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

Adam Boult (www.bou.lt)

March 23, 2024

Contents

Part I

To DOS

Chapter 1

Partitioning using Master Boot Record (MBR)

1.1 Introduction

1.1.1 Intro

Master boot record is start of disk which says how disks sectors aka blocks are laid out.

Chapter 2

File systems including FAT32 (File Allocation Table)

2.1 Introduction

2.1.1 FAT8

The table is a linked list of cluster locations. at end of linked list is end of file. cluster iss fixed amount of space. bitness is size of values linking to clusters. max size = bitness * cluster size.

No garbage management or inodes.

2.1.2 FAT12

Introduced after FAT8.

2.1.3 FAT32

Introduced after FAT12.

Chapter 3

First-stage boot loaders and motherboard firmware, including BIOS

3.1 Introduction

3.1.1 Introduction

The motherboard has the first-stage boot loader. This if firmware with its own read-only memory.

3.1.2 Basic Input/Output System (BIOS)

Chapter 4

DOS fdisk

4.1 Introduction

4.1.1 Introduction

Same name as util-linux program.

Part II

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

Chapter 5

File systems including ext2

5.1 Introduction

5.1.1 Introduction

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

5.1.2 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.

5.1.3 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 mimimum 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.

5.1.4 ext

5.1.5 ext2

Chapter 6

Journalling file systems including ext4

6.1 Introduction

6.1.1 Introduction

6.1.2 ext3

6.1.3 ext4

Chapter 7

Partitioning using Globally Unique IDentifiers (GUID) Partion Table (GPT)

7.1 Introduction

7.1.1 MBR and GPT

Drives are divided into sectors. Each sector contains a fixed number of bytes, eg 4096 bytes.

File systems can be partitioned using either MBR or GPT:

- Master Boot Record (MBR)
- Globally Unique IDentifiers (GUID) Partition Table (GPT)

GPT is newer than MBR.

7.1.2 Boot sectors

The first sector of a disk is the boot sector. This applies to both MBR and GPT.

GPT does not use the boot sector, and it is just kept for compatibility reasons. An EFI system partition is used instead, as discussed later.

Chapter 8

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

8.1 Partitioning drives using util-linux

8.1.1 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

8.1.2 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"

8.1.3 cfdisk

Curses ndisk

8.1.4 lsblk

See devices in /dev/

8.1.5 fsck

Fix file system.

8.1.6 gdisk

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

8.1.7 wipefs

8.2 GNU parted

8.2.1 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 9

Formatting partitions using util-linux: mkfs

9.1 Formatting drives

9.1.1 Introduction

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

9.1.2 Making ext4 partitions

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

9.1.3 Making FAT partitions

also for grub boot?

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

9.1.4 Swap

swap, though this is discussed later

Part III

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

Chapter 10

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

10.1 Introduction

10.1.1 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.

10.1.2 Coreboot and Libreboot

10.1.3 Android boot loaders

Chapter 11

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

11.1 Introduction

11.1.1 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".

11.1.2 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.

11.2 Other

11.2.1 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.

11.2.2 EFISTUB

Allows EFI firmware to load kernel as EFI executable.

11.2.3 memtest

Run memtest from grub

Part IV

Memory Mapping Units (MMUs):

Part V

Address Generation Units (AGUs):

Part VI

Linux kernel

Chapter 12

Loading the Linux kernel from the boot partition

12.1 Introduction

12.1.1 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 arond files: * "generic block layer" * "block device drivers" * i/o scheduler + memory management subsystsem * 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 13

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

13.1 Introduction

13.1.1 Introduction

13.1.2 /sbin

/sbin is where main binaries are stored.

13.1.3 /proc

/proc has kernal files?

13.1.4 /sys

13.1.5 /lib

/lib has libraries for /sbin.

13.1.6 /etc

/etc has conf?

Chapter 14

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

14.1 Introduction

14.1.1 Introduction

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

14.1.2 openrc

14.1.3 runit

14.1.4 /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 15

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

15.1 Introduction

15.1.1 Introduction

can use swap file or swap partition

mount swap:

+ need spare partition in partition table

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

15.2 /etc/fstab

15.2.1 fstab

Can add entry into /etc/fstab.

Eg:

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

15.3 swapfiles

15.3.1 Introduction

Chapter 16

/dev/shm, /tmp and tmpfs

16.1 Introduction

16.1.1 Introduction

Part VIII

Linux multi stuff?

Chapter 17

Batch processing

17.1 Introduction

17.1.1 Introduction

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

DOS is like this maybe?

Chapter 18

Interrupts

18.1 Introduction

18.1.1 Introduction

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

Chapter 19

Concurrency control

19.1 Introduction

19.1.1 Introduction

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

Part IX

Pseudo-character devices

Chapter 20

Pseudo-character device files

20.1 Introduction

20.1.1 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.

20.1.2 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 21

Loop devices

21.1 Introduction

21.1.1 Introduction

/dev/loop<x>

Part X

Shells

Chapter 22

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

22.1 Introduction

22.1.1 Introduction

ctrl z to sleep

22.1.2 jobs

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

22.1.3 fg

wakes up sleeping things. (short for foreground)

22.1.4 Introduction

shebang at top.

22.1.5 Pipes

|

22.1.6 Multiple jobs

`multiple commands (&), trailing &,`

22.1.7 Control flow

&&
||

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

22.1.8 Writing to files

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

raise error?

22.1.9 Getting interactive input

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

22.1.10 Variables

defining variables.

env.

22.1.11 Functions

functions.

22.1.12 Passing variables to shell scripts

passing variables to sh script (-, –?)

22.1.13 xargs

22.1.14 Stream and batch data

stream vs batch data here or elsewhere?

22.1.15 Other commmands

exit. sleep? timer?

Chapter 23

Keyboards and locales

23.1 Introduction

23.1.1 Introduction

loadkeys

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

Chapter 24

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

24.1 Introduction

24.1.1 mount and umount

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

can use –mkdir

`mount --mkdir`

24.1.2 kill

24.1.3 dmesg

Show kernel messages

24.1.4 more

24.1.5 whereis

24.1.6 cal

24.1.7 su

run as different user

24.1.8 hwclock

24.1.9 lscpu

24.1.10 fallocate

24.1.11 chsh and /etc/shells

CHoose SHell.

Valid shells listed in /etc/shells

Chapter 25

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

25.1 Introduction

25.1.1 Linux modules

mods are in /lib/modules/

25.1.2 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 26

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

26.1 Introduction

26.1.1 Introduction

26.1.2 pwd

26.1.3 ls and dir

26.1.4 du

sizes of files in folder.

26.1.5 stat

Chapter 27

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

27.1 Introduction

27.1.1 Introduction

27.1.2 cat

27.1.3 head and tail

head(first x lines)

tail(last x lines)

27.1.4 nl

Number of lines. Prints file along with line number.

Chapter 28

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

28.1 Introduction

28.1.1 cp, dd and install

28.1.2 mv

28.1.3 rm and shred

28.1.4 mkdir

28.1.5 rmdir

28.1.6 touch

28.1.7 ln

28.1.8 readlink

Expands symlinks.

Chapter 29

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

29.1 Introduction

29.1.1 tr

29.1.2 cut

29.1.3 split

29.1.4 sort

Sort lines of text files.

29.1.5 uniq

Return unique lines only.

Chapter 30

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

30.1 Introduction

30.1.1 Introduction

Chapter 31

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

31.1 Introduction

31.1.1 Introduction

31.1.2 md5sum

31.1.3 sha1sum

31.1.4 sha256sum

31.1.5 sha512sum

31.1.6 crc32sum

Chapter 32

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

32.1 Introduction

32.1.1 Introduction

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

Chapter 33

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

33.1 Introduction

33.1.1 df

amount of Disk Free)

33.1.2 date

33.1.3 uptime

33.1.4 uname

to get info on kernel etc

arch (same as uname -m)

33.1.5 env and printenv

33.1.6 nproc

Chapter 34

GNU Core Utilities: Maths with seq, factor and numfmt

34.1 Introduction

34.1.1 Introduction

Chapter 35

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

35.1 Introduction

35.1.1 Introduction

Chapter 36

GNU Core Utilities: SELinux with runcon and chcon

36.1 Introduction

36.1.1 Introduction

Chapter 37

GNU Core Utilities: Printing with echo, printf and yes

37.1 Introduction

37.1.1 Introduction

Chapter 38

GNU Core Utilities: tee

38.1 Introduction

38.1.1 Introduction

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

Chapter 39

GNU Core Utilities: sleep

39.1 Introduction

39.1.1 Introduction

Sleep for specified time.

Part XII

Managing users and groups with shadow-utils

Chapter 40

Home directories in /root and /home/¡user¿

40.1 Introduction

40.1.1 Introduction

/root is root home directory.

/home/[user] folders.

Chapter 41

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

41.1 Introduction

41.1.1 Introduction

41.1.2 /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.

41.1.3 /etc/shadow

Contains user names and hashed passwords.

Only readable by root.

Chapter 42

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

42.1 Introduction

42.1.1 passwd

is file with info on users

/etc/passwd

contains hash of password: /etc/shadow

42.1.2 logout

Chapter 43

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

43.1 Introduction

43.1.1 useradd

`useradd`

43.1.2 userdel

`userdel`

what happens to files with user as owner?

Chapter 44

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

44.1 Introduction

44.1.1 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 45

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

45.1 Introduction

45.1.1 who

Who is logged in and what they are doing.

45.1.2 whoami

Part XIV

Pluggable Authentication Modules (PAM)

Chapter 46

Pluggable Authentication Modules (PAM)

46.1 Introduction

46.1.1 Introduction

Part XV

GNU findutils

Chapter 47

GNU findutils: xargs, find, locate, updatedb

47.1 Introduction

47.1.1 xargs

47.1.2 find

47.1.3 locate

47.1.4 updatedb

Part XVI

GNU text editors

Chapter 48

ed, ex and vi

48.1 ed

48.1.1 Introduction

48.2 ex

48.2.1 Introduction

48.3 vi

48.3.1 Introduction

zz; ZZ

text folding.

48.3.2 Input mode

exit with escape

48.3.3 Basic editing

cursor before or after

`i/a`

new line above or below

`O/o`

undo

`u`

48.3.4 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

48.3.5 Opening other files

`:e[dit] FILE_PATH`

`:vi[sual] FILE_PATH`

Chapter 49

GNU nano

49.1 Introduction

49.1.1 Introduction

Part XVII

procps

Chapter 50

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

50.1 Introduction

50.1.1 Introduction

50.1.2 ps

See processes one off then return to terminal.

50.1.3 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 51

GNU man-db: man and whatis

51.1 Introduction

51.1.1 Introduction

51.1.2 man

51.1.3 whatis

one line version of man

Chapter 52

grep

52.1 Introduction

52.1.1 Introduction

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

Chapter 53

sed

53.1 Introduction

53.1.1 sed

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

Chapter 54

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

54.1 Introduction

54.1.1 sudo

Chapter 55

GNU which

55.1 Introduction

55.1.1 Introduction

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

Part XIX

Scripting

Chapter 56

awk

56.1 Introduction

56.1.1 Introduction

rewrite grep as awk command as an example (actiona return all, pattern is regex) can do patterns inside actions too 0$returnwholeline$1 return first col Maths in awk? rewrite sed in awk? rewrite cat etc in awk?

Part XX

Additional shells

Chapter 57

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

57.1 Introduction

57.1.1 Introduction

57.1.2 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)

57.1.3 bash_history

contains history of bash commands

`~/.bash_history`

57.1.4 bashrc

`~/.bashrc`

can customise prompt strings here.

57.1.5 bashprofile

```
.bashprofile
```

Chapter 58

csh and tsch

58.1 Introduction

58.1.1 Introduction

Chapter 59

ksh

59.1 Introduction

59.1.1 Introduction

Chapter 60

zsh

60.1 Introduction

60.1.1 Introduction

Part XXI

Archiving and compressing

Chapter 61

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

61.1 Introduction

61.1.1 Introduction

Makes multiple files into a single file.

61.1.2 pax

61.1.3 tar

Tape archive

xvf flags to untar

61.1.4 cpio

Chapter 62

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

62.1 Introduction

62.1.1 Introduction

62.1.2 gzip and gunzip

62.1.3 zcat

Chapter 63

tar and zip

63.1 Introduction

63.1.1 tar

63.1.2 zip

Part XXII

Systemd

Chapter 64

Systemd

64.1 Introduction

64.1.1 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

64.2 Replacing cron with systemd

64.2.1 General commands

List installed

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

```
systemctl status
```

running

```
systemctl list-units
```

```
systemctl daemon-reload
```

see if thing failed:

```
systemctl --failed
```

64.2.2 journalctl

see logs: journalctl (part of systemd?)

```
journalctl
```

64.2.3 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 reenable <unit>
```

```
systemctl mask <unit>
```

```
systemctl unmask <unit>
```

```
systemctl edit <unit>
```

```
systemctl revert <unit>
```

64.3 Replacing GRUB with systemd-boot

64.3.1 systemd-boot

Alternative to GRUB which supports UEFI.

64.3.2 systemd-stub

64.4 The systemd implementation of /tmp

64.4.1 Introduction

64.5 Mounting with systemd

64.5.1 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.

64.6 systemd-cryptenroll

64.6.1 Introduction

Can manage physical security tokens and passwords for LUKS2.

64.7 systemd-homed

64.7.1 Introduction

Allows the creation of portable users.

Part XXIII

Alternatives to Systemd

Chapter 65

cron

65.1 Introduction

65.1.1 Introduction