

Linux Startup

by Mohamad H. Danesh

System Startup

- At first BIOS check the basic subsystem of computer.
- If all goes well, the BIOS will then look for a bootable volume.
- Next, the BIOS will look for boot code in the partition boot sector.
 - The first sector
 - 512 bytes (**446 bytes**:boot loader such as LILO or GRUB, **64 bytes**:partition table, **2 bytes**:special code).
 - This area usually contains a boot loader.

Boot process

- System startup (BIOS)
 - Search for boot devices,
- Stage 1 boot loader
 - Looking through Partition Table to find and load the secondary boot loader
- Stage 2 boot loader
 - Splash screen: display a list of available kernels
 - in `/etc/grub.conf`
 - Soft link to link `/etc/grub/menu.lst`
 - Load kernel image from fs to memory
 - Invoke kernel image

Boot process (cont)

- Kernel
 - initialise devices
 - mounts root filesystem
 - runs /sbin/init which is process number 1 (PID=1)

Sequence of Events

- Details about the specific sequence of events or what happens when the Linux kernel is loaded.
 - Initialize the CPU(s).
 - Set up kernel memory.
 - Initialize system devices.
 - Start memory handling.
 - Set up and mount the file system.
 - Start the **init** command.

init Command

- After the kernel has detected computer's hardware and load the correct device driver, init is started.
 - The last step of kernel booting.
- It's the parent of all processes.
 - PID = 1
- init role:
 - The primary role of init is to create processes from a script stored in **/etc/inittab**.
 - Default run level
 - Running scripts in **/etc/rc.d**.
 - /etc/rc.d/rc<number>.d

Run Levels

- A run-level is a software configuration of the system which allows only a selected group of processes to exist.

Run Levels (Cont.)

- 0 – halt (/etc/rc.d/rc0.d/)
- 1 – single user mode (/etc/rc.d/rc1.d/)
- 2 – multiuser, without NFS (/etc/rc.d/rc2.d/)
- 3 – full multiuser mode (/etc/rc.d/rc3.d/)
- 4 – unused (/etc/rc.d/rc4.d/)
- 5 – X11 (/etc/rc.d/rc5.d/)
- 6 – reboot (/etc/rc.d/rc6.d/)

/etc/inittab

- This file describes how the init process should setup the system in a certain run level.
- /etc/inittab format:
 - Id:run-level:action:process
 - Id: a unique identifier for the entry.
 - Run-level: run level number.
 - Action: which action should be taken (wait, boot, initdefault, ...)
 - Process: the process to be executed.

/etc/inittab (cont)

- Example:
 - id:4:initdefault:l
 - 0:0:wait:/etc/rc.d/rc.0l
 - 6:6:wait:/etc/rc.d/rc.6x
 - 1:4:wait:/etc/rc.d/rc.4
- The boot process uses these parameters to identify the default runlevel and the files that will be used by that runlevel.

/etc/rc.d

- rc
 - Performs master control of which scripts to execute.
- rc.sysinit
 - The system initialization script
- rc.local
 - Used for local initialization
- /etc/rc.d/rc*.d/ (* : 0–6)
 - Each file is merely a soft link to a script under init.d

/etc/rc.d/rc.sysinit

- This file is interpreted by init once at boot time.
- It contains bash shell script logic to perform some the following:
 - Sets the system hostname
 - Reads in network configuration data
 - Prints welcome banner for login
 - Configures the kernel
 - Sets up the system time
 - Sets the console and keyboard mapping

Run Level and Services

- The scripts (in /etc/rc.d/rc*.d/) are actually symbolic links to system service scripts under the /etc/rc.d/init.d/ directory.
- Some scripts begin with the letter **K**
- Some other scripts begin with the letter **S**
- There is a number followed by **K/S**
- Example:
 - **K12**mysql
 - **S10**network

Why symlinks?

- Not to put the same script in each runlevel directory of rc*.d
- Won't be able to define the order as per which the script will be executed during startup and shutdown.

Change Run Level

- The boot process(init command) will decide the runlevel to select and from that will decide the rc*.d script files to be run
- the scripts to be run are actually symbolic links to files in the directory /etc/init.d
 - So all you have to do is to write your startup script, place it in /etc/init.d,
 - and then create a symbolic link to it from the appropriate runlevel directory (or runlevel file, if that's what your system uses).