Booting Linux system

Contents

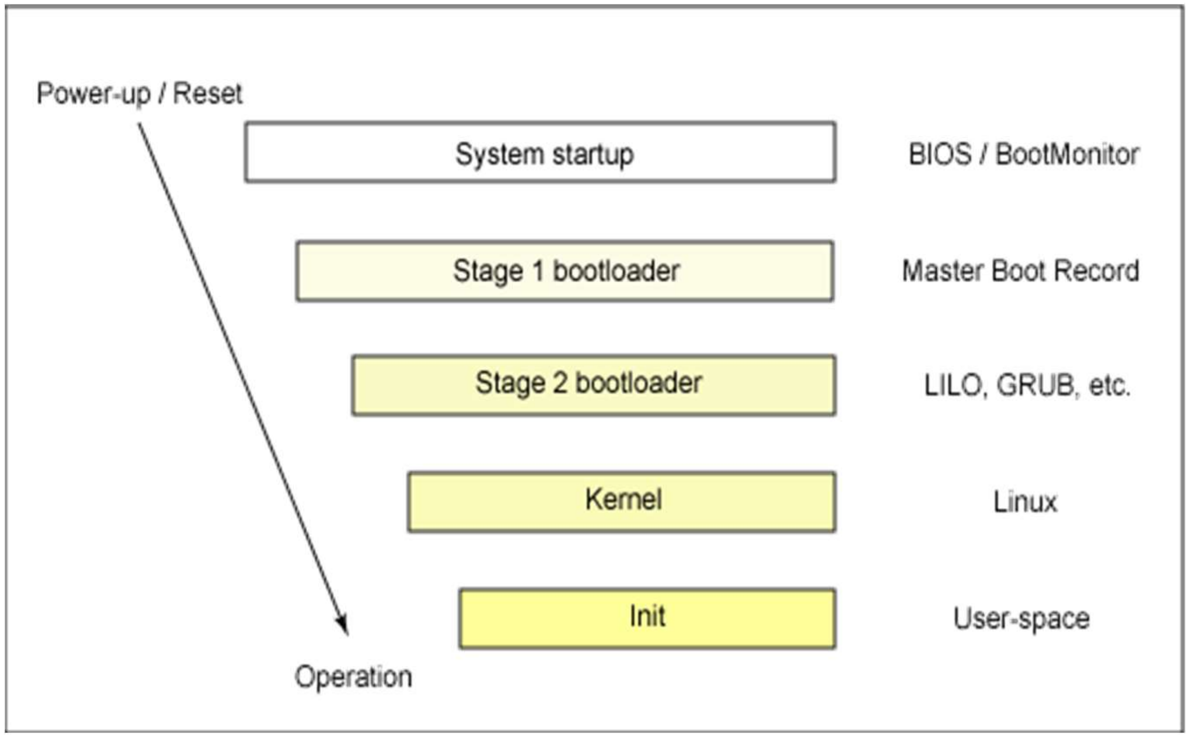
1. The overview of booting

The goals of booting

- Start up hardware
- Check device status
- Boot up applications for users

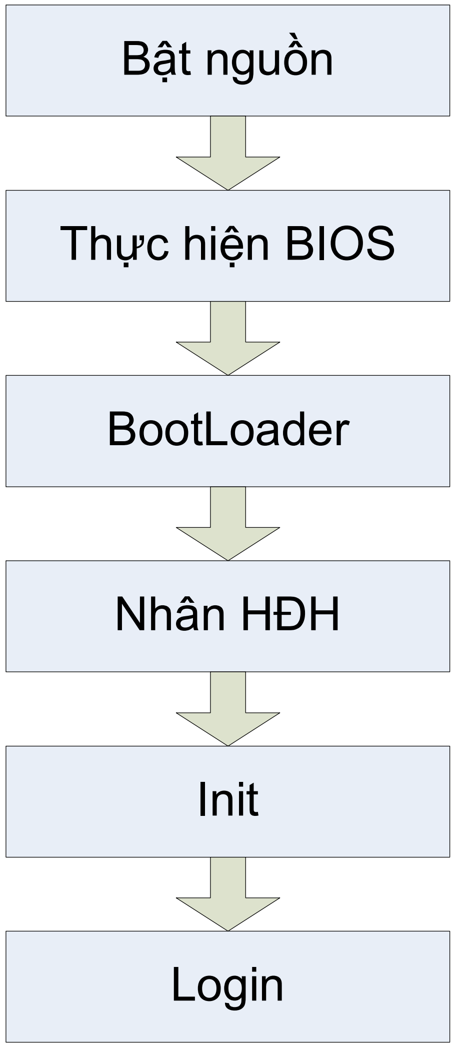
Specifically, when booting a PC

- Start up hardware
- Start up MBR
- Execute booting software (OS menu)
- Start up OS kernel
- Start up user applications
- Depending on systems, some stages can be combined together



Booting progress in Linux

- Power on
  - The system automatically checks hardware devices
- Execute BIOS program
  - Configure peripherals
  - Access main storage devices
- Start up booting program (bootloader)
  - Load OS kernel
  - Start up services for OS
- OS execute init process
  - Start up the process and working environment
- According to configuration, init starts up user interface



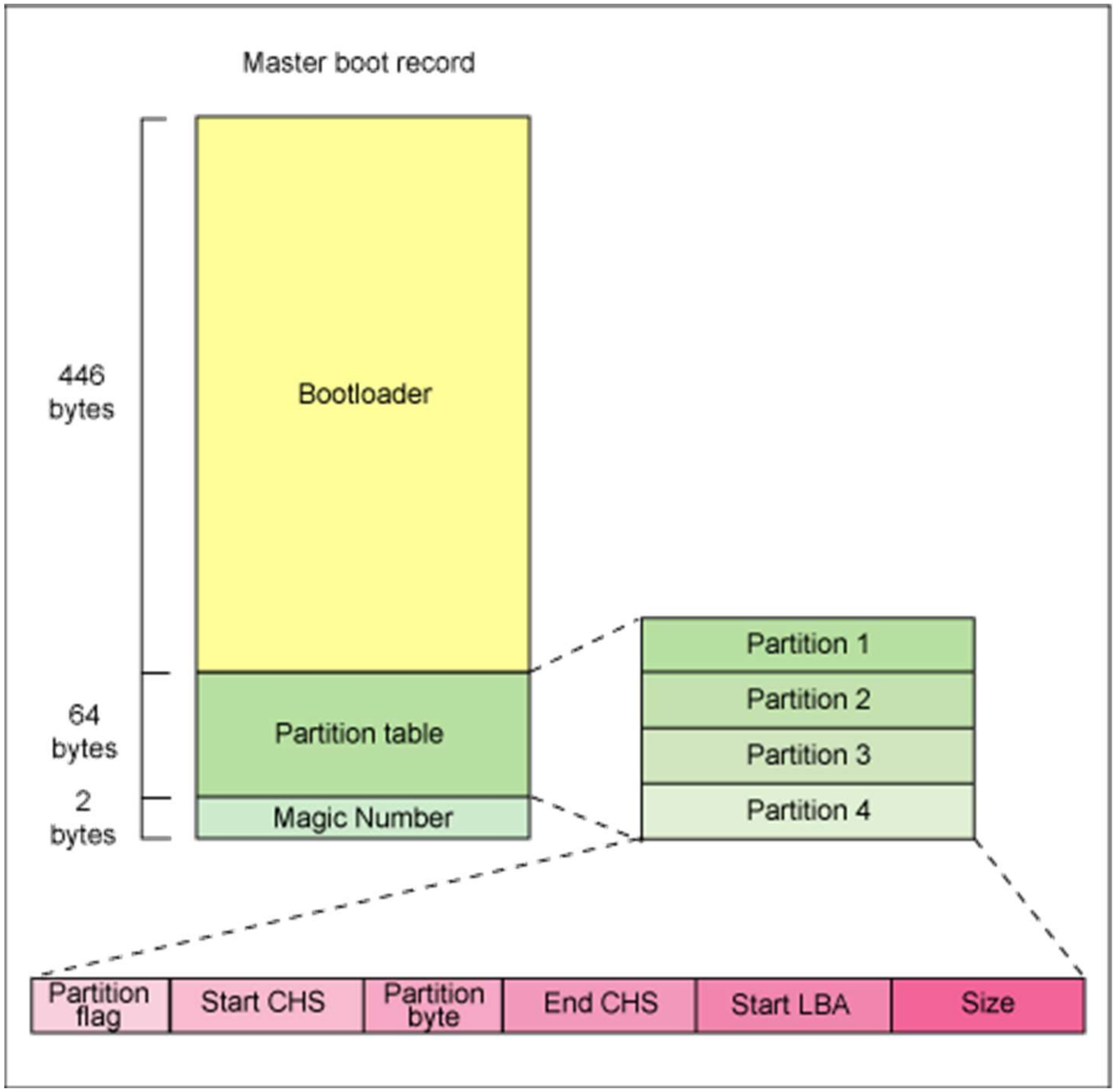
Booting hardware devices

- Depending on physical systems
- On PC: BIOS
  - POST
  - Allocate and mark peripherals
  - Determine the booting device
  - Execute MBR
  - MBR
    - Bootloader
    - Partition Table: how the logical partitions are organized
  - Execute boot record

MBR-Master Boot Record

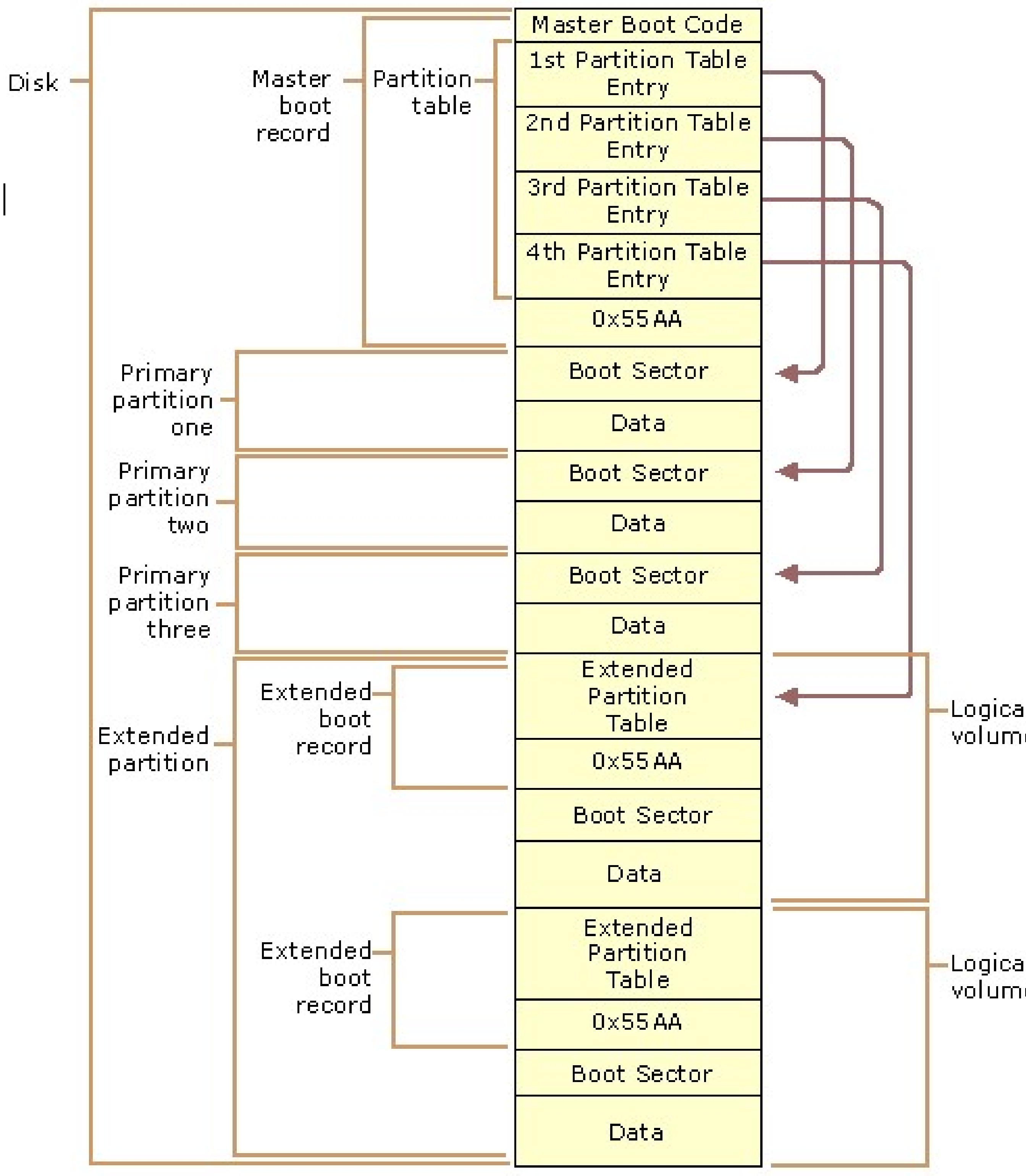
- MBR the first sector of a physical hard drive
- MBR is outside of Partition Table
- MBR:
  - Contains Partition Table
  - Contains bootloader
- Each partition has its own Boot Record, containing script/program to start up OS inside the partition table

MBR-Master Boot Record



Example of a complex MB

1. Load partition table of the Active partition
2. Find the first sector of the active partition
3. Load first sector into memory
4. Switch the control to the bootloader



Notes

- Each physical disk can have up to 4 primary partitions
- One of 4 primary partitions can be converted to many logical partitions

Bootloader

- A small program to load OS kernel
- Location
  - 1st sector of HDD: 1st stage boot loader, in MBR
  - 1st sector of each partition : 2nd stage boot loader.
- Functions
  - Load OS kernel to memory
  - Load 2nd stage bootloader to memory
  - Call bootloader in boot sector of other partition.
- Simple
  - No authentication
  - No protection (Boot sector virus)

2. Load OS kernel

- MBR or boot sector can directly load OS kernel
  - Only use simple and low-level disk-reading operations
  - Cannot read big files, complex locations (Ex: LBA)
- Practical method
  - MBR loads a small program (but larger than MBR) and let this program load OS kernel
  - More complex, more steps
  - OS kernel can be more sophisticated
  - Phức tạp hơn, nhiều bước hơn
  - Nhân HĐH có thể phức tạp hơn
- Example: ntosloader, lilo, grub

Lilo Boot Loader

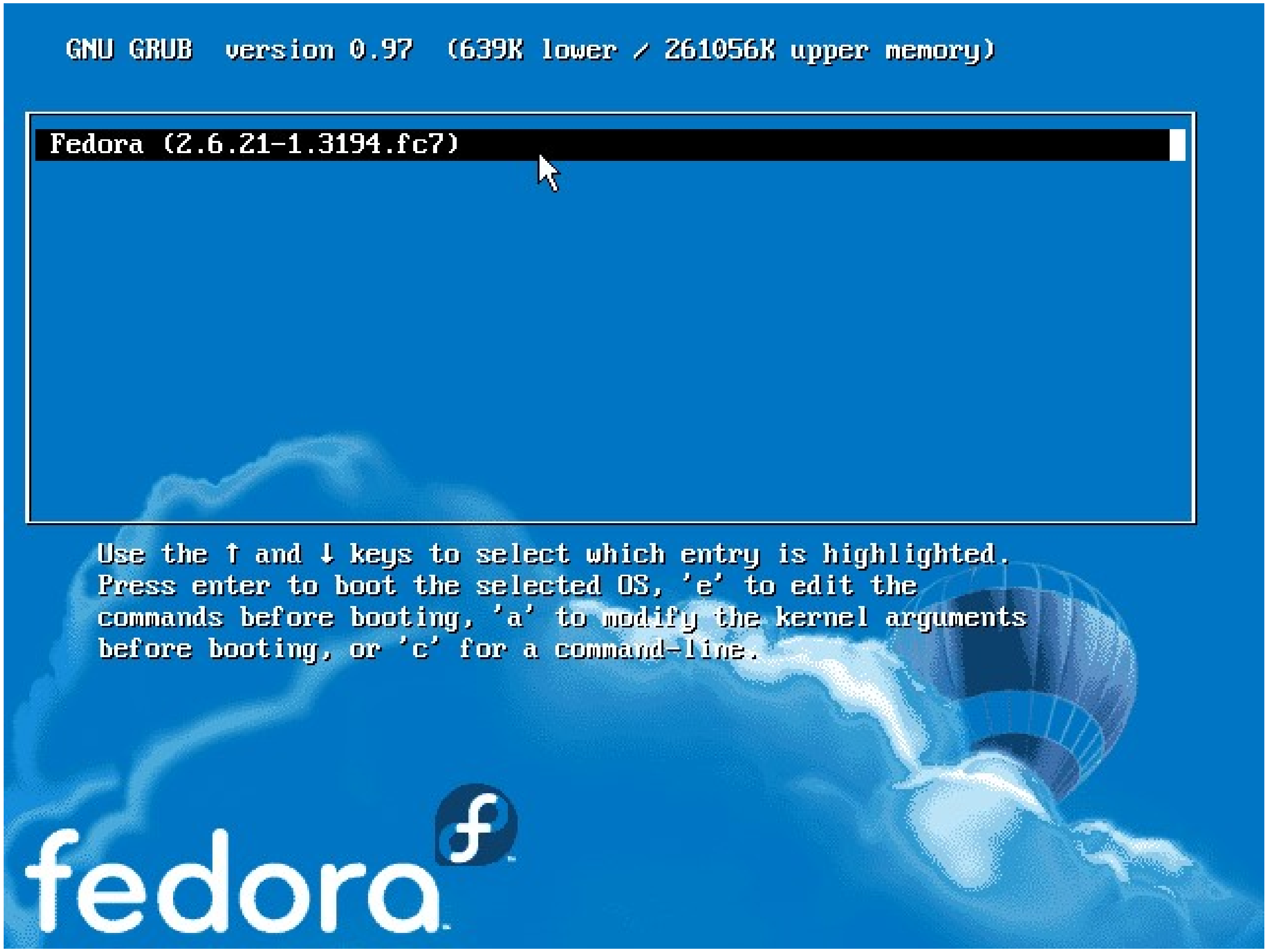
```
boot = /dev/hda #boot loader ở MBR
delay = 40
compact
vga = normal
root = /dev/hda1
read-only
image = /zImage-2.5.99
        label = try # tên ở menu khởi
động
image = /zImage-1.0.9
        label = 1.0.9
other = /dev/hda3
        label = dos
        table = /dev/hda
```

- Location: MBR of HDD or first sector of a partition
- To simplifty, OS kernel is stored in /boot
- Allow to select OS to boot
- Lilo configuration
  - /etc/lilo.conf
  - Use lilo command to
    - Read configuration file
    - Write changes to MBR
  - Can check the configuration before writing

LILO Boot step

- L- Loader OK
- LI- Second stage Loader OK
- LIL? Found Kernel but cannot load
- LIL- Wrong kernel format
- LILO- Sucessful

Grub bootloader



```
GNU GRUB  version 0.97  (639K lower / 261056K upper memory)

Fedora (2.6.21-1.3194.fc7)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, 'a' to modify the kernel arguments
before booting, or 'c' for a command-line.
```

Grub bootloader

- Grand Unified Bootloader
- At MBR
- Use to load grub loader
- Grub loader loads kernel OS in /boot

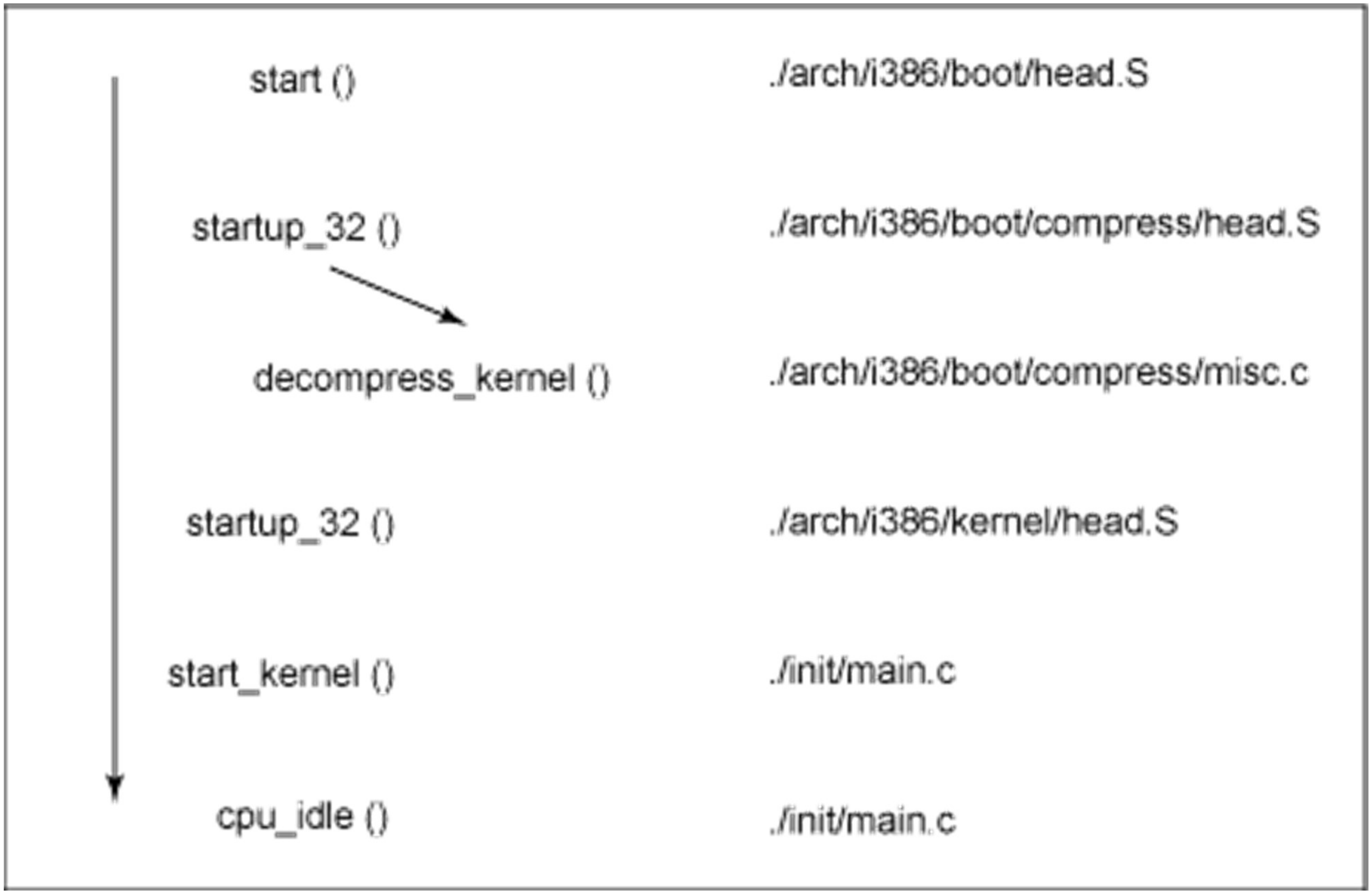
Grub configuration

- Grub 2: /boot/grub2/grub.conf
- MBR doesn’t change after modifying /boot/grub2/grub.conf
- The updating progress will be done by the grub of step 2
- Allow to change parameters while booting

Booting parameters

- Vga: text screen while booting
- Root: the disk of root directory /
- Label: Name of OS when booting
- Other parameters

Kernel boot


start ()
./arch/i386/boot/head.S
startup_32 ()
./arch/i386/boot/compress/head.S
decompress_kernel ()
./arch/i386/boot/compress/misc.c
startup_32 ()
./arch/i386/kernel/head.S
start_kernel ()
./init/main.c
cpu_idle ()
./init/main.c

2. SysVInit and run-level

- Run-level
  - After loading OS kernel, some tasks will be executed
  - The first one is init
  - Other tasks/processes will be loaded according to user requirements
  - There are 6 different run-levels
  - Each run-level includes different activated tasks

Run-level in Debian distributions

Run-level	Description
0	Halt
1	Single user, no graphic, no network
2-5	Multiple user, graphic, network
6	Restart

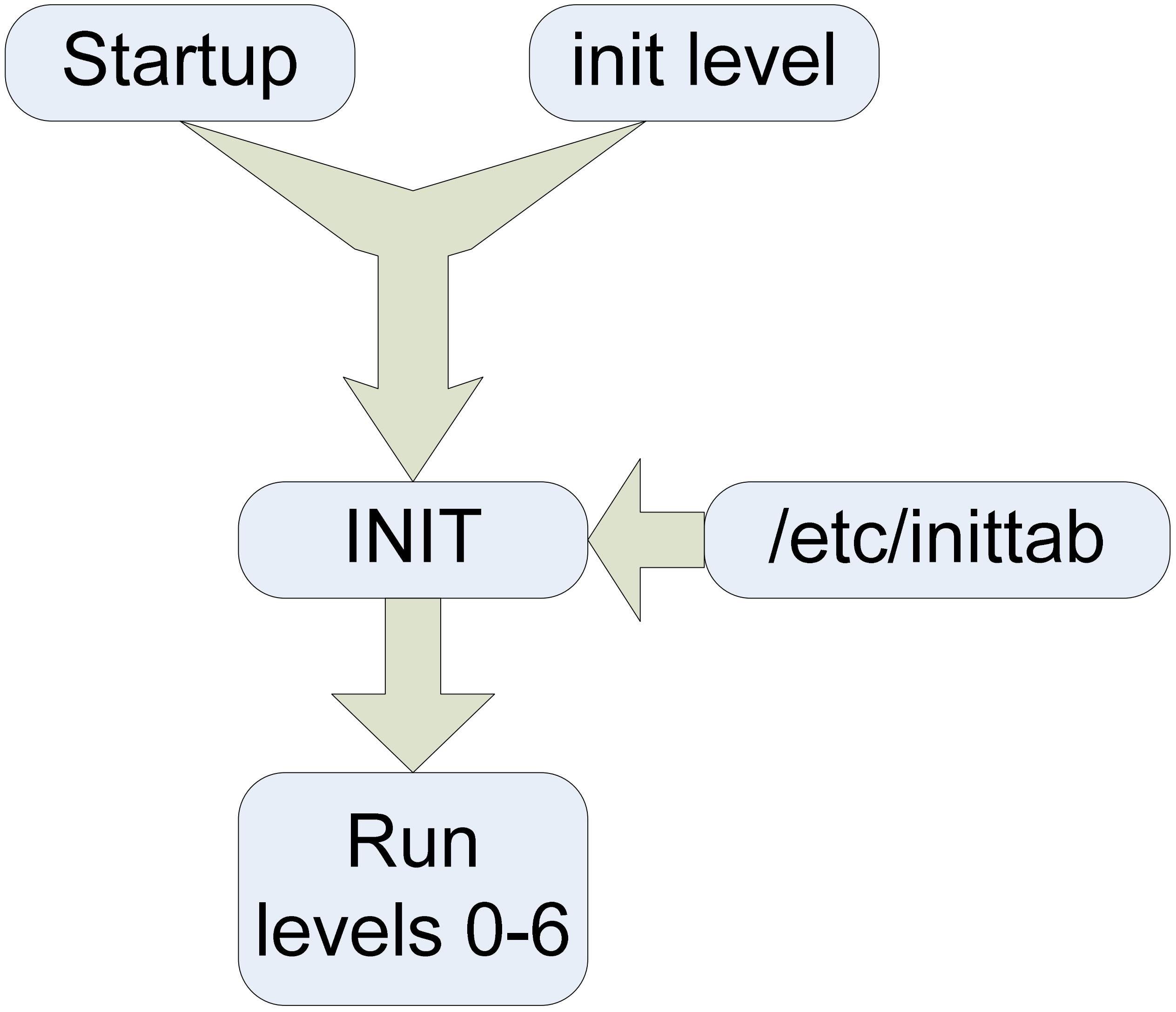
Run-level in Redhat distributions

Run-level	Description
0	Turn off
1	Single user, no graphic, no network, no service
2	Multiple user, no graphic, no network
3	Multiple user, no graphic with network
4	No use
5	Multiple user, graphic, network
6	Restart
S	Single user, no graphic, no network, no service

Manage run-levels

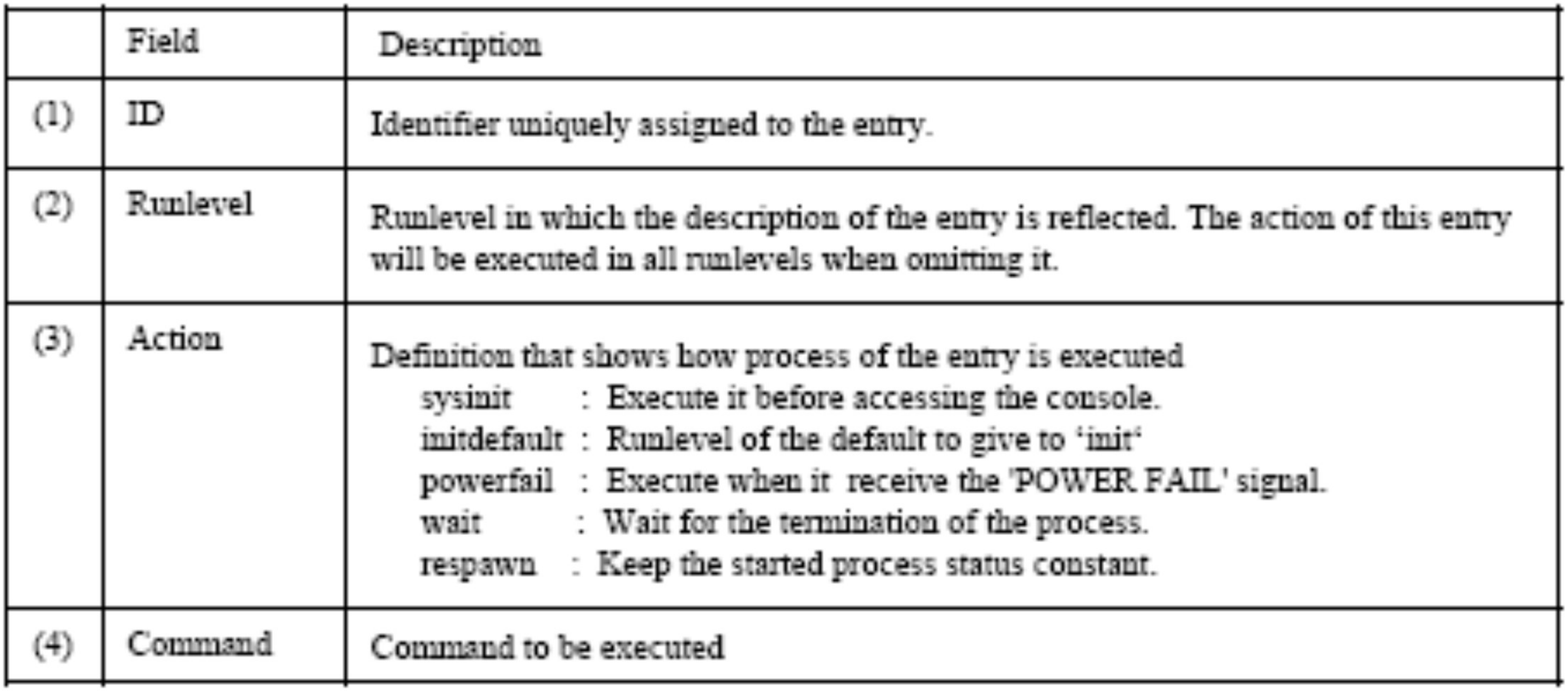
- runlevel: find the current and previous runlevels
- init
- telinit: change system run-level
- initctl: allows a system administrator to communicate and interact with the Upstart init(8) daemon

Manage run-levels

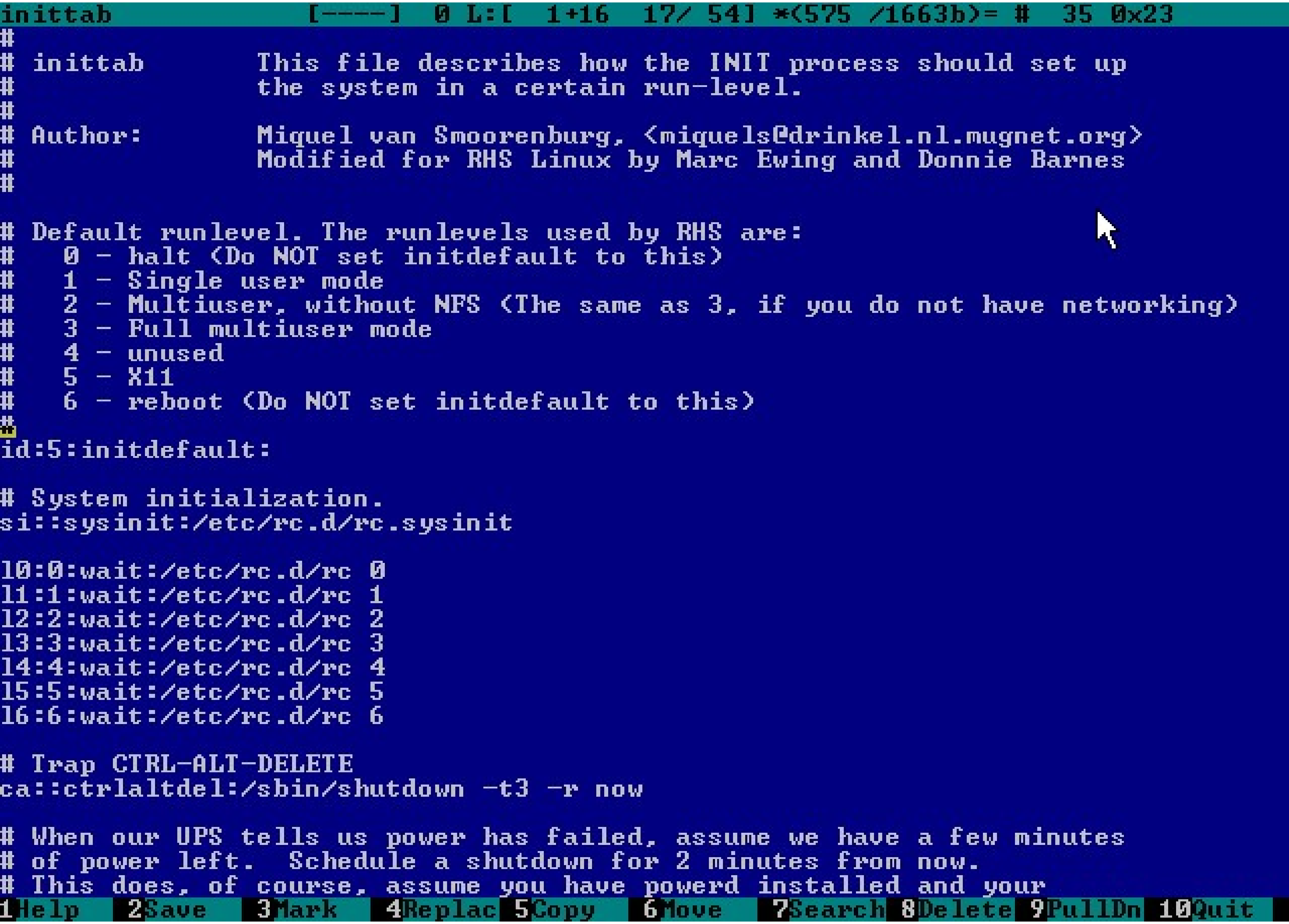


inittab

```
ID : runlevel : action : command
(1)    (2)        (3)        (4)
```

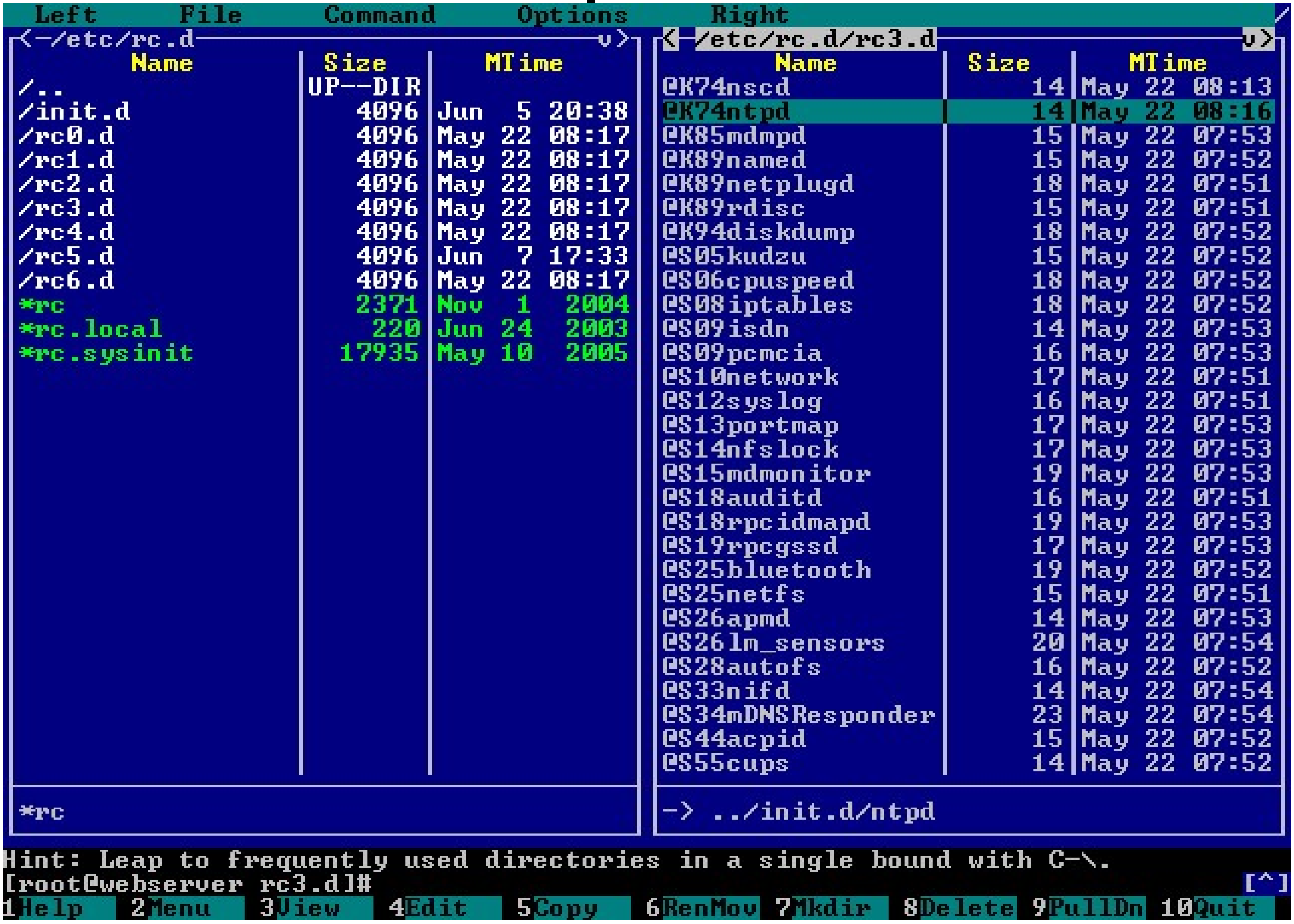


	Field	Description
(1)	ID	Identifier uniquely assigned to the entry.
(2)	Runlevel	Runlevel in which the description of the entry is reflected. The action of this entry will be executed in all runlevels when omitting it.
(3)	Action	Definition that shows how process of the entry is executed sysinit : Execute 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:       Miquel 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
```

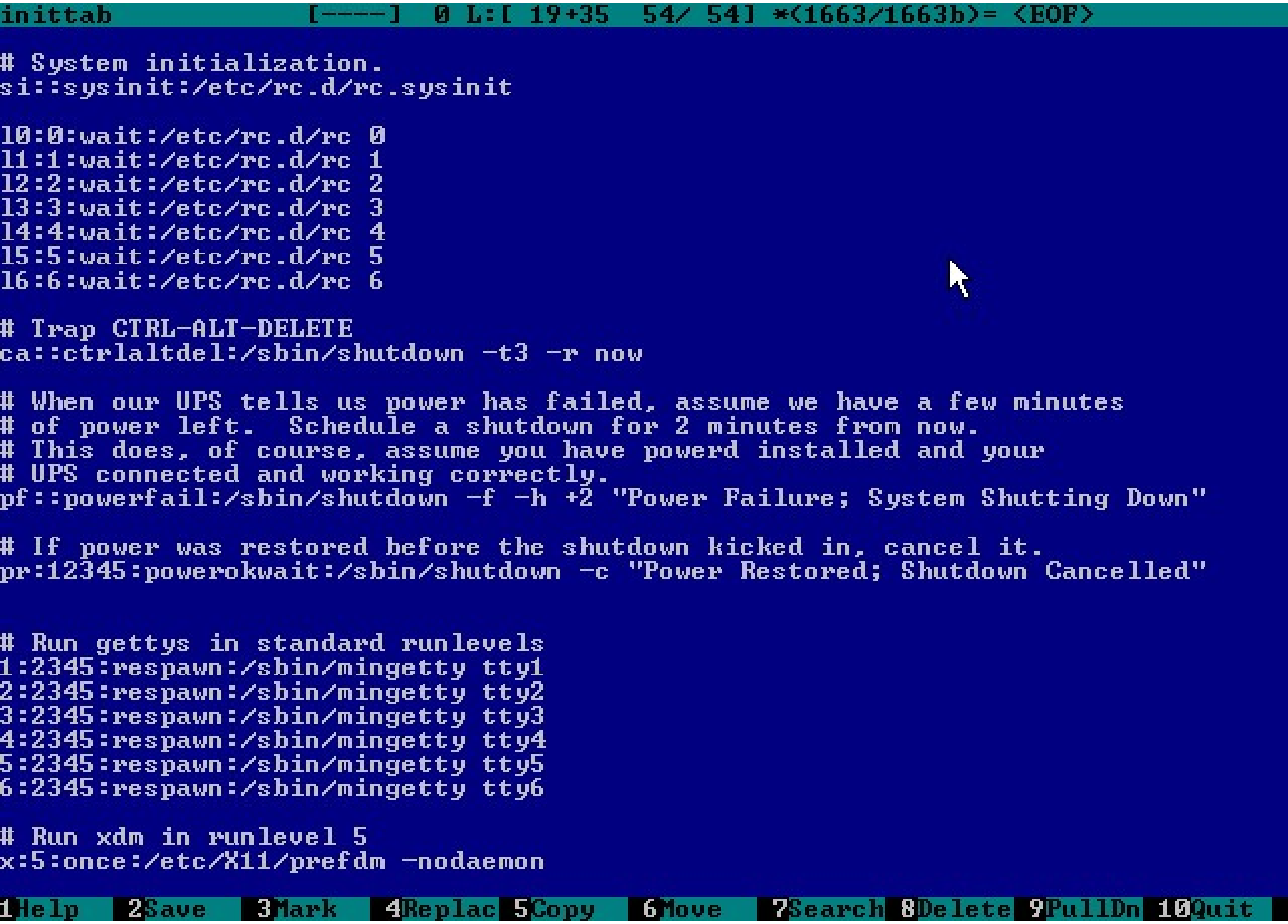
rc: startup directories



```
  Left       File       Command      Options      Right
┌<─/etc/rc.d──────────────────────────────v>┐┌< /etc/rc.d/rc3.d──────────────────────v>┐
│      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
```

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

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

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

Most of current Linux distribution abandoned init / inittab !!!!!!!!

Why not init / inittab?

- Init is a process with PID=1
- If init cannot start, the system will fall to "kernel panic" state.
- A starup process can only be started after the previous one was successfully loaded → Slow and unstable booting
- Replacement
  1. **Upstart** – WAS used for Ubuntu GNU/Linux (support up to Ubuntu 16). It was designed to start up asynchronously.
  2. **Epoch** – was built around the simplification of managing services and processes.
  3. **Mudar** – was written on Python, is used for Pardus GNU/Linux, is designed to start up asynchronously.
  4. **systemd** – was designed to run process parallel, is uded for many current Linux systems such as Ubuntu, Fedora, OpenSuSE, Arch, RHEL, CentOS, etc.

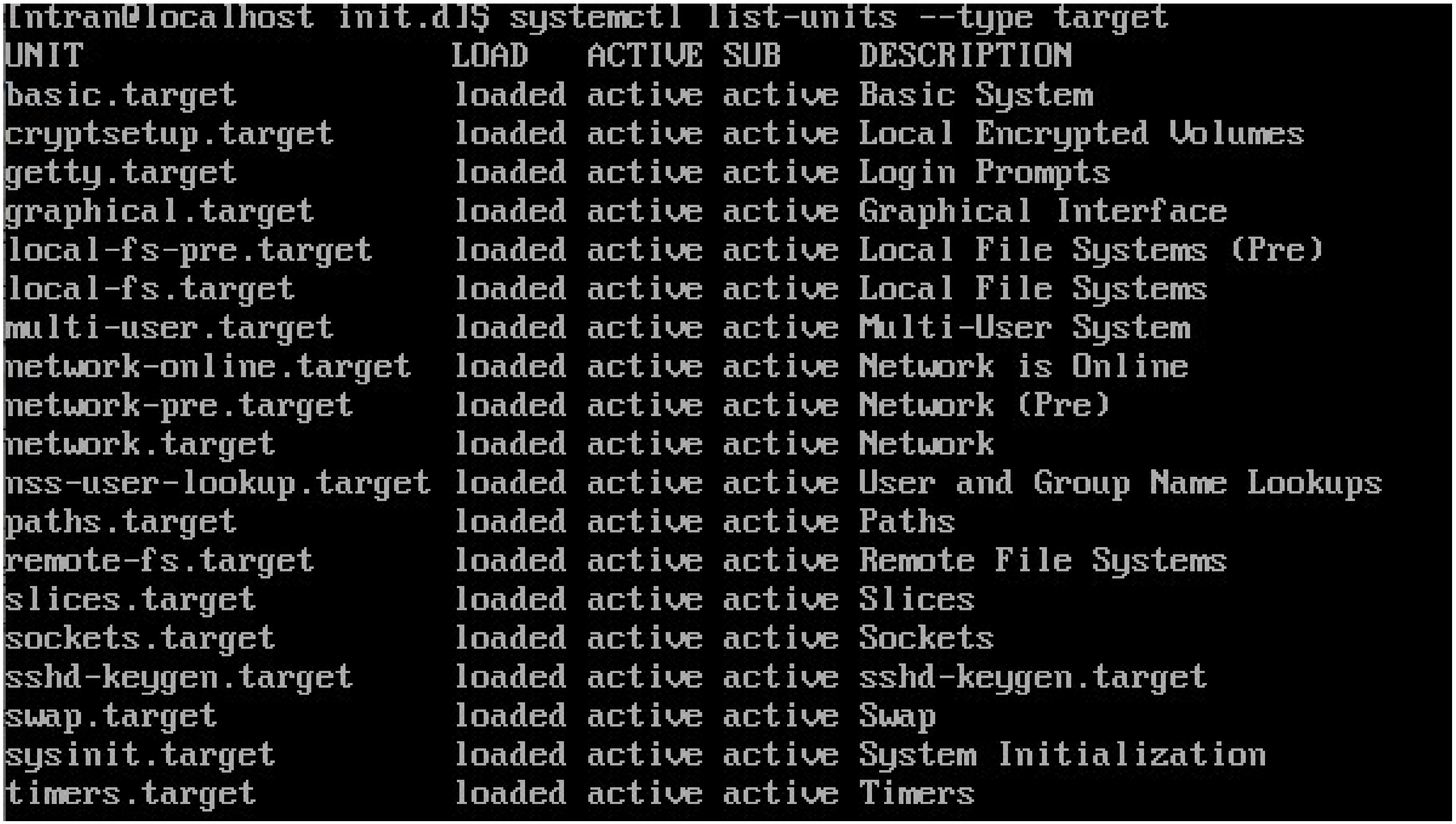
Target in systemd

- Target is used to group services to synchronous points for services
- Replace runlevel
- Each target is considered as a OS state and services will be activated depending on OS states

Runlevel at systemd

- Target system: set of OS states and regulation of services and processes need to be run at that state
- New targets are equivalent with old runlevel
  - poweroff.target *(runlevel 0)*: Power off
  - rescue.target *(runlevel 1)*: start rescue shell
  - multi-user.target *(runlevel 2,3,4)*: multiple users at console mode
  - graphical.target *(runlevel 5)*: graphical mode and network service
  - reboot.target *(runlevel 6)*: restart system

Examples



```
[ntran@localhost init.d]$ systemctl list-units --type target
UNIT                      LOAD   ACTIVE SUB    DESCRIPTION
basic.target              loaded active active Basic System
cryptsetup.target         loaded active active Local Encrypted Volumes
getty.target              loaded active active Login Prompts
graphical.target          loaded active active Graphical Interface
local-fs-pre.target       loaded active active Local File Systems (Pre)
local-fs.target           loaded active active Local File Systems
multi-user.target         loaded active active Multi-User System
network-online.target     loaded active active Network is Online
network-pre.target        loaded active active Network (Pre)
network.target            loaded active active Network
nss-user-lookup.target    loaded active active User and Group Name Lookups
paths.target              loaded active active Paths
remote-fs.target          loaded active active Remote File Systems
slices.target             loaded active active Slices
sockets.target            loaded active active Sockets
sshd-keygen.target        loaded active active sshd-keygen.target
swap.target               loaded active active Swap
sysinit.target            loaded active active System Initialization
timers.target             loaded active active Timers
```

Structure of a service in systemd

- Each service is represented by a “service unit” (with the tail of name as .service), saved in /lib/systemd/system/ (or /run/systemd/system or /etc/systemd/system)
  - etc > run > lib : priority
- Content of a service service-name.service is similar to a INI file in Windows
- Each service’s configuration can be modified by file *service-name.service.d/*.conf* in the same folder
- Change files of a service
  - systemctl edit [--full] application
  - Note: while modifying, the directory with the name “*.d” will be created inside /…/systemd/system for each service. Removing this folder means deleting modified configurations
  - systemctl daemon-reload

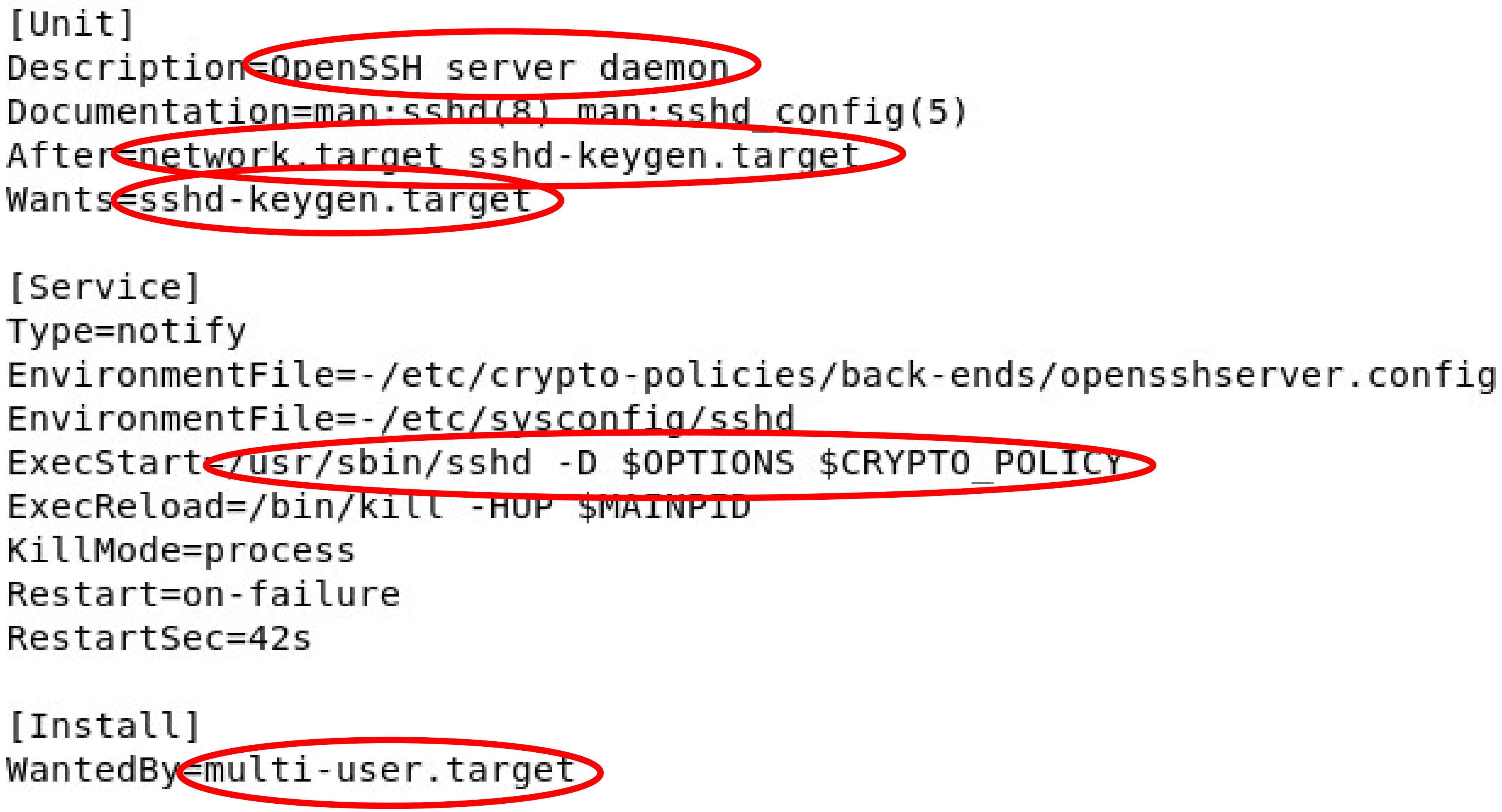
How to create a service with systemd

- Create a file inside the directory systemd/system under a service-name.service with the following format

```
[Unit]
Description=<description about this service>

[Service]
User=<user e.g. root>
WorkingDirectory=<directory_of_script e.g. /root>
ExecStart=<script which needs to be executed>
Restart=always

[Install]
WantedBy=multi-user.target
```

- Reload the daemon service by systemctl
  - sudo systemctl daemon-reload
- Restart the service
  - sudo systemctl start your-service.service

Example of sshd.service



```
[Unit]
Description=OpenSSH server daemon
Documentation=man:sshd(8) man:sshd_config(5)
After=network.target sshd-keygen.target
Wants=sshd-keygen.target

[Service]
Type=notify
EnvironmentFile=-/etc/crypto-policies/back-ends/opensshserver.config
EnvironmentFile=-/etc/sysconfig/sshd
ExecStart=/usr/sbin/sshd -D $OPTIONS $CRYPTO_POLICY
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s

[Install]
WantedBy=multi-user.target
```

systemctl

- The command is used to control other parts of systemd
- Only root or users with root privilege (through sudo)
- Simplify the configuration of service/ application while booting
- Manage all through UNIT

Some commands of systemctl

- List information
  - systemctl [list-units] [--all] [--state=inactive]
  - systemctl [list-units] [--all] [--type=service]
  - systemctl list-unit-files
- Manage information of UNIT
  - systemctl cat UNIT
  - systemctl list-dependencies UNIT
  - systemctl show sshd.service
- Some commands to change runlevel
  - systemctl rescue/halt/poweroff/reboot

Manage services/programs with systemd

- systemctl [start/stop/restart/reload] application
  - start/stop/restart/reload a service
  - Restart vs reload?
  - Nor sure? → reload-or-restart
- systemctl [enable/disable] application
  - enable/disable a application while booting
- systemctl status application.service
- systemctl [is-active/is-enabled/is-failed] application.service

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