– General Linux 1 –

(Linux Professional Institute Certification)

[a]

```
   .~.
   /V\      by: geoffrey robertson
  // \\         geoffrey@zip.com.au
  @._.@
```

$Id: slides.tex,v 1.9 2003/05/30 04:45:24 waratah Exp $

List of Slides

General Linux 1

- (1.3) GNU and UNIX Commands [30]
- (2.4) Devices, Linux Filesystems & FHS [21]
- (2.6) Boot, Initialisation, Shutdown, Runlevels [6]
- (1.8) Documentation [8]
- (2.11) Administrative Tasks [24]

103 – (1.3) GNU and UNIX Commands [30]

1. Work on the command line [4]
2. Process text streams using filters [7]
3. Perform basic file management [2]
4. Use streams, pipes, and redirects [3]
5. Create, monitor, and kill processes [7]
6. Modify process execution priorities [2]
7. Search text files using regular expressions [3]
8. Perform basic file editing operations using vi [2]

103.1 – Work on the command line [4]

103.1 – Objective

Candidate should be able to interact with shells and commands using the command line. This includes typing valid commands and command sequences, defining, referencing and exporting environment variables, using command history and editing facilities, invoking commands in the path and outside the path, using command substitution, applying commands recursively through a directory tree and using man to find about commands.

103.1 – Key files, terms and utilities

- .
- bash
- echo
- env
- exec
- export
- man
- pwd
- set
- unset
- ~.bash_history
- ~/.profile

103.1 – test

this is the beginning

103.1 – test

this is the beginning

blah

103.1 – test

this is the beginning

blah blah blah

note... thing

103.1 – test

this is the beginning

blah blah blah blah

note... thing

103.1 – Shell Prompt

The command prompt is held in the shell variable $PS1 (Prompt String 1).

103.1 – Shell Prompt

The command prompt is held in the shell variable $PS1 (Prompt String 1).

- Simple User prompt:

```
$ echo $PS1  ↩
\$
$
```

103.1 – Shell Prompt

The command prompt is held in the shell variable $PS1 (Prompt String 1).

- Simple User prompt:

```
$ echo $PS1  ↩
\$
$
```

- Fancy user prompt:

```
$ export PS1="[\u@\\h:\W]\$ " ↩
[geoffrey@marsbar:geoffrey]$
```

103.1 – Shell Prompt

The command prompt is held in the shell variable $PS1 (Prompt String 1).

- Simple User prompt:

  ```
  $ echo $PS1  ←↩
  \$
  $
  ```

- Fancy user prompt:

  ```
  $ export PS1="[\u@\\h:\W]\$ " ←↩
  [geoffrey@marsbar:geoffrey]$
  ```

- Fancy prompt for the superuser:

  ```
  [root@marsbar root]# echo $PS1 ←↩
  \[\033[1;31m\][\u@\h \W]# \[\033[0m\]
  [root@marsbar root]#
  ```

103.1 – Command Execution

In order for bash to execute a command entered at the prompt, the command must be one of the following:

103.1 – Command Execution

In order for bash to execute a command entered at the prompt, the command must be one of the following:

- An alias: `$ alias ll='ls -l'` ↩

103.1 – Command Execution

In order for bash to execute a command entered at the prompt, the command must be one of the following:

- An alias: `$ alias ll='ls -l'` ↩
- A bash builtin: e.g. `$ cd` ↩

103.1 – Command Execution

In order for bash to execute a command entered at the prompt, the command must be one of the following:

- An alias: `$ alias ll='ls -l'` ↩
- A bash builtin: e.g. `$ cd` ↩
- An executable in a directory that is listed in the `$PATH` variable:

  ```
  $ which ls ↩
  /bin/ls
  ```

103.1 – Command Execution

In order for bash to execute a command entered at the prompt, the command must be one of the following:

- An alias: `$ alias ll='ls -l'` ←↩
- A bash builtin: e.g. `$ cd` ←↩
- An executable in a directory that is listed in the `$PATH` variable:

```
$ which ls ↩
/bin/ls
```

- Explicitly defined by absolute or relative path

```
$ ~/my_scripts/my.sh ↩
hi
$ cd ~/my_scripts ↩
$ ./my.sh ↩
```

103.1 – PATH

The PATH environmental variable holds a list of directories. These directories hold executable programs.

103.1 – PATH

The PATH environmental variable holds a list of directories. These directories hold executable programs.

- View the PATH variable:

```
$ echo $PATH
/usr/kerberos/bin:/bin:/usr/bin:/usr/bin/X11:\
/usr/local/bin:/usr/bin:/usr/X11R6/bin:/home/\
geoffrey/bin:/home/geoffrey/bin
```

103.1 – PATH

The PATH environmental variable holds a list of directories. These directories hold executable programs.

- View the PATH variable:

```
$ echo $PATH
/usr/kerberos/bin:/bin:/usr/bin:/usr/bin/X11:\
/usr/local/bin:/usr/bin:/usr/X11R6/bin:/home/\
geoffrey/bin:/home/geoffrey/bin
```

- General user PATH is configured in /etc/profile

103.1 – PATH

The PATH environmental variable holds a list of directories. These directories hold executable programs.

- View the PATH variable:

```
$ echo $PATH
/usr/kerberos/bin:/bin:/usr/bin:/usr/bin/X11:\
/usr/local/bin:/usr/bin:/usr/X11R6/bin:/home/\
geoffrey/bin:/home/geoffrey/bin
```

- General user PATH is configured in /etc/profile
- PATH may be modified in ~/.bash_profile

103.1 – Command history

103.1 – Command history

- Previously used commands accessed through the use of the arrow keys

103.1 – Command history

- Previously used commands accessed through the use of the arrow keys
- The size of the history file is determined by `HISTSIZE`

103.1 – Command history

- Previously used commands accessed through the use of the arrow keys
- The size of the history file is determined by `HISTSIZE`
- The command `history` lists the last `HISTSIZE` commands

```
$ history
...
 1007  xhost +
 1008  history
$
```

103.1 – Command history

- Previously used commands accessed through the use of the arrow keys
- The size of the history file is determined by `HISTSIZE`
- The command `history` lists the last `HISTSIZE` commands

```
$ history
...
 1007  xhost +
 1008  history
$
```

- The command `fc` opens the command in the default editor for editing

```
$ fc 1007
xhost +
~
~
```

103.1 –

103.1 –

103.2 – Process text streams using filters [7]

103.2 – Objective

Candidate should be able to apply filters to text streams. Tasks include sending text files and output streams through text utility filters to modify the output, and using standard UNIX commands found in the GNU textutils package.

103.3 – Perform basic file management [2]

103.3 – Objective

Candidate should be able to use the basic UNIX commands to copy, move, and remove files and directories. Tasks include advanced file management operations such as copying multiple files recursively, removing directories recursively, and moving files that meet a wildcard pattern. This includes using simple and advanced wildcard specifications to refer to files, as well as using find to locate and act on files based on type, size, or time.

103.4 – Use streams, pipes, and redirects [3]

103.4 – Objective

Candidate 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.

103.5 – Create, monitor, and kill processes [7]

103.5 – Objective

Candidate should be able to manage processes. This includes knowing how to run jobs in the foreground and background, bring a job from the background to the foreground and vice versa, start a process that will run without being connected to a terminal and signal a program to continue running after logout. Tasks also include monitoring active processes, selecting and sorting processes for display, sending signals to processes, killing processes and identifying and killing X applications that did not terminate after the X session closed.

103.5 – Create, monitor, and kill processes [7]

103.5 – Objective

Candidate should be able to manage processes. This includes knowing how to run jobs in the foreground and background, bring a job from the background to the foreground and vice versa, start a process that will run without being connected to a terminal and signal a program to continue running after logout. Tasks also include monitoring active processes, selecting and sorting processes for display, sending signals to processes, killing processes and identifying and killing X applications that did not terminate after the X session closed.

103.5 – Key files, terms and utilities

- `ps`
- `pstree`
- `top`
- `kill`
- `nohup`
- `jobs`
- `bg`
- `fg`
- `&`
- `^Z`

103.5 – Processes

- A process is an executable loaded in memory.
- Linux is a multitasking operating system and so runs many processes concurrently.

103.5 – Processes

- A process is an executable loaded in memory.
- Linux is a multitasking operating system and so runs many processes concurrently.
- INIT (PID 1) is the mother of all processes.

103.5 – Processes

- A process is an executable loaded in memory.
- Linux is a multitasking operating system and so runs many processes concurrently.
- INIT (PID 1) is the mother of all processes.
- Programs, daemons, shells and commands are all processes.

103.5 – Processes

- A process is an executable loaded in memory.
- Linux is a multitasking operating system and so runs many processes concurrently.
- INIT (PID 1) is the mother of all processes.
- Programs, daemons, shells and commands are all processes.
- The kernel automatically manages processes.

103.5 – Processes

- A process is an executable loaded in memory.
- Linux is a multitasking operating system and so runs many processes concurrently.
- INIT (PID 1) is the mother of all processes.
- Programs, daemons, shells and commands are all processes.
- The kernel automatically manages processes.
- Normally processes live, execute and die without intervention from users.

103.5 – Process Attributes and Concepts

The kernel starts the first process: `init` which has PID 1

103.5 – Process Attributes and Concepts

The kernel starts the first process: `init` which has PID 1

Lifetime: Each process starts when it's command is executed, and lives till it dies or is killed.

103.5 – Process Attributes and Concepts

The kernel starts the first process: `init` which has PID 1

Lifetime: Each process starts when it's command is executed, and lives till it dies or is killed.

Process ID (PID): Every process has a unique number assigned to it when it is started.

103.5 – Process Attributes and Concepts

The kernel starts the first process: `init` which has PID 1

Lifetime: Each process starts when it's command is executed, and lives till it dies or is killed.

Process ID (PID): Every process has a unique number assigned to it when it is started.

User ID and Group ID: Processes have the privileges associated with the user / group who started them.

103.5 – Process Attributes and Concepts

The kernel starts the first process: `init` which has PID 1

Lifetime: Each process starts when it's command is executed, and lives till it dies or is killed.

Process ID (PID): Every process has a unique number assigned to it when it is started.

User ID and Group ID: Processes have the privileges associated with the user / group who started them.

Parent processes (PPID): Shell processes are descendants of init and commands run from them are child processes.

103.5 – Process Attributes and Concepts

The kernel starts the first process: `init` which has PID 1

Lifetime: Each process starts when it's command is executed, and lives till it dies or is killed.

Process ID (PID): Every process has a unique number assigned to it when it is started.

User ID and Group ID: Processes have the privileges associated with the user / group who started them.

Parent processes (PPID): Shell processes are descendants of init and commands run from them are child processes.

Environment: Each process inherits a set of *environmental variables* from it's parent process.

103.5 – Process Attributes and Concepts

The kernel starts the first process: `init` which has PID 1

Lifetime: Each process starts when it's command is executed, and lives till it dies or is killed.

Process ID (PID): Every process has a unique number assigned to it when it is started.

User ID and Group ID: Processes have the privileges associated with the user / group who started them.

Parent processes (PPID): Shell processes are descendants of init and commands run from them are child processes.

Environment: Each process inherits a set of *environmental variables* from it's parent process.

Current Working Directory: Each process starts with a default directory.

103.5 – Process Monitoring

Processes have to be monitored so as to check their health and use of system resources.

103.5 – Process Monitoring

Processes have to be monitored so as to check their health and use of system resources.

- ps

```
$ ps aux |grep ssh
root  866  0.0  0.3  2676 1268 ?   S  07:56 0:00 /usr/sbin/sshd
```

103.5 – Process Monitoring

Processes have to be monitored so as to check their health and use of system resources.

- ps

```
$ ps aux |grep ssh
root  866  0.0  0.3  2676 1268 ?   S  07:56 0:00 /usr/sbin/sshd
```

- pstree

```
$ pstree
init-+-alarmd
     |-apmd
     |-kdeinit-+-autorun
     |         |-kdeinit---emacs
```

103.5 – Process Monitoring

Processes have to be monitored so as to check their health and use of system resources.

- ps

```
$ ps aux |grep ssh
root  866  0.0  0.3  2676 1268 ?   S  07:56 0:00 /usr/sbin/sshd
```

- pstree

```
$ pstree
init-+-alarmd
     |-apmd
     |-kdeinit-+-autorun
     |         |-kdeinit---emacs
```

- top

```
$ top
 PID USER     PRI NI  SIZE  RSS SHARE STAT %CPU %MEM  TIME COMMAND
1792 geoffrey  11  0  8796 8796  7932 S     0.3  2.2  0:01 kdeinit
1590 root      14  0 57512  13M  2572 R     0.1  3.6  0:41 X
2857 geoffrey  14  0  1056 1056   836 R     0.1  0.2  0:01 top
```

103.5 – Process Management

Normally the kernel automatically manages processes. However sometimes processes have to be started, stopped, restarted and killed.

103.5 – Process Management

Normally the kernel automatically manages processes. However sometimes processes have to be started, stopped, restarted and killed.

- Starting a process:

```
#  /usr/sbin/httpd
ps aux |grep httpd
root      2987  0.0  0.4  4512 1584 ?   /usr/sbin/httpd
apache    3003  0.0  0.4  4656 1672 ?   /usr/sbin/httpd
```

103.5 – Process Management

Normally the kernel automatically manages processes. However sometimes processes have to be started, stopped, restarted and killed.

- Starting a process:

```
#  /usr/sbin/httpd
ps aux |grep httpd
root      2987  0.0  0.4  4512 1584 ?   /usr/sbin/httpd
apache    3003  0.0  0.4  4656 1672 ?   /usr/sbin/httpd
```

- Occasionally processes die and have to be restarted.

103.5 – Process Management

Normally the kernel automatically manages processes. However sometimes processes have to be started, stopped, restarted and killed.

- Starting a process:

```
#  /usr/sbin/httpd
ps aux |grep httpd
root      2987  0.0  0.4  4512 1584 ?   /usr/sbin/httpd
apache    3003  0.0  0.4  4656 1672 ?   /usr/sbin/httpd
```

- Occasionally processes die and have to be restarted.
- Processes may go beserk and have to be killed.

```
# kill -9 1234
```

103.5 – Process Management

Normally the kernel automatically manages processes. However sometimes processes have to be started, stopped, restarted and killed.

- Starting a process:

```
#  /usr/sbin/httpd
ps aux |grep httpd
root      2987  0.0  0.4  4512 1584 ?   /usr/sbin/httpd
apache    3003  0.0  0.4  4656 1672 ?   /usr/sbin/httpd
```

- Occasionally processes die and have to be restarted.
- Processes may go beserk and have to be killed.

```
# kill -9 1234
```

- After configuration changes processes may have to be restarted so as to re-read their configuration files.

```
# service xinetd restart
Stopping xinetd:                  [  OK  ]
Starting xinetd:                  [  OK  ]
```

103.5 – What is multitasking?

Multitasking is used to describe the situation where one processor (CPU) is used to perform multiple tasks concurrently.

- Only one task or program is executing instructions on the CPU.
- The CPU must be regularly switched between each program and others.
- This process is known as a *task switch*.
- At each *task switch* the Linux kernel must save the *context* of the CPU.
- The operating system uses the saved context when it switches back to the task the next time it gets some CPU time scheduled to it.

103.5 – Task Scheduling

The total number of slices, when, how often and for how long the CPU is switched is determined by the multitasking algorithm and is handled by a software component within the kernel know as the *task scheduler*.

There are three basic types of task scheduling:

Nonpreemptive: A task must relinquish the CPU before a task switch occurs.

Preemptive: The kernel takes away the CPU from a task without notice.

Realtime: Tasks are prioritised. High priority tasks must complete before a task switch.

103.5 – What is a Process?

The term process is a fundamental abstraction: Two of the more traditional definitions of a process are:

> "A program in execution."
>
> "A single program running in its own virtual address space"

In practice, a process is simply an executable that has been loaded into memory and is either running or ready to run on the system.

103.5 – Process types

Processes under Linux fall into three basic categories:

Interactive Process: An interactive process is a process initiated from (and controlled by) a shell. Interactive processes may be in foreground or background.
(Example: `ls, ls &`)

Batch Process: A batch process is a process that is not associated with a terminal but is submitted to a queue to be executed sequentially.
(Example `slocate` started by `cron`)

Daemon Process: A daemon process is a process that runs in the background until it's required. This kind of processes is usually initiated when Linux boots.
(Example: `inetd`, `lpd`)

103.5 – Elements associated with a process

For each process running on the system, the kernel needs to keep a list of resources used by that process.

Some of these resources include:

- tty association (`tty_struct`)
- file system (eg current directory & open files) (`fs_struct`, `files_struct`)
- memory allocation (`mm_struct`)
- Signals received (`signal_struct`)

103.5 – Process States

At any given point in time, a process is in one of 5 states:

TASK_RUNNING: The process is either executing on the CPU or waiting to be executed.

TASK_INTERRUPTIBLE: The process is sleeping until something becomes true. Raising a hardware interrupt, waiting for a system resource etc are examples of a condition that might wake the process up. If a signal is received by the process (eg KILL -HUP) the process will also be woken up.

TASK_UNINTERRUPTIBLE: Like the previous state except that delivering a signal will not wake the process up.

TASK_STOPPED: Process execution has stopped. A process enters this state after receiving a SIGSTOP signal. A debugger may use this to step through a program.

TASK_ZOMBIE: Process execution has stopped but the kernel has not yet ?cleaned up? the resources allocated to the process.

103.5 – The Process Family Tree

Every process (with the sole exception of the kernel), must be created by another process. The terms *parent*, *child* and *sibling* (or sometimes *father*, *son* and *brother* in a patriarchal sense) are used to describe the relationships between processes.

As an example consider the following line executed from the bash prompt:

```
[andy@Node4] andy]$ ls & df -h &
```

The following relationships are true:

- The `ls` and `df` processes are both siblings to each other.
- The `bash` process (ie the shell) is the parent to both `ls` and `df`.
- The `ls` process has `bash` as its parent.
- The `df` process has `bash` as its parent.

103.5 – The Kernel is at the Top of the Family Tree

- When linux boots, the first thing it does is load the kernel into memory and start executing itself.
- One of the first things it does once execution starts, is to spawn a process called init, which in turn spawns other processes.
- In this sense, the kernel is at the top of the family tree, with only one child process called init.
- Init in turn has many children and probably many grandchildren.

```
Kernel -->
          Init -->
                  all other processes -->
                                         even more processes -->
```

103.5 – Process IDs

In order for the kernel to keep track of all processes and their descendants, a process ID is assigned to every process running on the system. Process IDs are just numbers and run from 0 to 32767. The number 32767 is the largest signed integer available with a sixteen bit word size and is used to maintain backward compatibility with 16 bit architectures.

There are two PIDs (process IDs) that are always the same:

- kernel PID is always 0
- init PID is always 1

Each time a new process is created, a new PID is allocated and is equal to the last PID issued plus one. Once the last PID is reached, the PID wraps back around to zero and the next available PID is used (note that 0 and 1 will never be available). This scheme is a little like the assignment of telephone numbers: When a telephone service is disconnected, rather than just assigning the old telephone number to a new subscriber, the old number remains out of use until all other numbers have been used up. This saves "wrong numbers" to the new subscriber from callers who have not yet realised that the old number is no longer connected to the person they were trying to reach. In a similar vein, the kernel does this to minimise “wrong numbers” from other processes who have not yet worked out that their intended process no longer exists. This is especially true for Interprocess Communication (IPC) which uses the PID to identify a target process.

103.5 – Displaying Process Information

There are two utilities used to display the state of running processes:

- `ps`
- `top`

The ps command is used to display a "snapshot" of all processes running on the system at the time the `ps` command was executed.

The `top` command is used to display a real-time display of all processes running on the system. Top can also be used in interactive mode to `kill` or `renice` (change priority) of a process.

103.5 – Process Monitoring—`ps`

The `ps` command has a huge number of switches. The switches can be subdivided into two main groups:

- Process selection (which processes to display)
- Output control (how and what output should be displayed)

  usage: `ps [options]`

103.5 – `ps` options

The switches that need to be known for the purposes of LPIC are as follows:

`a` Display processes for all users

`txx` Display processes within controlling terminal `txx`

`u` Display user information for the process

`l` Display in long format with detailed information

`s` Display signal information

`m` Display memory information

`x` Display processes without a controlling terminal

`S` Display CPU time and page faults of child processes

`-C cmd` Search for instances of command `cmd`.

`-f` Forest mode shows process family trees.

`-w` Wide format

103.5 – ps options

```
$ ps ?
ERROR: Garbage option.
********* simple selection *********  ********* selection by list *********
-A all processes                      -C by command name
-N negate selection                   -G by real group ID (supports names)
-a all w/ tty except session leaders  -U by real user ID (supports names)
-d all except session leaders         -g by session leader OR by group name
-e all processes                      -p by process ID
T  all processes on this terminal     -s processes in the sessions given
a  all w/ tty, including other users  -t by tty
g  all, even group leaders!           -u by effective user ID (supports names)
r  only running processes             U  processes for specified users
x  processes w/o controlling ttys     t  by tty
*********** output format **********  *********** long options ***********
-o,o user-defined  -f full            --Group --User --pid --cols
-j,j job control   s  signal          --group --user --sid --rows
-O,O preloaded -o  v  virtual memory  --cumulative --format --deselect
-l,l long          u  user-oriented   --sort --tty --forest --version
                   X  registers       --heading --no-heading
                    ********* misc options *********
-V,V show version       L  list format codes  f  ASCII art forest
-m,m show threads       S  children in sum    -y change -l format
-n,N set namelist file  c  true command name  n  numeric WCHAN,UID
-w,w wide output        e  show environment   -H process hierarchy
```

103.5 – `ps` field names & their meanings

`USER` The user who started the process

`PID` The process ID

`%CPU` Shows the cputime / realtime percentage.

`%MEM` The fraction of RSS divided by the total size of RAM

`VSZ` Size of virtual memory used by the process

`RSS` Resident set size (Data & Text segments only) in Kb

`TTY` The TTY associated with this process

`STAT` The current status (DRSTZW¡NL) (details next slide)

`TIME` CPU time in MINS:SECS

`COMMAND` The full command line used to start the process

103.5 – `ps` Status Field

```
$ ps aux
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  1304   72 ?        S    Mar21   0:19 init
```

D uninterruptible sleep (usually IO)

R runnable (on run queue)

S sleeping

T traced or stopped

Z a defunct (“zombie”) process

W has no resident pages

< high-priority process

N low-priority task

L has pages locked into memory (for real-time and custom IO)

103.5 – ps Status Field

```
$ ps aux
USER        PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
root          1  0.0  0.2  1384  516 ?        S    11:43   0:04 init [5]
root          2  0.0  0.0     0    0 ?        SW   11:43   0:00 [keventd]
root          3  0.0  0.0     0    0 ?        SW   11:43   0:00 [kapm-idled]
root          5  0.0  0.0     0    0 ?        SW   11:43   0:00 [kswapd]
root          6  0.0  0.0     0    0 ?        SW   11:43   0:00 [kreclaimd]
root          7  0.0  0.0     0    0 ?        SW   11:43   0:00 [bdflush]
root          8  0.0  0.0     0    0 ?        SW   11:43   0:00 [kupdated]
root          9  0.0  0.0     0    0 ?        SW<  11:43   0:00 [mdrecoveryd]
root        103  0.0  0.0     0    0 ?        SW   11:44   0:00 [kjournald]
root        474  0.0  0.2  1444  620 ?        S    11:44   0:00 syslogd -m 0
root        479  0.0  0.4  2080 1152 ?        S    11:44   0:00 klogd -2
rpc         497  0.0  0.2  1632  708 ?        S    11:44   0:00 portmap
rpcuser     525  0.0  0.3  1624  796 ?        S    11:44   0:00 rpc.statd
ntp         735  0.0  0.8  2088 2080 ?        SL   11:44   0:00 ntpd -U ntp
root        759  0.0  0.3  5784  856 ?        S    11:44   0:00 ypbind
root        763  0.0  0.3  5784  856 ?        S    11:44   0:00 ypbind
.....
andy       1176  0.0  0.5  2620 1508 pts/0    S    11:46   0:00 bash
root       1343  0.0  0.7  3000 1816 tty1     S    15:21   0:00 ssh node10
andy       1664  0.0  0.3  2824  924 pts/1    R    21:52   0:00 ps -aux
```

103.5 – Process Monitoring—`$ pstree`

```
$ pstree
init-+-anacron---run-parts---cfengine
     |-5*[apache-ssl]
     |-atd
     |-bash---startx---xinit-+-X
     |                       `-enlightenment-+-E-Clock.epplet
     |                                       |-E-Cpu.epplet
     |                                       |-Emix.epplet
     |                                       |-Eterm---bash-+-abiword---AbiWord
     |                                       |              `-mozilla-bin---moz
     |                                       |-Eterm---bash---bash
     |                                       |-Eterm---bash
     |                                       |-Eterm---bash---gv---gs
     |                                       |-Eterm---bash---mutt
     |                                       |-Eterm---bash---emacs-+-ispell
     |                                       |                      `-xdvi---gs
     |                                       |-Eterm---bash---pstree
     |                                       `-Eterm---bash---man---pager
     |-cron
     |-gcache
     |-6*[getty]
     |-inetd---nmbd
     |-junkbuster
```

103.5 – `$ pstree` options

Three commonly used options for `pstree`:

-a Show command line arguments.

```
|-xfs -daemon
|-xfstt --port 7101 --daemon --user nobody
`-zope-z2 /usr/sbin/zope-z2
   `-python /usr/sbin/zope-z2
```

-n Sort processes with the same ancestor by PID

-p Show PIDs.

```
init(1)-+-anacron(27095)---run-parts(27755)---cfengine(27765)
        |-apache-ssl(27188)
        |-apache-ssl(27189)
```

103.5 – Process Monitoring—`$ pstree`

103.5 – Process Monitoring—`$ top`

The ”top” command provides a continuously updated, real-time look at process activity, memory and swap file usage plus CPU activity.

It also shows what processes are running and by whom.

103.5 – Process Monitoring—`$ top`

The ”top” command provides a continuously updated, real-time look at process activity, memory and swap file usage plus CPU activity.

It also shows what processes are running and by whom.

- Its primary use is as an administration and system information tool. It provides an extension to the functionality of the ”ps” command.

103.5 – Process Monitoring—`$ top`

The ”top” command provides a continuously updated, real-time look at process activity, memory and swap file usage plus CPU activity.

It also shows what processes are running and by whom.

- Its primary use is as an administration and system information tool. It provides an extension to the functionality of the ”ps” command.
- It makes it easy to find an errand process and ”kill” that process. It also has an interactive interface whereby options can be passed while the command is actually running. All in all, a very useful tool.

103.5 – `$ top` ↩

```
 9:16am  up 13 days,  8:05,  8 users,  load average: 0.05, 0.05, 0.00
86 processes: 84 sleeping, 1 running, 1 zombie, 0 stopped
CPU states:  2.3% user,  0.7% system,  0.0% nice, 96.8% idle
Mem:  900236K av, 546472K used, 353764K free,  0K shrd,  37552K buff
Swap: 329324K av,  34784K used, 294540K free          190764K cached

  PID USER   PRI  NI  SIZE  RSS SHARE STAT LIB %CPU %MEM  TIME COMMAND
10281 root    16 -10 97952 6452  1584 S <    0  3.9  0.7 56:57 X
12547 geoff  16   0  1728 1728   764 R      0  0.9  0.1  0:01 top
10284 geoff  12   0  3012 2568  1352 S      0  0.7  0.2 50:49 enlight
12173 geoff  10   0  9340 9340  3768 S      0  0.3  1.0  0:11 emacs
12543 geoff   9   0  3328 3328  2072 S      0  0.1  0.3  0:00 Eterm
    1 root    9   0   116   72    52 S      0  0.0  0.0  0:19 init
    2 root    9   0     0    0     0 SW     0  0.0  0.0  0:01 keventd
```

103.5 – top's command line options

Note: dashes not required.

-b Batch mode. Useful for sending output from top to other programs or to a file. Output is plain text.

-d Delay between screen updates. (default 5 seconds)

-i

103.5 – top’s upper screen

```
 9:16am  up 13 days,  8:05,  8 users,  load average: 0.05, 0.05, 0.00
86 processes: 84 sleeping, 1 running, 1 zombie, 0 stopped
CPU states:  2.3% user,  0.7% system,  0.0% nice, 96.8% idle
Mem:  900236K av, 546472K used, 353764K free,  0K shrd,  37552K buff
Swap: 329324K av,  34784K used, 294540K free          190764K cached
```

- The current system time:
- The ”up time” of the system:
- How many users are logged in.
- The ”load average” : the average number of processes ready to run over the last 1,5 and 15 minutes
- ”CPU States” shows the percentage of CPU time spent in usermode, system mode and at idle.
- ”MEM” shows a complete set of statistics on current memory usage.
- ”SWAP” gives us the same details as ”MEM” but for the swap space.

103.5 – top's lower screen

```
  PID USER   PRI  NI  SIZE  RSS SHARE STAT LIB %CPU %MEM  TIME COMMAND
10281 root    16 -10 97952 6452  1584 S <     0  3.9  0.7 56:57 X
12547 geoff  16   0  1728 1728   764 R       0  0.9  0.1  0:01 top
```

PID The process ID of each task.

USER The user name of the task's owner.

PRI The priority of the task.

NI The nice value of the task. Negative nice values are higher priority.

SIZE The size of the task's code plus data plus stack space, in kilobytes, is shown here.

RSS The total amount of physical memory used by the task,in kilobytes, is shown here. For ELF processes used library pages are counted here, for a.out processes not.

SHARE The amount of shared memory used by the task is shown in this column.

ctd. . .

103.5 – top's lower screen

```
  PID USER   PRI  NI  SIZE  RSS SHARE STAT LIB %CPU %MEM  TIME COMMAND
10281 root    16 -10 97952 6452  1584 S <    0  3.9  0.7 56:57 X
12547 geoff  16   0  1728 1728   764 R      0  0.9  0.1  0:01 top
```

STAT The state of the task is shown here.

The state is either

S sleeping

D uninterruptible sleep

R running

Z zombies

T stopped or trace

These states are modified by trailing < for a process with negative nice value, N for a process with positive nice value, W for a swapped out process (this does not work correctly for kernel processes).

%CPU The task's share of the CPU time since the last screen update, expressed as a percentage of total CPU time per processor.

%MEM The task's share of the physical memory.

103.5 – `top`: selected interactive commands

ˆL Redraw the screen

f|F Add and remove fields

h|? Displays a help screen

S Toggle cumulative mode

I Toggle between Irix and Solaris views (SMP-only)

k Kill a task (with any signal)

r Renice a task

T Sort by time / cumulative time

s Set the delay in seconds between updates

q Quit

103.5 – `top`'s interactive commands

space Update display

ˆL Redraw the screen

f|F Add and remove fields

o|O Change order of displayed fields

h|? Displays a help screen

S Toggle cumulative mode

i Toggle display of idle processes

I Toggle between Irix and Solaris views (SMP-only)

c Toggle display of command name/line

l Toggle display of load average

m Toggle display of memory information

t Toggle display of summary information

k Kill a task (with any signal)

r Renice a task

N Sort by pid (Numerically)

A Sort by age

P Sort by CPU usage

M Sort by resident memory usage

T Sort by time / cumulative time

u Show only a specific user

n|**#** Set the number of process to show

s Set the delay in seconds between updates

W Write configuration file /.toprc

q Quit

103.5 – ~/.toprc

```
$ cat toprc ↩
AbcDgHIjklMnoTP|qrsuzyV{EFWx
2
```

103.5 – Killing Processes

103.5 – Job Control

There are three commands and a pretzel used for job control.

- `jobs`
- `fg`
- `bg`
- `&`

They are bash built-ins:

```
$ type jobs fg bg ←↩
jobs is a shell builtin
fg is a shell builtin
bg is a shell builtin
```

For more information, see the Job Control section of man bash.

103.5 – & — Direct the shell to execute a command in the background.

Example:

```
$ xeyes ←
```

Notice the `xeyes` process is started in the foreground and you have no prompt. The user is locked out of further interaction with the shell until a process is stopped, terminated or completed.

Now start the xeyes process in the background.

```
$ xeyes & ←
[1] 1650
$
```

Two numbers are listed and the prompt is now also displayed waiting for another command.

103.5 – Job Control

```
$ xeyes & ↩
[1] 1650
$
```

- The [1] is the programs job id, a unique number for the shell starting from 1.
- The 1650 is the process id (pid), which identifies the process across the entire system.
- Either of these numbers can be used to interact with the program through bash.

103.5 – Background Processing

The best candidates for background processing are programs that do not require user input, as these programs will keep on waiting until input is provided.

Programs that send their results to standard output (The screen), will do so even if running in the background. If the user is performing another operation, the results may be difficult to interpret. The output from these processes can be redirected to a file.

```
$ wc bigfile > bigfile.wc & ↩
[1] 1654
$
```

103.5 – The `jobs` command

$ jobs ↩ :

Lists all commands stopped, or running in the background.

Options :

`-l` List pid

Example :

Start some processes in the background and suspend a foreground process.

```
$ jobs ↩
[1]+  Stopped                    less job_control.txt
[2]-  Running                    xeyes &
$
```

103.5 – The `fg` command

$ fg ↩ :

Shell built-in used to force a suspended or background process to continue running in the foreground.

Example :

Use the 'jobs' command to find job id.

```
$ jobs ↩
[1]+  Stopped                    less job_control.txt
[2]-  Running                    xeyes &
$
```

Use `fg` to bring xeyes to foreground.

```
$ fg 2 ↩
xeyes
```

A `%` used with the job id is equivalent to `fg 2`.

```
$ %2 ↩
xeyes
```

103.5 – The `fg` command

A job can also be referred to by a string that uniquely identifies the beginning of the command line used to start a job. A '%' can also be used with a unique string.

```
$ fg x ↩
xeyes
```

or

```
$ %x ↩
xeyes
```

If `fg` is issued without any argument, the job with the '+' in the job list is brought to the foreground.

```
$ fg ↩
xeyes
```

103.5 – The `bg` command

`$ bg` ←↩ :

Used to force a suspended process to continue running in the background.

Example :

Use the 'jobs' command to find job id.

```
$ jobs ←↩
[1]-  Stopped              find -name myfile >myfile.found  (wd: /)
[2]+  Stopped              less job_control.txt
[3]   Running              xeyes &
$
```

Job 1 shows the 'find' command was started in the foreground and then suspended.To start 'find' in the background, use the 'bg' command or '%'.

Example :

`$ bg 1 ←↩` or `$ bg f ←↩` or `$ %1 & ←↩` or `$ %f & ←↩`

103.6 – Modify process execution priorities [2]

103.6 – Objective

Candidate should be able to manage process execution priorities. Tasks include running a program with higher or lower priority, determining the priority of a process and changing the priority of a running process.

103.7 – Search text files using regular expressions [3]

103.7 – Objective

The candidate 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.

103.8 – Perform basic file editing operations using vi [2]

103.8 – Objective

Candidate must be able to edit text files using vi. This objective includes vi navigation, basic vi nodes, inserting, editing, deleting, copying, and finding text.

104 – (2.4) Devices, Linux Filesystems & FHS [21]

1. Create partitions and filesystems [3]
2. Maintain the integrity of filesystems [5]
3. Control mounting and unmounting filesystems [3]
4. Managing disk quota [1]
5. Use file permissions to control access to files [3]
6. Manage file ownership [2]
7. Create and change hard and symbolic links [2]
8. Find system files and place files in the correct location [2]

104.1 – Create partitions and filesystems [3]

104.1 – Objective

Candidates should be able to configure disk partitions and then create filesystems on media such as hard disks. This objective includes using various mkfs commands to set up partitions to various filesystems, including ext2, ext3, reiserfs, vfat, and xfs.

104.1 – Key files, terms, and utilities include:

`fdisk` `mkfs`

104.1 – Resources

TBA

104.1 – devices—major and minor numbers

104.1 – devices—major and minor numbers

- All devices have a major and a minor number, e.g. floppy disks have the major number of 2 and the first floppy disk has a minor number of 0.

104.1 – devices—major and minor numbers

- All devices have a major and a minor number,
  e.g. floppy disks have the major number of 2
  and the first floppy disk has a minor number of 0.
- All devices are represented by device nodes in `/dev/`

104.1 – devices—major and minor numbers

- All devices have a major and a minor number,
  e.g. floppy disks have the major number of 2
  and the first floppy disk has a minor number of 0.
- All devices are represented by device nodes in `/dev/`
- Device nodes may be made thus:

```
# mknod -b /dev/fd0 2 0 ↩
```

104.1 – devices—major and minor numbers

- All devices have a major and a minor number,
  e.g. floppy disks have the major number of 2
  and the first floppy disk has a minor number of 0.
- All devices are represented by device nodes in `/dev/`
- Device nodes may be made thus:

```
# mknod -b /dev/fd0 2 0 ↩
```

- Should you inadvertently loose your device nodes a new set can be generated by running the utility `/dev/MAKEDEV`

104.1 – devices—major and minor numbers

- All devices have a major and a minor number,
  e.g. floppy disks have the major number of 2
  and the first floppy disk has a minor number of 0.
- All devices are represented by device nodes in `/dev/`
- Device nodes may be made thus:

  ```
  # mknod -b /dev/fd0 2 0 ↩
  ```

- Should you inadvertently loose your device nodes a new set can be generated by running the utility `/dev/MAKEDEV`
- Access to devices may be modified by changing the permissions to the device node. However, there is usually a better way.

104.1 – block devices

Block devices are generally random access rotating memory but may use other technologies.

- ide hard disk—`/dev/hdc`
- scsi hard disk—`/dev/sdb`
- scsi cdrom—`/dev/sr2`
- ide zip disk—`/dev/hdd`
- floppy disk—`/dev/fd5`
- ide cdrom disk—`/dev/hdc`
- scsi cd-rw disk—`/dev/sr4`

104.1 – disk partitions

On i386 systems disks may be sliced up into to 15 partitions.

104.1 – disk partitions

On i386 systems disks may be sliced up into to 15 partitions.

- The disk must have at least 1 primary partition.

  `/dev/hda1`

104.1 – disk partitions

On i386 systems disks may be sliced up into to 15 partitions.

- The disk must have at least 1 primary partition.

  ```
  /dev/hda1
  ```

- There may be up to 4 primary partitions.

  ```
  /dev/hda1  (primary)
  /dev/hda2  (primary)
  /dev/hda3  (primary)
  /dev/hda4  (primary)
  ```

104.1 – disk partitions

On i386 systems disks may be sliced up into to 15 partitions.

- The disk must have at least 1 primary partition.

  ```
  /dev/hda1
  ```

- There may be up to 4 primary partitions.

  ```
  /dev/hda1  (primary)
  /dev/hda2  (primary)
  /dev/hda3  (primary)
  /dev/hda4  (primary)
  ```

- One of the primary partitions may be made into an extended partition.

104.1 – disk partitions

On i386 systems disks may be sliced up into to 15 partitions.

- The disk must have at least 1 primary partition.

  ```
  /dev/hda1
  ```

- There may be up to 4 primary partitions.

  ```
  /dev/hda1  (primary)
  /dev/hda2  (primary)
  /dev/hda3  (primary)
  /dev/hda4  (primary)
  ```

- One of the primary partitions may be made into an extended partition.
- The one extended partition must hold between 1 and 12 logical partitions.

  ```
  /dev/hda1  (primary)
  /dev/hda2  (extended)
  /dev/hda5  (logical)
  /dev/hda6  (logical)
  ```

104.1 – `fdisk` disk partitioning tool

```
# fdisk ↩

Usage: fdisk [-l] [-b SSZ] [-u] device
E.g.: fdisk /dev/hda  (for the first IDE disk)
  or: fdisk /dev/sdc  (for the third SCSI disk)
  or: fdisk /dev/eda  (for the first PS/2 ESDI drive)
  or: fdisk /dev/rd/c0d0   (for RAID devices)
  or: fdisk /dev/ida/c0d0  (for RAID devices)
  ...
#
```

104.1 – `fdisk` disk partitioning tool

```
# fdisk ↩

Usage: fdisk [-l] [-b SSZ] [-u] device
E.g.: fdisk /dev/hda  (for the first IDE disk)
  or: fdisk /dev/sdc  (for the third SCSI disk)
  or: fdisk /dev/eda  (for the first PS/2 ESDI drive)
  or: fdisk /dev/rd/c0d0   (for RAID devices)
  or: fdisk /dev/ida/c0d0  (for RAID devices)
  ...
#
```

-b sectorsize old kernels only

-l list the partition table only

-u sectors not cylinders

-s print size in blocks

104.1 – fdisk

```
# fdisk /dev/hda ↩

The number of cylinders for this disk is set to 3648.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): _
```

104.1 – `fdisk` menu

```
Command (m for help): m ↩
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
```

104.1 – `fdisk` print the partition table

```
Command (m for help): p ↩

Disk /dev/hda: 255 heads, 63 sectors, 3648 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot   Start      End     Blocks    Id  System
/dev/hda1   *        1      768   6168928+    c  Win95 FAT32 (LBA)
/dev/hda2          769     3648  23133600     5  Extended
/dev/hda5          769      780     96358+   83  Linux
/dev/hda6          781      904    995998+   82  Linux swap
/dev/hda7          905     2363  11719386    83  Linux
/dev/hda8         2364     2485    979933+   83  Linux

Command (m for help): _
```

104.1 – `fdisk` exercise on a floppy

Warning it makes no sense to use `fdisk` on a floppy.

```
# fdisk /dev/fd0

Command (m for help): p

Disk /dev/fd0: 2 heads, 18 sectors, 80 cylinders
Units = cylinders of 36 * 512 bytes

    Device Boot    Start       End    Blocks   Id  System
/dev/fd0p1             1        20       351    1  FAT12
/dev/fd0p2            21        25        90   83  Linux
/dev/fd0p3            26        80       990    5  Extended
/dev/fd0p5            26        30        81   83  Linux
/dev/fd0p6            31        70       711   83  Linux
/dev/fd0p7            71        80       171   83  Linux
```

104.1 – using `sfdisk` on a floppy

```
# sfdisk -f /dev/fd0
Disk /dev/fd0: 80 cylinders, 2 heads, 18 sectors/track
Old situation:
Units = cylinders of 18432 bytes, blocks of 1024 bytes,counting from 0

   Device Boot Start     End   #cyls   #blocks   Id  System
/dev/fd0p1          0+     19      20-       351    1  FAT12
/dev/fd0p2         20      24       5         90   83  Linux
/dev/fd0p3         25      79      55        990    5  Extended
/dev/fd0p4          0       -       0          0    0  Empty
/dev/fd0p5         25+     29       5-        81   83  Linux
/dev/fd0p6         30+     69      40-       711   83  Linux
/dev/fd0p7         70+     79      10-       171   83  Linux
Input in the following format; absent fields get a default value.
<start> <size> <type [E,S,L,X,hex]> <bootable [-,*]> <c,h,s> <c,h,s>
Usually you only need to specify <start>, <size> (and perhaps <type>).

/dev/fd0p1 :
```

104.1 – using **sfdisk** on hda

```
# sfdisk -f /dev/hda ↩
Disk /dev/hda: 3648 cylinders, 255 heads, 63 sectors/track
Old situation:
Units = cylinders 8225280 bytes, blocks of 1024 bytes, counting from 0

   Device Boot Start     End   #cyls   #blocks   Id  System
/dev/hda1   *      0+    767     768-  6168928+   c  Win95 FAT32 (LBA)
/dev/hda2        768    3647    2880  23133600    5  Extended
/dev/hda3          0       -       0         0    0  Empty
/dev/hda4          0       -       0         0    0  Empty
/dev/hda5        768+    779      12-    96358+  83  Linux
/dev/hda6        780+    903     124-   995998+  82  Linux swap
/dev/hda7        904+   2362    1459- 11719386   83  Linux
/dev/hda8       2363+   2484     122-   979933+  83  Linux
Input in the following format; absent fields get a default value.
<start> <size> <type [E,S,L,X,hex]> <bootable [-,*]> <c,h,s> <c,h,s>
Usually you only need to specify <start>, <size> (and perhaps <type>).
/dev/hda1 : _
```

104.1 – GNU parted—partition table manipulator

```
# parted /dev/hda ↩
  (parted) help ↩
  check MINOR                        do a simple check on the filesystem
  cp [FROM-DEVICE] FROM-MINOR TO-MINOR copy fs to another partition
  help [COMMAND]                     prints general help, help on COMMAND
  mklabel LABEL-TYPE                 create a new disklabel (part. table)
  mkfs MINOR FS-TYPE                 make a fs FS-TYPE on partititon MINOR
  mkpart PART-TYPE [FS-TYPE] START END  make a partition
  mkpartfs PART-TYPE FS-TYPE START END  make a partition with a fs
  move MINOR START [END]               move partition MINOR
  name MINOR NAME                    name partition MINOR NAME
  print                              display the partition table
  quit                               exit program
  resize MINOR START END             resize filesystem on partition MINOR
  rm MINOR                           delete partition MINOR
  select DEVICE                      choose the device to edit
  set MINOR FLAG STATE               change a flag on partition MINOR
(parted)
```

104.1 – GNU parted—partition table manipulator

This utility has similar uses as the commercial utility ”Partition Magic”. It can resize partitions.

```
(parted) p
Disk geometry for /dev/hda: 0.000-28615.781 megabytes
Disk label type: msdos
Minor    Start        End     Type      Filesystem  Flags
1          0.031   6024.375  primary   FAT         boot, lba
2       6024.375  28615.781  extended
5       6024.406   6118.505  logical   ext2
6       6118.537   7091.191  logical   linux-swap
7       7091.222  18535.935  logical   ext2
8      18535.966  19492.932  logical
(parted)
```

104.1 – `fips.exe`—resize FAT partitions

104.1 – File System Types

ext2 Linux standard file system (type 83)

ext3 New journaling file system

reiserfs Journaling file system

qnx4 qnx4 file system

minix Used by Minux and where space is at a premium

iso9660 Standard file system on CDROMs

msdos Microsoft FAT16 file system

vfat Microsoft FAT32 file system

ntfs Microsoft NT file system

hfs Apple file system

nfs Network File System

proc Linux process file system

swap Linux swap file system (type 82)

smbfs Samba

104.1 – `mkfs`—make filesystem

One the a partition is made and has been given a type it must

104.1 – `mkswap`

104.1 –

104.1 –

104.1 –

104.2 – Maintain the integrity of filesystems [5]

104.2 – Objective

Candidates should be able to verify the integrity of filesystems, monitor free space and inodes, and repair simple filesystem problems. This objective includes the commands required to maintain a standard filesystem, as well as the extra data associated with a journaling filesystem.

104.2 – Key files, terms, and utilities include:

```
du   df   fsck  e2fsck   mke2fs
debugfs   dumpe2fs   tune2fs
```

104.2 – `du`—disk usage

104.2 – `df`—

104.2 – `fsck`—check and repair a Linux file system

- fsck is used to check and optionally repair a one or more Linux file systems.
- filesys can be a device name (e.g. /dev/hdc1, /dev/sdb2), a mount point (e.g. /, /usr, /home), or an ext2 label or UUID specifier (e.g. UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root).
- The fsck program will try to run filesystems on different physical disk drives in parallel to reduce total amount time to check all of the filesystems.
- Options:
  - -A Walk through the /etc/fstab file and try to check all file systems in one run. This option is typically used from the /etc/rc system initialization file, instead of multiple commands for checking a single file system.
  - -a Automatically repair the file system without any questions (use this option with caution).

104.2 – `e2fsck`—check a Linux second extended file system

e2fsck is used to check a Linux second extended file system (e2fs). E2fsck also supports ext2 filesystems containing a journal, which are also sometimes known as ext3 filesystems.

104.2 – `mke2fs`—create a Linux second extended file system

mke2fs is used to create a Linux second extended file system on a device (usually a disk partition). device is the special file corresponding to the device (e.g /dev/hdXX). blocks-count is the number of blocks on the device. If omitted, mke2fs automagically figures the file system size.

```
mke2fs [  -c | -l filename ] [ -b block-size ] [ -f fragment-
    size ] [ -i bytes-per-inode ] [ -j ]  [  -J  journal-
    options  ]  [  -N number-of-inodes ] [ -n ] [ -m reserved-
    blocks-percentage ] [ -o creator-os ] [  -O  feature[,...]
    ] [ -q ] [ -r fs-revision-level ] [ -R raid-options ] [ -v
    ] [ -F ] [ -L volume-label ] [ -M last-mounted-directory ]
    [ -S ] [ -T filesystem-type ] [ -V ] device [ blocks-count]
```

104.2 – `debugfs`—ext2 file system debugger

The debugfs program is a file system debugger. It can be used to examine and change the state of an ext2 file system.

debugfs is an interactive debugger. It understands a number of commands.

cat filespec Dump the contents of the inode filespec to stdout.

cd filespec Change the current working directory to filespec.

chroot filespec Change the root directory to be the directory file spec.

close Close the currently open file system.

?

?

?

?

?

104.2 – `dumpe2fs`—dump filesystem information

dumpe2fs prints the super block and blocks group information for the filesystem present on device.

104.2 – `tune2fs`—adjust tunable filesystem parameters on ext2 fs

tune2fs adjusts tunable filesystem parameters on a Linux second extended filesystem.

```
tune2fs [ -l ] [ -c max-mount-counts ] [ -e  errors-behavior  ]
   [ -i interval-between-checks ] [ -j ] [ -J journal
     options ] [ -m reserved-blocks-percentage ] [ -r reserved
     blocks-count  ]  [ -s sparse-super-flag ] [ -u user ] [ -g
     group ] [ -C mount-count ] [ -L volume-name ] [  -M  last
     mounted-directory  ]  [ -O [^]feature[,...]  ] [ -U UUID ]
     device
```

104.3 – Control mounting and unmounting filesystems [3]

104.3 – Objective

Candidates should be able to configure the mounting of a filesystem. This objective includes the ability to manually mount and unmount filesystems, configure filesystem mounting on bootup, and configure user mountable removeable filesystems such as tape drives, floppies, and CDs.

104.3 – Key files, terms, and utilities

`/etc/fstab` `mount` `umount`

104.3 – Resources of Interest

Appropriate man pages.

104.3 – Linux file system

Unlike MSDOS based operating systems that use named volumes with separate file systems (C: drive, A: drive, etc.) Linux and other unicies have a unified filesystem with volumes "grafted" in to a single tree at various mount points. The mount points are arbitrary. Typical mount points for removable media are as follows:

- Floppy disk: device /dev/fd0 mounted at `/mnt/floppy`
- ZIP drive: mounted at `/ZIP`
- CDROM disk: device `/dev/hdc` mounted at `/mnt/cdrom`
- CD Writer: device `/dev/sd2` mounted at `/burner`
- Network drive: mounted at `/mnt/nfs/database`
- NT Server network drive: `/mnt/samba-vol`

104.3 – First Level Directories

```
$ tree -L 1 ↩
/
|-- backups
|-- bin
|-- boot
|-- boot~
|-- dev
|-- dvd
|-- etc
|-- home
|-- incomming -> /usr/local/incomming/
|-- lib
|-- lost+found
|-- mnt
    |--floppy
    `--cdrom
|-- proc
|-- root
|-- sbin
|-- tmp
|-- usr
|-- var
`-- zip
```

104.3 – The `mount` Command

```
# mount -t type /dev/device_node /mount_point ↩
```

104.3 – The `mount` Command

```
# mount -t type /dev/device_node /mount_point ↩
```

type The type of file system e.g. vfat or ext2

device_node The device node for the appropriate block device e.g. `/dev/fd0`

mount_point The existing directory where the file system will be come accessible. e.g. `/mnt/floppy`

104.3 – The mount Command

```
# mount -t type /dev/device_node /mount_point ↩
```

type The type of file system e.g. vfat or ext2

device_node The device node for the appropriate block device e.g. /dev/fd0

mount_point The existing directory where the file system will be come accessible. e.g. /mnt/floppy

104.3 – mount example

```
# touch /flb/im_gona_be_hidden ↩
# ls /flb ↩
im_gona_be_hidden
# mount -t msdos /dev/fd1 /flb ↩
mount: block device /dev/fd1 is write-protected, read-only
# ls /flb ↩
ecp5-1.c  ecp5-4.c  ecp5-6.c
#
```

104.3 – Filesystem Types

```
# mount -t type ...
```

ext2 Linux standard file system

ext3 New journaling file system

reiserfs Journaling file system

qnx4 qnx I presume

minix Used by Minux and where space is a premium

iso9660 Standard file system on CDROMs

msdos Microsoft FAT16 file system

vfat Microsoft FAT32 file system

ntfs Microsoft NT file system

hfs Apple file system

nfs Network File System

smbfs Samba

104.3 – configuration: `/etc/fstab`

The file `/etc/fstab` is a table of static mount information. The mount command references this table. Edit this file to reflect your system.

```
$ cat /etc/fstab ↩
# <file system><mount pt><type>  <options>                <dump> <pass>

/dev/sda1   /            ext2     defaults,errors=remount-ro    0 1
/dev/sda2   /tmp         ext2     rw                            0 2
/dev/sda3   /var         ext2     rw                            0 2
/dev/sda4   none         swap     sw                            0 0

/dev/sr5    /writer      iso9660  defaults,ro,user,noauto       0 0
/dev/hdb    /dvd         iso9660  defaults,ro,user,noauto       0 0

/dev/fd0    /mnt/floppy auto      defaults,user,noauto          0 0
```

104.3 – Abreviated `mount` commands

Using entries in the `/etc/fstab` configuration file mount commands may be abbreviated:

- Mount a floppy

  `$ mount /mnt/floppy` ←˒

- Mount the DVD:

  `$ mount /dvd` ←˒

104.3 – Abreviated `mount` commands

Using entries in the `/etc/fstab` configuration file mount commands may be abbreviated:

- Mount a floppy

  `$ mount /mnt/floppy` ↩

- Mount the DVD:

  `$ mount /dvd` ↩

104.3 – Unmounting filesystems `umount`

- Unmount a floppy

  `$ umount /mnt/floppy` ↩

- Unmount the DVD:

  `$ umount /dvd` ↩

104.3 – /etc/mtab

/etc/mtab is a dynamic table of currently mounted file systems.

104.3 – `/etc/mtab`

`/etc/mtab` is a dynamic table of currently mounted file systems.

- This file is secret kernel business.

104.3 – `/etc/mtab`

`/etc/mtab` is a dynamic table of currently mounted file systems.

- This file is secret kernel business.
- Do *not* edit this file.

104.3 – `/etc/mtab`

`/etc/mtab` is a dynamic table of currently mounted file systems.

- This file is secret kernel business.
- Do *not* edit this file.
- Ever.

104.3 – /etc/mtab

/etc/mtab is a dynamic table of currently mounted file systems.

- This file is secret kernel business.
- Do *not* edit this file.
- Ever.

```
$ cat /etc/mtab ↩
/dev/hda6 / ext2 rw 0 0
none /proc proc rw 0 0
none /dev/pts devpts rw,gid=5,mode=620 0 0
/dev/hda1 /mnt/disk vfat rw 0 0
```

104.3 – `/etc/mtab`

`/etc/mtab` is a dynamic table of currently mounted file systems.

- This file is secret kernel business.
- Do *not* edit this file.
- Ever.

```
$ cat /etc/mtab ↩
/dev/hda6 / ext2 rw 0 0
none /proc proc rw 0 0
none /dev/pts devpts rw,gid=5,mode=620 0 0
/dev/hda1 /mnt/disk vfat rw 0 0
```

```
$ mount ↩
/dev/hda6 on / type ext2 (rw)
none on /proc type proc (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda1 on /mnt/disk type vfat (rw)
```

104.3 – Disk usage—`du`

```
$ du ↩
24       ./vmware/win2000
961136   ./vmware/nt4
961164   ./vmware
20       ./.netscape/cache
4        ./.netscape/archive
968164   .
```

104.3 – Disk usage—`du`

```
$ du ↩
24        ./vmware/win2000
961136    ./vmware/nt4
961164    ./vmware
20        ./.netscape/cache
4         ./.netscape/archive
968164    .
```

104.3 – Disk filesystem usage—`df`

```
$ df ↩
Filesystem 1k-blocks      Used Available Use% Mounted on
/dev/hda6    3755460   3041736    522956  86% /
/dev/hda1    2044240    614664   1429576  31% /mnt/disk
```

104.4 – Managing disk quota [1]

104.4 – Objective

Candidates should be able to manage disk quotas for users. This objective includes setting up a disk quota for a filesystem, editing, checking, and generating user quota reports.

104.4 – Key files, terms, and utilities include:

See 104-4.tex

104.5 – Use file permissions to control access to files [3]

104.5 – Objective

Candidates should be able to control file access through permissions. This objective includes access permissions on regular and special files as well as directories. Also included are access modes such as suid, sgid, and the sticky bit, the use of the group field to grant file access to workgroups, the immutable flag, and the default file creation mode.

104.5 – Key files, terms, and utilities include:

`chmod` `umask` `chattr`

104.6 – Manage file ownership [2]

104.6 – Objective

Candidates should be able to control user and group ownership of files. This objective includes the ability to change the user and group owner of a file as well as the default group owner for new files.

104.6 – Key files, terms, and utilities include:

`chmod` `chown` `chgrp`

104.7 – Create and change hard and symbolic links [2]

104.7 – Objective

Candidates should be able to create and manage hard and symbolic links to a file. This objective includes the ability to create and identify links, copy files through links, and use linked files to support system administration tasks.

104.7 – Key files, terms, and utilities include:

`ln`

104.7 – `ln` link

A *link* is a pseudofile that creates a shortcut to the original file located elsewhere on the filesystem.

Symbolic links

Hard links

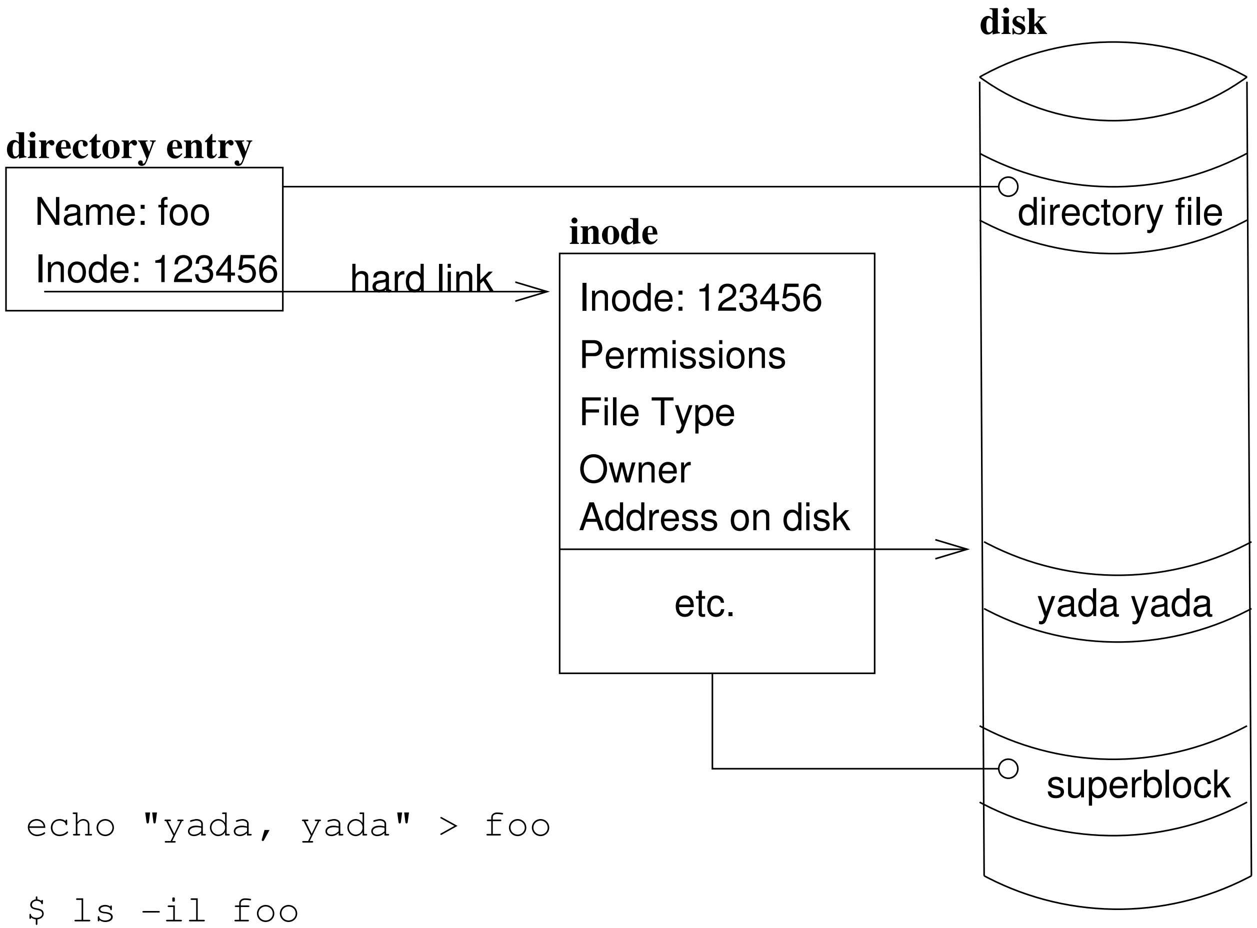
104.7 – Linux files and *inodes*

Creating a file with a command such as:

```
$ cat -n "Hello" > foo
```

1. An *inode* number in the superblock is allocated to the file
2. The files inode is populated with information
3. A directory entry (Hard link) is made in a directory file
4. The file's data is written to a place on the disk pointed to by the *inode*

104.7 – Linux files and *inodes*




```
echo "yada, yada" > foo

$ ls -il foo

123456 -rw-r--r--  1 fred user 10 Mar 20 12:35 foo
```

104.7 – The *inode* information

Some of the information contained in a file's inode can be displayed with the `ls` command:

```
$ ls -il foo
2723514 -rw-r--r--   2 geoff geoff   16 Mar 22 09:38 /tmp/foo
```

A more complete view of inode information may be had with `stat`:

```
$ stat foo
  File: "/tmp/foo"
  Size: 5               Blocks: 2        IO Block: 4096   Regular File
Device: 802h/2050d      Inode: 2723514  Links: 1
Access: (0644/-rw-r--r--) Uid: ( 1000/geoffrey) Gid: ( 1000/geoffrey)
Access: Fri Mar 22 08:25:00 2002
Modify: Fri Mar 22 09:38:19 2002
Change: Fri Mar 22 09:52:26 2002
```

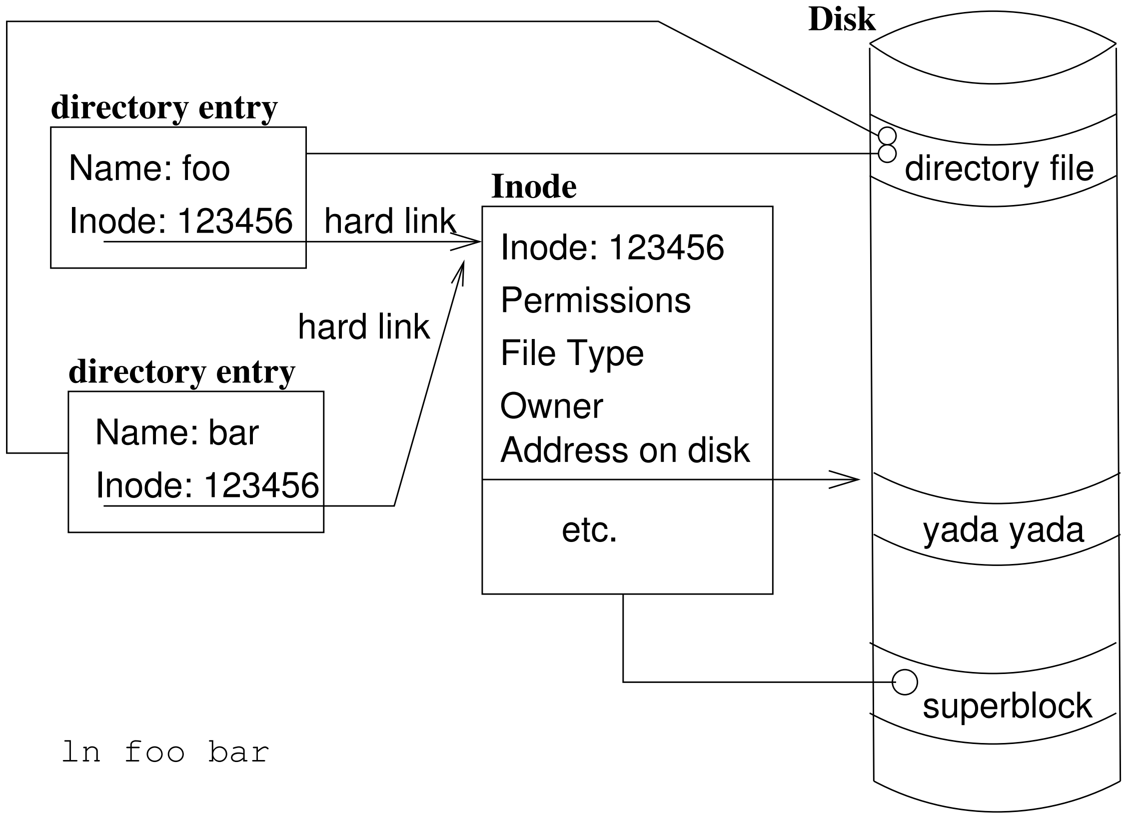
104.7 – *Hard links* are directory entries

A file may have one or more *hard links* to it. Additional *hard links* are made with the `ln` command:

```
$ ln foo bar
$ ls -il foo bar
2723514 -rw-r--r--    2 geoff geoff    5 Mar 22 09:38 bar
2723514 -rw-r--r--    2 geoff geoff    5 Mar 22 09:38 foo
```

- there is only one file on the disk
- it has one inode
- it has two names (hard links)

104.7 – Hard links are directory entries




```
ln foo bar

$ ls -il

123456 -rw-r--r--   2 fred user 10 Mar 20 12:35 bar

123456 -rw-r--r--   2 fred user 10 Mar 20 12:35 foo
```

104.7 – foo a.k.a. bar

```
$ chmod 640 foo
$ ls -il foo bar
2723514 -rw-r-----    2 geoff geoff    5 Mar 22 09:38 bar
2723514 -rw-r-----    2 geoff geoff    5 Mar 22 09:38 foo

$ echo " there" >> bar
$ head foo bar
==> foo <==
Hello there
==> bar <==
Hello there

$ ls -il foo bar
2723514 -rw-r-----    12 geoff geoff    5 Mar 22 09:38 bar
2723514 -rw-r-----    12 geoff geoff    5 Mar 22 09:38 foo
```

104.7 – Hard link constraints

- Hard links are confined within the volume:

```
$ ln foo ~/fred
ln: creating hard link ‘/home/geoffrey/fred’ to ‘foo’:
Invalid cross-device link
```

- Directories may not have multiple hard links:

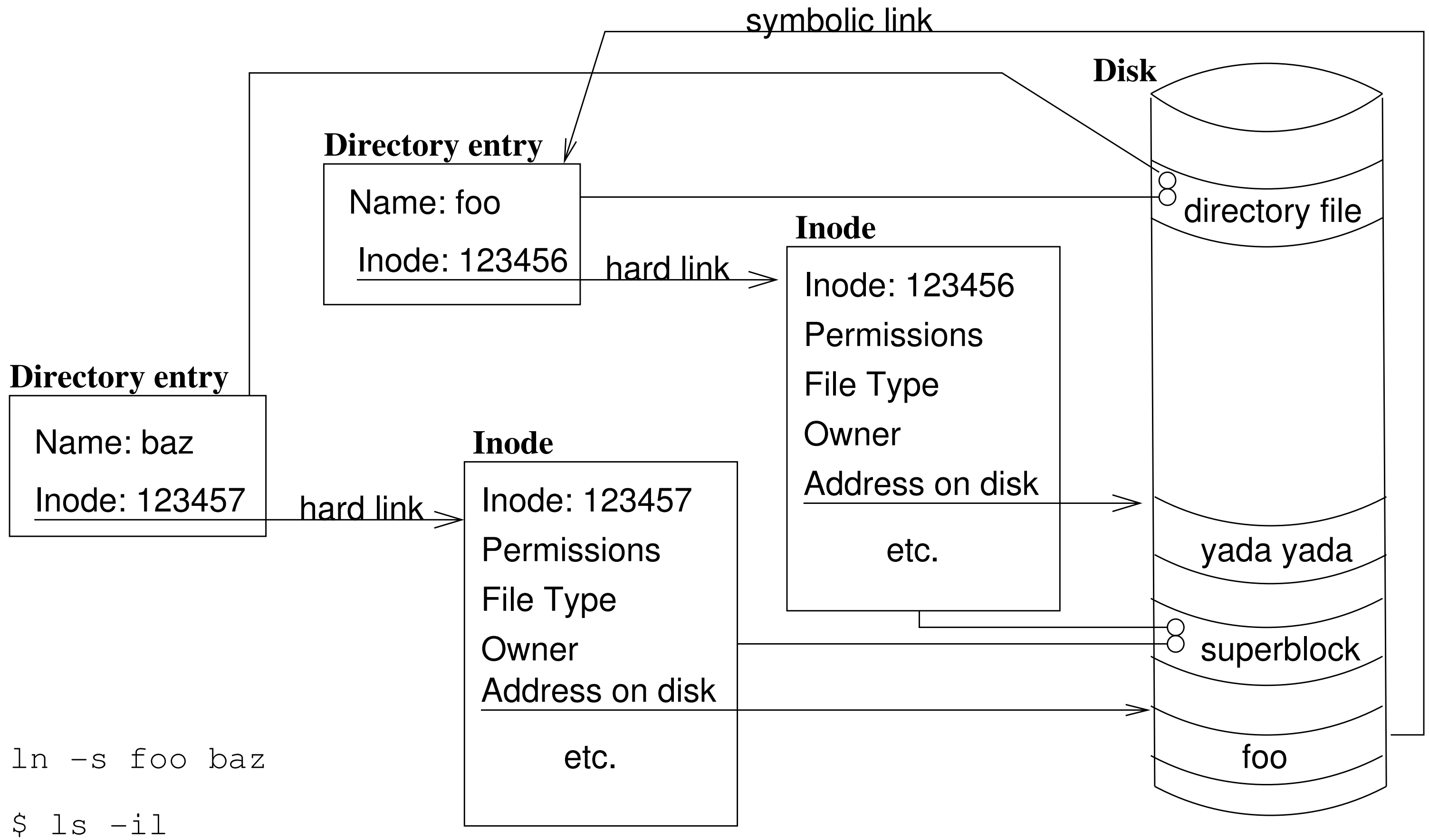
```
$ ln /tmp doh
ln: ‘/tmp’: hard link not allowed for directory
```

- `ln` declines to clobber unless forced:

```
$ ln foo bar
ln: ‘bar’: File exists
$ ln -f foo bar
$
```

104.7 – Symbolic links

104.7 – A *symbolic link* is a file that points to another




```
ln -s foo baz

$ ls -il

123457 lrwxrwxrwx   2 fred user 3  Mar 20 12:35 baz -> foo

123456 -rw-r--r--   2 fred user 10 Mar 20 12:35 foo
```

104.7 – Linux files and *inodes*

104.7 – Linux files and *inodes*

104.7 – Linux files and *inodes*

104.8 – Find system files, place files in the correct location [2]

104.8 – Objective

Candidates should be thoroughly familiar with the Filesystem Hierarchy Standard, including typical file locations and directory classifications. This objective includes the ability to find files and commands on a Linux system.

104.8 – Key files, terms, and utilities include:

`find` `locate` `slocate` `updatedb`
`whereis` `which` `/etc/updatedb.conf`

104.8 – Subsections of 104.8

- Data Types
- The filesystem layout
- find
- locate, slocate, updatedb and /etc/updatedb.conf
- whereis, which and apropos

104.8 – Data Types—FHS Classification

Data sharing scope of data used in networked environments:

104.8 – Data Types—FHS Classification

Data sharing scope of data used in networked environments:

Shareable Data used on multiple hosts across a network—data files, many executables and common configuration files.

104.8 – Data Types—FHS Classification

Data sharing scope of data used in networked environments:

- **Shareable** Data used on multiple hosts across a network—data files, many executables and common configuration files.
- **Non-shareable** Files relevant to a paticular host—system logs, etc.

104.8 – Data Types—FHS Classification

Data sharing scope of data used in networked environments:

- **Shareable** Data used on multiple hosts across a network—data files, many executables and common configuration files.
- **Non-shareable** Files relevant to a paticular host—system logs, etc.

Data modification How data changes.

104.8 – Data Types—FHS Classification

Data sharing scope of data used in networked environments:

- **Shareable** Data used on multiple hosts across a network—data files, many executables and common configuration files.
- **Non-shareable** Files relevant to a paticular host—system logs, etc.

Data modification How data changes.

- **Variable** Data that changes normal frequent processes—log files, user data etc.

104.8 – Data Types—FHS Classification

Data sharing scope of data used in networked environments:

- **Shareable** Data used on multiple hosts across a network—data files, many executables and common configuration files.
- **Non-shareable** Files relevant to a paticular host—system logs, etc.

Data modification How data changes.

- **Variable** Data that changes normal frequent processes—log files, user data etc.
- **Static** Data that does not normally change day to day—binary programs and most of /etc

104.8 – FHS Data Types

	Shareable	Non-shareable
Static	`/usr`	`/etc`
	`/usr/local`	`/boot`
Variable	`/vavr/mail`	`/var/log`
	`/home`	`/proc`

104.8 – The Linux Filesystem

```
$ ./tree -L 2 -d /
/
|-- bin
|-- boot
|   |-- grub
|   `-- lost+found
|-- dev
|   `-- video
|-- etc
|   |-- CORBA
|   |-- X11
|   |-- alchemist
|   |-- cron.d
|   |-- cron.daily
|   |-- cron.hourly
|   |-- cron.monthly
```

104.8 – The Linux Filesystem

```
|-- home
|    |-- geoffrey
|    `-- lost+found
|-- initrd
|-- lib
|    |-- i686
|    |-- iptables
|    |-- kbd
|    |-- modules
|    `-- security
|-- lost+found
|-- misc
|-- mnt
|    |-- cdrom
|    |-- floppy
|    `-- hda8
```

104.8 – The Linux Filesystem

```
|-- opt
|    '-- prenticehall_cyberclassroom
|-- proc
|    |-- 1

|    |-- scsi
|    |-- sys

|-- root
|-- sbin
|-- tmp
|-- usr
|    |-- X11R6
|    |-- apps
|    |-- bin
```

104.8 – The Linux Filesystem

```
|-- var
|    |-- arpwatch
|    |-- cache
|    |-- db

|    |-- lock
|    |-- log
|    |-- mail -> spool/mail

‘-- win
     |-- Adobe Albums
     |-- Dell
     |-- My Documents

|
‘-- zip
```

104.8 – The First Level Directories

`/boot` Files for the boot loader.

104.8 – The First Level Directories

`/boot` Files for the boot loader.

`/dev` Device files for accessing devices.

104.8 – The First Level Directories

/boot Files for the boot loader.

/dev Device files for accessing devices.

/etc Configuration files.

104.8 – The First Level Directories

`/boot` Files for the boot loader.

`/dev` Device files for accessing devices.

`/etc` Configuration files.

`/home` Home directories for system users.

104.8 – The First Level Directories

`/boot` Files for the boot loader.

`/dev` Device files for accessing devices.

`/etc` Configuration files.

`/home` Home directories for system users.

`/lib` Shared libraries and kernel modules.

104.8 – The First Level Directories

`/boot` Files for the boot loader.

`/dev` Device files for accessing devices.

`/etc` Configuration files.

`/home` Home directories for system users.

`/lib` Shared libraries and kernel modules.

`/mnt` Mount points for temporary partitions.

104.8 – The First Level Directories

`/opt` Location for 3rd party applications.

104.8 – The First Level Directories

`/opt` Location for 3rd party applications.

`/proc` Virtual filesystem created by the kernel.

104.8 – The First Level Directories

`/opt` Location for 3rd party applications.

`/proc` Virtual filesystem created by the kernel.

`/root` Home for the superuser.

104.8 – The First Level Directories

`/opt` Location for 3rd party applications.

`/proc` Virtual filesystem created by the kernel.

`/root` Home for the superuser.

`/sbin` Esential utilities for system administration

104.8 – The First Level Directories

`/opt` Location for 3rd party applications.

`/proc` Virtual filesystem created by the kernel.

`/root` Home for the superuser.

`/sbin` Esential utilities for system administration

`/tmp` Storage for temporary files—cleared at boot time.

104.8 – The First Level Directories

`/opt` Location for 3rd party applications.

`/proc` Virtual filesystem created by the kernel.

`/root` Home for the superuser.

`/sbin` Esential utilities for system administration

`/tmp` Storage for temporary files—cleared at boot time.

`/usr` Hierarchy of non esential executables etc.

104.8 – The First Level Directories

`/opt` Location for 3rd party applications.

`/proc` Virtual filesystem created by the kernel.

`/root` Home for the superuser.

`/sbin` Esential utilities for system administration

`/tmp` Storage for temporary files—cleared at boot time.

`/usr` Hierarchy of non esential executables etc.

`/var` Hierarchy of variable data.

104.8 – The Filesystem partitions

```
geoffrey@mintie:~$ df -h
Filesystem         Size  Used Avail Use% Mounted on
/dev/sda1          3.2G  2.7G  357M  89% /
/dev/sda2          182M   57M  116M  33% /tmp
/dev/sda3          273M   89M  170M  35% /var
/dev/hda1           99M   21M   72M  23% /boot
/dev/hda2           99M   13M   80M  14% /boot~
/dev/hda3          9.6G  8.6G  624M  94% /home
/dev/hda5           14G  7.1G  6.6G  52% /usr/local
/dev/hda6          4.8G 1021M  3.5G  22% /var/cache/apt/archives
/dev/hda7          4.8G   20k  4.5G   1% /backups
/dev/hda8          2.8G  318M  2.3G  12% /var/lib
/dev/hdc1          1.9G  1.2G  739M  61% /mnt/hdc1
```

104.8 – The Filesystem partitions

/ `(root)` Small partition containing the files esential for booting the system.

104.8 – The Filesystem partitions

/ `(root)` Small partition containing the files esential for booting the system.

/boot Files for the boot loader.

104.8 – The Filesystem partitions

/ (root) Small partition containing the files esential for booting the system.

/boot Files for the boot loader.

/home Home directories for system users.

104.8 – The Filesystem partitions

/ (root) Small partition containing the files esential for booting the system.

/boot Files for the boot loader.

/home Home directories for system users.

/opt Location for 3rd party applications.

104.8 – The Filesystem partitions

/ `(root)` Small partition containing the files esential for booting the system.

/boot Files for the boot loader.

/home Home directories for system users.

/opt Location for 3rd party applications.

/tmp Storage for temporary files—cleared at boot time.

104.8 – The Filesystem partitions

/ `(root)` Small partition containing the files esential for booting the system.

/boot Files for the boot loader.

/home Home directories for system users.

/opt Location for 3rd party applications.

/tmp Storage for temporary files—cleared at boot time.

/usr Hierarchy of non esential executables etc.

104.8 – The Filesystem partitions

/ `(root)` Small partition containing the files esential for booting the system.

`/boot` Files for the boot loader.

`/home` Home directories for system users.

`/opt` Location for 3rd party applications.

`/tmp` Storage for temporary files—cleared at boot time.

`/usr` Hierarchy of non esential executables etc.

`/var` Hierarchy of variable data.

104.8 – The `root` Filesystem: Esentials

`/bin`

`/dev`

`/etc`

`/lib`

`/mnt`

`/root`

`/sbin`

`/lib`

104.8 – The `root` Filesystem: Esentials

`/bin`

`/dev`

`/etc`

`/lib`

`/mnt`

`/root`

`/sbin`

`/lib`

Also required: **`/boot`** and **`swap`**

104.8 – The `/usr` Filesystem

`/usr/X11R6` Files for XFree86.

104.8 – The `/usr` Filesystem

`/usr/X11R6` Files for XFree86.

`/usr/bin` User commands not needed for emergency maintenance.

104.8 – The `/usr` Filesystem

`/usr/X11R6` Files for XFree86.

`/usr/bin` User commands not needed for emergency maintenance.

`/usr/games` Area for console games.

104.8 – The `/usr` Filesystem

`/usr/X11R6` Files for XFree86.

`/usr/bin` User commands not needed for emergency maintenance.

`/usr/games` Area for console games.

`/usr/include` Header files for C and C++.

104.8 – The `/usr` Filesystem

`/usr/X11R6` Files for XFree86.

`/usr/bin` User commands not needed for emergency maintenance.

`/usr/games` Area for console games.

`/usr/include` Header files for C and C++.

`/usr/lib` Shared program libraries. (+perl modules)

104.8 – The `/usr` Filesystem

`/usr/X11R6` Files for XFree86.

`/usr/bin` User commands not needed for emergency maintenance.

`/usr/games` Area for console games.

`/usr/include` Header files for C and C++.

`/usr/lib` Shared program libraries. (+perl modules)

`/usr/local` Hierarchy for local files not in distribution.

104.8 – The `/usr` Filesystem

`/usr/X11R6` Files for XFree86.

`/usr/bin` User commands not needed for emergency maintenance.

`/usr/games` Area for console games.

`/usr/include` Header files for C and C++.

`/usr/lib` Shared program libraries. (+perl modules)

`/usr/local` Hierarchy for local files not in distribution.

`/usr/sbin` System administration utilities not essential for emergency system maintenance.

104.8 – The `/usr` Filesystem

`/usr/X11R6` Files for XFree86.

`/usr/bin` User commands not needed for emergency maintenance.

`/usr/games` Area for console games.

`/usr/include` Header files for C and C++.

`/usr/lib` Shared program libraries. (+perl modules)

`/usr/local` Hierarchy for local files not in distribution.

`/usr/sbin` System administration utilities not essential for emergency system maintenance.

`/usr/share` Hierarchy of data files that are hardware independent. (e.g. documentation)

104.8 – The `/usr` Filesystem

`/usr/X11R6` Files for XFree86.

`/usr/bin` User commands not needed for emergency maintenance.

`/usr/games` Area for console games.

`/usr/include` Header files for C and C++.

`/usr/lib` Shared program libraries. (+perl modules)

`/usr/local` Hierarchy for local files not in distribution.

`/usr/sbin` System administration utilities not essential for emergency system maintenance.

`/usr/share` Hierarchy of data files that are hardware independent. (e.g. documentation)

`/usr/src` Linux source code.

104.8 – The `/var` Filesystem

`/var/account` Can be used for process accounting data.

104.8 – The `/var` Filesystem

`/var/account` Can be used for process accounting data.

`/var/cache` Temporary storage of intermaediate data.

104.8 – The `/var` Filesystem

`/var/account` Can be used for process accounting data.

`/var/cache` Temporary storage of intermaediate data.

`/var/crash` Crash dumps.

104.8 – The `/var` Filesystem

`/var/account` Can be used for process accounting data.

`/var/cache` Temporary storage of intermaediate data.

`/var/crash` Crash dumps.

`/var/games` Console game saves and score data.

104.8 – The `/var` Filesystem

`/var/account` Can be used for process accounting data.

`/var/cache` Temporary storage of intermaediate data.

`/var/crash` Crash dumps.

`/var/games` Console game saves and score data.

`/var/lock` Lock files for various programs.

104.8 – The `/var` Filesystem

`/var/account` Can be used for process accounting data.

`/var/cache` Temporary storage of intermaediate data.

`/var/crash` Crash dumps.

`/var/games` Console game saves and score data.

`/var/lock` Lock files for various programs.

`/var/log` Main repository for system log files.

104.8 – The `/var` Filesystem

`/var/account` Can be used for process accounting data.

`/var/cache` Temporary storage of intermaediate data.

`/var/crash` Crash dumps.

`/var/games` Console game saves and score data.

`/var/lock` Lock files for various programs.

`/var/log` Main repository for system log files.

`/var/mail` System mailbox containing mail files for users.

104.8 – The `/var` Filesystem

`/var/opt` Transient srorage for 3rd party programs.

104.8 – The `/var` Filesystem

`/var/opt` Transient srorage for 3rd party programs.

`/var/run` Files describing the current state of the system.

104.8 – The `/var` Filesystem

`/var/opt` Transient srorage for 3rd party programs.

`/var/run` Files describing the current state of the system.

`/var/spool` Information that is queued for processing. (e.g. print jobs)

104.8 – The `/var` Filesystem

`/var/opt` Transient srorage for 3rd party programs.

`/var/run` Files describing the current state of the system.

`/var/spool` Information that is queued for processing. (e.g. print jobs)

`/var/state` Data for applications with multiple instances.

104.8 – The `/var` Filesystem

`/var/opt` Transient srorage for 3rd party programs.

`/var/run` Files describing the current state of the system.

`/var/spool` Information that is queued for processing. (e.g. print jobs)

`/var/state` Data for applications with multiple instances.

`/var/tmp` A more persistant version of `/tmp`.

104.8 – The `/var` Filesystem

`/var/opt` Transient srorage for 3rd party programs.

`/var/run` Files describing the current state of the system.

`/var/spool` Information that is queued for processing. (e.g. print jobs)

`/var/state` Data for applications with multiple instances.

`/var/tmp` A more persistant version of `/tmp`.

`/var/yp` Database files for NIS.

104.8 –

104.8 – `find` - search for files in a directory hierarchy

The find command is one of the fundamental tools of Unix.

It is a tool that is constantly rediscovered as you perform more and more complex operations with it.

The man page of this simple tool is 555 lines long.

To find the file `foo.txt` somewhere:

```
$find / -name foo.txt
```

104.8 – `find` - basic use

- The most basic use of find is:

  `$find <directory> -name "<mask>"`

- To find a missing file somewhere in you home directory

  `$ find ~ -name missing.file`

  where ~ is shorthand for your home directory.

104.8 – `find` - basic use

- The most basic use of find is:

  `$find <directory> -name "<mask>"`

- To find a missing file somewhere in you home directory

  `$ find ~ -name missing.file`

  where ~ is shorthand for your home directory.

- You can also use masks like `"*.txt.gz"` but you must put it in quotes.
  — Why do you have to put it in quotes?

104.8 – `find` - basic use

- The most basic use of find is:

  ```
  $find <directory> -name "<mask>"
  ```

- To find a missing file somewhere in you home directory

  ```
  $ find ~ -name missing.file
  ```

  where ~ is shorthand for your home directory.

- You can also use masks like `"*.txt.gz"` but you must put it in quotes.
  — Why do you have to put it in quotes?

- When you use an * in a bash command line it is interpreted as a file expansion and it is looked for in the current directory and if it does exist it is substituted before the command is sent to find. If it is not found then your shell may generate an error message (for example csh, I think).

104.8 – `find` - basic options

According to the man page ’find - search for files in a directory hierarchy’ This is true but you can also find directories as well, like the filesystems .

First we will start with some basic options:

`-print` list the filename (default, never really use it).

`-exec` run a command

`-ok` run a command after prompting for confirmation.

`-ls` list file like `‘ls -dils‘`, is a lot of file information.

104.8 – `find` - advanced options

Advanced doing options, I am sure you will use these one day:

`-prune` don't descend past this directory.

`-printf` print a filename based on format like C printf.

`-print0` print but end with a null character.

`-fprintf` <**`fn`**> print a format string to a filename, (scripting??)

`-fprint` <**`fn`**> print filenames to a file.

`-fls` <**`fn`**> ls to a file

104.8 – `find` - entry selection by age

Most of these options take a *number*, +*number* or -*number*.

A little explanation is required first.

`-atime 2` Will pick any file accessed two days ago.

`-atime -2` Will pick any file access more than two days ago

`-atime +2` Will pick any file accessed in the last day.

104.8 – `find` - entry selection by age

Most of these options take a *number*, *+number* or *-number*.

A little explanation is required first.

`-atime n` files on access date

`-ctime n` files on creation date (note chmod mucks this up

`-mtime n` files on modification date

`-anewer <file>` files on access date based on another file.

`-cnewer<file>` files on creation date based on another file.

`-newer <file>` files on modification date based on another file

Example: delete all files older than 7 days in the `/data` directory who have an `.A` extension.

104.8 – `find` - entry selection by age

Most of these options take a *number*, *+number* or *-number*.

A little explanation is required first.

`-atime n` files on access date

`-ctime n` files on creation date (note chmod mucks this up

`-mtime n` files on modification date

`-anewer <file>` files on access date based on another file.

`-cnewer<file>` files on creation date based on another file.

`-newer <file>` files on modification date based on another file

Example: delete all files older than 7 days in the `/data` directory who have an `.A` extension.

```
$ find /data -name "*.A" -ctime -7 -exec rm {} \;
```

104.8 – `find` - entry selection by age

- A script may run a command and then ’touch’ a tag file to give a timestamp when it was run.
- Assume that the last thing a script does is touch modification.tag in the `/parms` directory.
- Write a command line that lists all details of files modified in the `/apps/source/` directory based on this tag file.

104.8 – **find** - entry selection by age

- A script may run a command and then 'touch' a tag file to give a timestamp when it was run.
- Assume that the last thing a script does is touch modification.tag in the `/parms` directory.
- Write a command line that lists all details of files modified in the `/apps/source/` directory based on this tag file.

```
$ find /data -mnewer /parms/modification.tag -ls
```

104.8 – `find` - entry selection by owner and group

One problem with the Unix authentification system, when you delete a userid you end up with magic numbers on a directory listing. It is handy to be able to change the ownership on all files from the exiting staff member to the new person working on those projects.

`-nouser` users numeric id does not have and entry in /etc/passwd

`-nogroup` group numeric id does not have an entry in /etc/group

`-uid n` User by number

`-user name` User by name

`-gid n` Group by number

`-group name` Group by name

I recently converted from Redhat to Debian. I installed a new harddisk and mounted the old one as `/mnt/old1`. I notice that when I do `ls -al` I get a username of 500 in the directory listing. Change all the occurrences of 500 to the username of ken.

104.8 – `find` - entry selection by owner and group

One problem with the Unix authentification system, when you delete a userid you end up with magic numbers on a directory listing. It is handy to be able to change the ownership on all files from the exiting staff member to the new person working on those projects.

`-nouser` users numeric id does not have and entry in /etc/passwd

`-nogroup` group numeric id does not have an entry in /etc/group

`-uid n` User by number

`-user name` User by name

`-gid n` Group by number

`-group name` Group by name

I recently converted from Redhat to Debian. I installed a new harddisk and mounted the old one as `/mnt/old1`. I notice that when I do `ls -al` I get a username of 500 in the directory listing. Change all the occurrences of 500 to the username of ken.

```
# find /mnt/old1 -uid 500 -exec chown ken {} \;
```

104.8 – `find` - entry selection by Inode number and links

You have a directory listing, the hard link count is greater than 1. ... lecturers note wait and ask class how we know this ....

You have no idea where the other hard link is and you want to locate the other version to see what impact a change may have.

`-inode n`

104.8 – `find` - entry selection by Inode number and links

You have a directory listing, the hard link count is greater than 1. ... lecturers note wait and ask class how we know this ....

You have no idea where the other hard link is and you want to locate the other version to see what impact a change may have.

`-inode n`

```
# find -inode ????
```

104.8 – `find` - entry selection by regex and xdev

Advanced options on what entries we select:

`-iregex` Use regex rather than standard file masks.

Options on how we go through the directories:

`-xdev` don't go into other file systems.

104.8 – `locate`, `slocate` and `updatedb`

`locate` is used to find files on the system using a database rather than actually having to search the entire directory tree.

```
$ locate slide.tex ↩
/home/geoffrey/tafe/adp/bash/slide.tex
```

104.8 – `locate`, `slocate` and `updatedb`

`locate` is used to find files on the system using a database rather than actually having to search the entire directory tree.

```
$ locate slide.tex ↩
/home/geoffrey/tafe/adp/bash/slide.tex
```

- `updatedb` is the command that updates the database.

104.8 – `locate`, `slocate` and `updatedb`

`locate` is used to find files on the system using a database rather than actually having to search the entire directory tree.

```
$ locate slide.tex ↩
/home/geoffrey/tafe/adp/bash/slide.tex
```

- `updatedb` is the command that updates the database.
- The database is kept in `/var/lib/locate/locatedb`

104.8 – `locate`, `slocate` and `updatedb`

`locate` is used to find files on the system using a database rather than actually having to search the entire directory tree.

```
$ locate slide.tex ↩
/home/geoffrey/tafe/adp/bash/slide.tex
```

- `updatedb` is the command that updates the database.
- The database is kept in `/var/lib/locate/locatedb`
- `slocate` The Security Enhanced version of the GNU Locate— now replaces all of these. `locate` and `updatedb` become a symbolic links to `slocate`

104.8 – `locate` vs. `slocate`

104.8 – `locate` vs. `slocate`

- The `locate` command as now been replaced by `slocate` and is now a Symbolic link to `slocate`.

104.8 – `locate` vs. `slocate`

- The `locate` command as now been replaced by `slocate` and is now a Symbolic link to `slocate`.
- The `updatedp` command as now been replaced by `slocate` and is now a Symbolic link to `slocate`.

104.8 – `locate` vs. `slocate`

- The `locate` command as now been replaced by `slocate` and is now a Symbolic link to `slocate`.
- The `updatedp` command as now been replaced by `slocate` and is now a Symbolic link to `slocate`.

```
$ ls -l /usr/bin/ |grep locate ↩
lrwxrwxrwx  1 root root         7 Mar 27 10:44 locate -> slocate*
-rwxr-xr-x  1 root root      9228 Nov  5  2000 locate.notslocate*
-rwxr-sr-x  1 root slocate 24504 Feb  4 03:57 slocate*
lrwxrwxrwx  1 root root         7 Mar 27 10:44 updatedb -> slocate*
-rwxr-xr-x  1 root root      5860 Nov  5  2000 updatedb.notslocate*
```

104.8 – `locate` vs. `slocate`

- The `locate` command as now been replaced by `slocate` and is now a Symbolic link to `slocate`.
- The `updatedp` command as now been replaced by `slocate` and is now a Symbolic link to `slocate`.

```
$ ls -l /usr/bin/ |grep locate ↩
lrwxrwxrwx  1 root root         7 Mar 27 10:44 locate -> slocate*
-rwxr-xr-x  1 root root      9228 Nov  5  2000 locate.notslocate*
-rwxr-sr-x  1 root slocate 24504 Feb  4 03:57 slocate*
lrwxrwxrwx  1 root root         7 Mar 27 10:44 updatedb -> slocate*
-rwxr-xr-x  1 root root      5860 Nov  5  2000 updatedb.notslocate*
```

- The database is now in `/var/lib/slocate/slocatedb`
- The database is no longer world readable

104.8 – `slocate` - Security Enhanced version of the GNU Locate

104.8 – `slocate` - Security Enhanced version of the GNU Locate

- Slocate is used to find files on the system without actually having to search the entire directory tree.

104.8 – `slocate` - Security Enhanced version of the GNU Locate

- Slocate is used to find files on the system without actually having to search the entire directory tree.
- A database of all files on the system is created and is then used by slocate to reveal the files actual location.

104.8 – `slocate` - Security Enhanced version of the GNU Locate

- Slocate is used to find files on the system without actually having to search the entire directory tree.
- A database of all files on the system is created and is then used by slocate to reveal the files actual location.
- It is important to note that slocate may return a result which is no longer valid since the directory structure may have been modified since the slocate database was last created.

104.8 – `slocate` - Security Enhanced version of the GNU Locate

- Slocate is used to find files on the system without actually having to search the entire directory tree.
- A database of all files on the system is created and is then used by slocate to reveal the files actual location.
- It is important to note that slocate may return a result which is no longer valid since the directory structure may have been modified since the slocate database was last created.
- For example, you create a file called poobar.txt, create the slocate database and then remove poobar.txt.

104.8 – `slocate` - Security Enhanced version of the GNU Locate

- Slocate is used to find files on the system without actually having to search the entire directory tree.
- A database of all files on the system is created and is then used by slocate to reveal the files actual location.
- It is important to note that slocate may return a result which is no longer valid since the directory structure may have been modified since the slocate database was last created.
- For example, you create a file called poobar.txt, create the slocate database and then remove poobar.txt.
- Slocate will still return poobar.txt?s original location until the slocate database is recreated.

104.8 – `slocate` - Security Enhanced version of the GNU Locate

- Slocate is used to find files on the system without actually having to search the entire directory tree.
- A database of all files on the system is created and is then used by slocate to reveal the files actual location.
- It is important to note that slocate may return a result which is no longer valid since the directory structure may have been modified since the slocate database was last created.
- For example, you create a file called poobar.txt, create the slocate database and then remove poobar.txt.
- Slocate will still return poobar.txt?s original location until the slocate database is recreated.
- Slocate can be used in two modes:
  - Search mode:- To locate an actual file within the database
  - Database creation mode:- To build the database

104.8 – `locate` usage

```
locate [-qi]  [-d <path>]  [-r <regexp>]  <search string>...
```

Options:

-q Quiet mode. Suppress error messages.

-i Does a case insensitive search.

-d Specify a database to use.

-r Pass a regular expression instead of a search string.

104.8 – `locate` examples

`locate ls` `$ locate ls` ←↩

```
...
 /etc/X11/xkb/symbols/xfree68/ataritt
/etc/X11/xkb/symbols/xfree68/amiga
/etc/alternatives/tclsh
...
```

`locate -r "/ls$"` `$ locate -r "/ls$"` ←↩

```
/home/geoffrey/tafe/mos/compress/ls
/usr/lib/bitchx/help/8_Scripts/ls
/bin/ls
```

The above example illustrates the need for a regex option to `locate`. In the first example there will be lots of hits. In the second there is only one (the actual `ls` command).

As well as searching for a file in the database, `locate` can also build the search database.

104.8 – Database creation with `slocate`

As well as searching for a file in the database, slocate can also build the search database.

`-u` Create slocate database starting at path `/`.

`-U` <**`dir`**> Create slocate database starting at path $< dir >$.

`-c` Parse original GNU Locate's `/etc/updatedb.conf`

`-e` <**`dir1...`**>
Exclude directories from the slocate database when using the -u or -U options.

`-f` <**`fs...`**> Exclude file system types from the slocate database

-l Security level. 0–> security off, 1–> security on

`-q` Quiet mode. Error messages are suppressed.

`-o` <**`file`**> Specify the name of the database file to create

`-v` Be verbose

104.8 – Database creation with `slocate`— Examples

104.8 – Database creation with **`slocate`**— Examples

- Create a database for all directories under `/usr` and place the resulting database file into `slocate.db` in andy's home directory.

  ```
  # slocate -U /usr -o /home/andy/slocate.db  ↩
  ```

104.8 – Database creation with **slocate**— Examples

- Create a database for all directories under `/usr` and place the resulting database file into `slocate.db` in andy's home directory.

```
# slocate -U /usr -o /home/andy/slocate.db
```

- Create a database for all directories under /usr, excluding directories under /usr/man and place the resulting database file into slocate.db in andy?s home directory.

```
# slocate  -U /usr   -e /usr/man  -o /home/andy/slocate.db
```

104.8 – Update `slocate` database—`update`

`updatedb` is simply a link to slocate that implies the -u option. (Excerpt from the man page:- man updatedb)

```
$ ls -l ‘which updatedb‘ ↩
lrwxrwxrwx  1 root  root 7 Mar 27 10:44 /usr/bin/updatedb -> slocate*
```

`updatedb` is typically executed periodically via cron.

104.8 – `/etc/updatedb.conf`

104.8 – /etc/updatedb.conf

- The updatedb (or slocate) tool can use a configuration file to decide which directories and file systems are included when the database is created. This file is normally located in /etc/updatedb.conf

104.8 – `/etc/updatedb.conf`

- The `updatedb` (or `slocate`) tool can use a configuration file to decide which directories and file systems are included when the database is created. This file is normally located in `/etc/updatedb.conf`
- The following is a list of keywords that are recognised by updatedb (slocate) and their equivalent command line options

  PRUNEFS `<fs_type1 fs_type2...>` - Option `-f`

  PRUNEPATHS `<dir1 dir2 dir3...>` - Opion `-e`

104.8 – /etc/updatedb.conf

- The `updatedb` (or `slocate`) tool can use a configuration file to decide which directories and file systems are included when the database is created. This file is normally located in `/etc/updatedb.conf`
- The following is a list of keywords that are recognised by updatedb (slocate) and their equivalent command line options

  PRUNEFS `<fs_type1 fs_type2...>` - Option `-f`

  PRUNEPATHS `<dir1 dir2 dir3...>` - Opion `-e`
- Example `updatedb.conf`

```
PRUNEFS="devpts NFS nfs afs proc smbfs autofs auto iso9660"
PRUNEPATHS="/tmp /usr/tmp /var/tmp /afs /net?
export PRUNEFS
export PRUNEPATHS
```

104.8 – `slocate` Exercises

1. Create an `slocate` database in your home directory including all directories from `/` down.
2. Using the database created in step 1, locate all files with `rm` in the filename
3. Using the database created in step 1, locate the executable file `rm` using a regex. (ie `/some/path/rm`)
4. Create an slocate database in your home directory include all directories from `/` down but excluding the `/bin` directory.
5. Repeat (2) and (3) above. Do you notice anything different ?
6. After backing up your existing `/etc/updatedb.conf`, say

   ```
   # cp /etc/updatedb.conf /etc/updatedb.conf.orig ↩
   ```

   edit `/etc/updatedb.conf` to perform the same actions as in step (4).
7. When you have finished this exercise restore your original `/etc/updatedb.conf`.

104.8 – `whereis`

104.8 – `which`

104.8 – `/etc/updatedb.conf`

104.8 –

106 – (2.6)Boot, Initialization, Shutdown, Runlevels [6]

1. Boot the system [3]
2. Change runlevels and shutdown or reboot system [3]

106.1 – Boot the system [3]

106.1 – Objective

Candidates should be able to guide the system through the booting process. This includes giving commands to the boot loader and giving options to the kernel at boot time, and checking the events in the log files.

106.1 – Key files, terms, and utilities include:

`dmesg` `/var/log/messages`

`/etc/conf.modules` or `/etc/modules.conf`

106.1 – Resources

TBA

106.2 – Change runlevels and shutdown or reboot system [3]

106.2 – Objective

Candidates should be able to manage the runlevel 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 runlevel, and properly terminate processes. This objective also includes setting the default runlevel.

106.2 – Key files, terms, and utilities include:

`shutdown` `init` `/etc/inittab`

106.2 – Resources

TBA

106.2 –

106.2 –

- a

106.2 –

- a
- b

106.2 –

- a
- b
- c

106.2 –

- a
- b
- c

- d

106.2 –

- a
- b
- c

- d
- e

108 – (1.8) Documentation [8]

1. Use and manage local system documentation [5]
2. Find Linux documentation on the Internet [2]
3. Write System Documentation [1]
4. Provide User Support [1]
5. Notify users on system-related issues [1]

108.1 – Use and manage local system documentation [5]

108.1 – Objective

Candidates should be able to use and administer the man facility and the material in `/usr/share/doc/`. This objective includes finding relevant man pages, searching man page sections, finding commands and man pages related to them, and configuring access to man sources and the man system. It also includes using system documentation stored in

108.1 – Key files, terms, and utilities include:

`man` `apropos` `whatis` `MANPATH`

108.1 – Resources

TBA

108.2 – Find Linux documentation on the Internet [2]

108.2 – Objective

Candidates should be able to find and use Linux documentation. This objective includes using Linux documentation at sources such as the Linux Documentation Project (LDP), vendor and third-party websites, newsgroups, newsgroup archives, and mailing lists.

108.2 – Key files, terms, and utilities include:

`not applicable`

108.2 – Resources

TBA

108.3 – Notify users on system-related issues [1]

108.3 – Objective

Candidates should be able to notify the users about current issues related to the system. This objective includes automating the communication process, e.g. through logon messages.

108.3 – Key files, terms, and utilities include:

`/etc/issue` `/etc/issue.net` `/etc/motd`

108.3 – Resources

TBA

108.3 –

108.3 –

- a

108.3 –

- a
- s

108.3 –

- a
- s
- s

108.3 –

- a
- s
- s

- s

108.3 –

- a
- s
- s

- s
- s

111 – (2.11) Administrative Tasks [24]

1. Manage users and group accounts and related system files [7]
2. Tune the user environment and system environment variables [4]
3. Configure and use system log files to meet administrative and security needs [3]
4. Automate system administration tasks by scheduling jobs to run in the future [4]
5. Maintain an effective data backup strategy [3]
6. Maintain system time [3]

111.1 – Manage user/group accounts, related system files [7]

111.1 – Objective

Candidate should be able to add, remove, suspend and change user accounts. Tasks include to add and remove groups, to change user/group info in passwd/group databases. The objective also includes creating special purpose and limited accounts.

111.1 – Key files, terms, and utilities include:

`chageg passwd groupadd groupdel groupmod grpconv grpunconv passwd pwconv pwunconv useradd userdel usermod /etc/passwd /etc/shadow /etc/group /etc/gshadow`

111.1 – Resources

Chapter 9 - Managing User Accounts: The Linux System Administrators' Guide

Manpages for `useradd usermod userdel groupadd groupmod groupdel useradd passwd chage`

111.1 – The /etc/passwd file

111.1 – `useradd` - Create a new user or update new user information

`# useradd [options] user`

Options:

`-c` “comment” GCOS field—Users name

`-d homedir` Use `homedir` as the users home directory

`-D` List and optionally change the default values

`-m` Create and populate the users home directory

`-s shell` Use `shell` as the users shell

111.1 – `useradd`—examples

- The defaults:

```
# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
```

111.1 – `useradd`—examples

- The defaults:

```
# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
```

- Vanilla :

```
# useradd quincy
```

111.1 – **useradd**—examples

- The defaults:

```
# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
```

- Vanilla :

```
# useradd quincy
```

- With nuts:

```
# useradd -m -c "Quincy Shnorks" -s zsh quincy
```

111.1 – `passwd` - change user password

- The command:

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

111.1 – `passwd` - change user password

- The command:

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

- `passwd` changes passwords for user and group accounts.

111.1 – `passwd` - change user password

- The command:

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

- `passwd` changes passwords for user and group accounts.
- A normal user may only change the password for their own account

111.1 – `passwd` - change user password

- The command:

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

- `passwd` changes passwords for user and group accounts.
- A normal user may only change the password for their own account
- The super user may change the password for any account.

111.1 – `passwd` - change user password

- The command:

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

- `passwd` changes passwords for user and group accounts.
- A normal user may only change the password for their own account
- The super user may change the password for any account.
- The administrator of a group may change the password for the group.

111.1 – `passwd` - change user password

- The command:

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

- `passwd` changes passwords for user and group accounts.
- A normal user may only change the password for their own account
- The super user may change the password for any account.
- The administrator of a group may change the password for the group.
- `passwd` also changes account information, such as the full name of the user, their login shell, or password expiry dates and intervals.

111.1 – `passwd` example

- A user may change their own password:

```
$ passwd
Changing password for geoffrey
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

111.1 – `passwd` example

- A user may change their own password:

```
$ passwd
Changing password for geoffrey
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

- Assuming root privileges:

```
$ su -
Password:
```

111.1 – `passwd` example

- A user may change their own password:

```
$ passwd
Changing password for geoffrey
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

- Assuming root privileges:

```
$ su -
Password:
```

- root may change any passwd:

```
# passwd geoffrey
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

111.1 – passwd guidelines

As a general guideline, passwords should consist of 6 to 8 characters including one or more from each of following sets:

- Lower case alphabetics
- Upper case alphabetics
- Digits 0 through 9
- Punctuation marks

Examples of secure and memorable passwds:

- The Owl and the Pussycat went to sea—`TO&tPw2s`
- I got it—`1_got_IT`
- One and all.— `1_&_All.`

111.1 –

111.1 –

- a

111.1 –

- a
- s

111.1 –

- a
- s
- s

111.1 –

- a
- s
- s

- s

111.1 –

- a
- s
- s

- s
- s

111.2 – Tune user and system environment variables [4]

111.2 – Objective

Candidate should be able to modify global and user profiles. This includes setting environment variables, maintaining skel directories for new user accounts and setting command search path with the proper directory.

111.2 – Key files, terms, and utilities include:

`env export set unset /etc/profile /etc/skel`

111.2 – Resources

TBA

111.3 – Config & use system log files for admin / security [3]

111.3 – Objective

Candidate should be able to configure system logs. This objective includes managing the type and level of information logged, manually scanning log files for notable activity, monitoring log files, arranging for automatic rotation and archiving of logs and tracking down problems noted in logs.

111.3 – Key files, terms, and utilities include:

`logrotate /etc/syslog.conf /var/log/* tail -f`

111.3 – Resources

TBA

111.4 – Automate system admin tasks by job scheduling[4]

111.4 – Objective

Candidate 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. Task include managing cron and at jobs and configuring user access to cron and at services.

111.4 – Key files, terms, and utilities include:

`at atq crontab /etc/anacrontab /etc/at.deny /etc/at.allow /etc/crontab /etc/cron.allow /etc/cron.deny /var/spool/cron/*`

111.4 – Resources

TBA

111.5 – Maintain an effective data backup strategy [3]

111.5 – Objective

Candidate should be able to plan a backup strategy and backup filesystems automatically to various media. Tasks include dumping a raw device to a file or vice versa, performing partial and manual backups, verifying the integrity of backup files and partially or fully restoring backups.

111.5 – Key files, terms, and utilities include:

`cpio dd dump restore tar`

111.5 – Resources

TBA

111.5 – System Backup Tools

Backing up a system is a vital part of system administration. A good backup strategy involves the following key features:

111.5 – System Backup Tools

Backing up a system is a vital part of system administration. A good backup strategy involves the following key features:

- Frequent backups (at least daily)

111.5 – System Backup Tools

Backing up a system is a vital part of system administration. A good backup strategy involves the following key features:

- Frequent backups (at least daily)
- Backup media is maintained on a rotating basis (eg 28 sets of backup media, rotated daily)

111.5 – System Backup Tools

Backing up a system is a vital part of system administration. A good backup strategy involves the following key features:

- Frequent backups (at least daily)
- Backup media is maintained on a rotating basis (eg 28 sets of backup media, rotated daily)
- At least one copy of the backup should be kept off-site

111.5 – System Backup Tools

Backing up a system is a vital part of system administration. A good backup strategy involves the following key features:

- Frequent backups (at least daily)
- Backup media is maintained on a rotating basis (eg 28 sets of backup media, rotated daily)
- At least one copy of the backup should be kept off-site
- Backup must have been tested

111.5 – Backup Tools

There are many strategies used for system backups. These range from copying a simple file to entire disk partitions to an entire disk.

Some tools available under linux are:

111.5 – Backup Tools

There are many strategies used for system backups. These range from copying a simple file to entire disk partitions to an entire disk.

Some tools available under linux are:

- dump / restore

111.5 – Backup Tools

There are many strategies used for system backups. These range from copying a simple file to entire disk partitions to an entire disk.

Some tools available under linux are:

- dump / restore
- rsync

111.5 – Backup Tools

There are many strategies used for system backups. These range from copying a simple file to entire disk partitions to an entire disk.

Some tools available under linux are:

- dump / restore
- rsync
- cpio

111.5 – Backup Tools

There are many strategies used for system backups. These range from copying a simple file to entire disk partitions to an entire disk.

Some tools available under linux are:

- dump / restore
- rsync
- cpio
- ar

111.5 – Backup Tools

There are many strategies used for system backups. These range from copying a simple file to entire disk partitions to an entire disk.

Some tools available under linux are:

- dump / restore
- rsync
- cpio
- ar
- tar

111.5 – Dump & Restore

`Dump` Saves source files or filesystems to backup media
(Like the DOS backup utility)

`Restore` Reinstate the backup to files or filesystems
(Like the DOS restore utility)

111.5 – `dump` usage

- Dump has 22 options, many of which concern themselves with various tape device setup options (density, tape length, etc.)

111.5 – `dump` usage

- Dump has 22 options, many of which concern themselves with various tape device setup options (density, tape length, etc.)
- To dump the `/boot` directory (in my case a seperate partition), do the following:

  ```
  $ /sbin/dump -0u -f  /root/boot.backup /boot ↩
  ```

111.5 – `dump` usage

- Dump has 22 options, many of which concern themselves with various tape device setup options (density, tape length, etc.)
- To dump the `/boot` directory (in my case a seperate partition), do the following:

  `$ /sbin/dump -0u -f  /root/boot.backup /boot ↩`

  `-0` (zero) tells dump to do the entire directory

111.5 – **`dump`** usage

- Dump has 22 options, many of which concern themselves with various tape device setup options (density, tape length, etc.)
- To dump the `/boot` directory (in my case a seperate partition), do the following:

```
$ /sbin/dump -0u -f  /root/boot.backup /boot ↩
```

`-0` (zero) tells dump to do the entire directory

`-u` update the file `/etc/dumpdates`

111.5 – `dump` usage

- Dump has 22 options, many of which concern themselves with various tape device setup options (density, tape length, etc.)
- To dump the `/boot` directory (in my case a seperate partition), do the following:

```
$ /sbin/dump -0u -f  /root/boot.backup /boot ↩
```

`-0` (zero) tells dump to do the entire directory

`-u` update the file `/etc/dumpdates`

`-f` write the backup to a file

111.5 – Dump Usage Example:

```
[root@Node4] root]# df -h /boot
Filesystem            Size  Used Avail Use% Mounted on
/dev/hda5             197M  4.1M  183M   3% /boot
```

111.5 – Dump Usage Example:

```
[root@Node4] root]# df -h /boot
Filesystem            Size  Used Avail Use% Mounted on
/dev/hda5             197M  4.1M  183M   3% /boot


[root@Node4] root]# dump -0u -f /root/boot.backup /boot
  DUMP: Date of this level 0 dump: Tue May  7 20:30:36 2002
  DUMP: Dumping /dev/hda5 (/boot) to /root/boot.backup
  DUMP: Label: /boot
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 4224 tape blocks.
  DUMP: Volume 1 started with block 1 at: Tue May  7 20:30:37 2002
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing /root/boot.backup
  DUMP: Volume 1 completed at: Tue May  7 20:30:37 2002
  DUMP: Volume 1 4210 tape blocks (4.11MB)
  DUMP: 4210 tape blocks (4.11MB) on 1 volume(s)
  DUMP: finished in less than a second
  DUMP: Date of this level 0 dump: Tue May  7 20:30:36 2002
  DUMP: Date this dump completed:  Tue May  7 20:30:37 2002
  DUMP: Average transfer rate: 0 kB/s
  DUMP: DUMP IS DONE
```

111.5 – Restore

Restore takes the data created by the dump command and puts it back to the filesystem.

111.5 – Restore

Restore takes the data created by the dump command and puts it back to the filesystem.

Restore can be used in one of these modes:

111.5 – Restore

Restore takes the data created by the dump command and puts it back to the filesystem.

Restore can be used in one of these modes:

- Compare - Verify backup file against filesystem

111.5 – Restore

Restore takes the data created by the dump command and puts it back to the filesystem.

Restore can be used in one of these modes:

- Compare - Verify backup file against filesystem
- Interactive restore - A small shell like interface to selectively restore files

111.5 – Restore

Restore takes the data created by the dump command and puts it back to the filesystem.

Restore can be used in one of these modes:

- Compare - Verify backup file against filesystem
- Interactive restore - A small shell like interface to selectively restore files
- Non-interactive restore - Restore files without asking.

111.5 – Restore

Restore takes the data created by the dump command and puts it back to the filesystem.

Restore can be used in one of these modes:

- Compare - Verify backup file against filesystem
- Interactive restore - A small shell like interface to selectively restore files
- Non-interactive restore - Restore files without asking.
- List mode - List contents of a backup file

111.5 – Restore

Restore takes the data created by the dump command and puts it back to the filesystem.

Restore can be used in one of these modes:

- Compare - Verify backup file against filesystem
- Interactive restore - A small shell like interface to selectively restore files
- Non-interactive restore - Restore files without asking.
- List mode - List contents of a backup file
- Selective restore - Restore files listed on the command line

111.5 – Restore Usage Example

This is the sequence for a `dump`—disaster—`restore` cycle:

111.5 – Restore Usage Example

This is the sequence for a `dump`—disaster—`restore` cycle:

- Make a backup:

```
# dump -0u -f /root/boot.backup /boot ↩
```

111.5 – Restore Usage Example

This is the sequence for a `dump`—disaster—`restore` cycle:

- Make a backup:

  `# dump -0u -f /root/boot.backup /boot ↩`

- Disk dies or some other catastrophe ....

111.5 – Restore Usage Example

This is the sequence for a `dump`—disaster—`restore` cycle:

- Make a backup:

  ```
  # dump -0u -f /root/boot.backup /boot ↩
  ```

- Disk dies or some other catastrophe ....

- Disaster recovery:

  ```
  # mke2fs /dev/hda1 ↩
  # mount /dev/hda1 /mnt ↩
  # cd /mnt ↩
  # restore -r -f /root/boot.backup ↩
  ```

111.5 – Restore Usage Example

This is the sequence for a `dump`—disaster—`restore` cycle:

- Make a backup:

  ```
  # dump -0u -f /root/boot.backup /boot ↩
  ```

- Disk dies or some other catastrophe ....
- Disaster recovery:

  ```
  # mke2fs /dev/hda1 ↩
  # mount /dev/hda1 /mnt ↩
  # cd /mnt ↩
  # restore -r -f /root/boot.backup ↩
  ```

(As much as I have faith in the restore command, your not going to see a real live example!!)

111.5 – Dump and Restore Demonstration

This is the sequence for a `dump`—disaster—`restore` cycle:

111.5 – Dump and Restore Demonstration

This is the sequence for a `dump`—disaster—`restore` cycle:

- Make a backup:

  `# dump -0 -f /root/boot.backup /boot` ↩

111.5 – Dump and Restore Demonstration

This is the sequence for a `dump`—disaster—`restore` cycle:

- Make a backup:

  ```
  # dump -0 -f /root/boot.backup /boot ↩
  ```

- Disk dies or some other catastrophe ....

111.5 – Dump and Restore Demonstration

This is the sequence for a `dump`—disaster—`restore` cycle:

- Make a backup:

  ```
  # dump -0 -f /root/boot.backup /boot ↩
  ```

- Disk dies or some other catastrophe ....

- Disaster recovery:

  ```
  # cd / ↩
  # restore -r -f /root/boot.backup ↩
  ```

111.5 – CPIO - file formats

The archive file can be stored in any of the following formats:

- binary (obsolete due to lack of portability)
- old ASCII or new ASCII
- crc
- HPUX binary or HPUX old ASCII
- old tar or POSIX.1 tar

111.5 – CPIO - Copy Out (Create) Usage

```
cpio -o [options]  < filenames_to_archive  [> archive file]
```

- The `-o` option can be replaced with the long `--create` option
- Output will be sent to stdout unless the `-F <file>` option is used
- The list of files to archive must come from stdin

111.5 – CPIO - Copy In (Extract) Usage

```
cpio -i [options]  [file_patterns_to_restore]  [< archive file]
```

- The `-i` option can be replaced with the long `--extract` option
- The input archive will come from stdin unless the `-F <file>` option is used
- If no restore file pattern is given, all files will be restored

111.5 – CPIO - Copy Pass Usage

```
cpio -p [options]  destination_directory  < filenames_to_copy
```

- The `-p` option can be replaced with the long `--pass-through` option
- The destination directory is mandatory
- The list of files to copy must come from stdin

111.5 – Options to `cpio` (for LPIC purposes)

111.5 – Options to `cpio` (for LPIC purposes)

`-a` Reset access time on files (as if they hadn't been read by cpio)

111.5 – Options to `cpio` (for LPIC purposes)

`-a` Reset access time on files (as if they hadn't been read by cpio)

`-A` Used to append to an archive (in copy-out mode)

111.5 – Options to `cpio` (for LPIC purposes)

`-a` Reset access time on files (as if they hadn't been read by cpio)

`-A` Used to append to an archive (in copy-out mode)

`--block-size N` Set the block size to N * 512 bytes

111.5 – Options to `cpio` (for LPIC purposes)

`-a` Reset access time on files (as if they hadn't been read by cpio)

`-A` Used to append to an archive (in copy-out mode)

`--block-size N` Set the block size to N * 512 bytes

`-d` Create leading directories

111.5 – Options to `cpio` (for LPIC purposes)

`-a` Reset access time on files (as if they hadn't been read by cpio)

`-A` Used to append to an archive (in copy-out mode)

`--block-size N` Set the block size to N * 512 bytes

`-d` Create leading directories

`-E <file>` Read file for additional file patterns to restore (copy-in mode)

111.5 – Options to `cpio` (for LPIC purposes)

`-a` Reset access time on files (as if they hadn't been read by cpio)

`-A` Used to append to an archive (in copy-out mode)

`--block-size N` Set the block size to N * 512 bytes

`-d` Create leading directories

`-E` <**`file`**> Read file for additional file patterns to restore (copy-in mode)

`-f` Copy files that don't match specified patterns

111.5 – Options to `cpio` (for LPIC purposes)

`-a` Reset access time on files (as if they hadn't been read by cpio)

`-A` Used to append to an archive (in copy-out mode)

`--block-size N` Set the block size to N * 512 bytes

`-d` Create leading directories

`-E` <**`file`**> Read file for additional file patterns to restore (copy-in mode)

`-f` Copy files that don't match specified patterns

`-F` <**`file`**> Specify an archive filename rather than stdin or stdout

111.5 – Options to `cpio` (for LPIC purposes)

`-a` Reset access time on files (as if they hadn't been read by cpio)

`-A` Used to append to an archive (in copy-out mode)

`--block-size N` Set the block size to N * 512 bytes

`-d` Create leading directories

`-E` <**`file`**> Read file for additional file patterns to restore (copy-in mode)

`-f` Copy files that don't match specified patterns

`-F` <**`file`**> Specify an archive filename rather than stdin or stdout

`-0` **(zero)** Filenames are terminated by null instead of newline (copy-out & copy-pass)

111.5 – Options to `cpio` (for LPIC purposes)

`-a` Reset access time on files (as if they hadn't been read by cpio)

`-A` Used to append to an archive (in copy-out mode)

`--block-size N` Set the block size to N * 512 bytes

`-d` Create leading directories

`-E` <**`file`**> Read file for additional file patterns to restore (copy-in mode)

`-f` Copy files that don't match specified patterns

`-F` <**`file`**> Specify an archive filename rather than stdin or stdout

`-0` **(zero)** Filenames are terminated by null instead of newline (copy-out & copy-pass)

`-r` Allows files to be renamed interactively

`-u` Replace files without asking

`-v` Verbosely list files that are processed

111.5 – CPIO - Examples

111.5 – CPIO - Examples

- Create a fully qualified list of filenames under `/boot`:

  ```
  # find /boot > list ↩
  ```

111.5 – CPIO - Examples

- Create a fully qualified list of filenames under `/boot`:

  ```
  # find /boot > list ↩
  ```

- Make an archive of `/boot` called `boot.backup` using default bin format:

  ```
  # cpio -o -F boot.backup -v < list ↩
  ```

111.5 – CPIO - Examples

- Create a fully qualified list of filenames under `/boot`:

  ```
  # find /boot > list ↩
  ```

- Make an archive of `/boot` called `boot.backup` using default bin format:

  ```
  # cpio -o -F boot.backup -v < list ↩
  ```

- Make an archive of `/boot` called `boot.backup` using new tar format

  ```
  # cpio -o -F boot.backup -H ustar -v < list ↩
  ```

111.5 – CPIO - Examples

- Create a fully qualified list of filenames under `/boot`:

  `# find /boot > list ↩`

- Make an archive of `/boot` called `boot.backup` using default bin format:

  `# cpio -o -F boot.backup -v < list ↩`

- Make an archive of `/boot` called `boot.backup` using new tar format

  `# cpio -o -F boot.backup -H ustar -v < list ↩`

- Use tar to list the contents of the `boot.backup` tar archive

  `# tar -tvf boot.backup ↩`

111.5 – CPIO - Examples

111.5 – CPIO - Examples

- List the contents of the `boot.backup` archive

```
# cpio -i -F boot.backup --list ↩
```

111.5 – CPIO - Examples

- List the contents of the `boot.backup` archive

  `# cpio -i -F boot.backup --list ←↩`

- Restore all files in `boot.backup` to original locations

  `# cpio -i -F boot.backup ←↩`

111.5 – CPIO - Examples

- List the contents of the `boot.backup` archive

  `# cpio -i -F boot.backup --list` ↩

- Restore all files in `boot.backup` to original locations

  `# cpio -i -F boot.backup` ↩

- Restore all files in under `/boot/grub` in `boot.backup` to original locations

  `# cpio -i -F boot.backup /boot/grub/*` ↩

111.5 – CPIO - Examples

- List the contents of the `boot.backup` archive

  ```
  # cpio -i -F boot.backup --list ↩
  ```

- Restore all files in `boot.backup` to original locations

  ```
  # cpio -i -F boot.backup ↩
  ```

- Restore all files in under `/boot/grub` in `boot.backup` to original locations

  ```
  # cpio -i -F boot.backup /boot/grub/* ↩
  ```

- Make a replica of `/boot` under `/root/boot`

  ```
  # cpio -p /root <list ↩
  ```

111.6 – Maintain system time [3]

111.6 – Objective

Candidate should be able to properly maintain the system time and synchronize the clock over NTP. Tasks include setting the system date and time, setting the BIOS clock to the correct time in UTC, configuring the correct timezone for the system and configuring the system to correct clock drift to match NTP clock.

111.6 – Key files, terms, and utilities include:

`date hwclock ntpd ntpdate /usr/share/zoneinfo /etc/timezone /etc/localtime /etc/ntp.conf /etc/ntp.drift`

111.6 – Resources

TBA

111.6 – Date: Display or Set System Date & Time

The date command without any options will print the current date and time. The date will be relative to any timezone set for the machine.

```
[andy@Node4]$ date ↩
Tue May 21 09:57:51 EST 2002
```

111.6 – Date: Options to the Date command

-I Output an ISO-8601 compliant date (YYYY-MM-DD)

```
$ date -I ↩
2002-05-21
```

-R Output an RFC-822 compliant date (Local time + GMT 0ffset)

```
$ date -R ↩
Tue, 21 May 2002 10:14:09 +1000
```

-r <**file**> Display the last modification time of file

```
$ date -r ~/ivr/va/src/va.c ↩
Mon May 20 12:55:48 EST 2002
```

111.6 – Date: Options to the Date command

-d <**STRING**> Display date described by string instead of now

```
$ date -d "last Monday 4 years ago" ↩
Mon May 18 00:00:00 EST 1998
```

-u Display UTC time & date instead of localtime

```
$ date ↩
Tue May 21 10:55:34 EST 2002
$ date -u ↩
Tue May 21 00:55:34 UTC 2002
```

111.6 – Date: Options to the Date command

-s <**date**> Set the system time (must be superuser)

```
# date -s "Tue May 21 10:03:06 EST 2002" ↩
Tue May 21 10:03:06 EST 2002
```

+FORMAT Display date in user defined format

```
$ date +"Today is %A, %d %B, %Y" ↩
Today is Tuesday, 21 May, 2002
```

111.6 – hwclock: RTC <==>System clock

Hwclock is used to do the following:

- Set the system clock from the Hardware clock
- Set the hardware clock from the system clock
- Show the time/date held by the RTC
- Adjust the RTC to account for clock drift

The *Real Time Clock (RTC)* is the hardware clock and is located on the motherboard of the system. This is what keeps track of the time when the system is not powered up. The *system clock* is maintained in the Linux kernel and is used while the system is running.

111.6 – hwclock: Set System clock to Hardware clock

- To set the system time from the RTC, use the following option to hwclock:

```
hwclock -s  (or hwclock --hctosys)
```

111.6 – hwclock: Set System clock to Hardware clock

- To set the system time from the RTC, use the following option to hwclock:

```
hwclock -s  (or hwclock --hctosys)
```

- To set the RTC from the system time, use this option:

```
hwclock -w  (or hwclock --systohc)
```

111.6 – hwclock: Set System clock to Hardware clock

- To set the system time from the RTC, use the following option to hwclock:

```
hwclock -s  (or hwclock --hctosys)
```

- To set the RTC from the system time, use this option:

```
hwclock -w  (or hwclock --systohc)
```

- To display the contents of the RTC, use this option:

```
hwclock -r  (or hwclock --show)
```

111.6 – hwclock: Set System clock to Hardware clock

- To set the system time from the RTC, use the following option to hwclock:

  ```
  hwclock -s  (or hwclock --hctosys)
  ```

- To set the RTC from the system time, use this option:

  ```
  hwclock -w  (or hwclock --systohc)
  ```

- To display the contents of the RTC, use this option:

  ```
  hwclock -r  (or hwclock --show)
  ```

- To adjsut the RTC for clock drift, use this option:

  ```
  hwclock -a  (or hwclock --adjust)
  ```

111.6 – hwclock: Set System clock to Hardware clock

- To set the system time from the RTC, use the following option to hwclock:

  `hwclock -s  (or hwclock --hctosys)`

- To set the RTC from the system time, use this option:

  `hwclock -w  (or hwclock --systohc)`

- To display the contents of the RTC, use this option:

  `hwclock -r  (or hwclock --show)`

- To adjsut the RTC for clock drift, use this option:

  `hwclock -a  (or hwclock --adjust)`

  Note that the file `/etc/adjtime` is used to hold information about the extent to which (and direction) your RTC drifts

111.6 – NTP - Network Time Protocol

NTP is a time protocol used to synchronise a systems clock to master time source. For example, the CSIRO maintains a nationwide time source with atomic clock accuracy. As a user I can synchronise my system to that time source by sending a request to the CSIRO's ntp server.

111.6 – NTP - Network Time Protocol

NTP is a time protocol used to synchronise a systems clock to master time source. For example, the CSIRO maintains a nationwide time source with atomic clock accuracy. As a user I can synchronise my system to that time source by sending a request to the CSIRO's ntp server.

Features and properties of NTP include:

- NTP takes into account the time taken to send/receive NTP packets
- Uses the UDP protocol
- Uses Port 123 plus one other unpriveledged port (1024:65535)
- Can operate in both client & server modes
- There are 3 versions of the protocol (ntp1, ntp2 & ntp3)
- Available for Unix & Windows machines.