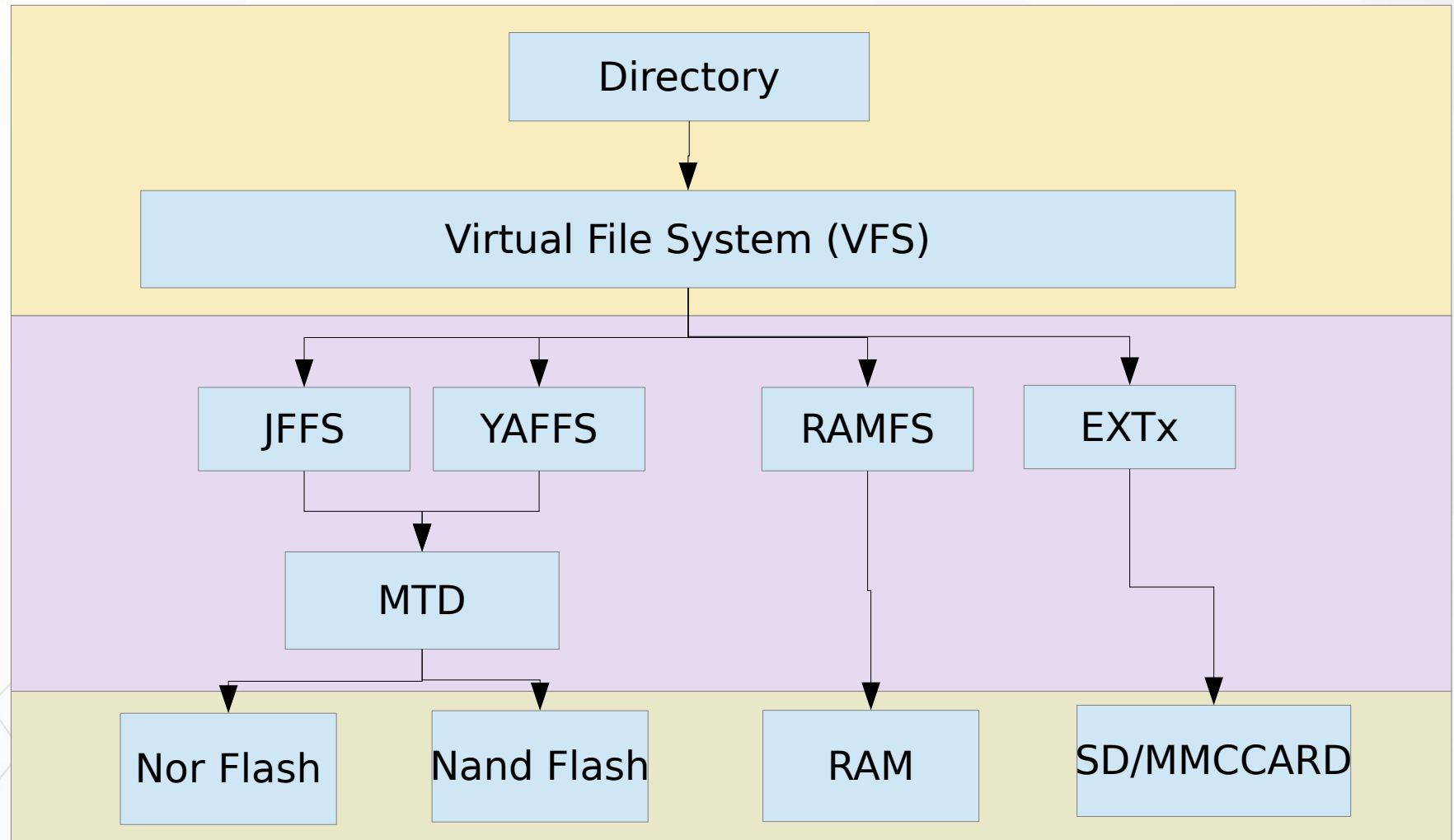


CH 7 Root File System

File System in Linux





File System

- A file system defines how files are **named**, **stored**, and **retrieved** from a storage device
- For desktop users
 - FAT32, NFS, EXT3, EXT4
- For embedded system
 - Cramfs
 - JFFS2
 - Squashfs
 - YAFFS2
 - EXT2, EXT3

File System and Kernel

➤ [CMD] make menuconfig

- File systems

```
[ ] Second extended fs support (NEW)
[ ] The Extended 3 (ext3) filesystem (NEW)
[ ] The Extended 4 (ext4) filesystem (NEW)
[ ] Reiserfs support (NEW)
[ ] JFS filesystem support (NEW)
[ ] XFS filesystem support (NEW)
[ ] GFS2 file system support (NEW)
[ ] Btrfs filesystem support (NEW)
[ ] NILFS2 file system support (NEW)
[ ] F2FS filesystem support (NEW)
[ ] Direct Access (DAX) support (NEW)
[ ] FS Encryption (Per-file encryption) (NEW)
[*] Dnotify support (NEW)
[*] Inotify support for userspace (NEW)
[ ] Filesystem wide access notification (NEW)
[ ] Quota support (NEW)
[ ] Kernel automounter version 4 support (also supports v3) (NEW)
[ ] FUSE (Filesystem in Userspace) support (NEW)
```

Mount a File System Driver

➤ Make sure which File-System be supported

- [CMD] cat /proc/filesystems

```
rock@rockpi4b:~$ cat /proc/filesystems
nodev    sysfs
nodev    rootfs
nodev    ramfs
nodev    bdev
nodev    proc
nodev    cpuset
nodev    cgroup
nodev    cgroup2
nodev    tmpfs
nodev    devtmpfs
nodev    configfs
nodev    debugfs
nodev    tracefs
nodev    securityfs
nodev    sockfs
nodev    pipefs
nodev    rpc_pipefs
nodev    devpts
        ext3
        ext2
        ext4
        squashfs
```



Mount a File System Driver

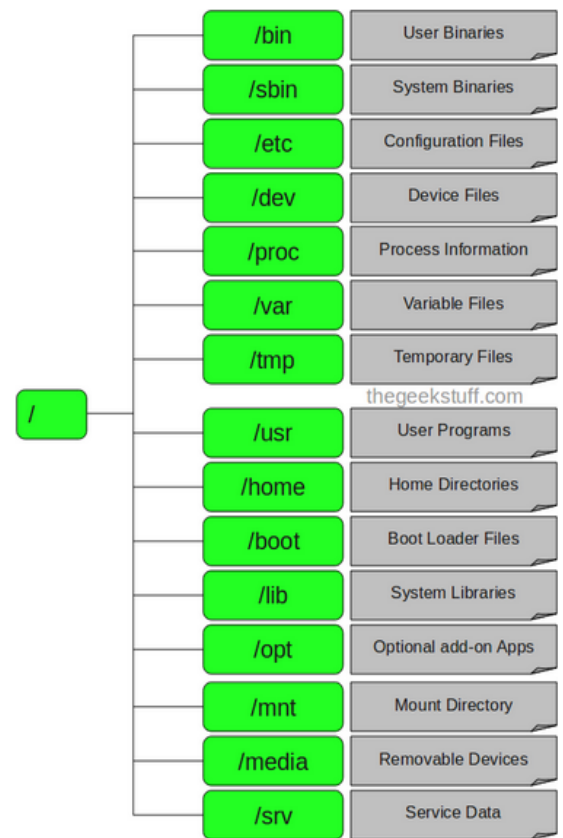


Make sure which File-System be supported

- `mount -t ${FILE_SYS_TYPE} ${DISK} ${MOUNT_FOLDER}`
- Fake Image_Disk Test
 - [CMD] `sudo dd if=/dev/zero of=/home/rock/vdidk.img bs=1M count=5 status=progress`
 - [CMD] `sudo mkfs.vfat ./vdidk.img`
 - **[CMD] `sudo mount -t vfat ./vdidk.img /tmp/vfat_folder/`**
 - [CMD] `lsblk -f`

Root File System

Root File System Structure





Root

- Every single file and directory starts from the root directory
- Only root user has write privilege under this directory
- Please note that /root is root user's home directory, which is not same as /



/bin – User Binaries

- Contains binary executables.
- Common linux commands you need to use in single-user modes are located under this directory.
- Commands used by all the users of the system are located here.
- For example: ps, ls, ping, grep, cp.



/sbin – System Binaries

- Just like /bin, /sbin also contains binary executables.
- But, the linux commands located under this directory are used typically by system administrator, for system maintenance purpose.
- For example: iptables, reboot, fdisk, ifconfig, swapon



/etc – Configuration Files

- Contains configuration files required by all programs.
- This also contains startup and shutdown shell scripts used to start/stop individual programs.
- For example: `/etc/resolv.conf`, `/etc/logrotate.conf`



/dev – Device Files

- Contains device files.
- These include terminal devices, usb, or any device attached to the system.
- For example: /dev/tty1, /dev/usbmon0



/proc – Process Information

- Contains information about system process.
- This is a pseudo filesystem contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.
- This is a virtual filesystem with text information about system resources. For example: /proc/uptime



/tmp – Temporary Files

- Directory that contains temporary files created by system and users.
- Files under this directory are deleted when system is rebooted.



/usr – User Programs

- Contains binaries, libraries, documentation, and source-code for second level programs.
- **/usr/bin** contains binary files for user programs. If you can't find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp
- **/usr/sbin** contains binary files for system administrators. If you can't find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel
- **/usr/lib** contains libraries for /usr/bin and /usr/sbin
- **/usr/local** contains users programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2



/home – Home Directories

- » Home directories for all users to store their personal files.
- » For example: /home/john, /home/nikita



/boot – Boot Loader Files

- » Contains boot loader related files.
- » Kernel initrd, vmlinuz, grub files are located under /boot



/lib – System Libraries

- Contains library files that supports the binaries located under /bin and /sbin
- Library filenames are either ld* or lib*.so.*

Linux System Initial Program

System V

» Unix System **Five (V)**

» AT&T developed

» The first initial process is → init

- PID=1

» SystemV handles startup processes through shell scripts in /etc/init*

- /etc/inittab

- /etc/init.d/ and /etc/init.d/rcS

System V

➤ /etc/init.d/S*

- initial system script

➤ Start a Service

- [CMD] etc/init.d/S50sshd start

[CMD] etc/init.d/S50sshd stop

```
[root@rk3399:/etc/init.d]# ls
S01logging      S22hdmion      S50link_iq      S80dnsmasq
S10init         S30dbus        S50sshd         S99input-event-daemon
S10udev         S40network     S50telnet       rcK
S20urandom      S41dhcpcd      S50usbdevice    rcS
S21mountall.sh  S50launcher_   S66load_wifi_modules
```

System D

➤ System Daemon

➤ /sbin/init -> /lib/systemd/systemd

➤ PID=1

➤ SystemD is the new system that many distros are moving to

➤ SystemD handles startup processes through .service files



System D Configure File

➤ The unit configuration files are loaded from a set of paths

- `"/lib/systemd/system"`:
 - OS default configuration files
- `"/etc/systemd/system"`:
 - system administrator configuration files
 - override the OS default
- `"/run/systemd/system"`:
 - un-time generated configuration files
 - override the installed configuration files

Service Control

 `systemctl ${CTL} ${SERVICE}`

- [CMD] `systemctl enable ssh`
- [CMD] `systemctl status ssh`
- [CMD] `systemctl start ssh`
- [CMD] `systemctl stop ssh`

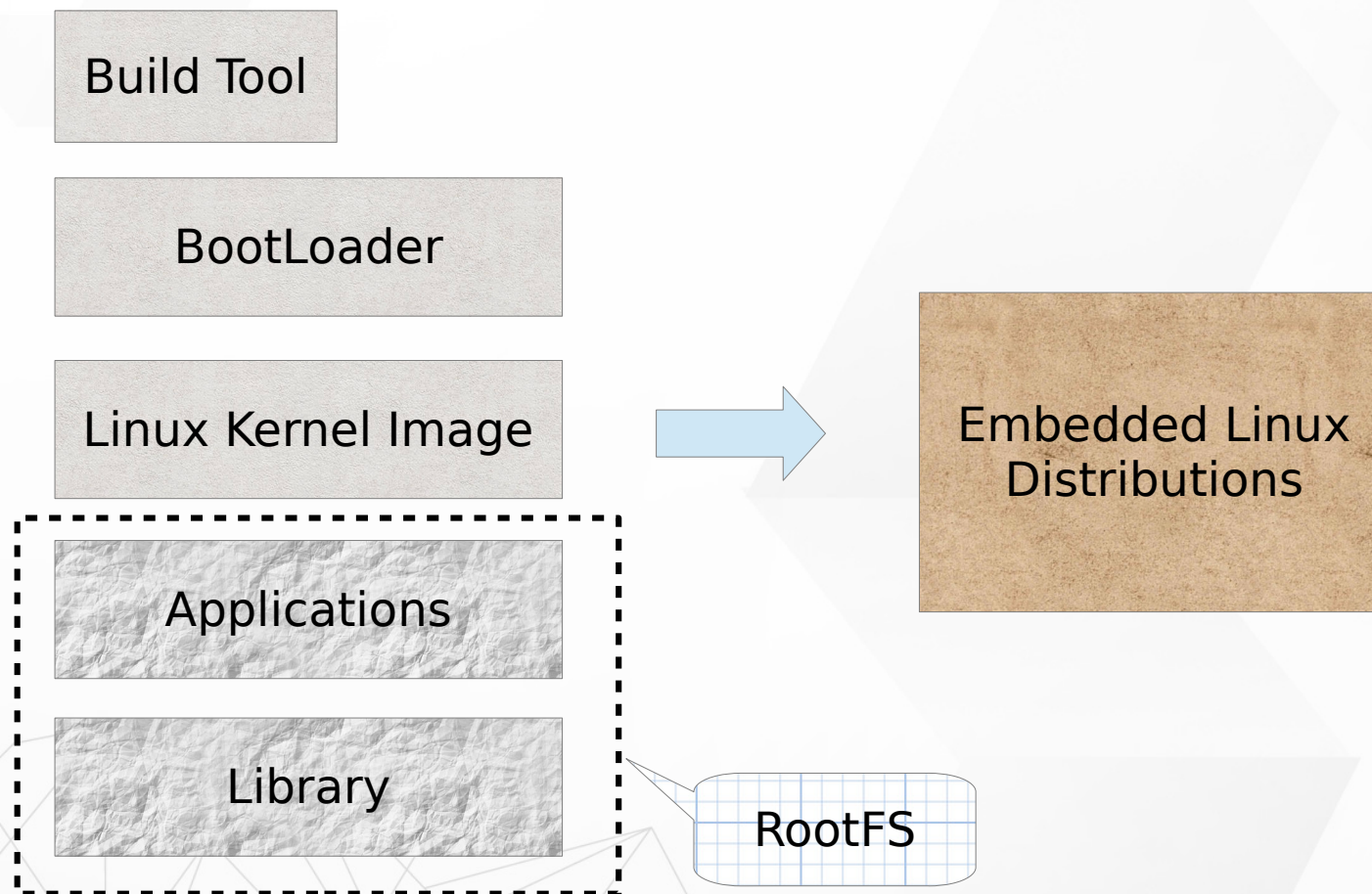
Linux Distribution



Linux Distribution

- Boot-loader
- Linux kernel
- RootFS
- Application
- Library
- Linux driver modules

Build Distribution by Tool



System integration

<https://bootlin.com/doc/training/buildroot/buildroot-slides.pdf>

	Pros	Cons
Building everything manually	Full flexibility Learning experience	Dependency hell Need to understand a lot of details Version compatibility Lack of reproducibility
Binary distribution Debian, Ubuntu, Fedora, etc.	Easy to create and extend	Hard to customize Hard to optimize (boot time, size) Hard to rebuild the full system from source Large system Uses native compilation (slow) No well-defined mechanism to generate an image Lots of mandatory dependencies Not available for all architectures
Build systems Buildroot, Yocto, PTXdist, etc.	Nearly full flexibility Built from source: customization and optimization are easy Fully reproducible Uses cross-compilation Have embedded specific packages not necessarily in desktop distros Make more features optional	Not as easy as a binary distribution Build time

Debian



RockPi4B Debian

➤ <https://wiki.radxa.com/Rockpi4/dev/Debian>

➤ <https://github.com/radxa/rk-rootfs-build/tree/rockchip-debian>