Operating System Installation

René Serral-Gracià Xavier Martorell-Bofill[1]

[1]Universitat Politècnica de Catalunya (UPC)

September 16, 2014

Lectures

1. System administration introduction
2. **Operating System installation**
3. User management
4. Application management
5. System monitoring
6. Filesystem Maintenance
7. Local services
8. Network services
9. Security and Protection
10. Virtualization

Outline

Outline

Goals

Abilities

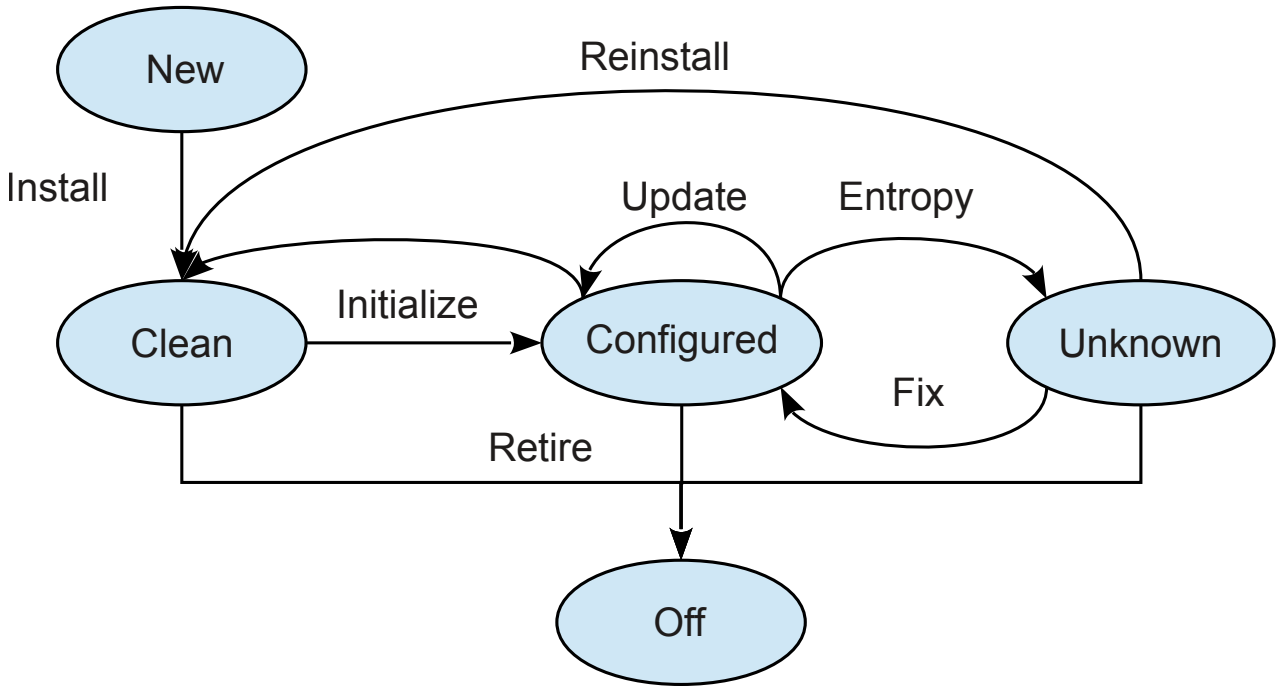
- Installation scheduling
  - Disk Partitioning
  - File System creation
  - Swap area dimensioning
- Basic configuration
  - System Startup and Shutdown

Configuration Commands and files

- `fdisk`, `mkfs`, `mkswap`, `mount`, `swapon`
- `shutdown`, `halt`, `reboot`, `poweroff`
- `init`, `/etc/inittab`, `/etc/rc*.d/`, `/etc/fstab`

Outline

Equipment Life-cycle[1]



- Sysadmin goals:
  - Understand the existence of the states and their transitions
  - Maximize the amount of time in the *"Configured"* state

[1] Rémy Evard. *"An analysis of UNIX system configuration"*. 11th Systems Administration Conference (LISA 97)

Equipment Life-cycle

States

- **New:** new equipment
- **Clean:** equipment with the installed OS but without any maintenance task
- **Configured:** configured equipment according to the environment requirements
- **Unknown:** unconfigured or outdated equipment
- **Off:** discarded equipment due to its age or hardware failure

Equipment Life-cycle

Transitions

- **Install:** OS installation
- **Initialize:** Initial set of required changes to have the equipment configured in the work environment
- **Update:** Insert new functionalities, apply patches and security updates
- **Entropy:** Gradual degradation process leaving the equipment in unknown state
- **Fix:** take the necessary actions to set the equipment back to configured state
- **Reinstall:** massive update of the OS. Usually forced by an attack, goal shift in the equipment, or configuration errors
- **Retire:** final retirement of the equipment

Outline

System installation

1. Goals
2. Dimensioning
3. HW Acquisition
4. Disk preparation
5. Protected network setup
6. Install / OS & Software update
7. Service configuration / adaptation
8. Security policy enforcement
9. Final location network setup
10. Label / Document the followed steps
11. Monitor... goto 5

Previous tasks

1. Goals

Which is the purpose of the new equipment?

- Desktop
  - Document editing?
  - Compiling?
- Server
  - E-mail? Web? Proxy? DNS? Files?
  - Primary? Secondary?
- Amount of expected users
- Security requirements

Previous tasks

2. Dimensioning
   - CPU
   - Memory
   - Disk
   - Redundancy
3. Buy HW
   - OS Compatibility (drivers!)
   - List of features
     - IRQs, DMA, and/or ports...

Installation

4. Disk preparation
   - Partitioning
   - Swap area preparation
   - Format and prepare the filesystems
5. Connect the equipment into a secure network
   - So during the installation the machine is protected
6. Install / Update OS & Software
   - Choose OS / Distribution
   - Select the package update list

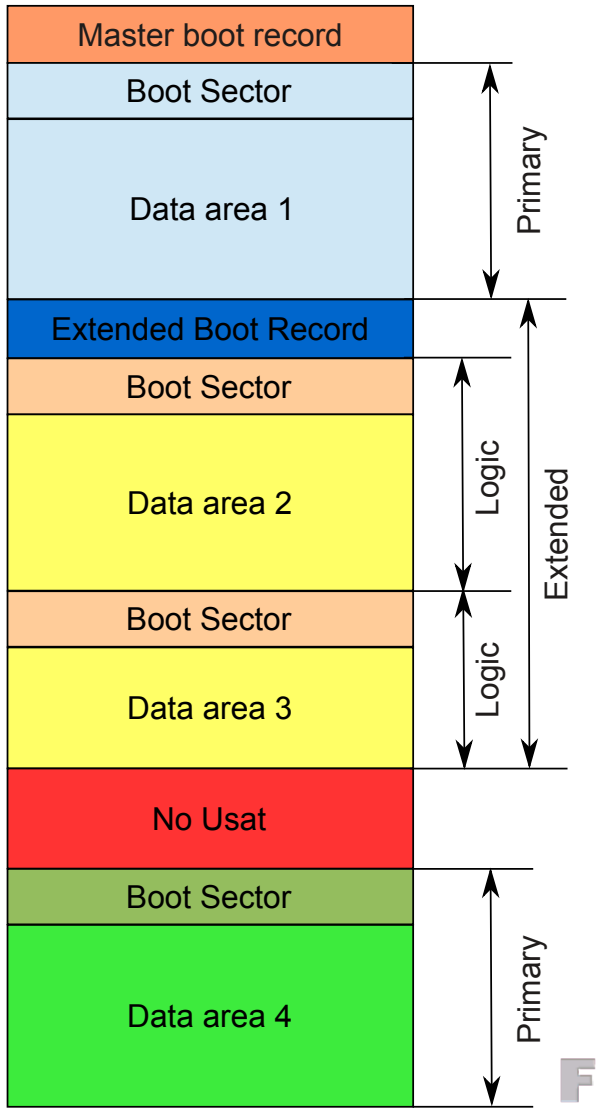
Installation

7. Service configuration
   - Adapt them to the work environment
8. Implement security policies
   - Offer only the necessary services
9. Connect to the network
   - To the final location
10. Label / Document the followed steps
    - In case it is necessary to repeat them, to apply them on other machines, ...
11. Monitoring... goto 6

Outline

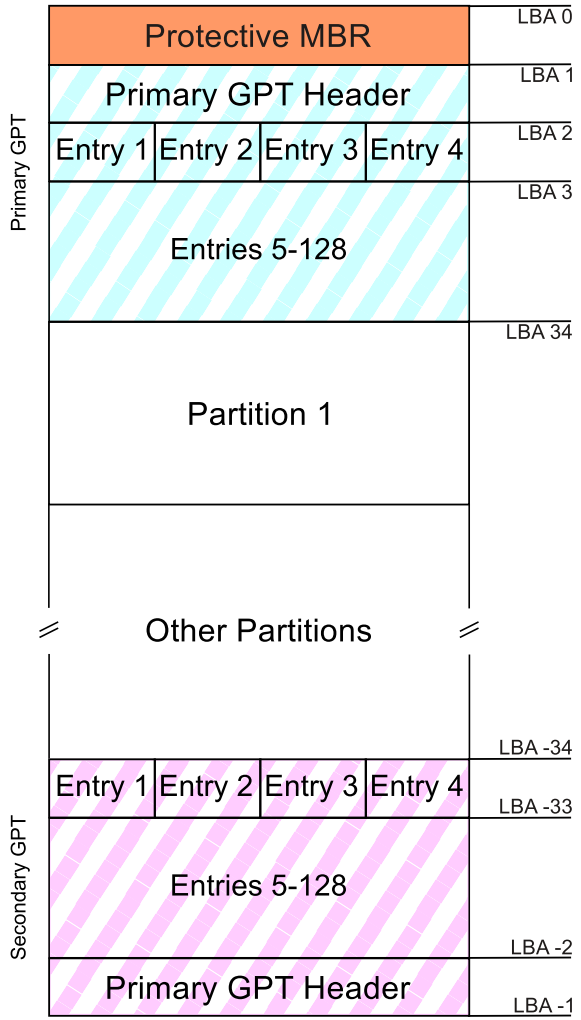
Types of partitions (PC)

- Up to 4 “primary” partitions in the Master Boot Record
  - Or 3 primary and 1 extended
  - . . . or 2 primary and 2 extended. . . (not supported by all OSs)
- Primary partition
  - May contain a filesystem
- Extended partition
  - Can only contain logical partitions
- Logical partitions
  - May contain a filesystem



Types of partitions – GUID Partition Table (GPT)

- Up to 128 partitions with the default size of GPT
- There is no disctintion of primary and extended partitions anymore, now it is identified by UUID
  - The partition type is determined by the Operating System, which assigns its own IDs



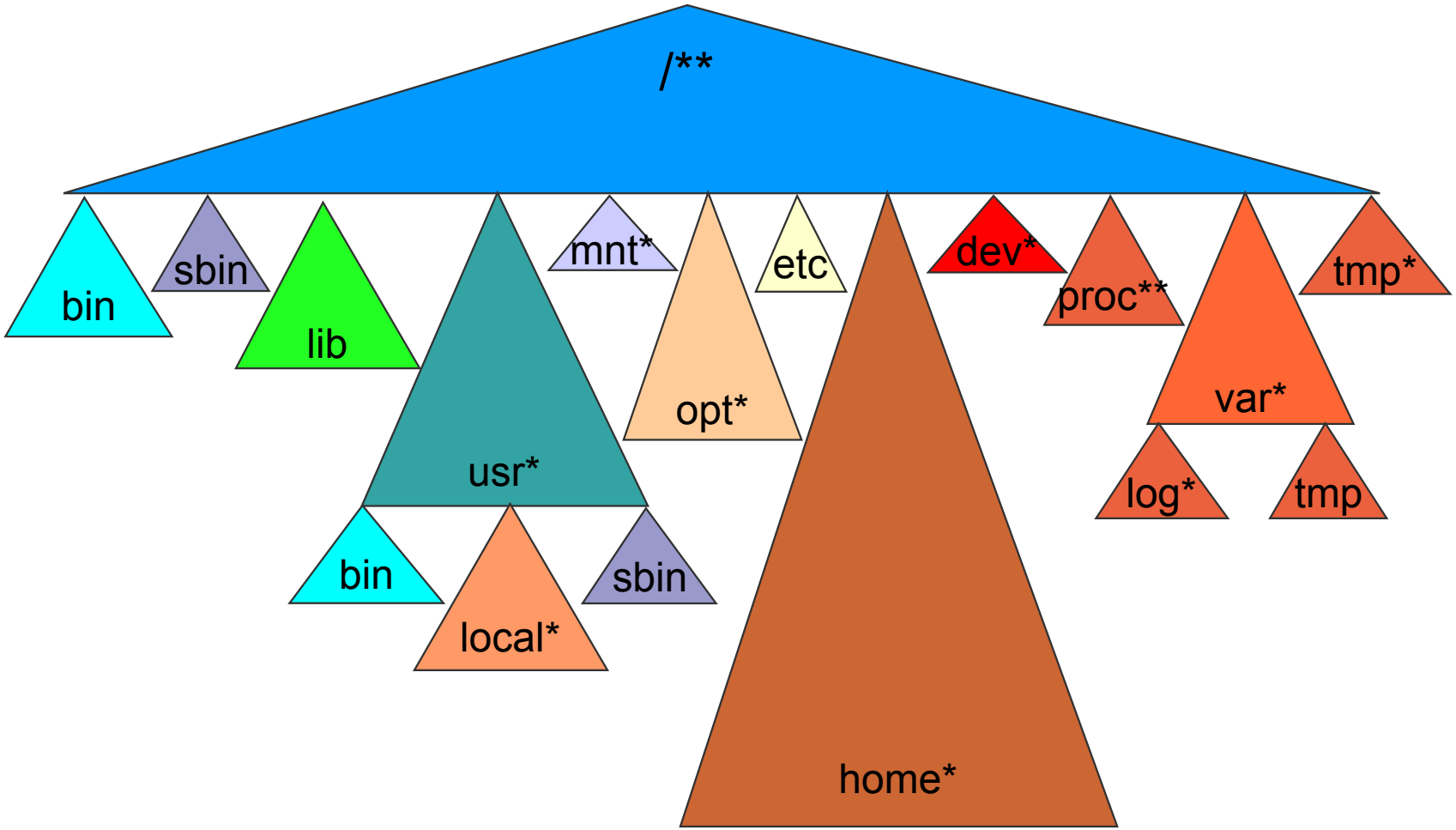
Partitions: concept and justification

Divide one disk into several independent disks

- Each partition is completely isolated from the others
  - Error isolation
  - More security
- Backup management different for each partition
  - Faster
  - More convenient
  - *Read-only* or not much changed partitions
- Information reuse among OS

Problem: hard disk fragmentation

Filesystem structure in UNIX



* Can be mounted filesystems
** Must be mounted filesystems

Filesystem structure in UNIX

- /bin and /sbin
  - Executables needed during boot time
  - `ifconfig`, `mount`, `ls`, `cat`, ...
- /usr/bin and /usr/sbin
  - Operating system applications
  - man, apropos, ...
  - adduser, deluser, ...
- /usr/local/bin and /usr/local/sbin (or /opt)
  - Specific applications
- $HOME/bin
  - End-user applications

Filesystem structure in UNIX

/var

- Dynamic content
  - Accounting
    - Information about end-user activity
  - Spool
    - Mail
    - Cron/at
    - lpd
  - Run
    - Pid's of running daemons
  - Log
    - System logs

Filesystem preparation/format

- **mkfs** *-t tipus [opcions] dispositiu*
  - type: ext3, ext4, reiserfs, vfat, brtfs,. . .
  - options (filesystem dependent)
    - block size
    - number of inode
    - number of blocks (usually autodetected)
    - . . .
- **tune2fs** *[-l] [-j]* . . .
  - Filesystem ext[234] parameter configuration
    - Filesystem check interval
    - Journal creation
    - . . .

Exercise – En grup

- If we put all the directories labelled with $*$ and $**$ in their own partition. Determine a correct size for each partition
- Why the rest of the directoris cannot be on a partition by themselves

Exercise – En grup

- If we put all the directories labelled with $*$ and $**$ in their own partition. Determine a correct size for each partition
  - Such size normally depends of the particular needs for that installation. Usually a regular Linux installation needs around 15*GB*
- Why the rest of the directoris cannot be on a partition by themselves
  - The content is necessary during the boot process. Potentially before mounting the filesystems

Mount

- **mount** *[options] device directory*
  - -t <filesystem type>
    - Indicate the type of the filesystem
  - -a
    - mount all the filesystems in /etc/fstab
  - -o <FS options>
    - ro = read-only
    - remount
    - noexec, nodev, nosuid
    - user

/etc/fstab

- Indicates how to mount the filesystems

Device	M. point	FS	Options	D	F
/dev/sda1	/boot	ext3	defaults	0	2
/dev/sda2	/	ext4	defaults	0	1
/dev/sda5	/var	ext3	defaults	0	2
/dev/sda6	/tmp	ext3	defaults	0	2
/dev/sda7	/home	ext3	defaults	0	2
none	/dev/pts	devpts	gid=5,mode=620	0	0
none	/proc	proc	defaults	0	0
none	/sys	sysfs	defaults	0	0
/dev/sda3	swap	swap	defaults	0	0
/dev/scd0	/mnt/cdrom	auto	ro,noauto,user	0	0

Exercise – In group

- We have a server with 100 users, with a disk quota of 5Gb per user. The system has a 1TB harddisk. Indicate how can you partition it and the size of each partition.

Exercise – In group

- We have a server with 100 users, with a disk quota of 5Gb per user. The system has a 1TB harddisk. Indicate how can you partition it and the size of each partition.
  - The users need a total of $\sim 500GB$. $\sim 5GB$ for the base system[2], then lacking more information we leave a total of $\sim 10GB$ for applications.
    Then we will have 3 different partitions, the root partition `/dev/sda1` with $6GB$, the user's partition `/dev/sda2` using $600Gb$, $12Gb$ for applications `/dev/sda5`, and finally $8GB$ for the swap partition`/dev/sda6`. We leave the rest of the disk unpartitioned
    For safety we leave a threshold of $10 - 20\%$ in terms of space for each partition

[2] Assuming a Linux Debian installation

Exercise – In group

- List the required commands in order to be able to mount the filesystems indicated in the previous exercise, knowing that the application partition must be read-only.

Exercise – In group

- List the required commands in order to be able to mount the filesystems indicated in the previous exercise, knowing that the application partition must be read-only.
  - `/dev/sda1` → it must be mounted from /etc/fstab
  - `/dev/sda2` → `mount /dev/sda2 /home`
  - `/dev/sda5` → `mount -o ro /dev/sda5 /usr`

Exercise – In group

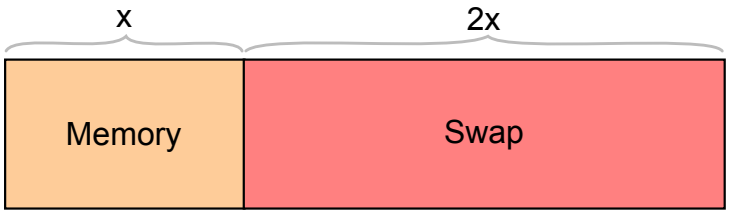
- List the required commands in order to be able to mount the filesystems indicated in the previous exercise, knowing that the application partition must be read-only.
  - `/dev/sda1` → it must be mounted from /etc/fstab
  - `/dev/sda2` → `mount /dev/sda2 /home`
  - `/dev/sda5` → `mount -o ro /dev/sda5 /usr`
- Can you devise any situation where more partitions could be necessary?

Exercise – In group

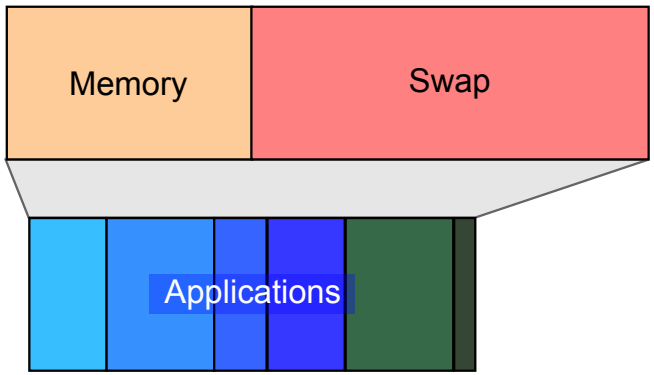
- List the required commands in order to be able to mount the filesystems indicated in the previous exercise, knowing that the application partition must be read-only.
  - `/dev/sda1` → it must be mounted from /etc/fstab
  - `/dev/sda2` → `mount /dev/sda2 /home`
  - `/dev/sda5` → `mount -o ro /dev/sda5 /usr`
- Can you devise any situation where more partitions could be necessary?
  - If the server had some specific requirements, for example a very large web page, we could be interested in having `/var/www` in a different partition

Swap area

- Rule of thumb
  - Swap = 2 * physical memory



- Realment
  - Foresee memory requirements and choose it accordingly



Swap area implementation

- As a disk partition
  - Better if divided into multiple devices
- Special file
  - Pre-created and completely reserved... it cannot have any "holes"
    - Holes??? in a file???

      ```
      dd if=/dev/zero of=swapfile bs=1024 count=65536
      ```
  - **Be careful!**
    - File protections
    - The is sensible information from the swapped out processes

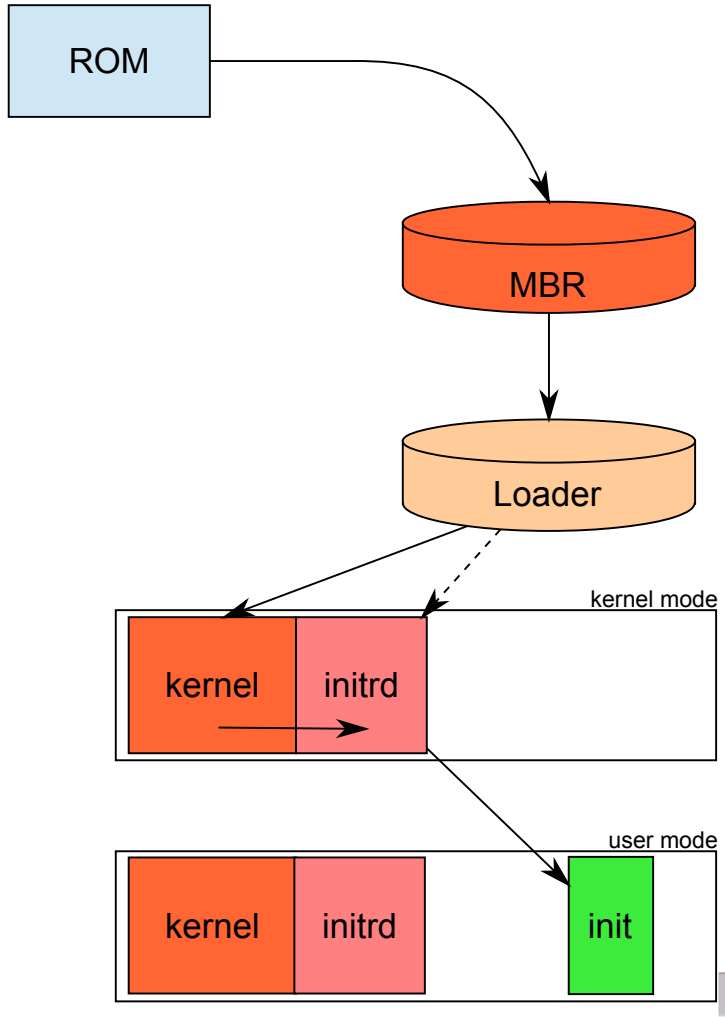
Swap area Creation/Preparation

- **mkswap** *device | file*
  - Creates a swap area — is equivalent to swap area “format”
- **swapon** *[options] [device | file]*
  - -p *priority*
    - The swap with more priority is used before
    - Round-Robin if equal priority
  - -a
    - Activates all the swaps defined in /etc/fstab
- **swapoff** *[options] [device | file]*
  - Disables a given swap area
  - -a
    - Disables all the ones defined into /etc/fstab

Outline

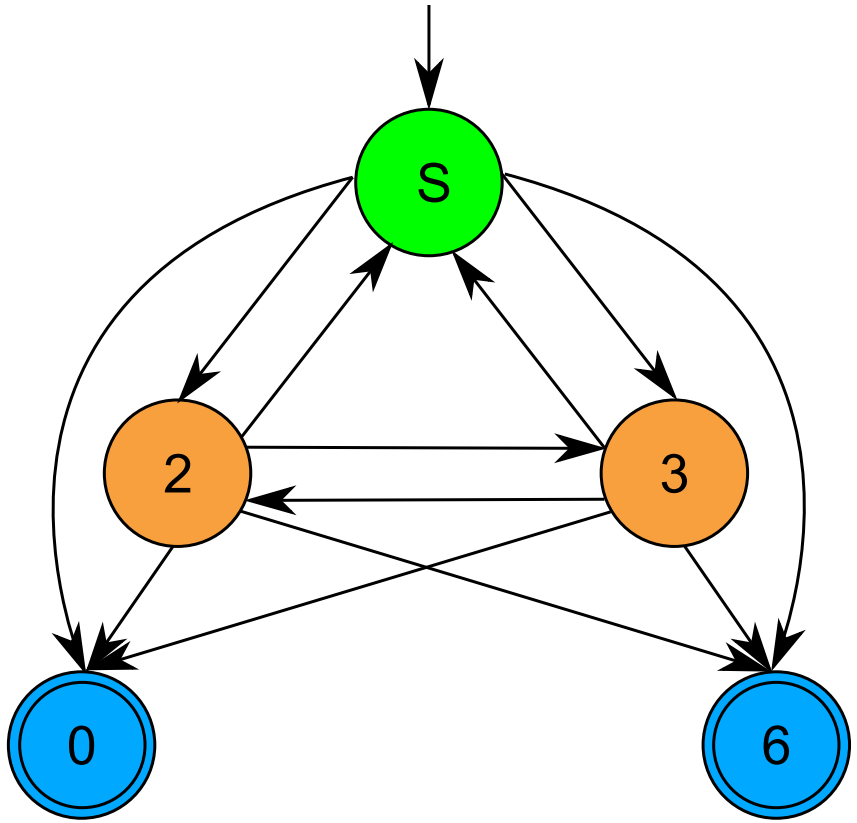
System initialization

- ROM
  - Hardware initialization
    - CPUs,. . .
- kernel
  - Hardware detection
  - Kernel mode configuration
- initrd
  - Device configuration
- init
  - User space configurations



Runlevel

- **init** it has diferent runlevel
  - S,1: single user
  - 2-5: multi-user
    - 2: without network
    - 3: with network
    - 4: network + X
  - 0: halt
  - 6: reboot
- **init** *run-level*
  - changes the runlevel



Init/shutdown Service Scripts

/etc/init.d

- Accept standard parameters
  - /etc/init.d/**servei** *start|stop|restart|reload|...*
    - **start:** starts the service
    - **stop:** stops the service
    - **restart:** stop+start
    - **reload:** if possible restarts the service without killing it (HUP)
  - And other specific to some services
    - **status**
    - **setup**
    - ...

Unix System-V init style

`/etc/rcX.d`

- One directory per runlevel
- Scripts running at runlevel X
  - Usually are soft-links to actual scripts in /etc/init.d
- The name indicates its priority (01-99)
  - [*S*|*K*] <priority>name
    e.g.: `S40networking`, `K74bluetooth`
- When changing the runlevel first the system runs the K and then S with priority order (small first – alphabetically)
- They can be managed using update-rc.d

  → Lab session

Dependency based init

Upstart

- Compatible with System V (Scripts and parameters)
- Totally asynchronous
- Service init/shutdown in parallel
  - Makefile style controlled dependencies
- It allows contron and monitoring of the running services

Systemd

- Partially compatible with System V or BSD
- Only available in Linux
- It allows hardware detection via `udev`

Initialization – End

/etc/rc.local

- Local configuration Shell script
  - Executed at the end of multiuser runlevels
- Example:

```
#!/bin/bash
# start hard drive temperature monitor daemon
/usr/local/bin/hddtemp -d /dev/sda

# In case hddtemp fails for any reason
exit 0
```

System shutdown

Actions to perform

- Stop all services — Network + locals
- Stop all the processes
- Sync all buffer caches
- Umount all the filesystem
- Stop/reboot the system

Commands

- **shutdown**: allows shutdown/reboot at a given time
- **reboot**, **halt**, **poweroff**, ...
  - Currently all optiosn use ACPI extensions
- init 0, init 6

Personal work

- Privileges and protection
  - Owners and groups
  - Privileges (r, w, x)
  - Umask
  - Setuid, setgid
- User management related commands
  - chmod, chown, id, newgrp
  - useradd/adduser, userdel
  - chfn, chsh, passwd
  - groupadd, groupdel