**Installation procedure of Linux**

**Preparing to Install Linux**

After you have obtained a distribution of Linux, you’re ready to prepare your system for installation. This takes a certain degree of planning, especially if you’re already running other operating systems. In the following sections we’ll describe how to plan for the Linux installation.

**Installation overview**: - in general the method used to install the software is as follows:

1) **Repartition your hard drive(s)**

. If you have other operating systems already installed, you will need to repartition the drives in order to allocate space for Linux.

2) **Boot the Linux installation media.**

Each distribution of Linux has some kind of installation media— usually a “boot floppy”—which is used to install the software. Booting this media will either present you with some kind of installation program, which will step you through the Linux installation, or allow you to install the software by hand.

3) **Create Linux partitions**

After repartitioning to allocate space for Linux, you create Linux partitions on that empty space. This is accomplished with the Linux fdisk program.

4) **Create file systems and swap space.**

At this point, you will create one or more file systems, used to store files, on the newly-created partitions. In addition, if you plan to use swap space, you will create the swap space on one of your Linux partitions.

5**) Install the software on the new file systems**.

Finally, you will install the Linux software on your newly-created file systems.

**Hardware Requirements**

Red Hat Enterprise Linux versions on both 32-bit and 64-bit platforms have the same system requirements.

Operating System:- Red Hat Enterprise Linux 4 or 5 with the latest patches and

upgrades.

CPU Type Pentium 4 or higher; 2 GHz or higher

Memory/RAM 1 GB minimum, up to the system limit.

Hard Disk 4 GB minimum

Other: - To run the Directory Server using port numbers less than 1024,

such as the default port 389, you must setup and start the Directory

Server as **root**, but it is not necessary to run the Directory Server

as **root**.

A minimum screen resolution of 800 by 600 pixels is required to display the Tivoli Storage Manager Backup-Archive Client Java GUI and Web GUI.

**Software requirements:-**

|  |  |
| --- | --- |
| Type of software | Minimum software requirements |
| Software | * Red Hat Enterprise Linux 5.0 * SUSE Linux Enterprise Server 10 and 11 (including Novell OES distributions)   For SuSE Linux Enterprise Server 10 or 11 64-bit support, you need to install the **glibc-locale-32bit** package if you run the client on a locale other than en\_US or EN\_US. |
| Linux x86/x86\_64 GPFS™ support: | GPFS 3.2, 3.2.1, and 3.3 |
| Applications | Java JRE 5 or 6 |
| Web browser | A Firefox 3.0.14 or higher browser for the Web client and to access online help and documentation |

**Create partitions:-**

In most Linux systems, you can use the fdisk utility to create a new partition and to do other disk management operations.

As a tool with a text interface, fdisk requires typing the commands on the fdisk command line. The following fdisk commands may be helpful, for creating partitions.

|  |  |
| --- | --- |
| **Options** | **Description** |
| m | Displays the available commands. |
| p | Displays the list of existing partitions on your hda drive. Unpartitioned space is not listed. |
| n | Creates a new partition. |
| q | Exits fdisk without saving your changes. |
| l | Lists partition types. |
| w | Writes changes to the partition table. |

**To create a new partition on Linux**

1. Start a terminal.
2. Start fdisk using the following command**: “/sbin/fdisk /dev/had”**

Where, /dev/hda stands for the hard drive that you want to partition.

1. In fdisk, to create a new partition, type the following command: **“n”**

-When prompted to specify the **Partition type**, type p to create a primary partition or e to create an extended one. There may be up to four primary partitions. If you want to create more than four partitions, make the last partition extended, and it will be a container for other logical partitions.

-When prompted for the **Number**, in most cases, type 3 because a *typical* Linux virtual machine has two partitions by default.

-When prompted for the **Start cylinder**, type a starting cylinder number or press **Return** to use the first cylinder available.

-When prompted for the **Last cylinder**, press **Return** to allocate all the available space or specify the size of a new partition in cylinders if you do not want to use all the available space.

By default, fdisk creates a partition with a **System ID** of 83. If you're unsure of the partition's **System ID**, use the “1”, command to check it.

1. Use the **“w”**, command to write the changes to the partition table.
2. Restart the virtual machine by entering the **“reboot”**, command.
3. When restarted, create a file system on the new partition. We recommend that you use the same file system as on the other partitions. In most cases it will be either the Ext3 or ReiserFS file system. For example, to create the Ext3 file system, enter the following command**: “/sbin/mkfs -t ext3 /dev/hda3”**
4. Create a directory that will be a mount point for the new partition. For example, to name it data, enter: **“mkdir /data”**
5. Mount the new partition to the directory you have just created by using the following command: **“mount /dev/hda3 /data”**
6. Make changes in your static file system information by editing the /etc/fstab file in any of the available text editors. For example, add the following string to this file:

**“/dev/hda3 /data ext3 defaults 0 0”**

In this string /dev/hda3 is the partition you have just created, /data is a mount point for the new partition, Ext3 is the file type of the new partition. For the exact meaning of other items in this string, consult the Linux documentation for the mount and fstab commands.

1. Save the **“/etc/fstab file .**

**Configuration of X system**

The X Window System (sometimes referred to as "X" or as "X Windows") is an [open](http://searchcio-midmarket.techtarget.com/definition/open), cross-[platform](http://searchservervirtualization.techtarget.com/definition/platform), [client/server](http://searchnetworking.techtarget.com/definition/client-server) system for managing a windowed graphical user interface in a [distributed](http://searchcio-midmarket.techtarget.com/definition/distributed) network. In general, such systems are known as [windowing system](http://searchwinit.techtarget.com/definition/windowing-system)s. In X Window, the client-server relationship is reversed from the usual. Remote computers contain applications that make client requests for display management services in each PC or workstation. X Window is primarily used in networks of interconnected [mainframe](http://searchdatacenter.techtarget.com/definition/mainframe)s, [minicomputer](http://search400.techtarget.com/definition/minicomputer)s, and [workstation](http://searchmobilecomputing.techtarget.com/definition/workstation)s. It is also used on the [X terminal](http://searchnetworking.techtarget.com/definition/X-terminal), which is essentially a workstation with display management capabilities but without its own applications. (The X terminal can be seen as a predecessor of the network PC or [thin client](http://searchnetworking.techtarget.com/definition/thin-client) computer.)

The X Window System was the result of research efforts in the early 1980s at Stanford University and MIT, aided by IBM, to develop a platform-independent graphics protocol. The X Window System is an open standard that is managed by the X.Org consortium. Although Microsoft has its own platform-dependent windowing system (an integral part of the Windows 95/98/NT operating systems), there are vendor-supplied X Windows products that can be installed to run on these systems.

**Users & Groups Management:-**

User is used to log in and work on a system. A group is created and then the name is assigned by user names or may be different. The group name is assigned to directories and files. Each directory and file has a group name. A group can be specified permissions for directories and files. The users that are members of a group acquire permissions that the group has been assigned to a directory or file. This is a method of easily assigning permissions to multiple users for directories and files.

*A user can be a member of more than one group.*

*A group cannot have another group as a member.*

There are three types of accounts on a Linux system:

1. **Root account:** This is also called superuser and would have complete and unfettered control of the system. A superuser can run any commands without any restriction. This user should be assumed as a system administrator.
2. **System accounts:** System accounts are those needed for the operation of system-specific components for example mail accounts and the sshd accounts. These accounts are usually needed for some specific function on your system, and any modifications to them could adversely affect the system.
3. **User accounts:** User accounts provide interactive access to the system for users and groups of users. General users are typically assigned to these accounts and usually have limited access to critical system files and directories.

Linux supports a concept of Group Account which logically groups a number of accounts. Every account would be a part of any group account. Linux group’s plays important role in handling file permissions and process management.

**Create Users:-**

A ‘**useradd**‘ command is a low-level utility that is used for adding/creating

user accounts in **Linux.**

When we run ‘useradd‘ command in Linux terminal, it performs following major things:

1. It edits /etc/passwd, /etc/shadow, /etc/group and /etc/gshadow files for the newly

created User account.

2. Creates and populate a home directory for the new user.

3. Sets permissions and ownerships to home directory.

The **‘adduser’** is much similar to ‘**useradd’** command. Username must be unique. When we add a new user in Linux with ‘**useradd**‘command it gets created in locked state and to unlock that user account, we need to set a password for that account with ‘**passwd**’ command.

**useradd or adduser Command**

Create a new user account (you must be login root user).

***Syntax: # useradd [user\_name]***

The user home directory is /home/user\_name.

**Example :** # useradd roman

# passwd roman

Changing password for user roman

New LINUX password:

Retype new LINUX password:

Passwd: all authentication tokens updated successfully.

Once a new user created, its entry automatically added to the

‘**/etc/passwd**’ file. The file is used to store user’s information.

**usermod command**

To change the attributes of an existing user such as, change user’s home directory, login name, login shell, password expiry date. Usermod command operate on

 We must have existing user accounts to execute usermod command.

 Only superuser (root) is allowed to execute usermod command.

When we execute ‘usermod’ command in terminal, the following files are used and affected.

1. **/etc/passwd –** User account information.

2. **/etc/shadow –** Secure account information.

3. **/etc/group –** Group account information.

4. **/etc/gshadow –** Secure group account information.

5. **/etc/login.defs –** Shadow password suite configuration.

***Syntax: # usermod options [user\_name]***

**Option Description Example**

-c Adding Information to # usermod –c "This is roman" roman

User Account Add information on ‘**roman’**

**Passwd Command**

Change the password on your current account. If you are login by root user,

you can change the password for any user using a passwd command.

***Syntax: # passwd [ user\_name ]***

Every time you try to log in to a computer system, you are asked for your user ID

and password. Although users can be set up without passwords, most users will

have a password, which they will use when logging in to a computer system.

When you are typing the password (both old and new), Linux does not display

them. Whenever you change your password, the new password must follow some of the rules that govern passwords are as follows:

 The minimum number of alphabetic characters.

 The length should be at least 7 characters.

 The maximum number of times a single character can be used in a password.

 Avoid using strings of characters followed by numbers, like xyz01.

 Use both uppercase and lowercase letters mixed with numbers

***Example: # passwd ECS***

***New password :***

***Confirm password:***

**Userdel Command**

Remove an account (you must be a root). The user's home directory and the

undelivered mail must be dealt with separately (manually because you have to decide what to do with the files).

***Syntax: # userdel [user\_name]***

***Example:*** # userdel ecs

**[-r ] -** To remove the user, their home directory and their files

***Example:*** # userdel -r ecs.

**Create Group :**

**Group command** is used to add a new group to the system. The groups are

required for easy maintenance of users. Multiple users can be part of a group and a

user can be part of more than one group. So there is many to many mapping of users to groups. **/etc/group -** Defines the groups on the system. There are two types of groups under Linux operating systems:

 Primary user group.

 Secondary or supplementary user group.

***Groupadd Command***

The groupadd command creates a new group account using the values specified on the command line and the default values from the system. The new group will be entered into the system files as needed.

***Syntax : # groupadd options [ group\_name ]***

***Example:*** # groupadd Sangola

**Option Description Example**

-r create a system account # groupadd –r javaproject

-f If we try to add some # groupadd javaproject

already existing group, groupadd: group javaproject exists

Forcefully create a same # groupadd –f javaproject

name group.

***Groupmod Command***

The groupmod command modifies the definition of the specified GROUP

***Syntax : # groupmod options [ group\_name ]***

**Option Description Example**

-n The name of the group # groupmod –n bettergroup newgroup

will be changed from Change the group “newgroup” to “bettergroup”.

GROUP to NEW\_GROUP.

**Groupdel Command**

The groupdel command is used to delete(remove) a group and modifies the system

account files, deleting all entries that refer to group. The named group must exist.

You must manually check all filesystems to insure that no files remain with the named group as the file group ID.

***Syntax : # groupdel [ group\_name ]***

***Example:*** # groupdel Sangola

**Assign Permissions to users & group:-**

**File and Directory permissions ☹(*File Security)***

Permissions are important for keeping your data safe and secure. Utilizing

permission settings in Linux can benefit you and those you want to give access to your files and you don’t need to open up everything just to share one file or directory. You can group individual users together and change permissions on folders (called directories in Linux) and files and you don’t have to be in the same workgroup or be part of a domain for them to access those files. You can change permissions on one file and share that out to a single group or multiple groups. Fine grained security over your files places you in the driver seat in control of your own data.

Password security protects a system from access by an unauthorized person. In a multi-user system, each authorized user can have access to files that belong to other users or to the system files and utilities. In such an environment, it becomes essential that files should be protected so that another user does not misuse one user’s files and system files are not detected or modified.

Linux allows users to protect their own files from other users working on the

system. It also provides protection for system files. Method of protecting their own files from other users working on the system. It also provides protection for system files.

The types of files in Linux.

The concept of File Owner, Group Owner and Other Users.

Files access permissions.

**Chmod [ Changing Mode of Files & Directories ]**

It is useful to provide file security. This is used for change the access

permission (mode) of one or more files. Only the owner of a file or a privileged user

may change the mode. Mode can be numeric or an expression in the form of who

opcode permission multiple modes are separated by commas.

***Syntax: $ chmod [operation or category] permission filename***

**Who:**

**u :** for changing for File Owner only. Example: chmod u+r x.c

**g :** for changing Group Owner only. Example: chmod g-r x.c

**o :** for changing for all Other Users only. Example: chmod o+x x.c

**a** : all (default).





**Permission Symbol : +** : Add permission

**-** : Remove permission

**=** : Assign permission

**Permission : r** – read

**w** – write

**x** – execute

**s** – set user ID

**u** – user’s parent permission

**g** – group’s parent permission

**o** - other’s parent permission

Alternatively, the ls –l command displays file permission in three column. it specify by a three types permission i.e read(r), write(w), execute(x).The first column designates owner permission. The second column group permission and the third column other’s permission. Permissions are calculated by adding the fallowing values. **4 - Read, 2 – write 1 - execute**

***Example:***

$ chmod u+x bca

This command assign (+) executable permission for owner or user (u).

$ chmod ugo+x bca

The ‘ugo’ combines all three categories user, group,others to assign execute

Permission.

$ chmod go-rw bca

The removes(-) a read & write permission of group & others.

$ chmod u=x , go=rw file

This removes all existing permission and replaces them with execute permission for

user & read, write permission for group & others.

$ chmod 777 mydata

This is all permission set to a mydata file pervious all permission removes. the use

the numeric permission can be added to mydata.

$ chmod g+rx-w home/sangola/mca

The set permission to a mca directory, the read & execute permission for the group for mca directory but removes the write permission.

***Chown [ Change the Owner ]***

This command used for change the owner of the files. When other users may be

access the files, only the owner can change its permission. But you can change the

owner of the file from yourself to the other user. The chown commands transfers

control over a file to another user. And change the owner of a file .full control to the

new owner.

***Syntax : $ chown new owner name filename***

***Example :*** $ chown comp mydata

d- rw -r- -r-- 1 **comp** group 302 step 4 09:45 mydata

Change the owner name of the file mydata. **comp** is a owner name of the mydata.

***Chgrp [ Change the Group ]***

This command used for change the group owner of the files. A user can change

the group owner of a file, but only to a group to which she also belongs.

***Syntax : $ chgrp new group name filename***

***Example :*** *$ chown* ***ECS*** *mydata*

d- rw -r- -r-- 1 comp **ECS** 302 step 4 09:45 mydata

Change the group name of the file mydata. **ECS** is a group name of the mydata.

**Difference between chown and chgrp**

1) **chown** command is used to change ownership as well as group name associated to different one, where as **chgrp** can change only group associated to it.

2) Many people say that regular user only able to use **chgrp** to change the group if the user belongs to them. But it’s not true a user can use **chown** and **chgrp** irrespective to change group to one of their group because chown is located in /bin folder so every can use it with some limited access.