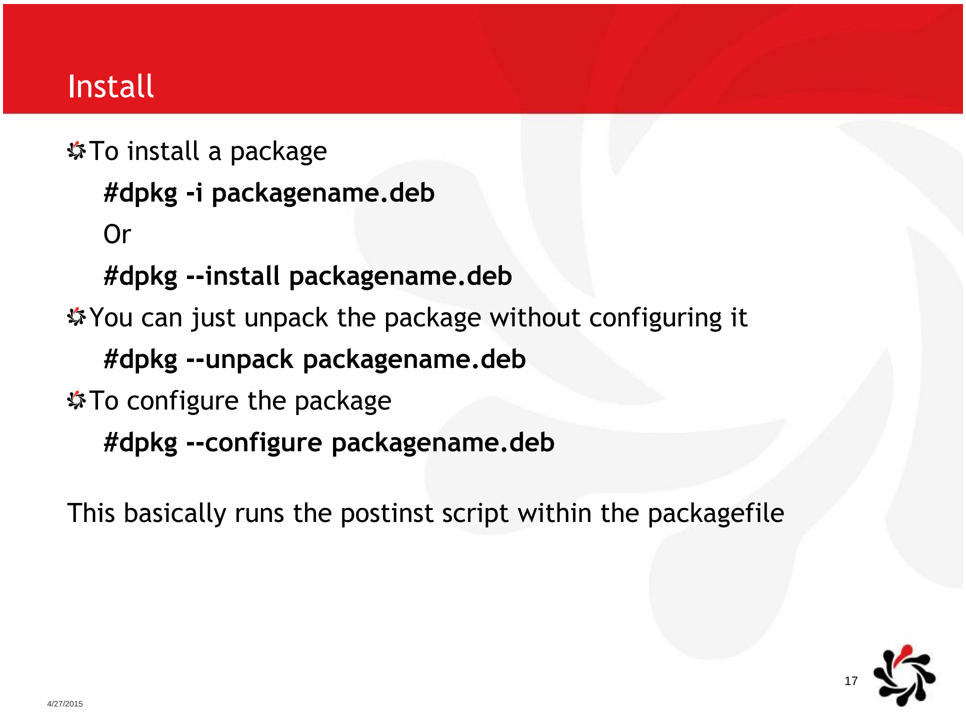


## Install

- To install a package dpkg goes through the following
  - Extracts the control file from the package
  - If a previous version of the package exists then it will run a pre-rm of the old package
  - Runs the preinst script of the new package if this exists
  - Unpack new files and backup old files
  - Then configures the packages (--configure)

4/27/2015

16



## Install

- To install a package

  **#dpkg -i packagename.deb**

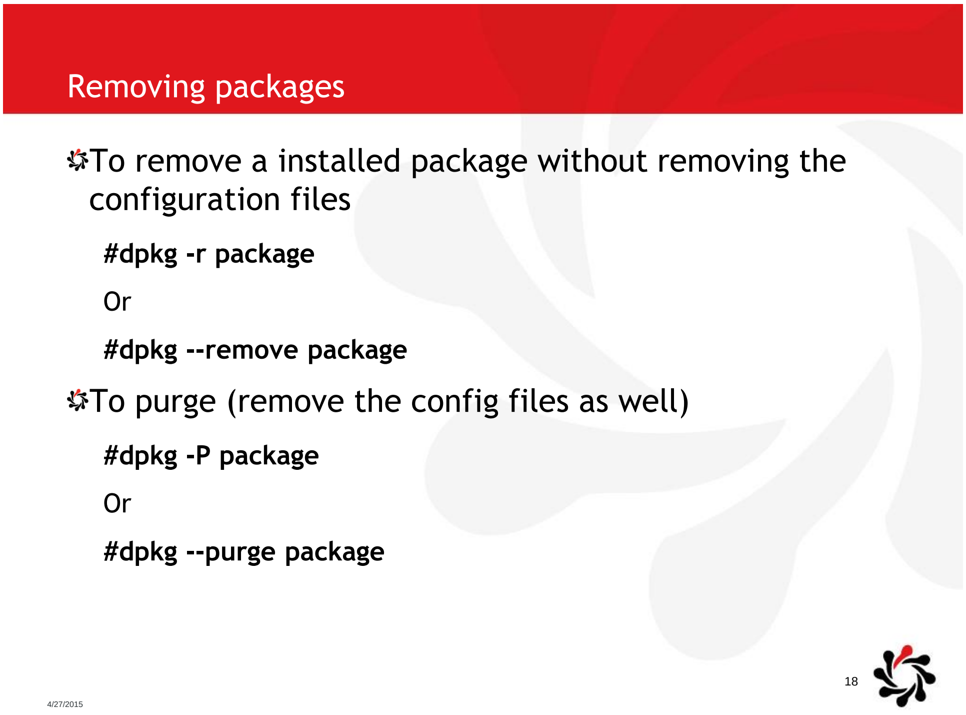
  Or

  **#dpkg --install packagename.deb**
- You can just unpack the package without configuring it

  **#dpkg --unpack packagename.deb**
- To configure the package

  **#dpkg --configure packagename.deb**

This basically runs the postinst script within the packagefile



## Removing packages

- To remove a installed package without removing the configuration files

  **#dpkg -r package**

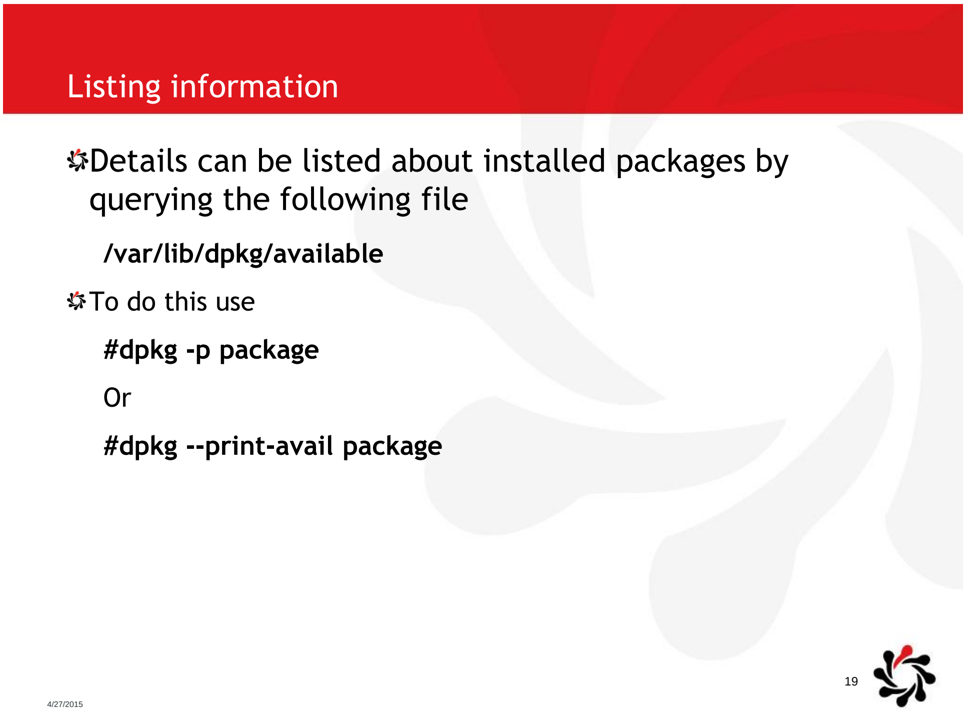
  Or

  **#dpkg --remove package**
- To purge (remove the config files as well)

  **#dpkg -P package**

  Or

  **#dpkg --purge package**



## Listing information

- Details can be listed about installed packages by querying the following file

  **/var/lib/dpkg/available**

- To do this use

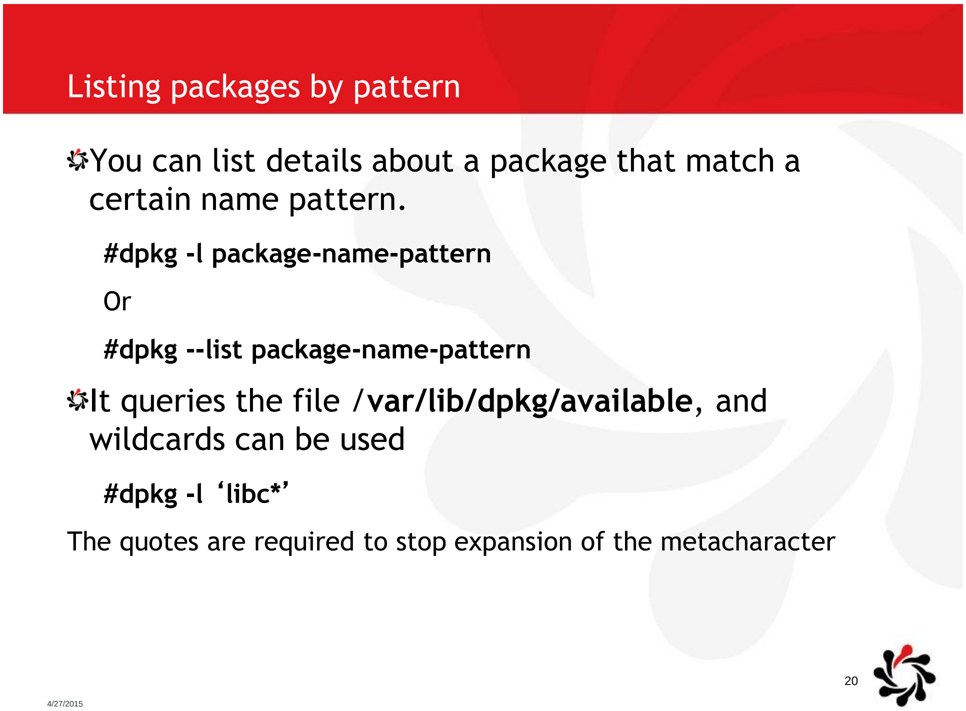
```
#dpkg -p package
```

  Or

```
#dpkg --print-avail package
```

4/27/2015

19



## Listing packages by pattern

- You can list details about a package that match a certain name pattern.

```
#dpkg -l package-name-pattern
```

  Or

```
#dpkg --list package-name-pattern
```

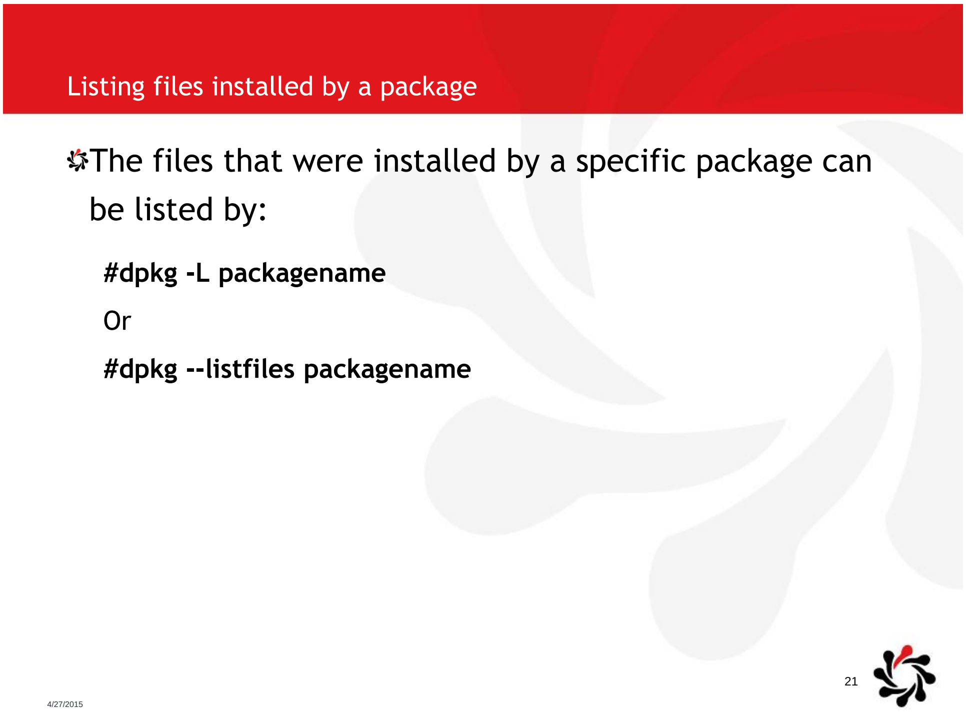
- It queries the file /**var/lib/dpkg/available**, and wildcards can be used

```
#dpkg -l ‘libc*’
```

The quotes are required to stop expansion of the metacharacter

4/27/2015

20



## Listing files installed by a package

- The files that were installed by a specific package can be listed by:

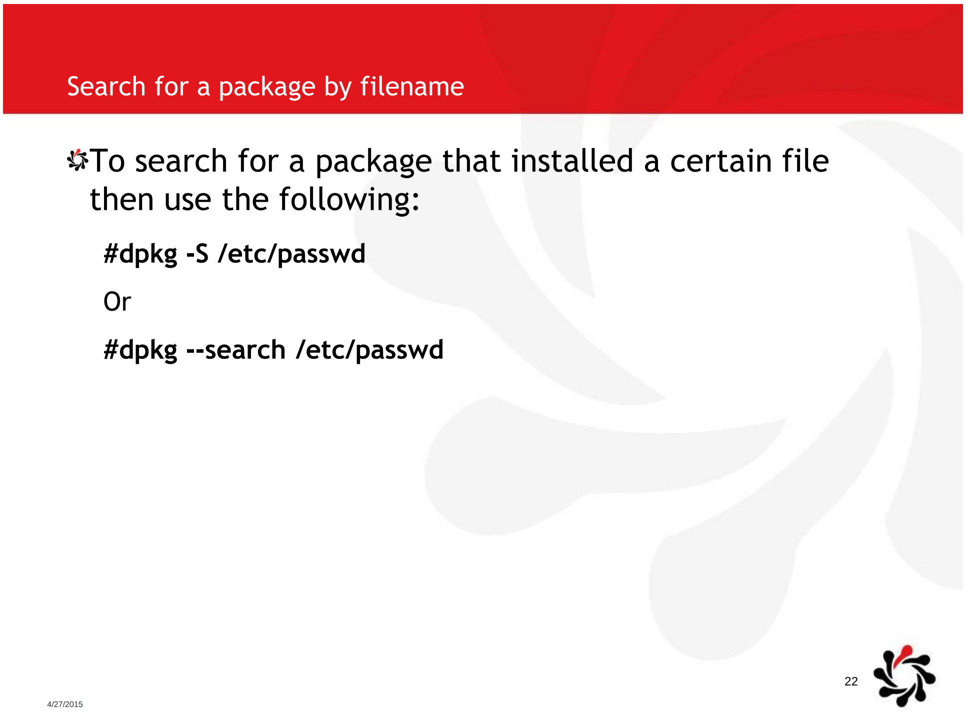
**#dpkg -L packagename**

Or

**#dpkg --listfiles packagename**

4/27/2015

21



## Search for a package by filename

- To search for a package that installed a certain file then use the following:

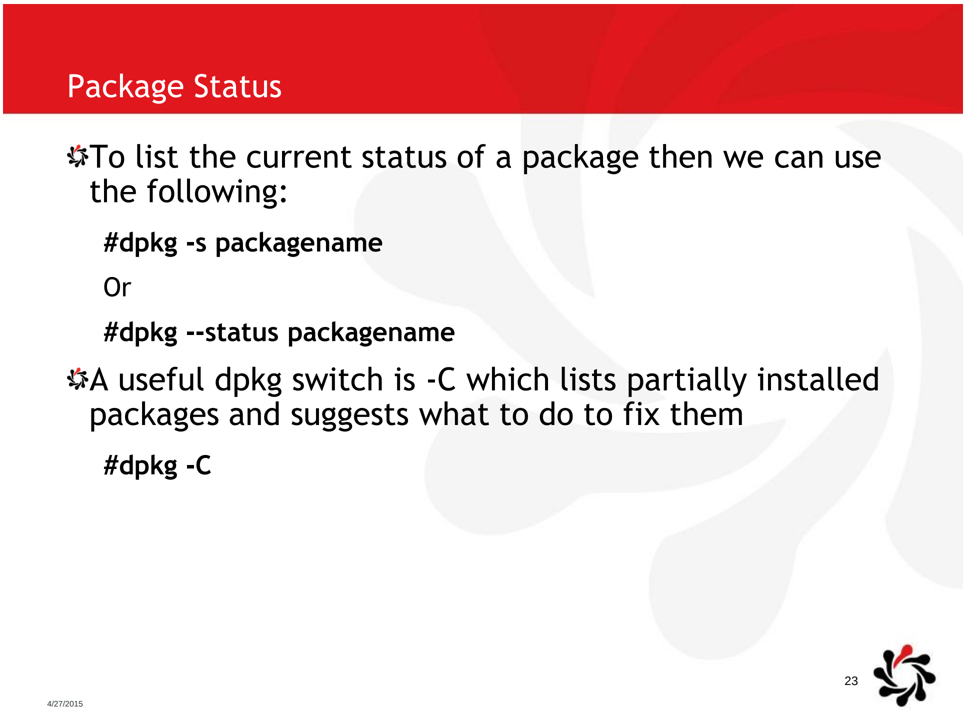
**#dpkg -S /etc/passwd**

Or

**#dpkg --search /etc/passwd**

4/27/2015

22



## Package Status

- To list the current status of a package then we can use the following:

  **#dpkg -s packagename**

  Or

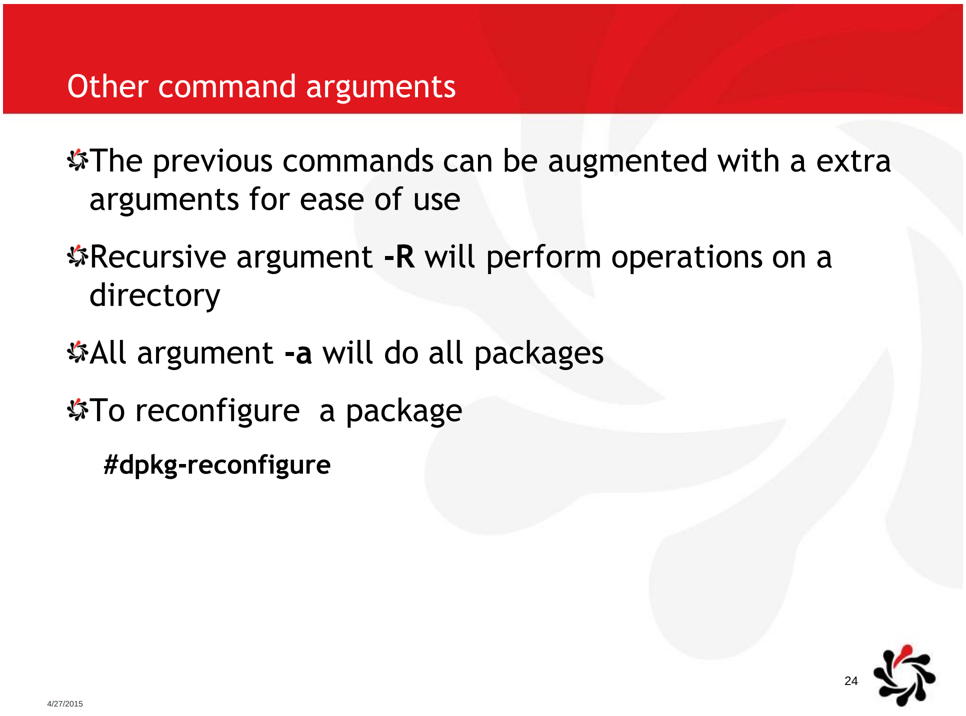
  **#dpkg --status packagename**

- A useful dpkg switch is -C which lists partially installed packages and suggests what to do to fix them

  **#dpkg -C**

4/27/2015

23



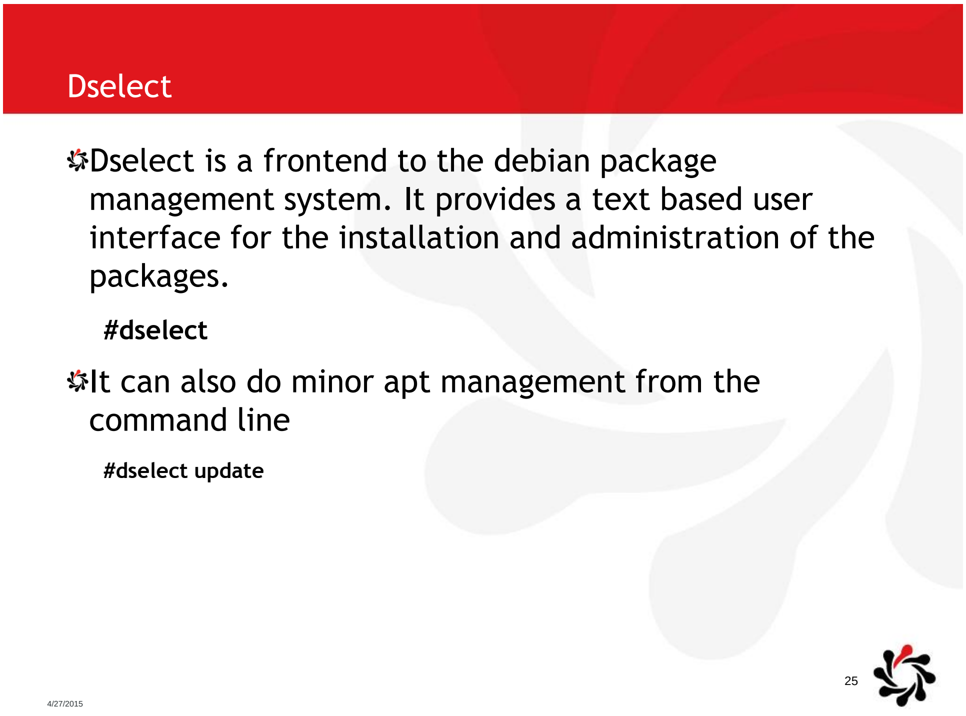
## Other command arguments

- The previous commands can be augmented with a extra arguments for ease of use
- Recursive argument **-R** will perform operations on a directory
- All argument **-a** will do all packages
- To reconfigure  a package

  **#dpkg-reconfigure**

4/27/2015

24



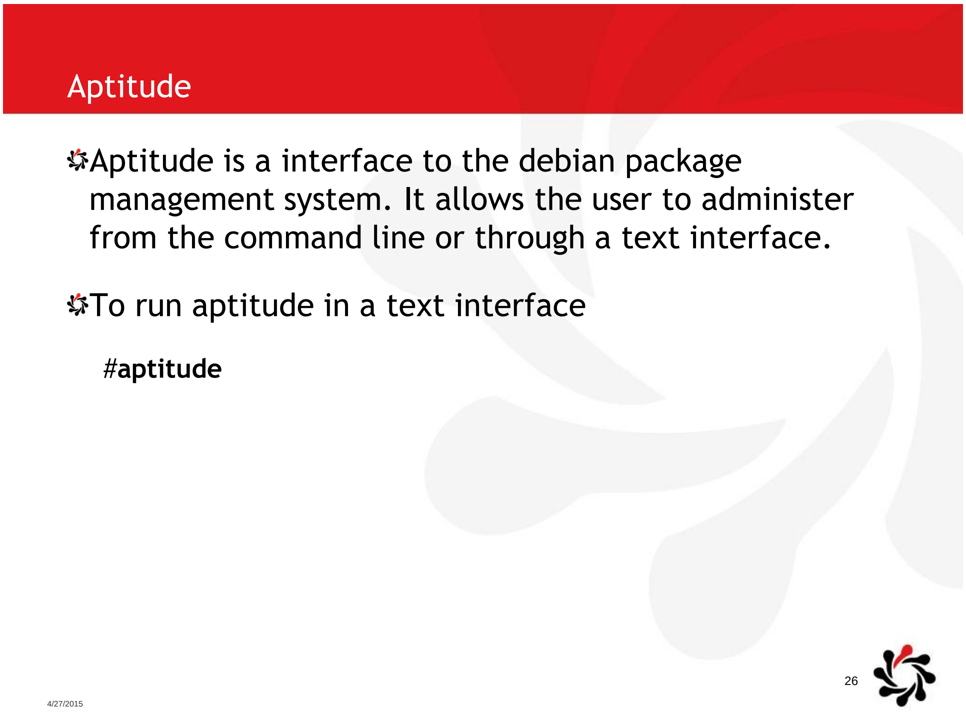
## Dselect

- Dselect is a frontend to the debian package management system. It provides a text based user interface for the installation and administration of the packages.

```
#dselect
```

- It can also do minor apt management from the command line

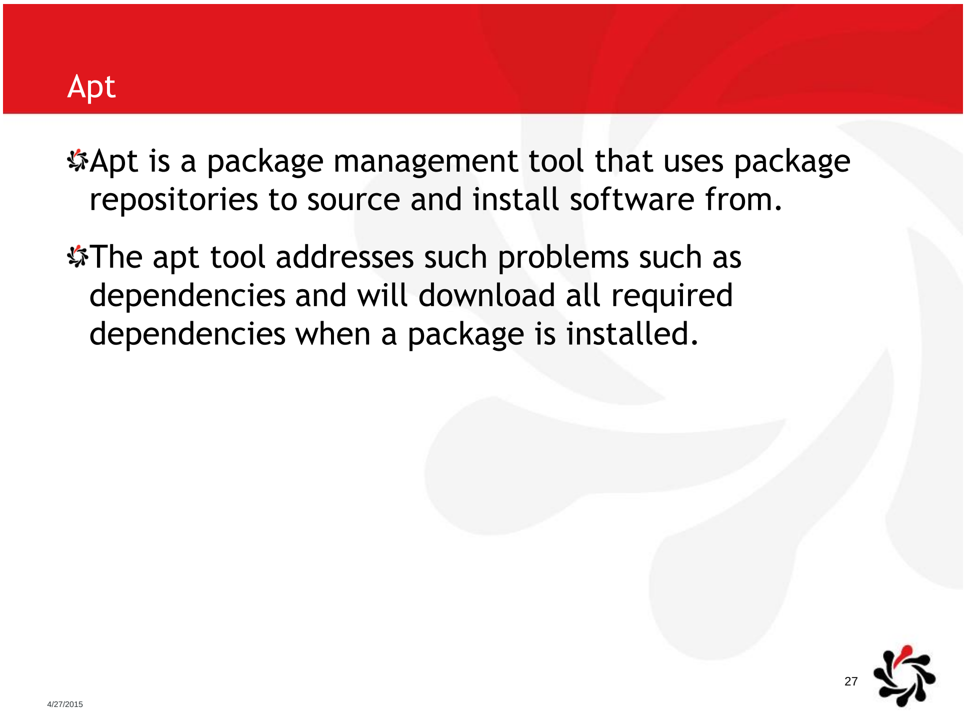
```
#dselect update
```



## Aptitude

- Aptitude is a interface to the debian package management system. It allows the user to administer from the command line or through a text interface.
- To run aptitude in a text interface

```
#aptitude
```

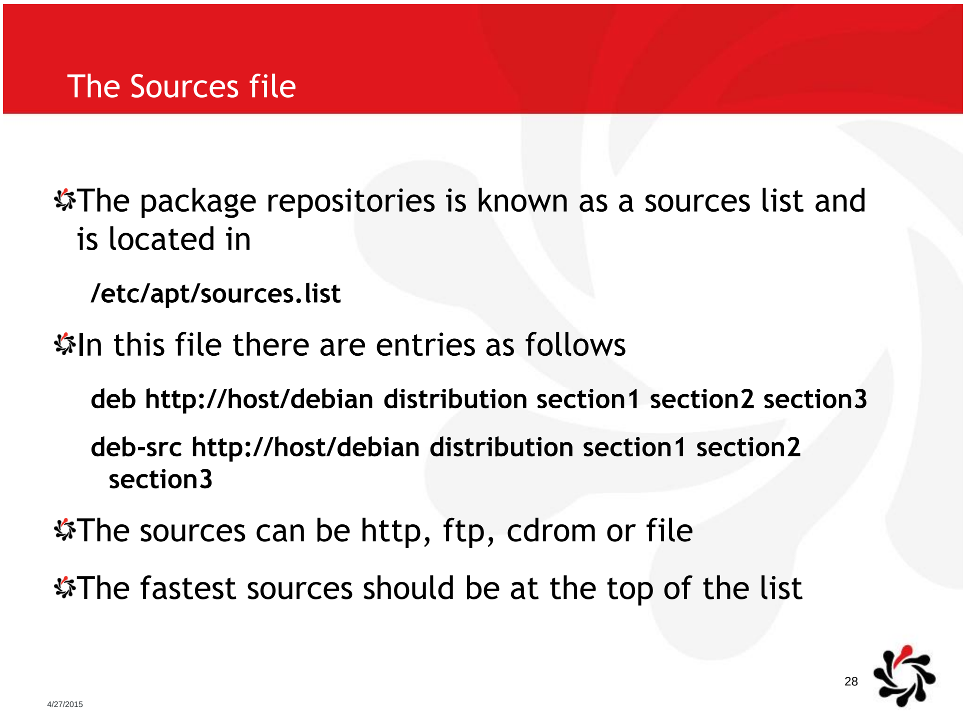


## Apt

- Apt is a package management tool that uses package repositories to source and install software from.
- The apt tool addresses such problems such as dependencies and will download all required dependencies when a package is installed.

27

4/27/2015



## The Sources file

- The package repositories is known as a sources list and is located in

  **/etc/apt/sources.list**

- In this file there are entries as follows

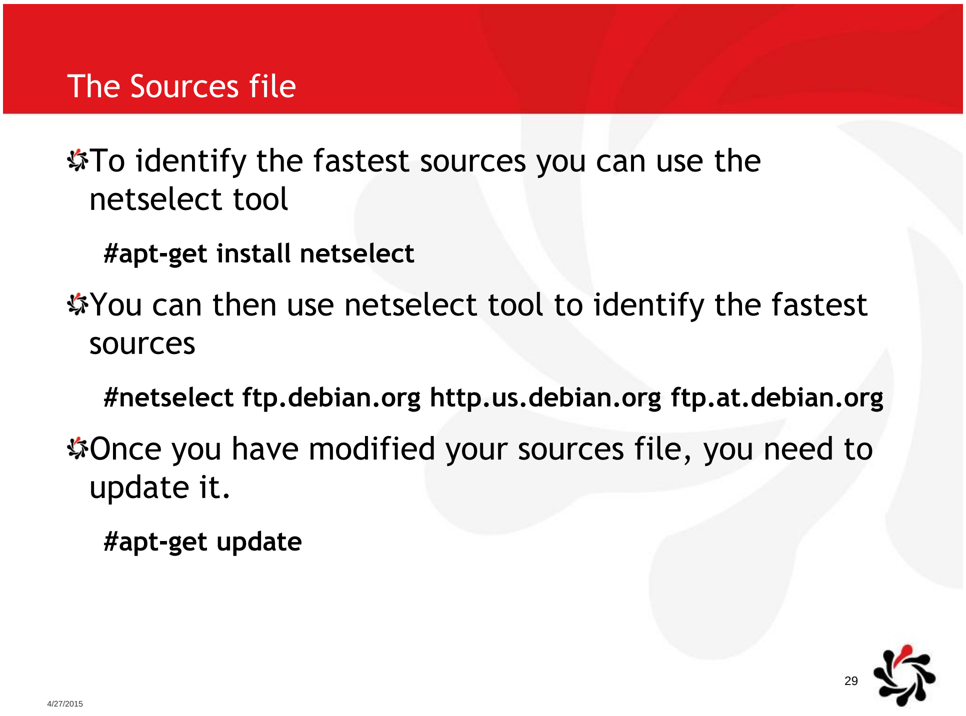
  **deb http://host/debian distribution section1 section2 section3**

  **deb-src http://host/debian distribution section1 section2 section3**

- The sources can be http, ftp, cdrom or file
- The fastest sources should be at the top of the list

28

4/27/2015



## The Sources file

- To identify the fastest sources you can use the netselect tool

  **#apt-get install netselect**

- You can then use netselect tool to identify the fastest sources

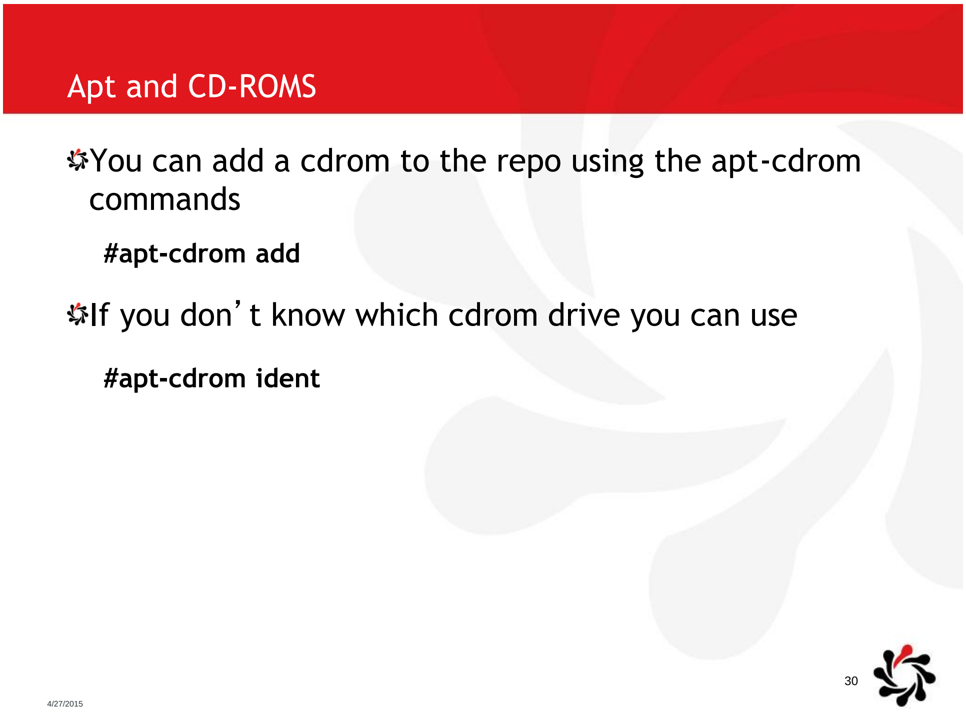
  **#netselect ftp.debian.org http.us.debian.org ftp.at.debian.org**

- Once you have modified your sources file, you need to update it.

  **#apt-get update**

4/27/2015

29



## Apt and CD-ROMS

- You can add a cdrom to the repo using the apt-cdrom commands

  **#apt-cdrom add**

- If you don’t know which cdrom drive you can use

  **#apt-cdrom ident**

4/27/2015

30

## Searching for packages

- The apt-cache tool is used for searching for packages that you want to install

```
#apt-cache search atari
```

- This will list all packages found with atari in their details.  To then see just one of those packages details

```
#apt-cache show stella
```

## Install/remove packages

- Apt-get tool will install a package with all its dependencies

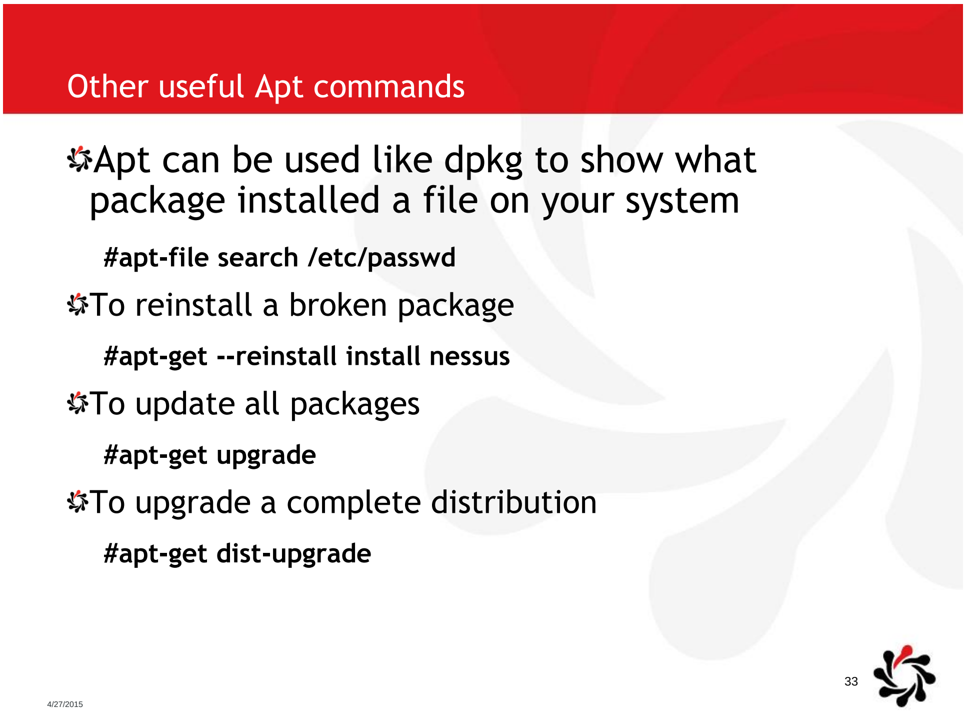
```
#apt-get install nessus
```

- To remove a package from the system, but leave the configuration files

```
#apt-get remove nessus
```

- To remove a package and all its configuration files

```
#apt-get --purge remove nessus
```



## Other useful Apt commands

- Apt can be used like dpkg to show what package installed a file on your system

```
#apt-file search /etc/passwd
```

- To reinstall a broken package

```
#apt-get --reinstall install nessus
```

- To update all packages

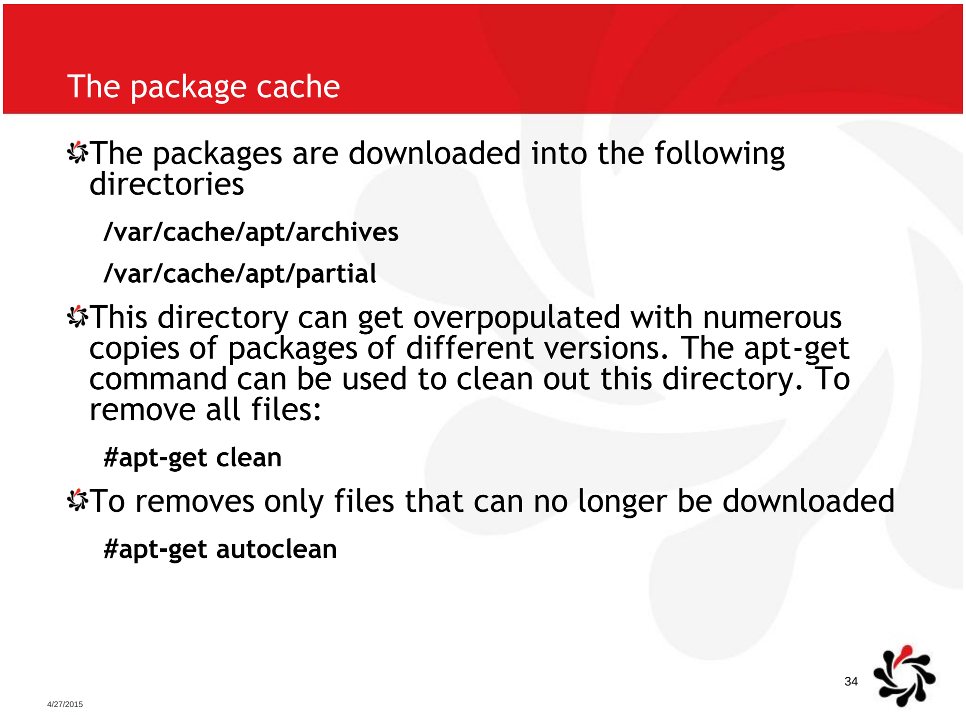
```
#apt-get upgrade
```

- To upgrade a complete distribution

```
#apt-get dist-upgrade
```

33

4/27/2015



## The package cache

- The packages are downloaded into the following directories

```
/var/cache/apt/archives
/var/cache/apt/partial
```

- This directory can get overpopulated with numerous copies of packages of different versions. The apt-get command can be used to clean out this directory. To remove all files:

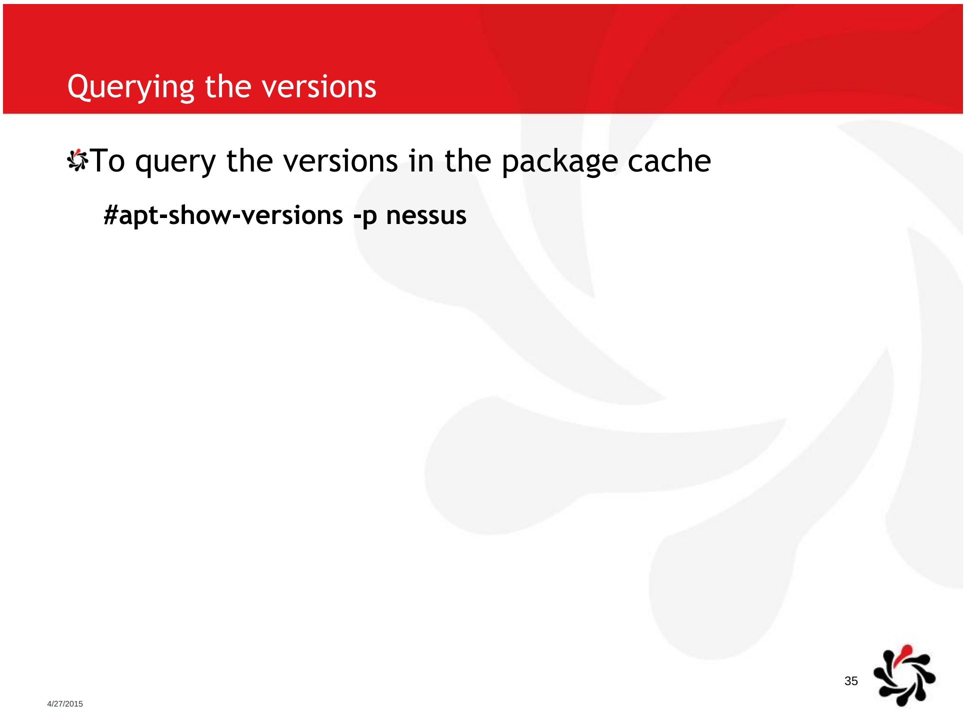
```
#apt-get clean
```

- To removes only files that can no longer be downloaded

```
#apt-get autoclean
```

34

4/27/2015



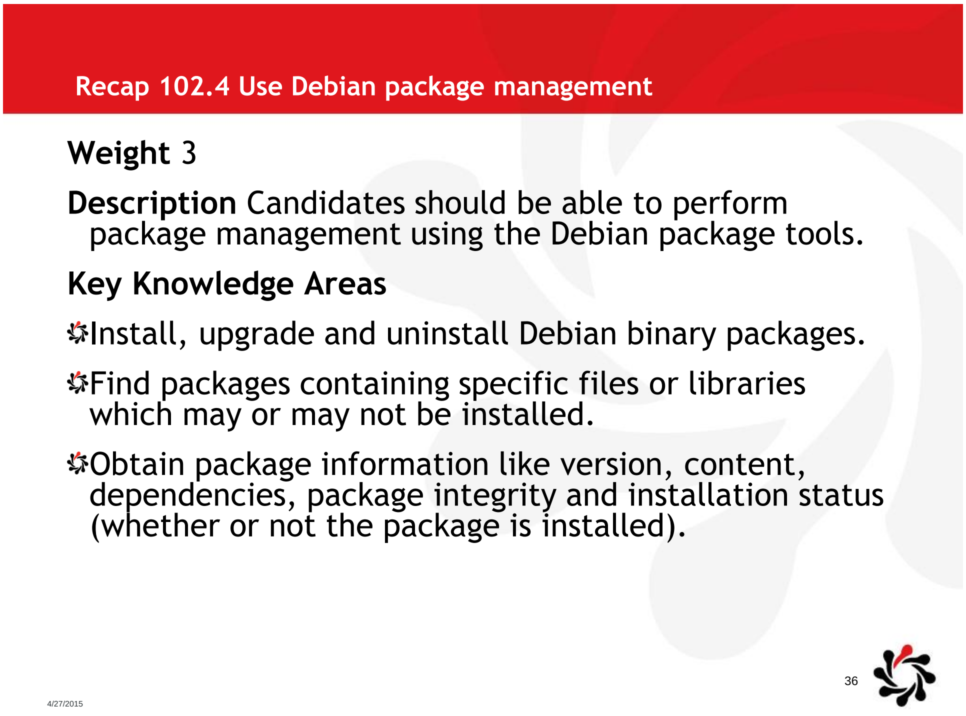
## Querying the versions

- To query the versions in the package cache

```
#apt-show-versions -p nessus
```

4/27/2015

35



## Recap 102.4 Use Debian package management

**Weight** 3

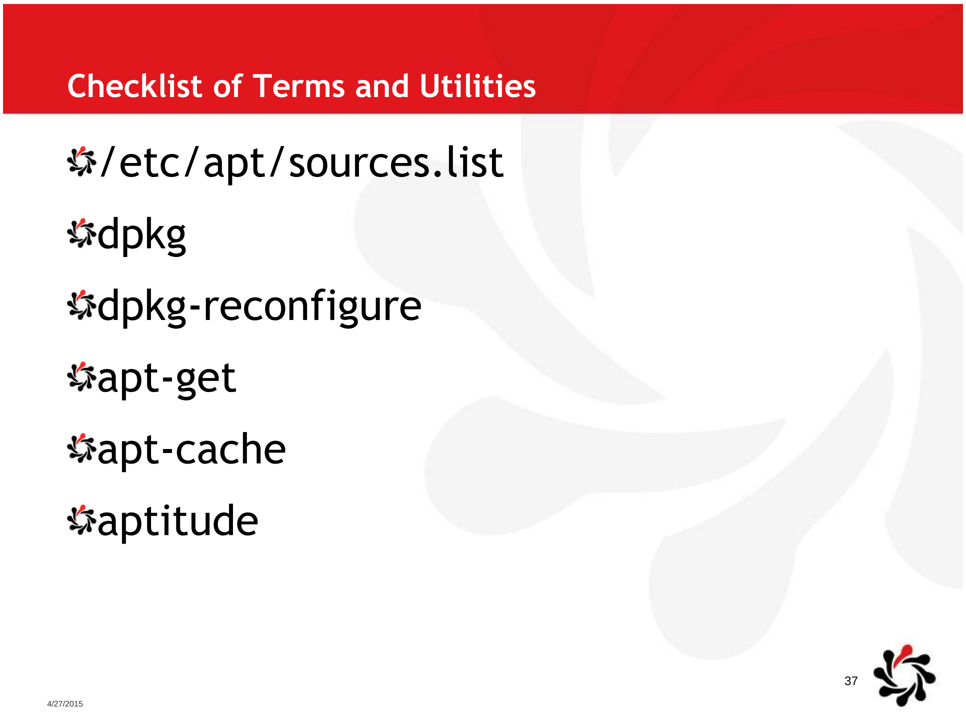
**Description** Candidates should be able to perform package management using the Debian package tools.

**Key Knowledge Areas**

- Install, upgrade and uninstall Debian binary packages.
- Find packages containing specific files or libraries which may or may not be installed.
- Obtain package information like version, content, dependencies, package integrity and installation status (whether or not the package is installed).

4/27/2015

36

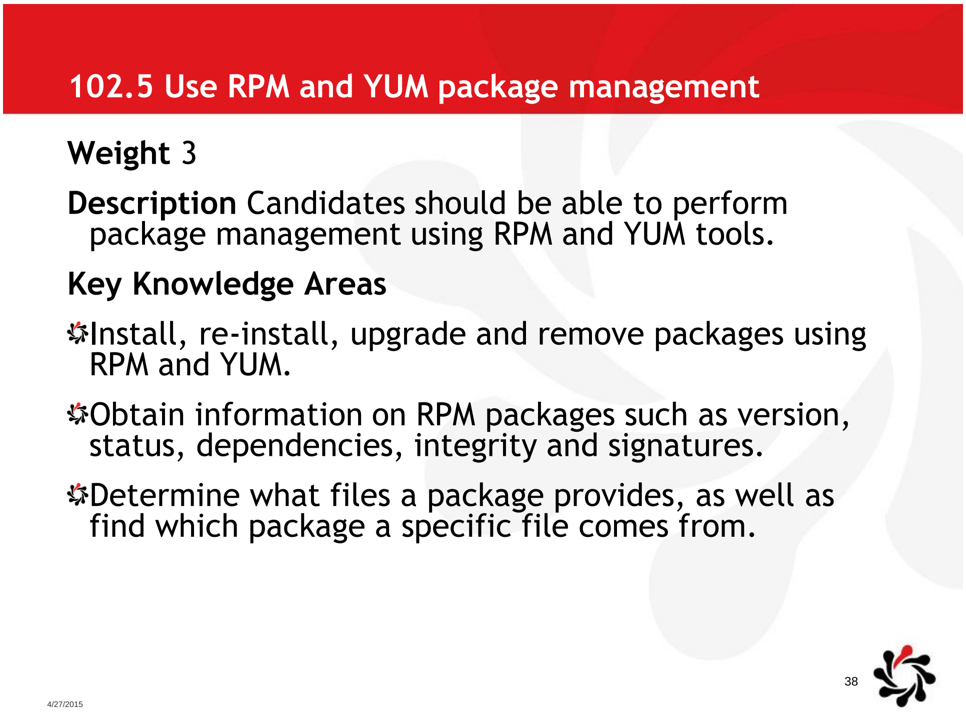


## Checklist of Terms and Utilities

- /etc/apt/sources.list
- dpkg
- dpkg-reconfigure
- apt-get
- apt-cache
- aptitude

4/27/2015

37



## 102.5 Use RPM and YUM package management

**Weight** 3

**Description** Candidates should be able to perform package management using RPM and YUM tools.

**Key Knowledge Areas**

- Install, re-install, upgrade and remove packages using RPM and YUM.
- Obtain information on RPM packages such as version, status, dependencies, integrity and signatures.
- Determine what files a package provides, as well as find which package a specific file comes from.

4/27/2015

38

## RPM

- Rpm is the basic tool for installing redhat packages onto any compatible system. It can be used to install, remove, upgrade, query and verify rpm packages.
- It consists of:
  - The **rpm** tool
  - The **rpm database**
  - The **rpmrc** configuration file
- Packages must be downloaded before being installed

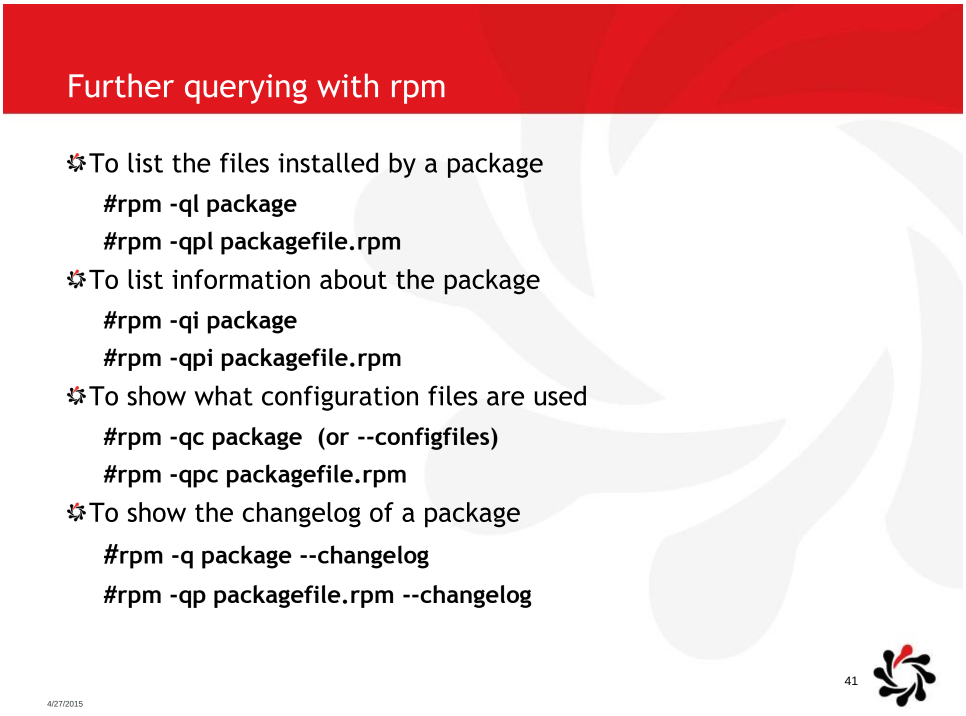
## Querying with rpm

- The rpm tool can query a package, whether or not it is installed. Installed and non installed packages contain the same information
- On an uninstalled **package file** use the following:

```
#rpm -qp packagefile.rpm
```

- When querying an installed package, the rpm database is queried.

```
#rpm -q package
```



## Further querying with rpm

- To list the files installed by a package

```
#rpm -ql package
#rpm -qpl packagefile.rpm
```

- To list information about the package

```
#rpm -qi package
#rpm -qpi packagefile.rpm
```

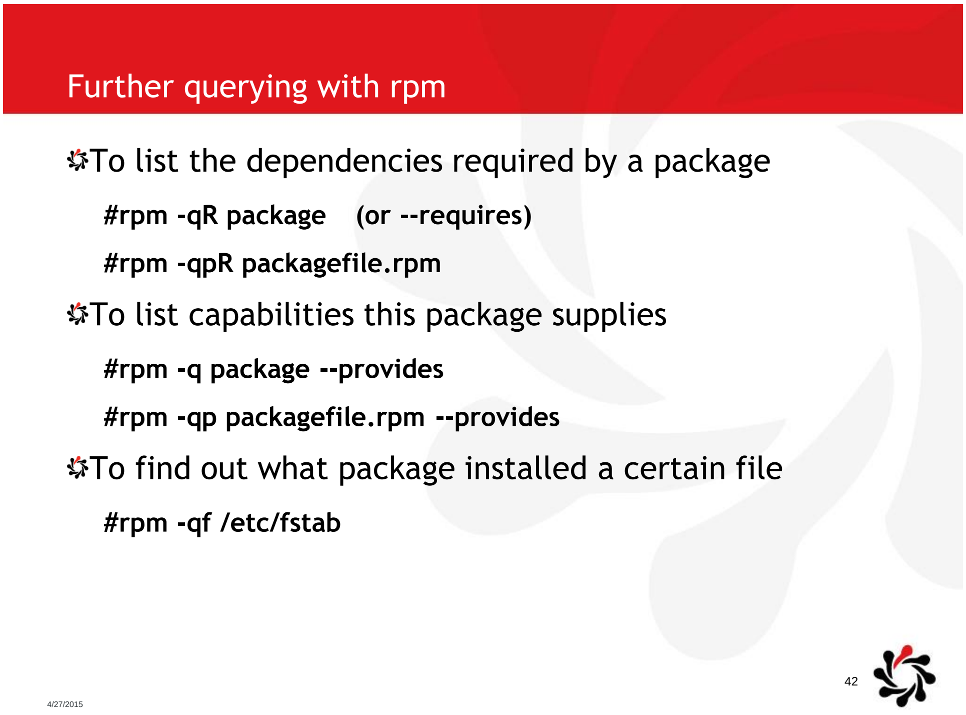
- To show what configuration files are used

```
#rpm -qc package  (or --configfiles)
#rpm -qpc packagefile.rpm
```

- To show the changelog of a package

```
#rpm -q package --changelog
#rpm -qp packagefile.rpm --changelog
```

4/27/2015 41



## Further querying with rpm

- To list the dependencies required by a package

```
#rpm -qR package    (or --requires)
#rpm -qpR packagefile.rpm
```

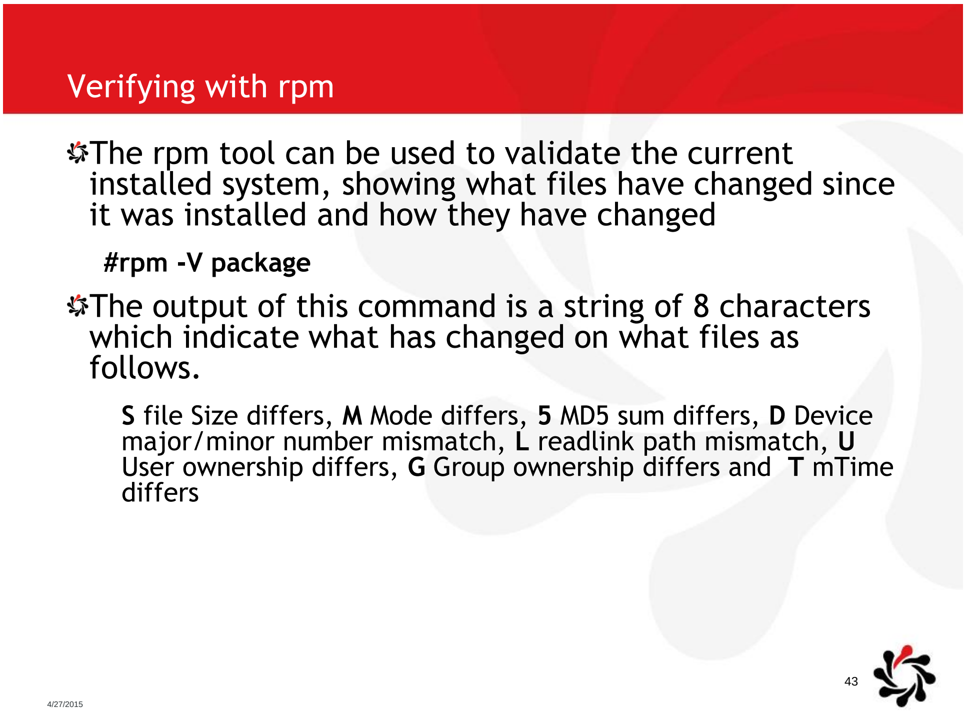
- To list capabilities this package supplies

```
#rpm -q package --provides
#rpm -qp packagefile.rpm --provides
```

- To find out what package installed a certain file

```
#rpm -qf /etc/fstab
```

4/27/2015 42



## Verifying with rpm

- The rpm tool can be used to validate the current installed system, showing what files have changed since it was installed and how they have changed

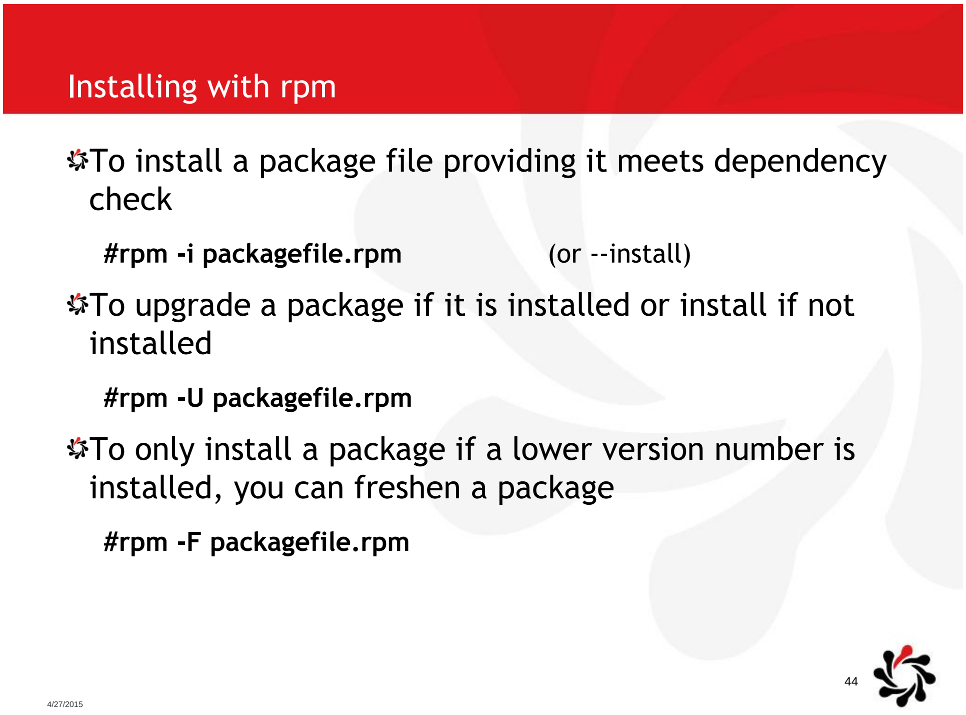
```
#rpm -V package
```

- The output of this command is a string of 8 characters which indicate what has changed on what files as follows.

  **S** file Size differs, **M** Mode differs, **5** MD5 sum differs, **D** Device major/minor number mismatch, **L** readlink path mismatch, **U** User ownership differs, **G** Group ownership differs and **T** mTime differs

4/27/2015

43



## Installing with rpm

- To install a package file providing it meets dependency check

  **#rpm -i packagefile.rpm** (or --install)

- To upgrade a package if it is installed or install if not installed

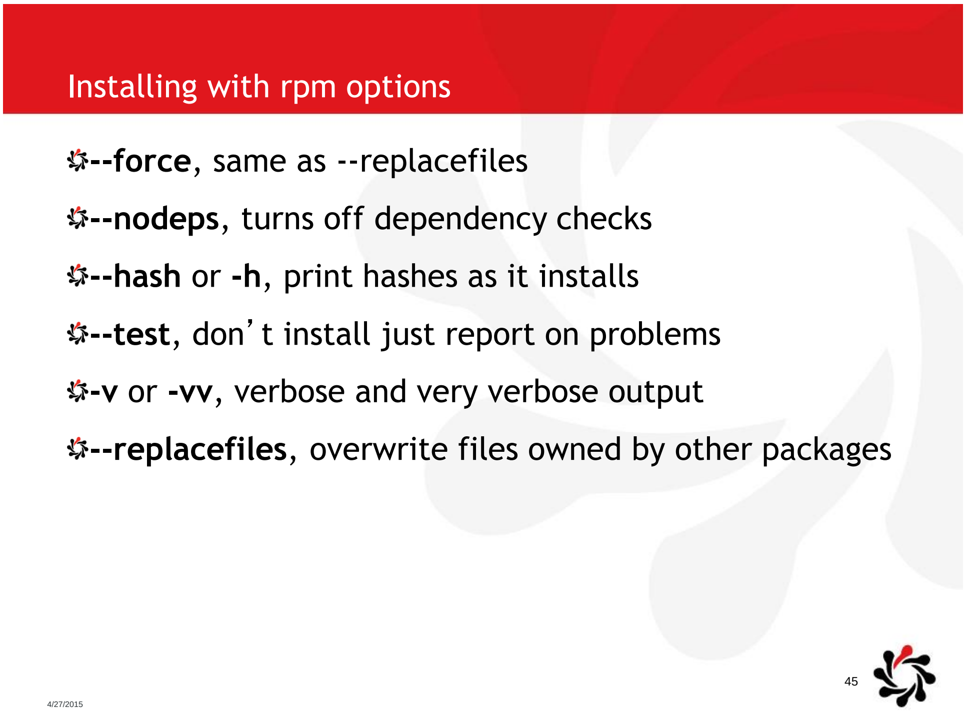
```
#rpm -U packagefile.rpm
```

- To only install a package if a lower version number is installed, you can freshen a package

```
#rpm -F packagefile.rpm
```

4/27/2015

44

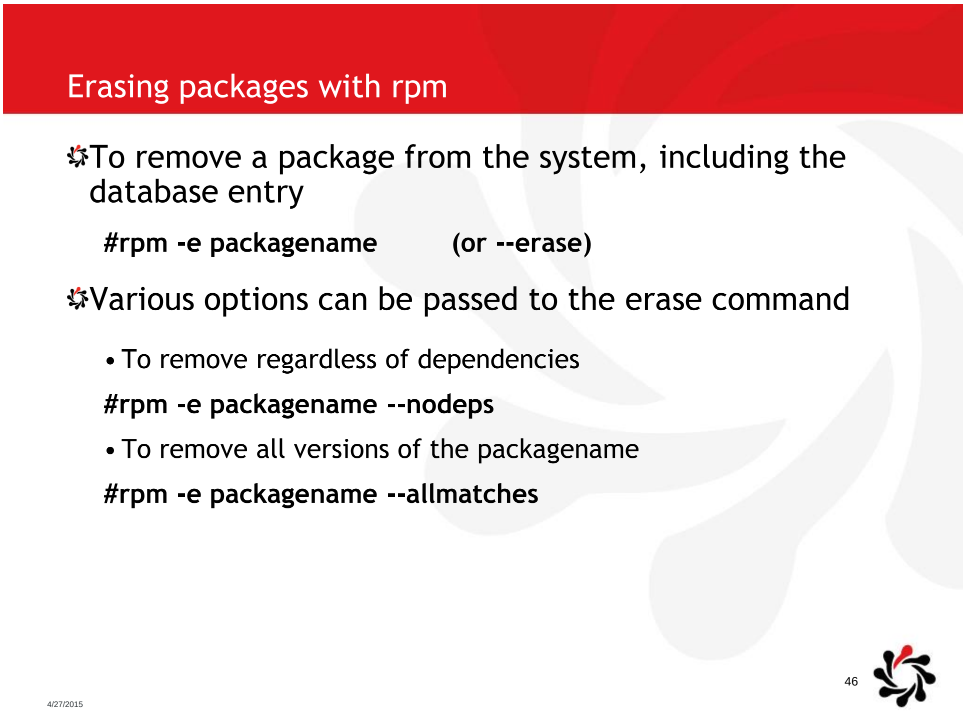


## Installing with rpm options

- **--force**, same as --replacefiles
- **--nodeps**, turns off dependency checks
- **--hash** or **-h**, print hashes as it installs
- **--test**, don’t install just report on problems
- **-v** or **-vv**, verbose and very verbose output
- **--replacefiles**, overwrite files owned by other packages

4/27/2015

45



## Erasing packages with rpm

- To remove a package from the system, including the database entry

  **#rpm -e packagename (or --erase)**

- Various options can be passed to the erase command
  - To remove regardless of dependencies

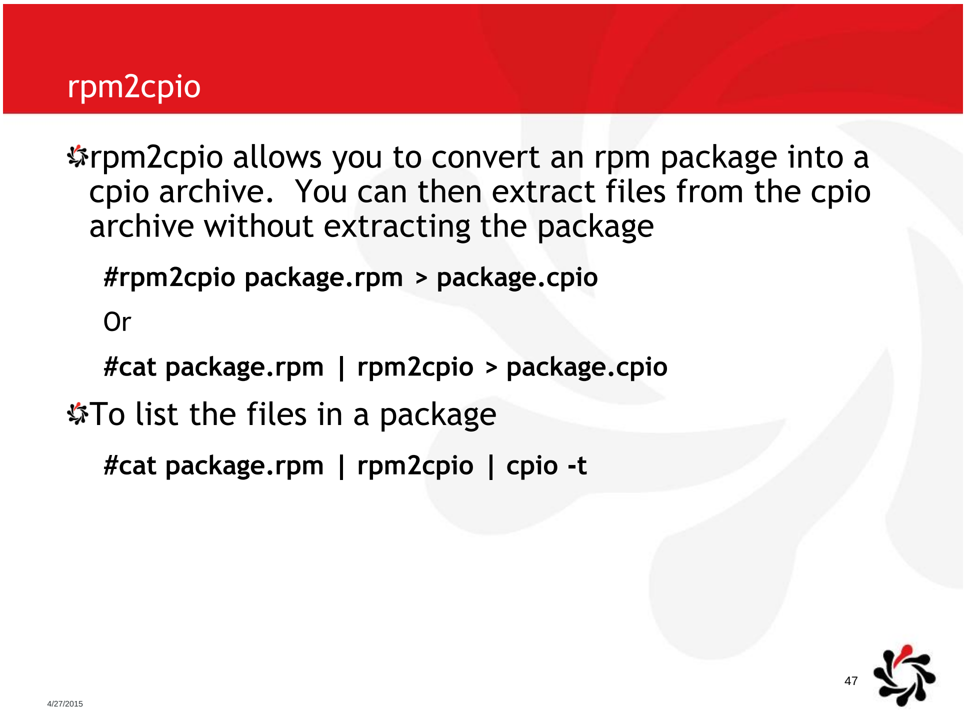
    **#rpm -e packagename --nodeps**

  - To remove all versions of the packagename

    **#rpm -e packagename --allmatches**

4/27/2015

46



## rpm2cpio

- rpm2cpio allows you to convert an rpm package into a cpio archive.  You can then extract files from the cpio archive without extracting the package

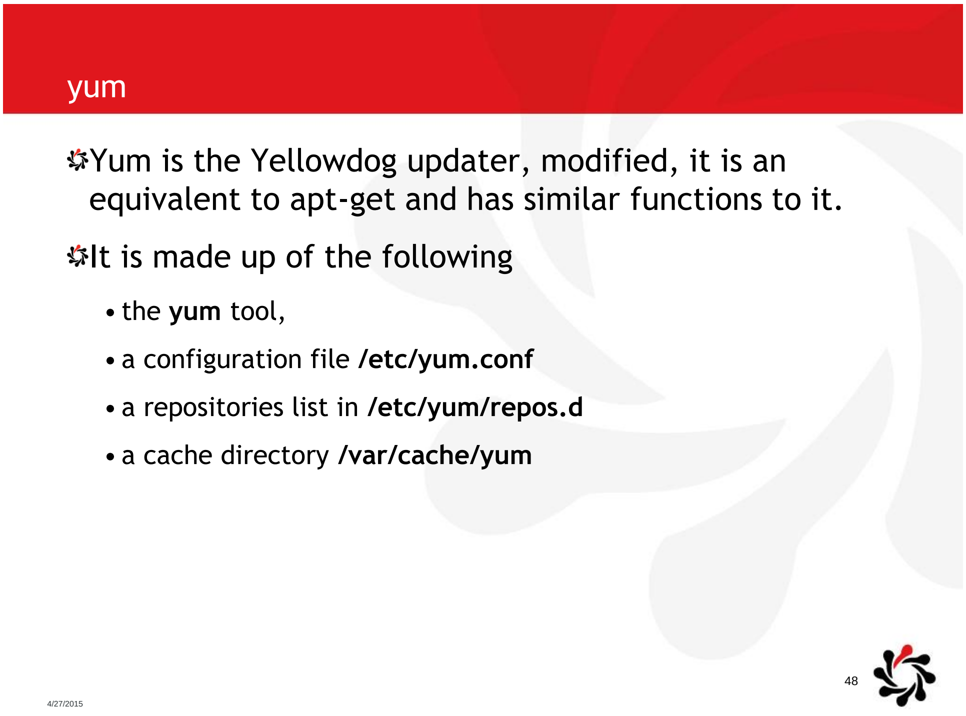
```
#rpm2cpio package.rpm > package.cpio
```

Or

```
#cat package.rpm | rpm2cpio > package.cpio
```

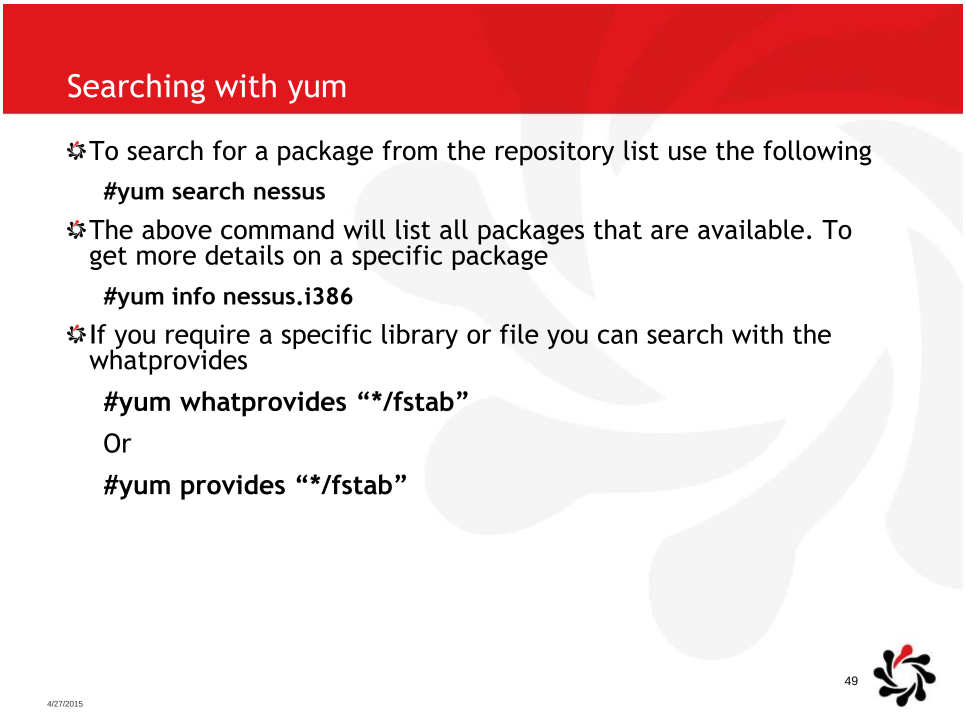
- To list the files in a package

```
#cat package.rpm | rpm2cpio | cpio -t
```



## yum

- Yum is the Yellowdog updater, modified, it is an equivalent to apt-get and has similar functions to it.
- It is made up of the following
  - the **yum** tool,
  - a configuration file **/etc/yum.conf**
  - a repositories list in **/etc/yum/repos.d**
  - a cache directory **/var/cache/yum**



## Searching with yum

- To search for a package from the repository list use the following

```
#yum search nessus
```

- The above command will list all packages that are available. To get more details on a specific package

```
#yum info nessus.i386
```

- If you require a specific library or file you can search with the whatprovides

```
#yum whatprovides "*/fstab"
```

Or

```
#yum provides "*/fstab"
```

49

4/27/2015

## Listing packages with yum

- To list package and their states, use the yum list

```
#yum list [all | glob_exp1] [glob_exp2] [...]
```

  - List all available and installed packages.

```
#yum list available [glob_exp1] [...]
```

  - List all packages in the yum repositories available to be installed.

```
#yum list updates [glob_exp1] [...]
```

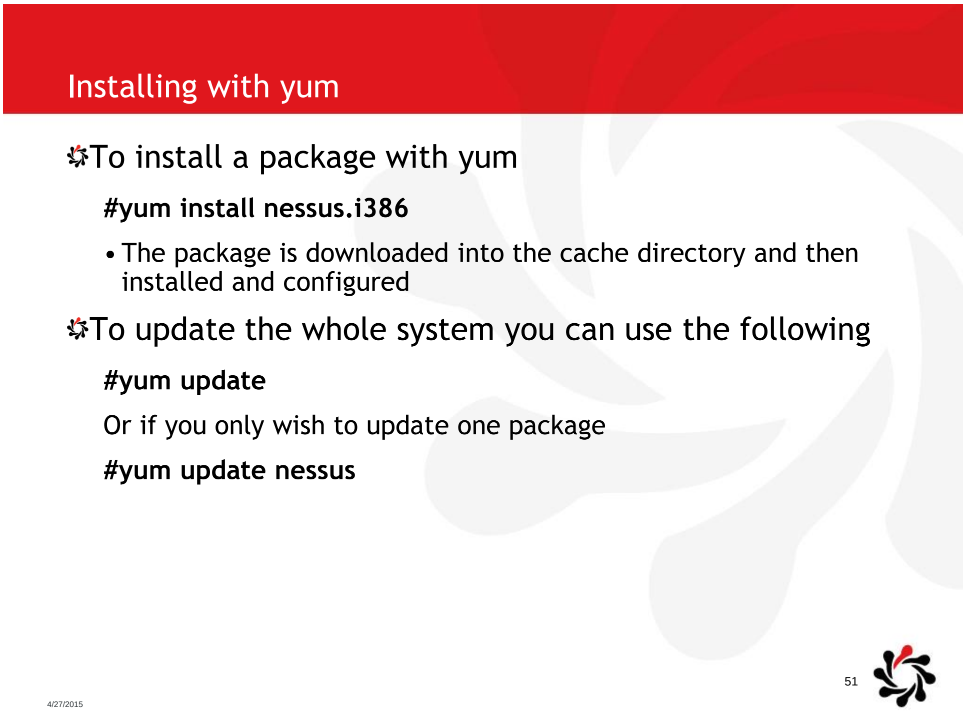
  - List all packages with updates available in the yum repositories.

```
#yum list installed [glob_exp1] [...]
```

  - List the packages specified by *args*. If an argument does not match the name of an available package, it is assumed to be a shell-style glob and any matches are printed.

50

4/27/2015



## Installing with yum

- To install a package with yum

```
#yum install nessus.i386
```

  - The package is downloaded into the cache directory and then installed and configured

- To update the whole system you can use the following

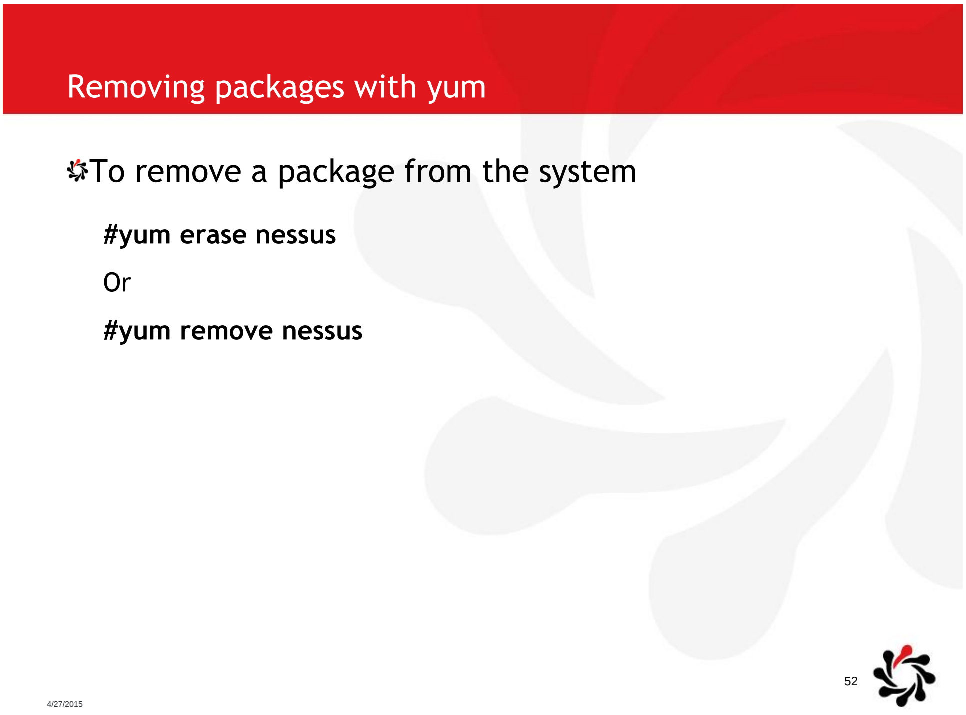
```
#yum update
```

  Or if you only wish to update one package

```
#yum update nessus
```

4/27/2015

51



## Removing packages with yum

- To remove a package from the system

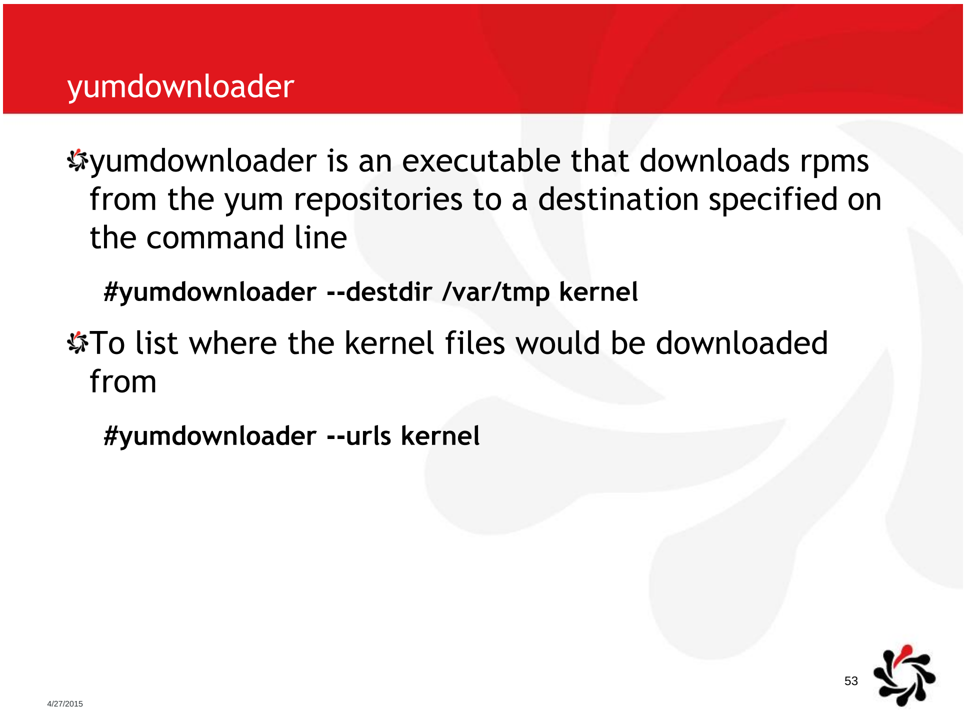
```
#yum erase nessus
```

  Or

```
#yum remove nessus
```

4/27/2015

52



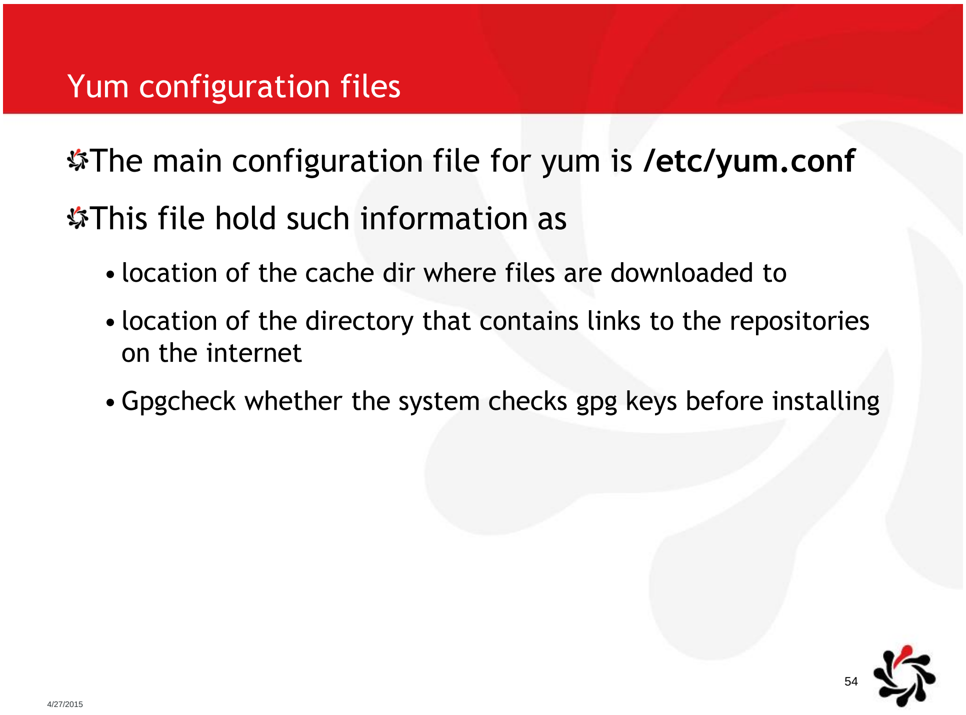
## yumdownloader

- yumdownloader is an executable that downloads rpms from the yum repositories to a destination specified on the command line

```
#yumdownloader --destdir /var/tmp kernel
```

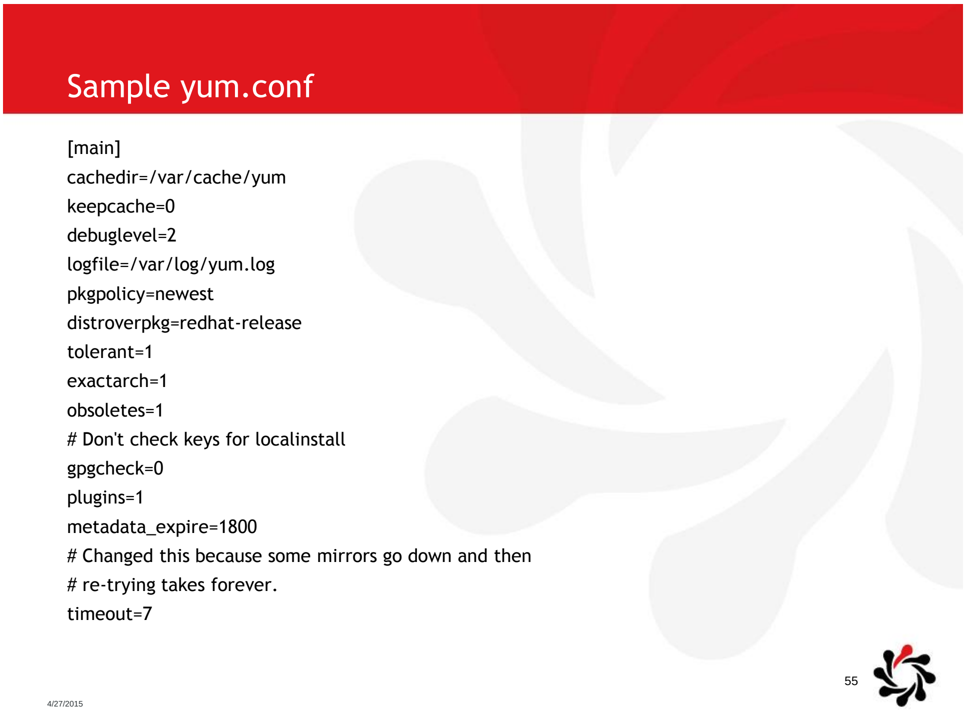
- To list where the kernel files would be downloaded from

```
#yumdownloader --urls kernel
```



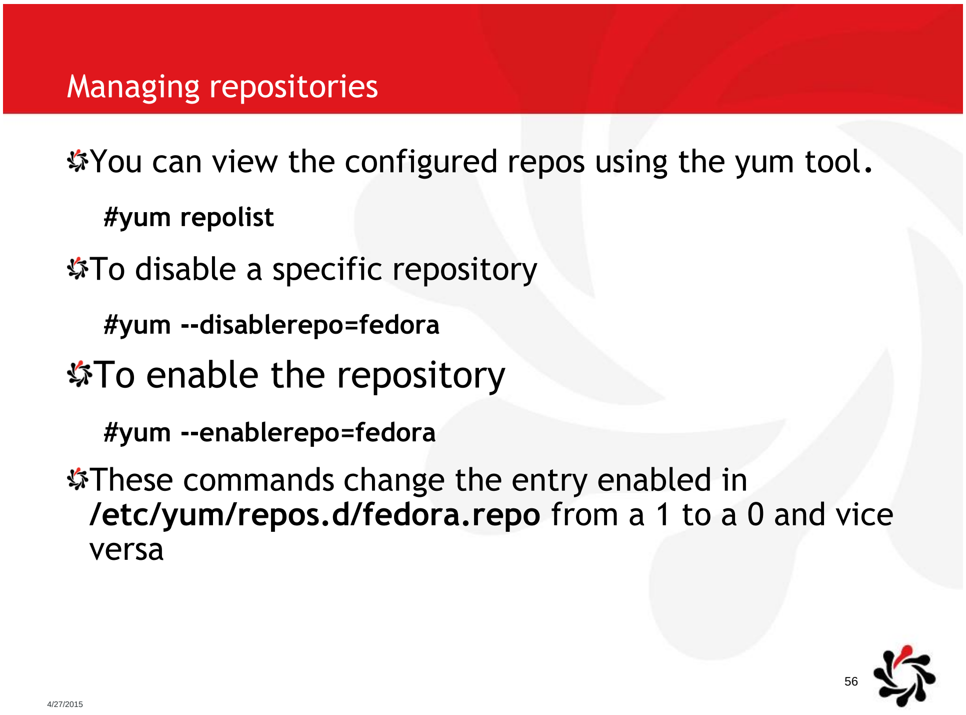
## Yum configuration files

- The main configuration file for yum is **/etc/yum.conf**
- This file hold such information as
  - location of the cache dir where files are downloaded to
  - location of the directory that contains links to the repositories on the internet
  - Gpgcheck whether the system checks gpg keys before installing



## Sample yum.conf

```
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
pkgpolicy=newest
distroverpkg=redhat-release
tolerant=1
exactarch=1
obsoletes=1
# Don't check keys for localinstall
gpgcheck=0
plugins=1
metadata_expire=1800
# Changed this because some mirrors go down and then
# re-trying takes forever.
timeout=7
```



## Managing repositories

- You can view the configured repos using the yum tool.

  **#yum repolist**

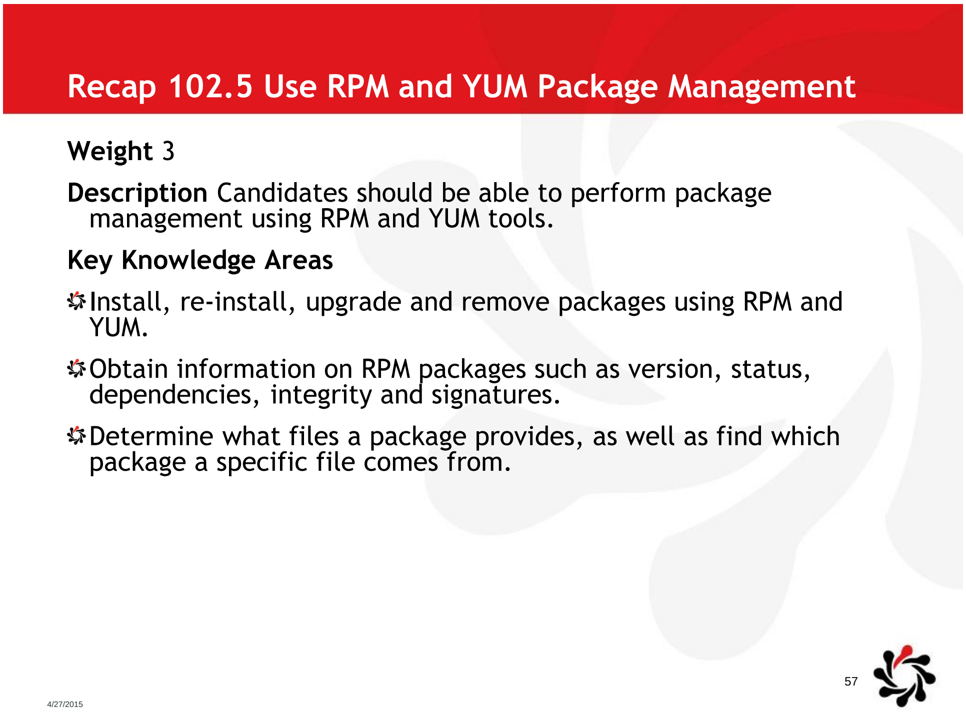
- To disable a specific repository

  **#yum --disablerepo=fedora**

- To enable the repository

  **#yum --enablerepo=fedora**

- These commands change the entry enabled in **/etc/yum/repos.d/fedora.repo** from a 1 to a 0 and vice versa



## Recap 102.5 Use RPM and YUM Package Management

**Weight** 3

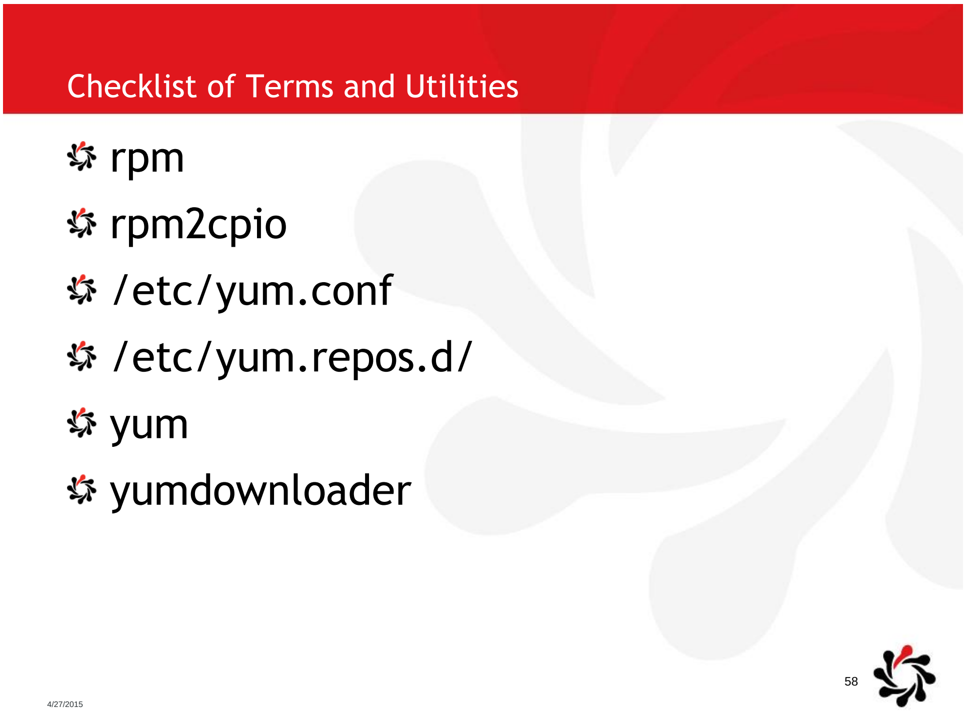
**Description** Candidates should be able to perform package management using RPM and YUM tools.

**Key Knowledge Areas**

- Install, re-install, upgrade and remove packages using RPM and YUM.
- Obtain information on RPM packages such as version, status, dependencies, integrity and signatures.
- Determine what files a package provides, as well as find which package a specific file comes from.

4/27/2015

57

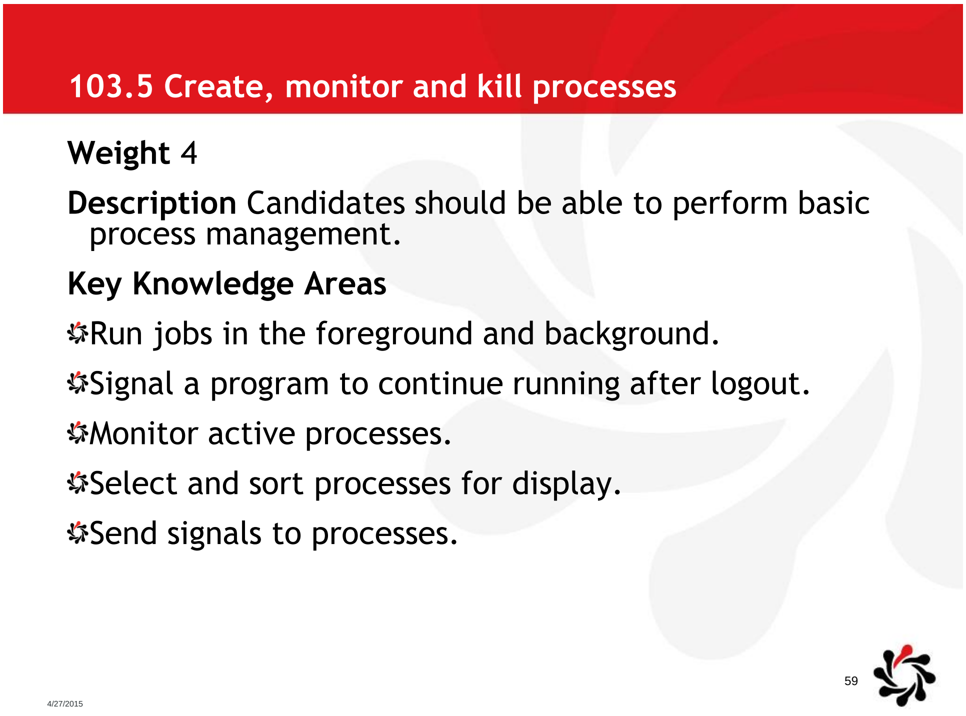


## Checklist of Terms and Utilities

- rpm
- rpm2cpio
- /etc/yum.conf
- /etc/yum.repos.d/
- yum
- yumdownloader

4/27/2015

58



## 103.5 Create, monitor and kill processes

**Weight** 4

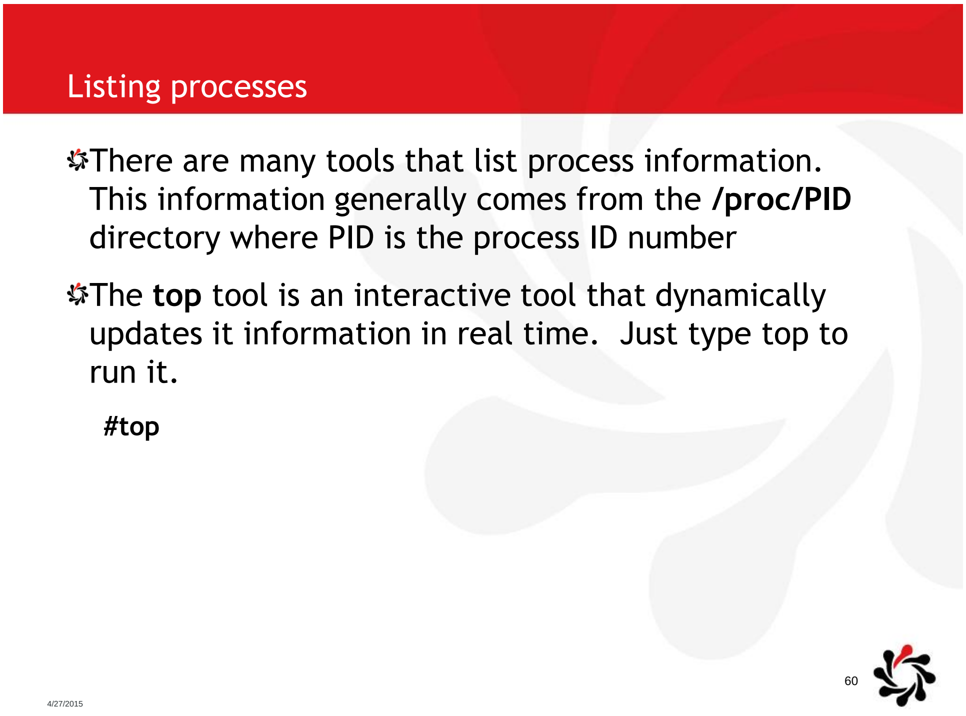
**Description** Candidates should be able to perform basic process management.

**Key Knowledge Areas**

- Run jobs in the foreground and background.
- Signal a program to continue running after logout.
- Monitor active processes.
- Select and sort processes for display.
- Send signals to processes.

4/27/2015

59



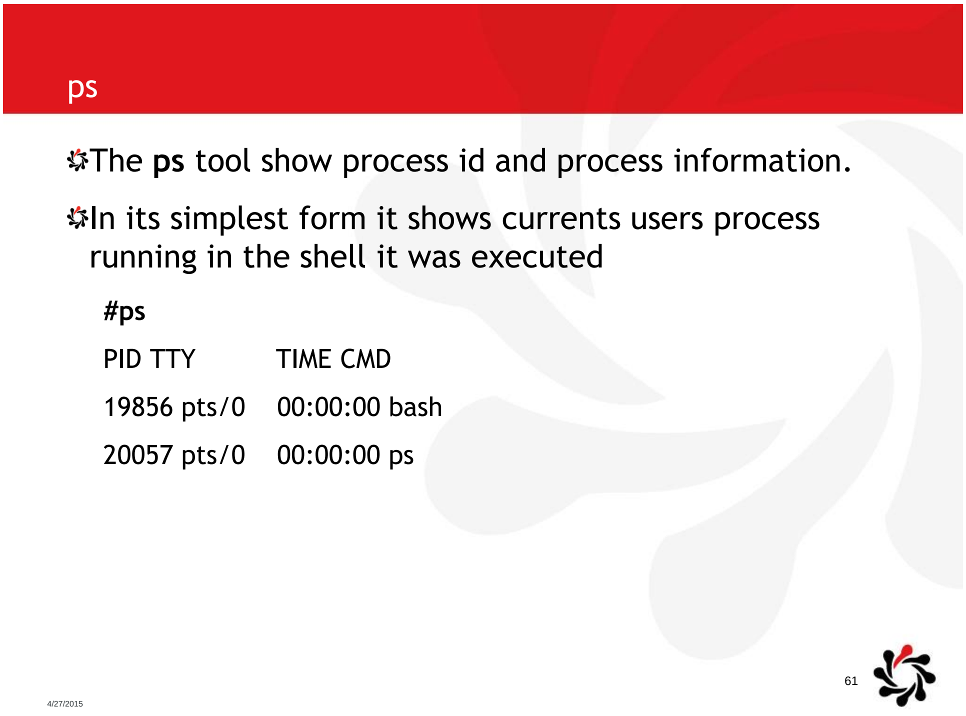
## Listing processes

- There are many tools that list process information. This information generally comes from the **/proc/PID** directory where PID is the process ID number
- The **top** tool is an interactive tool that dynamically updates it information in real time.  Just type top to run it.

```
#top
```

4/27/2015

60



## ps

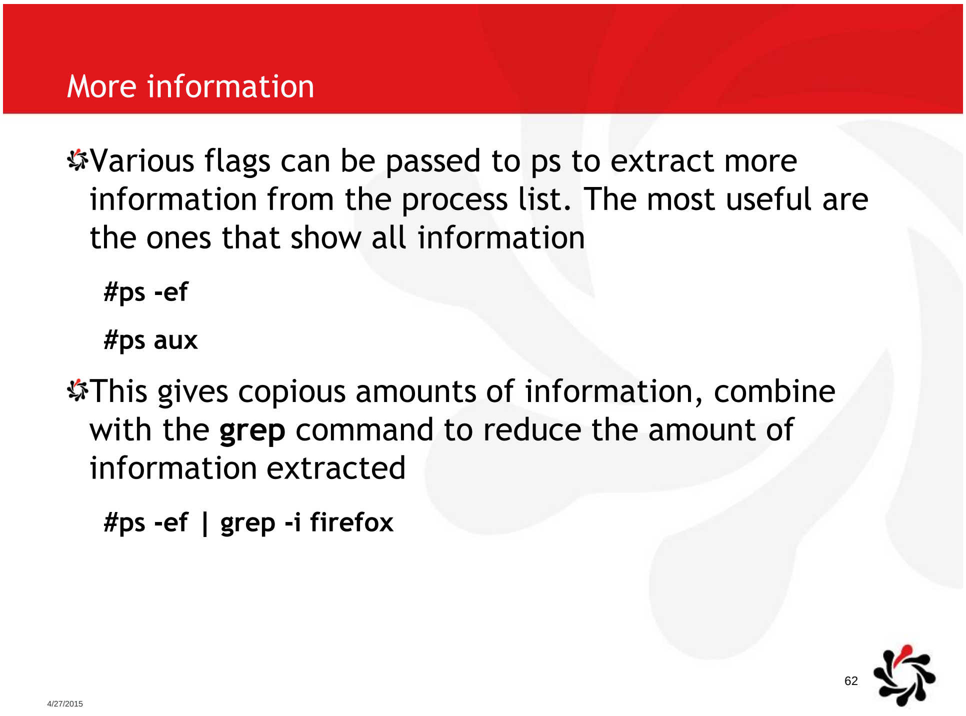
- The **ps** tool show process id and process information.
- In its simplest form it shows currents users process running in the shell it was executed

**#ps**

```
PID TTY         TIME CMD
19856 pts/0    00:00:00 bash
20057 pts/0    00:00:00 ps
```

61

4/27/2015



## More information

- Various flags can be passed to ps to extract more information from the process list. The most useful are the ones that show all information

**#ps -ef**

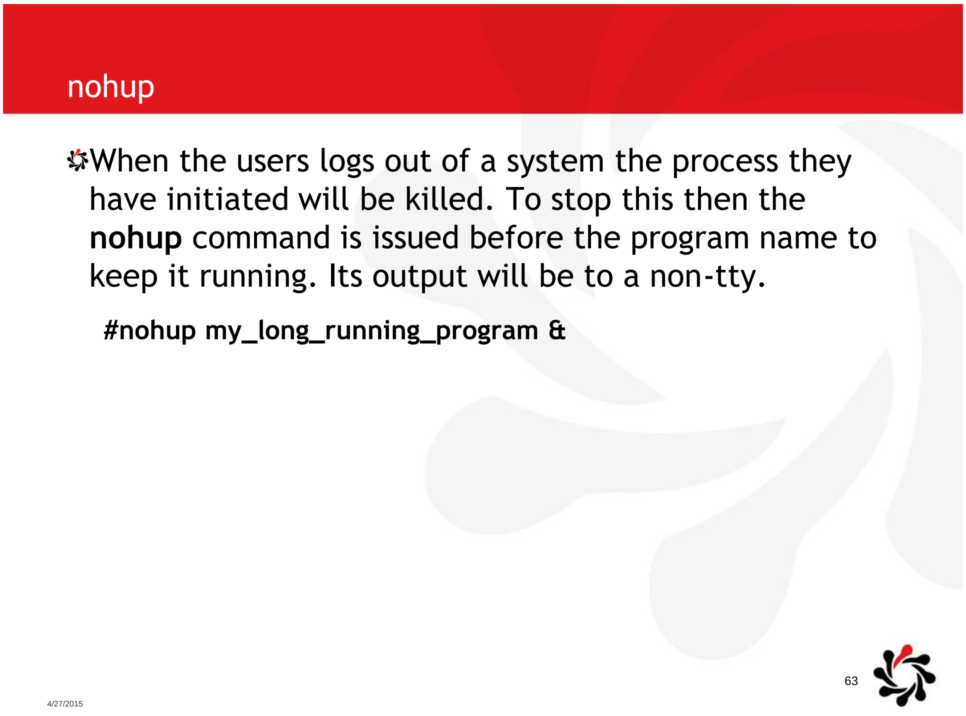
**#ps aux**

- This gives copious amounts of information, combine with the **grep** command to reduce the amount of information extracted

**#ps -ef | grep -i firefox**

62

4/27/2015



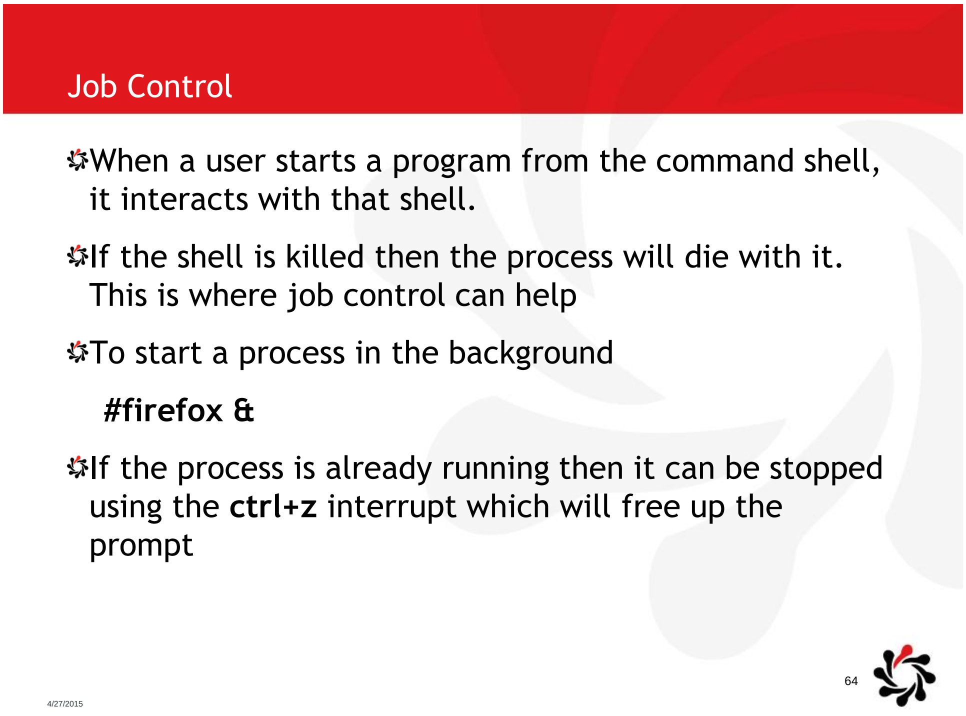
## nohup

- When the users logs out of a system the process they have initiated will be killed. To stop this then the **nohup** command is issued before the program name to keep it running. Its output will be to a non-tty.

```
#nohup my_long_running_program &
```

4/27/2015

63



## Job Control

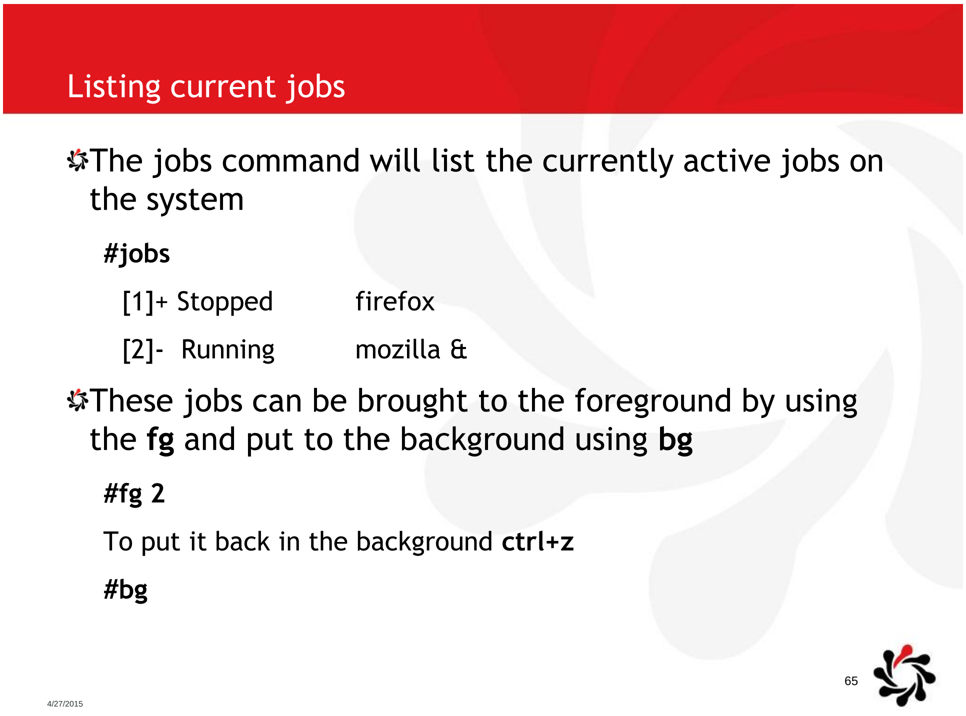
- When a user starts a program from the command shell, it interacts with that shell.
- If the shell is killed then the process will die with it. This is where job control can help
- To start a process in the background

```
#firefox &
```

- If the process is already running then it can be stopped using the **ctrl+z** interrupt which will free up the prompt

4/27/2015

64



## Listing current jobs

- The jobs command will list the currently active jobs on the system

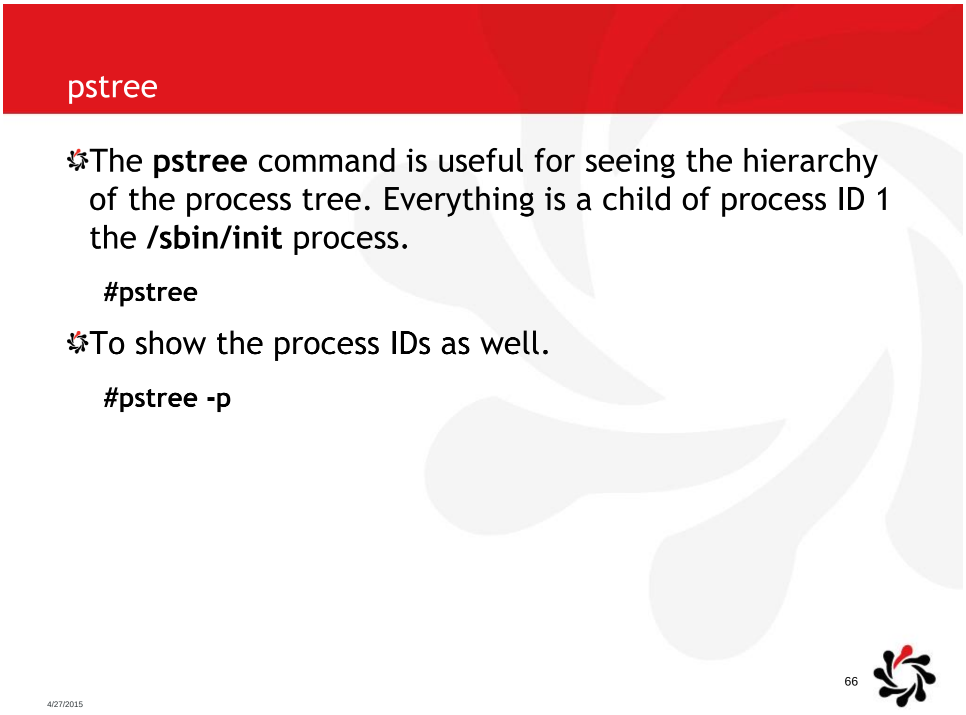
```
#jobs
  [1]+ Stopped          firefox
  [2]-  Running         mozilla &
```

- These jobs can be brought to the foreground by using the **fg** and put to the background using **bg**

**#fg 2**

To put it back in the background **ctrl+z**

**#bg**



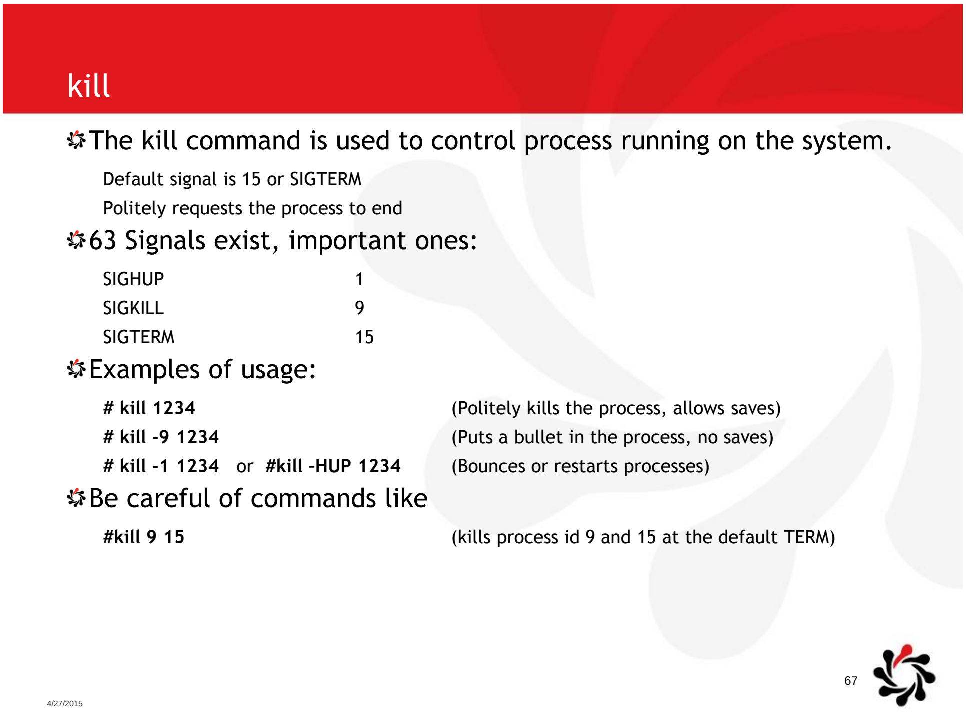
## pstree

- The **pstree** command is useful for seeing the hierarchy of the process tree. Everything is a child of process ID 1 the **/sbin/init** process.

**#pstree**

- To show the process IDs as well.

**#pstree -p**



## kill

- The kill command is used to control process running on the system.
  - Default signal is 15 or SIGTERM
  - Politely requests the process to end
- 63 Signals exist, important ones:

| | |
|---|---|
| SIGHUP | 1 |
| SIGKILL | 9 |
| SIGTERM | 15 |

- Examples of usage:

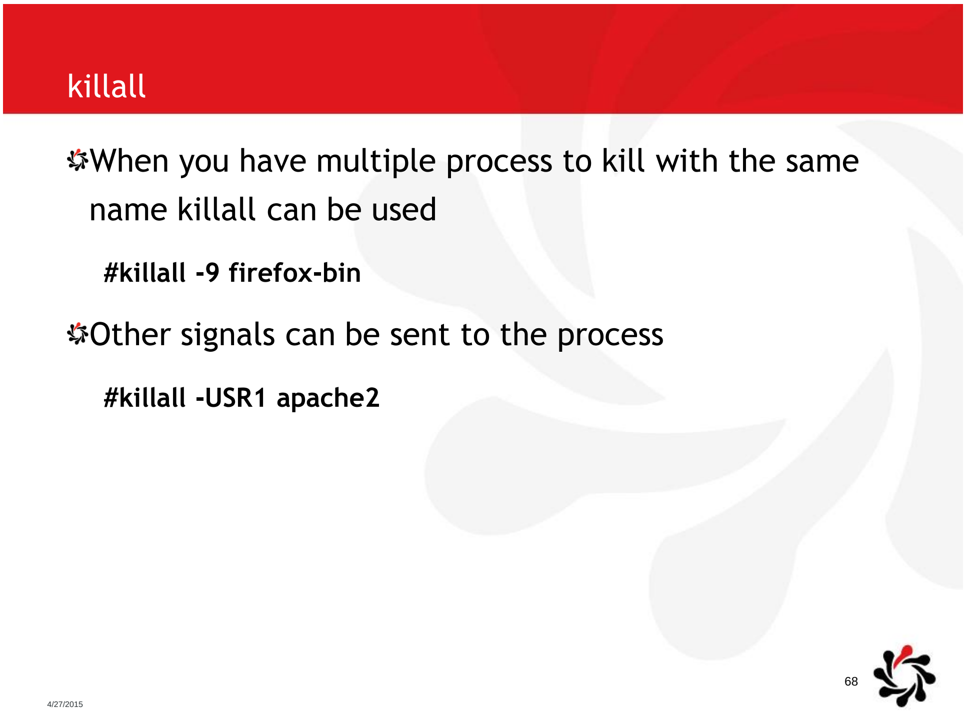
| | |
|---|---|
| **# kill 1234** | (Politely kills the process, allows saves) |
| **# kill -9 1234** | (Puts a bullet in the process, no saves) |
| **# kill -1 1234** or **#kill -HUP 1234** | (Bounces or restarts processes) |

- Be careful of commands like

| | |
|---|---|
| **#kill 9 15** | (kills process id 9 and 15 at the default TERM) |

67

4/27/2015



## killall

- When you have multiple process to kill with the same name killall can be used

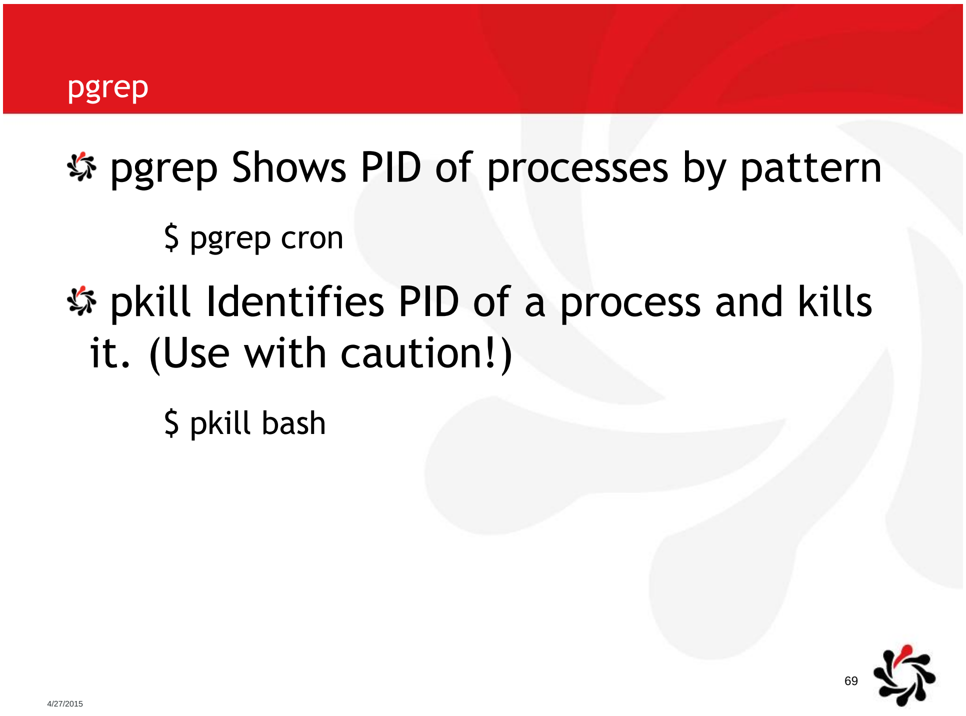
  **#killall -9 firefox-bin**

- Other signals can be sent to the process

  **#killall -USR1 apache2**

68

4/27/2015



## pgrep

- pgrep Shows PID of processes by pattern

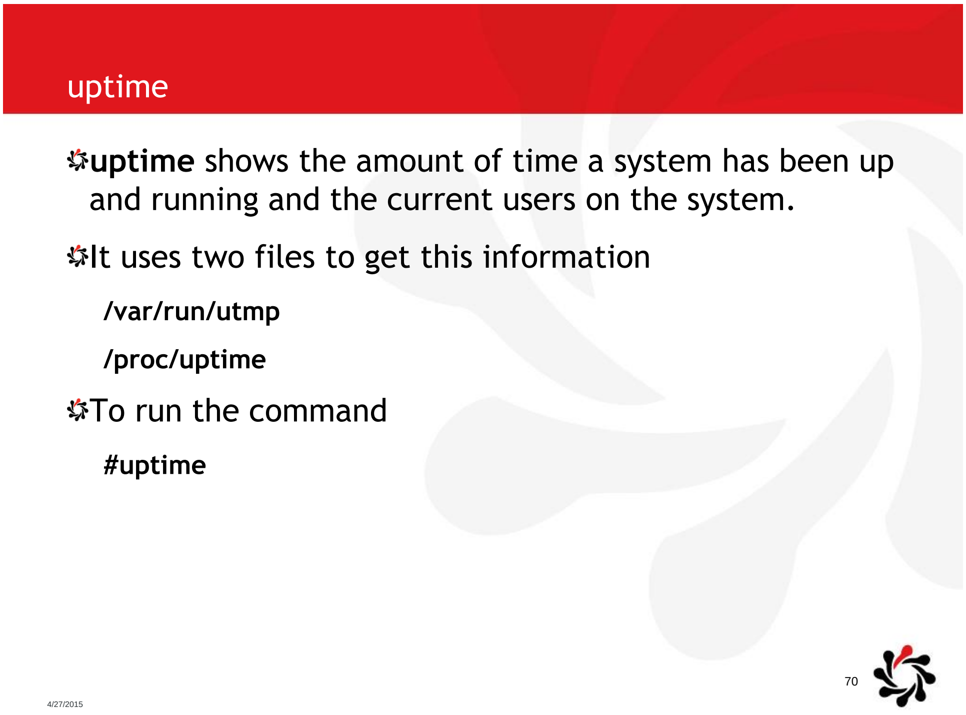
```
$ pgrep cron
```

- pkill Identifies PID of a process and kills it. (Use with caution!)

```
$ pkill bash
```

69

4/27/2015



## uptime

- **uptime** shows the amount of time a system has been up and running and the current users on the system.
- It uses two files to get this information

  **/var/run/utmp**

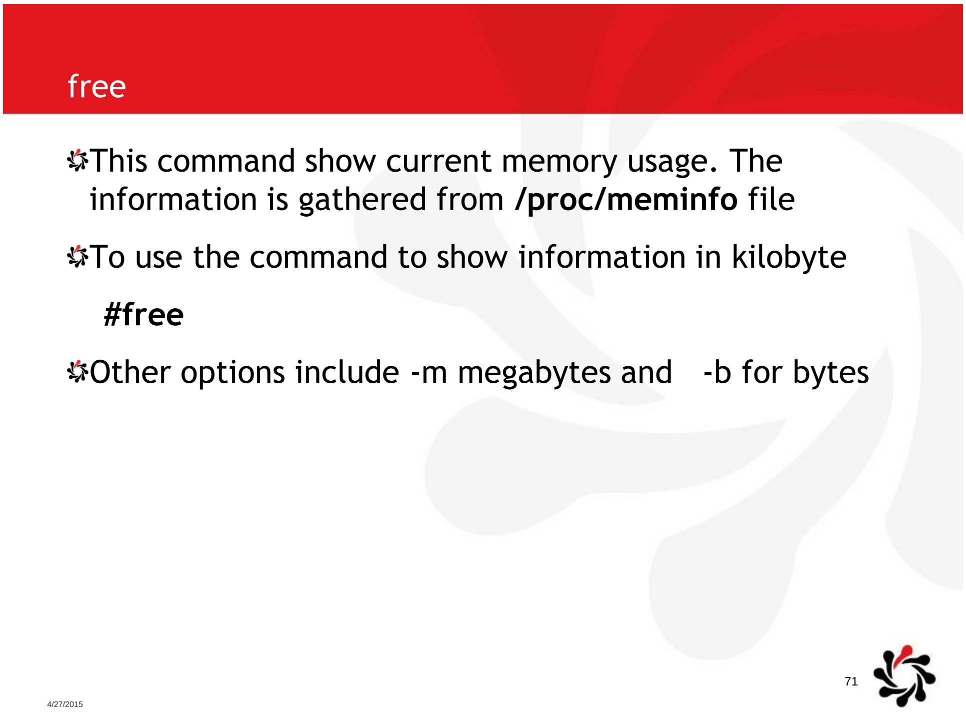
  **/proc/uptime**

- To run the command

```
#uptime
```

70

4/27/2015



## free

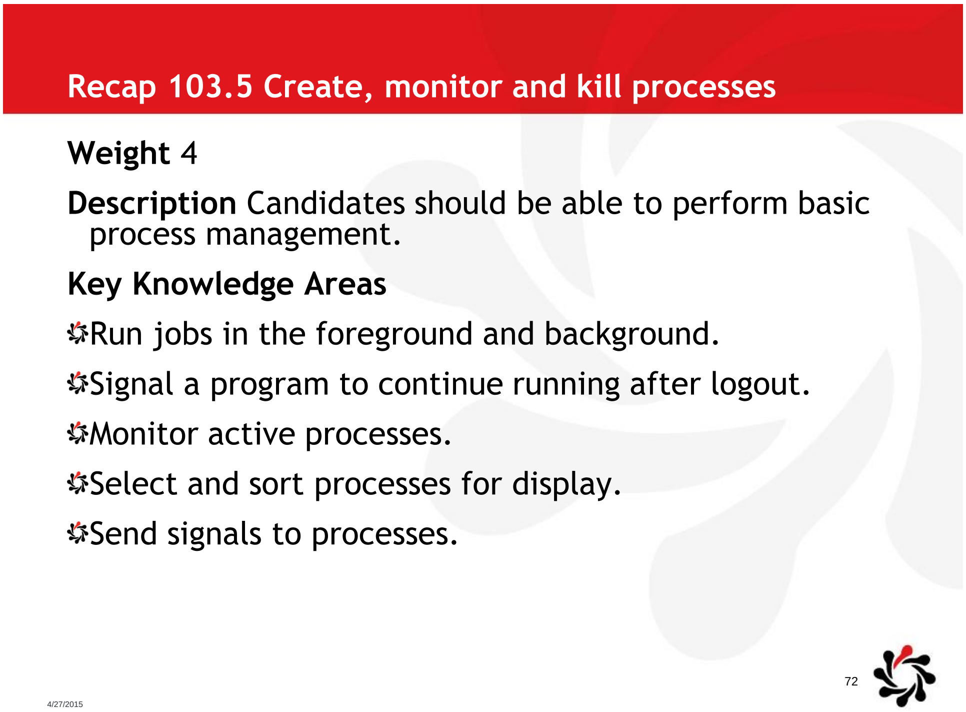
- This command show current memory usage. The information is gathered from **/proc/meminfo** file
- To use the command to show information in kilobyte

```
#free
```

- Other options include -m megabytes and   -b for bytes

4/27/2015

71



## Recap 103.5 Create, monitor and kill processes

**Weight** 4

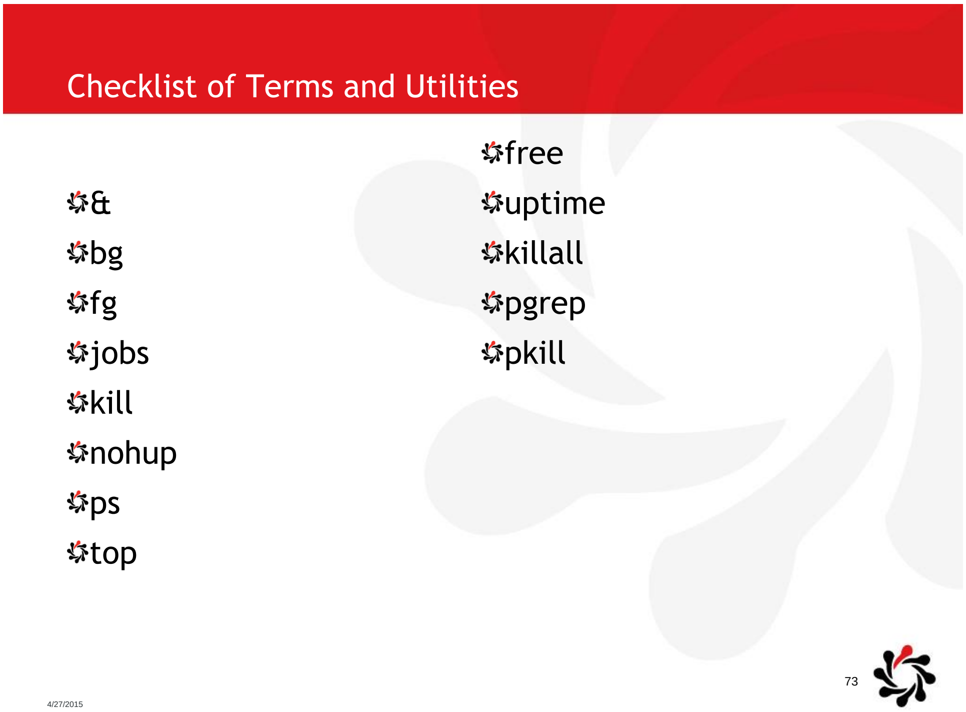
**Description** Candidates should be able to perform basic process management.

**Key Knowledge Areas**

- Run jobs in the foreground and background.
- Signal a program to continue running after logout.
- Monitor active processes.
- Select and sort processes for display.
- Send signals to processes.

4/27/2015

72

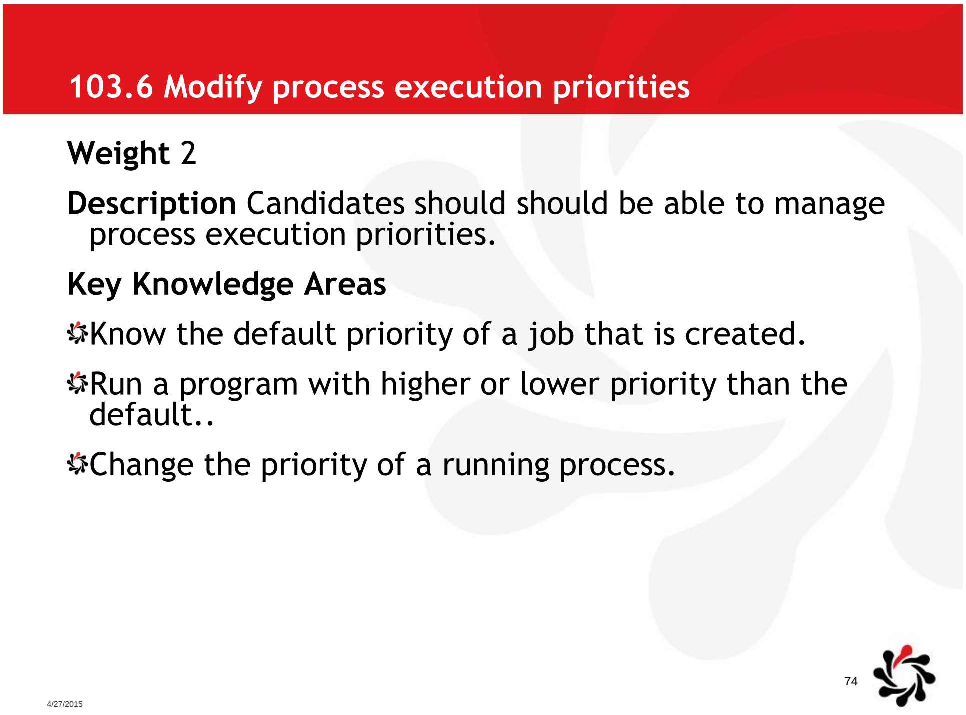


## Checklist of Terms and Utilities

- &
- bg
- fg
- jobs
- kill
- nohup
- ps
- top
- free
- uptime
- killall
- pgrep
- pkill

4/27/2015

73



## 103.6 Modify process execution priorities

**Weight** 2

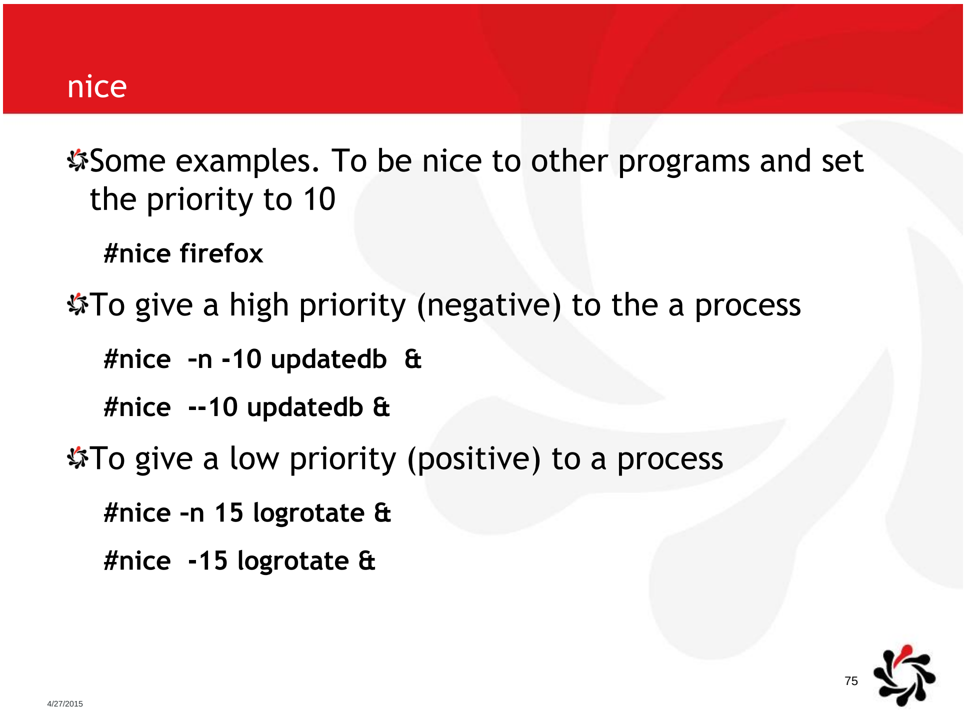
**Description** Candidates should should be able to manage process execution priorities.

**Key Knowledge Areas**

- Know the default priority of a job that is created.
- Run a program with higher or lower priority than the default..
- Change the priority of a running process.

4/27/2015

74



## nice

- Some examples. To be nice to other programs and set the priority to 10

```
#nice firefox
```

- To give a high priority (negative) to the a process

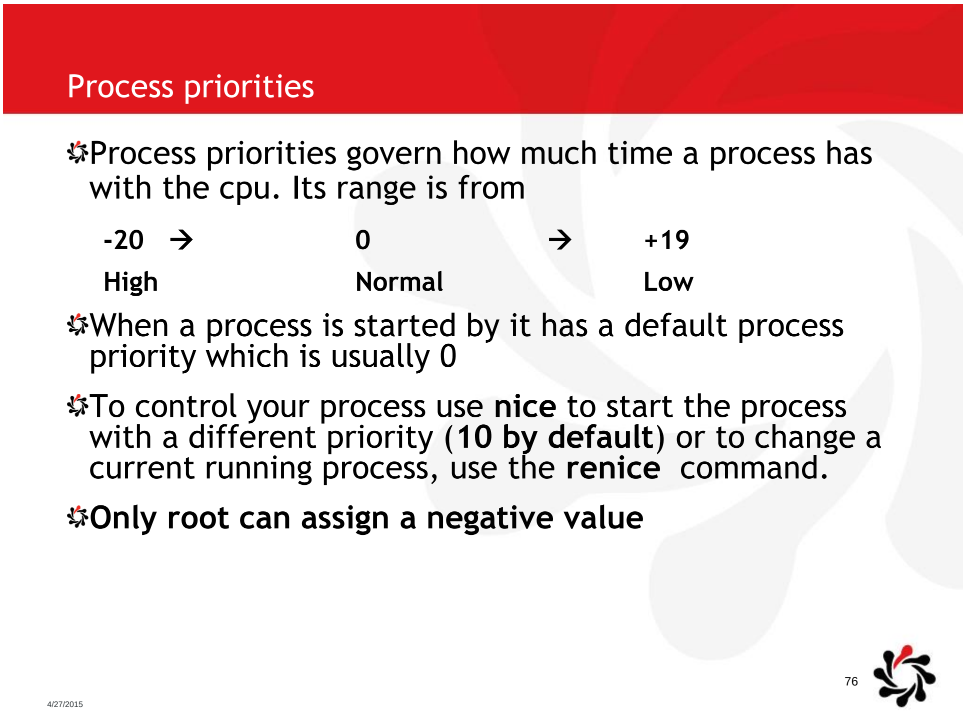
```
#nice  -n -10 updatedb  &
#nice  --10 updatedb &
```

- To give a low priority (positive) to a process

```
#nice -n 15 logrotate &
#nice  -15 logrotate &
```

75

4/27/2015



## Process priorities

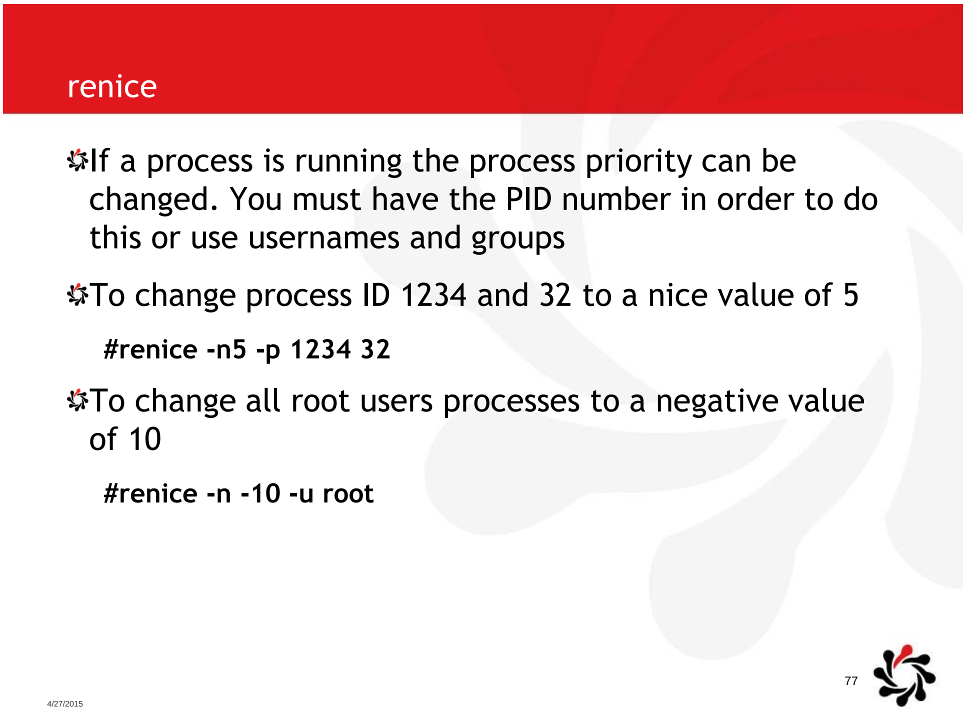
- Process priorities govern how much time a process has with the cpu. Its range is from

| | | | |
|---|---|---|---|
| **-20 →** | **0** | **→** | **+19** |
| **High** | **Normal** | | **Low** |

- When a process is started by it has a default process priority which is usually 0
- To control your process use **nice** to start the process with a different priority (**10 by default**) or to change a current running process, use the **renice** command.
- **Only root can assign a negative value**

76

4/27/2015



## renice

- If a process is running the process priority can be changed. You must have the PID number in order to do this or use usernames and groups
- To change process ID 1234 and 32 to a nice value of 5

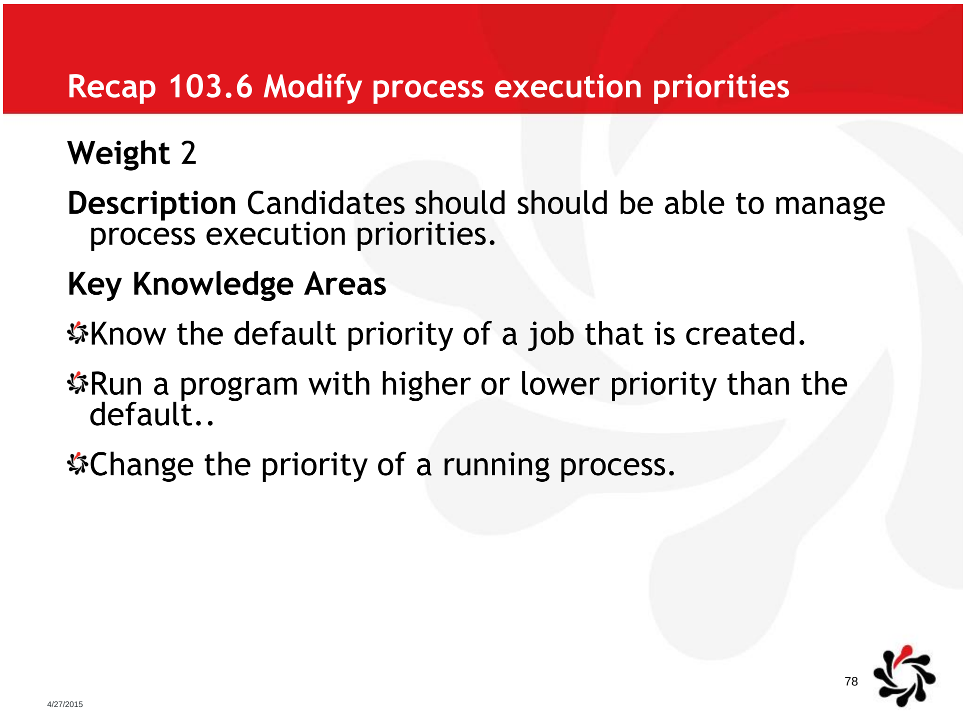
```
#renice -n5 -p 1234 32
```

- To change all root users processes to a negative value of 10

```
#renice -n -10 -u root
```

4/27/2015

77



## Recap 103.6 Modify process execution priorities

**Weight** 2

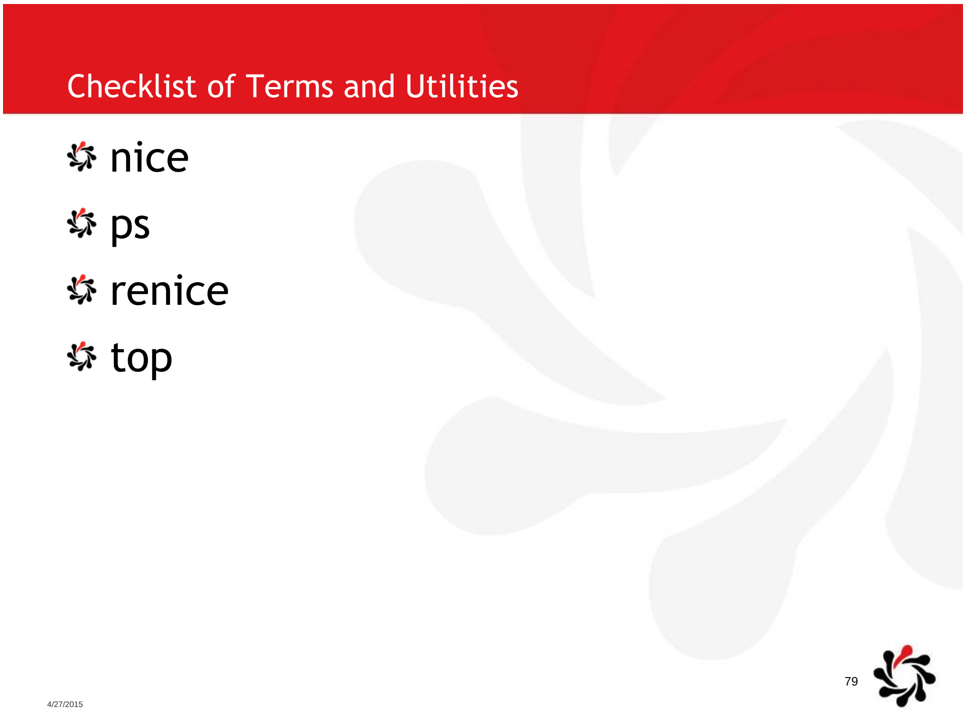
**Description** Candidates should should be able to manage process execution priorities.

**Key Knowledge Areas**

- Know the default priority of a job that is created.
- Run a program with higher or lower priority than the default..
- Change the priority of a running process.

4/27/2015

78

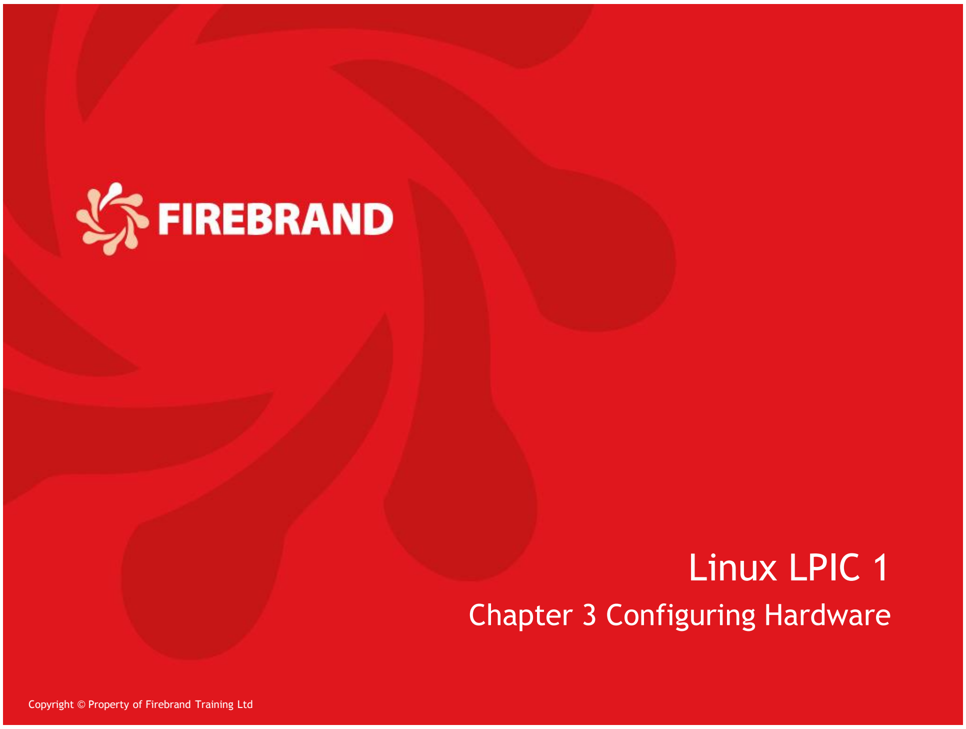


## Checklist of Terms and Utilities

- nice
- ps
- renice
- top

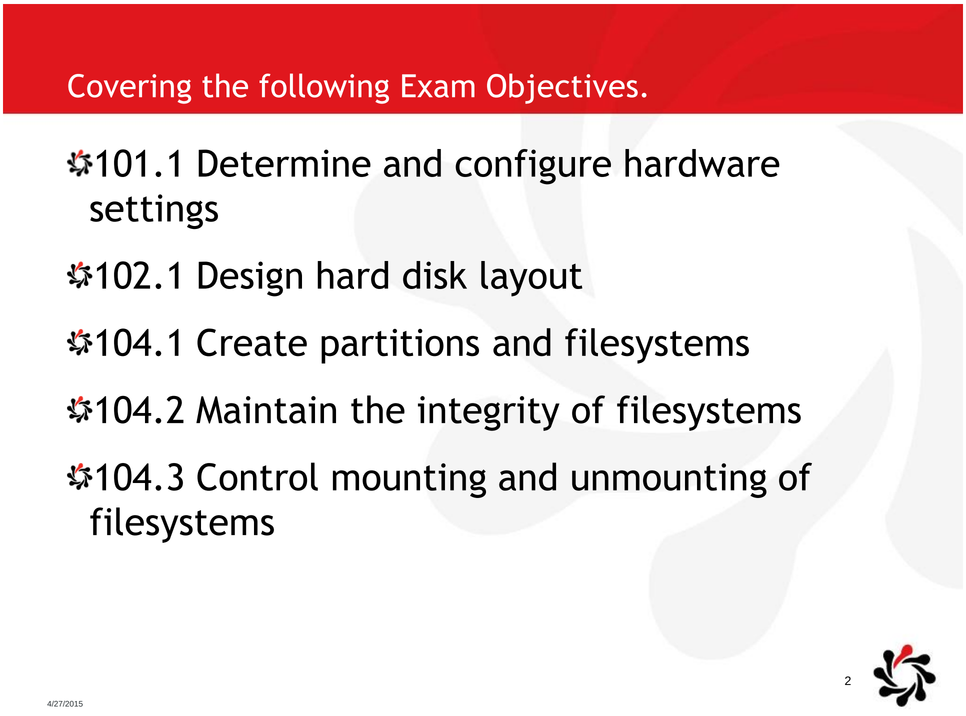
79

4/27/2015



# Linux LPIC 1

## Chapter 3 Configuring Hardware



## Covering the following Exam Objectives.

- 101.1 Determine and configure hardware settings
- 102.1 Design hard disk layout
- 104.1 Create partitions and filesystems
- 104.2 Maintain the integrity of filesystems
- 104.3 Control mounting and unmounting of filesystems

## 101.1 Determine Hardware and configure hardware settings

**Weight 2**

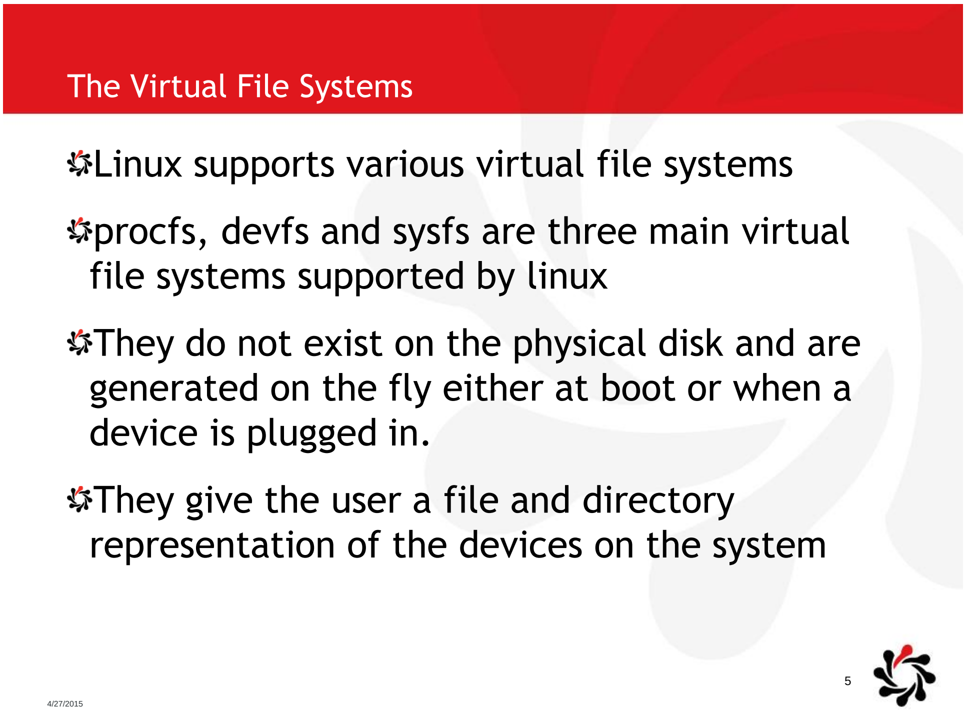
**Description:** Candidates should be able to determine and configure fundamental system hardware.

***Key Knowledge Areas:***

- Enable and disable integrated peripherals
- Config systems with/without external peripherals such as keyboard
- Differentiate between the various types of mass storage devices
- Know the differences between coldplug and hotplug devices
- Determine hardware resources for devices
- Tools and utils to list various hardware information (e.g. lsusb, lspci, etc.)
- Tools and utilities to manipulate USB devices
- Conceptual understanding of sysfs, udev, dbus

## Hardware detection

- Hardware must be detected in order for it to be useable by the Linux kernel.  This section covers how the system works and how to troubleshoot it.

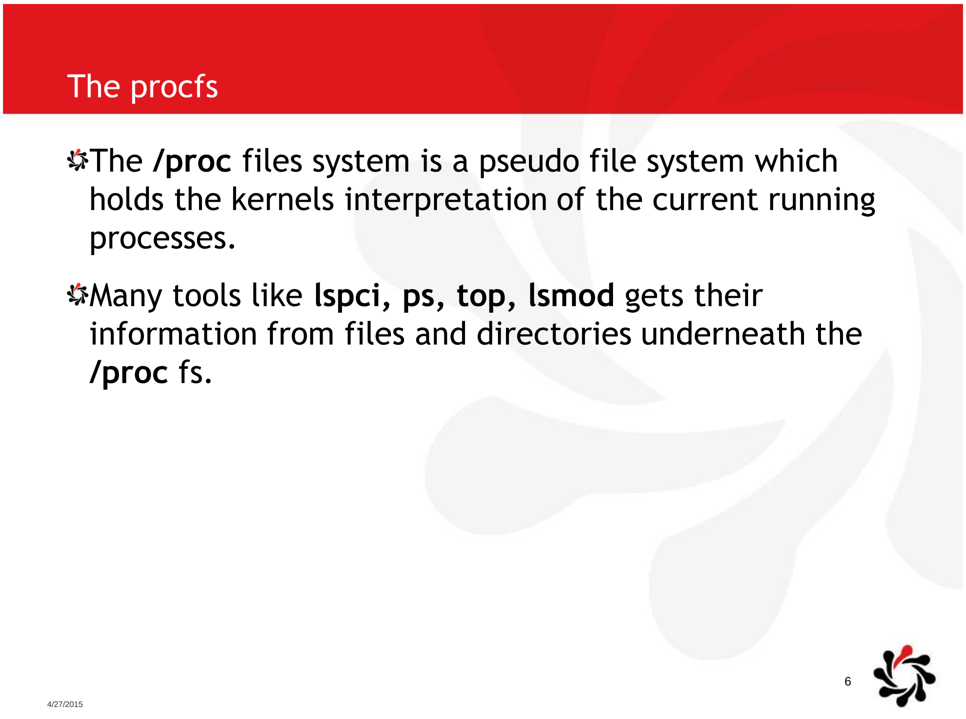


## The Virtual File Systems

- Linux supports various virtual file systems
- procfs, devfs and sysfs are three main virtual file systems supported by linux
- They do not exist on the physical disk and are generated on the fly either at boot or when a device is plugged in.
- They give the user a file and directory representation of the devices on the system

5

4/27/2015

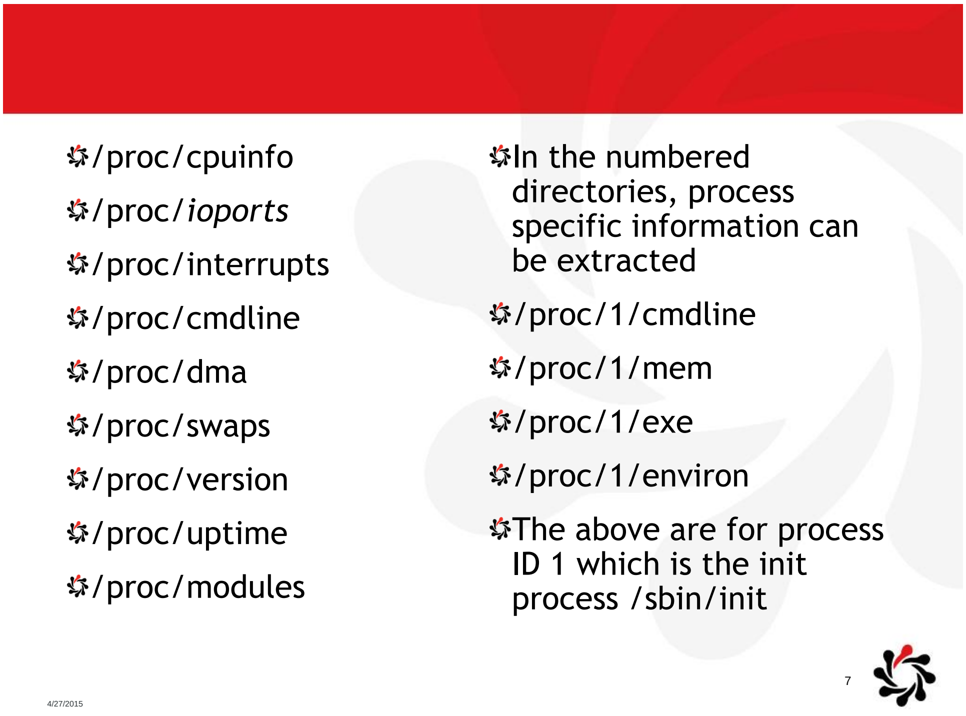


## The procfs

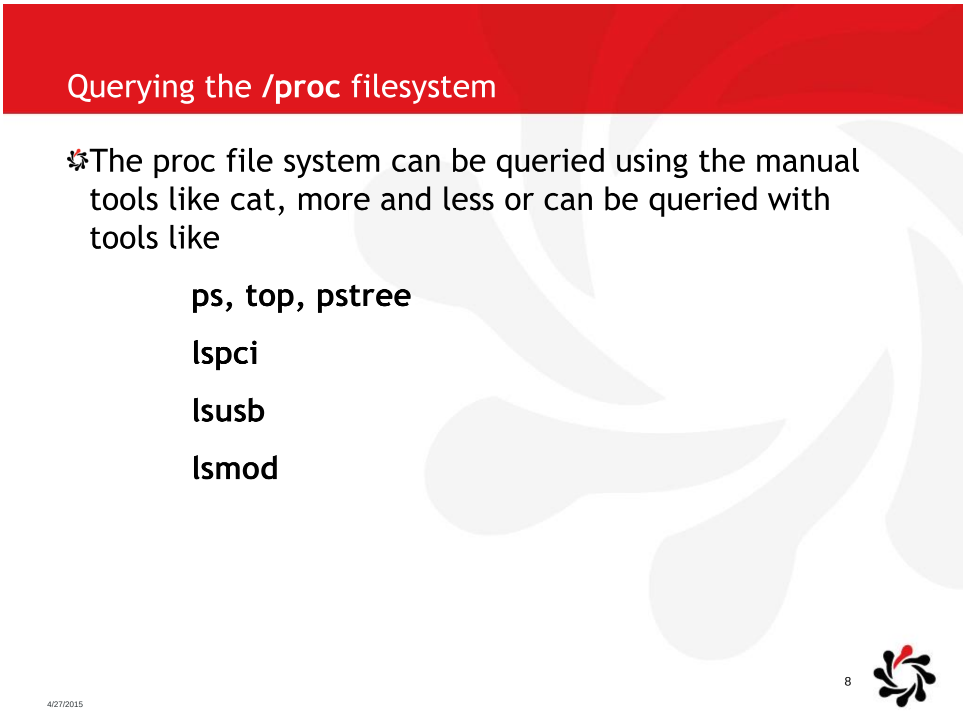
- The **/proc** files system is a pseudo file system which holds the kernels interpretation of the current running processes.
- Many tools like **lspci, ps, top, lsmod** gets their information from files and directories underneath the **/proc** fs.

6

4/27/2015



- /proc/cpuinfo
- /proc/*ioports*
- /proc/interrupts
- /proc/cmdline
- /proc/dma
- /proc/swaps
- /proc/version
- /proc/uptime
- /proc/modules
- In the numbered directories, process specific information can be extracted
- /proc/1/cmdline
- /proc/1/mem
- /proc/1/exe
- /proc/1/environ
- The above are for process ID 1 which is the init process /sbin/init



## Querying the **/proc** filesystem

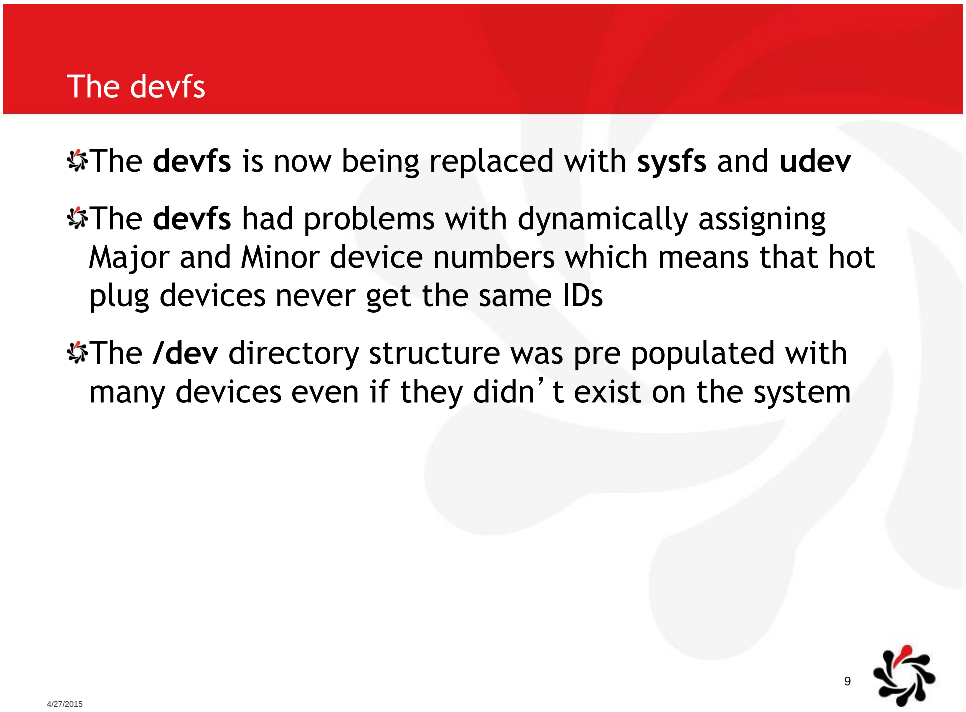
- The proc file system can be queried using the manual tools like cat, more and less or can be queried with tools like

  **ps, top, pstree**

  **lspci**

  **lsusb**

  **lsmod**

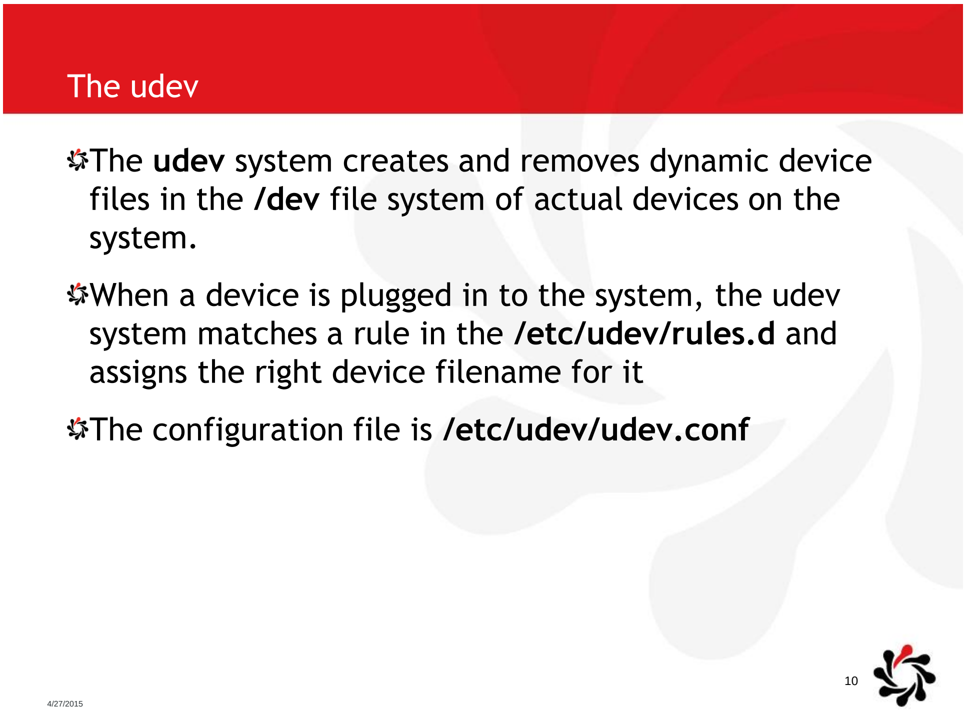


## The devfs

- The **devfs** is now being replaced with **sysfs** and **udev**
- The **devfs** had problems with dynamically assigning Major and Minor device numbers which means that hot plug devices never get the same IDs
- The **/dev** directory structure was pre populated with many devices even if they didn’t exist on the system

9

4/27/2015

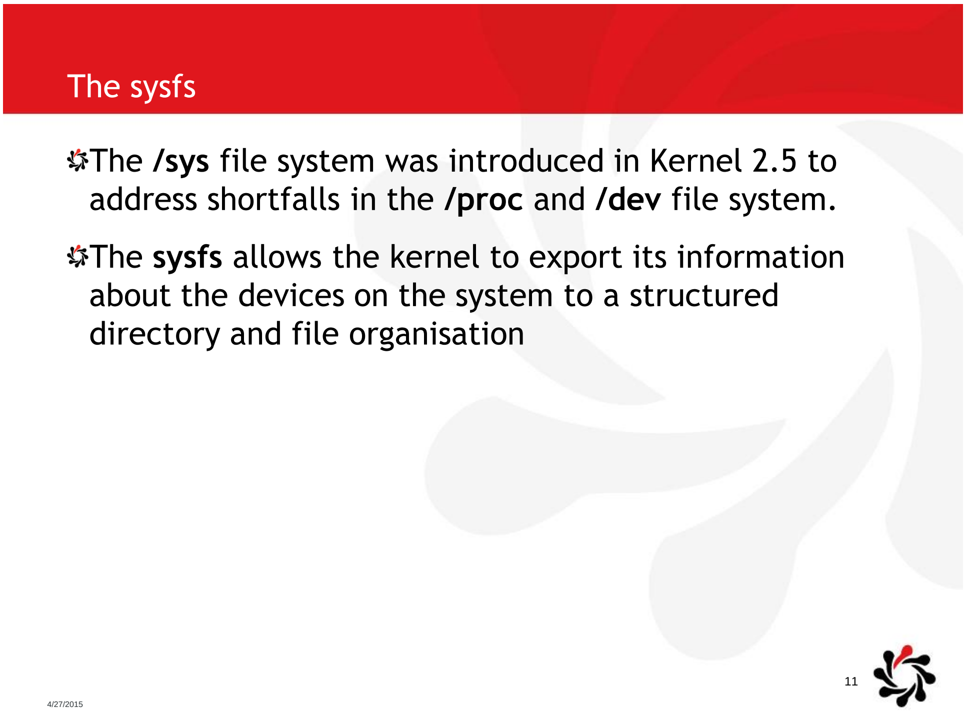


## The udev

- The **udev** system creates and removes dynamic device files in the **/dev** file system of actual devices on the system.
- When a device is plugged in to the system, the udev system matches a rule in the **/etc/udev/rules.d** and assigns the right device filename for it
- The configuration file is **/etc/udev/udev.conf**

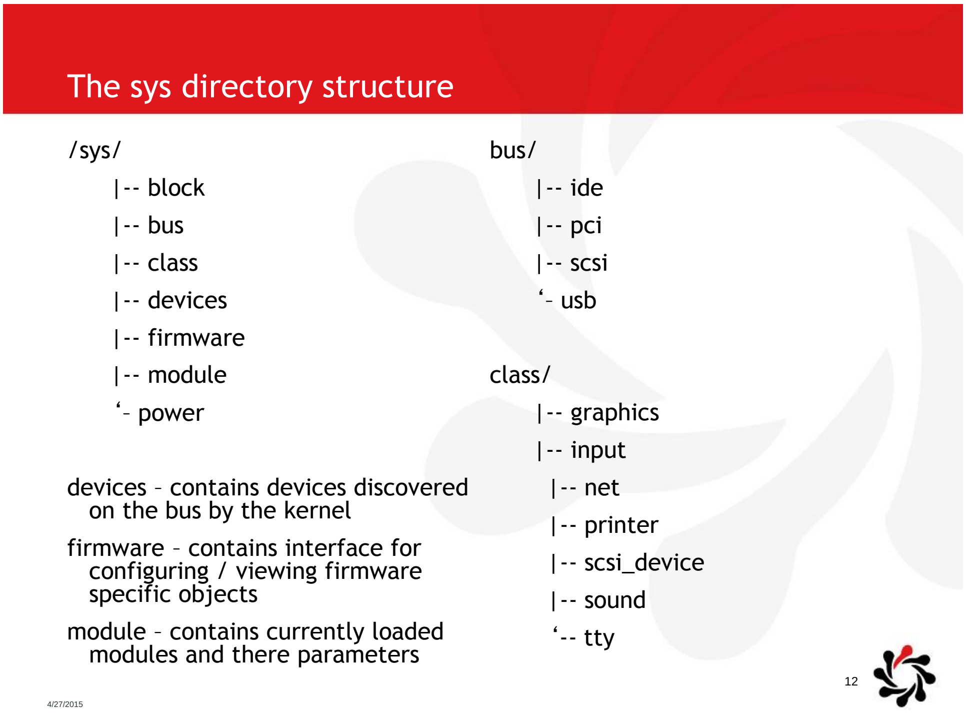
10

4/27/2015



## The sysfs

- The **/sys** file system was introduced in Kernel 2.5 to address shortfalls in the **/proc** and **/dev** file system.
- The **sysfs** allows the kernel to export its information about the devices on the system to a structured directory and file organisation



## The sys directory structure

```
/sys/
    |-- block
    |-- bus
    |-- class
    |-- devices
    |-- firmware
    |-- module
    ‘- power
```

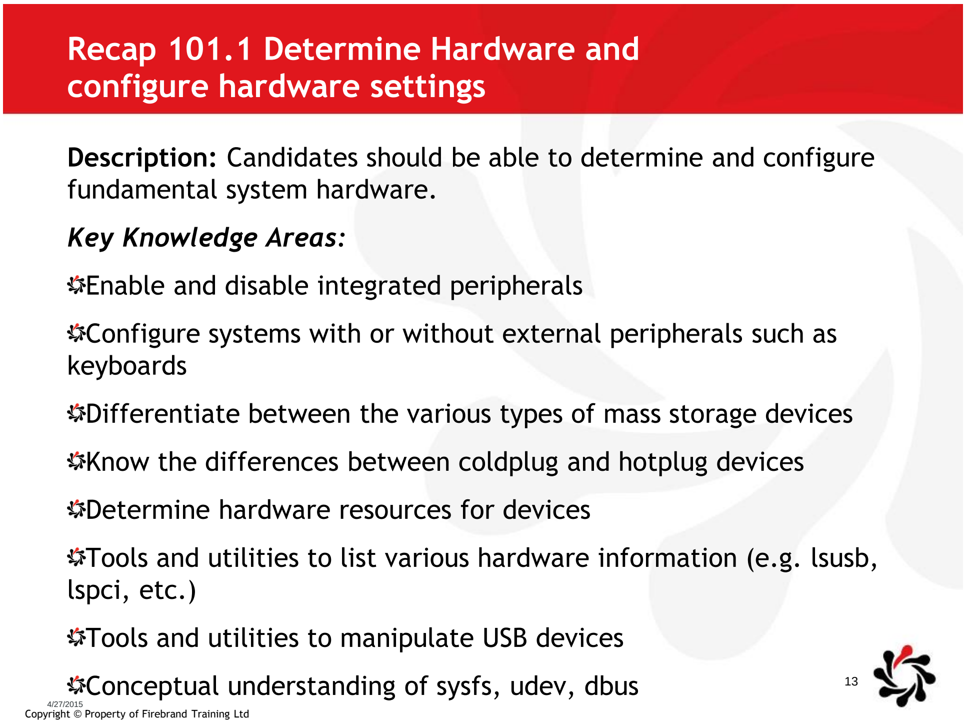
devices – contains devices discovered on the bus by the kernel

firmware – contains interface for configuring / viewing firmware specific objects

module – contains currently loaded modules and there parameters

```
bus/
    |-- ide
    |-- pci
    |-- scsi
    ‘- usb

class/
    |-- graphics
    |-- input
      |-- net
      |-- printer
      |-- scsi_device
      |-- sound
      ‘-- tty
```



## Recap 101.1 Determine Hardware and configure hardware settings

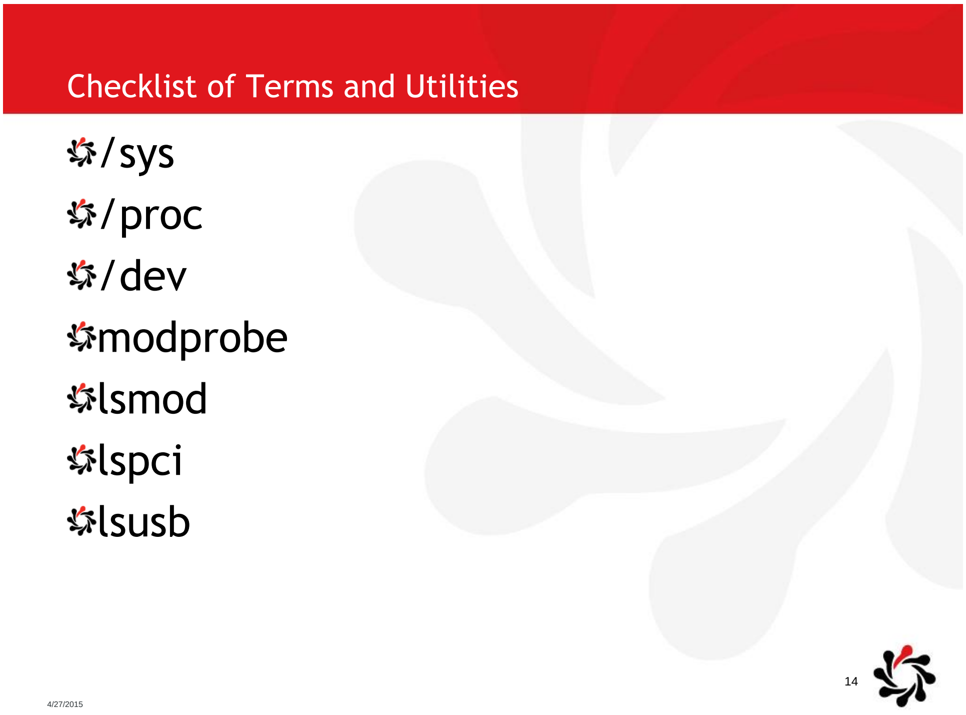
**Description:** Candidates should be able to determine and configure fundamental system hardware.

***Key Knowledge Areas:***

- Enable and disable integrated peripherals
- Configure systems with or without external peripherals such as keyboards
- Differentiate between the various types of mass storage devices
- Know the differences between coldplug and hotplug devices
- Determine hardware resources for devices
- Tools and utilities to list various hardware information (e.g. lsusb, lspci, etc.)
- Tools and utilities to manipulate USB devices
- Conceptual understanding of sysfs, udev, dbus

4/27/2015

13

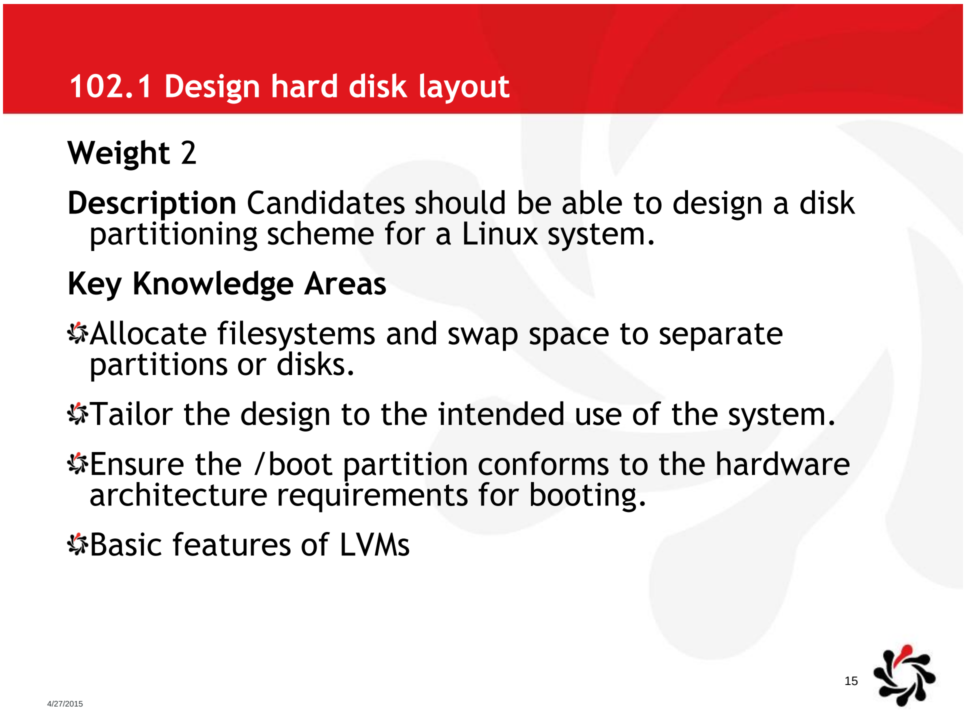


## Checklist of Terms and Utilities

- /sys
- /proc
- /dev
- modprobe
- lsmod
- lspci
- lsusb

14

4/27/2015



## 102.1 Design hard disk layout

**Weight** 2

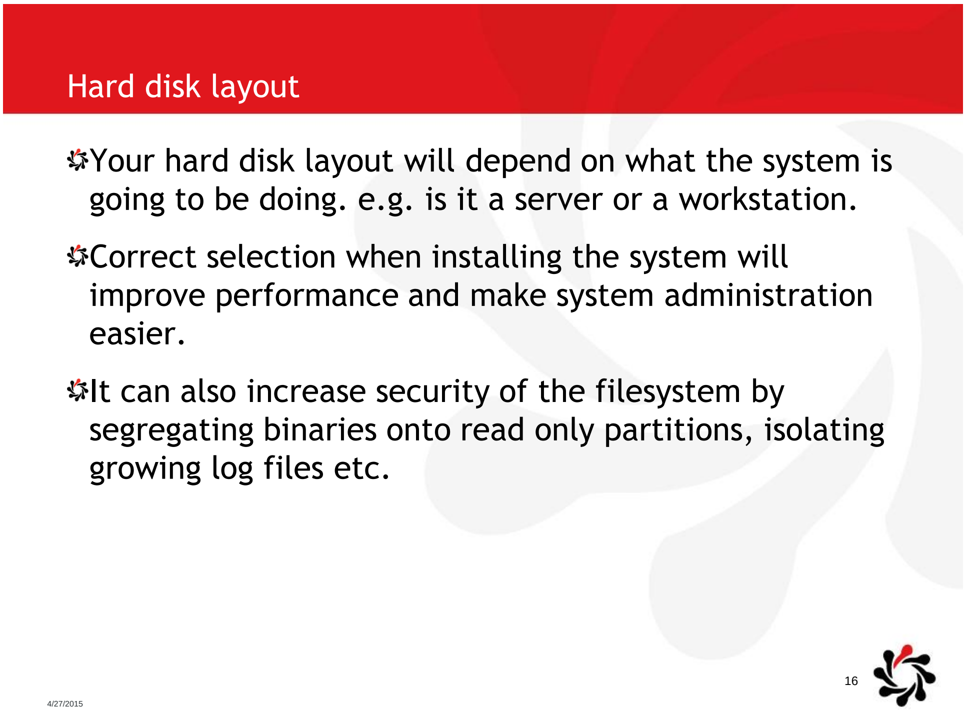
**Description** Candidates should be able to design a disk partitioning scheme for a Linux system.

**Key Knowledge Areas**

- Allocate filesystems and swap space to separate partitions or disks.
- Tailor the design to the intended use of the system.
- Ensure the /boot partition conforms to the hardware architecture requirements for booting.
- Basic features of LVMs

15

4/27/2015

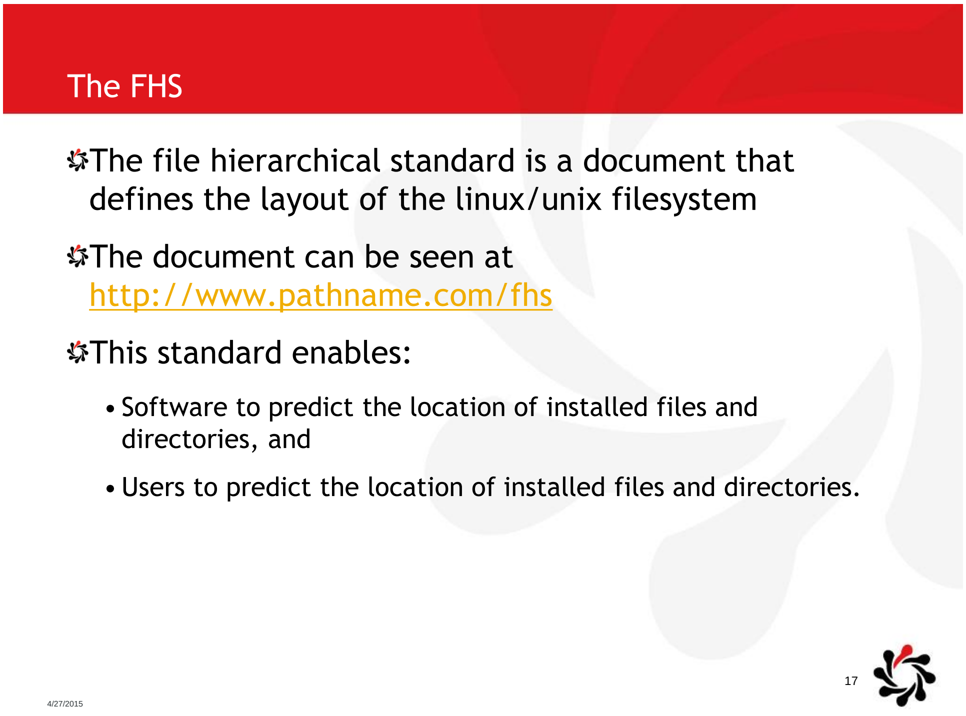


## Hard disk layout

- Your hard disk layout will depend on what the system is going to be doing. e.g. is it a server or a workstation.
- Correct selection when installing the system will improve performance and make system administration easier.
- It can also increase security of the filesystem by segregating binaries onto read only partitions, isolating growing log files etc.

16

4/27/2015

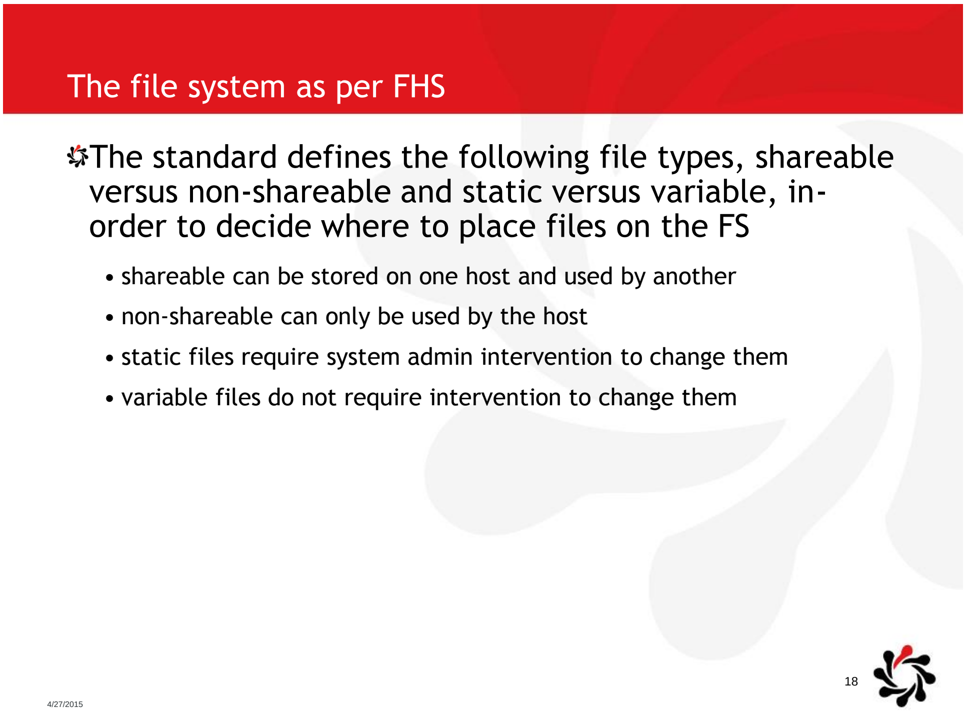


## The FHS

- The file hierarchical standard is a document that defines the layout of the linux/unix filesystem
- The document can be seen at [http://www.pathname.com/fhs](http://www.pathname.com/fhs)
- This standard enables:
  - Software to predict the location of installed files and directories, and
  - Users to predict the location of installed files and directories.

4/27/2015

17

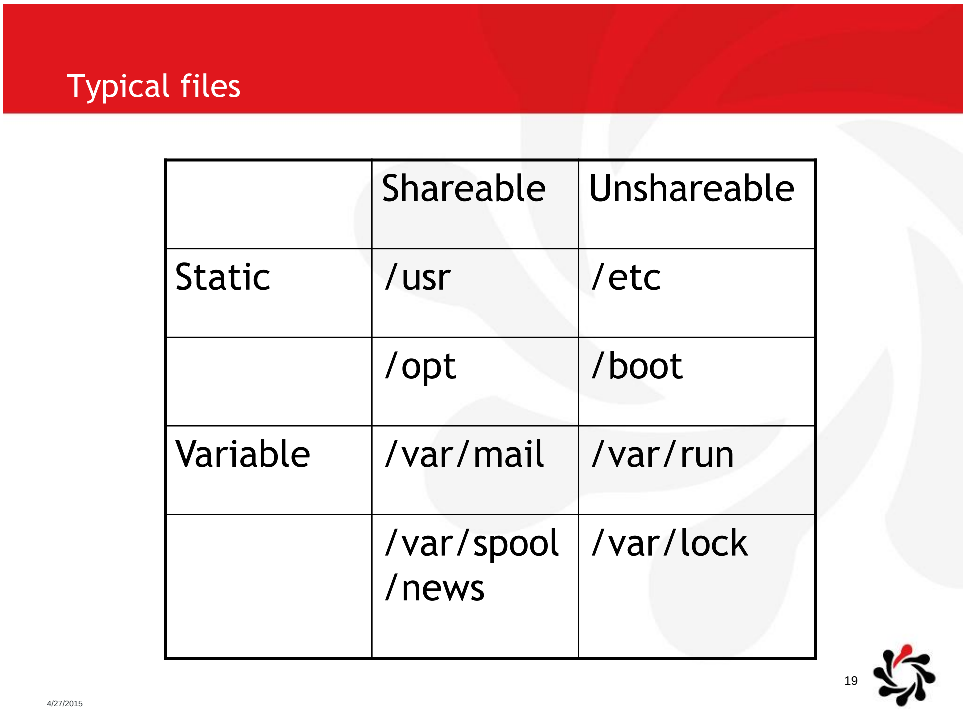


## The file system as per FHS

- The standard defines the following file types, shareable versus non-shareable and static versus variable, in-order to decide where to place files on the FS
  - shareable can be stored on one host and used by another
  - non-shareable can only be used by the host
  - static files require system admin intervention to change them
  - variable files do not require intervention to change them

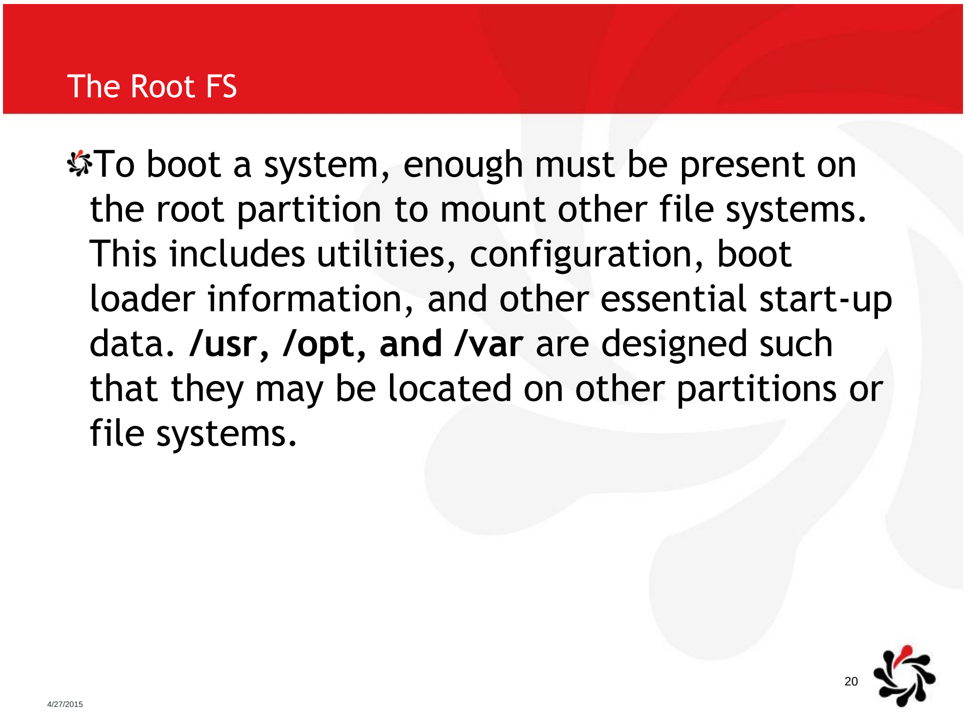
4/27/2015

18



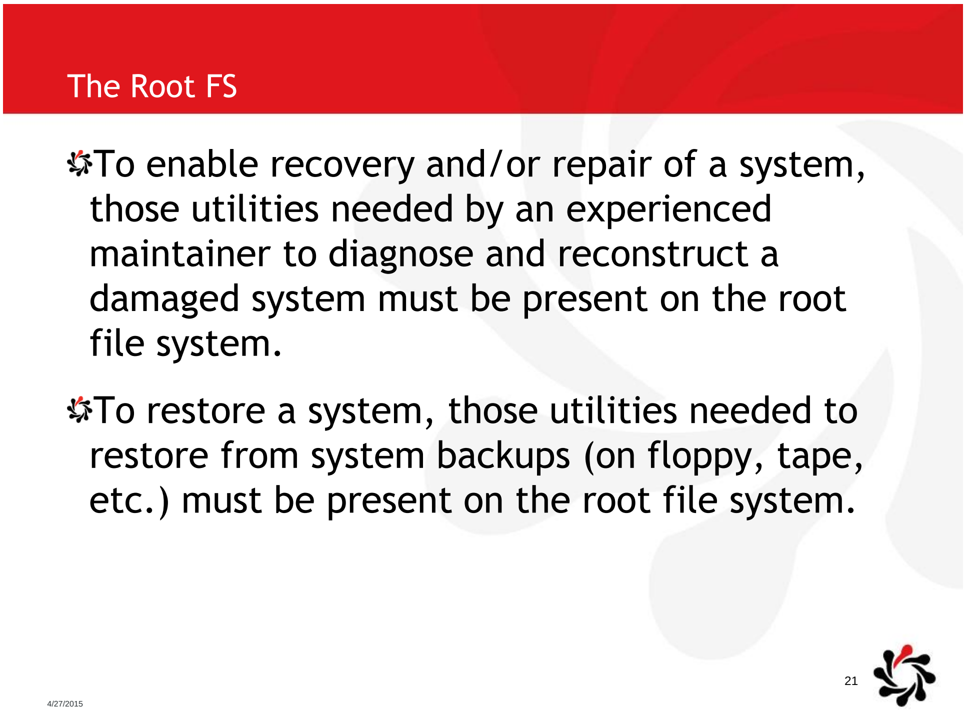
## Typical files

| | Shareable | Unshareable |
|---|---|---|
| Static | /usr | /etc |
| | /opt | /boot |
| Variable | /var/mail | /var/run |
| | /var/spool<br>/news | /var/lock |



## The Root FS

- To boot a system, enough must be present on the root partition to mount other file systems. This includes utilities, configuration, boot loader information, and other essential start-up data. **/usr, /opt, and /var** are designed such that they may be located on other partitions or file systems.

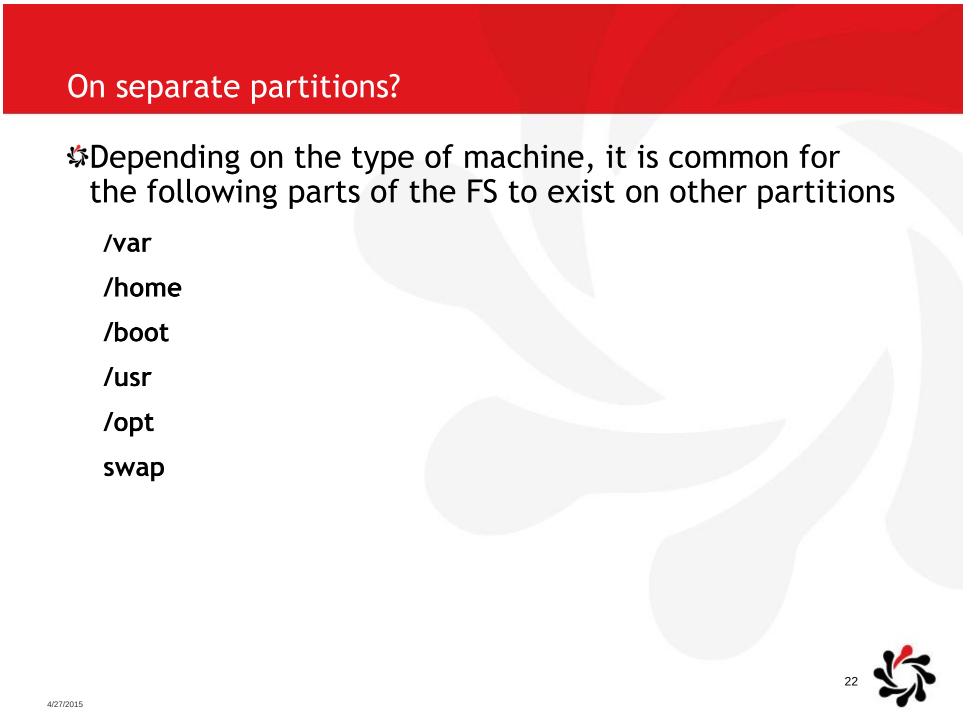


## The Root FS

- To enable recovery and/or repair of a system, those utilities needed by an experienced maintainer to diagnose and reconstruct a damaged system must be present on the root file system.
- To restore a system, those utilities needed to restore from system backups (on floppy, tape, etc.) must be present on the root file system.

4/27/2015

21

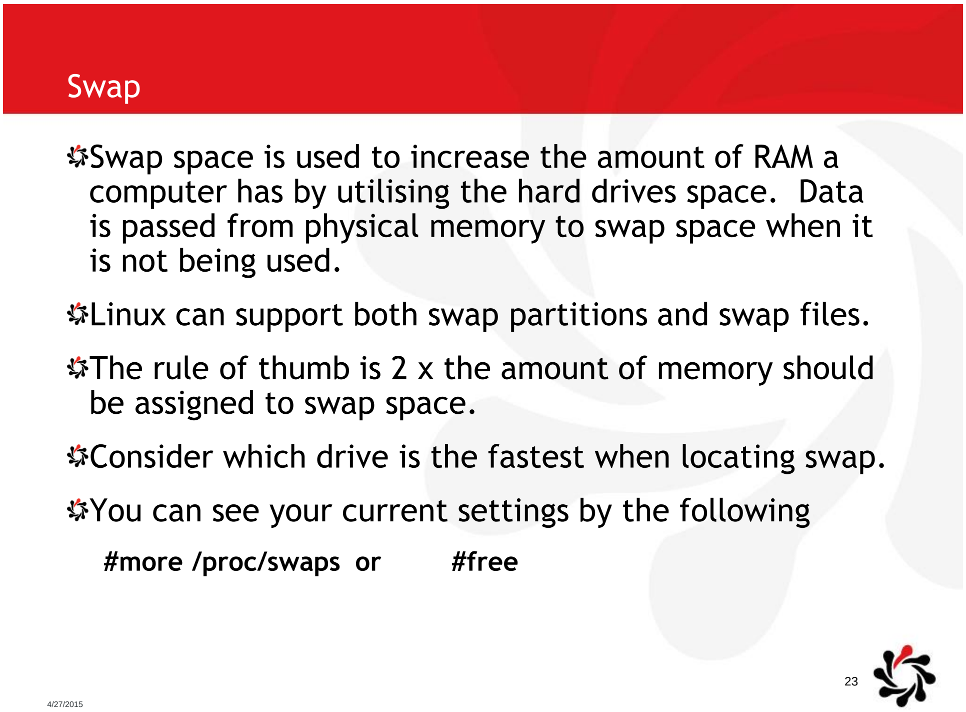


## On separate partitions?

- Depending on the type of machine, it is common for the following parts of the FS to exist on other partitions
  - /var
  - /home
  - /boot
  - /usr
  - /opt
  - swap

4/27/2015

22



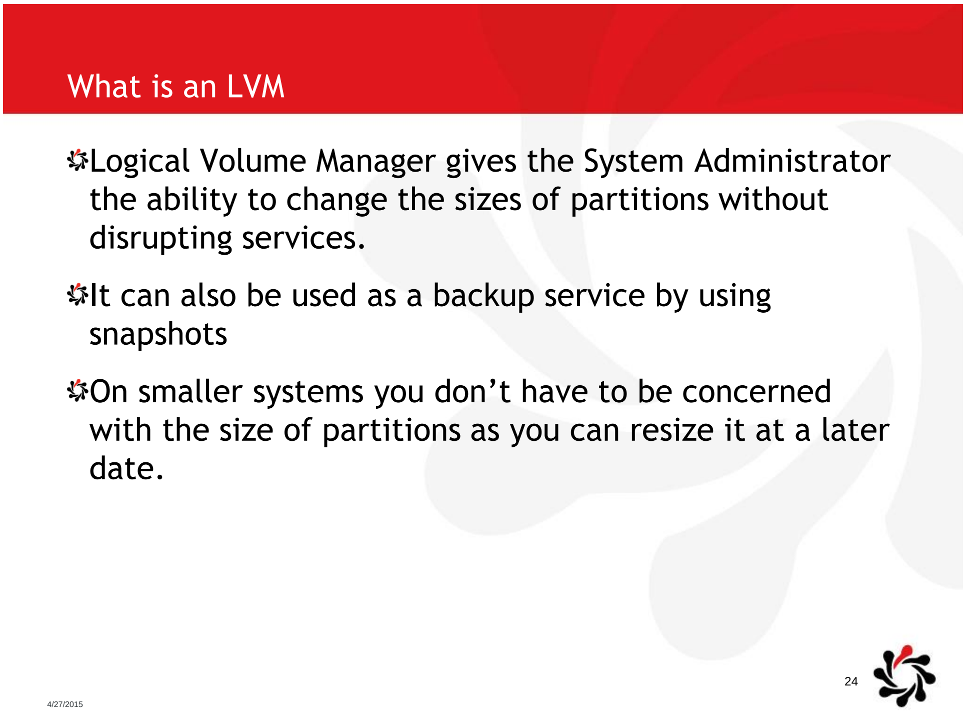
## Swap

- Swap space is used to increase the amount of RAM a computer has by utilising the hard drives space.  Data is passed from physical memory to swap space when it is not being used.
- Linux can support both swap partitions and swap files.
- The rule of thumb is 2 x the amount of memory should be assigned to swap space.
- Consider which drive is the fastest when locating swap.
- You can see your current settings by the following

```
#more /proc/swaps  or        #free
```

4/27/2015

23



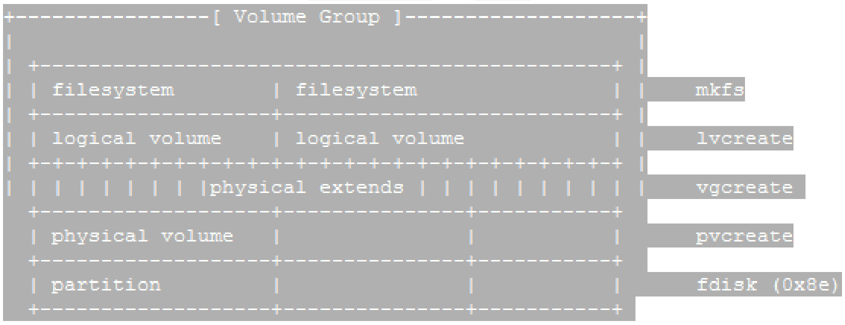
## What is an LVM

- Logical Volume Manager gives the System Administrator the ability to change the sizes of partitions without disrupting services.
- It can also be used as a backup service by using snapshots
- On smaller systems you don’t have to be concerned with the size of partitions as you can resize it at a later date.

4/27/2015

24

## Overview of a Volume group



```
+----------------[ Volume Group ]-------------------+
|                                                   |
| +-----------------------------------------------+ |
| | filesystem        | filesystem                | |   mkfs
| +-------------------+---------------------------+ |
| | logical volume    | logical volume            | |   lvcreate
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
| | | | | | | | |physical extends | | | | | | | | | |   vgcreate
  +-------------------+---------------+-----------+
  | physical volume   |               |           |   pvcreate
  +-------------------+---------------+-----------+
  | partition         |               |           |   fdisk (0x8e)
  +-------------------+---------------+-----------+
```

4/27/2015

## Creating an LVM

- An LVM can be created with a minimum of one partition, but to see its full use, then more than one partition is required.
- If you create an LVM on a single partition, you can, at a later date add another physical volume to the original LVM to extend the original capacity.

4/27/2015

## Volume snapshot

- An LVM snapshot is an exact copy of an LVM partition that has all the data from the LVM volume from the time the snapshot was created.
- The advantage of LVM snapshots is that they can be used to greatly reduce the amount of time that your services/databases are down during backups because a snapshot is usually created in fractions of a second.
- After the snapshot has been created, you can back up the snapshot while your services and databases are in normal operation.

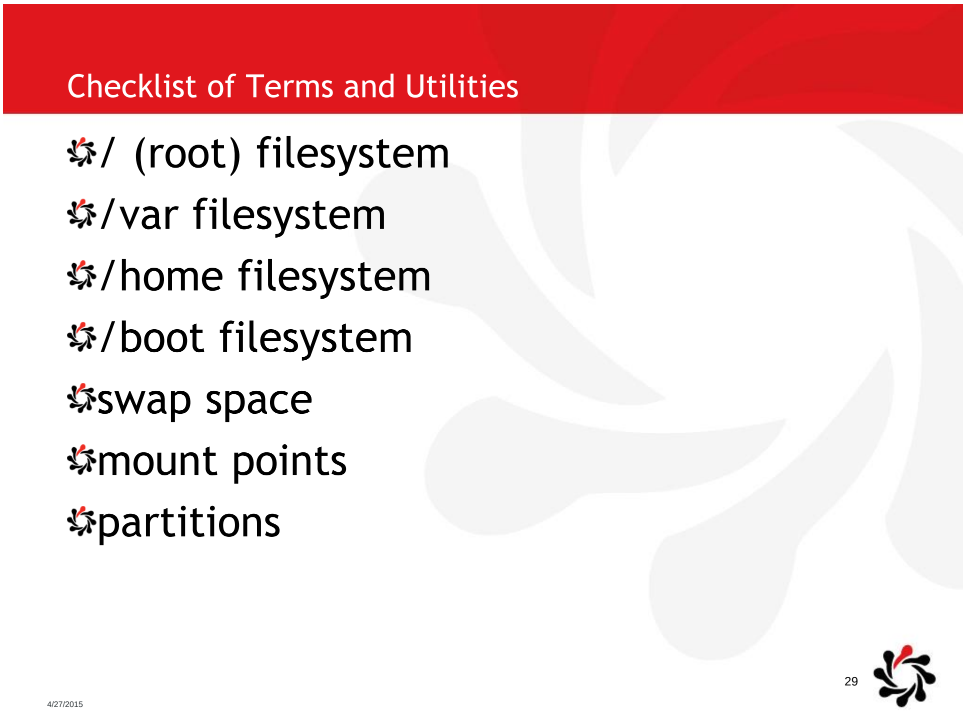
## Recap 102.1 Design hard disk layout

**Weight** 2

**Description** Candidates should be able to design a disk partitioning scheme for a Linux system.

**Key Knowledge Areas**

- Allocate filesystems and swap space to separate partitions or disks.
- Tailor the design to the intended use of the system.
- Ensure the /boot partition conforms to the hardware architecture requirements for booting.
- Basic features of LVMs

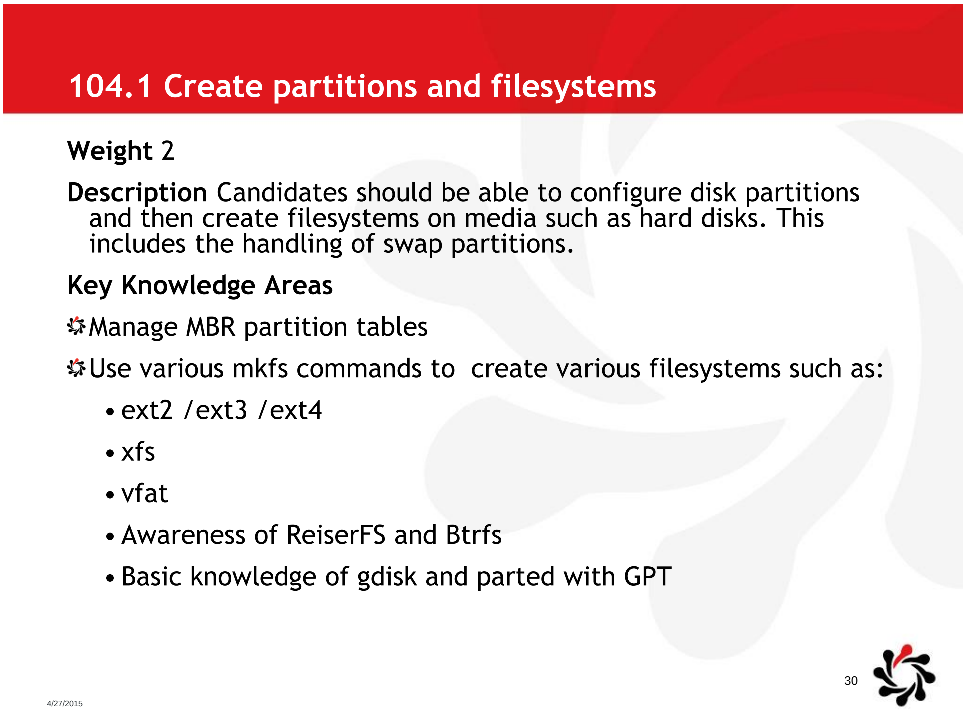


## Checklist of Terms and Utilities

- / (root) filesystem
- /var filesystem
- /home filesystem
- /boot filesystem
- swap space
- mount points
- partitions

4/27/2015

29



## 104.1 Create partitions and filesystems

**Weight** 2

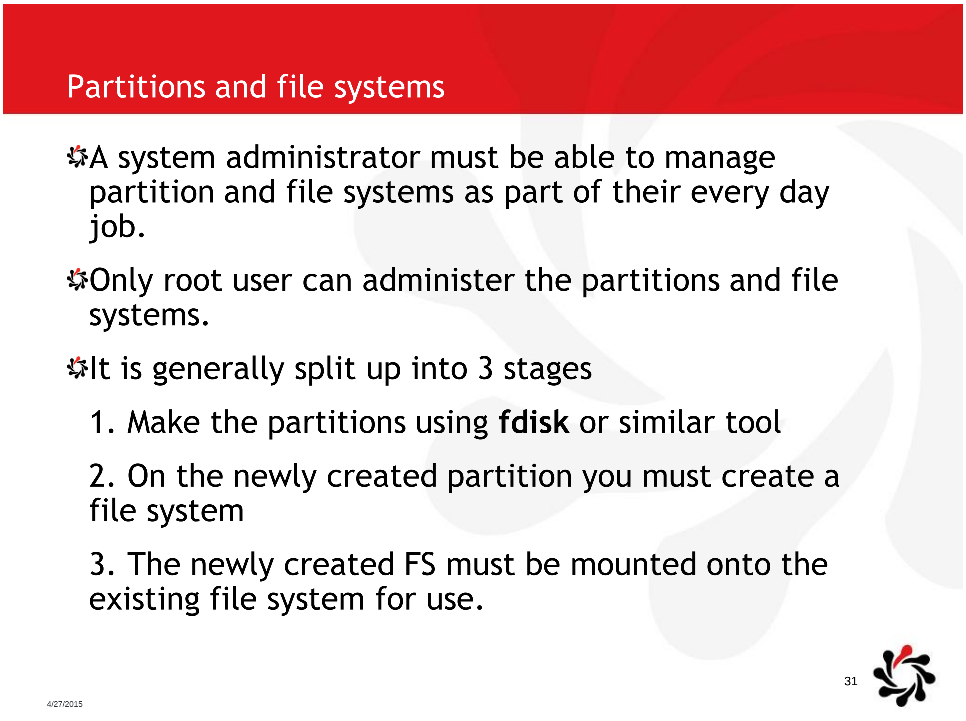
**Description** Candidates should be able to configure disk partitions and then create filesystems on media such as hard disks. This includes the handling of swap partitions.

**Key Knowledge Areas**

- Manage MBR partition tables
- Use various mkfs commands to create various filesystems such as:
  - ext2 /ext3 /ext4
  - xfs
  - vfat
  - Awareness of ReiserFS and Btrfs
  - Basic knowledge of gdisk and parted with GPT

4/27/2015

30



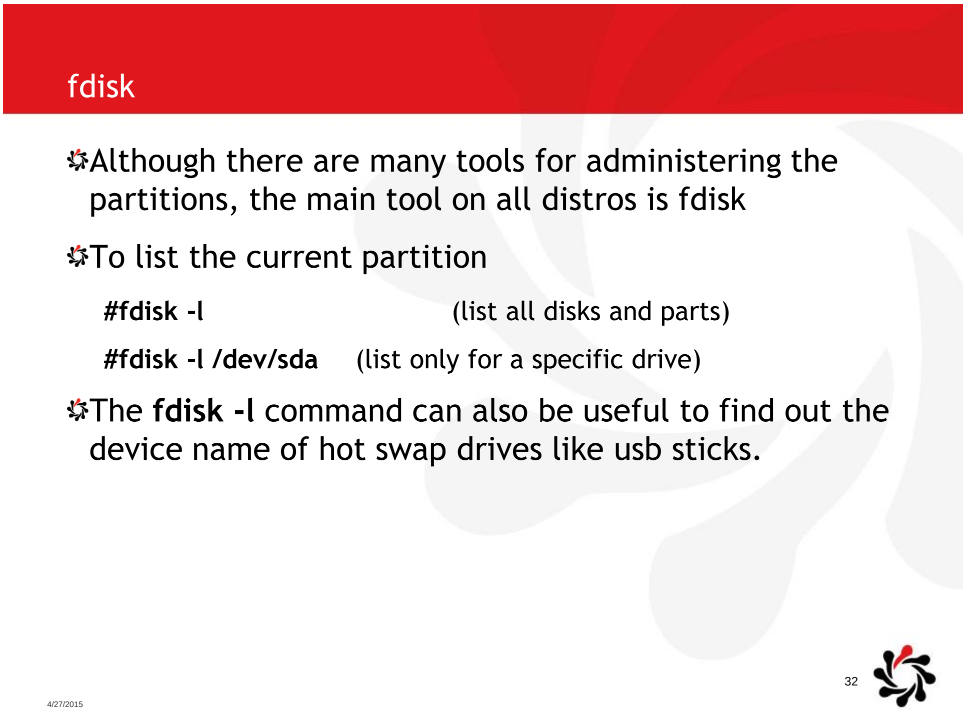
## Partitions and file systems

- A system administrator must be able to manage partition and file systems as part of their every day job.
- Only root user can administer the partitions and file systems.
- It is generally split up into 3 stages

  1. Make the partitions using **fdisk** or similar tool

  2. On the newly created partition you must create a file system

  3. The newly created FS must be mounted onto the existing file system for use.



## fdisk

- Although there are many tools for administering the partitions, the main tool on all distros is fdisk
- To list the current partition

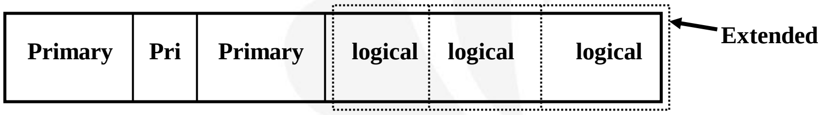
```
#fdisk -l                    (list all disks and parts)
#fdisk -l /dev/sda     (list only for a specific drive)
```

- The **fdisk -l** command can also be useful to find out the device name of hot swap drives like usb sticks.

## Creating a new partition

- You can only have up to 4 primary partitions on a drive.
- If you require more than 4, then an extended partition must exist as one of the four primary partitions
- Inside the extended partition, you must create a logical drive.
- The 1st logical drive within the extended partition will always be **5**

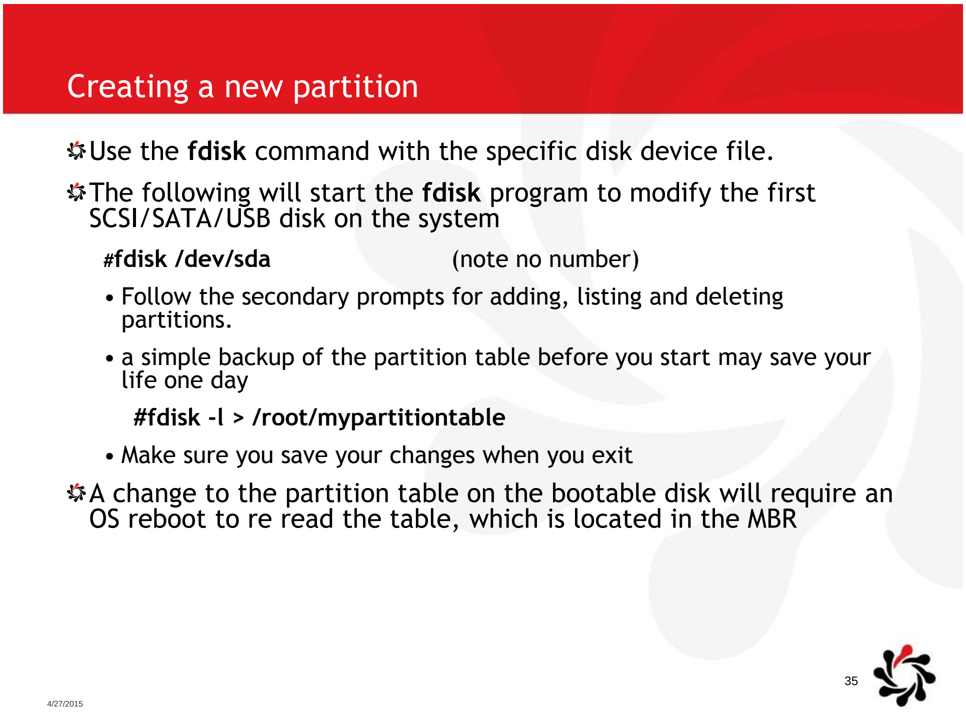
## A typical disk



Primary | Pri | Primary | logical | logical | logical

Extended

- The above represents a primary master IDE drive with the following device names
- **/dev/hda1** Primary Partition
- **/dev/hda2** Primary Partition
- **/dev/hda3** Primary Partition
- **/dev/hda4** Extended Container
- **/dev/hda5** 1st Logical drive within the extended partition
- **/dev/hda6** 2nd Logical drive within the extended partition
- **/dev/hda7** Last logical drive within the extended partition



## Creating a new partition

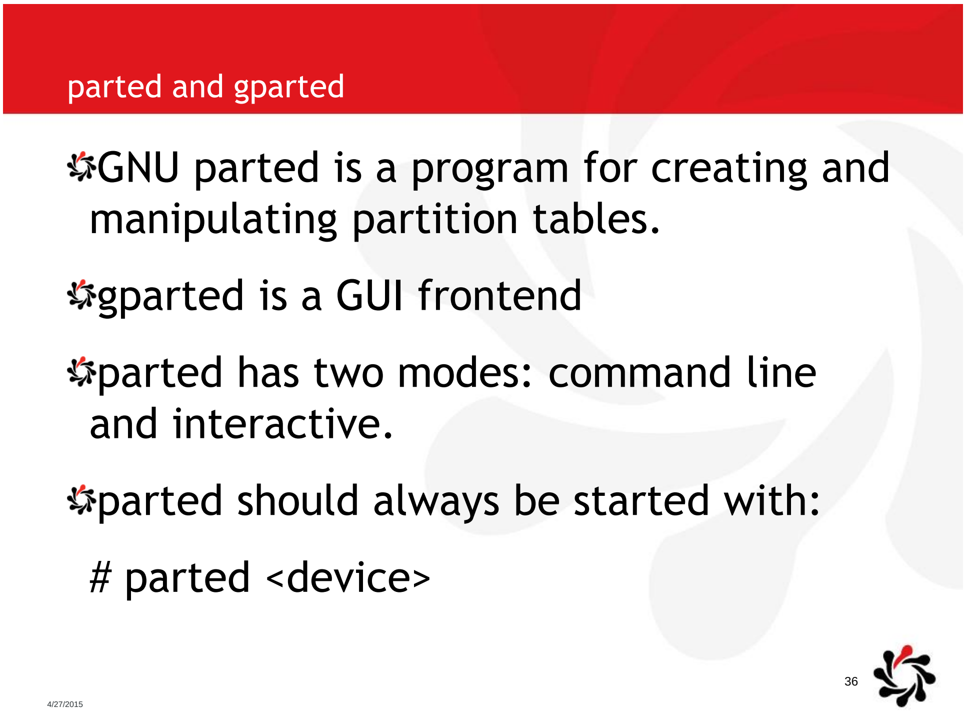
- Use the **fdisk** command with the specific disk device file.
- The following will start the **fdisk** program to modify the first SCSI/SATA/USB disk on the system

  **#fdisk /dev/sda** (note no number)

  - Follow the secondary prompts for adding, listing and deleting partitions.
  - a simple backup of the partition table before you start may save your life one day

    **#fdisk -l > /root/mypartitiontable**

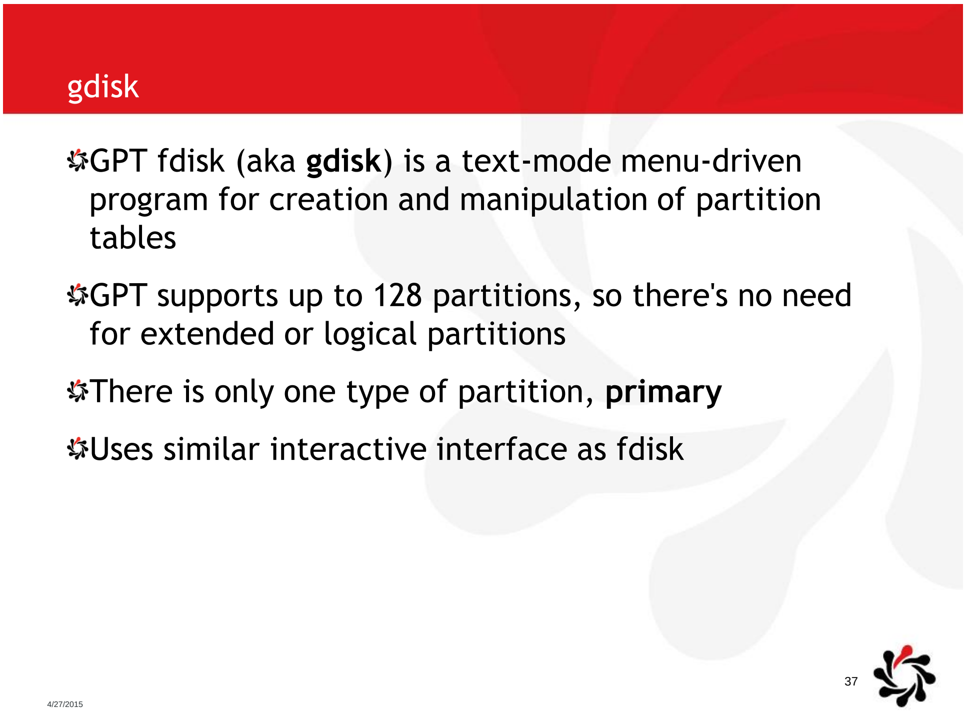
  - Make sure you save your changes when you exit
- A change to the partition table on the bootable disk will require an OS reboot to re read the table, which is located in the MBR



## parted and gparted

- GNU parted is a program for creating and manipulating partition tables.
- gparted is a GUI frontend
- parted has two modes: command line and interactive.
- parted should always be started with:

  # parted <device>

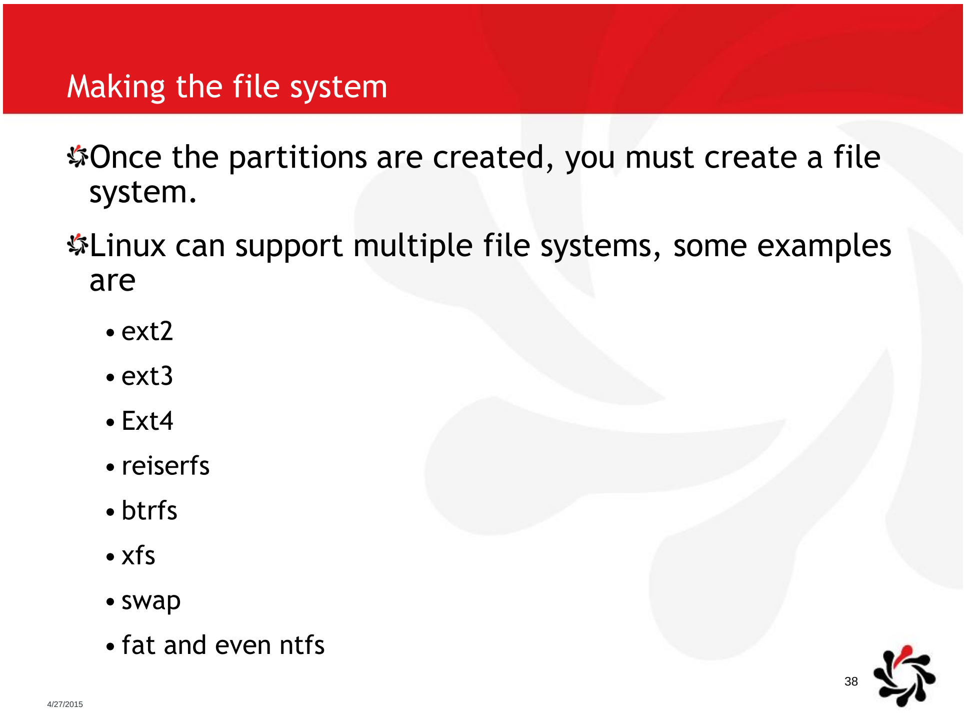


## gdisk

- GPT fdisk (aka **gdisk**) is a text-mode menu-driven program for creation and manipulation of partition tables
- GPT supports up to 128 partitions, so there's no need for extended or logical partitions
- There is only one type of partition, **primary**
- Uses similar interactive interface as fdisk

37

4/27/2015

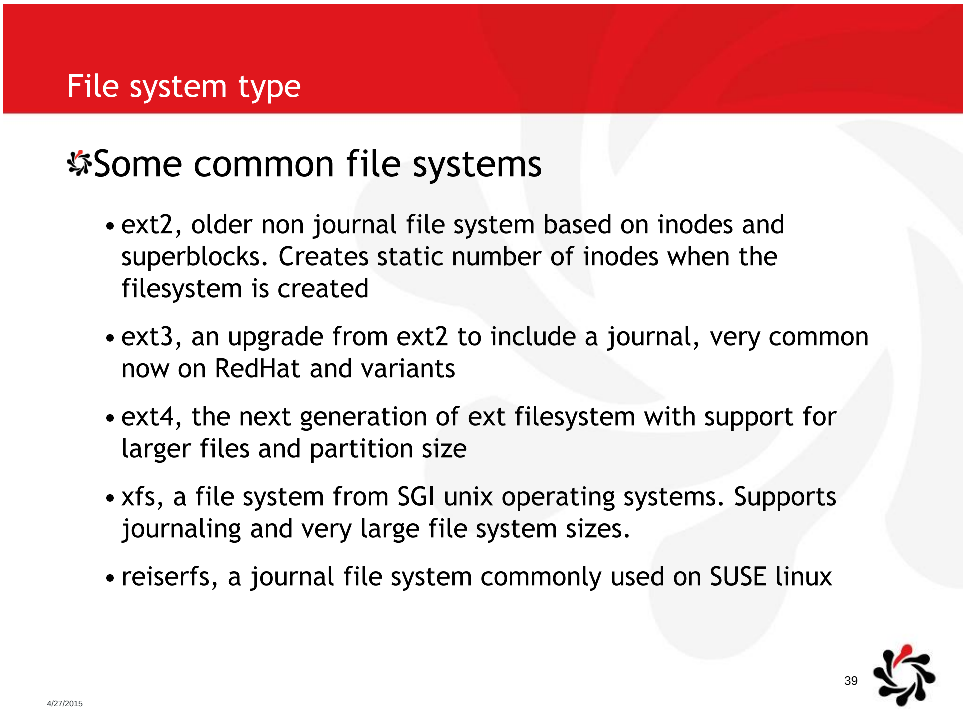


## Making the file system

- Once the partitions are created, you must create a file system.
- Linux can support multiple file systems, some examples are
  - ext2
  - ext3
  - Ext4
  - reiserfs
  - btrfs
  - xfs
  - swap
  - fat and even ntfs

38

4/27/2015

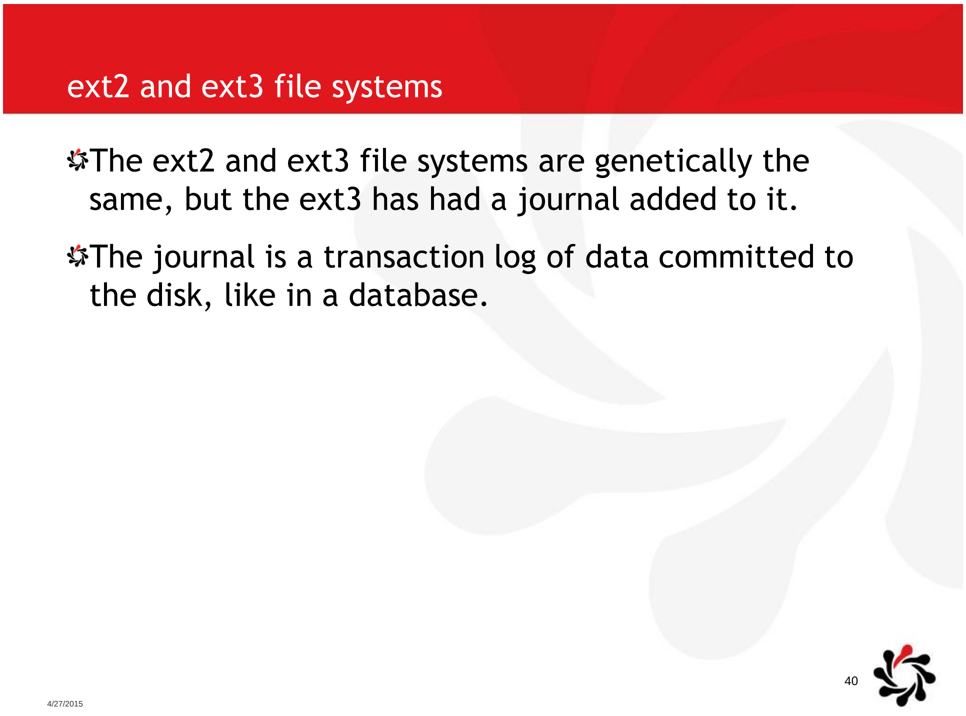


## File system type

- Some common file systems
  - ext2, older non journal file system based on inodes and superblocks. Creates static number of inodes when the filesystem is created
  - ext3, an upgrade from ext2 to include a journal, very common now on RedHat and variants
  - ext4, the next generation of ext filesystem with support for larger files and partition size
  - xfs, a file system from SGI unix operating systems. Supports journaling and very large file system sizes.
  - reiserfs, a journal file system commonly used on SUSE linux

39

4/27/2015

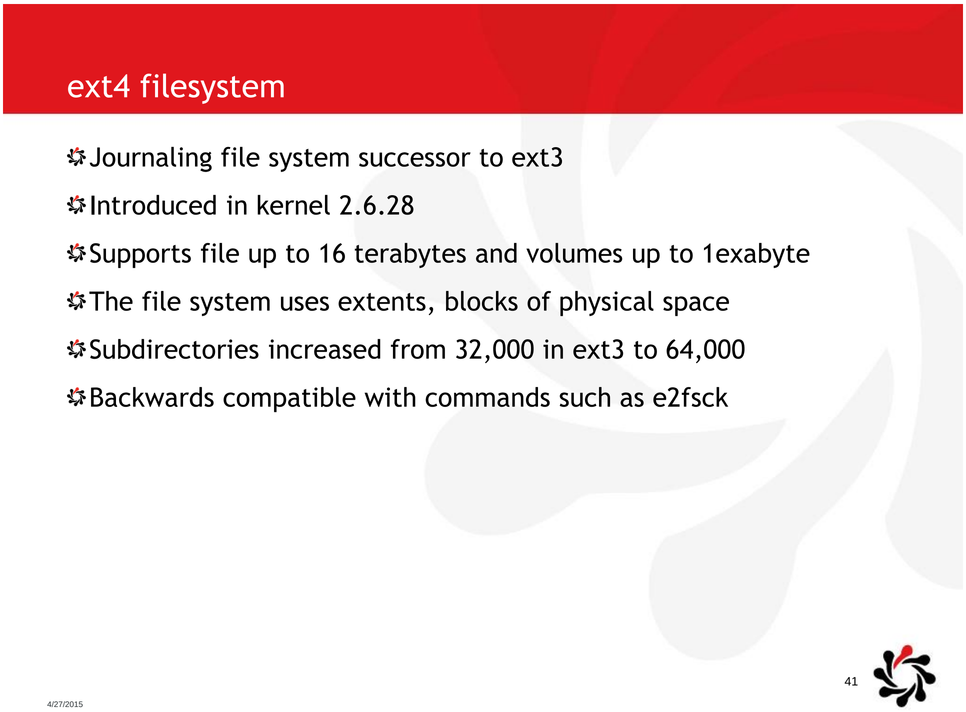


## ext2 and ext3 file systems

- The ext2 and ext3 file systems are genetically the same, but the ext3 has had a journal added to it.
- The journal is a transaction log of data committed to the disk, like in a database.

40

4/27/2015

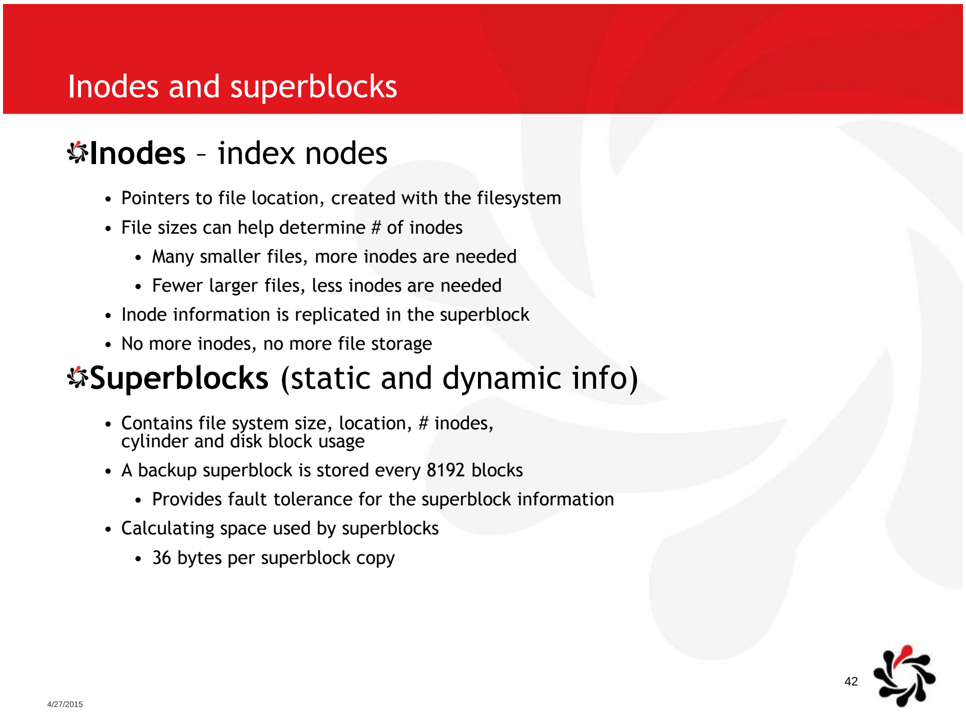


## ext4 filesystem

- Journaling file system successor to ext3
- Introduced in kernel 2.6.28
- Supports file up to 16 terabytes and volumes up to 1exabyte
- The file system uses extents, blocks of physical space
- Subdirectories increased from 32,000 in ext3 to 64,000
- Backwards compatible with commands such as e2fsck

4/27/2015

41

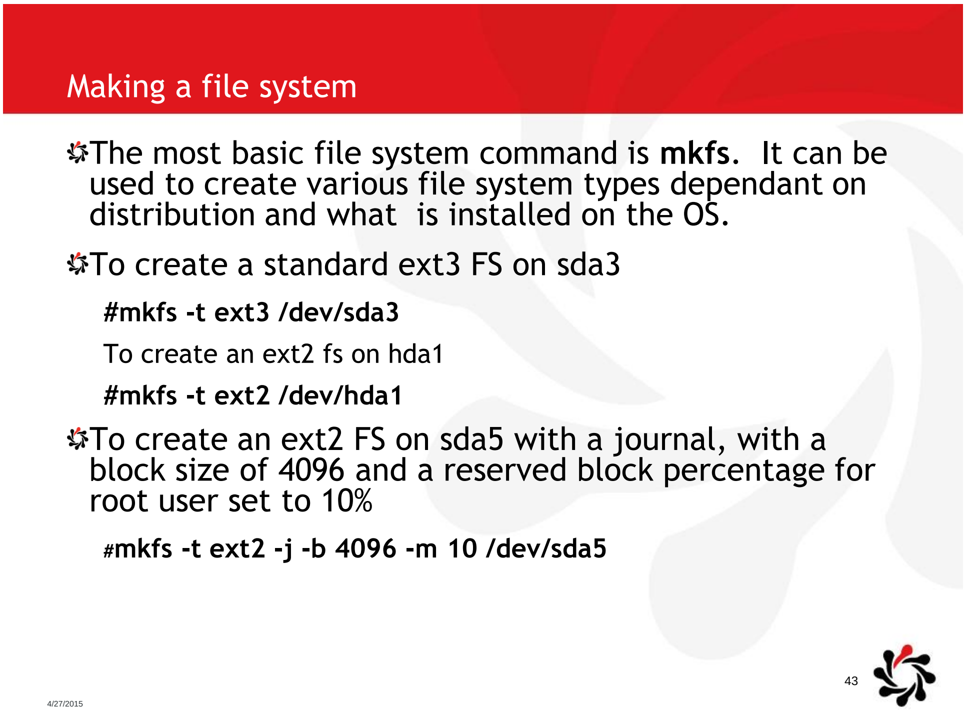


## Inodes and superblocks

- **Inodes** – index nodes
  - Pointers to file location, created with the filesystem
  - File sizes can help determine # of inodes
    - Many smaller files, more inodes are needed
    - Fewer larger files, less inodes are needed
  - Inode information is replicated in the superblock
  - No more inodes, no more file storage
- **Superblocks** (static and dynamic info)
  - Contains file system size, location, # inodes, cylinder and disk block usage
  - A backup superblock is stored every 8192 blocks
    - Provides fault tolerance for the superblock information
  - Calculating space used by superblocks
    - 36 bytes per superblock copy

4/27/2015

42



## Making a file system

- The most basic file system command is **mkfs**. It can be used to create various file system types dependant on distribution and what is installed on the OS.
- To create a standard ext3 FS on sda3

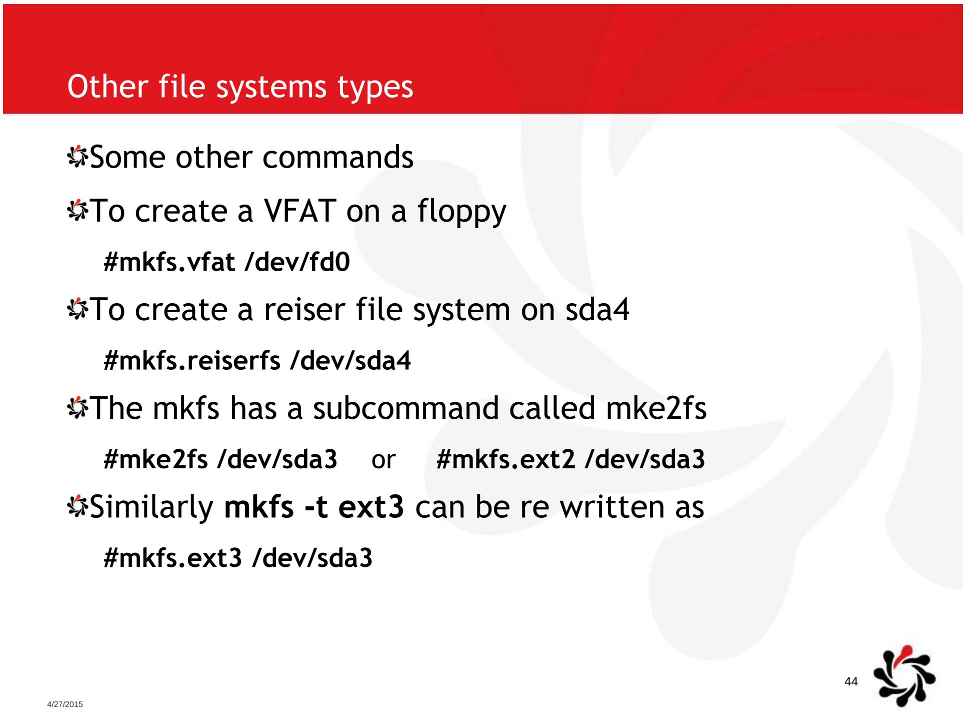
```
#mkfs -t ext3 /dev/sda3
```

To create an ext2 fs on hda1

```
#mkfs -t ext2 /dev/hda1
```

- To create an ext2 FS on sda5 with a journal, with a block size of 4096 and a reserved block percentage for root user set to 10%

```
#mkfs -t ext2 -j -b 4096 -m 10 /dev/sda5
```



## Other file systems types

- Some other commands
- To create a VFAT on a floppy

```
#mkfs.vfat /dev/fd0
```

- To create a reiser file system on sda4

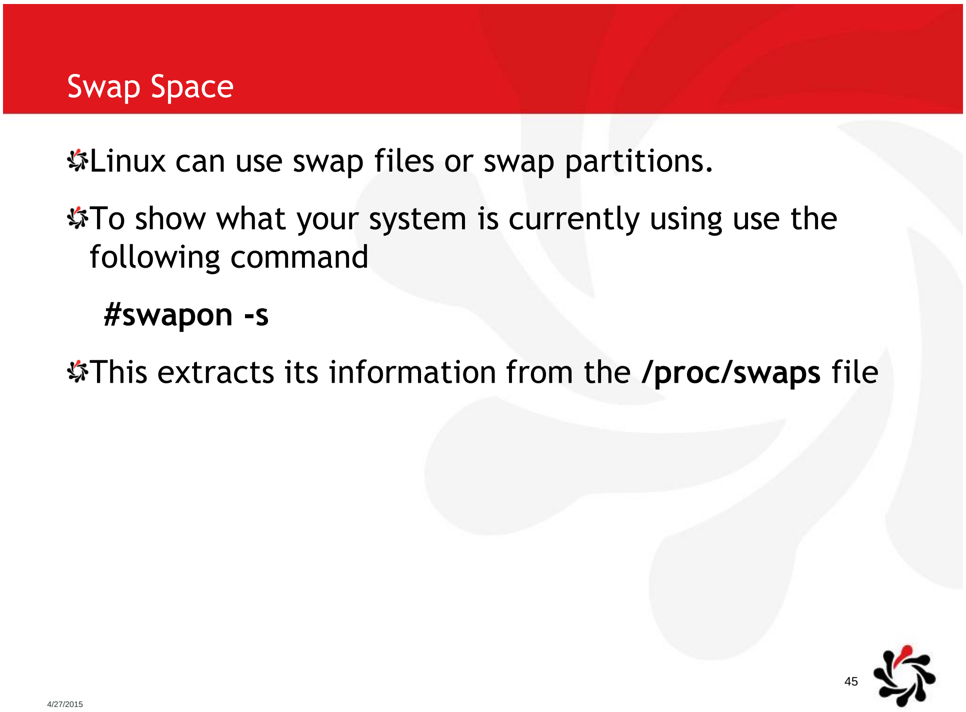
```
#mkfs.reiserfs /dev/sda4
```

- The mkfs has a subcommand called mke2fs

```
#mke2fs /dev/sda3    or    #mkfs.ext2 /dev/sda3
```

- Similarly **mkfs -t ext3** can be re written as

```
#mkfs.ext3 /dev/sda3
```



## Swap Space

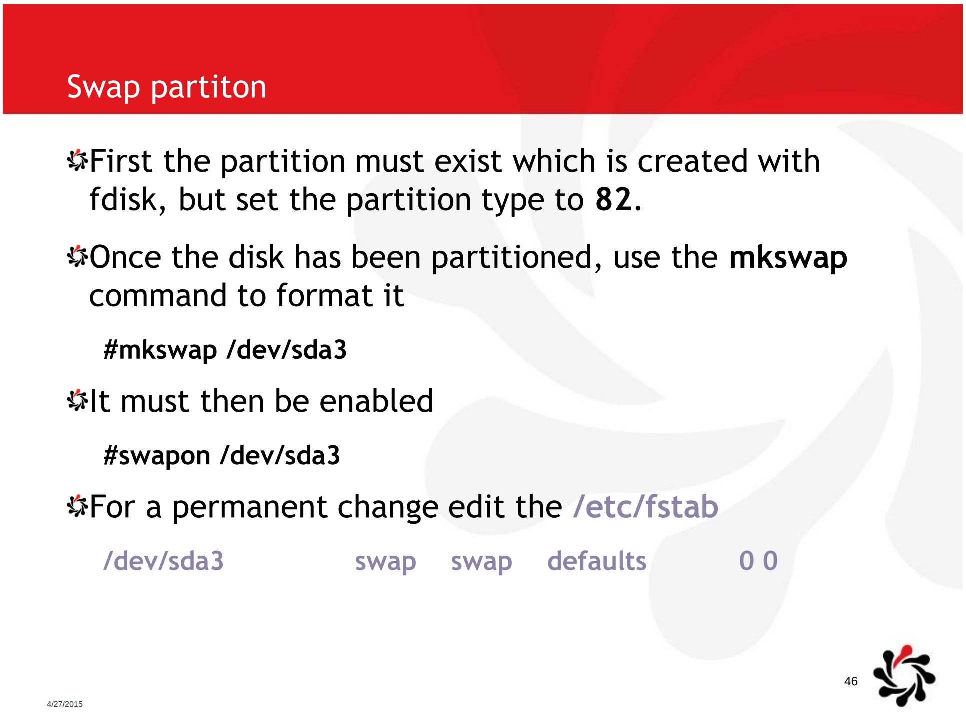
- Linux can use swap files or swap partitions.
- To show what your system is currently using use the following command

```
#swapon -s
```

- This extracts its information from the **/proc/swaps** file

45

4/27/2015



## Swap partiton

- First the partition must exist which is created with fdisk, but set the partition type to **82**.
- Once the disk has been partitioned, use the **mkswap** command to format it

```
#mkswap /dev/sda3
```

- It must then be enabled

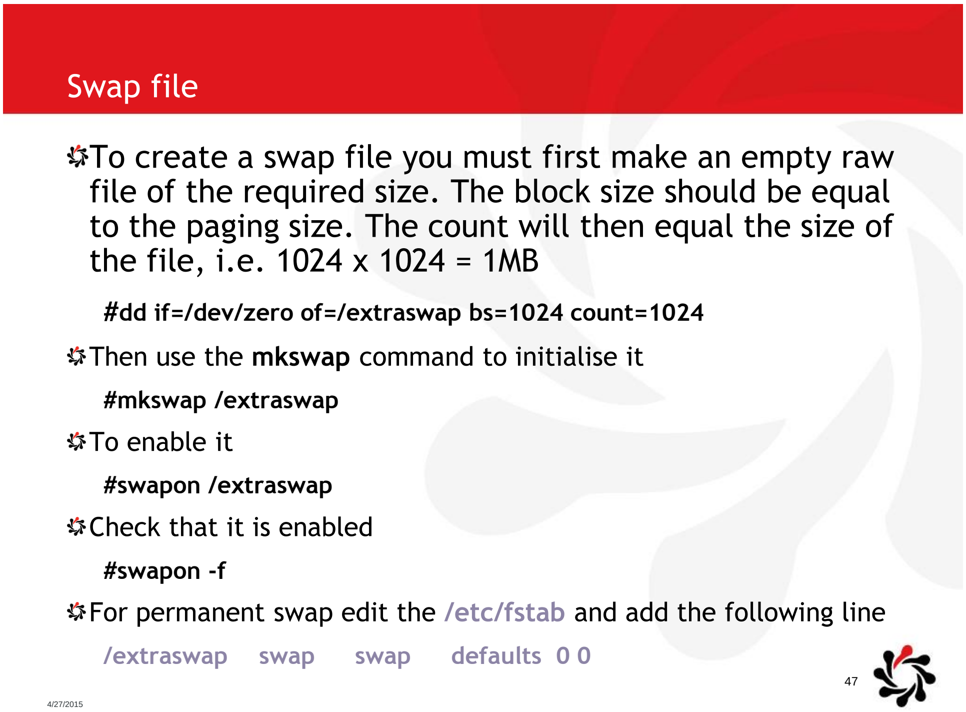
```
#swapon /dev/sda3
```

- For a permanent change edit the /etc/fstab

```
/dev/sda3            swap    swap    defaults          0 0
```

46

4/27/2015



## Swap file

- To create a swap file you must first make an empty raw file of the required size. The block size should be equal to the paging size. The count will then equal the size of the file, i.e. 1024 x 1024 = 1MB

```
#dd if=/dev/zero of=/extraswap bs=1024 count=1024
```

- Then use the **mkswap** command to initialise it

```
#mkswap /extraswap
```

- To enable it

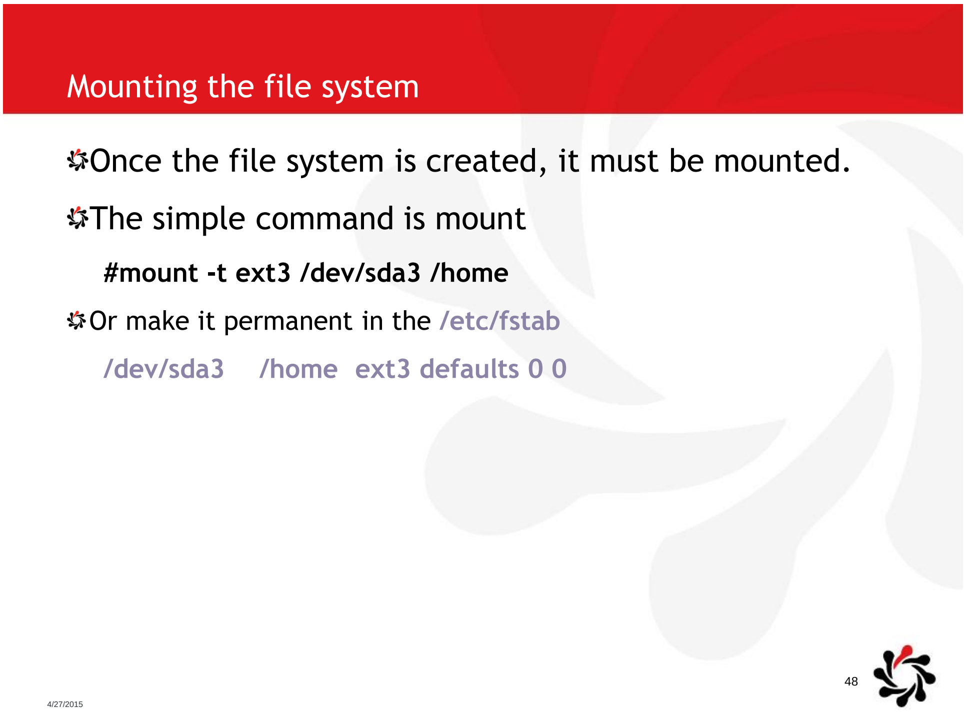
```
#swapon /extraswap
```

- Check that it is enabled

```
#swapon -f
```

- For permanent swap edit the /etc/fstab and add the following line

```
/extraswap    swap    swap    defaults  0 0
```



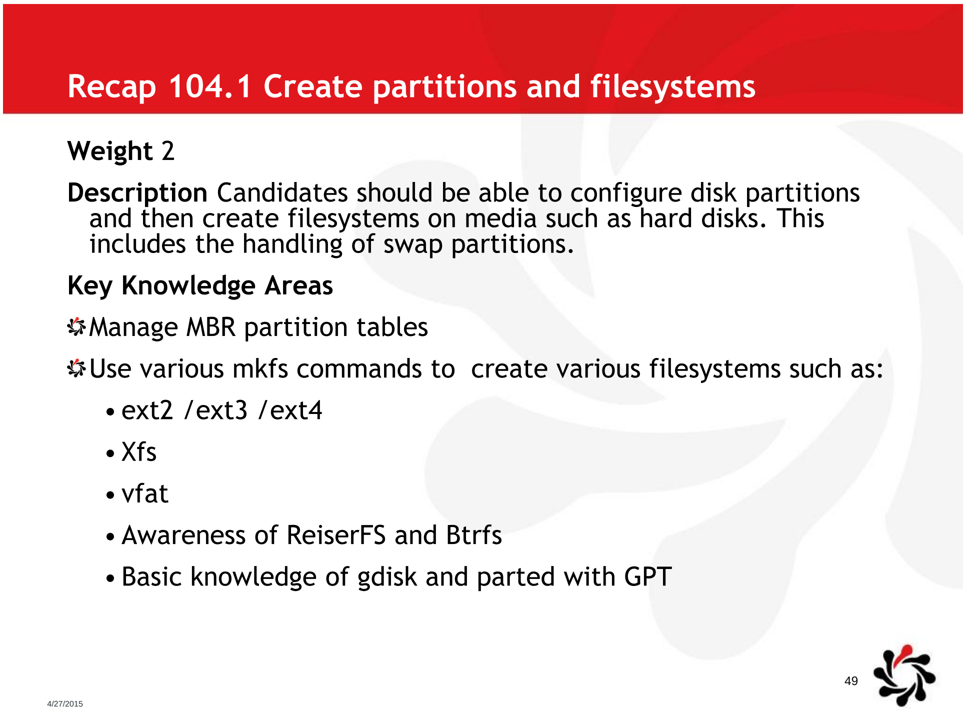
## Mounting the file system

- Once the file system is created, it must be mounted.
- The simple command is mount

```
#mount -t ext3 /dev/sda3 /home
```

- Or make it permanent in the /etc/fstab

```
/dev/sda3    /home  ext3 defaults 0 0
```



## Recap 104.1 Create partitions and filesystems

**Weight** 2

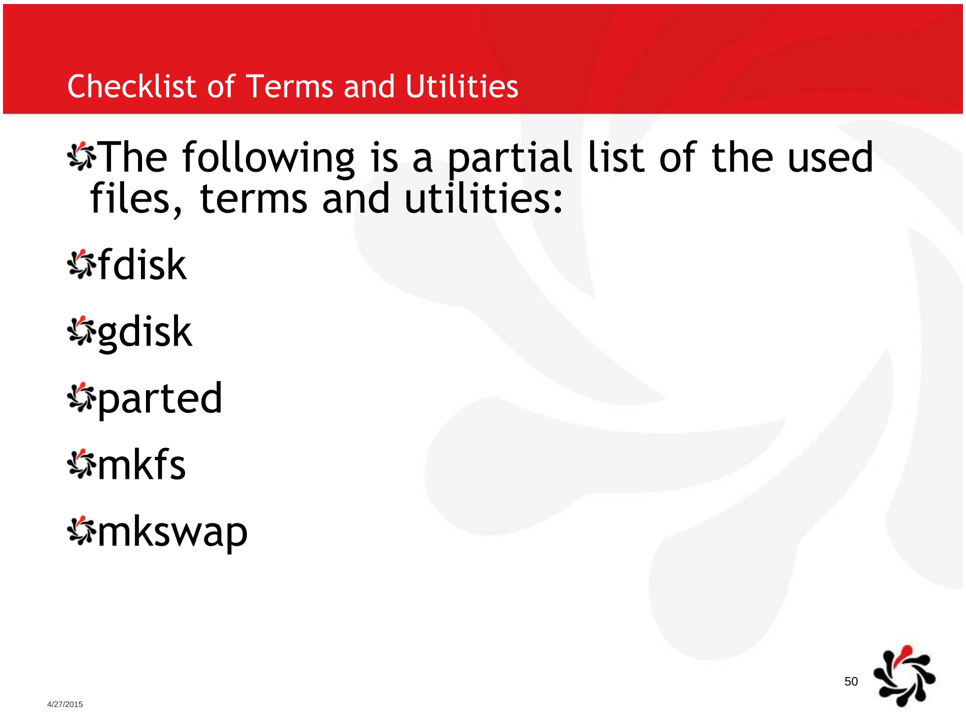
**Description** Candidates should be able to configure disk partitions and then create filesystems on media such as hard disks. This includes the handling of swap partitions.

**Key Knowledge Areas**

- Manage MBR partition tables
- Use various mkfs commands to create various filesystems such as:
  - ext2 /ext3 /ext4
  - Xfs
  - vfat
  - Awareness of ReiserFS and Btrfs
  - Basic knowledge of gdisk and parted with GPT

4/27/2015

49



## Checklist of Terms and Utilities

- The following is a partial list of the used files, terms and utilities:
- fdisk
- gdisk
- parted
- mkfs
- mkswap

4/27/2015

50

## 104.2 Maintain the integrity of filesystems

**Weight** 2

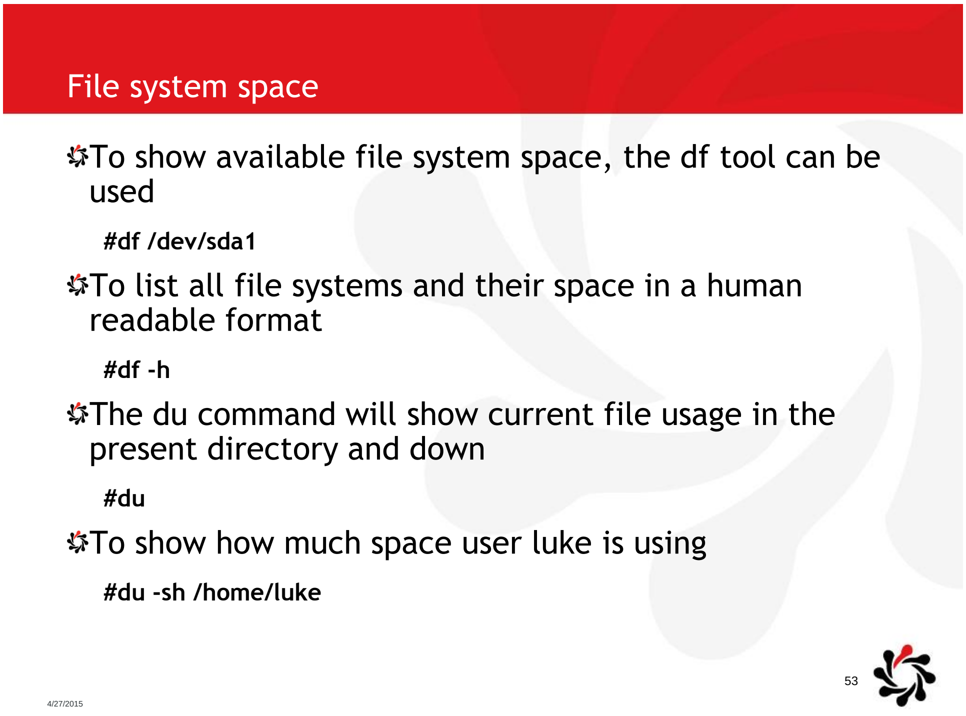
**Description** Candidates should be able to maintain a standard filesystem, as well as the extra data associated with a journaling filesystem.

**Key Knowledge Areas**

- Verify the integrity of filesystems.
- Monitor free space and inodes.
- Repair simple filesystem problems.

## File system integrity

- The system administrator must make sure that the file system remains functional and has enough space on it on a daily basis.
- There are several tools for monitoring space and some tools for checking and fixing the file system meta data.



## File system space

- To show available file system space, the df tool can be used

```
#df /dev/sda1
```

- To list all file systems and their space in a human readable format

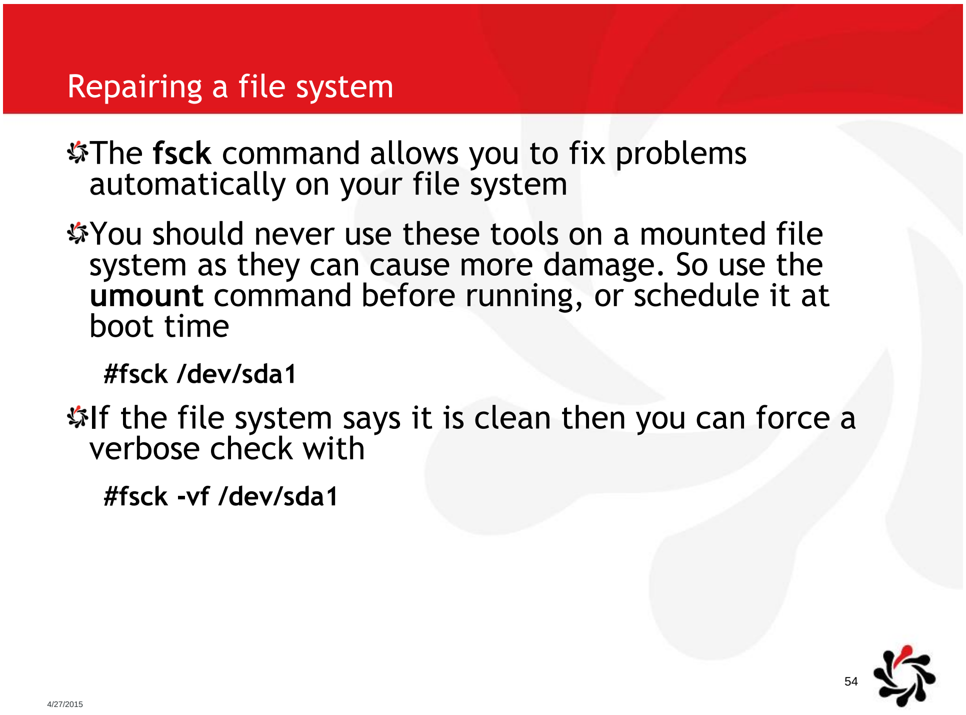
```
#df -h
```

- The du command will show current file usage in the present directory and down

```
#du
```

- To show how much space user luke is using

```
#du -sh /home/luke
```



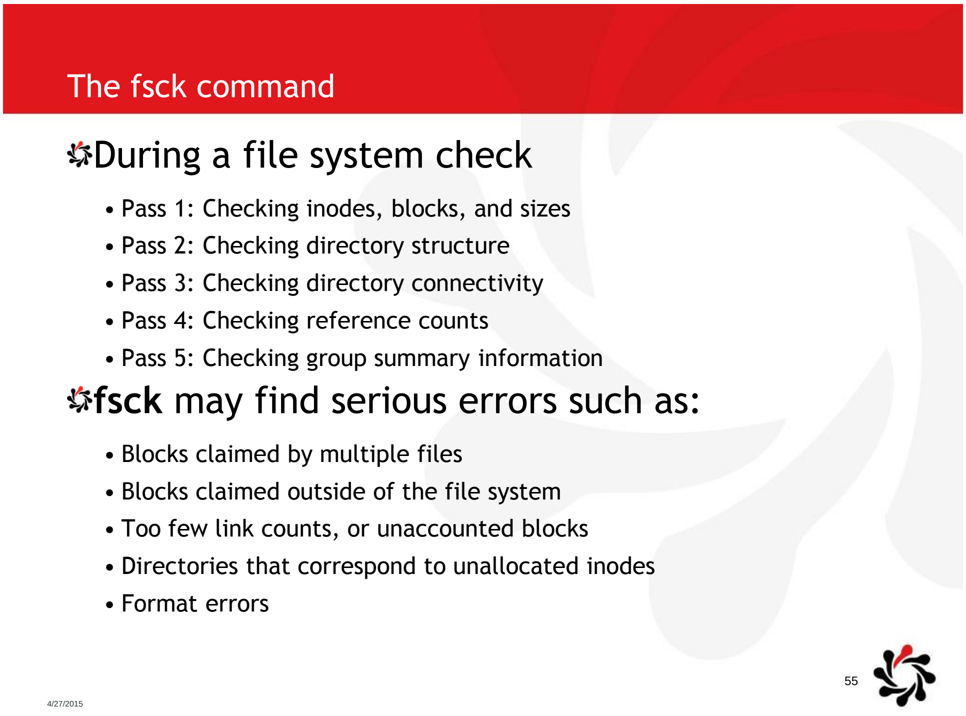
## Repairing a file system

- The **fsck** command allows you to fix problems automatically on your file system
- You should never use these tools on a mounted file system as they can cause more damage. So use the **umount** command before running, or schedule it at boot time

```
#fsck /dev/sda1
```

- If the file system says it is clean then you can force a verbose check with

```
#fsck -vf /dev/sda1
```

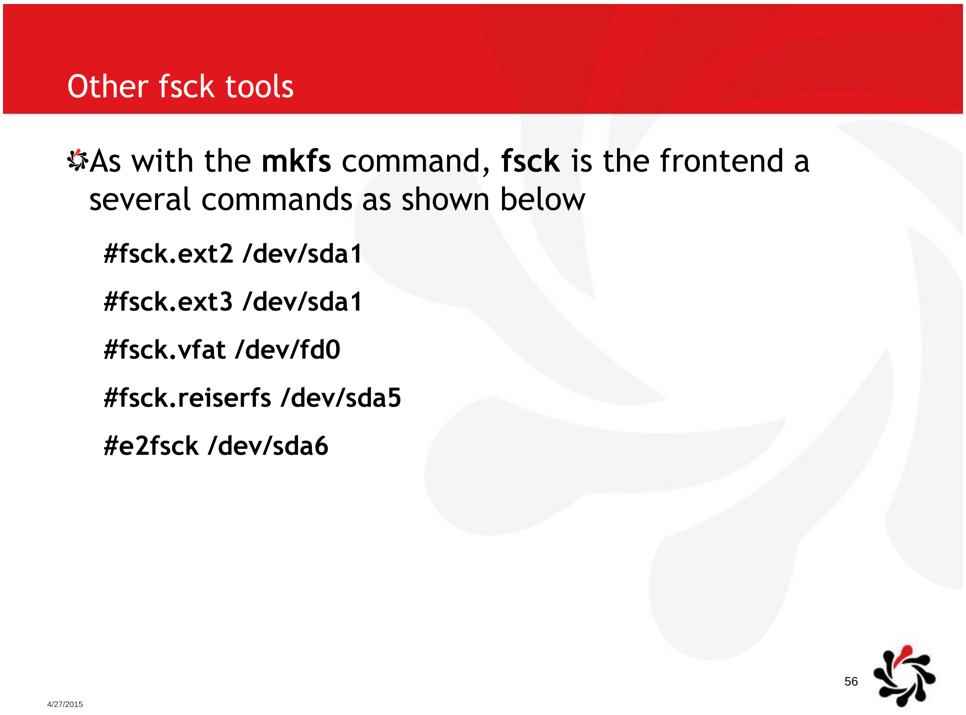


## The fsck command

- During a file system check
  - Pass 1: Checking inodes, blocks, and sizes
  - Pass 2: Checking directory structure
  - Pass 3: Checking directory connectivity
  - Pass 4: Checking reference counts
  - Pass 5: Checking group summary information
- **fsck** may find serious errors such as:
  - Blocks claimed by multiple files
  - Blocks claimed outside of the file system
  - Too few link counts, or unaccounted blocks
  - Directories that correspond to unallocated inodes
  - Format errors

55

4/27/2015



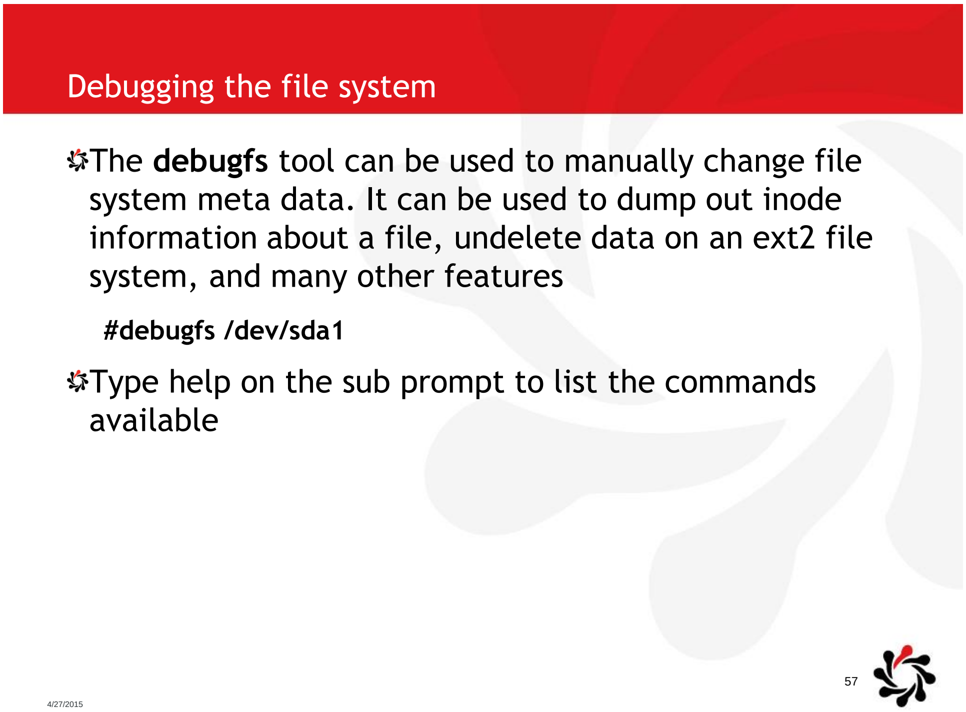
## Other fsck tools

- As with the **mkfs** command, **fsck** is the frontend a several commands as shown below

```
#fsck.ext2 /dev/sda1
#fsck.ext3 /dev/sda1
#fsck.vfat /dev/fd0
#fsck.reiserfs /dev/sda5
#e2fsck /dev/sda6
```

56

4/27/2015

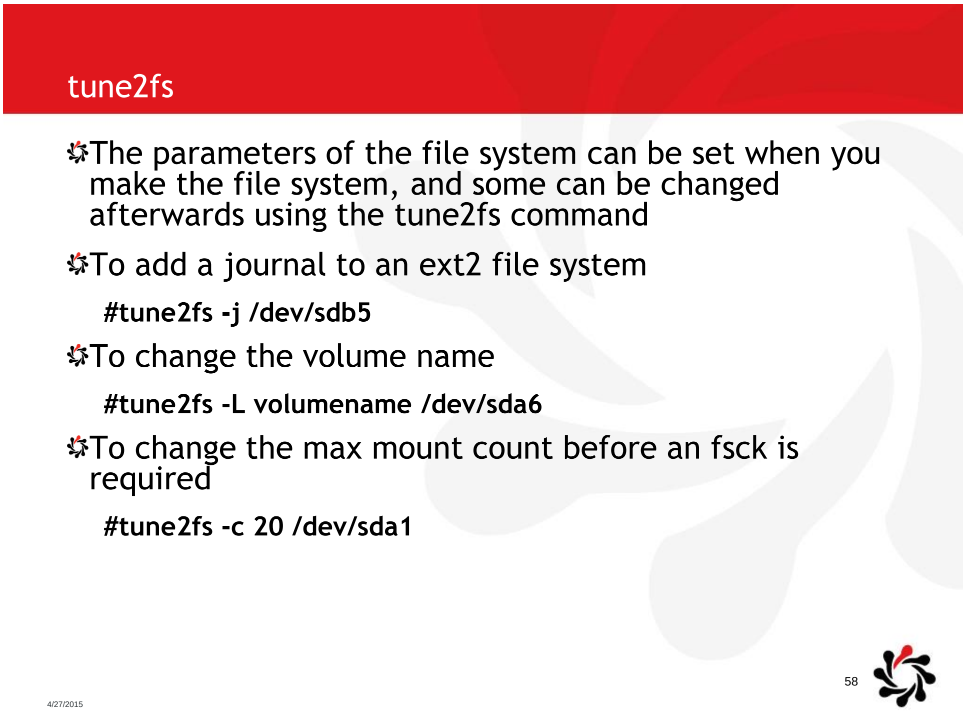


## Debugging the file system

- The **debugfs** tool can be used to manually change file system meta data. It can be used to dump out inode information about a file, undelete data on an ext2 file system, and many other features

```
#debugfs /dev/sda1
```

- Type help on the sub prompt to list the commands available



## tune2fs

- The parameters of the file system can be set when you make the file system, and some can be changed afterwards using the tune2fs command
- To add a journal to an ext2 file system

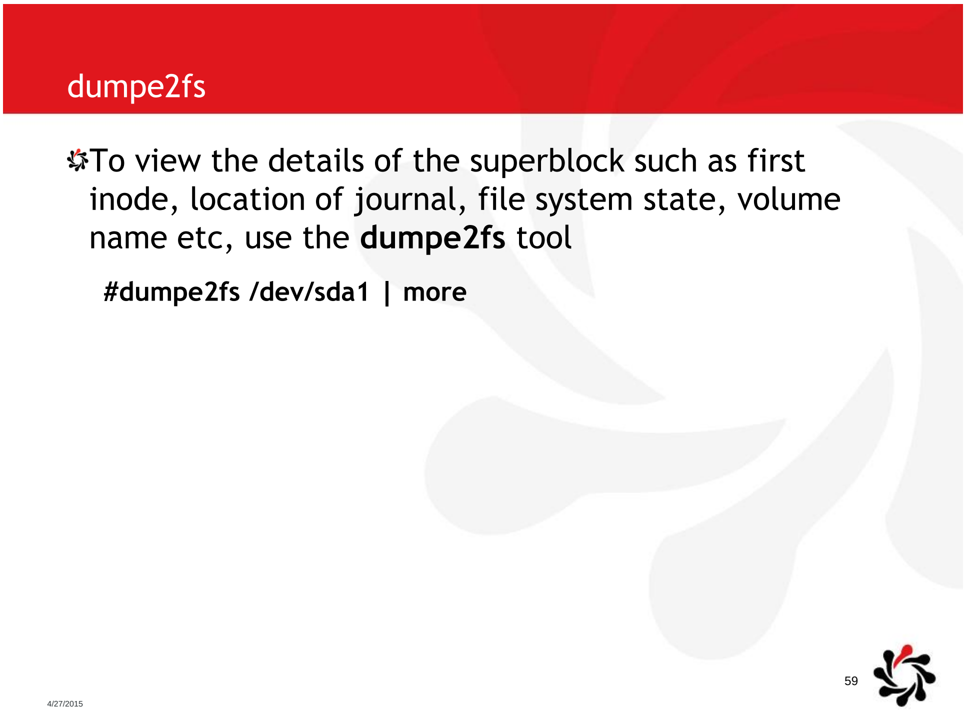
```
#tune2fs -j /dev/sdb5
```

- To change the volume name

```
#tune2fs -L volumename /dev/sda6
```

- To change the max mount count before an fsck is required

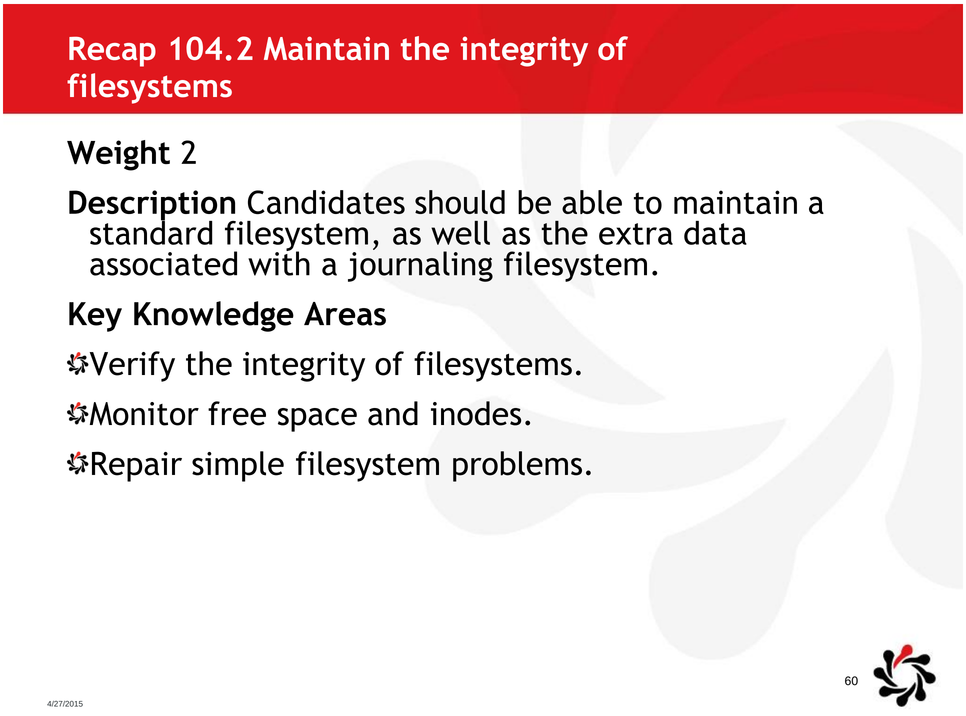
```
#tune2fs -c 20 /dev/sda1
```



## dumpe2fs

- To view the details of the superblock such as first inode, location of journal, file system state, volume name etc, use the **dumpe2fs** tool

```
#dumpe2fs /dev/sda1 | more
```



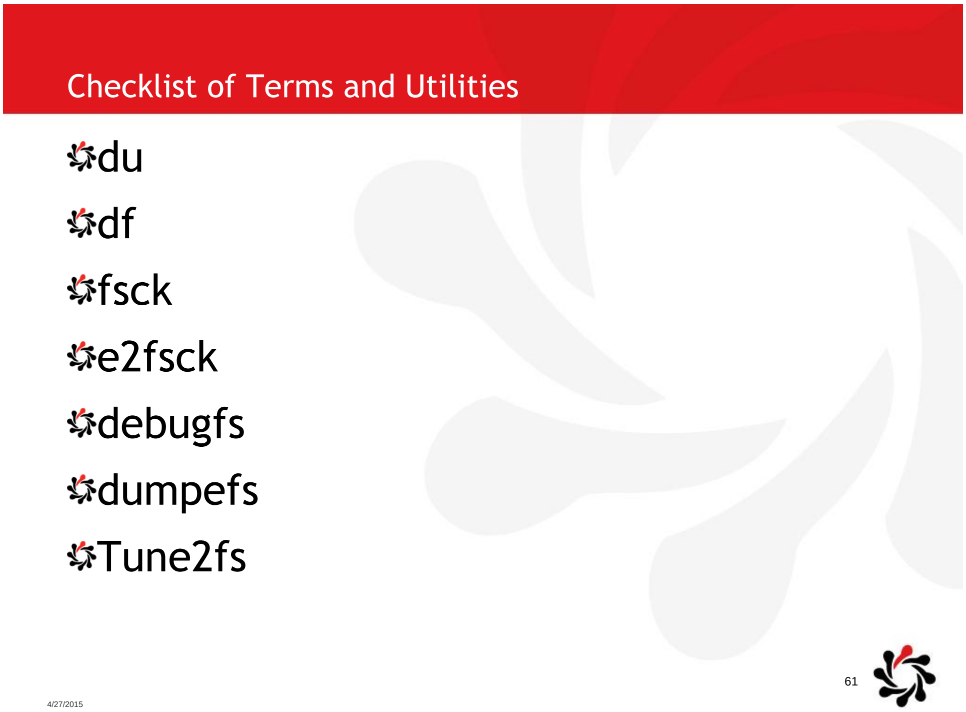
## Recap 104.2 Maintain the integrity of filesystems

**Weight** 2

**Description** Candidates should be able to maintain a standard filesystem, as well as the extra data associated with a journaling filesystem.

**Key Knowledge Areas**

- Verify the integrity of filesystems.
- Monitor free space and inodes.
- Repair simple filesystem problems.

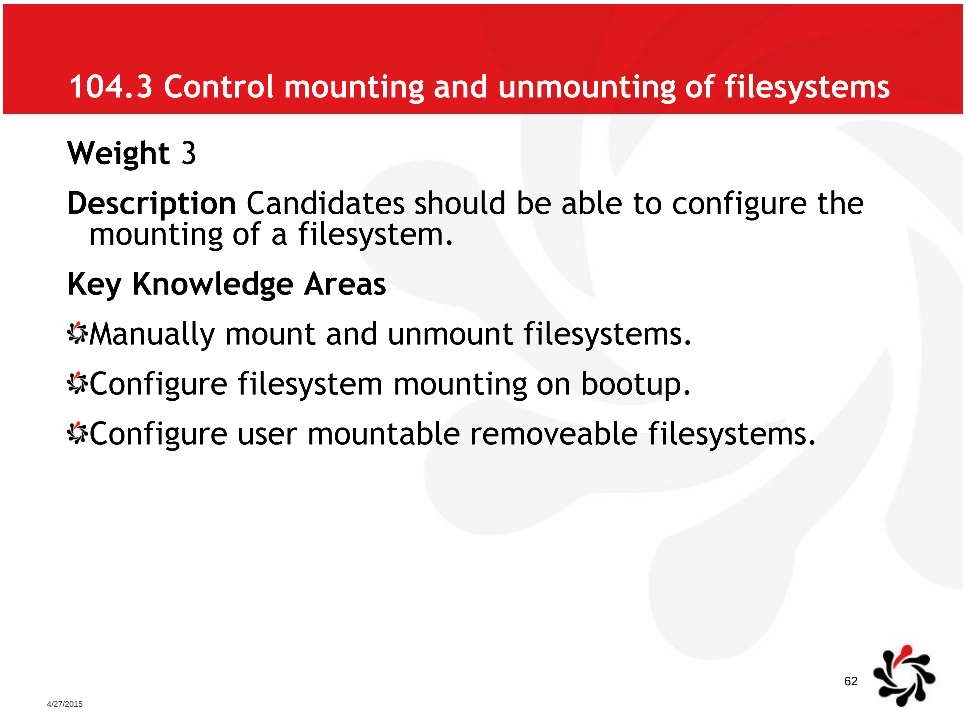


## Checklist of Terms and Utilities

- du
- df
- fsck
- e2fsck
- debugfs
- dumpefs
- Tune2fs

4/27/2015

61



## 104.3 Control mounting and unmounting of filesystems

**Weight** 3

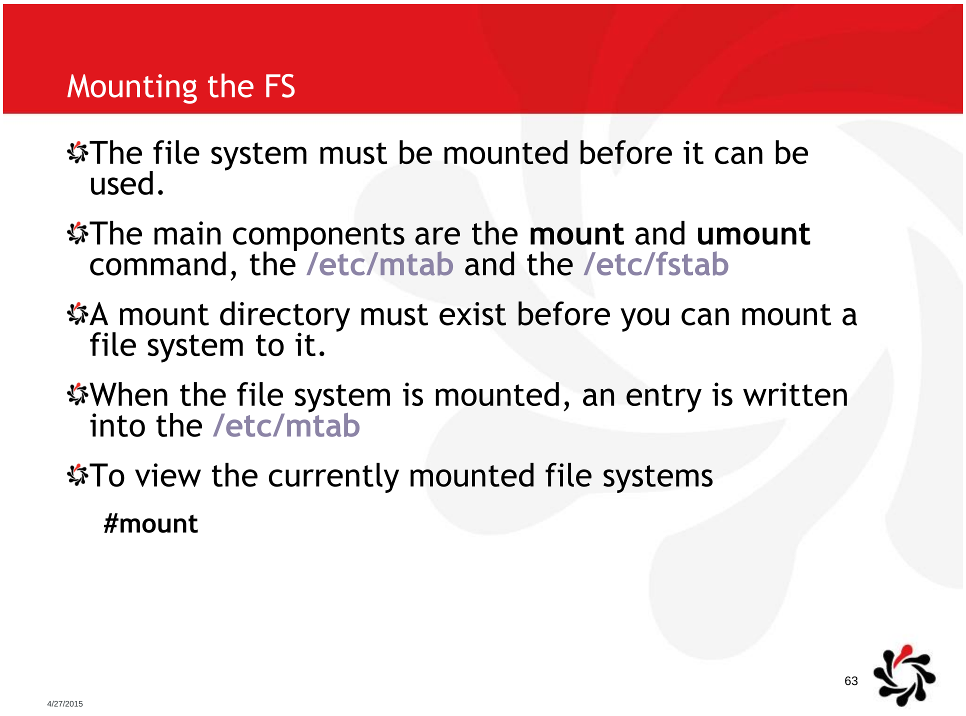
**Description** Candidates should be able to configure the mounting of a filesystem.

**Key Knowledge Areas**

- Manually mount and unmount filesystems.
- Configure filesystem mounting on bootup.
- Configure user mountable removeable filesystems.

4/27/2015

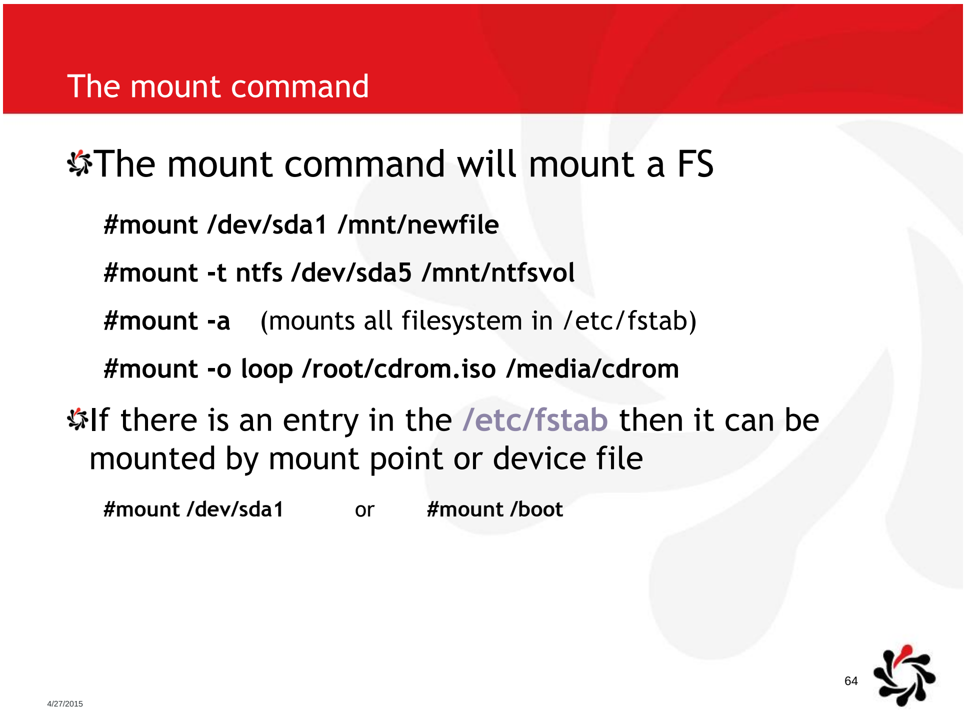
62



## Mounting the FS

- The file system must be mounted before it can be used.
- The main components are the **mount** and **umount** command, the /etc/mtab and the /etc/fstab
- A mount directory must exist before you can mount a file system to it.
- When the file system is mounted, an entry is written into the /etc/mtab
- To view the currently mounted file systems

```
#mount
```



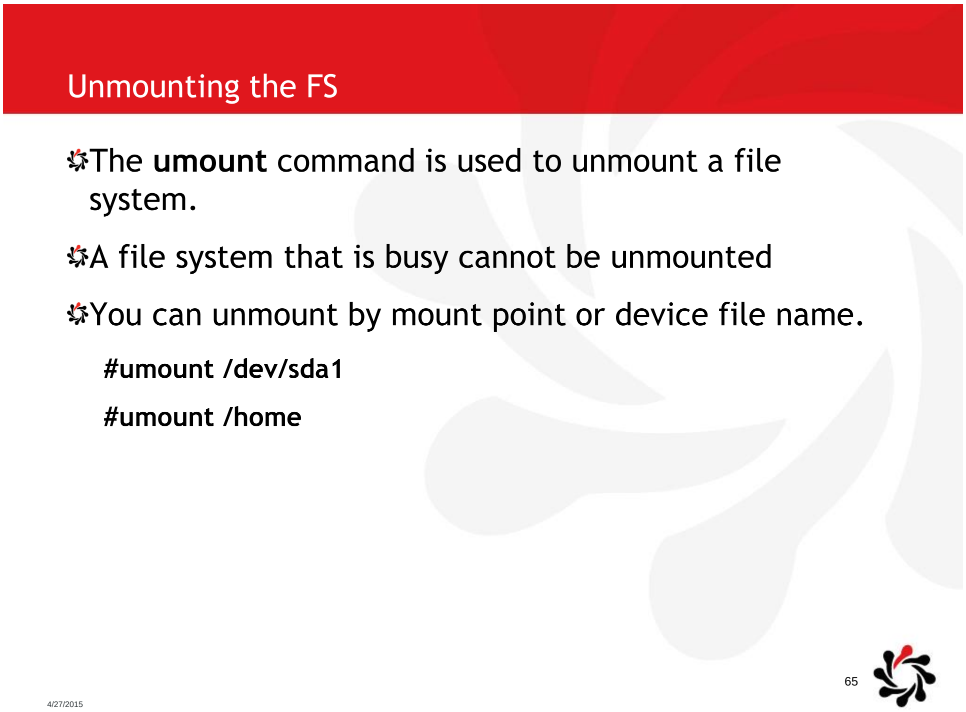
## The mount command

- The mount command will mount a FS

```
#mount /dev/sda1 /mnt/newfile
#mount -t ntfs /dev/sda5 /mnt/ntfsvol
#mount -a    (mounts all filesystem in /etc/fstab)
#mount -o loop /root/cdrom.iso /media/cdrom
```

- If there is an entry in the /etc/fstab then it can be mounted by mount point or device file

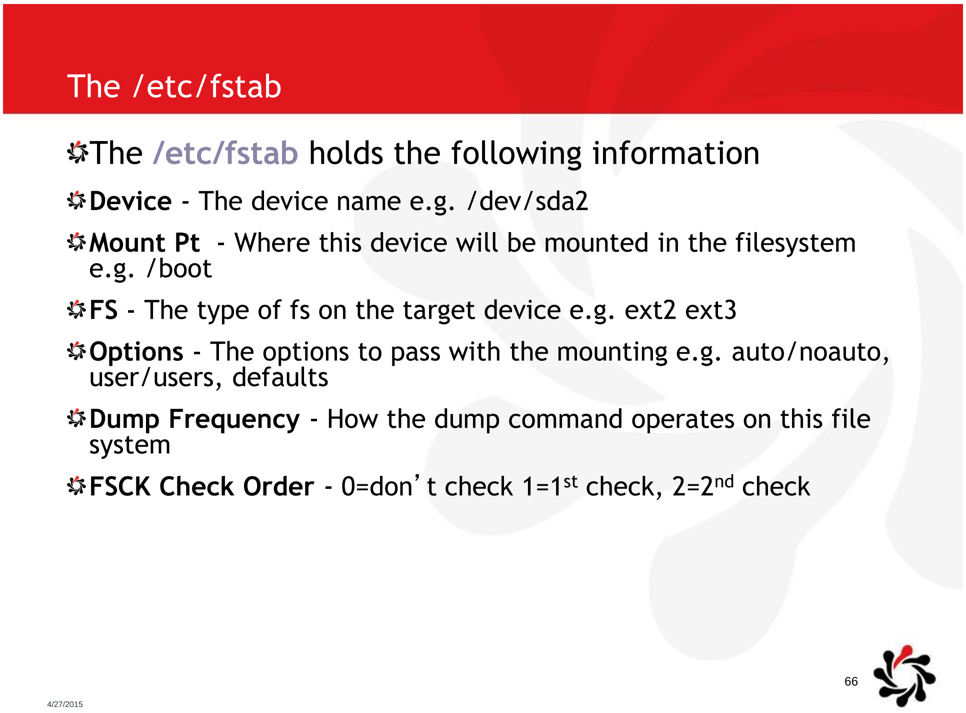
```
#mount /dev/sda1        or       #mount /boot
```



## Unmounting the FS

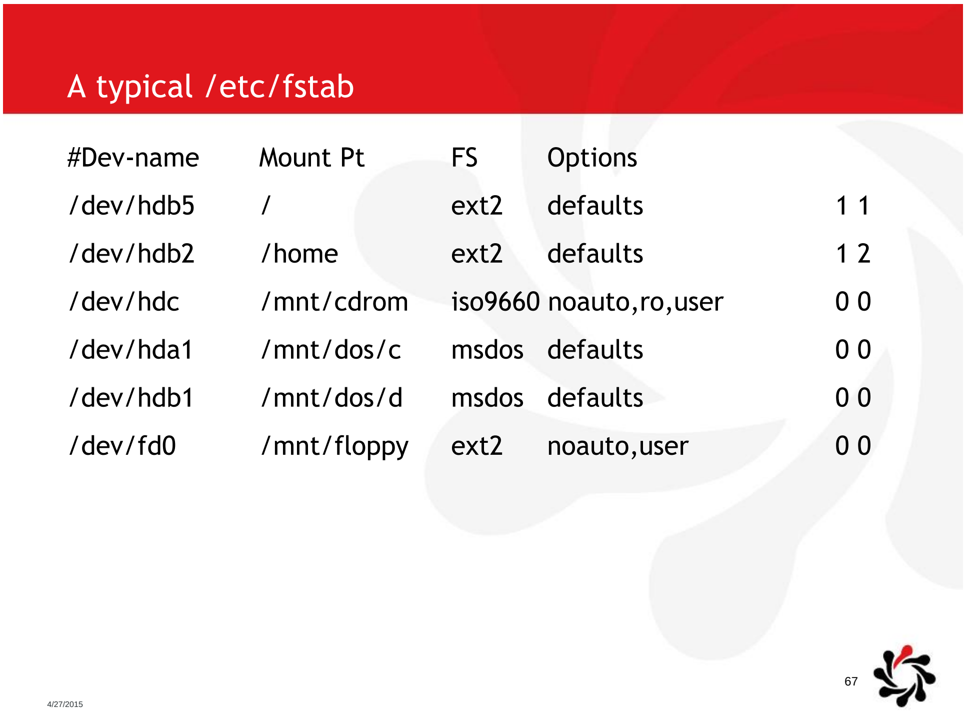
- The **umount** command is used to unmount a file system.
- A file system that is busy cannot be unmounted
- You can unmount by mount point or device file name.

```
#umount /dev/sda1
#umount /home
```



## The /etc/fstab

- The /etc/fstab holds the following information
- **Device** - The device name e.g. /dev/sda2
- **Mount Pt** - Where this device will be mounted in the filesystem e.g. /boot
- **FS** - The type of fs on the target device e.g. ext2 ext3
- **Options** - The options to pass with the mounting e.g. auto/noauto, user/users, defaults
- **Dump Frequency** - How the dump command operates on this file system
- **FSCK Check Order** - 0=don't check 1=1st check, 2=2nd check

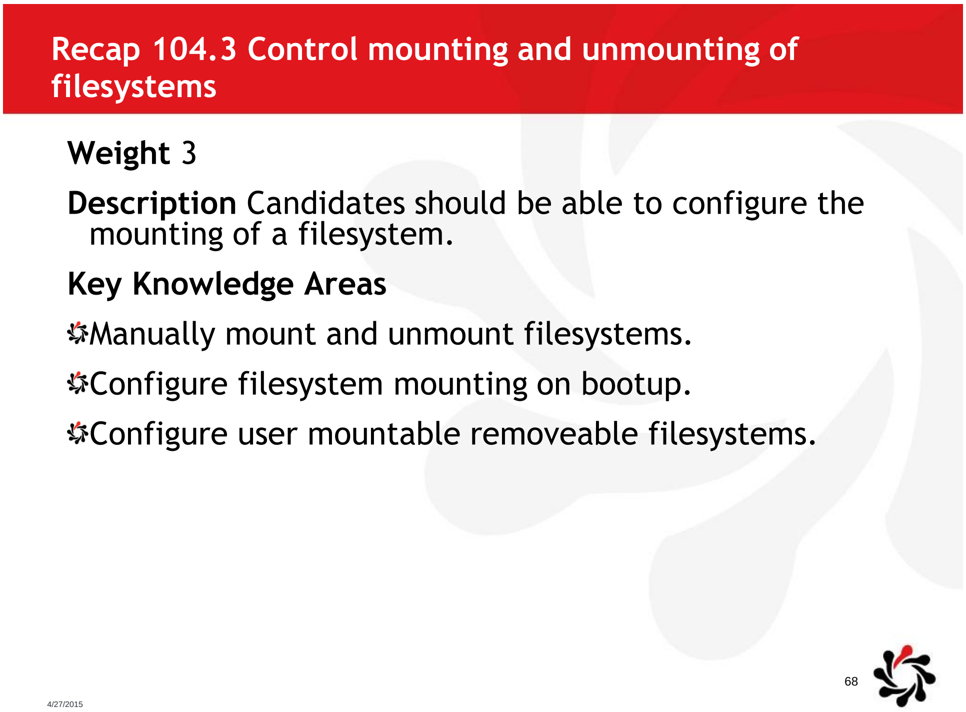


## A typical /etc/fstab

```
#Dev-name     Mount Pt      FS      Options
/dev/hdb5     /             ext2    defaults              1 1
/dev/hdb2     /home         ext2    defaults              1 2
/dev/hdc      /mnt/cdrom    iso9660 noauto,ro,user        0 0
/dev/hda1     /mnt/dos/c    msdos   defaults              0 0
/dev/hdb1     /mnt/dos/d    msdos   defaults              0 0
/dev/fd0      /mnt/floppy   ext2    noauto,user           0 0
```

4/27/2015

67



## Recap 104.3 Control mounting and unmounting of filesystems

**Weight** 3

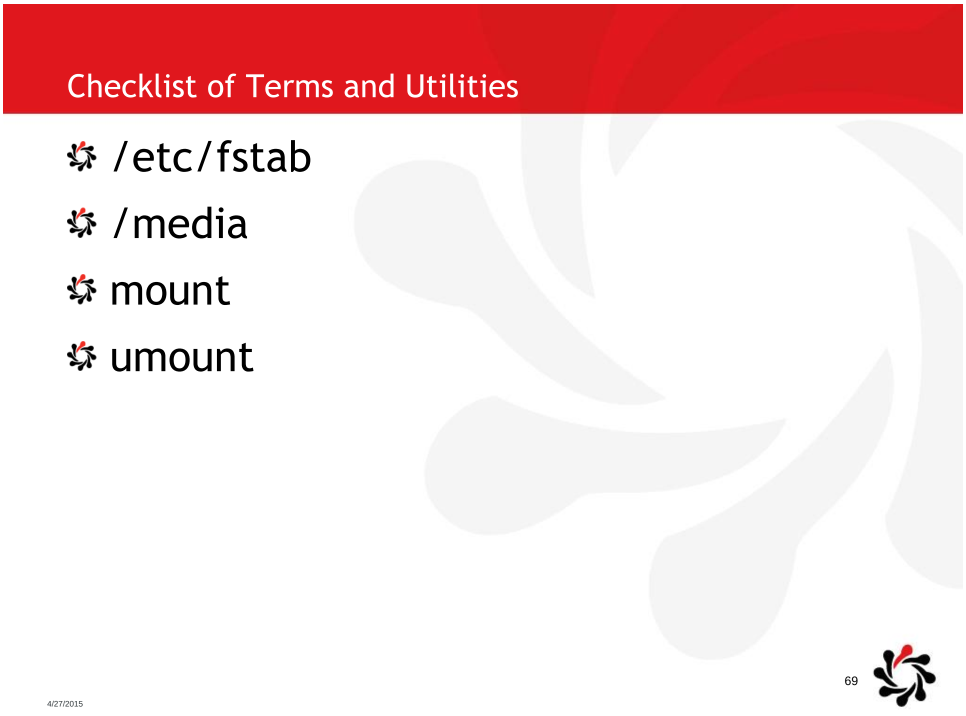
**Description** Candidates should be able to configure the mounting of a filesystem.

**Key Knowledge Areas**

- Manually mount and unmount filesystems.
- Configure filesystem mounting on bootup.
- Configure user mountable removeable filesystems.

4/27/2015

68

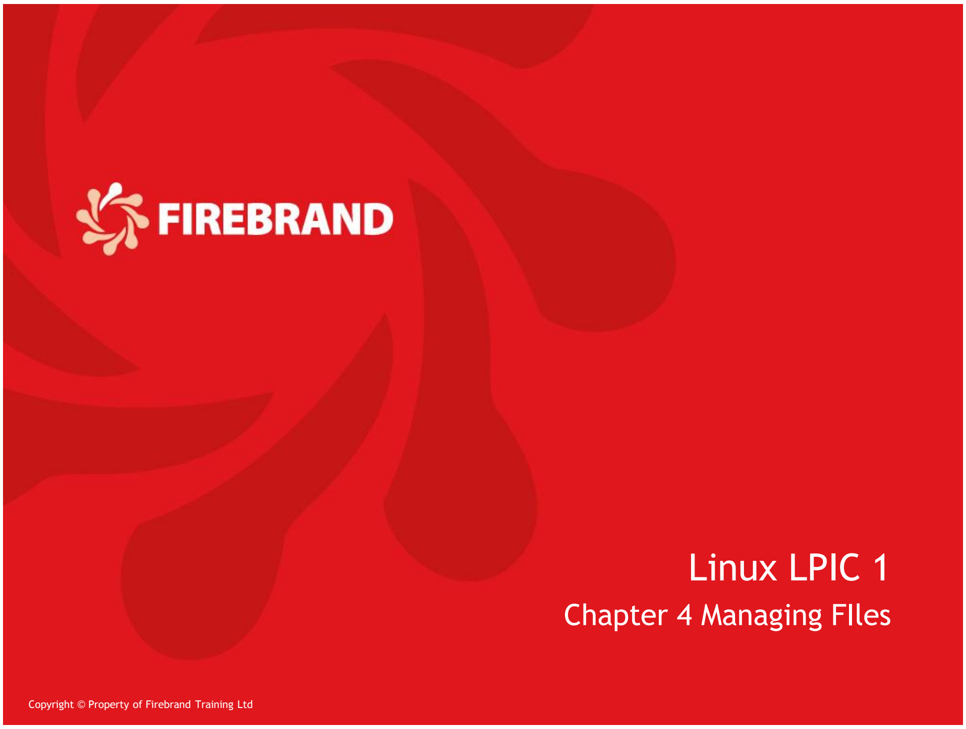


## Checklist of Terms and Utilities

- /etc/fstab
- /media
- mount
- umount

4/27/2015

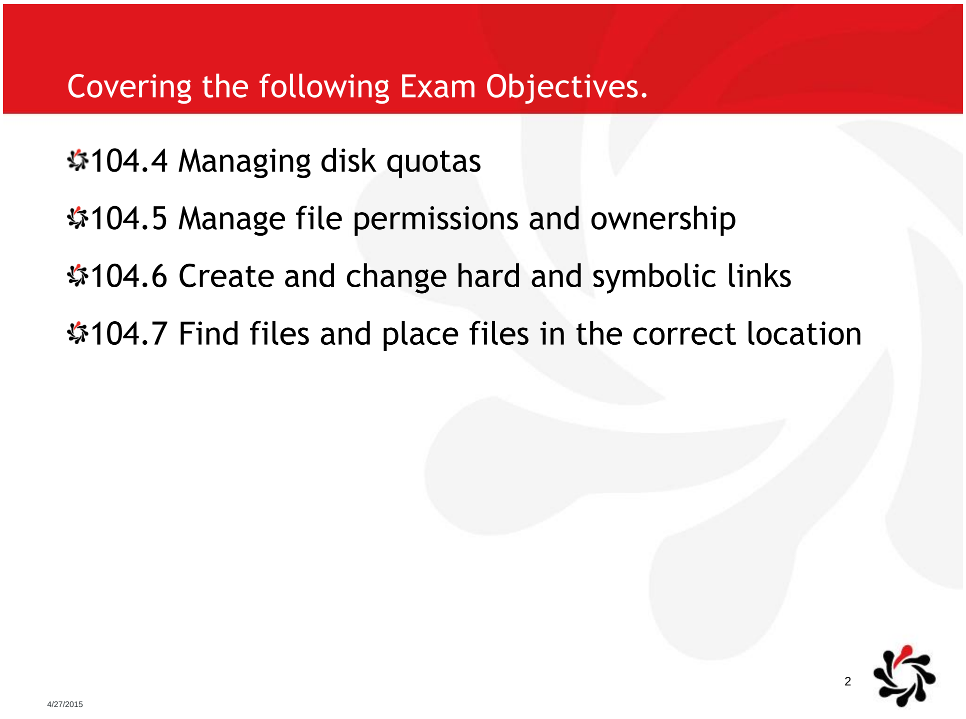
69



FIREBRAND

# Linux LPIC 1

## Chapter 4 Managing FIles

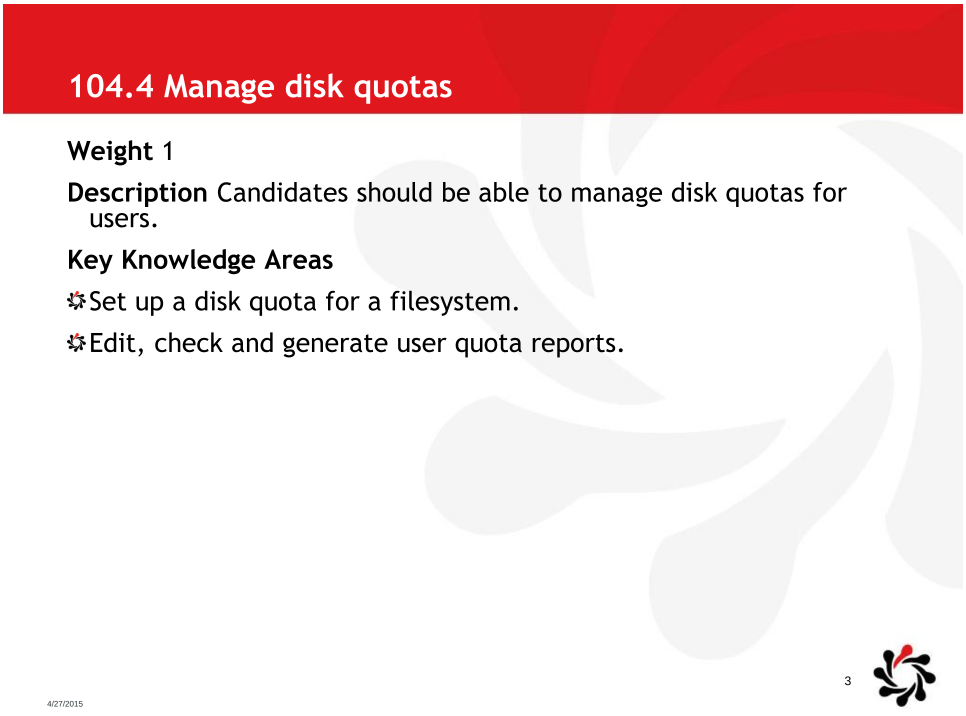


## Covering the following Exam Objectives.

- 104.4 Managing disk quotas
- 104.5 Manage file permissions and ownership
- 104.6 Create and change hard and symbolic links
- 104.7 Find files and place files in the correct location

4/27/2015

2



## 104.4 Manage disk quotas

**Weight** 1

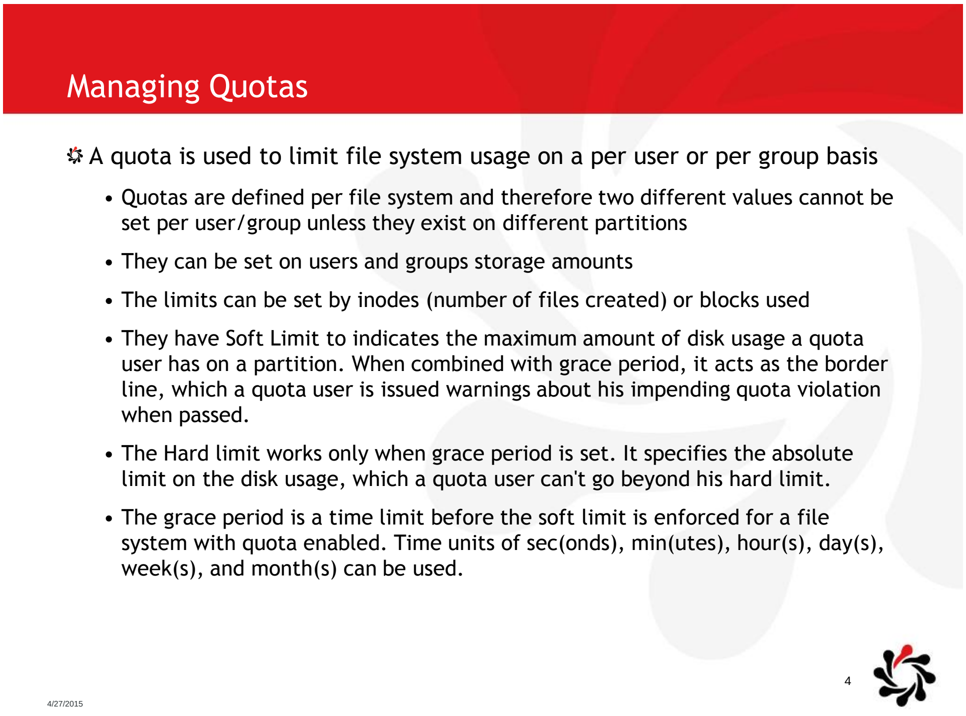
**Description** Candidates should be able to manage disk quotas for users.

**Key Knowledge Areas**

- Set up a disk quota for a filesystem.
- Edit, check and generate user quota reports.

4/27/2015

3



## Managing Quotas

- A quota is used to limit file system usage on a per user or per group basis
  - Quotas are defined per file system and therefore two different values cannot be set per user/group unless they exist on different partitions
  - They can be set on users and groups storage amounts
  - The limits can be set by inodes (number of files created) or blocks used
  - They have Soft Limit to indicates the maximum amount of disk usage a quota user has on a partition. When combined with grace period, it acts as the border line, which a quota user is issued warnings about his impending quota violation when passed.
  - The Hard limit works only when grace period is set. It specifies the absolute limit on the disk usage, which a quota user can't go beyond his hard limit.
  - The grace period is a time limit before the soft limit is enforced for a file system with quota enabled. Time units of sec(onds), min(utes), hour(s), day(s), week(s), and month(s) can be used.

4/27/2015

4

Create the **aquota.user** and **aquota.group** files in the fs root that you wish to apply quotas to.

**#touch aquota.user aquota.group**

Set the permissions relative to who you want to control quotas

**#chmod 660 aquota.***

Edit the /etc/fstab file and add the “**usrquota** and **grpquota**” entry to the options column

Remount the entry

**#mount –o remount,rw /entry**

Initialize the quotas with

**#quotacheck -avug**

- This will check the configured filesystem
- Populate the **aquota.user** and **aquota.group** files

Turn quotas on with

**#quotaon -avug**

To set the soft/hard limits, use the following

**#edquota -u user**

To set the grace period before invoking quotas, use

**#edquota -t**

Add quotacheck and quotaon to boot files if required

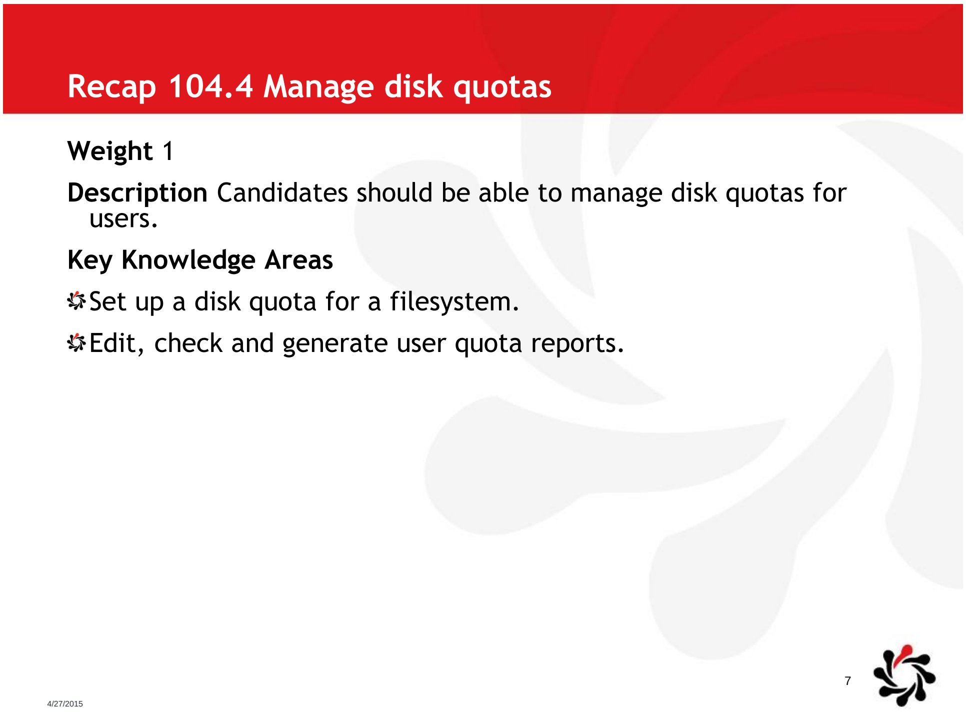
Getting information with quota

**#quota user1** (shows quota statistics for user1)

If no quota defined for user1, shows “none”

Reporting quota info with repquota

**#repquota –avug** (prints user info and statistics)



## Recap 104.4 Manage disk quotas

**Weight** 1

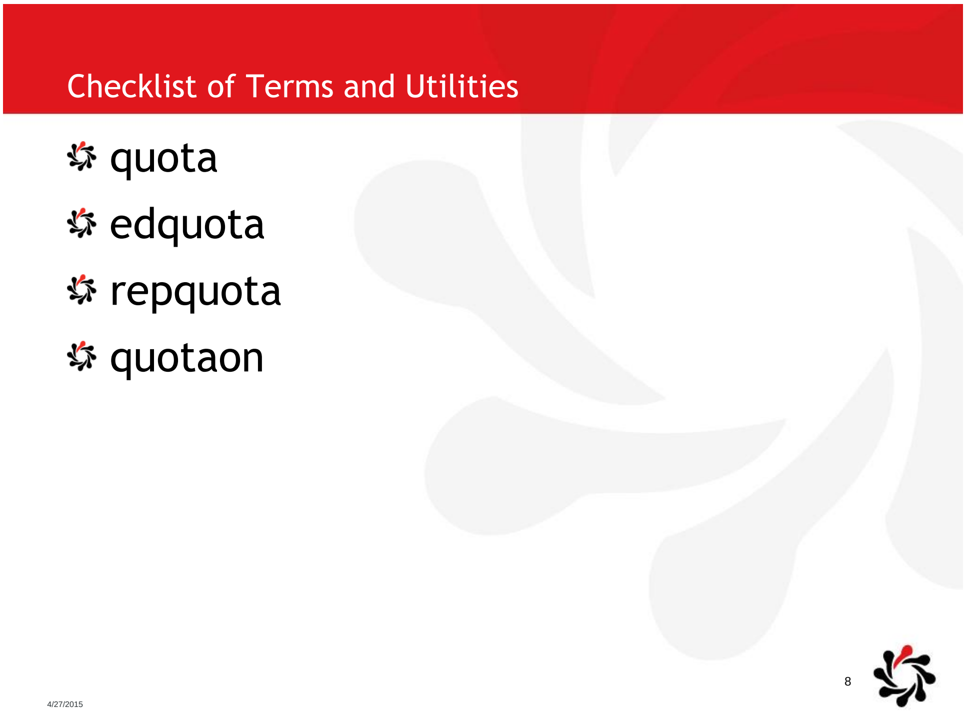
**Description** Candidates should be able to manage disk quotas for users.

**Key Knowledge Areas**

- Set up a disk quota for a filesystem.
- Edit, check and generate user quota reports.

7

4/27/2015



## Checklist of Terms and Utilities

- quota
- edquota
- repquota
- quotaon

8

4/27/2015

## 104.5 Manage file permissions and ownership

**Weight** 3

**Description** Candidates should be able to control file access through the proper use of permissions and ownerships.

**Key Knowledge Areas**

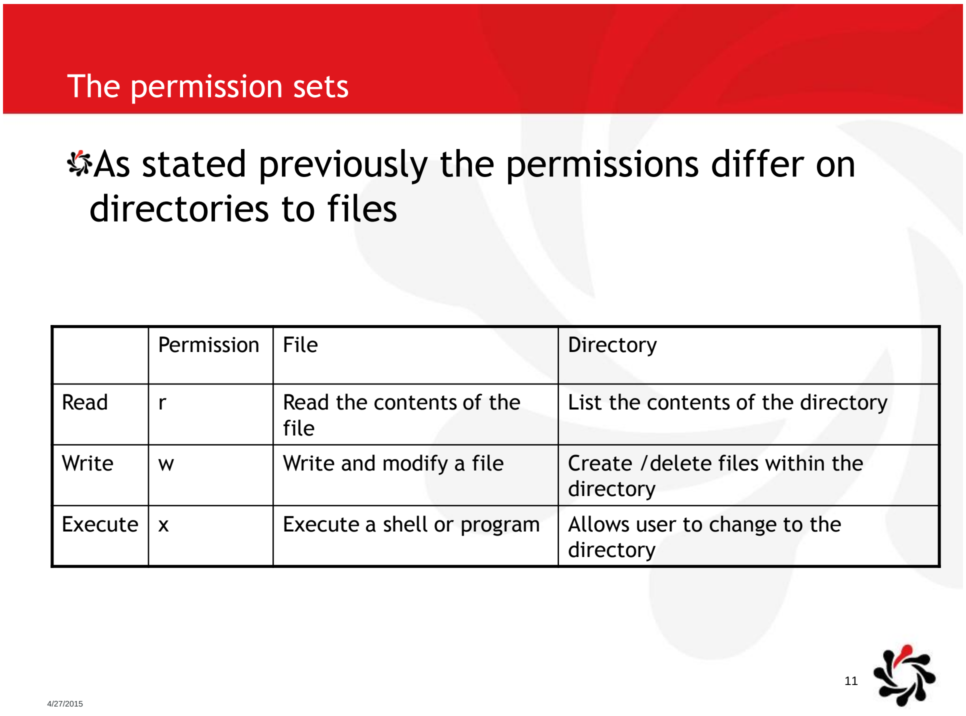
- Manage access permissions on regular and special files as well as directories.
- Use access modes such as suid, sgid and the sticky bit to maintain security.
- Know how to change the file creation mask.
- Use the group field to grant file access to group members.

## The basic permissions

- Permissions can be assigned to files and directories, and they have different meanings when applied.
- A typical permission set looks like

| Type | Owner | Group | Other | Owner name | Group Owner | File name |
|---|---|---|---|---|---|---|
| -/d/l | rwx | rwx | rwx | luke | users | filename |

- Type - This describes if it is a file/directory/link
- Owner - The permissions for the item if you  where the owner of the item.
- Group  - The permission set for the item if you where a member of the group
- Other  - The permission set if you did not fall into the owner or group
- Owner - The owner of the item
- Group Owner - The Group who owns the item



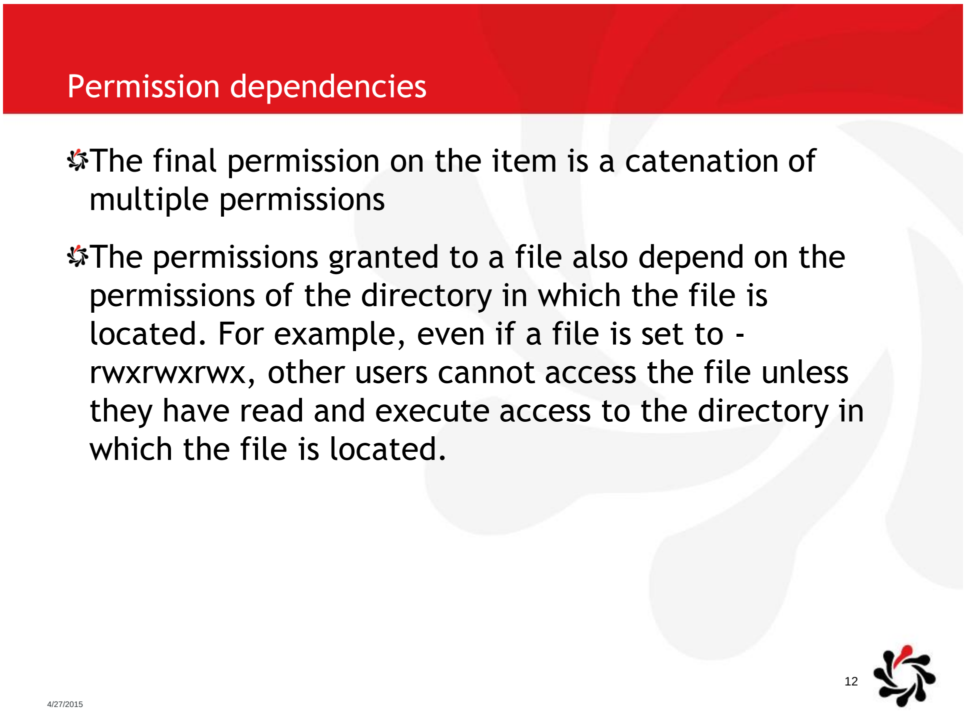
## The permission sets

- As stated previously the permissions differ on directories to files

| | Permission | File | Directory |
|---|---|---|---|
| Read | r | Read the contents of the file | List the contents of the directory |
| Write | w | Write and modify a file | Create /delete files within the directory |
| Execute | x | Execute a shell or program | Allows user to change to the directory |

4/27/2015

11

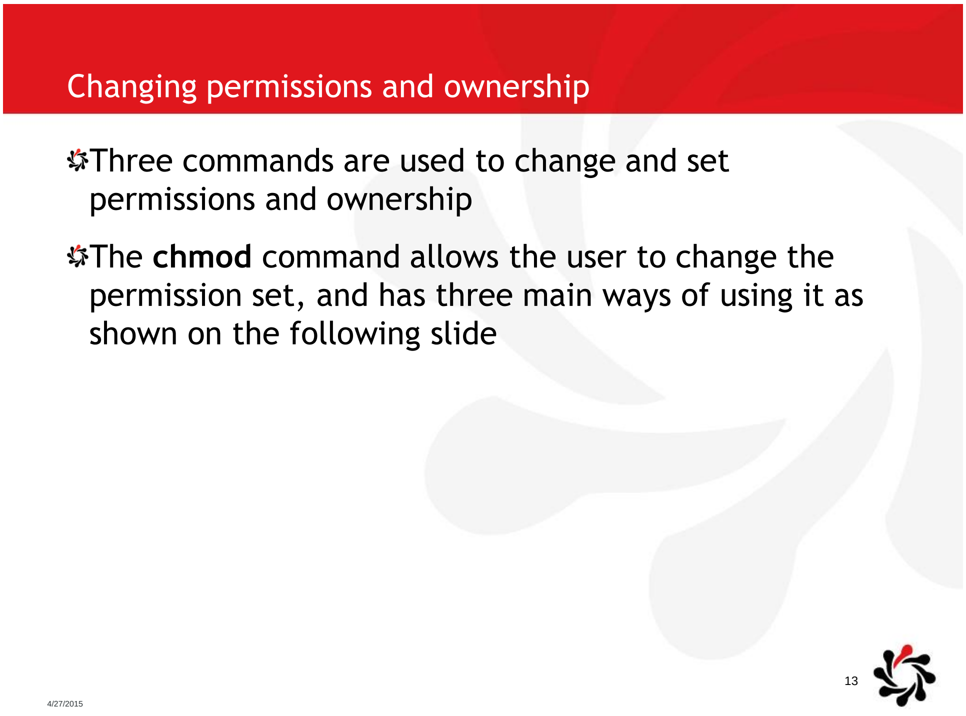


## Permission dependencies

- The final permission on the item is a catenation of multiple permissions
- The permissions granted to a file also depend on the permissions of the directory in which the file is located. For example, even if a file is set to -rwxrwxrwx, other users cannot access the file unless they have read and execute access to the directory in which the file is located.

4/27/2015

12

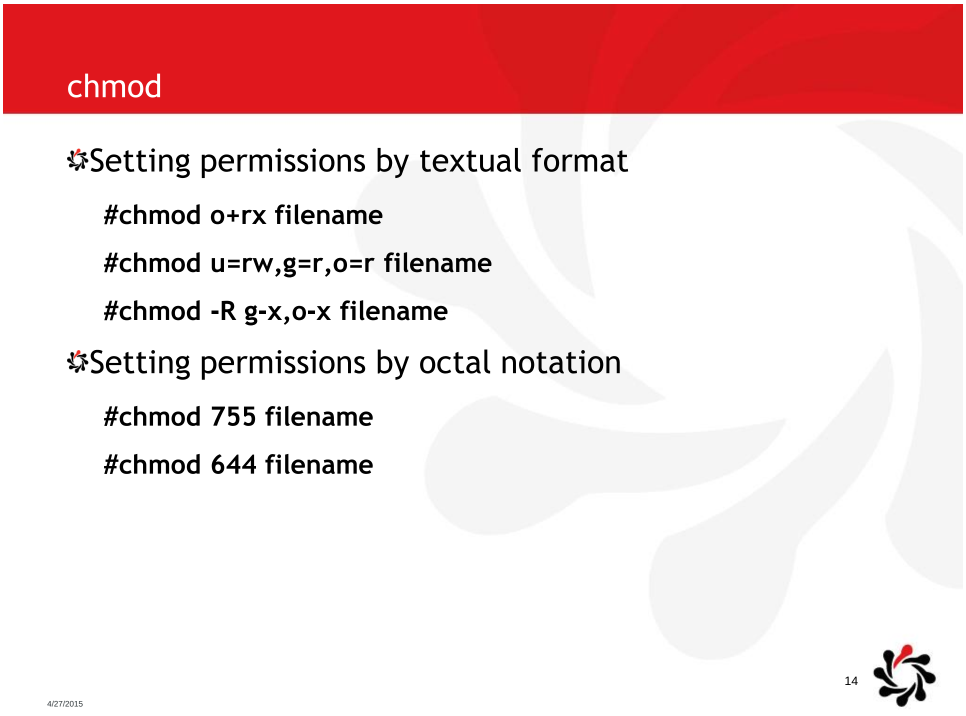


## Changing permissions and ownership

- Three commands are used to change and set permissions and ownership
- The **chmod** command allows the user to change the permission set, and has three main ways of using it as shown on the following slide

4/27/2015

13



## chmod

- Setting permissions by textual format

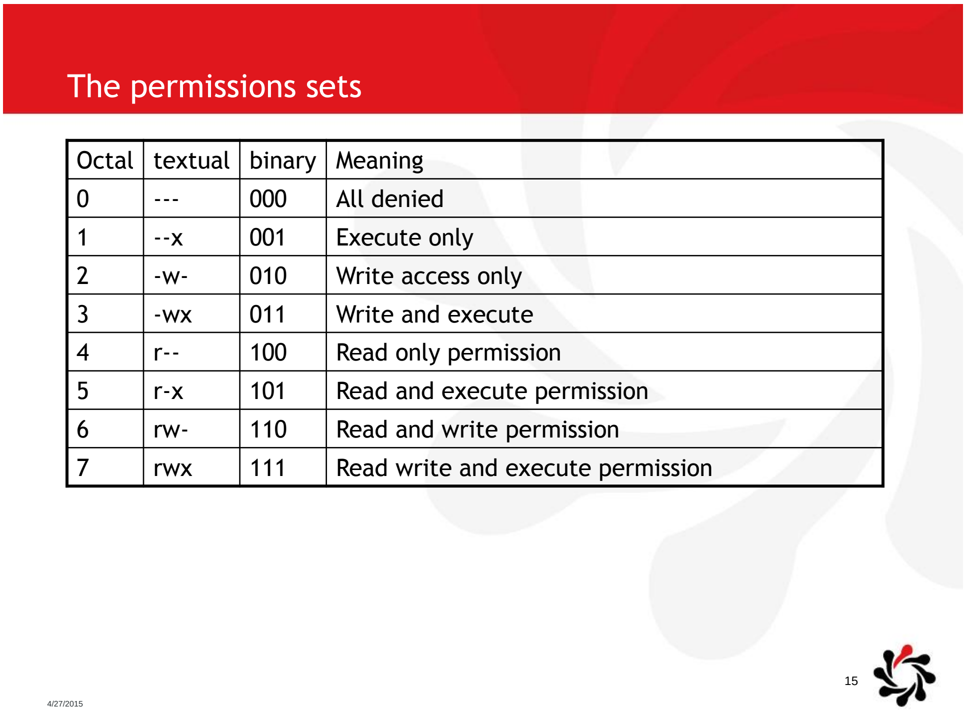
```
#chmod o+rx filename
#chmod u=rw,g=r,o=r filename
#chmod -R g-x,o-x filename
```

- Setting permissions by octal notation

```
#chmod 755 filename
#chmod 644 filename
```

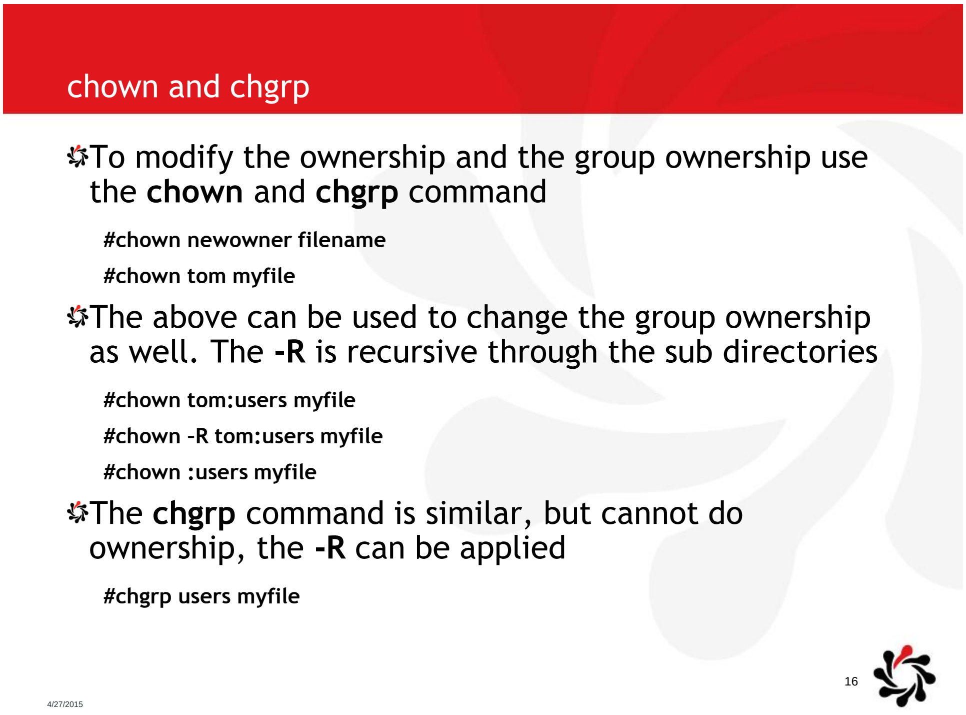
4/27/2015

14



## The permissions sets

| Octal | textual | binary | Meaning |
|---|---|---|---|
| 0 | --- | 000 | All denied |
| 1 | --x | 001 | Execute only |
| 2 | -w- | 010 | Write access only |
| 3 | -wx | 011 | Write and execute |
| 4 | r-- | 100 | Read only permission |
| 5 | r-x | 101 | Read and execute permission |
| 6 | rw- | 110 | Read and write permission |
| 7 | rwx | 111 | Read write and execute permission |



## chown and chgrp

- To modify the ownership and the group ownership use the **chown** and **chgrp** command

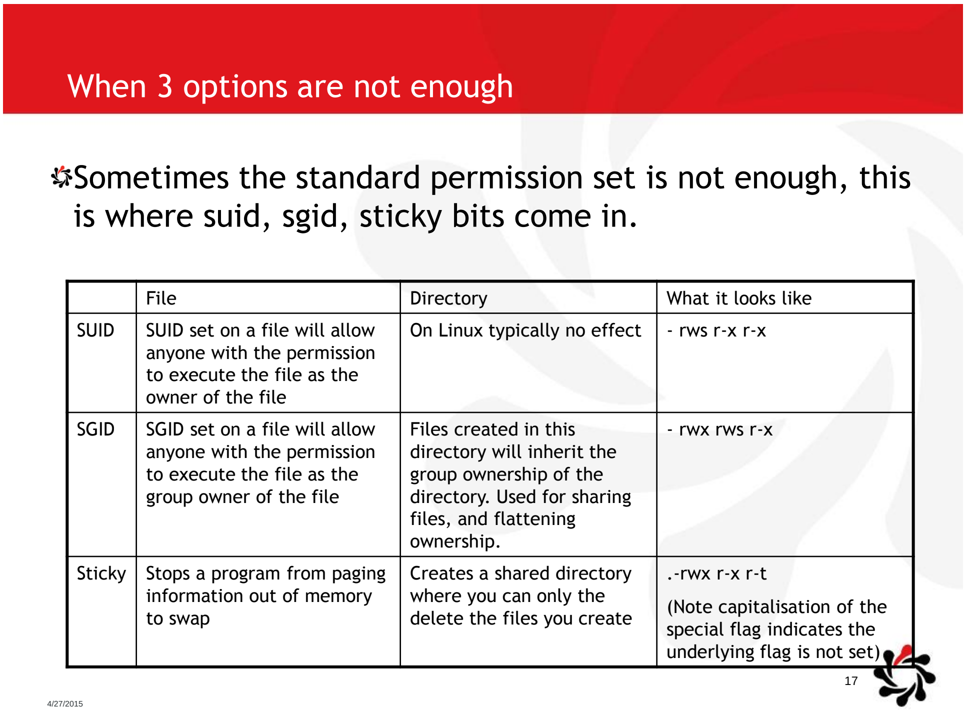
```
#chown newowner filename
#chown tom myfile
```

- The above can be used to change the group ownership as well. The **-R** is recursive through the sub directories

```
#chown tom:users myfile
#chown -R tom:users myfile
#chown :users myfile
```

- The **chgrp** command is similar, but cannot do ownership, the **-R** can be applied

```
#chgrp users myfile
```



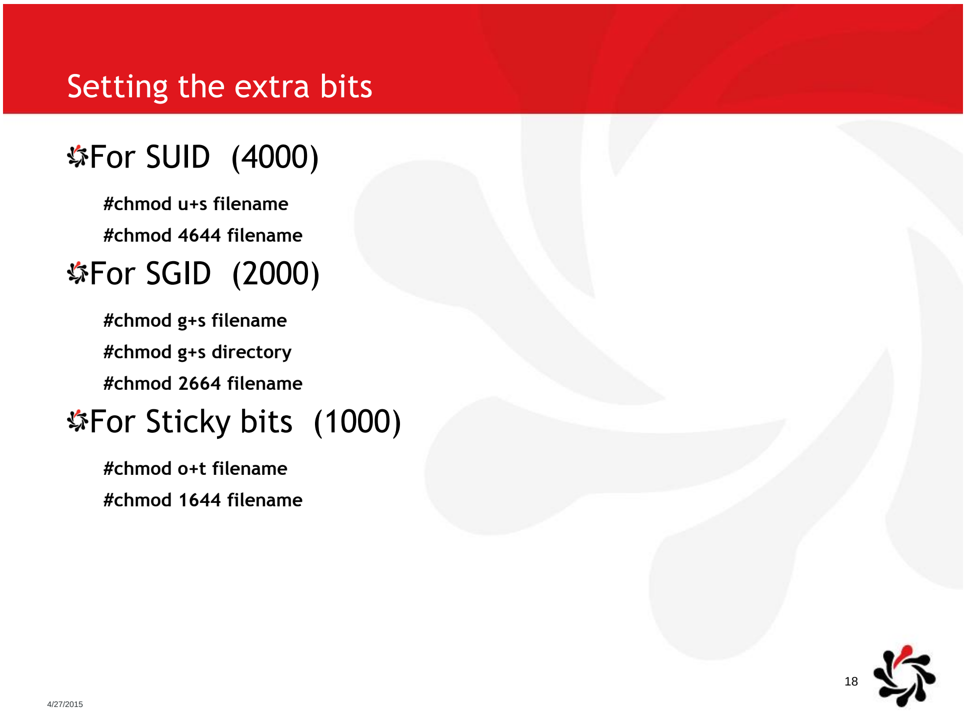
## When 3 options are not enough

- Sometimes the standard permission set is not enough, this is where suid, sgid, sticky bits come in.

| | File | Directory | What it looks like |
|---|---|---|---|
| SUID | SUID set on a file will allow anyone with the permission to execute the file as the owner of the file | On Linux typically no effect | - rws r-x r-x |
| SGID | SGID set on a file will allow anyone with the permission to execute the file as the group owner of the file | Files created in this directory will inherit the group ownership of the directory. Used for sharing files, and flattening ownership. | - rwx rws r-x |
| Sticky | Stops a program from paging information out of memory to swap | Creates a shared directory where you can only the delete the files you create | .-rwx r-x r-t<br>(Note capitalisation of the special flag indicates the underlying flag is not set) |

17

4/27/2015



## Setting the extra bits

- For SUID (4000)

```
#chmod u+s filename
#chmod 4644 filename
```

- For SGID (2000)

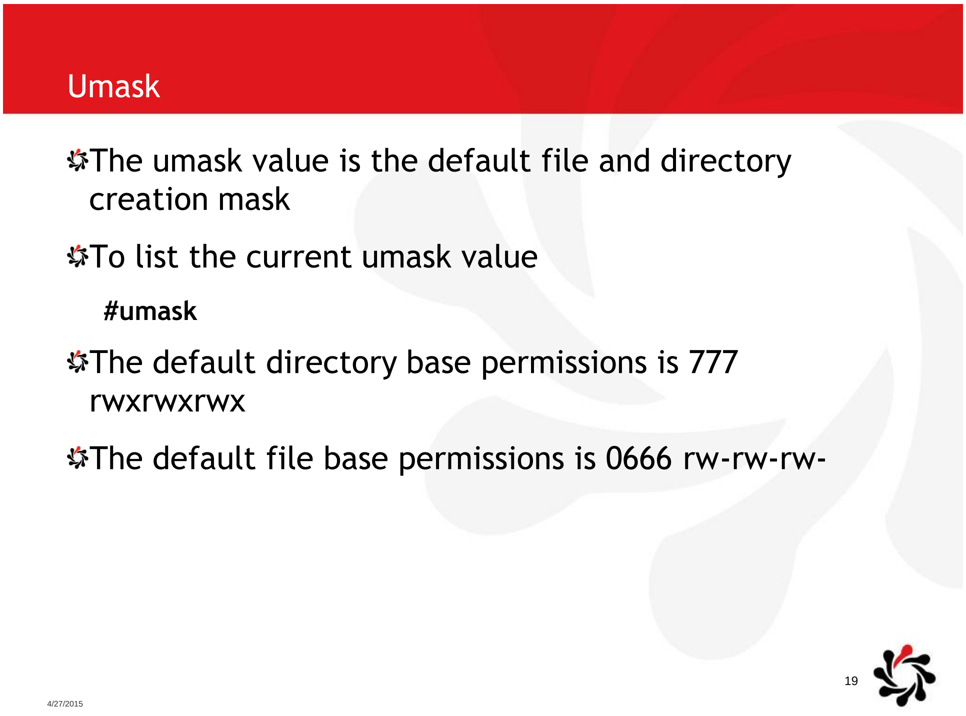
```
#chmod g+s filename
#chmod g+s directory
#chmod 2664 filename
```

- For Sticky bits (1000)

```
#chmod o+t filename
#chmod 1644 filename
```

18

4/27/2015



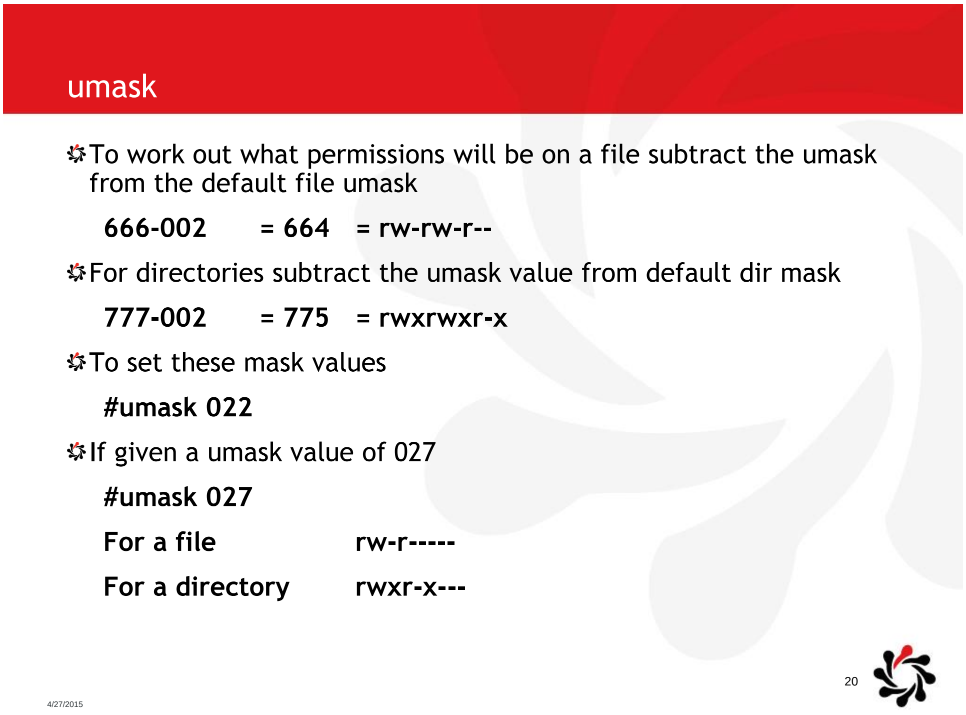
## Umask

- The umask value is the default file and directory creation mask
- To list the current umask value

  **#umask**
- The default directory base permissions is 777 rwxrwxrwx
- The default file base permissions is 0666 rw-rw-rw-

19

4/27/2015



## umask

- To work out what permissions will be on a file subtract the umask from the default file umask

  **666-002 = 664 = rw-rw-r--**
- For directories subtract the umask value from default dir mask

  **777-002 = 775 = rwxrwxr-x**
- To set these mask values

  **#umask 022**
- If given a umask value of 027

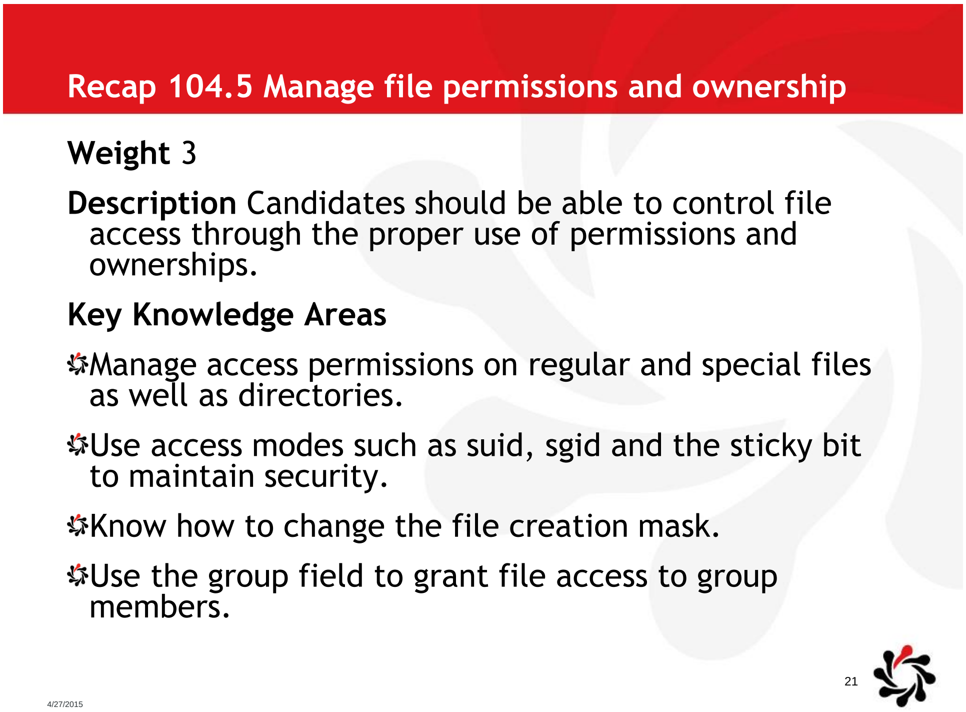
  **#umask 027**

  **For a file rw-r-----**

  **For a directory rwxr-x---**

20

4/27/2015



## Recap 104.5 Manage file permissions and ownership

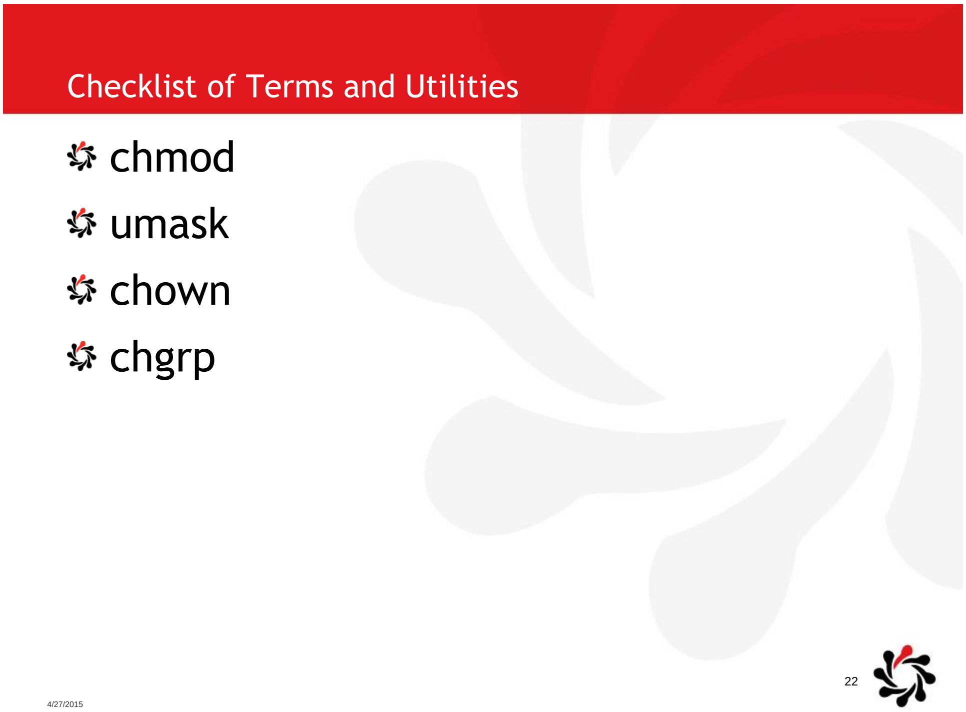
**Weight** 3

**Description** Candidates should be able to control file access through the proper use of permissions and ownerships.

**Key Knowledge Areas**

- Manage access permissions on regular and special files as well as directories.
- Use access modes such as suid, sgid and the sticky bit to maintain security.
- Know how to change the file creation mask.
- Use the group field to grant file access to group members.

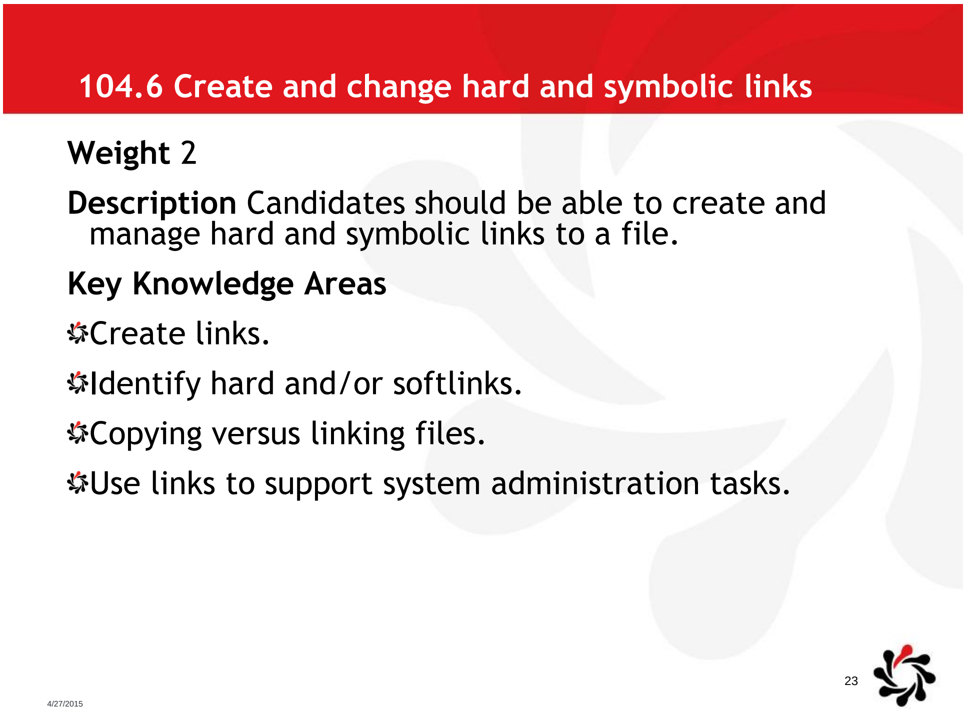
4/27/2015 21



## Checklist of Terms and Utilities

- chmod
- umask
- chown
- chgrp

4/27/2015 22



## 104.6 Create and change hard and symbolic links

**Weight** 2

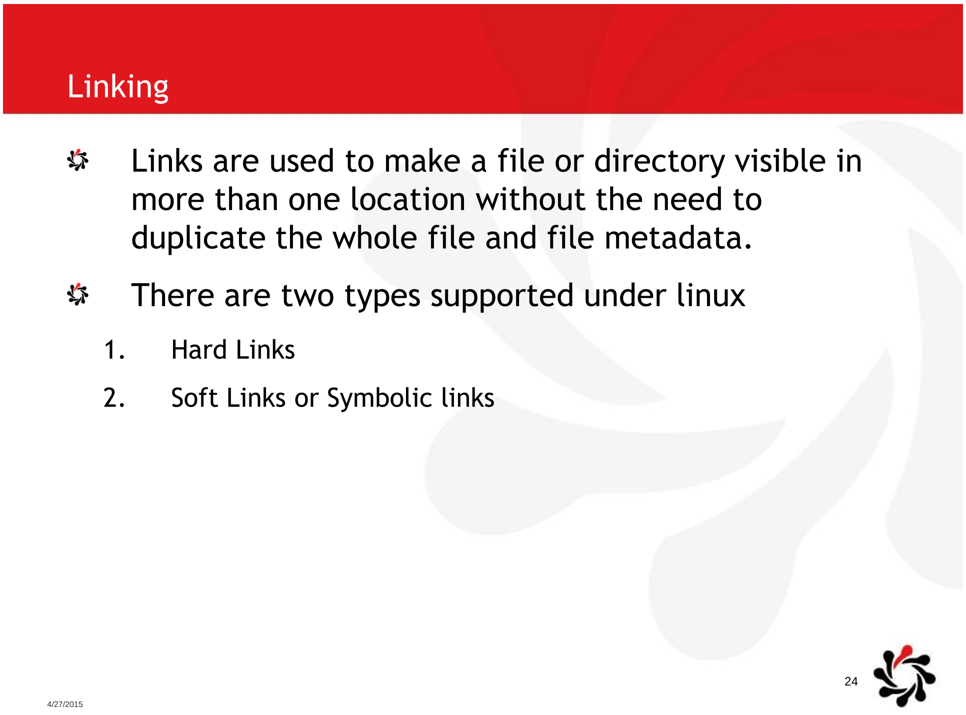
**Description** Candidates should be able to create and manage hard and symbolic links to a file.

**Key Knowledge Areas**

- Create links.
- Identify hard and/or softlinks.
- Copying versus linking files.
- Use links to support system administration tasks.

4/27/2015

23



## Linking

- Links are used to make a file or directory visible in more than one location without the need to duplicate the whole file and file metadata.
- There are two types supported under linux
  1. Hard Links
  2. Soft Links or Symbolic links

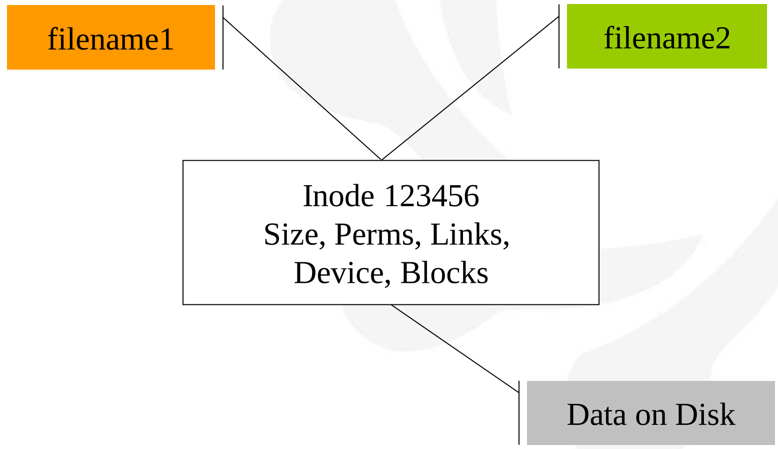
4/27/2015

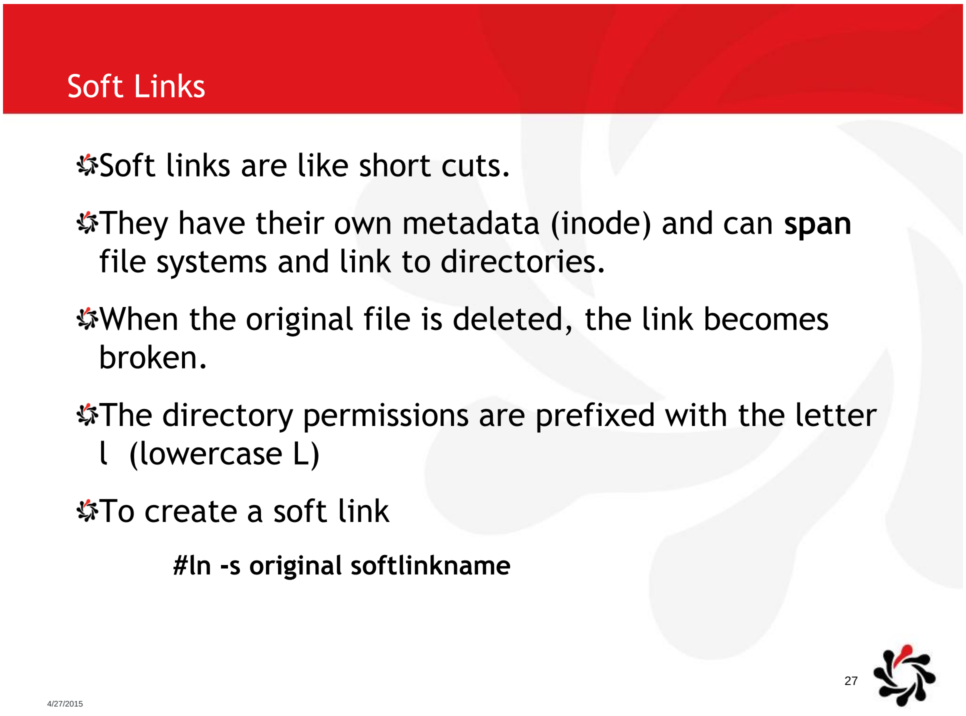
24

## Hard Links

- Hard links are a multiple filenames pointing to one piece of metadata (inode).
- This means they all share the same permissions.
- When the inode link count reaches zero, all filenames have been unlinked from the metadata, and therefore it is deleted.
- Hard links cannot span file system and cannot link to directories
- To create a hard link

```
#ln original hardlinkname
```





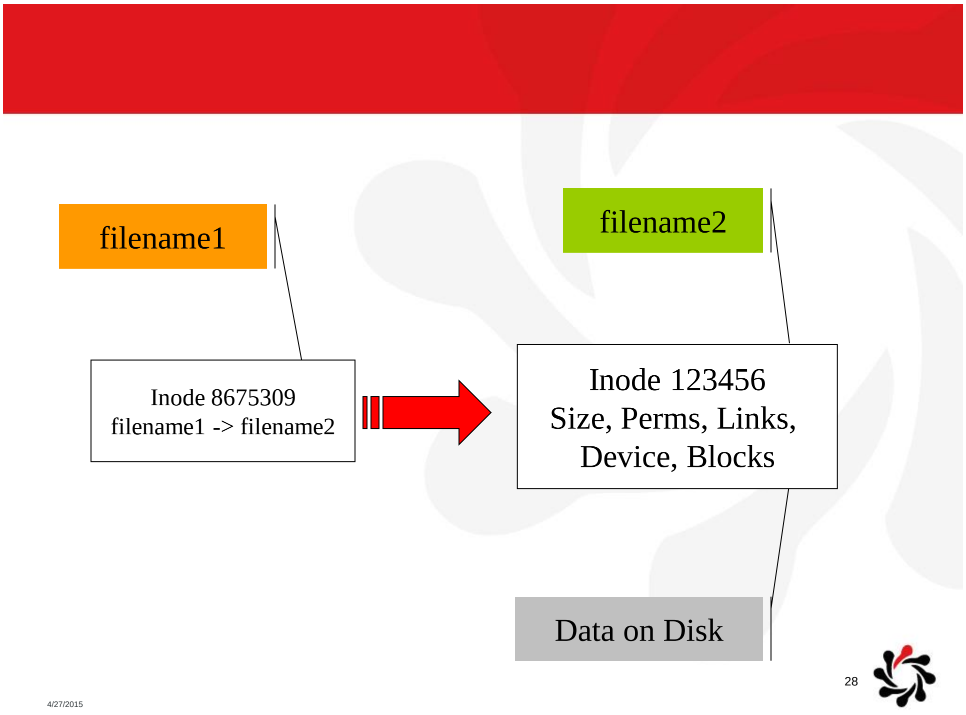
## Soft Links

- Soft links are like short cuts.
- They have their own metadata (inode) and can **span** file systems and link to directories.
- When the original file is deleted, the link becomes broken.
- The directory permissions are prefixed with the letter l (lowercase L)
- To create a soft link

```
#ln -s original softlinkname
```

4/27/2015

27



filename1

filename2

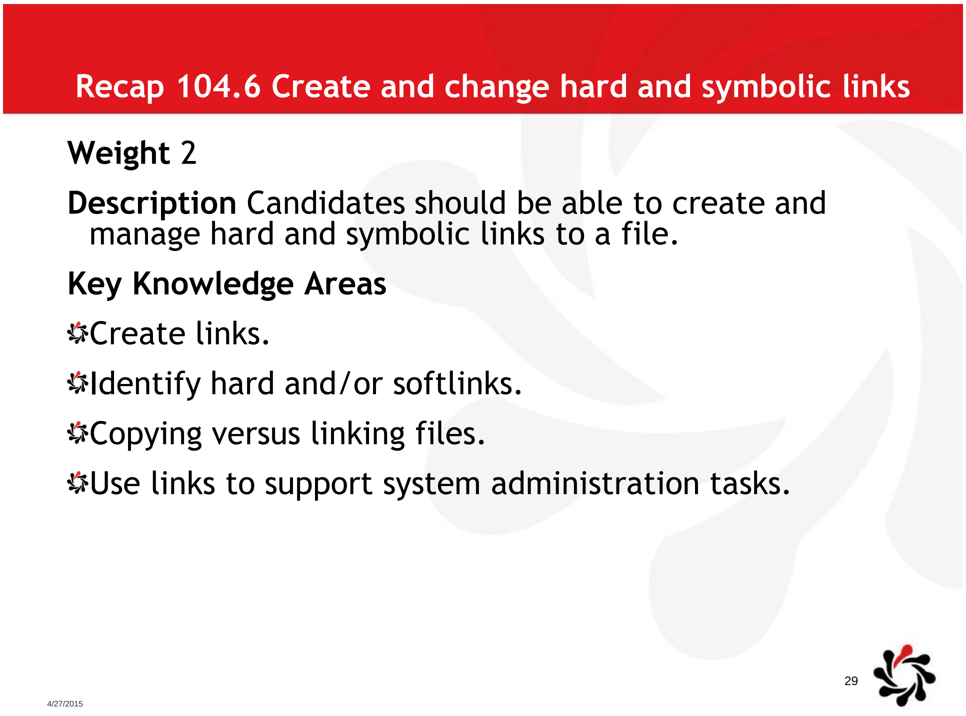
Inode 8675309
filename1 -> filename2

Inode 123456
Size, Perms, Links,
Device, Blocks

Data on Disk

4/27/2015

28



## Recap 104.6 Create and change hard and symbolic links

**Weight** 2

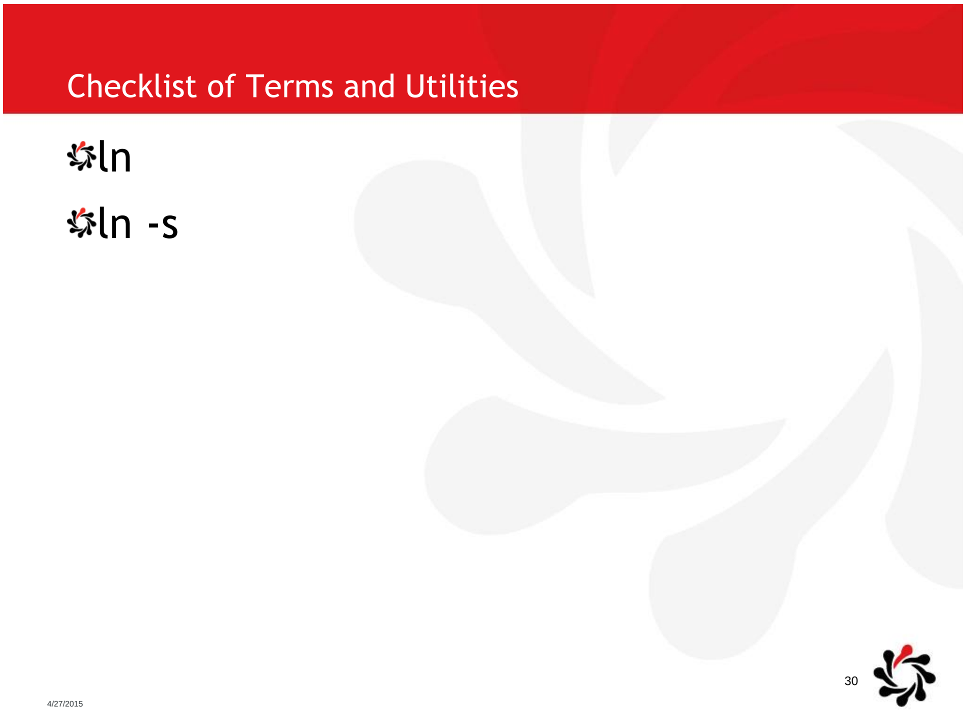
**Description** Candidates should be able to create and manage hard and symbolic links to a file.

**Key Knowledge Areas**

- Create links.
- Identify hard and/or softlinks.
- Copying versus linking files.
- Use links to support system administration tasks.

4/27/2015

29



## Checklist of Terms and Utilities

- ln
- ln -s

4/27/2015

30

## 104.7 Find system files and place files in the correct location

**Weight** 2

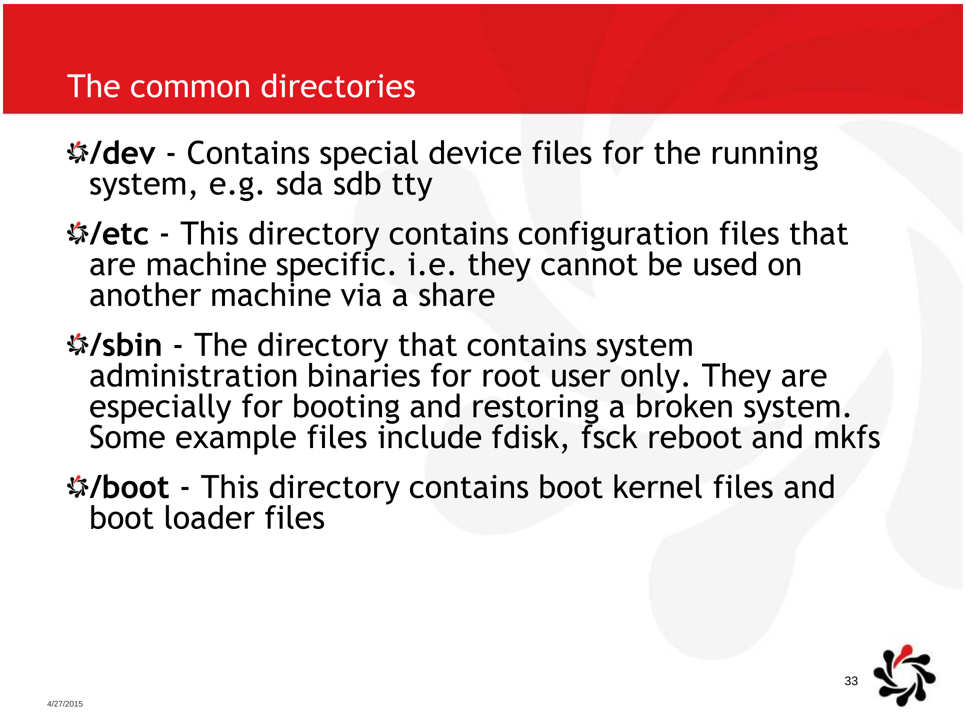
**Description** Candidates should be thoroughly familiar with the File system Hierarchy Standard (FHS), including typical file locations and directory classifications.

**Key Knowledge Areas**

- Understand the correct locations of files under the FHS.
- Find files and commands on a Linux system.
- Know the location and propose of important file and directories as defined in the FHS.

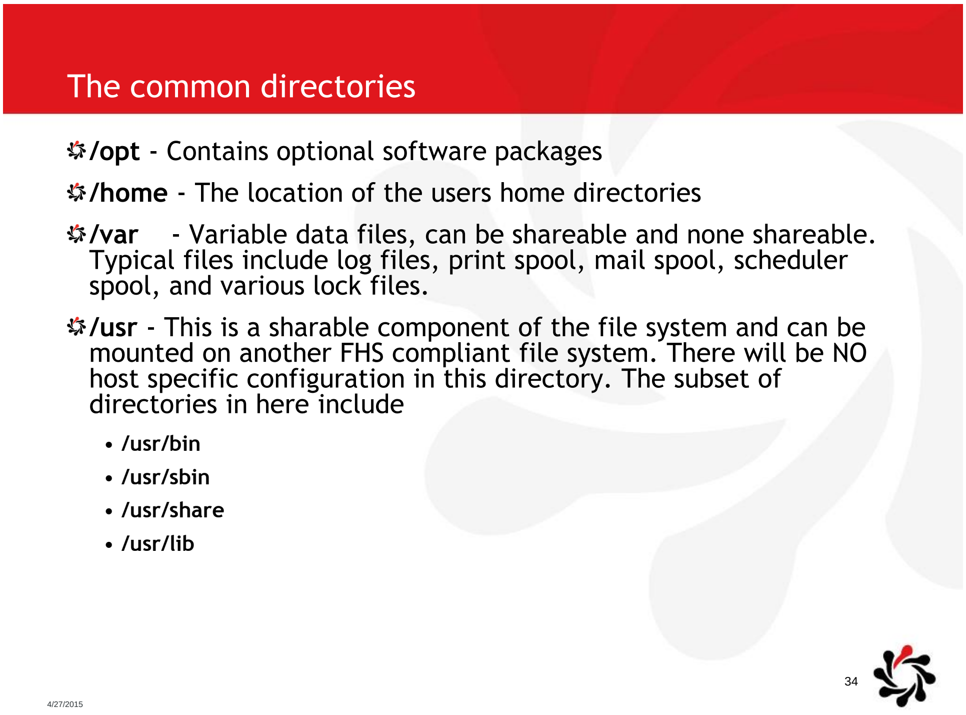
## The FHS

- The FHS as described earlier is a standard that defines the directory structure and what should be located in each directory.
- The current version of this document is V2.3 can be found at
  - http://www.pathname.com/fhs
- The document can help software designers to write valid installation scripts for there software and aid system admins in knowing where to find specific files on the system



## The common directories

- **/dev** - Contains special device files for the running system, e.g. sda sdb tty
- **/etc** - This directory contains configuration files that are machine specific. i.e. they cannot be used on another machine via a share
- **/sbin** - The directory that contains system administration binaries for root user only. They are especially for booting and restoring a broken system. Some example files include fdisk, fsck reboot and mkfs
- **/boot** - This directory contains boot kernel files and boot loader files



## The common directories

- **/opt** - Contains optional software packages
- **/home** - The location of the users home directories
- **/var** - Variable data files, can be shareable and none shareable. Typical files include log files, print spool, mail spool, scheduler spool, and various lock files.
- **/usr** - This is a sharable component of the file system and can be mounted on another FHS compliant file system. There will be NO host specific configuration in this directory. The subset of directories in here include
  - /usr/bin
  - /usr/sbin
  - /usr/share
  - /usr/lib

## The common directories

- **/tmp** - Is a store for temporary data for applications running. The directory has a stickybit set on it so anyone can write into it, but only the owner can delete it
- **/bin** - This directory is for general user commands e.g. grep, touch, pwd
- **/lib** - This contains the library files used by ELF binaries, **shareable** libraries should be stored in **/usr/lib**

## Examples of the find command

- The find command is used to exhaustively search the file system, regardless of path
- To find a filename fstab starting the search in your pwd

```
#find . -name fstab
```

- Find any file owned by luke which has changed in the last 24 hours

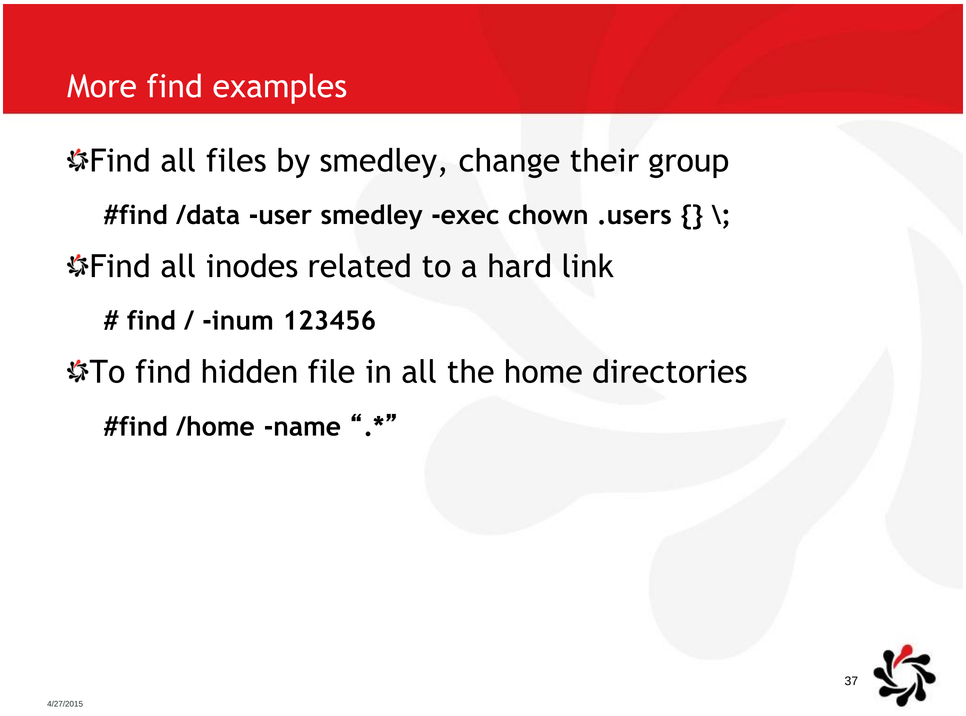
```
#find /home -user luke -mtime 0
```

- Find any file with an SUID set on it

```
#find / -perm +4000
```

- Find any file owned by group users, stop at 10 sub dirs.

```
#find / -group users -maxdepth 10
```



## More find examples

- Find all files by smedley, change their group

  **#find /data -user smedley -exec chown .users {} \;**

- Find all inodes related to a hard link

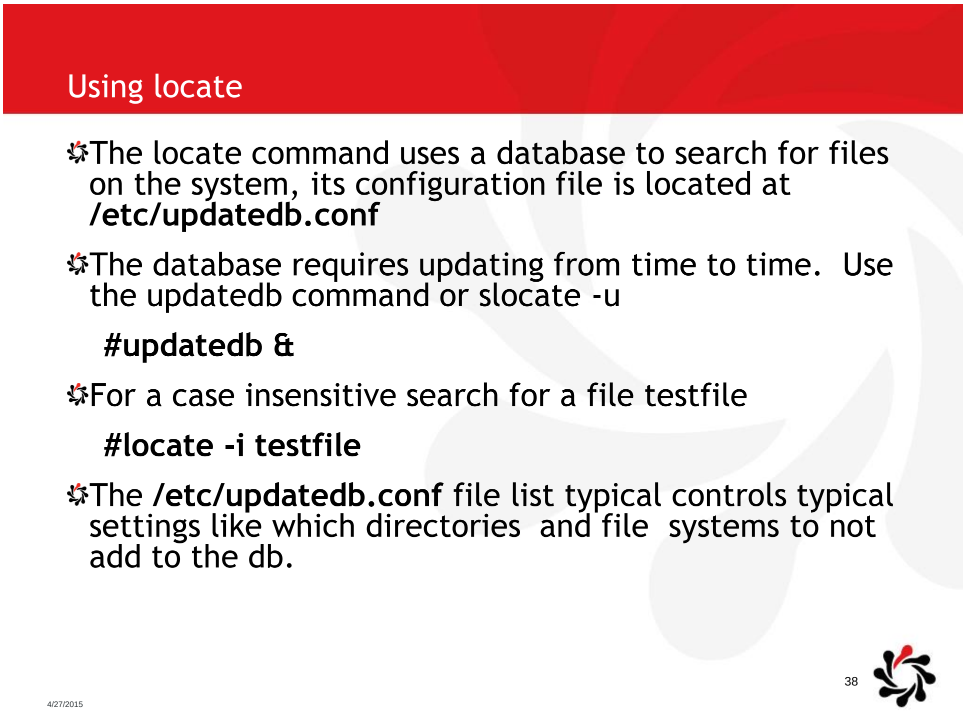
  **# find / -inum 123456**

- To find hidden file in all the home directories

  **#find /home -name “.*”**

37

4/27/2015



## Using locate

- The locate command uses a database to search for files on the system, its configuration file is located at **/etc/updatedb.conf**
- The database requires updating from time to time.  Use the updatedb command or slocate -u

  **#updatedb &**

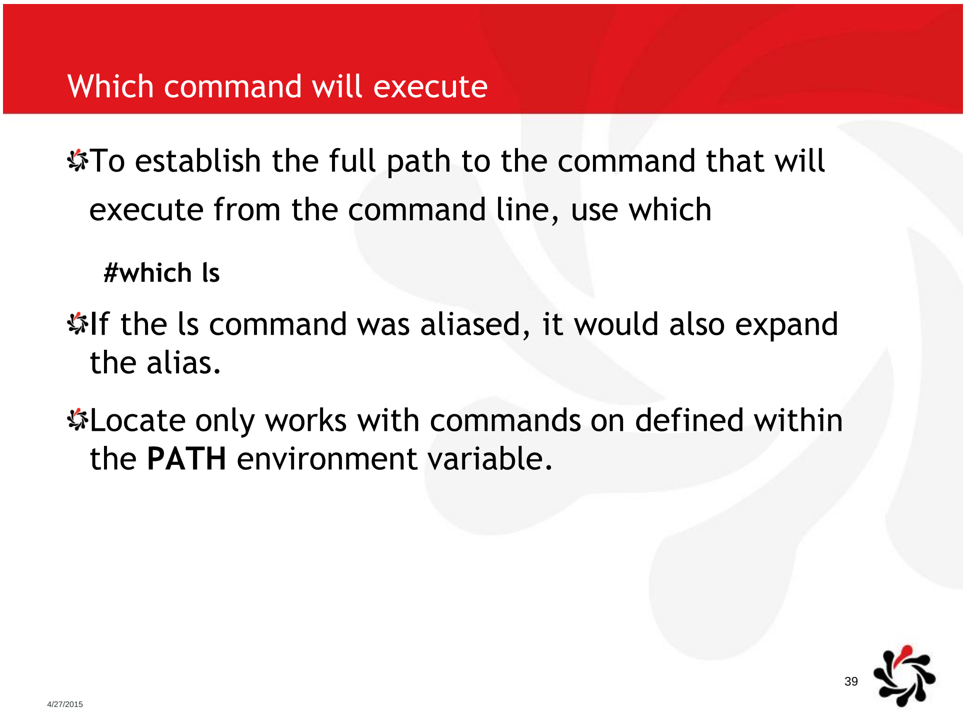
- For a case insensitive search for a file testfile

  **#locate -i testfile**

- The **/etc/updatedb.conf** file list typical controls typical settings like which directories  and file  systems to not add to the db.

38

4/27/2015

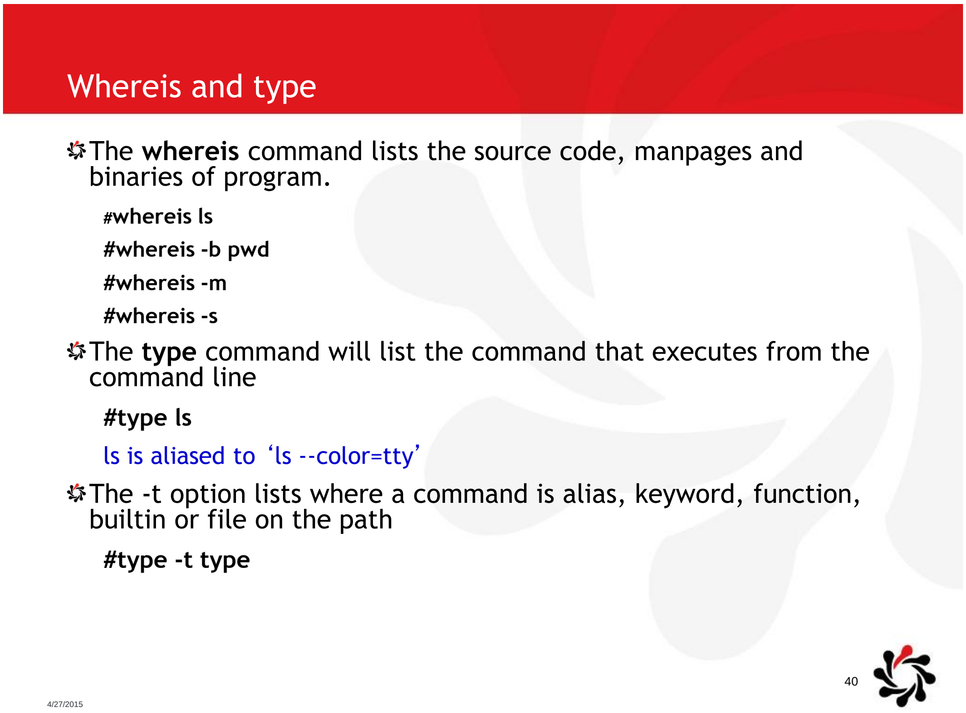


## Which command will execute

- To establish the full path to the command that will execute from the command line, use which

```
#which ls
```

- If the ls command was aliased, it would also expand the alias.
- Locate only works with commands on defined within the **PATH** environment variable.



## Whereis and type

- The **whereis** command lists the source code, manpages and binaries of program.

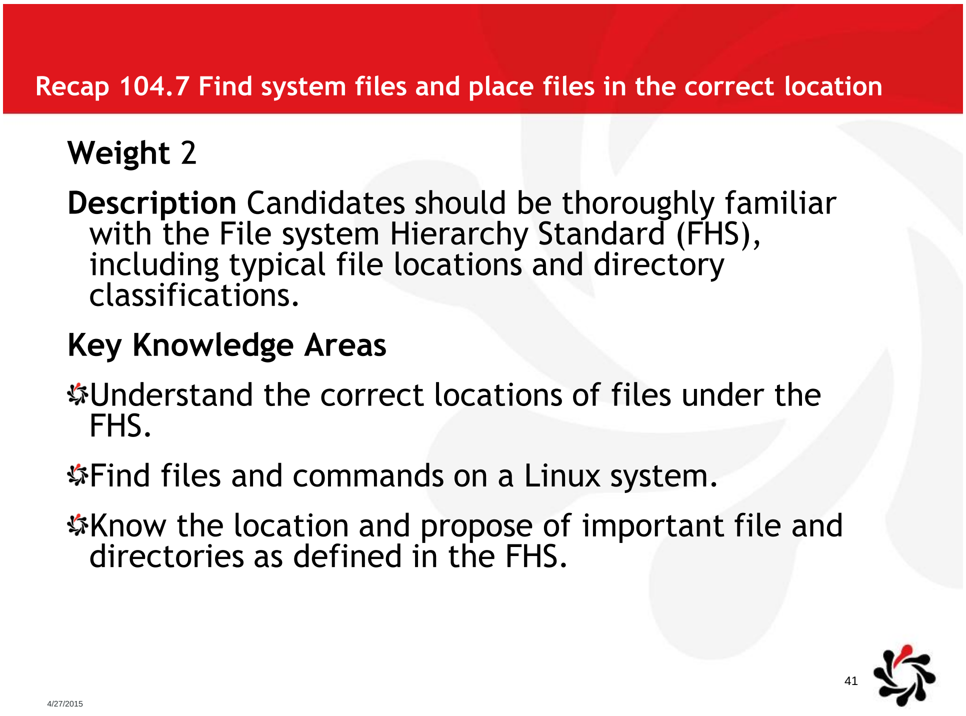
```
#whereis ls
#whereis -b pwd
#whereis -m
#whereis -s
```

- The **type** command will list the command that executes from the command line

```
#type ls
ls is aliased to ‘ls --color=tty’
```

- The -t option lists where a command is alias, keyword, function, builtin or file on the path

```
#type -t type
```



## Recap 104.7 Find system files and place files in the correct location

**Weight** 2

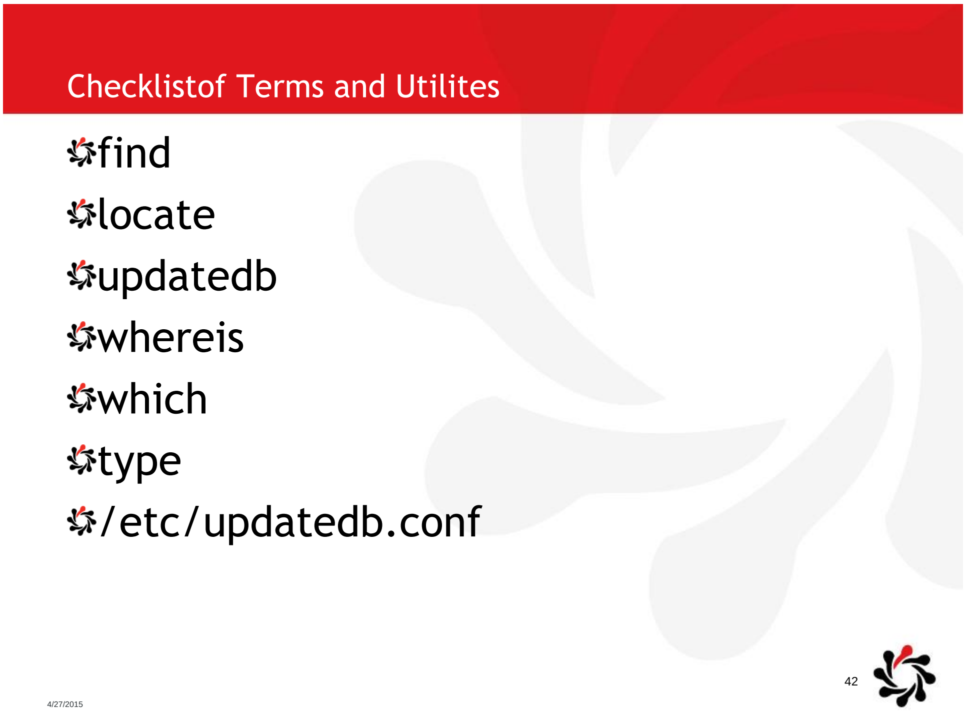
**Description** Candidates should be thoroughly familiar with the File system Hierarchy Standard (FHS), including typical file locations and directory classifications.

**Key Knowledge Areas**

- Understand the correct locations of files under the FHS.
- Find files and commands on a Linux system.
- Know the location and propose of important file and directories as defined in the FHS.

4/27/2015

41

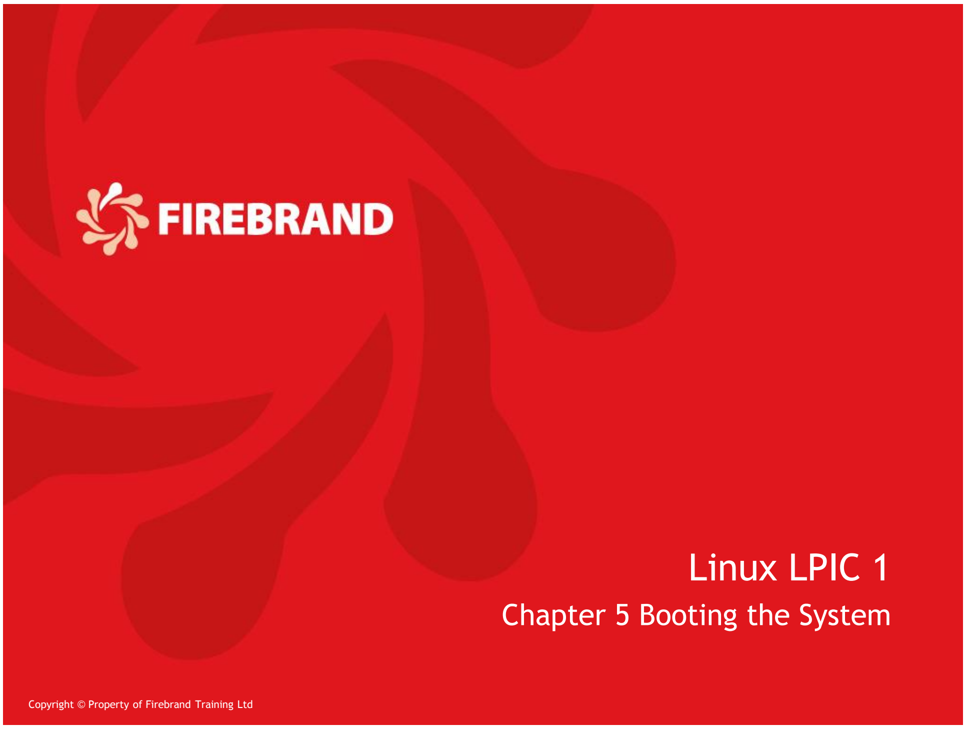


## Checklistof Terms and Utilites

- find
- locate
- updatedb
- whereis
- which
- type
- /etc/updatedb.conf

4/27/2015

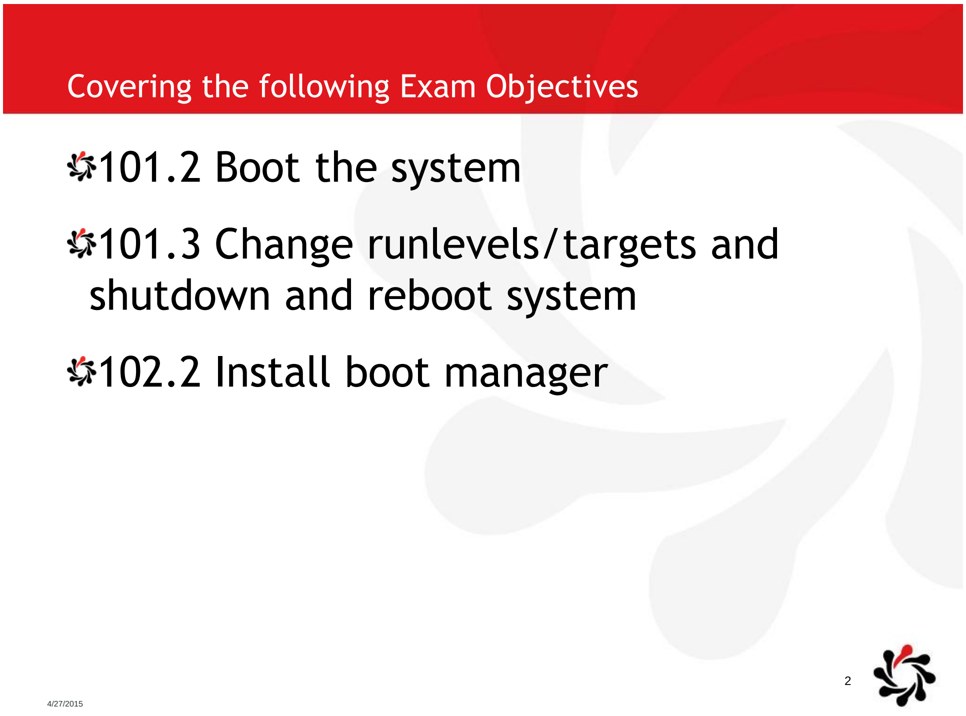
42



FIREBRAND

# Linux LPIC 1

## Chapter 5 Booting the System

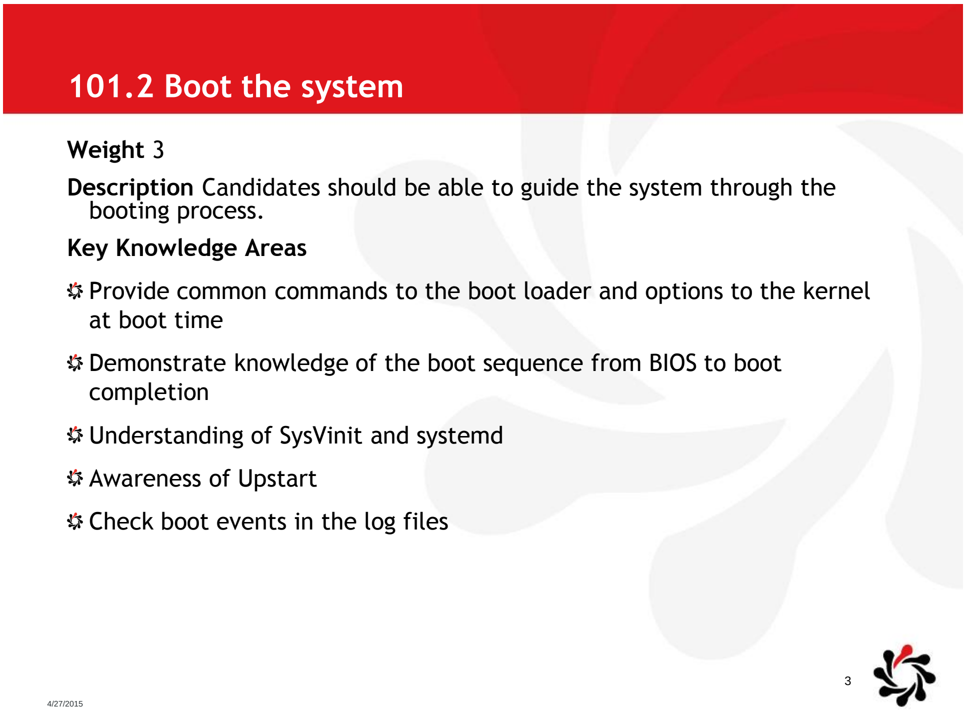


## Covering the following Exam Objectives

- 101.2 Boot the system
- 101.3 Change runlevels/targets and shutdown and reboot system
- 102.2 Install boot manager

4/27/2015

2



## 101.2 Boot the system

**Weight** 3

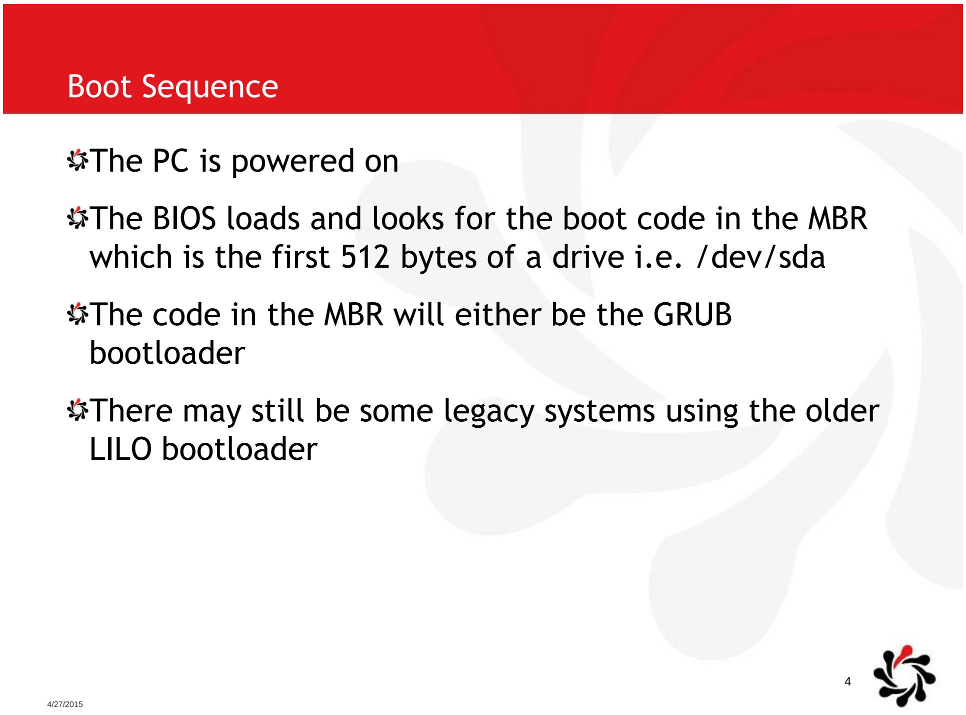
**Description** Candidates should be able to guide the system through the booting process.

**Key Knowledge Areas**

- Provide common commands to the boot loader and options to the kernel at boot time
- Demonstrate knowledge of the boot sequence from BIOS to boot completion
- Understanding of SysVinit and systemd
- Awareness of Upstart
- Check boot events in the log files

4/27/2015

3

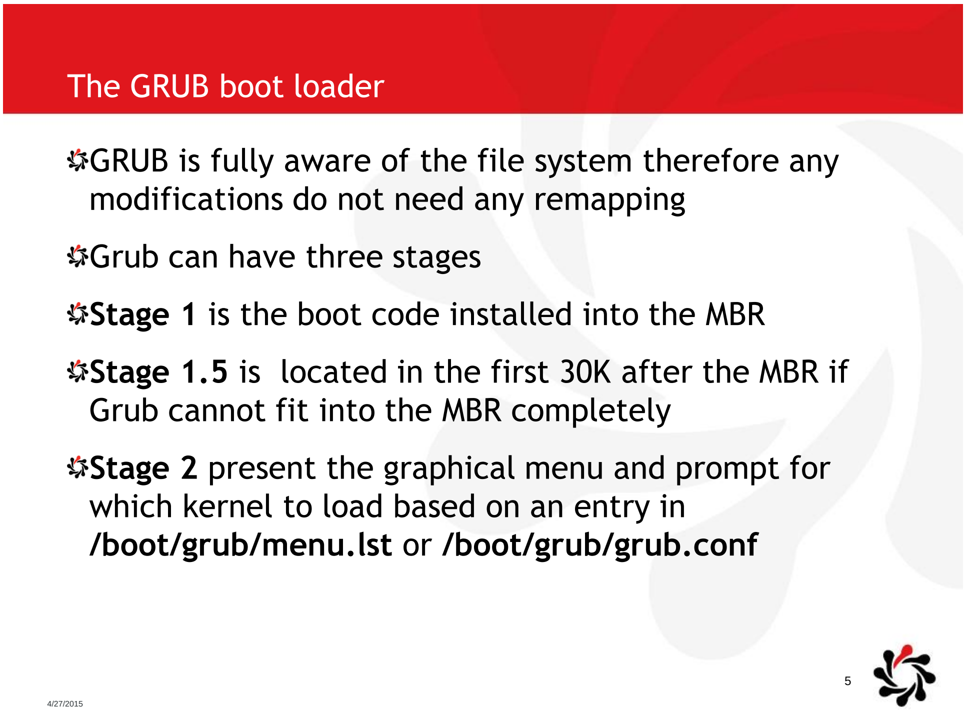


## Boot Sequence

- The PC is powered on
- The BIOS loads and looks for the boot code in the MBR which is the first 512 bytes of a drive i.e. /dev/sda
- The code in the MBR will either be the GRUB bootloader
- There may still be some legacy systems using the older LILO bootloader

4/27/2015

4

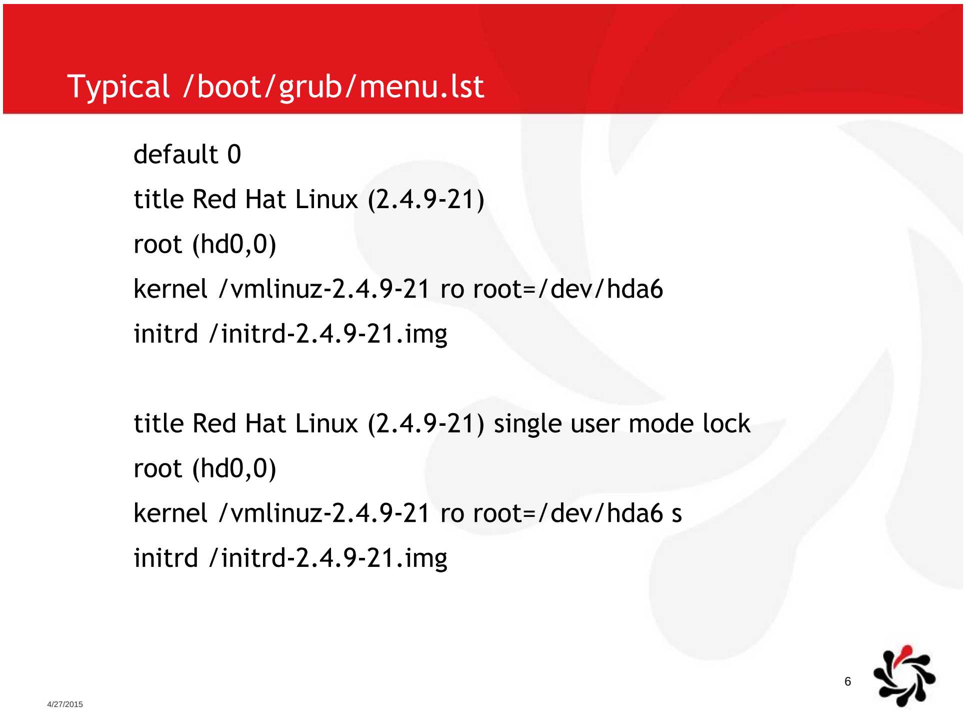


## The GRUB boot loader

- GRUB is fully aware of the file system therefore any modifications do not need any remapping
- Grub can have three stages
- **Stage 1** is the boot code installed into the MBR
- **Stage 1.5** is  located in the first 30K after the MBR if Grub cannot fit into the MBR completely
- **Stage 2** present the graphical menu and prompt for which kernel to load based on an entry in **/boot/grub/menu.lst** or **/boot/grub/grub.conf**

4/27/2015

5



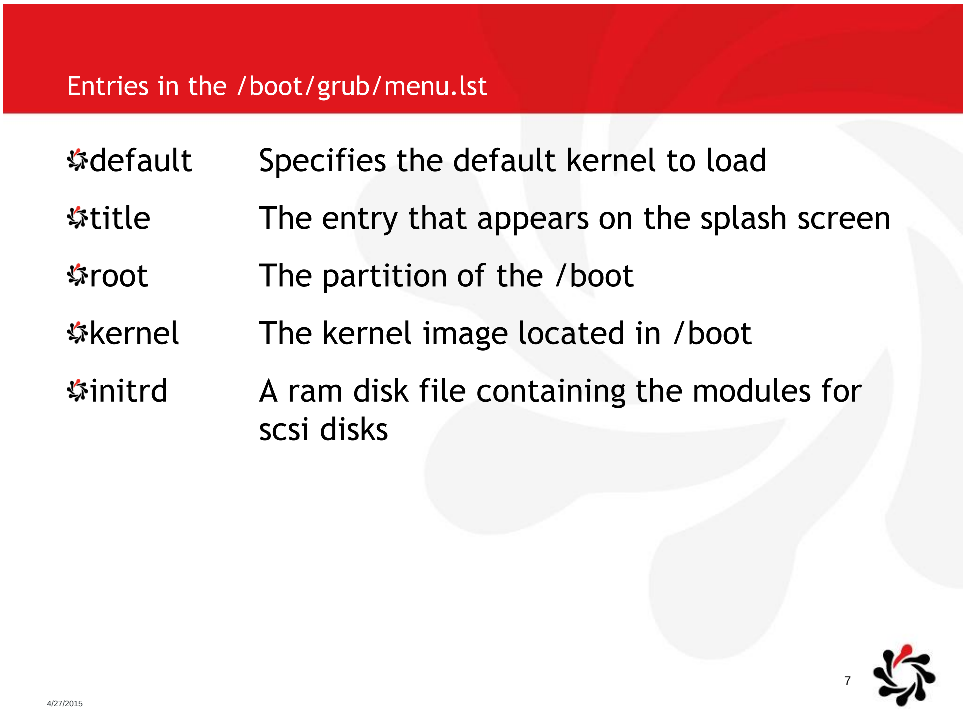
## Typical /boot/grub/menu.lst

```
default 0
title Red Hat Linux (2.4.9-21)
root (hd0,0)
kernel /vmlinuz-2.4.9-21 ro root=/dev/hda6
initrd /initrd-2.4.9-21.img

title Red Hat Linux (2.4.9-21) single user mode lock
root (hd0,0)
kernel /vmlinuz-2.4.9-21 ro root=/dev/hda6 s
initrd /initrd-2.4.9-21.img
```

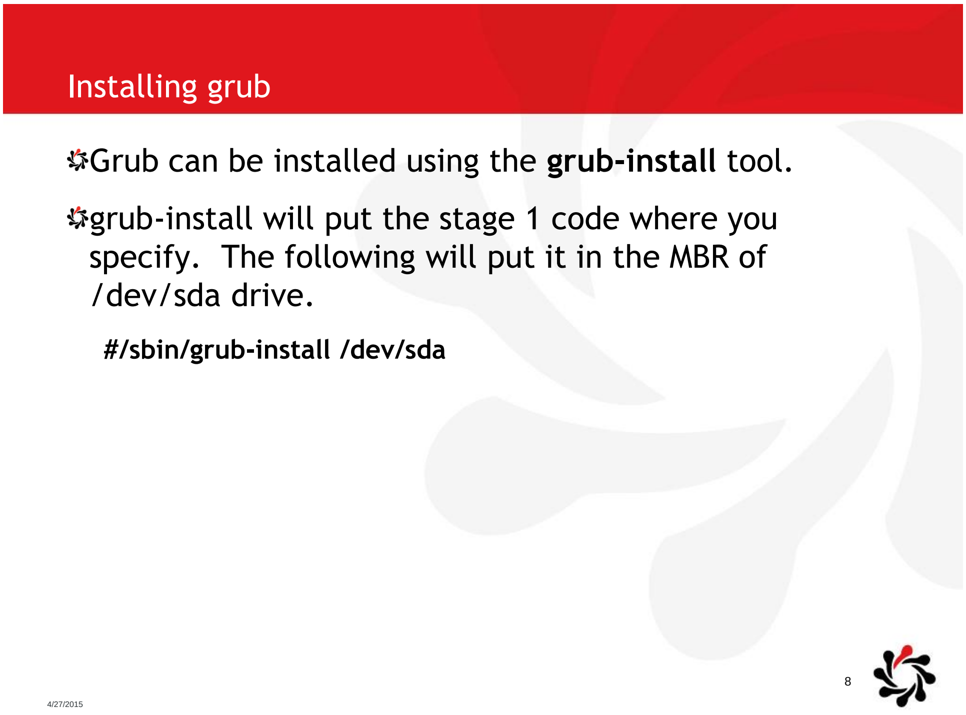
4/27/2015

6



## Entries in the /boot/grub/menu.lst

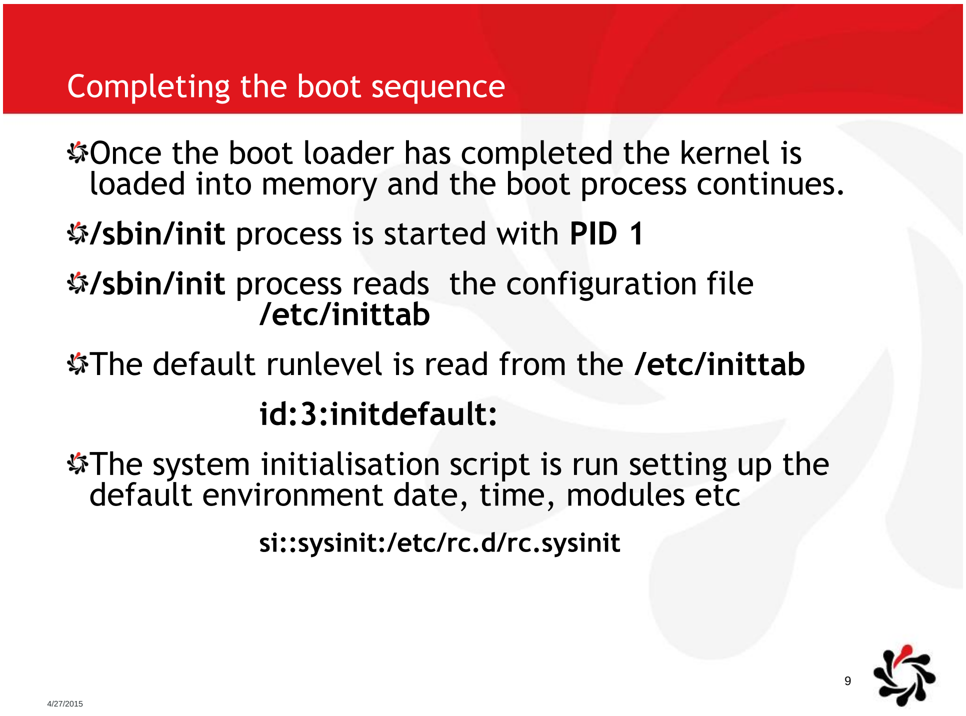
| | |
|---|---|
| default | Specifies the default kernel to load |
| title | The entry that appears on the splash screen |
| root | The partition of the /boot |
| kernel | The kernel image located in /boot |
| initrd | A ram disk file containing the modules for scsi disks |



## Installing grub

- Grub can be installed using the **grub-install** tool.
- grub-install will put the stage 1 code where you specify. The following will put it in the MBR of /dev/sda drive.

```
#/sbin/grub-install /dev/sda
```



## Completing the boot sequence

- Once the boot loader has completed the kernel is loaded into memory and the boot process continues.
- **/sbin/init** process is started with **PID 1**
- **/sbin/init** process reads the configuration file **/etc/inittab**
- The default runlevel is read from the **/etc/inittab**

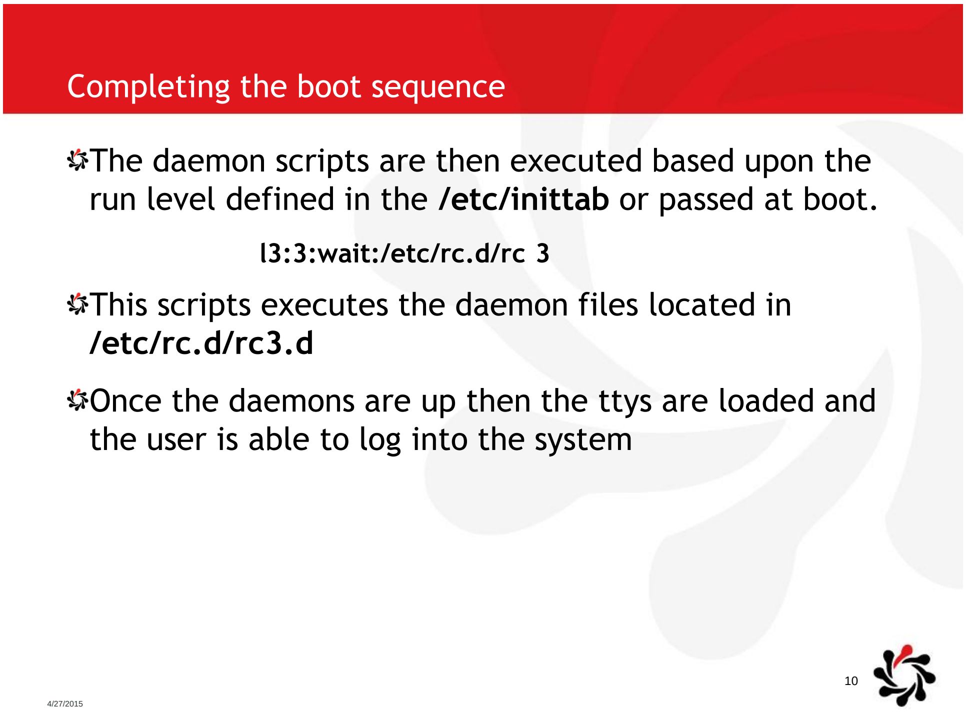
```
id:3:initdefault:
```

- The system initialisation script is run setting up the default environment date, time, modules etc

```
si::sysinit:/etc/rc.d/rc.sysinit
```

4/27/2015

9



## Completing the boot sequence

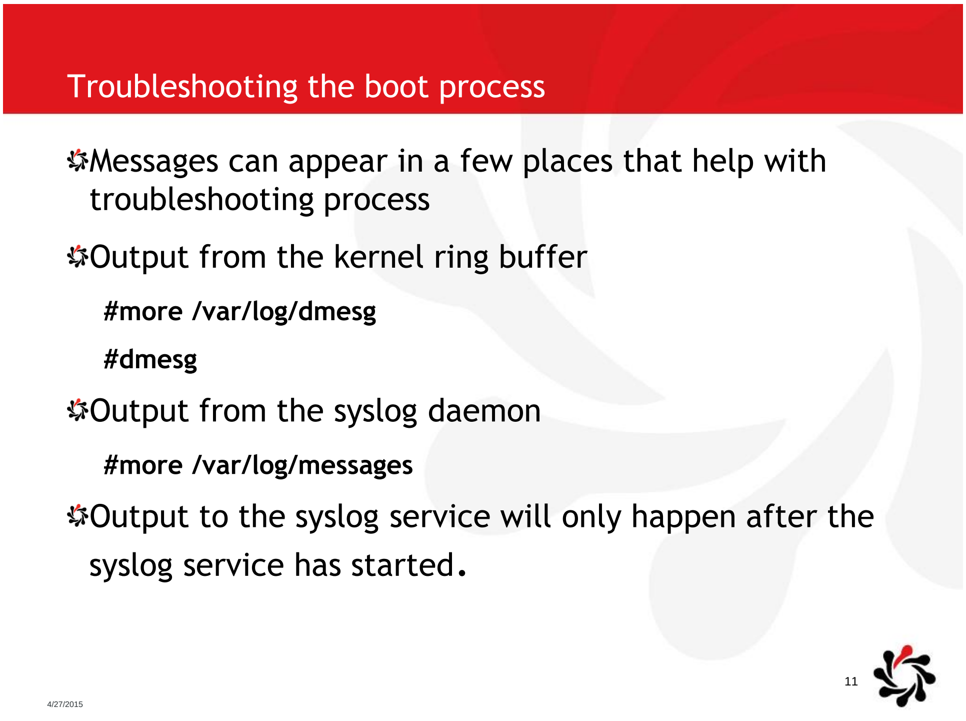
- The daemon scripts are then executed based upon the run level defined in the **/etc/inittab** or passed at boot.

```
l3:3:wait:/etc/rc.d/rc 3
```

- This scripts executes the daemon files located in **/etc/rc.d/rc3.d**
- Once the daemons are up then the ttys are loaded and the user is able to log into the system

4/27/2015

10



## Troubleshooting the boot process

- Messages can appear in a few places that help with troubleshooting process
- Output from the kernel ring buffer

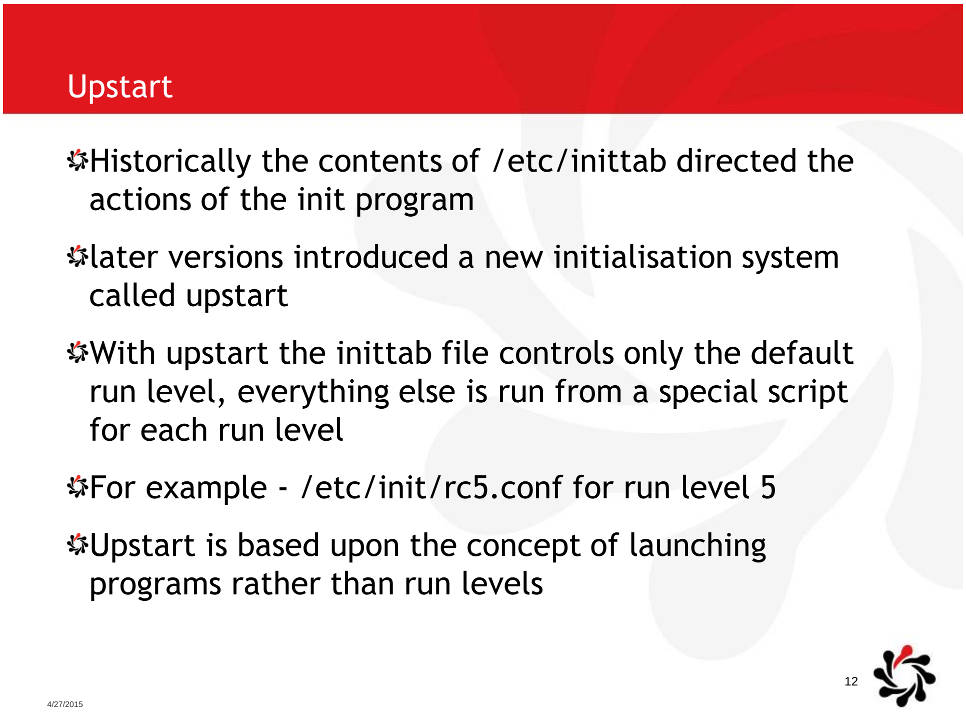
```
#more /var/log/dmesg
#dmesg
```

- Output from the syslog daemon

```
#more /var/log/messages
```

- Output to the syslog service will only happen after the syslog service has started.

4/27/2015 11



## Upstart

- Historically the contents of /etc/inittab directed the actions of the init program
- later versions introduced a new initialisation system called upstart
- With upstart the inittab file controls only the default run level, everything else is run from a special script for each run level
- For example - /etc/init/rc5.conf for run level 5
- Upstart is based upon the concept of launching programs rather than run levels

4/27/2015 12

## systemd

- There is now a new system to replace upstart called systemd. Has pid 1
- Designed to be backwards compatible with SysV init scripts
- Features such as parallel startup of system services at boot time
- On-demand activation of daemons, support
- Dependency-based service control logic.
- Config files are located in /etc/systemd/system
- Services are defined in /lib/systemd/system
- Most services are still defined in /etc/rc.d/init.d

13

4/27/2015

## Recap 101.2 Boot the system

**Weight** 3

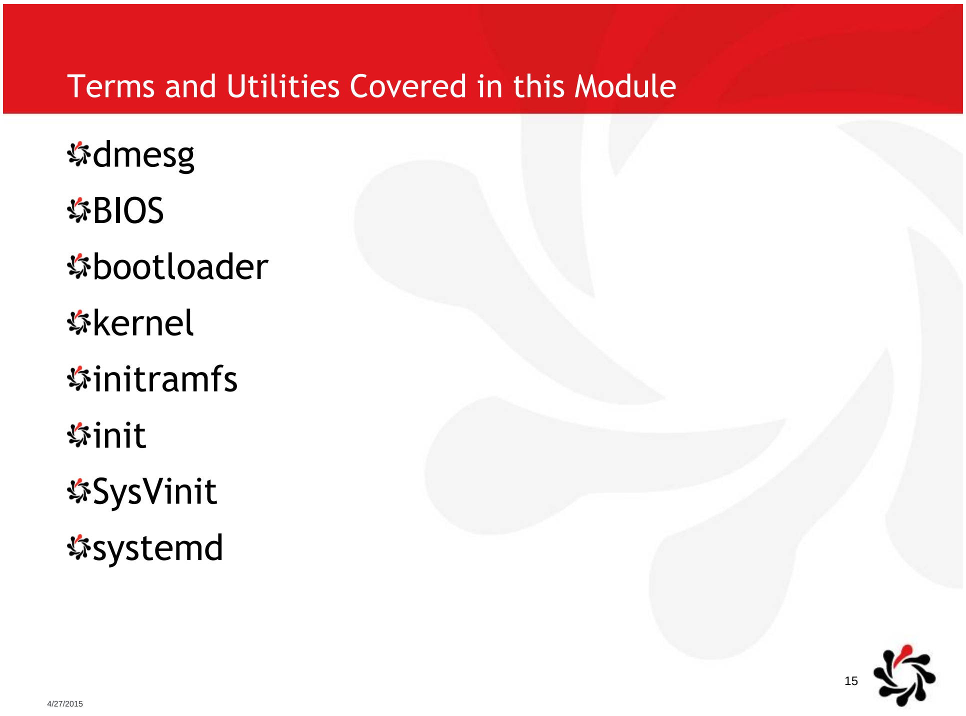
**Description** Candidates should be able to guide the system through the booting process.

**Key Knowledge Areas**

- Provide common commands to the boot loader and options to the kernel at boot time
- Demonstrate knowledge of the boot sequence from BIOS to boot completion
- Understanding of SysVinit and systemd
- Awareness of Upstart
- Check boot events in the log files

14

4/27/2015

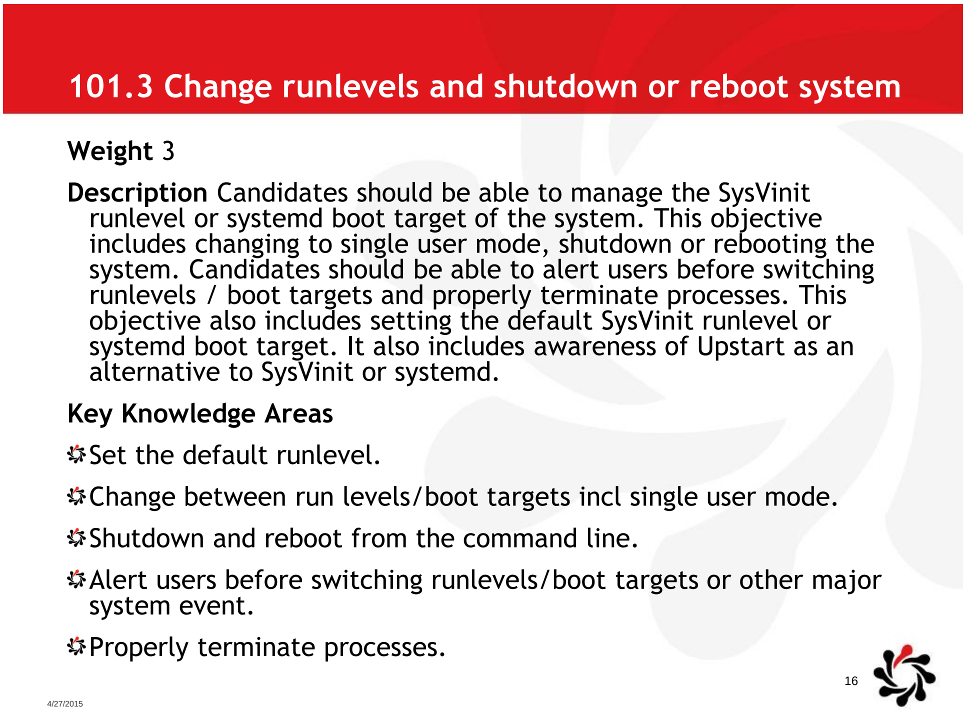


## Terms and Utilities Covered in this Module

- dmesg
- BIOS
- bootloader
- kernel
- initramfs
- init
- SysVinit
- systemd

4/27/2015

15



## 101.3 Change runlevels and shutdown or reboot system

**Weight** 3

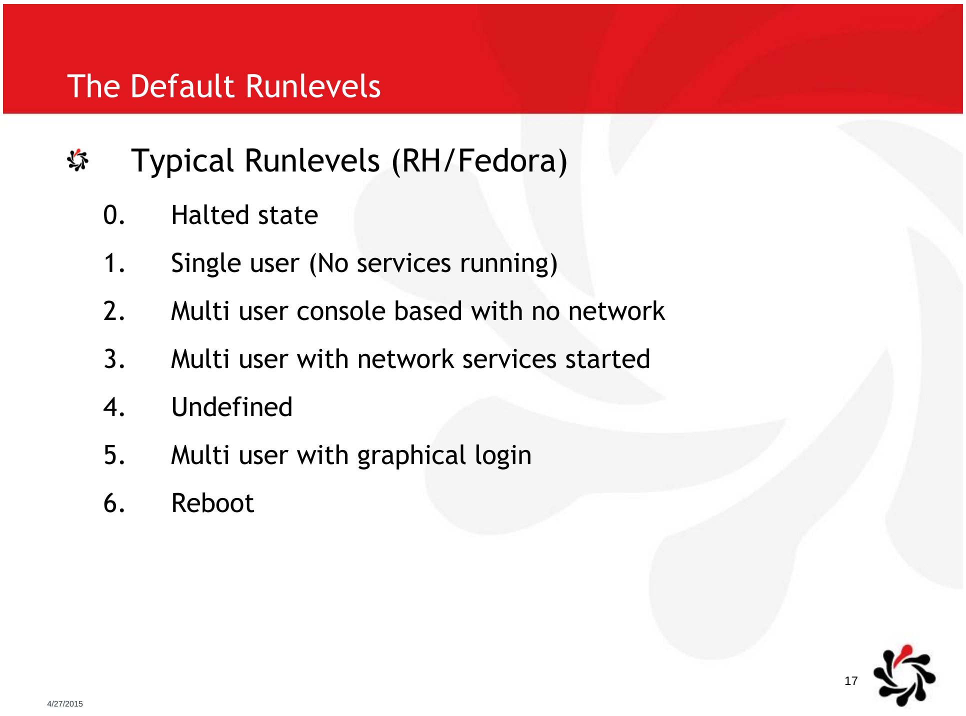
**Description** Candidates should be able to manage the SysVinit runlevel or systemd boot target of the system. This objective includes changing to single user mode, shutdown or rebooting the system. Candidates should be able to alert users before switching runlevels / boot targets and properly terminate processes. This objective also includes setting the default SysVinit runlevel or systemd boot target. It also includes awareness of Upstart as an alternative to SysVinit or systemd.

**Key Knowledge Areas**

- Set the default runlevel.
- Change between run levels/boot targets incl single user mode.
- Shutdown and reboot from the command line.
- Alert users before switching runlevels/boot targets or other major system event.
- Properly terminate processes.

4/27/2015

16



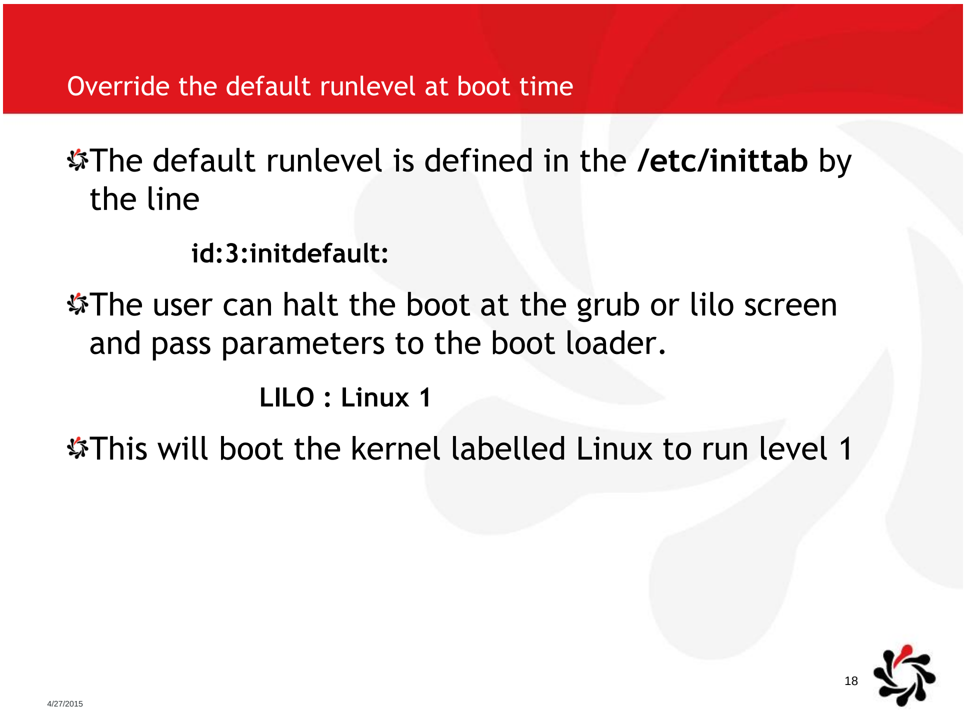
## The Default Runlevels

- Typical Runlevels (RH/Fedora)

0. Halted state
1. Single user (No services running)
2. Multi user console based with no network
3. Multi user with network services started
4. Undefined
5. Multi user with graphical login
6. Reboot

4/27/2015

17



## Override the default runlevel at boot time

- The default runlevel is defined in the **/etc/inittab** by the line

**id:3:initdefault:**

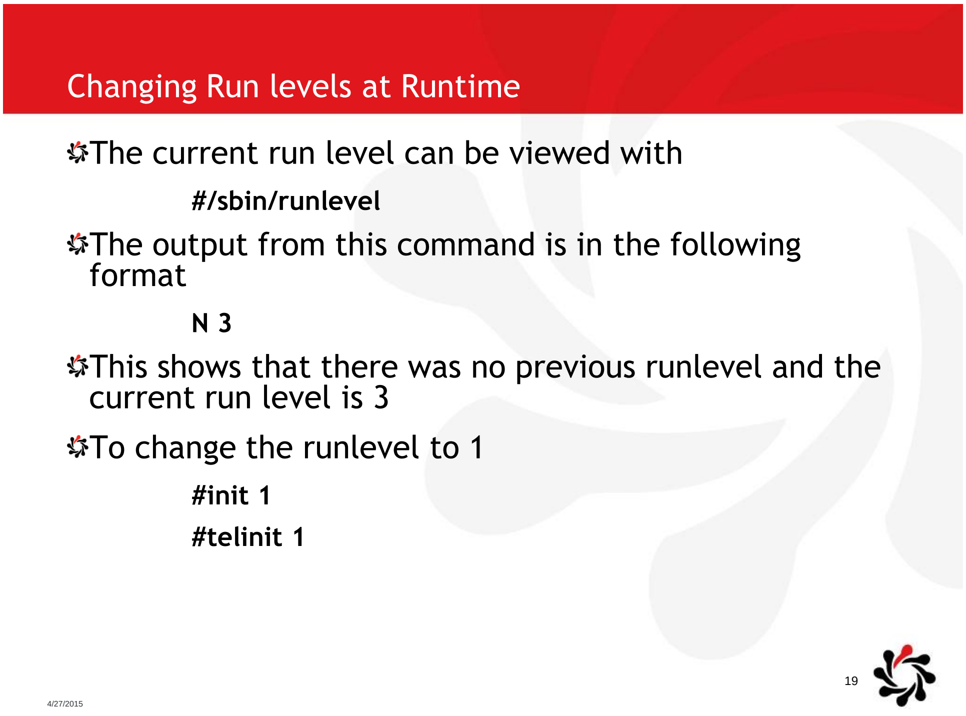
- The user can halt the boot at the grub or lilo screen and pass parameters to the boot loader.

**LILO : Linux 1**

- This will boot the kernel labelled Linux to run level 1

4/27/2015

18



## Changing Run levels at Runtime

- The current run level can be viewed with

```
#/sbin/runlevel
```

- The output from this command is in the following format

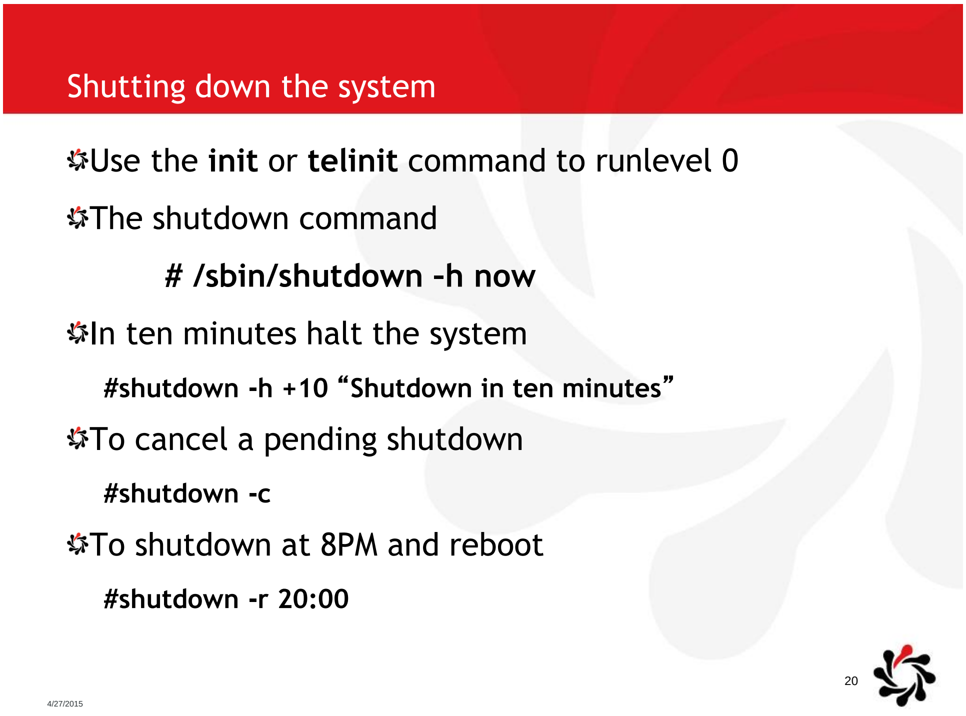
```
N 3
```

- This shows that there was no previous runlevel and the current run level is 3
- To change the runlevel to 1

```
#init 1
#telinit 1
```

19

4/27/2015



## Shutting down the system

- Use the **init** or **telinit** command to runlevel 0
- The shutdown command

```
# /sbin/shutdown -h now
```

- In ten minutes halt the system

```
#shutdown -h +10 “Shutdown in ten minutes”
```

- To cancel a pending shutdown

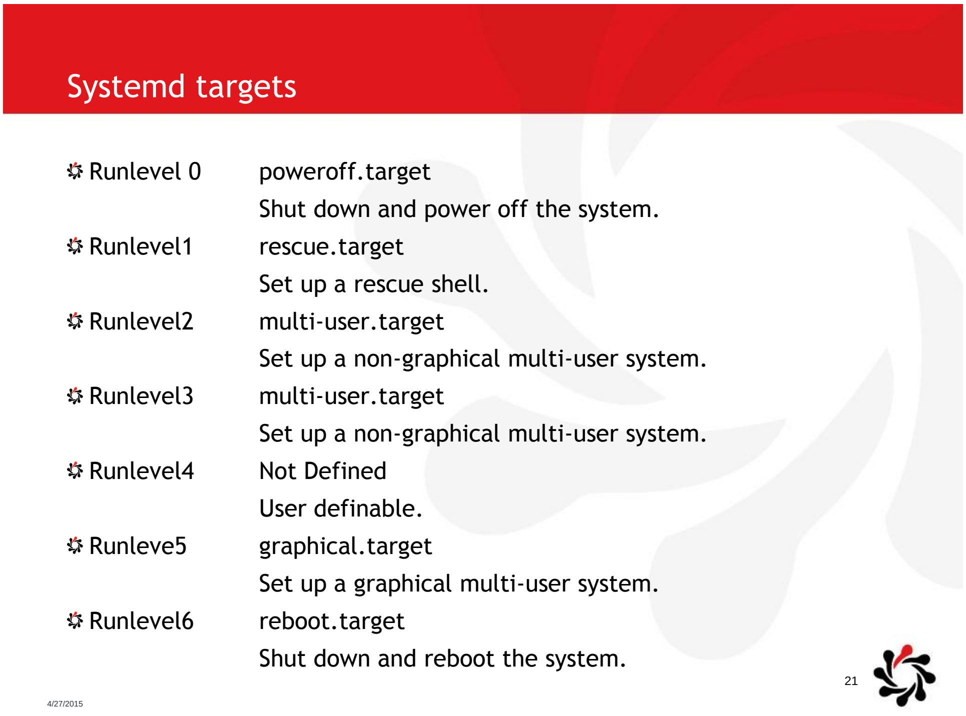
```
#shutdown -c
```

- To shutdown at 8PM and reboot

```
#shutdown -r 20:00
```

20

4/27/2015

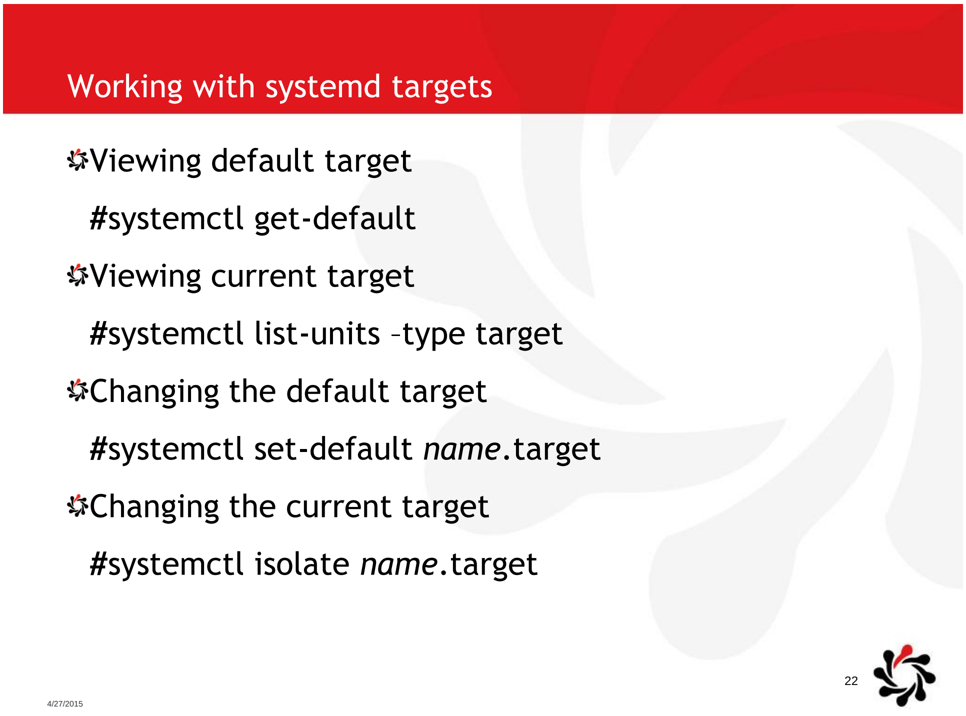


## Systemd targets

| | | |
|---|---|---|
| Runlevel 0 | poweroff.target | Shut down and power off the system. |
| Runlevel1 | rescue.target | Set up a rescue shell. |
| Runlevel2 | multi-user.target | Set up a non-graphical multi-user system. |
| Runlevel3 | multi-user.target | Set up a non-graphical multi-user system. |
| Runlevel4 | Not Defined | User definable. |
| Runleve5 | graphical.target | Set up a graphical multi-user system. |
| Runlevel6 | reboot.target | Shut down and reboot the system. |

21

4/27/2015



## Working with systemd targets

- Viewing default target

```
#systemctl get-default
```

- Viewing current target

```
#systemctl list-units –type target
```

- Changing the default target

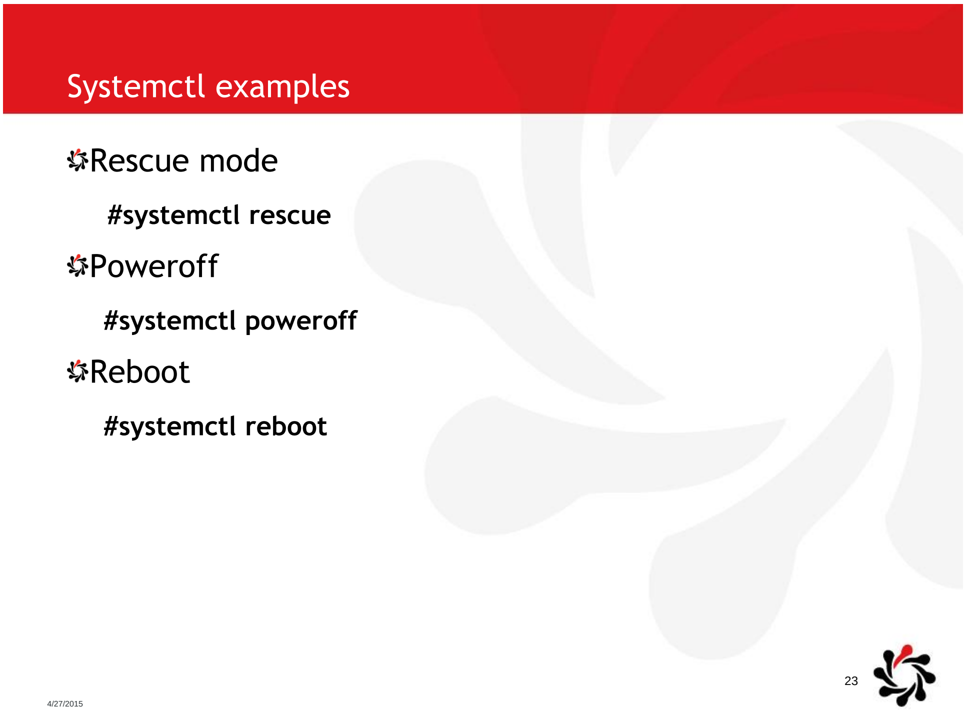
```
#systemctl set-default name.target
```

- Changing the current target

```
#systemctl isolate name.target
```

22

4/27/2015



## Systemctl examples

- Rescue mode

```
#systemctl rescue
```

- Poweroff

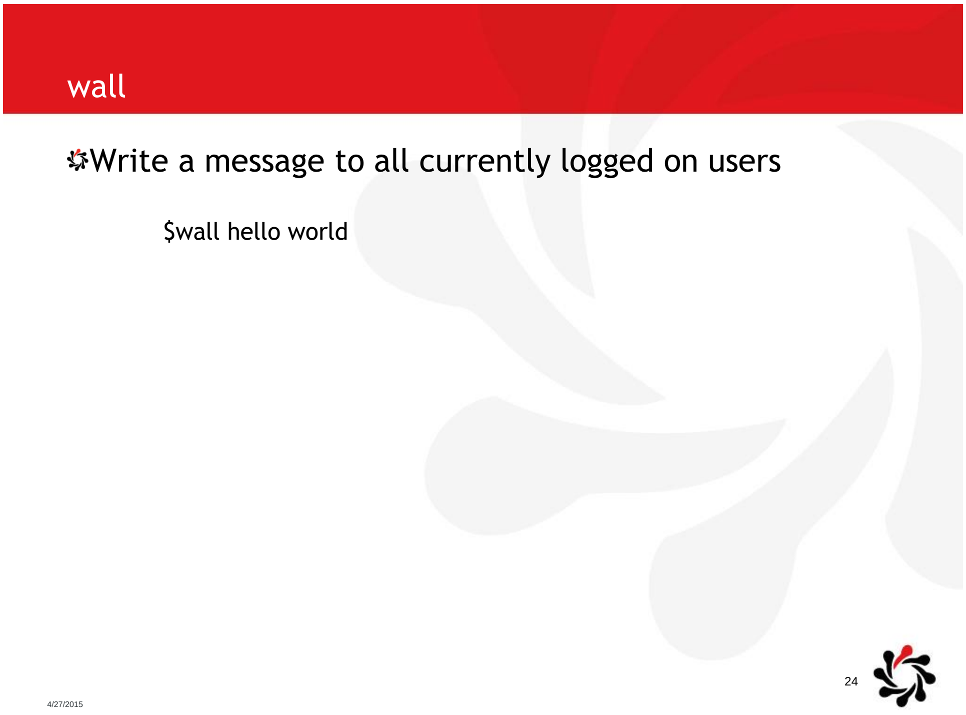
```
#systemctl poweroff
```

- Reboot

```
#systemctl reboot
```

23

4/27/2015



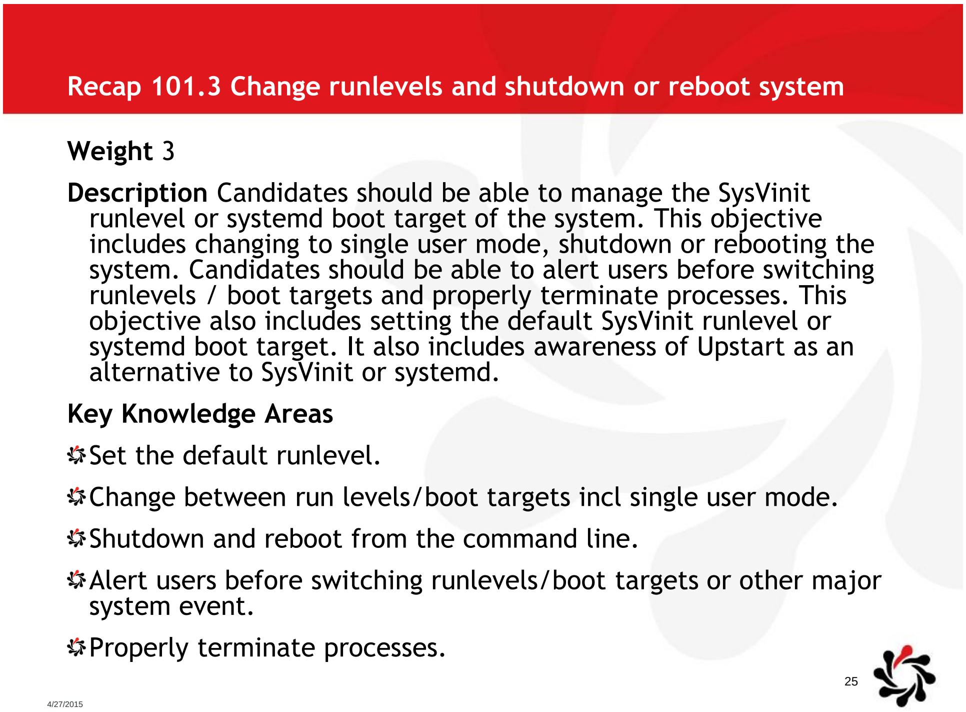
## wall

- Write a message to all currently logged on users

```
$wall hello world
```

24

4/27/2015



## Recap 101.3 Change runlevels and shutdown or reboot system

**Weight** 3

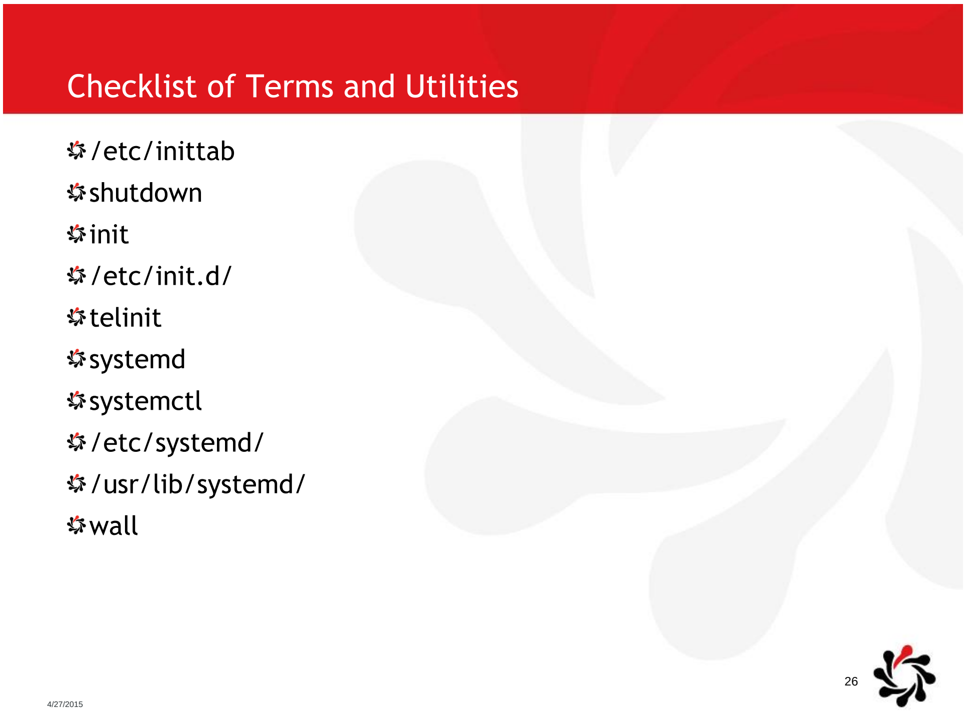
**Description** Candidates should be able to manage the SysVinit runlevel or systemd boot target of the system. This objective includes changing to single user mode, shutdown or rebooting the system. Candidates should be able to alert users before switching runlevels / boot targets and properly terminate processes. This objective also includes setting the default SysVinit runlevel or systemd boot target. It also includes awareness of Upstart as an alternative to SysVinit or systemd.

**Key Knowledge Areas**

- Set the default runlevel.
- Change between run levels/boot targets incl single user mode.
- Shutdown and reboot from the command line.
- Alert users before switching runlevels/boot targets or other major system event.
- Properly terminate processes.

25

4/27/2015

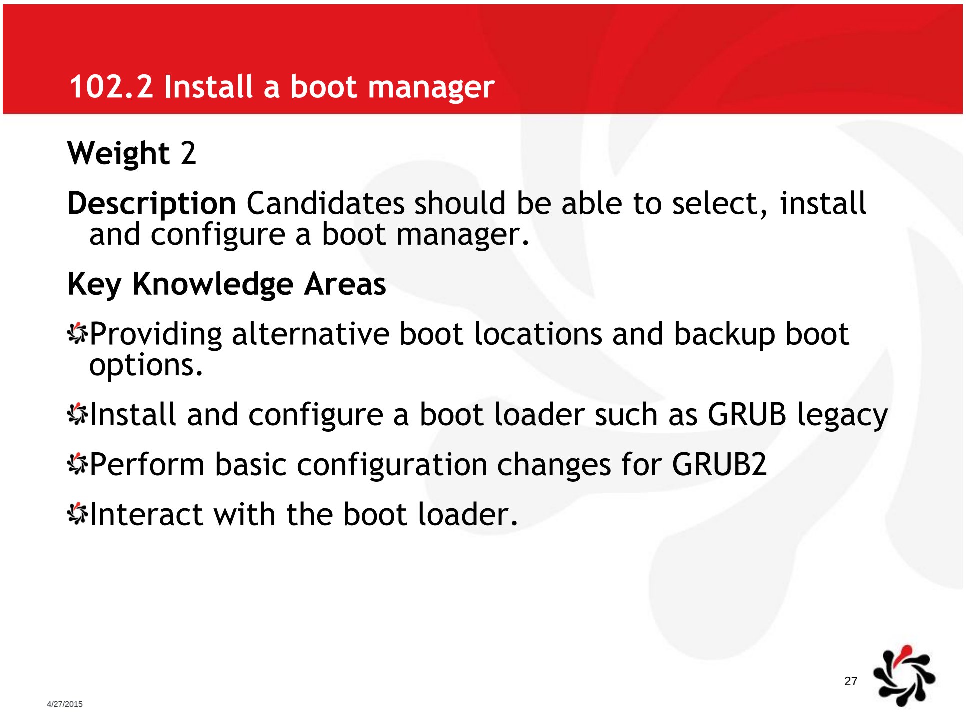


## Checklist of Terms and Utilities

- /etc/inittab
- shutdown
- init
- /etc/init.d/
- telinit
- systemd
- systemctl
- /etc/systemd/
- /usr/lib/systemd/
- wall

26

4/27/2015



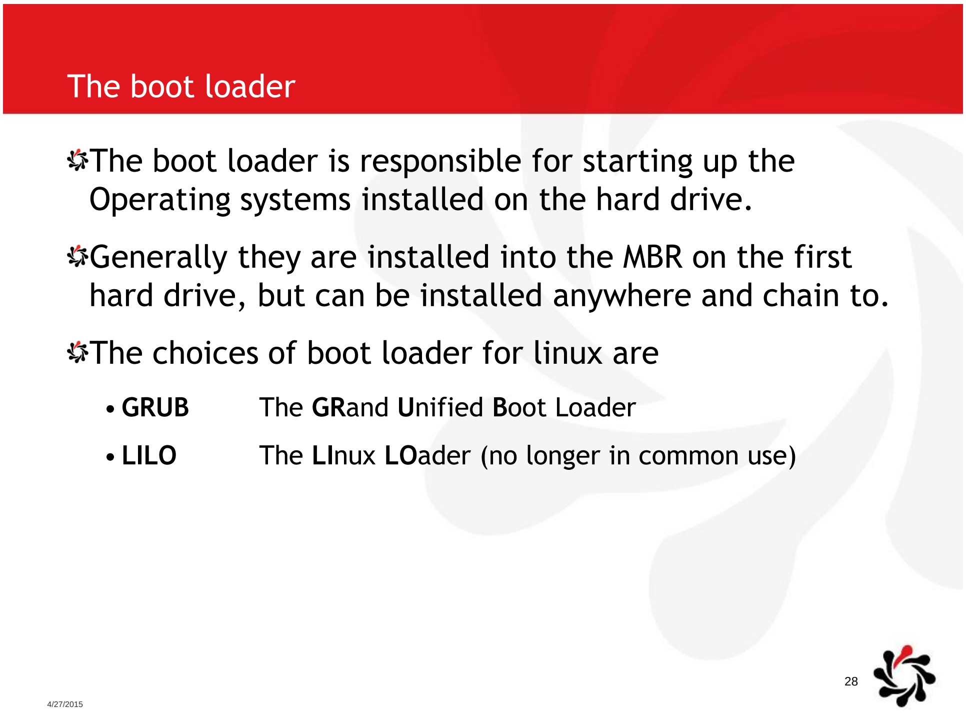
## 102.2 Install a boot manager

**Weight** 2

**Description** Candidates should be able to select, install and configure a boot manager.

**Key Knowledge Areas**

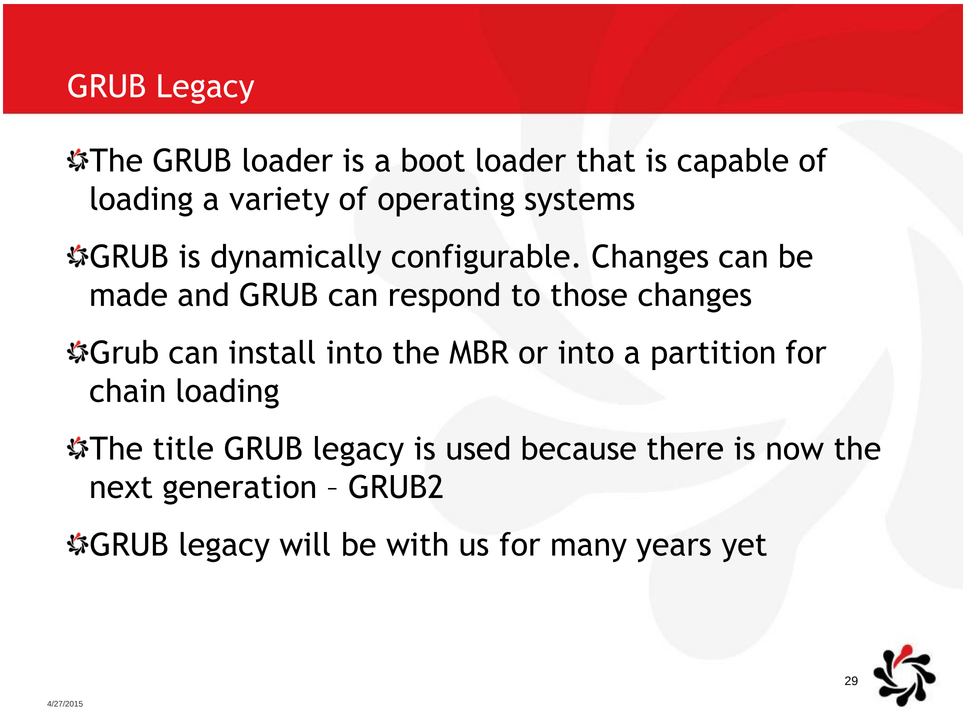
- Providing alternative boot locations and backup boot options.
- Install and configure a boot loader such as GRUB legacy
- Perform basic configuration changes for GRUB2
- Interact with the boot loader.



## The boot loader

- The boot loader is responsible for starting up the Operating systems installed on the hard drive.
- Generally they are installed into the MBR on the first hard drive, but can be installed anywhere and chain to.
- The choices of boot loader for linux are

| | |
|---|---|
| • **GRUB** | The **GR**and **U**nified **B**oot Loader |
| • **LILO** | The **LI**nux **LO**ader (no longer in common use) |

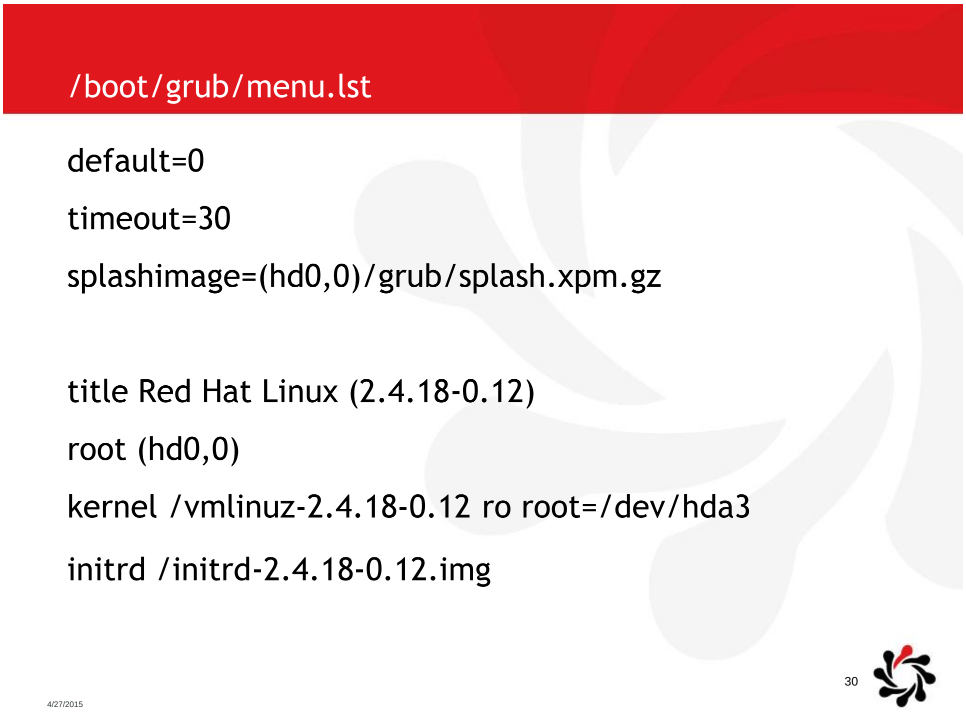


## GRUB Legacy

- The GRUB loader is a boot loader that is capable of loading a variety of operating systems
- GRUB is dynamically configurable. Changes can be made and GRUB can respond to those changes
- Grub can install into the MBR or into a partition for chain loading
- The title GRUB legacy is used because there is now the next generation – GRUB2
- GRUB legacy will be with us for many years yet

4/27/2015

29



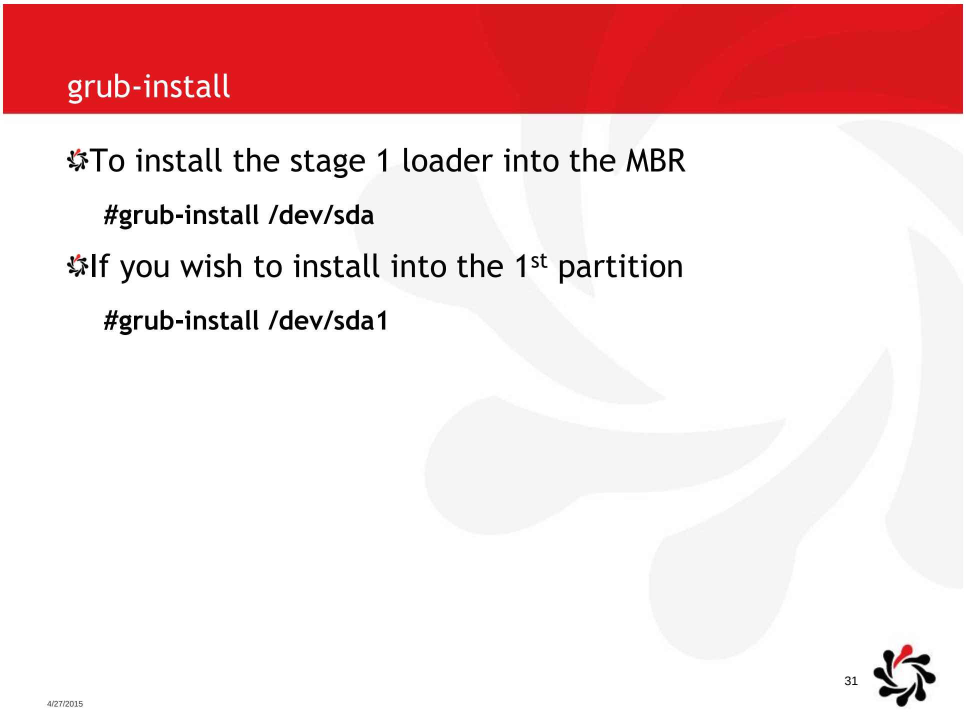
## /boot/grub/menu.lst

```
default=0
timeout=30
splashimage=(hd0,0)/grub/splash.xpm.gz

title Red Hat Linux (2.4.18-0.12)
root (hd0,0)
kernel /vmlinuz-2.4.18-0.12 ro root=/dev/hda3
initrd /initrd-2.4.18-0.12.img
```

4/27/2015

30



## grub-install

- To install the stage 1 loader into the MBR

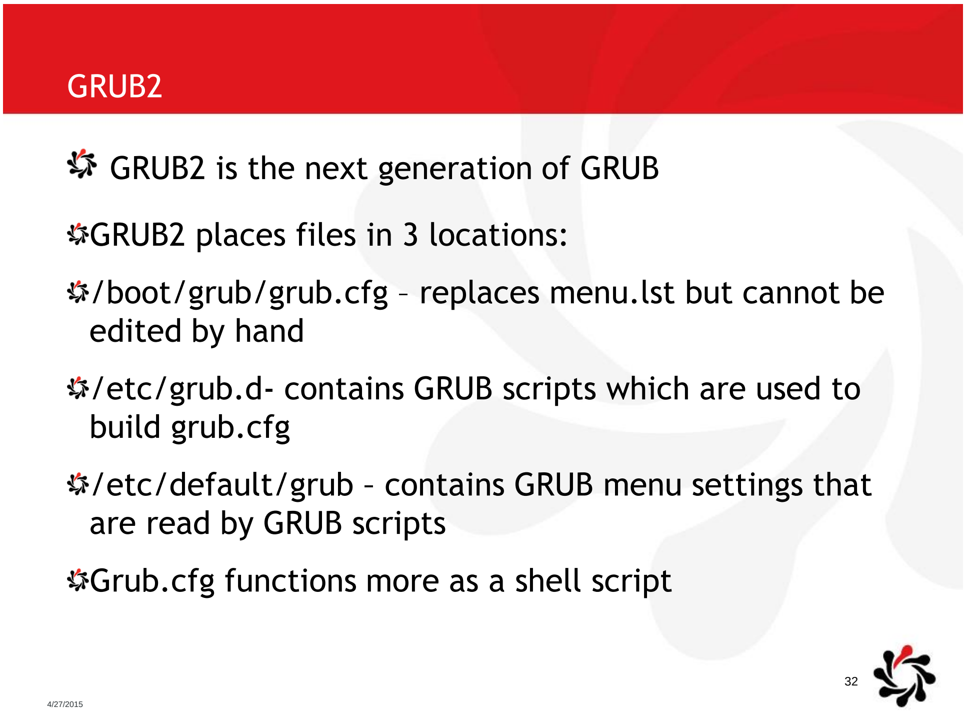
```
#grub-install /dev/sda
```

- If you wish to install into the 1st partition

```
#grub-install /dev/sda1
```

4/27/2015

31

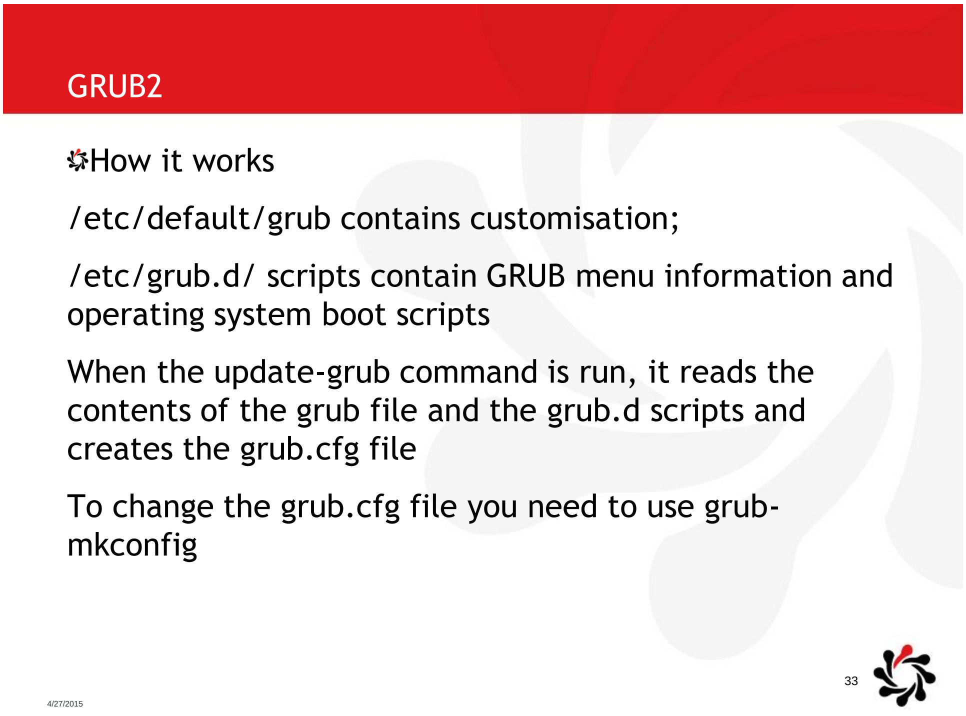


## GRUB2

- GRUB2 is the next generation of GRUB
- GRUB2 places files in 3 locations:
- /boot/grub/grub.cfg – replaces menu.lst but cannot be edited by hand
- /etc/grub.d- contains GRUB scripts which are used to build grub.cfg
- /etc/default/grub – contains GRUB menu settings that are read by GRUB scripts
- Grub.cfg functions more as a shell script

4/27/2015

32



## GRUB2

- How it works

/etc/default/grub contains customisation;

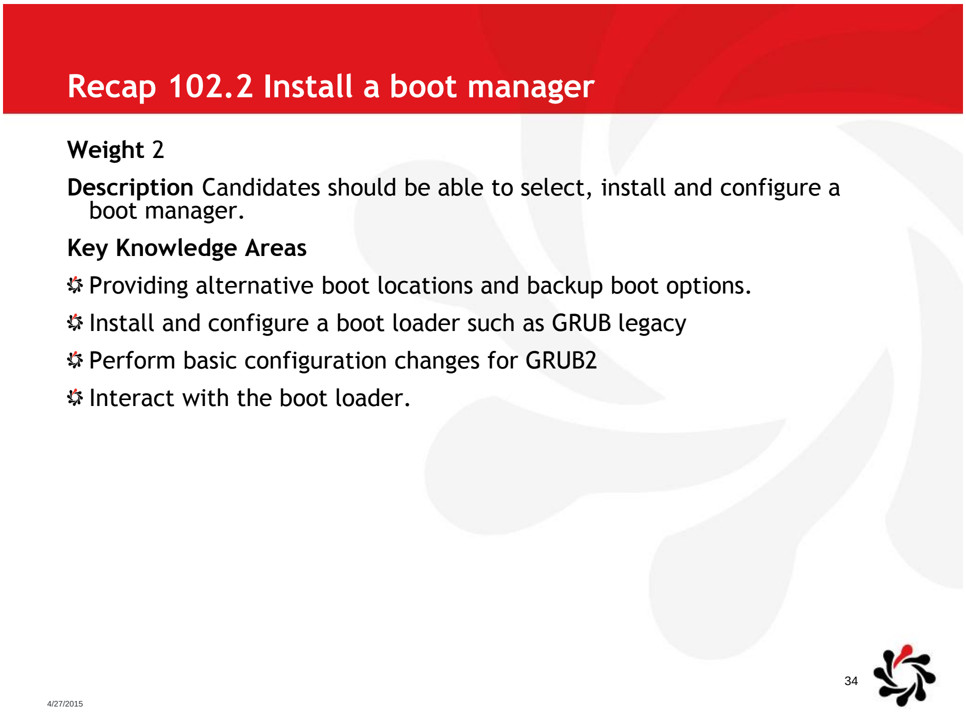
/etc/grub.d/ scripts contain GRUB menu information and operating system boot scripts

When the update-grub command is run, it reads the contents of the grub file and the grub.d scripts and creates the grub.cfg file

To change the grub.cfg file you need to use grub-mkconfig

33

4/27/2015



## Recap 102.2 Install a boot manager

**Weight** 2

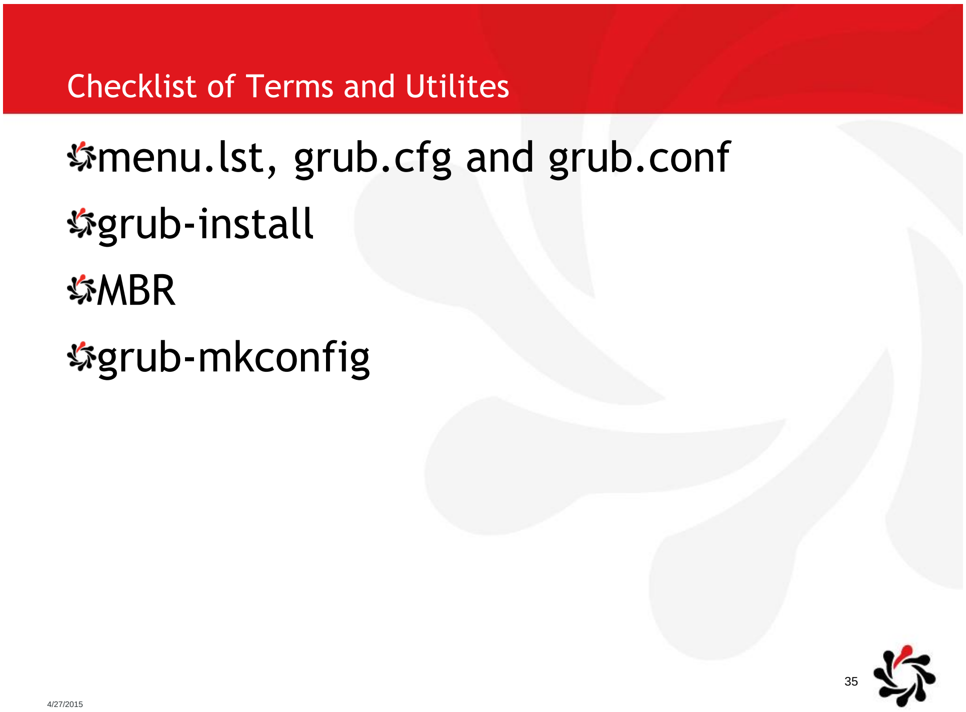
**Description** Candidates should be able to select, install and configure a boot manager.

**Key Knowledge Areas**

- Providing alternative boot locations and backup boot options.
- Install and configure a boot loader such as GRUB legacy
- Perform basic configuration changes for GRUB2
- Interact with the boot loader.

34

4/27/2015

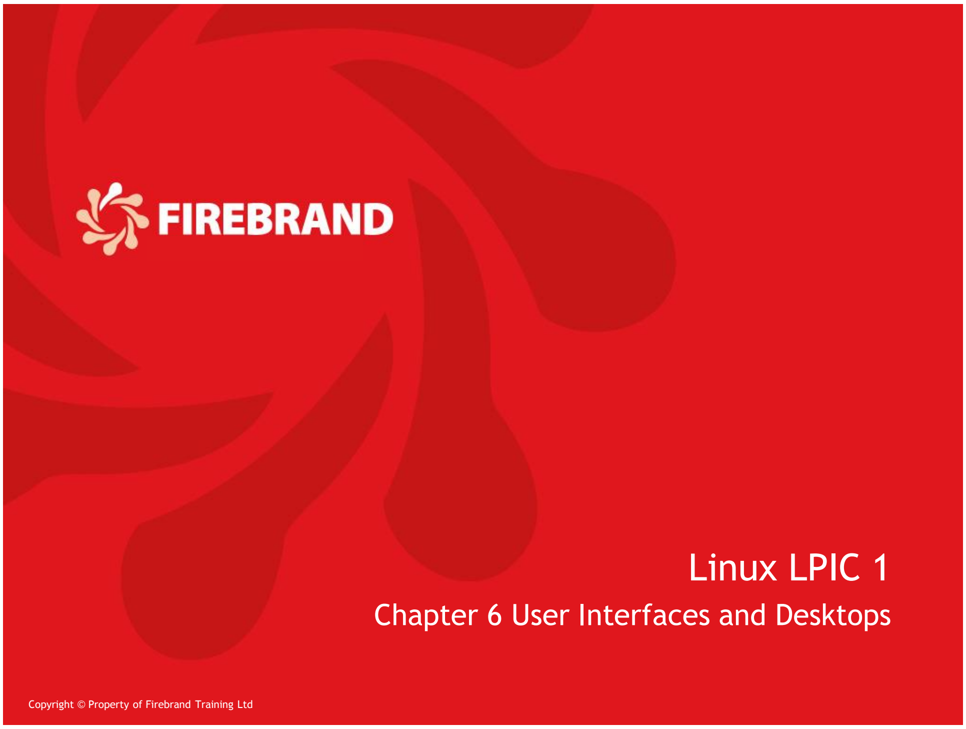


## Checklist of Terms and Utilites

- menu.lst, grub.cfg and grub.conf
- grub-install
- MBR
- grub-mkconfig

35

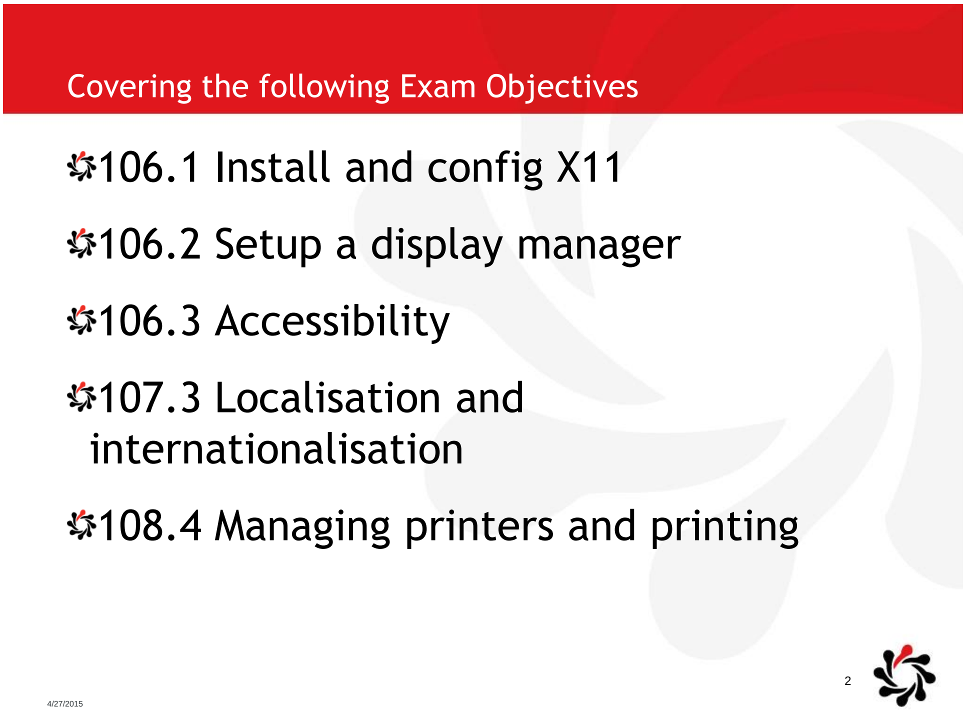
4/27/2015



FIREBRAND

# Linux LPIC 1

## Chapter 6 User Interfaces and Desktops

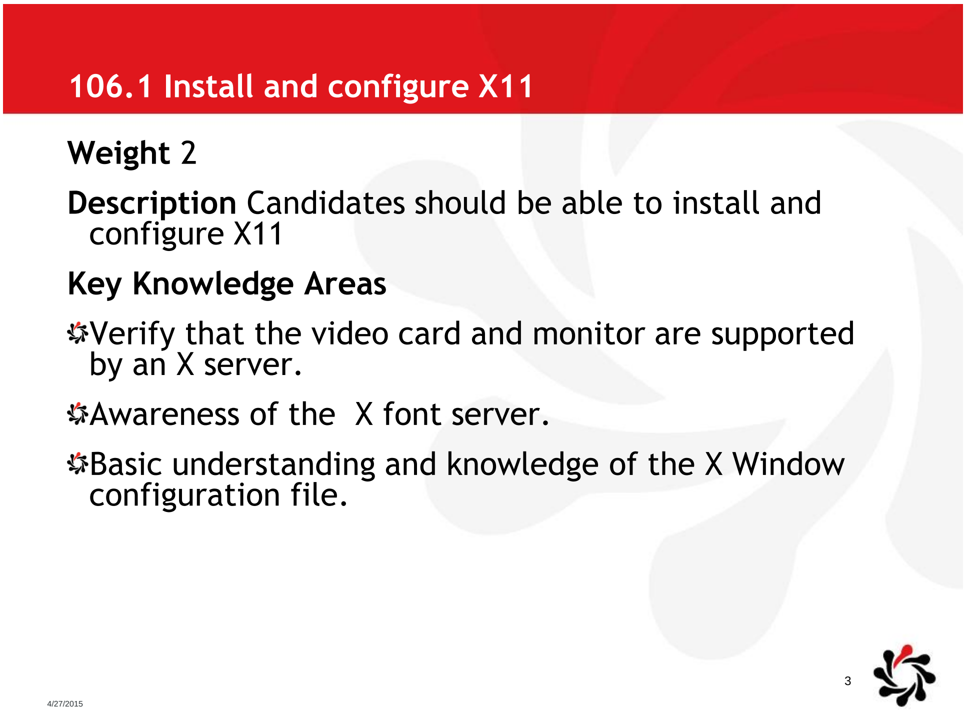


## Covering the following Exam Objectives

- 106.1 Install and config X11
- 106.2 Setup a display manager
- 106.3 Accessibility
- 107.3 Localisation and internationalisation
- 108.4 Managing printers and printing

4/27/2015

2



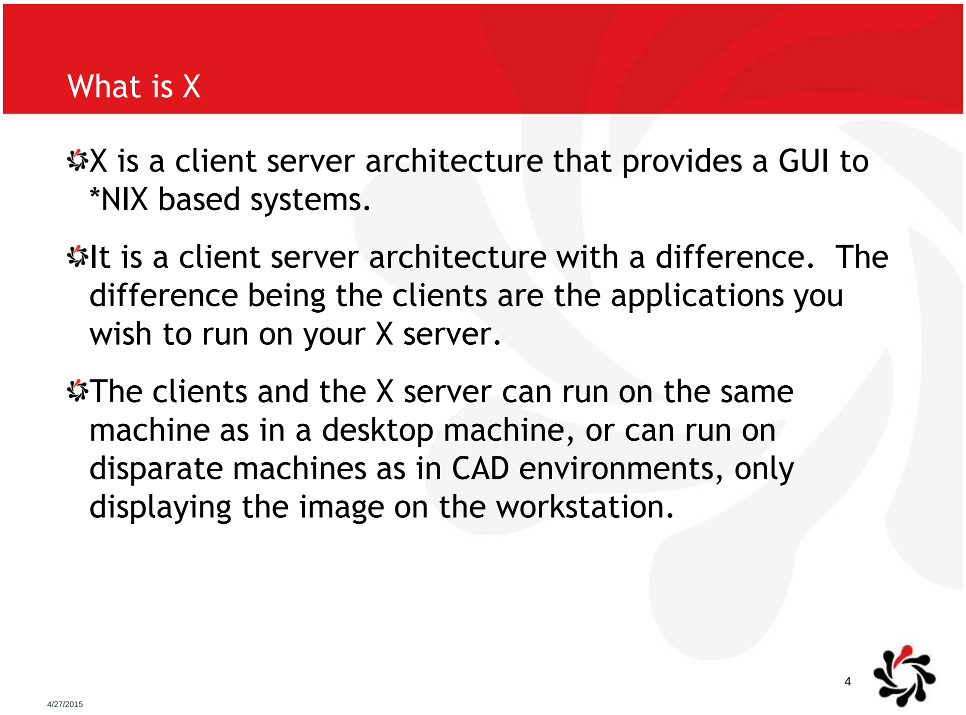
## 106.1 Install and configure X11

**Weight** 2

**Description** Candidates should be able to install and configure X11

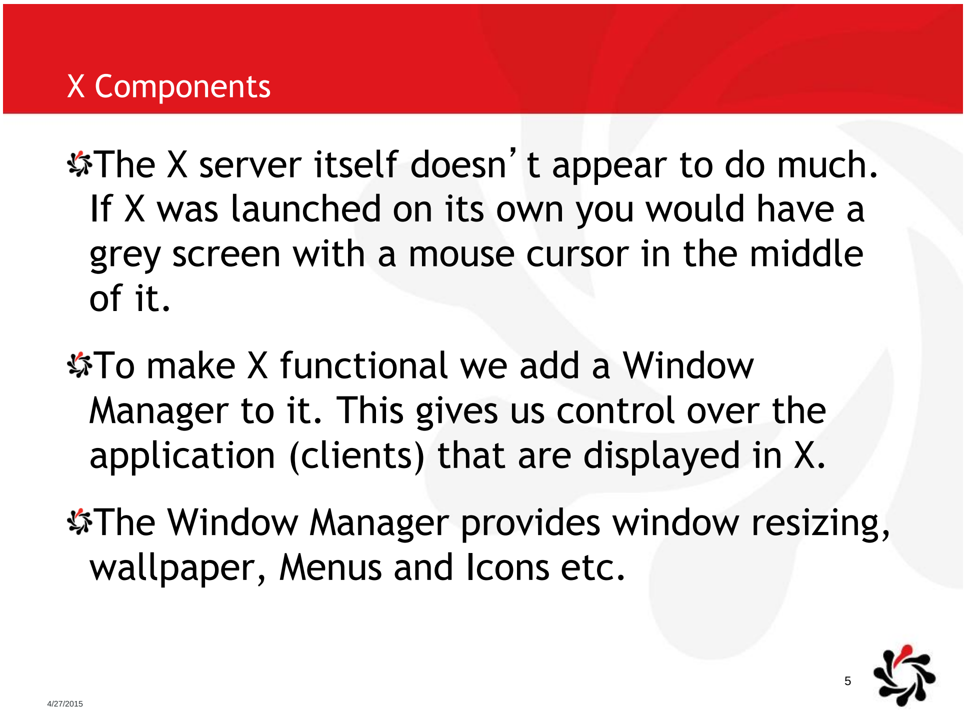
**Key Knowledge Areas**

- Verify that the video card and monitor are supported by an X server.
- Awareness of the  X font server.
- Basic understanding and knowledge of the X Window configuration file.



## What is X

- X is a client server architecture that provides a GUI to *NIX based systems.
- It is a client server architecture with a difference.  The difference being the clients are the applications you wish to run on your X server.
- The clients and the X server can run on the same machine as in a desktop machine, or can run on disparate machines as in CAD environments, only displaying the image on the workstation.



## X Components

- The X server itself doesn’t appear to do much. If X was launched on its own you would have a grey screen with a mouse cursor in the middle of it.
- To make X functional we add a Window Manager to it. This gives us control over the application (clients) that are displayed in X.
- The Window Manager provides window resizing, wallpaper, Menus and Icons etc.

5

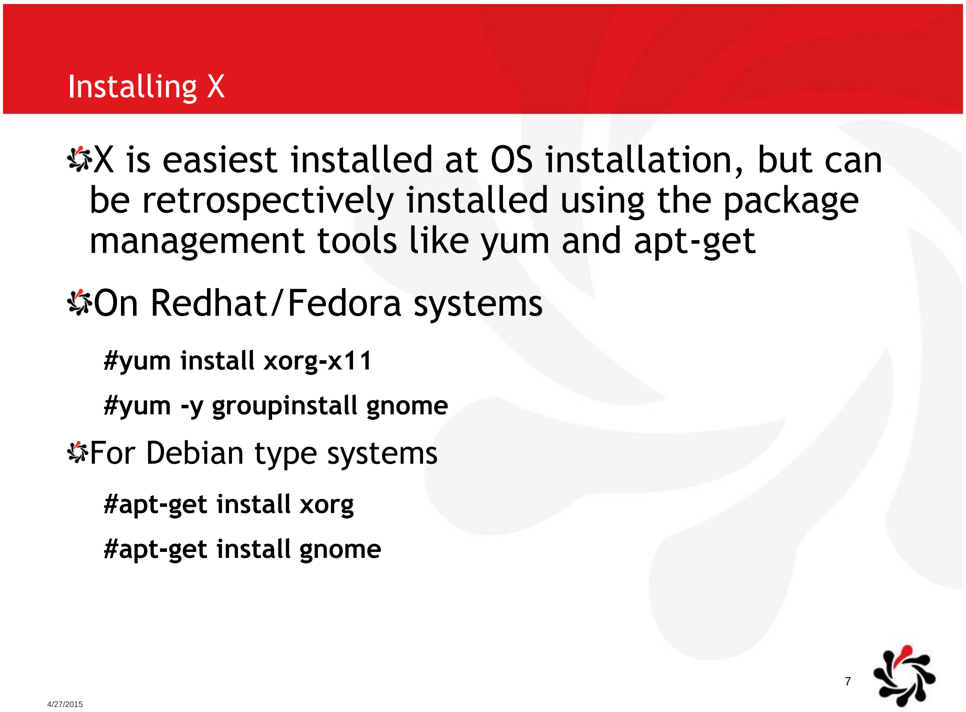
4/27/2015

## The Desktop Environment

- The Desktop Environment is a collection of items to provide a common look and feel to your desktop.
- It would include a Window Manager and also common interface tools like File Manager, Control Panel etc. Two common Desktop environments are
  - KDE
  - GNOME

6

4/27/2015



## Installing X

- X is easiest installed at OS installation, but can be retrospectively installed using the package management tools like yum and apt-get
- On Redhat/Fedora systems

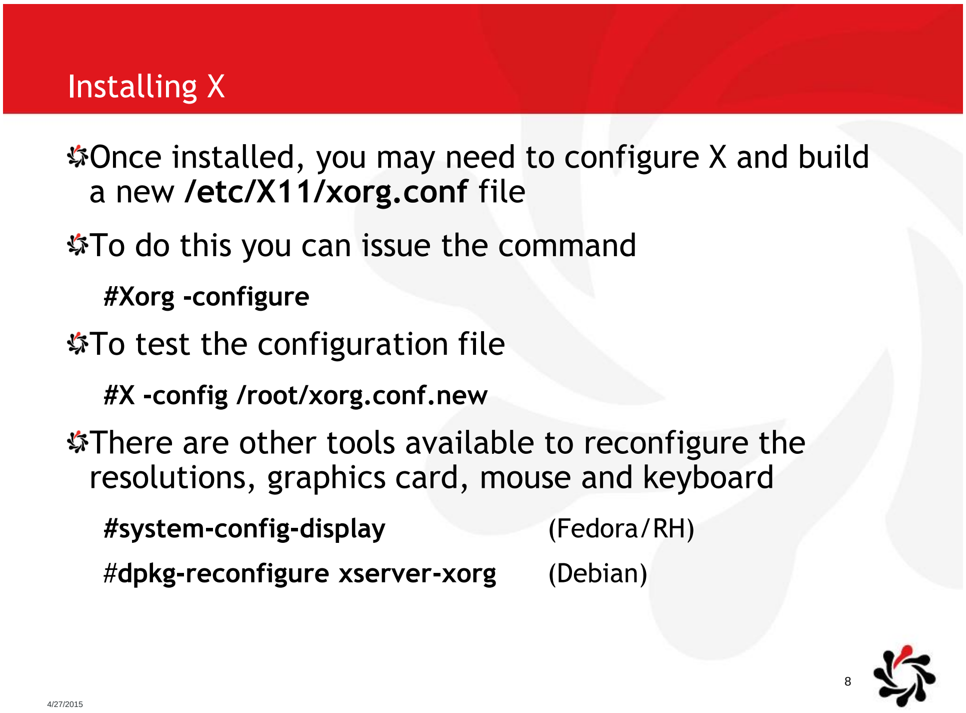
```
#yum install xorg-x11
#yum -y groupinstall gnome
```

- For Debian type systems

```
#apt-get install xorg
#apt-get install gnome
```

4/27/2015

7



## Installing X

- Once installed, you may need to configure X and build a new **/etc/X11/xorg.conf** file
- To do this you can issue the command

```
#Xorg -configure
```

- To test the configuration file

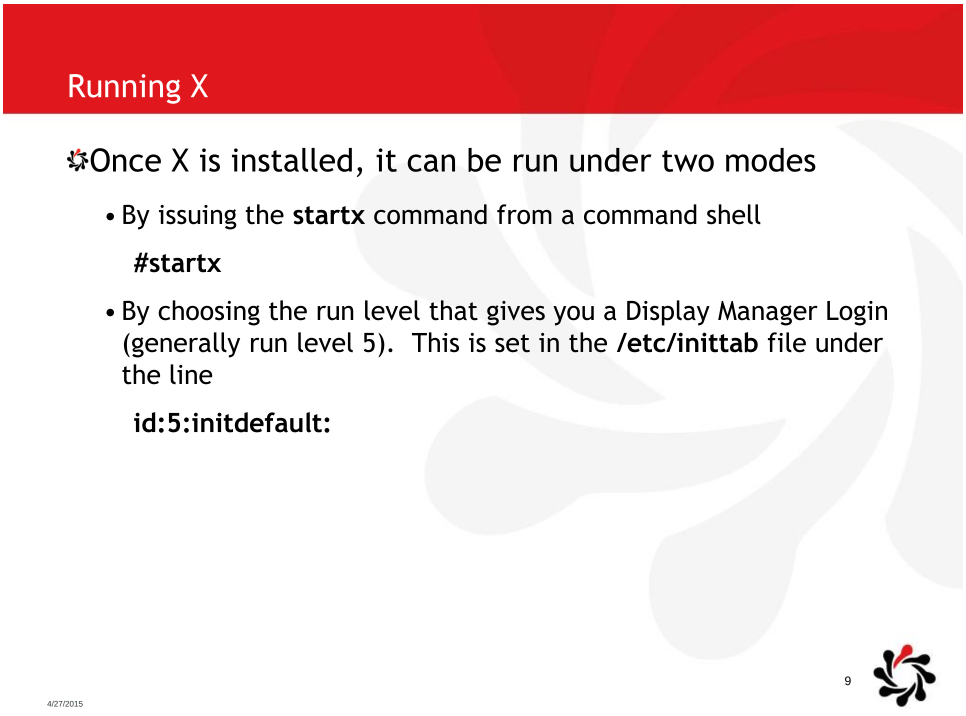
```
#X -config /root/xorg.conf.new
```

- There are other tools available to reconfigure the resolutions, graphics card, mouse and keyboard

```
#system-config-display          (Fedora/RH)
#dpkg-reconfigure xserver-xorg  (Debian)
```

4/27/2015

8



## Running X

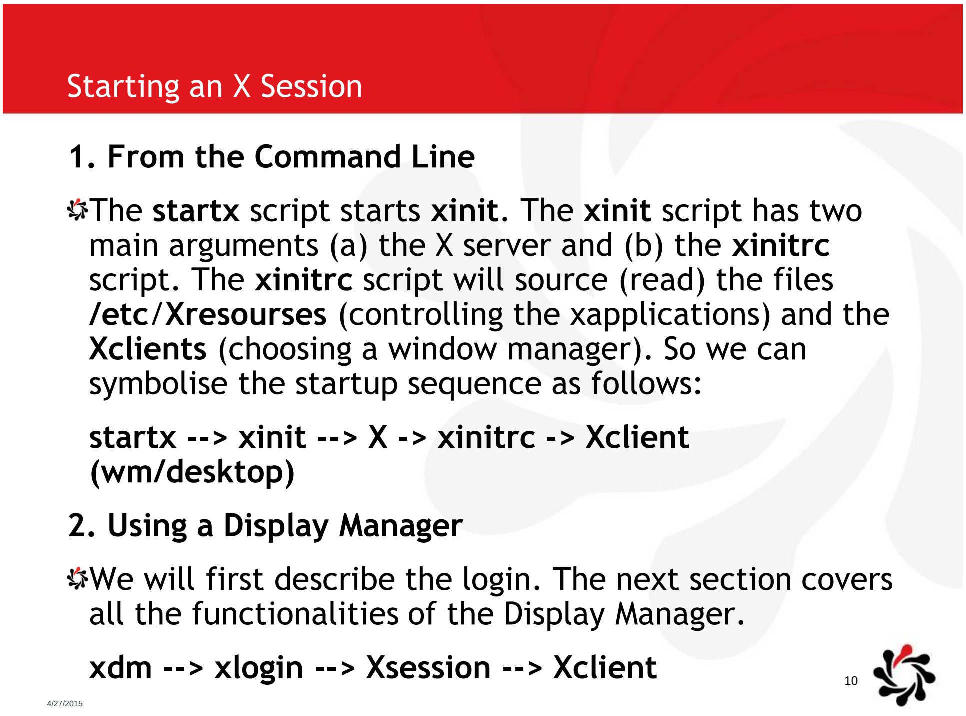
- Once X is installed, it can be run under two modes
  - By issuing the **startx** command from a command shell

    **#startx**
  - By choosing the run level that gives you a Display Manager Login (generally run level 5). This is set in the **/etc/inittab** file under the line

    **id:5:initdefault:**

4/27/2015

9



## Starting an X Session

### 1. From the Command Line

- The **startx** script starts **xinit**. The **xinit** script has two main arguments (a) the X server and (b) the **xinitrc** script. The **xinitrc** script will source (read) the files **/etc/Xresourses** (controlling the xapplications) and the **Xclients** (choosing a window manager). So we can symbolise the startup sequence as follows:

  **startx --> xinit --> X -> xinitrc -> Xclient (wm/desktop)**

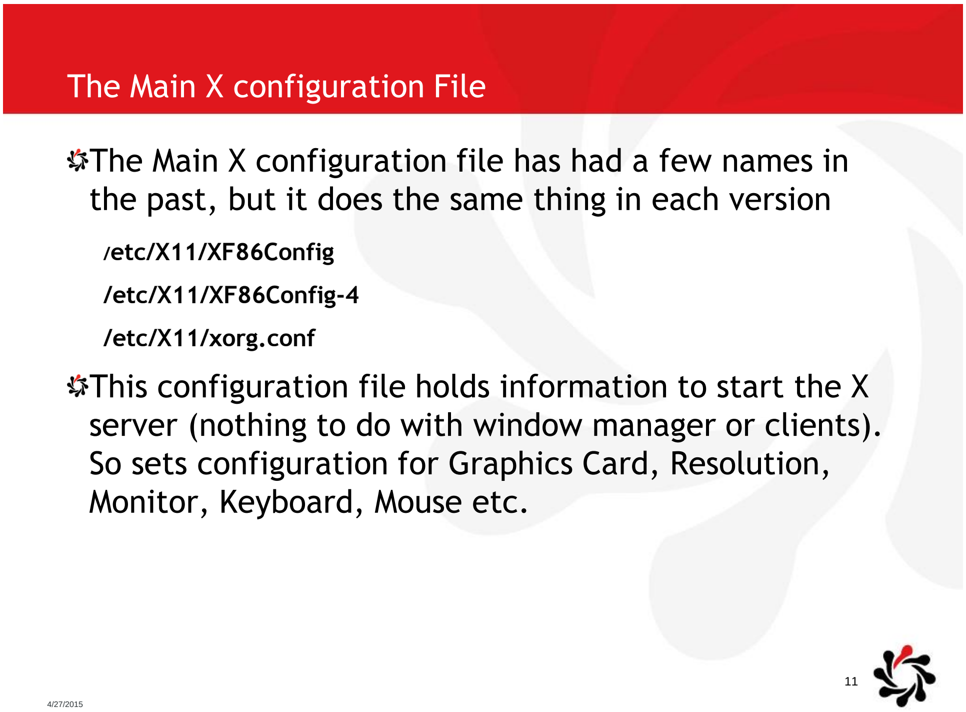
### 2. Using a Display Manager

- We will first describe the login. The next section covers all the functionalities of the Display Manager.

  **xdm --> xlogin --> Xsession --> Xclient**

4/27/2015

10



## The Main X configuration File

- The Main X configuration file has had a few names in the past, but it does the same thing in each version

  **/etc/X11/XF86Config**

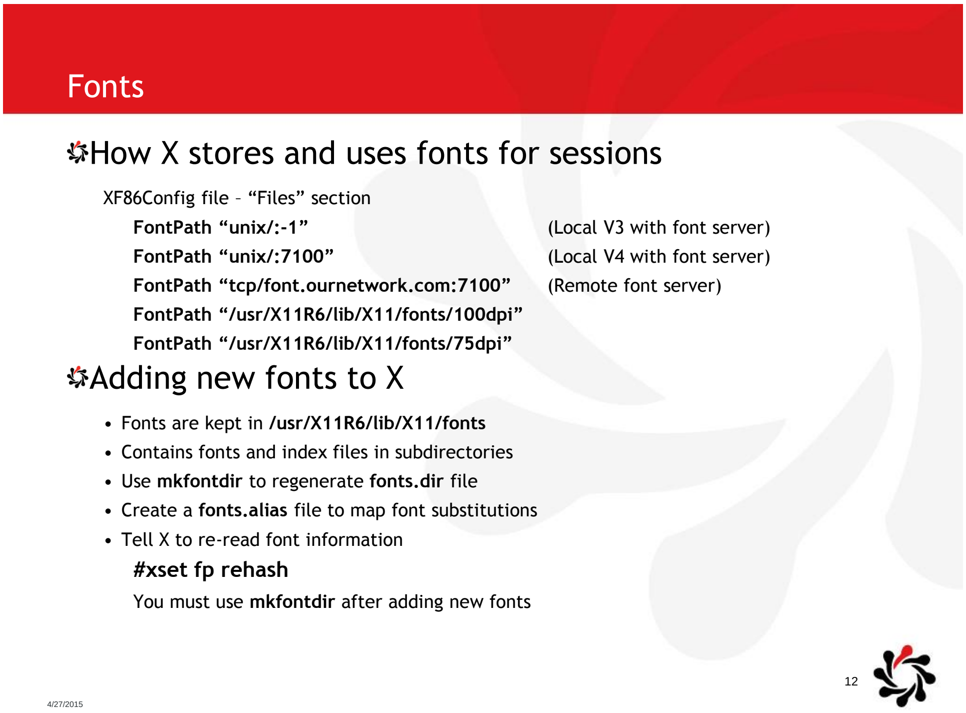
  **/etc/X11/XF86Config-4**

  **/etc/X11/xorg.conf**

- This configuration file holds information to start the X server (nothing to do with window manager or clients). So sets configuration for Graphics Card, Resolution, Monitor, Keyboard, Mouse etc.

4/27/2015

11



## Fonts

- How X stores and uses fonts for sessions

  XF86Config file – “Files” section

  | | |
  |---|---|
  | **FontPath “unix/:-1”** | (Local V3 with font server) |
  | **FontPath “unix/:7100”** | (Local V4 with font server) |
  | **FontPath “tcp/font.ournetwork.com:7100”** | (Remote font server) |
  | **FontPath “/usr/X11R6/lib/X11/fonts/100dpi”** | |
  | **FontPath “/usr/X11R6/lib/X11/fonts/75dpi”** | |

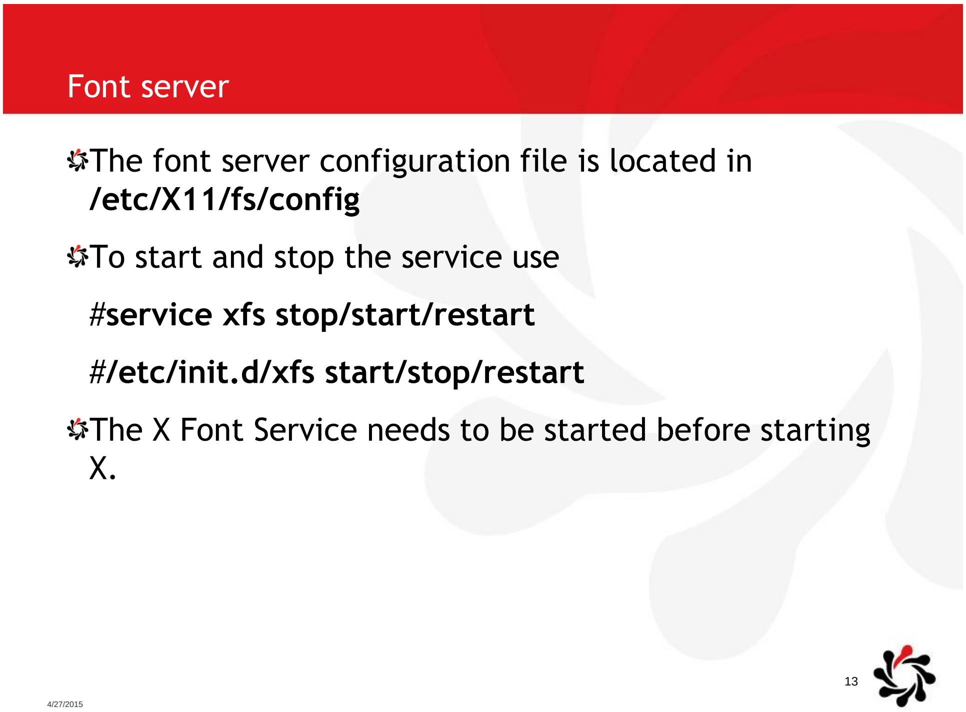
- Adding new fonts to X
  - Fonts are kept in **/usr/X11R6/lib/X11/fonts**
  - Contains fonts and index files in subdirectories
  - Use **mkfontdir** to regenerate **fonts.dir** file
  - Create a **fonts.alias** file to map font substitutions
  - Tell X to re-read font information

    **#xset fp rehash**

    You must use **mkfontdir** after adding new fonts

4/27/2015

12



## Font server

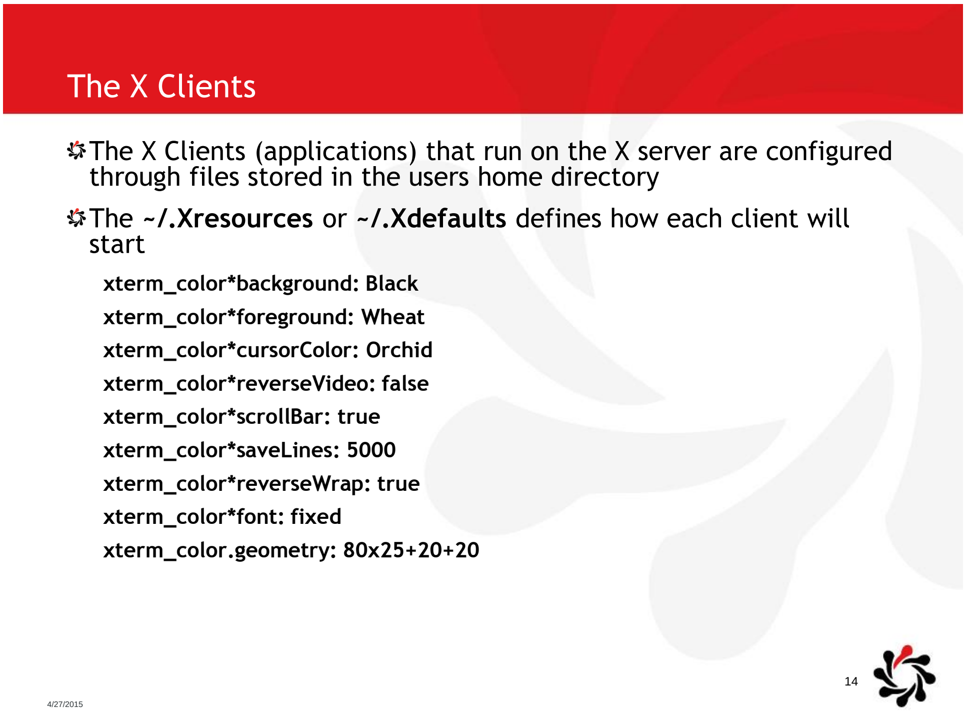
- The font server configuration file is located in **/etc/X11/fs/config**
- To start and stop the service use

```
#service xfs stop/start/restart
#/etc/init.d/xfs start/stop/restart
```

- The X Font Service needs to be started before starting X.

4/27/2015

13



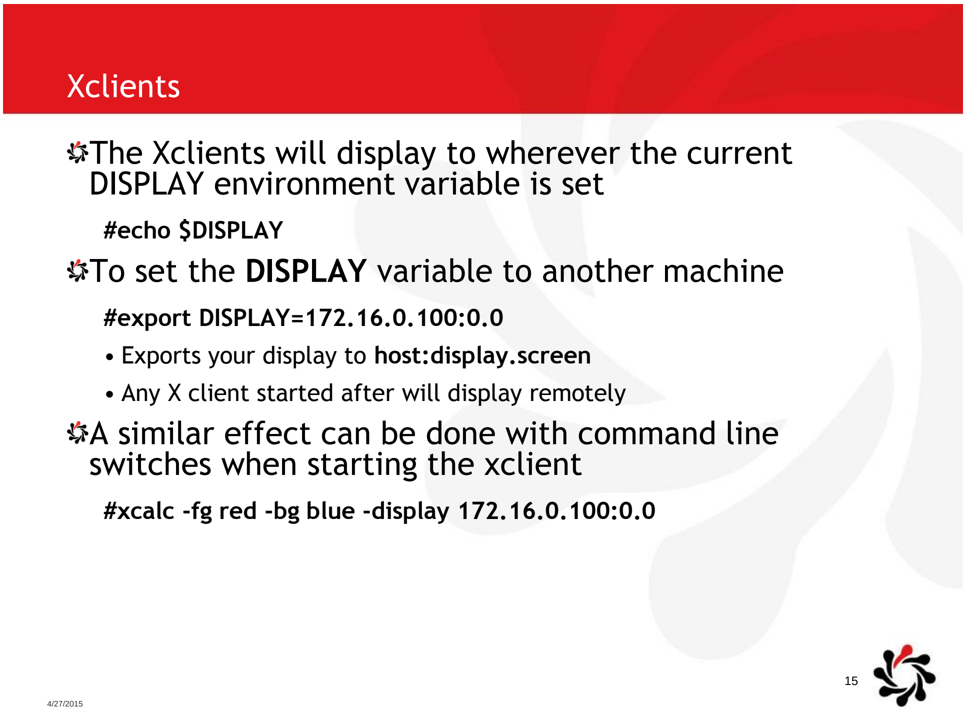
## The X Clients

- The X Clients (applications) that run on the X server are configured through files stored in the users home directory
- The **~/.Xresources** or **~/.Xdefaults** defines how each client will start

```
xterm_color*background: Black
xterm_color*foreground: Wheat
xterm_color*cursorColor: Orchid
xterm_color*reverseVideo: false
xterm_color*scrollBar: true
xterm_color*saveLines: 5000
xterm_color*reverseWrap: true
xterm_color*font: fixed
xterm_color.geometry: 80x25+20+20
```

4/27/2015

14



## Xclients

- The Xclients will display to wherever the current DISPLAY environment variable is set

```
#echo $DISPLAY
```

- To set the **DISPLAY** variable to another machine

```
#export DISPLAY=172.16.0.100:0.0
```

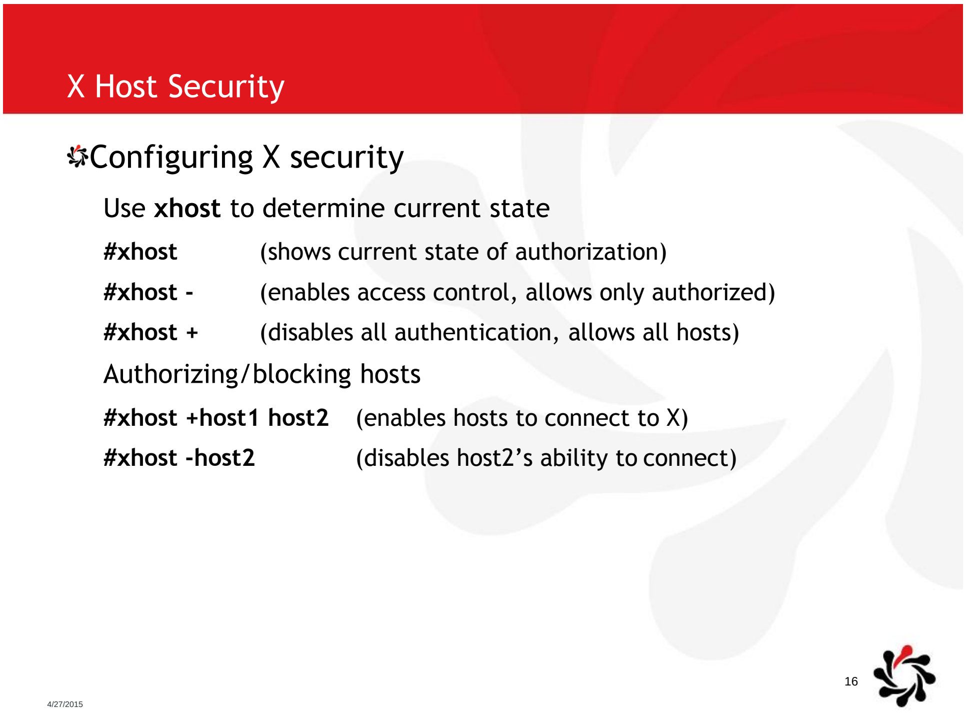
  - Exports your display to **host:display.screen**
  - Any X client started after will display remotely

- A similar effect can be done with command line switches when starting the xclient

```
#xcalc -fg red -bg blue -display 172.16.0.100:0.0
```

4/27/2015

15



## X Host Security

- Configuring X security

Use **xhost** to determine current state

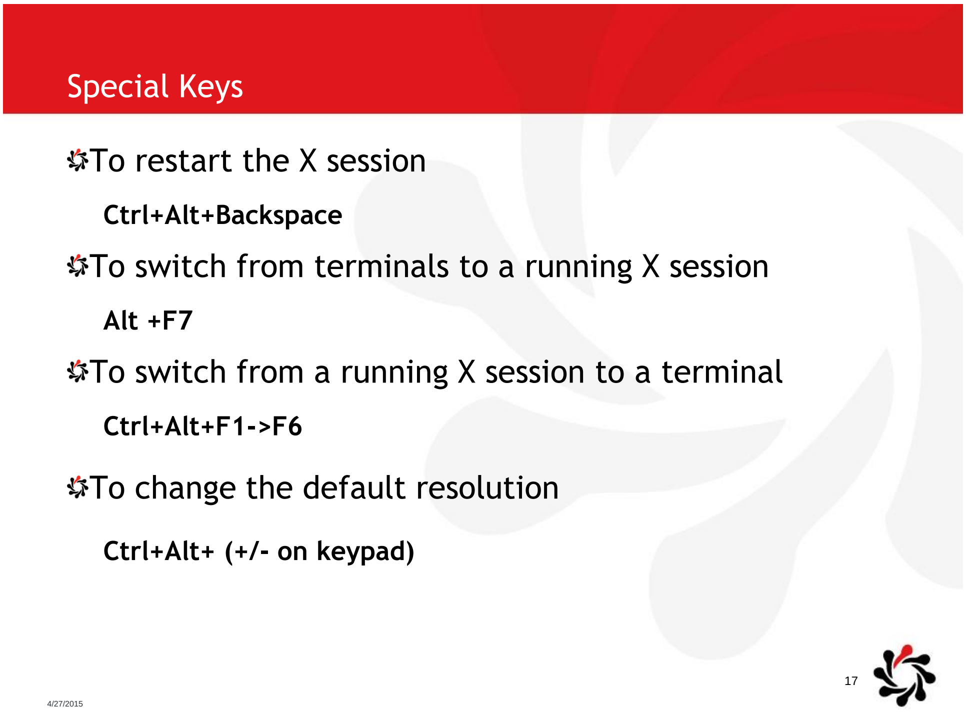
| | |
|---|---|
| **#xhost** | (shows current state of authorization) |
| **#xhost -** | (enables access control, allows only authorized) |
| **#xhost +** | (disables all authentication, allows all hosts) |

Authorizing/blocking hosts

| | |
|---|---|
| **#xhost +host1 host2** | (enables hosts to connect to X) |
| **#xhost -host2** | (disables host2's ability to connect) |

4/27/2015

16



## Special Keys

- To restart the X session

  **Ctrl+Alt+Backspace**

- To switch from terminals to a running X session

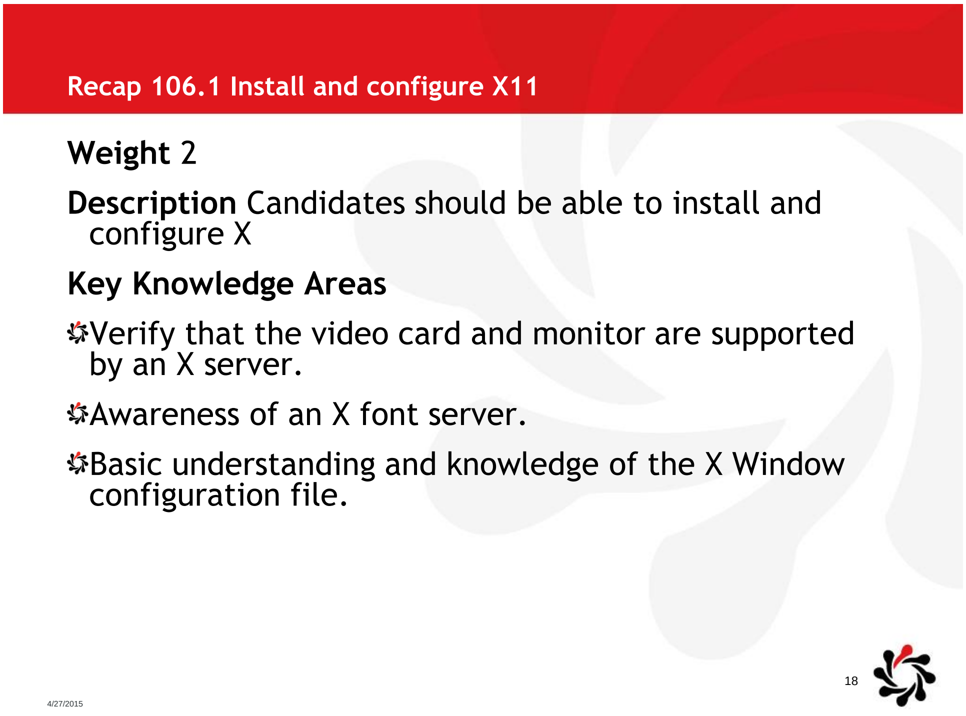
  **Alt +F7**

- To switch from a running X session to a terminal

  **Ctrl+Alt+F1->F6**

- To change the default resolution

  **Ctrl+Alt+ (+/- on keypad)**



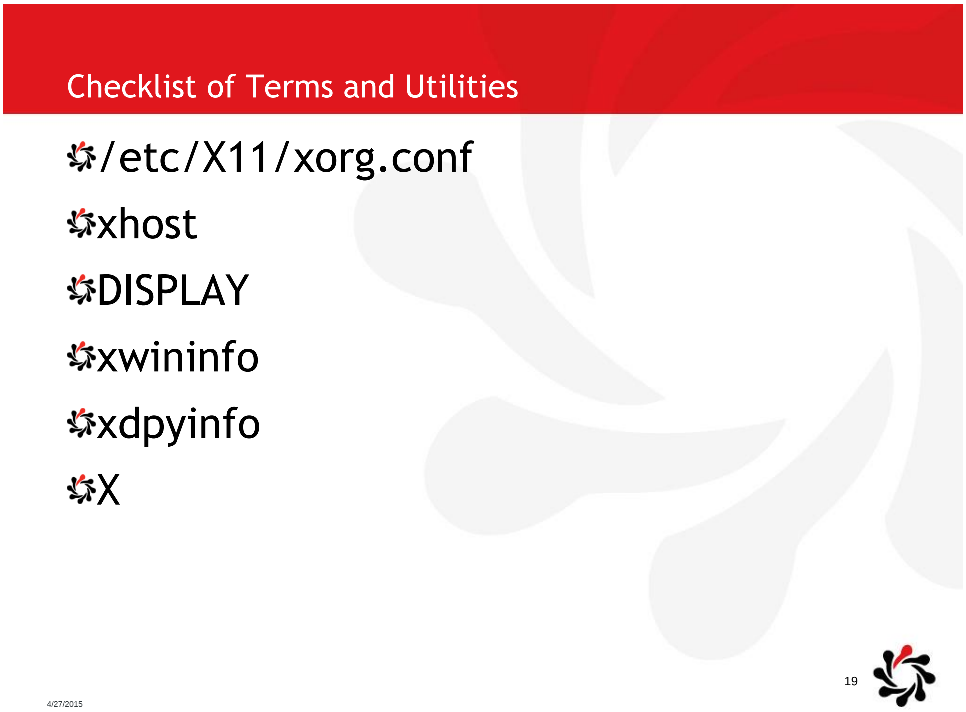
## Recap 106.1 Install and configure X11

**Weight** 2

**Description** Candidates should be able to install and configure X

**Key Knowledge Areas**

- Verify that the video card and monitor are supported by an X server.
- Awareness of an X font server.
- Basic understanding and knowledge of the X Window configuration file.

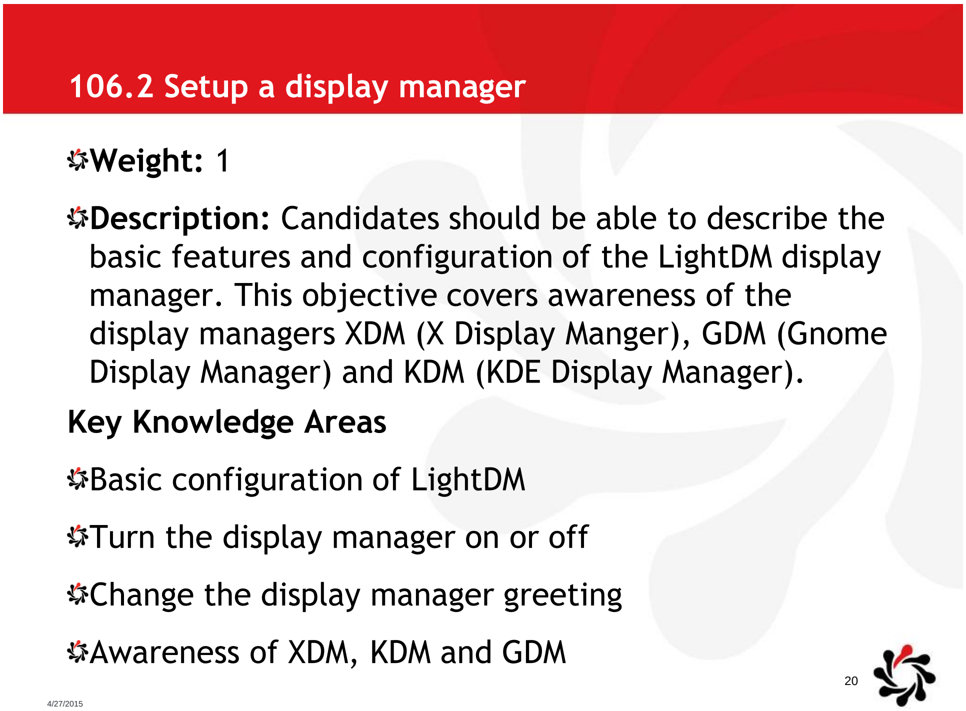


## Checklist of Terms and Utilities

- /etc/X11/xorg.conf
- xhost
- DISPLAY
- xwininfo
- xdpyinfo
- X

19

4/27/2015



## 106.2 Setup a display manager

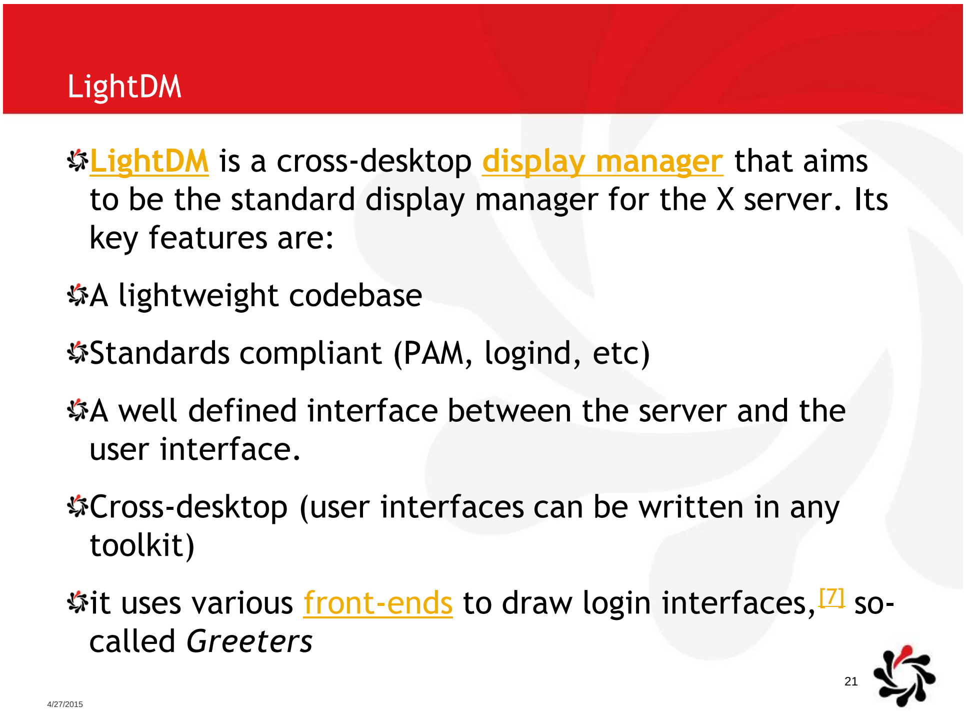
- **Weight:** 1
- **Description:** Candidates should be able to describe the basic features and configuration of the LightDM display manager. This objective covers awareness of the display managers XDM (X Display Manger), GDM (Gnome Display Manager) and KDM (KDE Display Manager).

**Key Knowledge Areas**

- Basic configuration of LightDM
- Turn the display manager on or off
- Change the display manager greeting
- Awareness of XDM, KDM and GDM

20

4/27/2015

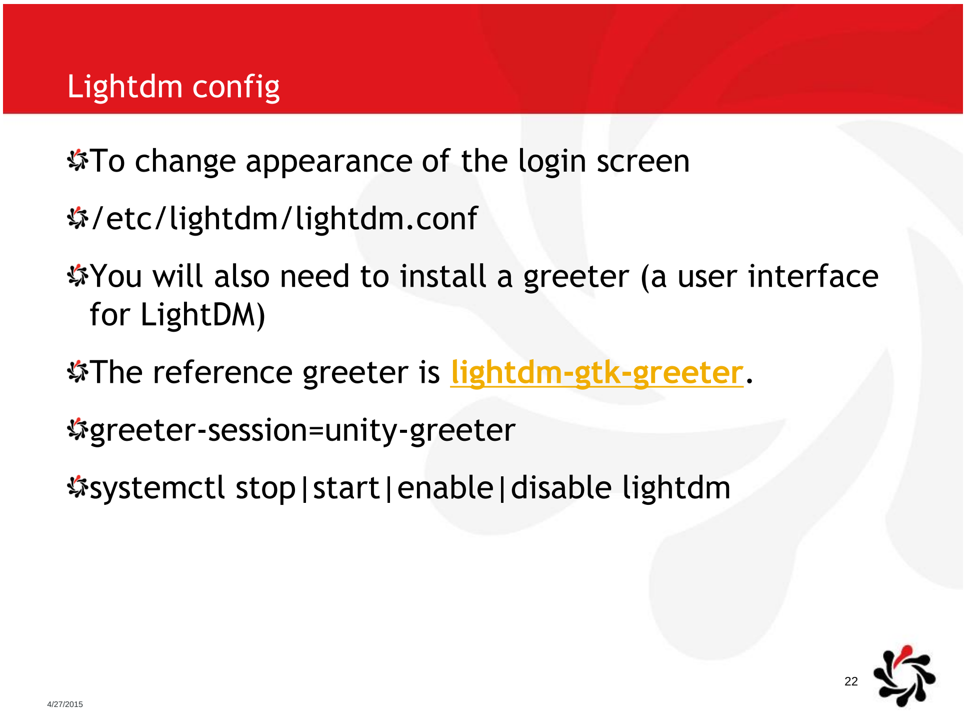


## LightDM

- LightDM is a cross-desktop display manager that aims to be the standard display manager for the X server. Its key features are:
- A lightweight codebase
- Standards compliant (PAM, logind, etc)
- A well defined interface between the server and the user interface.
- Cross-desktop (user interfaces can be written in any toolkit)
- it uses various front-ends to draw login interfaces,[7] so-called *Greeters*

4/27/2015

21

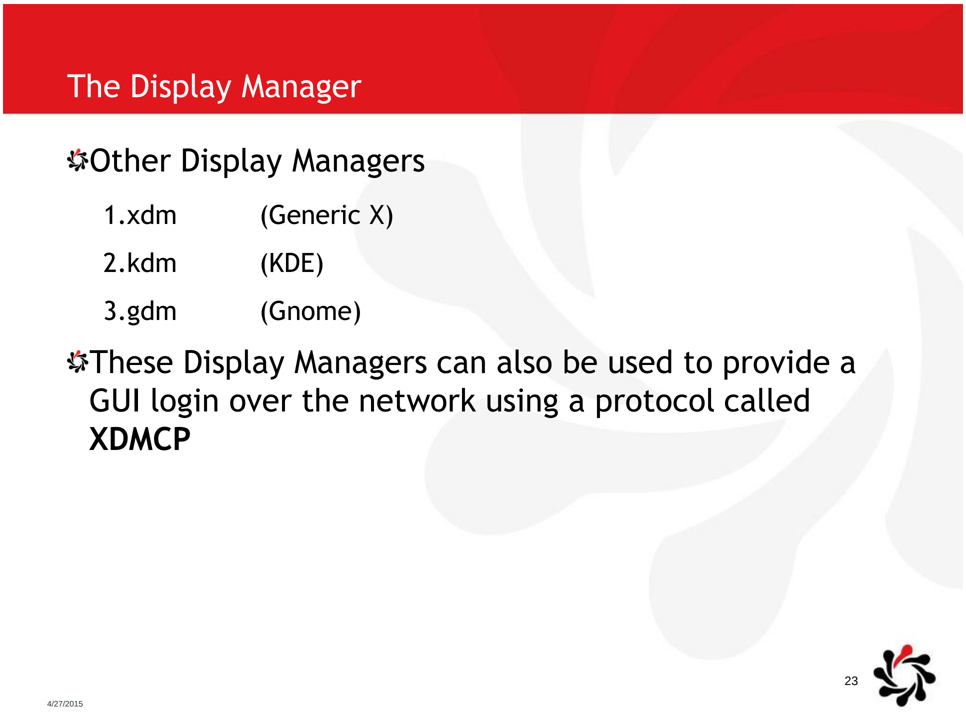


## Lightdm config

- To change appearance of the login screen
- /etc/lightdm/lightdm.conf
- You will also need to install a greeter (a user interface for LightDM)
- The reference greeter is lightdm-gtk-greeter.
- greeter-session=unity-greeter
- systemctl stop|start|enable|disable lightdm

4/27/2015

22



## The Display Manager

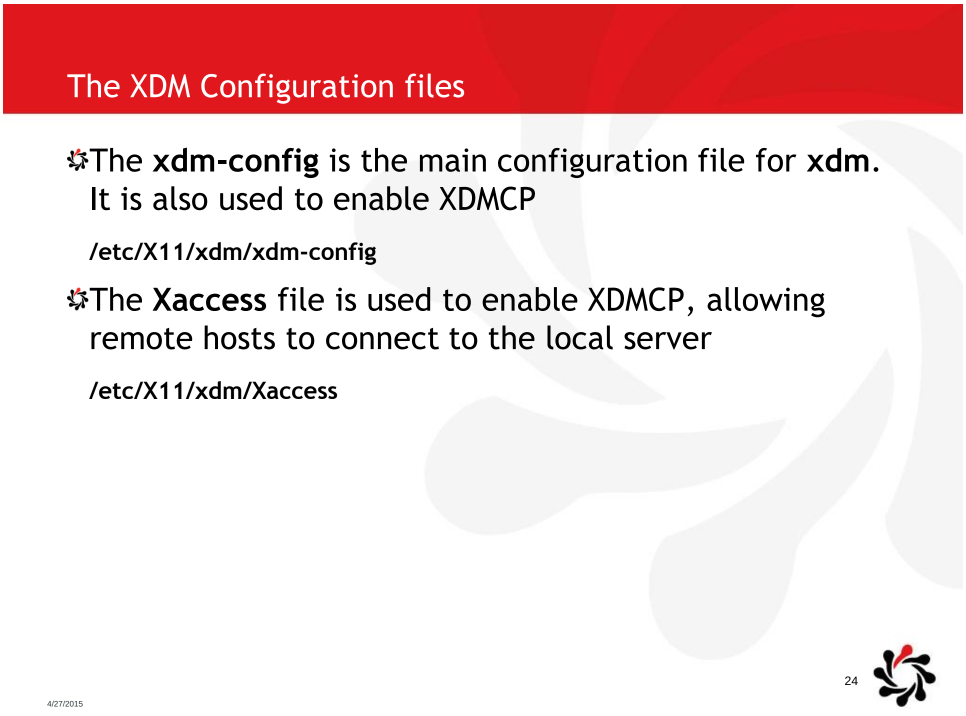
- Other Display Managers

| | |
|---|---|
| 1.xdm | (Generic X) |
| 2.kdm | (KDE) |
| 3.gdm | (Gnome) |

- These Display Managers can also be used to provide a GUI login over the network using a protocol called **XDMCP**

23

4/27/2015



## The XDM Configuration files

- The **xdm-config** is the main configuration file for **xdm**. It is also used to enable XDMCP

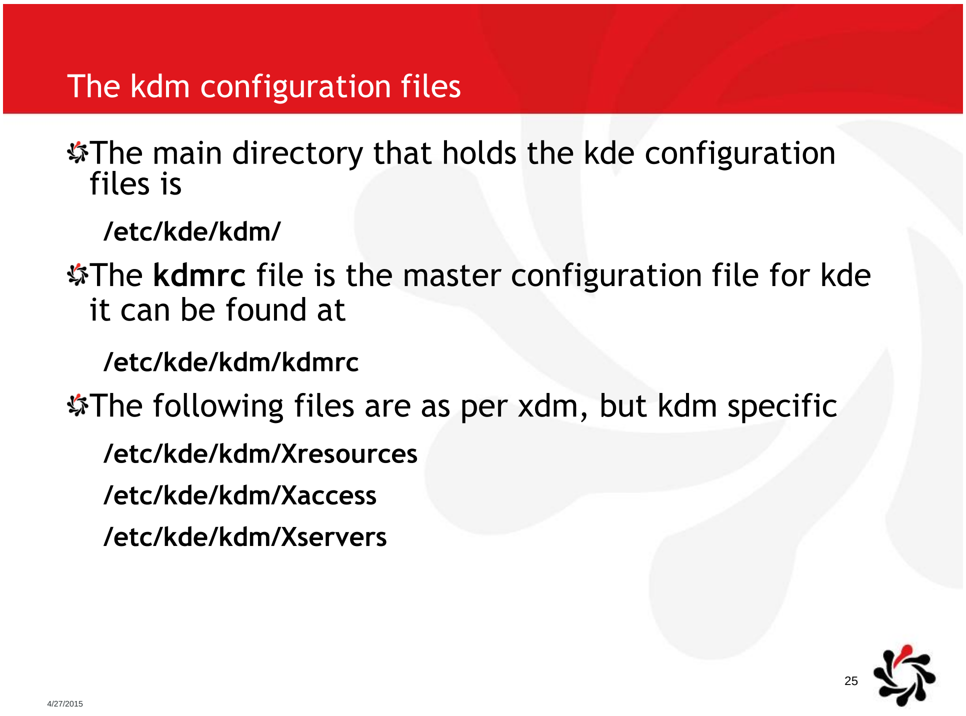
  **/etc/X11/xdm/xdm-config**

- The **Xaccess** file is used to enable XDMCP, allowing remote hosts to connect to the local server

  **/etc/X11/xdm/Xaccess**

24

4/27/2015



## The kdm configuration files

- The main directory that holds the kde configuration files is

  **/etc/kde/kdm/**

- The **kdmrc** file is the master configuration file for kde it can be found at

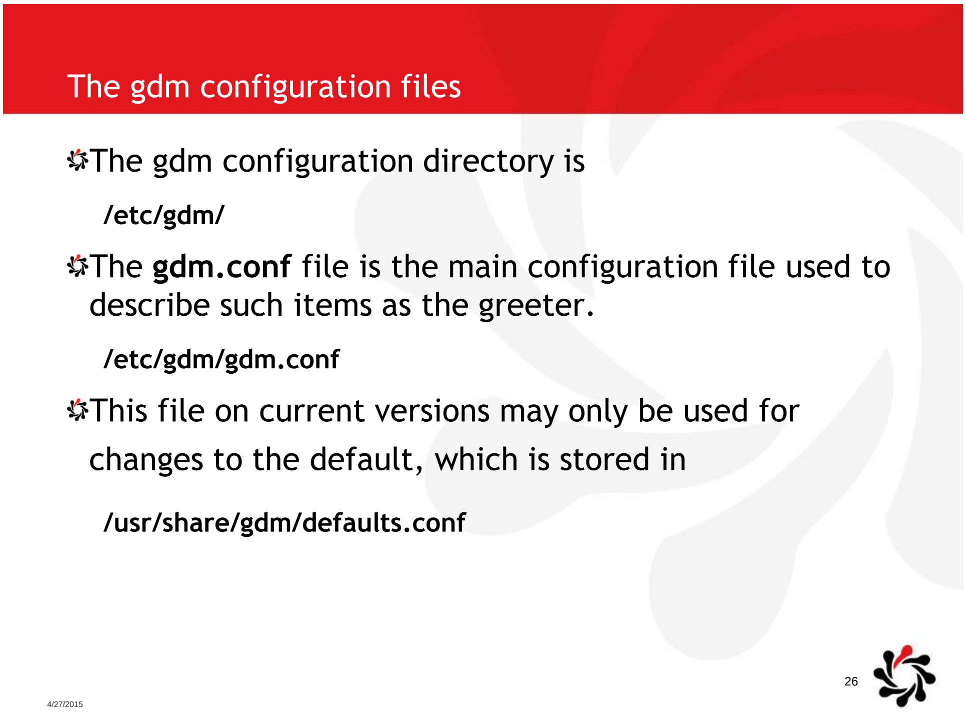
  **/etc/kde/kdm/kdmrc**

- The following files are as per xdm, but kdm specific

  **/etc/kde/kdm/Xresources**

  **/etc/kde/kdm/Xaccess**

  **/etc/kde/kdm/Xservers**



## The gdm configuration files

- The gdm configuration directory is

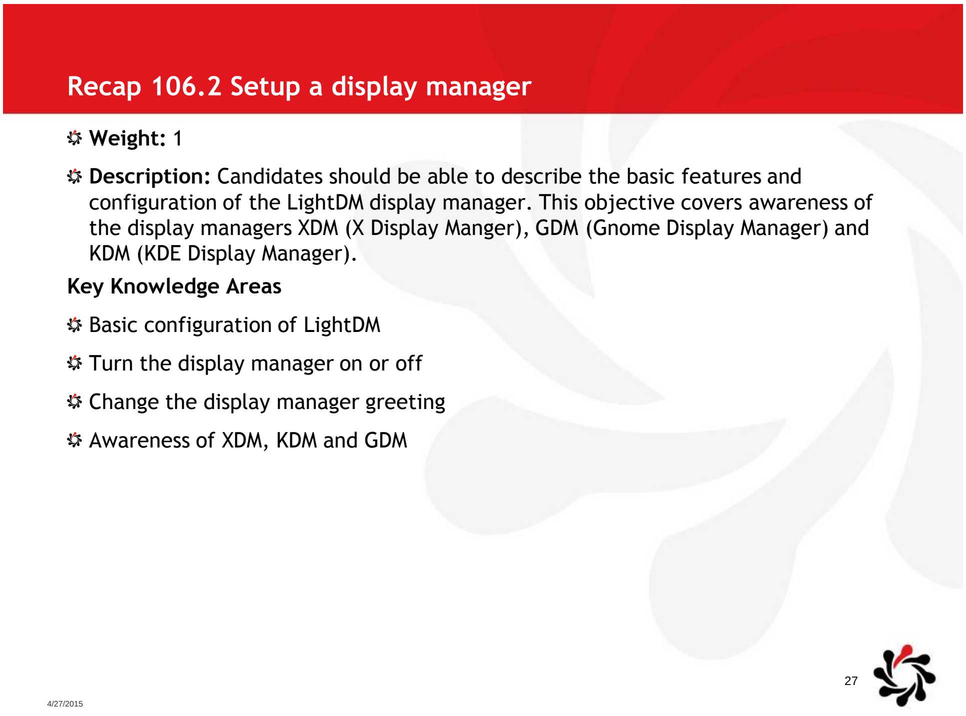
  **/etc/gdm/**

- The **gdm.conf** file is the main configuration file used to describe such items as the greeter.

  **/etc/gdm/gdm.conf**

- This file on current versions may only be used for changes to the default, which is stored in

  **/usr/share/gdm/defaults.conf**

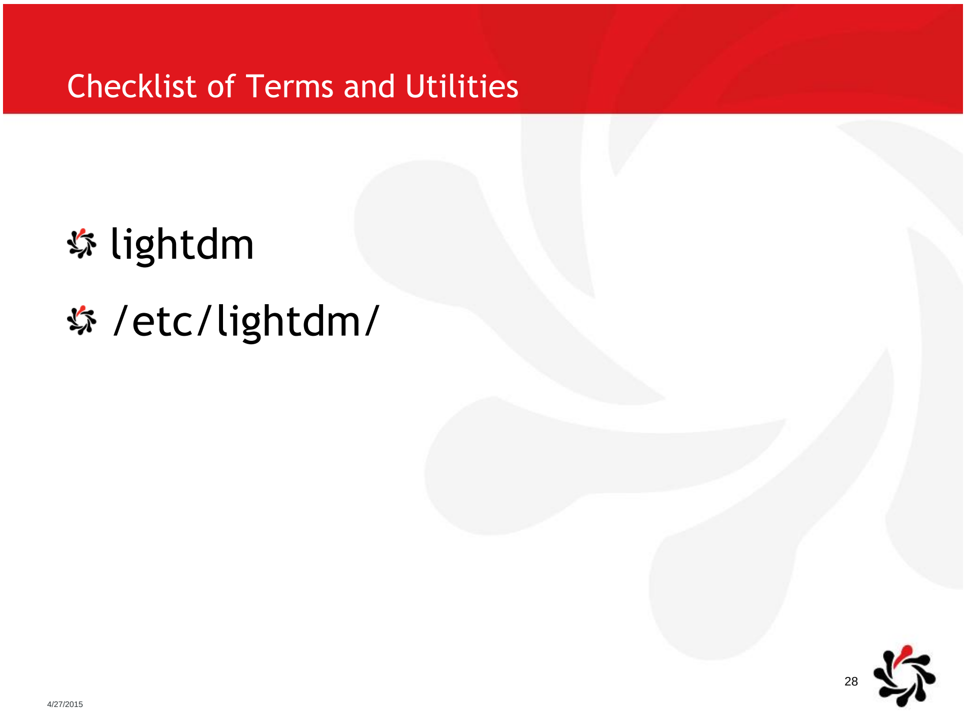


## Recap 106.2 Setup a display manager

- **Weight:** 1
- **Description:** Candidates should be able to describe the basic features and configuration of the LightDM display manager. This objective covers awareness of the display managers XDM (X Display Manger), GDM (Gnome Display Manager) and KDM (KDE Display Manager).

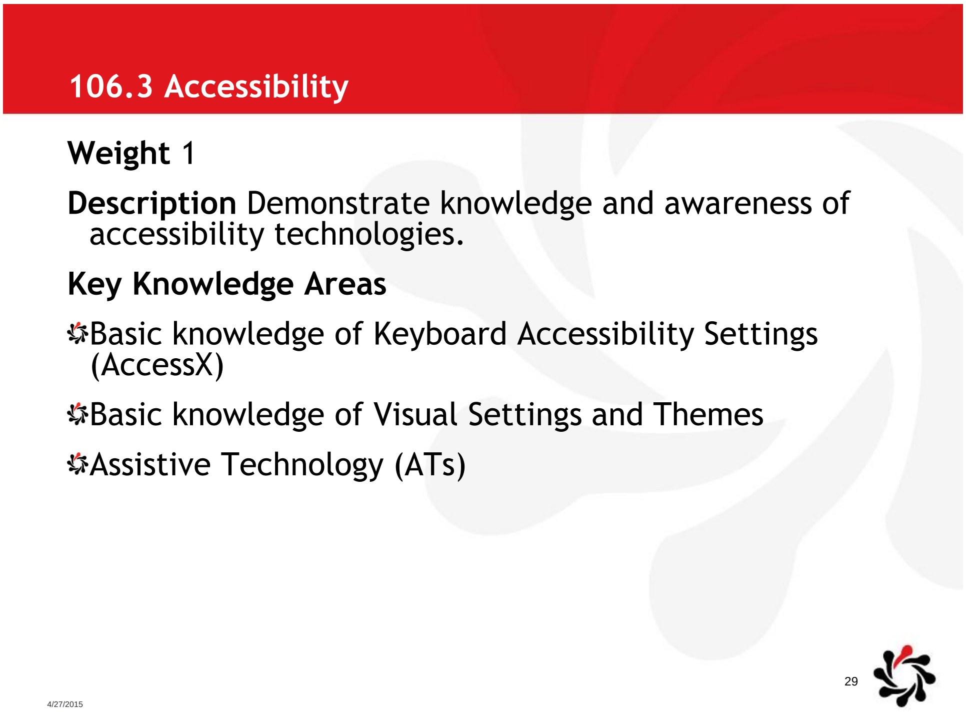
**Key Knowledge Areas**

- Basic configuration of LightDM
- Turn the display manager on or off
- Change the display manager greeting
- Awareness of XDM, KDM and GDM



## Checklist of Terms and Utilities

- lightdm
- /etc/lightdm/



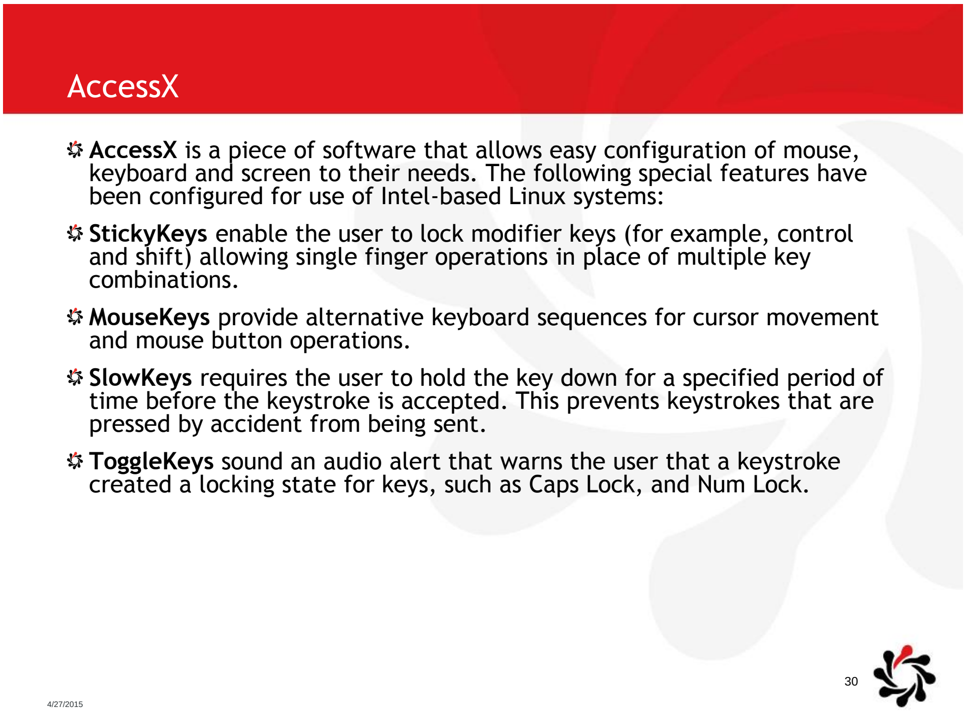
## 106.3 Accessibility

**Weight** 1

**Description** Demonstrate knowledge and awareness of accessibility technologies.

**Key Knowledge Areas**

- Basic knowledge of Keyboard Accessibility Settings (AccessX)
- Basic knowledge of Visual Settings and Themes
- Assistive Technology (ATs)



## AccessX

- **AccessX** is a piece of software that allows easy configuration of mouse, keyboard and screen to their needs. The following special features have been configured for use of Intel-based Linux systems:
- **StickyKeys** enable the user to lock modifier keys (for example, control and shift) allowing single finger operations in place of multiple key combinations.
- **MouseKeys** provide alternative keyboard sequences for cursor movement and mouse button operations.
- **SlowKeys** requires the user to hold the key down for a specified period of time before the keystroke is accepted. This prevents keystrokes that are pressed by accident from being sent.
- **ToggleKeys** sound an audio alert that warns the user that a keystroke created a locking state for keys, such as Caps Lock, and Num Lock.

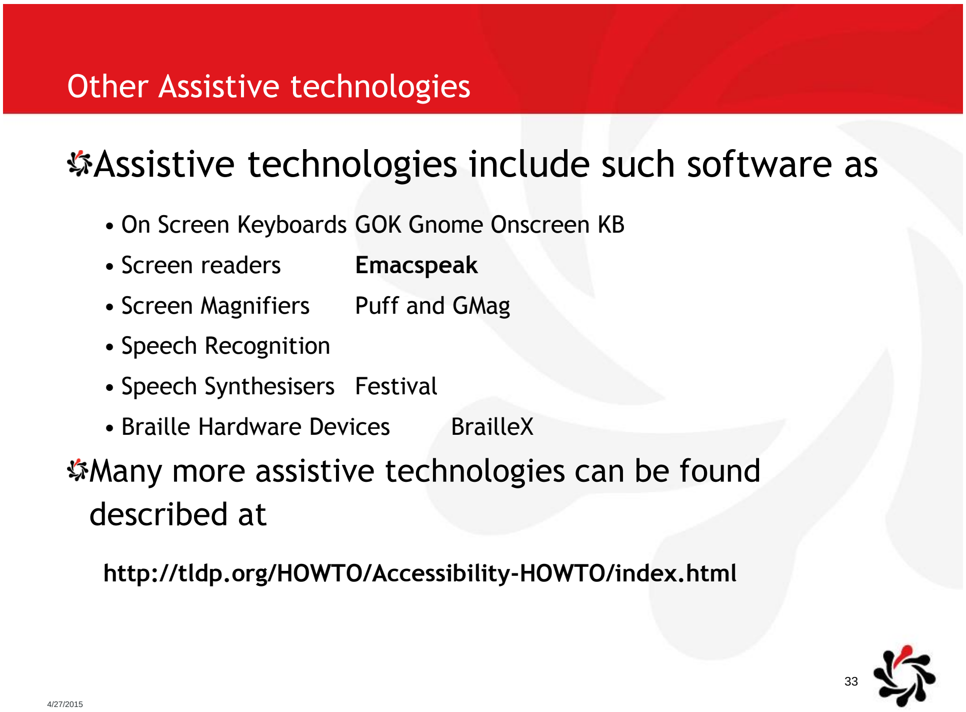
## AccessX

- **RepeatKeys** allow a user with limited coordination additional time to release keys before multiple key sequences are sent to the application.
- **BounceKeys** or **Delay Keys** have a delay between keystrokes. This function can help prevent the system from accepting unintentional keystrokes.
- **AccessX Time Out** Allows the user to set the time when the features will become disabled after no keyboard activity. This feature is especially useful to a user who may type on a terminal and turn-on the settings, leave the computer and another non-handicap user begins to type.
- **Video Mode Changing** lets users change their video screen mode on demand.
- **Control Panel** allows the user to apply the settings before saving, save the user's settings, tab through the panel (for those who cannot use a mouse), give the user the option to restore the to the default settings, and more.

## ORCA

- **Orca** is a open source, flexible, extensible assistive technology for people with visual impairments. Using various combinations of speech synthesis, braille, and magnification.
- Orca can be downloaded from

  http://live.gnome.org/Orca/



## Other Assistive technologies

- Assistive technologies include such software as
  - On Screen Keyboards GOK Gnome Onscreen KB
  - Screen readers **Emacspeak**
  - Screen Magnifiers Puff and GMag
  - Speech Recognition
  - Speech Synthesisers Festival
  - Braille Hardware Devices BrailleX
- Many more assistive technologies can be found described at

  **http://tldp.org/HOWTO/Accessibility-HOWTO/index.html**

4/27/2015

33

## 106.3 Accessibility

**Weight** 1

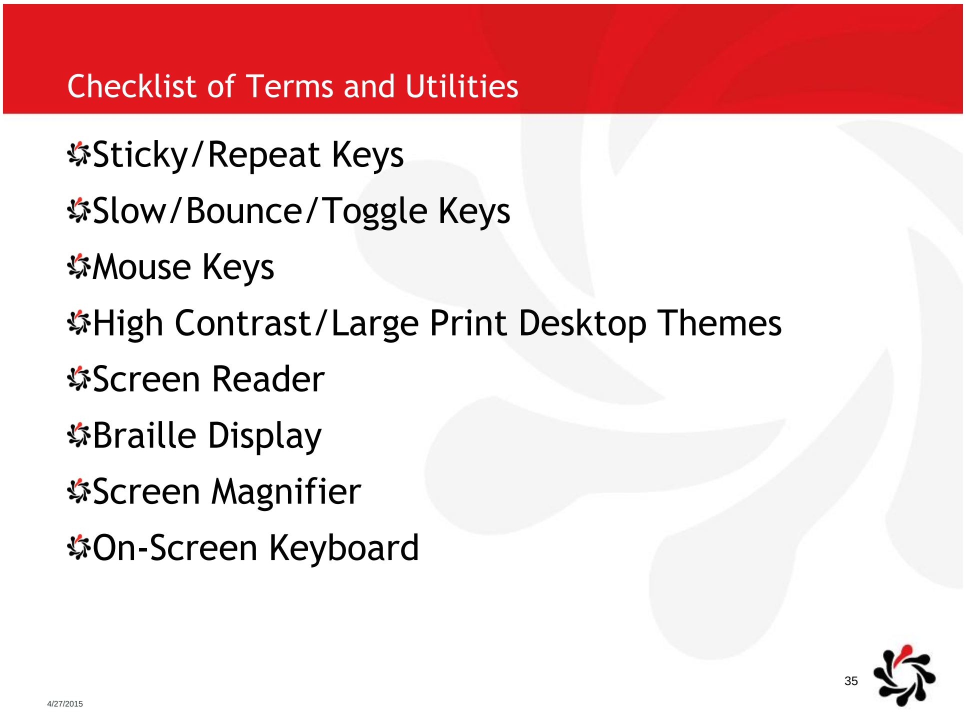
**Description** Demonstrate knowledge and awareness of accessibility technologies.

**Key Knowledge Areas**

- Basic knowledge of Keyboard Accessibility Settings (AccessX)
- Basic knowledge of Visual Settings and Themes
- Assistive Technology (ATs)

4/27/2015

34

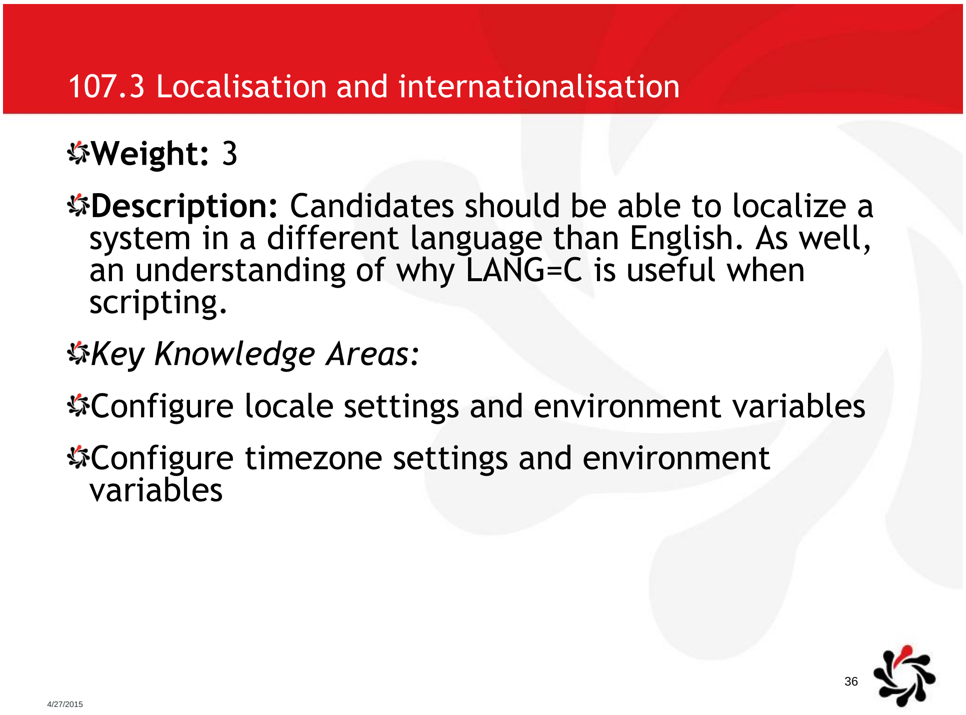


## Checklist of Terms and Utilities

- Sticky/Repeat Keys
- Slow/Bounce/Toggle Keys
- Mouse Keys
- High Contrast/Large Print Desktop Themes
- Screen Reader
- Braille Display
- Screen Magnifier
- On-Screen Keyboard

4/27/2015

35

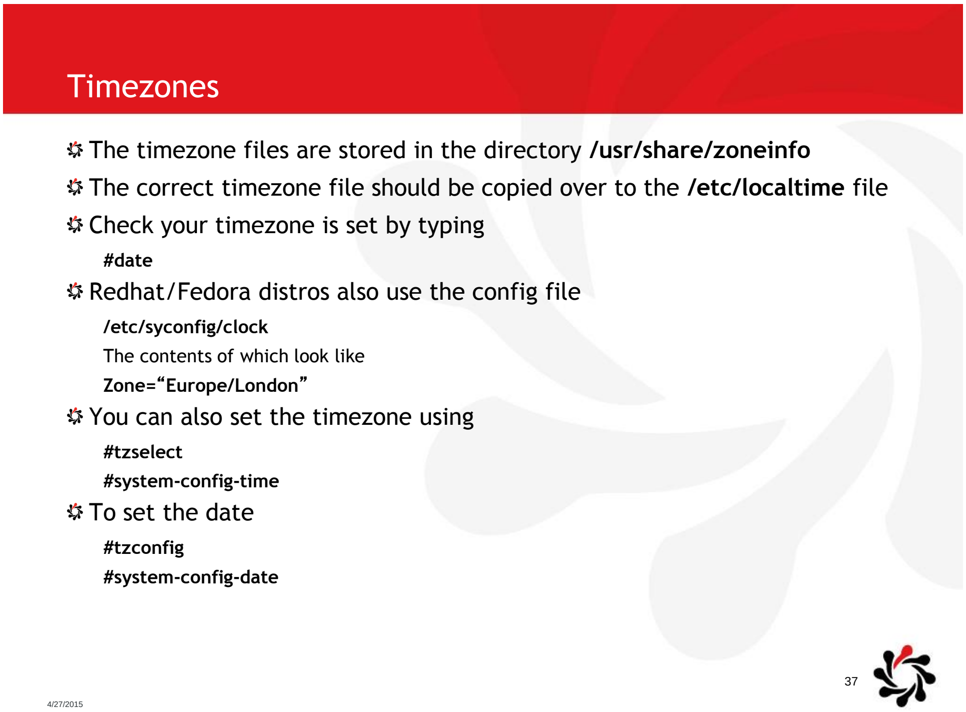


## 107.3 Localisation and internationalisation

- **Weight:** 3
- **Description:** Candidates should be able to localize a system in a different language than English. As well, an understanding of why LANG=C is useful when scripting.
- *Key Knowledge Areas:*
- Configure locale settings and environment variables
- Configure timezone settings and environment variables

4/27/2015

36



## Timezones

- The timezone files are stored in the directory **/usr/share/zoneinfo**
- The correct timezone file should be copied over to the **/etc/localtime** file
- Check your timezone is set by typing

```
#date
```

- Redhat/Fedora distros also use the config file

  **/etc/syconfig/clock**

  The contents of which look like

  **Zone="Europe/London"**

- You can also set the timezone using

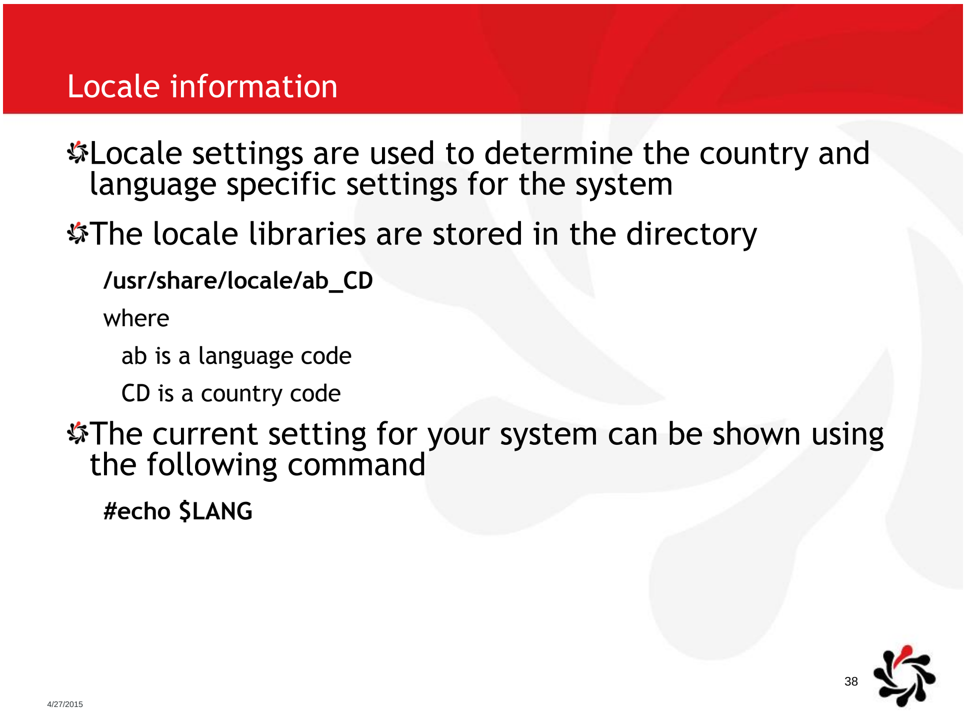
```
#tzselect
#system-config-time
```

- To set the date

```
#tzconfig
#system-config-date
```

37

4/27/2015



## Locale information

- Locale settings are used to determine the country and language specific settings for the system
- The locale libraries are stored in the directory

  **/usr/share/locale/ab_CD**

  where

  ab is a language code

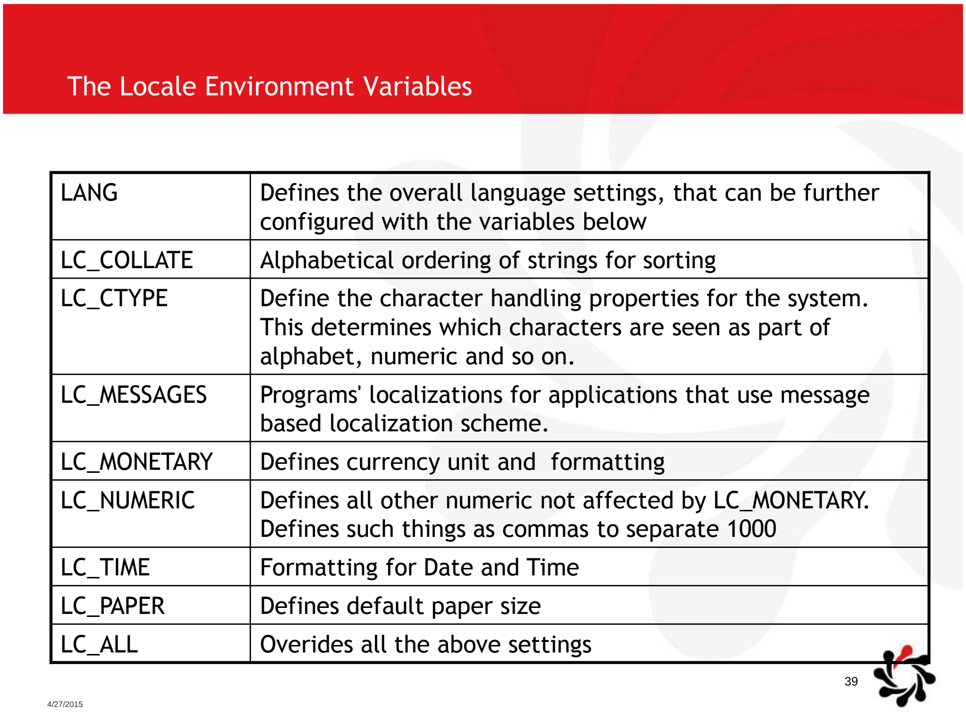
  CD is a country code

- The current setting for your system can be shown using the following command

```
#echo $LANG
```

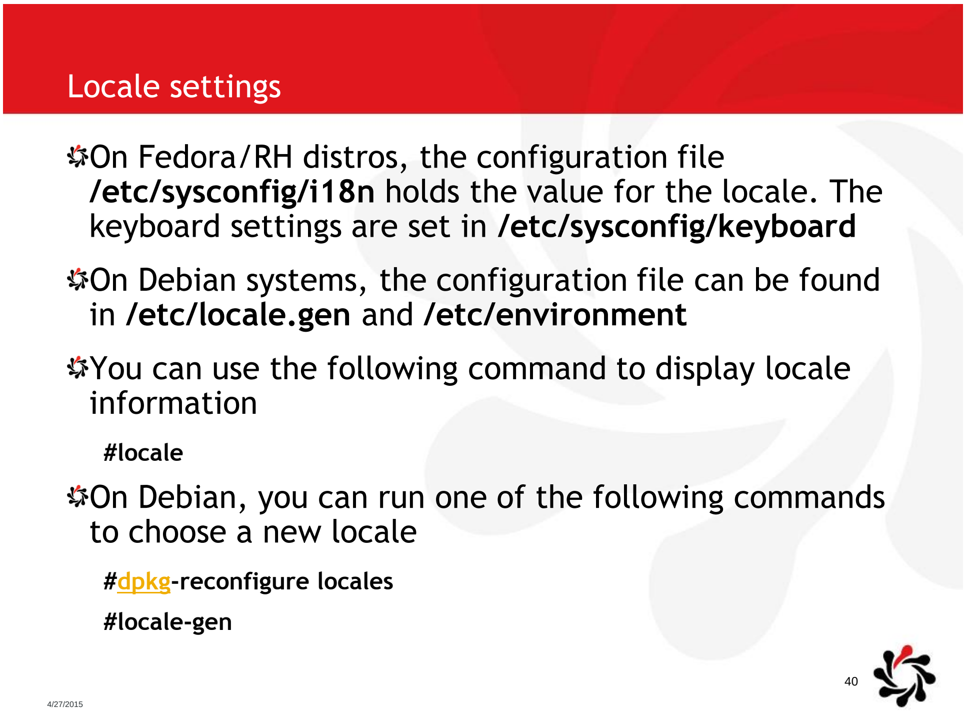
38

4/27/2015



## The Locale Environment Variables

| | |
|---|---|
| LANG | Defines the overall language settings, that can be further configured with the variables below |
| LC_COLLATE | Alphabetical ordering of strings for sorting |
| LC_CTYPE | Define the character handling properties for the system. This determines which characters are seen as part of alphabet, numeric and so on. |
| LC_MESSAGES | Programs' localizations for applications that use message based localization scheme. |
| LC_MONETARY | Defines currency unit and formatting |
| LC_NUMERIC | Defines all other numeric not affected by LC_MONETARY. Defines such things as commas to separate 1000 |
| LC_TIME | Formatting for Date and Time |
| LC_PAPER | Defines default paper size |
| LC_ALL | Overides all the above settings |



## Locale settings

- On Fedora/RH distros, the configuration file **/etc/sysconfig/i18n** holds the value for the locale. The keyboard settings are set in **/etc/sysconfig/keyboard**
- On Debian systems, the configuration file can be found in **/etc/locale.gen** and **/etc/environment**
- You can use the following command to display locale information

```
#locale
```

- On Debian, you can run one of the following commands to choose a new locale

```
#dpkg-reconfigure locales
#locale-gen
```

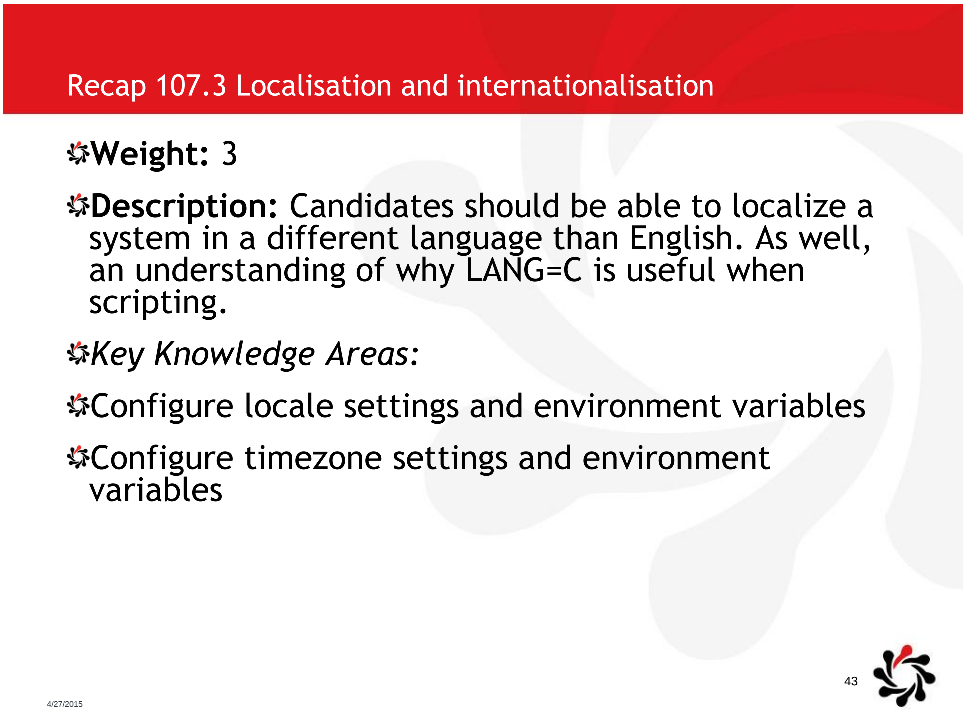
## LANG=C

- The LANG=C locale settings allows standardisation of output regardless of locale setting.  Many different languages have different alphabets which means commands like **sort** will output differently
- Therefore when using scripting set the locale LANG=C
- Other output may be standardised if you are in a foreign country

```
#LANG=C ifconfig -a
#LANG=de_DE ifocnfig -a
```

## iconv library

- Libiconv is a gnu utility to convert between traditional encodings like ISO8859 and ASCII to full Unicode.

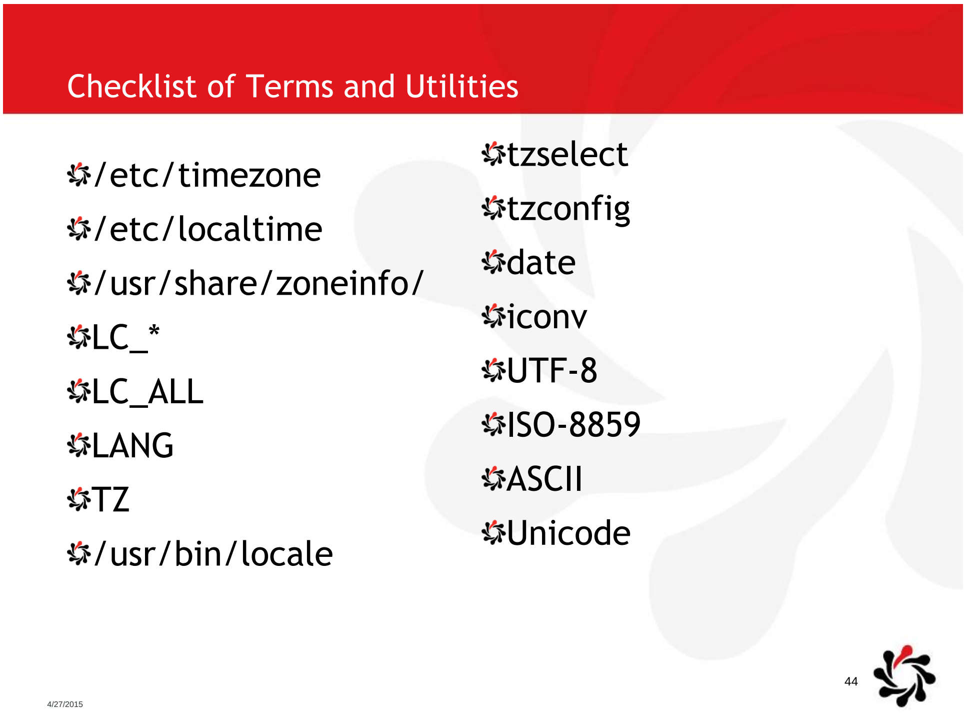


## Recap 107.3 Localisation and internationalisation

- **Weight:** 3
- **Description:** Candidates should be able to localize a system in a different language than English. As well, an understanding of why LANG=C is useful when scripting.
- *Key Knowledge Areas:*
- Configure locale settings and environment variables
- Configure timezone settings and environment variables

4/27/2015

43



## Checklist of Terms and Utilities

- /etc/timezone
- /etc/localtime
- /usr/share/zoneinfo/
- LC_*
- LC_ALL
- LANG
- TZ
- /usr/bin/locale

- tzselect
- tzconfig
- date
- iconv
- UTF-8
- ISO-8859
- ASCII
- Unicode

4/27/2015

44

## 108.4 Manage printers and printing

**Weight** 2

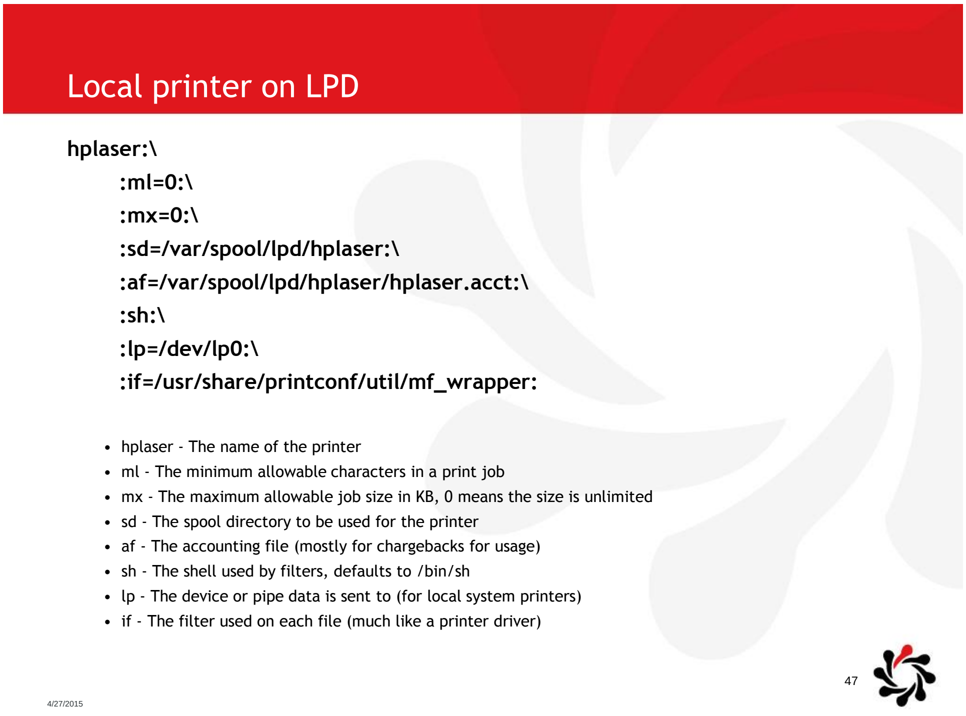
**Description** Candidates should be able to manage print queues and user print jobs using CUPS and the LPD compatibility interface.

**Key Knowledge Areas**

- Basic CUPS configuration (for local and remote printers).
- Manage user print queues.
- Troubleshoot general printing problems.
- Add and remove jobs from configured printer queues.

## LPD

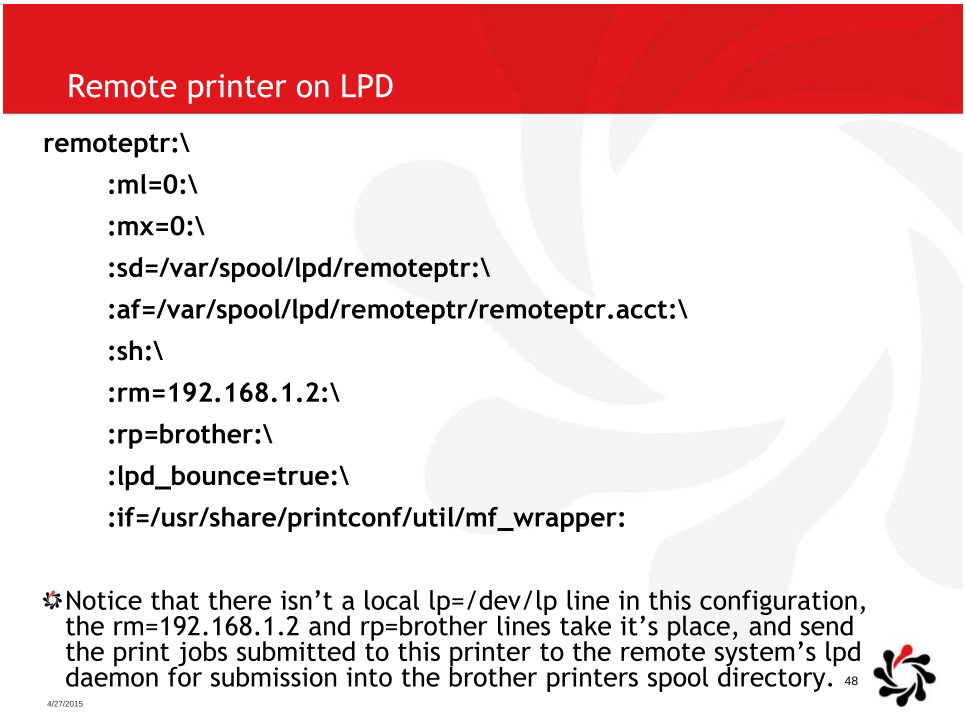
- The LPD daemon is an older printing daemon.
- It has a single configuration file that lists all the printers configured on the system. This file is **/etc/printcap**
- Security on the printer was done through a basic file called **/etc/hosts.lpd**, which if exists, had to list all hosts that were allowed to print



## Local printer on LPD

```
hplaser:\
        :ml=0:\
        :mx=0:\
        :sd=/var/spool/lpd/hplaser:\
        :af=/var/spool/lpd/hplaser/hplaser.acct:\
        :sh:\
        :lp=/dev/lp0:\
        :if=/usr/share/printconf/util/mf_wrapper:
```

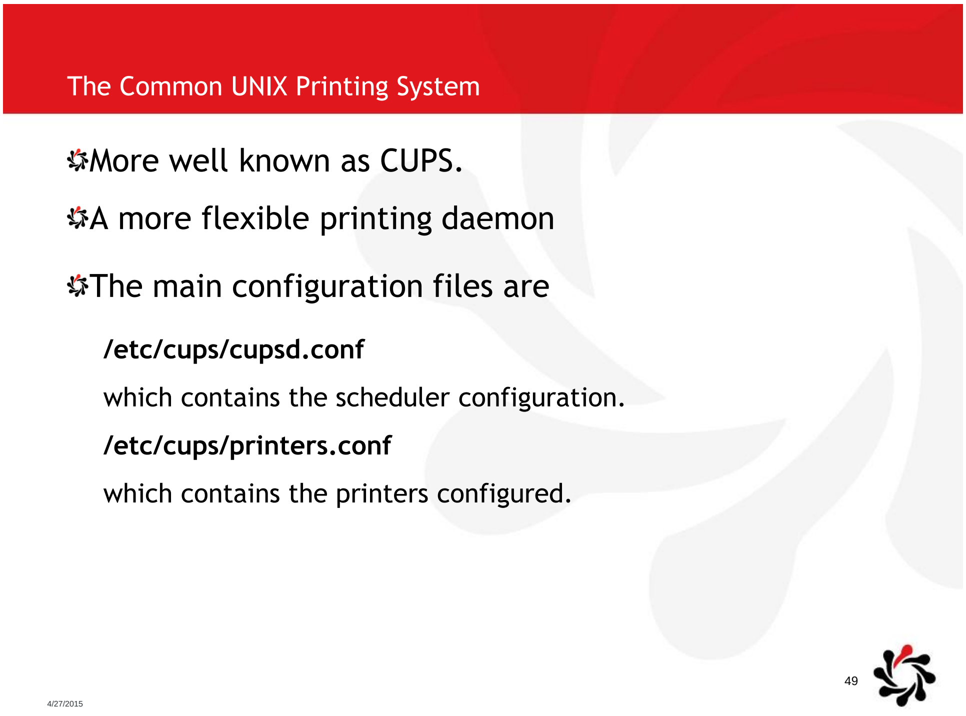
- hplaser - The name of the printer
- ml - The minimum allowable characters in a print job
- mx - The maximum allowable job size in KB, 0 means the size is unlimited
- sd - The spool directory to be used for the printer
- af - The accounting file (mostly for chargebacks for usage)
- sh - The shell used by filters, defaults to /bin/sh
- lp - The device or pipe data is sent to (for local system printers)
- if - The filter used on each file (much like a printer driver)



## Remote printer on LPD

```
remoteptr:\
        :ml=0:\
        :mx=0:\
        :sd=/var/spool/lpd/remoteptr:\
        :af=/var/spool/lpd/remoteptr/remoteptr.acct:\
        :sh:\
        :rm=192.168.1.2:\
        :rp=brother:\
        :lpd_bounce=true:\
        :if=/usr/share/printconf/util/mf_wrapper:
```

- Notice that there isn’t a local lp=/dev/lp line in this configuration, the rm=192.168.1.2 and rp=brother lines take it’s place, and send the print jobs submitted to this printer to the remote system’s lpd daemon for submission into the brother printers spool directory.



## The Common UNIX Printing System

- More well known as CUPS.
- A more flexible printing daemon
- The main configuration files are

  **/etc/cups/cupsd.conf**

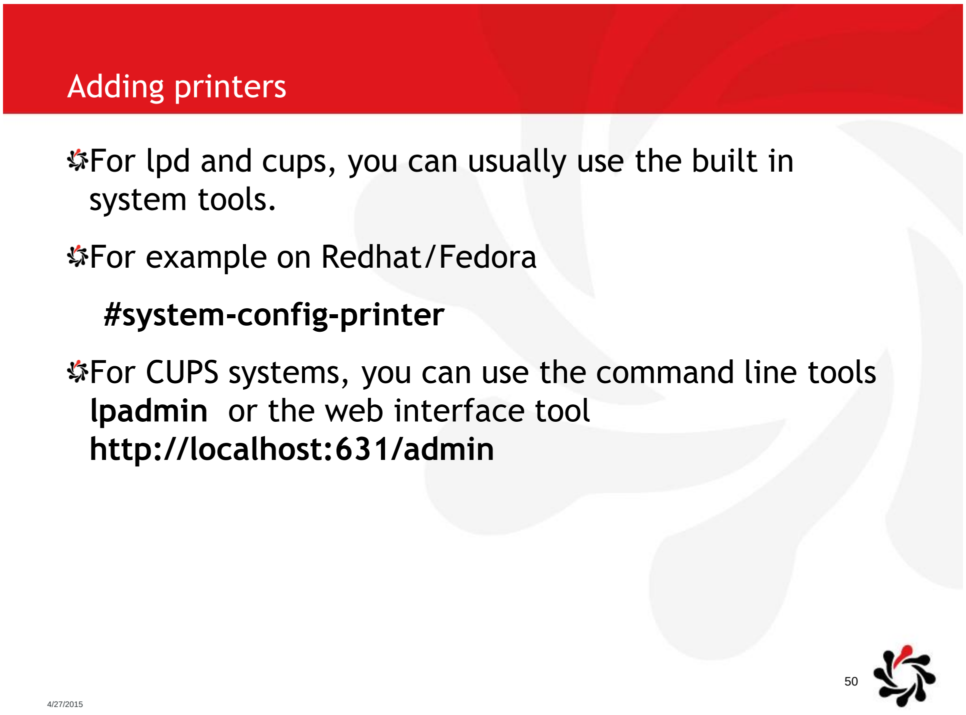
  which contains the scheduler configuration.

  **/etc/cups/printers.conf**

  which contains the printers configured.

4/27/2015

49



## Adding printers

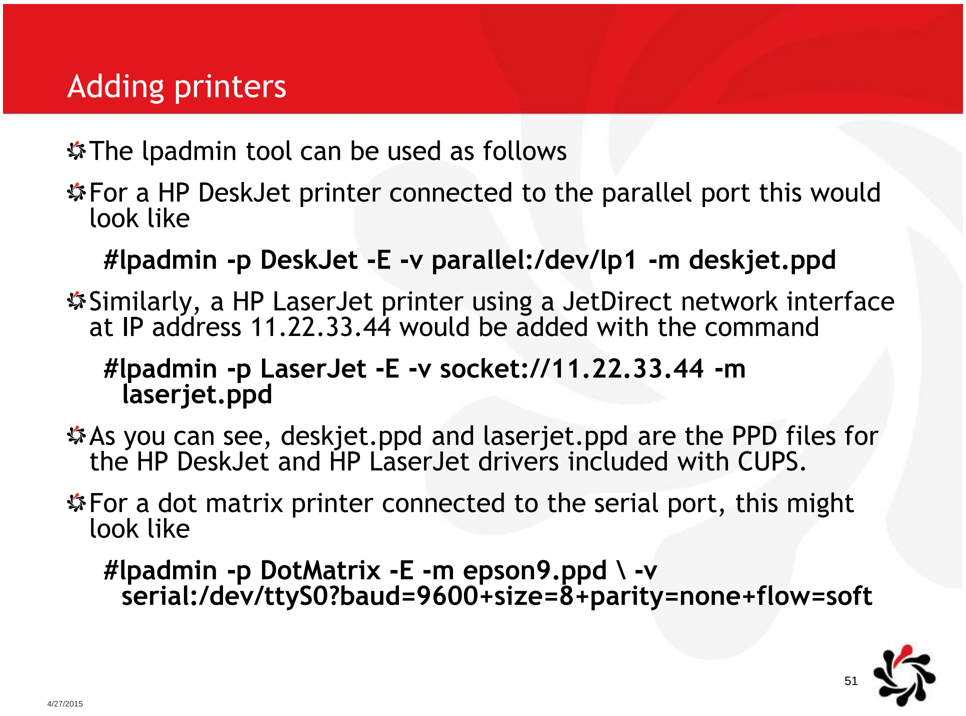
- For lpd and cups, you can usually use the built in system tools.
- For example on Redhat/Fedora

  **#system-config-printer**

- For CUPS systems, you can use the command line tools **lpadmin** or the web interface tool **http://localhost:631/admin**

4/27/2015

50



## Adding printers

- The lpadmin tool can be used as follows
- For a HP DeskJet printer connected to the parallel port this would look like

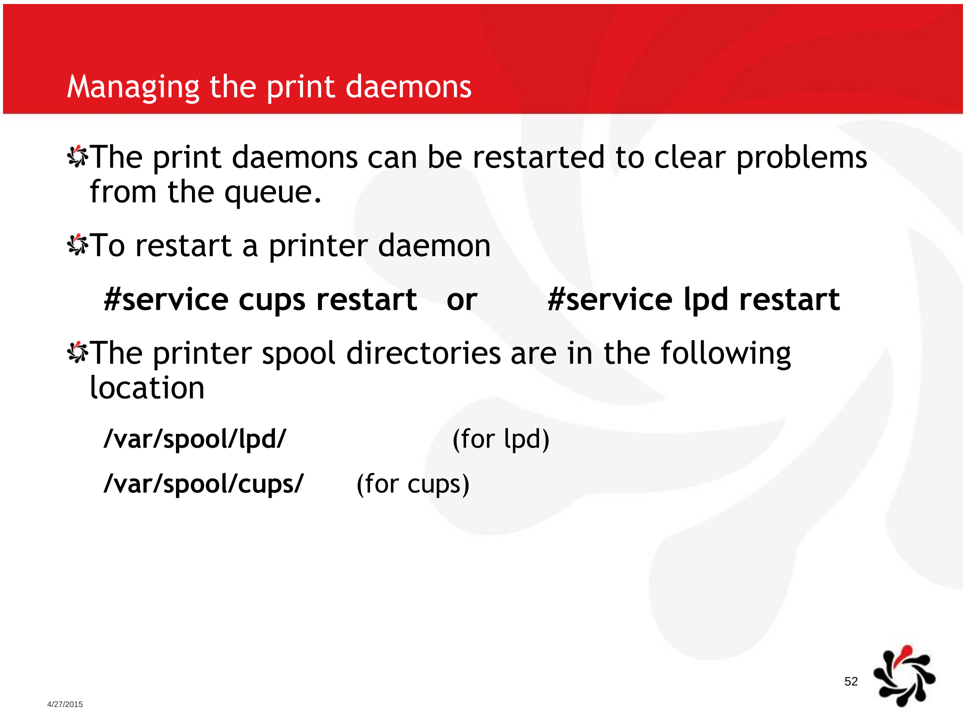
```
#lpadmin -p DeskJet -E -v parallel:/dev/lp1 -m deskjet.ppd
```

- Similarly, a HP LaserJet printer using a JetDirect network interface at IP address 11.22.33.44 would be added with the command

```
#lpadmin -p LaserJet -E -v socket://11.22.33.44 -m
  laserjet.ppd
```

- As you can see, deskjet.ppd and laserjet.ppd are the PPD files for the HP DeskJet and HP LaserJet drivers included with CUPS.
- For a dot matrix printer connected to the serial port, this might look like

```
#lpadmin -p DotMatrix -E -m epson9.ppd \ -v
  serial:/dev/ttyS0?baud=9600+size=8+parity=none+flow=soft
```



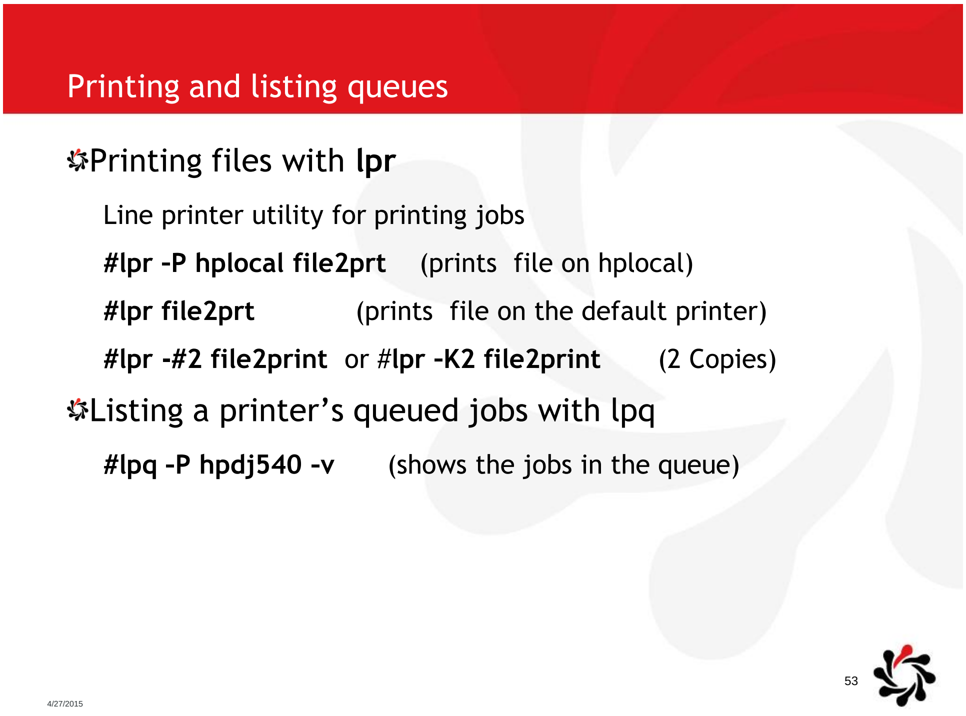
## Managing the print daemons

- The print daemons can be restarted to clear problems from the queue.
- To restart a printer daemon

```
#service cups restart   or       #service lpd restart
```

- The printer spool directories are in the following location

```
/var/spool/lpd/                  (for lpd)
/var/spool/cups/      (for cups)
```



## Printing and listing queues

- Printing files with **lpr**

  Line printer utility for printing jobs

  **#lpr -P hplocal file2prt** (prints file on hplocal)

  **#lpr file2prt** (prints file on the default printer)

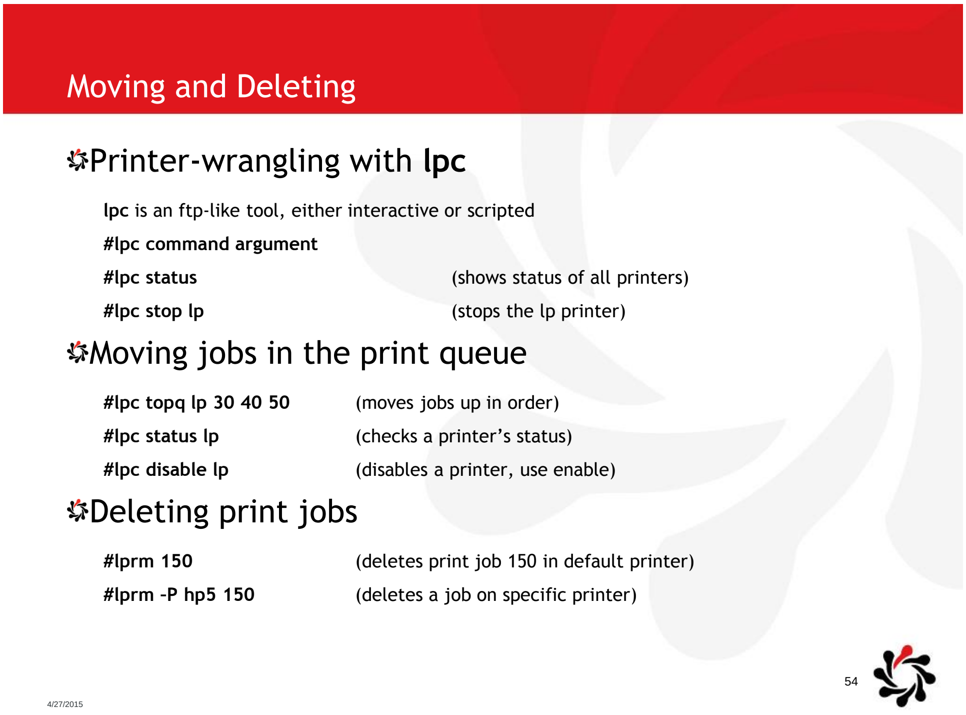
  **#lpr -#2 file2print** or #**lpr -K2 file2print** (2 Copies)

- Listing a printer’s queued jobs with lpq

  **#lpq -P hpdj540 -v** (shows the jobs in the queue)

53

4/27/2015



## Moving and Deleting

- Printer-wrangling with **lpc**

  **lpc** is an ftp-like tool, either interactive or scripted

  **#lpc command argument**

| | |
|---|---|
| **#lpc status** | (shows status of all printers) |
| **#lpc stop lp** | (stops the lp printer) |

- Moving jobs in the print queue

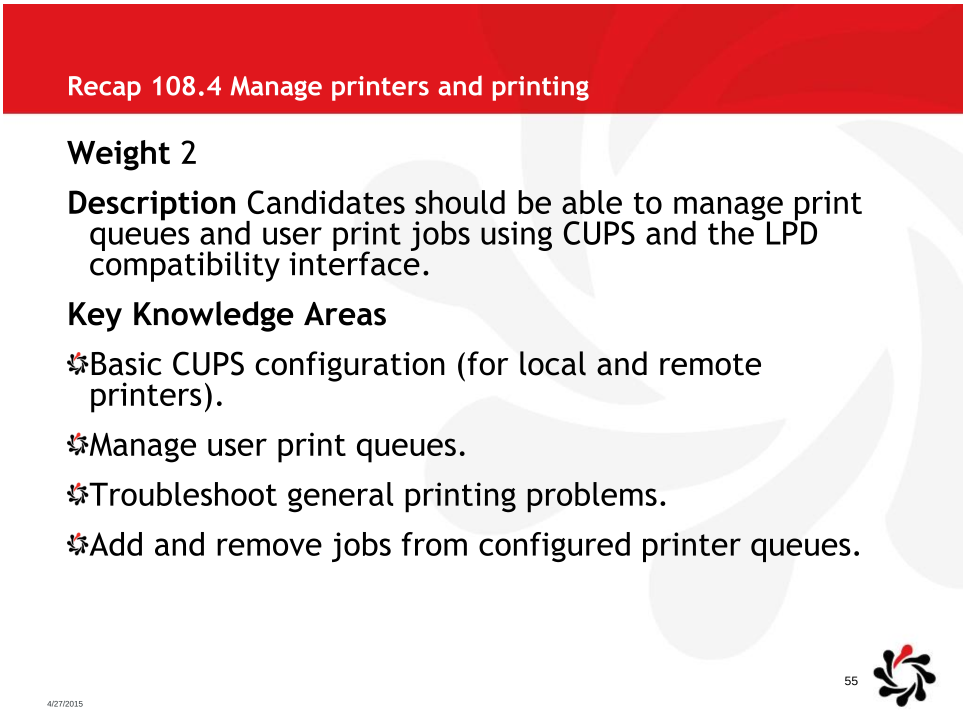
| | |
|---|---|
| **#lpc topq lp 30 40 50** | (moves jobs up in order) |
| **#lpc status lp** | (checks a printer’s status) |
| **#lpc disable lp** | (disables a printer, use enable) |

- Deleting print jobs

| | |
|---|---|
| **#lprm 150** | (deletes print job 150 in default printer) |
| **#lprm -P hp5 150** | (deletes a job on specific printer) |

54

4/27/2015



## Recap 108.4 Manage printers and printing

**Weight** 2

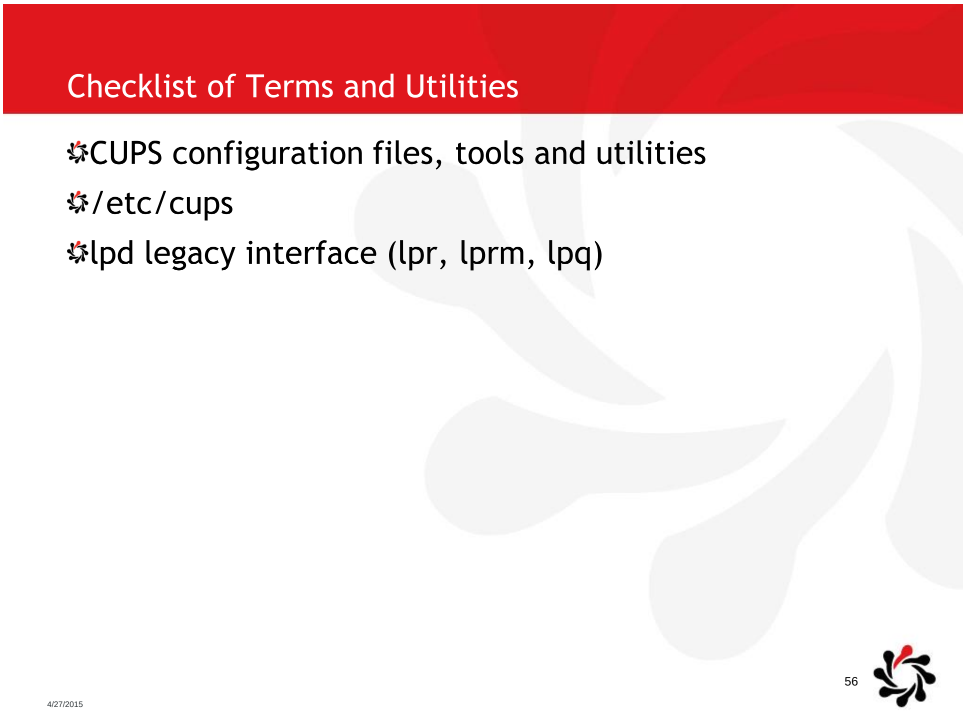
**Description** Candidates should be able to manage print queues and user print jobs using CUPS and the LPD compatibility interface.

**Key Knowledge Areas**

- Basic CUPS configuration (for local and remote printers).
- Manage user print queues.
- Troubleshoot general printing problems.
- Add and remove jobs from configured printer queues.

4/27/2015

55

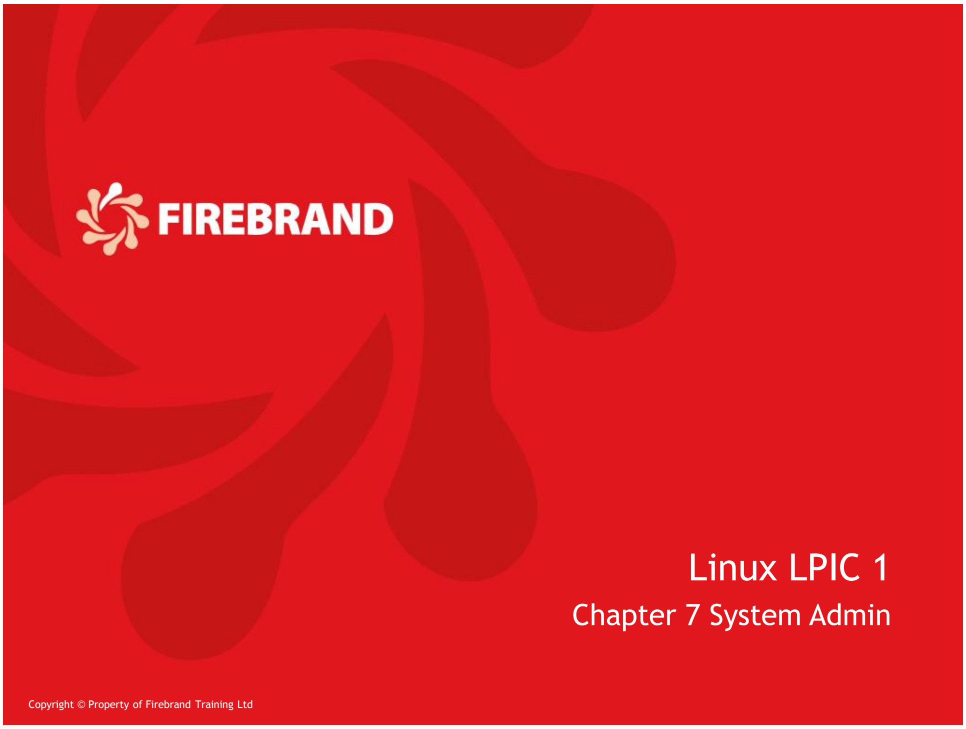


## Checklist of Terms and Utilities

- CUPS configuration files, tools and utilities
- /etc/cups
- lpd legacy interface (lpr, lprm, lpq)

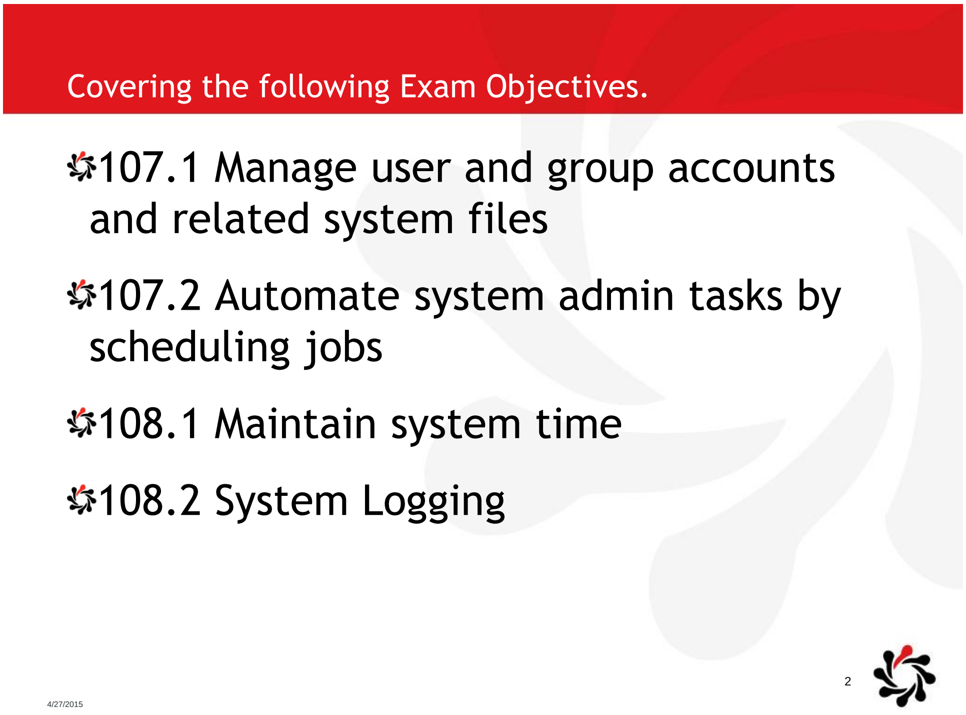
4/27/2015

56



# Linux LPIC 1

Chapter 7 System Admin

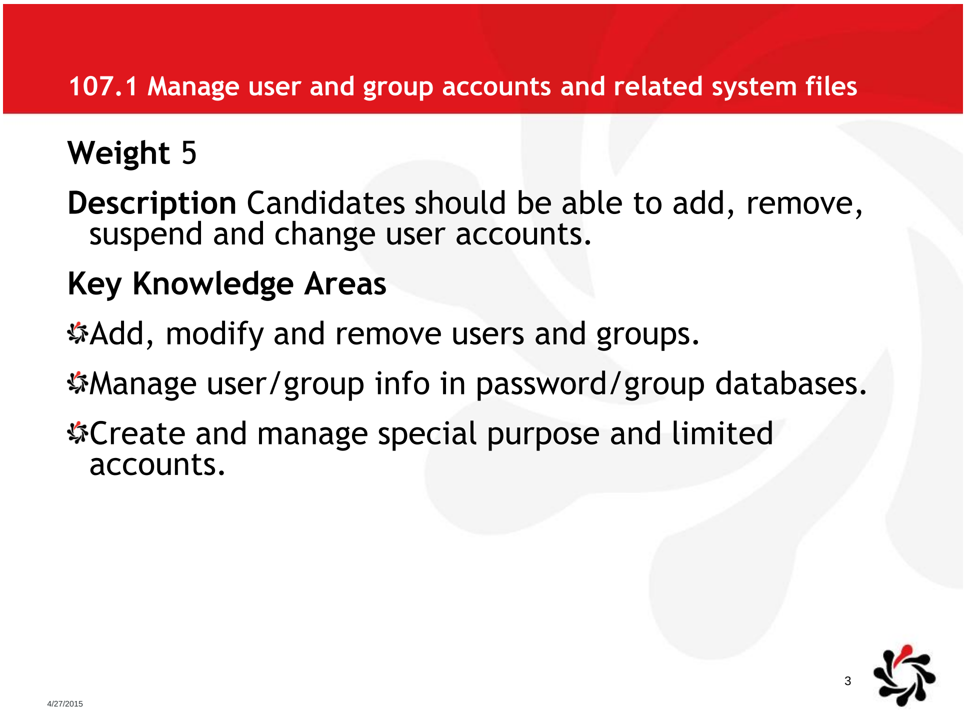


## Covering the following Exam Objectives.

- 107.1 Manage user and group accounts and related system files
- 107.2 Automate system admin tasks by scheduling jobs
- 108.1 Maintain system time
- 108.2 System Logging

4/27/2015

2



## 107.1 Manage user and group accounts and related system files

**Weight** 5

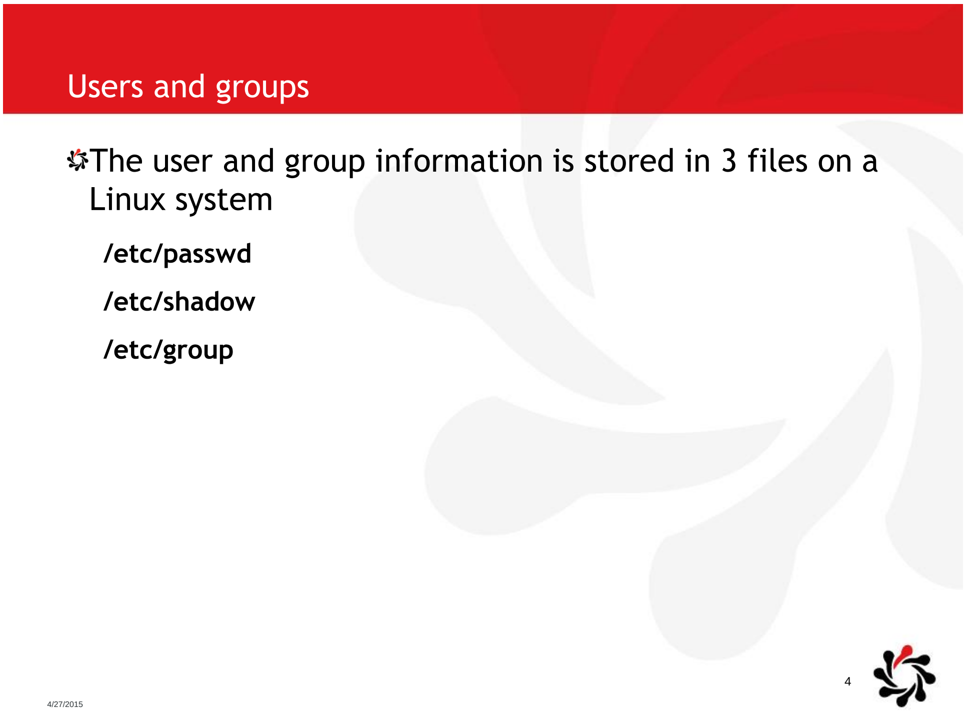
**Description** Candidates should be able to add, remove, suspend and change user accounts.

**Key Knowledge Areas**

- Add, modify and remove users and groups.
- Manage user/group info in password/group databases.
- Create and manage special purpose and limited accounts.

3

4/27/2015



## Users and groups

- The user and group information is stored in 3 files on a Linux system

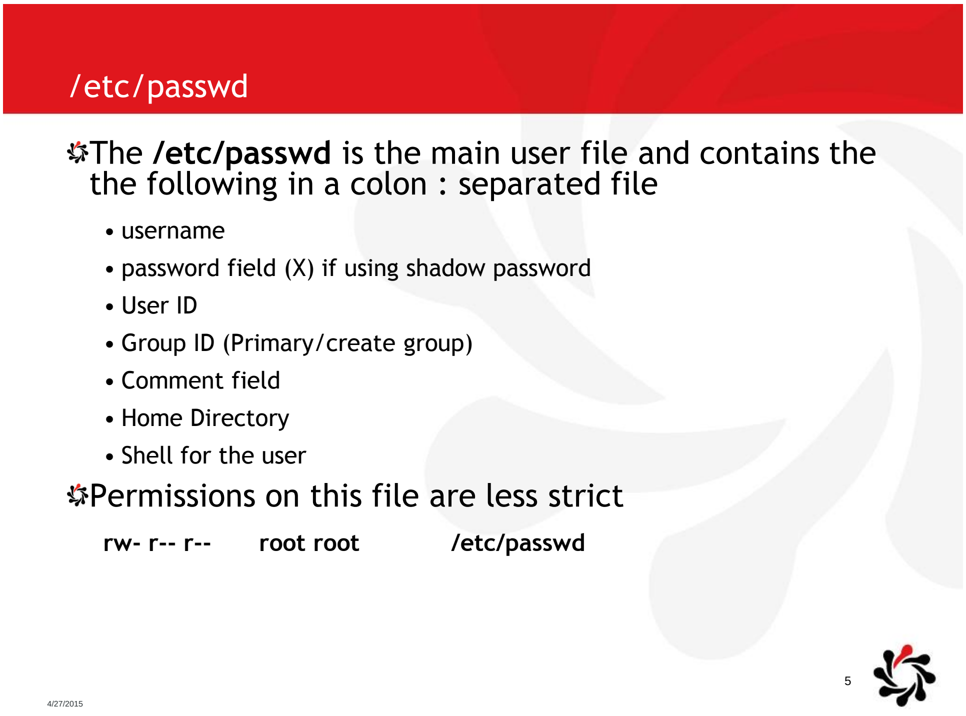
  **/etc/passwd**

  **/etc/shadow**

  **/etc/group**

4

4/27/2015



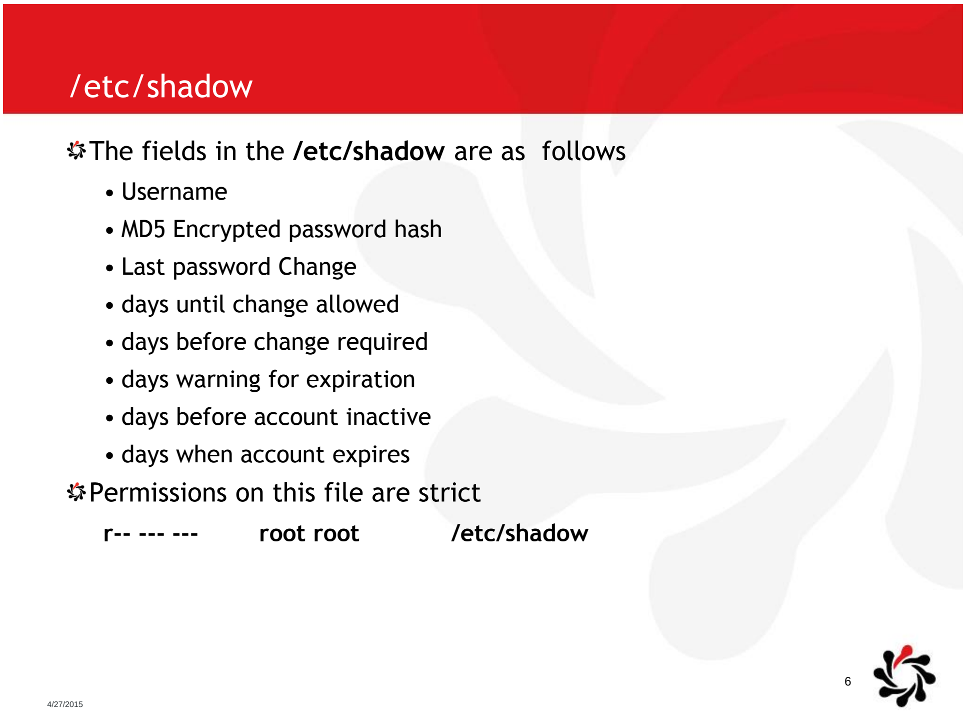
## /etc/passwd

- The **/etc/passwd** is the main user file and contains the the following in a colon : separated file
  - username
  - password field (X) if using shadow password
  - User ID
  - Group ID (Primary/create group)
  - Comment field
  - Home Directory
  - Shell for the user
- Permissions on this file are less strict

**rw- r-- r--      root root      /etc/passwd**

5

4/27/2015



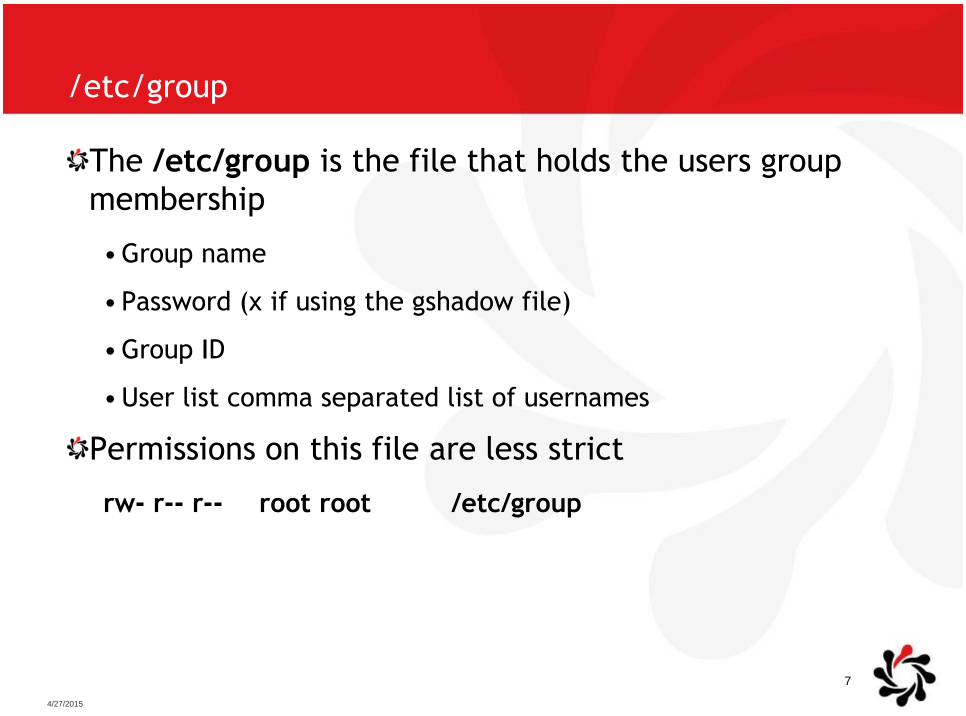
## /etc/shadow

- The fields in the **/etc/shadow** are as follows
  - Username
  - MD5 Encrypted password hash
  - Last password Change
  - days until change allowed
  - days before change required
  - days warning for expiration
  - days before account inactive
  - days when account expires
- Permissions on this file are strict

**r-- --- ---      root root      /etc/shadow**

6

4/27/2015



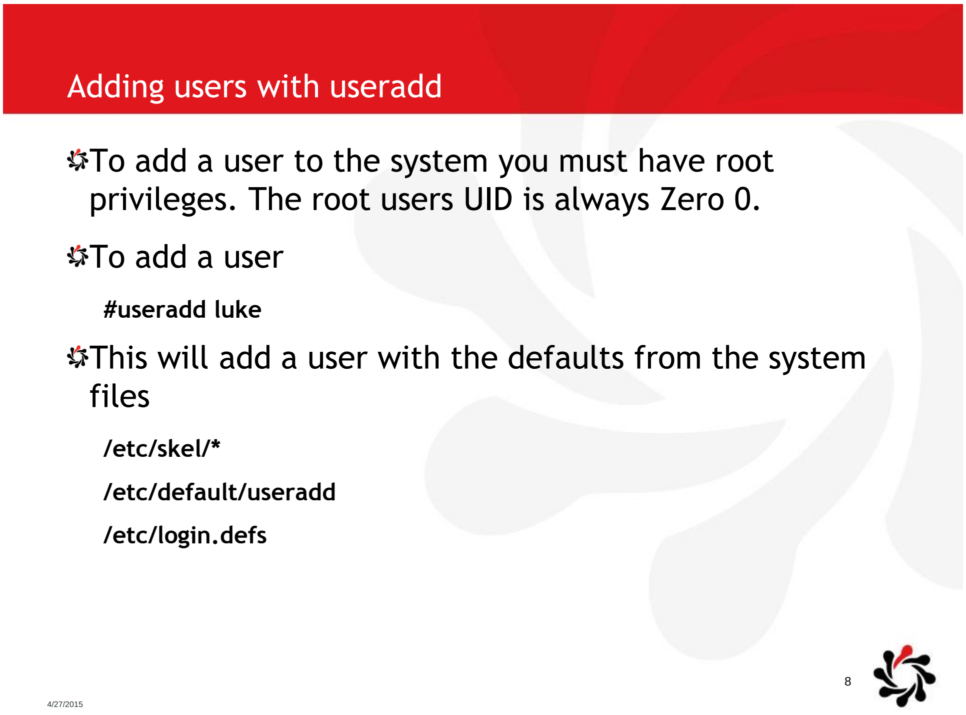
## /etc/group

- The **/etc/group** is the file that holds the users group membership
  - Group name
  - Password (x if using the gshadow file)
  - Group ID
  - User list comma separated list of usernames
- Permissions on this file are less strict

**rw- r-- r--     root root          /etc/group**

4/27/2015

7



## Adding users with useradd

- To add a user to the system you must have root privileges. The root users UID is always Zero 0.
- To add a user

**#useradd luke**

- This will add a user with the defaults from the system files

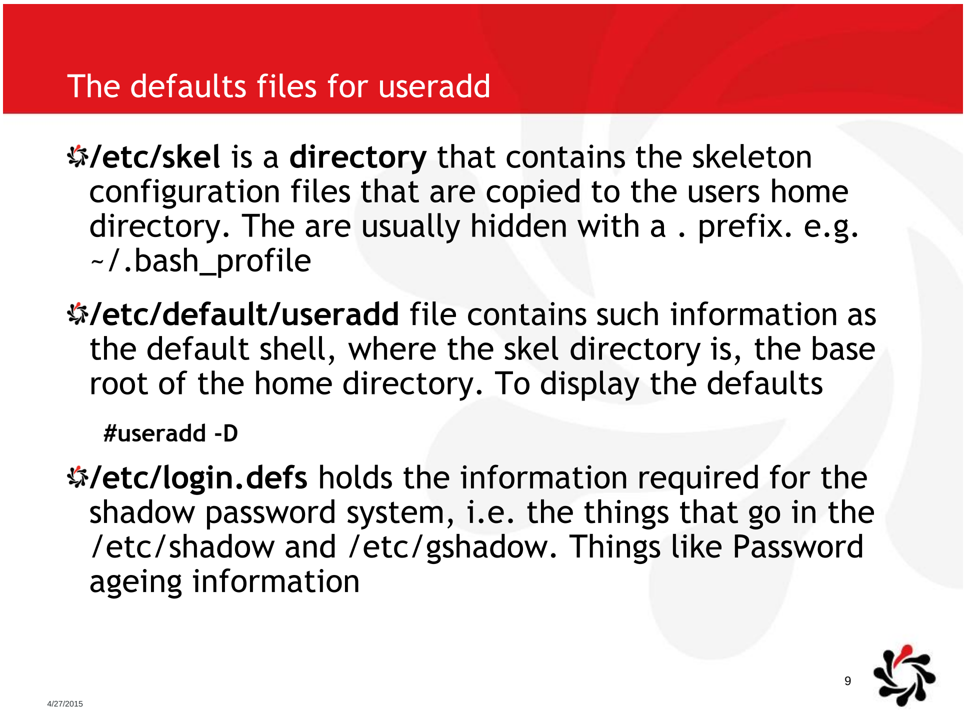
**/etc/skel/***

**/etc/default/useradd**

**/etc/login.defs**

4/27/2015

8



## The defaults files for useradd

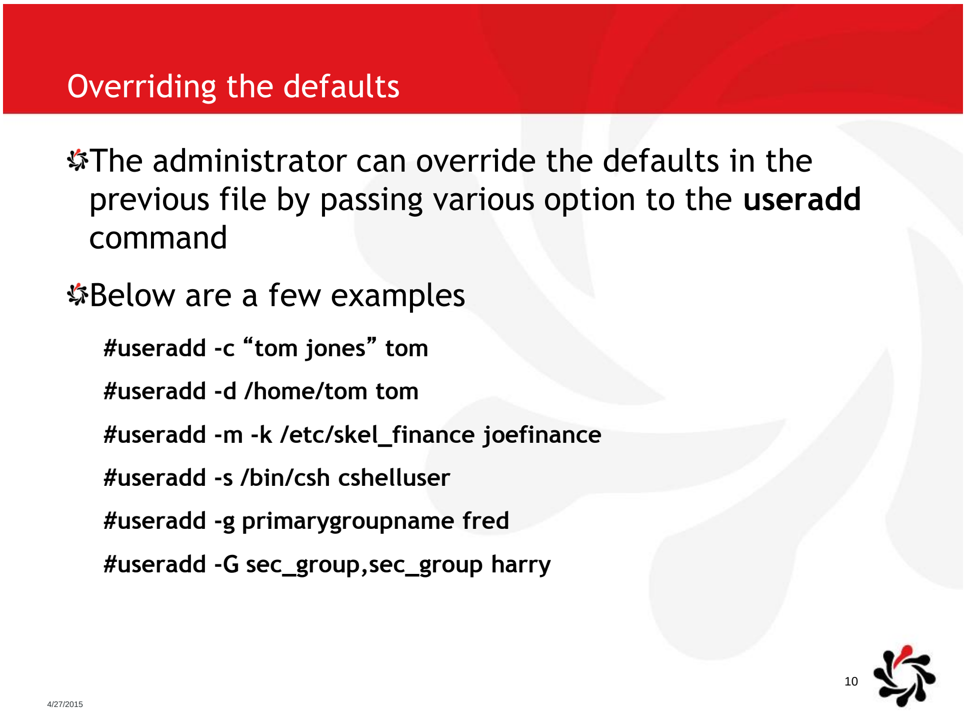
- **/etc/skel** is a **directory** that contains the skeleton configuration files that are copied to the users home directory. The are usually hidden with a . prefix. e.g. ~/.bash_profile
- **/etc/default/useradd** file contains such information as the default shell, where the skel directory is, the base root of the home directory. To display the defaults

```
#useradd -D
```

- **/etc/login.defs** holds the information required for the shadow password system, i.e. the things that go in the /etc/shadow and /etc/gshadow. Things like Password ageing information

9

4/27/2015



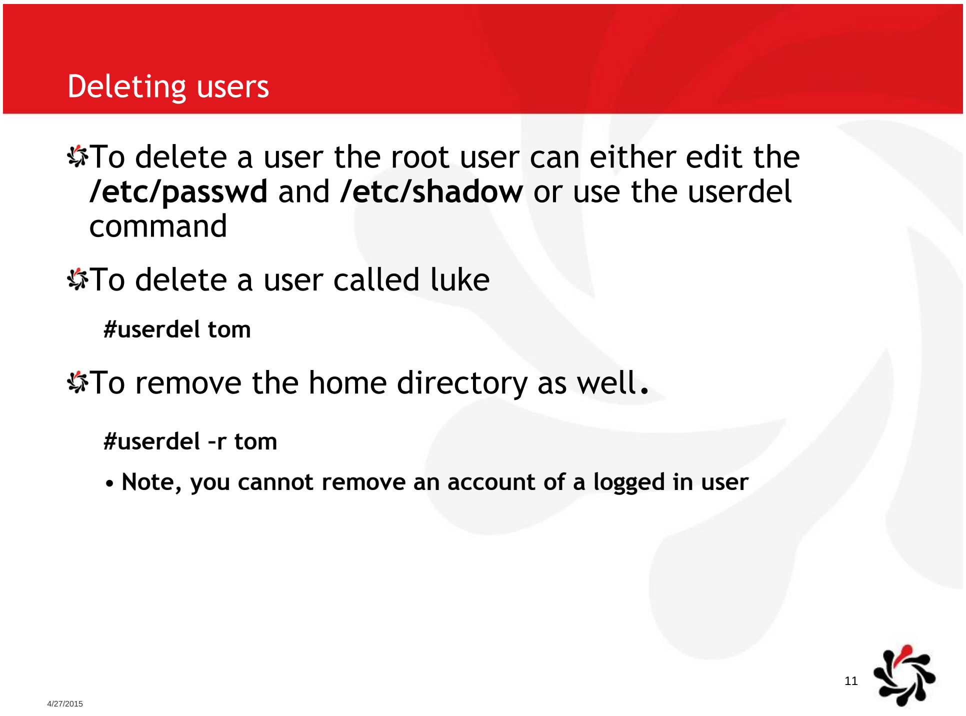
## Overriding the defaults

- The administrator can override the defaults in the previous file by passing various option to the **useradd** command
- Below are a few examples

```
#useradd -c “tom jones” tom
#useradd -d /home/tom tom
#useradd -m -k /etc/skel_finance joefinance
#useradd -s /bin/csh cshelluser
#useradd -g primarygroupname fred
#useradd -G sec_group,sec_group harry
```

10

4/27/2015



## Deleting users

- To delete a user the root user can either edit the **/etc/passwd** and **/etc/shadow** or use the userdel command
- To delete a user called luke

```
#userdel tom
```

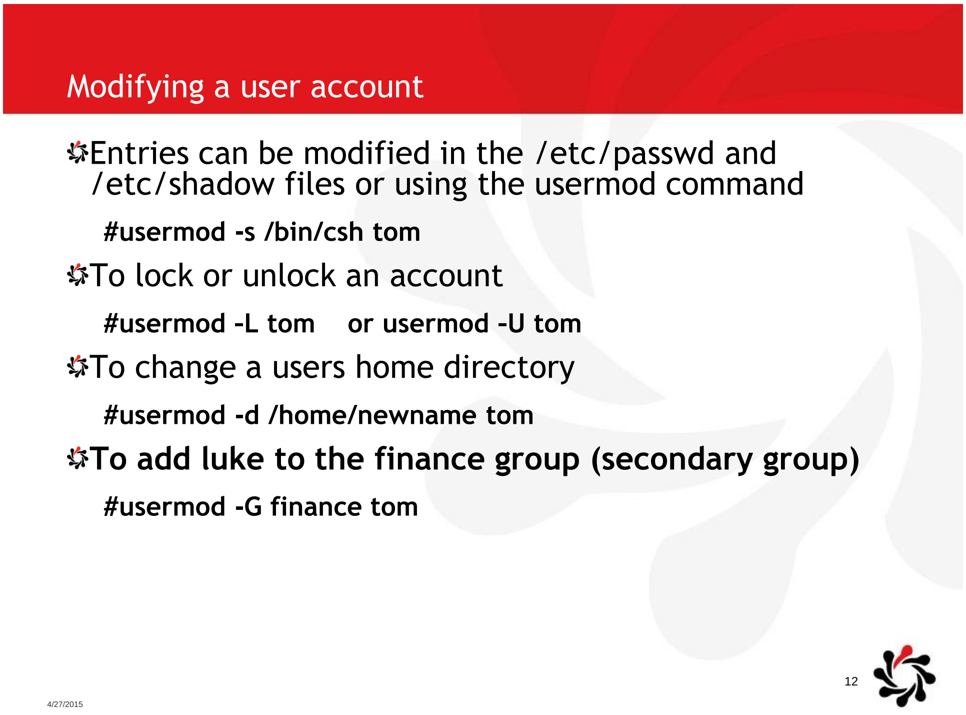
- To remove the home directory as well.

```
#userdel -r tom
```

  - **Note, you cannot remove an account of a logged in user**

11

4/27/2015



## Modifying a user account

- Entries can be modified in the /etc/passwd and /etc/shadow files or using the usermod command

```
#usermod -s /bin/csh tom
```

- To lock or unlock an account

```
#usermod -L tom     or usermod -U tom
```

- To change a users home directory

```
#usermod -d /home/newname tom
```

- **To add luke to the finance group (secondary group)**

```
#usermod -G finance tom
```

12

4/27/2015

## Group management

- The users primary group (-g) is stored in the /etc/passwd in 4th field, commonly known as the creation group. This is a GID value which relates to the file /etc/group. The secondary groups (-G) are access groups. There is a group shadow password file /etc/gshadow
- To add a group to the /etc/group and /etc/gshadow use groupadd

```
#groupadd users
```

- To add a group called finance with a GID of 9999

```
#groupadd -g 9999 finance
```

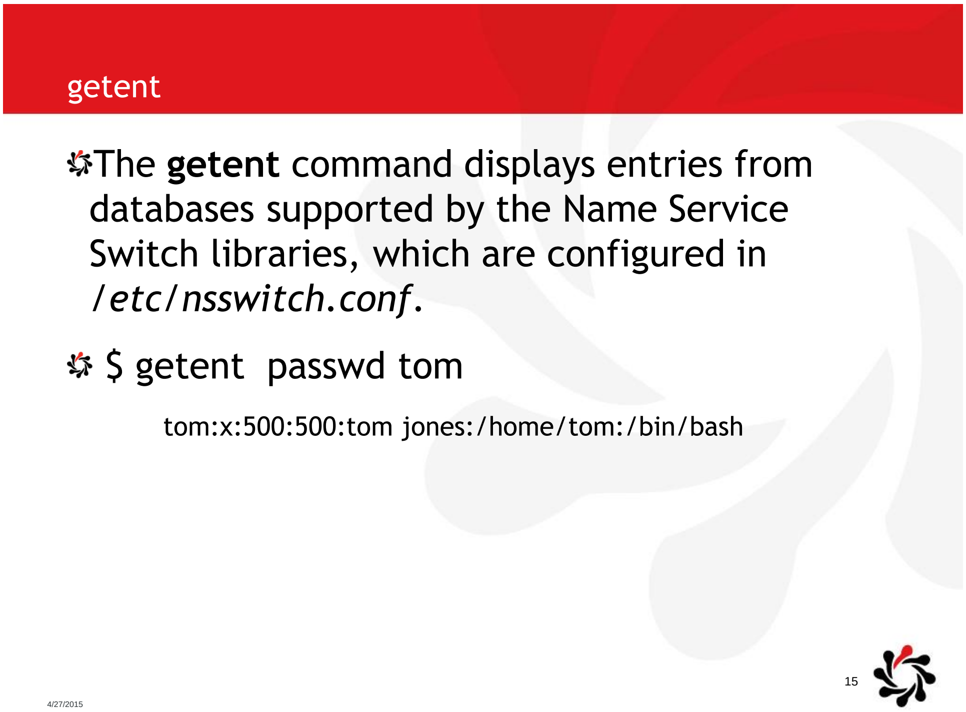
## Modifying and deleting groups

- To modify a group use groupmod

```
#groupmod -n newgroupname oldgroupname
```

- To delete a group from the **/etc/group**

```
#groupdel finance
```



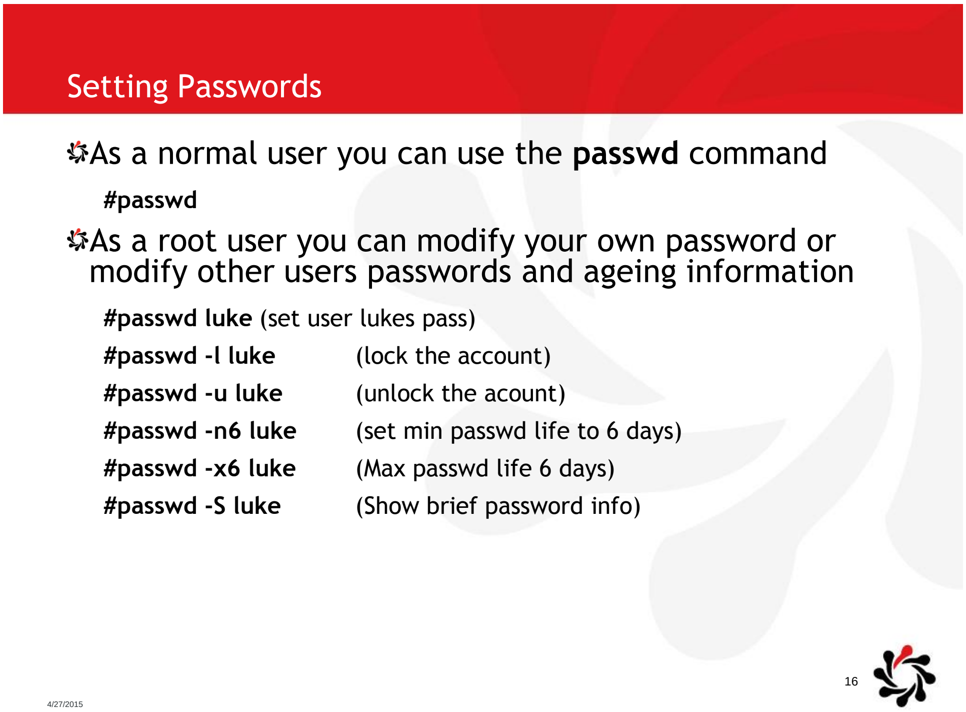
## getent

- The **getent** command displays entries from databases supported by the Name Service Switch libraries, which are configured in */etc/nsswitch.conf*.
- $ getent  passwd tom

```
tom:x:500:500:tom jones:/home/tom:/bin/bash
```

15

4/27/2015



## Setting Passwords

- As a normal user you can use the **passwd** command

  **#passwd**

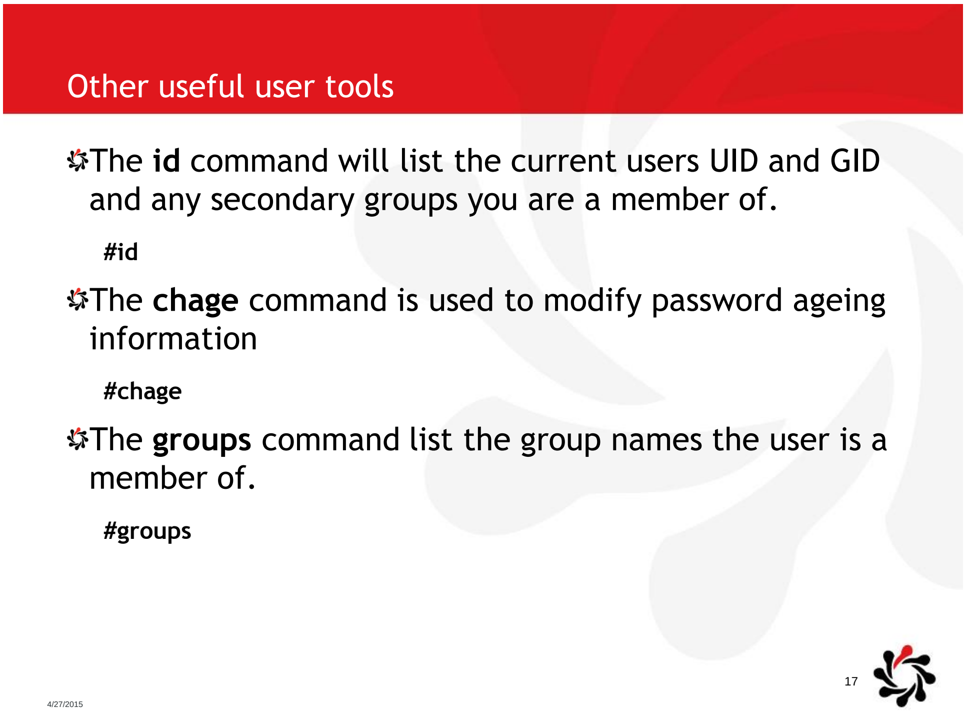
- As a root user you can modify your own password or modify other users passwords and ageing information

  **#passwd luke** (set user lukes pass)

```
#passwd -l luke        (lock the account)
#passwd -u luke        (unlock the acount)
#passwd -n6 luke       (set min passwd life to 6 days)
#passwd -x6 luke       (Max passwd life 6 days)
#passwd -S luke        (Show brief password info)
```

16

4/27/2015



## Other useful user tools

- The **id** command will list the current users UID and GID and any secondary groups you are a member of.

```
#id
```

- The **chage** command is used to modify password ageing information

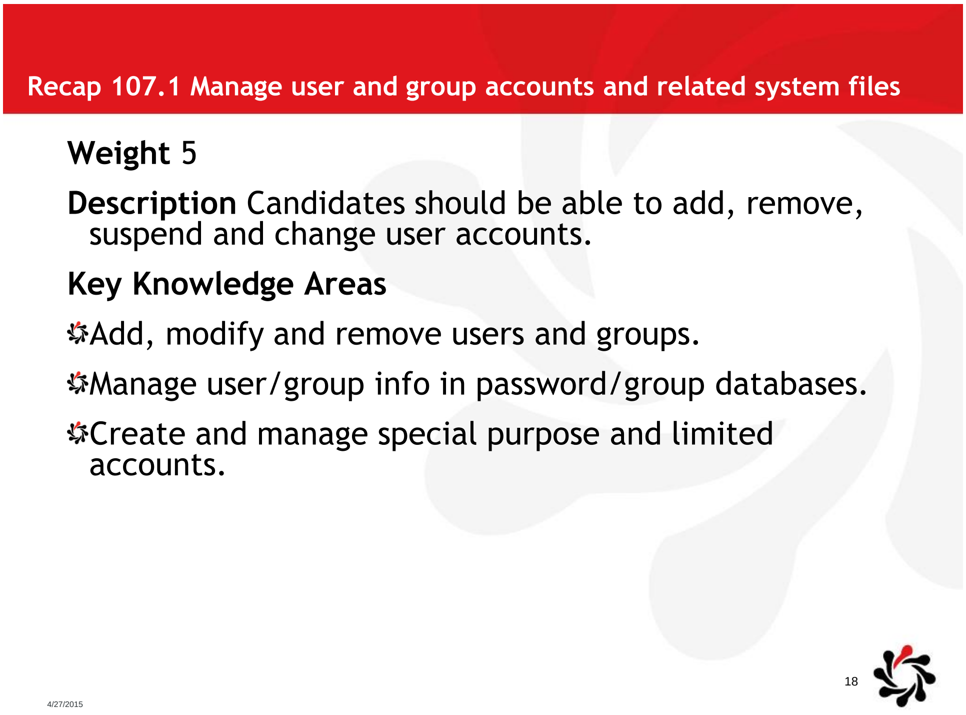
```
#chage
```

- The **groups** command list the group names the user is a member of.

```
#groups
```

4/27/2015

17



## Recap 107.1 Manage user and group accounts and related system files

**Weight** 5

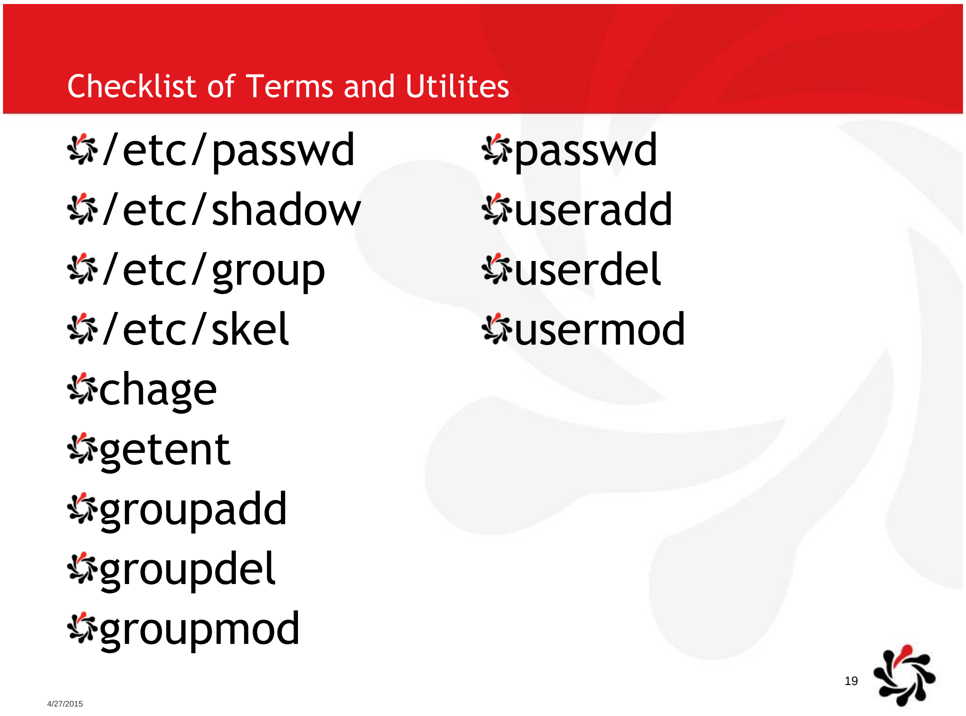
**Description** Candidates should be able to add, remove, suspend and change user accounts.

**Key Knowledge Areas**

- Add, modify and remove users and groups.
- Manage user/group info in password/group databases.
- Create and manage special purpose and limited accounts.

4/27/2015

18

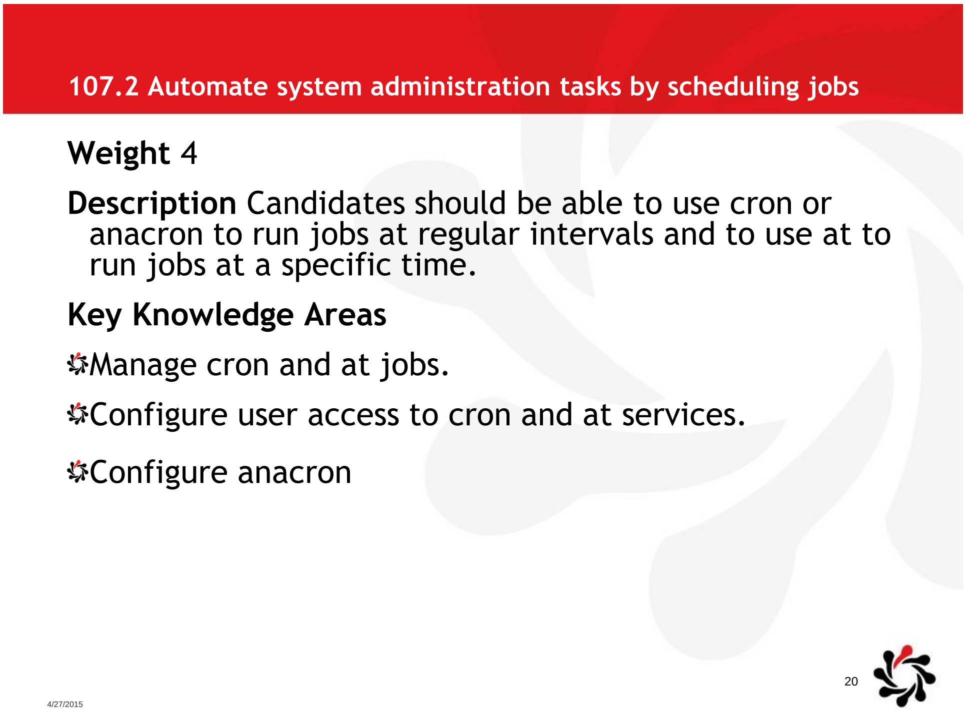


## Checklist of Terms and Utilites

- /etc/passwd
- /etc/shadow
- /etc/group
- /etc/skel
- chage
- getent
- groupadd
- groupdel
- groupmod
- passwd
- useradd
- userdel
- usermod

4/27/2015

19



## 107.2 Automate system administration tasks by scheduling jobs

**Weight** 4

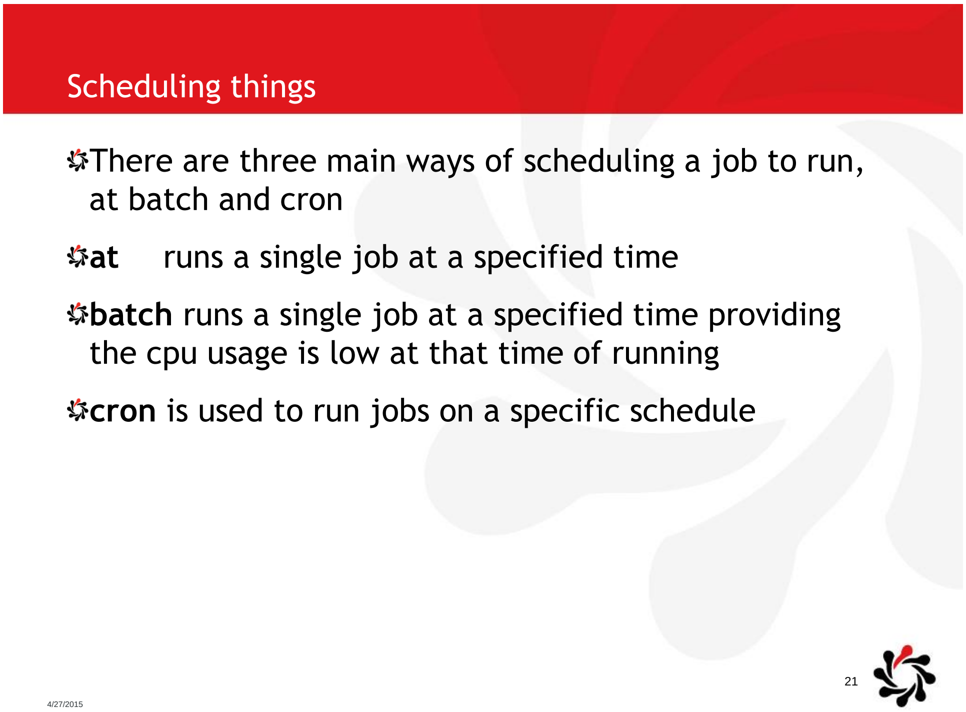
**Description** Candidates should be able to use cron or anacron to run jobs at regular intervals and to use at to run jobs at a specific time.

**Key Knowledge Areas**

- Manage cron and at jobs.
- Configure user access to cron and at services.
- Configure anacron

4/27/2015

20

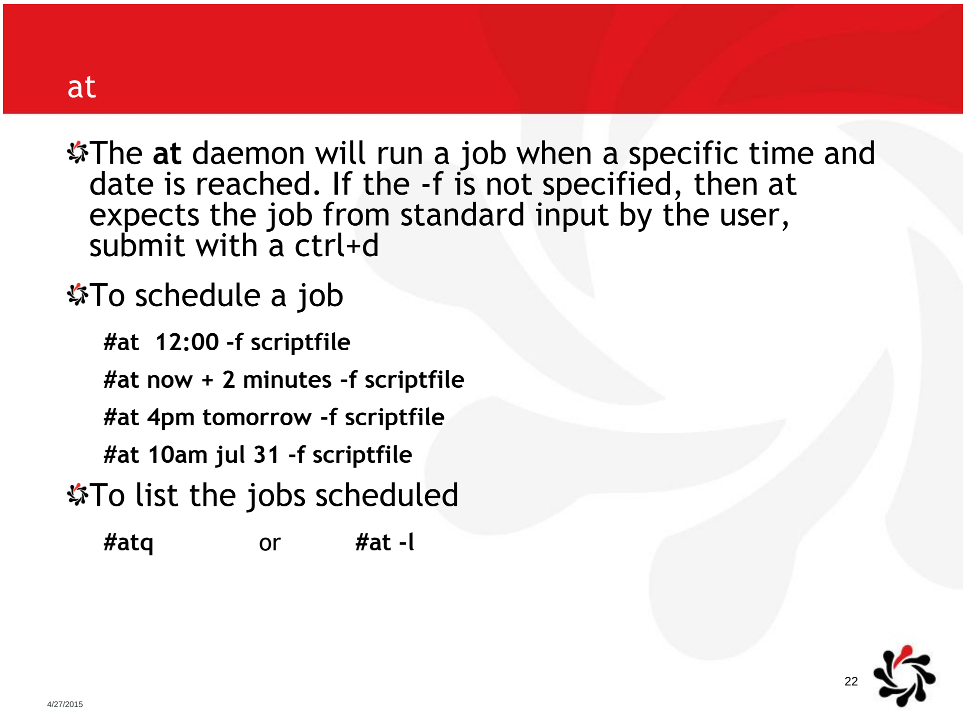


## Scheduling things

- There are three main ways of scheduling a job to run, at batch and cron
- **at** runs a single job at a specified time
- **batch** runs a single job at a specified time providing the cpu usage is low at that time of running
- **cron** is used to run jobs on a specific schedule

4/27/2015

21



## at

- The **at** daemon will run a job when a specific time and date is reached. If the -f is not specified, then at expects the job from standard input by the user, submit with a ctrl+d
- To schedule a job

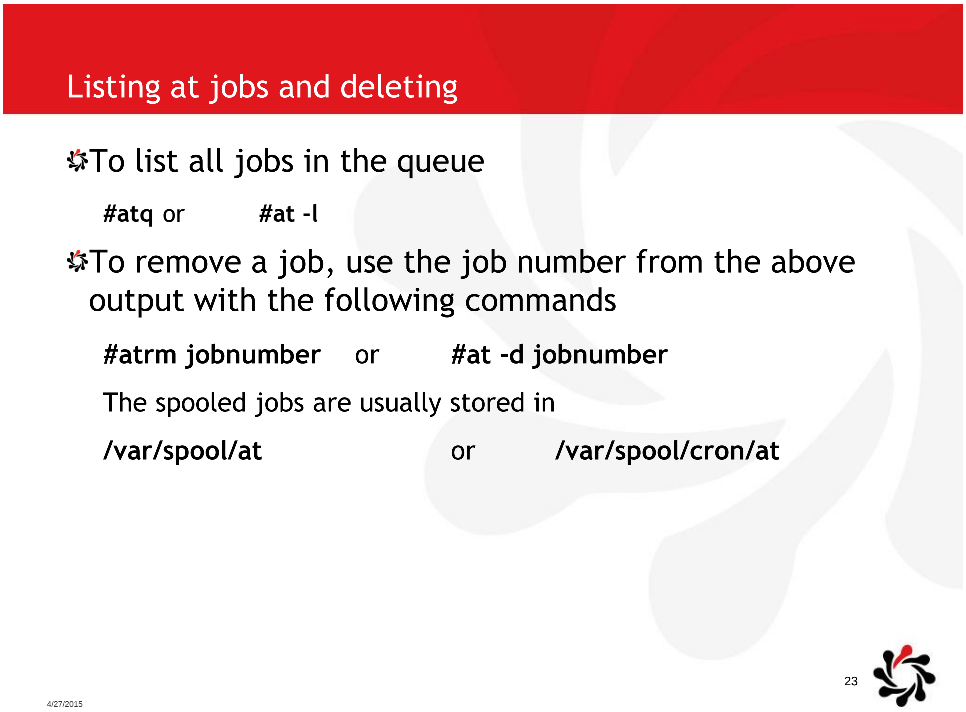
```
#at  12:00 -f scriptfile
#at now + 2 minutes -f scriptfile
#at 4pm tomorrow -f scriptfile
#at 10am jul 31 -f scriptfile
```

- To list the jobs scheduled

```
#atq            or          #at -l
```

4/27/2015

22



## Listing at jobs and deleting

- To list all jobs in the queue

  **#atq** or **#at -l**

- To remove a job, use the job number from the above output with the following commands

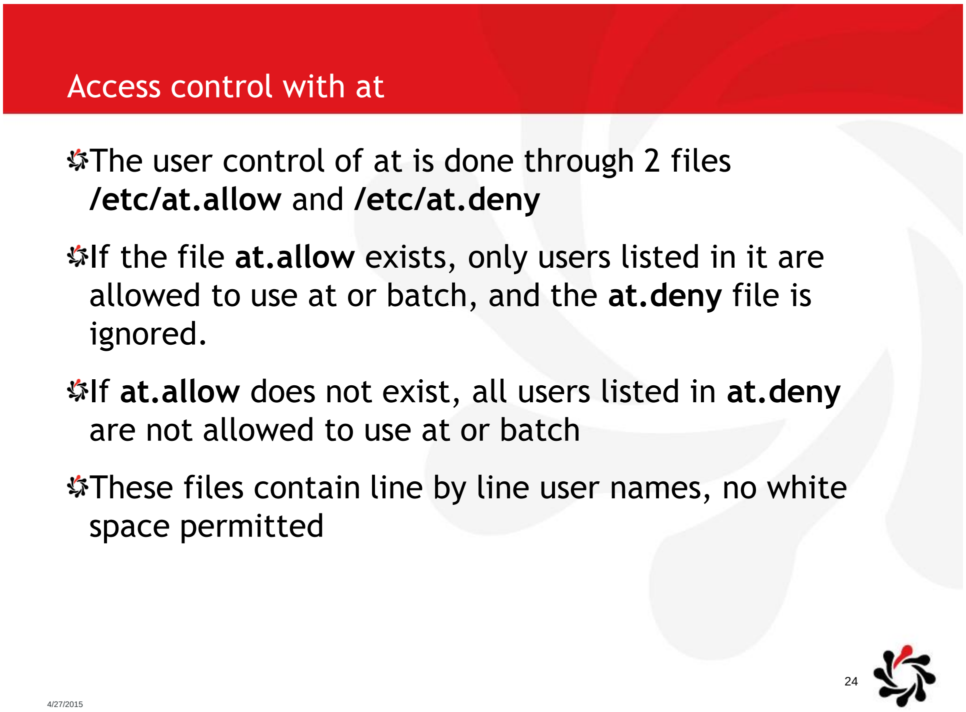
  **#atrm jobnumber** or **#at -d jobnumber**

  The spooled jobs are usually stored in

  **/var/spool/at** or **/var/spool/cron/at**

4/27/2015

23

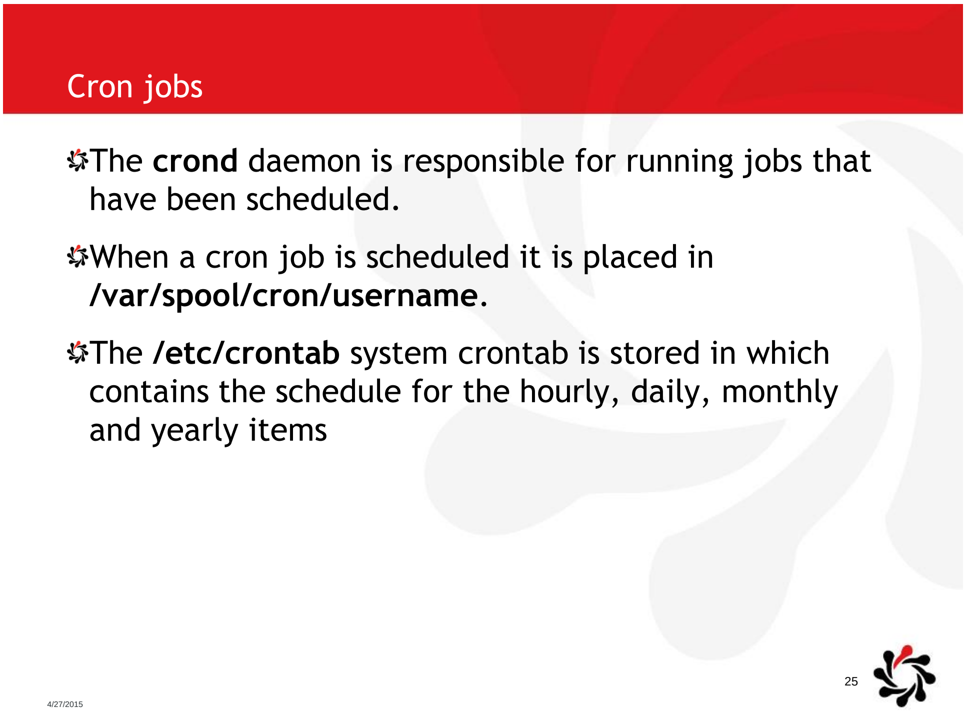


## Access control with at

- The user control of at is done through 2 files **/etc/at.allow** and **/etc/at.deny**
- If the file **at.allow** exists, only users listed in it are allowed to use at or batch, and the **at.deny** file is ignored.
- If **at.allow** does not exist, all users listed in **at.deny** are not allowed to use at or batch
- These files contain line by line user names, no white space permitted

4/27/2015

24

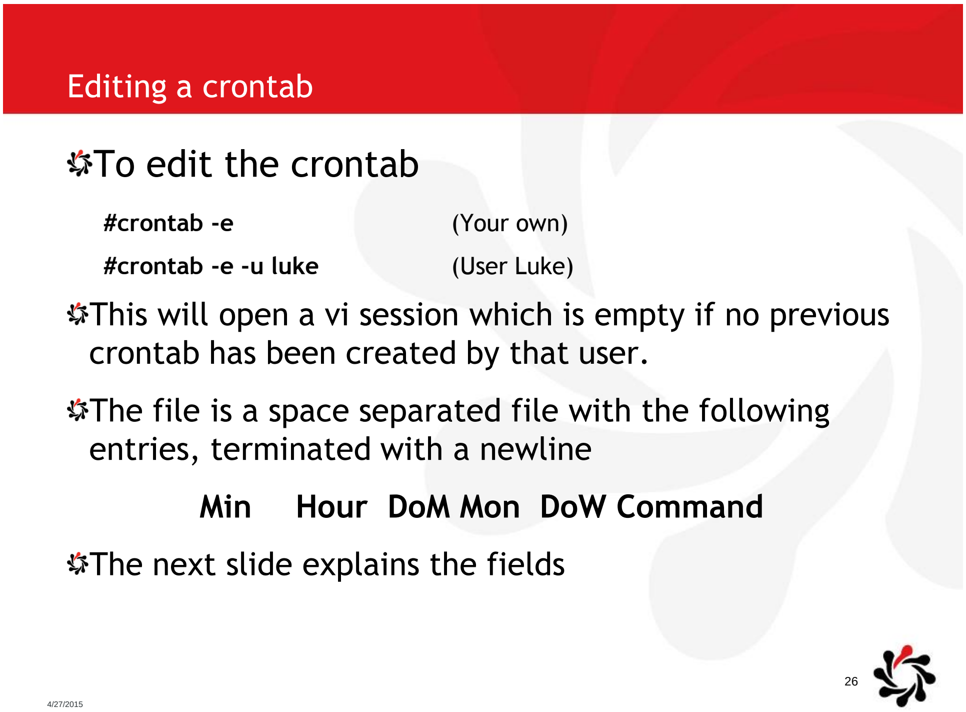


## Cron jobs

- The **crond** daemon is responsible for running jobs that have been scheduled.
- When a cron job is scheduled it is placed in **/var/spool/cron/username**.
- The **/etc/crontab** system crontab is stored in which contains the schedule for the hourly, daily, monthly and yearly items

4/27/2015

25



## Editing a crontab

- To edit the crontab

  **#crontab -e** (Your own)

  **#crontab -e -u luke** (User Luke)

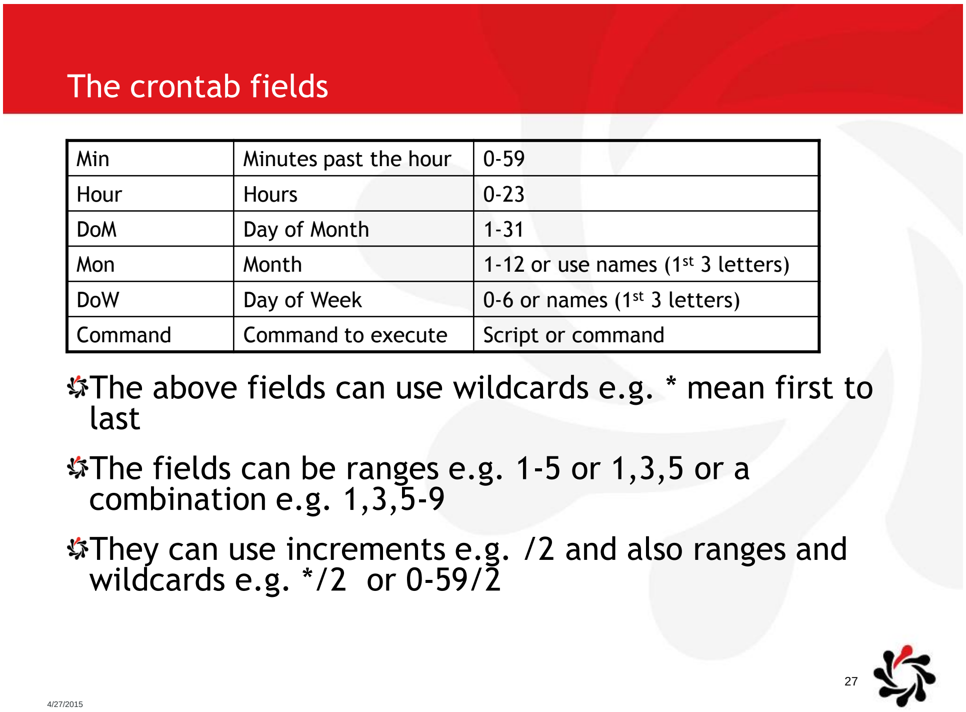
- This will open a vi session which is empty if no previous crontab has been created by that user.
- The file is a space separated file with the following entries, terminated with a newline

  **Min Hour DoM Mon DoW Command**

- The next slide explains the fields

4/27/2015

26



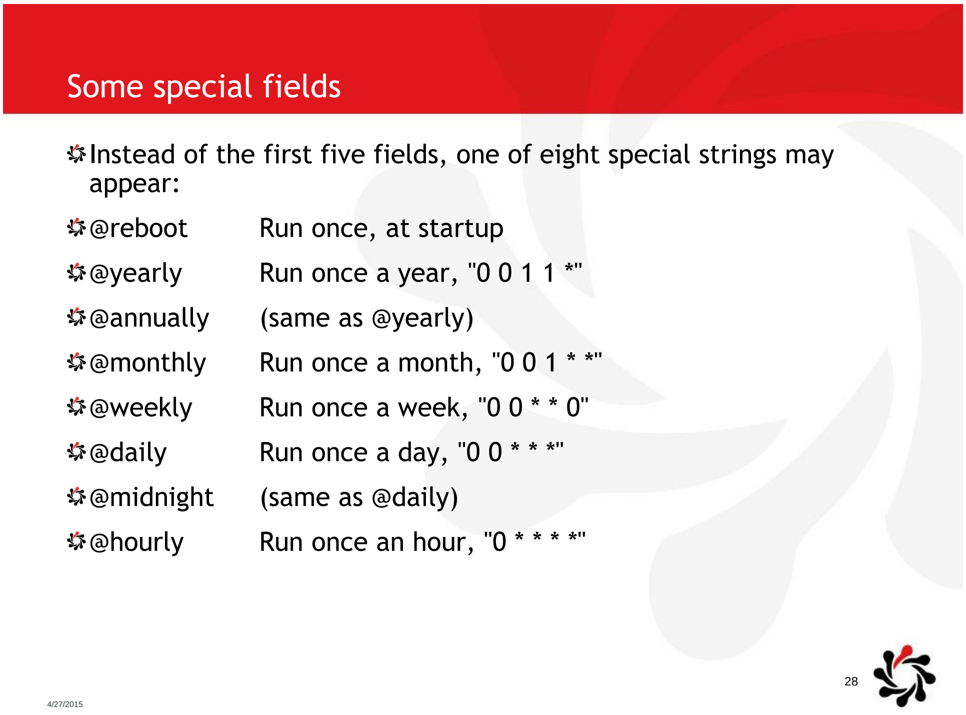
## The crontab fields

| | | |
|---|---|---|
| Min | Minutes past the hour | 0-59 |
| Hour | Hours | 0-23 |
| DoM | Day of Month | 1-31 |
| Mon | Month | 1-12 or use names (1st 3 letters) |
| DoW | Day of Week | 0-6 or names (1st 3 letters) |
| Command | Command to execute | Script or command |

- The above fields can use wildcards e.g. * mean first to last
- The fields can be ranges e.g. 1-5 or 1,3,5 or a combination e.g. 1,3,5-9
- They can use increments e.g. /2 and also ranges and wildcards e.g. */2  or 0-59/2

4/27/2015

27



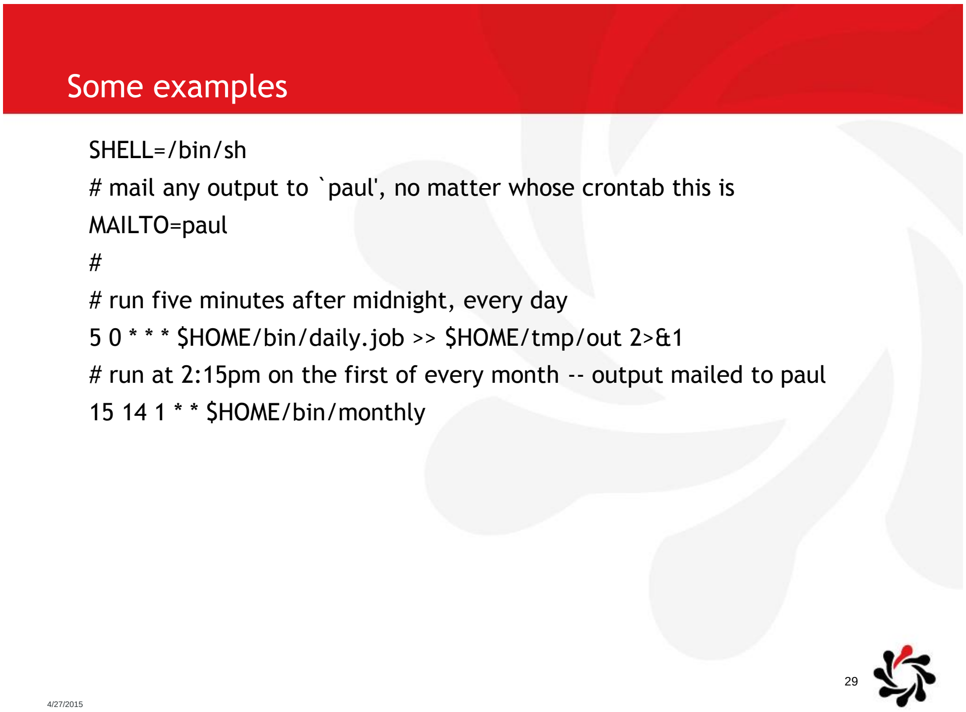
## Some special fields

- Instead of the first five fields, one of eight special strings may appear:

| | |
|---|---|
| @reboot | Run once, at startup |
| @yearly | Run once a year, "0 0 1 1 *" |
| @annually | (same as @yearly) |
| @monthly | Run once a month, "0 0 1 * *" |
| @weekly | Run once a week, "0 0 * * 0" |
| @daily | Run once a day, "0 0 * * *" |
| @midnight | (same as @daily) |
| @hourly | Run once an hour, "0 * * * *" |

4/27/2015

28

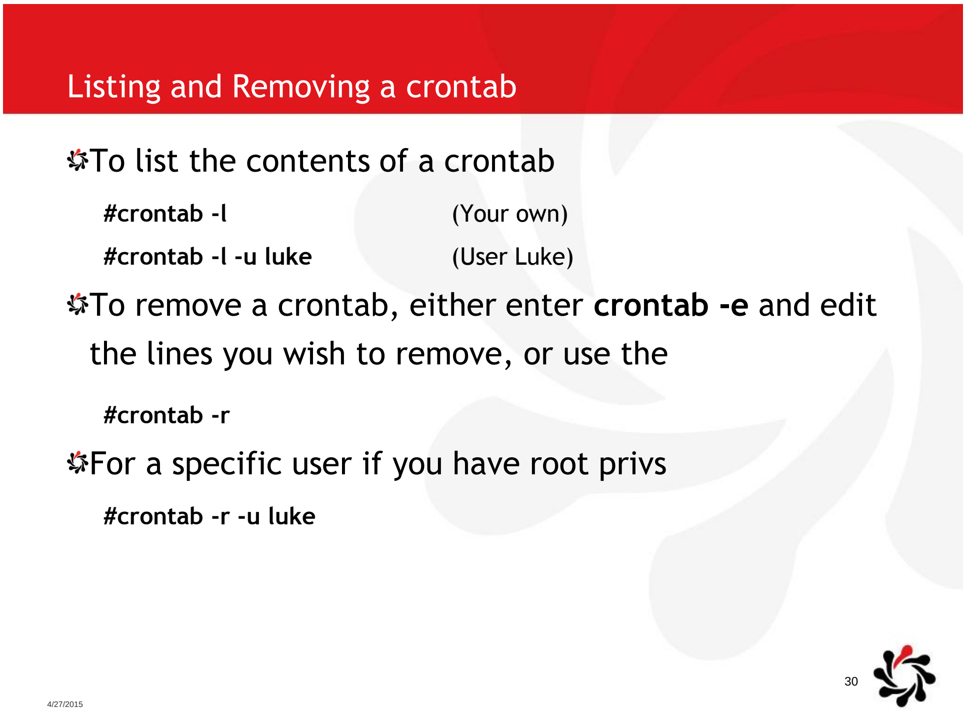


## Some examples

```
SHELL=/bin/sh
# mail any output to `paul', no matter whose crontab this is
MAILTO=paul
#
# run five minutes after midnight, every day
5 0 * * * $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 2:15pm on the first of every month -- output mailed to paul
15 14 1 * * $HOME/bin/monthly
```

29

4/27/2015



## Listing and Removing a crontab

- To list the contents of a crontab

| | |
|---|---|
| **#crontab -l** | (Your own) |
| **#crontab -l -u luke** | (User Luke) |

- To remove a crontab, either enter **crontab -e** and edit the lines you wish to remove, or use the

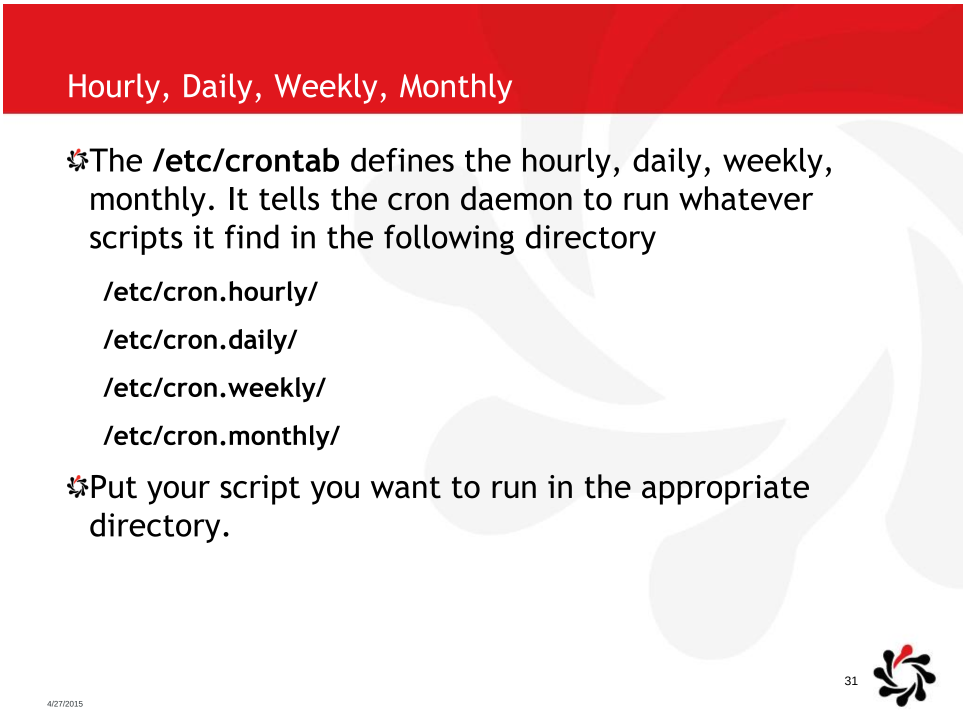
**#crontab -r**

- For a specific user if you have root privs

**#crontab -r -u luke**

30

4/27/2015



## Hourly, Daily, Weekly, Monthly

- The **/etc/crontab** defines the hourly, daily, weekly, monthly. It tells the cron daemon to run whatever scripts it find in the following directory

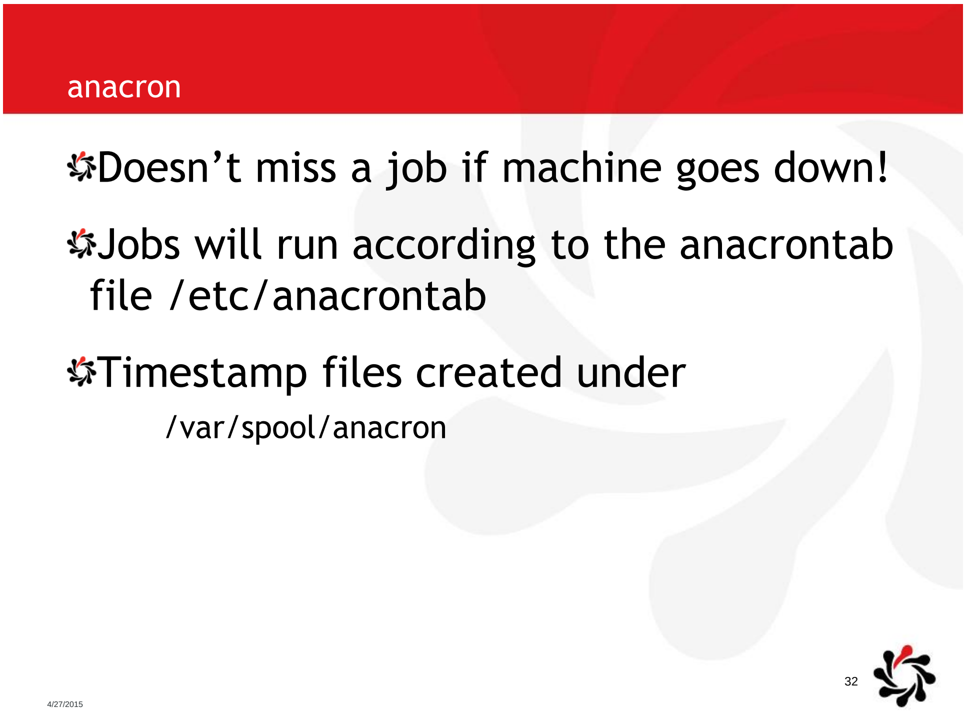
  /etc/cron.hourly/

  /etc/cron.daily/

  /etc/cron.weekly/

  /etc/cron.monthly/

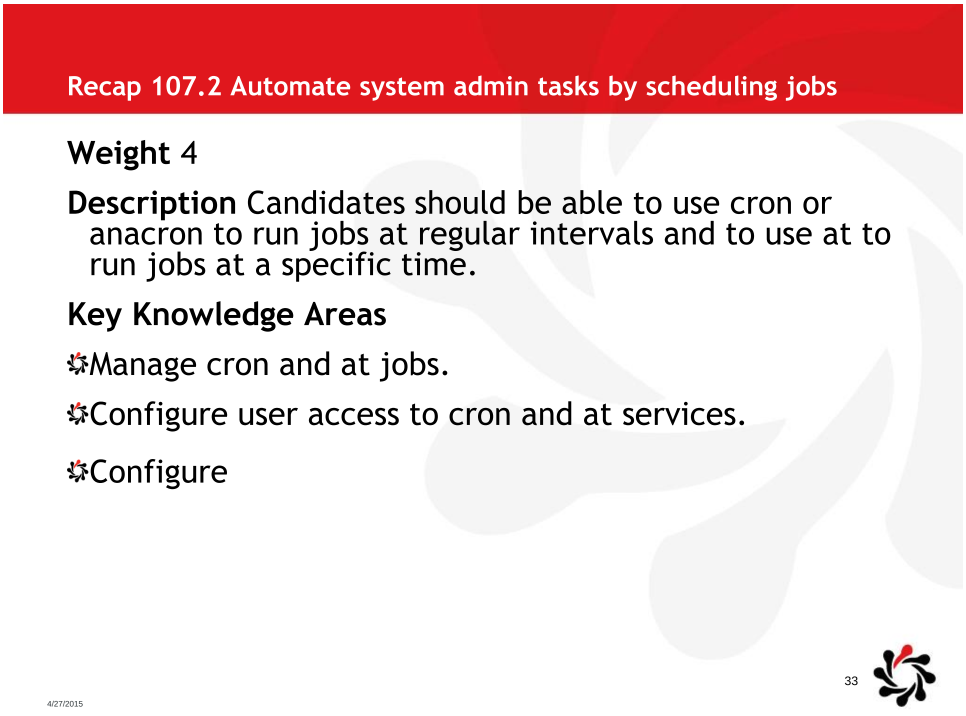
- Put your script you want to run in the appropriate directory.



## anacron

- Doesn’t miss a job if machine goes down!
- Jobs will run according to the anacrontab file /etc/anacrontab
- Timestamp files created under

  /var/spool/anacron



## Recap 107.2 Automate system admin tasks by scheduling jobs

**Weight** 4

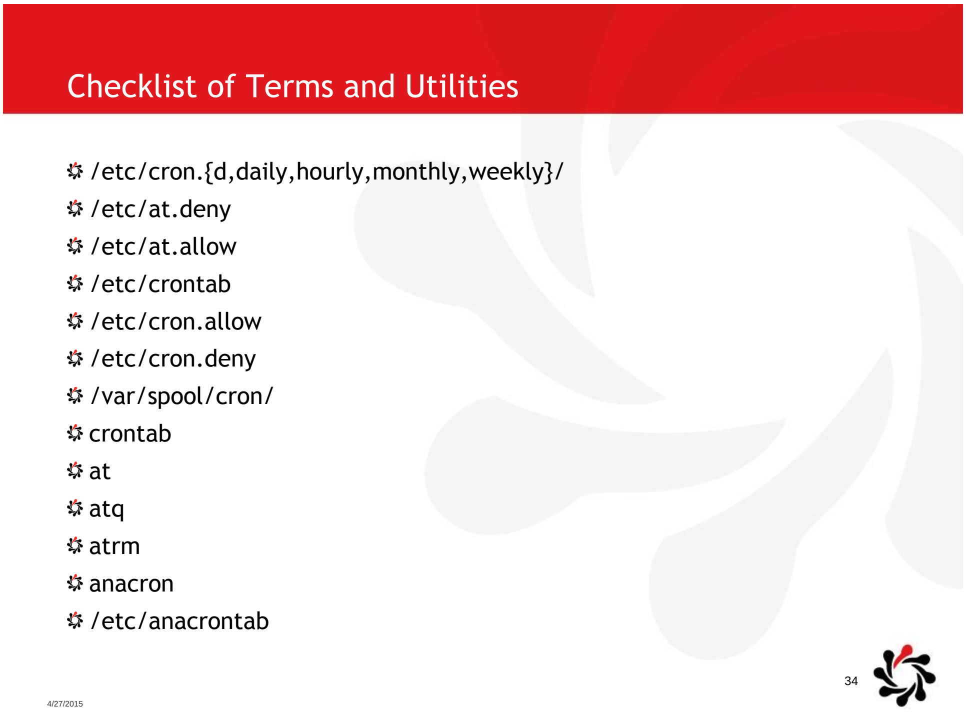
**Description** Candidates should be able to use cron or anacron to run jobs at regular intervals and to use at to run jobs at a specific time.

**Key Knowledge Areas**

- Manage cron and at jobs.
- Configure user access to cron and at services.
- Configure

4/27/2015

33

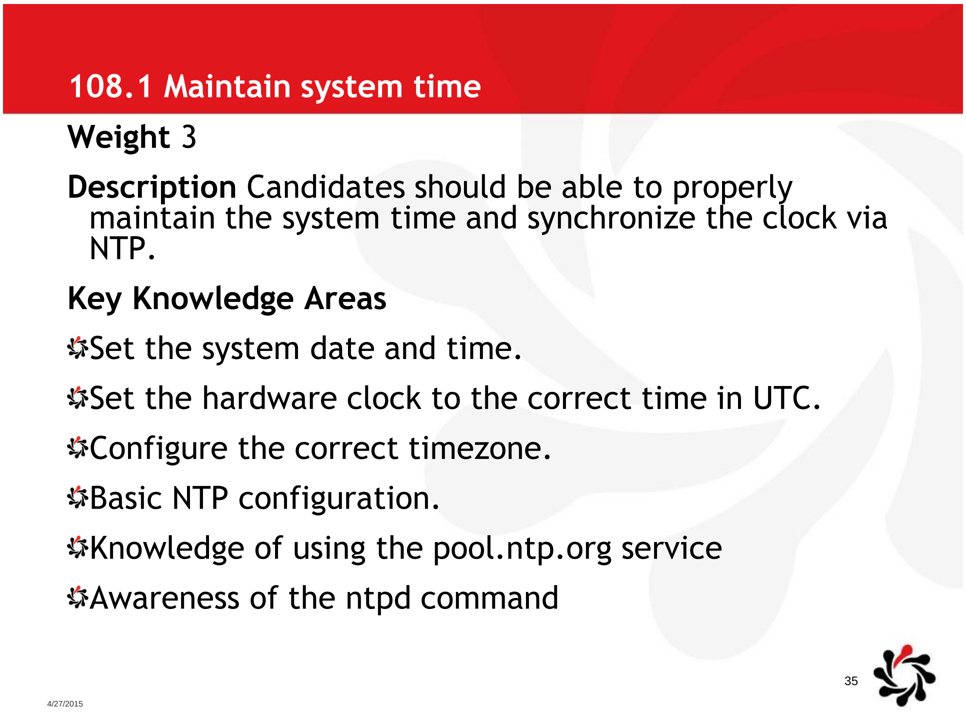


## Checklist of Terms and Utilities

- /etc/cron.{d,daily,hourly,monthly,weekly}/
- /etc/at.deny
- /etc/at.allow
- /etc/crontab
- /etc/cron.allow
- /etc/cron.deny
- /var/spool/cron/
- crontab
- at
- atq
- atrm
- anacron
- /etc/anacrontab

4/27/2015

34



## 108.1 Maintain system time

**Weight** 3

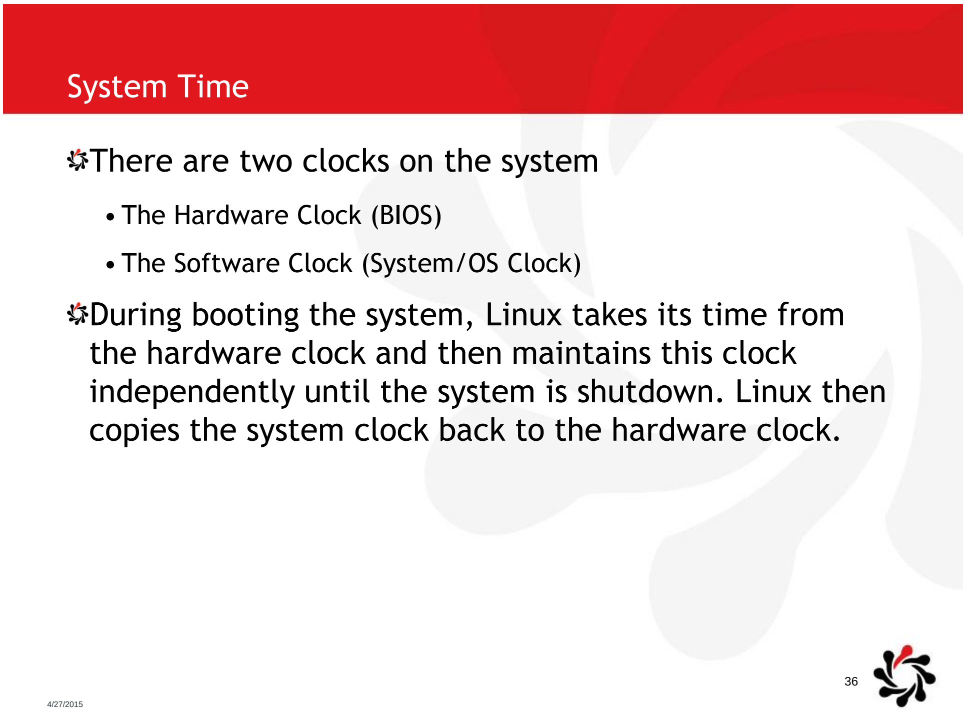
**Description** Candidates should be able to properly maintain the system time and synchronize the clock via NTP.

**Key Knowledge Areas**

- Set the system date and time.
- Set the hardware clock to the correct time in UTC.
- Configure the correct timezone.
- Basic NTP configuration.
- Knowledge of using the pool.ntp.org service
- Awareness of the ntpd command

35

4/27/2015

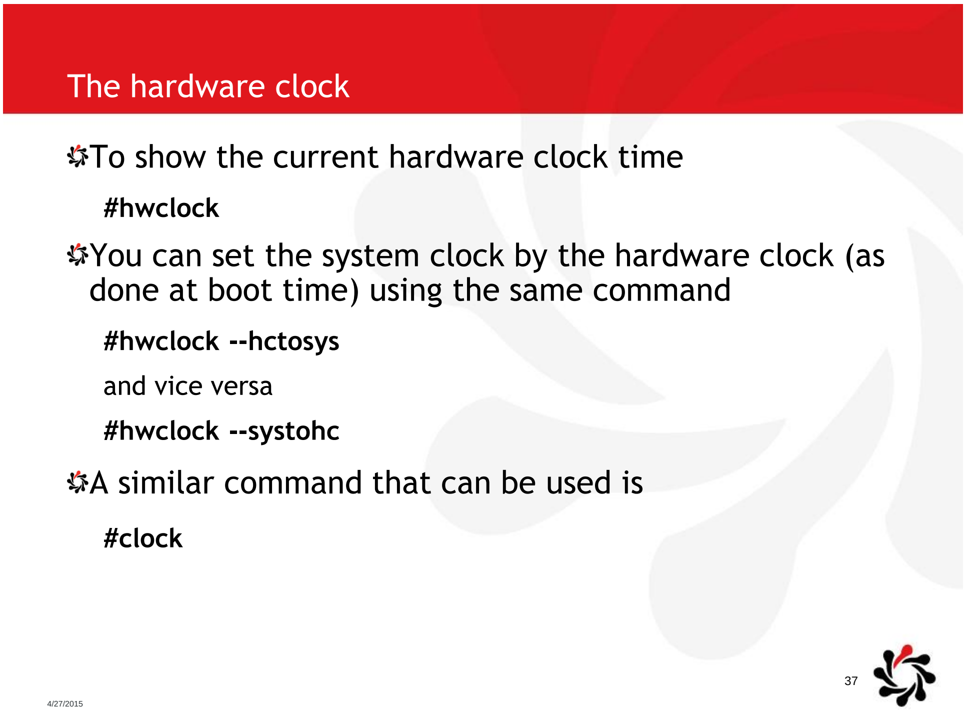


## System Time

- There are two clocks on the system
  - The Hardware Clock (BIOS)
  - The Software Clock (System/OS Clock)
- During booting the system, Linux takes its time from the hardware clock and then maintains this clock independently until the system is shutdown. Linux then copies the system clock back to the hardware clock.

36

4/27/2015



## The hardware clock

- To show the current hardware clock time

```
#hwclock
```

- You can set the system clock by the hardware clock (as done at boot time) using the same command

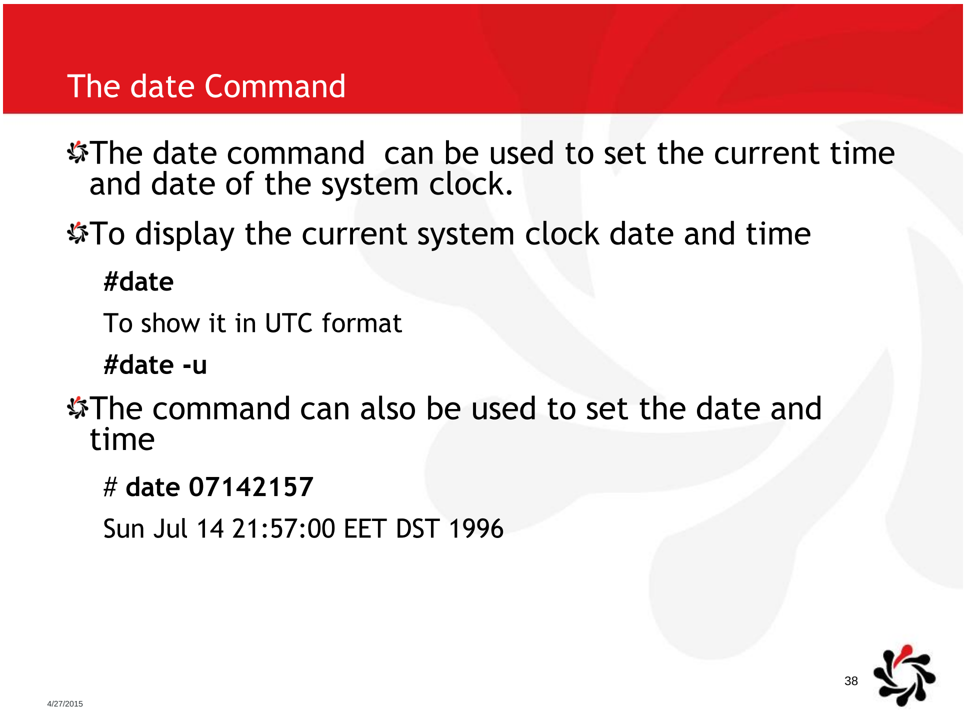
```
#hwclock --hctosys
```

and vice versa

```
#hwclock --systohc
```

- A similar command that can be used is

```
#clock
```



## The date Command

- The date command  can be used to set the current time and date of the system clock.
- To display the current system clock date and time

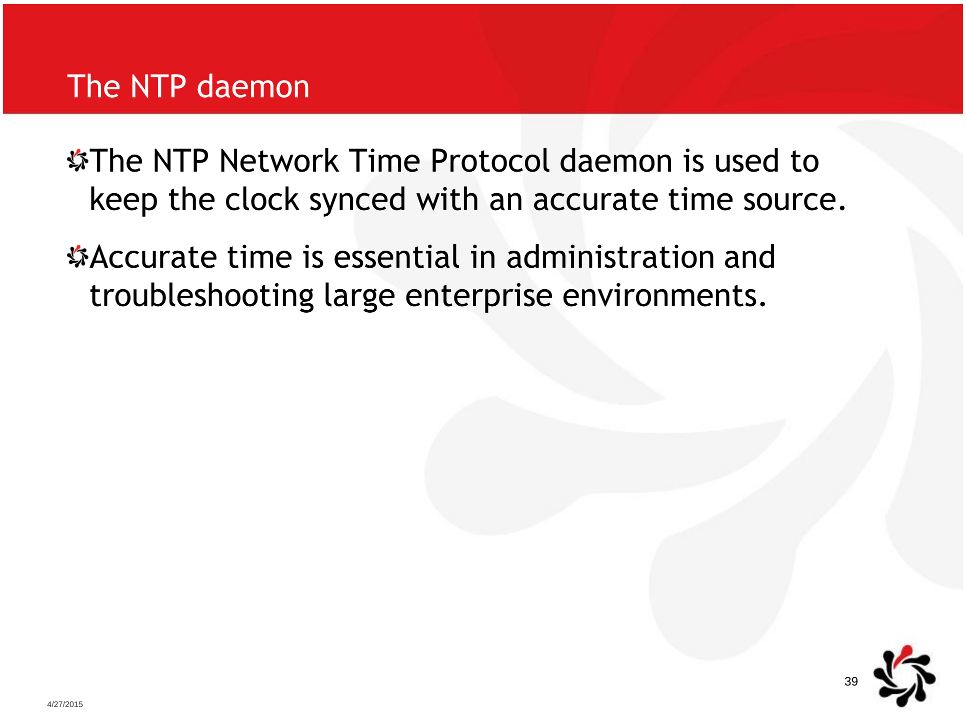
```
#date
```

To show it in UTC format

```
#date -u
```

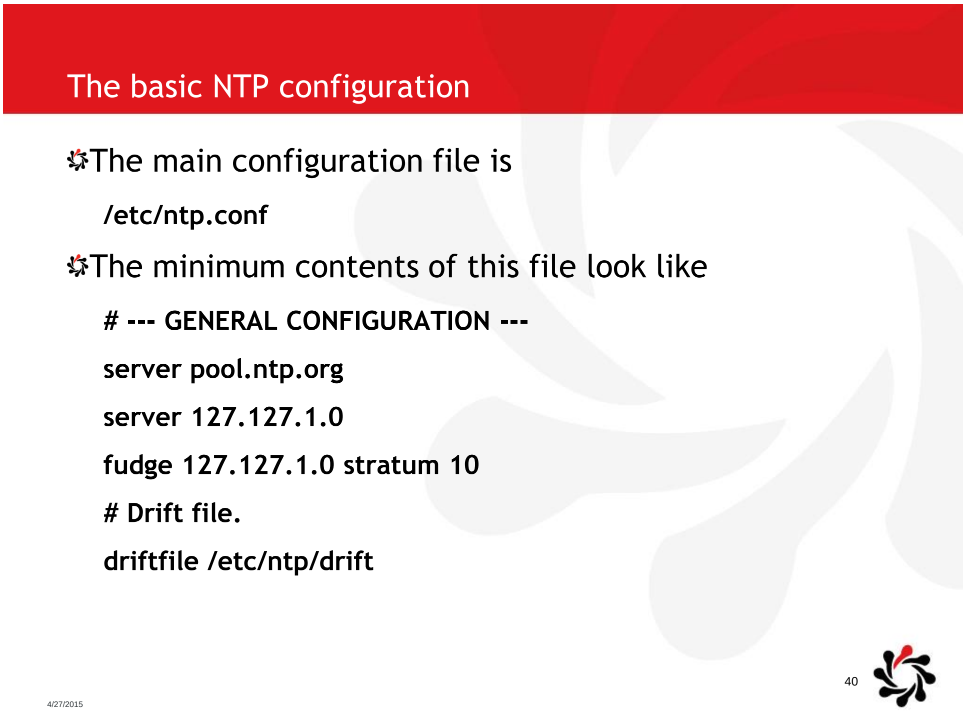
- The command can also be used to set the date and time

```
# date 07142157
Sun Jul 14 21:57:00 EET DST 1996
```



## The NTP daemon

- The NTP Network Time Protocol daemon is used to keep the clock synced with an accurate time source.
- Accurate time is essential in administration and troubleshooting large enterprise environments.

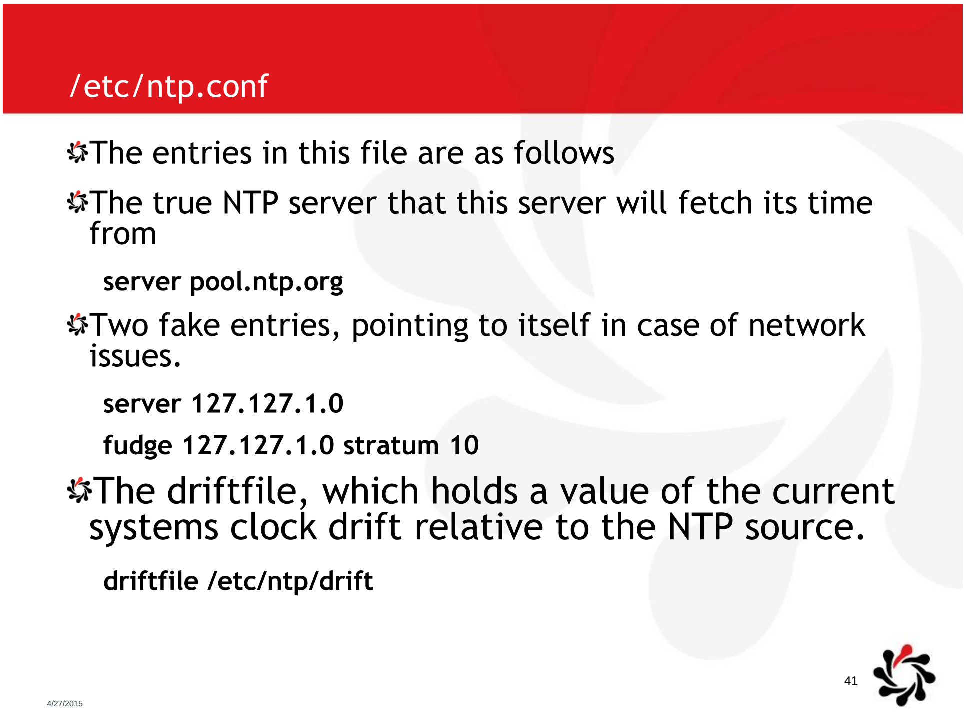


## The basic NTP configuration

- The main configuration file is

  /etc/ntp.conf
- The minimum contents of this file look like

```
# --- GENERAL CONFIGURATION ---
server pool.ntp.org
server 127.127.1.0
fudge 127.127.1.0 stratum 10
# Drift file.
driftfile /etc/ntp/drift
```



## /etc/ntp.conf

- The entries in this file are as follows
- The true NTP server that this server will fetch its time from

```
server pool.ntp.org
```

- Two fake entries, pointing to itself in case of network issues.

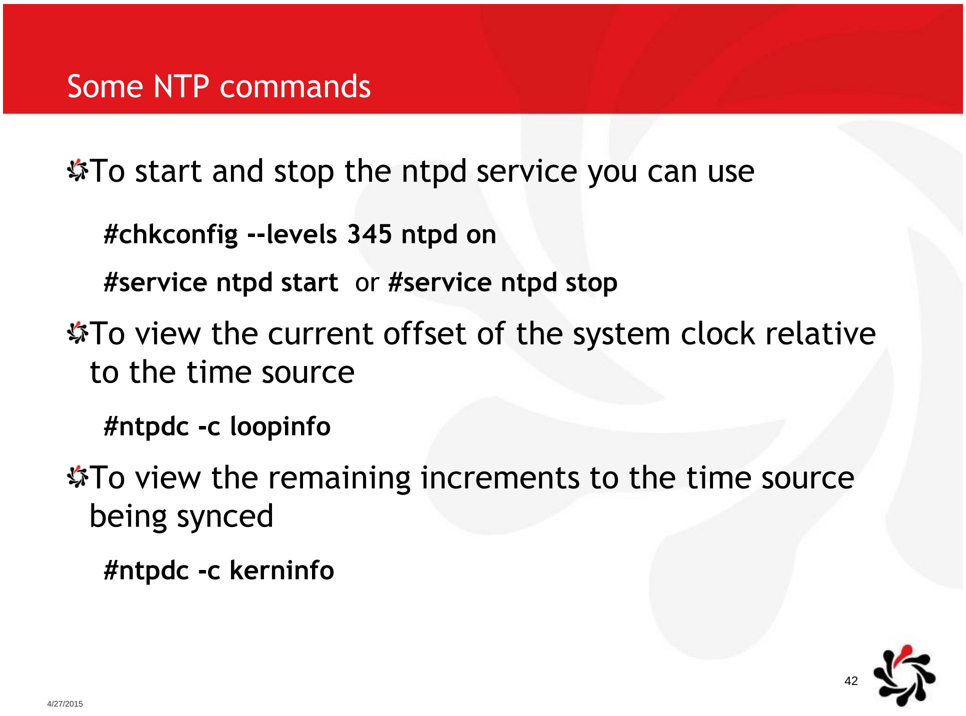
```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
```

- The driftfile, which holds a value of the current systems clock drift relative to the NTP source.

```
driftfile /etc/ntp/drift
```

4/27/2015

41



## Some NTP commands

- To start and stop the ntpd service you can use

```
#chkconfig --levels 345 ntpd on
#service ntpd start  or #service ntpd stop
```

- To view the current offset of the system clock relative to the time source

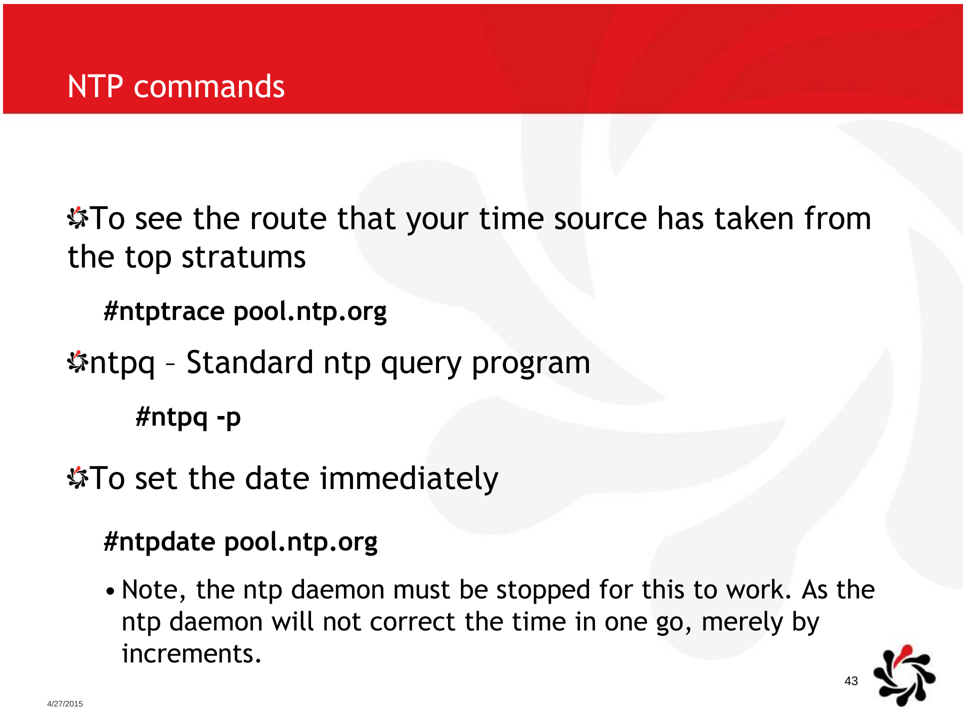
```
#ntpdc -c loopinfo
```

- To view the remaining increments to the time source being synced

```
#ntpdc -c kerninfo
```

4/27/2015

42



## NTP commands

- To see the route that your time source has taken from the top stratums

  **#ntptrace pool.ntp.org**

- ntpq – Standard ntp query program

  **#ntpq -p**

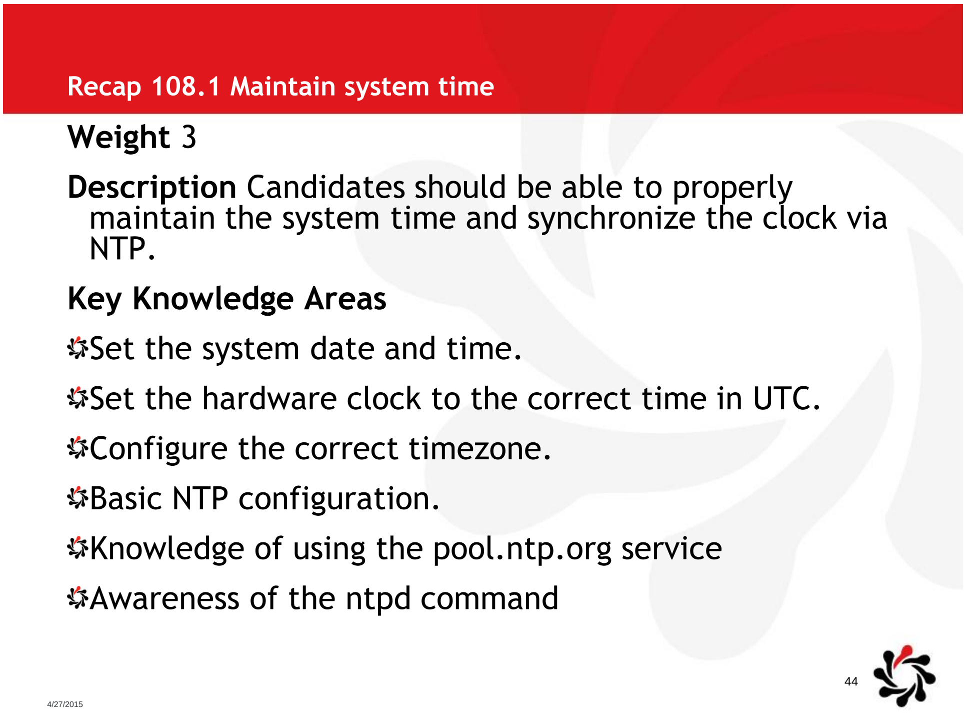
- To set the date immediately

  **#ntpdate pool.ntp.org**

  - Note, the ntp daemon must be stopped for this to work. As the ntp daemon will not correct the time in one go, merely by increments.

43

4/27/2015



## Recap 108.1 Maintain system time

**Weight** 3

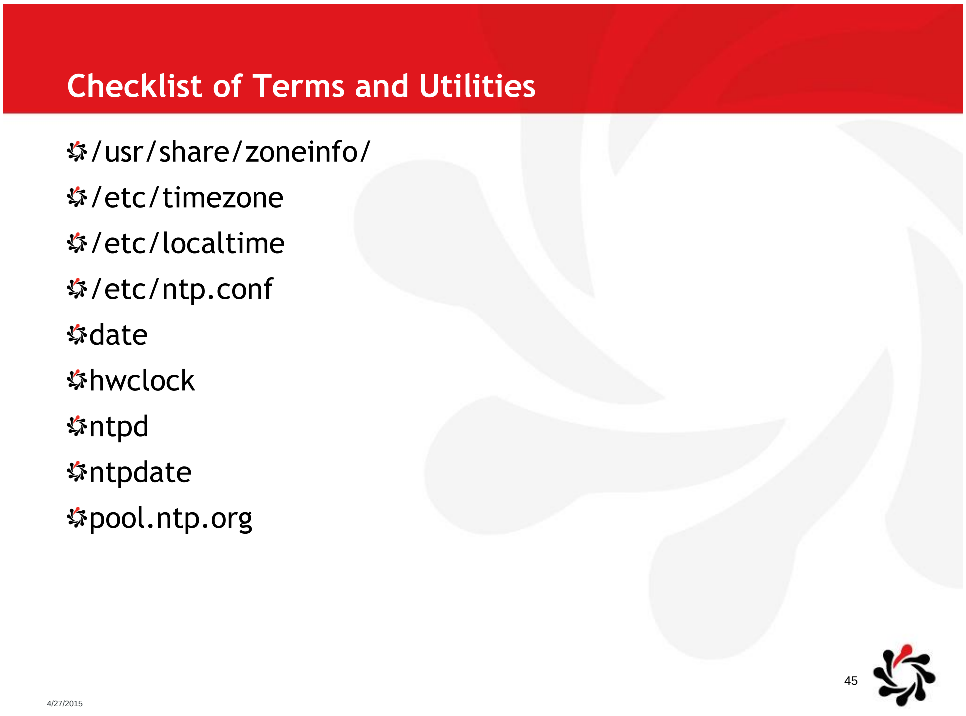
**Description** Candidates should be able to properly maintain the system time and synchronize the clock via NTP.

**Key Knowledge Areas**

- Set the system date and time.
- Set the hardware clock to the correct time in UTC.
- Configure the correct timezone.
- Basic NTP configuration.
- Knowledge of using the pool.ntp.org service
- Awareness of the ntpd command

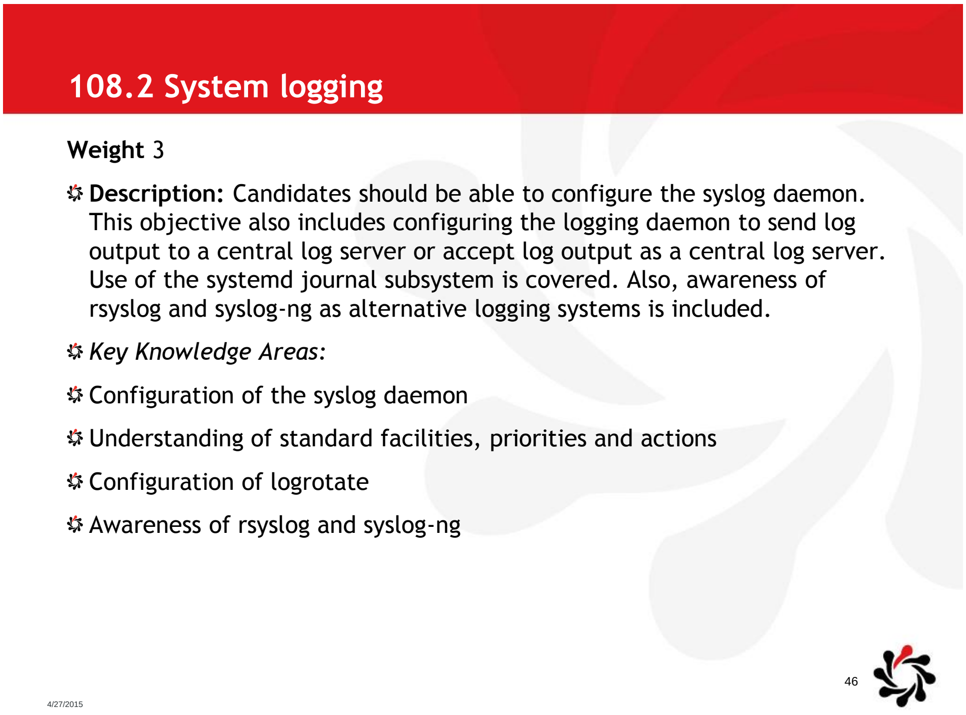
44

4/27/2015



## Checklist of Terms and Utilities

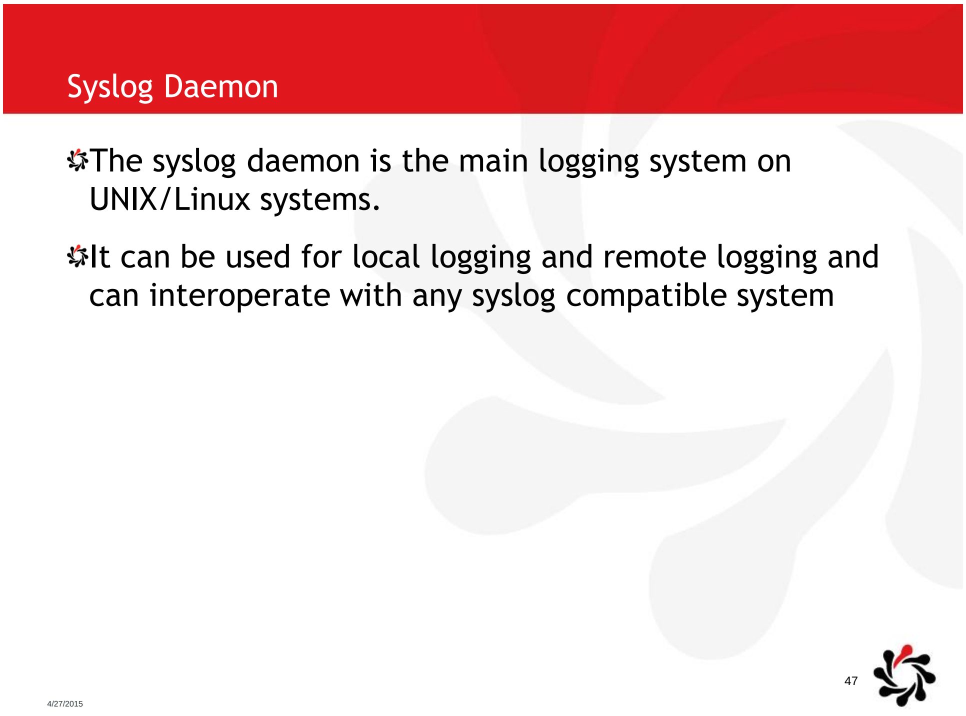
- /usr/share/zoneinfo/
- /etc/timezone
- /etc/localtime
- /etc/ntp.conf
- date
- hwclock
- ntpd
- ntpdate
- pool.ntp.org



## 108.2 System logging

**Weight** 3

- **Description:** Candidates should be able to configure the syslog daemon. This objective also includes configuring the logging daemon to send log output to a central log server or accept log output as a central log server. Use of the systemd journal subsystem is covered. Also, awareness of rsyslog and syslog-ng as alternative logging systems is included.
- *Key Knowledge Areas:*
- Configuration of the syslog daemon
- Understanding of standard facilities, priorities and actions
- Configuration of logrotate
- Awareness of rsyslog and syslog-ng

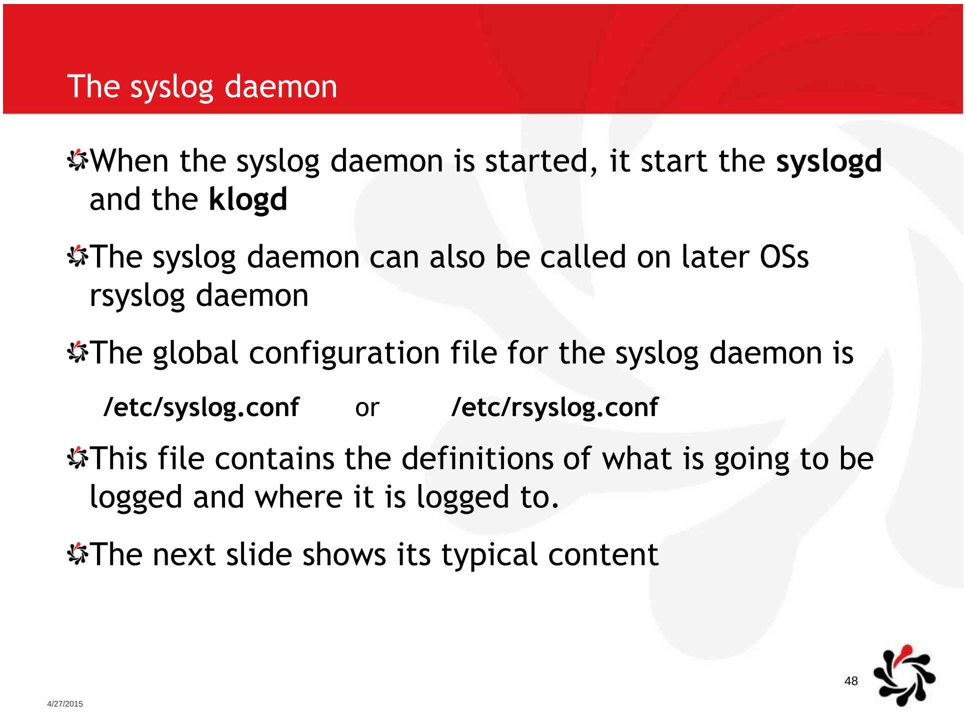


## Syslog Daemon

- The syslog daemon is the main logging system on UNIX/Linux systems.
- It can be used for local logging and remote logging and can interoperate with any syslog compatible system

47

4/27/2015



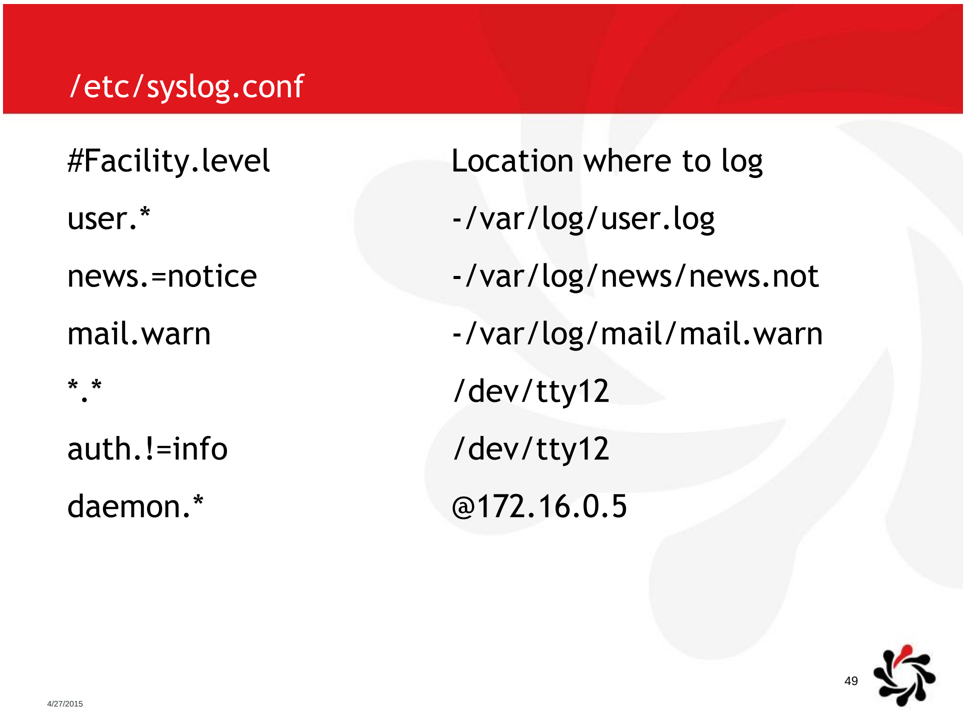
## The syslog daemon

- When the syslog daemon is started, it start the **syslogd** and the **klogd**
- The syslog daemon can also be called on later OSs rsyslog daemon
- The global configuration file for the syslog daemon is

  **/etc/syslog.conf** or **/etc/rsyslog.conf**
- This file contains the definitions of what is going to be logged and where it is logged to.
- The next slide shows its typical content

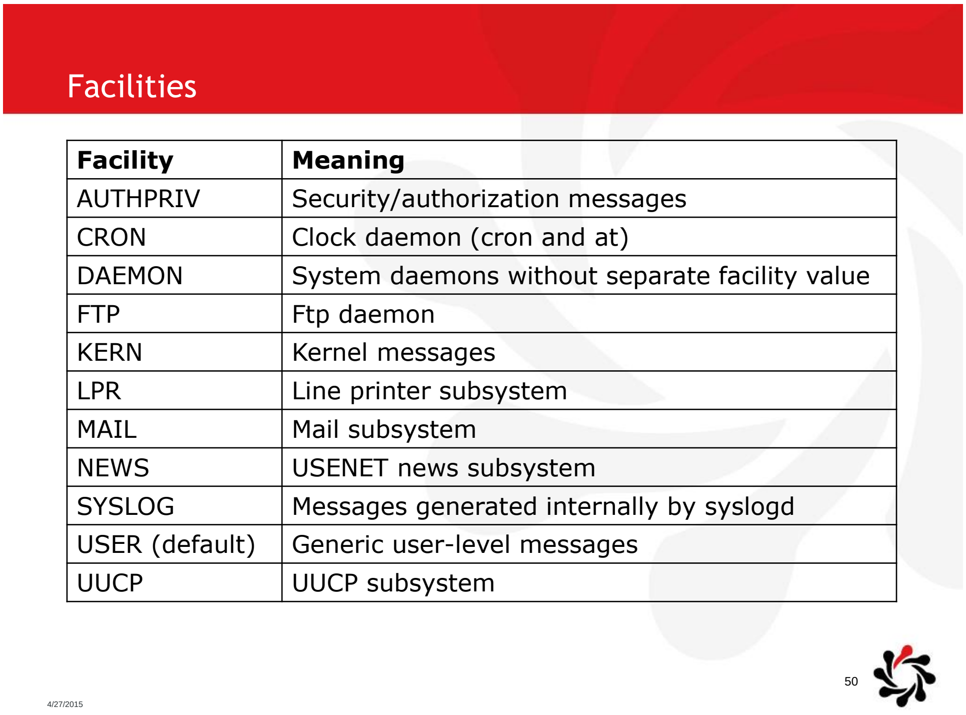
48

4/27/2015



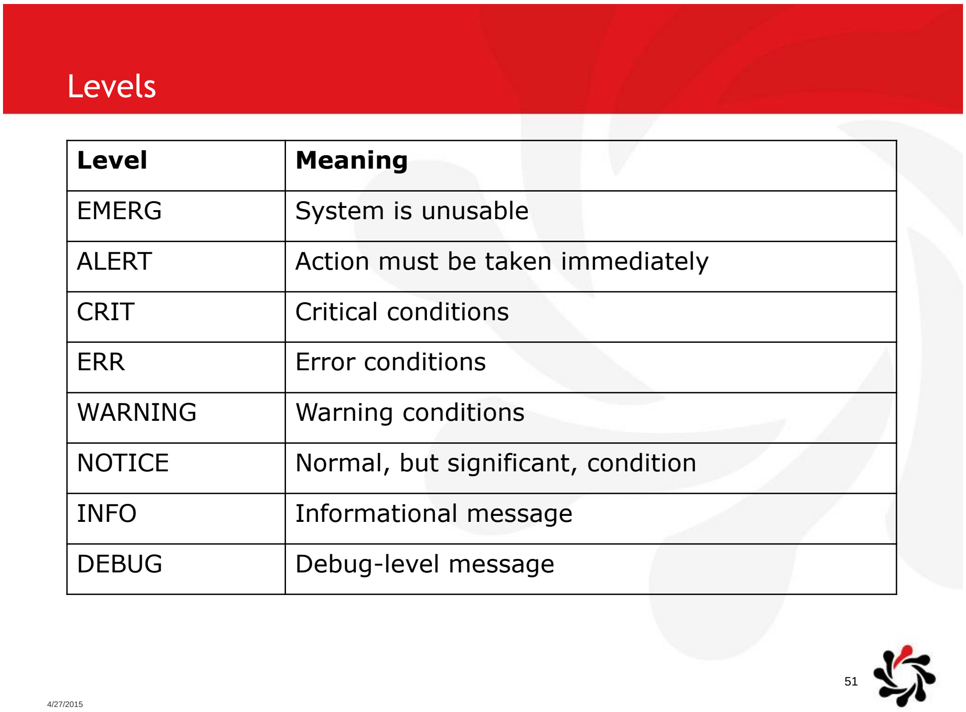
## /etc/syslog.conf

```
#Facility.level          Location where to log
user.*                   -/var/log/user.log
news.=notice             -/var/log/news/news.not
mail.warn                -/var/log/mail/mail.warn
*.*                      /dev/tty12
auth.!=info              /dev/tty12
daemon.*                 @172.16.0.5
```



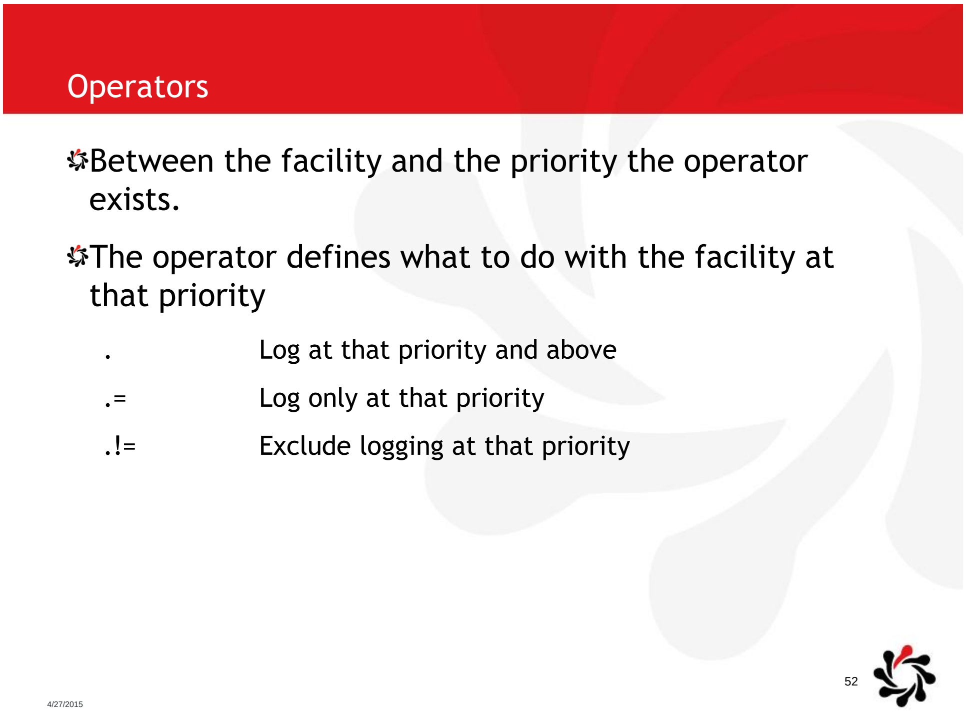
## Facilities

| **Facility** | **Meaning** |
|---|---|
| AUTHPRIV | Security/authorization messages |
| CRON | Clock daemon (cron and at) |
| DAEMON | System daemons without separate facility value |
| FTP | Ftp daemon |
| KERN | Kernel messages |
| LPR | Line printer subsystem |
| MAIL | Mail subsystem |
| NEWS | USENET news subsystem |
| SYSLOG | Messages generated internally by syslogd |
| USER (default) | Generic user-level messages |
| UUCP | UUCP subsystem |



## Levels

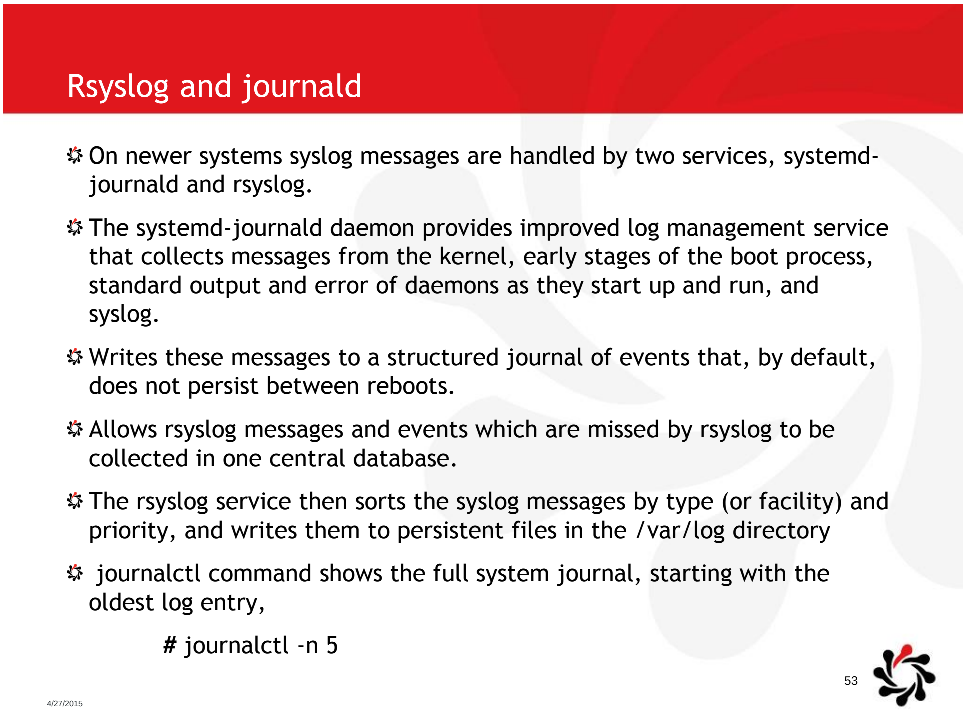
| Level | Meaning |
|---|---|
| EMERG | System is unusable |
| ALERT | Action must be taken immediately |
| CRIT | Critical conditions |
| ERR | Error conditions |
| WARNING | Warning conditions |
| NOTICE | Normal, but significant, condition |
| INFO | Informational message |
| DEBUG | Debug-level message |



## Operators

- Between the facility and the priority the operator exists.
- The operator defines what to do with the facility at that priority

| | |
|---|---|
| . | Log at that priority and above |
| .= | Log only at that priority |
| .!= | Exclude logging at that priority |



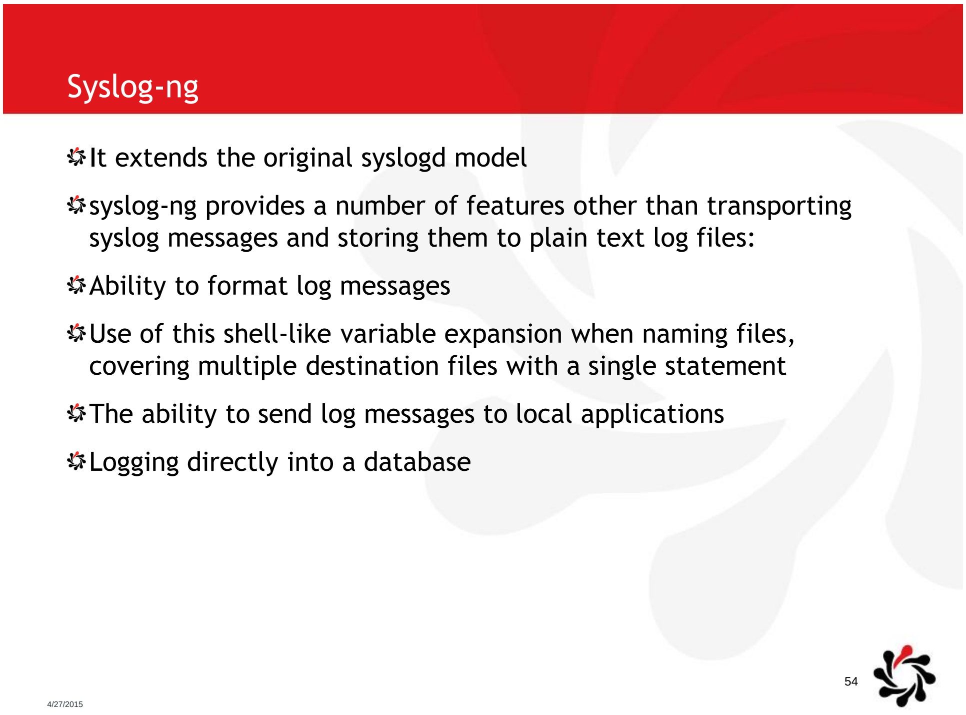
## Rsyslog and journald

- On newer systems syslog messages are handled by two services, systemd-journald and rsyslog.
- The systemd-journald daemon provides improved log management service that collects messages from the kernel, early stages of the boot process, standard output and error of daemons as they start up and run, and syslog.
- Writes these messages to a structured journal of events that, by default, does not persist between reboots.
- Allows rsyslog messages and events which are missed by rsyslog to be collected in one central database.
- The rsyslog service then sorts the syslog messages by type (or facility) and priority, and writes them to persistent files in the /var/log directory
- journalctl command shows the full system journal, starting with the oldest log entry,

```
# journalctl -n 5
```

53

4/27/2015

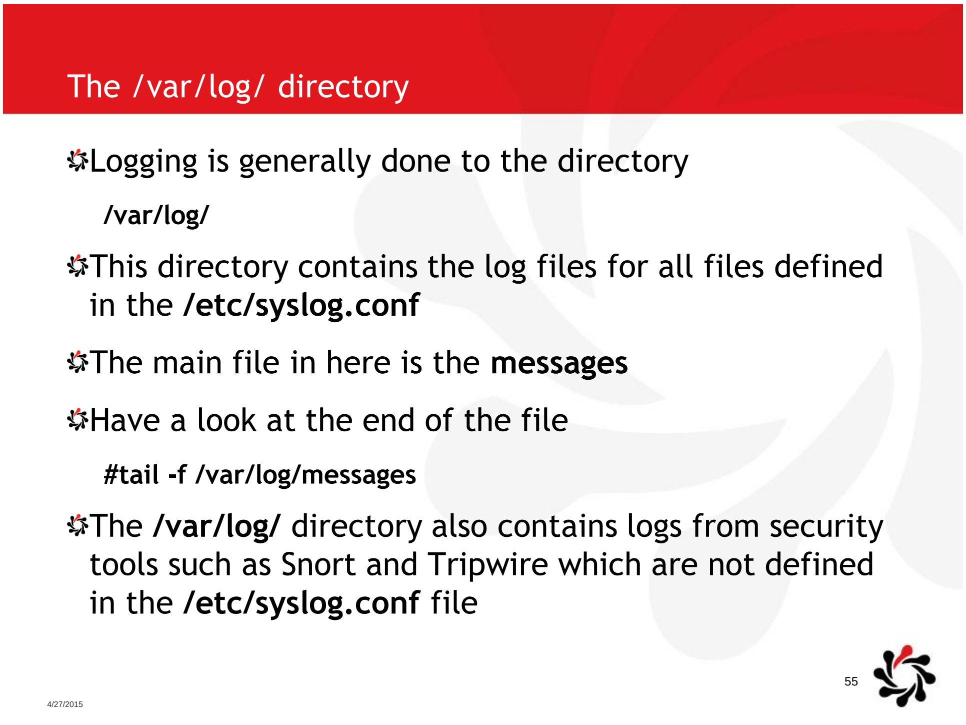


## Syslog-ng

- It extends the original syslogd model
- syslog-ng provides a number of features other than transporting syslog messages and storing them to plain text log files:
- Ability to format log messages
- Use of this shell-like variable expansion when naming files, covering multiple destination files with a single statement
- The ability to send log messages to local applications
- Logging directly into a database

54

4/27/2015



## The /var/log/ directory

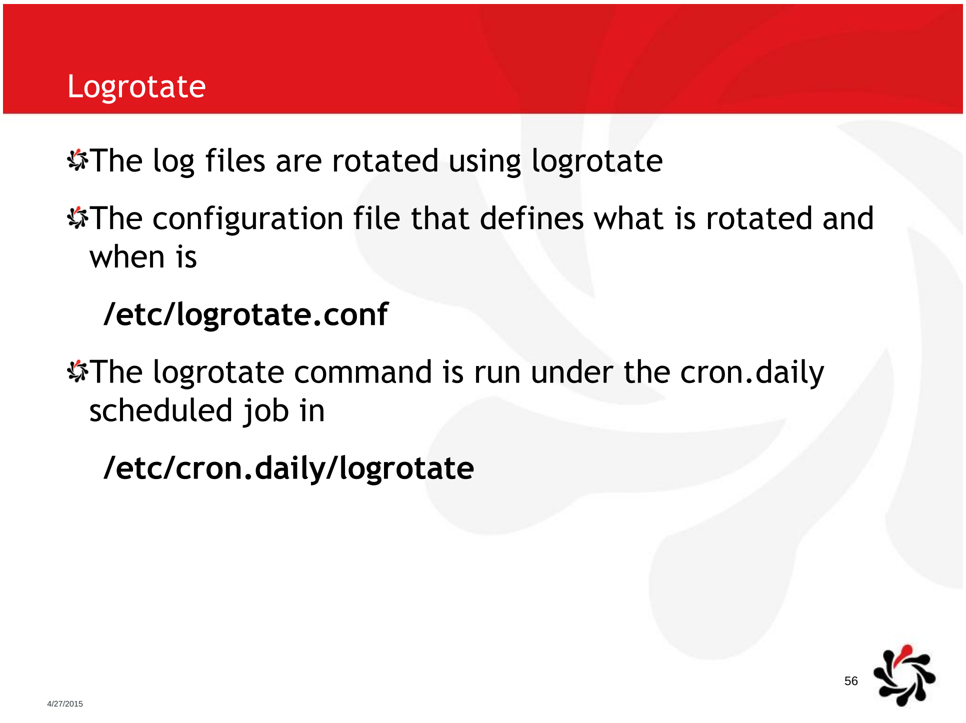
- Logging is generally done to the directory

  **/var/log/**

- This directory contains the log files for all files defined in the **/etc/syslog.conf**
- The main file in here is the **messages**
- Have a look at the end of the file

  **#tail -f /var/log/messages**

- The **/var/log/** directory also contains logs from security tools such as Snort and Tripwire which are not defined in the **/etc/syslog.conf** file



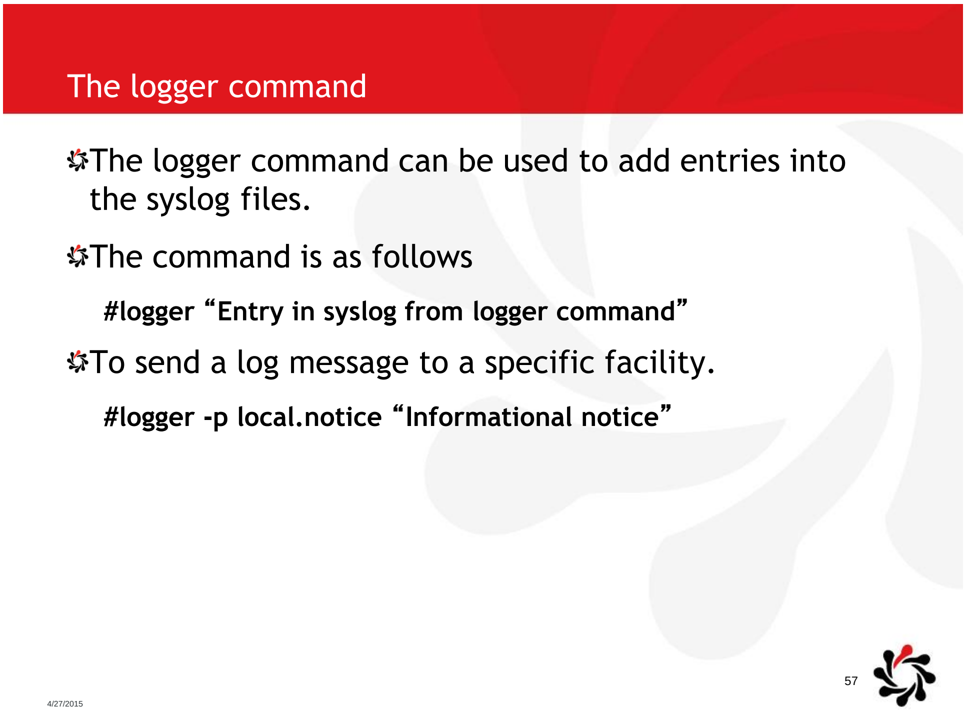
## Logrotate

- The log files are rotated using logrotate
- The configuration file that defines what is rotated and when is

  **/etc/logrotate.conf**

- The logrotate command is run under the cron.daily scheduled job in

  **/etc/cron.daily/logrotate**



## The logger command

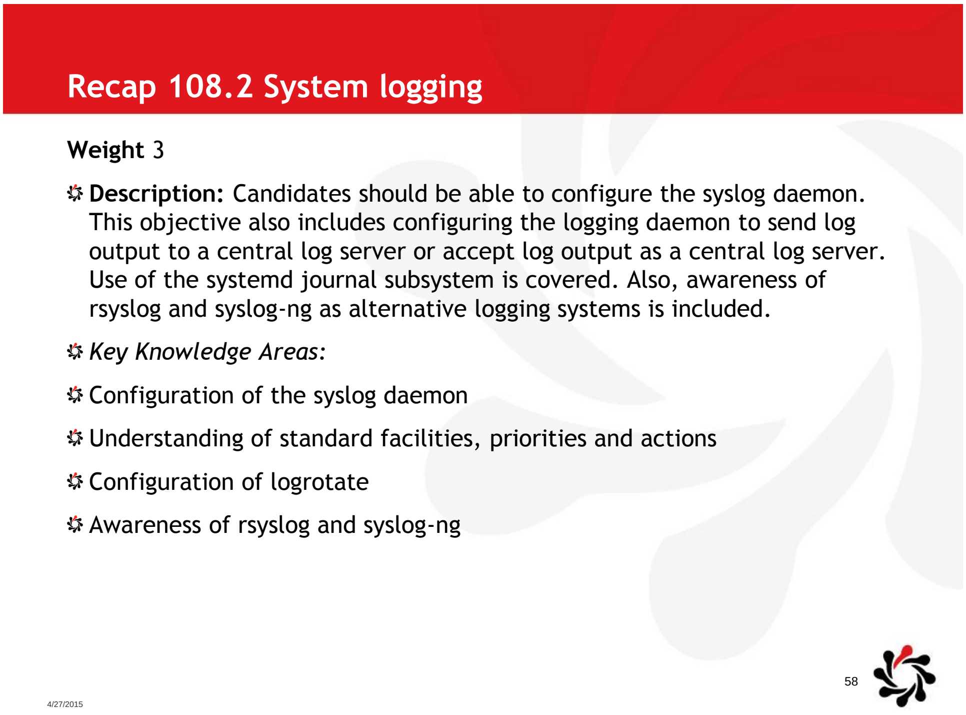
- The logger command can be used to add entries into the syslog files.
- The command is as follows

```
#logger "Entry in syslog from logger command"
```

- To send a log message to a specific facility.

```
#logger -p local.notice "Informational notice"
```

4/27/2015 57

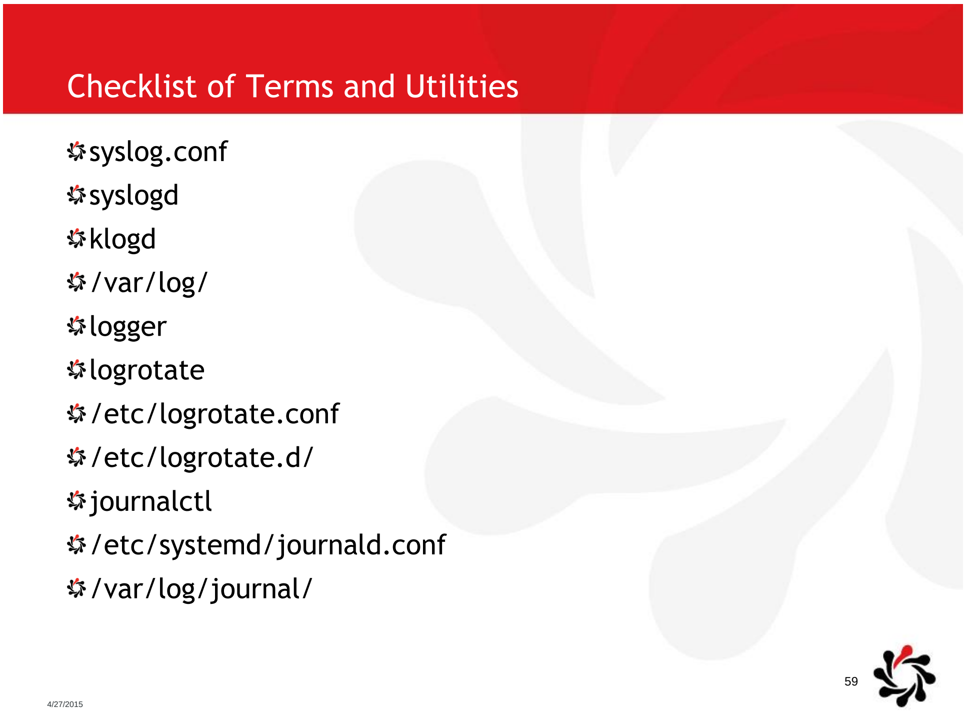


## Recap 108.2 System logging

**Weight** 3

- **Description:** Candidates should be able to configure the syslog daemon. This objective also includes configuring the logging daemon to send log output to a central log server or accept log output as a central log server. Use of the systemd journal subsystem is covered. Also, awareness of rsyslog and syslog-ng as alternative logging systems is included.
- *Key Knowledge Areas:*
- Configuration of the syslog daemon
- Understanding of standard facilities, priorities and actions
- Configuration of logrotate
- Awareness of rsyslog and syslog-ng

4/27/2015 58

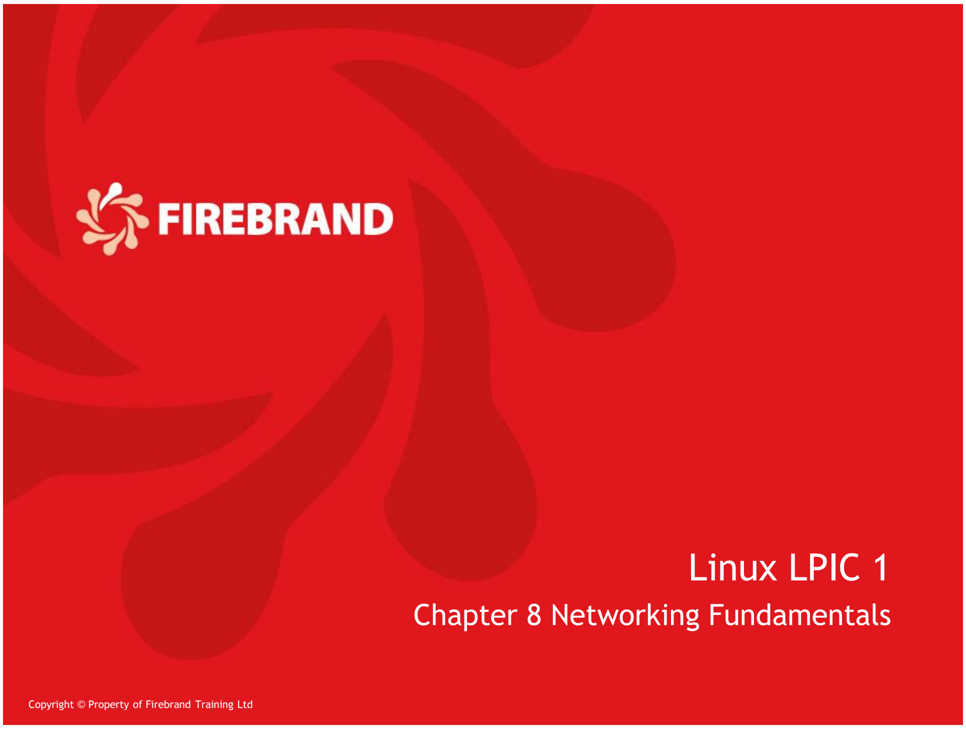


## Checklist of Terms and Utilities

- syslog.conf
- syslogd
- klogd
- /var/log/
- logger
- logrotate
- /etc/logrotate.conf
- /etc/logrotate.d/
- journalctl
- /etc/systemd/journald.conf
- /var/log/journal/

59

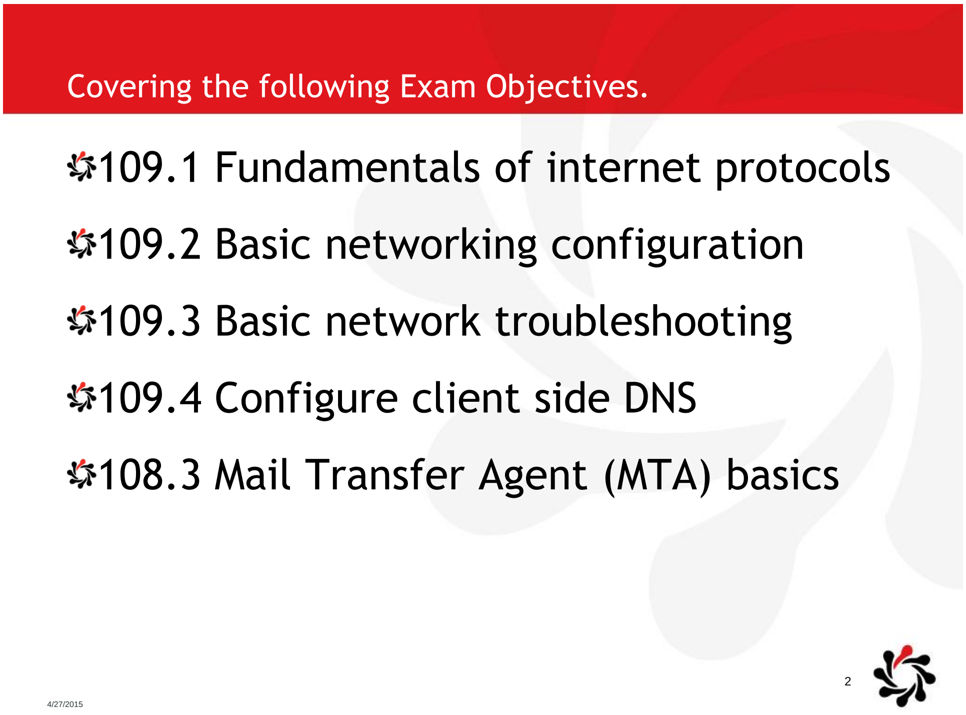
4/27/2015



FIREBRAND

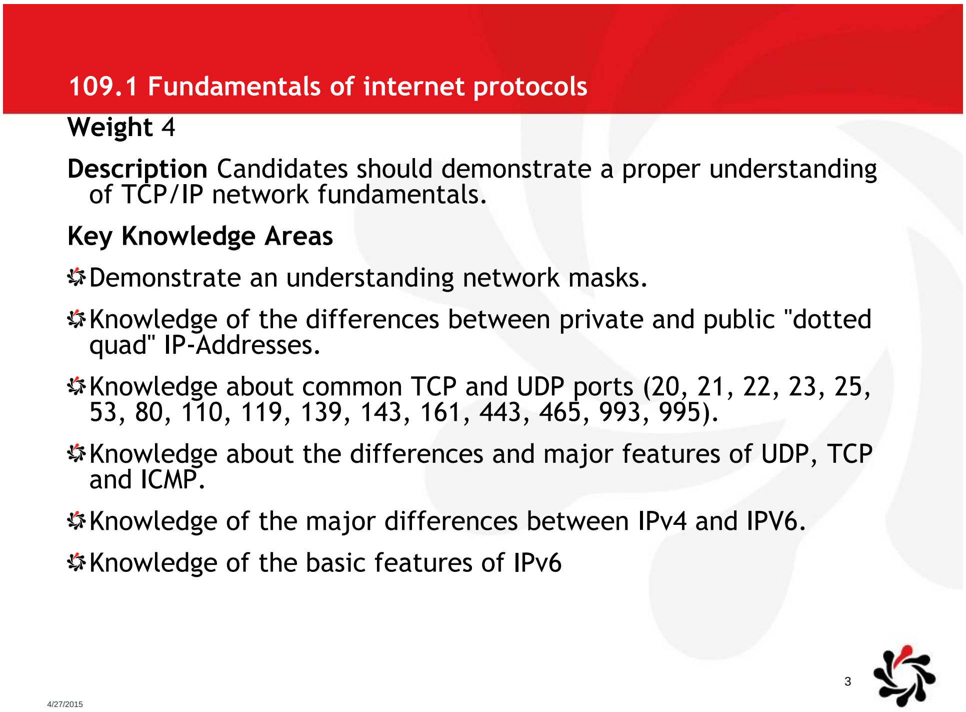
# Linux LPIC 1

## Chapter 8 Networking Fundamentals



## Covering the following Exam Objectives.

- 109.1 Fundamentals of internet protocols
- 109.2 Basic networking configuration
- 109.3 Basic network troubleshooting
- 109.4 Configure client side DNS
- 108.3 Mail Transfer Agent (MTA) basics



## 109.1 Fundamentals of internet protocols

**Weight** 4

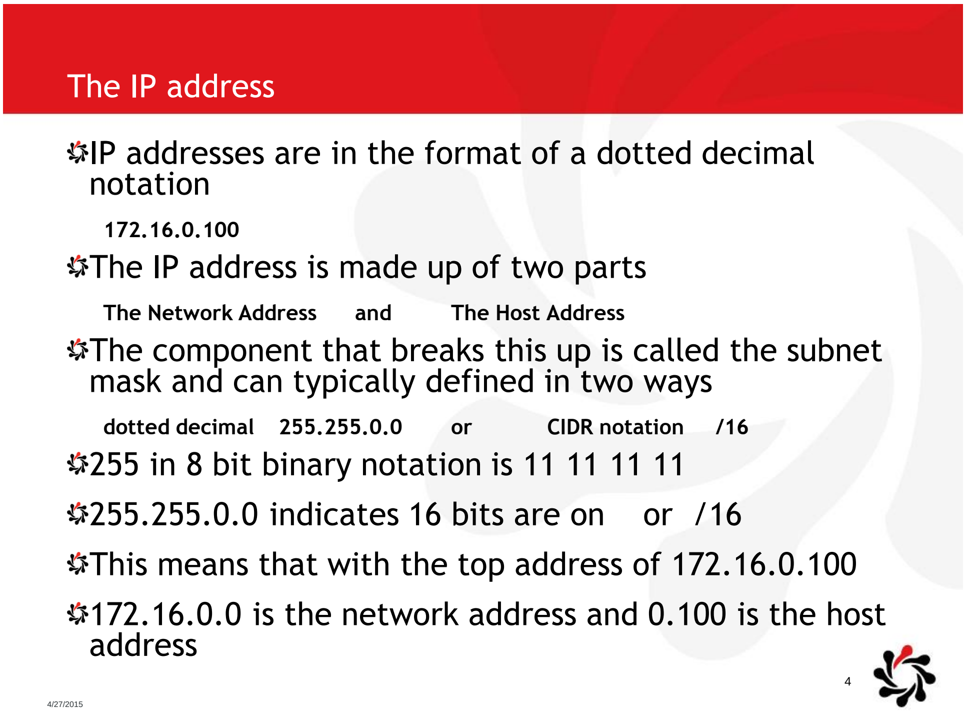
**Description** Candidates should demonstrate a proper understanding of TCP/IP network fundamentals.

**Key Knowledge Areas**

- Demonstrate an understanding network masks.
- Knowledge of the differences between private and public "dotted quad" IP-Addresses.
- Knowledge about common TCP and UDP ports (20, 21, 22, 23, 25, 53, 80, 110, 119, 139, 143, 161, 443, 465, 993, 995).
- Knowledge about the differences and major features of UDP, TCP and ICMP.
- Knowledge of the major differences between IPv4 and IPV6.
- Knowledge of the basic features of IPv6

3

4/27/2015



## The IP address

- IP addresses are in the format of a dotted decimal notation

  **172.16.0.100**

- The IP address is made up of two parts

  **The Network Address and The Host Address**

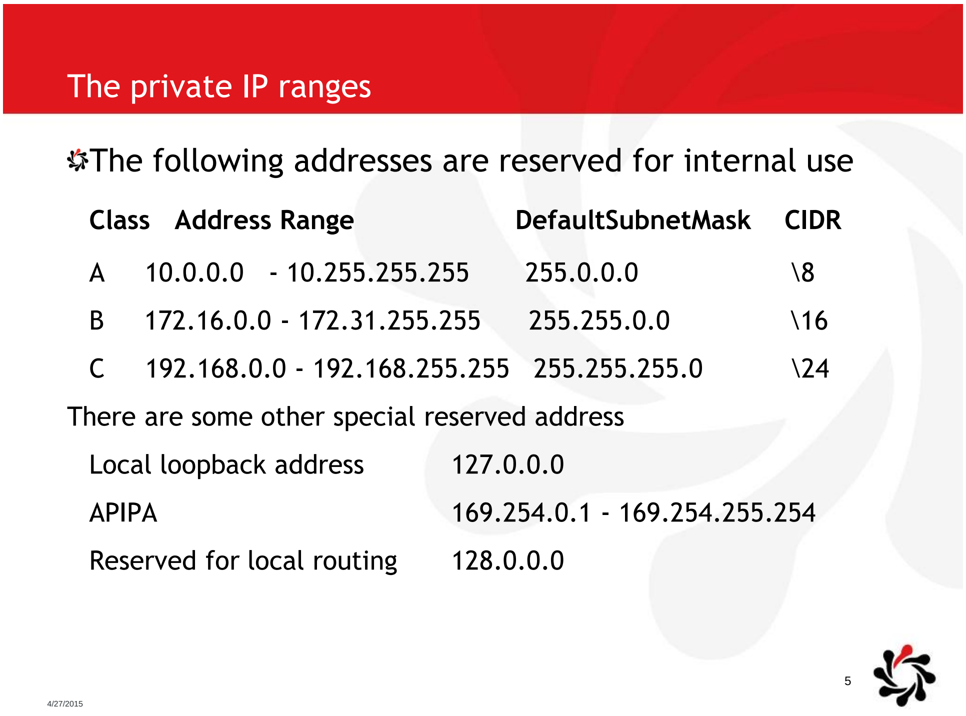
- The component that breaks this up is called the subnet mask and can typically defined in two ways

  **dotted decimal 255.255.0.0 or CIDR notation /16**

- 255 in 8 bit binary notation is 11 11 11 11
- 255.255.0.0 indicates 16 bits are on or /16
- This means that with the top address of 172.16.0.100
- 172.16.0.0 is the network address and 0.100 is the host address

4

4/27/2015



## The private IP ranges

- The following addresses are reserved for internal use

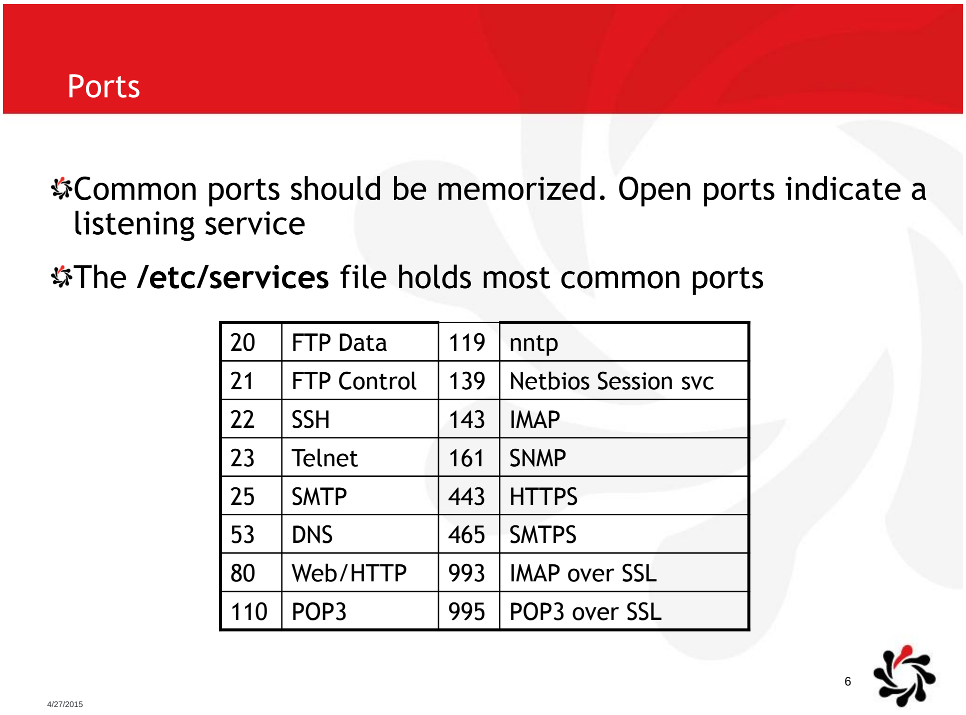
| Class | Address Range | DefaultSubnetMask | CIDR |
|---|---|---|---|
| A | 10.0.0.0 - 10.255.255.255 | 255.0.0.0 | \8 |
| B | 172.16.0.0 - 172.31.255.255 | 255.255.0.0 | \16 |
| C | 192.168.0.0 - 192.168.255.255 | 255.255.255.0 | \24 |

There are some other special reserved address

| | |
|---|---|
| Local loopback address | 127.0.0.0 |
| APIPA | 169.254.0.1 - 169.254.255.254 |
| Reserved for local routing | 128.0.0.0 |

4/27/2015

5



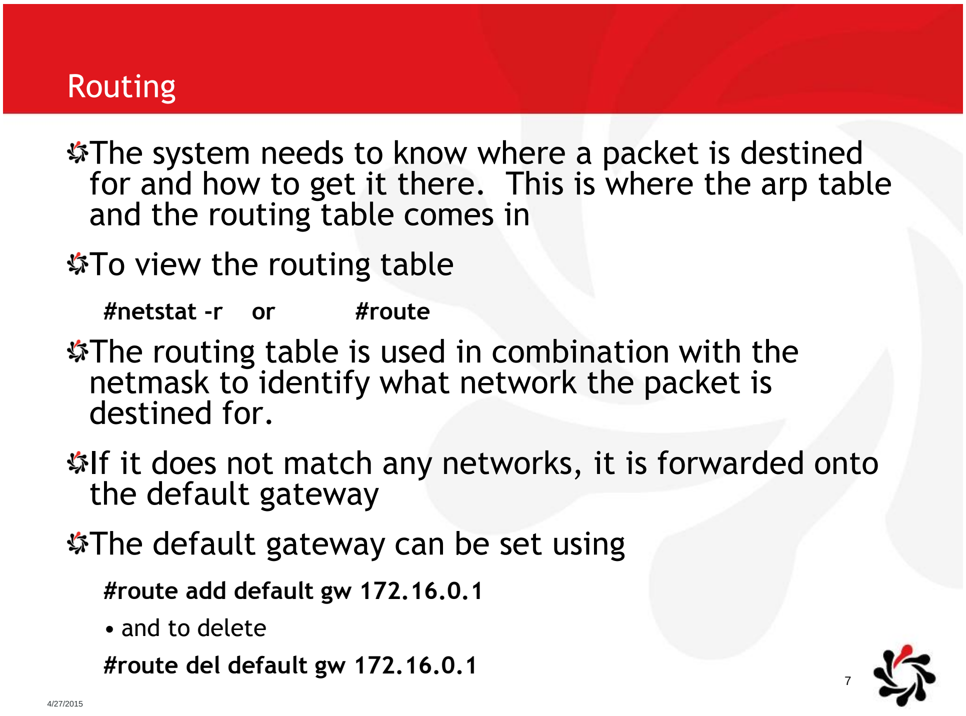
## Ports

- Common ports should be memorized. Open ports indicate a listening service
- The **/etc/services** file holds most common ports

| | | | |
|---|---|---|---|
| 20 | FTP Data | 119 | nntp |
| 21 | FTP Control | 139 | Netbios Session svc |
| 22 | SSH | 143 | IMAP |
| 23 | Telnet | 161 | SNMP |
| 25 | SMTP | 443 | HTTPS |
| 53 | DNS | 465 | SMTPS |
| 80 | Web/HTTP | 993 | IMAP over SSL |
| 110 | POP3 | 995 | POP3 over SSL |

4/27/2015

6



## Routing

- The system needs to know where a packet is destined for and how to get it there. This is where the arp table and the routing table comes in
- To view the routing table

```
#netstat -r    or          #route
```

- The routing table is used in combination with the netmask to identify what network the packet is destined for.
- If it does not match any networks, it is forwarded onto the default gateway
- The default gateway can be set using

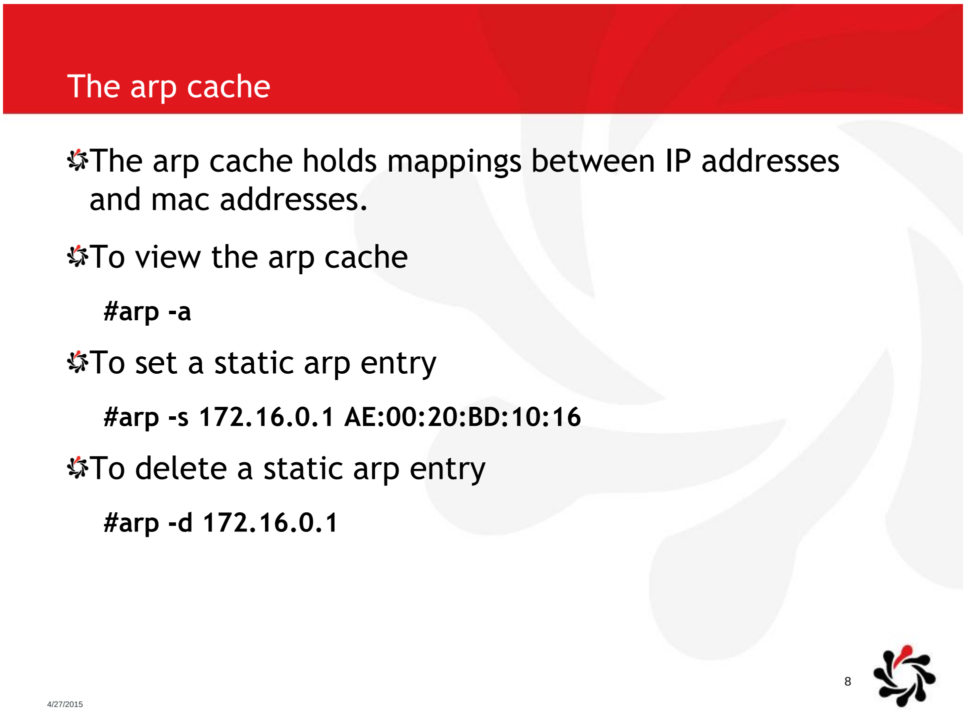
```
#route add default gw 172.16.0.1
```

  - and to delete

```
#route del default gw 172.16.0.1
```

4/27/2015

7



## The arp cache

- The arp cache holds mappings between IP addresses and mac addresses.
- To view the arp cache

```
#arp -a
```

- To set a static arp entry

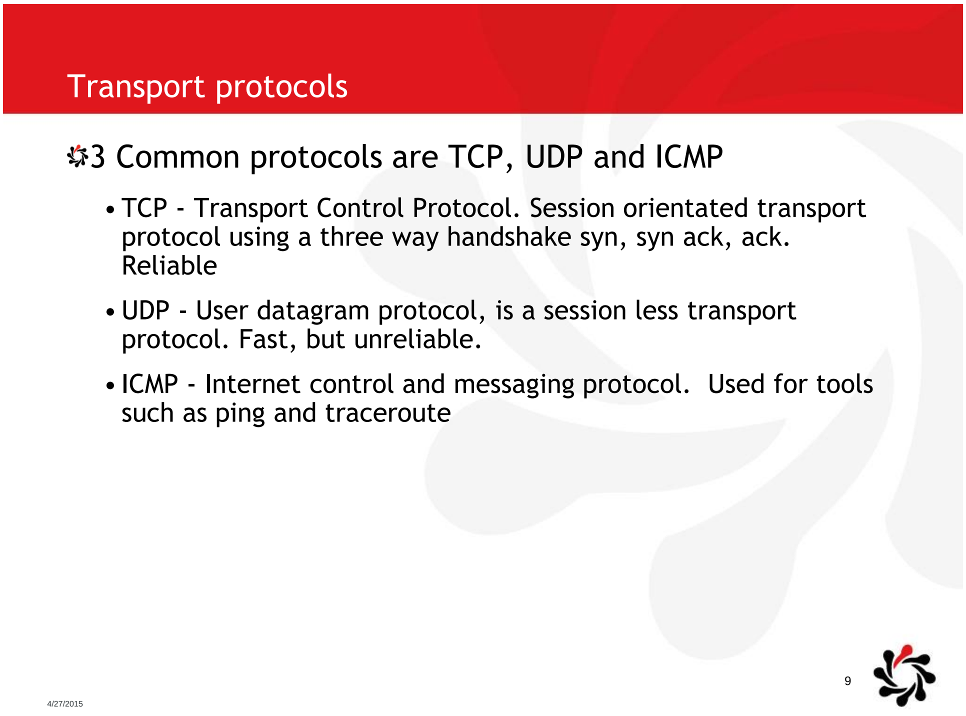
```
#arp -s 172.16.0.1 AE:00:20:BD:10:16
```

- To delete a static arp entry

```
#arp -d 172.16.0.1
```

4/27/2015

8

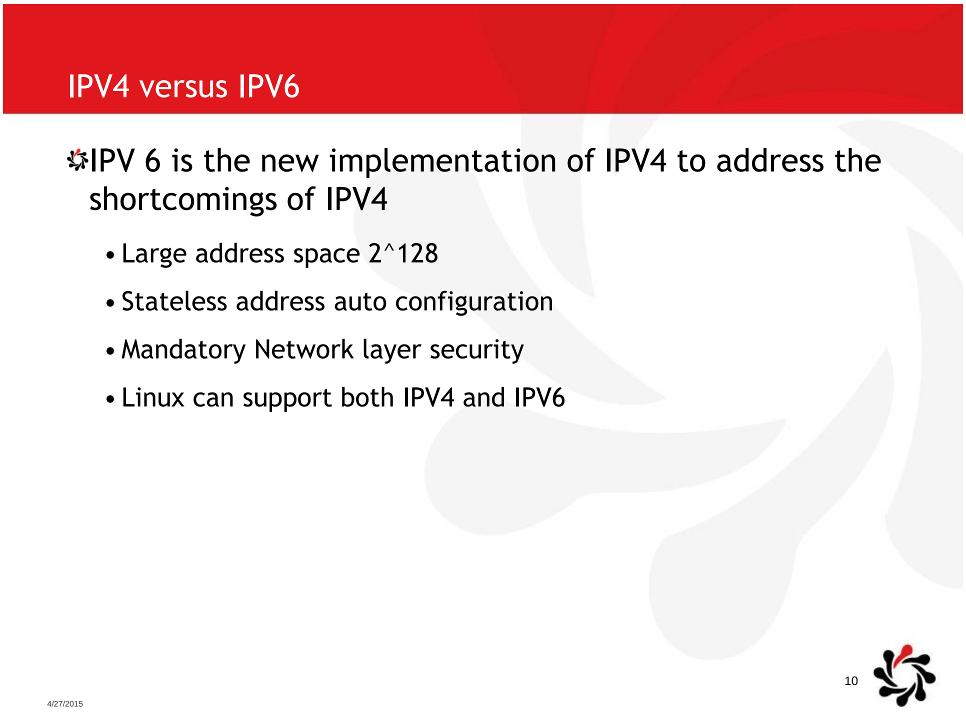


## Transport protocols

- 3 Common protocols are TCP, UDP and ICMP
  - TCP - Transport Control Protocol. Session orientated transport protocol using a three way handshake syn, syn ack, ack. Reliable
  - UDP - User datagram protocol, is a session less transport protocol. Fast, but unreliable.
  - ICMP - Internet control and messaging protocol. Used for tools such as ping and traceroute

9

4/27/2015

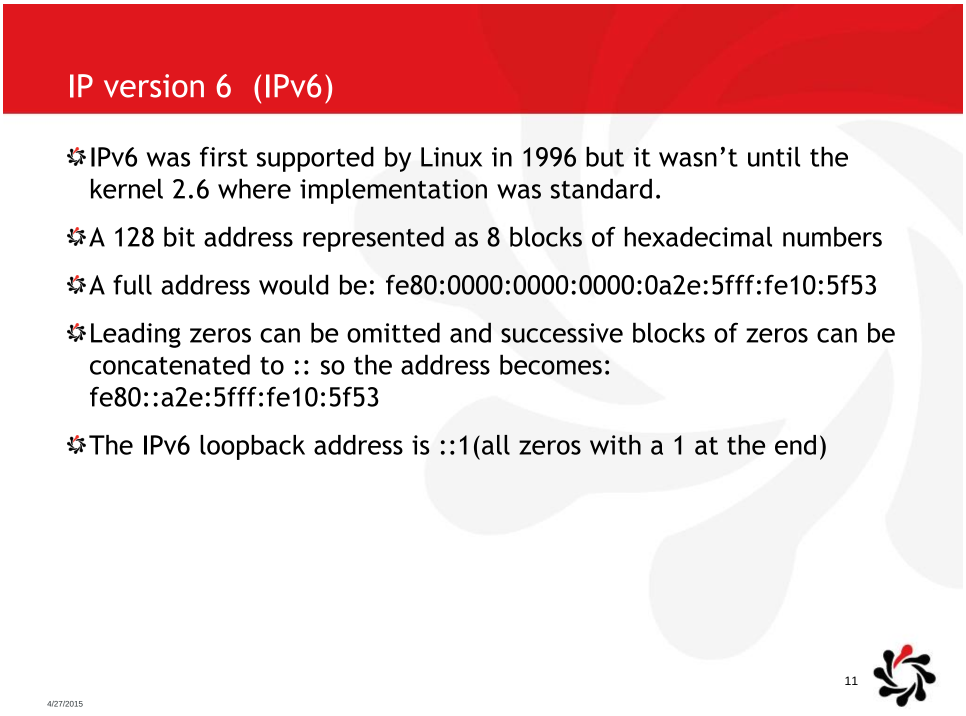


## IPV4 versus IPV6

- IPV 6 is the new implementation of IPV4 to address the shortcomings of IPV4
  - Large address space 2^128
  - Stateless address auto configuration
  - Mandatory Network layer security
  - Linux can support both IPV4 and IPV6

10

4/27/2015

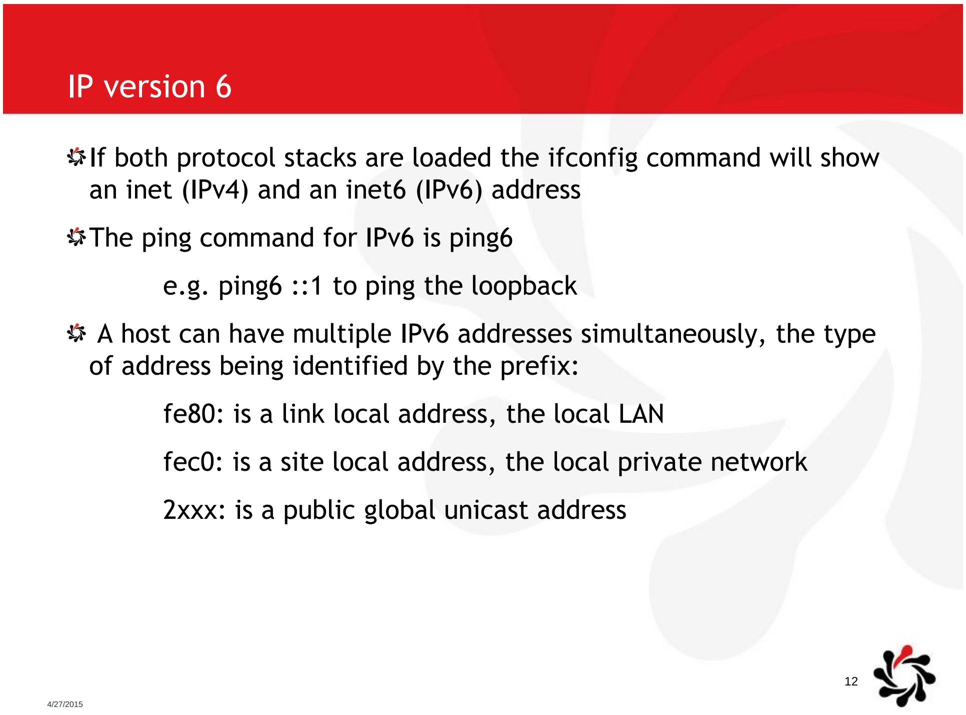


## IP version 6  (IPv6)

- IPv6 was first supported by Linux in 1996 but it wasn’t until the kernel 2.6 where implementation was standard.
- A 128 bit address represented as 8 blocks of hexadecimal numbers
- A full address would be: fe80:0000:0000:0000:0a2e:5fff:fe10:5f53
- Leading zeros can be omitted and successive blocks of zeros can be concatenated to :: so the address becomes: fe80::a2e:5fff:fe10:5f53
- The IPv6 loopback address is ::1(all zeros with a 1 at the end)

4/27/2015

11



## IP version 6

- If both protocol stacks are loaded the ifconfig command will show an inet (IPv4) and an inet6 (IPv6) address
- The ping command for IPv6 is ping6

  e.g. ping6 ::1 to ping the loopback

- A host can have multiple IPv6 addresses simultaneously, the type of address being identified by the prefix:

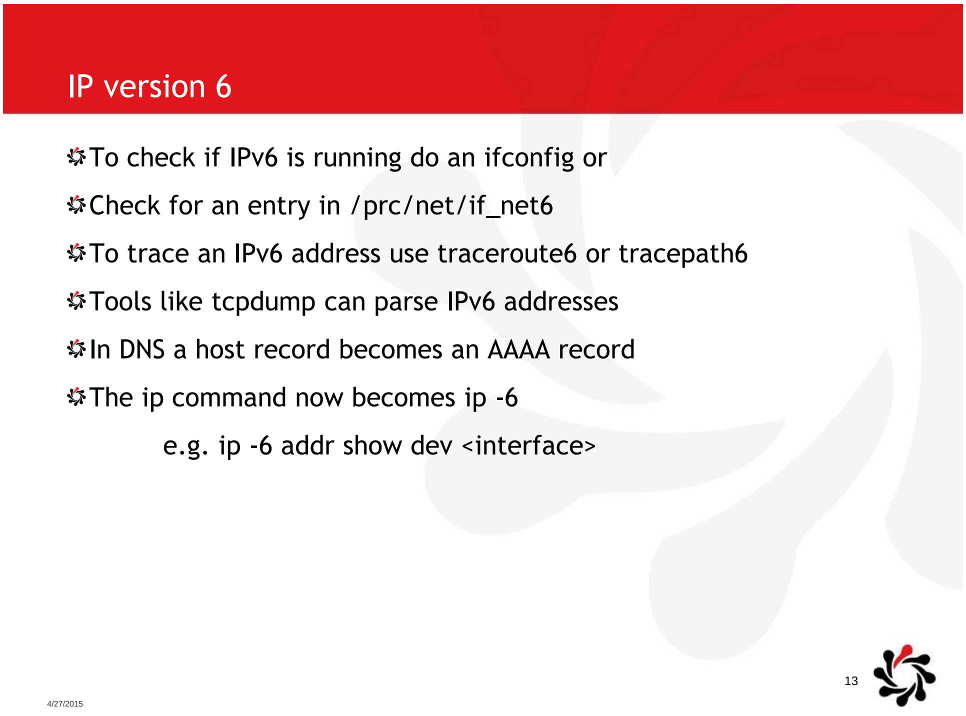
  fe80: is a link local address, the local LAN

  fec0: is a site local address, the local private network

  2xxx: is a public global unicast address

4/27/2015

12



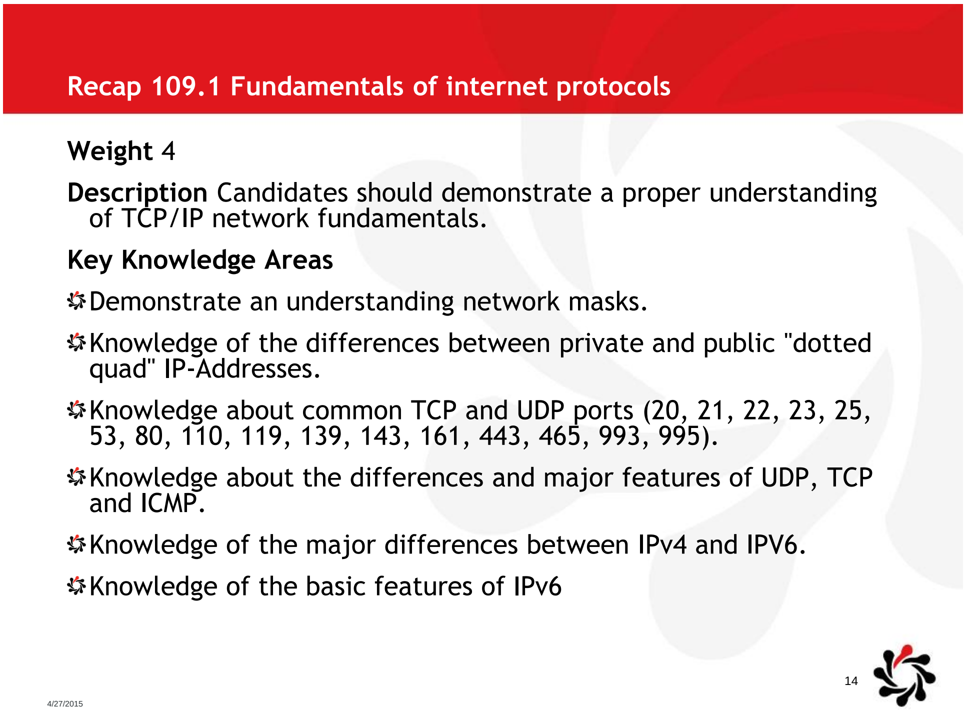
## IP version 6

- To check if IPv6 is running do an ifconfig or
- Check for an entry in /prc/net/if_net6
- To trace an IPv6 address use traceroute6 or tracepath6
- Tools like tcpdump can parse IPv6 addresses
- In DNS a host record becomes an AAAA record
- The ip command now becomes ip -6

  e.g. ip -6 addr show dev <interface>

13

4/27/2015



## Recap 109.1 Fundamentals of internet protocols

**Weight** 4

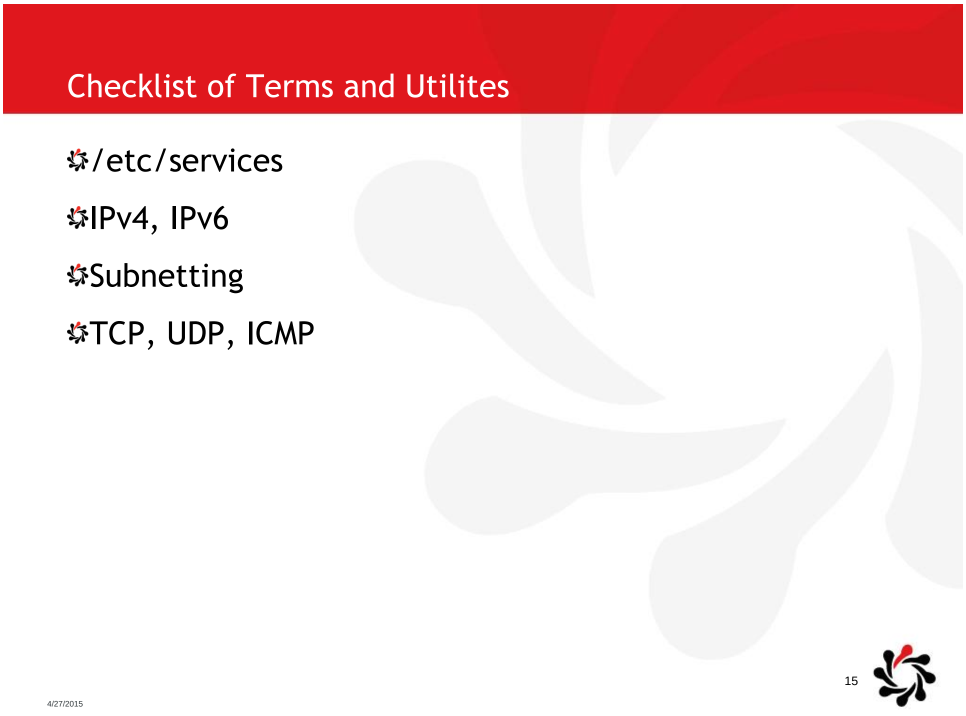
**Description** Candidates should demonstrate a proper understanding of TCP/IP network fundamentals.

**Key Knowledge Areas**

- Demonstrate an understanding network masks.
- Knowledge of the differences between private and public "dotted quad" IP-Addresses.
- Knowledge about common TCP and UDP ports (20, 21, 22, 23, 25, 53, 80, 110, 119, 139, 143, 161, 443, 465, 993, 995).
- Knowledge about the differences and major features of UDP, TCP and ICMP.
- Knowledge of the major differences between IPv4 and IPV6.
- Knowledge of the basic features of IPv6

14

4/27/2015

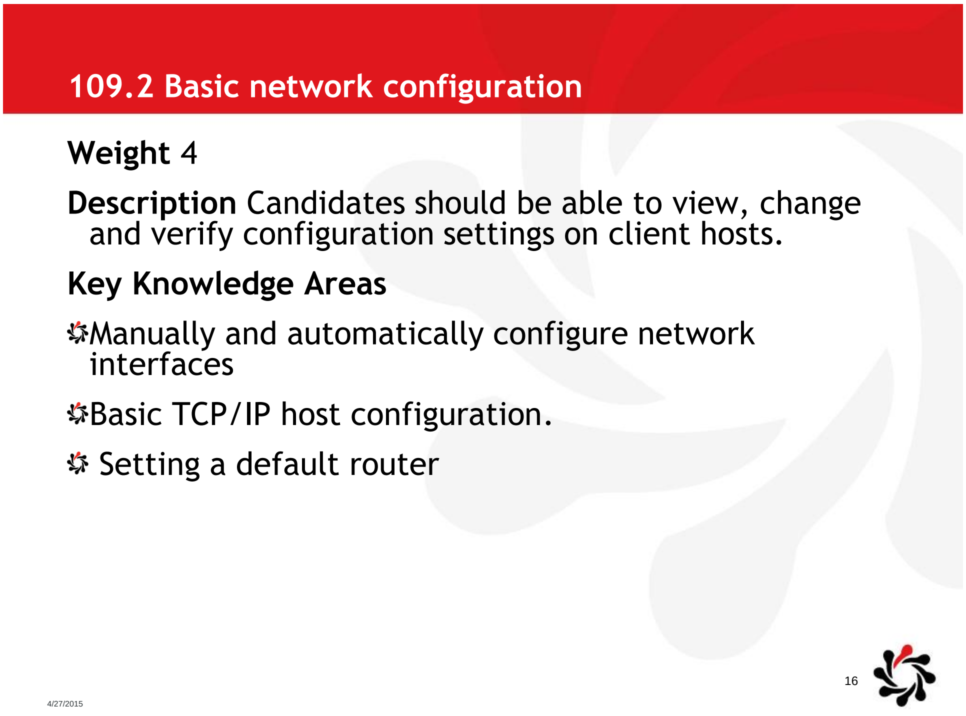


## Checklist of Terms and Utilites

- /etc/services
- IPv4, IPv6
- Subnetting
- TCP, UDP, ICMP

15

4/27/2015



## 109.2 Basic network configuration

**Weight** 4

**Description** Candidates should be able to view, change and verify configuration settings on client hosts.

**Key Knowledge Areas**

- Manually and automatically configure network interfaces
- Basic TCP/IP host configuration.
- Setting a default router

16

4/27/2015

## The hostname

- The hostname is configured through the file **/etc/hostname**
- The entry should be a FQDN Fully qualified domain name i.e. server.lpiclass.local
- The hostname can also be in other files as well e.g. **/etc/sysconfig/network**

```
HOSTNAME=server.lpiclass.local
```

## The ifconfig command

- To configure an interface manually use the ifconfig command

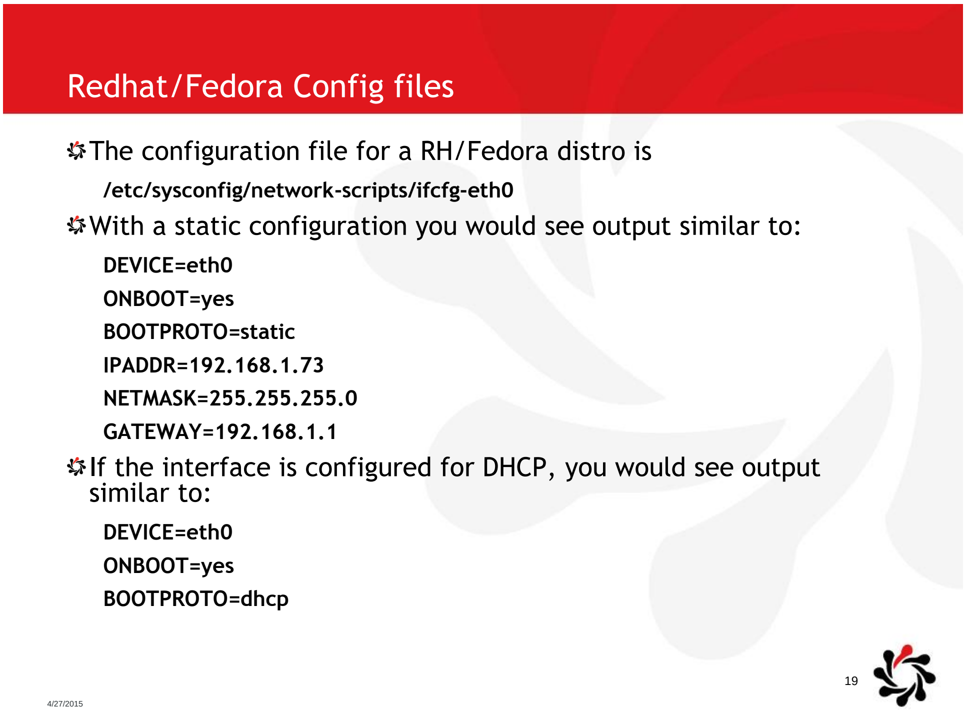
```
#ifconfig eth0 172.16.0.101 netmask 255.255.0.0 up
```

- To take a network interface down

```
#ifconfig eth0 down
```

- If the configuration files exist, then the admin can use

```
#ifup eth0
#ifdown eth0
```



## Redhat/Fedora Config files

- The configuration file for a RH/Fedora distro is

  /etc/sysconfig/network-scripts/ifcfg-eth0

- With a static configuration you would see output similar to:

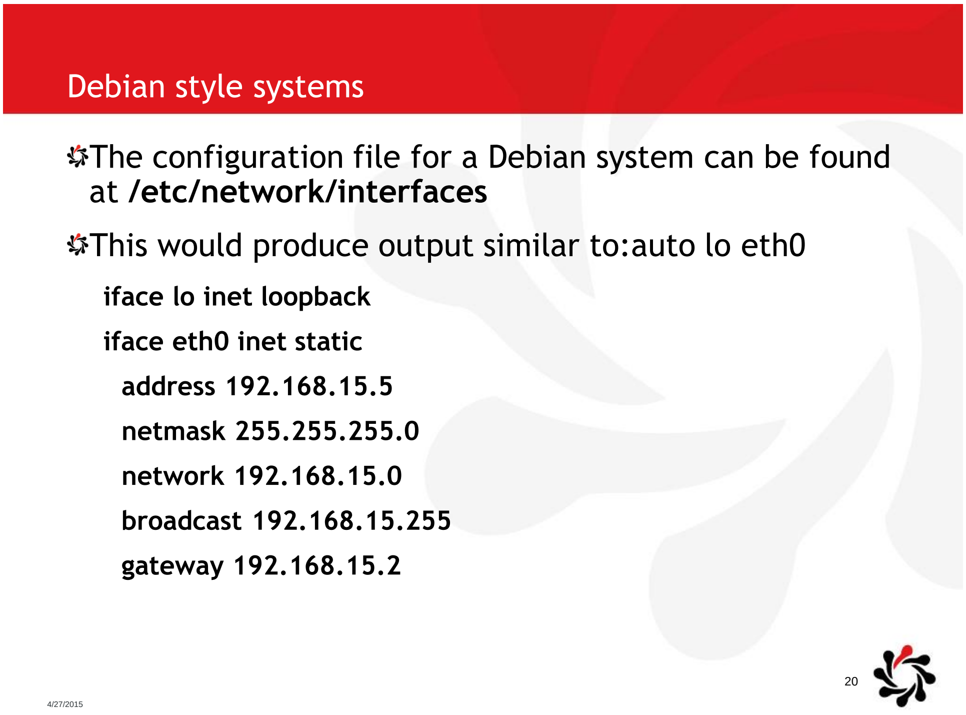
```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.1.73
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
```

- If the interface is configured for DHCP, you would see output similar to:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
```

19

4/27/2015



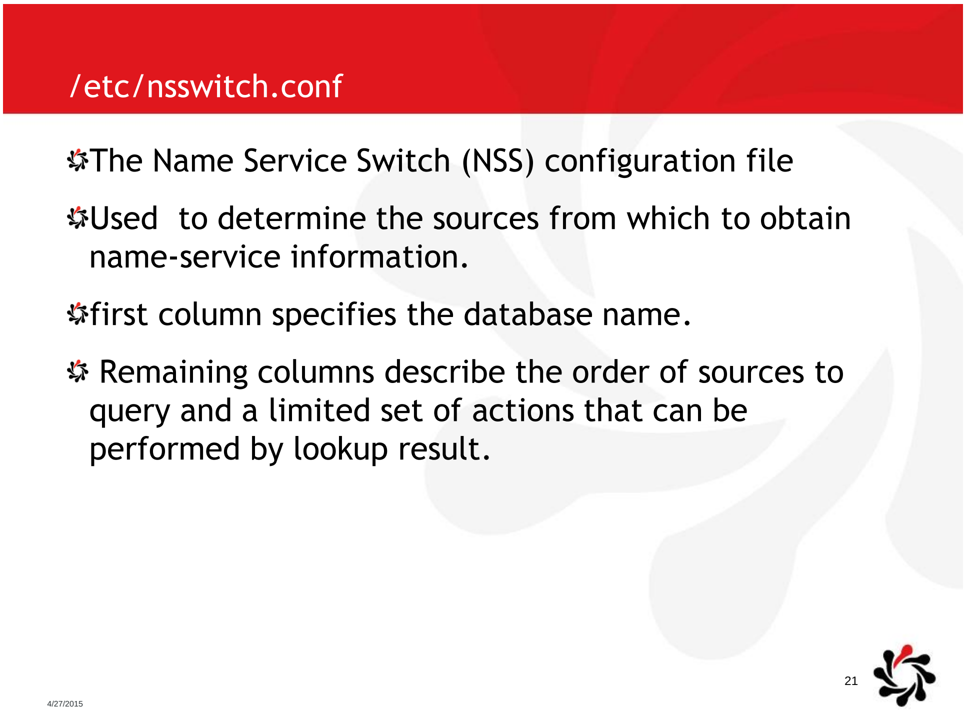
## Debian style systems

- The configuration file for a Debian system can be found at **/etc/network/interfaces**
- This would produce output similar to:auto lo eth0

```
iface lo inet loopback
iface eth0 inet static
  address 192.168.15.5
  netmask 255.255.255.0
  network 192.168.15.0
  broadcast 192.168.15.255
  gateway 192.168.15.2
```

20

4/27/2015

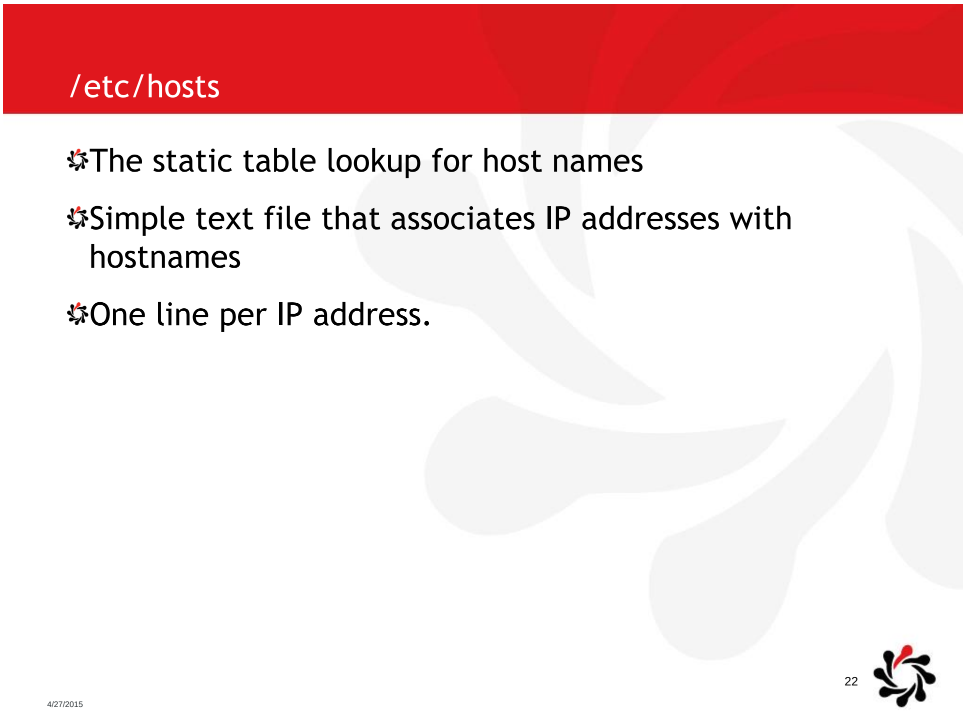


## /etc/nsswitch.conf

- The Name Service Switch (NSS) configuration file
- Used  to determine the sources from which to obtain name-service information.
- first column specifies the database name.
- Remaining columns describe the order of sources to query and a limited set of actions that can be performed by lookup result.

4/27/2015

21

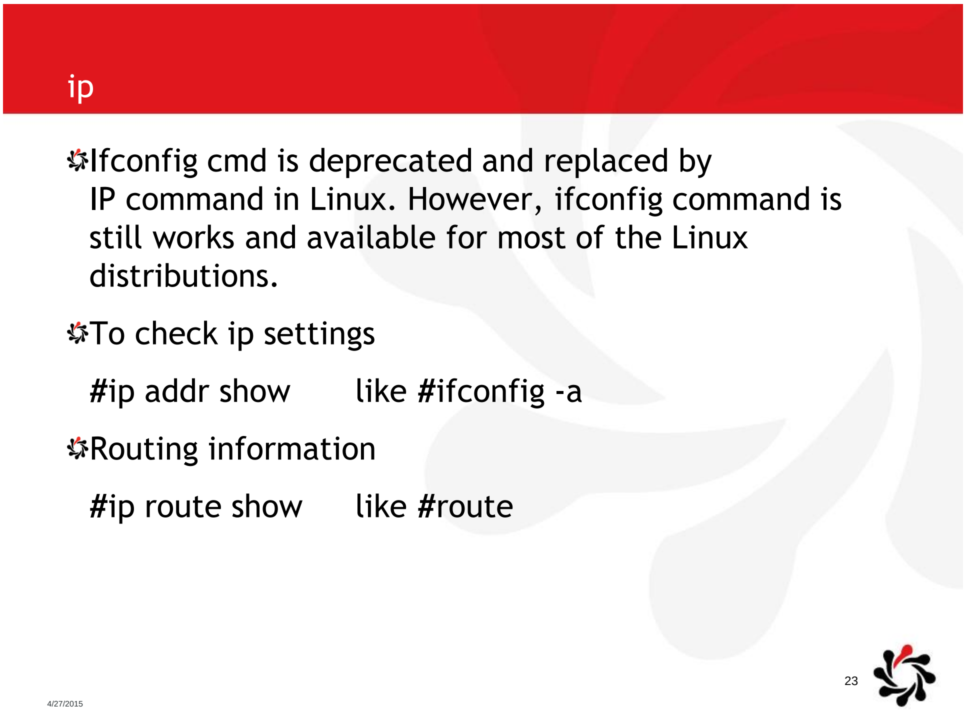


## /etc/hosts

- The static table lookup for host names
- Simple text file that associates IP addresses with hostnames
- One line per IP address.

4/27/2015

22



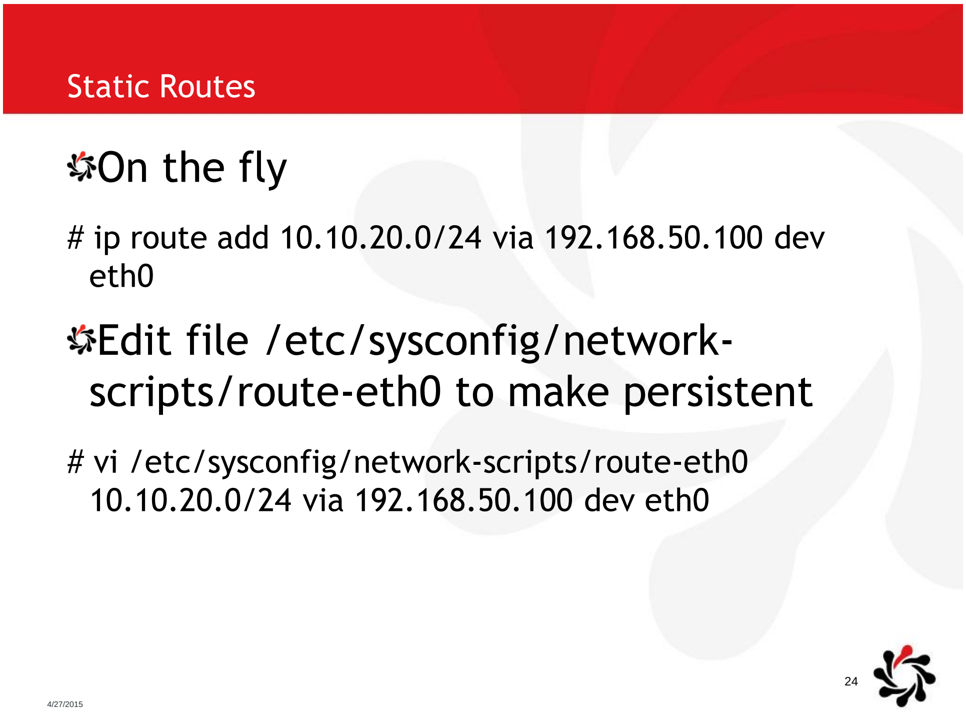
## ip

- Ifconfig cmd is deprecated and replaced by IP command in Linux. However, ifconfig command is still works and available for most of the Linux distributions.
- To check ip settings

```
#ip addr show        like #ifconfig -a
```

- Routing information

```
#ip route show       like #route
```



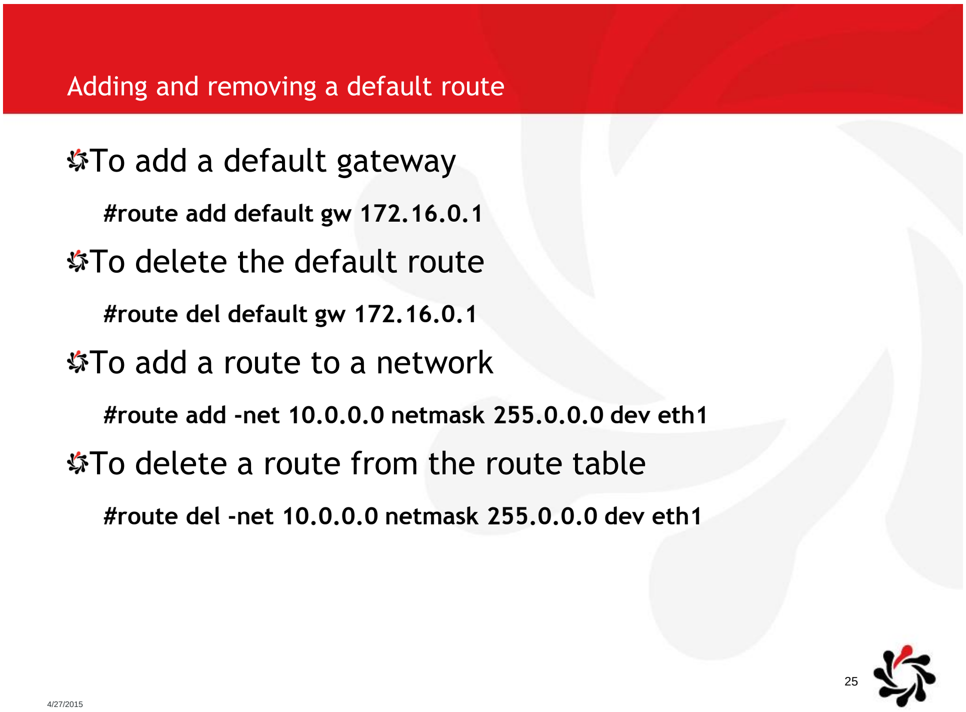
## Static Routes

- On the fly

```
# ip route add 10.10.20.0/24 via 192.168.50.100 dev eth0
```

- Edit file /etc/sysconfig/network-scripts/route-eth0 to make persistent

```
# vi /etc/sysconfig/network-scripts/route-eth0
10.10.20.0/24 via 192.168.50.100 dev eth0
```



## Adding and removing a default route

- To add a default gateway

```
#route add default gw 172.16.0.1
```

- To delete the default route

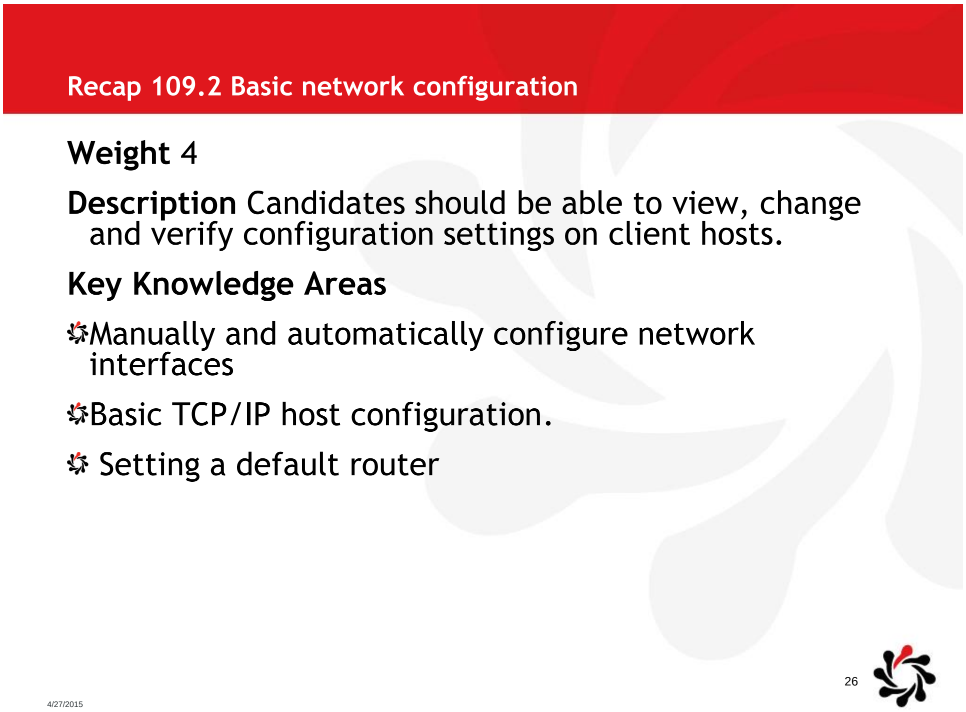
```
#route del default gw 172.16.0.1
```

- To add a route to a network

```
#route add -net 10.0.0.0 netmask 255.0.0.0 dev eth1
```

- To delete a route from the route table

```
#route del -net 10.0.0.0 netmask 255.0.0.0 dev eth1
```



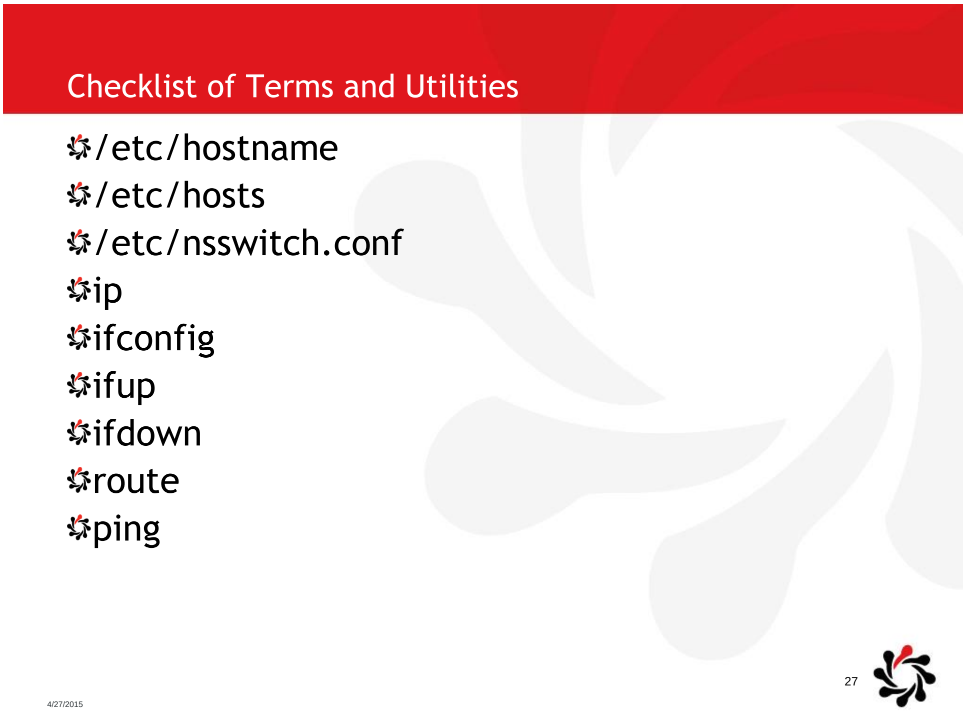
## Recap 109.2 Basic network configuration

**Weight** 4

**Description** Candidates should be able to view, change and verify configuration settings on client hosts.

**Key Knowledge Areas**

- Manually and automatically configure network interfaces
- Basic TCP/IP host configuration.
- Setting a default router

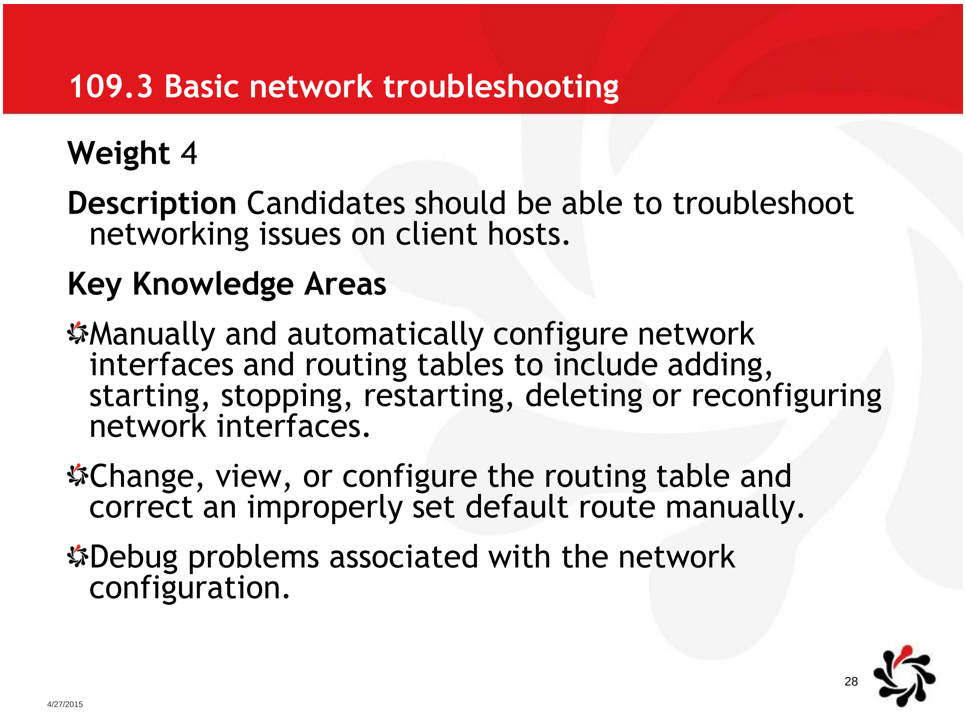


## Checklist of Terms and Utilities

- /etc/hostname
- /etc/hosts
- /etc/nsswitch.conf
- ip
- ifconfig
- ifup
- ifdown
- route
- ping

27

4/27/2015



## 109.3 Basic network troubleshooting

**Weight** 4

**Description** Candidates should be able to troubleshoot networking issues on client hosts.

**Key Knowledge Areas**

- Manually and automatically configure network interfaces and routing tables to include adding, starting, stopping, restarting, deleting or reconfiguring network interfaces.
- Change, view, or configure the routing table and correct an improperly set default route manually.
- Debug problems associated with the network configuration.

28

4/27/2015

## Basic troubleshooting

- The interface status can be listed using the ifconfig command.  The following command will list the interface configuration for eth0

```
#ifconfig eth0
```

- You can also take down the interface and bring it back up with ifup and ifdown

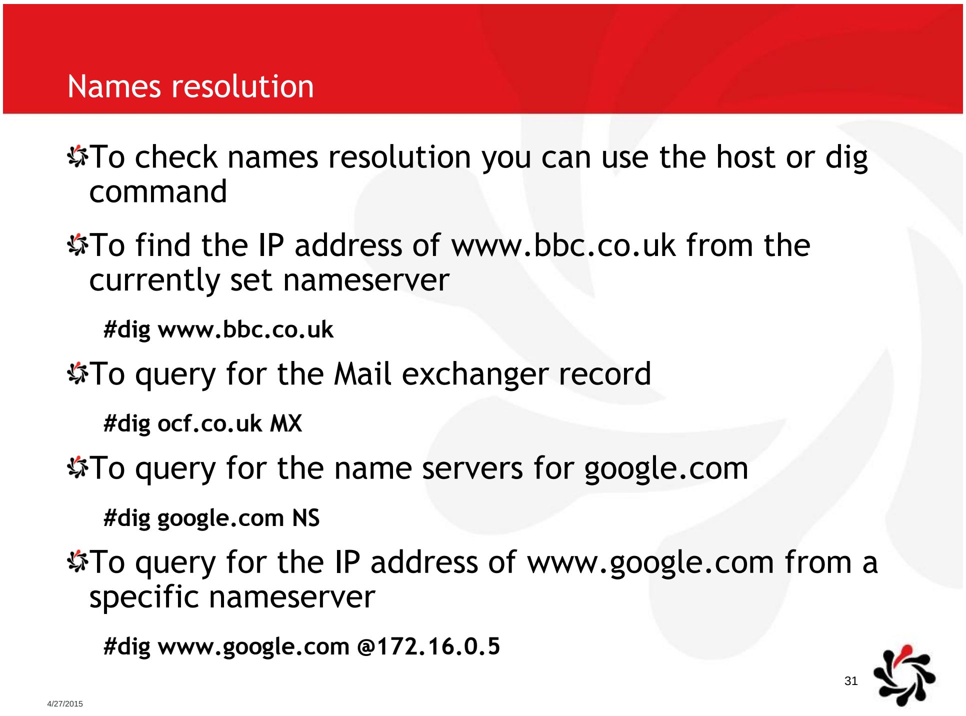
```
#ifdown eth0
#ifup eth0
```

## Routing

- The route table can be viewed with the route command

```
  #route                 or       #netstat -rn
Destination    Gateway      Genmask    Flags  Metric  Ref   Use    Iface
172.16.0.0        *      255.255.255.0   U       0       0     0      eth0
default      172.16.01     0.0.0.0      UG      0       0     0      eth0
```

- The default entry can be also seen as 0.0.0.0 under the destination, and a Genmask of 255.255.255.255 indicates the route is a host.



## Names resolution

- To check names resolution you can use the host or dig command
- To find the IP address of www.bbc.co.uk from the currently set nameserver

```
#dig www.bbc.co.uk
```

- To query for the Mail exchanger record

```
#dig ocf.co.uk MX
```

- To query for the name servers for google.com

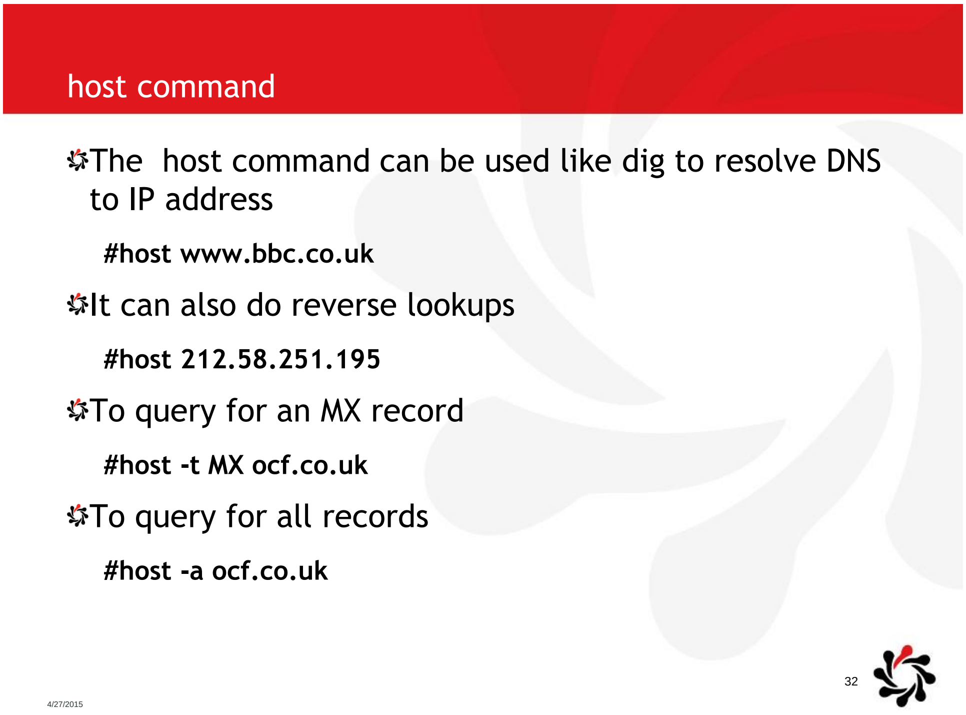
```
#dig google.com NS
```

- To query for the IP address of www.google.com from a specific nameserver

```
#dig www.google.com @172.16.0.5
```

31

4/27/2015



## host command

- The  host command can be used like dig to resolve DNS to IP address

```
#host www.bbc.co.uk
```

- It can also do reverse lookups

```
#host 212.58.251.195
```

- To query for an MX record

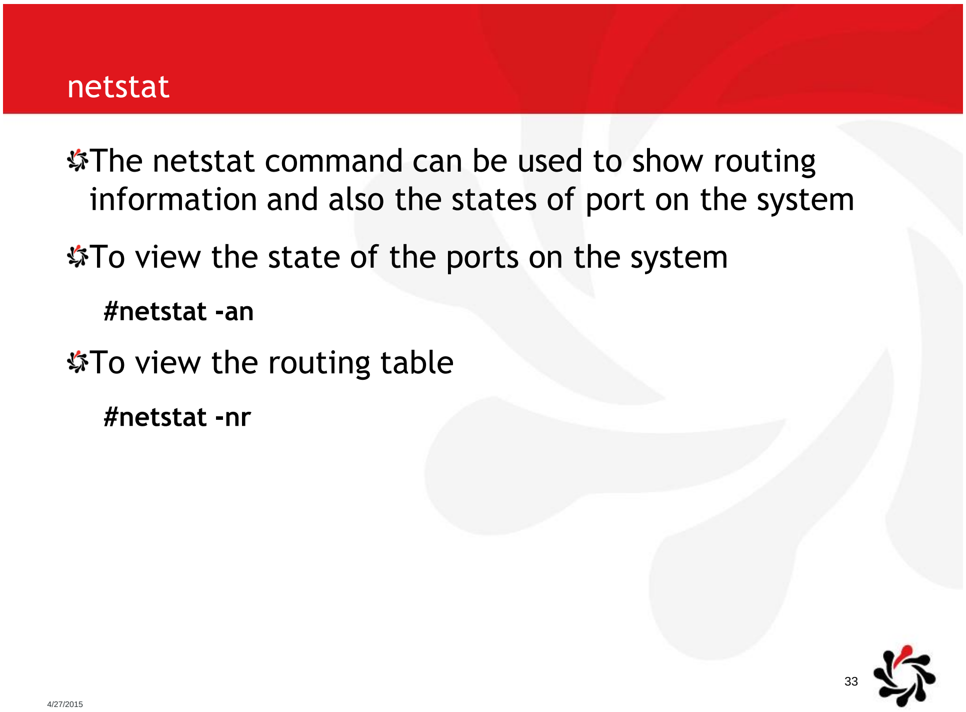
```
#host -t MX ocf.co.uk
```

- To query for all records

```
#host -a ocf.co.uk
```

32

4/27/2015



## netstat

- The netstat command can be used to show routing information and also the states of port on the system
- To view the state of the ports on the system

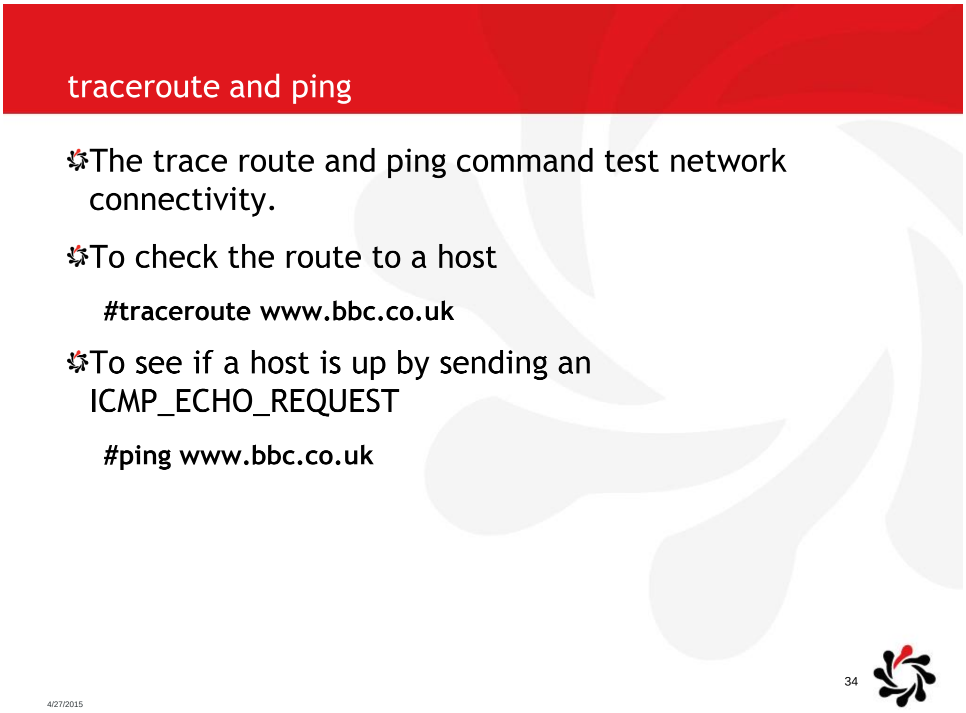
```
#netstat -an
```

- To view the routing table

```
#netstat -nr
```

4/27/2015

33



## traceroute and ping

- The trace route and ping command test network connectivity.
- To check the route to a host

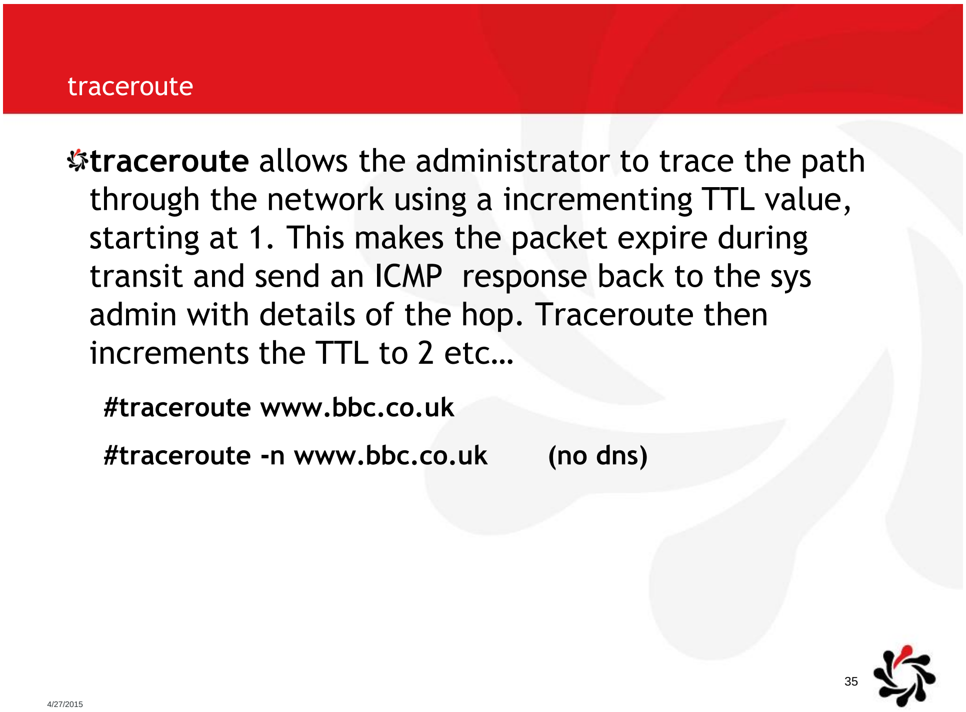
```
#traceroute www.bbc.co.uk
```

- To see if a host is up by sending an ICMP_ECHO_REQUEST

```
#ping www.bbc.co.uk
```

4/27/2015

34

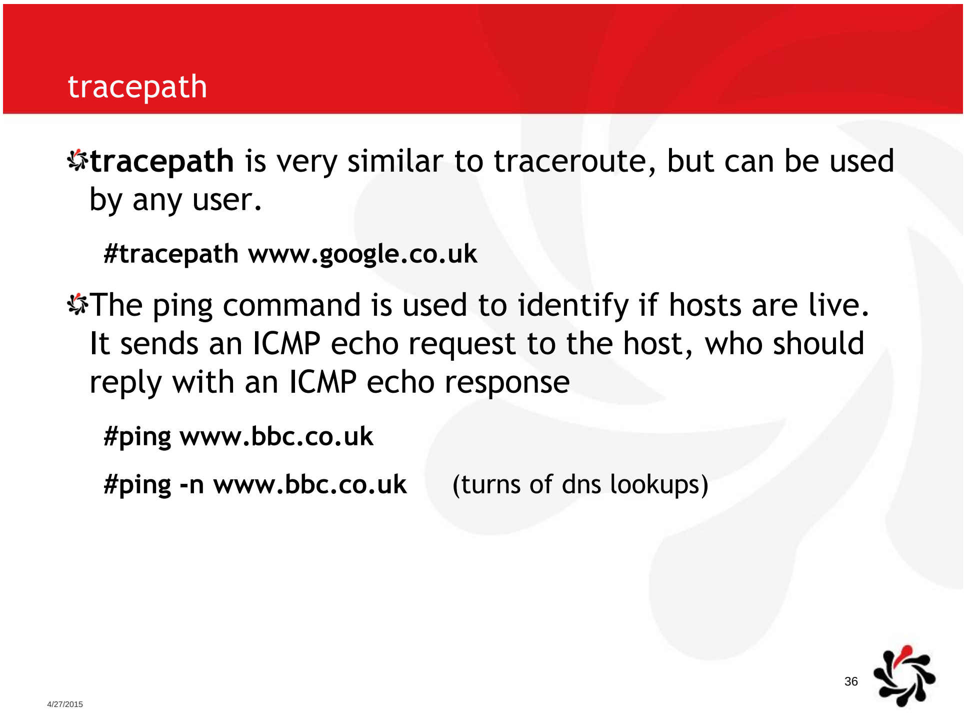


## traceroute

- **traceroute** allows the administrator to trace the path through the network using a incrementing TTL value, starting at 1. This makes the packet expire during transit and send an ICMP response back to the sys admin with details of the hop. Traceroute then increments the TTL to 2 etc…

```
#traceroute www.bbc.co.uk
#traceroute -n www.bbc.co.uk      (no dns)
```

4/27/2015 35



## tracepath

- **tracepath** is very similar to traceroute, but can be used by any user.

```
#tracepath www.google.co.uk
```

- The ping command is used to identify if hosts are live. It sends an ICMP echo request to the host, who should reply with an ICMP echo response

```
#ping www.bbc.co.uk
#ping -n www.bbc.co.uk     (turns of dns lookups)
```

4/27/2015 36

## Recap 109.3 Basic network troubleshooting

**Weight** 4

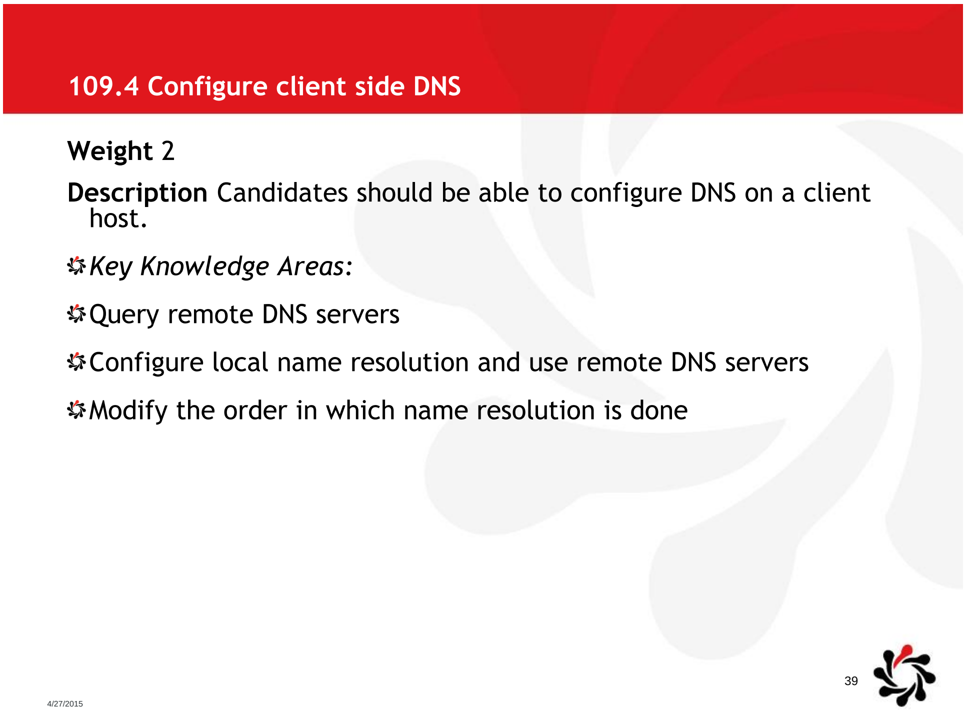
**Description** Candidates should be able to troubleshoot networking issues on client hosts.

**Key Knowledge Areas**

- Manually and automatically configure network interfaces and routing tables to include adding, starting, stopping, restarting, deleting or reconfiguring network interfaces.
- Change, view, or configure the routing table and correct an improperly set default route manually.
- Debug problems associated with the network configuration.

## Terms and Utilities Covered in this Module

- ifconfig
- ip
- ifup
- ifdown
- route
- host
- Hostname
- dig
- netstat
- ping
- traceroute



## 109.4 Configure client side DNS

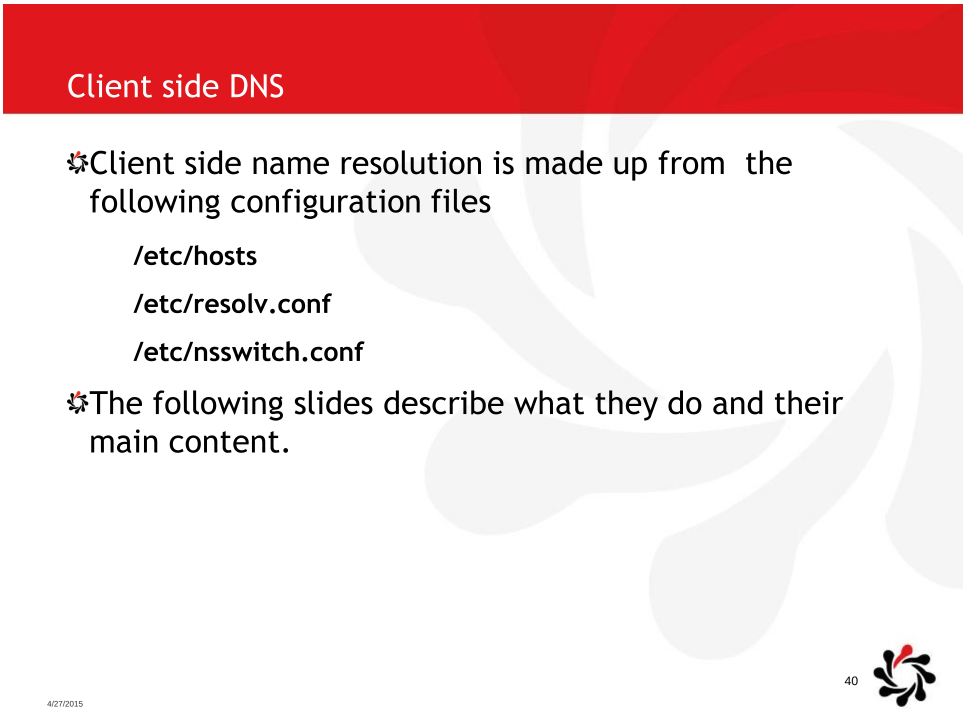
**Weight** 2

**Description** Candidates should be able to configure DNS on a client host.

- *Key Knowledge Areas:*
- Query remote DNS servers
- Configure local name resolution and use remote DNS servers
- Modify the order in which name resolution is done

4/27/2015

39



## Client side DNS

- Client side name resolution is made up from  the following configuration files

  **/etc/hosts**

  **/etc/resolv.conf**

  **/etc/nsswitch.conf**

- The following slides describe what they do and their main content.

4/27/2015

40

## /etc/nsswitch.conf

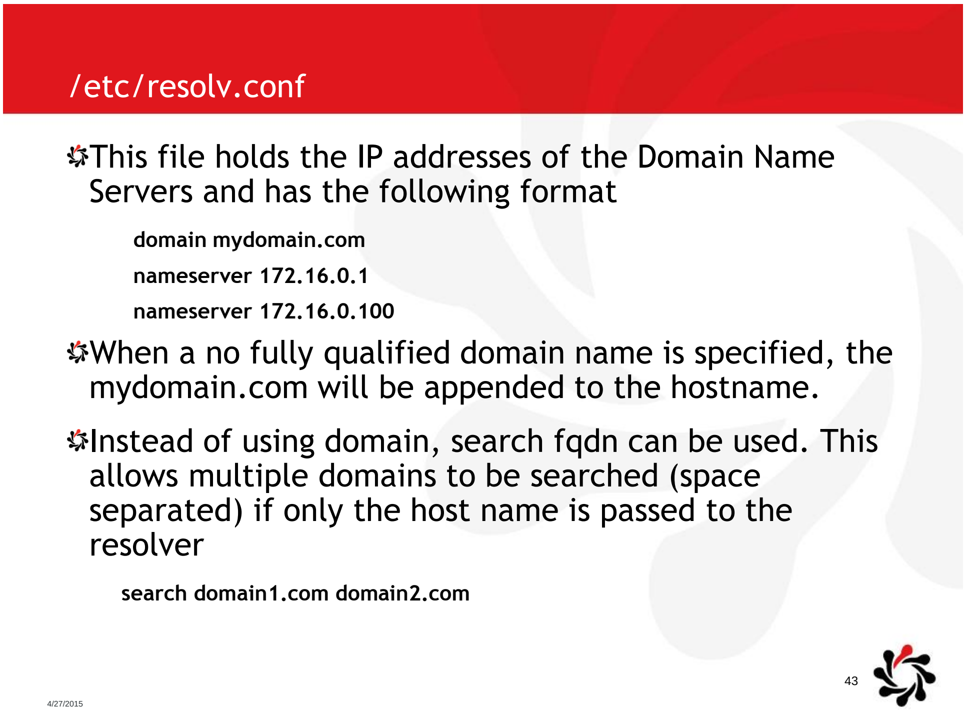
- The name server switch file defines the order of resolution for resolving DNS names. There are many lines, but the important one for names resolution is

```
hosts          files dns nis
```

## /etc/hosts

- This file is used to statically map IP addresses to host names.
- The format of the file is

```
172.16.0.5  server.lpi.org   server
```



## /etc/resolv.conf

- This file holds the IP addresses of the Domain Name Servers and has the following format

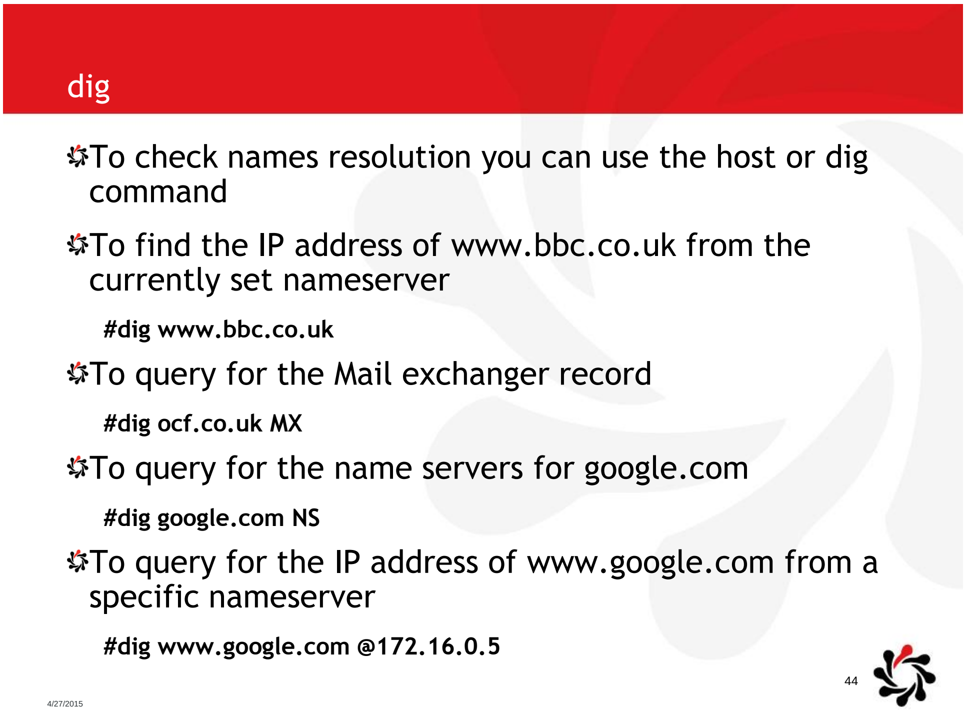
```
domain mydomain.com
nameserver 172.16.0.1
nameserver 172.16.0.100
```

- When a no fully qualified domain name is specified, the mydomain.com will be appended to the hostname.
- Instead of using domain, search fqdn can be used. This allows multiple domains to be searched (space separated) if only the host name is passed to the resolver

```
search domain1.com domain2.com
```

4/27/2015

43



## dig

- To check names resolution you can use the host or dig command
- To find the IP address of www.bbc.co.uk from the currently set nameserver

```
#dig www.bbc.co.uk
```

- To query for the Mail exchanger record

```
#dig ocf.co.uk MX
```

- To query for the name servers for google.com

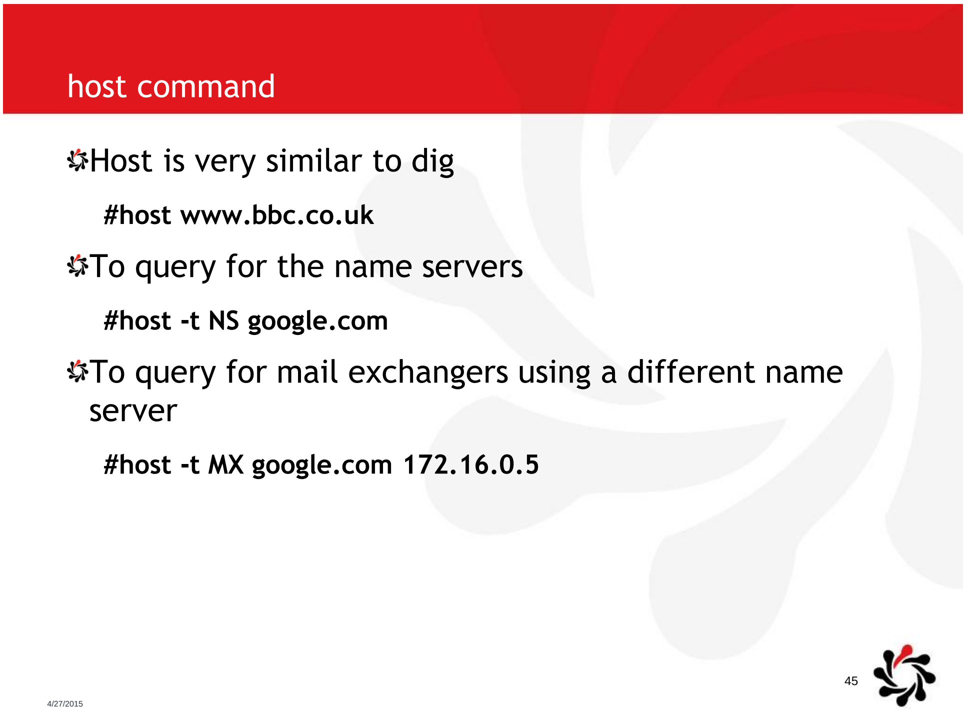
```
#dig google.com NS
```

- To query for the IP address of www.google.com from a specific nameserver

```
#dig www.google.com @172.16.0.5
```

4/27/2015

44



## host command

- Host is very similar to dig

```
#host www.bbc.co.uk
```

- To query for the name servers

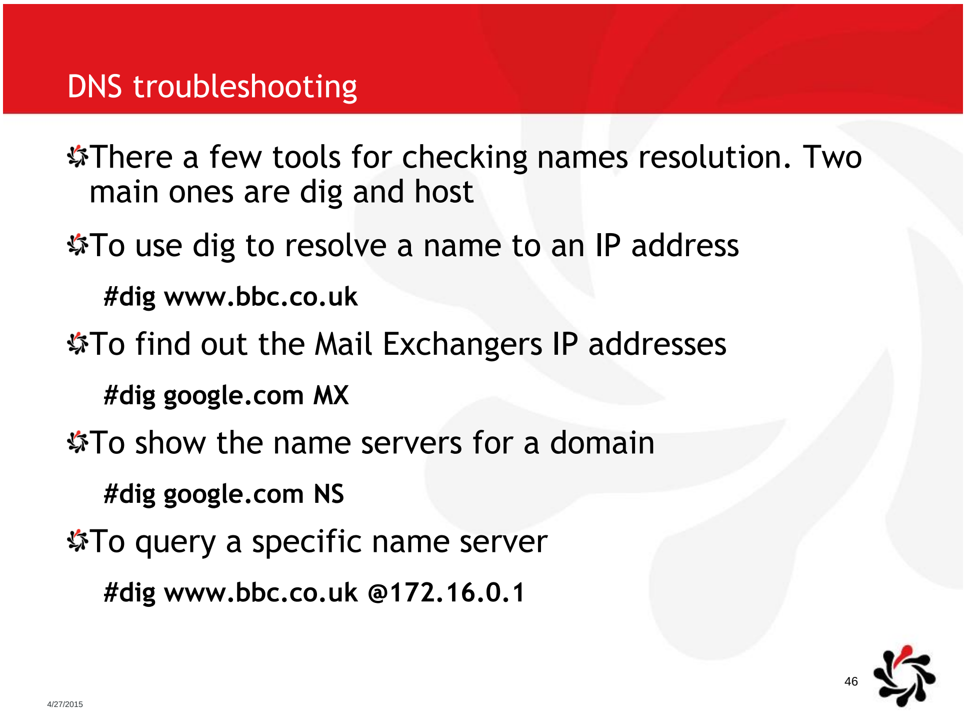
```
#host -t NS google.com
```

- To query for mail exchangers using a different name server

```
#host -t MX google.com 172.16.0.5
```

45

4/27/2015



## DNS troubleshooting

- There a few tools for checking names resolution. Two main ones are dig and host
- To use dig to resolve a name to an IP address

```
#dig www.bbc.co.uk
```

- To find out the Mail Exchangers IP addresses

```
#dig google.com MX
```

- To show the name servers for a domain

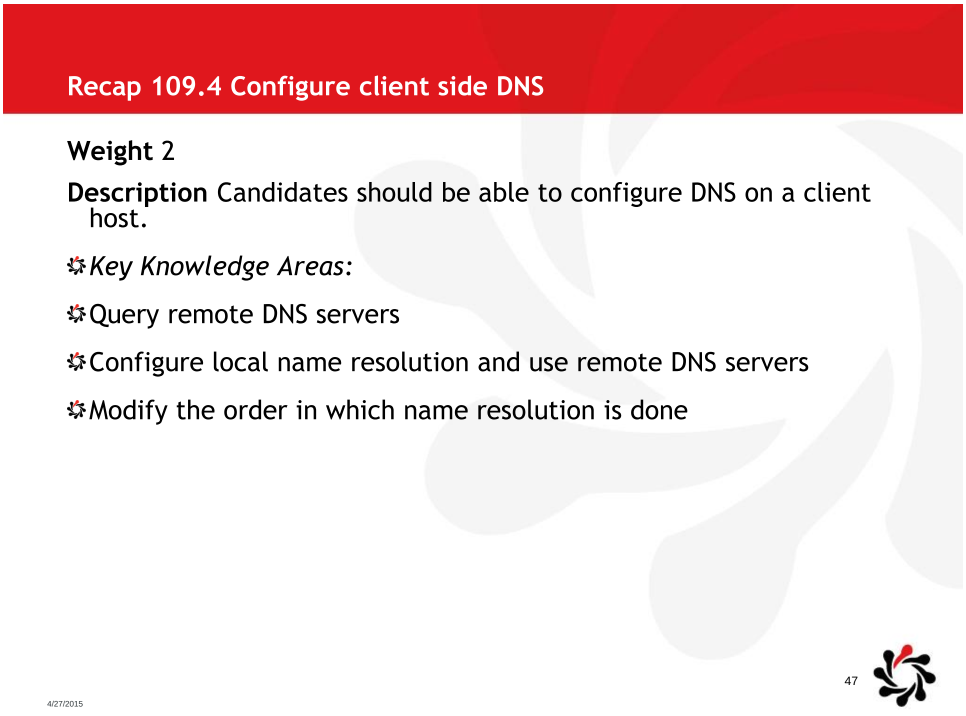
```
#dig google.com NS
```

- To query a specific name server

```
#dig www.bbc.co.uk @172.16.0.1
```

46

4/27/2015



## Recap 109.4 Configure client side DNS

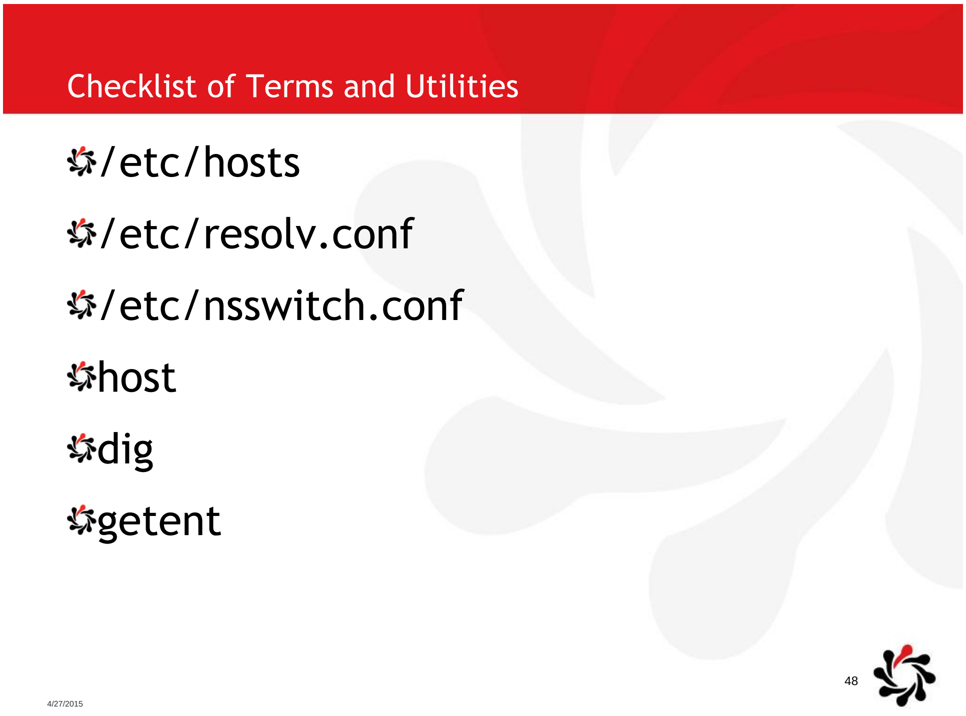
**Weight** 2

**Description** Candidates should be able to configure DNS on a client host.

- *Key Knowledge Areas:*
- Query remote DNS servers
- Configure local name resolution and use remote DNS servers
- Modify the order in which name resolution is done

47

4/27/2015

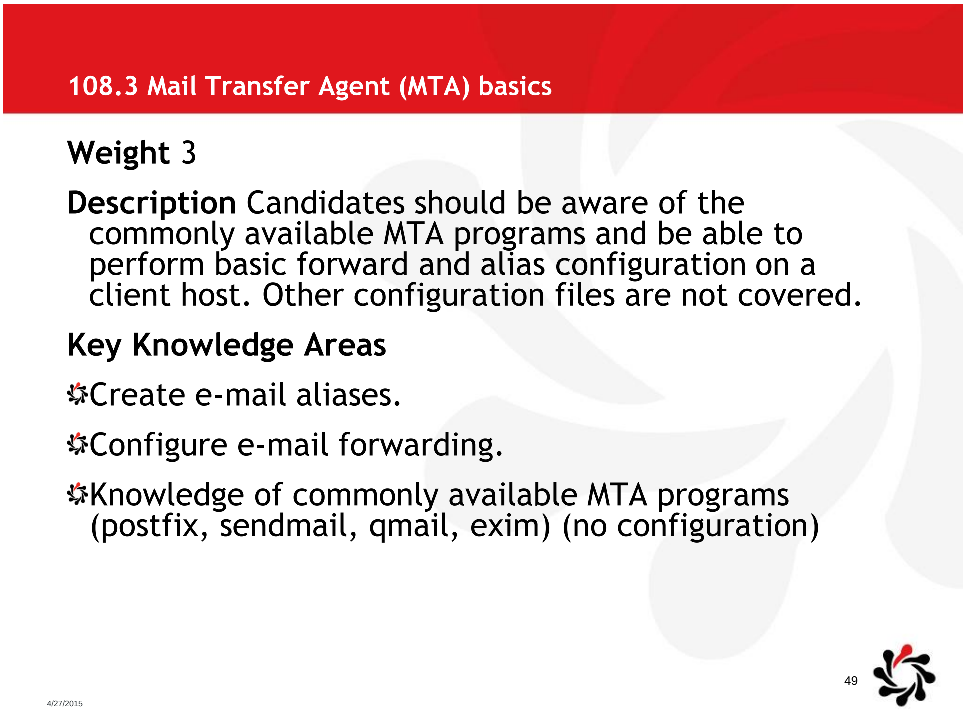


## Checklist of Terms and Utilities

- /etc/hosts
- /etc/resolv.conf
- /etc/nsswitch.conf
- host
- dig
- getent

48

4/27/2015



## 108.3 Mail Transfer Agent (MTA) basics

**Weight** 3

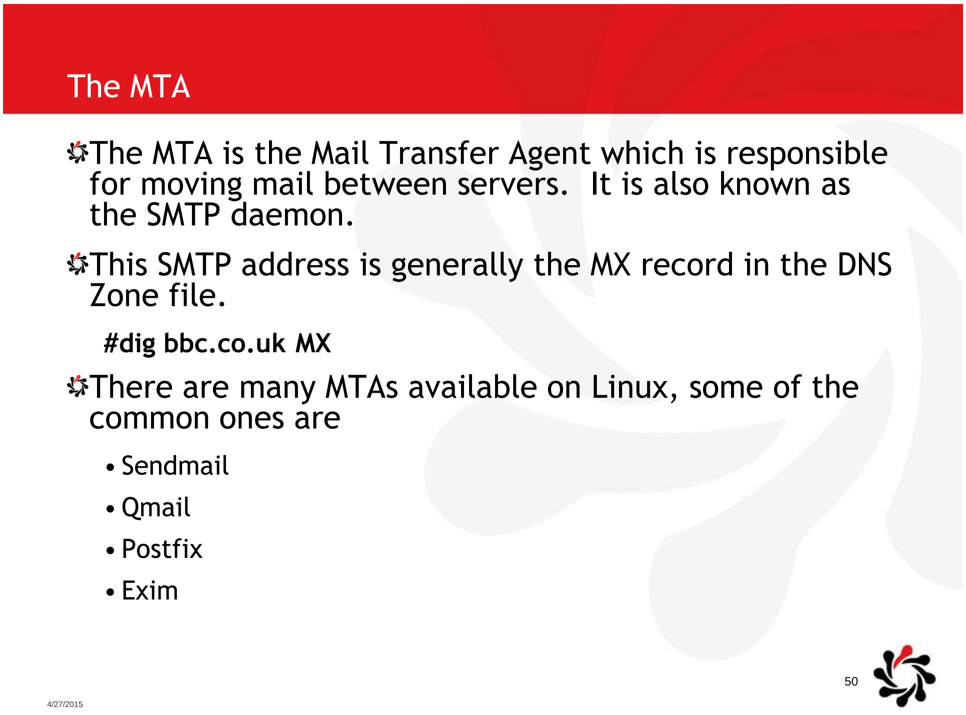
**Description** Candidates should be aware of the commonly available MTA programs and be able to perform basic forward and alias configuration on a client host. Other configuration files are not covered.

**Key Knowledge Areas**

- Create e-mail aliases.
- Configure e-mail forwarding.
- Knowledge of commonly available MTA programs (postfix, sendmail, qmail, exim) (no configuration)

4/27/2015

49



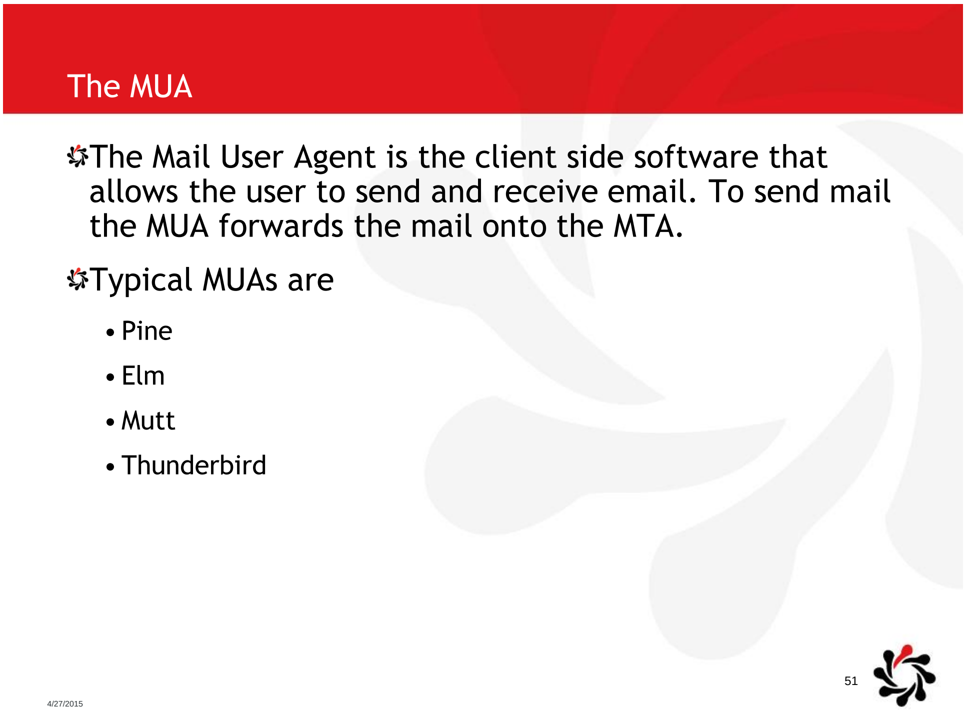
## The MTA

- The MTA is the Mail Transfer Agent which is responsible for moving mail between servers.  It is also known as the SMTP daemon.
- This SMTP address is generally the MX record in the DNS Zone file.

  **#dig bbc.co.uk MX**
- There are many MTAs available on Linux, some of the common ones are
  - Sendmail
  - Qmail
  - Postfix
  - Exim

4/27/2015

50

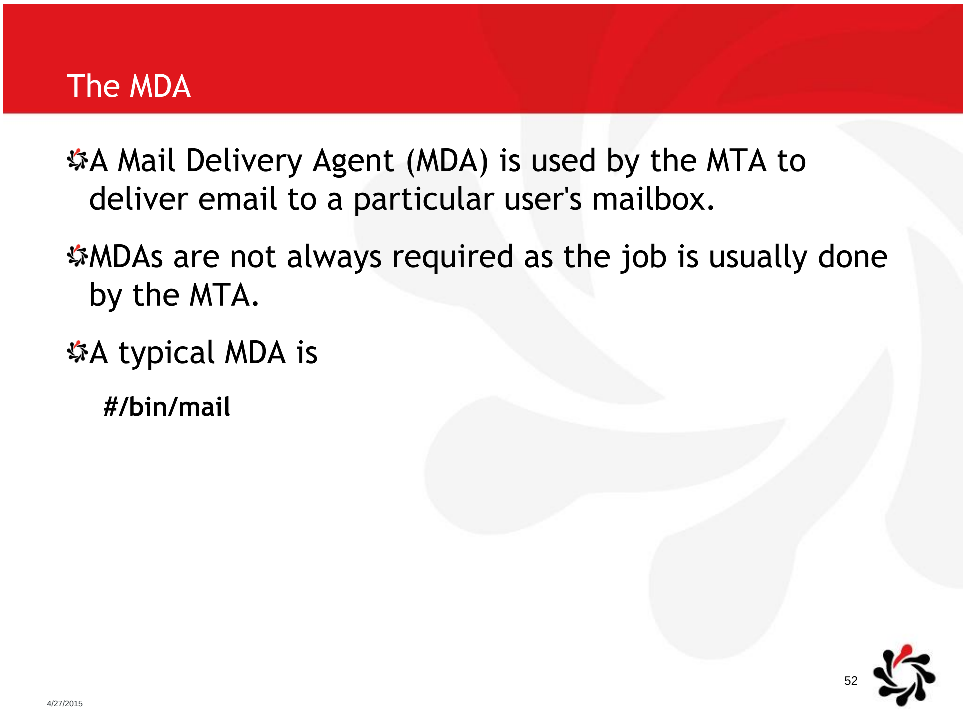


## The MUA

- The Mail User Agent is the client side software that allows the user to send and receive email. To send mail the MUA forwards the mail onto the MTA.
- Typical MUAs are
  - Pine
  - Elm
  - Mutt
  - Thunderbird

4/27/2015

51



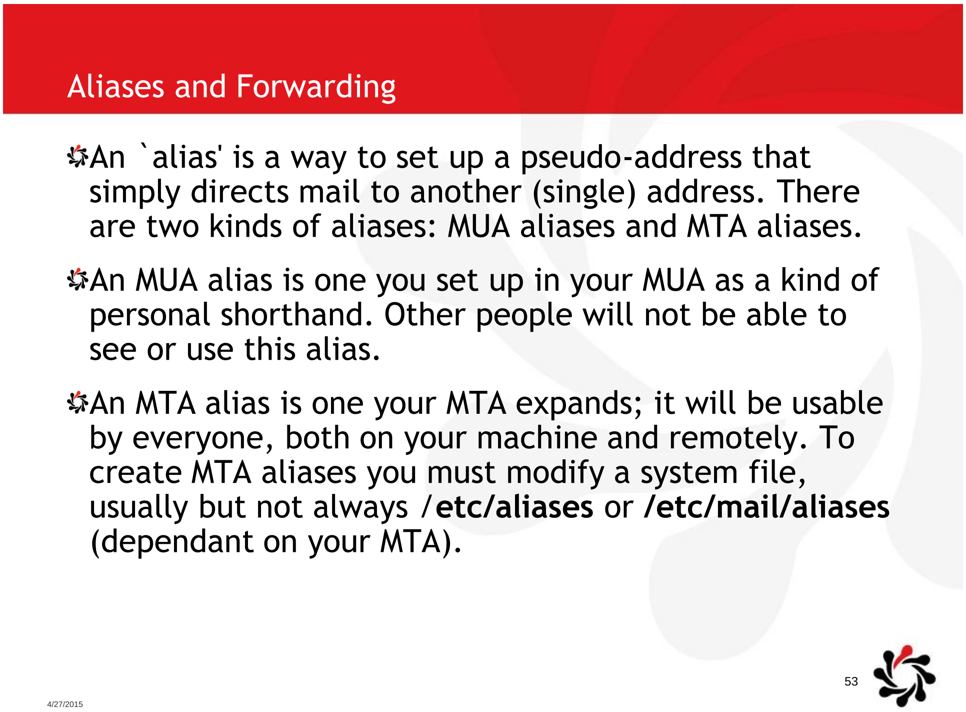
## The MDA

- A Mail Delivery Agent (MDA) is used by the MTA to deliver email to a particular user's mailbox.
- MDAs are not always required as the job is usually done by the MTA.
- A typical MDA is

**#/bin/mail**

4/27/2015

52

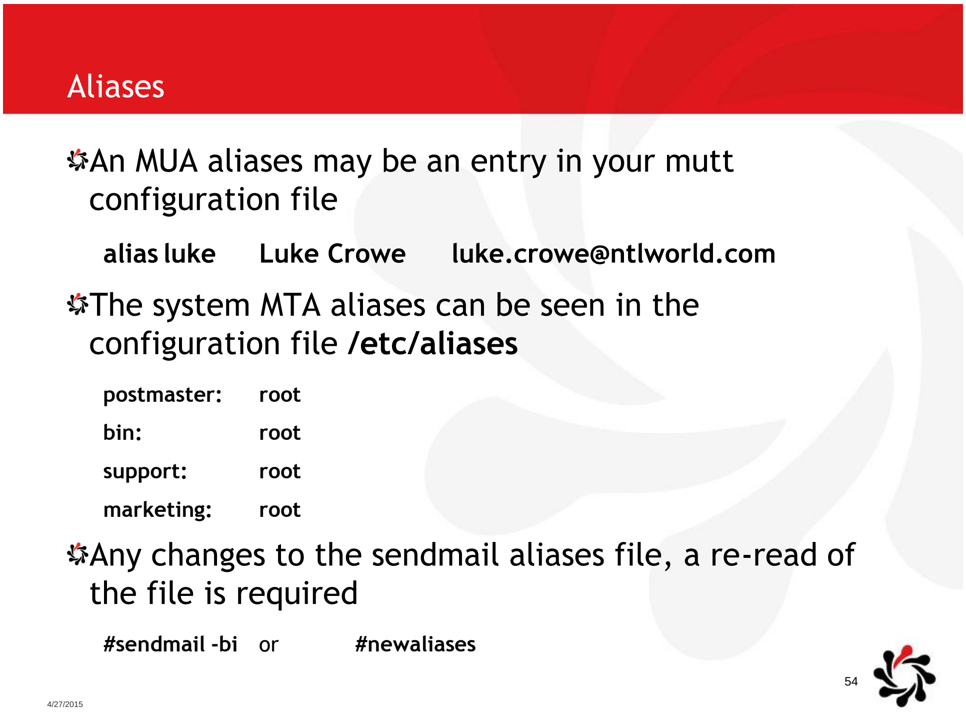


## Aliases and Forwarding

- An `alias' is a way to set up a pseudo-address that simply directs mail to another (single) address. There are two kinds of aliases: MUA aliases and MTA aliases.
- An MUA alias is one you set up in your MUA as a kind of personal shorthand. Other people will not be able to see or use this alias.
- An MTA alias is one your MTA expands; it will be usable by everyone, both on your machine and remotely. To create MTA aliases you must modify a system file, usually but not always /**etc/aliases** or **/etc/mail/aliases** (dependant on your MTA).

4/27/2015

53



## Aliases

- An MUA aliases may be an entry in your mutt configuration file

```
alias luke      Luke Crowe      luke.crowe@ntlworld.com
```

- The system MTA aliases can be seen in the configuration file **/etc/aliases**

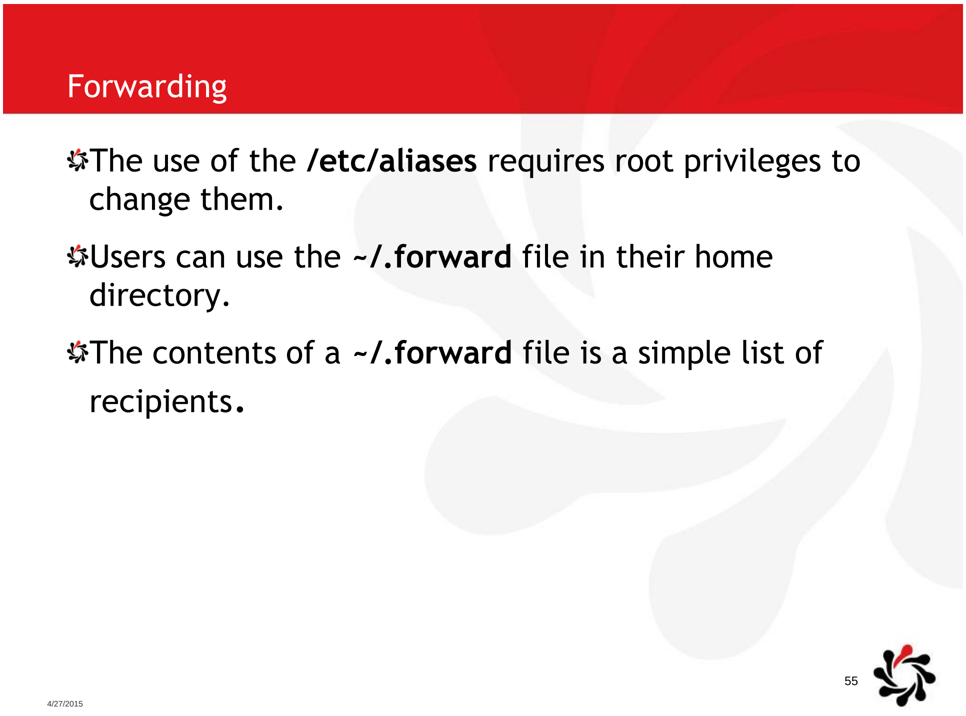
```
postmaster:     root
bin:            root
support:        root
marketing:      root
```

- Any changes to the sendmail aliases file, a re-read of the file is required

```
#sendmail -bi   or       #newaliases
```

4/27/2015

54

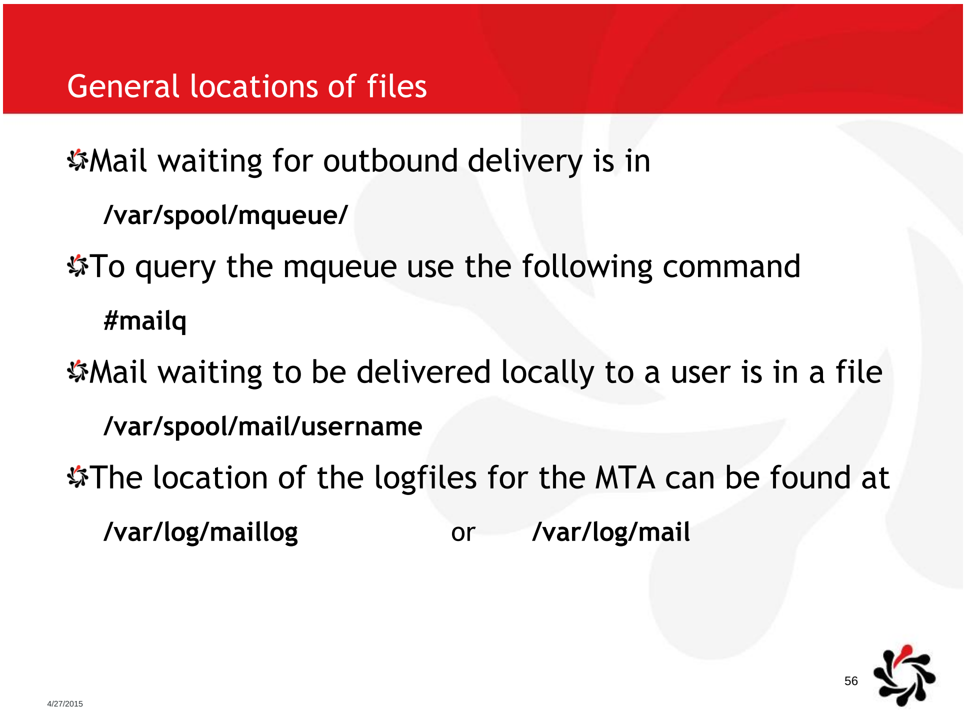


## Forwarding

- The use of the **/etc/aliases** requires root privileges to change them.
- Users can use the **~/.forward** file in their home directory.
- The contents of a **~/.forward** file is a simple list of recipients.

4/27/2015

55



## General locations of files

- Mail waiting for outbound delivery is in

  **/var/spool/mqueue/**
- To query the mqueue use the following command

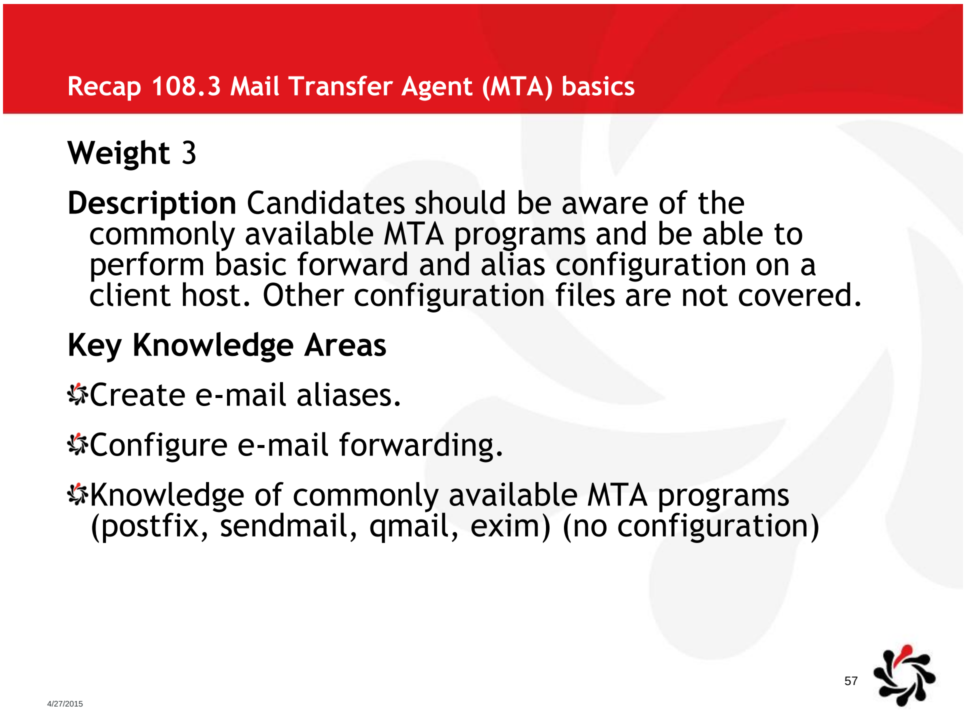
  **#mailq**
- Mail waiting to be delivered locally to a user is in a file

  **/var/spool/mail/username**
- The location of the logfiles for the MTA can be found at

  **/var/log/maillog** or **/var/log/mail**

4/27/2015

56



## Recap 108.3 Mail Transfer Agent (MTA) basics

**Weight** 3

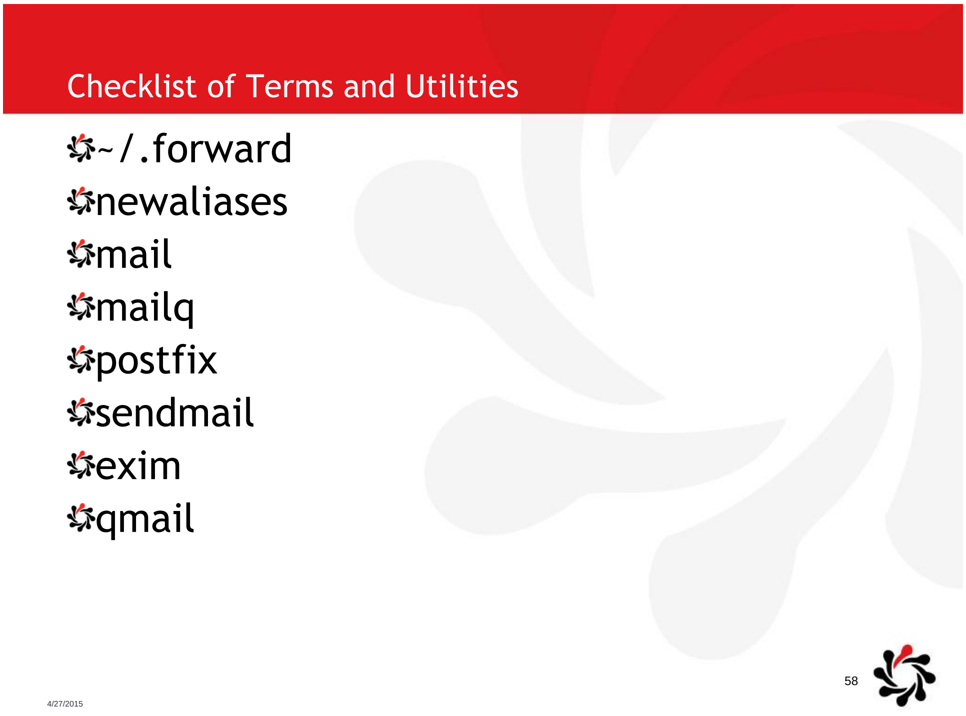
**Description** Candidates should be aware of the commonly available MTA programs and be able to perform basic forward and alias configuration on a client host. Other configuration files are not covered.

**Key Knowledge Areas**

- Create e-mail aliases.
- Configure e-mail forwarding.
- Knowledge of commonly available MTA programs (postfix, sendmail, qmail, exim) (no configuration)

57

4/27/2015

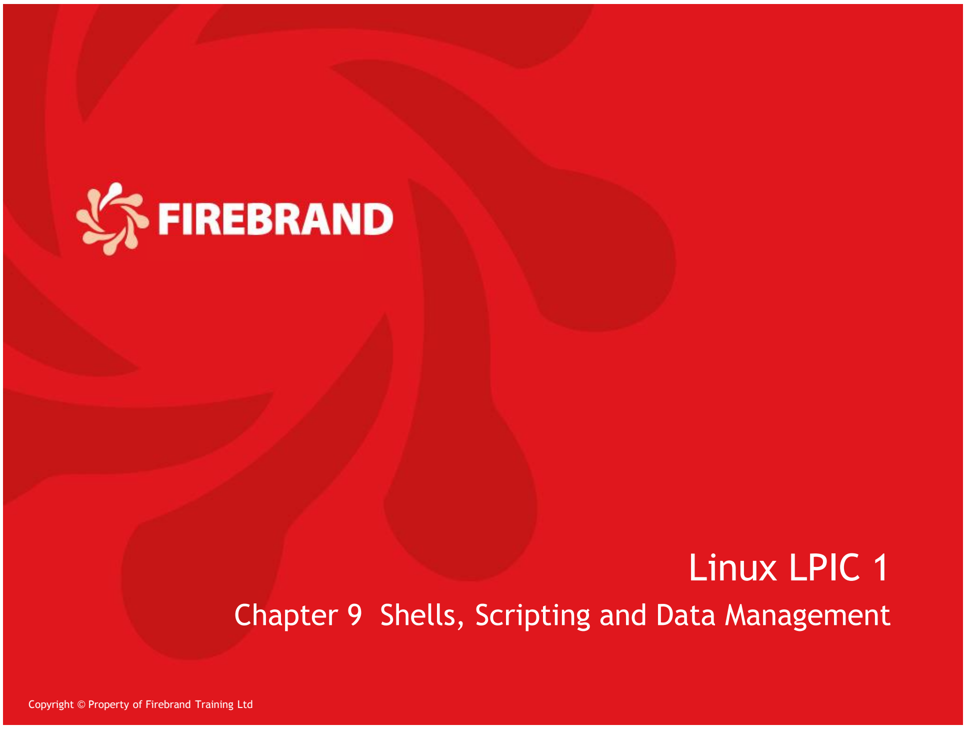


## Checklist of Terms and Utilities

- ~/.forward
- newaliases
- mail
- mailq
- postfix
- sendmail
- exim
- qmail

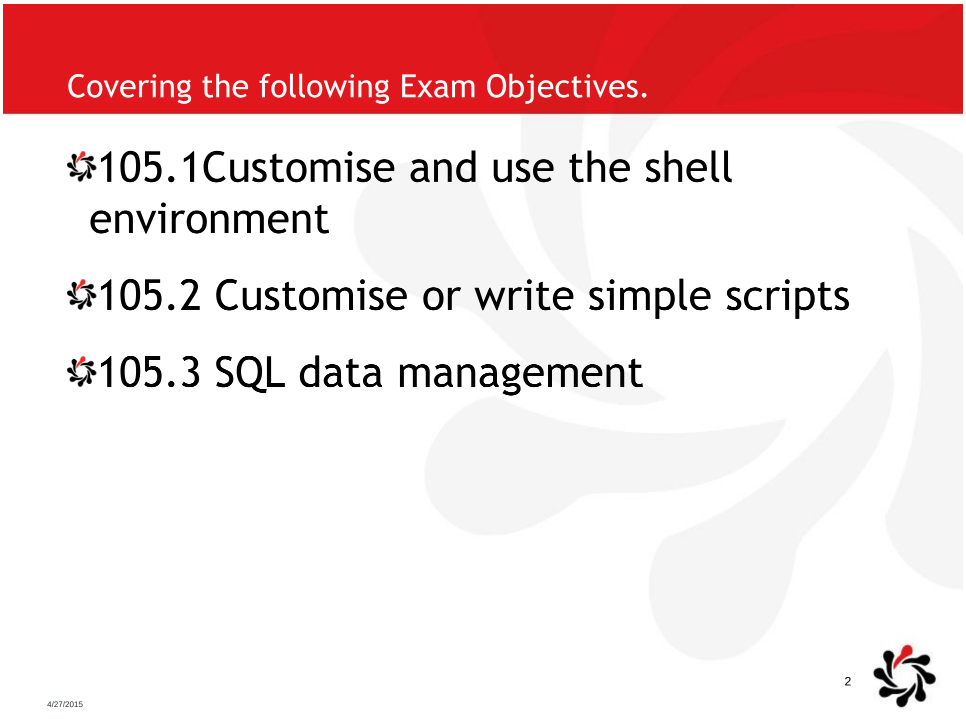
58

4/27/2015



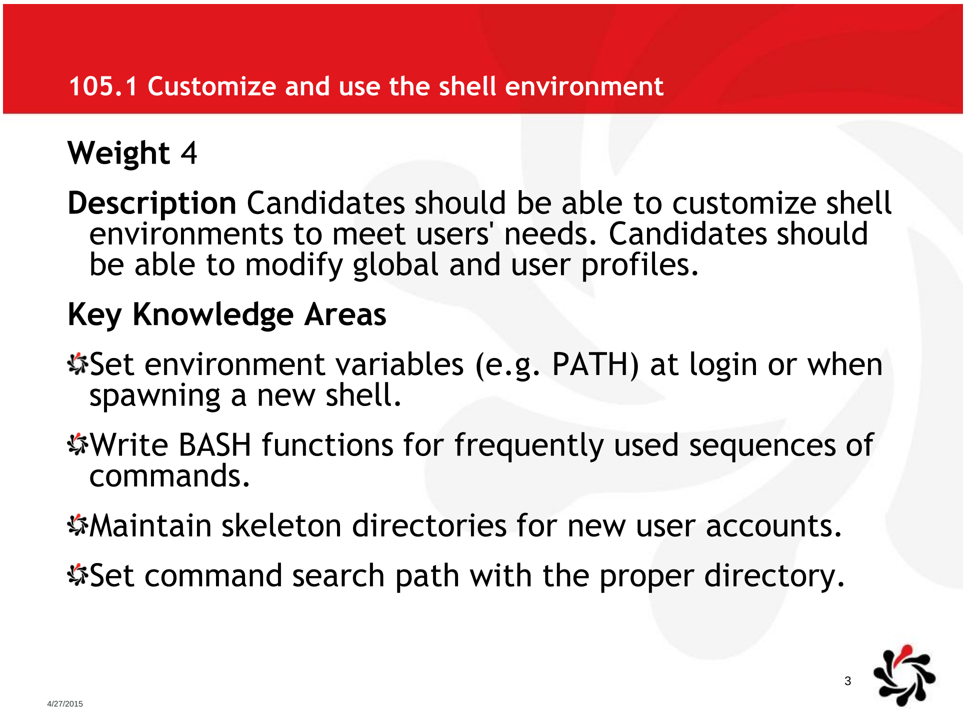
# Linux LPIC 1

## Chapter 9  Shells, Scripting and Data Management



## Covering the following Exam Objectives.

- 105.1Customise and use the shell environment
- 105.2 Customise or write simple scripts
- 105.3 SQL data management



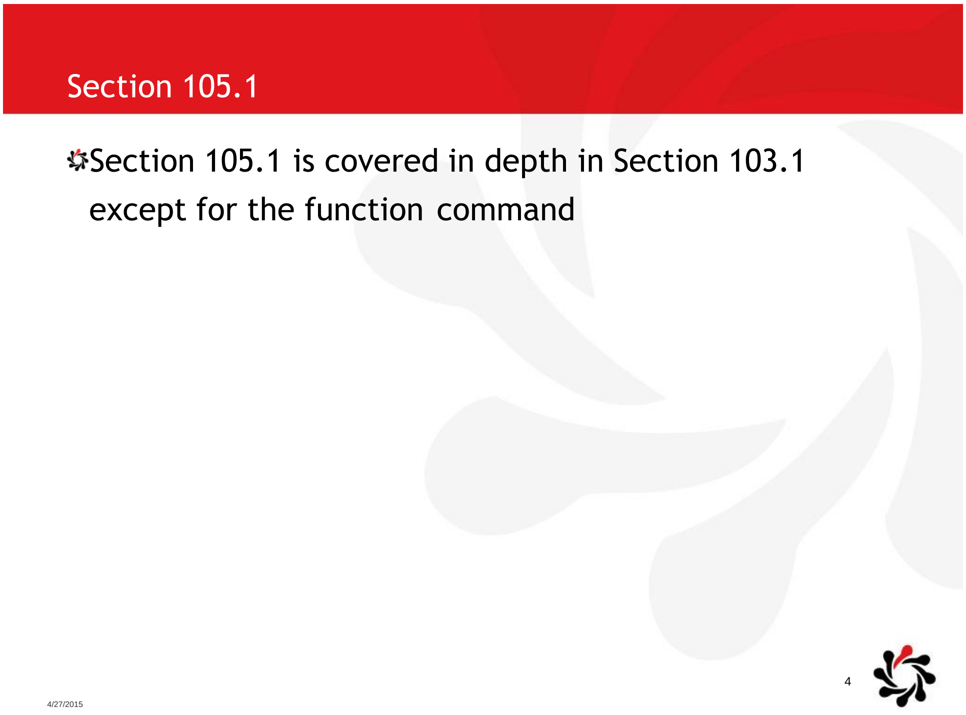
## 105.1 Customize and use the shell environment

**Weight** 4

**Description** Candidates should be able to customize shell environments to meet users' needs. Candidates should be able to modify global and user profiles.

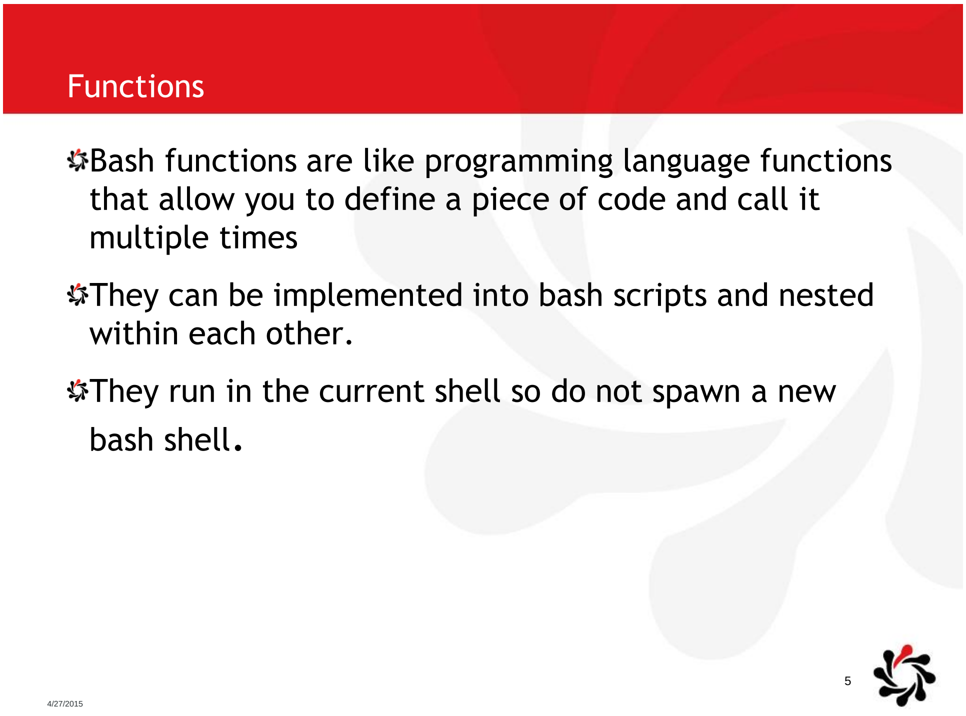
**Key Knowledge Areas**

- Set environment variables (e.g. PATH) at login or when spawning a new shell.
- Write BASH functions for frequently used sequences of commands.
- Maintain skeleton directories for new user accounts.
- Set command search path with the proper directory.



## Section 105.1

- Section 105.1 is covered in depth in Section 103.1 except for the function command

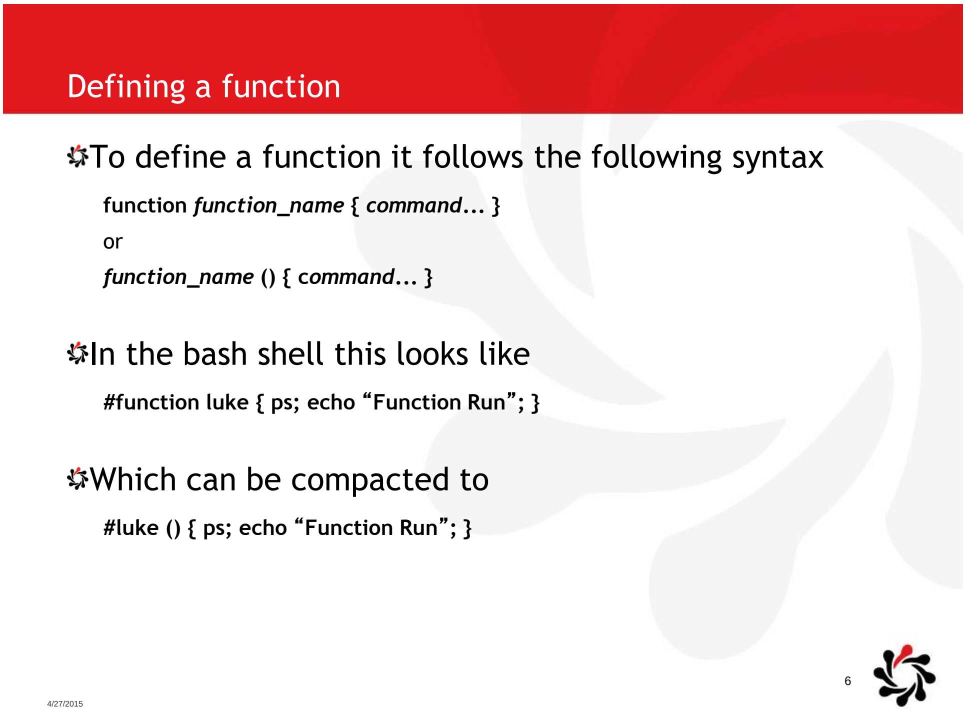


## Functions

- Bash functions are like programming language functions that allow you to define a piece of code and call it multiple times
- They can be implemented into bash scripts and nested within each other.
- They run in the current shell so do not spawn a new bash shell.

4/27/2015

5



## Defining a function

- To define a function it follows the following syntax

```
function function_name { command... }
or
function_name () { command... }
```

- In the bash shell this looks like

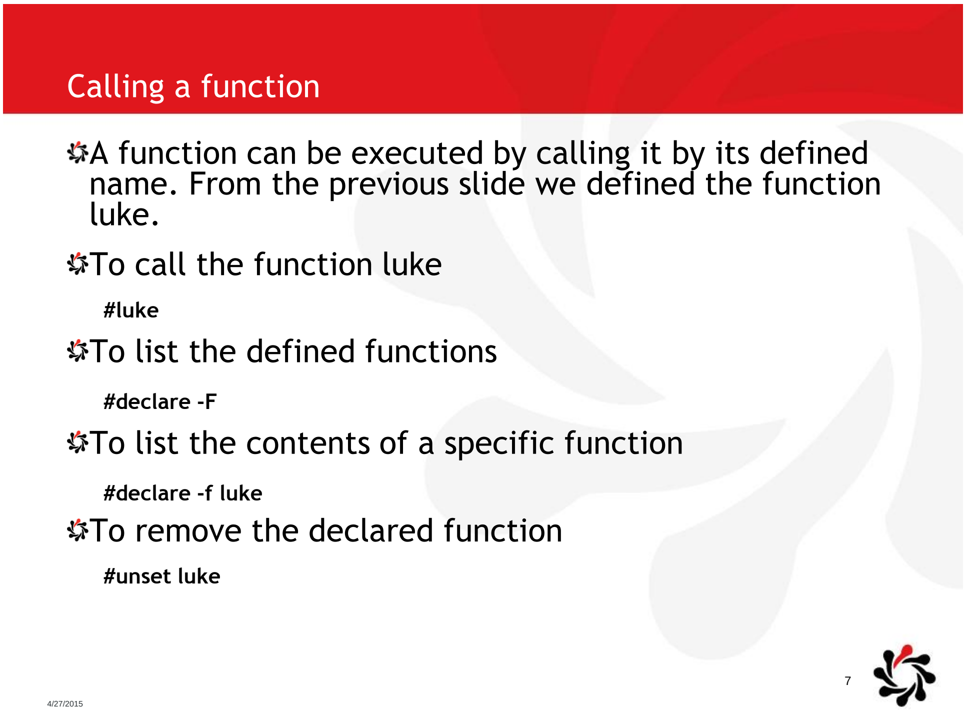
```
#function luke { ps; echo “Function Run”; }
```

- Which can be compacted to

```
#luke () { ps; echo “Function Run”; }
```

4/27/2015

6



## Calling a function

- A function can be executed by calling it by its defined name. From the previous slide we defined the function luke.
- To call the function luke

```
#luke
```

- To list the defined functions

```
#declare -F
```

- To list the contents of a specific function

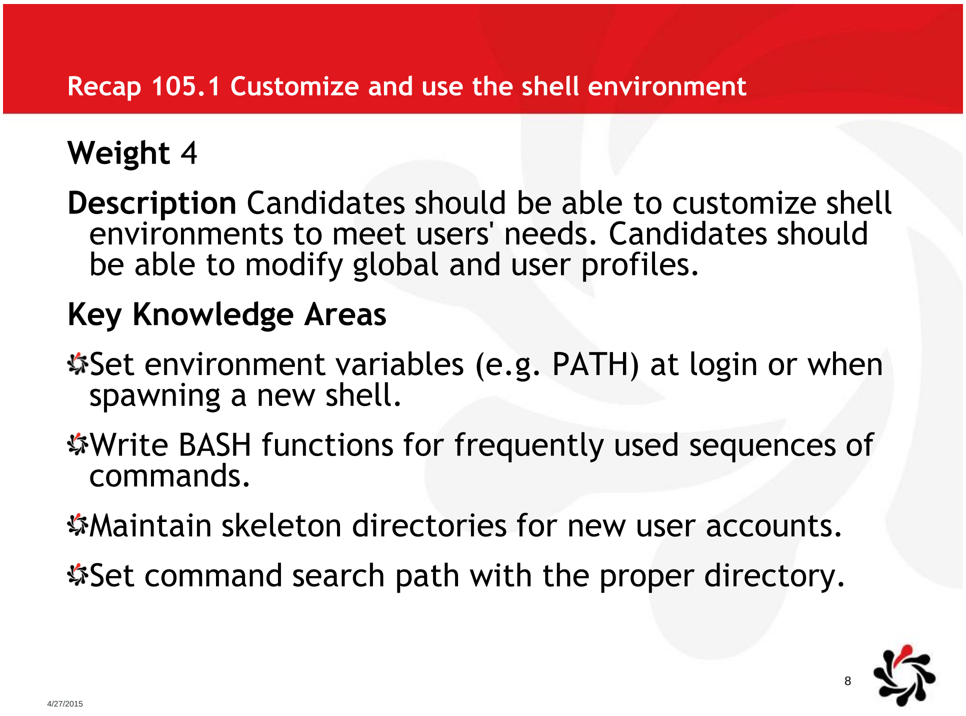
```
#declare -f luke
```

- To remove the declared function

```
#unset luke
```

4/27/2015

7



## Recap 105.1 Customize and use the shell environment

**Weight** 4

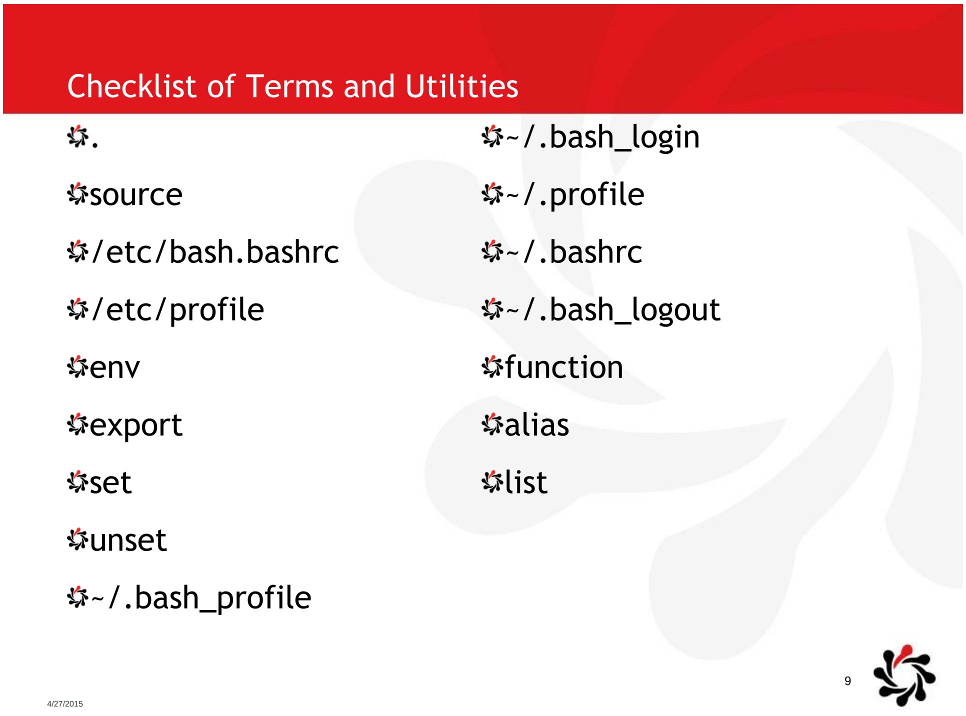
**Description** Candidates should be able to customize shell environments to meet users' needs. Candidates should be able to modify global and user profiles.

**Key Knowledge Areas**

- Set environment variables (e.g. PATH) at login or when spawning a new shell.
- Write BASH functions for frequently used sequences of commands.
- Maintain skeleton directories for new user accounts.
- Set command search path with the proper directory.

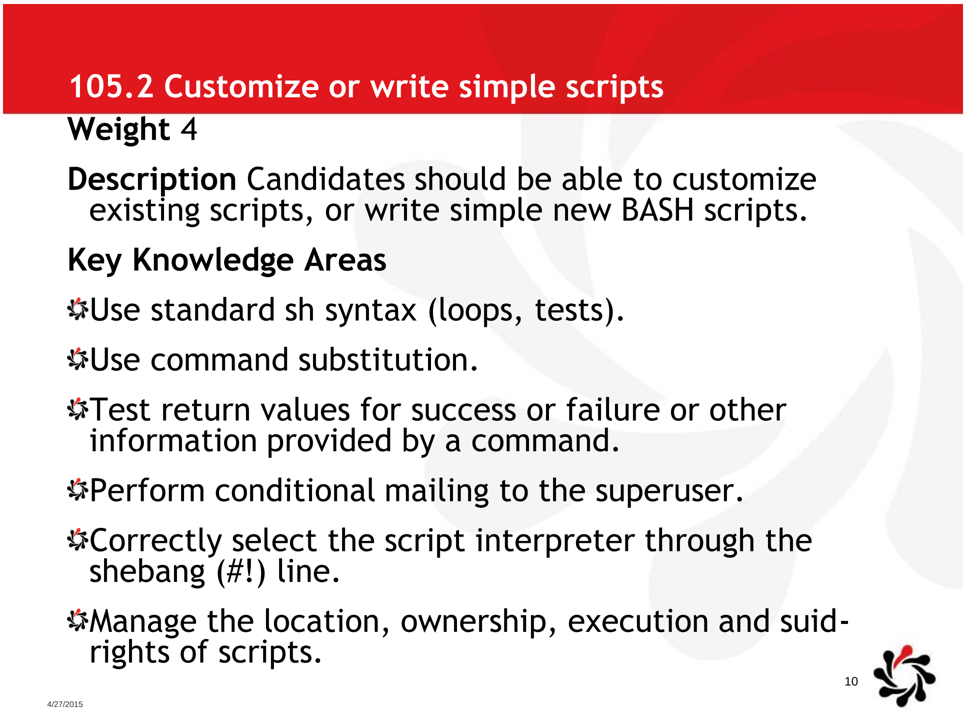
4/27/2015

8



## Checklist of Terms and Utilities

- .
- source
- /etc/bash.bashrc
- /etc/profile
- env
- export
- set
- unset
- ~/.bash_profile
- ~/.bash_login
- ~/.profile
- ~/.bashrc
- ~/.bash_logout
- function
- alias
- list



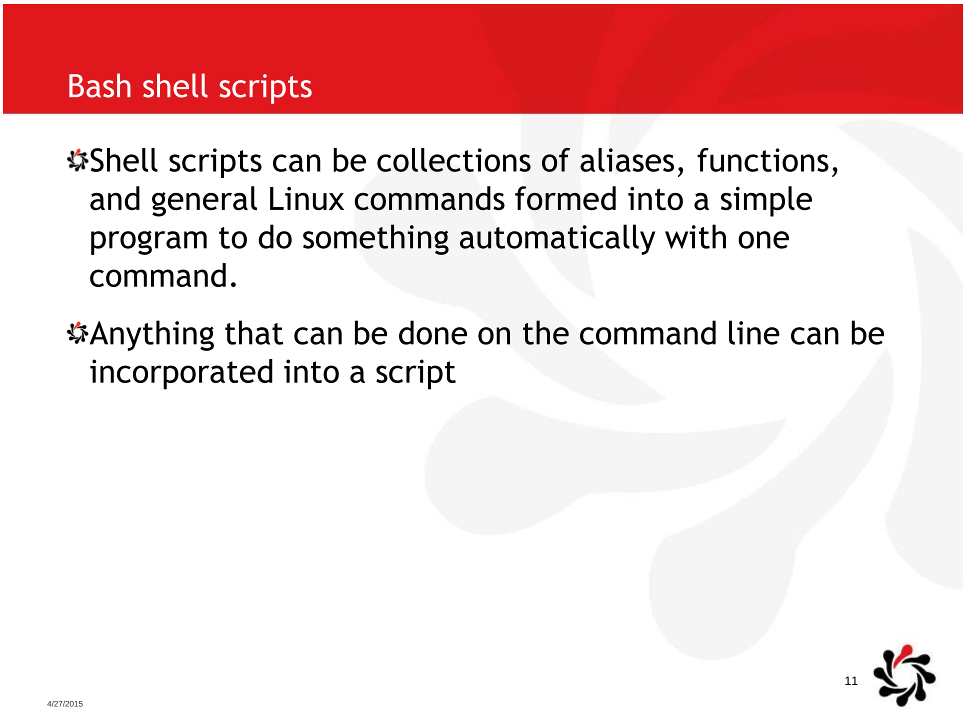
## 105.2 Customize or write simple scripts

**Weight** 4

**Description** Candidates should be able to customize existing scripts, or write simple new BASH scripts.

**Key Knowledge Areas**

- Use standard sh syntax (loops, tests).
- Use command substitution.
- Test return values for success or failure or other information provided by a command.
- Perform conditional mailing to the superuser.
- Correctly select the script interpreter through the shebang (#!) line.
- Manage the location, ownership, execution and suid-rights of scripts.

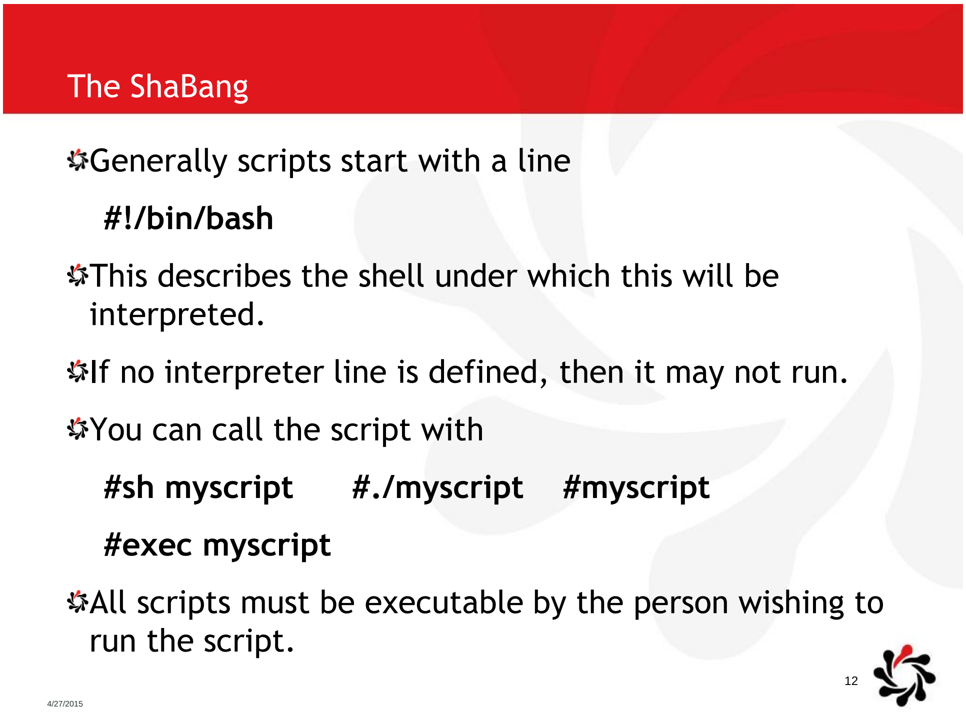


## Bash shell scripts

- Shell scripts can be collections of aliases, functions, and general Linux commands formed into a simple program to do something automatically with one command.
- Anything that can be done on the command line can be incorporated into a script

11

4/27/2015



## The ShaBang

- Generally scripts start with a line

  **#!/bin/bash**

- This describes the shell under which this will be interpreted.
- If no interpreter line is defined, then it may not run.
- You can call the script with

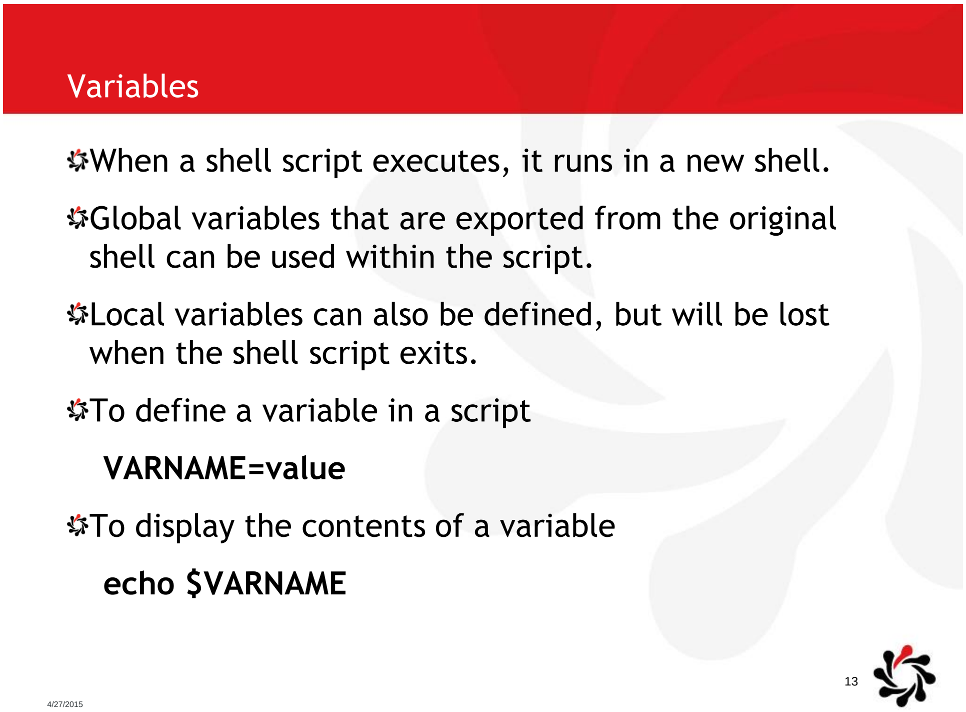
  **#sh myscript**  **#./myscript**  **#myscript**

  **#exec myscript**

- All scripts must be executable by the person wishing to run the script.

12

4/27/2015



## Variables

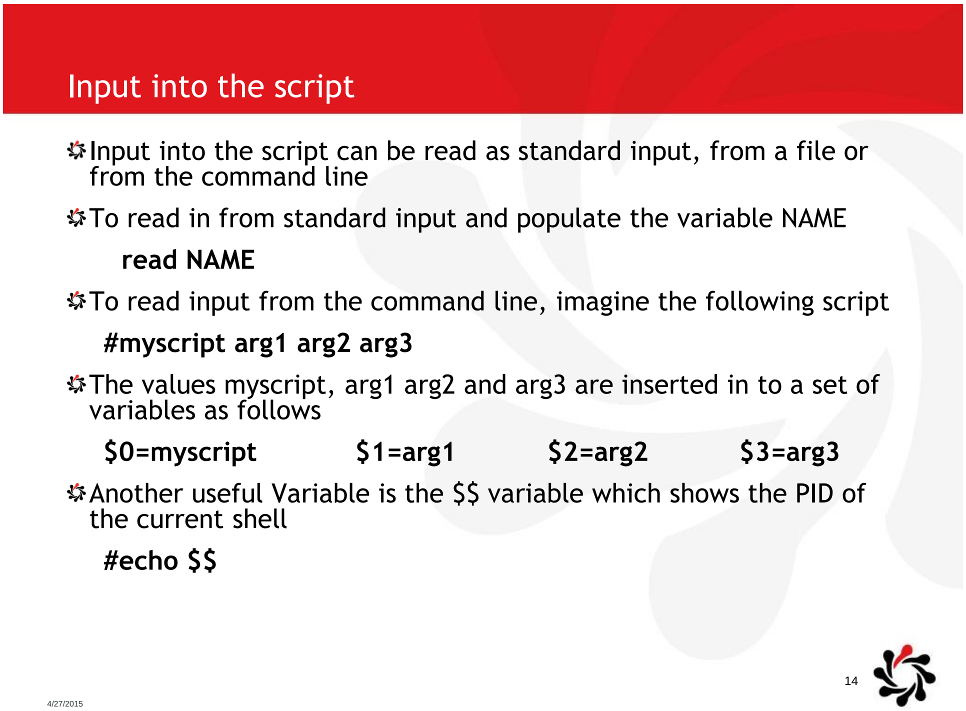
- When a shell script executes, it runs in a new shell.
- Global variables that are exported from the original shell can be used within the script.
- Local variables can also be defined, but will be lost when the shell script exits.
- To define a variable in a script

```
VARNAME=value
```

- To display the contents of a variable

```
echo $VARNAME
```

4/27/2015 13



## Input into the script

- Input into the script can be read as standard input, from a file or from the command line
- To read in from standard input and populate the variable NAME

```
read NAME
```

- To read input from the command line, imagine the following script

```
#myscript arg1 arg2 arg3
```

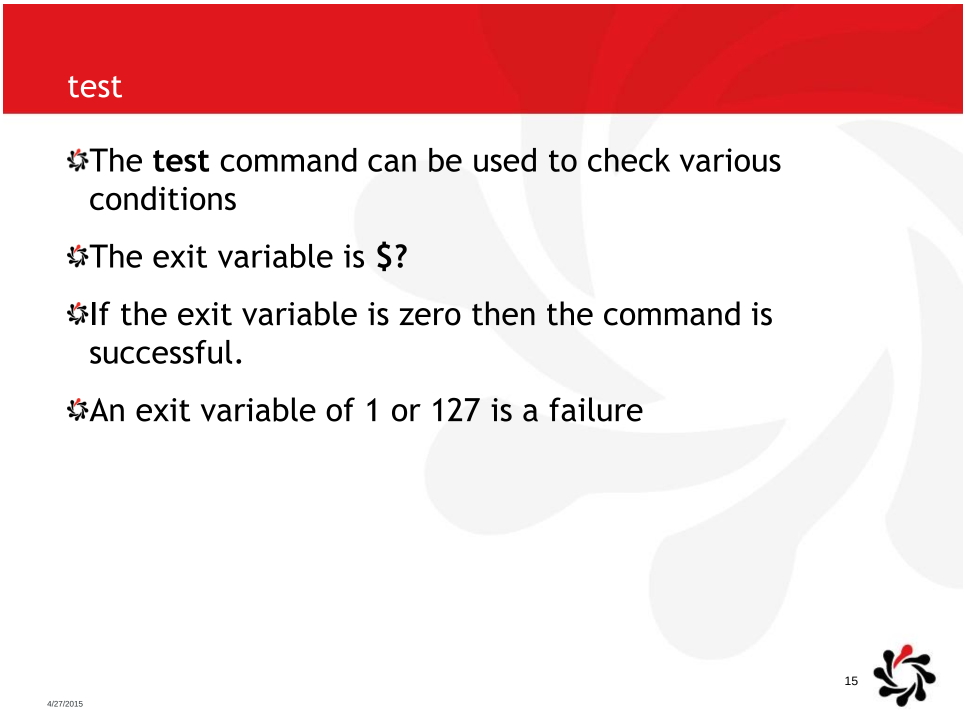
- The values myscript, arg1 arg2 and arg3 are inserted in to a set of variables as follows

```
$0=myscript      $1=arg1      $2=arg2      $3=arg3
```

- Another useful Variable is the $$ variable which shows the PID of the current shell

```
#echo $$
```

4/27/2015 14

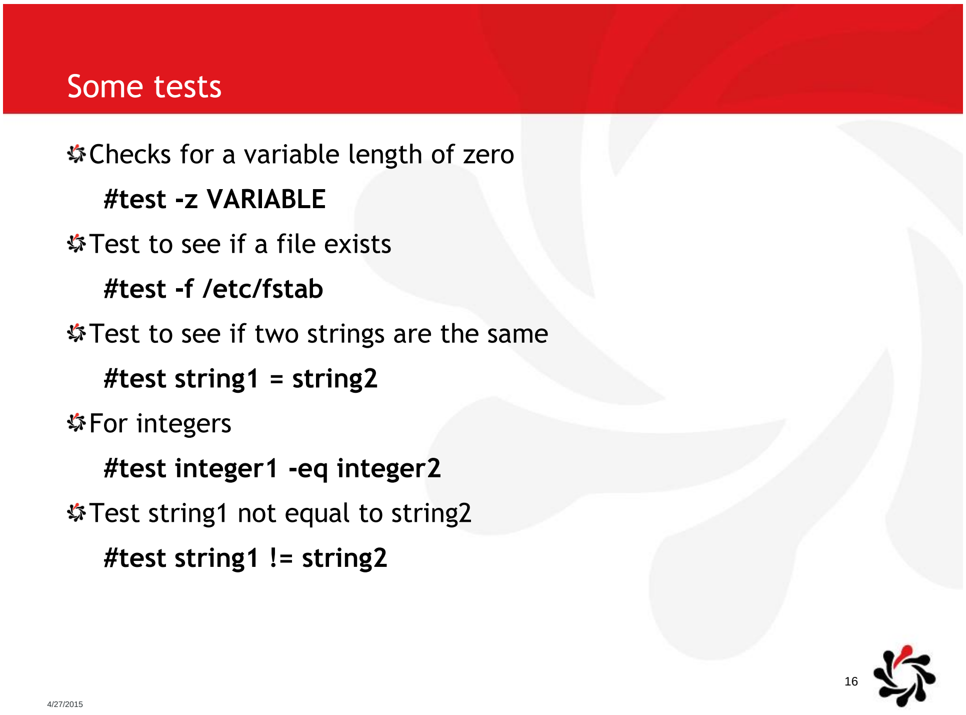


## test

- The **test** command can be used to check various conditions
- The exit variable is **$?**
- If the exit variable is zero then the command is successful.
- An exit variable of 1 or 127 is a failure

4/27/2015

15



## Some tests

- Checks for a variable length of zero

  **#test -z VARIABLE**
- Test to see if a file exists

  **#test -f /etc/fstab**
- Test to see if two strings are the same

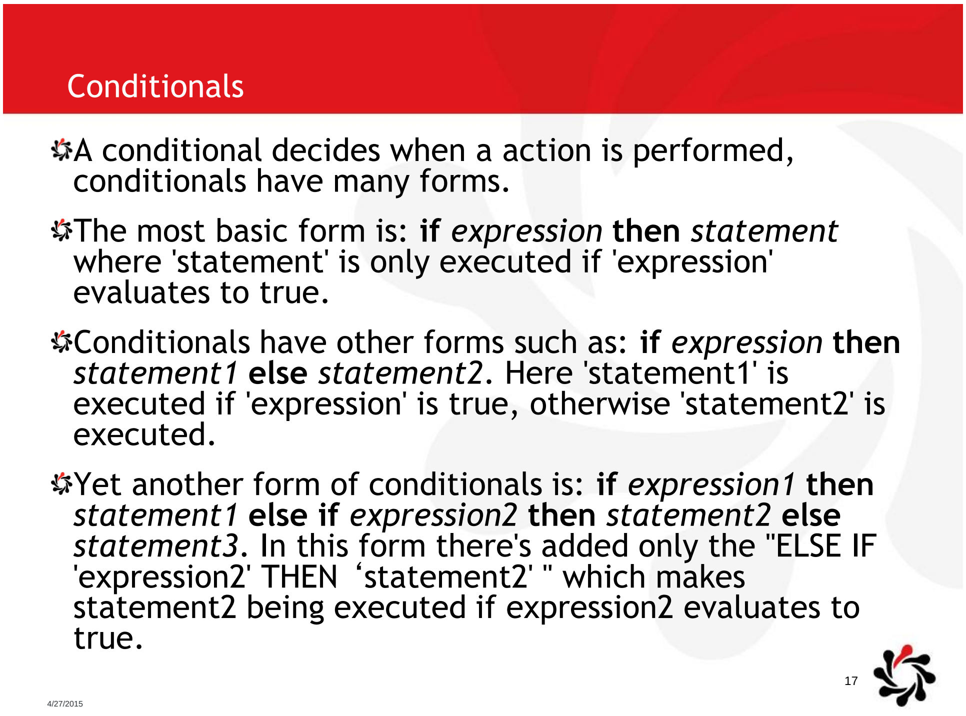
  **#test string1 = string2**
- For integers

  **#test integer1 -eq integer2**
- Test string1 not equal to string2

  **#test string1 != string2**

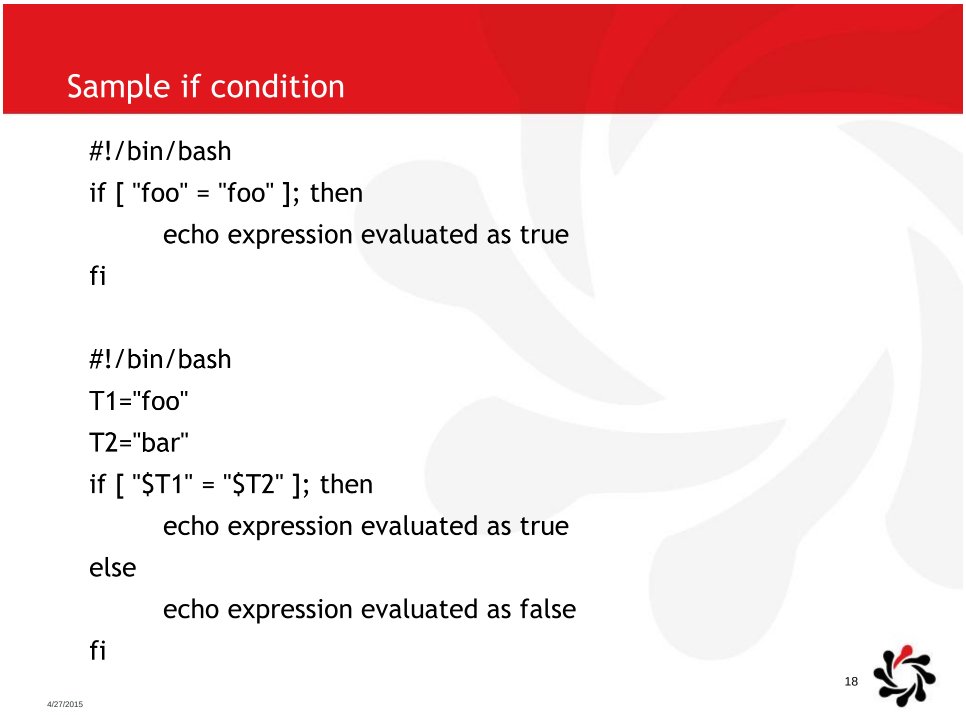
4/27/2015

16



## Conditionals

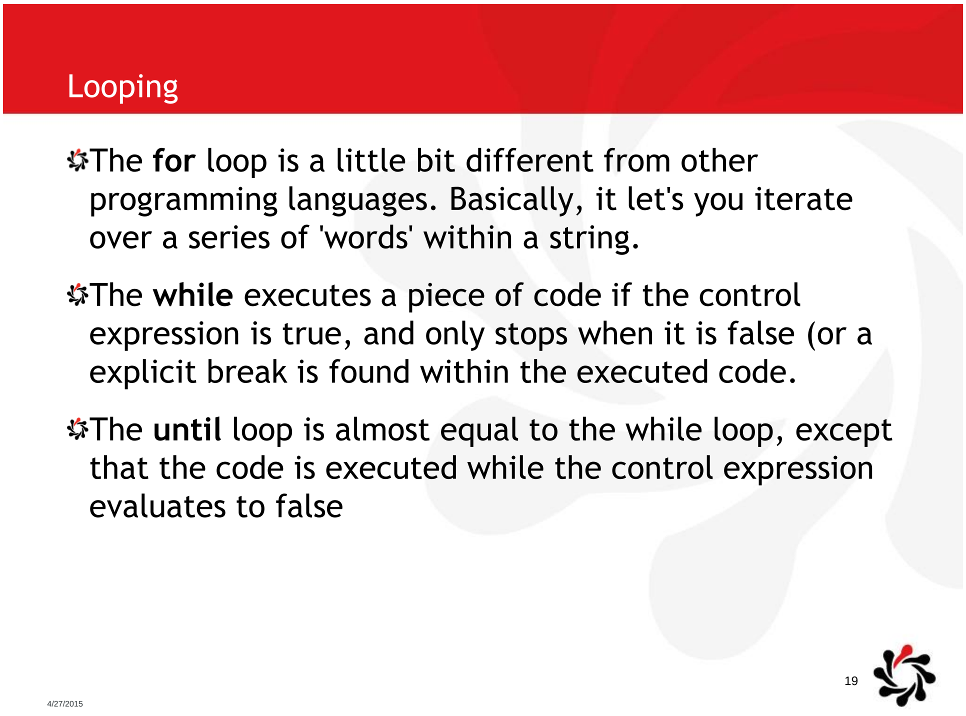
- A conditional decides when a action is performed, conditionals have many forms.
- The most basic form is: **if** *expression* **then** *statement* where 'statement' is only executed if 'expression' evaluates to true.
- Conditionals have other forms such as: **if** *expression* **then** *statement1* **else** *statement2*. Here 'statement1' is executed if 'expression' is true, otherwise 'statement2' is executed.
- Yet another form of conditionals is: **if** *expression1* **then** *statement1* **else if** *expression2* **then** *statement2* **else** *statement3*. In this form there's added only the "ELSE IF 'expression2' THEN ‘statement2' " which makes statement2 being executed if expression2 evaluates to true.



## Sample if condition

```
#!/bin/bash
if [ "foo" = "foo" ]; then
        echo expression evaluated as true
fi

#!/bin/bash
T1="foo"
T2="bar"
if [ "$T1" = "$T2" ]; then
        echo expression evaluated as true
else
        echo expression evaluated as false
fi
```

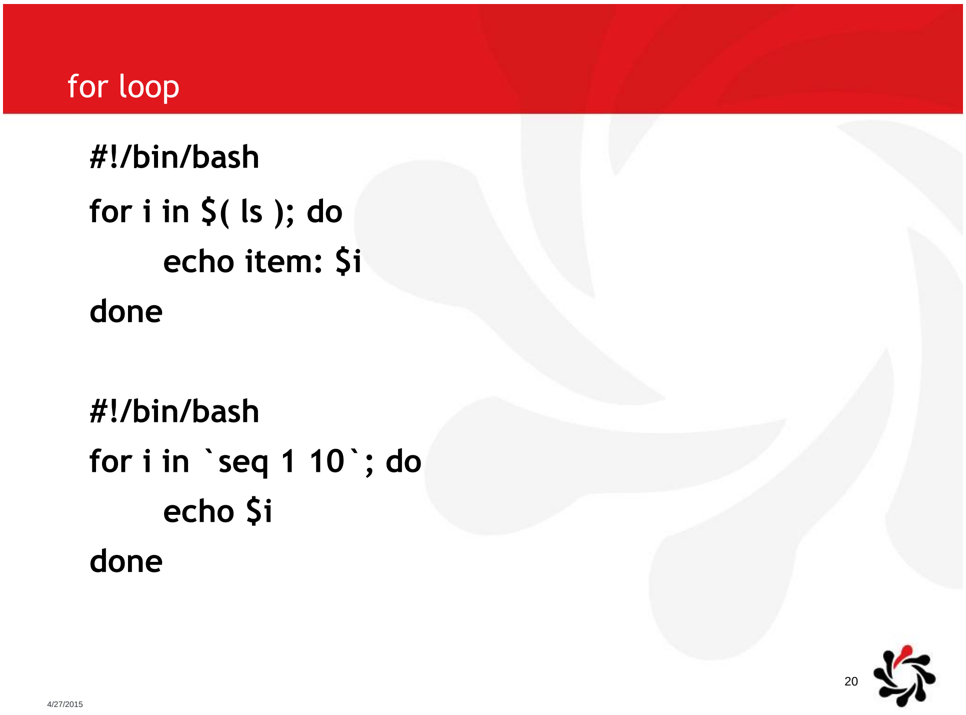


## Looping

- The **for** loop is a little bit different from other programming languages. Basically, it let's you iterate over a series of 'words' within a string.
- The **while** executes a piece of code if the control expression is true, and only stops when it is false (or a explicit break is found within the executed code.
- The **until** loop is almost equal to the while loop, except that the code is executed while the control expression evaluates to false

4/27/2015

19



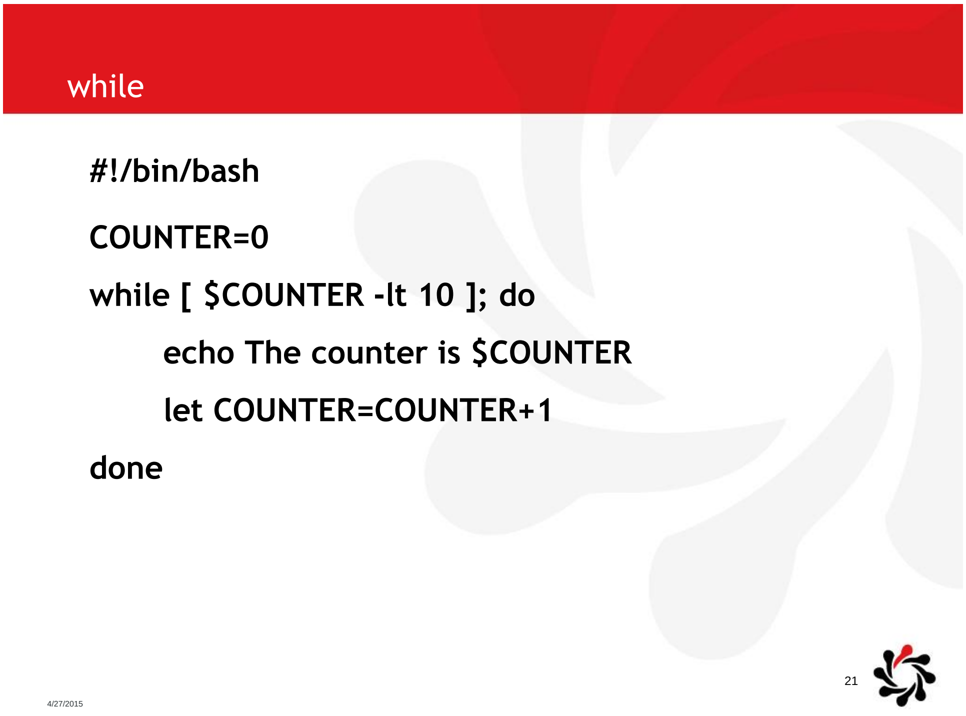
## for loop

```
#!/bin/bash
for i in $( ls ); do
        echo item: $i
done

#!/bin/bash
for i in `seq 1 10`; do
        echo $i
done
```

4/27/2015

20

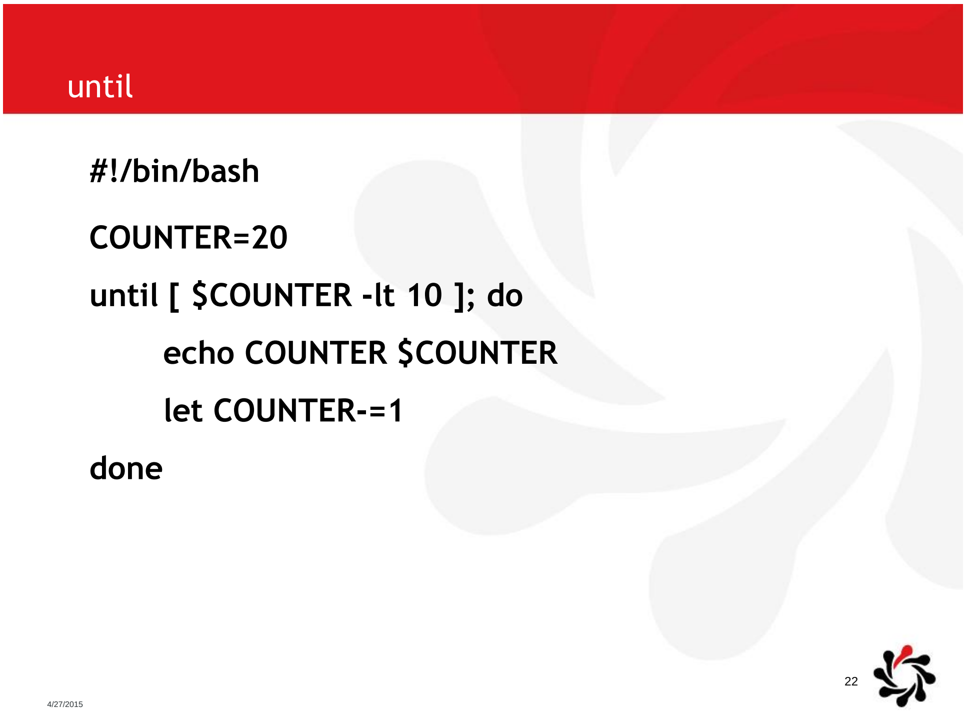


## while

```
#!/bin/bash
COUNTER=0
while [ $COUNTER -lt 10 ]; do
        echo The counter is $COUNTER
        let COUNTER=COUNTER+1
done
```

21

4/27/2015

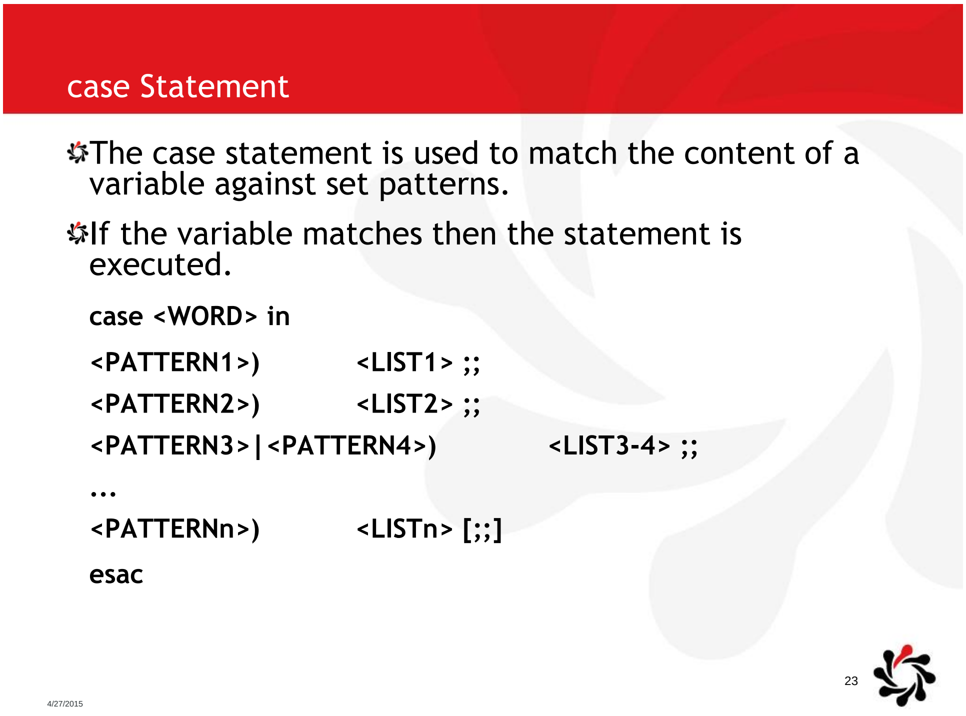


## until

```
#!/bin/bash
COUNTER=20
until [ $COUNTER -lt 10 ]; do
        echo COUNTER $COUNTER
        let COUNTER-=1
done
```

22

4/27/2015



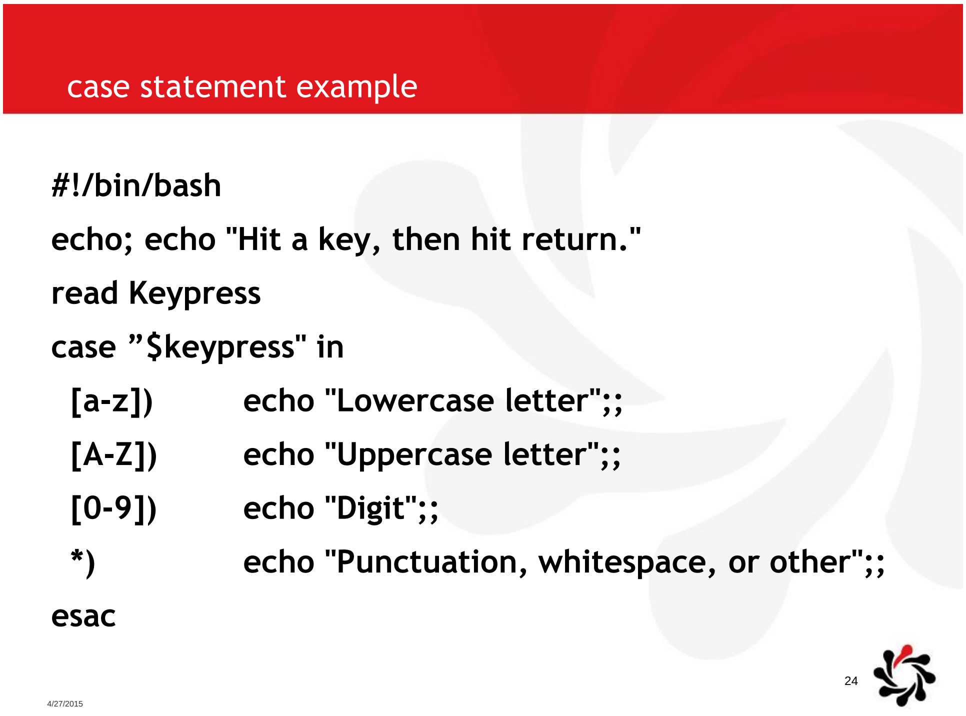
## case Statement

- The case statement is used to match the content of a variable against set patterns.
- If the variable matches then the statement is executed.

```
case <WORD> in
<PATTERN1>)          <LIST1> ;;
<PATTERN2>)          <LIST2> ;;
<PATTERN3>|<PATTERN4>)          <LIST3-4> ;;
...
<PATTERNn>)          <LISTn> [;;]
esac
```

4/27/2015

23



## case statement example

```
#!/bin/bash
echo; echo "Hit a key, then hit return."
read Keypress
case ”$keypress" in
  [a-z])       echo "Lowercase letter";;
  [A-Z])       echo "Uppercase letter";;
  [0-9])       echo "Digit";;
  *)           echo "Punctuation, whitespace, or other";;
esac
```

4/27/2015

24

## Sequence

- The seq command can be used to create a sequence of numbers
- To produce a sequence starting at 1 through to a number

**#seq 10**

To start at a specific number

**#seq 5 10**

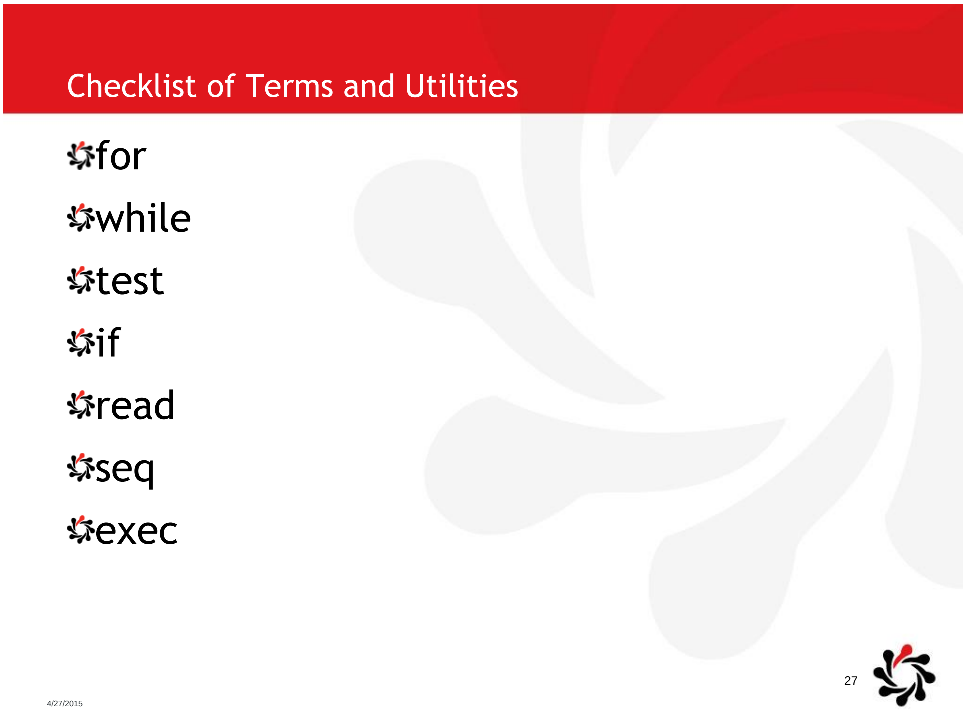
## Recap 105.2 Customize or write simple scripts

**Weight** 4

**Description** Candidates should be able to customize existing scripts, or write simple new BASH scripts.

**Key Knowledge Areas**

- Use standard sh syntax (loops, tests).
- Use command substitution.
- Test return values for success or failure or other information provided by a command.
- Perform conditional mailing to the superuser.
- Correctly select the script interpreter through the shebang (#!) line.
- Manage the location, ownership, execution and suid-rights of scripts.

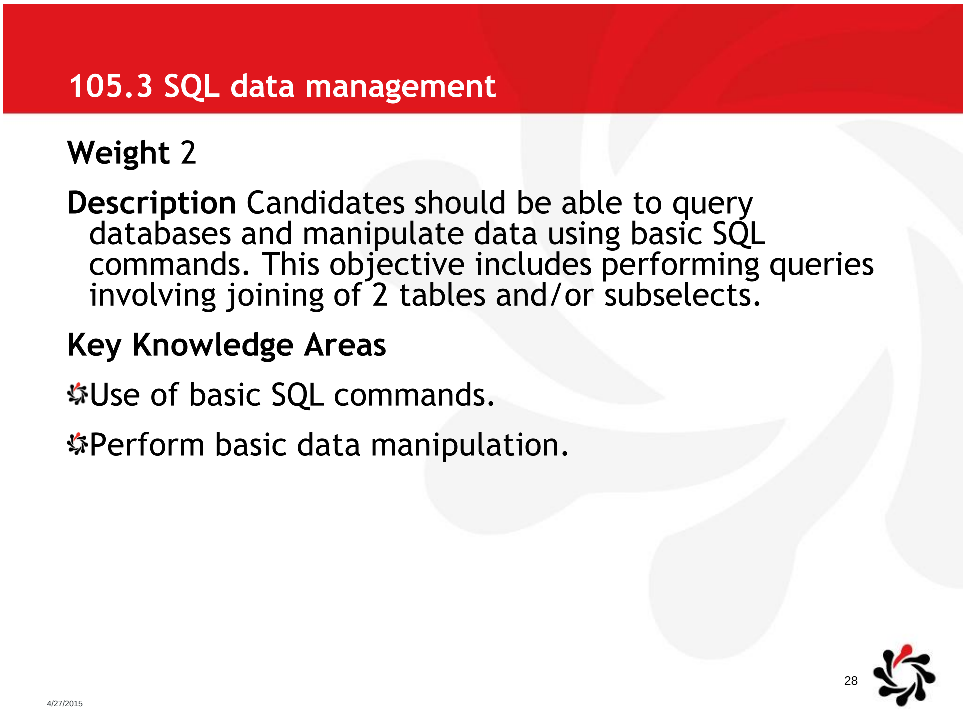


## Checklist of Terms and Utilities

- for
- while
- test
- if
- read
- seq
- exec

27

4/27/2015



## 105.3 SQL data management

**Weight** 2

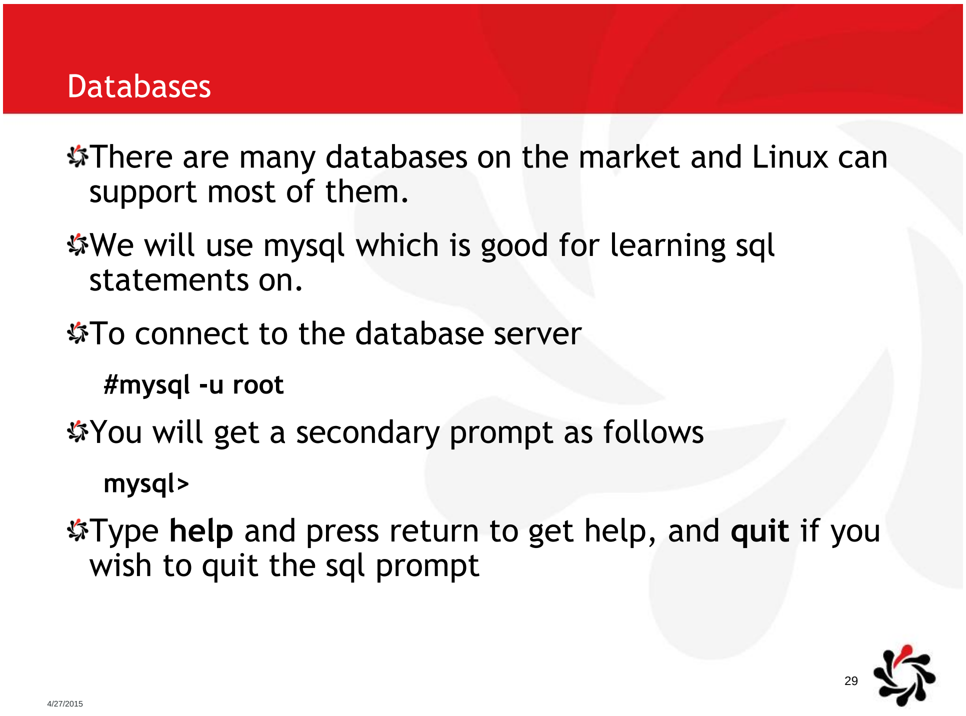
**Description** Candidates should be able to query databases and manipulate data using basic SQL commands. This objective includes performing queries involving joining of 2 tables and/or subselects.

**Key Knowledge Areas**

- Use of basic SQL commands.
- Perform basic data manipulation.

28

4/27/2015



## Databases

- There are many databases on the market and Linux can support most of them.
- We will use mysql which is good for learning sql statements on.
- To connect to the database server

```
#mysql -u root
```

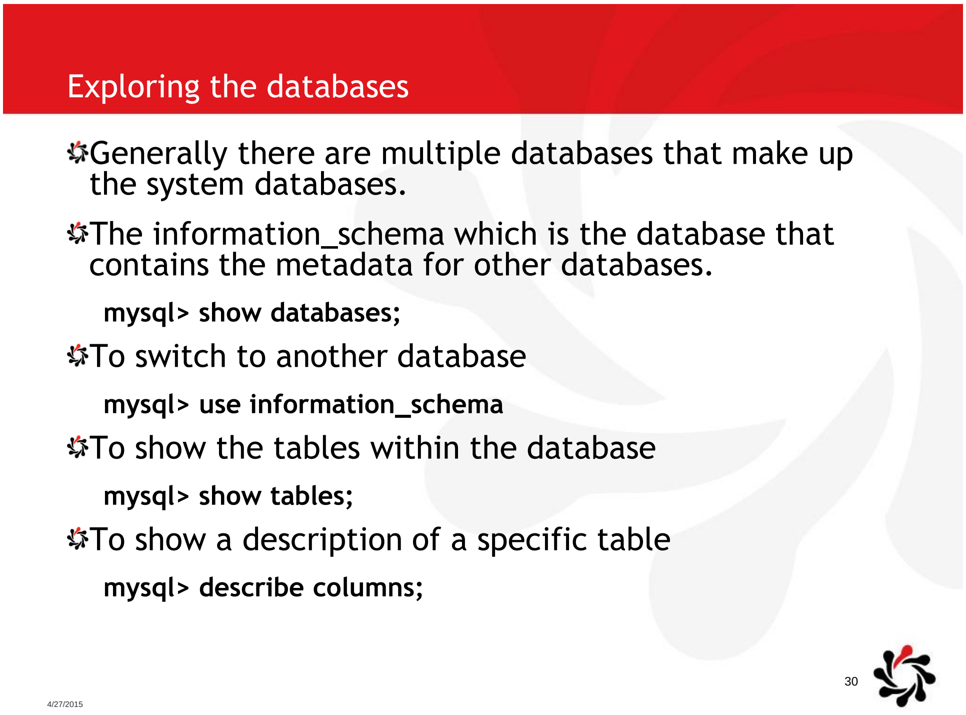
- You will get a secondary prompt as follows

```
mysql>
```

- Type **help** and press return to get help, and **quit** if you wish to quit the sql prompt

4/27/2015

29



## Exploring the databases

- Generally there are multiple databases that make up the system databases.
- The information_schema which is the database that contains the metadata for other databases.

```
mysql> show databases;
```

- To switch to another database

```
mysql> use information_schema
```

- To show the tables within the database

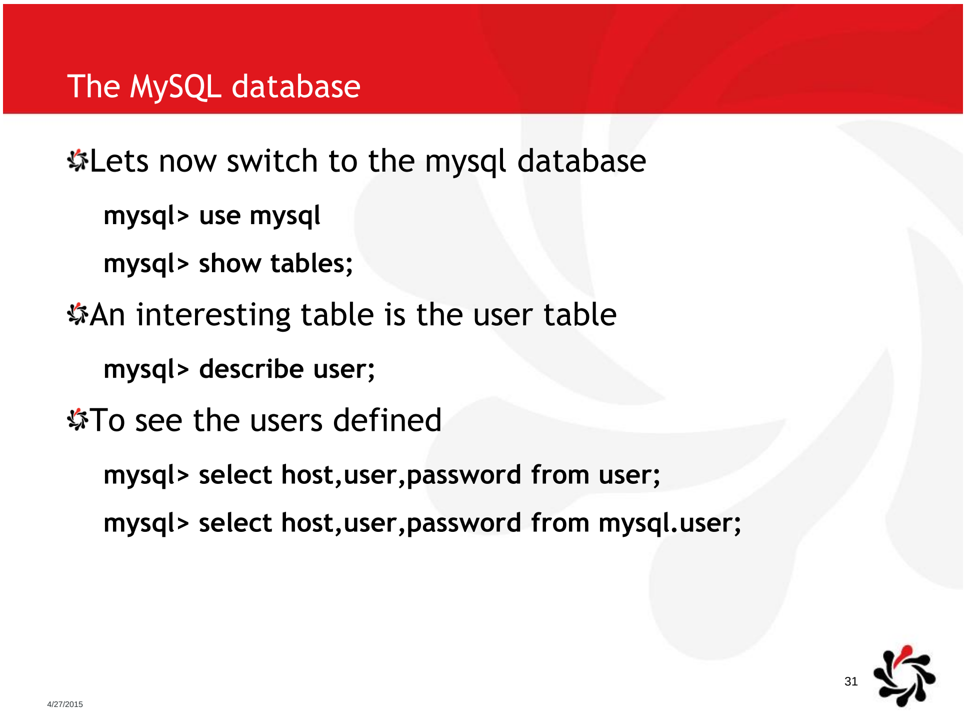
```
mysql> show tables;
```

- To show a description of a specific table

```
mysql> describe columns;
```

4/27/2015

30



## The MySQL database

- Lets now switch to the mysql database

```
mysql> use mysql
mysql> show tables;
```

- An interesting table is the user table

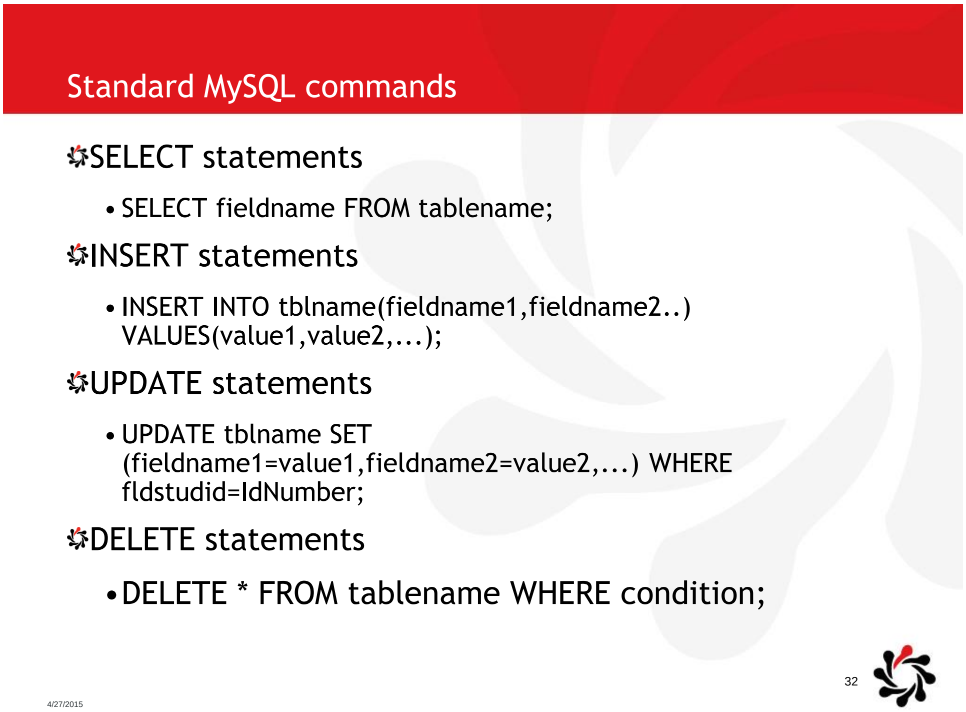
```
mysql> describe user;
```

- To see the users defined

```
mysql> select host,user,password from user;
mysql> select host,user,password from mysql.user;
```

4/27/2015

31

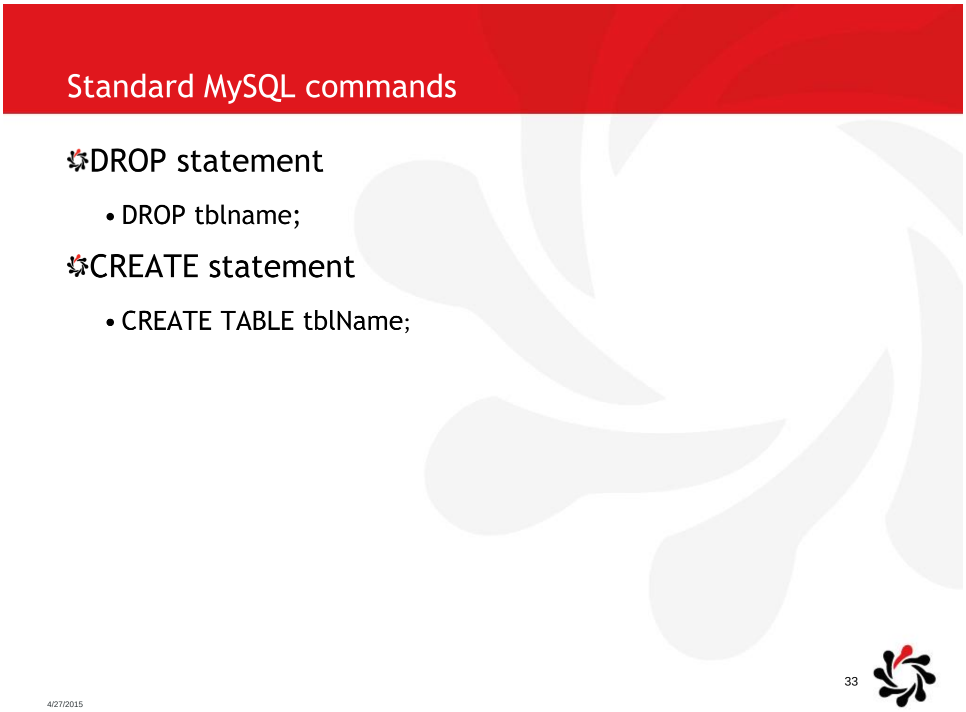


## Standard MySQL commands

- SELECT statements
  - SELECT fieldname FROM tablename;
- INSERT statements
  - INSERT INTO tblname(fieldname1,fieldname2..) VALUES(value1,value2,...);
- UPDATE statements
  - UPDATE tblname SET (fieldname1=value1,fieldname2=value2,...) WHERE fldstudid=IdNumber;
- DELETE statements
  - DELETE * FROM tablename WHERE condition;

4/27/2015

32

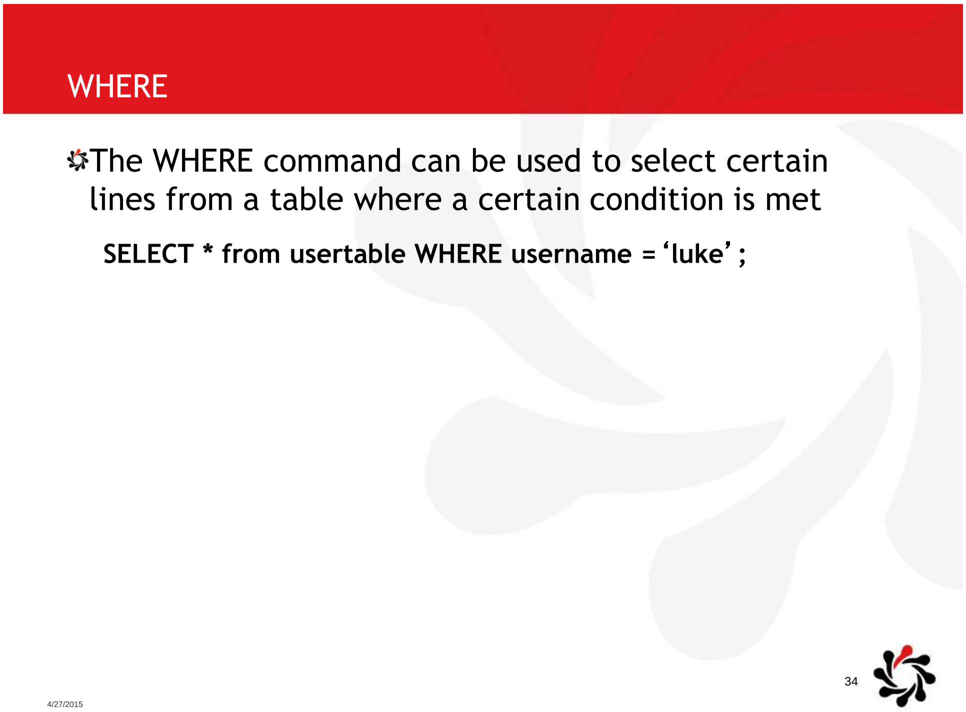


## Standard MySQL commands

- DROP statement
  - DROP tblname;
- CREATE statement
  - CREATE TABLE tblName;

4/27/2015

33



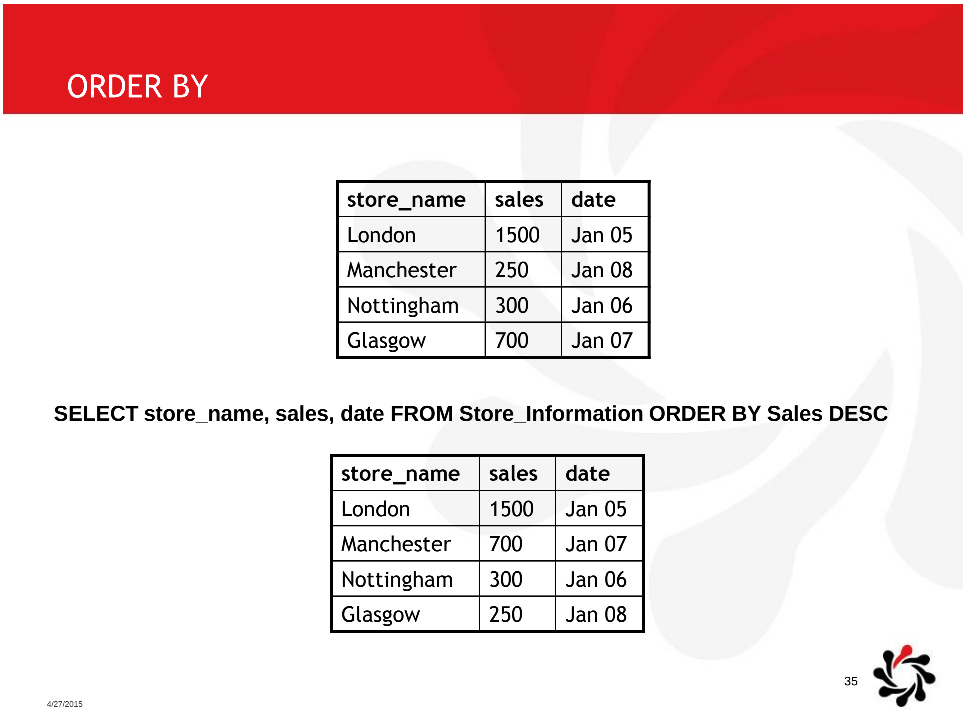
## WHERE

- The WHERE command can be used to select certain lines from a table where a certain condition is met

  **SELECT * from usertable WHERE username = ‘luke’ ;**

4/27/2015

34



## ORDER BY

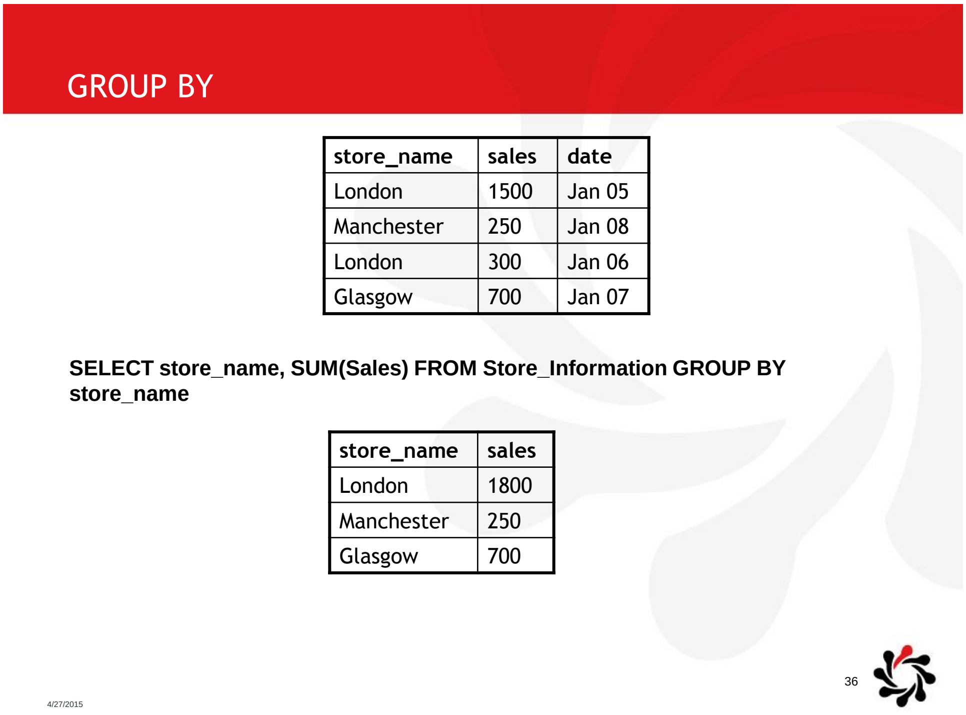
| store_name | sales | date |
|---|---|---|
| London | 1500 | Jan 05 |
| Manchester | 250 | Jan 08 |
| Nottingham | 300 | Jan 06 |
| Glasgow | 700 | Jan 07 |

**SELECT store_name, sales, date FROM Store_Information ORDER BY Sales DESC**

| store_name | sales | date |
|---|---|---|
| London | 1500 | Jan 05 |
| Manchester | 700 | Jan 07 |
| Nottingham | 300 | Jan 06 |
| Glasgow | 250 | Jan 08 |

4/27/2015

35



## GROUP BY

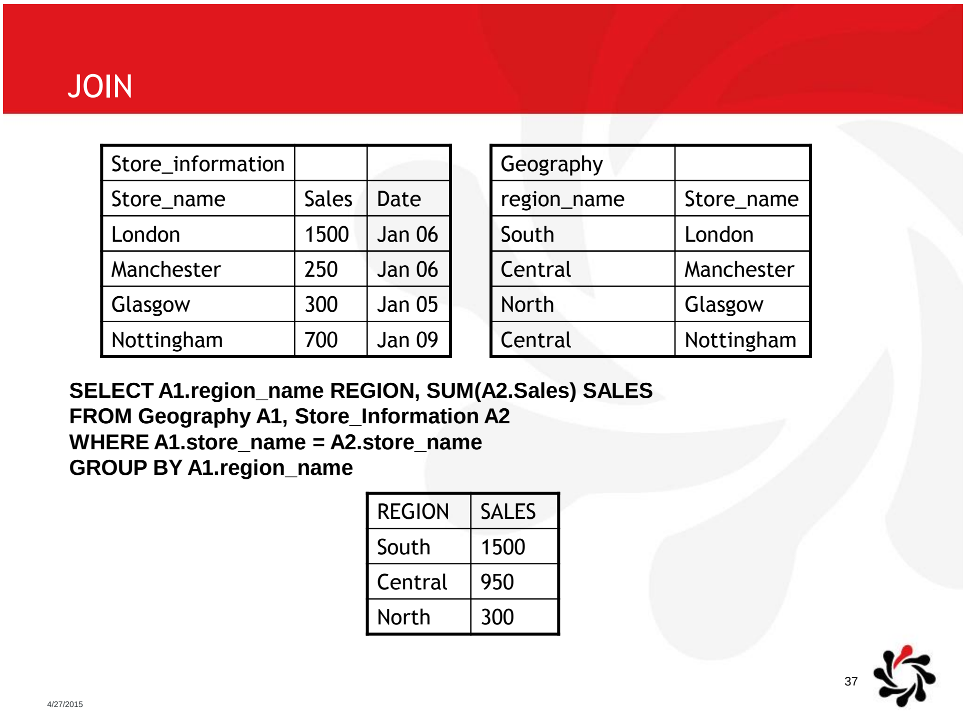
| store_name | sales | date |
|---|---|---|
| London | 1500 | Jan 05 |
| Manchester | 250 | Jan 08 |
| London | 300 | Jan 06 |
| Glasgow | 700 | Jan 07 |

**SELECT store_name, SUM(Sales) FROM Store_Information GROUP BY store_name**

| store_name | sales |
|---|---|
| London | 1800 |
| Manchester | 250 |
| Glasgow | 700 |

4/27/2015

36



## JOIN

| Store_information | | |
|---|---|---|
| Store_name | Sales | Date |
| London | 1500 | Jan 06 |
| Manchester | 250 | Jan 06 |
| Glasgow | 300 | Jan 05 |
| Nottingham | 700 | Jan 09 |

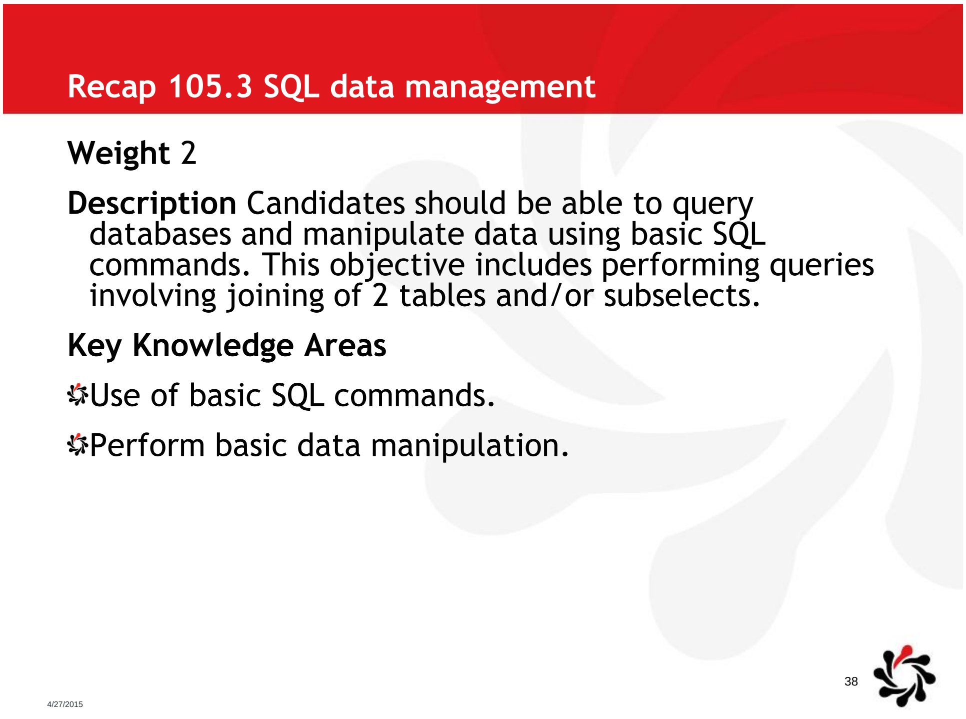
| Geography | |
|---|---|
| region_name | Store_name |
| South | London |
| Central | Manchester |
| North | Glasgow |
| Central | Nottingham |

```
SELECT A1.region_name REGION, SUM(A2.Sales) SALES
FROM Geography A1, Store_Information A2
WHERE A1.store_name = A2.store_name
GROUP BY A1.region_name
```

| REGION | SALES |
|---|---|
| South | 1500 |
| Central | 950 |
| North | 300 |

37

4/27/2015



## Recap 105.3 SQL data management

**Weight** 2

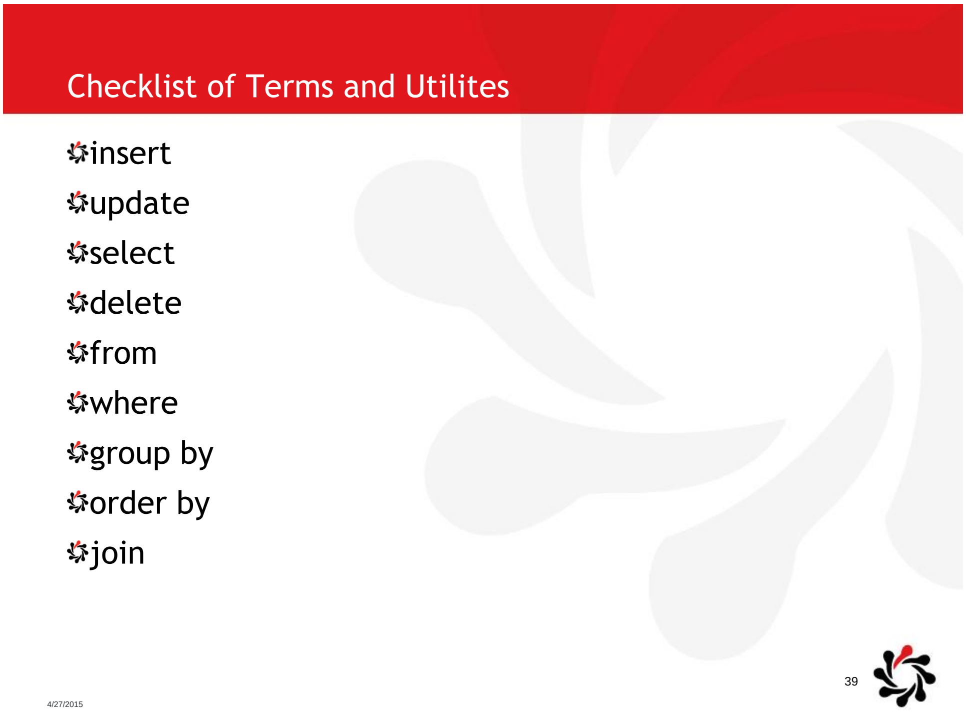
**Description** Candidates should be able to query databases and manipulate data using basic SQL commands. This objective includes performing queries involving joining of 2 tables and/or subselects.

**Key Knowledge Areas**

- Use of basic SQL commands.
- Perform basic data manipulation.

38

4/27/2015



## Checklist of Terms and Utilites

- insert
- update
- select
- delete
- from
- where
- group by
- order by
- join

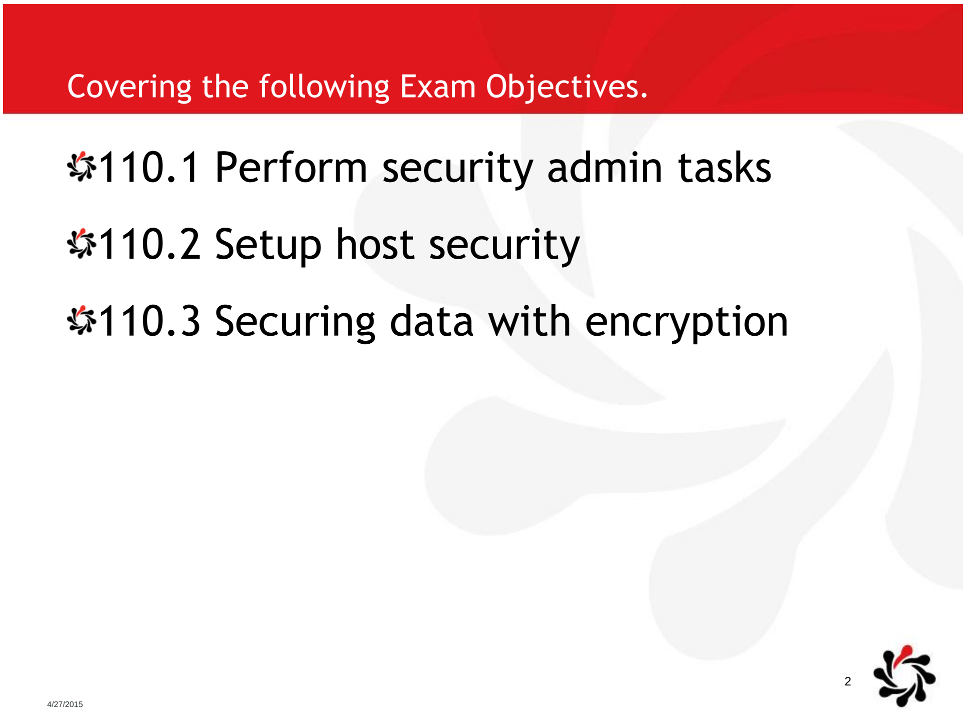
39

4/27/2015

FIREBRAND

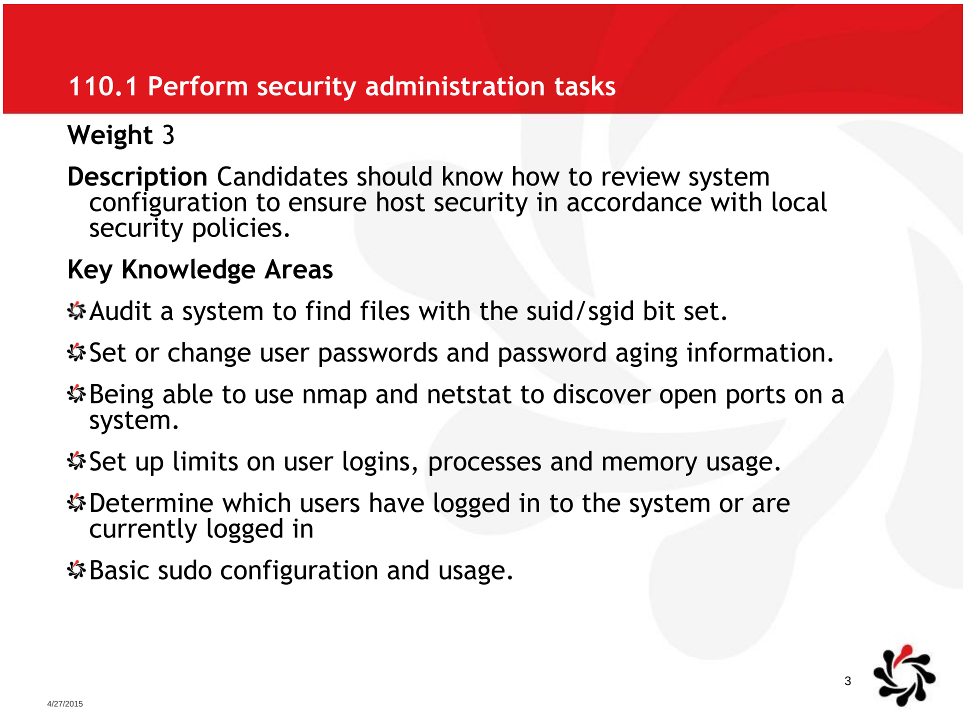
# Linux LPIC 1

## Chapter 10 Security



## Covering the following Exam Objectives.

- 110.1 Perform security admin tasks
- 110.2 Setup host security
- 110.3 Securing data with encryption



## 110.1 Perform security administration tasks

**Weight** 3

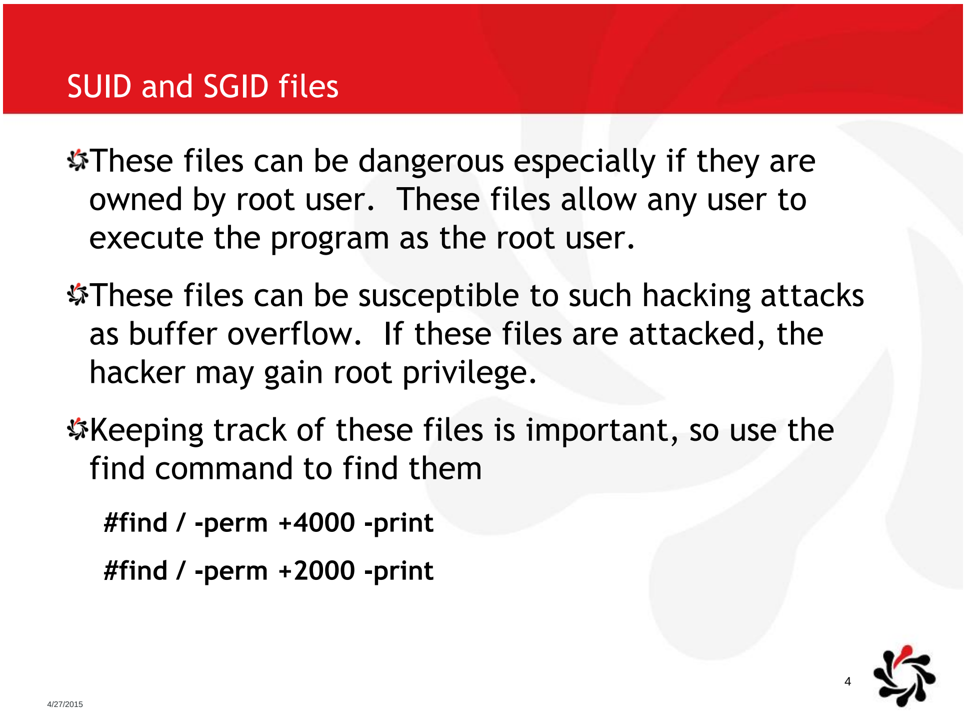
**Description** Candidates should know how to review system configuration to ensure host security in accordance with local security policies.

**Key Knowledge Areas**

- Audit a system to find files with the suid/sgid bit set.
- Set or change user passwords and password aging information.
- Being able to use nmap and netstat to discover open ports on a system.
- Set up limits on user logins, processes and memory usage.
- Determine which users have logged in to the system or are currently logged in
- Basic sudo configuration and usage.

4/27/2015

3



## SUID and SGID files

- These files can be dangerous especially if they are owned by root user. These files allow any user to execute the program as the root user.
- These files can be susceptible to such hacking attacks as buffer overflow. If these files are attacked, the hacker may gain root privilege.
- Keeping track of these files is important, so use the find command to find them

```
#find / -perm +4000 -print
#find / -perm +2000 -print
```

4/27/2015

4

## Hidden Files

- Users can hide files from normal ls listings, you can use the find command to find these specific files

```
#find / -name “.*”
```

5

4/27/2015

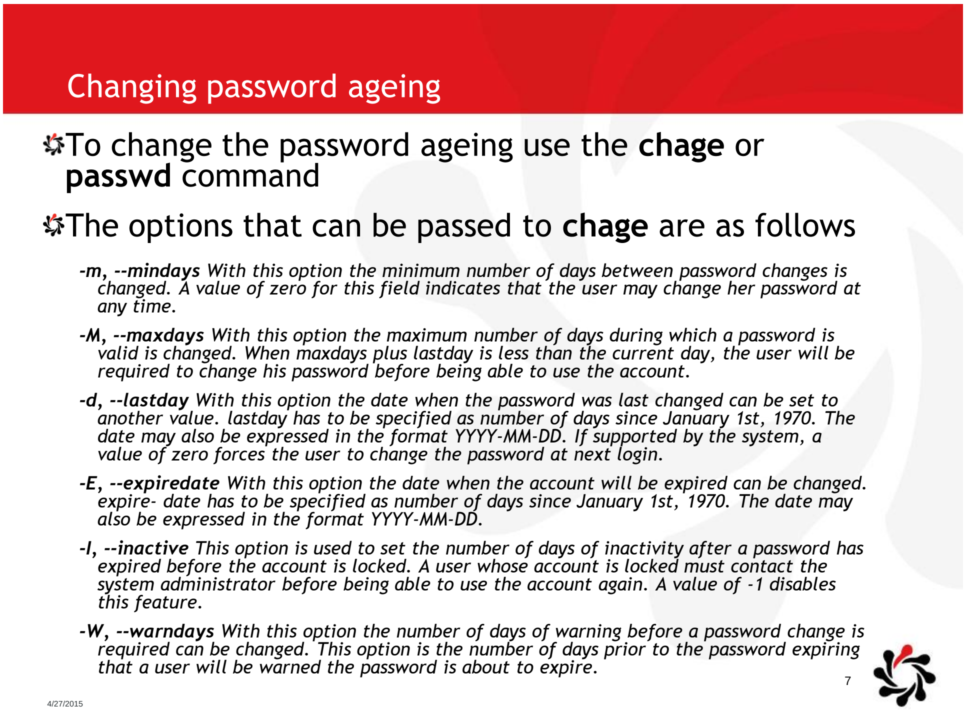
## Password Ageing

- Password ageing allows you to force changes of users passwords.
- The password ageing information is stored in **/etc/shadow** file.
- To defaults for the password ageing are stored in **/etc/login.defs**
- To view a users password ageing

```
#chage -l tom
```

6

4/27/2015



## Changing password ageing

- To change the password ageing use the **chage** or **passwd** command
- The options that can be passed to **chage** are as follows

  ***-m, --mindays*** *With this option the minimum number of days between password changes is changed. A value of zero for this field indicates that the user may change her password at any time.*

  ***-M, --maxdays*** *With this option the maximum number of days during which a password is valid is changed. When maxdays plus lastday is less than the current day, the user will be required to change his password before being able to use the account.*

  ***-d, --lastday*** *With this option the date when the password was last changed can be set to another value. lastday has to be specified as number of days since January 1st, 1970. The date may also be expressed in the format YYYY-MM-DD. If supported by the system, a value of zero forces the user to change the password at next login.*

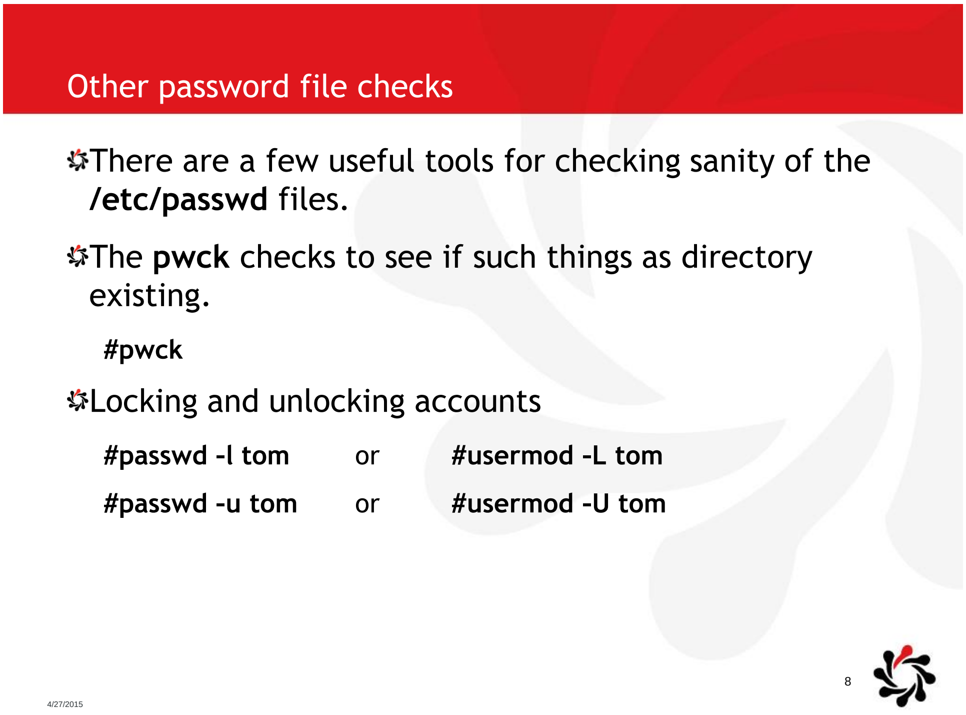
  ***-E, --expiredate*** *With this option the date when the account will be expired can be changed. expire- date has to be specified as number of days since January 1st, 1970. The date may also be expressed in the format YYYY-MM-DD.*

  ***-I, --inactive*** *This option is used to set the number of days of inactivity after a password has expired before the account is locked. A user whose account is locked must contact the system administrator before being able to use the account again. A value of -1 disables this feature.*

  ***-W, --warndays*** *With this option the number of days of warning before a password change is required can be changed. This option is the number of days prior to the password expiring that a user will be warned the password is about to expire.*

4/27/2015

7



## Other password file checks

- There are a few useful tools for checking sanity of the **/etc/passwd** files.
- The **pwck** checks to see if such things as directory existing.

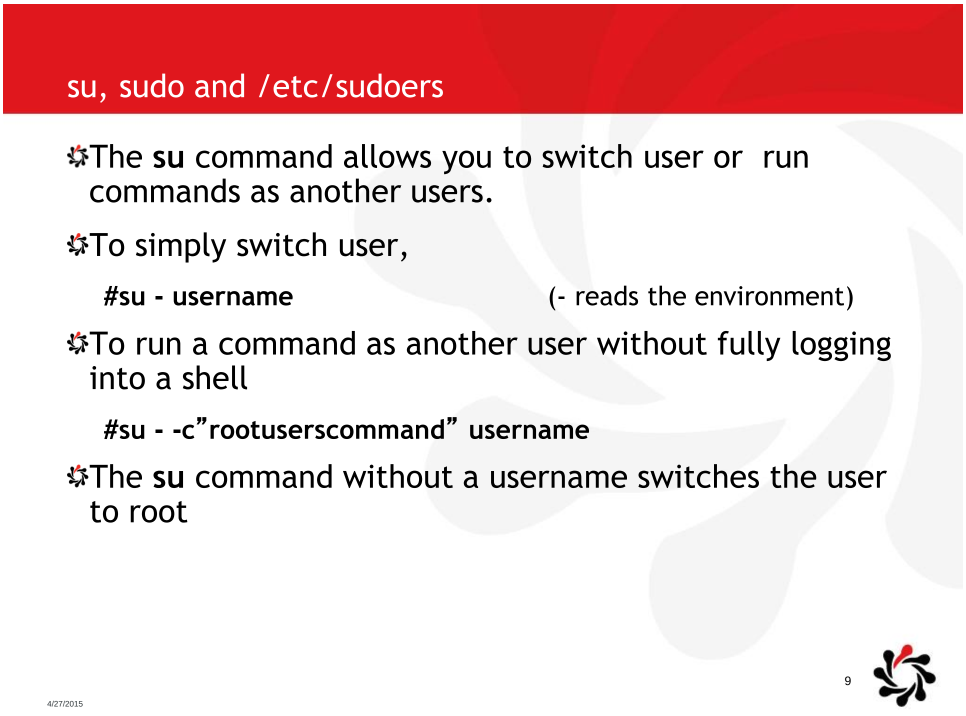
```
#pwck
```

- Locking and unlocking accounts

```
#passwd -l tom     or     #usermod -L tom
#passwd -u tom     or     #usermod -U tom
```

4/27/2015

8



## su, sudo and /etc/sudoers

- The **su** command allows you to switch user or  run commands as another users.
- To simply switch user,

```
#su - username                    (- reads the environment)
```

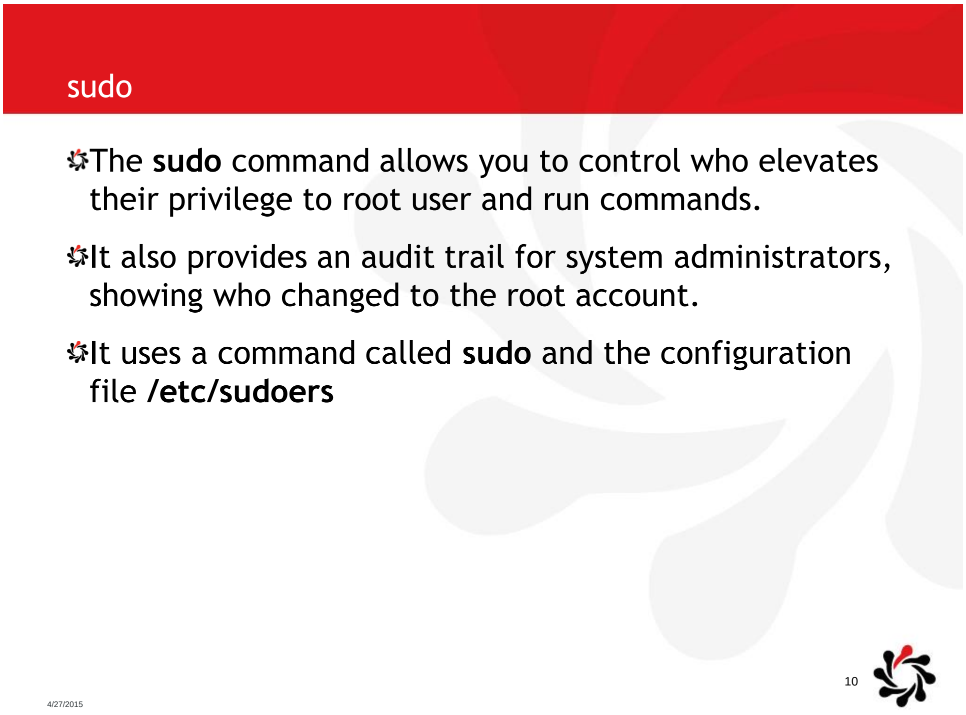
- To run a command as another user without fully logging into a shell

```
#su - -c”rootuserscommand” username
```

- The **su** command without a username switches the user to root

9

4/27/2015

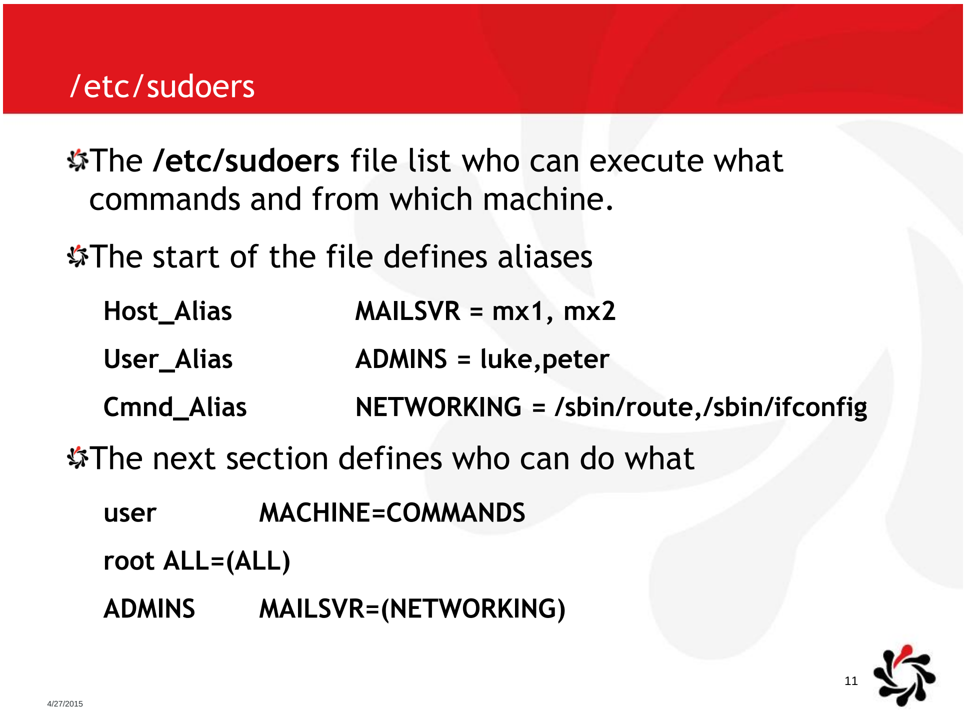


## sudo

- The **sudo** command allows you to control who elevates their privilege to root user and run commands.
- It also provides an audit trail for system administrators, showing who changed to the root account.
- It uses a command called **sudo** and the configuration file **/etc/sudoers**

10

4/27/2015



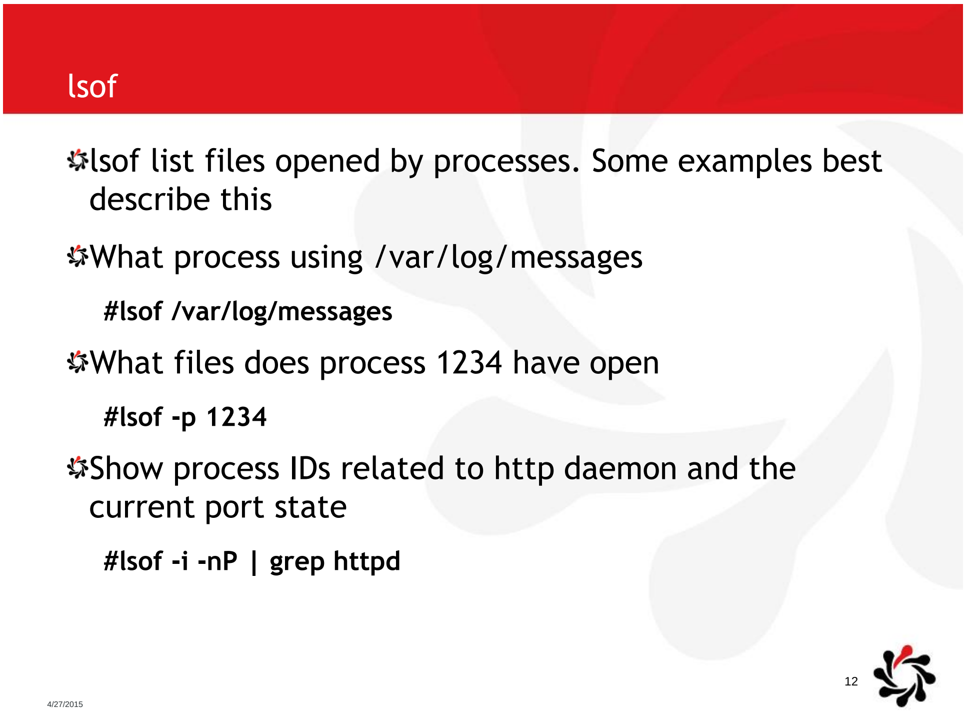
## /etc/sudoers

- The **/etc/sudoers** file list who can execute what commands and from which machine.
- The start of the file defines aliases

```
Host_Alias        MAILSVR = mx1, mx2
User_Alias        ADMINS = luke,peter
Cmnd_Alias        NETWORKING = /sbin/route,/sbin/ifconfig
```

- The next section defines who can do what

```
user        MACHINE=COMMANDS
root ALL=(ALL)
ADMINS      MAILSVR=(NETWORKING)
```



## lsof

- lsof list files opened by processes. Some examples best describe this
- What process using /var/log/messages

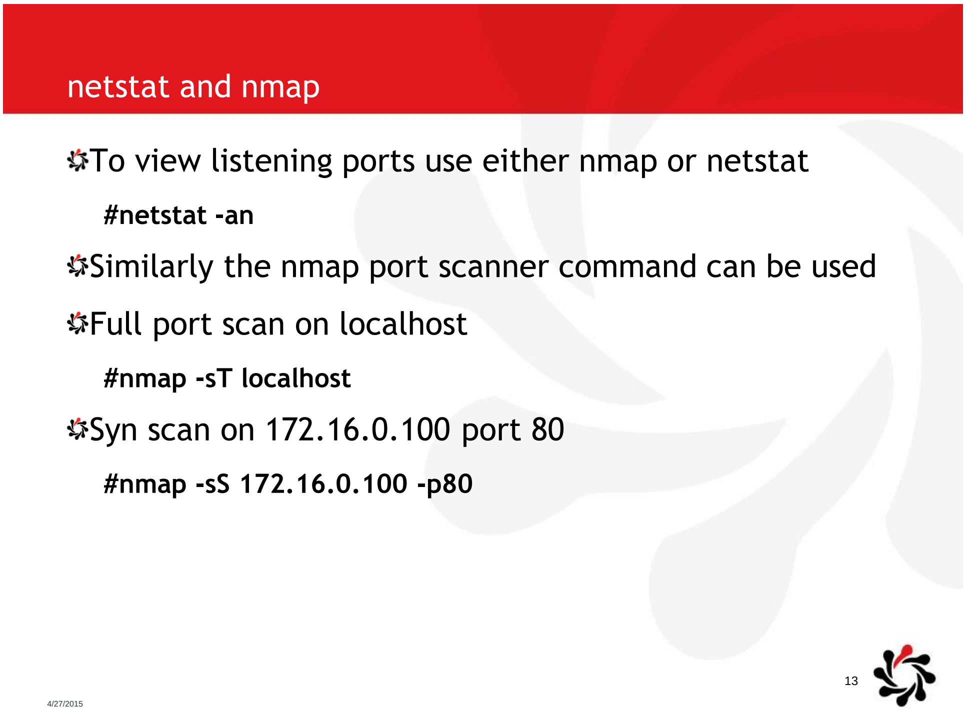
```
#lsof /var/log/messages
```

- What files does process 1234 have open

```
#lsof -p 1234
```

- Show process IDs related to http daemon and the current port state

```
#lsof -i -nP | grep httpd
```



## netstat and nmap

- To view listening ports use either nmap or netstat

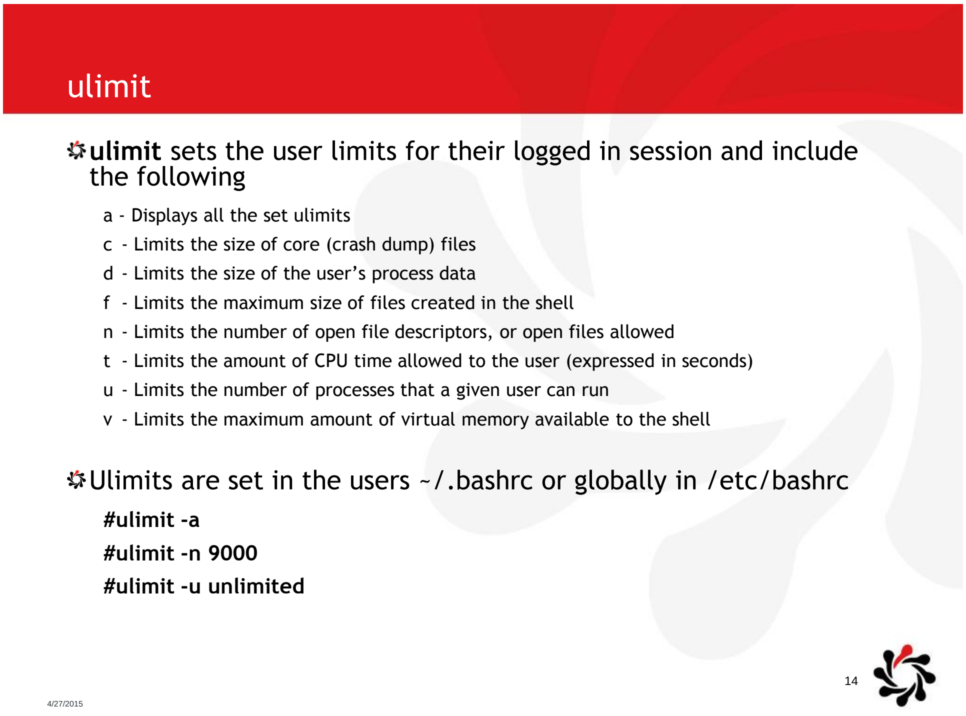
```
#netstat -an
```

- Similarly the nmap port scanner command can be used
- Full port scan on localhost

```
#nmap -sT localhost
```

- Syn scan on 172.16.0.100 port 80

```
#nmap -sS 172.16.0.100 -p80
```



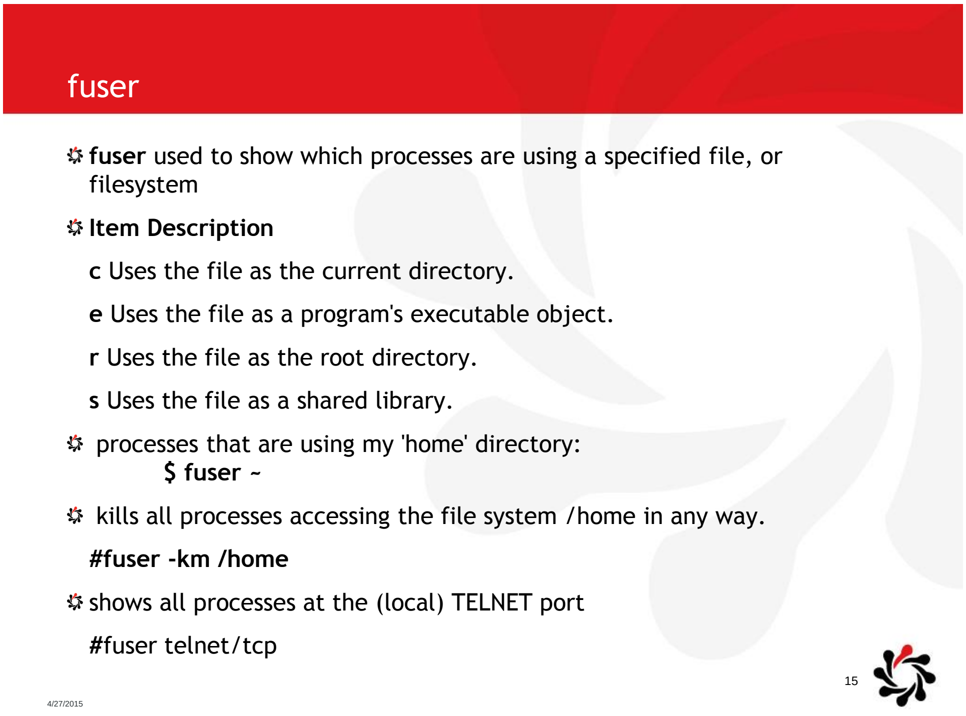
## ulimit

- **ulimit** sets the user limits for their logged in session and include the following

  a - Displays all the set ulimits
  c - Limits the size of core (crash dump) files
  d - Limits the size of the user's process data
  f - Limits the maximum size of files created in the shell
  n - Limits the number of open file descriptors, or open files allowed
  t - Limits the amount of CPU time allowed to the user (expressed in seconds)
  u - Limits the number of processes that a given user can run
  v - Limits the maximum amount of virtual memory available to the shell

- Ulimits are set in the users ~/.bashrc or globally in /etc/bashrc

```
#ulimit -a
#ulimit -n 9000
#ulimit -u unlimited
```



## fuser

- **fuser** used to show which processes are using a specified file, or filesystem
- **Item Description**

  **c** Uses the file as the current directory.

  **e** Uses the file as a program's executable object.

  **r** Uses the file as the root directory.

  **s** Uses the file as a shared library.
- processes that are using my 'home' directory:

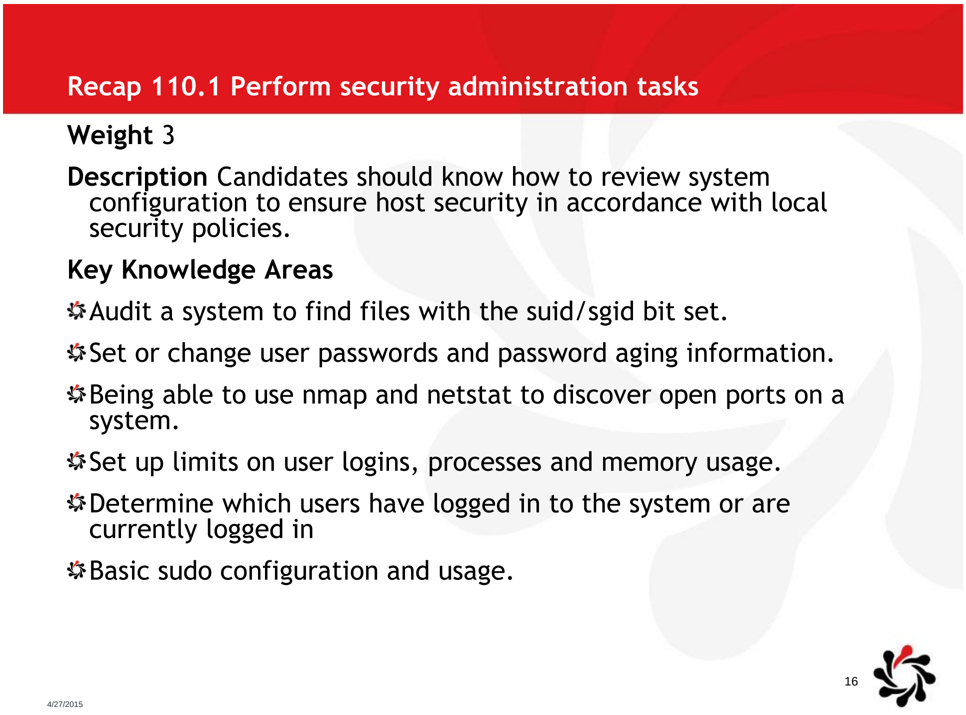
  **$ fuser ~**
- kills all processes accessing the file system /home in any way.

  **#fuser -km /home**
- shows all processes at the (local) TELNET port

  #fuser telnet/tcp

15

4/27/2015



## Recap 110.1 Perform security administration tasks

**Weight** 3

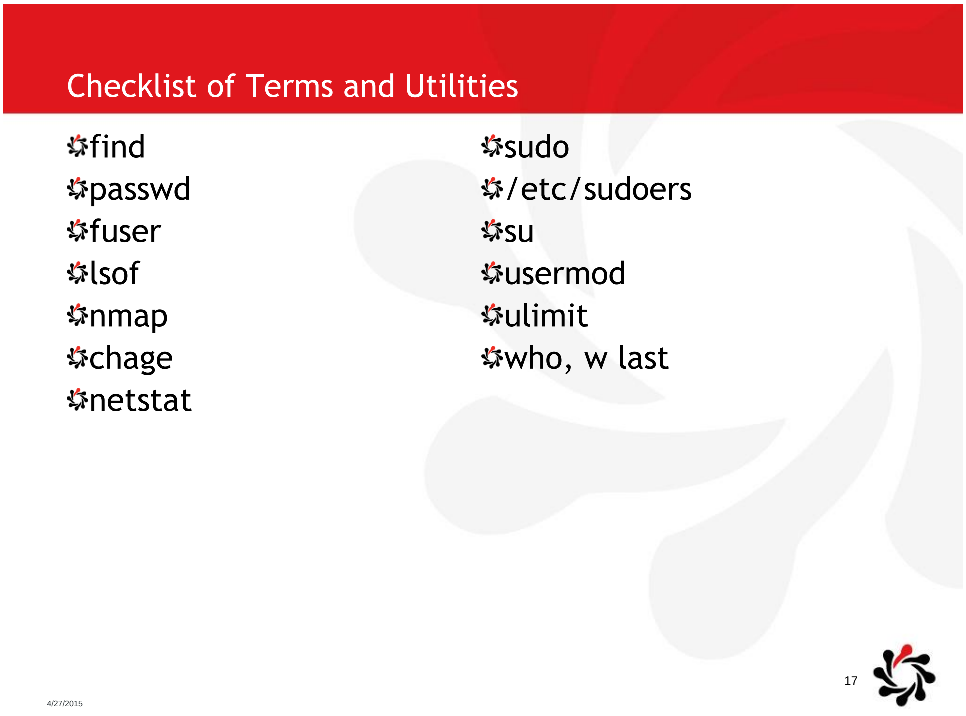
**Description** Candidates should know how to review system configuration to ensure host security in accordance with local security policies.

**Key Knowledge Areas**

- Audit a system to find files with the suid/sgid bit set.
- Set or change user passwords and password aging information.
- Being able to use nmap and netstat to discover open ports on a system.
- Set up limits on user logins, processes and memory usage.
- Determine which users have logged in to the system or are currently logged in
- Basic sudo configuration and usage.

16

4/27/2015

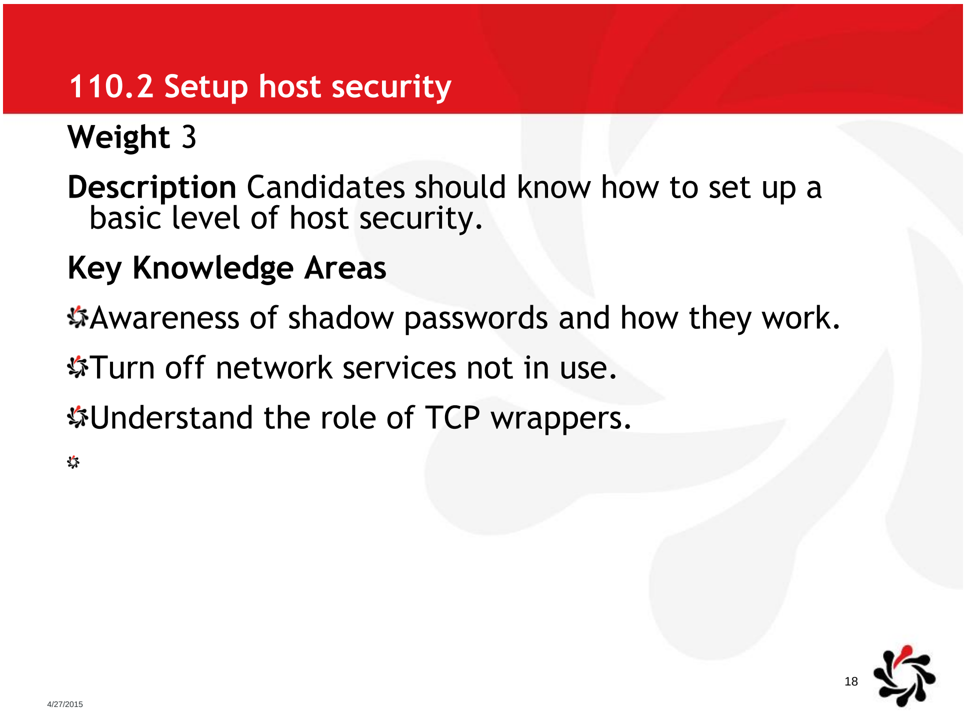


## Checklist of Terms and Utilities

- find
- passwd
- fuser
- lsof
- nmap
- chage
- netstat
- sudo
- /etc/sudoers
- su
- usermod
- ulimit
- who, w last

4/27/2015

17



## 110.2 Setup host security

**Weight** 3

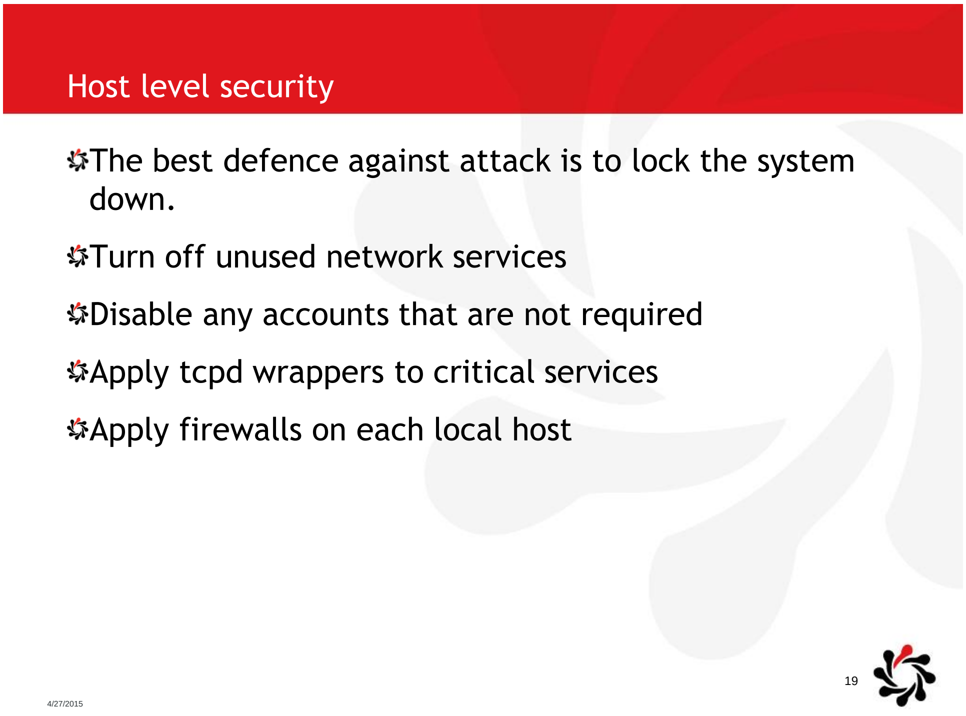
**Description** Candidates should know how to set up a basic level of host security.

**Key Knowledge Areas**

- Awareness of shadow passwords and how they work.
- Turn off network services not in use.
- Understand the role of TCP wrappers.

4/27/2015

18

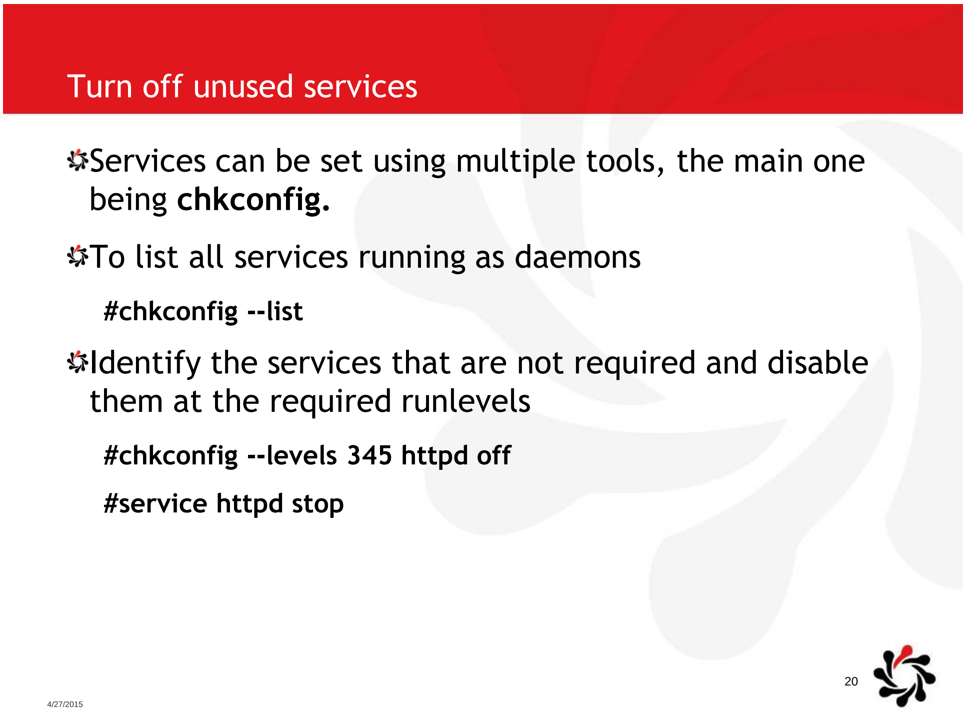


## Host level security

- The best defence against attack is to lock the system down.
- Turn off unused network services
- Disable any accounts that are not required
- Apply tcpd wrappers to critical services
- Apply firewalls on each local host

19

4/27/2015



## Turn off unused services

- Services can be set using multiple tools, the main one being **chkconfig.**
- To list all services running as daemons

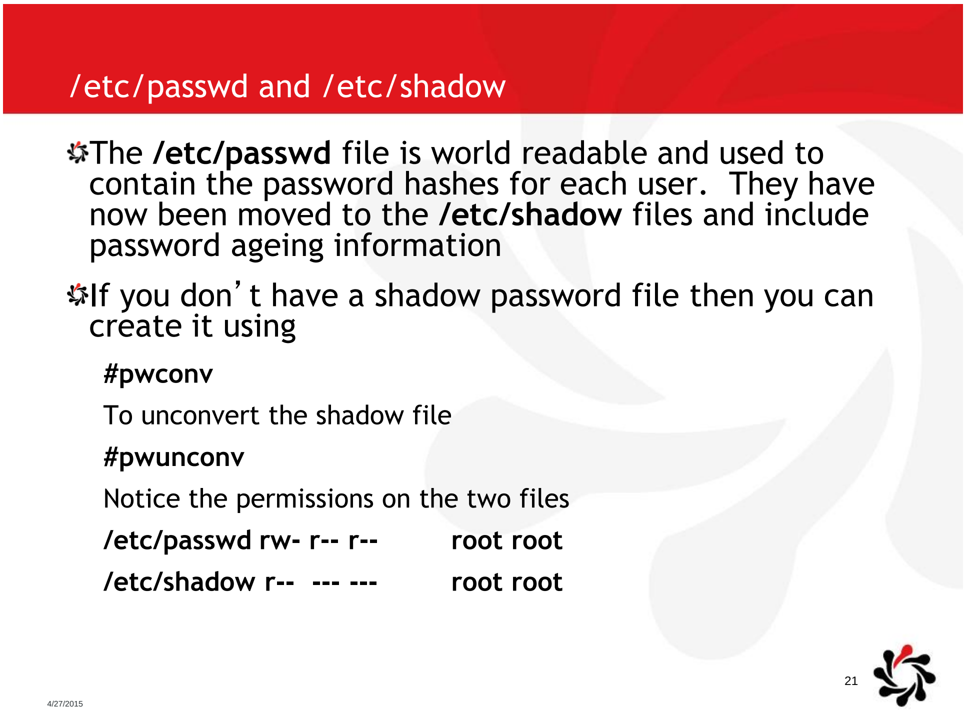
```
#chkconfig --list
```

- Identify the services that are not required and disable them at the required runlevels

```
#chkconfig --levels 345 httpd off
#service httpd stop
```

20

4/27/2015



## /etc/passwd and /etc/shadow

- The **/etc/passwd** file is world readable and used to contain the password hashes for each user.  They have now been moved to the **/etc/shadow** files and include password ageing information
- If you don’t have a shadow password file then you can create it using

```
#pwconv
```

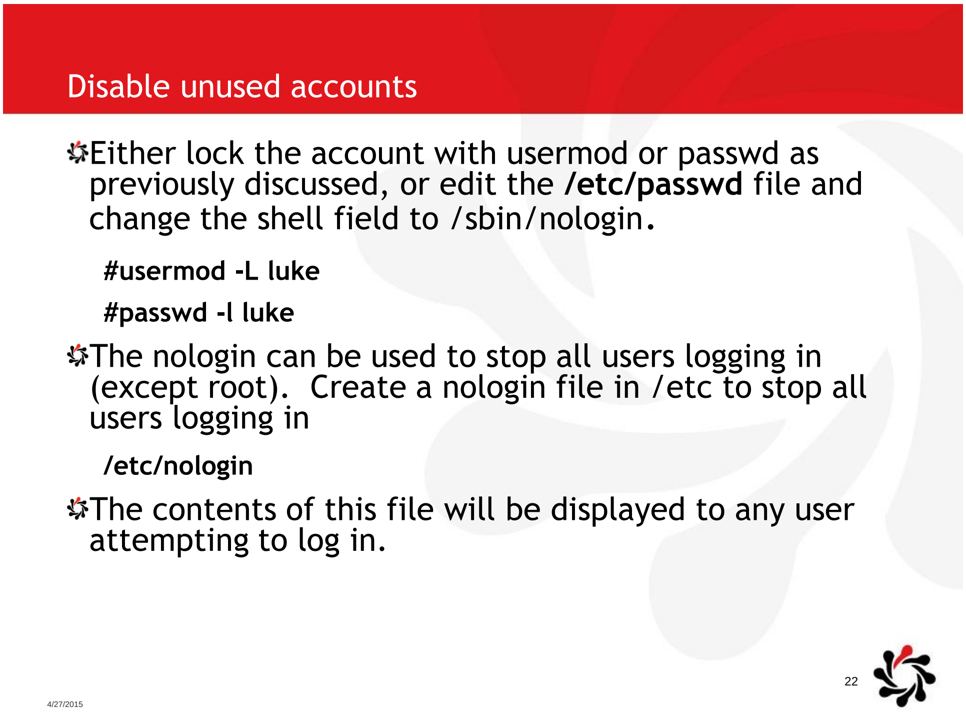
To unconvert the shadow file

```
#pwunconv
```

Notice the permissions on the two files

```
/etc/passwd rw- r-- r--        root root
/etc/shadow r--  --- ---       root root
```

4/27/2015 21



## Disable unused accounts

- Either lock the account with usermod or passwd as previously discussed, or edit the **/etc/passwd** file and change the shell field to /sbin/nologin.

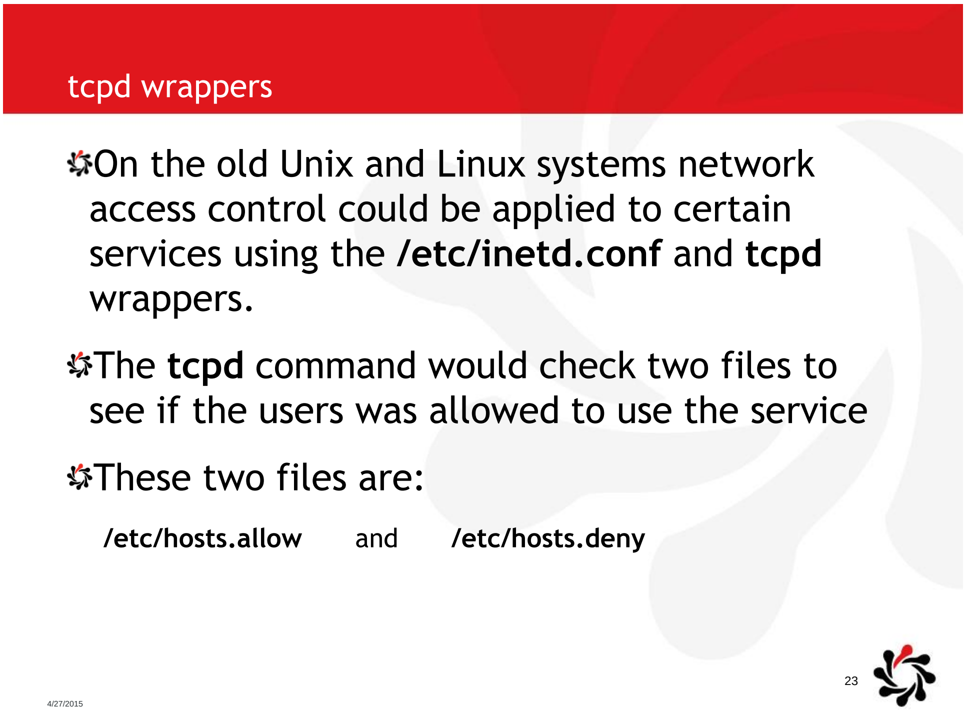
```
#usermod -L luke
#passwd -l luke
```

- The nologin can be used to stop all users logging in (except root).  Create a nologin file in /etc to stop all users logging in

```
/etc/nologin
```

- The contents of this file will be displayed to any user attempting to log in.

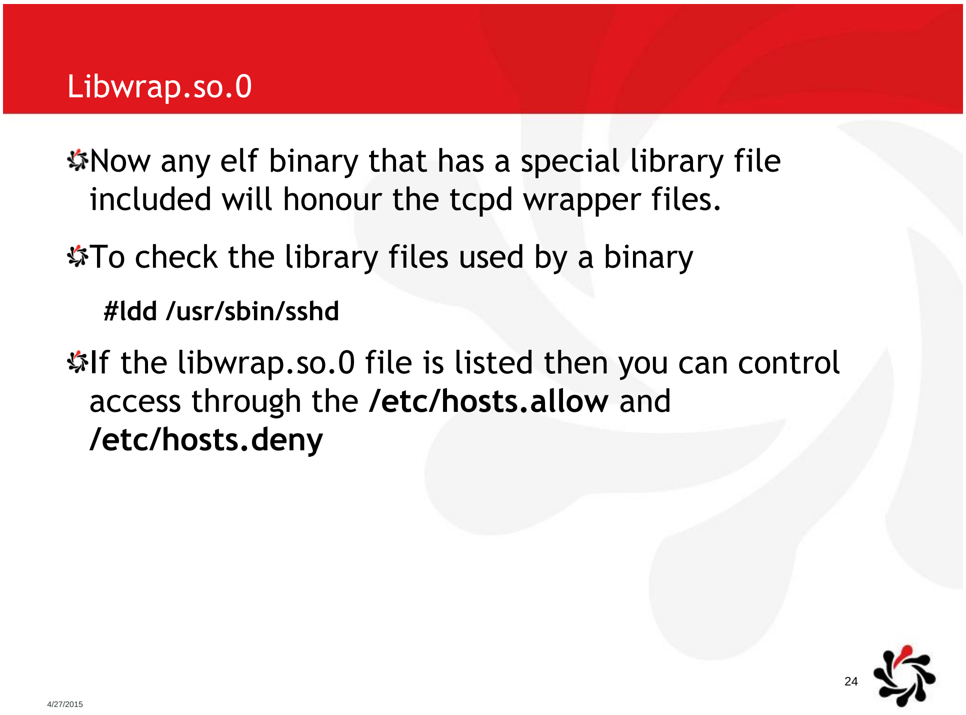
4/27/2015 22



## tcpd wrappers

- On the old Unix and Linux systems network access control could be applied to certain services using the **/etc/inetd.conf** and **tcpd** wrappers.
- The **tcpd** command would check two files to see if the users was allowed to use the service
- These two files are:

**/etc/hosts.allow** and **/etc/hosts.deny**

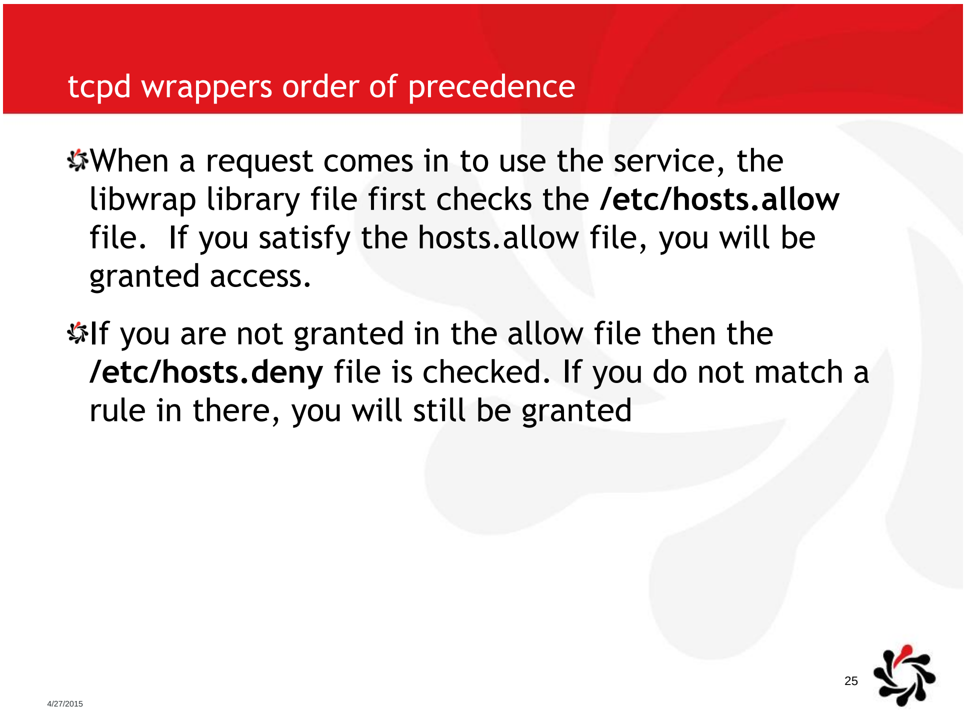


## Libwrap.so.0

- Now any elf binary that has a special library file included will honour the tcpd wrapper files.
- To check the library files used by a binary

```
#ldd /usr/sbin/sshd
```

- If the libwrap.so.0 file is listed then you can control access through the **/etc/hosts.allow** and **/etc/hosts.deny**

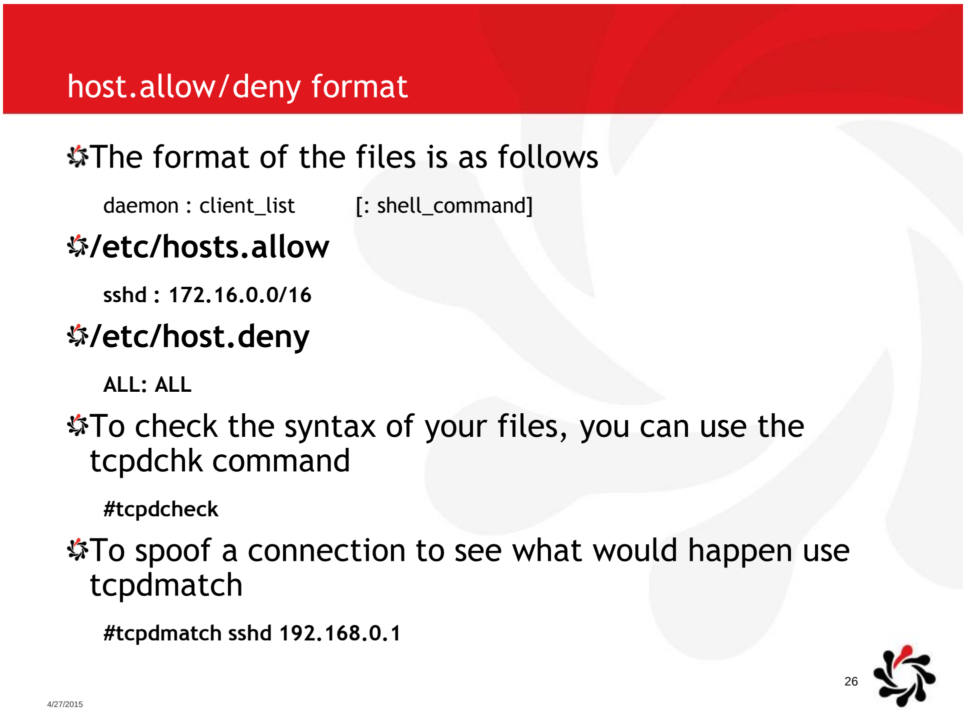


## tcpd wrappers order of precedence

- When a request comes in to use the service, the libwrap library file first checks the **/etc/hosts.allow** file.  If you satisfy the hosts.allow file, you will be granted access.
- If you are not granted in the allow file then the **/etc/hosts.deny** file is checked. If you do not match a rule in there, you will still be granted

4/27/2015

25



## host.allow/deny format

- The format of the files is as follows

```
daemon : client_list        [: shell_command]
```

- **/etc/hosts.allow**

```
sshd : 172.16.0.0/16
```

- **/etc/host.deny**

```
ALL: ALL
```

- To check the syntax of your files, you can use the tcpdchk command

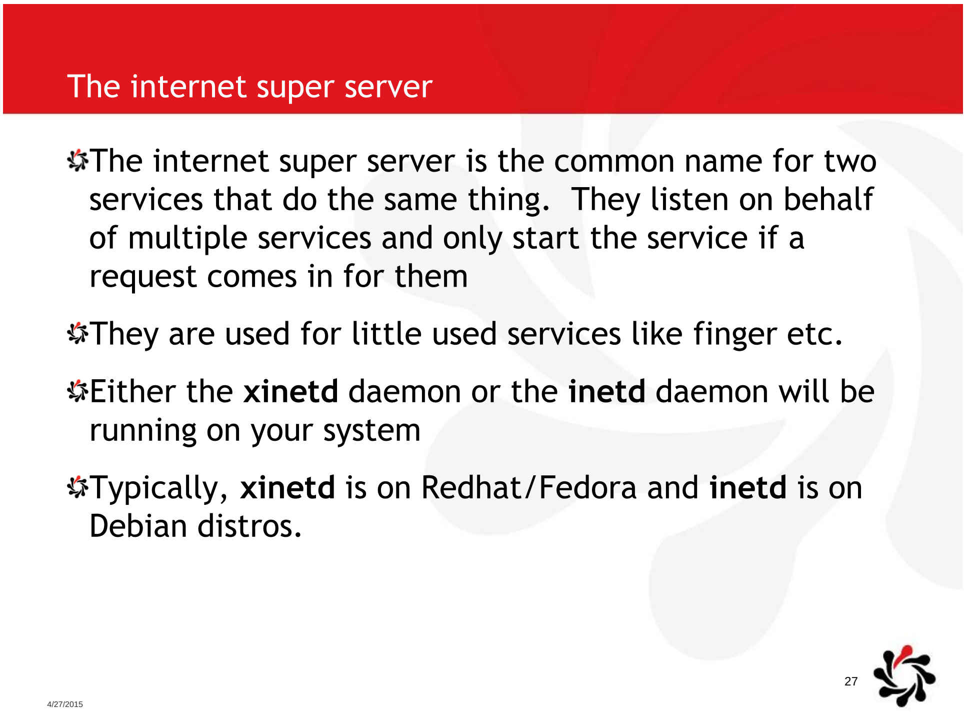
```
#tcpdcheck
```

- To spoof a connection to see what would happen use tcpdmatch

```
#tcpdmatch sshd 192.168.0.1
```

4/27/2015

26

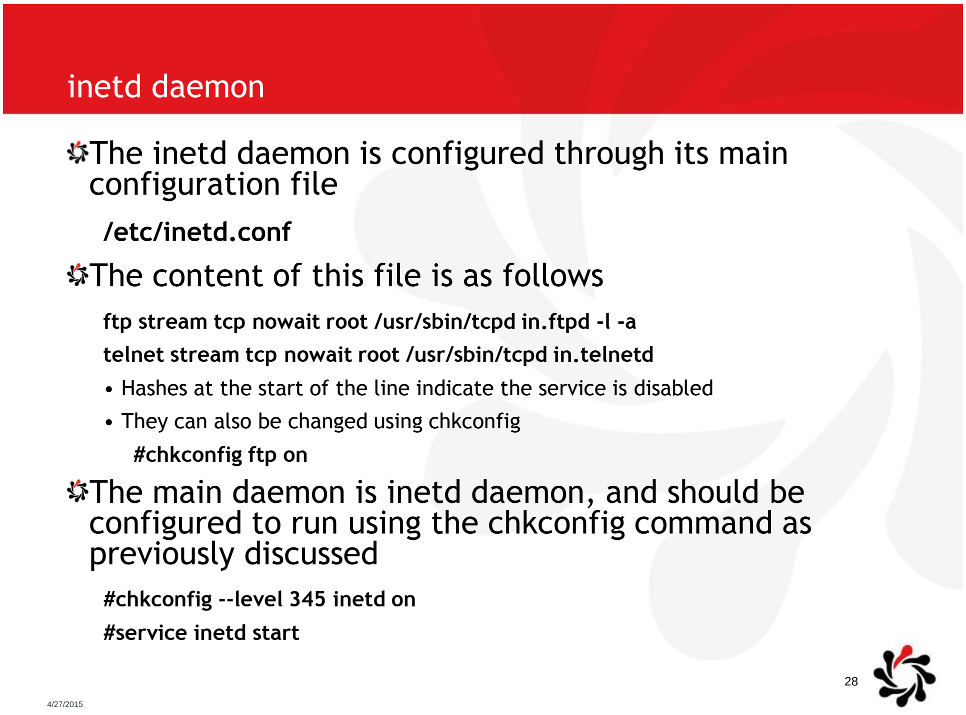


## The internet super server

- The internet super server is the common name for two services that do the same thing. They listen on behalf of multiple services and only start the service if a request comes in for them
- They are used for little used services like finger etc.
- Either the **xinetd** daemon or the **inetd** daemon will be running on your system
- Typically, **xinetd** is on Redhat/Fedora and **inetd** is on Debian distros.

27

4/27/2015



## inetd daemon

- The inetd daemon is configured through its main configuration file

  **/etc/inetd.conf**
- The content of this file is as follows

```
ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
```

  - Hashes at the start of the line indicate the service is disabled
  - They can also be changed using chkconfig

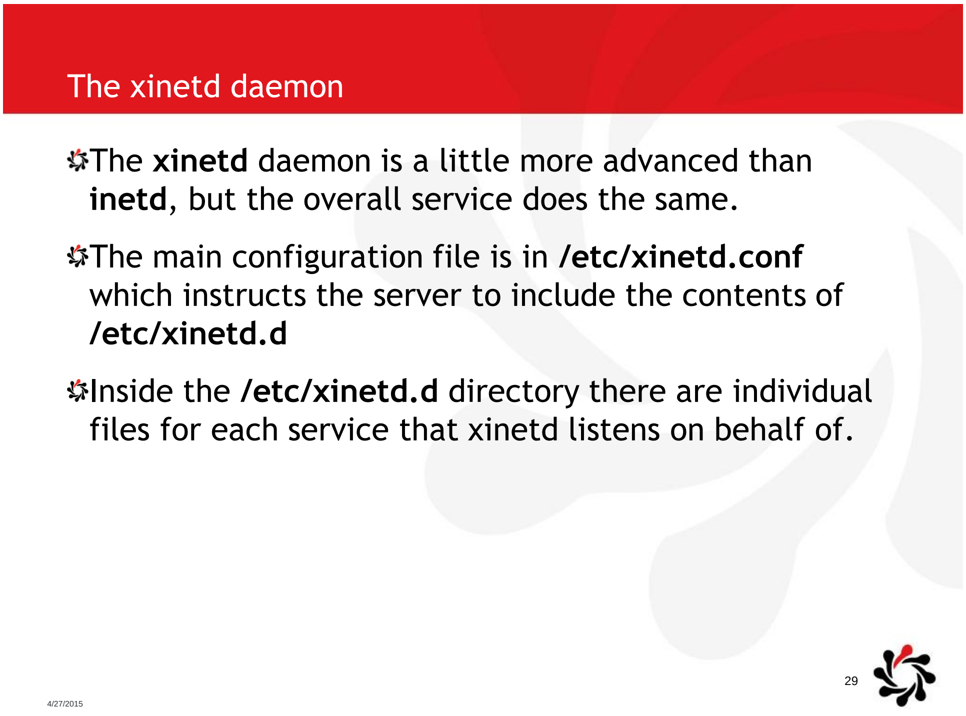
```
#chkconfig ftp on
```

- The main daemon is inetd daemon, and should be configured to run using the chkconfig command as previously discussed

```
#chkconfig --level 345 inetd on
#service inetd start
```

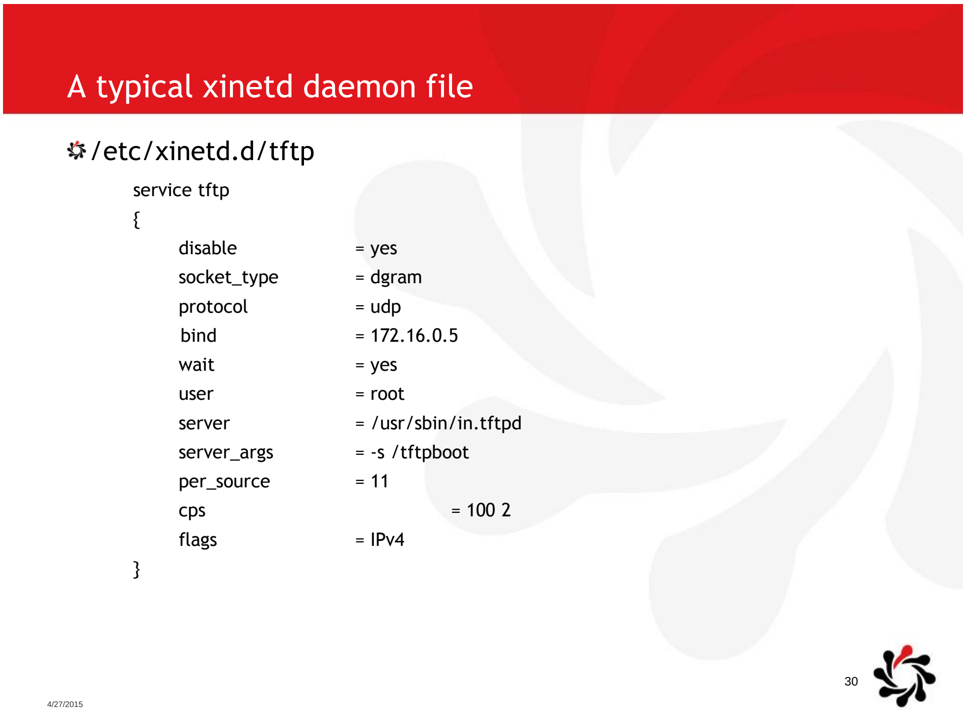
28

4/27/2015



## The xinetd daemon

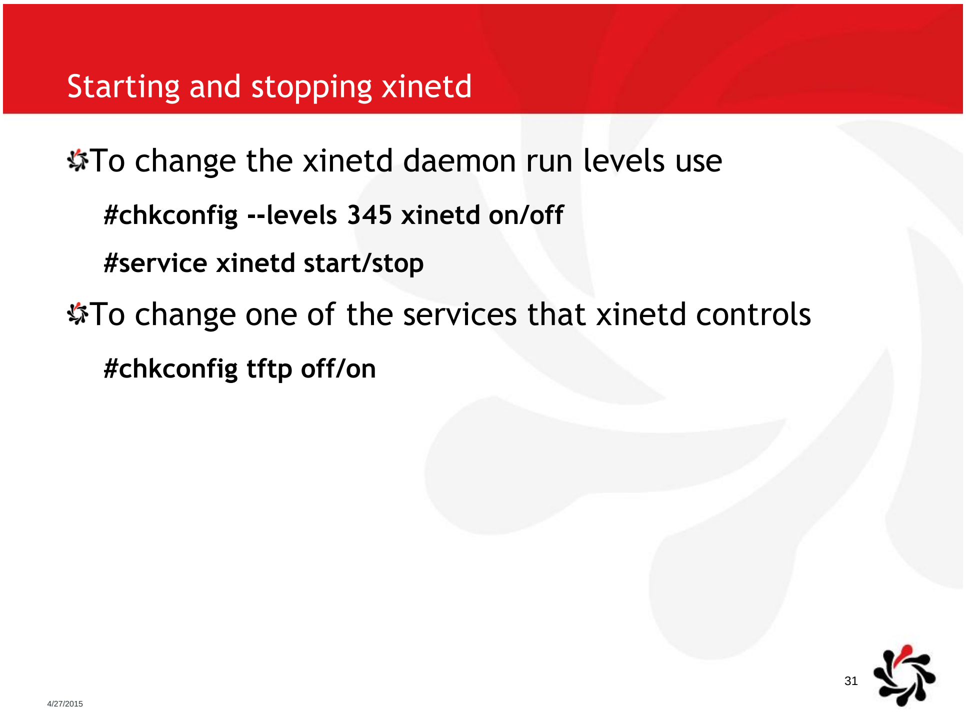
- The **xinetd** daemon is a little more advanced than **inetd**, but the overall service does the same.
- The main configuration file is in **/etc/xinetd.conf** which instructs the server to include the contents of **/etc/xinetd.d**
- Inside the **/etc/xinetd.d** directory there are individual files for each service that xinetd listens on behalf of.



## A typical xinetd daemon file

- /etc/xinetd.d/tftp

```
service tftp
{
        disable                 = yes
        socket_type             = dgram
        protocol                = udp
        bind                    = 172.16.0.5
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /tftpboot
        per_source              = 11
        cps                                 = 100 2
        flags                   = IPv4
}
```



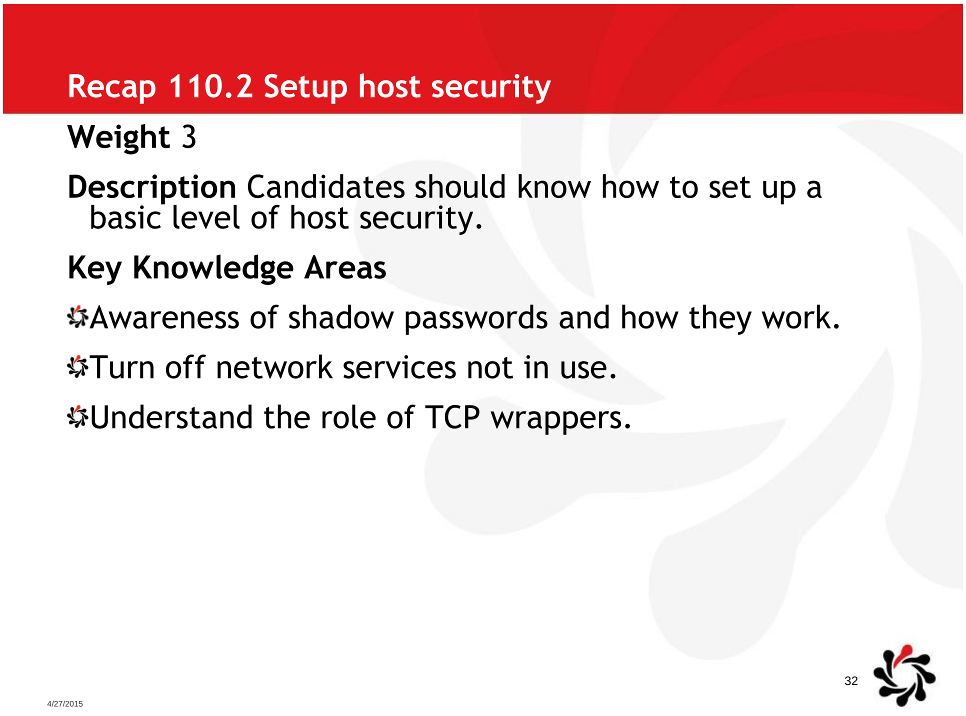
## Starting and stopping xinetd

- To change the xinetd daemon run levels use

```
#chkconfig --levels 345 xinetd on/off
#service xinetd start/stop
```

- To change one of the services that xinetd controls

```
#chkconfig tftp off/on
```



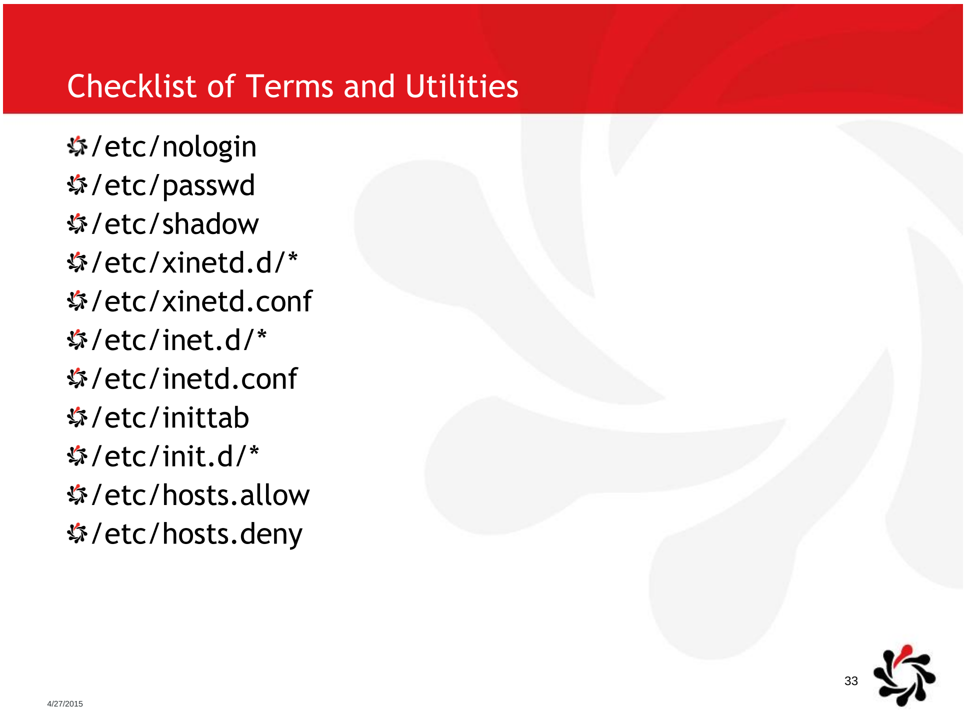
## Recap 110.2 Setup host security

**Weight** 3

**Description** Candidates should know how to set up a basic level of host security.

**Key Knowledge Areas**

- Awareness of shadow passwords and how they work.
- Turn off network services not in use.
- Understand the role of TCP wrappers.

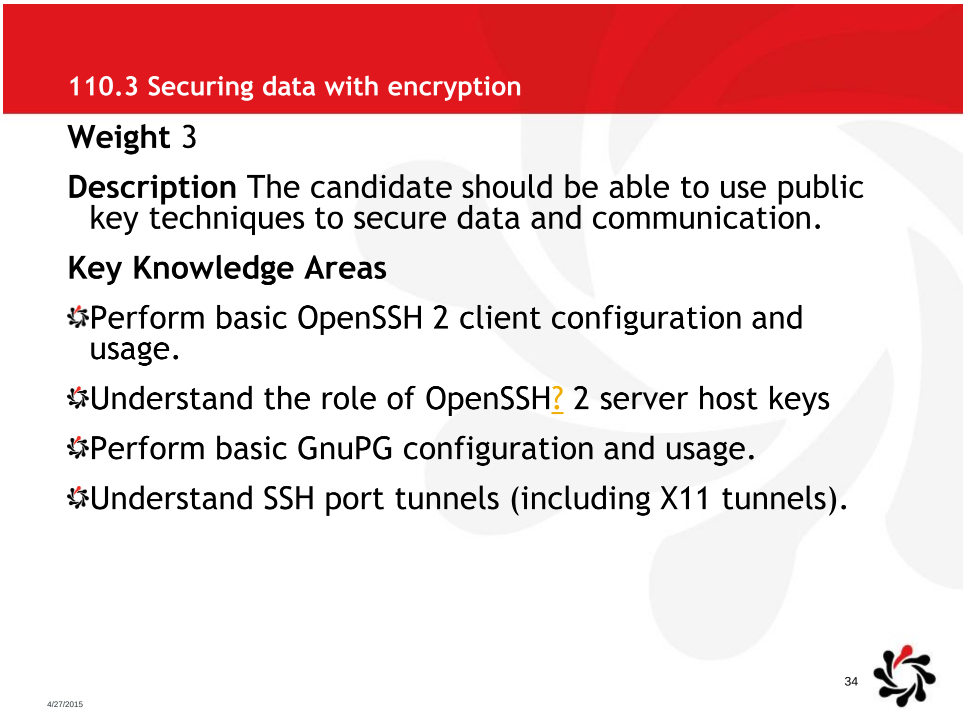


## Checklist of Terms and Utilities

- /etc/nologin
- /etc/passwd
- /etc/shadow
- /etc/xinetd.d/*
- /etc/xinetd.conf
- /etc/inet.d/*
- /etc/inetd.conf
- /etc/inittab
- /etc/init.d/*
- /etc/hosts.allow
- /etc/hosts.deny

33

4/27/2015



## 110.3 Securing data with encryption

**Weight** 3

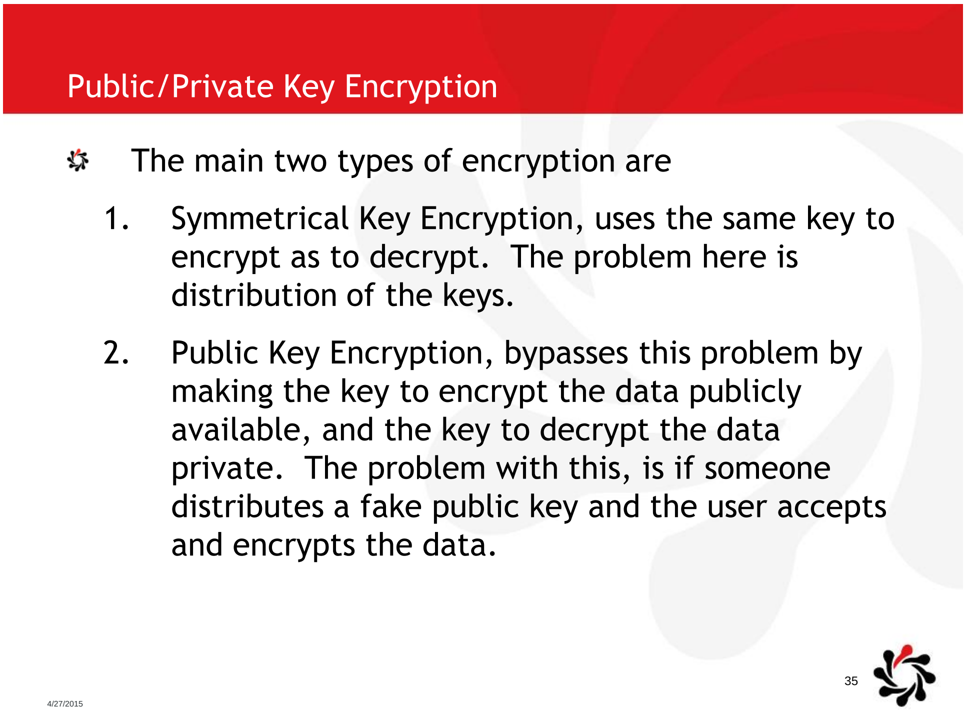
**Description** The candidate should be able to use public key techniques to secure data and communication.

**Key Knowledge Areas**

- Perform basic OpenSSH 2 client configuration and usage.
- Understand the role of OpenSSH? 2 server host keys
- Perform basic GnuPG configuration and usage.
- Understand SSH port tunnels (including X11 tunnels).

34

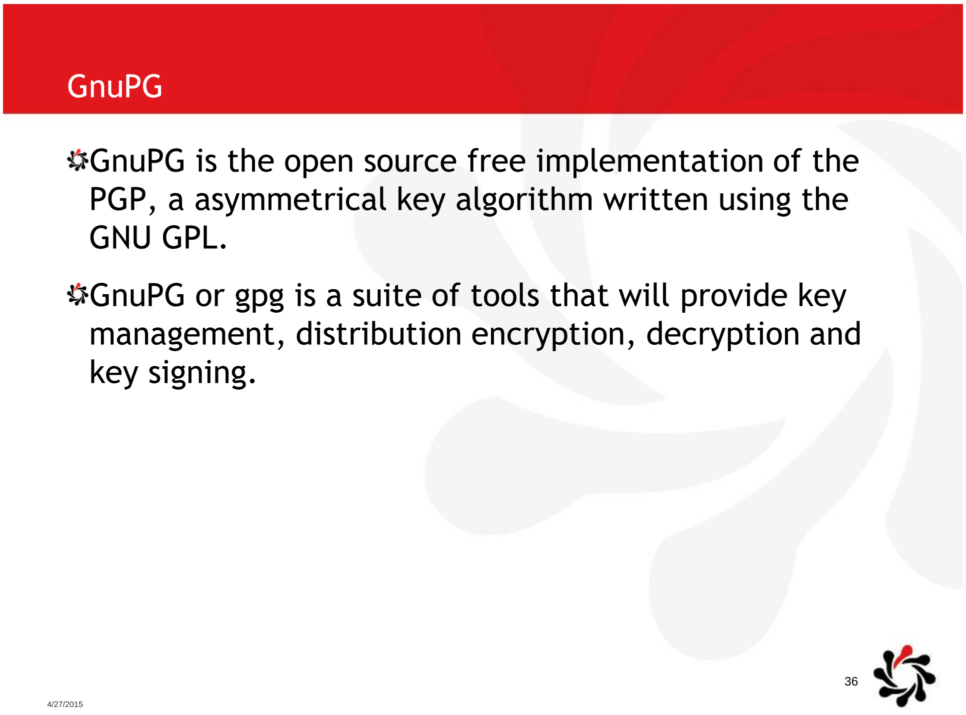
4/27/2015



## Public/Private Key Encryption

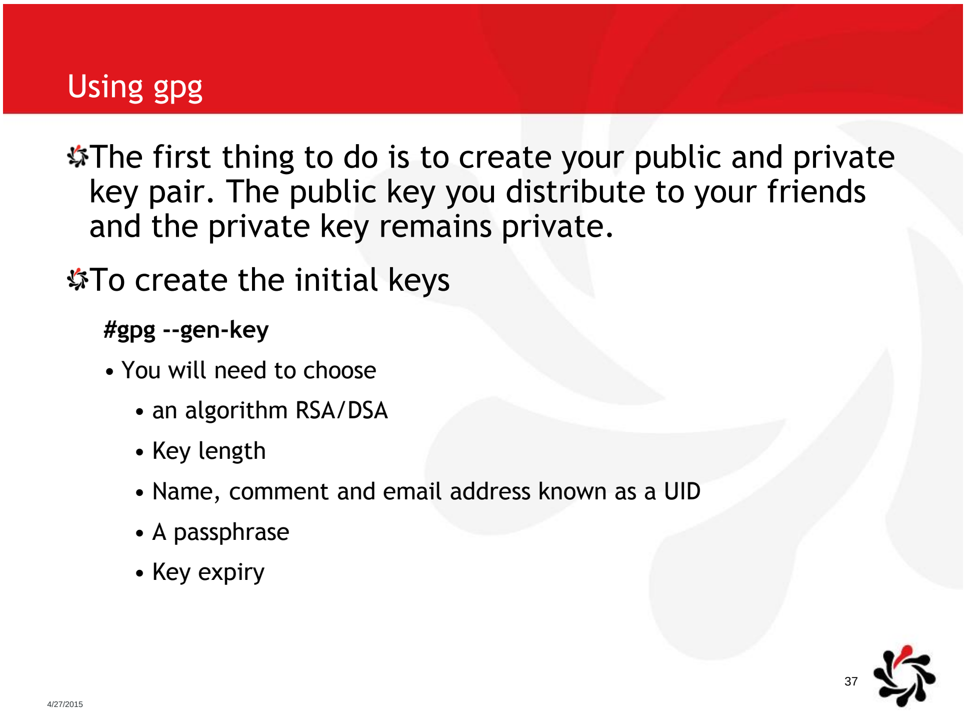
- The main two types of encryption are

1. Symmetrical Key Encryption, uses the same key to encrypt as to decrypt. The problem here is distribution of the keys.
2. Public Key Encryption, bypasses this problem by making the key to encrypt the data publicly available, and the key to decrypt the data private. The problem with this, is if someone distributes a fake public key and the user accepts and encrypts the data.



## GnuPG

- GnuPG is the open source free implementation of the PGP, a asymmetrical key algorithm written using the GNU GPL.
- GnuPG or gpg is a suite of tools that will provide key management, distribution encryption, decryption and key signing.



## Using gpg

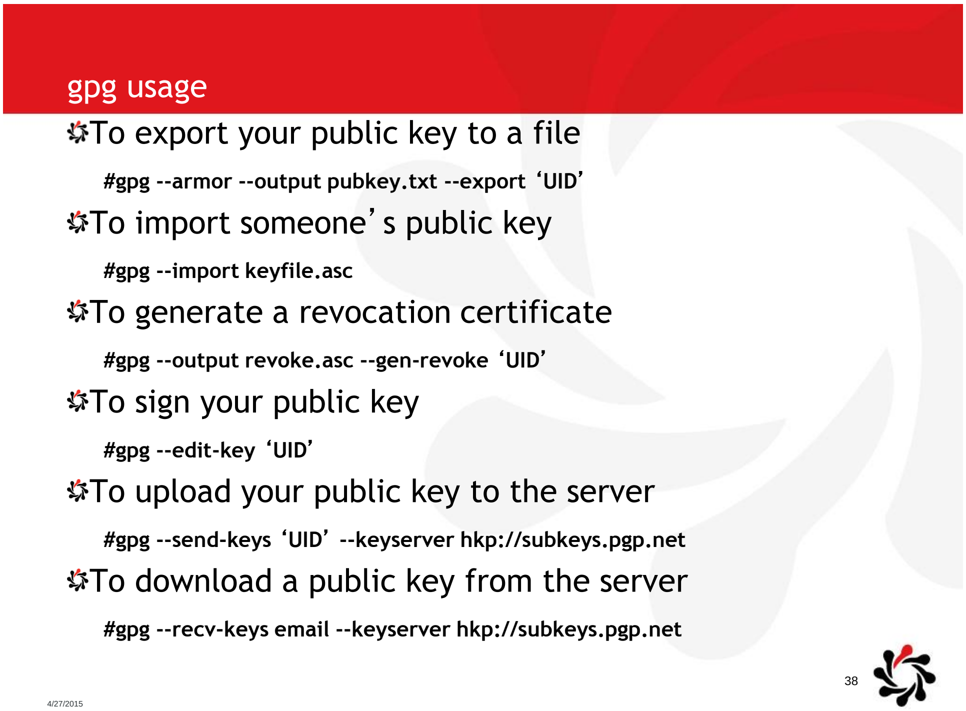
- The first thing to do is to create your public and private key pair. The public key you distribute to your friends and the private key remains private.
- To create the initial keys

  ```
  #gpg --gen-key
  ```

  - You will need to choose
    - an algorithm RSA/DSA
    - Key length
    - Name, comment and email address known as a UID
    - A passphrase
    - Key expiry

37

4/27/2015



## gpg usage

- To export your public key to a file

  ```
  #gpg --armor --output pubkey.txt --export ‘UID’
  ```

- To import someone’s public key

  ```
  #gpg --import keyfile.asc
  ```

- To generate a revocation certificate

  ```
  #gpg --output revoke.asc --gen-revoke ‘UID’
  ```

- To sign your public key

  ```
  #gpg --edit-key ‘UID’
  ```

- To upload your public key to the server

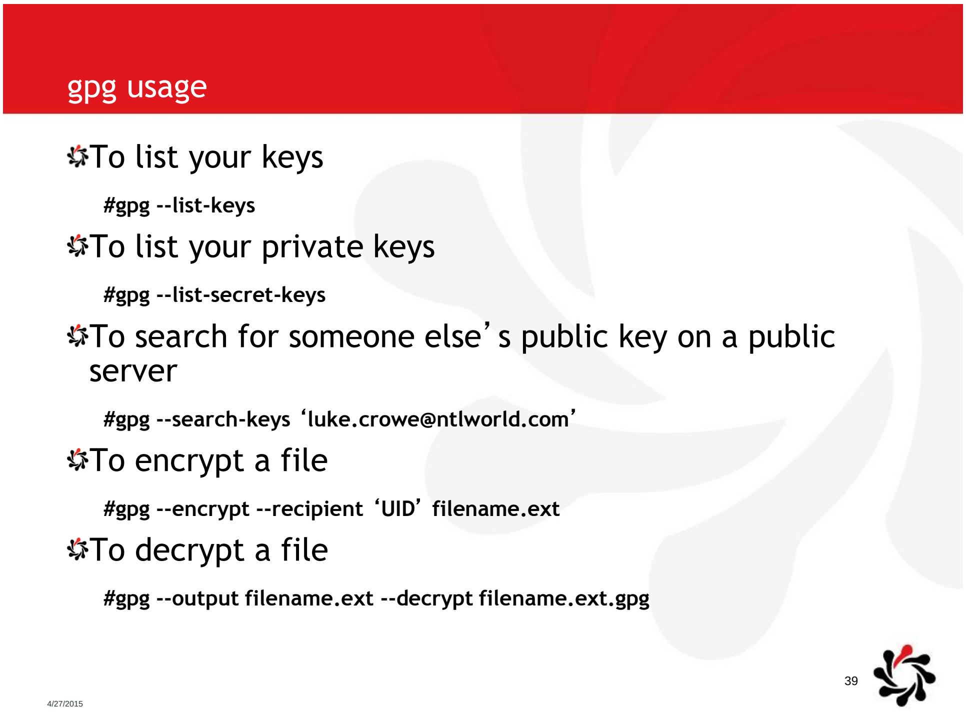
  ```
  #gpg --send-keys ‘UID’ --keyserver hkp://subkeys.pgp.net
  ```

- To download a public key from the server

  ```
  #gpg --recv-keys email --keyserver hkp://subkeys.pgp.net
  ```

38

4/27/2015



## gpg usage

- To list your keys

```
#gpg --list-keys
```

- To list your private keys

```
#gpg --list-secret-keys
```

- To search for someone else’s public key on a public server

```
#gpg --search-keys ‘luke.crowe@ntlworld.com’
```

- To encrypt a file

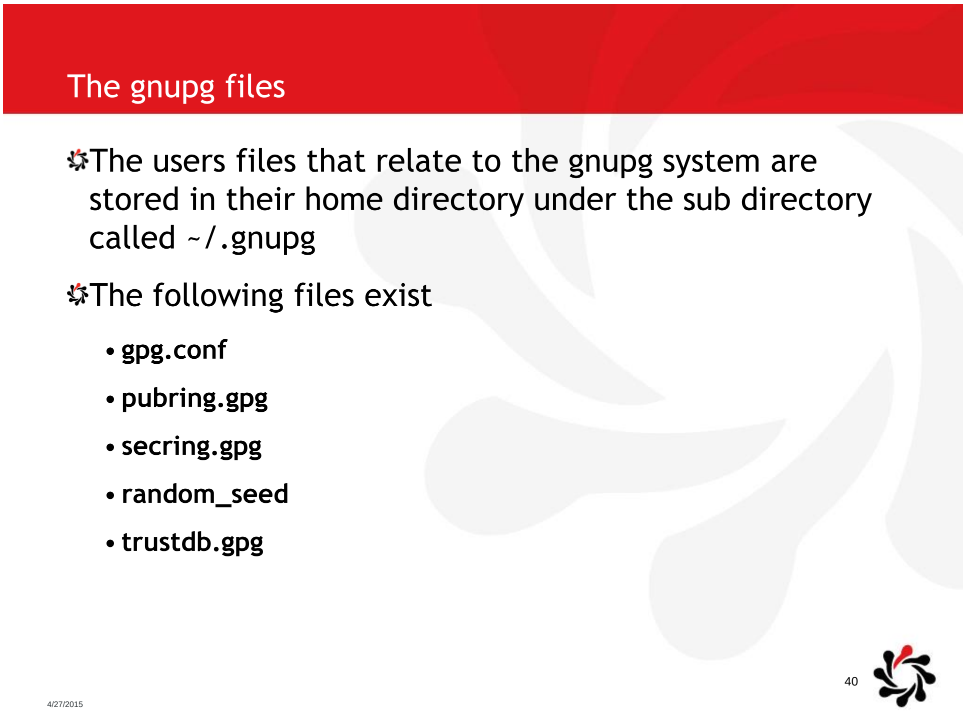
```
#gpg --encrypt --recipient ‘UID’ filename.ext
```

- To decrypt a file

```
#gpg --output filename.ext --decrypt filename.ext.gpg
```

39

4/27/2015

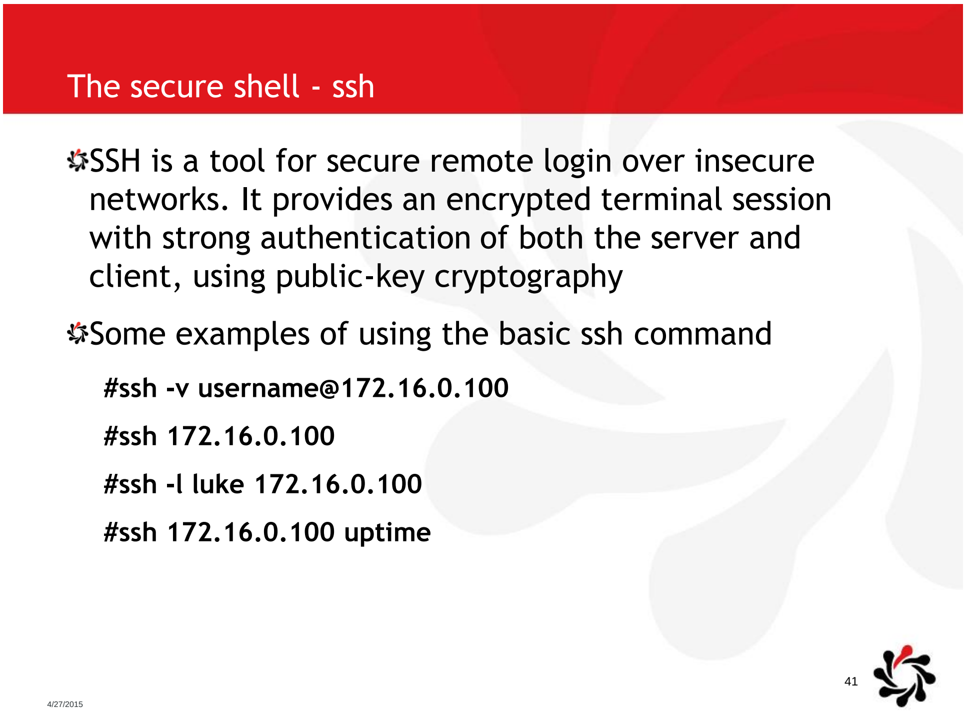


## The gnupg files

- The users files that relate to the gnupg system are stored in their home directory under the sub directory called ~/.gnupg
- The following files exist
  - **gpg.conf**
  - **pubring.gpg**
  - **secring.gpg**
  - **random_seed**
  - **trustdb.gpg**

40

4/27/2015



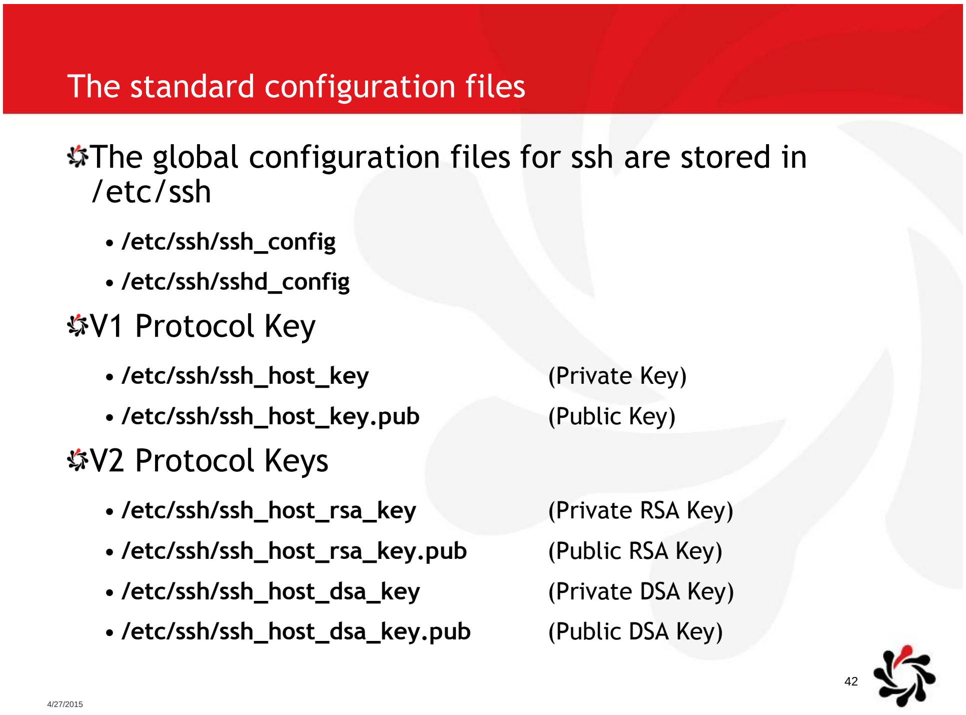
## The secure shell - ssh

- SSH is a tool for secure remote login over insecure networks. It provides an encrypted terminal session with strong authentication of both the server and client, using public-key cryptography
- Some examples of using the basic ssh command

```
#ssh -v username@172.16.0.100
#ssh 172.16.0.100
#ssh -l luke 172.16.0.100
#ssh 172.16.0.100 uptime
```

4/27/2015

41



## The standard configuration files

- The global configuration files for ssh are stored in /etc/ssh
  - /etc/ssh/ssh_config
  - /etc/ssh/sshd_config
- V1 Protocol Key

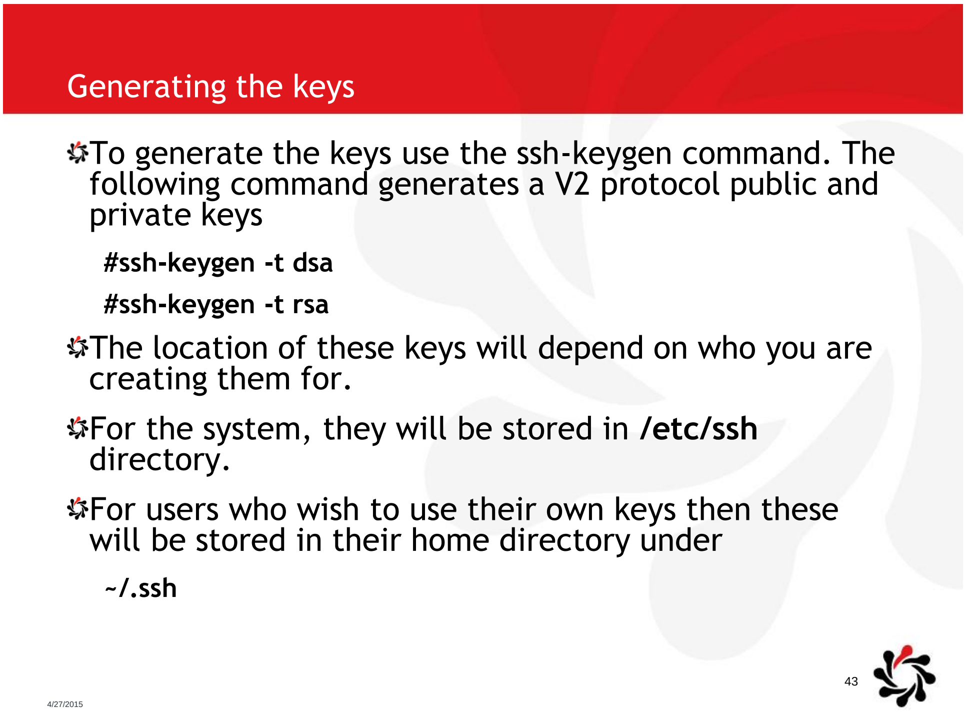
| | |
|---|---|
| /etc/ssh/ssh_host_key | (Private Key) |
| /etc/ssh/ssh_host_key.pub | (Public Key) |

- V2 Protocol Keys

| | |
|---|---|
| /etc/ssh/ssh_host_rsa_key | (Private RSA Key) |
| /etc/ssh/ssh_host_rsa_key.pub | (Public RSA Key) |
| /etc/ssh/ssh_host_dsa_key | (Private DSA Key) |
| /etc/ssh/ssh_host_dsa_key.pub | (Public DSA Key) |

4/27/2015

42



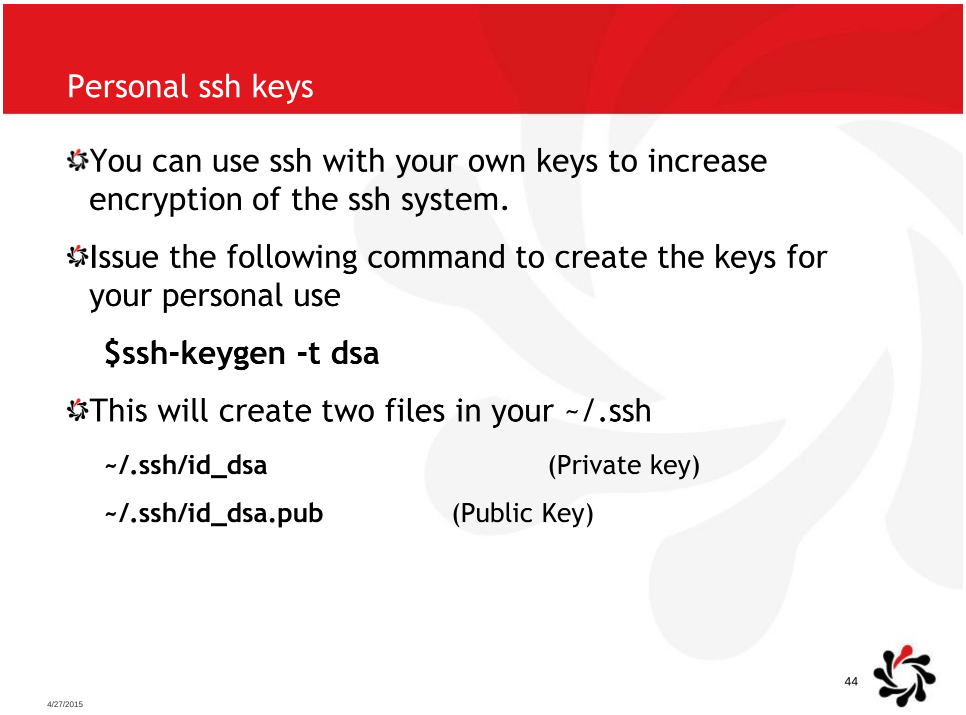
## Generating the keys

- To generate the keys use the ssh-keygen command. The following command generates a V2 protocol public and private keys

```
#ssh-keygen -t dsa
#ssh-keygen -t rsa
```

- The location of these keys will depend on who you are creating them for.
- For the system, they will be stored in **/etc/ssh** directory.
- For users who wish to use their own keys then these will be stored in their home directory under

  **~/.ssh**



## Personal ssh keys

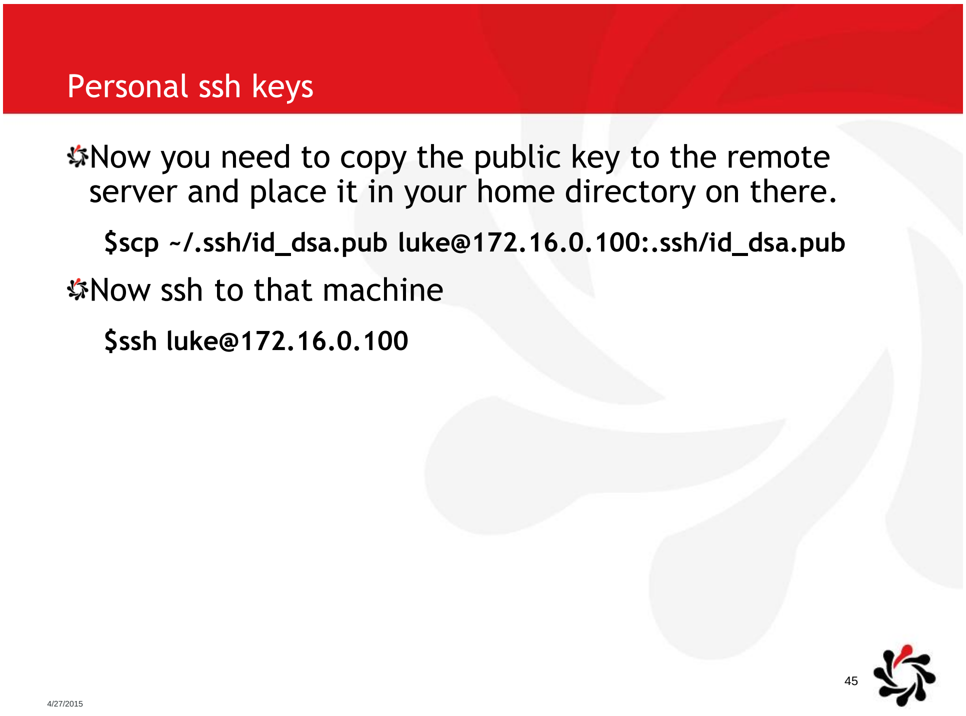
- You can use ssh with your own keys to increase encryption of the ssh system.
- Issue the following command to create the keys for your personal use

```
$ssh-keygen -t dsa
```

- This will create two files in your ~/.ssh

  **~/.ssh/id_dsa** (Private key)

  **~/.ssh/id_dsa.pub** (Public Key)



## Personal ssh keys

- Now you need to copy the public key to the remote server and place it in your home directory on there.

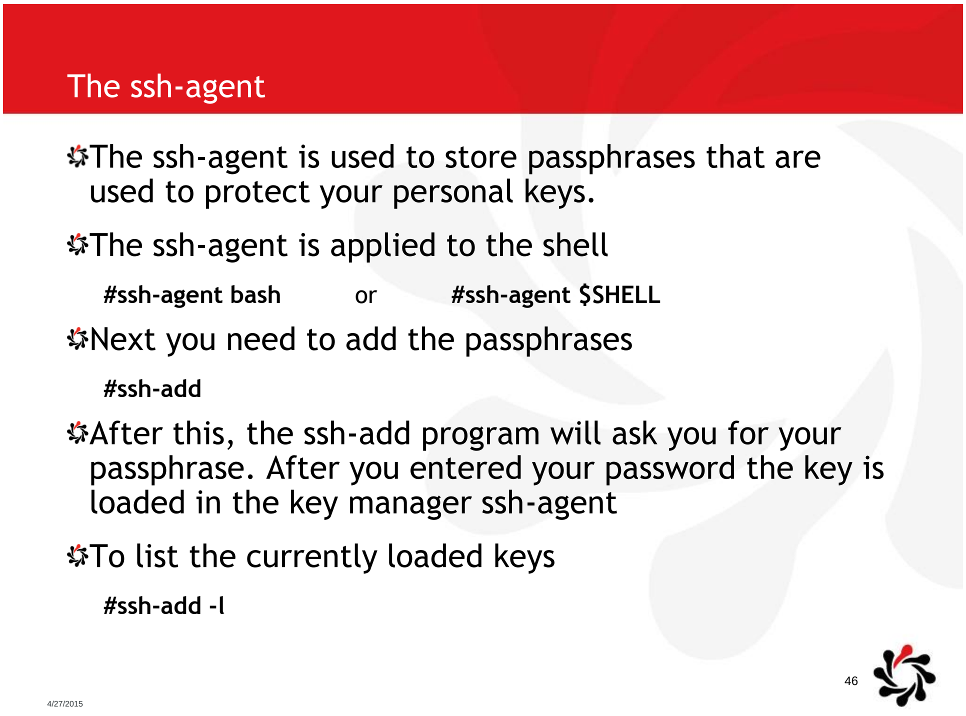
```
$scp ~/.ssh/id_dsa.pub luke@172.16.0.100:.ssh/id_dsa.pub
```

- Now ssh to that machine

```
$ssh luke@172.16.0.100
```

4/27/2015

45



## The ssh-agent

- The ssh-agent is used to store passphrases that are used to protect your personal keys.
- The ssh-agent is applied to the shell

```
#ssh-agent bash          or          #ssh-agent $SHELL
```

- Next you need to add the passphrases

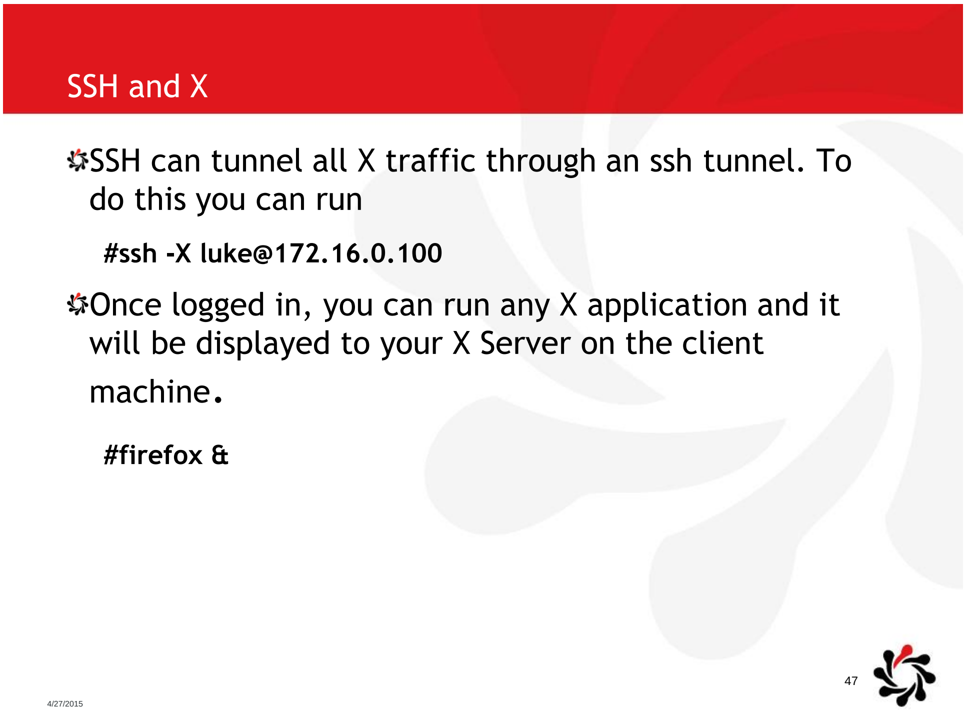
```
#ssh-add
```

- After this, the ssh-add program will ask you for your passphrase. After you entered your password the key is loaded in the key manager ssh-agent
- To list the currently loaded keys

```
#ssh-add -l
```

4/27/2015

46



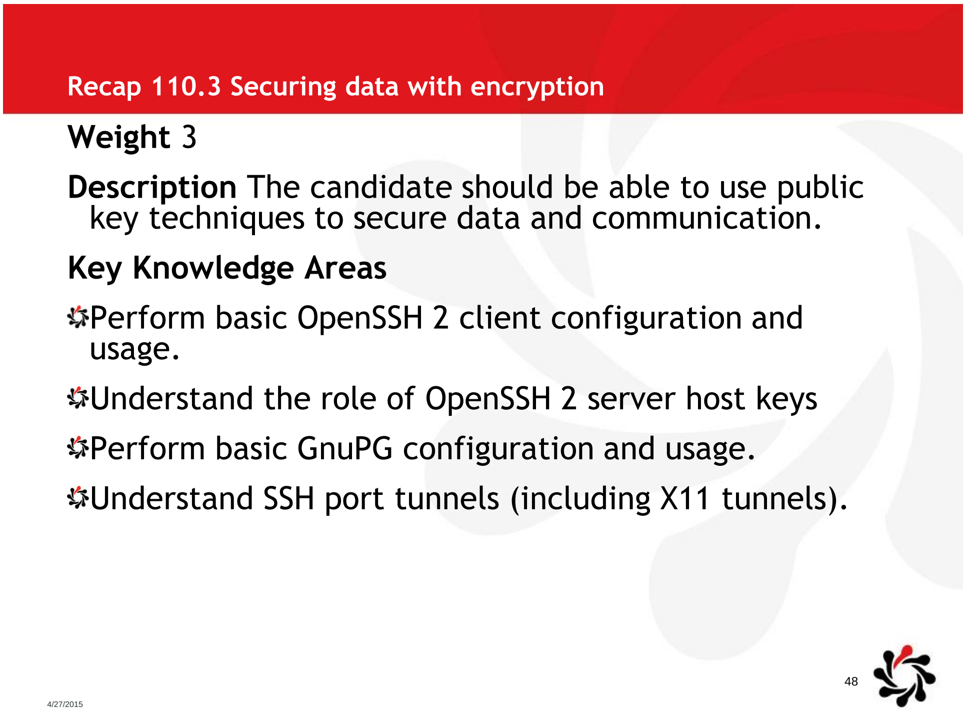
## SSH and X

- SSH can tunnel all X traffic through an ssh tunnel. To do this you can run

```
#ssh -X luke@172.16.0.100
```

- Once logged in, you can run any X application and it will be displayed to your X Server on the client machine.

```
#firefox &
```



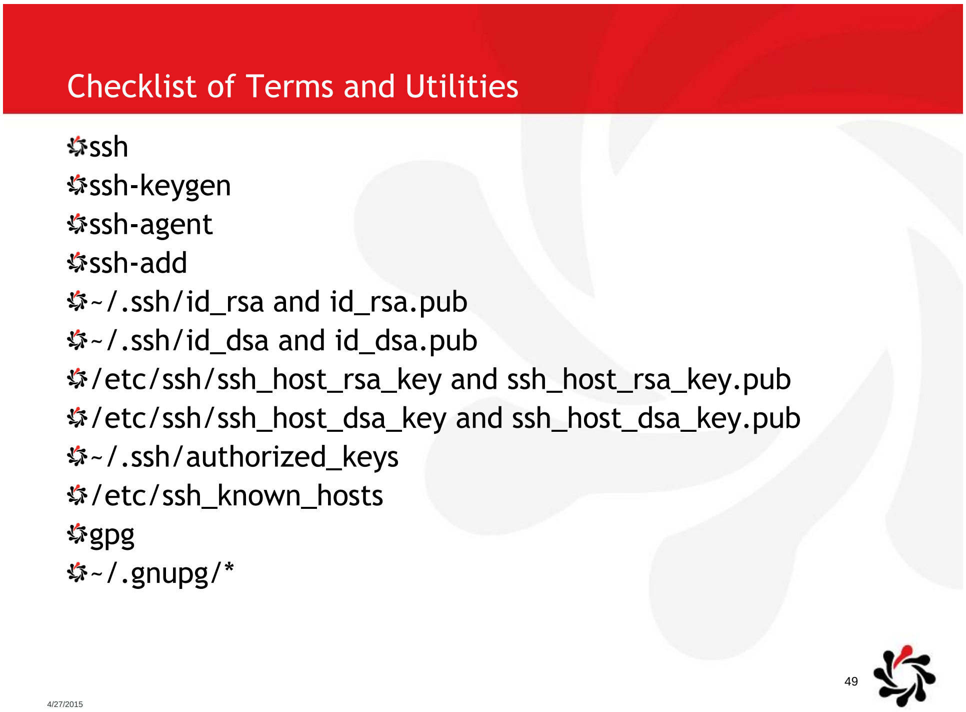
## Recap 110.3 Securing data with encryption

**Weight** 3

**Description** The candidate should be able to use public key techniques to secure data and communication.

**Key Knowledge Areas**

- Perform basic OpenSSH 2 client configuration and usage.
- Understand the role of OpenSSH 2 server host keys
- Perform basic GnuPG configuration and usage.
- Understand SSH port tunnels (including X11 tunnels).



## Checklist of Terms and Utilities

- ssh
- ssh-keygen
- ssh-agent
- ssh-add
- ~/.ssh/id_rsa and id_rsa.pub
- ~/.ssh/id_dsa and id_dsa.pub
- /etc/ssh/ssh_host_rsa_key and ssh_host_rsa_key.pub
- /etc/ssh/ssh_host_dsa_key and ssh_host_dsa_key.pub
- ~/.ssh/authorized_keys
- /etc/ssh_known_hosts
- gpg
- ~/.gnupg/*

49

4/27/2015