

RV64GQV_S, dynamic libraries, Memory
Mapping Units (MMUs), Address Generation
Units (AGUs), protection rings, CPU interrupts
and GNU/Linux/Systemd

Adam Boulton (www.bou.lt)

April 30, 2025

Contents

Preface	2
I To DOS	3
1 DOS fdisk	4
II File systems, block devices and tools for partitioning and formatting drives	5
2 File systems including ext2	6
3 Journalling file systems including ext4	8
4 Partitioning drives with GNU parted and util-linux: fdisk and cfdisk, and lsblk, wipefs and fsck	9
5 Formatting partitions using util-linux: mkfs	11
III UEFI and non-BIOS first-stage bootloaders, and second-stage boot loaders	12
6 More first-stage boot loaders: UEFI and coreboot/libreboot	13
7 The boot partition and second-stage boot loaders, including GRUB	14

<i>CONTENTS</i>	2
IV Memory Mapping Units (MMUs):	16
V Address Generation Units (AGUs):	17
VI Linux kernel	18
8 Loading the Linux kernel from the boot partition	19
9 Directory layout on Linux: /boot, /sbin, /proc, /sys, /etc and /lib	20
10 The init process, openrc and runit, and mounting using /etc/fstab	21
VII User space	23
11 Using swap partitions with util-linux: mkswap, swapon and swapoff, and swapfiles	24
12 /dev/shm, /tmp and tmpfs	25
VIII Linux multi stuff?	26
13 Batch processing	27
14 Interrupts	28
15 Concurrency control	29
IX Pseudo-character devices	30
16 Pseudo-character device files	31
17 Loop devices	32
X Shells	33
18 Interactive login shells, read-eval-print loop (REPL), the Bourne shell implementations ash and dash, including commands: cd, fg, exit, jobs	34
19 Keyboards and locales	37

20 Other util-linux programs, including lscpu; more; mount and umount; dmesg; hwclock; kill; whereis; cal; fallocation; su; chsh	38
21 Linux modules using kmod: lsmod, insmod, rmmod, modprobe and modinfo	40
 XI GNU coreutils: Basics	 41
22 GNU Core Utilities: Exploring folders using ls and dir, vdir, dircolors, du and stat	42
23 GNU Core Utilities: Reading files using cat, tac, head and tail, and nl, od, base32, base64 and basenc	43
24 GNU Core Utilities: Writing to files using cp, dd and install, mv, rm and shred, mkdir, rmdir, touch and ln, readlink, mknod, mkfifo, mktemp, sync, link, unlink, truncate, split and csplit	44
25 GNU Core Utilities: Reading and transforming text using tr, cut, split, ptx, sort, tsort, expand, unexpand and uniq, fmt, pr and fold	46
26 GNU Core Utilities: Reading from multiple files using paste, comm and join	47
27 GNU Core Utilities: Summarising files with wc and checksums (sum, cksum, b2sum, md5sum, sha1sum, sha224sum, sha256sum, sha512sum)	48
28 GNU Core Utilities: Modifying command invocation with chroot (and jails), env, nice, nohup, stdbuf and timeout	49
29 GNU Core Utilities: Getting system information with df, date, uptime, uname, env, printenv, nproc, pwd, stty, tty, printenv	50
30 GNU Core Utilities: Maths with seq, factor and numfmt	52
31 GNU Core Utilities: Conditionals with test, expr, true and false	53
32 GNU Core Utilities: SELinux with runcon and chcon	54
33 GNU Core Utilities: Printing with echo, printf and yes	55
34 GNU Core Utilities: tee	56
35 GNU Core Utilities: sleep	57

<i>CONTENTS</i>	4
XII Managing users and groups with shadow-utils	58
36 Home directories in <code>/root</code> and <code>/home/!user!</code>	59
37 <code>/etc/passwd</code> and <code>/etc/shadow</code> , and shadow-utils:	60
38 Setting passwords and logging out with <code>logout</code> and shadow-utils: <code>passwd</code>	61
39 Making and removing other users with shadow-utils: <code>useradd</code> and <code>suserdel</code>	62
40 Using groups with shadow utils: <code>usermod</code> , <code>groupadd</code> , <code>groupdel</code> , <code>groupmod</code> , <code>groups</code> , <code>gpasswd</code>	63
XIII GNU coreutils: groups and users	64
41 GNU Core Utilities: <code>who</code> and <code>whoami</code> , <code>chmod</code> , <code>chgrp</code> , <code>chown</code> , <code>users</code> , <code>logname</code> , <code>id</code> , <code>groups</code> , <code>pinky</code>	65
XIV Pluggable Authentication Modules (PAM)	66
42 Pluggable Authentication Modules (PAM)	67
XV GNU findutils	68
43 GNU findutils: <code>xargs</code> , <code>find</code> , <code>locate</code> , <code>updatedb</code>	69
XVI GNU text editors	70
44 <code>ed</code> , <code>ex</code> and <code>vi</code>	71
45 GNU <code>nano</code>	73
XVII <code>procs</code>	74
46 <code>procs</code> with <code>pgrep</code> , <code>pkill</code> , <code>pidwait</code> , <code>sysctl</code> , <code>free</code> , <code>top</code> , <code>watch</code> , <code>ps</code>	75
XVIII Other GNU programs	76
47 GNU <code>man-db</code> : <code>man</code> and <code>whatis</code>	77

<i>CONTENTS</i>	5
48 grep	78
49 sed	79
50 sudo, the /etc/sudoers file and disabling root login	80
51 GNU which	81
XIX Scripting	82
52 awk	83
XX Additional shells	84
53 Bourne-Again Shell (bash), including commands: history	85
54 csh and tsch	87
55 ksh	88
56 zsh	89
XXI Archiving and compressing	90
57 File archiving using GNU pax-archive: pax, tar and cpio	91
58 Compression using GNU zip (gzip): gzip, gunzip and zcat	92
59 tar and zip	93
XXII Systemd	94
60 Systemd	95
XXIII Alternatives to Systemd	98
61 cron	99

Preface

This is a live document, and is full of gaps, mistakes, typos etc.

Part I

To DOS

Chapter 1

DOS fdisk

1.1 Introduction

1.1.1 Introduction

Same name as util-linux program.

Part II

File systems, block devices and tools for partitioning and formatting drives

Chapter 2

File systems including ext2

2.1 Introduction

2.1.1 Introduction

What they are. tree? heap? concept of mapping from path to file

2.1.2 Garbage management of files

The link count is stored per file. If there are zero links to a file then the file system manager knows that the file can be deleted.

2.1.3 inodes

Bad blocks are noted in inode1.

Root is inode 2.

inode 0 is null.

Within a partition, each file or folder has a unique inode.

Each partition divided into blocks. numbered from 0. Blocks are the minimum size for readable or writable operations. Changing a block means needing to read the whole block, making a change and then rewriting the block.

Files can be stored across many blocks. Block don't have to be next to each other.

Because a block is the minimum size of any operation, there can only be one file per block, and each file takes up at least one block.

File names are the property of folders.

2.1.4 ext

2.1.5 ext2

Chapter 3

Journalling file systems including ext4

3.1 Introduction

3.1.1 Introduction

3.1.2 ext3

3.1.3 ext4

Chapter 4

Partitioning drives with GNU parted and util-linux: fdisk and cfdisk, and lsblk, wipefs and fsck

4.1 Partitioning drives using util-linux

4.1.1 Introduction

three options for partitioning are fdisk, gdisk and parted. parted generally seems the better option.

root partition if uefi, efi system partition (boot partition). also need boot partition if doing LVM or encryption on BIOS swap, though this is discussed later

4.1.2 fdisk

Same name as DOS fdisk.

fdisk is designed with MBR in mind, but later versions have some GPT support:

- `fdisk -l` (list things in `/dev/`) (or can use `lsblk`)
- `fdisk /dev/sda` (or whatever correct device is)
- this opens dialogue:
 - “d” to delete partitions
 - create a new table, using MBR or GPT

- create partitions (can press "n" for new)
- make one bootable
- "w" to write"

4.1.3 cfdisk

Curses ndisk

4.1.4 lsblk

See devices in /dev/

4.1.5 fsck

Fix file system.

4.1.6 gdisk

gdisk is similar to fdisk but aimed at GPT (is it part of util-linux though?)

4.1.7 wipefs

4.2 GNU parted

4.2.1 parted

partitioning using parted:

- supports MBR and GPT
- different to fdisk? needed if drives over 2TB?
- parted -l (list things in /dev/) (or can use lsblk)
- parted /dev/sda (or whatever correct device is)
- this opens dialogue:
 - see status with "print"
 - type "quit" when done
 - make gpt using "mklabel gpt"
 - make mbr using "mklabel msdos"
 - make partitions: "mkpart". is interactive
 - make one bootable? "set {partition} boot on"

Chapter 5

Formatting partitions using util-linux: mkfs

5.1 Formatting drives

5.1.1 Introduction

once partitions have been made, they show up on /dev/

5.1.2 Making ext4 partitions

```
mkfs.ext4 /dev/<textless_root_partition>
```

5.1.3 Making FAT partitions

also for grub boot?

```
mkfs.fat -F 32 /dev/<efi_system_partition>
```

5.1.4 Swap

swap, though this is discussed later

Part III

UEFI and non-BIOS first-stage bootloaders, and second-stage boot loaders

Chapter 6

More first-stage boot loaders: UEFI and coreboot/libreboot

6.1 Introduction

6.1.1 Unified Extensible Firmware Interface (UEFI)

Supports Secure Boot.

If `/sys/firmware/efi/` exists, the system is an EFI computer. Modern systems are UEFI rather than BIOS.

UEFI stores data in `.efi` file located in a hard drive, not a rom like in bios.

UEFI file stored in EFI system partition (ESP).

UEFI runs in 32/64 bit. BIOS in 16 bit. means uefi can support mouse and GUI.

UEFI supports disks over 2TB.

6.1.2 Coreboot and Libreboot

6.1.3 Android boot loaders

Chapter 7

The boot partition and second-stage boot loaders, including GRUB

7.1 Introduction

7.1.1 Introduction

The first-stage bootloader, eg BIOS, looks for a second-stage bootloader to load on a disk.

The second-stage bootloader loads the linux kernel then runs "init".

7.1.2 GRand Unified Bootloader (GRUB)

GRUB is a second-stage bootloader.

If the drive is partitioned using MBR, it is stored in the MBR.

With BIOS and GPT, there needs to be a separate boot partition for it. With UEFI and GPT, it can sit in the EFI partition.

7.2 Other

7.2.1 GRUB config

There are config files associated with GRUB:

- /etc/default/grub
- /etc/grub.d/

Running `update-grub` can reflect changes in the boot path.

7.2.2 EFISTUB

Allows EFI firmware to load kernel as EFI executable.

7.2.3 memtest

Run `memtest` from `grub`

Part IV

Memory Mapping Units (MMUs):

Part V

Address Generation Units (AGUs):

Part VI

Linux kernel

Chapter 8

Loading the Linux kernel from the boot partition

8.1 Introduction

8.1.1 Introduction

linux kernel hugepages + bigger than 4k standard + page table entry on linux
memory + Translation Lookaside Buffer + transparent hugepages

linux kernel stuff: + i/o subsystem stuff around files: * "generic block layer"
* "block device drivers" * i/o scheduler + memory management subsystem *
virtual memory * paging page replacement * page cache + process management
subsystem * signal handling * process/thread creation and termination * process
scheduler + IRQ (interrupt requests?) and dispatcher

Chapter 9

Directory layout on Linux: /boot, /sbin, /proc, /sys, /etc and /lib

9.1 Introduction

9.1.1 Introduction

9.1.2 /sbin

/sbin is where main binaries are stored.

9.1.3 /proc

/proc has kernel files?

9.1.4 /sys

9.1.5 /lib

/lib has libraries for /sbin.

9.1.6 /etc

/etc has conf?

Chapter 10

The init process, openrc and runit, and mounting using /etc/fstab

10.1 Introduction

10.1.1 Introduction

/sbin/init /etc/init/ /etc/init.d/ /etc/inittab

10.1.2 openrc

10.1.3 runit

10.1.4 /etc/fstab

Example from arch wiki:

# <device>	<dir>	<type>	<options>	<dump>	<fsck>
UUID=0a3407de-014b-458b-b5c1-848e92a327a3	/	ext4	noatime	0	1
UUID=f9fe0b69-a280-415d-a03a-a32752370dee	none	swap	defaults	0	0
UUID=b411dc99-f0a0-4c87-9e05-184977be8539	/home	ext4	noatime	0	2

Devices can also be eg /dev/sda2, but UUIDs safer.

dump refers to backing up disks

fsck says whether there should be a check first. 0 means no. 1 means 1 and is root. 2 means yes and is not root.

Options include:

+ rw (read and write) + suid (use set user IDs and group IDs from file system)
+ dev ("Interpret character or block special devices on the filesystem") + exec
(allow execution of binaries) + auto (can mount with -a) + nouser (don't allow
normal user to mount) + async

The option "defaults" uses all of these

Part VII

User space

Chapter 11

Using swap partitions with util-linux: mkswap, swapon and swapoff, and swapfiles

11.1 Introduction

11.1.1 Introduction

can use swap file or swap partition

mount swap:

+ need spare partition in partition table

```
mkswap /dev/<swap_partition>
```

```
swapon /dev/swap_partition
```

11.2 /etc/fstab

11.2.1 fstab

Can add entry into /etc/fstab.

Eg:

```
UUID=device_UUID none swap defaults 0 0
```

11.3 swapfiles

11.3.1 Introduction

Chapter 12

/dev/shm, /tmp and tmpfs

12.1 Introduction

12.1.1 Introduction

Part VIII

Linux multi stuff?

Chapter 13

Batch processing

13.1 Introduction

13.1.1 Introduction

multiple programs set to run one after another. virtual memory (and pages)
here? something on segmentation faults

DOS is like this maybe?

Chapter 14

Interrupts

14.1 Introduction

14.1.1 Introduction

Swap between processes (eg if user says to swap during, waiting for input, or printing). multi process needed for system management if even running 1 job? thread safety. address space layout randomisation. privilege. memory protection. avoiding deadlocks. job scheduler

Chapter 15

Concurrency control

15.1 Introduction

15.1.1 Introduction

separate to parallel or multi threading. overlapping lifetimes of programs can cause

Part IX

Pseudo-character devices

Chapter 16

Pseudo-character device files

16.1 Introduction

16.1.1 Introduction

character device file. just buffer for input buffer and output buffer. are fifo buffers. eg keyboard and printer of characters are character device files.

16.1.2 Specifics

/dev/zero

/dev/null

/dev/random

/dev/urandom

/dev/tty* + Terminals

/dev/pt* + Pseudo terminals

/dev/lb* + Line printers

/dev/fb* + Frame buffers

Chapter 17

Loop devices

17.1 Introduction

17.1.1 Introduction

`/dev/loop<x>`

Part X

Shells

Chapter 18

Interactive login shells,
read-eval-print loop
(REPL), the Bourne shell
implementations ash and
dash, including commands:
cd, fg, exit, jobs

18.1 Introduction

18.1.1 Introduction

ctrl z to sleep

18.1.2 jobs

”jobs” command to see sleeping jobs. can wake up with fg

18.1.3 fg

wakes up sleeping things. (short for foreground)

18.1.4 Introduction

shebang at top.

18.1.5 Pipes

|

18.1.6 Multiple jobs

multiple commands (&), trailing &,

18.1.7 Control flow

&&

||

control flow in sh (do while, case, for loop).

18.1.8 Writing to files

write to file with > (overwrite) and >> (append),
direct stderr to stdout with 2>&1.

raise error?

18.1.9 Getting interactive input

getting input from user as part of script. doing so in password way to hide input.

18.1.10 Variables

defining variables.

env.

18.1.11 Functions

functions.

18.1.12 Passing variables to shell scripts

passing variables to sh script (-, -?)

18.1.13 xargs

18.1.14 Stream and batch data

stream vs batch data here or elsewhere?

18.1.15 Other commands

exit. sleep? timer?

Chapter 19

Keyboards and locales

19.1 Introduction

19.1.1 Introduction

loadkeys

locale-gen function to eg set languages. see locale using "locale"

Chapter 20

Other util-linux programs,
including lscpu; more;
mount and umount; dmesg;
hwclock; kill; whereis; cal;
fallocate; su; chsh

20.1 Introduction

20.1.1 mount and umount

`mount /dev/<thing> /mnt/<name>`

can use `-mkdir`

`mount --mkdir`

20.1.2 kill

20.1.3 dmesg

Show kernel messages

CHAPTER 20. OTHER UTIL-LINUX PROGRAMS, INCLUDING LSCPU; MORE; MOUNT AND UMOUN

20.1.4 more

20.1.5 whereis

20.1.6 cal

20.1.7 su

run as different user

20.1.8 hwclock

20.1.9 lscpu

20.1.10 fallocate

20.1.11 chsh and /etc/shells

CChoose SHell.

Valid shells listed in /etc/shells

Chapter 21

Linux modules using kmod: lsmod, insmod, rmmod, modprobe and modinfo

21.1 Introduction

21.1.1 Linux modules

mods are in /lib/modules/

21.1.2 Linux module commands

Show loaded modules

`lsmod`

install mods

`insmod`

`rmmod`

load mod and dependencies

`modprobe`

Get information on a module

`modinfo`

Part XI

GNU coreutils: Basics

Chapter 22

GNU Core Utilities: Exploring folders using ls and dir, vdir, dircolors, du and stat

22.1 Introduction

22.1.1 Introduction

22.1.2 pwd

22.1.3 ls and dir

22.1.4 du

sizes of files in folder.

22.1.5 stat

Chapter 23

GNU Core Utilities: Reading files using cat, tac, head and tail, and nl, od, base32, base64 and basenc

23.1 Introduction

23.1.1 Introduction

23.1.2 cat

23.1.3 head and tail

head(first x lines)

tail(last x lines)

23.1.4 nl

Number of lines. Prints file along with line number.

Chapter 24

GNU Core Utilities: Writing to files using cp, dd and install, mv, rm and shred, mkdir, rmdir, touch and ln, readlink, mknod, mkfifo, mktemp, sync, link, unlink, truncate, split and csplit

24.1 Introduction

24.1.1 cp, dd and install

24.1.2 mv

24.1.3 rm and shred

24.1.4 mkdir

24.1.5 rmdir

24.1.6 touch

24.1.7 ln

24.1.8 readlink

Expands symlinks.

Chapter 25

GNU Core Utilities: Reading and transforming text using tr, cut, split, ptx, sort, tsort, expand, unexpand and uniq, fmt, pr and fold

25.1 Introduction

25.1.1 tr

25.1.2 cut

25.1.3 split

25.1.4 sort

Sort lines of text files.

25.1.5 uniq

Return unique lines only.

Chapter 26

GNU Core Utilities: Reading from multiple files using paste, comm and join

26.1 Introduction

26.1.1 Introduction

Chapter 27

GNU Core Utilities: Summarising files with `wc` and checksums (`sum`, `cksum`, `b2sum`, `md5sum`, `sha1sum`, `sha224sum`, `sha256sum`, `sha512sum`)

27.1 Introduction

27.1.1 Introduction

27.1.2 `md5sum`

27.1.3 `sha1sum`

27.1.4 `sha256sum`

27.1.5 `sha512sum`

27.1.6 `crc32sum`

Chapter 28

GNU Core Utilities: Modifying command invocation with chroot (and jails), env, nice, nohup, stdbuf and timeout

28.1 Introduction

28.1.1 Introduction

chroot (changes apparent root for processes, chroot jail?)

Chapter 29

GNU Core Utilities: Getting system information with df, date, uptime, uname, env, printenv, nproc, pwd, stty, tty, printenv

29.1 Introduction

29.1.1 df

amount of Disk Free)

29.1.2 date

29.1.3 uptime

29.1.4 uname

to get info on kernel etc

arch (same as uname -m)

CHAPTER 29. GNU CORE UTILITIES: GETTING SYSTEM INFORMATION WITH DF, DATE, UPTIME

29.1.5 env and printenv

29.1.6 nproc

Chapter 30

GNU Core Utilities: Maths with seq, factor and numfmt

30.1 Introduction

30.1.1 Introduction

Chapter 31

GNU Core Utilities: Conditionals with test, expr, true and false

31.1 Introduction

31.1.1 Introduction

Chapter 32

GNU Core Utilities: SELinux with runcon and chcon

32.1 Introduction

32.1.1 Introduction

Chapter 33

GNU Core Utilities: Printing with echo, printf and yes

33.1 Introduction

33.1.1 Introduction

Chapter 34

GNU Core Utilities: tee

34.1 Introduction

34.1.1 Introduction

Send things to standard output and files (ie T pipe).

Chapter 35

GNU Core Utilities: sleep

35.1 Introduction

35.1.1 Introduction

Sleep for specified time.

Part XII

Managing users and groups with shadow-utils

Chapter 36

Home directories in `/root` and `/home/!user!`

36.1 Introduction

36.1.1 Introduction

`/root` is root home directory.

`/home/[user]` folders.

Chapter 37

/etc/passwd and /etc/shadow, and shadow-utils:

37.1 Introduction

37.1.1 Introduction

37.1.2 /etc/passwd

Contains user names, full names, home directories and user shells.

Readable by anyone.

Used to contain hashes of passwords, but not anymore because vulnerable to dictionary attacks.

37.1.3 /etc/shadow

Contains user names and hashed passwords.

Only readable by root.

Chapter 38

Setting passwords and logging out with `logout` and `shadow-utils: passwd`

38.1 Introduction

38.1.1 `passwd`

is file with info on users

`/etc/passwd`

contains hash of password: `/etc/shadow`

38.1.2 `logout`

Chapter 39

Making and removing other users with shadow-utils: useradd and suserdel

39.1 Introduction

39.1.1 useradd

useradd

39.1.2 userdel

userdel

what happens to files with user as owner?

Chapter 40

Using groups with shadow utils: usermod, usermod, groupadd, groupdel, groupmod, groups, gpasswd

40.1 Introduction

40.1.1 Introduction

usermod to add user to group

users have primary group associated with just them, usually same name. can change using usermod.

groupadd, groupdel, groupmod, 777 etc. what happens to file when group deleted? command groups shows what groups a user is in

cont groups gpasswd to set passwords for groups. /etc/groups, /etc/gshadow

Part XIII

GNU coreutils: groups and users

Chapter 41

GNU Core Utilities: who and whoami, chmod, chgrp, chown, users, logname, id, groups, pinky

41.1 Introduction

41.1.1 who

Who is logged in and what they are doing.

41.1.2 whoami

Part XIV

Pluggable Authentication Modules (PAM)

Chapter 42

Pluggable Authentication Modules (PAM)

42.1 Introduction

42.1.1 Introduction

Part XV

GNU findutils

Chapter 43

GNU findutils: xargs, find, locate, updatedb

43.1 Introduction

43.1.1 xargs

43.1.2 find

43.1.3 locate

43.1.4 updatedb

Part XVI

GNU text editors

Chapter 44

ed, ex and vi

44.1 ed

44.1.1 Introduction

44.2 ex

44.2.1 Introduction

44.3 vi

44.3.1 Introduction

zz; ZZ

text folding.

44.3.2 Input mode

exit with escape

44.3.3 Basic editing

cursor before or after

i/a

new line above or below

O/o

undo

u

44.3.4 Command mode

quit

:q

write and quit

:wq

quit without saving

:q!

vi has .swp files. swap files. recovery file for file being edited.

vi copy paste

44.3.5 Opening other files

:e[dit] FILE_PATH

:vi[sual] FILE_PATH

Chapter 45

GNU nano

45.1 Introduction

45.1.1 Introduction

Part XVII

procps

Chapter 46

procps with pgrep, pkill, pidwait, sysctl, free, top, watch, ps

46.1 Introduction

46.1.1 Introduction

46.1.2 ps

See processes one off then return to terminal.

46.1.3 free

free is different to top because dumps to out, not interactive.

”free -m” shows free memory. distinction between free memory and available memory. free memory often very low because linux uses ram where possible

Part XVIII

Other GNU programs

Chapter 47

GNU man-db: man and whatis

47.1 Introduction

47.1.1 Introduction

47.1.2 man

47.1.3 whatis

one line version of man

Chapter 48

grep

48.1 Introduction

48.1.1 Introduction

g/re/p (globally search for a regular expression and print matching lines) grep, egrep, fgrep

Chapter 49

sed

49.1 Introduction

49.1.1 sed

minor scripting options but not central to concept. is regex thing.

Chapter 50

sudo, the /etc/sudoers file and disabling root login

50.1 Introduction

50.1.1 sudo

Chapter 51

GNU which

51.1 Introduction

51.1.1 Introduction

prints what would have been executed if the command was typed, with full path

Part XIX

Scripting

Chapter 52

awk

52.1 Introduction

52.1.1 Introduction

rewrite grep as awk command as an example (action a return all, pattern is regex) can do patterns inside actions too *0returnwholeline1* return first col
Maths in awk? rewrite sed in awk? rewrite cat etc in awk?

Part XX

Additional shells

Chapter 53

Bourne-Again Shell (bash), including commands: history

53.1 Introduction

53.1.1 Introduction

53.1.2 Prompt string

ps0/ps1/ps2/ps3/ps4

affect how terminal is presented:

ps0: what is displayed after command, before output ps1: what is displayed
before command (most used customisation)

53.1.3 `bash_history`

contains history of bash commands

`~/.bash_history`

53.1.4 `bashrc`

`~/.bashrc`

can customise prompt strings here.

53.1.5 bashprofile

.bashprofile

Chapter 54

csh and tsch

54.1 Introduction

54.1.1 Introduction

Chapter 55

ksh

55.1 Introduction

55.1.1 Introduction

Chapter 56

zsh

56.1 Introduction

56.1.1 Introduction

Part XXI

Archiving and compressing

Chapter 57

File archiving using GNU pax-archive: pax, tar and cpio

57.1 Introduction

57.1.1 Introduction

Makes multiple files into a single file.

57.1.2 pax

57.1.3 tar

Tape archive

xvf flags to untar

57.1.4 cpio

Chapter 58

Compression using GNU zip (gzip): gzip, gunzip and zcat

58.1 Introduction

58.1.1 Introduction

58.1.2 gzip and gunzip

58.1.3 zcat

Chapter 59

tar and zip

59.1 Introduction

59.1.1 tar

59.1.2 zip

Part XXII

Systemd

Chapter 60

Systemd

60.1 Introduction

60.1.1 Introduction

replaces init

systemd:

+ /usr/lib/systemd/system/ + /etc/systemd/system/

systemd init system (doesn't have runlevels)

/lib/systemd/system/nginx.service;

/etc/systemd/system/multi-user.target.wants/nginx.service;

/etc/inittab not on systemd

60.2 Replacing cron with systemd

60.2.1 General commands

List installed

```
systemctl list-unit-files
```

```
systemctl status
```

running

```
systemctl list-units
```

```
systemctl daemon-reload
```

see if thing failed:

```
systemctl --failed
```

60.2.2 journalctl

see logs: journalctl (part of systemd?)

```
journalctl
```

60.2.3 Unit specific commands

```
systemctl status <unit>
```

```
systemctl help <help>
```

```
systemctl is-enabled <unit>
```

```
systemctl start <unit>
```

```
systemctl start <unit>
```

```
systemctl stop <unit>
```

```
systemctl restart <unit>
```

```
systemctl reload <unit>
```

Starts at boot, or starts now.

```
systemctl enable <unit>
```

```
systemctl enable <unit> --now
```

```
systemctl disable <unit>
```

```
systemctl reenabale <unit>
```

```
systemctl mask <unit>
```

```
systemctl unmask <unit>
```

```
systemctl edit <unit>
```

```
systemctl revert <unit>
```

60.3 Replacing GRUB with systemd-boot

60.3.1 systemd-boot

Alternative to GRUB which supports UEFI.

60.3.2 systemd-stub

60.4 The systemd implementation of /tmp

60.4.1 Introduction

60.5 Mounting with systemd

60.5.1 Introduction

systemd-gpt-auto-generator

systemd.automount

Requires GPT.

If using systemd, don't need to manually create swap for partition in /etc/fstab, systemd will find it by going through partitions

Doesn't replace /etc/fstab, but means don't need to include drives on GPT there, or swap.

60.6 systemd-cryptenroll

60.6.1 Introduction

Can manage physical security tokens and passwords for LUKS2.

60.7 systemd-homed

60.7.1 Introduction

Allows the creation of portable users.

Part XXIII

Alternatives to Systemd

Chapter 61

cron

61.1 Introduction

61.1.1 Introduction