

— LPI Certification —
— General Linux 2 —
(Study Notes)^{1 2}

geoffrey hector robertson
geoffrey@zip.com.au

July 21, 2005

¹Copyright ©2002 Geoffrey Robertson. Permission is granted to make and distribute verbatim copies or modified versions of this document provided that this copyright notice and this permission notice are preserved on all copies under the terms of the GNU General Public License as published by the Free Software Foundation—either version 2 of the License or (at your option) any later version.

²RCS Id = Id: gl2.notes.tex,v 1.10 2003/12/01 12:57:24 waratah Exp

Contents

Topic 105: Kernel	15
Objective 105.1: Manage/Query kernel and kernel modules at runtime	15
1.1 Overview	15
1.1.1 Weight: []	15
1.1.2 Statement of Objective:	15
1.1.3 Key files, terms, and utilities include:	15
1.1.4 Resources of Interest:	15
1.2 Notes	16
1.3 Lab	16
1.4 Questions	16
Objective 105.2: Reconfigure, build, and install a custom kernel and kernel modules	17
2.1 Overview	17
2.1.1 Weight: []	17
2.1.2 Statement of Objective:	17
2.1.3 Key files, terms, and utilities:	17
2.1.4 Resources of Interest:	17
2.2 Notes	18
2.3 Lab	18
2.3.1 Build a Debian Kernel	18
2.4 Questions	19
Topic 106: Boot, Initialisation, Shutdown and Runlevels	23
Objective 106.1: Boot the system	23
1.1 Overview	23
1.1.1 Weight: []	23
1.1.2 Statement of Objective:	23
1.1.3 Key files, terms, and utilities include:	23
1.1.4 Resources of Interest:	23
1.2 Notes	24
1.3 Lab	24
1.4 Questions	24

Objective 106.2: Change runlevels and shutdown or reboot system	25
2.1 Overview	25
2.1.1 Weight: []	25
2.1.2 Statement of Objective:	25
2.1.3 Key files, terms, and utilities include:	25
2.1.4 Resources of Interest:	25
2.2 Notes	26
2.3 Lab	26
2.4 Questions	26
Topic 107: Printing	29
Objective 107.1: Manage Printers and Print Queues	29
1.1 Overview	29
1.1.1 Weight: []	29
1.1.2 Statement of Objective:	29
1.1.3 Key files, terms, and utilities include:	29
1.1.4 Resources of Interest:	29
1.2 Notes	30
1.2.1 lpc - Printer Control	30
1.2.2 lpc - Printer Control	30
1.2.3 lpc Commands	30
1.2.4 lpq - Display Printer Queue	31
1.2.5 lpq - Example	31
1.2.6 lprm - Remove job(s) from Queue	31
1.2.7 lprm - Example	31
1.3 Lab	32
1.4 Questions	32
Objective 107.3: Print files	33
3.1 Overview	33
3.1.1 Weight: []	33
3.1.2 Statement of Objective:	33
3.1.3 Key files, terms, and utilities include:	33
3.1.4 Resources of Interest:	33
3.2 Notes	34
3.2.1 lpr - Submit job to Print Queue	34
3.2.2 a2ps - Convert ASCII to Postscript	34
3.2.3 mpage - Print multiple pages per page	34
3.3 Lab	35
3.4 Questions	35
Objective 107.4: Install and Configure Local and Remote Printers	37
4.1 Overview	37
4.1.1 Weight: []	37
4.1.2 Statement of Objective:	37
4.1.3 Key files, terms, and utilities include:	37
4.1.4 Resources of Interest:	37
4.2 Notes	38

4.2.1	Linux Printing	38
4.2.2	Installing a Printer	38
4.2.3	/etc/printcap - The configuration file	38
4.2.4	/etc/printcap - The configuration file	39
4.2.5	Creating spool directory & log file	39
4.2.6	Controlling printer access	40
4.2.7	Print Filters	40
4.2.8	Key Point Summary	40
4.3	Lab	40
4.4	Questions	40

Topic 108: Documentation

43

Objective 108.1: Use and manage local system documentation

43

1.1	Overview	43
1.1.1	Weight: []	43
1.1.2	Statement of Objective:	43
1.1.3	Key files, terms, and utilities include:	43
1.1.4	Resources of Interest:	43
1.2	Notes	44
1.3	Lab	44
1.4	Questions	44

Objective 108.2: Find Linux documentation on the Internet

45

2.1	Overview	45
2.1.1	Weight: []	45
2.1.2	Statement of Objective:	45
2.1.3	Key files, terms, and utilities include:	45
2.1.4	Resources of Interest:	45
2.2	Notes	46
2.3	Lab	46
2.4	Questions	46

Objective 108.5: Notify Users on System-Related Issues

47

5.1	Overview	47
5.1.1	Weight: []	47
5.1.2	Statement of Objective:	47
5.1.3	Key files, terms, and utilities include:	47
5.1.4	Resources of Interest:	47
5.2	Notes on /etc/issue and /etc/motd	48
5.2.1	Customise the Local Login screen with /etc/issue	48
5.2.2	man issue	48
5.2.3	man getty—embedded in the /etc/issue	48
5.2.4	Telnet uses /etc/issue.net	49
5.2.5	Message of the Day—motd	49
5.2.6	Manpage for motd	49
5.3	Lab on /etc/issue and /etc/motd	50
5.4	Questions	50

Topic 109: Shells, Scripting, Programming, Compiling	53
Objective 109.1: Customise and use the shell environment	53
1.1 Overview	53
1.1.1 Weight: []	53
1.1.2 Statement of Objective:	53
1.1.3 Key files, terms, and utilities include:	53
1.1.4 Resources of Interest:	53
1.2 Notes	54
1.2.1 Bash Configuration Files	54
1.2.2 Bash Aliases	54
1.2.3 Bash Functions	54
1.2.4 Function Example	54
1.2.5 Valid Function Definitions	55
1.2.6 Invalid Function Definitions	55
1.2.7 Invalid Function Definitions	55
1.2.8 Example from Jeffrey Dean's Nutshell Book	55
1.3 Lab	55
1.3.1 Exercise	55
1.4 Questions	55
Objective 109.2: Customise or write simple scripts	57
2.1 Overview	57
2.1.1 Weight: []	57
2.1.2 Statement of Objective:	57
2.1.3 Key files, terms, and utilities:	57
2.1.4 Resources of Interest:	57
2.2 Notes	59
2.3 Lab	59
2.4 Questions	59
Topic 111: Administrative Tasks	63
Objective 111.1: Manage users and group accounts and related system files	63
1.1 Overview	63
1.1.1 Weight: []	63
1.1.2 Statement of Objective:	63
1.1.3 Key files, terms, and utilities include:	63
1.1.4 Resources of Interest:	63
1.2 Notes	64
1.3 Lab	64
1.4 Questions	64
Objective 111.2: Tune the user environment and system environment variables	65
2.1 Overview	65
2.1.1 Weight: []	65
2.1.2 Statement of Objective:	65

2.1.3	Key files, terms, and utilities include:	65
2.1.4	Resources of Interest:	65
2.2	Notes	66
2.3	Lab	66
2.4	Questions	66
Objective 111.3: Configure and use system log files to meet administrative and security needs		67
3.1	Overview	67
3.1.1	Weight: []	67
3.1.2	Statement of Objective:	67
3.1.3	Key files, terms, and utilities include:	67
3.1.4	Resources of Interest:	67
3.2	Notes	68
3.3	Lab	68
3.4	Questions	68
Objective 111.4: Automate system administration tasks by scheduling jobs to run in the future		69
4.1	Overview	69
4.1.1	Weight: []	69
4.1.2	Statement of Objective:	69
4.1.3	Key files, terms, and utilities include:	69
4.1.4	Resources of Interest:	69
4.2	Notes on Using <code>at</code> and <code>cron</code>	70
4.2.1	The <code>at</code> command	70
4.2.2	Example <code>at</code> time specifications	70
4.2.3	Queued jobs	70
4.2.4	<code>crontab</code>	70
4.2.5	<code>crontab</code> file format	71
4.2.6	<code>cron</code> from root	71
4.2.7	<code>anacron</code>	71
4.3	Exercises Using <code>at</code> and <code>cron</code>	72
4.3.1	Backups	72
4.3.2	Merry Christmas	72
4.3.3	Happy Easter	72
4.3.4	Backups again	72
4.3.5	Watch for Evil Longhairs	72
4.3.6	Every Easter	72
4.4	Solutions for Exercises Using <code>at</code> and <code>cron</code>	73
4.4.1	Backups	73
4.4.2	Merry Christmas	73
4.4.3	Happy Easter	73
4.4.4	Backups again	73
4.4.5	Watch for Evil Longhairs	73
4.4.6	Every Easter	73
4.5	Lab	74
4.6	Questions	74

Objective 111.5: Maintain an effective data backup strategy	75
5.1 Overview	75
5.1.1 Weight: []	75
5.1.2 Statement of Objective:	75
5.1.3 Key files, terms, and utilities include:	75
5.1.4 Resources of Interest:	75
5.2 Notes	76
5.2.1 Backup Overview	76
5.2.2 Backup & Restore methods	77
5.2.3 Software	79
5.2.4 Rotation & off-site strategies	79
5.3 Lab	80
5.4 Questions	80
Objective 111.6: Maintain system time	81
6.1 Overview	81
6.1.1 Weight: [4]	81
6.1.2 Statement of Objective:	81
6.1.3 Key files, terms, and utilities include:	81
6.1.4 Resources of Interest:	81
6.2 Notes	83
6.2.1 Display or Set System Date & Time: <code>date</code>	83
6.2.2 The Hardware Clock: <code>hwclock</code>	84
6.2.3 NTP - Network Time Protocol	84
6.2.4 Quick <code>ntp</code> install guide	85
6.2.5 <code>ntpdate</code> - Set system time & date	85
6.2.6 <code>ntpd</code> - The NTP daemon	86
6.2.7 <code>ntpd</code> usage & configuration	86
6.3 Lab	87
6.3.1 Explore the <code>ntp</code> documentation	87
6.3.2 Use the <code>date</code> command	87
6.3.3 Use the <code>hwclock</code> command	88
6.3.4 Explore the <code>ntp</code> family of commands	88
6.3.5 Setup <code>ntp</code>	88
6.4 Questions	88
Topic 112: Networking Fundamentals	91
Objective 112.1: Fundamentals of TCP/IP	91
1.1 Overview	91
1.1.1 Weight: []	91
1.1.2 Statement of Objective:	91
1.1.3 Key files, terms, and utilities include:	91
1.1.4 Resources of Interest:	91
1.2 Notes	93
1.3 Lab	93
1.4 Questions	93

Objective 112.3: TCP/IP configuration and troubleshooting	95
3.1 Overview	95
3.1.1 Weight: []	95
3.1.2 Statement of Objective:	95
3.1.3 Key files, terms, and utilities include:	95
3.1.4 Resources of Interest:	96
3.2 Notes	97
3.3 Lab	97
3.4 Questions	97
Objective 112.4: Configure Linux as a PPP client	99
4.1 Overview	99
4.1.1 Weight: []	99
4.1.2 Statement of Objective:	99
4.1.3 Key files, terms, and utilities include:	99
4.1.4 Resources of Interest:	99
4.2 Notes	100
4.3 Lab	100
4.4 Questions	100
Topic 113: Networking Services	103
Objective 113.1: Configure and manage inetd, xinetd, and related services	103
1.1 Overview	103
1.1.1 Weight: []	103
1.1.2 Statement of Objective:	103
1.1.3 Key files, terms, and utilities include:	103
1.1.4 Resources of Interest:	103
1.2 Notes	104
1.3 Lab	104
1.4 Questions	104
Objective 113.2: Operate and perform basic configuration of sendmail	105
2.1 Overview	105
2.1.1 Weight: []	105
2.1.2 Statement of Objective:	105
2.1.3 Key files, terms, and utilities include:	105
2.1.4 Resources of Interest:	105
2.2 Notes	106
2.3 Lab	106
2.4 Questions	106
Objective 113.3: Operate and perform basic configuration of Apache	107
3.1 Overview	107
3.1.1 Weight: []	107
3.1.2 Statement of Objective:	107
3.1.3 Key files, terms, and utilities include:	107
3.1.4 Resources of Interest:	107

3.2	Notes	108
3.2.1	Apache	108
3.2.2	Starting & Stopping Apache	108
3.2.3	apachectl	108
3.2.4	HTTPD Parameters	109
3.2.5	HTTPD Parameters	109
3.2.6	Configuring Apache	109
3.2.7	Site-wide Directives	109
3.2.8	Directory block Directives	110
3.2.9	Access Control	110
3.2.10	Other Directives	110
3.3	Lab	111
3.4	Questions	111
Objective 113.4: Properly manage the NFS, smb, and nmb daemons		113
4.1	Overview	113
4.1.1	Weight: []	113
4.1.2	Statement of Objective:	113
4.1.3	Key files, terms, and utilities include:	113
4.1.4	Resources of Interest:	113
4.2	Notes	114
4.3	Lab	114
4.4	Questions	114
Objective 113.5: Setup and Configure Basic DNS Services		115
5.1	Overview	115
5.1.1	Weight: []	115
5.1.2	Statement of Objective:	115
5.1.3	Key files, terms, and utilities include:	115
5.1.4	Resources of Interest:	115
5.2	Notes	116
5.2.1	Setup and Configure basic DNS services	116
5.2.2	Setup and Configure basic DNS services	116
5.2.3	DNS - DOMAIN NAME SERVICE	116
5.2.4	DNS - DOMAIN NAME SERVICE	116
5.2.5	RESOLVING A NAME	117
5.2.6	The nsswitch.conf file	117
5.2.7	The nsswitch.conf file	117
5.2.8	An example nsswitch file:	118
5.2.9	The resolv.conf file	118
5.2.10	BIND - Berkley Internet Name Domain	118
5.2.11	BIND Configuration	119
5.2.12	An Example Config file:	119
5.2.13	Zone files:	119
5.2.14	Zone Records:	120
5.2.15	Example Forward file /var/named/aes.zone	120
5.2.16	Example reverse file /var/named/1.27.10.in-addr.arpa.zone	121
5.2.17	Configuring a Caching only Nameserver	121
5.2.18	Testing DNS	121
5.2.19	nslookup	122

CONTENTS	11
5.2.20 dig	122
5.2.21 host	122
5.2.22 Exercise:	123
5.3 Lab	123
5.4 Questions	123
Objective 113.7: Set up secure shell (OpenSSH)	125
7.1 Overview	125
7.1.1 Weight: []	125
7.1.2 Statement of Objective:	125
7.1.3 Key files, terms, and utilities include:	125
7.1.4 Resources of Interest:	125
7.2 Notes on ssh	126
7.2.1 Versions	126
7.2.2 Commands	126
7.2.3 ssh Commands	126
7.2.4 scp Commands	126
7.2.5 sftp Commands	127
7.2.6 Advanced Usage	127
7.3 Lab	128
7.4 Questions	128
Topic 114: Security	131
Objective 114.1: Perform security administration tasks	131
1.1 Overview	131
1.1.1 Weight: []	131
1.1.2 Statement of Objective:	131
1.1.3 Key files, terms, and utilities include:	131
1.1.4 Resources of Interest:	131
1.2 Notes	132
1.3 Lab	132
1.4 Questions	132
Objective 114.2: Setup host security	133
2.1 Overview	133
2.1.1 Weight: []	133
2.1.2 Statement of Objective:	133
2.1.3 Key files, terms, and utilities include:	133
2.1.4 Resources of Interest:	133
2.2 Notes	134
2.3 Lab	134
2.4 Questions	134
Objective 114.3: Setup user level security	135
3.1 Overview	135
3.1.1 Weight: [2]	135
3.1.2 Statement of Objective:	135
3.1.3 Key files, terms, and utilities include:	135

3.1.4	Resources of Interest:	135
3.2	Set and View Disk Quotas	136
3.2.1	Enabling Quotas	136
3.2.2	Quota Limits	137
3.2.3	Setting up and configuring quotas.	137
3.2.4	Quota commands	138
3.2.5	139
3.2.6	139
3.2.7	139
3.2.8	139
3.3	Lab	139
3.4	Questions	139
List of topics		141
.1	Topics moved to General Linux 1	142

Topic 105

Kernel

Objective 105.1

Manage/Query kernel and kernel modules at runtime

1.1 Overview

1.1.1 Weight: []

1.1.2 Statement of Objective:

Candidates should be able to manage and/or query a kernel and kernel loadable modules. This objective includes using command-line utilities to get information about the currently running kernel and kernel modules. It also includes manually loading and unloading modules as appropriate. It also includes being able to determine when modules can be unloaded and what parameters a module accepts. Candidates should be able to configure the system to load modules by names other than their file name.

1.1.3 Key files, terms, and utilities include:

```
/lib/modules/kernel-version/modules.dep  
/etc/modules.conf & /etc/conf.modules  
depmod  
insmod  
lsmod  
rmmod  
modinfo  
modprobe  
uname
```

1.1.4 Resources of Interest:

TBA

1.2 Notes

1.3 Lab

1.4 Questions

Objective 105.2

Reconfigure, build, and install a custom kernel and kernel modules

2.1 Overview

2.1.1 Weight: []

2.1.2 Statement of Objective:

Candidates should be able to customise, build, and install a kernel and kernel loadable modules from source. This objective includes customising the current kernel configuration, building a new kernel, and building kernel modules as appropriate. It also includes installing the new kernel as well as any modules, and ensuring that the boot manager can locate the new kernel and associated files (generally located under /boot, see objective 1.102.2 for more details about boot manager configuration).

2.1.3 Key files, terms, and utilities:

```
/usr/src/linux/*
/usr/src/linux/.config
/lib/modules/kernel-version/*
/boot/*
make
make targets: config, menuconfig, xconfig, oldconfig, modules, install, modules_ins
```

2.1.4 Resources of Interest:

TBA

2.2 Notes

2.3 Lab

2.3.1 Build a Debian Kernel

Catch a fresh kernel

- Kernel sources are large so if you already have a recent kernel it is best to obtain patches to bring it up to date.
- Fresh kernels are to be had from

`http://www.kernel.org`

be sure to use a mirror near you.

- Place your kernel in a suitable place:
 - Under `/usr/src/` Note that this directory has group owner of `src`, so make yourself a member of the `src` group:

```
# usermod -G src fred ↔
```

```
- /tmp
```

```
- /home/fred
```

apt-get the Necessary Tools

```
# apt-get install debhelper modutils libncurses5-dev build-essential fakeroot ↔
```

Edit `/etc/pkg-source.conf`

```
# vi /etc/kernel-pkg.conf ↔
```

Unarchive the Source

```
$ cd /usr/src/ ↔
```

```
$ tar zxvf linux-2.5.34.tar.gz ↔
```

```
$ cd linux-2.5.34 ↔
```

Use Your Existing `.config` as a Starting Place

```
$ cp /boot/config-2.4.18-386 .config ↔ #YMMV
```

Configure Your Kernel

```
$ make oldconfig ↔
```

```
$ make xconfig ↔ #or menuconfig or config
```

2.4 Questions

Topic 106

Boot, Initialisation, Shutdown and Runlevels

Objective 106.1

Boot the system

1.1 Overview

1.1.1 Weight: []

1.1.2 Statement of 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.

1.1.3 Key files, terms, and utilities include:

```
dmesg /var/log/messages  
/etc/conf.modules or /etc/modules.conf  
LILO GRUB
```

1.1.4 Resources of Interest:

TBA

1.2 Notes

1.3 Lab

1.4 Questions

Objective 106.2

Change runlevels and shutdown or reboot system

2.1 Overview

2.1.1 Weight: []

2.1.2 Statement of 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.

2.1.3 Key files, terms, and utilities include:

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

2.1.4 Resources of Interest:

TBA

2.2 Notes

2.3 Lab

2.4 Questions

Topic 107

Printing

Objective 107.1

Manage Printers and Print Queues

1.1 Overview

1.1.1 Weight: []

1.1.2 Statement of Objective:

The candidate should be able to manage print queues and user print jobs. This objective includes monitoring print server and user print queues and troubleshooting general printing problems

1.1.3 Key files, terms, and utilities include:

lpc
lpq
lprm
lpr
/etc/printcap

1.1.4 Resources of Interest:

Printing-HOWTO

Printing-Usage-HOWTO

1.2 Notes

1.2.1 `lpc` - Printer Control

- `lpc` Is used to control a printer or print job
- Can be run interactively.
- Usage is:

```
lpc [command [argument]]
```

1.2.2 `lpc` - Printer Control

Example 1 - Non Interactive:

```
# lpc status ↵
Printer  Printing Spooling Jobs  Server Subserver
lp@Node4  enabled  enabled    0   none   none
```

Example 2 - Interactive

```
# lpc ↵
lpc> status
Printer  Printing Spooling Jobs  Server Subserver
lp@Node4  enabled  enabled    0   none   none
lpc> quit
```

1.2.3 `lpc` Commands

For a complete list of commands, use the command `lpc help`. Some of the more important commands to know are:

abort - Immediately terminate active spool & disable printing

disable - Stop spooling for this printer

enable - Start spooling for this printer

down - Disable spooling & printing

up - Enable spooling & printing

stop - Stop printing after current job is complete

start - Enable spooling & start printing

quit - Exit from interactive mode

help - Show all commands

1.2.4 lpq - Display Printer Queue

- Every print job is assigned a *job-id*
- You need the job-id to remove or reorder a job in the queue
- lpq shows the job-id along with information about the job.

Usage is:

```
# lpq [-P printer] [Job-id]
```

1.2.5 lpq - Example

Example - Show all jobs on default Queue

```
# lpq
Printer: lp@Node4 'lp0' (dest HPLjet@node10.aes)
Queue: 2 printable jobs
Server: pid 27354 active
Unspooler: pid 27356 active
Status: waiting for subserver to exit at 12:22:58.553
Rank  Owner/ID          Class Job Files  ....
1     root@Node4+353     A    353 /etc/hosts  ....
2     root@Node4+357     A    357 /etc/ntp.conf ....
Printer: HPLjet@Node10 'lp0' (printing disabled ...)
```

1.2.6 lprm - Remove job(s) from Queue

- lprm is used to remove jobs from a queue
- jobs can be removed by:
 - by job-id (use lpq to find out)
 - by user

Usage is:

```
# lprm [-P printer] [Job-id ...] [user ...]
```

1.2.7 lprm - Example

Example - Remove all jobs owned by root

```
# lprm root
Printer lp@Node4:
  checking perms 'root@Node4+353'
  dequeued 'root@Node4+353'
  checking perms 'root@Node4+357'
  dequeued 'root@Node4+357'
Printer HPLjet@Node10:
# lpq
```

1.3 Lab

1. Use a GUI printer configuration tool such as Red Hat's `printtool` to set up a local generic postscript printer. Make it the default.
2. Login as at least 2 different users. Have each user print some documents. As there is no printer attached these should remain queued.
3. View the queue as a normal user and try to dequeue your own print jobs and the print jobs belonging to others.
4. Login as root:
 - (a) View the queue with `lpq`.
 - (b) Use `lpc` to reorder the queue.
 - (c) Use `lprm` to remove some items from the queue.
5. Delete all print jobs from the print queue.
6. Restart `lpd`.

1.4 Questions

Objective 107.3

Print files

3.1 Overview

3.1.1 Weight: []

3.1.2 Statement of Objective:

Candidates should be able to manage print queues and manipulate print jobs. This objective includes adding and removing jobs from configured printer queues and converting text files to postscript for printing.

3.1.3 Key files, terms, and utilities include:

lpr
lpq
mpage

3.1.4 Resources of Interest:

Printing-HOWTO

Printing-Usage-HOWTO

3.2 Notes

3.2.1 lpr - Submit job to Print Queue

- lpr is used to submit a job to the print queue
- lpr can be run from the command line
- lpr assumes text is to be printed by default
- lpr uses `/etc/printcap` for printer settings

Usage of lpr is:

```
lpr [-Pprinter] [-#num] filename ...
```

Example:- Print `/etc/hosts` file to spool lp0

```
$ lpr -Plp0 /etc/hosts ↵
```

3.2.2 a2ps - Convert ASCII to Postscript

- Most Unix sites use postscript printers
- You can not send an ASCII file directly to a postscript printer
- To convert from ASCII to ps, use a2ps

Usage:

```
a2ps [-o out-file] in-file
```

Example:

```
$ a2ps -o myfile.ps myfile.txt ↵
```

```
$ cat myfile.ps > /dev/lp0 ↵
```

3.2.3 mpage - Print multiple pages per page

- mpage will print multiple pages per physical page
- Input can be either text or postscript
- 1, 2, 4 or 8 pages can be printed per page
- Output can be directed to printer or stdout

Usage:

```
mpage [-1248] [options] input-file ...
```

Example: Print 4 pages per page on A4

```
$ mpage -4 -b A4 bigfile.ps | lpr ←
```

3.3 Lab

3.4 Questions

Objective 107.4

Install and Configure Local and Remote Printers

4.1 Overview

4.1.1 Weight: []

4.1.2 Statement of Objective:

Candidate should be able to install a printer daemon, install and configure a print filter (e.g.: `apsfilter`, `magicfilter`). This objective includes making local and remote printers accessible for a Linux system, including postscript, non-postscript, and Samba printers.

4.1.3 Key files, terms, and utilities include:

```
lpd
/etc/printcap
/etc/apsfilter/*
/var/lib/apsfilter/*/
/etc/magicfilter/*/
/var/spool/lpd/*/
```

4.1.4 Resources of Interest:

Printing-HOWTO

Printing-Usage-HOWTO

www.linuxprinting.org

4.2 Notes

4.2.1 Linux Printing

- There are several packages available for linux printing:
 - LPR
 - LPRng
 - Cups
- LPR (or LPRng) is the default on most Linux distros
- Major components of the LPR subsystem are:
 - `lpd` - The printing daemon
 - `lpr` - A tool to submit jobs into the queue
 - `lprm` - A tool to remove jobs from the queue
 - `lpq` - A tool to view jobs in the queue
 - `lpc` - An administration tool for printers & queues

4.2.2 Installing a Printer

- There are two ways to install a printer under Linux:
- The easy way! - Use a GUI like `printtool`
- The hard way:
 - Edit `/etc/printcap`
 - Create the spool directory
 - Touch the log file
 - Restart `lpd`

4.2.3 `/etc/printcap` - The configuration file

`/etc/printcap` contains information about *all* printers on the system (including remote printers)

An example looks like:

```
HPLjet |lp|lp0:\
    :ml=0:\
    :mx=0:\
    :sd=/var/spool/lpd/HPLjet:\
    :sh:\
    :lp=/dev/lp0:\
    :lf=/var/spool/lpd/HPLjet/log:\
    :if=/usr/share/printconf/util/mf_wrapper:
```

4.2.4 /etc/printcap - The configuration file

Key points to note about `printcap` format:

- Comments start with a '#'
- Any line not starting with a colon or pipe is the start of a printer definition
- Each line of a definition ends in a backslash except the last line
- `lpd` must be restarted each time `/etc/printcap` is edited
- Spool directory & log file must be created manually

`if` Define the input filter

`lf` Define the printer log file

`lo` Define the lock file created when printer is in use

`mx` Define the maximum size of a print job

`rm` Specify printer is on remote machine. Eg : `rm=192.168.222.254:`

`rp` Define remote printer name. Eg : `rp=HPLjet:`

`sh` Tell `lpd` not to print banner pages

`sd` Specify spool directory

4.2.5 Creating spool directory & log file

The spool directory should be owned by `lp` and have permissions set to 700:

- `# mkdir /var/spool/lpd/HPLjet ←`
- `# chown lp:lp /var/spool/lpd/HPLjet ←`
- `# chmod 0700 /var/spool/lpd/HPLjet ←`

The log file should have permissions set to 666 and have the same ownership as the spool directory:

- `# touch /var/spool/lpd/HPLjet/log ←`
- `# chown lp:lp /var/spool/lpd/HPLjet/log ←`
- `# chmod 0666 /var/spool/lpd/HPLjet/log ←`

4.2.6 Controlling printer access

- Printer access is controlled through `/etc/hosts.lpd`
- If the file does not exist, all access is granted
- If the file exists, only those in the list will be granted access
- The format is: `[host [user]]`

Example: All access from `box2.c222`, only greg from `box3.c222`

```
box2.c222
box3.c222 greg
```

4.2.7 Print Filters

- A print filter converts data to be printed into a language that your printer understands
- There are several print filter packages:
 - Apsfilter
 - Magicfilter
 - Red Hat's PrintTool
 - Foomatic

4.2.8 Key Point Summary

- Most Linux Systems use LPR (or LPRng)
- Local & remote printer configs are stored in `/etc/printcap`
- The print spool directory & log file must be created manually
- Print access is controlled using `/etc/hosts.lpd`
- Print filters convert different data types to a language understood by the printer
- The `lpd` daemon is responsible for getting jobs from the user, putting them through the filter and delivering them to the spool directory.
- The `lpc` program is used to control the printer and print spools
- The `lpq` program is used to view the print queues
- The `lprm` program is used to remove jobs from the queues
- The `lpr` program is used to submit jobs into the queue.

4.3 Lab

4.4 Questions

Topic 108

Documentation

Objective 108.1

Use and manage local system documentation

1.1 Overview

1.1.1 Weight: []

1.1.2 Statement of 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 `/usr/share/doc/` and determining what documentation to keep in `/usr/share/doc/`.

1.1.3 Key files, terms, and utilities include:

1.1.4 Resources of Interest:

TBA

1.2 Notes

1.3 Lab

1.4 Questions

Objective 108.2

Find Linux documentation on the Internet

2.1 Overview

2.1.1 Weight: []

2.1.2 Statement of 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.

2.1.3 Key files, terms, and utilities include:

2.1.4 Resources of Interest:

TBA

2.2 Notes

2.3 Lab

2.4 Questions

Objective 108.5

Notify Users on System-Related Issues

5.1 Overview

5.1.1 Weight: []

5.1.2 Statement of 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.

5.1.3 Key files, terms, and utilities include:

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

5.1.4 Resources of Interest:

TBA

5.2 Notes on `/etc/issue` and `/etc/motd`

5.2.1 Customise the Local Login screen with `/etc/issue`

- The login screen for RH73 looks like this:

```
Red Hat Linux release 7.3 (Valhalla)
Kernel 2.4.18-3 on an i686
```

```
login:
```

- The `getty` process spawned by `init` presents the contents of the file `/etc/issue` and provides a `login:` prompt for the user.
- The file `/etc/issue`

```
$ cat /etc/issue ←
Red Hat Linux release 7.3 (Valhalla)
Kernel \r on an \m
```

5.2.2 `man issue`

```
ISSUE(5)                Linux Programmer's Manual                ISSUE(5)

NAME
    issue - pre-login message and identification file

DESCRIPTION
    The file /etc/issue is a text file which contains a message or
    system identification to be printed before the login prompt. It
    may contain various @char and \char sequences, if supported by
    getty(1).

FILES
    /etc/issue

SEE ALSO
    getty(1), motd(5)

Linux
1993-07-24                ISSUE(5)
```

5.2.3 `man getty`—embedded in the `/etc/issue`

```
\d    insert current day (localtime),
\l    insert line on which mingetty is running,
\m    inserts machine architecture (uname -m),
\n    inserts machine's network node hostname (uname -n),
\o    inserts domain name,
\r    inserts operating system release (uname -r),
\t    insert current time (localtime),
\s    inserts operating system name,
\u    resp. \U the current number of users which are currently
      logged in. \U inserts "n users", where as \u only inserts "n".
\v    inserts operating system version (uname -v).
```


5.2.4 Telnet uses /etc/issue.net

The following sequences are supported by telnetd:

```
%l      - show the current tty
%h, %n  - show the system node name (FQDN)
%D, %o  - show the name of the NIS domain
%d, %t  - show the current time and date
%s      - show the name of the operating system
%m      - show the machine (hardware) type
%r      - show the operating system release
%v      - show the operating system version
%%      - display a single '%' character
```

5.2.5 Message of the Day—motd

Login Sequence

- Contents of /etc/issue is displayed.
- getty displays the login prompt.
- /bin/login handles the login process.
 - User is authorised.
 - Contents of /etc/motd is displayed.
 - The login shell is executed.

5.2.6 Manpage for motd

```
MOTD(5)                Linux Programmer's Manual                MOTD(5)

NAME
    motd - message of the day

DESCRIPTION
    The contents of /etc/motd are displayed by login(1) after a successful login but just before it executes the login shell.

    The abbreviation "motd" stands for "message of the day", and this file has been traditionally used for exactly that (it requires much less disk space than mail to all users).

FILES
    /etc/motd

SEE ALSO
    login(1), issue(5)

Linux                1992-12-29                MOTD(5)
```

5.3 Lab on `/etc/issue` and `/etc/motd`

1. Before altering any of the system files back them up:

```
# cp issue issue.orig ↵
# cp issue.net issue.net.orig ↵
# cp motd motd.orig ↵
```

2. Edit `/etc/issue` and place a suitable message there. Try out some of the embedded codes from the man page such as `\t`. Login from another virtual terminal and check your results.
3. Edit `/etc/issue.net` and place a suitable message there. Try out some of the embedded codes from the man page such as `%t`. Telnet to you host from another system and check your results.
4. Edit `/etc/motd` and place a suitable message there. Login from another virtual terminal and check your results.
5. Replace the system files with the original versions:

```
# cp issue.orig issue ↵
# cp issue.net.orig issue.net ↵
# cp motd.orig motd ↵
```

6. Login and check that the systems are “as installed”.

5.4 Questions

Topic 109

**Shells, Scripting,
Programming, Compiling**

Objective 109.1

Customise and use the shell environment

1.1 Overview

1.1.1 Weight: []

1.1.2 Statement of Objective:

Candidate should be able to customise shell environments to meet users' needs. This objective includes setting environment variables (e.g. PATH) at login or when spawning a new shell. It also includes writing bash functions for frequently used sequences of commands.

1.1.3 Key files, terms, and utilities include:

```
~/.bash_profile
~/.bash_login
~/.profile
~/.bashrc
~/.bash_logout
~/.inputrc
function (Bash built-in command)
export
env
set (Bash built-in command)
unset (Bash built-in command)
```

1.1.4 Resources of Interest:

TBA

1.2 Notes

1.2.1 Bash Configuration Files

- When a user logs in to a bash shell the following configuration files are usually executed:

`/etc/profile` System wide profile, common to all users and shells

`~/ .bash_profile` Executed after `/etc/profile` at login

`~/ .bashrc` Executed after `/ .bash_profile` at login

- Note `/ .bashrc` is executed when any new bash shell is spawned

1.2.2 Bash Aliases

-

1.2.3 Bash Functions

- Functions work similarly to aliases but allow more complex constructions.
- They have the following syntax:

```
$ [ function ] NAME () { COMMAND_LIST; } ←
```

- Where

`function` Optional tag

`NAME ()` The name of the function

`COMMAND_LIST` The body of the function

- Functions may be stored in `~/ .bashrc`

1.2.4 Function Example

- This simple function prints the current working directory and the list of files in it:

```
$ function look () { pwd; ls; } ←
```

- This function would be used like this:

```
$ look ←
/home/geoffrey/lpic/general-linux-2/notes
CVS      _whizzy_gl2.notes.fmt
_whizzy_gl2.notes.pag
```

1.2.5 Valid Function Definitions

- `$ function look() { pwd; ls; }`
- `$ function look { pwd; ls; }`
- `$ look() { pwd; ls; }`
- `$ look()`

```
> {
> pwd;
> ls;
> }
```

1.2.6 Invalid Function Definitions

- `$ function look() pwd; ls;`
- `$ look() { pwd; ls }`
- `$ function look() {pwd; ls;}`

1.2.7 Invalid Function Definitions

- `$ function look() pwd; ls;`
- `$ look() { pwd; ls }`
- `$ function look() {pwd; ls;}`

1.2.8 Example from Jeffrey Dean's Nutshell Book

- `$ laps () { ←`

```
> ls -l $1
> ps aux | grep ` /usr/bin/basename $1 `
> }
```

- **Use the `laps ()` function:**

```
$ laps /usr/sbin/sshd ←
-rwxr-xr-x  1 root      root      276200 Jun 29 01:28 /usr/sbin/sshd
root      255  0.0  0.3  2792 1216 ?        S      Aug31   0:00 /usr/sbin/sshd
geoffrey  1187  0.0  0.1  1332  424 pts/1    R      14:39   0:00 grep sshd
```

1.3 Lab

1.3.1 Exercise

1.4 Questions

Objective 109.2

Customise or write simple scripts

2.1 Overview

2.1.1 Weight: []

2.1.2 Statement of Objective:

Candidate should be able to customise existing scripts, or write simple new (ba)sh scripts. This objective includes using standard sh syntax (loops, tests), using command substitution, testing command return values, testing of file status, and conditional mailing to the superuser. This objective also includes making sure the correct interpreter is called on the first (!) line of scripts. This objective also includes managing location, ownership, execution and suid-rights of scripts.

2.1.3 Key files, terms, and utilities:

```
while  
for  
test  
chmod
```

2.1.4 Resources of Interest:

1. Bash Programming Introduction (LDP HOWTO)

<http://www.linux.org/docs/ldp/howto/Bash-Prog-Intro-HOWTO.html>

2. Linux Shells by Example

by Ellie Quigley

ISBN 0-13-0141711-7

Prentice Hall

3. LINUX & UNIX Shell Programming

David Tansley

ISBN 0-201-67472-6

ADDISON-WESLEY

2.2. *NOTES*

59

2.2 Notes

2.3 Lab

2.4 Questions

Topic 111

Administrative Tasks

Objective 111.1

Manage users and group accounts and related system files

1.1 Overview

1.1.1 Weight: []

1.1.2 Statement of 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.

1.1.3 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

1.1.4 Resources of Interest:

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

Manpages useradd usermod userdel groupadd groupmod groupdel useradd
passwd chage

1.2 Notes

1.3 Lab

1.4 Questions

Objective 111.2

Tune the user environment and system environment variables

2.1 Overview

2.1.1 Weight: []

2.1.2 Statement of 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.

2.1.3 Key files, terms, and utilities include:

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

2.1.4 Resources of Interest:

TBA

2.2 Notes

2.3 Lab

2.4 Questions

Objective 111.3

Configure and use system log files to meet administrative and security needs

3.1 Overview

3.1.1 Weight: []

3.1.2 Statement of 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.

3.1.3 Key files, terms, and utilities include:

3.1.4 Resources of Interest:

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

3.2 Notes

3.3 Lab

3.4 Questions

Objective 111.4

Automate system administration tasks by scheduling jobs to run in the future

4.1 Overview

4.1.1 Weight: []

4.1.2 Statement of 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.

4.1.3 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/*
```

4.1.4 Resources of Interest:

TBA

4.2 Notes on Using `at` and `cron`

Notes from a talk by Angus Lees

4.2.1 The `at` command

`at` takes a time and a list of commands to run. Any output to `STDOUT` or `STDERR` will be mailed to the user running `at`.

```
$ at 2pm ↵
warning: commands will be executed using /bin/sh
at> date ↵
at> ^D ↵
job 3 at 2002-05-08 14:00
```

The current `umask`, working directory and environment (except for `TERM`, `DISPLAY` and `.`) are saved and restored before running the job (unlike `cron`).

The commands to run will be read from `STDIN` or from a file given with `-f`.

4.2.2 Example `at` time specifications

`at` allows a *very* flexible time format.

`17:36` Run at 5:36pm today or tomorrow.

`9pm May 8` Run at 9pm on May 8th.

`noon tomorrow` Run at 12pm tomorrow.

`now + 2 hours` Run in 2 hours.

See [at\(1\)](#) for more details.

4.2.3 Queued jobs

`atq` lists a user's pending jobs.

```
$ atq ↵
3 2002-05-08 14:00 a gus
```

`$ atrm 3` ↵ removes the queued job.

`$ at -c 3` ↵ dumps the job on `STDOUT`.

4.2.4 `crontab`

`cron` is a daemon that reads everyone's `crontab` information, spawning new tasks at the appropriate times.

`crontab file` Replace your `crontab` file with *file*.

`crontab -l` List your `crontab`.

`crontab -r` Remove your `crontab`.

`crontab -e` Edit your `crontab` (with `$EDITOR`).

4.2.5 crontab file format

A sample crontab file:

```
0 7 1 mon jan *
echo "sleep in, you dont feel so good"

# gratuitous noise
0 17 * * mon,wed,fri          wall%meeting in 5 minutes%

0 9-18 * * mon-fri           $HOME/bin/cron.bihourly
```

4.2.6 cron from root

A few extra issues arise when editing `/etc/crontab` (and similar “system” crontab files):

- Don’t use `crontab -e`, edit `/etc/crontab` directly.
- A new column (after timespec, before command) gives the user to the command run as.
- Distributions often create directories for “common” frequencies. It usually makes much more sense to place a script in there, rather than adding your own crontab lines.
Debian (for example) runs any scripts in `/etc/cron.{daily,weekly,monthly}` – but these are triggered from normal entries in `/etc/crontab`, so there’s no real mystery here.
- (*Debian specific?*) `/etc/cron.d/*` is read in addition to `/etc/crontab` (they also have the extra user field).

4.2.7 anacron

Apparently some people turn their machines off.

If your computer is always turned off at night (for example), then daily jobs which are usually scheduled to run in the wee hours, will never be run. This is a problem.

`anacron` fixes this by running any missed jobs after a reboot (or other times, like AC-on for laptops).

Since `anacron` can’t use the crontab files, it has its own simplified `/etc/anacrontab`.

If you only use the standard `/etc/cron.daily, monthly, weekly`, then no further configuration will be necessary. Otherwise, see [anacrontab\(5\)](#).

Note that the frequency of anacron jobs can only be specified in days.

4.3 Exercises Using `at` and `cron`

(Contributed by Angus Lees)

Solve the following problems using either `at` or `cron`.

4.3.1 Backups

Backup your home directory every Saturday night.

A simple `tar` to a fixed filename is fine.

4.3.2 Merry Christmas

Send a "Merry Christmas" email to yourself at midday every Christmas day.

4.3.3 Happy Easter

Send a "Happy Easter" email to yourself at 10am on the next Easter Sunday.

(`ncal -e` will give you the date of Easter Sunday)

4.3.4 Backups again

Backup your home directory at 10pm every week night.

4.3.5 Watch for Evil Longhairs

Every 5 minutes during business hours, check the process list for any processes named "crack". Email yourself the appropriate lines.

4.3.6 Every Easter

Send yourself an email at midday on *every* Easter Sunday.

(Answers on next page. No peeking)

4.4 Solutions for Exercises Using at and cron

4.4.1 Backups

```
# backup home directory at 10pm every saturday
0 20 * * sat    tar czf /tmp/$USER.tar.gz $HOME
```

Don't be that predictable in real life though, since people could create a `/tmp/$yourusername.tar.gz` symlink pointing to your carefully archived email, and your cronjob would overwrite your email every night!

Presumably in real life, you would be backing up to somewhere safer than `/tmp` anyway.

4.4.2 Merry Christmas

```
# email cards are so much cheaper..
0 12 25 12 *   echo "Merry Christmas"
```

Note that there's no need to call `mail` (or similar), since cron will mail us the command output anyway.

4.4.3 Happy Easter

Next Easter Sunday is 20th April, 2003.

```
$ at 10am April 20
at> echo "Happy Easter"
at> ^D
```

Again, note that `at` itself will mail us any output from the command.

4.4.4 Backups again

```
# backup home directory at 10pm every week night
0 20 * * mon-fri    tar czf /tmp/$USER.tar.gz $HOME
```

Note the earlier warning about writing to predictable filenames in `/tmp`.

4.4.5 Watch for Evil Longhairs

```
# check every 5 minutes, 9-5 week days
*/5 9-17 * * mon-fri    ps aux | grep crack
```

4.4.6 Every Easter

Easter Sunday keeps moving, hence we can't just set a cron job. What we *can* do is create an `at` job that will reschedule itself automatically.

Create a shell script somewhere, lets call it `$HOME/easter.sh`, containing:

```
#!/bin/sh

# payload
echo "Happy Easter"

# date of *next* easter sunday
nextyear=$(expr $(date +%Y) + 1)
nexteaster=$(ncal -e $nextyear)

# massage "20 April 2003" into "April 20 2003"
atdate=$(echo $nexteaster | awk '{print $2,$1,$3}')

# reschedule ourselves
echo $0 | at $atdate
```

Note that this is fairly fragile: if something goes wrong, it will not reschedule and you'll never know.

It would be arguably better to just write a simple shell loop that will schedule "echo Happy Easter" for the next 100 Easter Sundays.

4.5 Lab

4.6 Questions

Objective 111.5

Maintain an effective data backup strategy

5.1 Overview

5.1.1 Weight: []

5.1.2 Statement of 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.

5.1.3 Key files, terms, and utilities include:

`cpio dd dump restore tar`

5.1.4 Resources of Interest:

TBA

5.2 Notes

5.2.1 Backup Overview

Prepared by Grant Parnell

Decide what data is important and how long you can do without it.

- Is this used 24 x 7 or just business hours?
- During business hours how long can you do without it? 4 hours, 30 minutes, 5 minutes?
- How up-to-date is it required to get you running in an emergency?
- Are you backing up for archival or high availability or espionage?

Examples of Data

Static: Configurations of running servers. You need these 24x7 but they don't change much.

Databases / Transactions - financial & otherwise: These are updated frequently and need to balance. Associated with these are logs and duplication and other means of rollback and integrity checking. With databases it's often a good idea to dump them in a good portable format, especially if the inbuilt format is not cross platform or cross version compatible. EG 'mysqldump mydata ;mydata.dump' will give you a text file which can be used on most mysql versions and possibly adapted to other database packages.

Logs: People don't tend to read them unless something goes wrong in which case they're valuable. These need to be kept but don't need to be restored in a hurry.

Home directories: This is a mixed bag of everything but some policies could be instated to make the admin's life easier. EG Making specific sub-directories for things and assigning them different backup/restore priorities. Often the existence of a home directory is more important than the rest of the contents as it may make a user unable to login without it.

Code repositories: Programmers should be accustomed to doing regular backups anyway, they often need to revert to an old version to figure out what they broke. Any tools used such as CVS that have a central repository should be backed up almost as often as programmers commit code, at least once a day but they could probably cope with it being missing for half a day.

High availability - read only: Websites frequently used by your clients. They can contain dynamic data but customers don't update it. This sort of scenario lends itself to frequent replication to a backup server.

High availability - interactive: Taking a website again, this one might allow the customer to do such things as place orders. The website maintains some state information to allow building of an order. This is the most difficult, the state information can be stored in a replicated database. In the event of web server failure the other one comes into play and the customer may have to login again but the information is kept. (Otherwise complex designs and expensive hardware can be used to seamlessly migrate the state to the other webserver).

Important Linux directories

```

/var/spool/mail - daily backup
/var/lib/mysql - databases - backup the dumps, and possibly
                 the binary.
/var/log ?     - from "don't care" to "backup daily"
/etc          - backup config changes
/home        - be selective, but if you can't backup daily.
/home/<user>/mail - contains the user's mail folders
                (may also be 'Mail' or 'Maildir')
/home/<user>/ssh - If you login using ssh keys only, this is a
                 must have.
/usr/local   - locally installed apps & data
                Application specifics

```

5.2.2 Backup & Restore methods

Copy the files to another directory

This is the poor mans backup and does not offer much peace of mind. It does protect against accidental deletion and corruption by users. One advantage is that it can be very quick for things such as log files. You can also keep multiple copies, one for every day of the week for example. See `/etc/logrotate.conf`.

Backup to a standby partition

This has about the same level of peace of mind as the above. The backup partition can be left un-mounted after the backup. The backup is slower than the above but the restore operation can be quick. See also "Broken Mirror" method below.

Backup to tape

This is probably the most common backup used in the commercial world. It's easy to backup the lot every day provided you have the tape capacity. If you don't, you become more selective as to what to backup. There's a variety of software to do this but there's 3 main basic systems. Tar, cpio and dump. Often commercial software uses these basic systems and provide for labelling and indexing as well as multi-server capability from a simple GUI. The reason for using the basic systems is you can restore from them if you have to.

Backup to standby disk

This can offer peace of mind and a fairly cheap backup for people that don't require 24x7 service. Basically a removable drive bay houses another hard disk of similar capacity and the entire system is backed up. This can be done partition by partition or file by file using `dd`, `cpio` or `rsync`. Additional steps can be taken to ensure that the backup is also bootable. The backup drive should be removed once done and treated like a tape. The disadvantage here is that you most likely will need to power down the system twice for one backup. Alternately, if you have an external USB or fire-wire storage medium it becomes possible to do this without downtime.

Backup to CDROM/DVD

Under Linux (as far as I know) there's no software to directly write data without creating an image first. This means there must be sufficient space available. It would be possible to create a bootable CD with restore software and a compressed filesystem but I haven't seen this. It may be OK if you don't have a large filesystem or you have a DVD writer or you're not backing up everything.

RAID System

Not strictly a backup but a RAID system can protect against hard drive failure by providing redundancy. Data is written simultaneously to 2 or more hard drives and can include parity information. It does not protect against corrupt databases and people removing files. It will corrupt and remove files equally well on all disks. Linux can do RAID in software very well but the ideal is a hardware solution involving hot swappable disks so they can be replaced while the system is fully running. A RAID system can mean the difference between going on-site at 3am and saying "Oh dear, we'll replace that first thing in the morning". Just ensure that you do have a replacement readily available and do not have to wait a week.

RAID Tape array

In a similar manner to RAID 5 disks, data is written in parallel to 5 tape drives which increases throughput and data integrity.

Backup Server

All of the methods discussed so far involve direct transfer from server to backup medium. If you have a number of servers it may not be practical to install backup devices on each. Another way is to remotely access the required medium directly (`/dev/rmt0`) but arbitration of access can be an issue. An increasingly popular way is to provide a super-server with a huge amount of disk space capable of holding everything required by the other servers. Transferring the data can happen at any time in either a batch or continuous process. A batch would be say backup a whole directory at once whereas a continuous operation might be transmitting log information or database updates. The backup server itself may then employ any one or

more methods to perform backups of itself, possibly based on some statistical analysis. An example of this is a system called ADSM which employs RAID arrays, multiple tape drives, a tape robot with barcode reader and intelligent software that tells the operators which tapes are to go off-site and which ones it wants back. It essentially is a huge cache that stores frequently changing data locally and stores old data off-site.

Broken Mirror

If you've got about 100Gb of data on a mirrored pair of disks and only have a 10 minute backup window this may be for you. Basically you bring the system down, unhook one of the mirrors and replace it with another set of drives and bring the system up again. Mirroring starts from scratch during quiet time and should be finished before load picks up again. With the drive set you just un-hooked this can then be loaded into the standby server and backed up to tape over the course of many hours. Some high end servers can perform this operation without downtime as the hooking up can be done using inbuilt hardware or such things as dual-port fire-wire drive bays. All that is required in this case is an application shutdown, sync, dismount, re-mount, application start type operation.

5.2.3 Software

`dd` - can be used to copy raw disk blocks, even to tape (yuk). eg `dd if=/dev/hda1 of=/dev/hdb1`

`tar` - Tape ARchive - you all know how to unpack `tgz` files, and maybe even create them. Just remove the `'f'` option. It also can be an advantage not to use compression as some drives have this built in. Also, a portion of the tape being corrupt can ruin the rest of the data, whereas you can skip corrupt bits and pickup the next file if not compressed. eg `tar -c /home cd /tmp; tar -x`

`cpio` - cp I/O - Similar capabilities of `tar` but different methodology. EG `find /home — cpio -oB /dev/tape cd /tmp; cpio -idB /dev/tape`

`rsync` - remote sync - can sync a directory or whole filesystem by only transferring the changes between them. Be careful about trailing slashes. `rsync -a /home /backup/ rsync -a -e ssh /home backup@backup:/serverA/`

Arkeia - commercial package BRU - commercial package Amanda - Open source? Thousands more, some are client/server model and can backup multiple operating systems which is great.

See <http://www.linuxhelp.com.au/free.shtml> for our generic CPIO backup script.

5.2.4 Rotation & off-site strategies

It's no good having a backup if it's sitting next to the computer when there's a fire. You've got to have some off-site backups for really important stuff. On a small scale a friend of mine has a backup of all my music CD's I couldn't live without.

You could use this example strategy with any bulk medium but typically people refer to tapes or a set of tapes and for convenience I'll refer to a tape. If you can fit everything on one tape good for you, life is easy, backup the

lot daily. If you don't you'll have to do an incremental backup (ie what's changed) daily and do a whole backup with multiple tapes weekly. Take the weekly backup off-site home from work or over to a trustworthy friend's place. Once a month take a weekly backup to long term storage and keep it for 7 years or something if it's got all your tax info on it. It goes without saying the tapes should be labelled full/incremental and a date, hostname and what sequence in the set they are. Daily backup tapes may be rotated once a week with a new tape supplied once a week for a specific day of the week. Eg week1 will be all new tapes with one shipped off on Monday morning, week2 it'll be a new tape for Sunday morning, week3 it'll be Saturday morning's tape that's new. Alternately, some people believe the weekly or monthly should be on a fresh tape that's never been used.

With this strategy you get reasonable rotation of the tapes keeping costs down and for archival purposes, if you keep at least a months worth of data on the server you'll be able to go back to any point over the last few years and pull out a file. If you keep at least 3 months on hard disk you'll have 3 copies of this on 3 separate tapes because believe it or not they do fail and it will happen to you. To explain this more fully lets look at the following table and assume we have some wages data every week and the company's just started and there's 4 weeks per month.

server	has	weekly	tape	has	monthly	tape	has
wk1	wk1	wk1			-		
wk2	wk1-2	wk1-2			-		
wk3	wk1-3	wk1-3			-		
wk4	wk1-4	wk1-4,month1			wk1-4,month1		
wk5	wk1-5	wk1-5,month1			-		
wk6	wk1-6	wk1-6,month1			-		
wk7	wk1-7	wk1-7,month1			-		
wk8	wk1-8	wk1-8,month1-2			wk1-8,month1-2		
wk9	wk1-9	wk1-9,month1-2			-		
wk10	wk1-10	wk1-10,month1-2			-		
wk11	wk1-11	wk1-11,month1-2			-		
wk12	wk1-12	wk1-12,month1-3			wk1-12,month1-3		
wk13	wk2-13	wk2-13,month1-3			-		
wk14	wk3-14	wk3-14,month1-3			-		
wk15	wk4-15	wk4-15,month1-3			-		
wk16	wk5-16	wk5-16,month2-4			wk5-16,month2-4		
wk17	wk6-17	wk6-17,month2-4			-		
wk18	wk7-18	wk7-18,month2-4			-		
wk19	wk8-19	wk8-19,month2-4			-		
wk20	wk9-20	wk9-20,month3-5			wk9-20,month3-5		
wk21	wk10-21	wk10-21,month3-5			-		
....							

A complete backup and archive strategy should provide a means of going back to any point in time for critical data. Sometimes keeping the whole lot of data is not required. For example you could drop the weekly information and keep the monthly summary information and do a dedicated monthly backup for this data. The monthly data may be optimised and arranged for searching and an index provided but essentially contain all the information from the weekly data.

5.3 Lab

5.4 Questions

Objective 111.6

Maintain system time

6.1 Overview

6.1.1 Weight: [4]

6.1.2 Statement of Objective:

Candidate should be able to properly maintain the system time and synchronise 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.

6.1.3 Key files, terms, and utilities include:

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

6.1.4 Resources of Interest:

web <http://www.ntp.org>

Debian ntp-doc `/usr/share/doc/ntp-doc/index.html` on sarg.

LPI Linux Certification in a Nutshell :
by Jeffrey Dean
O'Reilly

LPIC 1 Certification Bible :
Angie Nash and Jason Nash
Hungry Minds

6.2 Notes

Prepared by Andrew Eager

6.2.1 Display or Set System Date & Time: `date`

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
```

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 Offset)

```
$ 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
```

-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
```

-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
```

6.2.2 The Hardware Clock: `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.

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 adjust 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

6.2.3 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 unprivileged 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.

The suite of tools

NTP normally comes in a package and contains the following binaries:

- `ntpd` - Network Time Protocol (NTP) daemon
- `ntpq` - standard NTP query program
- `ntpdc` - special NTP query program
- `ntpdate` - set the date and time via NTP
- `ntptrace` - trace a chain of NTP servers to the primary source
- `tickadj` - set time-related kernel variables
- `ntptime` - read kernel time variables
- `ntp-genkeys` - generate public and private keys

6.2.4 Quick ntp install guide

For anyone new to NTP, here's a quick guide to installing & setting up NTP.

- Install NTP package (`rpm -Uvh ntp-4.1.0-4.rpm`)
or
`apt-get install ntp`
- Modify `/etc/ntp.conf` to reflect time servers
- Start the service: `service ntpd start`
- Confirm operation using `ntpq` (command `pe`)

That's all there is to it! The hardest part is deciding which public time servers to use.

6.2.5 ntpdate - Set system time & date

- `Ntpdate` is a command line utility that will set the local machines time & date from the indicated remote time server(s).
- More than one server can be specified in order for `ntp` to get a better idea of the transit time and overall server accuracy.
- Running as a cron job is a simple way to maintain system time

Usage: `ntpdate [options] server ...`

```
# ntpdate ntp.nml.csiro.au
21 May 14:01:13 ntpdate[4002]: adjust time server 10.27.1.10
offset -0.000804 sec
```

This will set the local machines system time using server `ntp.nml.csiro.au`

6.2.6 ntpd - The NTP daemon

- NTPD is a better way to maintain the system time on a permanent basis.
- NTPD acts as both a client & server (Linux only).
- In server mode, other machines on the local network can use the server to set their own system clocks
- For Windows machines, automachron is available.
- NTPD also keeps track of RTC drift.

The NTP daemon is normally started up by the system initialisation scripts. On a Red-Hat system you can start the service by:

```
service ntpd start
```

6.2.7 ntpd usage & configuration

Usage: `ntpd [options] &`
 (normally done in the `/etc/init.d` scripts)
 NTPD is configured using these files:

- `/etc/ntp.conf` - Configuration file
- `/etc/ntp.drift` - RTC drift file
- `/etc/ntp.keys` - Key file (for authentication mode)

The only file of concern to the user is `ntp.conf`. The other files are all written to and read by the ntp applications.

Sample `ntp.conf` file

```
# Disable authentication mode
disable auth
restrict default ignore          # ignore all requests by default
server ntp.cs.mu.OZ.AU           # 128.250.36.2
server apphys16.mst.csiro.au     # 138.194.21.154
server ntp.nml.csiro.au          # 130.155.98.1
server 127.0.0.1                  # localhost
# Lift restrictions on time servers
restrict 128.250.36.2 nomodify    # time service only, no rt mods
restrict 138.194.21.154 nomodify
restrict 130.155.98.1 nomodify
# All local addresses are unrestricted
restrict 127.0.0.1
restrict 10.27.1.0 mask 255.255.255.0
# Set the default drift file
driftfile /etc/ntp/drift
```

Public Time Servers

A (partial) list of public time servers is shown below. When using these servers, it is considered polite to advise the administrator of the service that you intend on using it.

- `ntp.cs.mu.OZ.AU` (128.250.36.2)
- `apphys16.mst.csiro.au` (138.194.21.154)
- `ntp.nml.csiro.au` (130.155.98.1)

Testing NTP

Once you have the NTP daemon up & running, the easiest way of testing it is to use the `ntpq` utility.

```
# ntpq
ntpq> pe
      remote           refid          ...           delay    offset  jitter
=====
localhost.local 0.0.0.0      ...           0.000     0.000 4000.00
xmurgon.cs.mu.OZ .GPS.        ...           526.202  -206.43 208.270
+apphys16.mst.cs .ATOM.       ...           169.956   -5.576  87.828
*tictoc.tip.CSIR .ATOM.       ...           149.988  -24.328   6.761
ntpq> q
#
```

6.3 Lab

You should check that `ntp` is installed your system, if not, install the source, deb or rpm for `ntp`.

6.3.1 Explore the `ntp` documentation

1. See what commands are available: `$ ntp<tab><tab>` ↵
2. Check the info and man pages for the available commands.
3. Have a look at the homepage for `ntp` <http://www.ntp.org>
4. Have a look at the local documentation:
 - Debian3.0r1: `file:///usr/share/doc/ntp-doc/html/index.htm`
 - RedHat9.0: `file:///usr/share/doc/ntp-4.1.2/index.htm`

6.3.2 Use the `date` command

1. Scan the info and man pages for `date`.
2. Try out some of the options described in these notes.
3. Set the System Time using `date` if it is not correct.

6.3.3 Use the `hwclock` command

1. Scan the info and man pages for `hwclock`.
2. Try out some of the options described in these notes.

6.3.4 Explore the `ntp` family of commands

1. Scan the info and man pages for `ntp*`.
2. Try out some of the options described in these notes.

6.3.5 Setup `ntp`

Find a suitable *Secondary Time Server* near you.
<http://www.eeds.udel.edu/mills/ntp/>

1. Scan the info and man pages for `ntp*`.
2. Edit `/etc/ntp.conf` as described in these notes.
3. Set up a cron job as described in these notes.

6.4 Questions

Topic 112

Networking Fundamentals

Objective 112.1

Fundamentals of TCP/IP

1.1 Overview

1.1.1 Weight: []

1.1.2 Statement of Objective:

Candidates should demonstrate a proper understanding of network fundamentals. This objective includes the understanding of IP-addresses, network masks and what they mean (i.e. determine a network and broadcast address for a host based on its subnet mask in "dotted quad" or abbreviated notation or determine the network address, broadcast address and netmask when given an IP-address and number of bits). It also covers the understanding of the network classes and classless subnets (CIDR) and the reserved addresses for private network use. It includes the understanding of the function and application of a default route. It also includes the understanding of basic Internet protocols (IP, ICMP, TCP, UDP) and the more common TCP and UDP ports (20, 21, 23, 25, 53, 80, 110, 119, 139, 143, 161).

1.1.3 Key files, terms, and utilities include:

```
/etc/services  
ftp  
telnet  
host  
ping  
dig  
traceroute  
whois
```

1.1.4 Resources of Interest:

Linux Networking HOWTO - Joshua Drake :

<http://www.linuxdoc.org/HOWTO/Net-HOWTO/index.html>

The Linux Networking Overview HOWTO by Daniel Lopez Ridruejo :

<http://www.linuxdoc.org/HOWTO/Networking-Overview-HOWTO.html>

Linux Network Administrators Guide :

<http://www.linuxdoc.org/LDP/nag2/index.html>

1.2. *NOTES*

93

1.2 Notes

1.3 Lab

1.4 Questions

Objective 112.3

TCP/IP configuration and troubleshooting

3.1 Overview

3.1.1 Weight: []

3.1.2 Statement of Objective:

Candidates should be able to view, change and verify configuration settings and operational status for various network interfaces. This objective includes manual and automatic configuration of interfaces and routing tables. This especially means to add, start, stop, restart, delete or reconfigure network interfaces. It also means to change, view or configure the routing table and to correct an improperly set default route manually. Candidates should be able to configure Linux as a DHCP client and a TCP/IP host and to debug problems associated with the network configuration.

3.1.3 Key files, terms, and utilities include:

```
/etc/HOSTNAME or /etc/hostname
/etc/hosts
/etc/networks
/etc/host.conf
/etc/resolv.conf
/etc/nsswitch.conf
ifconfig
route
dhcpcd, dhcpclient, pump
host
hostname (domainname, dnsdomainname)
netstat
ping
traceroute
```

tcpdump
the network scripts run during system initialisation.

3.1.4 Resources of Interest:

Linux Networking HOWTO by Joshua Drake :

<http://www.linuxdoc.org/HOWTO/Net-HOWTO/index.html>

Linux Ethernet-Howto by Paul Gortmaker :

<http://www.linuxdoc.org/HOWTO/Ethernet-HOWTO.html>

3.2 Notes

3.3 Lab

3.4 Questions

Objective 112.4

Configure Linux as a PPP client

4.1 Overview

4.1.1 Weight: []

4.1.2 Statement of Objective:

Candidates should understand the basics of the PPP protocol and be able to configure and use PPP for outbound connections. This objective includes the definition of the chat sequence to connect (given a login example) and the setup commands to be run automatically when a PPP connection is made. It also includes initialisation and termination of a PPP connection, with a modem, ISDN or ADSL and setting PPP to automatically reconnect if disconnected.

4.1.3 Key files, terms, and utilities include:

```
/etc/ppp/options.*  
/etc/ppp/peers/*  
/etc/wvdial.conf  
/etc/ppp/ip-up  
/etc/ppp/ip-down  
wvdial  
pppd
```

4.1.4 Resources of Interest:

Linux PPP HOWTO Corwin Light-Williams and Joshua Drake :

<http://www.linuxdoc.org/HOWTO/PPP-HOWTO/index.html>

4.2 Notes

4.3 Lab

4.4 Questions

Topic 113

Networking Services

Objective 113.1

Configure and manage inetd, xinetd, and related services

1.1 Overview

1.1.1 Weight: []

1.1.2 Statement of Objective:

Candidates should be able to configure which services are available through inetd, use tcpwrappers to allow or deny services on a host-by-host basis, manually start, stop, and restart Internet services, configure basic network services including telnet and ftp. Set a service to run as another user instead of the default in inetd.conf.

1.1.3 Key files, terms, and utilities include:

```
/etc/inetd.conf  
/etc/hosts.allow  
/etc/hosts.deny  
/etc/services  
/etc/xinetd.conf  
/etc/xinetd.log
```

1.1.4 Resources of Interest:

TBA

104

CONTENTS

1.2 Notes

1.3 Lab

1.4 Questions

Objective 113.2

Operate and perform basic configuration of sendmail

2.1 Overview

2.1.1 Weight: []

2.1.2 Statement of Objective:

Candidate should be able to modify simple parameters in sendmail configuration files (including the "Smart Host" parameter, if necessary), create mail aliases, manage the mail queue, start and stop sendmail, configure mail forwarding and perform basic troubleshooting of sendmail. The objective includes checking for and closing open relay on the mailserver. It does not include advanced custom configuration of Sendmail.

2.1.3 Key files, terms, and utilities include:

```
/etc/sendmail.cf  
/etc/aliases or /etc/mail/aliases  
/etc/mail/*  
~/.forward  
mailq  
sendmail  
newaliases
```

2.1.4 Resources of Interest:

TBA

2.2 Notes

2.3 Lab

2.4 Questions

Objective 113.3

Operate and perform basic configuration of Apache

3.1 Overview

3.1.1 Weight: []

3.1.2 Statement of Objective:

Candidates should be able to modify simple parameters in Apache configuration files, start, stop, and restart httpd, arrange for automatic restarting of httpd upon boot. Does not include advanced custom configuration of Apache.

3.1.3 Key files, terms, and utilities include:

apachectl
httpd
httpd.conf

3.1.4 Resources of Interest:

Apache home page :

<http://www.apache.org>

3.2 Notes

3.2.1 Apache

- Apache is a web server (http daemon)
- Default on all Linux distros
- Most popular web server on the Internet
- Named after the number of patches to original source code.
- Provides both HTTP and HTTPS (SSL) as standard
- Other features added with modules (eg cgi)

3.2.2 Starting & Stopping Apache

- Apache can be started:
 - On demand through inetd or xinetd
 - As a daemon
- Normally started as daemon to reduce connect delay
- Uses standard SysV start/stop semantics
 - Debian: `/etc/init.d/apache`
 - RedHat: `/etc/rc.d/init.d/httpd`
- An alternative is `apachectl`

3.2.3 `apachectl`

`apachectl` is a management utility. To use it:

```
# apachectl <command>↵
```

<i>command</i>	<i>Function</i>
start	Start the daemon
stop	Stop the daemon
restart	Restart or start the daemon
fullstatus	Report status of server (requires lynx)
graceful	Gracefully restart the serve
configtest	Test config file syntax
help	Display commands

3.2.4 HTTPD Parameters

The `httpd` daemon can be run directly if needed. On Debian the daemon is called `apache`.

<i>Parameter</i>	<i>Function</i>
<code>-v</code>	Shows version
<code>-V</code>	Shows compile configuration
<code>-h</code>	List all cmd line parameters
<code>-l</code>	List compiled in modules
<code>-L</code>	List config directives
<code>-S</code>	Shows parsed settings (virtual hosts only)
<code>-t</code>	Test config file & doc root
<code>-T</code>	Test config file only

3.2.5 HTTPD Parameters

The following options take parameters:

<i>Parameter</i>	<i>Function</i>
<code>-D name</code>	Defines a name for use in <code>IfDefine name</code>
<code>-d directory</code>	Defines an alternate server root
<code>-f file</code>	Set a new configuration file
<code>-C "directive"</code>	Process directive before reading config file
<code>-c "directive"</code>	Process directive after reading config file

3.2.6 Configuring Apache

- Apache originally used 3 configuration files:
 - `httpd.conf` - Server settings
 - `srm.conf` - File types & doc specs
 - `access.conf` - Security settings
- All configuration is now done in `httpd.conf`
- Normally located in `/etc/httpd/conf`

3.2.7 Site-wide Directives

<i>Directive</i>	<i>Function</i>
<code>ServerAdmin</code>	Sets email address for admin
<code>ServerName</code>	Sets the name of the server
<code>DocumentRoot</code>	Sets the root for content served
<code>ServerRoot</code>	Sets root for server files
<code>ServerType</code>	<code>standalone</code> or <code>inetd</code>
<code>MinSpareServers</code>	No of free <code>httpd</code> 's before starting more
<code>MaxSpareServers</code>	No of free <code>httpd</code> 's before killing some
<code>StartServers</code>	No of <code>httpd</code> 's to start
<code>MaxClients</code>	Maximum no of <code>httpd</code> 's to run at once.

3.2.8 Directory block Directives

You can set directives so that they only apply to a particular part of the content directory tree. For example:

```
<Directory /Games>
    AllowOverride None
</Directory>
```

This says that the `.htaccess` file can not override settings for this directory

3.2.9 Access Control

This directive controls who can access what directories on your site. This is about the only directive that needs to be changed from an 'off-the-shelf' configuration if you don't want external users to access your site.

```
<Location />
    order deny,allow
    deny from all
    allow from 127.0.0.0/255.0.0.0
    allow from .c222
</Location>
```

This says to deny first then allow. The result is that only users in the `.c222` domain and the localhost will be able to access the server.

3.2.10 Other Directives

There are a large number of configuration directives. These are grouped as follows:

- Aliases & Redirects
- Default pages
- User Web Directories (site content in a users home)
- MIME types
- CGI files
- Directory Browsing
- Authentication
- Virtual hosts (multiple sites on one host)
- Logging directives

3.3 Lab

1. Confirm that you have apache installed on the system:

```
# rpm -q apache ↵
```

2. If it doesn't exist, install it:

```
# rpm -Uvh apache-*.rpm ↵
```

3. Setup apache so that only those in the .c222 domain can access the server. (See notes)

4. Startup apache:

```
# service httpd start ↵
```

5. Start up your browser and point it to your host `http://boxXX.c222`, where XX is your box number. You should see a default web page

6. Now make a symbolic link in `/var/www/html` called `homes` that points to the system home directory:

```
# ln -s /home /var/www/html/homes ↵
```

7. Point your browser to it: `http://boxXX.c222/homes/`

8. What happens when you try to browse the directories under `homes`?

9. Make a directory that is owned by `apache` called `public` with 0700 permissions:

```
# mkdir /home/public ↵
```

```
# chown apache:apache /home/public ↵
```

```
# chmod 0700 /home/public ↵
```

10. Put something in the directory:

```
# cp /etc/hosts /home/public ↵
```

11. Use the browser to view & download the `hosts` file located in `/home/public`.

12. Make a backup of the `index.html` file located in `/var/www/html` and then remove the file `index.html`

```
# cd /var/www/html ↵
```

```
# cp index.html index.bak ↵
```

```
# rm index.html ↵
```

13. What do you see if you try to browse `http://boxXX.c222`?

14. Copy back the original `index.html` file removed in the previous step.

15. Have a look at `index.html` to see what HTML looks like.

3.4 Questions

Objective 113.4

Properly manage the NFS, smb, and nmb daemons

4.1 Overview

4.1.1 Weight: []

4.1.2 Statement of Objective:

Candidate should know how to mount remote filesystems using NFS, configure NFS for exporting local filesystems, start, stop, and restart the NFS server. Install and configure Samba using the included GUI tools or direct edit of the `/etc/smb.conf` file (Note: this deliberately excludes advanced NT domain issues but includes simple sharing of home directories and printers, as well as correctly setting the `nmbd` as a WINS client).

4.1.3 Key files, terms, and utilities include:

```
/etc/exports  
/etc/fstab  
/etc/smb.conf  
mount  
umount
```

4.1.4 Resources of Interest:

TBA

114

CONTENTS

4.2 Notes

4.3 Lab

4.4 Questions

Objective 113.5

Setup and Configure Basic DNS Services

5.1 Overview

5.1.1 Weight: []

5.1.2 Statement of Objective:

Candidate should be able to configure hostname lookups and troubleshoot problems with local caching-only name server. Requires an understanding of the domain registration and DNS translation process. Requires understanding key differences in configuration files for bind 4 and bind 8.

5.1.3 Key files, terms, and utilities include:

```
/etc/hosts  
/etc/resolv.conf  
/etc/nsswitch.conf  
/etc/named.boot (v.4) or /etc/named.conf (v.8)  
named
```

5.1.4 Resources of Interest:

TBA

5.2 Notes

5.2.1 Setup and Configure basic DNS services

Objective

Candidate should be able to configure hostname lookups and troubleshoot problems with local caching-only name server. Requires an understanding of the domain registration and DNS translation process. Requires understanding key differences in configuration files for bind 4 and bind 8.

5.2.2 Setup and Configure basic DNS services

Key files, terms, and utilities

```
/etc/hosts
/etc/resolv.conf
/etc/nsswitch.conf
/etc/named.boot (v.4) or /etc/named.conf (v.8) (In Debian /etc/bind/named.conf)
named
```

5.2.3 DNS - DOMAIN NAME SERVICE

- The Internet works with numbers not names.

```
www.abc.gov.au is really 203.2.218.61
```

- DNS namespace is made up of a tree of domain names.
- At the top is root (.)
- Below this is the Top Level Domain (TLD)
- Below the TLD is the Second Level Domain.
- The Second level domain is handled by whoever 'owns' that domain
- Third & lower level domains are handled by the domain owner.

5.2.4 DNS - DOMAIN NAME SERVICE

- Example:

```
node1.office.my-domain.com
^         ^         ^         ^
|         |         |         |
|         |         |         -- Top level domain
|         |         -- Second level domain
|         - Subdomain
-- Hostname
```

- Domain names are fully qualified (FQDN) when a name is specified all the way down to the hostname.

5.2.5 RESOLVING A NAME

- A name is resolved using the following steps:
 - `/etc/nsswitch.conf` is checked to see what resolution method to use (eg: read `/etc/hosts`, use dns, use nis...)
 - `nsswitch` says USE DNS:
 - * Read `resolv.conf` to see what nameserver to use
 - * Send request to nameserver and wait for response
 - `nsswitch` says USE HOSTS
 - * Lookup `/etc/hosts` for a matching hostname

5.2.6 The `nsswitch.conf` file

- This is a file that determines what mechanisms are used by the host-name library calls to resolve names.
- The file contains lines with an identifier followed by a list of methods to use for name lookups.
- An example:


```
passwd: files nisplus nis
shadow: files nisplus nis
group: files nisplus nis
hosts: db files dns
```
- Note that the other entries like `passwd`, `shadow` and `group` are used for other applications like login and have nothing to do with DNS.

5.2.7 The `nsswitch.conf` file

- In the `hosts` line, we see that any hostname to be looked up will be done in the following order:
 1. Use local databases file (`.db` files in `/var/db`)
 2. Read `/etc/hosts`
 3. Search DNS
- The Search options can be one of:

```
nisplus (or nis+) - Consult NIS+ (Yellow Pages)
nis (or yp)       - Consult NIS
dns               - Use a DNS server
files            - Use local files like /etc/hosts
db               - Use local database files
compat           - Use NIS in compat mode
[NOTFOUND=return] - Stop searching and return host notfound
```

5.2.8 An example nsswitch file:

```
nisplus (or nis
passwd:    db files nisplus nis
shadow:    nisplus
group      db files nisplus nis

hosts:     db files nis dns

# Example - obey only what nisplus tells us...
#services: nisplus [NOTFOUND=return] files
#networks:  nisplus [NOTFOUND=return] files
#protocols: nisplus [NOTFOUND=return] files
#rpc:       nisplus [NOTFOUND=return] files
#ethers:    nisplus [NOTFOUND=return] files
#netmasks: nisplus [NOTFOUND=return] files

bootparams: nisplus [NOTFOUND=return] files

ethers:     files
netmasks:   files
networks:   files nis
protocols:  files nisplus
rpc:        files
services:   files nisplus

netgroup:   files nisplus

publickey:  nisplus

automount:  files nisplus
aliases:    files nisplus
```

5.2.9 The resolv.conf file

- This file configures how the system uses DNS. An example:

```
search aes
nameserver 10.27.1.10
nameserver 10.27.1.254
```

- The 'search' line says what to append to a non-fully qualified name: eg: ping node10 -> ping node10.aes
- The nameserver lines tell the hostname routines which dns server to send requests to. (If first lookup fails, use the second, third etc)

5.2.10 BIND - Berkley Internet Name Domain

- Bind is just one implementation of a DNS. Bind is to DNS what Apache is to http.
- Bind is configured with:

```
/etc/named.conf    - For BIND V8
/etc/named.boot    - For BIND V4
```

- Know that there is a difference between V4 & V8.
- Know how to configure V8 but not V4. (Different syntax)

5.2.11 BIND Configuration

- The configuration file contains subsections as follows:
 - Options → How named will operate
 - logging → What/how named will log information
 - Access Lists → Who can use named & what they can do
 - Remote Servers → Characteristics of remote servers
 - zones → Information about our defined domains

5.2.12 An Example Config file:

```
options {
    directory "/var/named/";
    forward only;
    forwarders {
        203.2.75.132;
        203.2.75.108;
    };
    query-source address * port 53;
    listen-on {
        10.27.1.10;
        127.0.0.1;
    };
    notify no;
};

#### The root zone ###
zone "." {
    type hint;
    file "named.ca";
};

#### A zone for localhost ###
zone "0.0.127.in-addr.arpa" {
    type master;
    file "0.0.127.in-addr.arpa.zone";
};

zone "localhost" {
    type master;
    file "localhost.zone";
};

### A local domain ###
zone "1.27.10.in-addr.arpa" {
    type master;
    file "1.27.10.in-addr.arpa.zone";
};

zone "aes" {
    type master;
    file "aes.zone";
};

key "key" {
    algorithm hmac-md5;
    secret "JoqlFqtnqcqurkhMOrrbQLYRcxSYXoNROvNTZBqWJFumleNkzOvEvTAbqpbMV";
};
```

5.2.13 Zone files:

- Each zone uses a file for:

- Hostname to IP address translations (Forward lookups)
- IP to Hostname translations (Reverse lookups)
- The names can be anything, but usually:
 - Forward file `-i ;domain;.zone`
 - Reverse file `-i ;Net-IP;.in-addr.arpa`
- Where the Net-IP is the network part of the IP address.

5.2.14 Zone Records:

SOA record Marks the start of a zone.

NS record Defines the name server for a zone or subdomain

MX record Define mail servers for domain

CNAME record Defines an alias for a hostname

LOC record Defines the physical location of the server

SRV record Defines what services are found where (eg ftp, http etc)

A record Defines hostname to IP address translations (forward file)

PTR record Defines IP address to hostname translations (reverse file)

5.2.15 Example Forward file `/var/named/aes.zone`

```
@      IN      SOA      node10.aes.  root.localhost (
                2 ; serial
                28800 ; refresh
                7200 ; retry
                604800 ; expire
                86400 ; ttl
                )

@      IN      NS      node10.aes.

node5  IN      MX      10      mail
node6  IN      MX      10      mail
node4  IN      MX      10      mail
node2  IN      MX      10      mail
node10 IN      MX      10      mail
gw     IN      MX      10      mail

node10 IN      A       10.27.1.10
node2  IN      A       10.27.1.2
node4  IN      A       10.27.1.4
node5  IN      A       10.27.1.5
node6  IN      A       10.27.1.6
cds    IN      A       10.27.1.99
gw     IN      A       10.27.1.254

ns     IN      CNAME   node10
mail   IN      CNAME   node10
node-4 IN      CNAME   node4
```


5.2.16 Example reverse file /var/named/1.27.10.in-addr.arpa.zone

```

@      IN      SOA      @      root.localhost (
                2 ; serial
                28800 ; refresh
                7200 ; retry
                604800 ; expire
                86400 ; ttk
                )

@      IN      NS      ns.aes.

2      IN      PTR     node2.aes.
4      IN      PTR     node4.aes.
5      IN      PTR     node5.aes.
6      IN      PTR     node6.aes.
10     IN      PTR     node10.aes.
99     IN      PTR     cds.aes.
254   IN      PTR     gw.aes.

```

5.2.17 Configuring a Caching only Nameserver

- **A caching only nameserver is simple to setup.**
The first time a name is needed, a normal lookup occurs (Authoritative)
The next time that name is needed, it is returned from cache (Non-authoritative)
- **Under /etc/named.conf in the options section, just make sure you have the following directives set:**

```

options {
    directory "/var/named/";
    forward only;
    forwarders {
        <First DNS to query>;
        <Second DNS to query>;
    };
    listen-on {
        <Your local IP address>;
        127.0.0.1;
    };
};

```

- **Leave the root zone (.) and localhost entries as they are.**

5.2.18 Testing DNS

- **To test DNS, use one of the following tools:**
 - nslookup (deprecated)
 - dig
 - host
- **To use in their simplest form, just add the hostname you wish to query as the first option to the command:**

```

nslookup node16.c222
dig node16.c222
host node16.c222

```

5.2.19 nslookup

- Usage: `nslookup [option] host-to-find [-name-server]`

Example:

```
$ nslookup node2.aes -10.27.1.10 ↵
```

- Note: `nslookup` is deprecated and may be removed from future releases. Consider using the 'dig' or 'host' programs instead. Run `nslookup` with the `-sil[ent]` option to prevent this message from appearing.

```
Server:          10.27.1.10
Address:         10.27.1.10#53
```

```
Name:   node2.aes
Address: 10.27.1.2
```

5.2.20 dig

- Usage: `dig [@name-server] host-to-find [query-type]`

- Example:

```
$ dig @10.27.1.10 node2.aes ↵
; <<>> DiG 9.2.0 <<>> @10.27.1.10 node2.aes
;; global options:  printcmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 43860
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;node2.aes.                IN      A

;; ANSWER SECTION:
node2.aes.                86400  IN      A      10.27.1.2

;; AUTHORITY SECTION:
aes.                      86400  IN      NS     node10.aes.

;; ADDITIONAL SECTION:
node10.aes.               86400  IN      A      10.27.1.10

;; Query time: 5 msec
;; SERVER: 10.27.1.10#53(10.27.1.10)
;; WHEN: Mon Sep  2 13:48:38 2002
;; MSG SIZE  rcvd: 80
```

5.2.21 host

- Usage: `host [option] host-to-find [name-server]`

- Example:

```
$ host node2.aes ↵
node2.aes has address 10.27.1.2
```

5.2.22 Exercise:

1. Install bind on your machine:

```
# rpm -Uvh bind-9*.rpm
```

2. Configure a Caching only nameserver on your machine. (Make all queries forward to 192.168.222.254)
3. Make changes to resolv.conf & nsswitch.conf as required (Default domain to use is c222)
4. Start the named.

```
# service named start
```

5. Test it out with the host node16.c222 using:

- nslookup
- dig
- host

6. Test again this time with the host box16

7. (For those who want a DNS challenge)

- (a) Setup a set of zones for the .c222 domain.
- (b) Insert the new zone into the main configuration file
- (c) Restart the named and test it.

5.3 Lab

5.4 Questions

Objective 113.7

Set up secure shell (OpenSSH)

7.1 Overview

7.1.1 Weight: []

7.1.2 Statement of Objective:

The candidate should be able to obtain and configure OpenSSH. This objective includes basic OpenSSH installation and troubleshooting, as well as configuring sshd to start at system boot.

7.1.3 Key files, terms, and utilities include:

```
/etc/hosts.allow  
/etc/hosts.deny  
/etc/nologin  
/etc/ssh/sshd_config  
/etc/ssh_known_hosts  
/etc/sshrd  
sshd  
ssh-keygen
```

7.1.4 Resources of Interest:

TBA

7.2 Notes on ssh

Notes from a talk by Angus Lees

“Secure SHell”

A functional replacement of the ancient `rsh` command, except with encryption and authentication.

7.2.1 Versions

Commercial SSH Finish company. Original authors of SSH.

OpenSSH Split from last free version of commercial SSH. Development led by OpenBSD team.

Draft “secsh” RFC.

Alternative implementations exist (Putty, Net::SSH::Perl, etc)

7.2.2 Commands

`ssh` Run a shell command on a remote host

`sshd` SSH server daemon

`scp` Copy files using SSH

`sftp` An ftp-like interface into `scp`

`ssh-keygen` Generate an SSH key pair

`ssh-agent`, `ssh-add` SSH key forwarding

7.2.3 ssh Commands

```
ssh [options] host [command]
```

Run a shell command on a remote host.

Acts like a normal shell command. ie: STDIN, STDOUT work as normal.

Without a *command*, `ssh` runs an interactive login.

7.2.4 scp Commands

```
scp user@host:path/file user2@host2:path/file2
```

Copy a file over `ssh`.

user defaults to current login, *user@host* maybe omitted for local files, *path* is relative to `$HOME`

7.2.5 sftp Commands

```
sftp user@host:path
```

ftp-like command line interface to scp.
Only provided with more recent ssh versions.

7.2.6 Advanced Usage

Remember that STDIN and STDOUT still work as normal (unlike telnet):

```
ssh tar zcf - /remotepath > localfile.tar.gz
```

X Forwarding

```
ssh -X host
```

Login to *host* and “forward” X11 connections back to the local Xserver.

A “fake” \$DISPLAY and xauth environment are created, and the X11 data is passed back over the same SSH connection.

Forwarding X over SSH is secure and easy, but slower than not doing it.

On a local LAN, the encryption is probably unnecessary—use normal X methods such as rstart instead (rstart can use ssh for authentication anyway).

Specialised X11 caching methods (eg: LBX) can get better performance than ssh compression over slow links.

There are concerns over connecting to a hostile remote machine and forwarding X back again, so don't forward X by default.

Port Forwarding

Arbitrary ports can also be forwarded over the SSH connection, to add security to other protocols (or bypass poor firewall policies...)

```
# .fetchmailrc example
poll localhost protocol pop3 port 11110:
  preconnect "ssh -C -f user@host.com \
    -L 11110:host.com:110 sleep 10"
```

SSH Keys

Public key authentication. More secure alternative to password login.

Generate a public/private “key pair” with ssh-keygen.

Keep the private key secret.

Append the public key into your (remote) `~/.ssh/authorized_keys` to allow access.

More powerful automation (scripting) possibilities.

SSH Authentication Agent

`ssh-agent` allows key information to be “forwarded” between its child processes—even across nested `ssh` sessions.

Start `ssh-agent` in your X-session or login scripts, and run `ssh-add` to add keys.

`ssh-askpass` is (basically) an X11 version of `ssh-add`.

SSH from win32

Putty Includes command line “`pscp.exe`” scp clone too.

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>

Winscp Graphical SCP client.

<http://winscp.vse.cz/eng>

Other “frontends”

KDE `kio.fish` Provides `ssh://konquerer` paths.

`tramp.el` Transparent access to remote files for emacs.

`rsh-compatible` Anything that can use `rsh` (eg: CVS)

7.3 Lab

7.4 Questions

Topic 114

Security

Objective 114.1

Perform security administration tasks

1.1 Overview

1.1.1 Weight: []

1.1.2 Statement of Objective:

Candidates should know how to review system configuration to ensure host security in accordance with local security policies. This objective includes how to configure TCP wrappers, find files with SUID/SGID bit set, verify packages, set or change user passwords and password aging information, update binaries as recommended by CERT, BUGTRAQ, and/or distribution's security alerts. Includes basic knowledge of ipchains and iptables.

1.1.3 Key files, terms, and utilities include:

```
/proc/net/ip_fwchains  
/proc/net/ip_fwnames  
/proc/net/ip_masquerade  
find  
ipchains  
passwd  
socket  
iptables
```

1.1.4 Resources of Interest:

TBA

1.2 Notes

1.3 Lab

1.4 Questions

Objective 114.2

Setup host security

2.1 Overview

2.1.1 Weight: []

2.1.2 Statement of Objective:

Candidate should know how to set up a basic level of host security. Tasks include syslog configuration, shadowed passwords, set up of a mail alias for root's mail and turning of all network services not in use.

2.1.3 Key files, terms, and utilities include:

```
/etc/inetd.conf or /etc/inet.d/*  
/etc/nologin  
/etc/passwd  
/etc/shadow  
/etc/syslog.conf
```

2.1.4 Resources of Interest:

TBA

2.2 Notes

2.3 Lab

2.4 Questions

Objective 114.3

Setup user level security

3.1 Overview

3.1.1 Weight: [2]

3.1.2 Statement of Objective:

Candidate should be able to configure user level security. Tasks include limits on user logins, processes, and memory usage.

3.1.3 Key files, terms, and utilities include:

quota
usermod

3.1.4 Resources of Interest:

TBA

3.2 Set and View Disk Quotas

Section prepared by Pia Smith

To achieve a general understanding of quotas. In particular the functions of each command, keeping in mind quotas are set on a per-filesystem basis.

3.2.1 Enabling Quotas

In order to use quotas they must first be enabled. To do this there are a few steps:

1. Firstly add the `userquota` and `grpquota` options to the relevant filesystems in `/etc/fstab`, as shown:

```
/dev/hda2 /home ext3 defaults,usrquota,grpquota 1 2
```

2. Then create the `quota.user` and `quota.group` files at the top of the filesystem, in this case, `/home`. Ensure that only root can read these files, like so:

```
fehng:~# touch /home/quota.user /home/quota.group
fehng:~# chmod 600 /home/quota.user /home/quota.group
```

3. We then initialise the `quota.*` files as databases by running `quotacheck`.

```
fehng:/home# quotacheck -augv
Cannot get exact used space... Results might be inaccurate.
quotacheck: Scanning /dev/hda2 [/home] done
quotacheck: Checked 143 directories and 689 files
```

4. Confirm that the databases have actually been initialised by making sure that the `quota.*` files are larger than 0.

5. Run `quotaon` to enable the quota system:

```
fehng:/home# quotaon -a
```

6. There are two further things to ensure quota is turned on when boots, and that the database is checked regularly:

- (a) To ensure quota is turned on upon system boot, add the following to the system's initialisation script (`/etc/rc.d/rc.sysinit` or similar):

```
if [ -x /sbin/quotacheck ]
then
    echo "Checking quotas."
    /sbin/quotacheck -augv
    echo "Done."
fi
if [ -x /sbin/quotaon ]
then
    echo "Enabling quotas."
```



```

    /sbin/quotaon -avug
fi

```

- (b) To ensure that the databases are checked regularly, add a script to one of the crontab system directories, (such as `/etc/cron.weekly/`) to run `quotacheck`:

```

#!/bin/bash
/sbin/quotacheck -avug

```

or a job in crontab to achieve the same thing.

The filesystem (in this case `/home`) is now ready to accept quotas on a per user or group basis.

3.2.2 Quota Limits

There are five types of quota limits that can be enforced:

Per-user hard limit this is the absolute maximum of a users allocated space, once reached the user cannot write anything else to the filesystem, and the currently worked upon file if saved is truncated and useless. The user doesn't lose what is in the current shell, so they can free up some space and then save the file.

Per-group hard limit this is the absolute maximum of a groups allocated space, once reached the group cannot write anything else to the filesystem, and the currently worked upon file if saved is truncated and useless. Users in the group don't lose what is in the current shell, so they can free up some space and then save the file.

Per-user soft limit an abstract limit enforced on users that is less than the hard limit, and once reached, the user enters the grace period. After the soft limit has been reached the user starts getting warnings printed on the terminal that the quota has been exceeded.

Per-group soft limit an abstract limit enforced on groups that is less than the hard limit, and once reached, the group enters the grace period. After the soft limit has been reached the group starts getting warnings printed on the terminal that the quota has been exceeded.

Grace Period Once a soft limit has been reached the user/group enters the grace period which is an abstract time before the hard limit is enforced, regardless of whether the hard limit is reached (assuming the user doesn't get their quota down below the soft limit in that time).

3.2.3 Setting up and configuring quotas.

The next move is to edit the quota reference for each user. We can get around this with scripts, but essentially this is not nice :)

We can actually edit the quota of a typical user on our system and then copy the attributes of that users quota to other users, as follows:

```
fehung:/home/greebo# edquota greebo
```

This edits the quota for user greebo, in this file we change the soft and hard limits to whatever we choose, example:

```
Disk quotas for user greebo (uid 1000):
Filesystem blocks soft hard inodes soft hard
/dev/hda2 538 29000 30000 689 0 0
```

The first soft and hard values are relevant to blocks and the second to inodes, here the user has a block soft and hard limit but no limit on inodes used.

We can then attribute these settings to the rest of the users on our system like so:

```
fehung:/home/greebo# edquota -p greebo $(awk -F: ' $3 > \
999 { print $1 }' /etc/passwd)
```

and can confirm this worked by running `edquota <randomuser>` to see whether the new settings copied across.

We can only modify the grace limit system wide. We do this by running `edquota -tu`, and changing the value.

3.2.4 Quota commands

`quota (1)`

`quota (1)` is used to display quotas on users and groups, using the `-u` switch for users and `-g` switch for groups:

```
fehung:/home# quota -uv greebo ↵
Disk quotas for user greebo (uid 1000):
Filesystem blocks quota limit grace files quota limit grace
/dev/hda2 538 29000 30000 689 0 0
```

`quotaon (1)`

`quotaon (1)` turns on the quota system, `quotaoff` turns it off. Easy!

`repquota (1)`

`repquota (1)` reports on the status on quotas. Common options are as follows:

```
-a reports on all quotas
-g reports on group quotas
-u reports on user quotas
-v verbose mode
```

Examples:

```
# repquota -v /home ↵
```

or

```
# repquota -a ↵
```

3.3. LAB

139

3.2.5

3.2.6

3.2.7

3.2.8

3.3 Lab

3.4 Questions

List of topics

- **105 Kernel**
 - Manage/Query kernel and kernel modules at runtime
 - Reconfigure, build, and install a custom kernel and kernel modules
- **106 Boot, Initialization, Shutdown and Runlevels**
 - Boot the system
 - Change runlevels and shutdown or reboot system
- **107 Printing**
 - Manage printers and print queues
 - Print files
 - Install and configure local and remote printers
- **108 Documentation**
 - Use and manage local system documentation
 - Find Linux documentation on the Internet
 - Notify users on system-related issues
- **109 Shells, Scripting, Programming, Compiling**
 - Customize and use the shell environment
 - Customize or write simple scripts
- **111 Administrative Tasks**
 - Manage users and group accounts and related system files
 - Tune the user environment and system environment variables
 - Configure and use system log files to meet administrative and security needs
 - Automate system administration tasks by scheduling jobs to run in the future
 - Maintain an effective data backup strategy
 - Maintain system time

- **112 Networking Fundamentals**
 - Fundamentals of TCP/IP
 - TCP/IP configuration and troubleshooting
 - Configure Linux as a PPP client
 -
- **113 Networking Services**
 - Configure and manage inetd, xinetd, and related services
 - Operate and perform basic configuration of sendmail
 - Operate and perform basic configuration of Apache
 - Properly manage the NFS, smb, and nmb daemons
 - Setup and configure basic DNS services
 - Configure ntp.conf and ntp.drift to be used by xntpd
- **Security**
 - Perform security administration tasks
 - Setup host security
 - Setup user level security
-

.1 Topics moved to General Linux 1

- –
- –
- –
- –
-
-