

Chapter 5: Unix Administration

Objectives

- System administration
 - The super user mode
 - Creating user accounts
 - Modifying and deleting user accounts
- Booting
 - Init and its run levels
- Partitioning
 - How to view and create partitions
 - The different partitions
 - The different file systems
 - Formatting and labeling a partition
 - How to remove partitions
 - How to resize partitions

3. System Administration

System administration is usually entrusted to a single person-the **system administrator**, also known as the **super user** or **root user**.

He has vast powers, having access to practically everything.

The success and stability of any UNIX installation depends, in great measure, on the effectiveness of the system administrator.

The job of the system administrator involves the management of the entire system-ranging from maintaining security, performing backups and managing the disk space to maintaining user accounts.

3.3 Creating User Accounts

Creating a user account involves setting of the following parameters:

- A user identification number (UID) and username

- A group identification number (GUID) and group name

- The home directory

- The login shell

- The mailbox in /var/mail

- The password.

Before creating a user account we first need to create the group

```
# groupadd -g 555 user15
```

Apart from **groupadd**, you can also use the **groupmod** and **groupdel** commands to modify and delete groups.

3.1 useradd: Adding a user

The `useradd` command adds new users to the system.

All the parameters related to the user have to be provided in the command line itself:

```
#useradd -u 567 -g user15 -c user20 -d  
/home/user20 -m user20
```

This quietly creates the user `user20` with a UID of 567 and group name `user15`.

The home directory is `/home/user20`.

The line `useradd` creates in `/etc/passwd` is shown in Fig 3.1

You now have to set the password for user20 with the command **passwd user20**.

Once all this had been done, the oracle user account is ready for use.

3.2 /etc/passwd and /etc/shadow: User Profiles

All user information except the password encryption is now stored in /etc/passwd.

It contained the password once, the reason why it continues to be known by that name.

The encryption itself is stored in /etc/shadow.

This is now the control file used by passwd to ascertain the legitimacy of the user's password.

Let us take the line pertaining to user2.

There are seven fields here, and their significance is noted below (in the order they appear in /etc/passwd)

Diagram illustrating the fields of the `user2` entry in `/etc/passwd`:

```
user2:x:503:502:user2:/home/user2:/bin/bash
```

Fields and their descriptions:

- Password Field (Not used) → `x`
- Numeric GUID → `503`
- Home Directory → `/home/user2`
- Username → `user2`
- Numeric UID → `502`
- GCOS Field → `user2`
- Login Shell → `/bin/bash`

Username:- The name you use to log on to the UNIX System (user2)

Password: No longer stored the password encryption but contains x

UID-The user's numerical identification (503). No two users have the same UID

GUID – The user's numerical group identification (502). This number is also the third field in /etc/group. Command like `ls` have to read this file to print the group name.

Comment or GECOS – User details, e.g. her name, address and so forth (user2). The name is used at the front of the email address. Any mail sent from this user account will show the sender as “user2.” `--user2@planets.com>`- assuming the user belongs to the domain shown.

Home Directory – The directory where the users end up on logging in (home/user2)

Login Shell-The first program executed after logging in. This is usually the shell (`/bin/bash`)

For every line in `etc/passwd`, there's a corresponding entry in `/etc/shadow`.

The relevant line in this file could look something like this:

```
user2:$1$NLzYyxqp$DASRGVljG0FOlsQKzJTg00:17135:0:9999:7:::
```

The password encryption is shown in the second field.

However, an intelligent hacker can use an encryption algorithm to generate a sequence of encrypted patterns.

Such algorithms are also widely available on the Internet.

It is quite possible that she might just find a match, so this file must be unreadable to all but the super user.

3.3 usermod and userdel: Modifying and Removing Users

usermod is used for modifying some of the parameters set with useradd.

Users sometimes need to change their login shell, or home directory, and the following command line sets /home/oracle as the home directory for user20

```
# usermod -d /home/oracle user20
```

Now if you use cat command to view /etc/passwd, you can see the home directory of user2 has been changed.

Users are removed from the system with **userdel**.

The following command removes the user user2 from the system.

This removes all entries pertaining to oracle from /etc/passwd, /etc/group and /etc/shadow.

The user's home directory doesn't get deleted in the process and has to be removed if required.

3.5 Booting

The startup procedures are controlled by automated shell scripts which are changed quite frequently. Yet the administrator needs to know the exact of steps the system follows during the this event
Things go wrong, especially during startup, she must be able to fix them.

3.5.1 init: The prime Mover

There are several processes initiated at system startup.

The kernel (/stand/unix, /kernel/genuix or /vmlinuz) is loaded into memory, and it then starts spawning further processes.

The most important of these is **/sbin/init** with PID 1, which is responsible for the birth of all subsequent processes.

You must know the behavioral pattern of init for two vital reasons:

It controls the run levels (the system states) and decides which processes to run (and Kill) for each run level

It spawns a **getty** process at every terminal or modern port so that users can log in

Init also makes sure all system daemons are running.

lpshed monitors the line printer spooler for jobs that have been queued for printing.

cron is the the system's chronograph.

httpd is the Web server daemon and **sendmail** looks for all incoming and outgoing mail.

init remains the parent and sometimes the grandparent of all the above processes.

3.5.2 **init Run Levels**

Init is responsible for keeping the system in different states-which we call **run levels**.

Each run level here is normally a single digit (0 to 6), or an s or S

A different process is scheduled to run at each of these states.

Normally, the system would be in any one these run levels.

0 – System shutdown

1 – System administration mode (local file systems mounted.)

2 – Multiuser mode without networking capabilities

3 – Multiuser mode with networking capabilities

6 – Shutdown and reboot mode

S or S – Single-user mode (file systems mounted).

We will not consider run levels 4 and 5 as they are either unused or used in situations which won't interest us.

When the system is booted, **init** first enters the run level 1—the system administration mode

The single-user mode is used by the administrator to perform her administrative tasks – like taking offline backup.

The role of run level 1 is system-variant; on some systems, 1 and S have identical roles.

Depending on your system, the normal multiuser mode is implemented in one or the two run levels-2 or 3.

You can change your run level by using the `init` command with the run level as argument.

But first to know the run level you are on

who -r

Then,

init 3

This will change to multiuser mode with networking. To change back, just be the super user and type `init 5`.

Partitions

- To see the partition table, log in as root using `su` command and the root's/your password. And type in the following

```
#parted /dev/sda
```

Then type in
print

in order to see the different partitions. You can see that, if you are using a red hat linux loaded on a virtual machines, it has already created the partitions for you-
1: /dev/sda1 2: /dev/sda2 3: /dev/sda3

Swap partitions are used to support virtual memory. In other words, data is written to a swap partition when there is not enough RAM to store the data your system is

- To make a partition, use the following command

mkpart primary ext3 1024 2048

The above is an example. You can make the partition extended-that is the second option.

Primary partitions can be used to load the operating system, while extended partitions are used for other types of files.

•File Systems

- ext3** — The ext3 file system is based on the ext2 file system and has one main advantage — journaling. Using a journaling file system reduces time spent recovering a file system after a crash as there is no need to fsck the file system. A maximum file system size of 16TB is supported for ext3.

- ext2** — An ext2 file system supports standard Unix file types (regular files, directories, symbolic links, etc). It provides the ability to assign long file names, up to 255 characters.

- xfs** — XFS is a highly scalable, high-performance file system that supports filesystems up to 16 exabytes (approximately 16 million terabytes), files up to 8 exabytes (approximately 8 million terabytes) and directory structures containing tens of millions of entries. XFS supports metadata journaling, which

File systems

- **ext4** — The ext4 file system is based on the ext3 file system and features a number of improvements.
- These include support for larger file systems and larger files, faster and more efficient allocation of disk space, no limit on the number of subdirectories within a directory, faster file system checking, and more robust journaling.
- A maximum file system size of 16TB is supported for ext4. The ext4 file system is selected by default and is highly recommended.

Note

- **vfat** — The VFAT file system is a Linux file system that is compatible with Microsoft Windows long filenames
- **Btrfs** — Btrfs is under development as a file system capable of addressing and managing more files, larger files, and larger volumes than the ext2, ext3, and ext4 file systems. Btrfs is designed to make the file system tolerant of errors, and to facilitate the detection and repair of errors when they occur. It uses checksums to ensure the validity of data and metadata, and maintains snapshots of the file system that can be used for backup or repair.

Because Btrfs is still experimental and under development, the installation program does not offer it by default. If you want to create a Btrfs partition on a drive, you must commence the installation process with the boot option `btrfs`.

Warning

Red Hat Enterprise Linux 6.9 includes Btrfs as a technology preview to allow you to experiment with this file system. You should not choose Btrfs for partitions that will contain valuable data or that are essential for the operation of important systems

Formatting and Labeling the Partition

To format and label the partition use the following procedure:

Procedure - Format and label the partition

1.The partition still does not have a file system. To create one use the following command:

```
# /sbin/mkfs -t ext3 /dev/sda6
```

Warning

Formatting the partition permanently destroys any data that currently exists on the partition.

2.Next, give the file system on the partition a label. For example, if the file system on the new partition is /dev/sda6 and you want to label it /work, use:

```
# e2label /dev/sda6 /work
```

By default, the installation program uses the mount point of the partition as the label to make sure the label is unique. You can use any label you want.

Afterwards, create a mount point (e.g. /work) as root.

Add to /etc/fstab

As root, edit the `/etc/fstab` file to include the new partition using the partition's UUID. Use the command `blkid -o list` for a complete list of the partition's UUID, or `blkid device` for individual device details.

The first column should contain `UUID=` followed by the file system's UUID. The second column should contain the mount point for the new partition, and the next column should be the file system type (for example, `ext3` or `swap`). If you need more information about the format, read the man page with the command `man fstab`.

If the fourth column is the word `defaults`, the partition is mounted at boot time. To mount the partition without rebooting, as root, type the command:
`mount /work`

Removing the partition

To remove a partition use the the rm command with the partition number. For example, in order to remove partition 3,

```
#rm 3
```

Resizing the partition

To resize a partition, for example partition 3, use the following command

```
#resize 3 1024 2048
```

After resizing the partition, use the print command to confirm that the partition has been resized correctly, is the correct partition type, and is the correct file system type.

After rebooting the system into normal mode, use the command df to make sure the partition was mounted and