

- * `lsmod` = list modules in kernel.
- * `insmod` = load a module.
- * `modprobe` = load dependencies & load the module
- * `rmmod` = unload the module.
- * `modinfo` = info of module.
- * `uname` = kernel version & info.
- * `lspci` = pci HW components.
- * `lsusb` = usb devices.
- * `dmesg` = log of things from kernel.
- * `menuconfig` = GUI for kernel configurations.

→ Device Field Engineer Interview. *

* Command line Environment. CLI

(1) Job Control.

Sleep: A process is sleeping / running in the background.

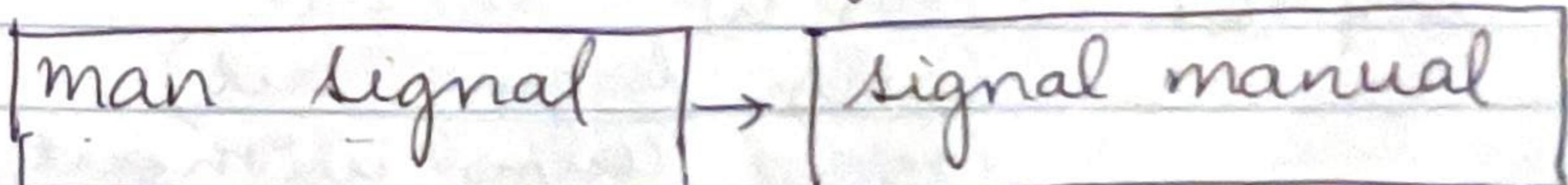
but if you want to stop it

`Ctrl + C`

This follows a Unix communication method called Signal

* Terminal sent a signal called

SIGINT: signal interrupt tells prog. to stop itself.



✓ SIGQUIT : quit program. (Ctrl + \)

✓ SIGTERM : Terminate (but not send through terminal)

Advantages in
remote sessions

ll-lah

ll=

(12)

SIGSTOP: Pause the execution of a program.

(bg %) SIGCONT: Continue the execution of a program.
↳ running in background.

SIGKILL: kill's prg

(8) Running in the background

INT QUIT HUP

CONT

Kill - STOP % 1

Pauses ~~stop~~ 1st process.

Kill - HUP % 1

↓
hangup

SIGHANGUP

SIGHUP: Hang up
(when logging out)
(reload config without
restarting the service)

Terminal multiplexer (tmux)

→ Sessions

→ Windows (are like tabs)

→ pane

to attach: [ctrl] + [A], [D]

to pane: [ctrl] + [A], ["]

↳ It is like a split screen.

Alias: alias ll="ll-lah" (nospaces
accepts single arg)

\$ alias ll = "ll-lah"

\$ ll
↳ alias is basically going to call
ll-lah.

\$ alias ll
↳ "ll-lah"

tells you what the alias is

Tomb

Dotfiles

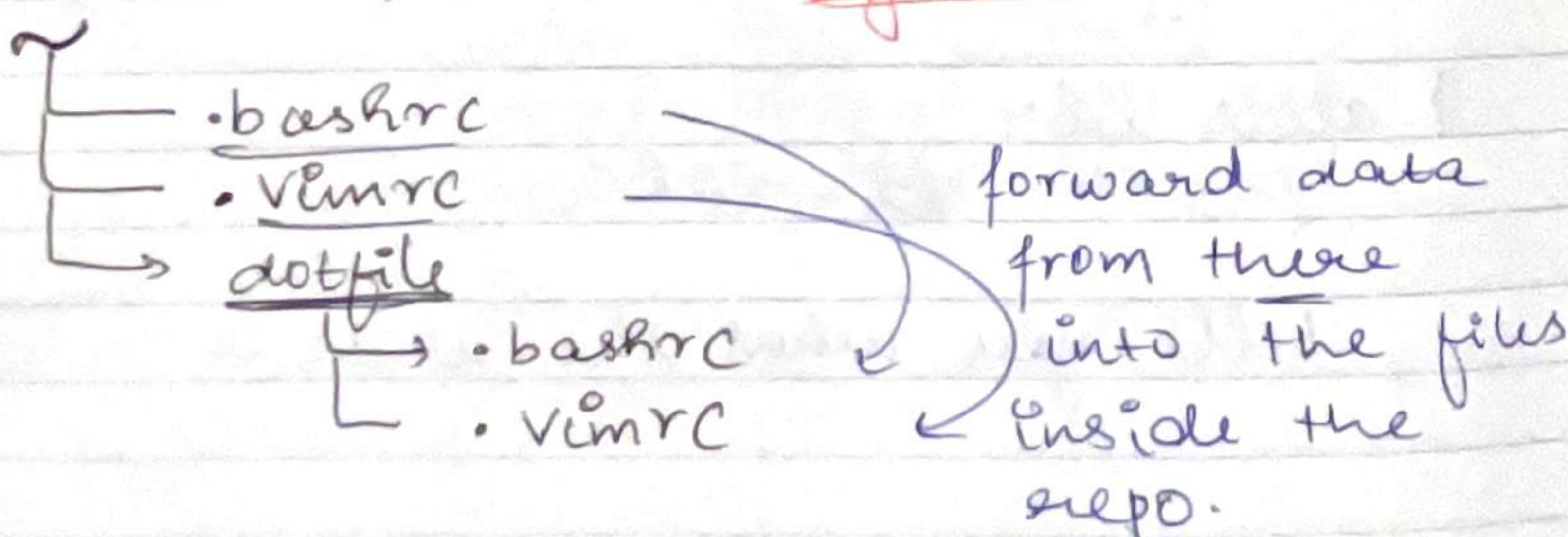
- bashrc } configuration file

`~/.bashrc` file is located in home directory.

→ We can add

- 1) Set alias
 - 2) Modify the prompts
 - 3) add dictionaries
 - 4) Customise env variable

Concept called Sym links



ln -s pathToFile SymLink

Special type of file that act as a pointer
(Short cut) to another file or directory
in linux.

Remote machines

→ SSH

↳ Secure Shell.

eg: ssh jj @ 192.168.246.142 → IP
ssh jj @ foobar.mit.edu
 { }
 username dns name

* SCP → SS + copy.

Linux in General: (GFG)

1) What is Linux

ans: Open source OS by Linus Torvalds.
It is unix like. ARM, x86.

2) Features of Linux

ans: Free, open-source, easily available.
Secure OS

↳ security auditing & pswd
authentication.

Personal software repo. (PPA)

CLI and firefox.

3) File permission in Linux

ans: Read 4
Write 2
Execute 1

4) What is Linux kernel?

ans: → customisable
→ middle layer b/w hardware and user

→ Subsystems:

- ↳ Process Scheduler
- ↳ Memory Management Unit
- ↳ Virtual file system
- ↳ The networking unit
- ↳ IPC unit.

few commands

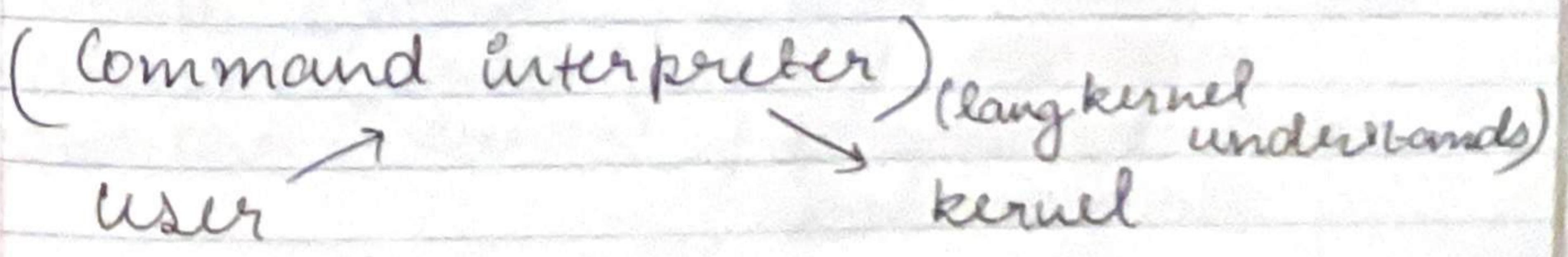
ls, pwd (path of working directory), cat
(Content of file), mkdir.

5) Explain LILO?

ans: LILO: linux loader.
It is the linux boot loader. It loads
linux OS into memory and starts
execution.

6) What is a shell?

ans: It is a user program that helps
user interact with kernel using
commands.



5 shells

1) **csh** (c shell)

Job control & spell check.

2) **ksh** (korn shell)

A high level shell for prg languages.

3) **ssh** (z shell)

4) **bash** (Bourne Again Shell)

Default shell for linux

5) **Fish** (Friendly interactive shell).
auto-suggestions.

7) **What is a swap space?**

ans: Linux uses extra space to expand
RAM.

↓
to hold concurrently
running processes.

8) difference b/w hard link and soft link.

ans:

Hard

Soft

- | | |
|---|--|
| ★ <u>faster</u> | ★ <u>slower</u> |
| ★ <u>includes original content</u> | ★ <u>includes path to the content</u> |
| ★ <u>uses less space</u> | ★ <u>uses more space</u> |
| ★ <u>Any change in link reflects in other files directly.</u> | ★ <u>Any change in links reflects in its hard link & original file directly.</u> |

9) How to create Symbolic link.

ans: Symlink or soft link are shortcuts to files and directories.

"ln -s path to file"

10) What are standard streams?

ans: Stdin (input) } communicate i/p
Stdout (output) } and o/p b/w
Std err (error) } programs & their environment

fdisk - l

- 11) how do you mount and umount file systems.

ans: use mount & umount

identify the partition (fdisk - l)

create a directory for mountpoint.

~~mkdir /mnt/mountpt~~

Sudo mount <partition> <mount-point - directory>

→ umount.

Sudo umount <mount-point - directory>.

Imp

12)

Troubleshoot Network connectivity in linux

ans:

1) Check Internet connection & cables.

2) check if network is configured

correctly or if your network interface has the correct IP.

3) ip route command to check default gateway is set properly.

4) DNS server config

/etc/resolv.conf

fdisk - l

- 5) check firewall if few rules are blocking access

↪ iptables.

- 6) restart network interface
↪ ifup or ifdown.

- 7) Reboot.

- 13) list all processes?

ans: ps

{
 ps - f full format
 ps - e displays all processes.
 ps auxf detailed list.

top

real-time data about process and resource usage.

htop

: colour-coded with few extra features.

- 14) chmod command?

ans: change file permissions

+,-,= 4 - read
r,w,x 2 - write
u,g,o,a 1 - execute

→ user
674 → group
→ other.

to make script executable +x.

eg. `chmod +x examples.sh`

verify changes. { ls - l }

15) How do you check disk space usage?

ans: (a) `df - disk full (-lh)`

used and available space.

(b) `du : disk usage`
(of current directory)

16) How to find process id

ans: `pgrep < process-1>`

17) rSync command.

ans? used to synchronise transfer
of file in linux.

Synchronise file b/w 2 directories
local system or a network.

`rSync (options) < sources < dest>`

18) How do you format disk.

ans:

- 1) "lsblk" list available partitions.
- 2) unmount any mounted disk.
- 3) find the type of file system EXT4, NTFS, XFS
- 4) run `"mkfs - <fileSys type> <partition>`

5) Mount disk again.

19) ulimit command?

ans: Increase / decrease the resource limit
`ulimit - u 50` } max number of
processes to 50.

20) find command

ans: Search for file using attributes

`find < directory > < file >`

Tomb

21) What is RAID in linux?

ans: Redundant Array independent disk
RAID 0, 1, 5, 6, 10

* used to increase system performance and data integrity.

RAID 0 → striping w/o redundancy
RAID 1 → Mirroring
RAID 5 → parity
RAID 6 → 2 set parity.
RAID 10 → striping & mirroring.

22) What is /proc file system?

ans: → VFS
→ information about system and kernel data structures.

→ interface to access system
→ perform debug tasks.
→ process related information.

23) What is strace command?

ans: Strace command is a diagnostic utility to trace and monitor system calls generated by a process.

- * how prg interact with kernel
- * debugging and troubleshooting task

24) How to administer linux servers?

ans:

- 1) Handle user acc & permissions
- 2) Security: firewall
- 3) System optimized for performance
- 4) Implement back up strategy
- 5) Resource management
- 6) Recovery plan.

25) What is virtual memory?

ans: It is a memory management technique give the appearance of large continuous block of memory to applications.

- * Supports multiprogramming.

26) iptables - Commands

ans: Netfilter firewall

Configure

→ netfilter firewall

Rules to NAT



packet filtering.

- * inspects network packets and manages them accordingly.

Imp

✓ 27) How to troubleshoot Linux OS that fails to boot

ans: (1) check warning/error message during boot process.

(2) check boot logs to find exact reason for error

(3) open GRUB bootloader and check boot options to solve booting problem.

(4) check hardware connections

(5) If error message is about kernel boot it with older version of kernel from GRUB.

(6) Identify last changes you made before boot.

28) Init process. → parent process

id = 1

ans: initialisation process, first process that begins when system boots.

→ System into a functioning state.

✓ 29) LVM? → logical volume for easy storage management.

ans: logical volume manager

↳ disk management approach in linux.

* subsystem that allows user to allocate disk space.

- {
- * resizing
 - * Volume mirroring
 - * snapshot
- }

30) grep

ans: Search for specific patterns in the file system or i/p stream.

grep "test" file.txt

✓ 31) how to check for status of service or daemon in linux.

ans: 'Systemctl' status <Service>
↳ can be used to start or stop service

32) What is the difference b/w /etc/passwd and /etc/shadow

<u>user information</u>	<u>encrypted passwd</u>
<u>home directory</u>	<u>security related detail</u>
<u>shell</u>	

✓ 33) how do you schedule recurring tasks in linux.

ans: Crontab

↳ specify when and how frequently a command or script should be executed.

34) Sed Command

ans: search and replace text.

Sed "s/foo/bar/g" file.txt

✓ 35) what are run levels in linux?

ans: Define different system states
↳ single user
↳ multiple user, etc.

also determine which program/service starts or stops during system startup and shutdown.

default: multuser - with GUI (RL5)

36) Network bonding

ans: Combining multiple network interface into a single interface. This improves throughput, ↑ bw, and ↑ redundancies.

Overall will work fine even if 1 fails.

37) SELinux? *look a little more
into this also AppArmor*

ans: Security - enhanced linux.
↳ security framework
↳ extra layer of security
↳ improve access control.
(prevent unauthorised access)

38) What is the purpose of sudoers
file in linux.

ans: Gives access to the file which
has permissions of ~~sudo~~ super
user.

Sudo Visudo

39) change ownership of file

ans: chown new user :userfile name
: user

new user : user file name-

40) ping

ans: To check for remote connection, sends
packets and waits to receive
corresponding packets.

✓ 41) netstat command?

ans: Shows active network connections,
routing table,
listening ports.

✓ 42) How to setup static IP.

ans: change configuration of network
interface. config file.
/etc/network/interface.

Trouble shooting Questions

1) How do you check system logs in linux?

ans: System logs
└── `/var/log`

└── `Syslog, messages, auth.log`
└── "tail" or "less" command.

2) Reasons of linux system running out of memory.

ans: memory leak, excessive memory usage
fragmentation

3) troubleshoot slow linux server

ans: tools like top
└── `htop`

└── Resource usage
└── monitor network traffic
└── find CPU bottleneck
└── Review app logs

4) Common causes of linux system running out of disk space

ans: large log files
fragmentation
memory leaks
runaway process generating excessive output
└── `fork()`

APT → advanced Package tool.
PPA → Personal Package Archive.

linux server

- user acc, pswd, authentication.
- firewall
- System is optimised for performance.
- Resource management.
- memory management & back up.
- Recovery plan.

Ubuntu and Debian

- ★ free
- ★ open source.

Debian

Desktop env

Multiple options
(GNOME, KDE)

HD req.

Minimal
good for older syst

install

Manual & custom

Release cycle

Conservative
stability focused

perform

lightweight, efficient

Ubuntu

GNOME with custom interface

Modern
opt for modern HD.

easy, stream-lined process

Regular with LTS

may use more resources

→ Debian

(1993)

- free
- open source.
- community driven.
- high Stability and security.
- server and critical app.

key features

1) Stability : long term support & robust.
(ideal for system w/o unexpected disruptions)

2) Package Management. APT
(APT) → advanced package tool.
Simple installation and updates.

3) Security : timely and reliable updates.

→ Ubuntu

(2004) 2004

- Created to make Linux easier to a broader audience.
- easy to use / user friendly.
- Snap packages. (Read more)
- also used for cloud support.

key features

- 1) LTS
- 2) user friendly.
- 3) APT Package management (Personal Package Arch)
- 4) Support multiple hardware Arch.

Common

- based on linux kernel
- (APT) package management
- open source

Debian

HD Req minimal req
suitable for resource constrained

Sudo users must add manually ✓

Security exceptional security and stable ✓

Package (APT), large packages. ✓

Perform opt for perform for older system. ✓

dpkg.

ubuntu.

higher req modern betty ✓

default user gets sudo privileges. ✓

Priorities security with regular updates. ✓

(APT), includes debian and more. ✓

designed for user friendly

apt.

Debian Versions

- Unstable : where new packages land.
- Testing : prepares for next stable rel.
- Stable : Production ready. ~2 year

ubuntu Versions

- Release every 2 months
- LTS comes every 2 years, 5+ year of support

Package Management

- debian
 - packages are built
 - ↳ debian unstable → Testing → Stable.
 - once packages are in stable they only get security updates.
 - deb managed by dpkg and APT.

- ubuntu
 - stabilize from debian unstable before ~~reach~~ ^{release}
 - * PPA - independent packages
 - (apt - get & apt)
 - (dpkg for low level)
 - (Snap for universal)
 - ↳ support via launch pad

~~book~~
launch pad: Web based platform.

/etc/lib/dpkg

Key directories

/etc/apt : APT config files

/etc/lib/dpkg : dpkg database

/var/cache/apt : Cached .deb files

/usr/bin : user installed binaries

backporting : using patches in newer version to fix bugs in older version.

Q) What are snap packages and why does ubuntu use them.

ans: Snaps are containerized packages that bundle dependencies

can run on multiple linux distribution

- ↳ cons : slower startups
- ↳ larger package size.

Q. How does the APT package manager work?

apt fetches, installs and removes packages

sudo apt update

upgrade

install <pkg>

remove <pkg>

autoremove

✓

✓

✓

✓

✓

Q. How to check which repo a package is from

ans: apt - show < pkg>

apt - show<pkg>

Q. How do you install packages from PPA.

ans: sudo add-apt

sudo apt update

sudo apt install <pkg>

sudo - add-apt
update
y

Q. How do you enable backports in ubuntu or debian?

ans: add backports repo to the source list

Q. How do you list installed pkg.

ans: apt list --installed

apt list -installed

Troubleshooting

Q. What do you do if package installation is broken?

ans: try fixing dependencies.

If that fails remove pkg.

Clean up and retry.

→ Embedded Systems Linux.

ubuntu 22.04 LTS

- * large overhead
- * Microprocessor usually required.
- * low control of hardware.
- * multiple threads and processes (portable application code)

SpaceX falcon 9 rely on linux to run.

- * Pre-made images
- * Write files yourself (punishment)
- * use tools to find / Generate files

Buildroot

OpenWrt → focused on networking gear
yocto project

{
steep learning curve
customizable
active community
}

Yocto project

→ Poky → bitbake (task execution function)
↳ reference distribution

↳ metadata layers
support packages
tools:

} used to generate images.

Boot process flow

1. Bootloader (U-Boot)

- * first piece of code executed after power up.
- * initialises hardware and loads the kernel
- * printenv, setenv, saveenv, bootcmd.
- { * know how to configure and compile U-boot for target board.

2. kernel

- * hardware abstraction layer HAL
- * handles process scheduling, memory management and drivers.

- * config kernel using 'menuconfig'

3. Rootfile Systems

- * contains user space app and libraries.
- * can be built using yocto, buildroot
- * ext 4.

linux boot process

next page

menu config

BIOS/UEFI → approach to disk storage different

common boot loader → LILO, GRUB2

1) Power ON ($\times 86$ / Systems)



2) BIOS / UEFI boots up.
(newer)

- * first piece of code to run.
- * initialise hardware.



3) POST (Power on self test)

- * checks if all parts are working
otherwise gives an error



4) Find and load the BOOTLOADER.

- hard drive → (most prob)
- * in the master boot record



5) Jobs for Bootloader.

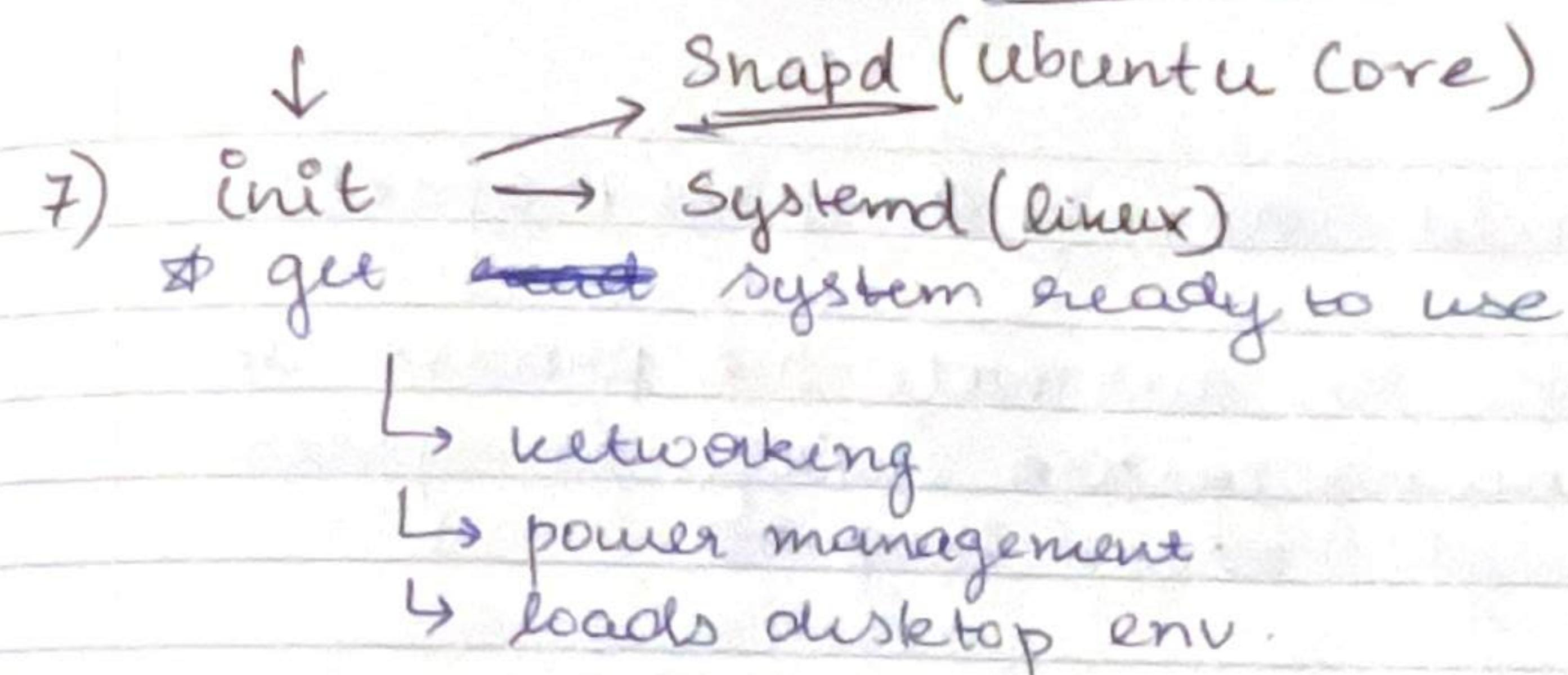
- * locate the OS kernel and load
- * start kernel code.



6) Kernel (DT)

- * decompress
- * take over the resources.
- * load device driver & other kernel modules.

GNU GRUB



In embedded System

(ARM)

1) Power on

2) Boot RDM (SoC - Specific)

- * small RDM code built-into the processor.
- * Searches for bootloader.

3) Boot loader

(U-BOOT)

- * initialises hardware
- * loads the kernel
- * loads the device tree

4) Kernel

- * initialises drivers and memory management

* mount root filesystem xfs

5) init

- ↳ Systemd
- ↳ Custom Script

* Buildroot and Yocto Project

Tool to automate the process of creating custom image for your single board computer.

→ Buildroot

- * ease of use
- * limited configuration settings
- * Requires full image rebuild.

→ Yocto Project → open embedded

- * Customizable (Build system and support for x86 and ARM.)
- * Active Community (Community)

✓ Q How would you modify the kernel configuration in Yocto?

ans:

- 1) Create a custom layer.
- 2) add the new layer to the build configuration.
- 3) Create a recipes-kernel append file.
- 4) modify kernel configuration.
 - (a) using configuration fragment.
 - (b) using a complete def config.

Q Power Management techniques.

Ans: → hardware based tech

- 1) Dynamic Voltage and frequency scaling.
- 2) Low power modes.
- 3) Clock gating
 - ↳ disable clock to idle component
- 4) Power gating.
 - ↳ cuts off power to unused block inside a chip.

DTS
DTC
DTB

→ Software based.

1) Dynamic power management.

2) Tickless kernel.

↳ MCU sleeps until event occurs.

3) Power aware task scheduling.

4) Peripheral power management.

~~Device Tree (Hardware description file)~~

* ~~list of devices that are available for the processor~~

→ ~~What hardware exists~~

→ ~~How devices are connected~~

→ ~~What drivers should be used~~

→ Device Tree → ~~ES~~ ES

* HD file in linux kernel to understand the hardware layout of an embedded system.

→ Tells us about hardware devices

→ how they are connected

(CPU's, memory, peripherals)

→ What drivers should be used.

* Why it should be used.

→ Hardware abstraction.

switch devices without changing kernel code.

→ device can be enabled/disabled by modifying kernel code

* Structure

→ Device tree source format

compiled into device tree blob that the bootloader passes to the kernel.

DTS

DTB

properties

- 1) compatible : identifies device type
(driver uses this)
- 2) reg : specifies memory and file.
- 3) interrupts : IRQ used by device.
- 4) status.
- 5) gpios : gpio pins assigned.

Q. How do you modify device tree for new peripheral?

- ans: → locate the DTS file
→ find the node related to the device
eg: SPI
→ change/modify the DTS file
 └ add all properties
→ enable SPI driver in kernel.

Q. how do you compile and load a device tree blob?

ans:

- 1) use a device tree compiler.
- 2) compile the DTS file with flags like -I, -O,
- 3) copy the DTB to the boot partition
- 4) load the new device tree
- 5) Reboot system.

DTC

Q. How do you debug device tree issues in embedded systems.

ans:

- 1) Check the loaded device tree
- 2) Check if device is detected
- 3) decompile and load
- 4) Check kernel logs.
- 5) Manually load the driver

Commands list

ls = list

ls -a = list all

pwd = path of working directory.

mkdir = to create new directory

pgrep = to get pid with P-name

iptable = network firewall config

grep = pattern in file system.

Systemctl = start, stop, status of service

Crontab = recurring processes.

Sed = search and replace text.

Sudo = Super user

sudo visudo = super user permission file

chown = change ownership

du = disk usage

df -f = disk full

ln -s = soft link & symbolic link

fdisk -f = identify partition

RT → top / htop / ps = process list

col RT ↴

* Strace = use to give trace of systemcalls

chmod = change permission of file

rsync = sync transfer of files

lsblk = list current disk layout

mkfs = format file/disk

iostat = check disk I/O

free -m = memory

chroot = change root

* ls

* ls -a { list all }

* ls -la { permissions }

* ls -l { permissions }

Filesys

/proc = System and kernel information

/etc/passwd = user info, home directory

/etc/shadow = encrypted passwords

/sys = system info
↳ CPU, firmware, devices

/etc/network/interface = network info.

→ /var/log = linux system logs

/etc/resolv.conf = DNS server config.