Tishk International University
Science Faculty
IT Department

Open Source OS (Linux)

Lecture04: Booting and Shutting Down

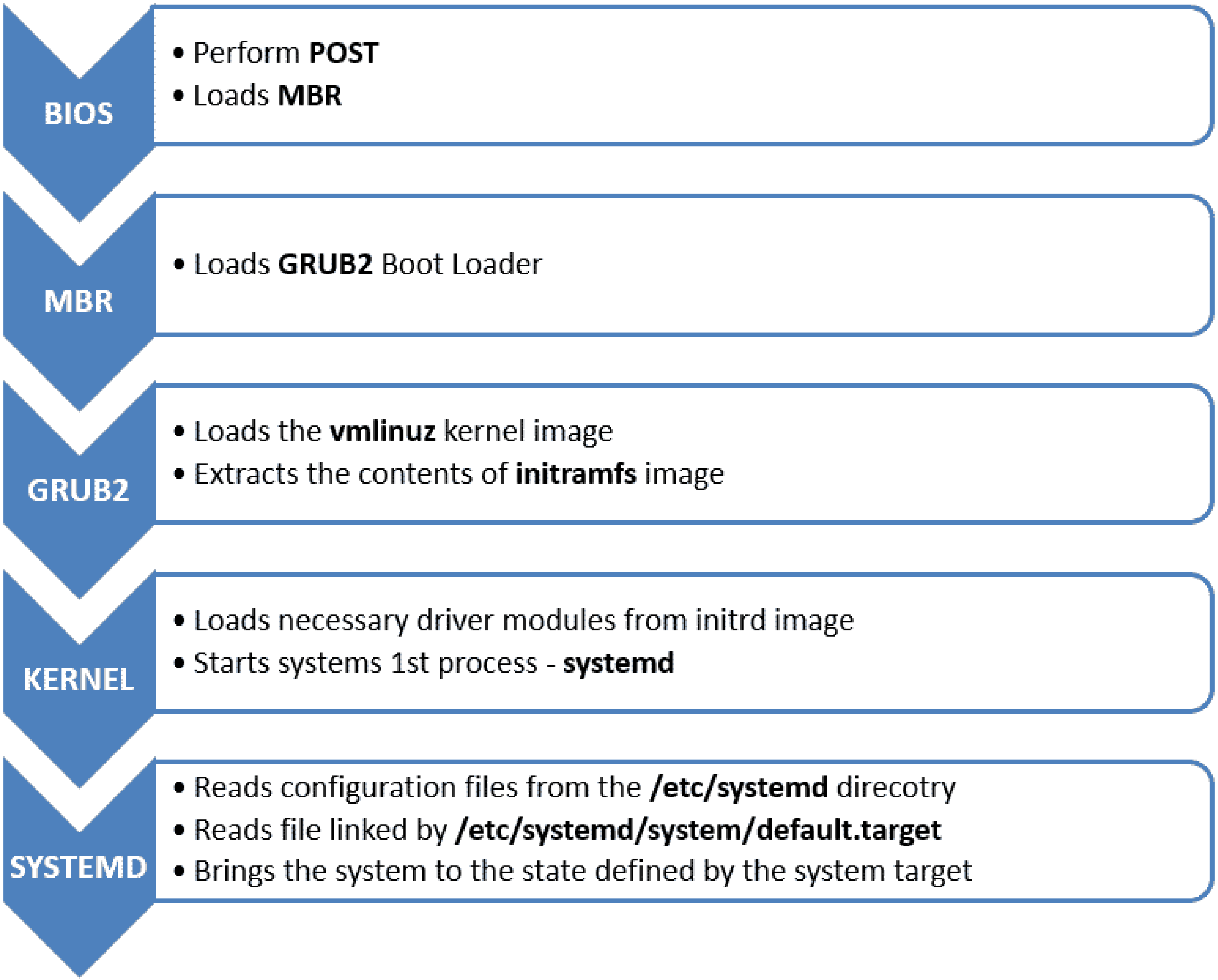
4th Grade -Fall Semester 2021-2022

Instructor: Alaa Ghazi

Lecture 4
Booting and Shutting Down

LET'S SHUT DOWN ALL HIS BODY FUNCTIONS THEN START THEM UP AGAIN...

Linux Booting Stages Diagram



BIOS

- Perform **POST**
- Loads **MBR**

MBR

- Loads **GRUB2** Boot Loader

GRUB2

- Loads the **vmlinuz** kernel image
- Extracts the contents of **initramfs** image

KERNEL

- Loads necessary driver modules from initrd image
- Starts systems 1st process - **systemd**

SYSTEMD

- Reads configuration files from the **/etc/systemd** direcotry
- Reads file linked by **/etc/systemd/system/default.target**
- Brings the system to the state defined by the system target

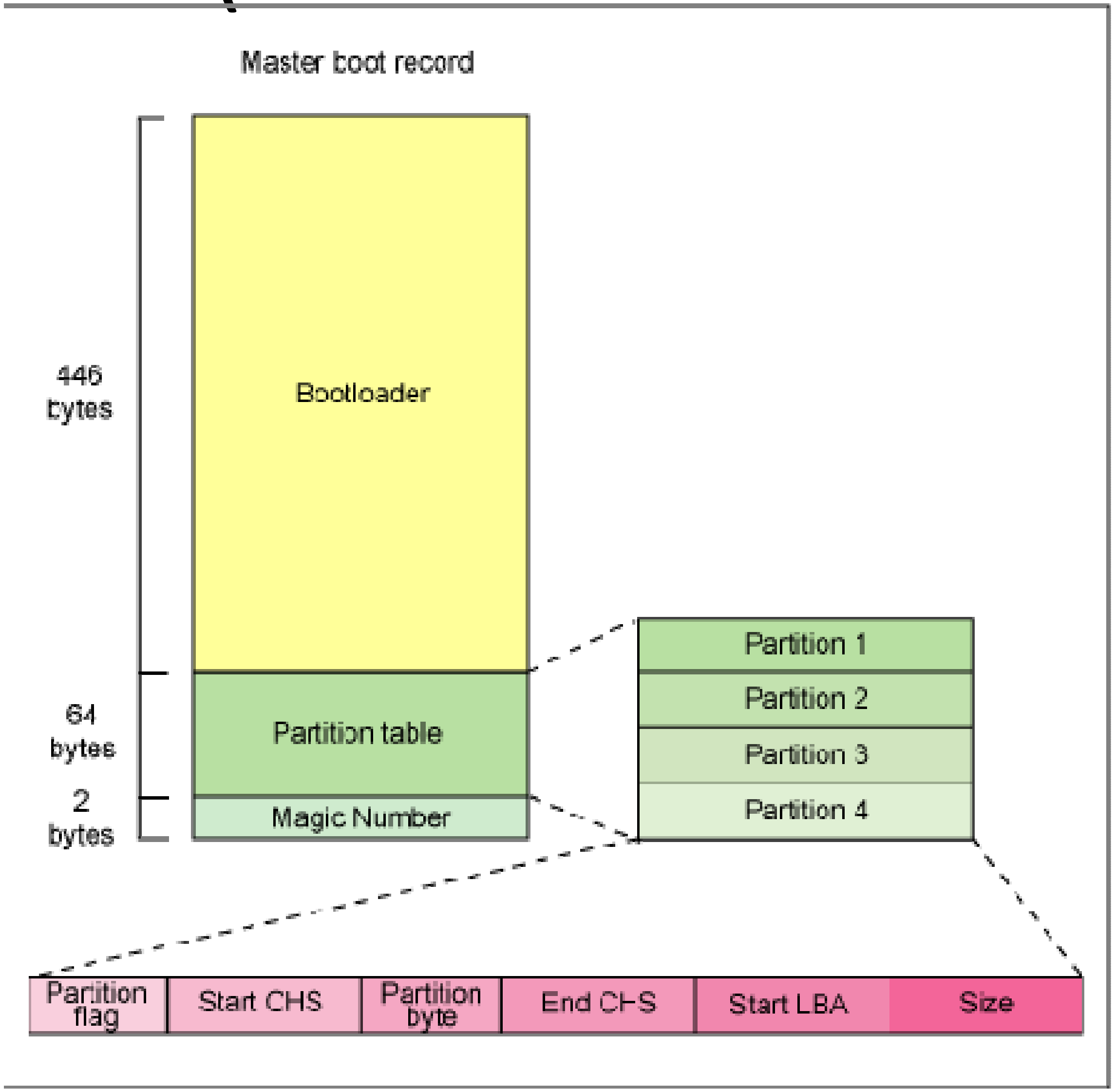
Booting Stage 1: BIOS

- The BIOS (Basic Input/Output System), performs the POST (power on self test) to detect, test and initialize system hardware components.
- Loads the MBR (Master boot record).
- If some hardware broke,System will notice.
- Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.
- So, in simple terms BIOS loads and executes the MBR boot loader.

Booting Stage 2:
MBR

- Master Boot Record (MBR) is the first sector of disk.
- It’s at first 512 bytes of the boot drive that is read into memory by the BIOS. Like the figure in next slide the MBR is divided into:
  - 446 bytes is data of boot loader,
  - 64 bytes contain the partition table for the disk, and
  - the last 2 bytes are the “Magic Number” which is used for error detection in sector.
- MBR discovers the bootable device and loads the GRUB2 boot loader into memory and transfers control over to it.
- So, in simple terms MBR loads and executes the GRUB boot loader.

MBR (Master Boot Record)



Booting Stage 3: GRUB Bootloader

- **GRUB** stands for **GRand Unified Bootloader** and it is the default boot loader program in most modern Linux distributions with Intel processors.
- GRUB v2 replaces the older version called the **Legacy GRUB**.
- **GRUB Main Features**:

1. By chain-loading it can load, almost all operating system's available.
2. GRUB has the knowledge of the file system (the older Linux loader LILO didn't understand file system).
3. Grub does not require the exact physical location of the operating system kernel in the hard disk. It just requires the hard disk number, and the partition number, along with the file name of the kernel.
4. Through GRUB command line, the booting options, like kernel parameter's and its location can be modified.

GRUB Mechanism

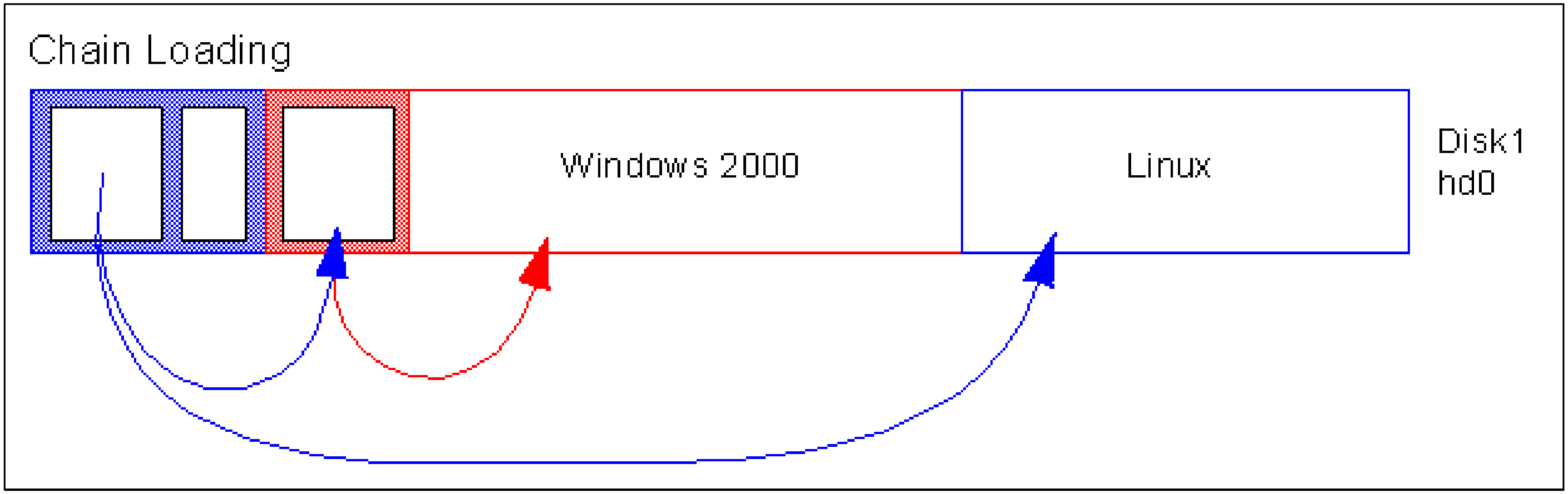
- GRUB displays a splash screen, waits for few seconds, if user don't enter anything, and If no other operating system is detected GRUB will boot straight into the default operating system and no menu will be displayed.
- If another operating system is detected the GRUB menu will display and user can select the OS to load.
- It searches and loads the compressed kernel image file located in

/boot /vmlinuz

- GRUB mounts the initial RAM disk (initrd) is an initial root file system that is mounted before the real root file system.
- So, in simple terms GRUB just loads and executes Kernel and initrd images.

GRUB Chain-Loading

- **Chain-Loading** is the possibility to load another boot loader to boot unsupported operating systems
- If you want to boot an unsupported operating system (e.g. Windows 2000), *chain-load* a boot loader for the operating system. Normally, this boot loader is embedded in the *boot sector* of the partition on which the operating system is installed.



GRUB Menu Settings

- Changes made in the GRUB menu-settings file

/etc/default/grub

will not take effect and overwrites GRUB configuration file

/boot/grub2/grub.cfg

until the below command is executed.

update-grub

- Below are some example settings:

GRUB_DEFAULT **Sets the default menu entry**

GRUB_HIDDEN_TIMEOUT **Wait this many seconds for the user to press a key.**

GRUB_HIDDEN_TIMEOUT_QUIET **Determines whether a countdown timer is displayed on a blank screen.**

GRUB_TIMEOUT **Sets the time period in seconds for the menu to be displayed before automatically booting.**

GRUB Command-Line Options

find - Finds files on all mountable partitions

root - Specifies the root device (a partition)

ls - List hard disks, devices or files

help - Gets interactive help for a command

boot - Boots the system from the specified kernel image

reboot - Reboots the system

GRUB Device Naming Convention

- Grub device naming convention is totally different from the operating system partition naming convention.
- GRUB uses a 0-based naming scheme for disk indexes but uses a 1-based naming scheme for partition indexes and uses the following format:

(hd*X, part-typeY*)

- Examples are shown below:

Device Name	**Description**
(hd0, msdos1)	Specifies the first DOS partition on the first disk.
(hd0, gpt2)	Specifies the second GPT partition on the first disk.
fd0	Floppy disk

Booting Stage 4: Loading The Kernel

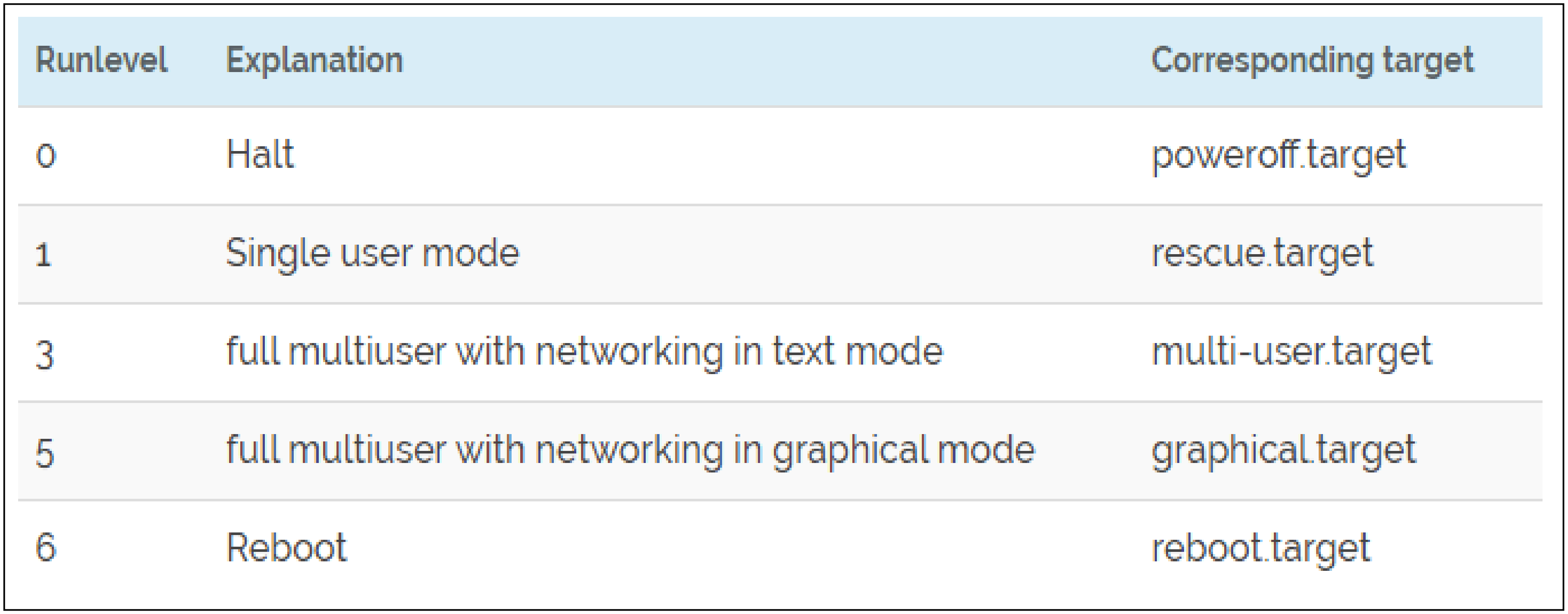
- Kernel is itself a program, and the first bootstrapping task is to get this program into memory so that it can be executed
- Linux system kernel - /boot/vmlinuz
- Kernel mounts filesystem in Two-stages:
  - initrd (init RAM disk): A transient root filesystem in RAM before a real root filesystem is available
  - The real root filesystem
- Kernel executes init program located in /sbin/init
- Kernel loads device drivers
- Kernel threads creation

Booting Stage 5: Systemd

- Systemd is the grandfather of all processes on a system.
- Systemd reads the file linked by /etc/systemd/system/default.target to determine the default system target (equivalent to run level).
- The system target file defines the services that systemd starts.
- Systemd performing system initialization tasks such as:
  1. Setting the host name
  2. Initializing the network
  3. Initializing Linux based on its configuration
  4. Printing a welcome banner
  5. Cleaning up directories in /var
  6. Starting swapping

Linux Run Levels

- **Runlevels**: are modes that Linux runs in. Each of these modes, or runlevels, has its list of processes and services that are either turned on or off.
- From the time Linux boots up, it's always in some runlevel.
- Below is comparison of different runlevels and related targets in Linux :



Runlevel	Explanation	Corresponding target
0	Halt	poweroff.target
1	Single user mode	rescue.target
3	full multiuser with networking in text mode	multi-user.target
5	full multiuser with networking in graphical mode	graphical.target
6	Reboot	reboot.target

Run Level Commands

- To change Linux runlevel systemd is used.
- Runlevels have been replaced with targets, although runlevel command, can still be used.
- The command to show current runlevel

runlevel

- To change the next boot run level used the command

sudo init *n*

Where *n* is the run level as per above table

Single-User Mode

- ***Single user mode***, (*maintenance mode):* is a mode of operation of running Linux that provides as few services as possible and only minimal functionality, with root shell is started on the console.
- It is useful for booting an operating system that has been damaged and is not capable of normal operation and for performing some diagnostic and repair tasks.
- Single User mode usage:

1. Check/repair the disk if there are disk problems
2. Reset root password if the password was forgotten.

- Demonstration of how to login to single-user mode and above tasks will be included in File System LAB.

Rebooting and Shutting Down

- When the system is initiated for Shutdown or Reboot, It will be notified to all logged-in users and processes. Also, it wont allow any new logins if the time argument is used.
- Improper shutdown can result problem in the system.
- To modify or make significant system changes, you should reboot just to make sure that the system comes up successfully.
- In modern Linux, the **systemd** utility is used to manage all services and processes of the system, so the legacy commands will use *systemctl command* to halt, poweroff, shutdown, or reboot respectively.

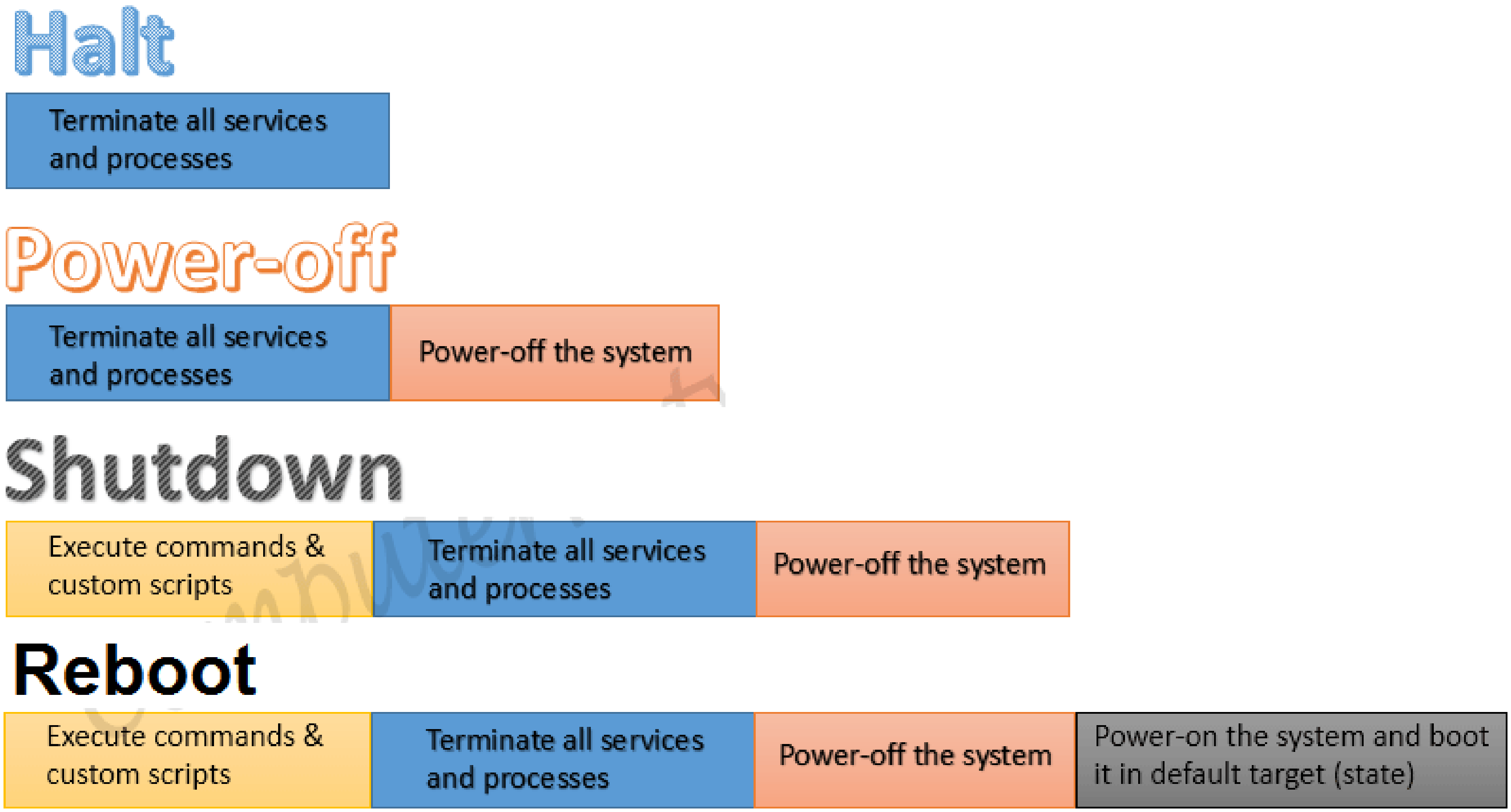
System Administrator Procedures for Shutting Down a Server

1. Get a console (keyboard and monitor) access for troubleshooting
2. Get a down time approval from management
3. Send notifications to all users.
4. Take backup.
5. Communicate DB and Application administrators to bring down their services with validation.
6. Reboot or shutdown the system
7. Ask the DB and Application administrators to bring up their services.
8. Monitor for some time to make sure everything is up again.

Essential Shutting Down Commands

- **Halt**: Terminate all processes and services and shutdown CPU but leaves the system in a powered-on state.
- **Power-off**: Turn off all services and process then send an ACPI command to the motherboard to terminate the power supply of the system.
- **Shutdown:** Execute shutdown scripts and then turn off all services and process then send an ACPI command to the motherboard to terminate the power supply of the system.
- **Reboot**: First shutdown the system then sends a reboot ACPI command to the motherboard to start the system again.

Essential Shutting Down Commands



Halt

Terminate all services and processes

Power-off

Terminate all services and processes | Power-off the system

Shutdown

Execute commands & custom scripts | Terminate all services and processes | Power-off the system

Reboot

Execute commands & custom scripts | Terminate all services and processes | Power-off the system | Power-on the system and boot it in default target (state)

Shutting Down Commands

- To halt the system, use any of below commands:

```
sudo shutdown -H now
sudo halt
```

- To power-off the system, use any of below commands:

```
sudo shutdown -P now
sudo poweroff
```

- To shut down the system, use any of below commands:

```
sudo shutdown
```

Shutting Down Commands

- To reboot the system, use any one of the following commands.

```
sudo shutdown -r
sudo reboot
```

- You can specify waiting time like example below:

system is going for halt after 15 minutes

```
sudo shutdown –H +15
```

- You can specify exact time like example below:

system is going for reboot at 09:35am

```
sudo shutdown –r 09:35
```

Advanced Power Management

- **Sleep (suspend or standby)**: In this mode, Linux saves the current state of the system into the RAM and cuts the power supply of all devices except the RAM, and when the system is started/waked-up again, the Linux restores the saved state from the RAM instead of booting the system, but if the power supply is turned off, the saved state is lost. The command is:

```
sudo systemctl suspend
```

- **Hibernate**: In this mode, Linux saves the current state of the system into the hard disk and shuts down the power supply of the system. Since the current state is saved in the disk, the saved stated does not depend on the battery or external power supply, so it is more reliable then the sleep mode. The command is:

```
sudo systemctl hibernate
```

LAB 04
Booting and Shut Down

LAB 04 TEST1: Default Configuration

- Reboot and see the behavior of the GRUB bootloader and
- Discover the default configuration using the command

sudo nano /etc/default/grub

- Provide screen shoot and your comments

LAB 04 TEST2: Changing Default Configuration

- Edit the default configuration using the command

sudo nano /etc/default/grub

- Change below parameters to indicted values

GRUB_HIDDEN_TIMEOUT=100

GRUB_HIDDEN_TIMEOUT_QUIET=false

- Run the commands

sudo update-grub

reboot

- Reboot and see the behavior of the GRUB bootloader
- Provide screen shoot and your comments

LAB 04 TEST3: Logging to GRUB Command Line

- Reboot and once you see the count down press and hold the right SHIFT
- Provide screen shoot and your comments
- Then press ‘c’ to enter GRUB command line
- Type the commands

ls

- Provide screen shoot and your comments

reboot

LAB 04 TEST4: MultiBoot

1. Install any other Linux OS on different virtual machine like:

(Xubuntu, Lubuntu, Fedora, CentOS, etc)

2. From virtualBox remove attachment to that hard disk from its original virtual machine through the menu:

- Machine > Settings > Storage > Controller SATA
- Right click on any virtual disk and select ‘Remove Attachment’ and Click OK

3. Attach this hard disk to Ubuntu virtual machine through the menu:

- Machine > Settings > Storage > Controller SATA
- Right click select ‘Hard Disk’
- Click Add and browse to the folder of the other virtual machine and select the virtual hard disk
- Click Choose, Click OK

LAB 04_ TEST4: MultiBoot

4. Start the Ubuntu machine and run the command

sudo update-grub

Provide screen shoot and your comments

5. Reboot and see the behavior of the GRUB boot-loader, provide screen shoot and your comments
6. Select the other operating system, and provide screen shoot and your comments.
7. Reboot and select Ubuntu.

LAB 04_ TEST5: Sleep and Hibernate

1. While Ubuntu is runing normally, put the command

sudo systemctl suspend

After few minutes provide screen shoot of VirtualBox main screen with your comments about the power state of the virtual machine and how could you restore the running state

2. While Ubuntu is runing normally, put the command

sudo systemctl hibernate

After few minutes provide screen shoot of VirtualBox main screen with your comments about the power state of the virtual machine and how could you restore the running state.