



# Linux Administration

## Boot Process on a Linux Computer

Lecturer:  
Di Dio Lavore, Galtarossa  
Politecnico di Milano  
[www.polimi.it](http://www.polimi.it)

---

© 2003 - Di Dio Lavore, Galtarossa

## Summary



- 
- The boot process
  - Bootstrapping the System
  - **GRUB**
  - **LILO**
  - After Loading
  - **INIT** - The first process
  - BSD boot system
  - System V boot system
  - Boot Administration
  - X-based Interfaces

# The boot process

---



- Linux system can boot in either automatic mode or manual mode
- The boot process consists in 6 steps:
  - ▶ Loading and initialization of the Kernel
  - ▶ Device detection and configuration
  - ▶ Creation of spontaneous system processes
  - ▶ Operator intervention (*manual boot only*)
  - ▶ Execution of system startup scripts
  - ▶ Creation of the login sessions

# Bootstrapping the System

---



- *Bootstrap* is the first operation that will be executed when the computer starts
- It comes from the expression “pull itself up by its own bootstraps”, because the computer must run the Operating System by itself
- The Computer search the Operating System in FD, CD, HD and others boot devices
- Booting from HD:  
the Computer loads the first 512 bytes from the HD, the Master Boot Record (MBR)
- MBR contains the *boot loader* that will load the Linux Kernel
- Linux has two famous Boot Managers: **GRUB** and **LILLO**

## GRUB (GRand Unified Boot loader)



- GRUB is a Boot Loader for different systems (GNU/Linux, GNU/Hurd...)
- It provides good interaction with users during the boot process
- At the boot time GRUB provides a prompt and the user can write the boot options. "Help" will display the possibilities

```
grub> help [ENTER]
...
root [DEVICE[HDBIAS]]
geometry DRIVE [CYLINDER HEAD SECTOR]
```

## GRUB (2)



- Booting options:

```
grub>kernel (hd0,0)/vmlinuz-2.4.2 root=/dev/hda6 ro [ENTER]
```

The system will boot with the root partition **/dev/hda6** (read-only) and will search the Linux Kernel (**/vmlinuz-2.4.2**) in the first partition of the first IDE disk

- GRUB supports the completion like the **BASH** shell
- All the boot options can be saved in a configuration file in **/boot/grub/grub.conf** (by default)
- The user can change this options without reinstalling grub

## LILO (Linux LOader)



- LILO is de-facto the Boot Manager for Linux
- It provides useful boot options for the boot process
- The `/etc/lilo.conf` file contains all the boot options
- The command `lilo` (as *root*) installs lilo in the MBR with the options saved in the `lilo.conf` file
- After a change in `lilo.conf` the user must type `lilo`

### NOTE:

Remember to run `lilo -t` to test the config file (lilo will display possible configuration errors)

- `fdisk /mbr` will erase lilo from the MBR and replace it with the DOS boot loader

## LILO (Lilo.conf: an example)



```
boot=/dev/hda #The boot device (MBR)
vga=normal    #The VGA mode for the console; normal=80x25
read-only     #Mount the system in read-only mode
prompt        #Display the boot prompt
timeout=200   #Set a timeout before booting
message=/boot/message #Display a welcome message

image=/boot/vmlinuz #The Kernel to launch
    label=linux      #The label for The Linux OS
    root=/dev/hda4   #The root partition
    password=stallmantorvalds #The boot password
    restricted       #Password only with parameters

other=/dev/hda1 #Another OS
    label=windows   #The label for the second OS
    table=/dev/hda  #The partition table for this OS
```

## LILO (Kernel Parameters)



- At boot time the user can pass some parameters to the Kernel to force a system behaviour
- Example: `boot: linux single`

OPTION	MEANING
<code>root=/dev/foo</code>	The root partition
<code>single</code>	Boot in single user mode
<code>init=/sbin/init</code>	The path of the INIT program
<code>ether=0,0,eth1</code>	An Ethernet device

## After Loading



- In a Linux Box the Kernel makes some hardware checks (RAM available...), then launches the **INIT** process
- The **INIT** process makes deeper checks (the script `kudzu` detects new hardware peripherals), some system settings (date,time,hostname...) and launches the system daemons (`apmd`, `syslogd`, `sendmail`...)
- **INIT** launches **GETTY** process, that allows the users to login. **GETTY** launches **LOGIN** that will wait for a login name and a password
- Now the system is ready and the users can login.

# INIT - The first process

---



- INIT is the first process to be executed
- INIT runs the system services
- In Red Hat, Mandrake and other distributions the user can select an interactive startup and which services will be run and which not
- Almost all distributions make the boot like the Unix System V
- Slackware makes the boot like Unix BSD

# BSD boot system

---



- BSD-based systems (like Slackware) set the runlevel in `/etc/inittab` and from this file launch the different initialization scripts
- Example: `rc:2345:wait:/etc/rc.d/rc.M`  
Default runlevels in Slackware:
  - 0 = halt
  - 1 = single user mode
  - 2 = unused
  - 3 = multiuser mode (default)
  - 4 = X11 with KDM/GDM/XWM
  - 5 = unused
  - 6 = reboot

## BSD boot system (2)



- All the scripts listed in `/etc/inittab` are in the `/etc/rc.d` directory

```
rc.0      #Kill all processes
rc.4      #Run the X Server
rc.6      #Like rc.0 but during reboot
rc.K      #Swithing to Runlevel 1
rc.M      #Daemons for multiuser mode
rc.S      #System initialization
rc.inet1  #Base networking system
rc.inet2  #Entire network system
...
```

## System V boot system



- All the initialization scripts are in located in `/etc/init.d`
- In `/etc/inittab` the user can select the runlevel
- Different directories link the system scripts according to the runlevel
- The links in a directory (to `/etc/init.d`) are prepared for a specific runlevel
- Directories location: `/etc/rcx.d`      `x=0..6`
- Others available files executed at boot time:

```
rc.local  #User defined scripts
rc.modules #Modules to load at boot time
```

## Shutdown the system

---



- Simple commands for system reboot or shutdown are:
  - ▶ `poweroff`
  - ▶ `shutdown -h now`
  - ▶ `halt`
  - ▶ `reboot`
- When the system will reboot or shutdown, the scripts in `/etc/rc[0|6].d` or `/etc/rc.d/rc.[0|6]` will be executed
- The system Administrator can send a message to inform all the users that the system will shutdown soon

## Boot Administration

---



- To change the current runlevel *root* can edit `/etc/inittab` and change the line:

```
id:n:initdefault: #n is the current runlevel
```

- System V: *root* can link or unlink the scripts in `/etc/init.d` to start or stop some services

```
Ex.: ln -s /etc/init.d/named /etc/rc3.d/S34named
```

- The “S” services (S34named) are started and the number shows an execution order
- The “K” services (K34named) are stopped. To switch to another runlevel some services must be stopped

## Boot Administration (2)

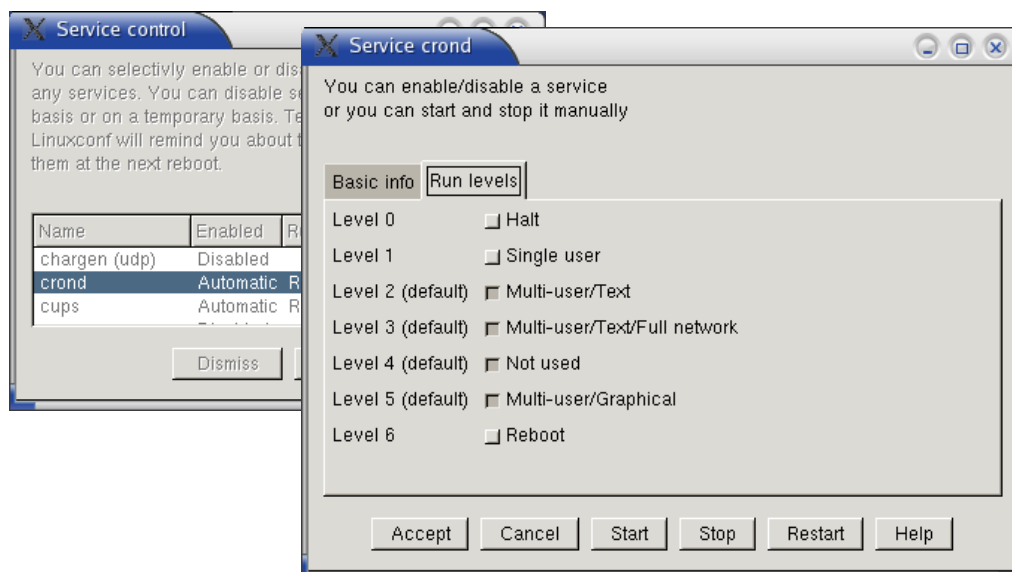


- BSD: *root* can make executable or not the services in `/etc/rc.d` (`chmod u+x /etc/rc.d/<script_name>`)
- Only the executable scripts are executed at boot time
- In Red Hat is available **SETUP** to configure the system behaviour
- It is a dialog-based script. You can select in a user interface the system services that will be started at boot time

## X-based Interfaces



- **LINUXCONF** provides a graphical interface to configure the system services and set the runlevel



## X-based Interfaces (2)



- **WEBMIN** is another tool (common in most distributions) that can configure the system runlevel and services

	Action	Start at boot?
<input type="checkbox"/>	<a href="#">alsa</a>	Yes
<input type="checkbox"/>	<a href="#">apmd</a>	Yes
<input type="checkbox"/>	<a href="#">atd</a>	Yes
<input type="checkbox"/>	<a href="#">crond</a>	Yes

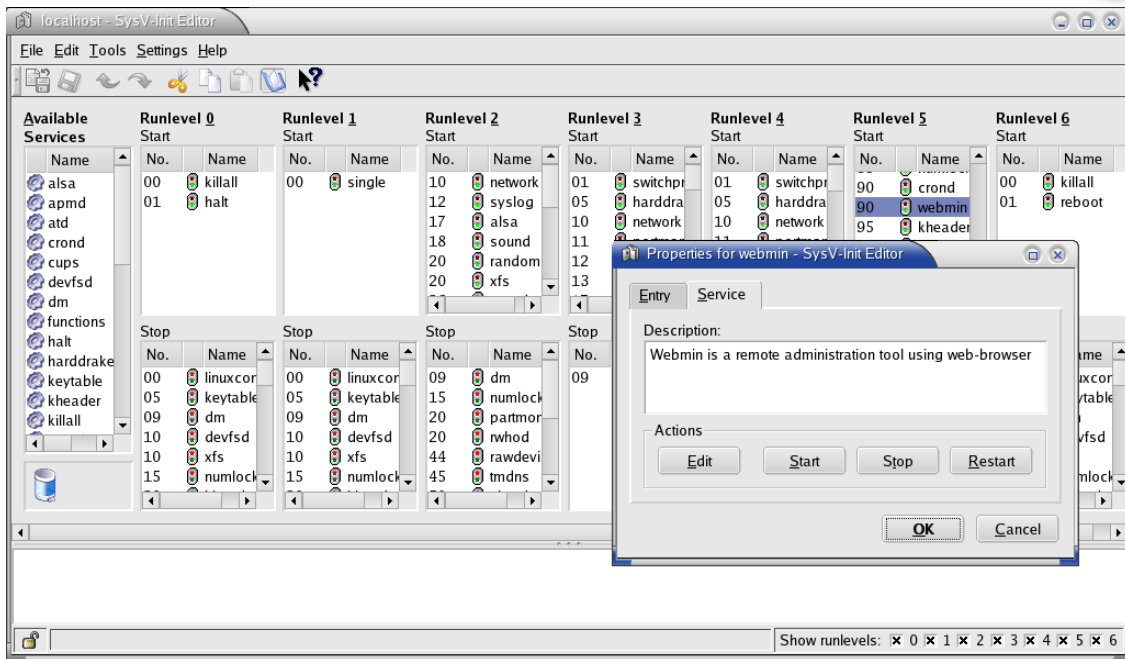


## X-based Interfaces (3)



- **KSYSV** is a KDE front-end to set up the services in a SysV-based system (like Mandrake) and can be used in others distributions
- All the front-end are not necessary to configure the boot system, but root can see the current configuration and select all the configuration options in a graphical interface

# KSYSV-A graphical tool for SysV booting system



## Reference



- “UNIX System Administration Handbook Third Edition” by Evi Nemeth, Garth Snyder, Scott Seebass, Trent R. Hein
- “Linux Administration Handbook” by Evi Nemeth, Garth Snyder, Trent R. Hein
- “Linux: The Complete Reference, Fifth Edition” by Richard L. Petersen
- “Bootprompt-HOWTO” - Paul Gortmaker ([www.tldp.org](http://www.tldp.org))

# System Commands

---



- `lilo` - installation of lilo in the MBR
- `poweroff` - shutdown of the system
- `shutdown` - like Poweroff but with more options
- `halt` - this command will kill all processes, unmount all the file systems, but don't use APM (Advanced Power Management) to turn off the computer
- `reboot` - this command will reboot the system
- `chmod` - change the permissions (read, write, execute) of a file or directory
- `ln` - link a file or a directory