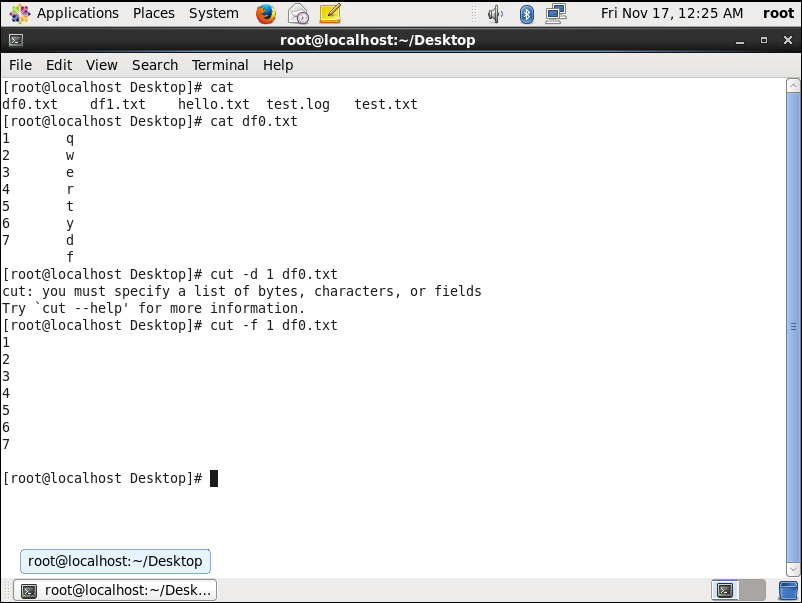
Buổi 4: Kỹ thuật dùng lệnh (Phần còn lại)

6. cut và tr:

Trên Linux, trong một file, dữ liệu sẽ được phân chia theo các dòng (line) và trường (field). Nếu grep là lệnh giúp chúng ta in ra các dòng chứa “pattern” mà ta chỉ định, thì cut sẽ giúp ta in ra các trường mà ta mong muốn với option –f.

Ví dụ: cut –f 1 df0.txt



Trên Linux, chức năng cơ bản đầu tiên của công cụ tr là giúp ta thay thế một hoặc nhiều ký tự thành một hoặc nhiều ký tự khác.

Ví dụ: tr h H < hello.txt

Hoặc là tr [a-z] [A-Z] < hello.txt ( [-] là cú pháp khai báo một loạt lần lượt các ký tự cần thay thế chứ không phải là một chuỗi).



Ngoài ra tr còn có một cách dùng nữa đó là xóa các dấu tab phân cách các trường bằng một ký tự mà ta chỉ định. Chẳng hạn:

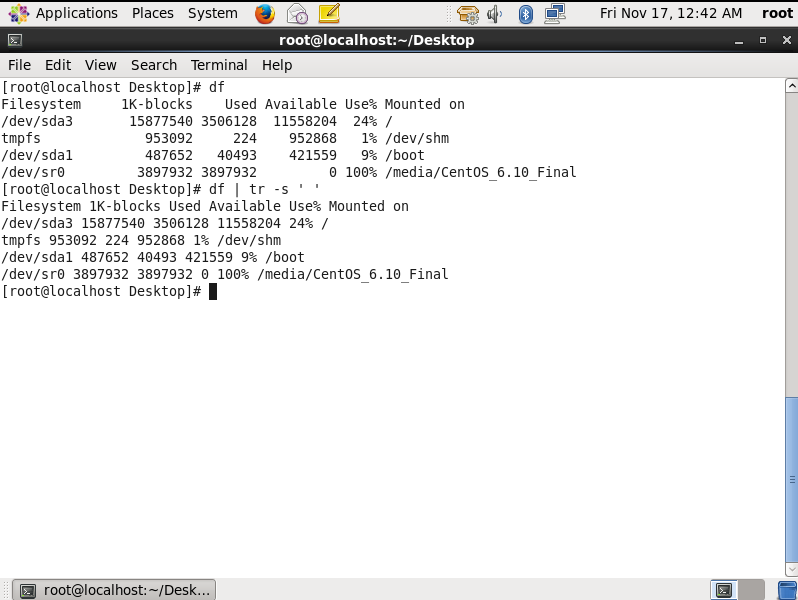
df (đây là công cụ giúp ta quản lý bộ nhớ thiết bị dưới dạng block)

Với hiển thị về mặt kỹ thuật rất khó nhìn như mặc định, ta sẽ thay các dấu tab không đều nhau ở trên bằng dấu cách. Bằng công cụ tr, ta làm như sau:

df | tr –s ‘ ‘

Với cách làm này thì sẽ không còn các trường nữa vì phân cách đã bị xóa bỏ và thay bằng dấu cách.

Ta sẽ có hiển thị ở màn hình như sau:



7. cat và more, tac và less:

Đây là nhóm lệnh in ra màn hình nội dung của file, tuy nhiên có vài điểm cần lưu ý ở đây:

cat: in ra toàn bộ nội dung từ trên xuống dưới. Với lệnh này thì ta dành cho máy đọc, phục vụ cho việc xử lý nội dung file, hay điều hướng kết quả in ra màn hình đến một file khác.

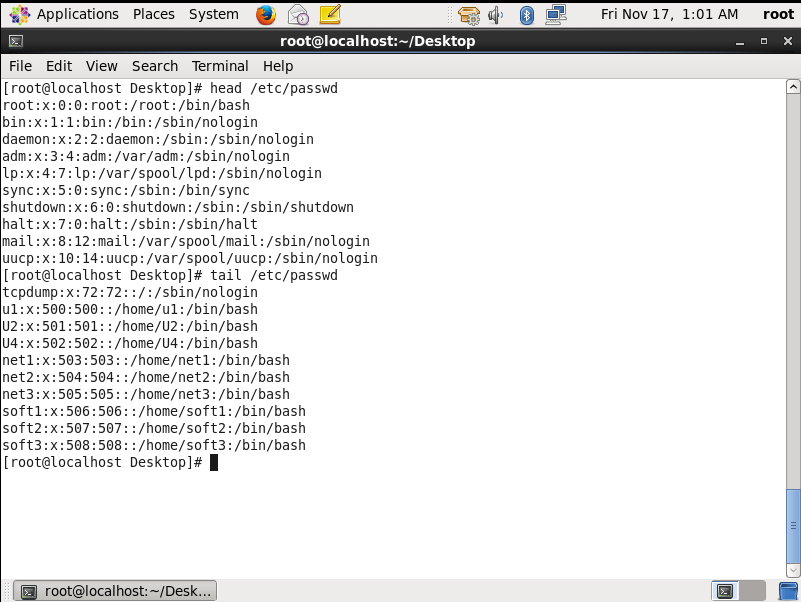
more: in ra nội dung từ trên xuống dưới nhưng không phải toàn bộ, mà là đến cuối màn hình thì không in nữa. Với lệnh này người dùng có thể xem từ từ nội dung in ra bằng nút enter, hoặc xem đến cuối luôn bằng nút cách. Lệnh này dùng cho người đọc.

Hai lệnh tac và less chức năng tương ứng lần lượt với cat và more, có điều hai lệnh này in ra nội dung từ dưới lên trên.

8. head và tail:

head: mặc định khi không có option in ra nội dung 10 dòng đầu tiên của 1 file

tail: mặc định khi không có option in ra nội dung 10 dòng cuối của 1 file



Lệnh tail còn 1 cách sử dụng nữa, đó là option –f dùng để đọc log.

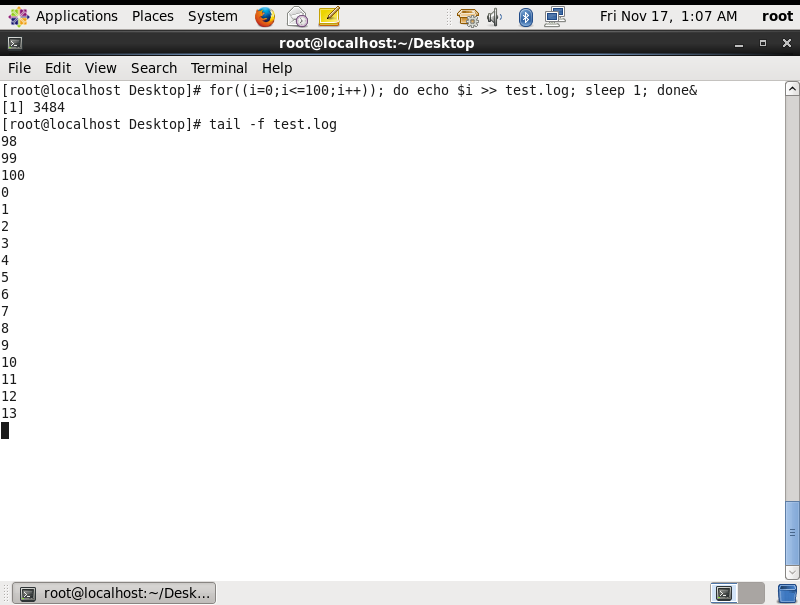
Ví dụ: Ta sẽ tạo một vòng lặp để sử dụng tail –f đọc kết quả vòng lặp ấy

For((i=0;i<100;i++)); do echo$i >> test.log; sleep 1; done&

Lưu ý:

sleep 1 nghĩa là cho chương trình tạm nghỉ 1s khi xong 1 vòng lặp, tránh việc chương trình chạy quá nhanh thì ta sẽ không biết được tail –f hoạt động ra sao.

done là cú pháp kết thúc vòng lặp, cú pháp này và dấu & không liên quan gì đến nhau cả. Dấu & là 1 ký tự đặc biệt đặt ở cuối dòng nhằm chạy chương trình này dưới dạng background.



9. Công cụ sed:

Chúng ta đã có công cụ thay đổi ký tự là tr, giờ đây với sed ta có thể thay thế hoặc xóa đi một chuỗi mà ta chỉ định. Cú pháp của câu lệnh như sau:

sed ‘ ‘ <input file>, trong đó bên trong thông điệp ‘ ‘ có nội dung như sau:

‘add cmd/str/rep\_str/flag(g)’

Trong đó:

add: địa chỉ (number line) chỉ định cho việc xóa hoặc thay thế

cmd: có hai lệnh là d(delete) và s(replace)

str: chuỗi chỉ định

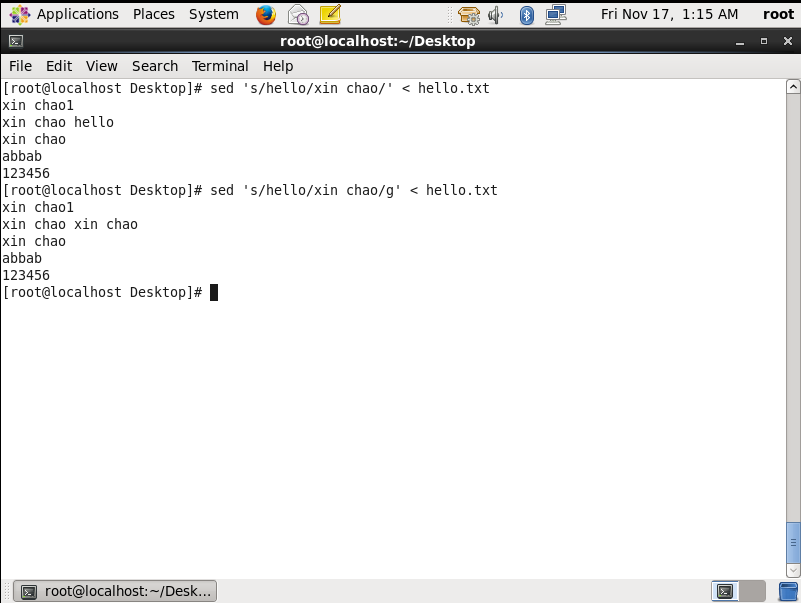
rep\_str: chuỗi thay thế

flag: gắn cờ g ( Lưu ý nếu không gắn cờ thì việc thay thế hay xóa đi sẽ không được thực hiện triệt để)

Ví dụ:

sed ‘s/hello/xin chao/’ < hello.txt

sed ‘s/hello/xin chao/g’ < hello.txt  
Cùng xem sự khác biệt khi không và có gắn cờ



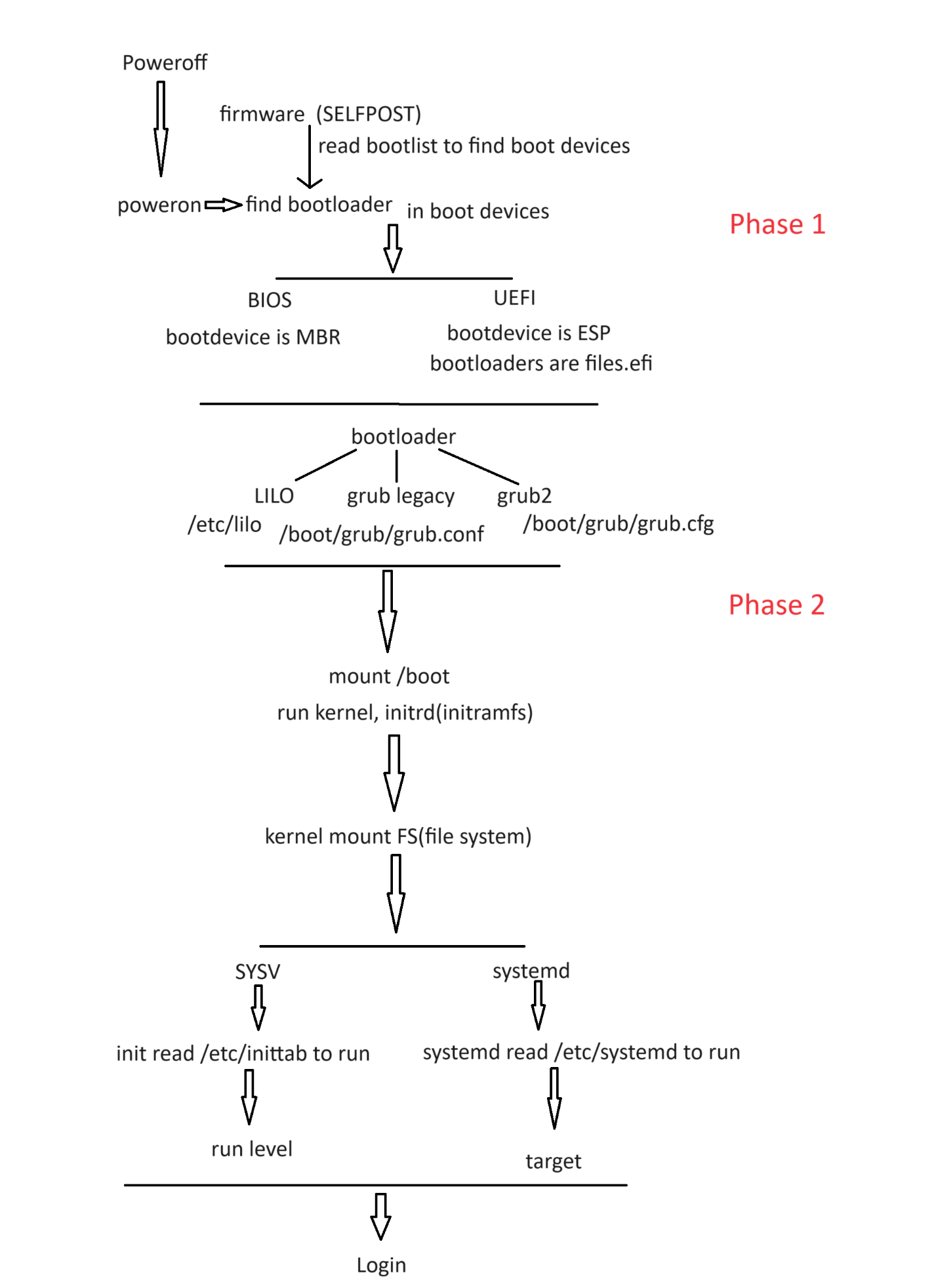
Nội dung buổi 4 kết thúc ở đây

Buổi 5: The startup

Quá trình khởi động của một hệ điều hành và máy chủ của nó là một quá trình tương đối phức tạp và phải trải qua nhiều bước. Ở buổi này ta cần nắm rõ khái niệm về firmware, trình quản lý khởi động (bootloader) và chương trình khởi chạy hệ điều hành (init) dưới dạng run level hoặc target.

Bài thi giữa kỳ:

* Có 3 lỗi thường xảy ra trong quá trình khởi động là những lỗi nào?
* Sửa các lỗi liên quan đến việc mất, lỗi file cấu hình của bootloader, hoặc bootloader bị hỏng.
* Hiểu rõ các phase trong quá trình khởi động của hệ điều hành Linux



1. Firmware startup:

+ **The BIOS startup:**

Giao diện menu đơn giản (menu BIOS): Cài đặt để kiểm soát cách hệ thống tìm thấy phần cứng và xác định BIOS nên sử dụng thiết bị nào để khởi động hệ điều hành (Trong mục boot của menu)

Hạn chế (limitations): BIOS chỉ có thể đọc dữ liệu có giá trị của một sector từ ổ cứng vào bộ nhớ để chạy (sector này không đủ dung lượng để tải toàn bộ hệ điều hành.). Bởi vậy nên hầu hết các hệ điều hành (bao gồm Linux và Microsoft Windows) đều chia phần khởi động quá trình thành hai phases.

Phase1: BIOS chạy chương trình bootloader

Bootloader thường nằm trong sector 0 của ổ đĩa, hay còn goi là MBR với kích thước 512 bytes.

Phase2: Bootloader trỏ đến vị trí của file kernel lưu trữ ở /boot trên hệ thống.

Sau đó chương trình init sẽ chạy hệ thống theo run level mặc định được cấu hình trong file /etc/inittab. Lúc này hệ điều hành đã khởi chạy và ta có thể đăng nhập vào hệ thống.

+ **The UEFI startup:**

* Intel đã tạo ra Extensible Firmware Interface (EFI) vào năm 1998 để giải quyết một số hạn chế của BIOS.
* UEFI chỉ định một phân vùng đĩa đặc biệt, được gọi là Phân vùng hệ thống EFI (ESP) để lưu trữ các chương trình bootloader.
* Bất kỳ kích thước nào của chương trình bootloader
* Nhiều chương trình bootloader cho nhiều hệ điều hành
* Phân vùng ESP có định dạng file là FAT32, phân vùng này lưu trữ các bootloader dưới dạng các file .efi.

2. Linux bootloaders:

+ **LILO:**

File config: /etc/lilo.conf

* Thay đổi file cấu hình hoặc cài đặt kernel mới sẽ phải cài lại bootsector bằng lệnh lilo
* Không chạy trên nền tảng UEFI

+ **GRUB Legacy** (Chúng ta sẽ làm bài bằng bootloader này)

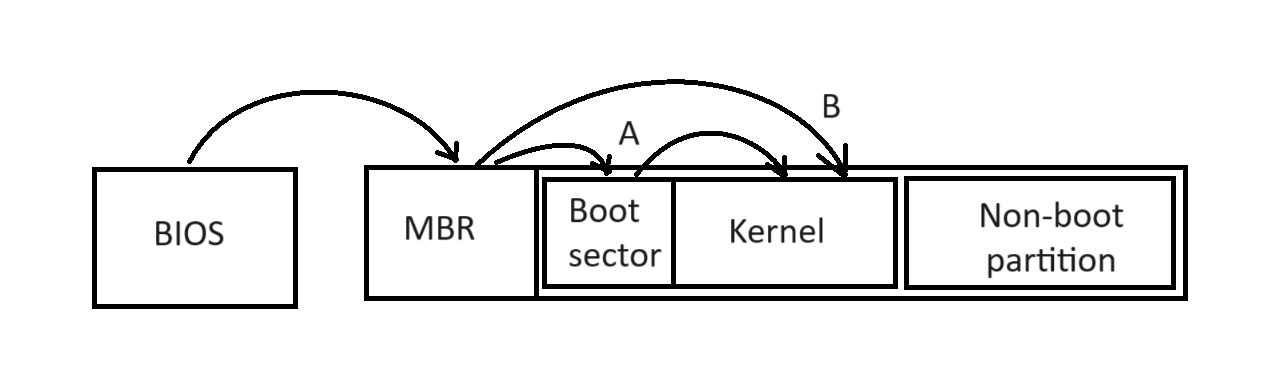
* File config: /boot/grub/grub.conf (bản CentOS 6 có menu.lst là 1 liên kết tượng trưng, lý do là vì một vài phiên bản cũ hơn của grub dùng cái tên này cho file cấu hình)
* Thay đổi file cấu hình không cần phải cài lại bootsector
* GRUB chạy cả trên hệ thống nền tảng BIOS và UEFI

**+ GRUB 2**

* Hỗ trợ các tính năng cao cấp hơn, linh hoạt và mạnh mẽ hơn phiên bản Legacy
* File cấu hình: /boot/grub/grub.cfg

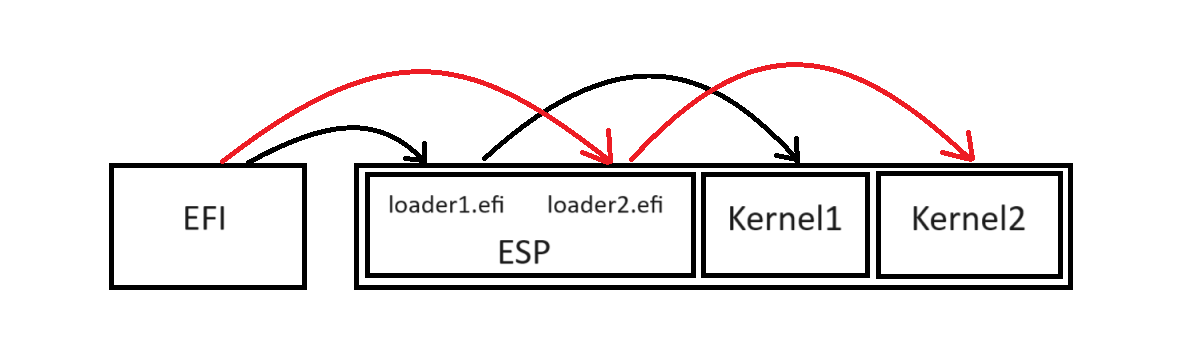
Trên BIOS và UEFI sẽ có cách tìm bootloader khác nhau, ta có thể xem ở minh họa bên dưới

**Với BIOS:**



* A: BIOS tìm thấy MBR, tuy nhiên ta lại muốn chạy hệ điều hành khác không phải do bootloader ở MBR quản lý, do đó bootloader này sẽ trao quyền lại cho bootloader ở bootsector khác quản lý hệ điều hành ta muốn chạy
* B: Ta chạy hệ điều hành mà bootloader ở MBR quản lý

**Với UEFI:**



* Quá trình này sẽ đơn giản hơn vì các bootloaders đều ở dạng file.efi trong phân vùng ESP. Ta chạy hệ điều hành nào thì gọi trỏ đến bootloader đó.
* EFI (/boot/efi): Điều khác biệt ở đây là với nền tảng này UEFI sẽ không tìm MBR như BIOS, mà nó sẽ tìm đến phân vùng ESP rồi chọn bootloader.

!Lưu ý: Lỗi đầu tiền trong ba lỗi đó chính là không tìm thấy bootloader. Lỗi này thuộc về lỗi firmware của nhà phát hành, hoặc đơn giản thứ tự ưu tiên trong mục boot của menu BIOS/UEFI chưa đúng. Các bạn lưu ý phần này.

3. Cấu hình configuration file của bootloader

**+ Modifying GRUB legacy configuration file**

Global GRUB Options

* /dev/sda1 ⬄ hd0,0 – nghĩa là đây là phân vùng đầu tiên của ổ đĩa thứ nhất
* GRUB không phân biệt được PATA, SATA hay SCSI drives
* **default=** option cho GRUB biết hệ điều hành nào để boot
* **Timeout=**<second>
* Background graphic: **splashimage=** tệp đồ họa được hiển thị làm background cho quá trình khởi động
* Title: Tên hệ điều hành

Per-Image Options

* GRUB root: **root**
* Kernel specification: **kernel**
* Initial RAM disk: **initrd**
* Chain loading: **chainloader**

**!Lưu ý:** Trong bài giữa kỳ phần sửa lỗi mất, lỗi file cấu hình bootloader, các bạn lưu ý phần Per-Image Options chứa các công cụ cơ bạn để các bạn thao tác cấu hình thủ công cho booloader GRUB quản lý quá trình khởi động.

Ví dụ: Nội dung file cấu hình có dạng sau (các bạn xem nội dung ở file /boot/grub/grub.conf)

default=0

timeout=5

splashimage=(hd0,0)/grub/splash.xpm.gz

hiddenmenu

title Red Hat Enterprise Linux Server (2.6.18-53.el5)

root (hd0,0)

kernel /vmlinuz-2.6.18-53.el5 roroot=/dev/VolGroup00/LogVol00 rhgbquiet

initrd /initrd-2.6.18-53.el5.img

title DOS

rootnoverify(hd0,1)

chainloader+1

+ **Modifying GRUB 2 configuration file**

Ở phiên bản này GRUB đánh số thứ tự của phân vùng (partition từ số 1): hd0,1- Phân vùng thứ nhất của ổ thứ nhất

**Rootnoverify** và **kernel** không còn được sử dụng

**/etc/grub.d/**: individual configuration files

**/etc/default/grub** configuration file: Global commands

Ví dụ: Đây là nội dung thường thấy trong grub.cfg

**menuentry**"CentOS Linux" {

set root=(hd1,1)

linux /boot/vmlinuzinitrd /initrd

}

**menuentry**"Windows" {

set root=(hd0,1)

}

4. Cài đặt GRUB

+ **Khi nào ta cài đặt?**

* Resizing hoặc chuyển GRUB root partition(phân vùng gốc) (Lưu ý không nhầm phân vùng gốc **/boot** với thư mục gốc **/**)
* Di chuyển toàn bộ cài đặt của bạn sang một ổ cứng mới
* Cài đặt lại hệ điều hành

Trong trường hợp mất, lỗi bootloader thì chúng ta sử dụng “Rescue installed system” trong đĩa CD. Nếu các bạn dùng máy ảo Vmware Workstation thì phải nhớ cài chính xác đĩa CD chứa nội dung về hệ điều hành nào chứ không để “auto detect”

+ **Cài đặt GRUB legacy**

* **grub-install ‘(hd0,0)’** (Phân vùng và ổ đĩa này là ví dụ, các bạn khi cài đặt nên xán định chính xác ổ đĩa và phân vùng mình muốn cài)
* **grub-install /dev/sda**
* **grub-install /dev/fd0**
* ...

!Lưu ý: Đây là cú pháp để cài lại bootloader bằng Rescue trong trường hợp lỗi bootloader bị mất, hỏng. Các bạn nắm rõ phần này để làm bài thi.

+ **Cài đặt GRUB 2**

Với GRUB 2 thì bạn không cần phải cài đặt lại nó mà chỉ cần rebuild lại file cài đặt chính là: grub-mkconfig > /boot/grub/grub.cfg

!Lưu ý: Thầy chỉ kiểm tra trên nền tảng sysv của CentOS 6 (hoặc các phiên bản enter prise Linux 6 đổ về trước) với bootloader là GRUB legacy. Phần về GRUB 2 và nền tảng systemd khuyến khích các bạn đọc thêm.

5. Boot Process

+ **System V**

Khi boot thành công, hệ thống sẽ sử dụng chương trình init để chạy các chương trình hiển thị tương ứng với các runlevel

|  |  |
| --- | --- |
| Run level | Chức năng |
| 0 | Tắt hệ thống |
| 1 | Single-user mode |
| 2 | Multiuser, but no networking support |
| 3 | Multiuser text login (giao diện CLI) |
| 4 | Undefined |
| 5 | Multiuser GUI login (giao diện đồ họa |
| 6 | Reboot |

Để xem runlevel đang chạy: runlevel

Để đổi run level ta có 2 cách:

1. Đổi tạm thời: init <number\_runlevel>

2. Đổi runlevel mặc định: sửa file cấu hình /etc/inittab

vi /etc/inittab

Lúc này nội dung file cấu hình có dạng

id:runlevels:action:process

Trong đó:

id - Trường này thường đại diện cho một định danh duy nhất cho một dòng cụ thể trong tệp

runlevels – [0-6] các runlevel là các fiel rc.0-rc.6 trong thư mục /etc/rc.d

action - Trường này chỉ ra hành động cụ thể nào sẽ được thực hiện khi hệ thống chuyển đến runlevel được chỉ định. Các hành động phổ biến: initdefault, respawn, wait.

process - Trường này xác định quy trình cụ thể (hoặc lệnh) mà hệ thống sẽ thực hiện khi có sự kiện thay đổi runlevel tương ứng với dòng cấu hình này.

Ví dụ: Cấu hình runlevel mặc định là 5

id:5:initdefault:

Tương tự cho các runlevel khác.

+ **Systemd**

Thay vì sử dụng shell scripts và runlevel, systemd sử dụng targets và units để kiểm soát các ứng dụng chạy bất kỳ lúc nào trên hệ thống.

**Units** là các đơn vị cơ bản của hệ thống, đại diện cho các thành phần khác nhau như dịch vụ, thiết bị, mount points, timer, và nhiều khái niệm khác. Các Units cung cấp cách để quản lý và theo dõi các phần của hệ thống Linux.

Để xem các units: systemctl list-units

**Target** Trong hệ thống systemd, một target là một nhóm các đơn vị (Units) có thể được sử dụng để đặt mục tiêu cho trạng thái cụ thể của hệ thống. Target đóng vai trò như là một cách để định rõ một môi trường hoặc chế độ làm việc cụ thể của hệ thống. Mỗi target tương ứng với một chế độ hoạt động nhất định. Giống như runlevel của sysv nhưng đa dạng và linh hoạt hơn. Ví dụ **multi-user.target** tương đương với **runlevel 3**, **graphical.target** tương đương với **runlevel 5**, và **rescue.target** tương đương với **single-user mode**.

Các targets được đặt trong thư mục /etc/systemd/system hoặc /lib/systemd/system với định dạng .target.

**systemctl** Để quản lý linh hoạt các target, ta có công cụ systemctl. Các bạn hãy dùng option --help để biết cách sử dụng cộng cụ này.

Ví dụ: Cài multi-user.target làm mặc định

systemctl set-default multi-user.target

grub legacy systemV

grub 2 system D