

Khởi động hệ thống

Nội dung

1. Tổng quan quá trình khởi động
2. Quá trình tải nhân hệ điều hành
3. Quá trình khởi động dịch vụ hệ thống.
Các mức thực hiện
4. Khởi tạo môi trường làm việc

1. Quá trình khởi động hệ thống máy tính

Mục tiêu của quá trình khởi động

- Khởi động các thành phần phần cứng

- Kiểm tra trạng thái thiết bị

- Khởi động các phần mềm cho NSD

Cụ thể khi khởi động PC

- Khởi động các thành phần phần cứng

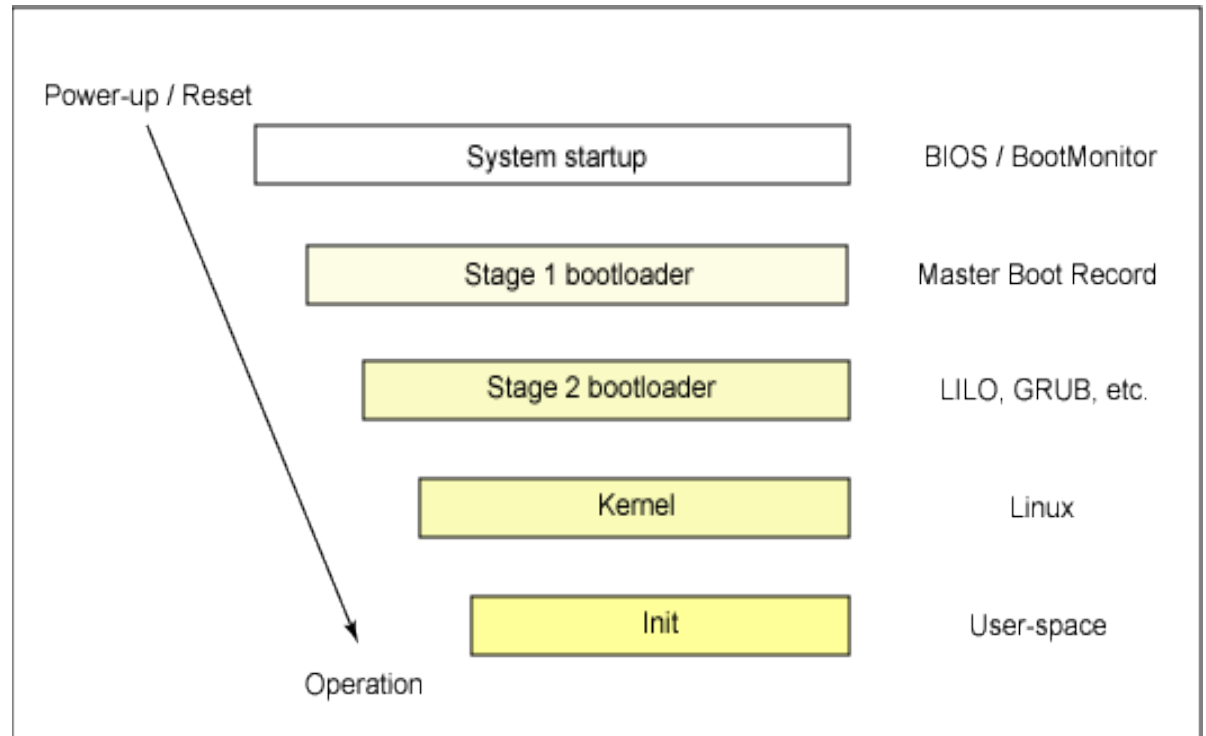
- Khởi động MBR

- Thực hiện chương trình quản lý khởi động (menu hệ điều hành)

- Khởi động nhân hệ điều hành

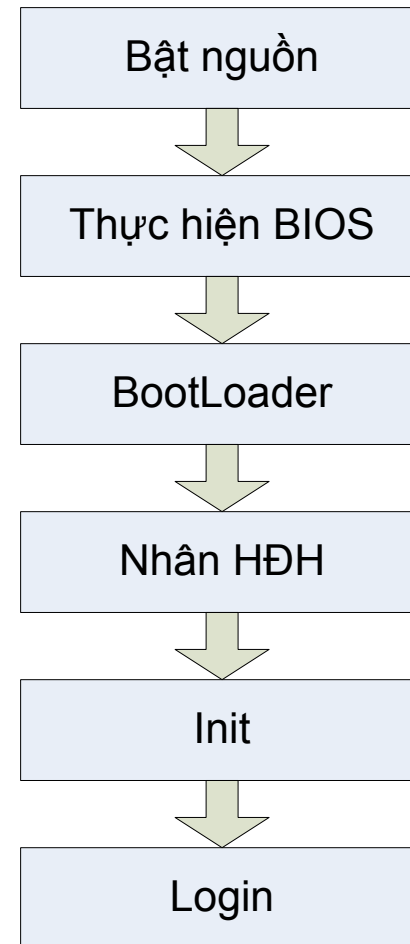
- Khởi động các chương trình phục vụ NSD

- Phụ thuộc yêu cầu các giai đoạn khởi động này có thể sát nhập



Quá trình khởi động Linux

- Bật nguồn điện
 - Hệ thống tự kiểm tra và thực hiện cấu hình phần cứng
- Chương trình BIOS được thực hiện
 - Cấu hình các thiết bị ngoại vi
 - Truy cập vào các thiết bị lưu trữ chính
- Chương trình khởi động được thực hiện
 - Tải nhân HĐH
 - Khởi động các dịch vụ của HĐH
- HĐH thực hiện tiến trình init
 - Khởi động các tiến trình và môi trường làm việc của hệ thống
- Theo cấu hình, init khởi động giao diện NSD



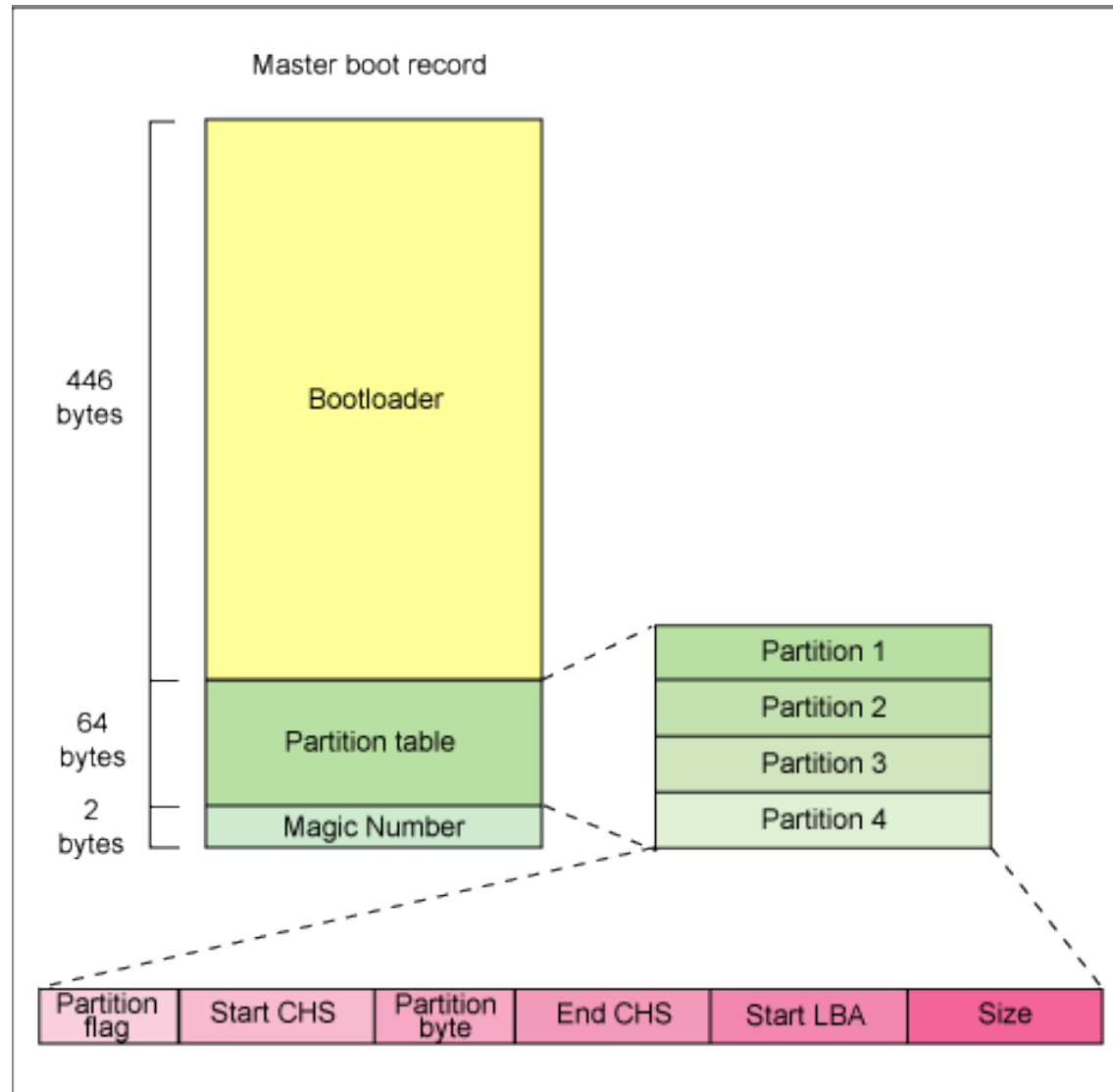
Khởi động hệ thống vật lý

- Phụ thuộc vào hệ thống vật lý
- Trên PC: BIOS
 - POST
 - Xác định và đánh dấu các thiết bị ngoại vi
 - Xác định thiết bị khởi động
 - Thực hiện MBR
 - MBR
 - Chương trình khởi động
 - Bảng các phân chương
 - Thực hiện boot record

MBR-Master Boot Record

- MBR là sector đầu tiên của ổ đĩa vật lý được phân chia thành nhiều ổ đĩa logic
- MBR nằm ngoài các phân chương
- MBR:
 - Chứa bảng các phân vùng chính
 - Chứa chương trình khởi động
- Mỗi phân vùng lại có một Boot Record riêng, chứa mã để khởi động Hệ điều hành nằm trong phân chương

MBR-Master Boot Record

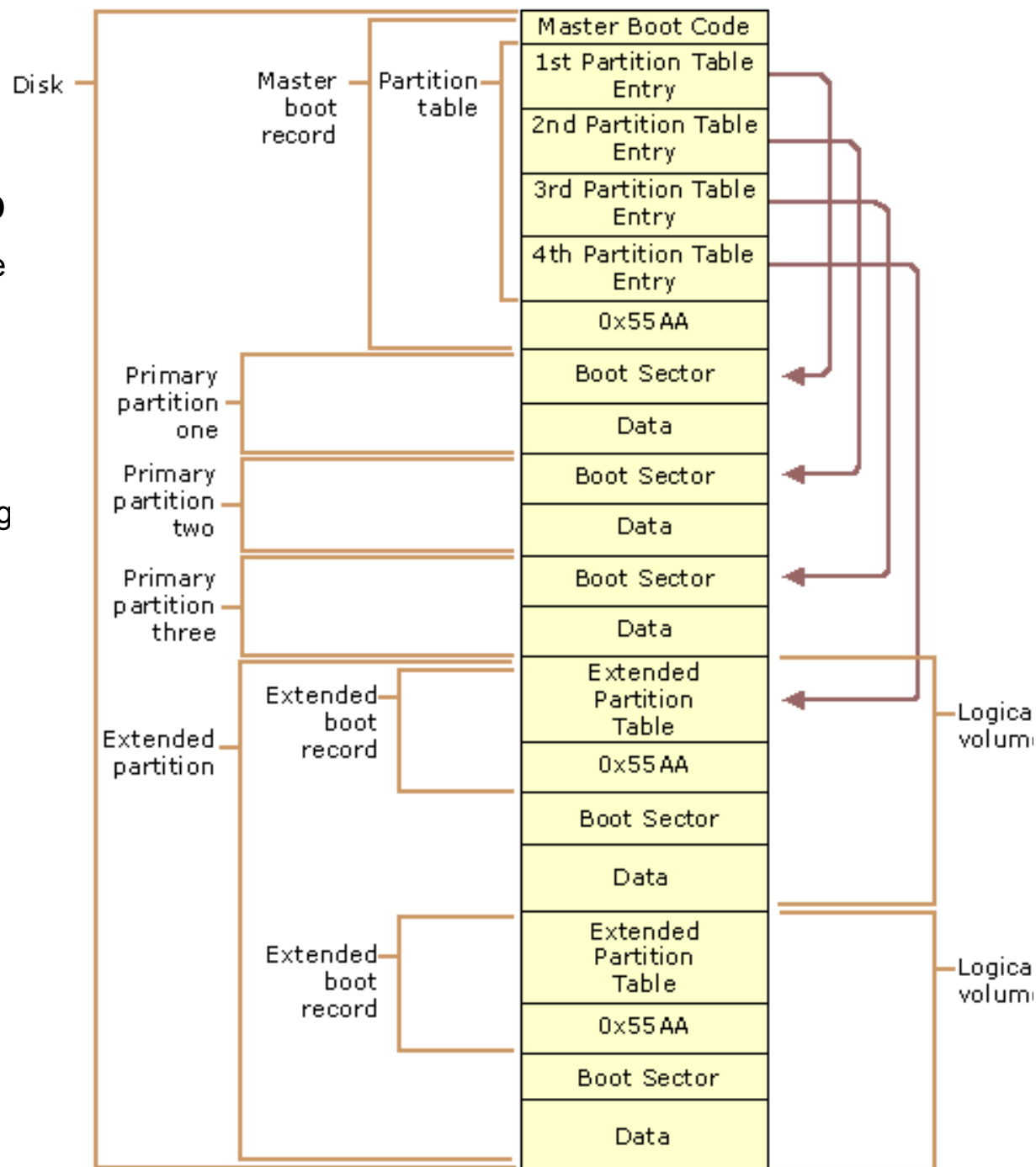


Các thành phần của MBR

- 446 bytes đầu tiên là chương trình boot loader → tải boot loader của một phân vùng tích cực
- 64 bytes tiếp theo là bảng các phân vùng
 - Tối đa 4 phân vùng
 - Flags, CHS (Cylinder-Head-Sector) đầu, Bytes, CHS cuối, LBA (Logical Block Addressing), size
- 2 bytes cuối cùng là magic number: đánh dấu cuối MBR.

Ví dụ về MBR phức tạp

1. Tải bảng phân vùng của Active partition (phân vùng tích cực)
2. Tìm sector đầu tiên của phân vùng tích cực
3. Nạp sector đầu tiên vào bộ nhớ
4. Chuyển điều khiển cho chương trình vừa được tải về



Chú ý

- Một máy tính có tối đa 4 ổ IDE
- Một ổ đĩa có thể có tối đa 4 phân vùng chính
- Trong 4 phân vùng chính, có 1 phân vùng có thể chuyển thành phân vùng mở rộng
- Trong phân vùng mở rộng, có thể có 4 phân vùng logic

Chương trình khởi động

- Chương trình nhỏ dùng để tải nhân ĐH
- Vị trí
 - 1st sector của HDD: 1st stage boot loader, trong MBR
 - 1st sector của phân vùng: 2nd stage boot loader.
- Chức năng
 - Tải nhân hệ điều hành vào bộ nhớ
 - Tải chương trình tải hệ điều hành vào bộ nhớ
 - Gọi chương trình khởi động trong boot sector của phân vùng khác
- Đơn giản
 - Không có xác thực
 - Không có bảo vệ (Boot sector virus)

2. Tải nhân hệ điều hành

- MBR hoặc boot sector có thể tải trực tiếp nhân hệ điều hành
 - Chỉ sử dụng các thao tác đọc đĩa bậc thấp và đơn giản
 - Không đọc được các tệp lớn, vị trí phức tạp (vd LBA)
- Thực tế
 - MBR tải một chương trình nhỏ (vẫn lớn hơn MBR) để chương trình này có thể tải nhân hệ điều hành
 - Phức tạp hơn, nhiều bước hơn
 - Nhân HĐH có thể phức tạp hơn
- CT nhỏ: ntosloader, lilo, grub

Lilo Boot Loader

```
boot = /dev/hda #boot loader ở MBR
delay = 40
compact
vga = normal
root = /dev/hda1
read-only
image = /zImage-2.5.99
    label = try # tên ở menu khởi
động
image = /zImage-1.0.9
    label = 1.0.9
other = /dev/hda3
    label = dos
    table = /dev/hda
```

- Đặt tại MBR của HDD hoặc Sector đầu tiên của Partition
- Để đơn giản, nhân hệ điều hành + các thành phần khác của HĐH cần thiết cho việc tải HĐH đặt ở trong thư mục/boot
- Cho phép lựa chọn HĐH để khởi động
- Cấu hình lilo
 - Thay đổi tệp /etc/lilo.conf
 - Sử dụng lệnh lilo để
 - Đọc nội dung tệp cấu hình
 - Ghi nội dung này vào MBR
- Có thể kiểm tra cấu hình trước khi ghi

LILO Boot step

- L- Loader OK
- LI- Second stage Loader OK
- LIL? Thấy Kernel không tải được
- LIL- Kernel sai qui cách
- LILO- Thành công

Grub bootloader



Grub bootloader

- Grand Unified Bootloader
- Được cài tại MBR
- Dùng để tải grub loader
- Grub loader tải hệ điều hành trong thư mục boot

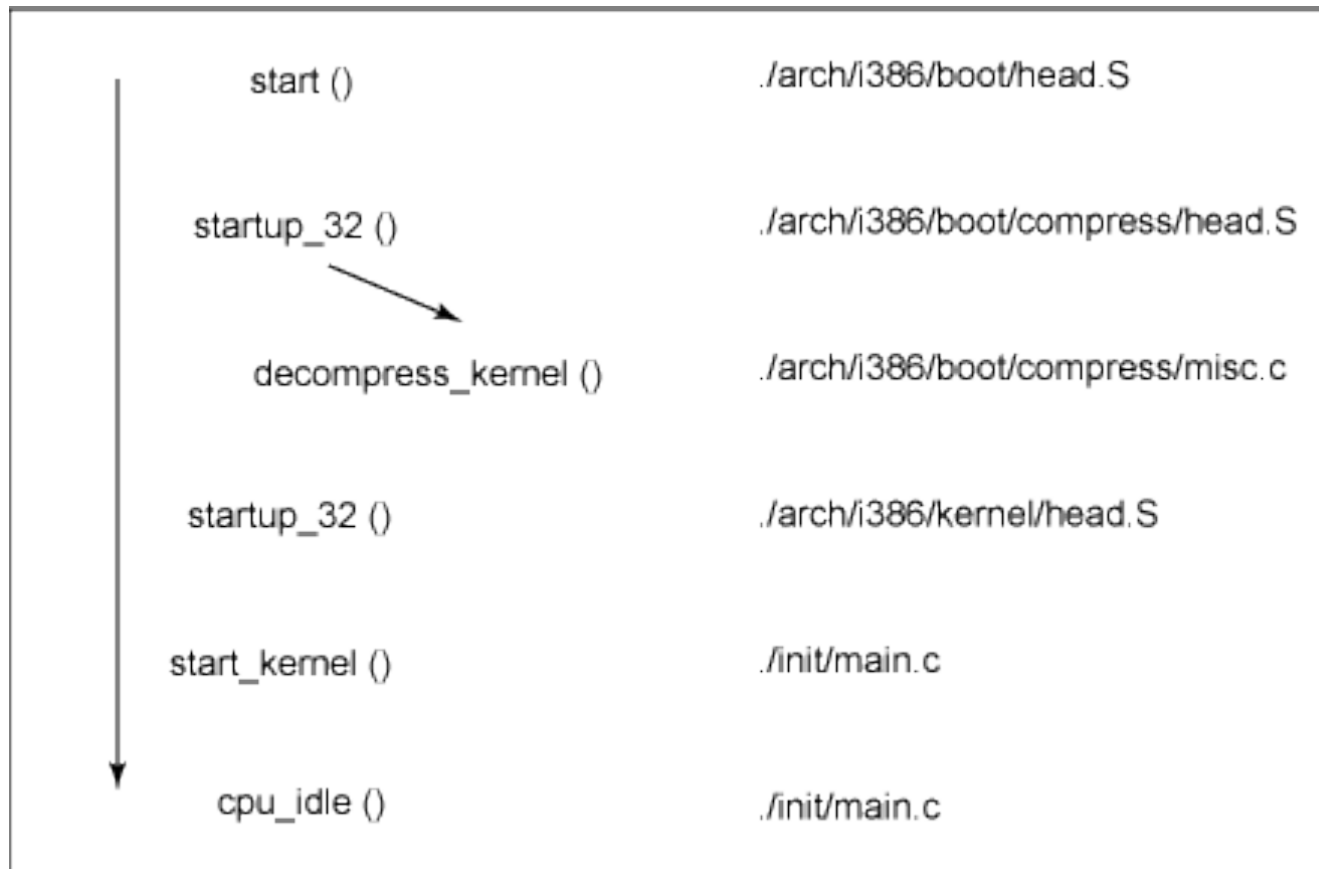
Cấu hình grub

- Grub 1.5: /boot/grub.conf
- MBR không thay đổi khi thay đổi boot/grub.conf
- Quá trình thực hiện thay đổi do Grub bước 2 thực hiện
- Cho phép thay đổi tham số boot khi khởi động máy

Các tham số khi boot

- Vga: các chế độ màn hình text khi boot
- Root: chỉ ra ổ đĩa sẽ là hệ thống tệp /
- Label: tên lựa chọn HĐH khi khởi động
- Các thông số thiết bị khác

Kernel boot



2. Khởi động các tác vụ hệ thống và mức thực hiện

- Các mức thực hiện
 - Sau khi tải nhân hệ điều hành, một số các tác vụ được thực hiện
 - Tác vụ đầu tiên là tác vụ init (chạy pstree để xem)
 - Các tác vụ khác được tải vào theo cấu hình yêu cầu của NSD
 - Các yêu cầu khác nhau được định nghĩa theo 6 mức thực hiện
 - Mỗi mức thực hiện bao gồm các tác vụ khác nhau được kích hoạt

Các mức thực hiện-run level

Mức thực hiện	Mô tả
0	Tắt máy
1	Đơn NSD, không đồ họa, không mạng, không dịch vụ
2	Đa NSD, không đồ họa, không mạng
3	Đa NSD, không đồ họa, mạng
4	Chưa dùng
5	Đa NSD, đồ họa, mạng
6	Khởi động lại
S	Đơn NSD, không đồ họa, không mạng, không dịch vụ

Mức thực hiện-Debian

Mức thực hiện	Mô tả
0	Halt
1	Đơn NSD, không đồ họa, không mạng
2-5	Đa NSD, đồ họa, mạng
6	Khởi động lại

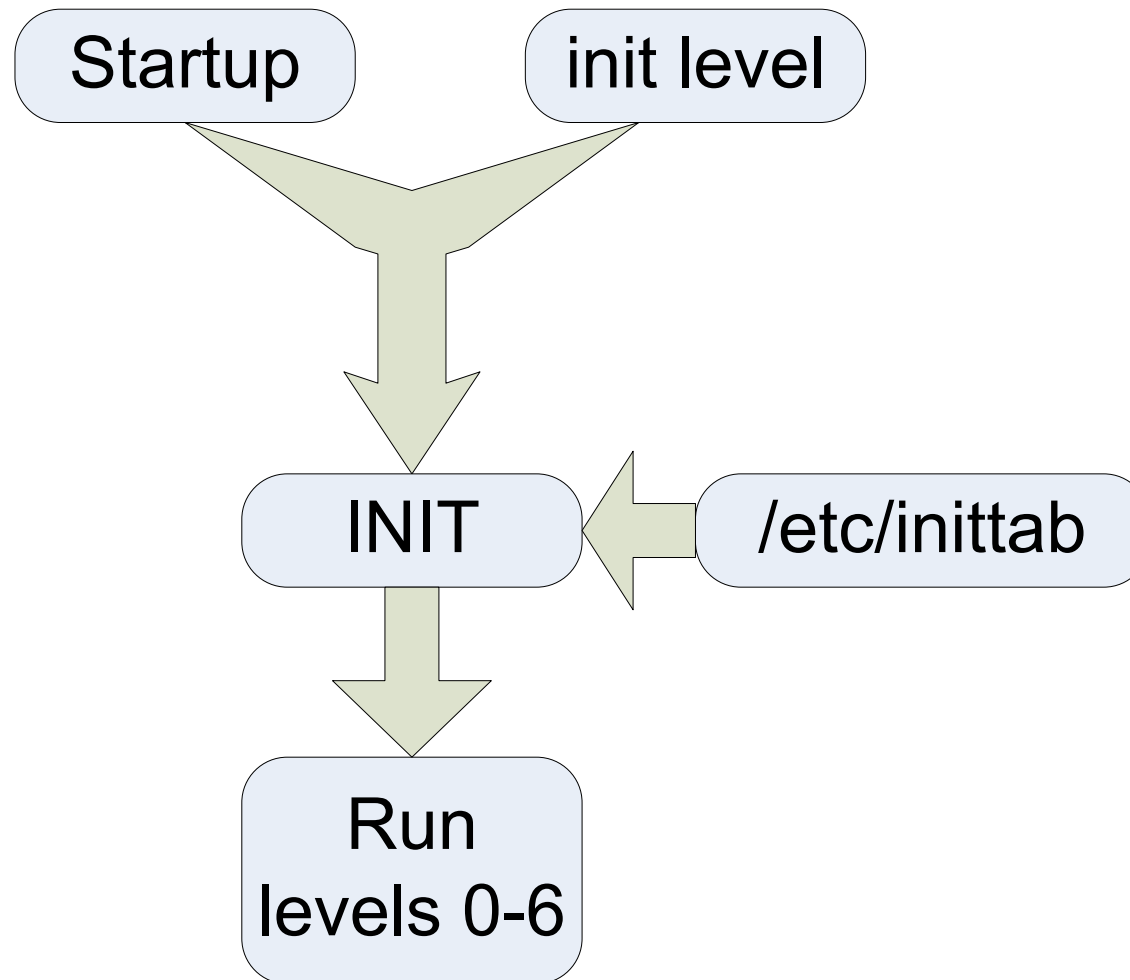
Các mức thực hiện-redhat

Mức thực hiện	Mô tả
0	Tắt máy
1	Đơn NSD, không đồ họa, không mạng, không dịch vụ
2	Đa NSD, không đồ họa, không mạng
3	Đa NSD, không đồ họa, mạng
4	Chưa dùng
5	Đa NSD, đồ họa, mạng
6	Khởi động lại
S	Đơn NSD, không đồ họa, không mạng, không dịch vụ

Câu lệnh quản lý mức thực hiện

- runlevel
- init
- telinit
- initctl

Các tác vụ hệ thống



/etc/inittab

- Chứa danh sách các tiến trình cần thực thi khi boot tùy theo các mức run-level

inittab

<u>ID</u>	:	<u>runlevel</u>	:	<u>action</u>	:	<u>command</u>
(1)		(2)		(3)		(4)

	Field	Description
(1)	ID	Identifier uniquely assigned to the entry.
(2)	Runlevel	Runlevel in which the description of the entry is reflected. The action of this entry will be executed in all runlevels when omitting it.
(3)	Action	Definition that shows how process of the entry is executed sysinit : Execute it before accessing the console. initdefault : Runlevel of the default to give to 'init' powerfail : Execute when it receive the 'POWER FAIL' signal. wait : Wait for the termination of the process. respawn : Keep the started process status constant.
(4)	Command	Command to be executed

```
inittab      [----]  0 L:[  1+16  17/ 54] *(575 /1663b)= #  35 0x23
```

```
#
# inittab      This file describes how the INIT process should set up
#               the system in a certain run-level.
#
# Author:      Miguel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#               Modified for RHS Linux by Marc Ewing and Donnie Barnes
#
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
```

```
id:5:initdefault:
```

```
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
```

```
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
```

```
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

```
# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
```

```
1Help  2Save  3Mark  4Replac 5Copy  6Move  7Search 8Delete 9PullDn 10Quit
```

rc: startup directories

Left			Right		
File	Command	Options	File	Command	Options
<- /etc/rc.d			<- /etc/rc.d/rc3.d		
Name	Size	MTime	Name	Size	MTime
/..	UP--DIR		@K74nscd	14	May 22 08:13
/init.d	4096	Jun 5 20:38	@K74ntpd	14	May 22 08:16
/rc0.d	4096	May 22 08:17	@K85mdmpd	15	May 22 07:53
/rc1.d	4096	May 22 08:17	@K89named	15	May 22 07:52
/rc2.d	4096	May 22 08:17	@K89netplugd	18	May 22 07:51
/rc3.d	4096	May 22 08:17	@K89rdisc	15	May 22 07:51
/rc4.d	4096	May 22 08:17	@K94diskdump	18	May 22 07:52
/rc5.d	4096	Jun 7 17:33	@S05kudzu	15	May 22 07:52
/rc6.d	4096	May 22 08:17	@S06cpuspeed	18	May 22 07:52
*rc	2371	Nov 1 2004	@S08iptables	18	May 22 07:52
*rc.local	220	Jun 24 2003	@S09isdn	14	May 22 07:53
*rc.sysinit	17935	May 10 2005	@S09pcmcia	16	May 22 07:53
			@S10network	17	May 22 07:51
			@S12syslog	16	May 22 07:51
			@S13portmap	17	May 22 07:53
			@S14nfslock	17	May 22 07:53
			@S15mdmonitor	19	May 22 07:53
			@S18auditd	16	May 22 07:51
			@S18rpcidmapd	19	May 22 07:53
			@S19rpcgssd	17	May 22 07:53
			@S25bluetooth	19	May 22 07:52
			@S25netfs	15	May 22 07:51
			@S26apmd	14	May 22 07:53
			@S26lm_sensors	20	May 22 07:54
			@S28autofs	16	May 22 07:52
			@S33nifd	14	May 22 07:54
			@S34mDNSResponder	23	May 22 07:54
			@S44acpid	15	May 22 07:52
			@S55cups	14	May 22 07:52
*rc			-> ../init.d/ntpd		

Hint: Leap to frequently used directories in a single bound with C-\.
 [root@webserver rc3.d]#

1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit

rc: startup directories

- **rc.d/** thư mục chứa các script, mỗi script cho một dịch vụ
- Các dịch vụ này được rc gọi khi khởi động
- **rc.shutdown** tắt các dịch vụ này khi shutdown
- Các script cũng được gọi khi thực hiện các hoạt động stop, start, restart, reload... điều khiển các dịch vụ

Login

- Để đăng nhập vào hệ thống, NSD cần có tài khoản
- Có thể đăng nhập vào terminal
- Mặc định, hệ thống linux có 6 terminal (tty1-tty6), tty; teletype writer
- tty 7 cho giao diện đồ họa
- Chuyển đổi giữa các giao diện, dùng Alt-Fx
- Thay đổi số lượng tty trong inittab

inittab [----] 0 L:[19+35 54/ 54] *(1663/1663b)= <EOF>

System initialization.

si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0

l1:1:wait:/etc/rc.d/rc 1

l2:2:wait:/etc/rc.d/rc 2

l3:3:wait:/etc/rc.d/rc 3

l4:4:wait:/etc/rc.d/rc 4

l5:5:wait:/etc/rc.d/rc 5

l6:6:wait:/etc/rc.d/rc 6

Trap CTRL-ALT-DELETE

ca::ctrlaltdel:/sbin/shutdown -t3 -r now

When our UPS tells us power has failed, assume we have a few minutes

of power left. Schedule a shutdown for 2 minutes from now.

This does, of course, assume you have powerd installed and your

UPS connected and working correctly.

pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

If power was restored before the shutdown kicked in, cancel it.

pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

Run gettys in standard runlevels

1:2345:respawn:/sbin/mingetty tty1

2:2345:respawn:/sbin/mingetty tty2

3:2345:respawn:/sbin/mingetty tty3

4:2345:respawn:/sbin/mingetty tty4

5:2345:respawn:/sbin/mingetty tty5

6:2345:respawn:/sbin/mingetty tty6

Run xdm in runlevel 5

x:5:once:/etc/X11/prefdm -nodaemon

1 Help 2 Save 3 Mark 4 Replac 5 Copy 6 Move 7 Search 8 Delete 9 PullDn 10 Quit

Dịch vụ đơn lẻ

- Chương trình được thực hiện bởi hệ thống
- Thực hiện bởi một script đặt trong thư mục `/etc/rc.d/init.d/` hoặc `/etc/init.d/`
- Các thư mục `/etc/rx#.d/` chứa các liên kết biểu tượng tới các script của dịch vụ
- K-tắt, S-bật

Script thực hiện dịch vụ

- Cung cấp các thao tác
 - Bật, tắt, khởi động lại, cấu hình lại, khởi động lại có điều kiện, trạng thái
 - Tạo ra các tệp log để lưu trạng thái dịch vụ
 - Kiểm tra các điều kiện cần thiết để thực hiện dịch vụ
- Ví dụ: `pico /etc/init.d/crond`

Các dịch vụ thực hiện khi khởi động

- Trong thư mục của các mức thực hiện, có các liên kết tới các script thực hiện các dịch vụ
- K=kill
- S=start
- Số thứ tự quyết định dịch vụ nào được khởi động trước
- Có thể được cấu hình
 - Bằng tay, câu lệnh, giao diện tương tác

chkconfig

- Công cụ quản lý các startup directory rc.d
- 5 thao tác
 - Hiển thị trạng thái khởi động của dịch vụ
 - Thêm dịch vụ
 - Bớt dịch vụ
 - Thay đổi trạng thái khởi động của dịch vụ
 - On/Off/Reset
- Trạng thái khởi động mặc định của dịch vụ
 - Lưu trong script của dịch vụ

TCP daemon

- Theo dõi các yêu cầu thiết lập kết nối
- Nếu cần thiết, khởi tạo dịch vụ để xử lý yêu cầu
 - Chuyển điều khiển cho dịch vụ (theo yêu cầu)
 - Chuyển điều khiển cho dịch vụ (một lần)
- → siêu server
 - inetd, xinetd

Chức năng của TCP daemon

- Tiết kiệm tài nguyên hệ thống
- Quản lý danh sách truy cập, logging,
- Các dịch vụ thông dụng được khai báo trong `/etc/services`

inetd

- Nghe các cổng được quy định cho các dịch vụ Internet: FTP, POP3, Telnet, ...
- Khi có gói tin TCP hoặc UDP đến một trong các cổng này, inetd kích hoạt server dịch vụ tương ứng
- Inet nối stdin, stdout, stderr của server dịch vụ với socket tại cổng
 - Dịch vụ ít tải: sử dụng bộ nhớ hiệu quả vì server dịch vụ không cần lo phần kết nối mạng do inetd chịu trách nhiệm
 - Dịch vụ tải lớn, thường xuyên: server riêng nghe cổng này. VD: httpd

Quản lý truy cập

- Inetd quản lý truy cập mạng thông qua 2 danh sách
 - /etc/host.allow
 - /etc/host.deny