Chapter-7: Shell script & System Administration (8L)

TOPICS:**2**. **System Administration**

Essential duties of UNIX System administrator, Starting and shutdown, Brief idea about user account management (username, password, home directory, group id, disk quota, terminal etc.)

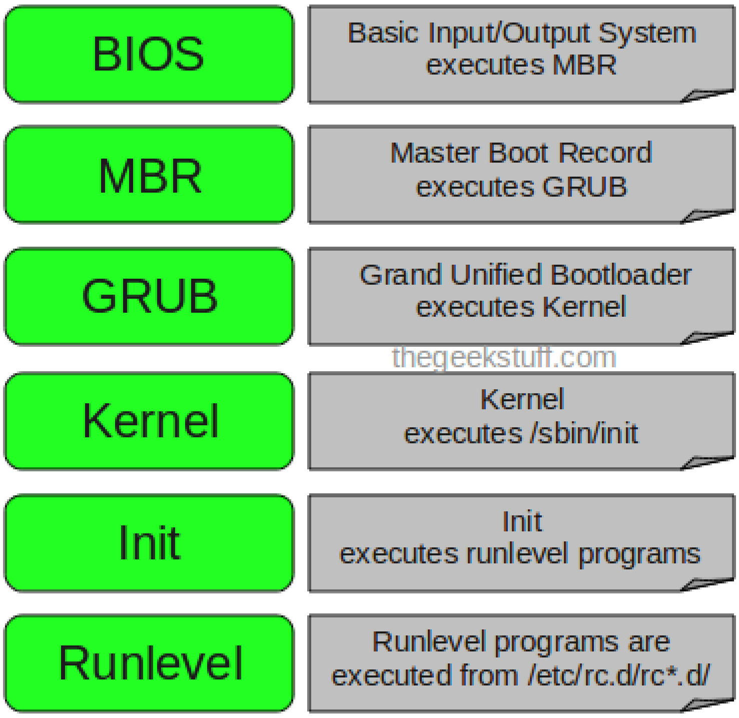
Essential duties of UNIX System administrator:

1. User administration (setup and maintaining account)
2. Maintaining system
3. Verify that peripherals are working properly
4. Quickly arrange repair for hardware in occasion of hardware failure
5. Monitor system performance
6. Create file systems
7. Install, update and remove software
8. Create a backup and recovery policy
9. Computer network setup and Monitor network communication, performance
10. Update system as soon as new version of OS and application software comes out
11. Implement the policies for the use of the computer system and network
12. Setup security policies for users. A sysadmin must have a strong grasp of computer security (e.g. firewalls and intrusion detection systems)
13. Documentation
14. User Password and identity management

Starting and shutdown:

Stages of Linux Boot Process (Startup Sequence)

The following are the 6 high level stages of a typical Linux boot process.



BIOS	Basic Input/Output System executes MBR
MBR	Master Boot Record executes GRUB
GRUB	Grand Unified Bootloader executes Kernel
Kernel	Kernel executes /sbin/init
Init	Init executes runlevel programs
Runlevel	Runlevel programs are executed from /etc/rc.d/rc*.d/

1. BIOS

- BIOS stands for Basic Input/Output System
- Performs some system integrity checks
- Searches, loads, and executes the boot loader program.
- It looks for boot loader in floppy, cd-rom, or hard drive. You can press a key (typically F12 of F2, but it depends on your system) during the BIOS startup to change the boot sequence.
- Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.
- So, in simple terms BIOS loads and executes the MBR boot loader.

2. MBR

- MBR stands for Master Boot Record.
- It is located in the 1st sector of the bootable disk. Typically /dev/hda, or /dev/sda
- MBR is less than 512 bytes in size. This has three components 1) primary boot loader info in 1st 446 bytes 2) partition table info in next 64 bytes 3) mbr validation check in last 2 bytes.
- It contains information about GRUB (or LILO in old systems).
- So, in simple terms MBR loads and executes the GRUB boot loader.

3. GRUB

- GRUB stands for Grand Unified Bootloader.

- If you have multiple kernel images installed on your system, you can choose which one to be executed.
- GRUB displays a splash screen, waits for few seconds, if you don't enter anything, it loads the default kernel image as specified in the grub configuration file.
- GRUB has the knowledge of the filesystem (the older Linux loader LILO didn't understand filesystem).
- Grub configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this). The following is sample grub.conf of CentOS.

```
#boot=/dev/sda

default=0

timeout=5

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

hiddenmenu

title CentOS (2.6.18-194.el5PAE)

          root (hd0,0)

          kernel /boot/vmlinuz-2.6.18-194.el5PAE ro root=LABEL=/

          initrd /boot/initrd-2.6.18-194.el5PAE.img
```

- As you notice from the above info, it contains kernel and initrd image.
- So, in simple terms GRUB just loads and executes Kernel and initrd images.

4. Kernel

- Mounts the root file system as specified in the "root=" in grub.conf
- Kernel executes the /sbin/init program
- Since init was the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1. Do a 'ps -ef | grep init' and check the pid.
- initrd stands for Initial RAM Disk.
- initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

5. Init

- Looks at the /etc/inittab file to decide the Linux run level.

- Following are the available run levels
    - 0 – halt
    - 1 – Single user mode
    - 2 – Multiuser, without NFS
    - 3 – Full multiuser mode
    - 4 – unused
    - 5 – X11 (GUI)
    - 6 – reboot
- Init identifies the default initlevel from /etc/inittab and uses that to load all appropriate program.
- Execute ‘grep initdefault /etc/inittab’ on your system to identify the default run level
- If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.
- Typically you would set the default run level to either 3 or 5.

6. Runlevel programs

- When the Linux system is booting up, you might see various services getting started. For example, it might say “starting sendmail .... OK”. Those are the runlevel programs, executed from the run level directory as defined by your run level.
- Depending on your default init level setting, the system will execute the programs from one of the following directories.
    - Run level 0 – /etc/rc.d/rc0.d/
    - Run level 1 – /etc/rc.d/rc1.d/
    - Run level 2 – /etc/rc.d/rc2.d/
    - Run level 3 – /etc/rc.d/rc3.d/
    - Run level 4 – /etc/rc.d/rc4.d/
    - Run level 5 – /etc/rc.d/rc5.d/
    - Run level 6 – /etc/rc.d/rc6.d/
- Please note that there are also symbolic links available for these directory under /etc directly. So, /etc/rc0.d is linked to /etc/rc.d/rc0.d.
- Under the /etc/rc.d/rc*.d/ directories, you would see programs that start with S and K.
- Programs starts with S are used during startup. S for startup.
- Programs starts with K are used during shutdown. K for kill.
- There are numbers right next to S and K in the program names. Those are the sequence number in which the programs should be started or killed.
- For example, S12syslog is to start the syslog deamon, which has the sequence number of 12. S80sendmail is to start the sendmail daemon, which has the sequence number of 80. So, syslog program will be started before sendmail.

Command Name: shutdown

SHUTDOWN(8) shutdown
SHUTDOWN(8)

NAME

shutdown - Halt, power-off or reboot the machine Starting and shutdown:

SYNOPSIS

shutdown [OPTIONS...] [TIME] [WALL...]

DESCRIPTION

shutdown may be used to halt, power-off or reboot the machine.

The first argument may be a time string (which is usually "now"). Optionally, this may be followed by a wall message to be sent to all logged-in users before going down.

-H, --halt
Halt the machine.

-P, --poweroff
Power-off the machine (the default).

-r, --reboot
Reboot the machine.

When used with no arguments, the `shutdown` command will power off the machine.

```
$ sudo shutdown
```

How to Shutdown the System at a Specified Time ?

The time argument can have two different formats. It can be an absolute time in the format `hh:mm` and relative time in the format `+m` where m is the number of minutes from now.

The following example will schedule system shutdown at 11 A.M:

```
sudo shutdown 11:00
```

The following example will schedule system `shutdown` in 10 minutes from now:

```
sudo shutdown +10
```

How to Shutdown the System Immediately

To shut down your system immediately you can use `+0` or its alias `now`:

```
$ sudo shutdown now
```

User Management Command(useradd,usermod,userdel):

User management is nothing but adding, deleting the users and assigning the passwords for the users in Linux. The same follows with groups. The important thing is this command needs root privilege for accessing other users or groups. Only the same user process can be done without the privilege.

To add a new User and to set password:

To add a new user you can use any two of the following User management commands.

Syntax

useradd < username>

or

adduser < username>

Example to create a user by name user1:

```
[root@linuxhelp ~]# useradd student
```

```
Or:
```

adduser < username>

To set the password for the newly created username:

Syntax

passwd < username>

Example:

```
[root@linuxhelp ~]# passwd student
Changing password for user student.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

To change the password:

To change the old password of the user that is already created use the following User management command.

Syntax:

passwd
The same in root will prompt you to change the root password. The other user' s password can also be changed from the root.

passwd < username>
It will change the specific user' s password and it won’ t ask the old password, since it’ s a root.

Example:

```
[user1@linuxhelp ~]$ passwd
Changing password for user student.
Changing password for student.
(current) UNIX password:
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

```
debasis@LAPTOP-H3N6JCNE:~$ sudo useradd student

debasis@LAPTOP-H3N6JCNE:~$ sudo passwd student

New password:

Retype new password:

passwd: password updated successfully
```

To delete a user account:

To delete an user account, use the following User management command.

Syntax
userdel -r [username]

Example:

[root@linuxhelp ~]# **userdel -r student**

Here, -r option is used to delete user along with the user’ s home directory and mail spool.

Q: Explain /etc/passwd File in UNIX.

Answer:

The full user account information is stored in /etc/passwd file. The entries in the /etc/passwd file has seven fields.

Syntax:
[username]:[x]:[UID]:[GID]:[Comment]:[Home directory]:[Default shell]

Example

```
user1:x:500:500::/home/student:/bin/bash
```

The 1st field indicates the user name

The 2nd field indicates the link to the /etc/shadow file

The 3rd and 4th field specifies the user id and group id of the user.

The 5th field is the comment about user’ s home directory

The 6th field indicates the path of the user’ s home directory

The 7th field indicates the user’ s parent shell

Q: Explain /etc/shadow file:

Answer:

```
user1:$6$n6muZW6t$aBhb40LDQhcjzpMM308ELvJkFE0ZpYZkO2w7oLofEu6YIa.O9lzmxxBkl
tF1Lm8TYdk5zNn6symdmTkdnUbEu0:16856:0:99999:7:::
```

The **1st field** indicates the **user name**
The 2nd field denotes the encrypted password of the user account
The 3rd field indicates last password change i.e., the date at which the user changed the password last time
The 4th field denotes the minimum number of days after which a user can change his password
The 5th field contains the password validity information if the password expires for a user then the user needs to change his password
The 6th field indicates the warning before the password expiration, the number of days for the warning alert before expiration is mentioned in this field
The 7th field denotes the number of days, if the user doesn’ t change the password after the expiration within the mentioned days, the account will be disabled
The 8th field indicates the expiry date of a user account

Group management Command:

Primary group:

To add a user to a Primary group, use the following user mod command as root,

Syntax:

usermod -g [groupname] [username]

Example

```
[root@linuxhelp ~]# usermod -g group1 student
Before adding the user to group
[root@linuxhelp ~]# id student

uid=500(student) gid=500(student) groups=500(student)
After adding the user to group

[root@linuxhelp ~]# id student
uid=500(student) gid=502(student) groups=502(student)
```

Here, If a user is added to a primary group, then the user gets the group id of the group to which it is added.

Adding a group:

To add a group, run the following User management command

Syntax

groupadd [groupname]

Example

```
[root@linuxhelp ~]# groupadd group1
```

OR:

Syntax:
groupadd [Group name]: [Group password]:[GID]:[Group members]

Example

```
group2: x:503:user
```

The 1st field indicates the name of the group
The 2nd field specifies the group password
The 3rd field indicates the group id
The 4th field contains the members of the group

Deleting a group:

To delete a group, use the following User management command

Syntax:

groupdel [groupname]

Example:

```
[root@linuxhelp ~]# groupdel group2
```

```
/etc./group File:
```

User mod commands:

After adding the user you can change user’ s information using the user mod commands

Syntax

usermod [options] [username]

Setting expiry date for a user:

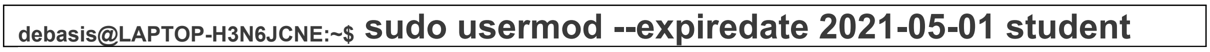
To set expiry date for an user, use the ' --expiredate' flag and mention the date followed by it.

Syntax

usermod --expiredate [date] [username]

Example

```
[root@linuxhelp ~]# usermod --expiredate 2021-02-24 student
```



```
debasis@LAPTOP-H3N6JCNE:~$ sudo usermod --expiredate 2021-05-01 student
```

disk quota:

Filesystem quota is a standard built-in feature found in Linux Kernel. Quotas determine the amount of space a file should have to support user activities.

The disk quotas also limit the number of files a user can create on the system.

Filesystems that support the quota system include xfs, ext2, ext4, and ext3 to mention a few. The assignment of quotas is specific to the filesystem and for each user.

An Introduction to the Linux Terminal

Terminal Emulator

A terminal emulator is a program that allows the use of the terminal in a graphical environment. As most people use an OS with a graphical user interface (GUI) for their day-to-day computer needs, the use of a terminal emulator is a necessity for most Linux server users.

Here are some free, commonly-used terminal emulators by operating system:

- **Mac OS X**: Terminal (default), iTerm 2
- **Windows**: PuTTY
- **Linux**: Terminal, KDE Konsole, XTerm

Each terminal emulator has its own set of features, but all of the listed ones work great and are easy to use.

Q: What is difference between terminal and console?
Answer:
The **console** is typically the primary interface for managing a computer, eg while it is still booting up.
A **terminal** is a session which can receive and send input and output for command-line programs.
The **console** is a special case of these. The shell is a program which is used for controlling and running programs.