***ABOUT AUTHOR***

*MY NAME IS ANKAM RAVI KUMAR HAVING MORE THAN 6YEARS OF SOLID INDUSTRY EXPERIENCE IN INFRASTRUCTURE MANAGEMENT / CUSTOMER SUPPORT / OPERATIONS AND TRAINING DOMAIN.*

***DEEP FUNCTIONAL AND MANAGEMENT KNOWLEDGE ACROSS THE FOLLOWING IT DOMAINS***

* *OPERATING SYSTEM MANAGEMENT SUCH HAS LINUX DIFFERENT FLAVORS, AIX, SOLARIS AND WINDOWS*
* *ENTERPRISE SERVER MANAGEMENT*
* *INSTALLING AND CONFIGURING BLADE SERVERS*
* *CORE STORAGE MANAGEMENT (OVERLAND STORAGE, EMC, IBM AND NETAPP)*
* *DATABASE MANAGEMENT (MS SQL, POSTGRESQL AND MYSQL)*
* *PROCESS MANAGEMENT (ITIL)*
* *VIRTUALIZATION MANAGEMENT (VSPHERE, VMWARE, KVM, HYPER-V AND XEN)*
* *BACKUP AND RECOVERY MANAGEMENT (NET VAULT, COMMVAULT AND SYMANTEC BACKUP EXEC)*
* *APPLICATION SERVER MANAGEMENT AND STORAGE CLUSTER MANAGEMENT*
* *DATA CENTER MANAGEMENT AND HOSTING SOLUTIONS*
* *SERVER MANAGEMENT*
* *PROGRAMMING LANGUAGES SUCH AS PHP AND HTML*
* *SCRIPTING LANGUAGES SHELL, PERL AND PYTHON*
* *ASSET MANAGEMENT AND PROCUREMENT. DESIGNED, PLANNED AND IMPLEMENTED THE*

*DATA CENTER, SERVER MONITORING AND SAN ENVIRONMENTS.*

*I AM SPECIALIZED IN MANAGING AND BUILDING THE TEAMS FOR IT SERVICES DELIVERY AND SERVICE SUPPORT, TRAINING AND OPERATIONS IN BOTH SMALLER AND LARGER COMPANIES. RICH EXPERIENCE AND STRONG EXPOSURE IN IT INFRASTRUCTURE & DATA CENTER MANAGEMENT.*

*IMPLEMENTATION OF MONITORING SOLUTIONS FOR SMALL. MEDIUM AND ENTERPRISE COMPANIES, USING TOOLS NAGIOS, NAGIOSXI, CACTI, SOLARWINDS AND OP MANAGER.*

*INDEX*

*1. INTRODUCTION………………………………………………………………………..4-4*

*2. WINDOWS VS LINUX………………………………………………………………….5-5*

*3. LINUX ARCHITECTURE………………………………………………………………6-6*

*4. BASIC AND COMMON COMMANDS………………………………………………….7-13*

*5. OS INSTALLATION…………………………………………………………………...14-34*

*6. BOOTING PROCESS…………………………………………………………………..35-36*

*7. BASH FEATURES……………………………………………………………………..37-40*

*8. LINUX DIRECTORY STRUCTURE……………………………………………………41-41*

*9. TEXT EDITORS………………………………………………………………………..42-44*

*10.USER ADMINISTRATION……………………………………………………………..45-47*

*11.PROFILE MANAGEMENT……………………………………………………………..48-50*

*12.STRING RELATED COMMANDS…………………………….………………………..51-54*

*13.FILE LINKS…………………………………………………………………………..55-55*

*14.PROCESS MANAGEMENT………………………………………………….………..56-57*

1. *SEARCH RELATED COMMANDS……………………………………………………..58-58*
2. *TASK AUTOMATION AND TASK SCHEDULING……………………………………..59-60*

*17.FILE SYSTEM………………………………………………………………………….61-70*

*18.LOGICAL VOLUME MANAGER………………………………………………………..71-77*

*19.RAID REDUNDANT ARRAY OF INDEPENDENT DISK………………………………..78-86*

*20.REDHAT PACKAGE MANAGER………………………………………………………..87-91*

*21.NETWORKING…………………………………………………………………………..92-96*

1. *KICK START INSTALLATION AND CONFIGURATION……………………………..97-100*

*23.QUOTA MANAGEMENT…………………………………………………………..…..101-103*

*24. REMOTE MANAGEMENT………………………………………………………..…..104-107*

*25. NETWORK FILE SYSTEM (NFS) ……………………………………………….…..108-110*

**1. Introduction**

UNIX is the first Operating system in the world, developed by Kem Thompson and Dennis Ritchie in 1969 at Bell Lab by AT&T Company

**IBM : AIX**

**SGI : IRIX**

**Sun : Solaris**

Free software foundation organization, they start a project by name GNU. The main aim of this project is to develop such an operating system that can run on any platform.

In 1991, a student Linuz Torvalds developed a kernel named Linux’s kernel plus GNU application called Linux operating system.

Linux is an open source technology.

Different companies that provide Linux in Market are Redhat, SuSe, Scientific, Centos, and Knoppix etc.

##### Features:

* + Linux is the fastest Operating system in the world. It runs 2 to 3 times fast than windows OS.
  + Linux is the much secured OS because there is no any problem of virus.
  + Linux file format is text format and windows file format is binary format.
  + Linux is very reliable OS because kernel of Linux is very stable as compare to windows kernel not crashed easily.
  + Kernel of Linux is very small in size it can be stored in floppy.
  + Linux uses the x-Window system which is advanced network windowing system. Using this system we can display output of any workstation monitor attached in the network.

##### Advantages:

* + Virus Proof Crash Prof Economical
  + Multiuser, Multi-Tasking and Multi processing capacity

##### Login Modes:

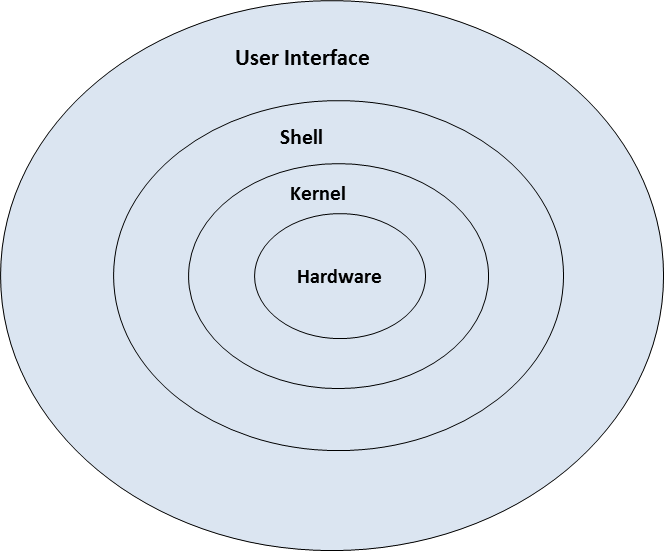
**Two modes:** 1.Text mode (CLI) 2.Graphical Mode (GUI)

Login to Text mode we have use (Ctrl+Alt+F1…..F6, F8….F12) (Ctrl+Alt+F7) for Graphical Mode

**2. Windows Vs Linux**

|  |  |
| --- | --- |
| **Windows** | **Linux** |
| * It is a proprietary software everything need to buy | * It is an open source software everything is free |
| * Less Secure | * More Secure |
| * More costly | * Less Cost compare to windows |
| * Chance to get a carpal tunnel syndrome | * There is no chance to get carpal tunnel syndrome |
| * Developed by Microsoft corporation | * Developed by Linus Torvalds |
| * Options need to select | * You can develop anything as per your requirement |
| * Kernel is not editable | * Kernel is editable |

##### 3. Linux Architecture:



***Kernel:*** It is a program. Kernel is a Core component of operating system, interacts directly with hardware and provides low level services to upper layer components.

***Shell:*** An interface to kernel, hiding complexity of kernel’s functions from users. Takes commands from user and executes kernel’s functions.

***User Interface:*** In information technology, the user interface (UI) is everything designed into an information device with which a human being may interact. Two types CLI (Command line interface) and GUI (Graphical User Interface).

***System Library*** - System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features. These libraries implements most of the functionalities of the operating system and do not requires kernel module's code access rights.

***System Utility*** - System Utility programs are responsible to do specialized, individual level tasks.

**4. Basic & Common Commands**

* All UNIX commands must always be entered in small case letters
* Between the command name and the options that may be available with the command there must always be a space or a tab, for example, ls –l. Here is the command whereas –l is the option and the two have been separated by space. The option is usually preceded by a minus (-) sign. The option available with a command are often known as swithes.
* Two or more options available with command can usually be combined, for example, the command ls –l –a is same as ls –la.
* If you make a typing mistake, press backspace to erase characters Don’t try back using arrow keys and then attempt deleting using the del key.
* To cancel the entire command before you press Enter, press ctrl+c Or del key.

##### Basic Commands:

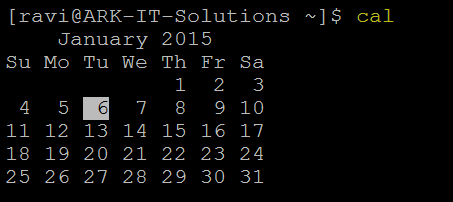
$ date #To see the date of the system.

$ date +%d-%Y-%H-%M #To see particular date format you always use date options



$ cal #To see the current month calendar.

$ cal 11 2014 #To display the calendar, Month November (11) and year 2014



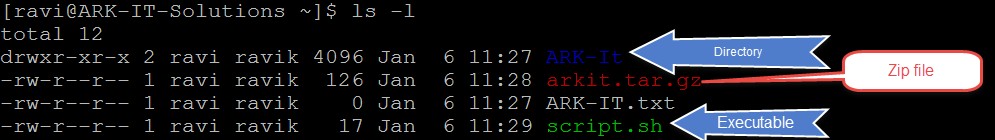
$ clear #To clear the screen

$ ls –l #List files, directories with their properties

$ ls #To list files and directories

$ ls –a #To list all hidden files and directories

$ ls –d #To list only directories



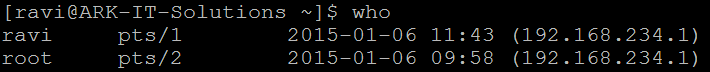
$ pwd #Print working directory

$ who am I #To see from which user you have logged in



$ who #To see all who is logged in yet this point of time from which IP

$ w #More details about user related info



$ uptime #To see the server uptime, boot time, users and load.



$ uname –a #Verify Operating system version, kernel version and architecture



$ touch <File Name> #Create an empty file / Multiple empty files yet a time

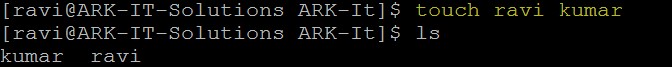
#### Options:

$ touch –am #it will change a file time to current time

$ touch –r file1 –B 30 file2 #it will create two files with 30 seconds time difference

$ cat /dev/null > file #To empty the data file

$ cat > <File Name> #Create an single file with text



$ rm –rf <File / Directory Name> #Delete files and directories forcefully

$rmdir <directory> #Delete directories only

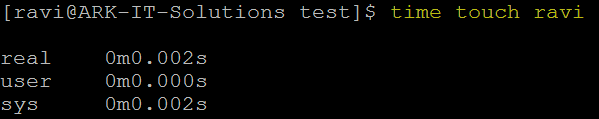
$ mkdir <Directory Name> #Create an empty directory / directories

$ mkdir –p <directory/directory/directory> #to create parent directories

$ cd <Path of the directory> #Change directory

$ cat <File Name> #View content of file

$ time #Calculate response time of the activity / command



$ hwclock $to see detailed date and time with time zone



$ cp <Source path> <Destination path> #Copy the files from one path to another path

$cp –Rv <source> <destination> #copy directories from source to destination

#### Options:

-R, -r, #copy directories recursively

-v #verbose to see progress of copy job

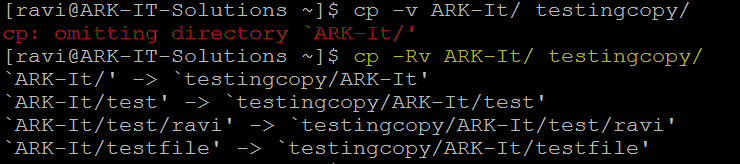
-p #preserve

-f #forcefully

-I #interactive: Ask before overwriting file



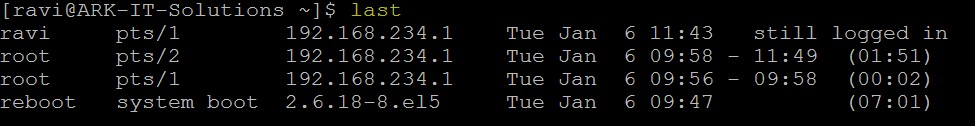
Copying directory must use –R to copy directories



$ mv <source> <destination> #Move files/directories

$ mv <old name> <new name> #Rename the file and directory

$ last #Check who logged in and when logged in duration



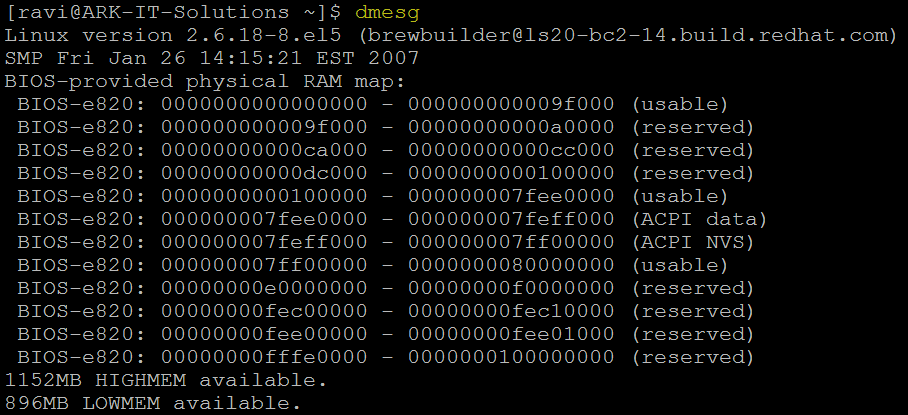
$ arch #to know architecture



$ reboot / init 6 #Restart server

$poweroff / init 0 #To shut down the server

$ dmesg #Check boot process logs

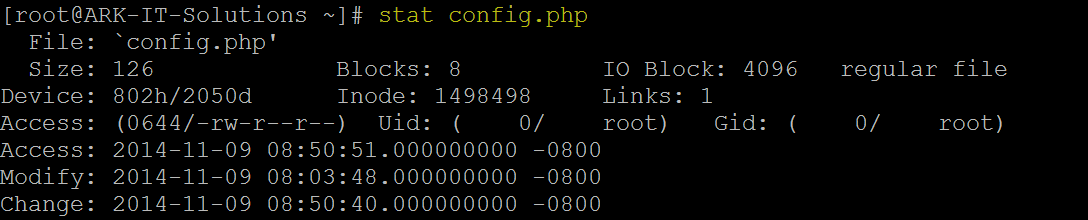


$ nsloookup <Server Address> #check dns resolution

$ dig <server address> #check dns resolution to debug

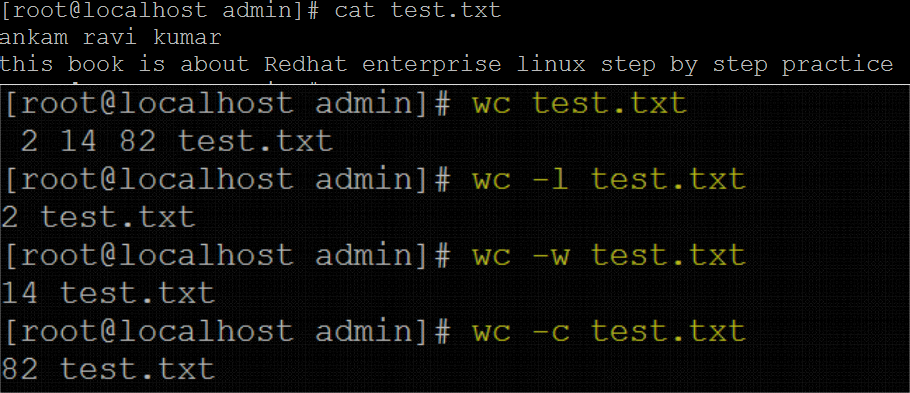
$ tree <directory> #it will show the tree of parent directory

$ stat <file name> #detailed information about file



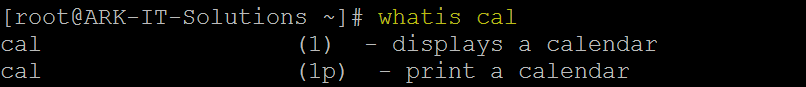
$ wc #word count, character count and line count

-l #Check line count -c #Character Coun -w #Word Count



#### Help relate commands:

$ whatis <Command Name> #It will display single line description about command



$ whereis <Command Name> #It will provide you path of the command





$ man <command> #manual page of the command

$ info <command> #information about the command

$ <command> --help #it will gives a command options and there usage

$ apropos <keyword> #to know about the command use

**5. OS Installation**



***Options:***

**Remove all partitions on selected drives and create default layout** this options will delete all the drive partitions and install OS

**Remove all Linux Partitions** this option will only remove Linux partitions and creates default layout, install OS

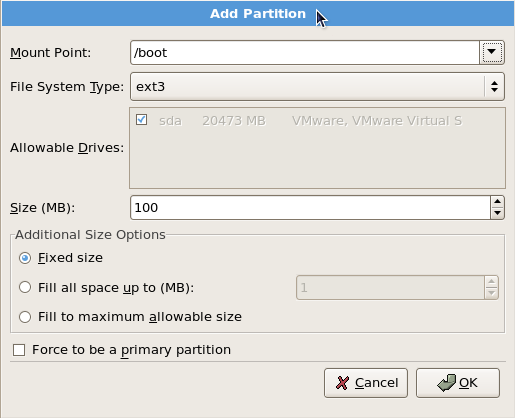
**Use free space** this option will install on free space of the drive creates default layout

**Create Custom layout** this option will not create any layout you have to create as per your requirement

This option will give more customization you to create your own layout (**Selected Customize Layout**)



Now you can able to see HDD space to create partitions follow click on **NEW**



After clicking on NEW you will see above popup

First create /boot because it needs normal partition to store MBR (master boot record). 100MB is enough for /boot partition.

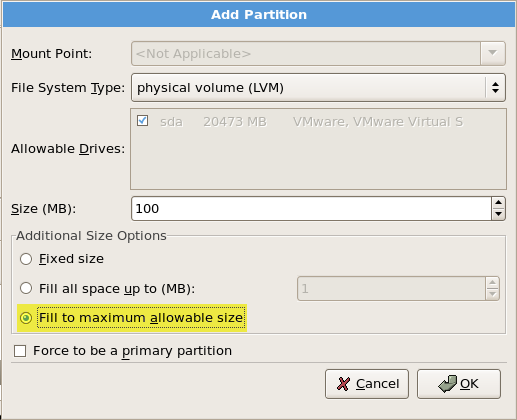
Mount point /boot

File system Type: ext3 OR ext4 Select fixed size Click **OK**

See above screen /boot partition is created.

***Note:*** Always remember to create LVM partitions OR RAID partitions is recommended, because later on if you want extend the ROOT file system is possible and easy.

To create LVM partition click on **NEW** you can able to see popup as below



For creating a LVM partition we have to create **Physical Volume** select file system type as **physical volume (LVM)**

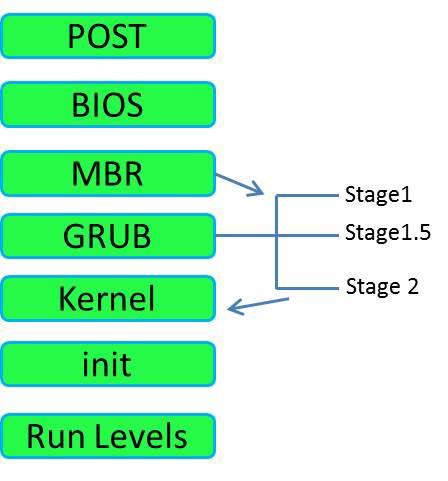
Select File system type as SWAP, provide logical volume name as “SWAP” for easy identification. Provide the swap size is always (RAM SIZE \* 2) in my case 2GB RAM. Click on **OK**

As you created above all the partitions are created. Click **NEXT Minimum Recommended Partitions**

Slash “/”

/boot SWAP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **6. Booting Process** | Document No. | : | RHEL Professional Guide |
| Author | : | Ankam Ravi Kumar |
| Web site | : | <http://ark-library.blogspot.in/> |
|



**POST:** Power on self-test - Hardware will check it self

**BIOS:** Basic input and output system will load

**MBR:** Master boot record – which will have 446Bytes in size first sector of the HDD.

|  |  |  |
| --- | --- | --- |
| 446 MBR | 64 Partition Table | 2 Active Sign |

**GRUB:** it will boot in three steps

Stage 1: The duty of the stage 1 is to identify where the stage is located to

Stage 1.5: The duty of the stage 1.5 is to synchronize with the deferent file system. LBA

– logical block addressing.

Once executed, core.img will load its configuration file and any other modules needed, particularly file system drivers; at installation time, it is generated from diskboot.img and configured to load the stage 2 by its file path.

Stage 2: it will list out all the available operating systems and it will boot with the default operating system.

**Kernel:** Kernel version and its installed modules will load

**Init:** init is the root daemon process after this process all the process will start under init

#### Run levels:

Run levels are 7

0 - Shutdown/Halt/Power off

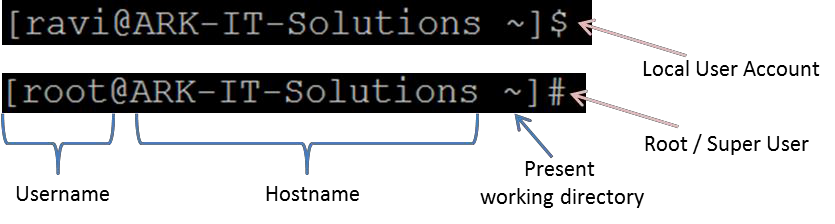
1. - Single User Mode
2. - Multiple user without Network
3. - Multiple users with Network (CLI)
4. - Unused
5. - GUI mode
6. - Reboot

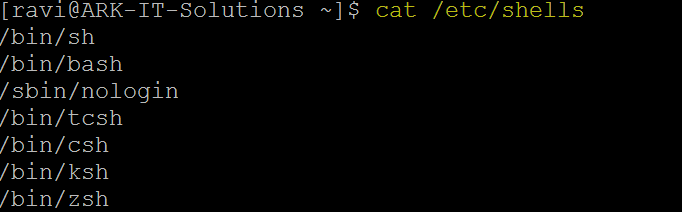
After completion of run level it will enter into the User interface will ask you for the credential to login.

1 sylinder = 7.8 MB

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **7. Bash Features** | Document No. | : | RHEL Professional Guide |
| Author | : | Ankam Ravi Kumar |
| Web site | : | <http://ark-library.blogspot.in/> |
| Page No. | : | 36 |

BASH: Bourne again shell





##### Command Line Completion:

Single Tab – it will provide the best option

Double Tab – it will provide all the possible options (Autocompletes

##### Command Line Editing:

Ctrl+a - It moves cursor to the Home line position Ctrl+b - Moves the cursor back one character

Ctrl+c - Sends the signal SIGINT to the current task, which aborts and close it. Ctrl+d - Close current shell prompt

Ctrl+e - It will move a cursor to end of the bash Ctrl+f - Moves cursor forward one character

Ctrl+g - Abort the research and restore the original file Ctrl+h - Deletes the previous character (Same as backspace)

Ctrl+k - It is used to delete the command from the courser to line home position Ctrl+l - Clear the screen

Ctrl+u - Clears the line content before the cursor and copies it into the clipboard Ctrl+y - Yank the content from the cursor position

Ctrl+z - Sends the signal SIGTSTP to the current task, which suspend it Ctrl+Shift+c - Copy selected text

Ctrl+Shift+v - Paste the copied content Ctrl+Shift+t - Tab

Ctrl+Shift+n - New Terminal Ctrl+Shift+w - Close Tab Ctrl+Shfift+q - Close a Terminal

Alt+b - Moves the cursor backward one word

Alt+c - Capitalizes the character under the cursor and moves to the end of word Alt+d - Cut the word after the cursor

Alt+f - Moves the cursor forward one word

Alt+l - Lowers the case of every character from the cursor’s position to the end Alt+. - Insert the last argument to the previous command

##### Command Line History:

$ history - this command will display all the previous executed commands

$ history –c - Clear the command history

$ !<number> - it executes mentioned number command

**Note:** Default history size is 1000 commands.

$ !<charectar> - it will display/execute matching character command

$ !! - it will execute last executed command

##### Sophisticated prompt control:

? - It will replace a single character

|  |  |  |
| --- | --- | --- |
| Example: | rm –rf a? - | it will delete the files with two characters |
|  | Rm –rf a?? - | it will delete the three characters files after “a” |

\* - replace multiple number of characters

Example: rm –f a\* - it will remove all the files which are starting with “a”

##### Piping and Redirecting:

***Redirecting*** input and output from standard stream to user defined place

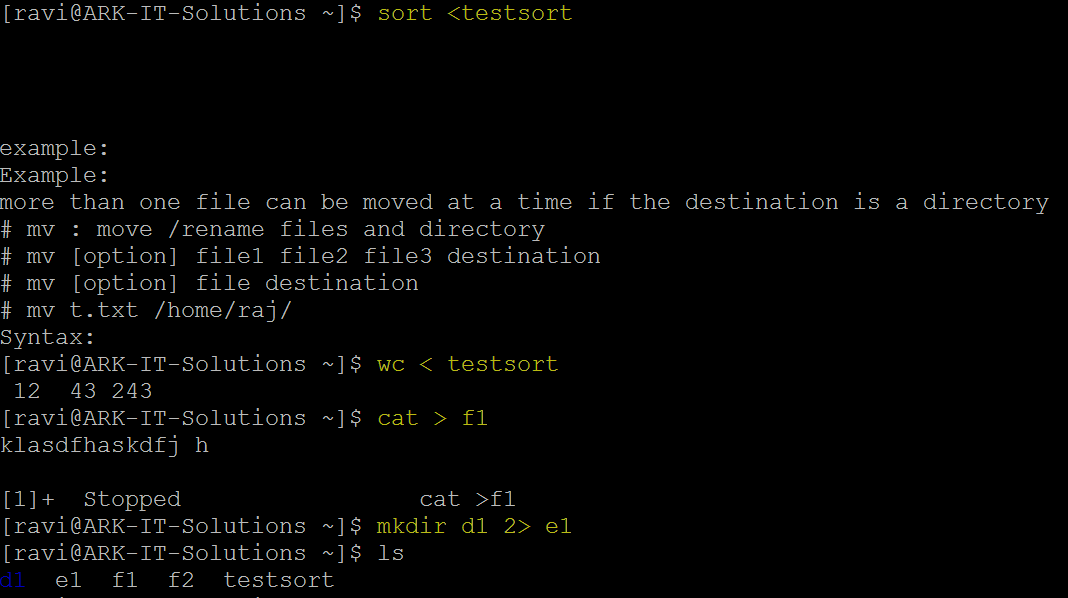
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | Input RD -0 |  | < |  |
| 2. | Output RD -1 | Values | > | Symbols |
| 3. | Error RD -2 |  | > |  |

Example: $ sort < <file name>

$ wc < <file name>

$ history > f2

$mkdir d1 2> e1



***Piping*** sending output or one command as a input to the another command

**|** Pipe  symbol. Page related command:

**Less:** is used to see the command output page by page in up and down way

**More:** we can’t go upward downward, just see the output fit to the screen



**8. Linux Directory Structure**

##### Overview

Everything in Linux can be reduced to a file. Partitions are associated with files such as

/dev/hda1. Hardware components are associated with files such as /dev/modem. The Filesystem Hierarchy Standard (FHS) is the official way to organize files in Unix and Linux directories.

##### Linux file system and directory structure

Several major directories are associated with all modern Unix/Linux operating systems. These directories organize user files, drivers, kernels, logs, programs, utilities, and more into different categories. The standardization of the FHS makes it easier for users of other Unix-based operating systems to understand the basics of Linux. All of the other directories shown in Table are subdirectories of the root directory, unless they are mounted separately.

|  |  |
| --- | --- |
| **Directory** | **Description** |
| / | The root directory, the top-level directory in the FHS. All other directories are subdirectories of root, which is always mounted on some partition. All directories that are not mounted on a separate partition are included in the root directory’s partition. |
| /bin | Essential command line utilities. Should not be mounted separately; otherwise, it could be difficult to get to these utilities when using a rescue disk. |
| /boot | Includes Linux startup files, including the Linux kernel. Can be small; 16MB is usually adequate for a typical modular kernel. If you use multiple kernels, such as for testing a kernel upgrade, increase the size of this partition accordingly. |
| /etc | Most basic configuration files. |
| /dev | Hardware and software device drivers for everything from floppy drives to terminals. Do not mount this directory on a separate partition. |
| /home | Home directories for almost every user. |
| /lib | Program libraries for the kernel and various command line utilities. Do not mount this directory on a separate partition. |
| /mnt | The mount point for removable media, including floppy drives, CD-ROMs, and Zip disks. |
| /opt | Applications such as WordPerfect or Star Office. |
| /proc | Currently running kernel-related processes, including device assignments such as IRQ ports, I/O addresses, and DMA channels. |
| /root | The home directory of the root user. |
| /sbin | System administration commands. Don't mount this directory separately. |
| /tmp | Temporary files. By default, Red Hat Linux deletes all files in this directory periodically. |
| /usr | Small programs accessible to all users. Includes many system administration commands and utilities. |
| /var | Variable data, including log files and printer spools. |

**9. Text Editors**

Console Based Text Editors:

1. Emacs
2. Nano
3. Vim/vi

Graphical (GUI) Editors:

1. Gedit
2. Gvim
3. Sublime (have to install separately)

Now we are going to discuss about vi/vim editor.

Vim is a highly configurable text editor built to enable efficient text editing. It is an improved version of the VI editor distributed with most UNIX systems. It is a tool, the use of which you must be learned. Program is written by Bram Moolenaar et al.

Usage of vi/vim, it has three modes:

1. Command mode
2. Insert Mode
3. Ex-mode

***Command mode:*** this mode is the default mode following things we can do

Yank/copy (yy) - copying the single line (nyy) n number of lines at a time you can copy.

p - Paste a content bellow the cursor

P - Paste content above the cursor

dd - Delete a single line ndd n number of line delete yet a time

x - Deleting a single character

dw - Deleting a single word

. - redo

u - Undo

J - Joining the line

r - Replace the character

#### Arrow Keys

h - Left

1. - Down
2. - Up
3. - Right Ctrl+d - Scroll Down Ctrl+u - Scroll UP

***Insert Mode:*** This mode is used to insert the text into the file

i - Inserting the content before the cursor position I - it will insert the character starting of the line

a - it will insert the character of the cursor append A - It will append the character from end of the line O - It will insert a new line above the cursor

* - It will insert a new line below the cursor s - Substitute the stream

***EX-Mode:*** Esc is used to change the mode. Press Escape key to enter EX-Mode

:w - save the modifications

:q - quit

:wq - save & quit

:w! - save forcefully

:q! - Quit forcefully

:set nu - setting the line numbers

:set nonu - Remove line numbers

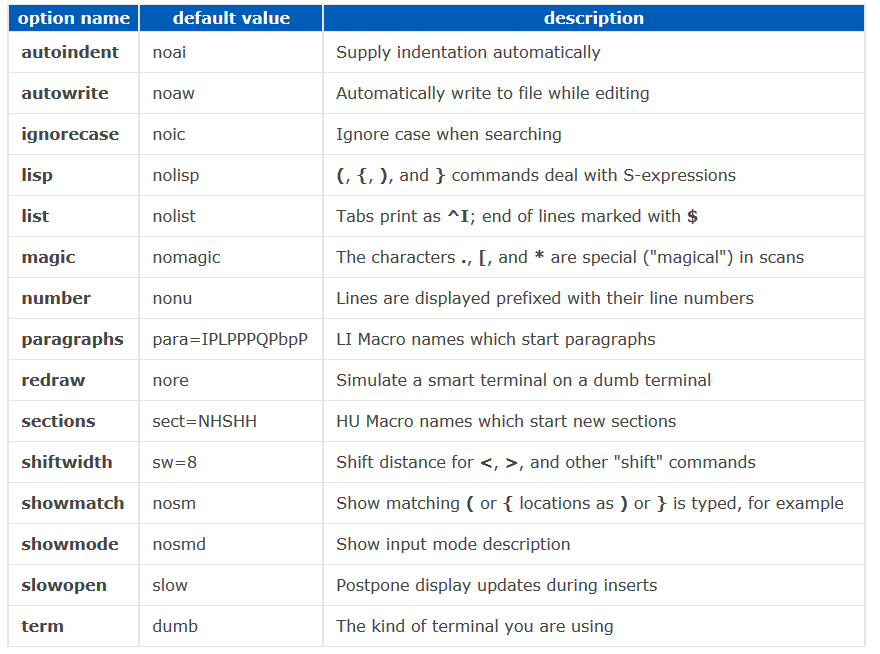
:<number> - it will goes to particular line

:%s/find string name/replace string/g - to replace the string group of lines

:r - reading the another file from here

:!<command> - command will execute from here

:r !date - output will save in current fil



|  |  |
| --- | --- |
|  | **10.User Administration** |
|
|
|

##### Type of Users:

**Root user** - Default user highly privileged UID is 0. This will create while installing the operating system

**System users** - is nothing but services, at the time of installing particular package. UID starts from 1 to 499.

**Local users** - after installing of the operating system admin user will create these users. UID starts from 500 to 65534.

* + After creating a user, user home directory will be created in default path /home.
  + One group is will be created with same user name (primary group)
  + Files from /etc/skel will be copied automatically to user home directory
  + /etc/passwd file is updated with user information
  + /etc/group file is update with primary group information Important files

/etc/passwd User Information

/etc/shadow User Passwords

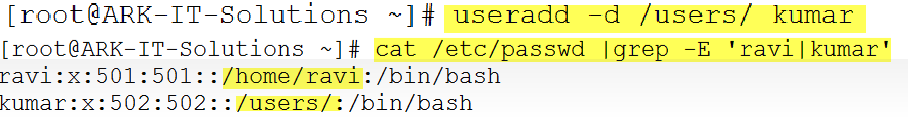
/etc/group Group Information

/etc/gshadow Group Passwords

##### User Administration Commands:

# useradd <user name> - To create specified local user

# useradd –d <home directory> <user name> - create a user with specified home path



# useradd –u <UID> <user name> - create user with specific UID.

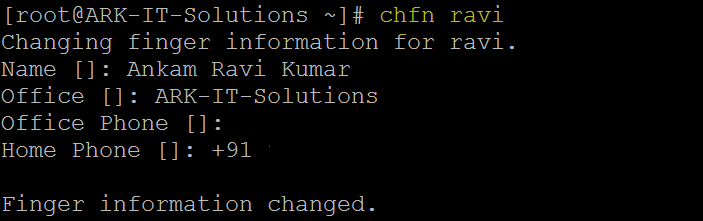
# passwd <user name> - change the user password

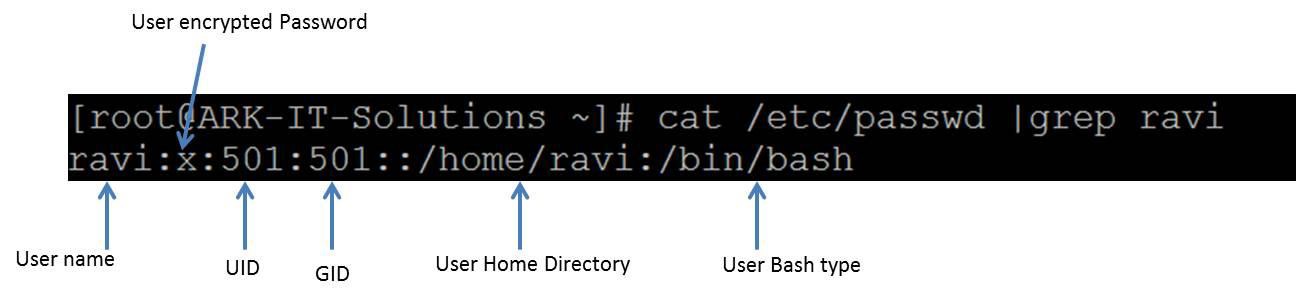
# userdel <user name> - delete user

# userdel <user name> - delete user including home directory

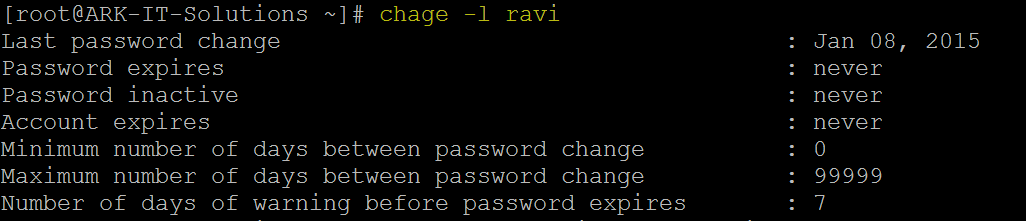
# finger <user name> - See user properties

# chfn <user name> - Change user information





# chage –l <user name> - to check user password expiry and account expiry information



# su - <user name> - Switch to other user account

# id <user name> - it will show the user id



# system-config-users - create and manage user account in GUI

##### Usermod command options:

-c = We can add comment field for the user account.

-d = To modify the directory for any existing user account.

-e = Using this option we can make the account expiry in specific period.

-g = Change the primary group for a User.

-G = To add a supplementary groups.

-a = To add anyone of the group to a secondary group.

-l = To change the login name from arkit to arkit\_admin.

-L = To lock the user account. This will lock the password so we can’t use the account.

-m = moving the contents of the home directory from existing home dir to new dir.

-p = To Use un-encrypted password for the new password. (NOT Secured).

-s = Create a Specified shell for new accounts.

-u = Used to Assigned UID for the user account between 0 to 999.

-U = To unlock the user accounts. This will remove the password lock and allow us to use the user account.

##### Creating Groups:

Group information is located/stored on /etc/group file.

# groupadd <group name> - Create a group with specified name

# usermod –G <group name> <user name> - Add user to group

# gpasswd –a ravi Administrators – Adds the user ravi to the group Administrators

# gpasswd –A ravi Administrators – give user ravi administrative rights to the group

# gpasswd –d ravi Administrators – remove user ravi from the group Administrators

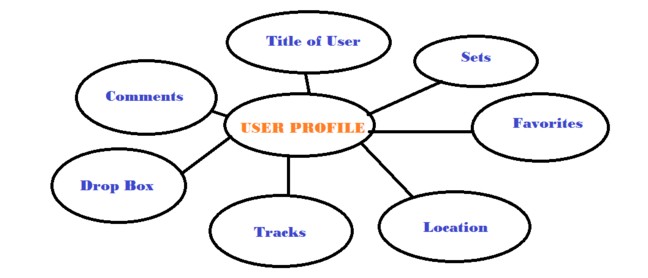
# groupdel <group name> - Delete group name

# groupmod –n <new group name> <old group name> - change group name

# newgrp - <group name> - Login into the group if successful, re-initializes the user environment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **11. Profile Management** | Document No. | : | RHEL Professional Guide |
| Author | : | Ankam Ravi Kumar |
| Web site | : | <http://ark-library.blogspot.in/> |
| Page No. | : | 47 |

A user profile is a visual display of personal data associated with a specific user, or a customized desktop environment. A profile refers therefore to the explicit digital representation of a person's identity. A user profile can also be considered as the computer representation of a user model.



/etc/profile - it contains system void variables, if you do any modification in this file it will effect to the administrator and local user profiles.

~/.bash\_profile - it contains user specific variables, if you do any modification in this file it will effect to that particular account only.

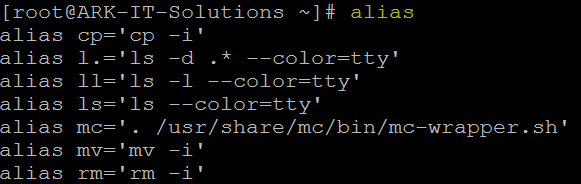
/etc/bashrc - it contains system void alias variables

~/.bashrc - it contains user specific alias variables

.bash\_history - it contains all executed commands history

##### Commands:

# alias - it will show the aliases



# unalias <alias name> - it will remove mentioned alias

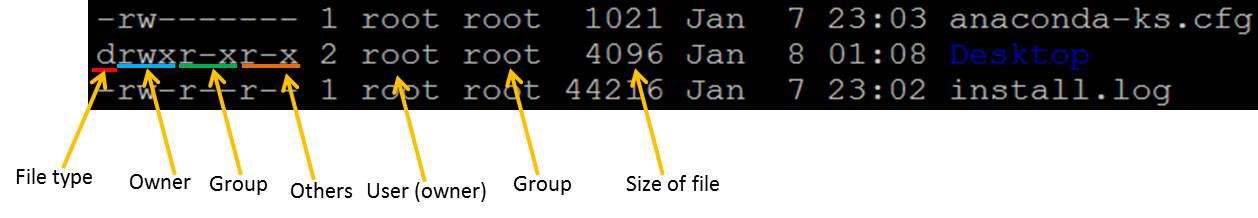
***Note:*** you can always define an alias using /etc/bashrc OR .bashrc files

##### File Permissions:

|  |  |  |
| --- | --- | --- |
| **Permission** | **Value** | **Number** |
| Read | r | 4 |
| Write | w | 2 |
| Execute | x | 1 |

Default permissions when you create a file or directory

|  |  |
| --- | --- |
| File Permissions | |
| File | 644 |
| Directory | 755 |



In above image explained about file permissions

##### Commands to Change file/directory permissions

Symbolic permissions

u - user/owner

g - Group

* - Others

1. - Write
2. - Execute

+ - Allow

* - deny

# chmod [options] <mode/permissions> <file/directory> - to change permissions file/folder

Example: chmod 744 file1

# chmod u+rwx file or directory : in case of user only

# chmod ug+rwx file or directoty : in case of user and group

# chmod u+w,g+r,o+x directory/file

# chmod u+rw,g+rw directory/file

# chmod u-r, g-w,o-rw directory/file

# chmod ugo+rwx file/directory

# chmod ugo-rwx file/directory

# chown [options] <new owner> <file/directory> - to change ownership of file/folder

Example: chown user2 file1

chown user1:group1 file2

# chgrp [options] <new group> <file/directory> - to change group of file/folder

Example: chgrp gorup2 file2

|  |  |
| --- | --- |
|  | **12. String Related Commands** |
|
|

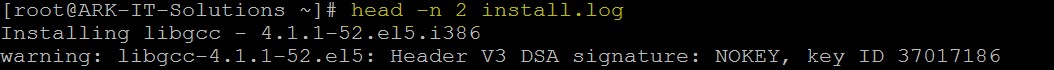
String related commands will help you to print/search file text as required

##### HEAD

Head prints the first N number of data of the given input. By default, it prints first 10 lines of each given file.

Example: head file2

head –n 2 file3 #number of lines



##### SORT

Sort is a simple and very useful command which will rearrange the lines in a text file so that they are sorted, numerically and alphabetically. By default, the rules for sorting are:

* + Lines starting with a number will appear before lines starting with a letter.
  + Lines starting with a letter that appears earlier in the alphabet will appear before lines starting with a letter that appears later in the alphabet.
  + Lines starting with a lowercase letter will appear before lines starting with the same letter in uppercase.

Example: sort –r file2



##### Options

-b ignores leading blanks

-d considers only blanks and alphanumeric characters

-f fold lower case to upper case characters

-g compare according to general numerical value

-i consider only printable characters

-M compare (unknown) < `JAN' < ... < `DEC'

-n compare according to string numerical value

-r reverse the result of comparisons

-c check whether input is sorted; does not sort

-k start a key at POS1, end it at POS2 (origin 1)

-m merges already sorted files; do not sort

-o write result to FILE instead of standard output

-s stabilize sort by disabling last-resort comparison

-S use SIZE for main memory buffer

-t use SEP instead of non-blank to blank transition

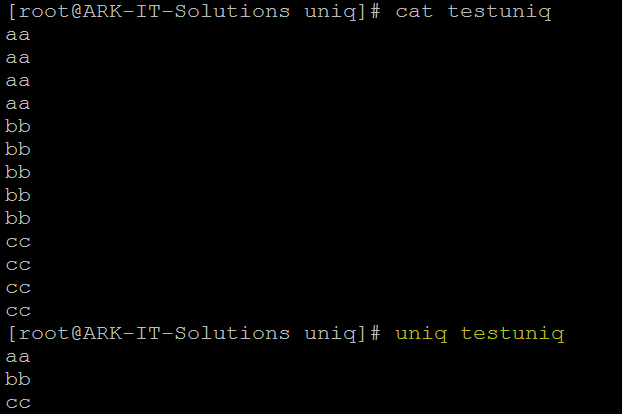
-T use DIR for temporaries, not $TMPDIR or /tmp

-z end lines with 0 byte, not newline

##### UNIQ

Uniq command is helpful to remove or detect duplicate entries in a file.

Example: uniq <file name> - it will print uniq values

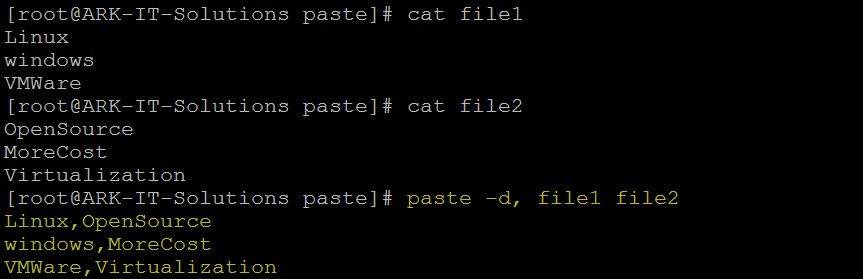


##### PASTE

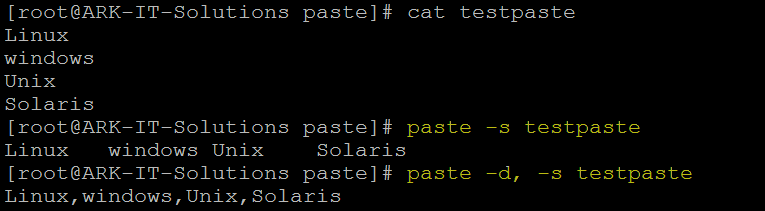
It is very useful for merging a single file and also for merging set of files as well.

* paste command examples for single file handling
* paste command examples for multiple files handling Example: paste –s file1

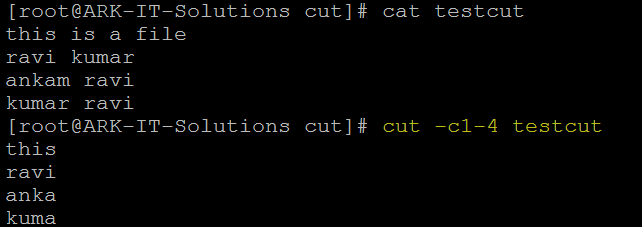
##### CUT



paste –d, -s file1

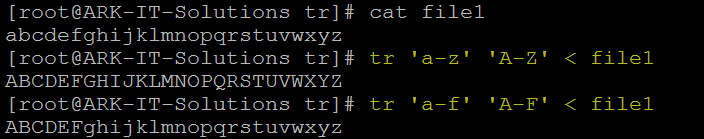


Cut is used for text processing. You can use this command to extract portion of text from a file by selecting columns.



##### TR

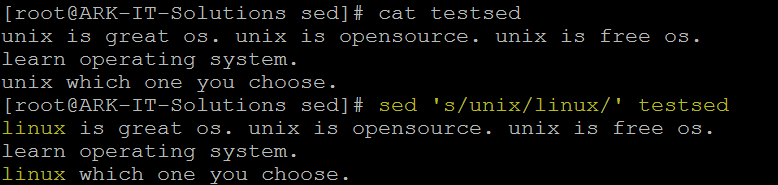
It will translate content of the file from one case to another case vice versa.



**Note:** if you want to change the file text to caps then redirect the output to another file

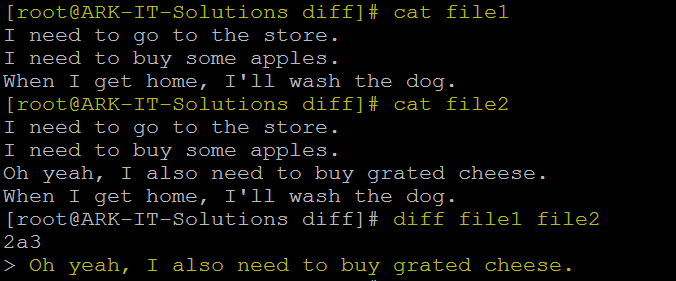
##### SED

Sed is a Stream Editor used for modifying the files in unix (or linux). Whenever you want to make changes to the file automatically, sed comes in handy to do this. Most people never learn its power; they just simply use sed to replace text. You can do many things apart from replacing text with sed.



##### DIFF

To compare the difference between two files text you can use this command



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **13. File Links** | Document No. | : | RHEL Professional Guide |
| Author | : | Ankam Ravi Kumar |
| Web site | : | <http://ark-library.blogspot.in/> |
| Page No. | : | 54 |

##### File Types

-b - block device file Example: HDD and pen drive

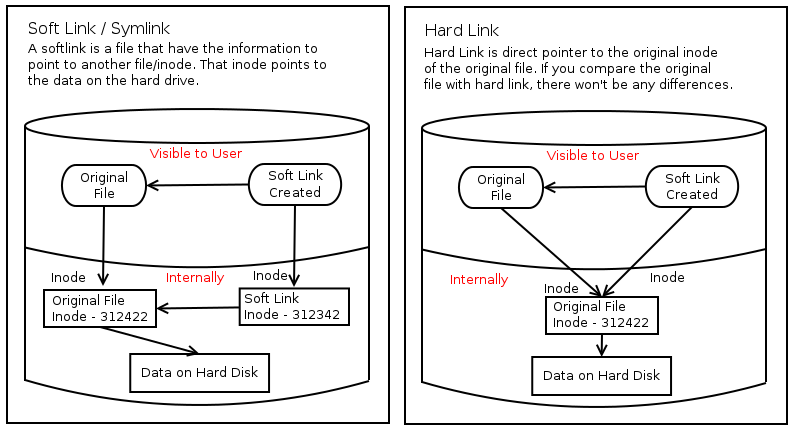
-d - directory file

- - common file

c - Character device file Example: terminal l - Linked file

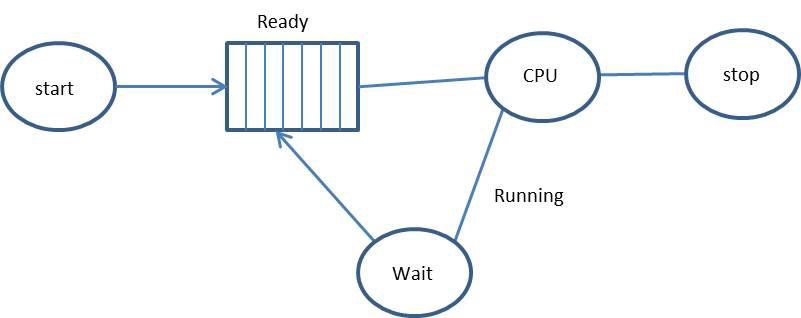
Linking means reflecting to the original file, In case of copy command updating is not possible after copying the file from the source to destination. In link updating is possible for both the files.

|  |  |
| --- | --- |
| **HARD Link** | **SOFT Link** |
| 1. The destination file is exact image of the source file. | 1. The destination file size is length of the source file name |
| 2. If source got deleted also even we can access the destination file | 2. if source got deleted we can’t access destination file |
| 3. inode numbers of source and destination are same | 3. inode numbers of source and destination are different |
| 4. We can’t put the hard link to different file system (partitions) because it will different. | 4. we can put a link between different file systems |
| 5. Ex: cp –l <source> <destination> | Ex: cp –s <source> <destination> |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **14. Process Management** | Document No. | : | RHEL Professional Guide |
| Author | : | Ankam Ravi Kumar |
| Web site | : | <http://ark-library.blogspot.in/> |
| Page No. | : | 55 |

The program in running action



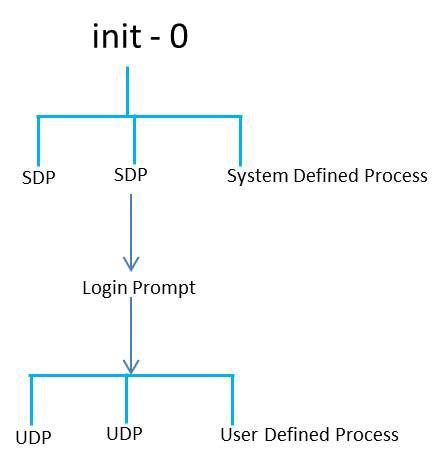
* 1. System Defined Process
  2. User Defined Process

System defined process is called as daemon, it is a program running for the service.

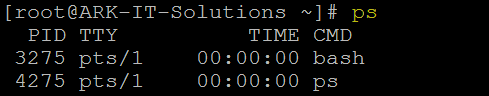
It will start when booting the operating system, we can also start manually.

User defined process is called executing commands.

Init is the parent process for all the processes. Process id is 0 always.



$ ps - it will display the process status correct terminal



pts/0 - sudo terminal

tty/1 - virtual terminal

Example: ps –a (a = all the processes)

ps –f - full description of the process ps –af

ps –u - for user processes ps –x - system processes

ps –ax - all terminals system processes To see the background running processes

$ jobs - to see background processes

$ fg

$ top - it will display dynamic running processes correct time, system uptime and number of users logged in, CPU load memory and processes.

##### Killing the processes:

# kill -9 <PID> - kill the process using Process ID

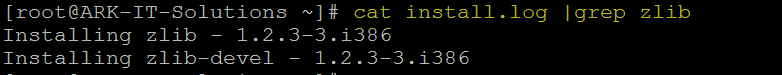
# kill –a <name> - kill the process using process name

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **15. Search Related Commands** | Document No. | : | RHEL Professional Guide |
| Author | : | Ankam Ravi Kumar |
| Web site | : | <http://ark-library.blogspot.in/> |
| Page No. | : | 57 |

Search commands will help you to search files/directories and file content.

##### Command GREP

Grand regular expression, it is used to search the file name and content of the file



##### Command FIND

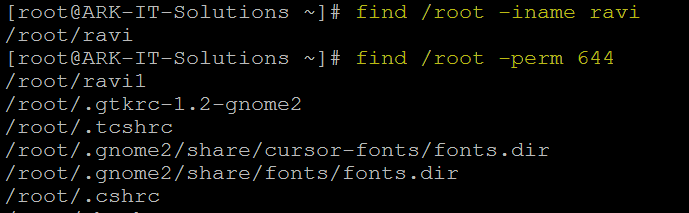
It is command to find files with different options.

# find / -iname <file name> - to search files with file name

# find / -perm 770 - to search files with their permissions

# find / -user <user name> - to search files with user ownership

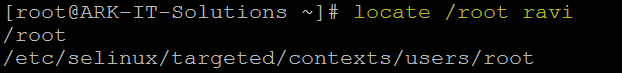
# find / -size 10M - to search files with their size



##### Command LOCATE

Locate command can be used to searching for the file based on the string name from

/ onwards.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **16. Task Automation and Task Scheduling** | Document No. | : | RHEL Professional Guide |
| Author | : | Ankam Ravi Kumar |
| Web site | : | <http://ark-library.blogspot.in/> |
| Page No. | : | 58 |

Scheduling is very useful when you prefer do some automation jobs. For scheduling the tasks in Linux we are using **at** and **crontab  *Command AT:***

$ at – it is used to scheduled one are more jobs (commands) for a single execution.

Example: $ at 10:30

$ at now+5 minutes

$ at tomorrow

$ at 10:30 july 15 2015

|  |  |  |  |
| --- | --- | --- | --- |
| **The expression** | | **Would translate to** | |
| noon | | 12:00 PM October 18 2014 | |
| midnight | | 12:00 AM October 19 2014 | |
| teatime | | 4:00 PM October 18 2014 | |
| tomorrow | | 10:00 AM October 19 2014 | |
| noon tomorrow | | 12:00 PM October 19 2014 | |
| next week | | 10:00 AM October 25 2014 | |
| next monday | | 10:00 AM October 24 2014 | |
| fri | | 10:00 AM October 21 2014 | |
| NOV | | 10:00 AM November 18 2014 | |
| 9:00 AM | | 9:00 AM October 19 2014 | |
| 2:30 PM | | 2:30 PM October 18 2014 | |
| 1430 | | 2:30 PM October 18 2014 | |
| 2:30 PM tomorrow | | 2:30 PM October 19 2014 | |
| 2:30 PM next month | | 2:30 PM November 18 2014 | |
| 2:30 PM Fri | | 2:30 PM October 21 2014 | |
| 2:30 PM 10/21 | | 2:30 PM October 21 2014 | |
| 2:30 PM Oct 21 | | 2:30 PM October 21 2014 | |
| 2:30 PM 10/21/2014 | | 2:30 PM October 21 2014 | |
| 2:30 PM 21.10.14 | | 2:30 PM October 21 2014 | |
| now + 30 minutes | | 10:30 AM October 18 2014 | |
| now + 1 hour | | 11:00 AM October 18 2014 | |
| now + 2 days | | 10:00 AM October 20 2014 | |
| 4 PM + 2 days | | 4:00 PM October 20 2014 | |
| now + 3 weeks | | 10:00 AM November 8 2014 | |
| now + 4 months | | 10:00 AM February 18 2015 | |
| now + 5 years | | 10:00 AM October 18 2019 | |

$ atq - is a command to see scheduled jobs

$ atrm <job number> – is a command to remove the schedules jobs

##### CRONTAB:

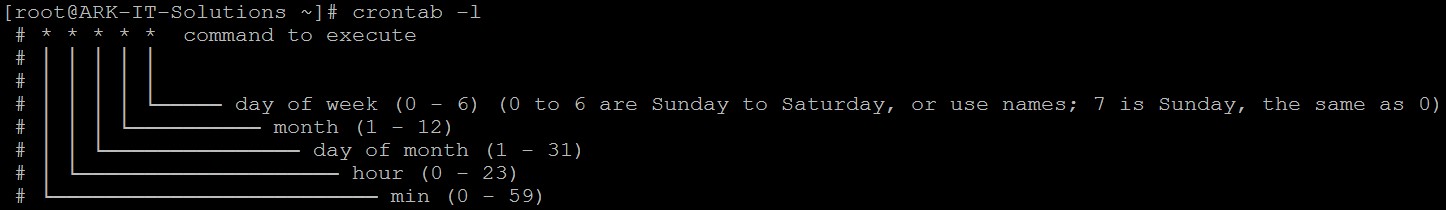
The crontab is a list of commands that you want to run on a regular schedule, and also the name of the command used to manage that list.

Example: crontab –e - to edit the jobs

crontab –l - to list the scheduled jobs

crontab –c –u <user name> - to see particular user jobs crontab –r - to remove crontab file

crontab format



# \* \* \* \* \* command to execute

# │ │ │ │ │

# │ │ │ │ │

# │ │ │ │ └───── day of week (0 - 6) (0 to 6 are Sunday to Saturday, or use names; 7 is Sunday, the same as 0)

# │ │ │ └────────── month (1 - 12)

# │ │ └─────────────── day of month (1 - 31)

# │ └──────────────────── hour (0 - 23)

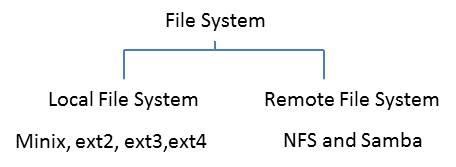
# └───────────────────────── min (0 - 59)

Example: 5,10,15 \* \* \* \* mail –s “Mail from root” [aravikumar48@gmail.com](mailto:aravikumar48@gmail.com)

This above example will send a mail yet every 5th 10th and 15th minute of hour.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **17. File System** | Document No. | : | RHEL Professional Guide |
| Author | : | Ankam Ravi Kumar |
| Web site | : | <http://ark-library.blogspot.in/> |
| Page No. | : | 60 |

A file system is the underlying structure a computer uses to organize data on a hard disk. If you are installing a new hard disk, you need to partition and format it using a file system before you can begin storing data or programs.



Minix - 14 characters, 64MB of storage

Ext - Extended file system, 255 characters, 2GB of storage

When you’re going to format the Linux file system using extend file system it will create blocks.

1). Master Block/Boot Block 2). Super Block

3). Inode Block

4). Data Block

1. Master block entry is located at partition table, only boot partition contains master blocks data. Remaining partitions master blocks are empty.
2. Super block just like an index to the book and it will holds to the information as follows
   1. Utilized inode numbers
   2. Free inode numbers
   3. Utilized data blocks
   4. Free data blocks

Super block holds all this information.

1. Inode table (index table) which holds all the information about files/directories like permissions, owner, group name, size and time stamps.

4096 bytes default block size 15 data blocks = inode

If data size is more than 100MB block size is 4096bytes. If data size is less than 100MB block size is 1024bytes.



Directory holds inode number of file and file name

1. Data block storage of files

##### Table

Now below is a very brief comparison of the most common file systems in use with the Linux world

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **File System** | **Max File Size** | **Max Partition Size** | **Journaling** | **Notes** |
| Fat16 | 2 GB | 2 GB | No | Legacy |
| Fat32 | 4 GB | 8 TB | No | Legacy |
| NTFS | 2 TB | 256 TB | Yes | (For Windows Compatibility) NTFS-3g is installed by default in Ubuntu, allowing Read/Write support |
| ext2 | 2 TB | 32 TB | No | Legacy |
| ext3 | 2 TB | 32 TB | Yes | Standard linux filesystem for many years. Best choice for super-standard installation. |
| ext4 | 16 TB | 1 EB | Yes | Modern iteration of ext3. Best choice for new installations where super-standard isn't necessary. |
| reiserFS | 8 TB | 16 TB | Yes | No longer well-maintained. |
| JFS | 4PB | 32PB | Yes (metadata) | Created by IBM - Not well maintained. |
| XFS | 8 EB | 8 EB | Yes (metadata) | Created by SGI. Best choice for a mix of stability and advanced journaling. |
| **GB = Gigabyte (1024 MB) :: TB = Terabyte (1024 GB) :: PB = Petabyte (1024 TB) :: EB = Exabyte (1024 PB)** | | | | |

##### Journaling:

A journaling file system is more reliable when it comes to data storage. Journaling file systems do not necessarily prevent corruption, but they do prevent inconsistency and are much faster at file system checks than non-journaled file systems. If a power failure happens while you are saving a file, the save will not complete and you end up with corrupted data and an inconsistent file system. Instead of actually writing directly to the part of the disk where the file is stored, a journaling file system first writes it to another part of the hard drive and notes the necessary changes to a log, then in the background it goes through each entry to the journal and begins to complete the task, and when the task is complete, it checks it off on the list. Thus the file system is always in a consistent state (the file got saved, the journal reports it as not completely saved, or the journal is inconsistent (but can be rebuilt from the file system)). Some journaling file systems can prevent corruption as well by writing data twice.

Journaling

| | | Writable more ordered more journal

1. Write block – metadata is stored in HDD
2. Ordered more – metadata and actual data
3. Journal will maintain more space to do journaling

##### Network File System (NFS):

A Network File System (NFS) allows remote hosts to mount file systems over a network and interact with those file systems as though they are mounted locally. This enables system administrators to consolidate resources onto centralized servers on the network.

##### SAMBA (SMB):

Samba is a software it is used to share the file and printer between Linux to Windows vice versa..

##### Creating Standard Partition:

Disk partitioning is the act of dividing a hard disk drive (HDD) into multiple logical storage units referred to as partitions, to treat one physical disk drive as if it were multiple disks, so that a different file system can be used on each partition.

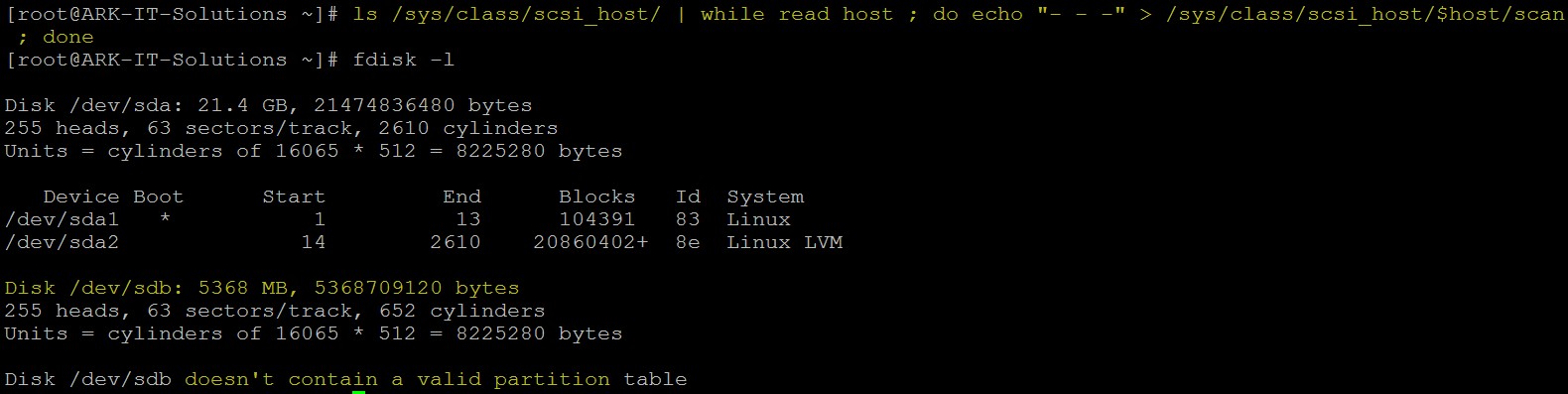
All the device files are stored in /dev/ directory. If your Hard disk is connected

|  |  |  |
| --- | --- | --- |
|  | SATA DRIVE | IDE DRIVE |
| Primary master | /dev/sda | /dev/hda |
| Primary slave | /dev/sdb | /dev/hdb |
| Secondary master | /dev/sdc | /dev/hdc |
| Secondary slave | /dev/sdd | /dev/hdd |

# fdisk - it is a utility used for creating, deleting, listing and checking partitions

In this situation i am using vmware environment so just added one disk 5GB in size to server.

# ls /sys/class/scsi\_host/ | while read host ; do echo "- - -" > /sys/class/scsi\_host/$host/scan ; done - To scan new hardware changes.

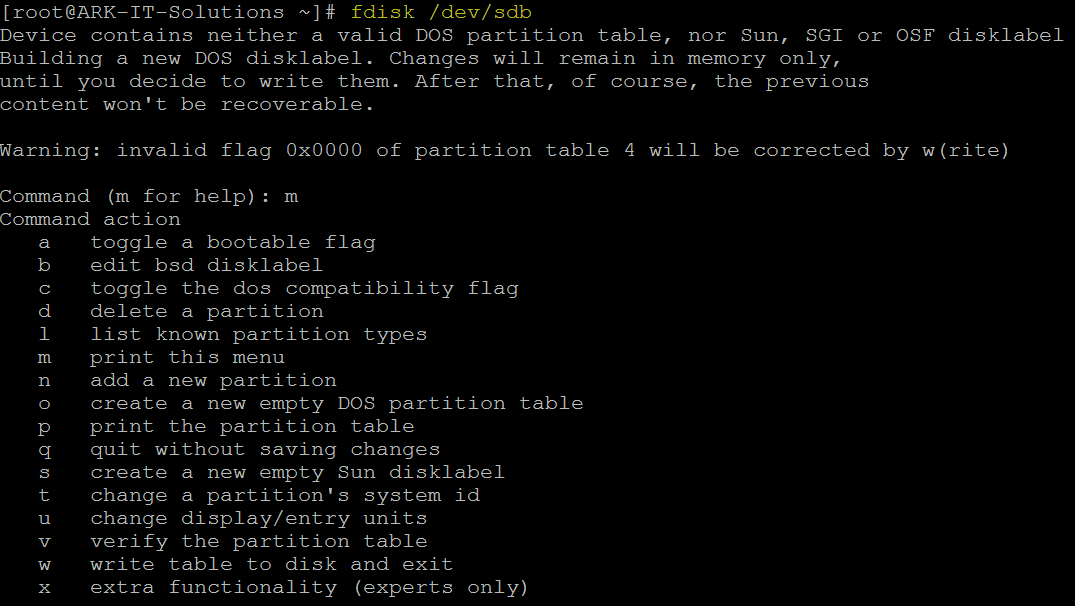


In above image we can able to see new hard disk is found it does not have valid partitions To create a partition we have to identify HDD name we can use fdisk utility to find.

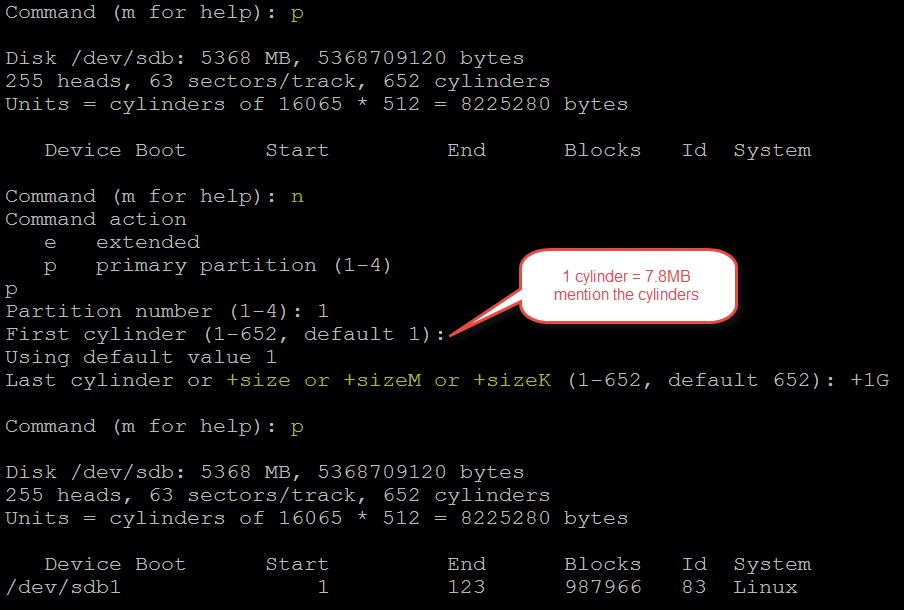
# fdisk –l – to identify HDD name

In this case our disk name is /dev/sdb

# fdisk /dev/sdb



List the options using m option



: p - print the partition list in fdisk utility

n – to create a new partition

mention the partition type primary of extended.

**Note:** (one extended OR 3 Primary 1 Extended) if you create extended partition first you can’t able to create a one more partition in same disk. We can create 3 primary and 1 extended partition.

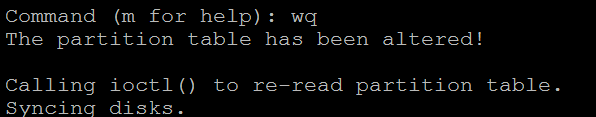
p for primary

enter the partition number (there is no partitions in our scenario entered 1)

if want to enter the space in first step then calculate cylinders (7.8MB\*number) if you want to provide the space using MB, KB and GB then just hit enter

+1G

Now see partition is created or not using p p



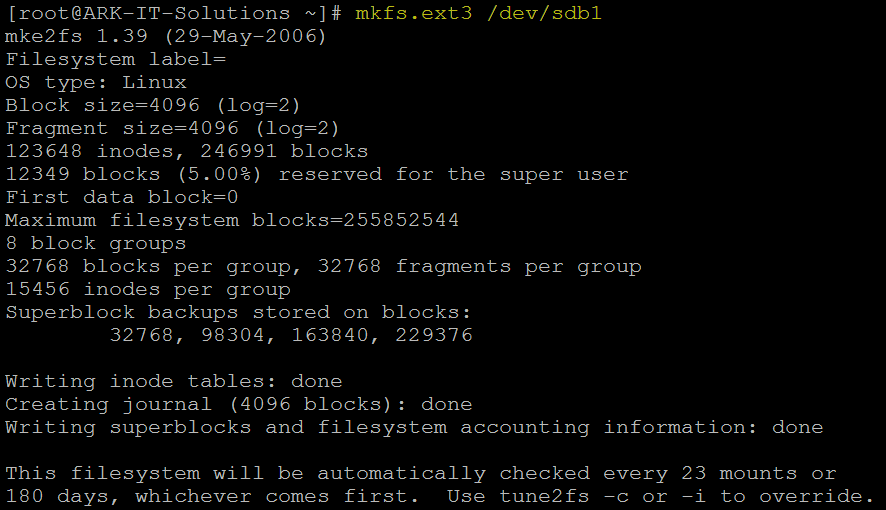
To save the partition information :wq



To update partition information to kernel

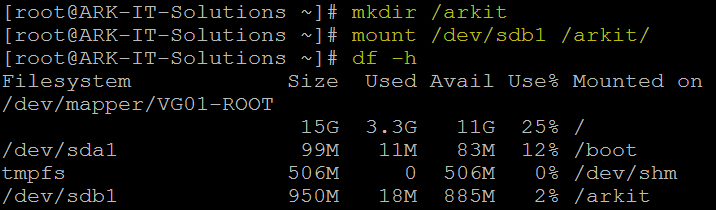
# partprobe /dev/sdb – to update partition information to partition table without reboot Partition created now we have to create a file system in partition

# mkfs.ext3 /dev/sdb1 - make file system in partition



After completion of creating a file system we have to mount the partition for use Create a directory to mount # mkdir /arkit

# mount /dev/sdb1 /arkit - to mount the file system



To check filee system size # df –h - check file system size

**Note:** SATA and SCSI we can create a 15 partitions and IDE 64 partitions only

**To access partition of windows in linux**

#mount –t vfat /dev/hdax /mnt

in order to check the label of any partition

#e2label /dev/hdax where x is number

**Mounting CD Rom**

# mount –t auto /dev/hdc /media/cdrom

-t : file type auto : file type

in order to check where cdrom is attached we can open the file fstab

#vi /etc/fstab

now in case of RHEl 3.0 we have to use command in order to unmount.

# umount /media/cdrom

and then eject the cdrom in case of RHEL 4.0 we simply type

# eject

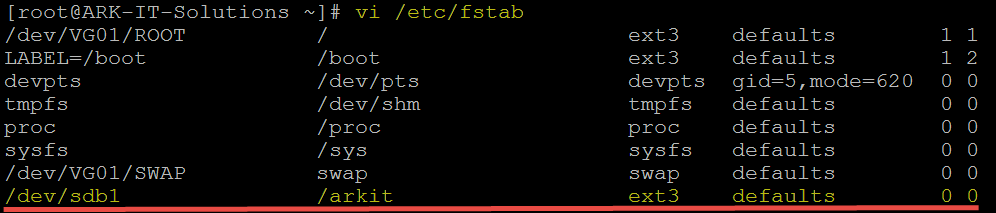
**Mounting Floppy**

#mount –t auto /dev/fd0 /media/floppy

in case of floppy we have to umount first then only we remove floppy otherwise all content of floppy may be lost or floppy may be physically damaged.

# umount /media/floppy

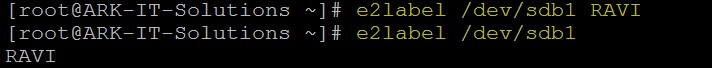
Permanent mount file system we have to edit /etc/fstab/ file.



add the entry as mentioned in above screenshot

# mount –a - to verify mentioned entry is correct, if not it will give you error message

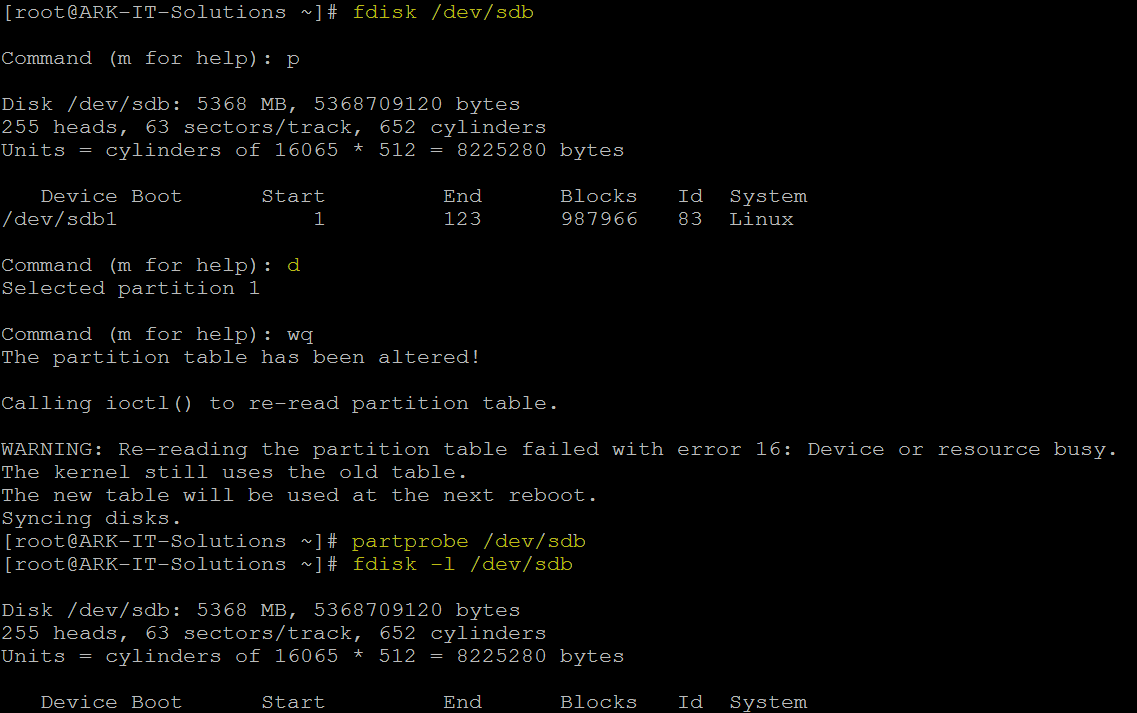
# e2label /dev/sdb1 RAVI - to add label to partition



# e2label /dev/sdb1 - to check disk label

##### DELETING PARTITION:

Deleting the partition will lead to lose the important data Remove the entry from /etc/fstab if it is mounted as permanent



# fdisk /dev/sdb

:d

If you have more than one partition it will ask for the partition id

:wq

# partprobe /dev/sdb

# fdisk –l /dev/sdb

##### Extending the Swap:

Create a partition

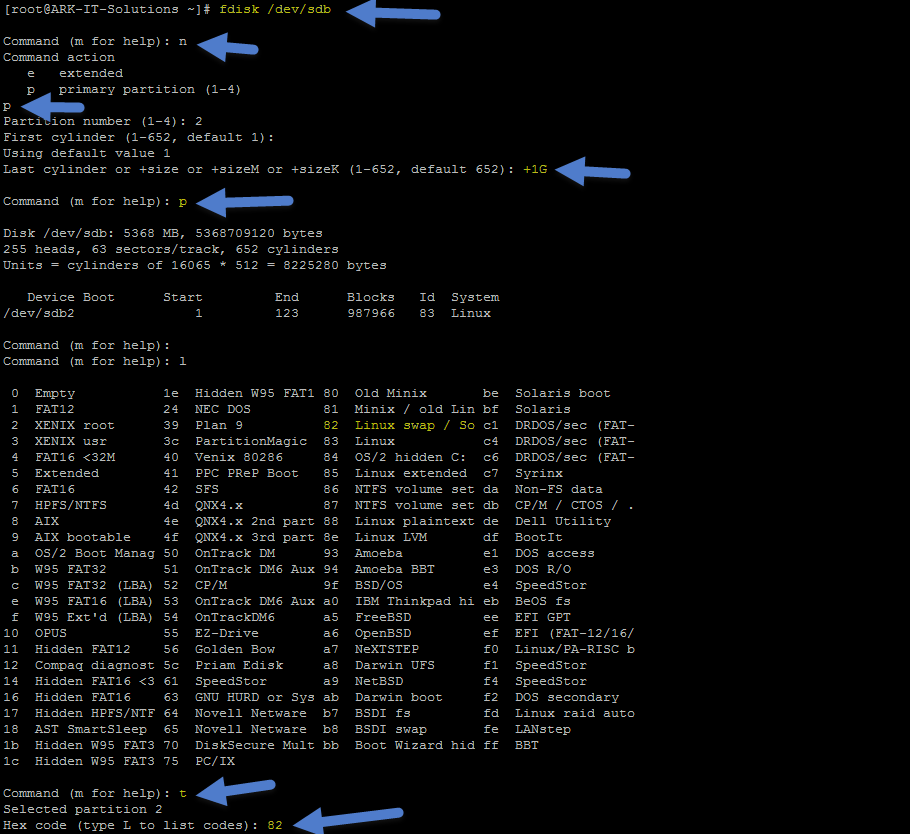
Change the partition type ID to 82 using t option in fdisk utility

# mkswap /dev/sdb2 - it will format in swap file system

# free - to check swap size

**Options:** -m in MB format

-G in GB format



# swapon /dev/sdb2 – in your case it may different.

To make permanent mount of swap file system edit the /etc/fstab file and add the entry

# mount –a - to refresh the /etc/fstab mounts

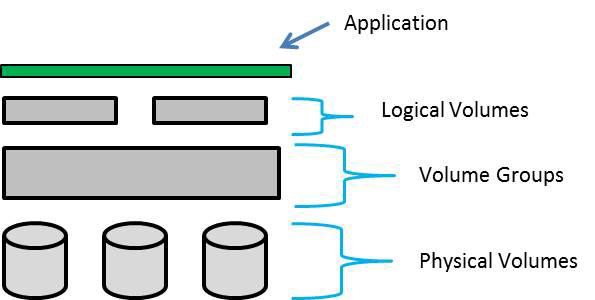
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **18. Logical Volume Manger (LVM)** | Document No. | : | RHEL Professional Guide |
| Author | : | Ankam Ravi Kumar |
| Web site | : | <http://ark-library.blogspot.in/> |
| Page No. | : | 70 |

LVM is a logical volume manager for the Linux kernel that manages disk drives and similar mass-storage devices. Using logical volume manager will give more flexibility to increase/reduce file system in more effective way and no loss of data.

##### Advantages

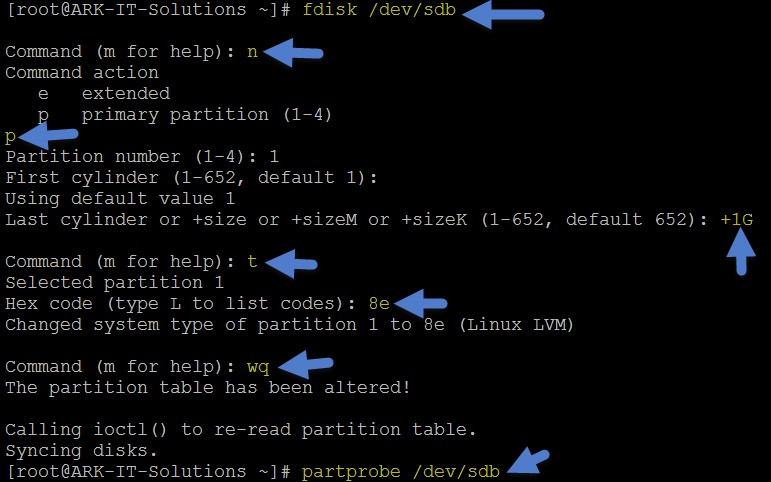
LVM gives you more flexibility than just using normal hard drive partitions:

* + Use any number of disks as one big disk.
  + Have logical volumes stretched over several disks.
  + Create small logical volumes and resize them "dynamically" as they get filled up.
  + Resize logical volumes regardless of their order on disk. It does not depend on the position of the LV within VG, there is no need to ensure surrounding available space.
  + Resize/create/delete logical and physical volumes online. File systems on them still need to be resized, but some (such as ext4) support online resizing.
  + Online/live migration of LV being used by services to different disks without having to restart services.
  + Snapshots allow you to back up a frozen copy of the file system, while keeping service downtime to a minimum.
  + Support for various device-mapper targets, including transparent file system encryption and caching of frequently used data.



To make LVM’s we will first create physical volumes, we will combine all the PV’s into the volume group and top of the VG’s we will create LVM’s.

Let’s create Partitions



**[root@ARK-IT-Solutions ~]# fdisk /dev/sdb**

Command (m for help): **n**

Command action e extended

p primary partition (1-4)

**p**

Partition number (1-4): 1

First cylinder (1-652, default 1): Using default value 1

**Last cylinder or +size or +sizeM or +sizeK (1-652, default 652): +1G**

Command (m for help): **t**

Selected partition 1

Hex code (type L to list codes): **8e**

Changed system type of partition 1 to 8e (Linux LVM)

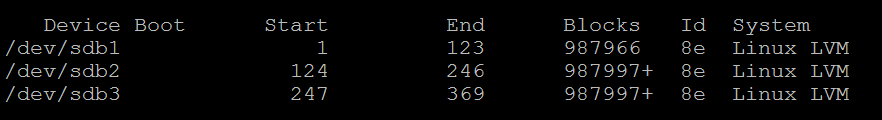
Command (m for help): **wq**

The partition table has been altered!

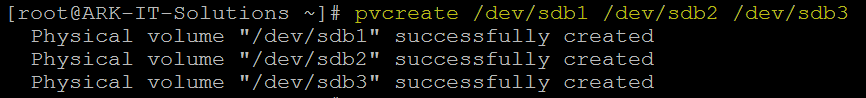
Calling ioctl() to re-read partition table. Syncing disks.

[root@ARK-IT-Solutions ~]# **partprobe /dev/sdb**

Repeat the above step for remain partition creations.



Creating Physical volumes

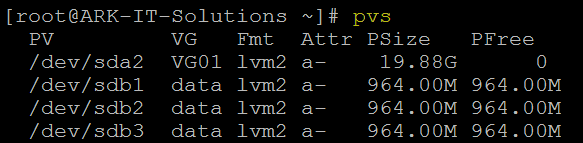


# pvcreate /dev/sdb1 /dev/sdb2 /dev/sdb3 – to create PV’s

# pvscan - to see if any PV’s are there

# pvdisplay - to see PV properties, attributes of a physical volume

# pvs - to see the PV’s information, produces formatted output about PV’s

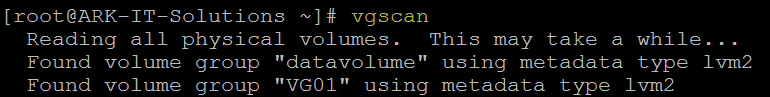


# vgcreate /dev/sdb1 /dev/sdb2 /dev/sdb3

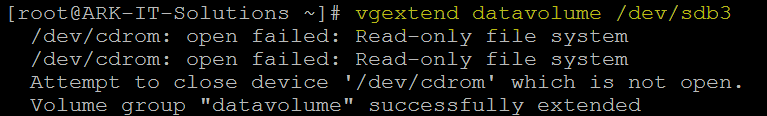


# vgdisplay - It will display all VG properties

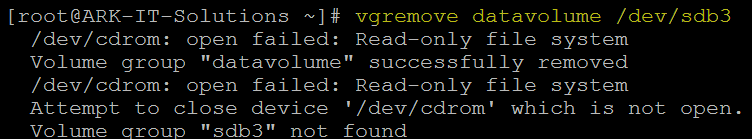
# vgscan - It will scan for all existing volume groups and rebuild caches



# vgextend <pv name> - allows you to add one or more initialized physical volumes to an existing volume group to extend it in size.



# vgremove <vg name> - to delete VG

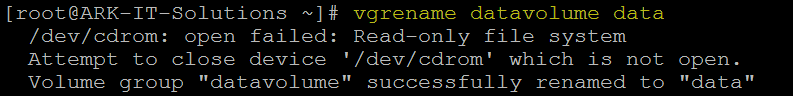


# vgreduce <vg name> <pv name> - to remove PV from VG



**Note:** Volume Group reducing will lead to data loss, we have to take a backup of complete VG then do above step. Do not practice above step in any production environment.

#vgrename <old vg name> <new vg name> - to rename VG name



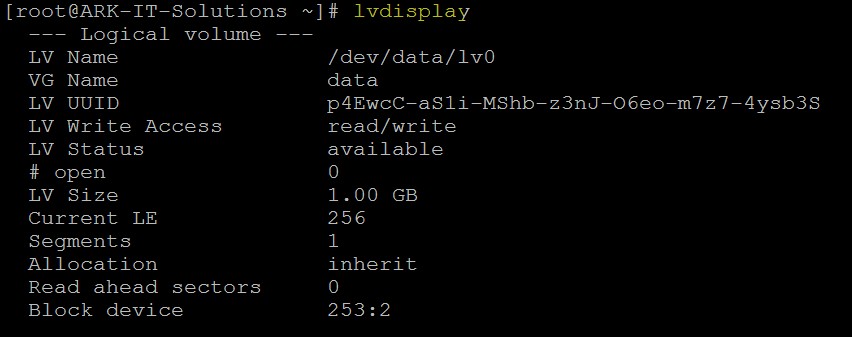
#vgmerge <vg1> <vg2> - To merge two VG’s as one group



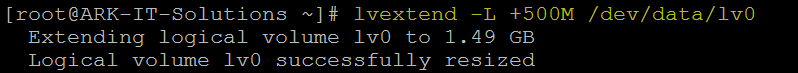
# lvcreate –n <lv name> -L +<size> <vg name> - create a logical volume in an existing VG



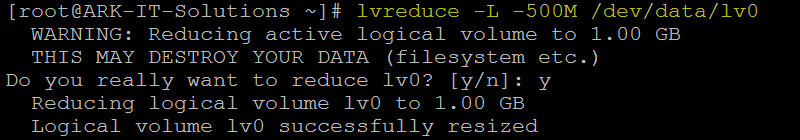
# lvdisplay - allows you to see the attributes of a logical volume like size, read/write status, snapshot information etc.



# lvextend –L +500M <lv path> - to extend the LV space



# lvreduce –L -500M <lv path> - to reduce the LV space



**Note:** reducing the logical volume space will lead to data loss, before reducing the lv space we have to take lv snapshot to restore back to same state. Do not practice this command until unless it’s required.

#### Command options:

lvchange Change the attributes of logical volume(s) lvconvert Change logical volume layout

lvcreate Create a logical volume

lvdisplay Display information about a logical volume lvextend Add space to a logical volume

lvmchange With the device mapper, this is obsolete and does nothing. lvmdisks can List devices that may be used as physical volumes lvmsadc Collect activity data

lvmsar Create activity report

lvreduce Reduce the size of a logical volume lvremove Remove logical volume(s) from the system lvrename Rename a logical volume

lvresize Resize a logical volume

lvs Display information about logical volumes lvscan List all logical volumes in all volume groups

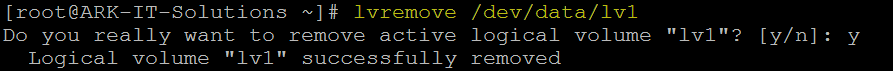
# lvm - to enter into the logical volume mode



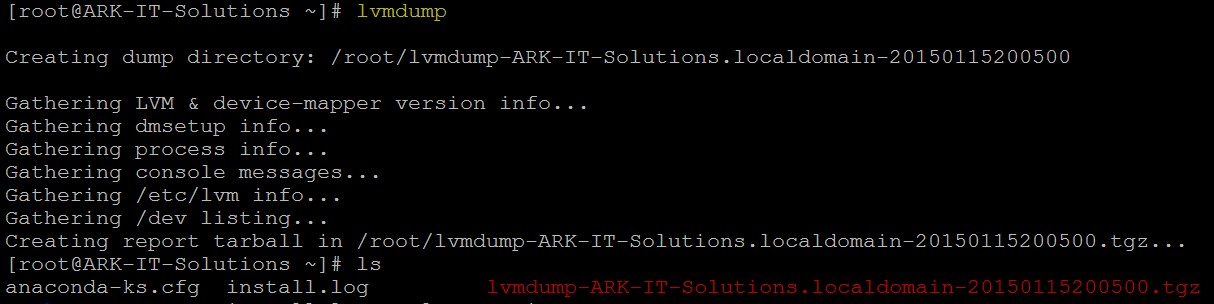
# lvrename - to rename the lv name



# lvremove - to delete the LV



# lvmdump - to collect all the information about LV’s, VG’s and PV’s



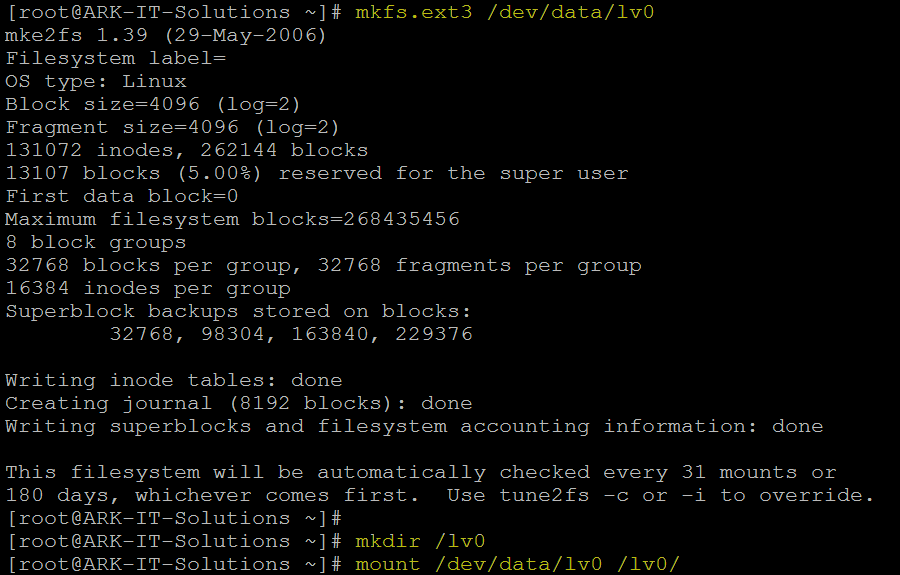
# lvmdiskscan - to see LV’s, VG’s and PV’s info with size

# lvs - to check logical volumes info

# **lvcreate –size 100m –snapshot –name snap /dev/data/lv0 -** creates a snapshot logical volume named /dev/data/snap which has access to the contents of the original logical volume named /dev/data/lv0 at snapshot logical volume creation time. If the original logical volume contains a file system, you can mount the snapshot logical volume on an arbitrary directory in order to access the contents of the file system to run a backup while the original file system continues to get updated.

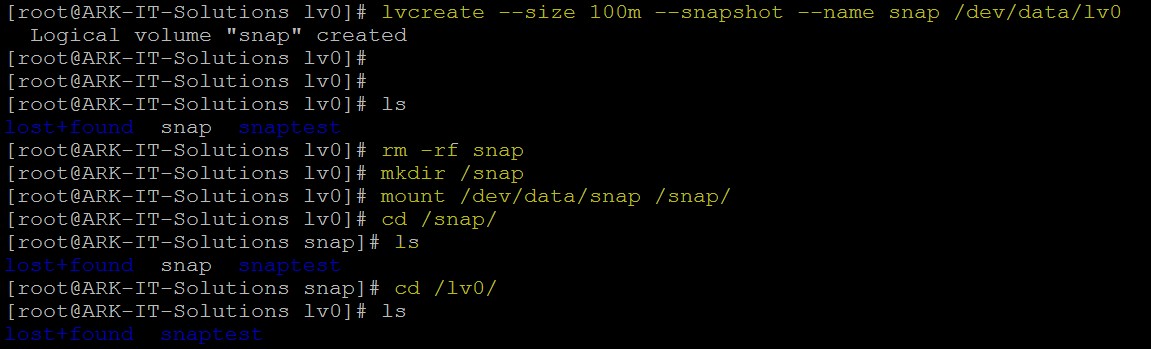


# mkfs.ext3 /dev/data/lv0 - to make file system in lv



# mount <lv path> <mount point> - to mount the lv

To example of snapshot



To restore the snapshot data mount the snapshot to mount point and copy the files from snapshot to original path

In order to delete the logical volume

#umount /lv0

#lvremove /dev/data/lv0

#vgremove data

#pvremove /dev/sdb1

#pvremove /dev/sdb2

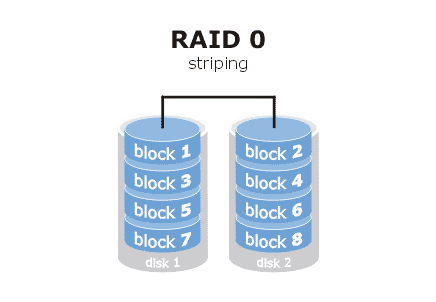
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **19. RAID Redundant array of independent disk** | Document No. | : | RHEL Professional Guide |
| Author | : | Ankam Ravi Kumar |
| Web site | : | <http://ark-library.blogspot.in/> |
| Page No. | : | 77 |

RAID (originally redundant array of inexpensive disks; now commonly redundant array of independent disks) is a data storage virtualization technology that combines multiple disk drive components into a logical unit for the purposes of data redundancy or performance improvement.

In RAID we have different RAID levels

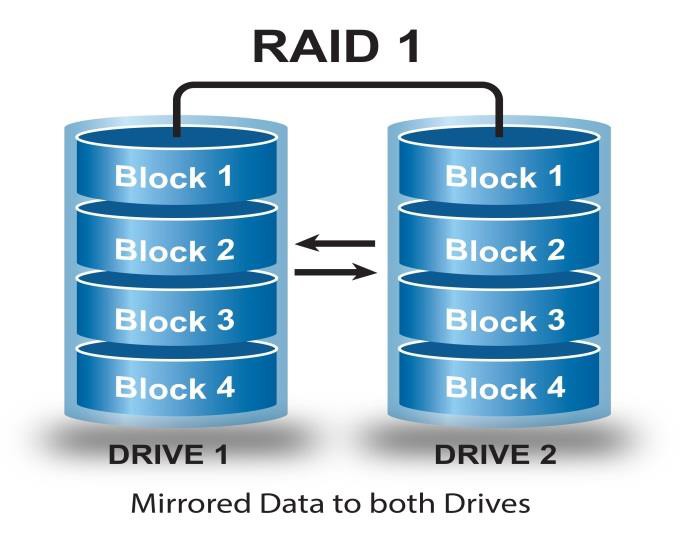
1. Level 0 - striped disk array without fault tolerance
2. Level 1 - mirroring and duplexing
3. Level 2 - error-correcting coding
4. Level 3 - bit-interleaved parity
5. Level 4 - dedicated parity drive
6. Level 5 - block interleaved distributed parity
7. Level 6 - independent data disks with double parity
8. Level 10 - a stripe of mirrors

***RAID Level 0:*** It’s just stripping. RAID Level 0 requires a minimum of 2 drives to implement.



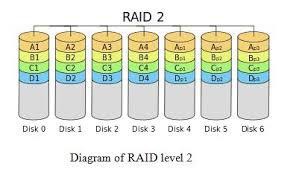
* RAID 0 implements a striped disk array, the data is broken down into blocks and each block is written to a separate disk drive
* I/O performance is greatly improved by spreading the I/O load across many channels and drives
* Best performance is achieved when data is striped across multiple controllers with only one drive per controller
* No parity calculation overhead is involved
* Very simple design
* Easy to implement

***RAID Level 1:*** For Highest performance, the controller must be able to perform two concurrent separate Reads per mirrored pair or two duplicate Writes per mirrored pair. Raid level 1 requires a minimum of 2 drives to implement.



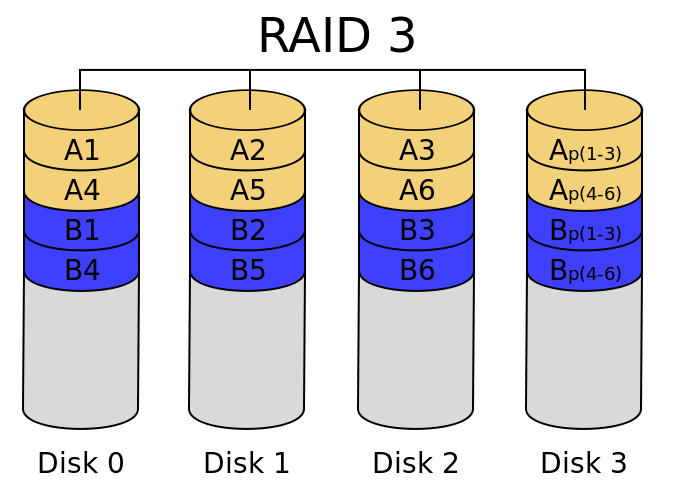
* One Write or two Reads possible per mirrored pair
* Twice the Read transaction rate of single disks, same Write transaction rate as single disks
* 100% redundancy of data means no rebuild is necessary in case of a disk failure, just a copy to the replacement disk
* Transfer rate per block is equal to that of a single disk
* Under certain circumstances, RAID 1 can sustain multiple simultaneous drive failures
* Simplest RAID storage subsystem design

***RAID Level 2:*** Each bit of data word is written to a data disk drive each data word has its Hamming Code ECC word recorded on the ECC disks. On Read, the ECC code verifies correct data or corrects single disk errors.



* "On the fly" data error correction
* Extremely high data transfer rates possible
* The higher the data transfer rate required, the better the ratio of data disks to ECC disks
* Relatively simple controller design compared to RAID levels 3,4 & 5

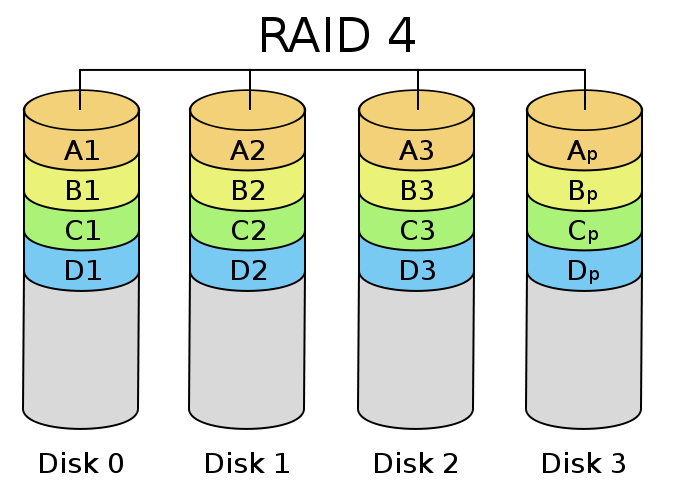
***RAID Level 3:*** Byte-level striping with dedicated parity, data block is subdivided ("striped") and written on the data disks. Stripe parity is generated on Writes, recorded on the parity disk and checked on Reads. Requires minimum 3 disks to implement



### Very high Read data transfer rate

* Very high Write data transfer rate
* Disk failure has an insignificant impact on throughput
* Low ratio of ECC (Parity) disks to data disks means high efficiency

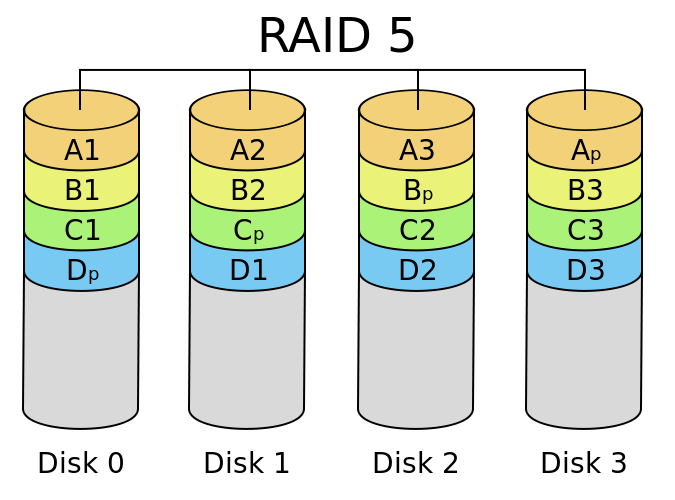
***RAID Level 4:*** Block-level striping with dedicated parity. Each entire block is written onto a data disk. Parity for same rank blocks is generated on Writes, recorded on the parity disk and checked on Reads. Requires minimum 3 disks



### Very high Read data transaction rate

* Low ratio of ECC (Parity) disks to data disks means high efficiency
* High aggregate Read transfer rate

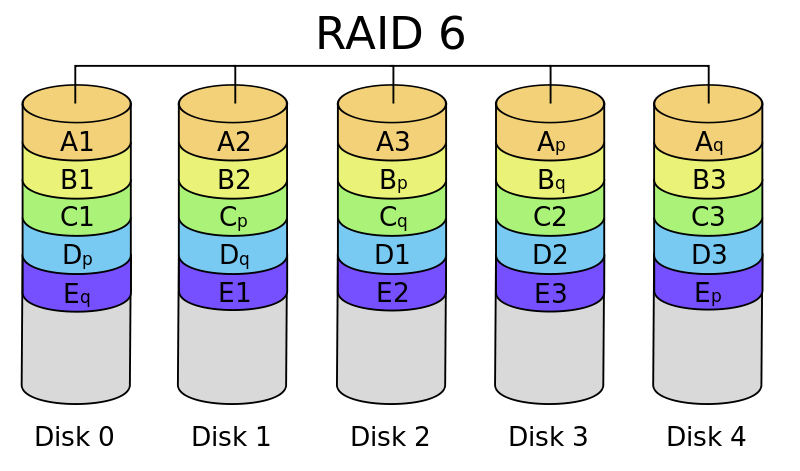
***RAID Level 5:*** Block-level striping with distributed parity. Each entire data block is written on a data disk; parity for blocks in the same rank is generated on Writes, recorded in a distributed location and checked on Reads. Requires minimum 3 disks to implement



### Highest Read data transaction rate

* Medium Write data transaction rate
* Low ratio of ECC (Parity) disks to data disks means high efficiency
* Good aggregate transfer rate

***RAID Level 6:*** Block-level striping with double distributed parity. Two independent parity computations must be used in order to provide protection against double disk failure. Two different algorithms are employed to achieve this purpose. Requires minimum 4 disks implement



### RAID 6 is essentially an extension of RAID level 5 which allows for additional fault tolerance by using a second independent distributed parity scheme (dual parity)

* Data is striped on a block level across a set of drives, just like in RAID 5, and a second set of parity is calculated and written across all the drives; RAID 6 provides for an extremely high data fault tolerance and can sustain multiple simultaneous drive failures
* RAID 6 protects against multiple bad block failures while non-degraded
* RAID 6 protects against a single bad block failure while operating in a degraded mode
* Perfect solution for mission critical applications

***RAID Level 10:*** Disks within the group are mirrored and groups are stripped, required minimum 4 disks to implement



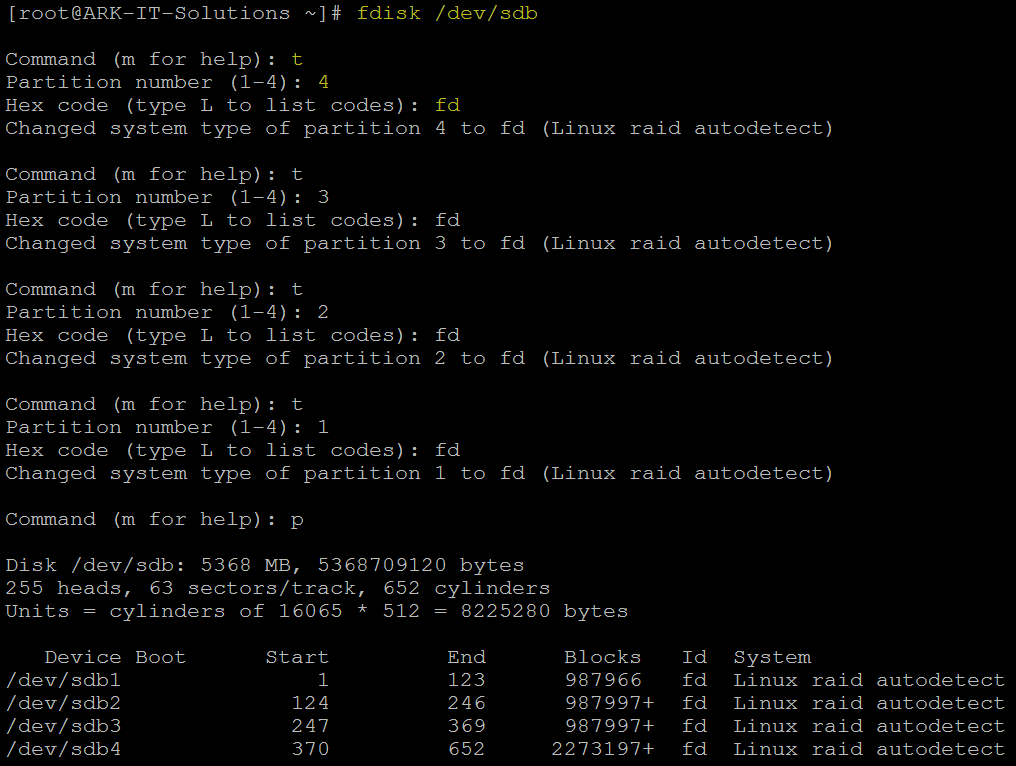
### RAID 10 is implemented as a striped array whose segments are RAID 1 arrays

* RAID 10 has the same fault tolerance as RAID level 1
* RAID 10 has the same overhead for fault-tolerance as mirroring alone
* High I/O rates are achieved by striping RAID 1 segments
* Under certain circumstances, RAID 10 array can sustain multiple simultaneous drive failures
* Excellent solution for sites that would have otherwise gone with RAID 1 but need some additional performance boost

Implementing the RAID will be two types 1.Software RAID and 2.Hardware RAID Let’s see the difference between those

|  |  |
| --- | --- |
| **SOFTWARE RAID** | **HARDWARE RAID** |
| 1. It will use computer system CPU | 1. It will use its own CPU |
| 2. Low cost compare to H/W RAID | 2.More cost compare to S/W RAID |
| 3. It has data integrity issues due to system crashes | 3.No data integrity issues |
| 4. No write-back cache | 4.It is capable of write-back cache |
| 5. Limited operating system migrations | 5.Can be migrated to any OS type |
| 6. Unprotected at boot (cannot manage or protect data at boot):Drive failure or corrupted data during boot and before the RAID software become active leads to an inoperable system | 6. Protected at boot: No negative impact on data availability  when boot drive has medium errors or fails completely |
| 7. Performance issues will be there | 7.No performance issues compare to S/W RAID |

Add new disks for RAID Creation



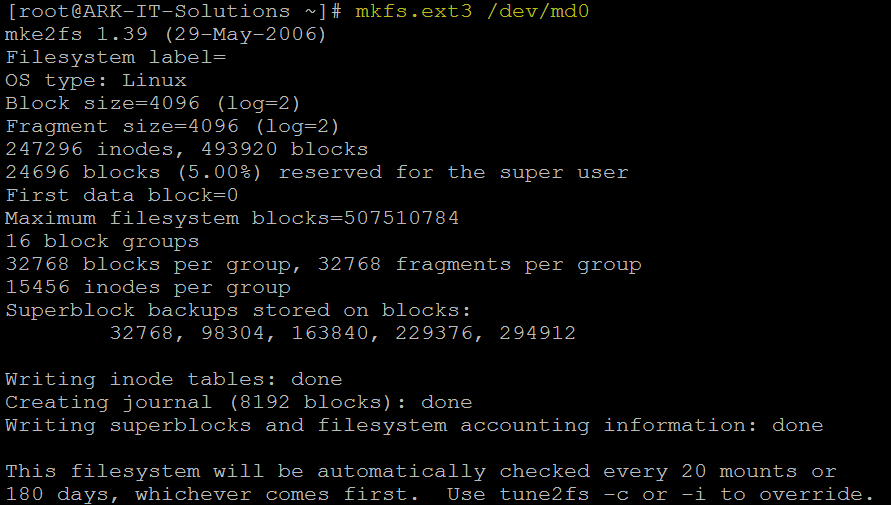
Create partitions using disks, partition type should be Linux raid AutoDetect (fd)

Creating RAID Device

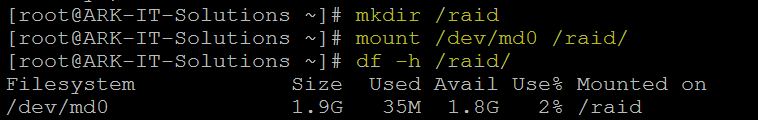
# mdadm --create /dev/md0 --level=5 --raid-disk=3 /dev/sdb1 /dev/sdb2 /dev/sdb3



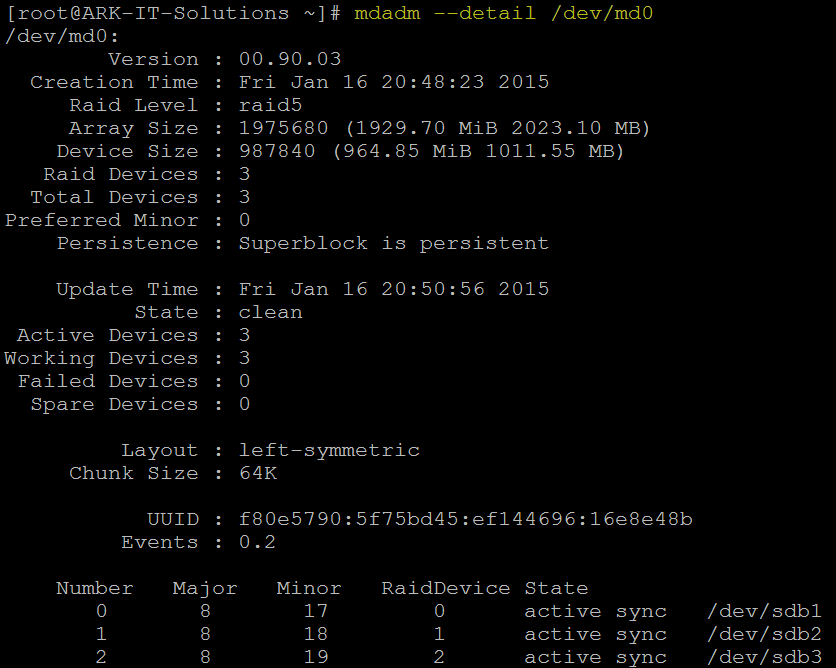
# mkfs.ext3 /dev/md0 - to make file system in RAID device



Mounting and using raid device



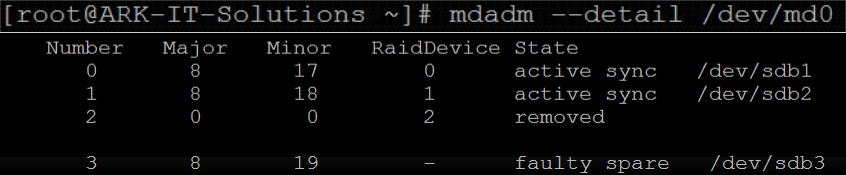
# mdadm –detail /dev/md0 - to see the raid device details



# mdadm /dev/md0 --fail /dev/sdb3 - to do manual failure of disk



To see the detail of raid device and any failed disks



# mdadm /dev/md0 --remove /dev/sdb3 - Removing failed drive from RAID



# mdadm /dev/md0 --add /dev/sdb4 - Adding New disk to RAID Device



To Destroy RAID Device and its related disk completely wipe

# umount /raid - to un mount the file system

# mdadm –stop /dev/md0 - Stopping RAID Device

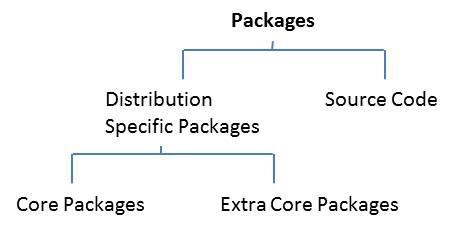
# mdadm –remove /dev/md0 - to remove RAID Device

# fdisk /dev/sdb use d to delete all the disks from OS



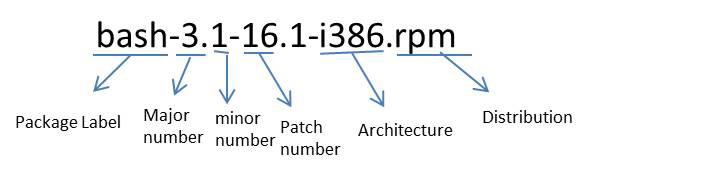
**20. Redhat Package Manager**

RPM is a powerful software manager. It can install, remove, query, and verify the software on your system. Setup of applications is called as package.



***Core Packages:*** These packages are available with you installation media

***Extra Core Packages:*** These extra core packages mean after OS installation, upgrade of package OR release is called as extra core packages.



Above is the explanation about rpm structure

# rpm <options> <package name> - to install, remove, query and upgrade RPM

Options:

-I - install

-v - verbose

-q - query

-e - deleting/erasing

-U - upgrade

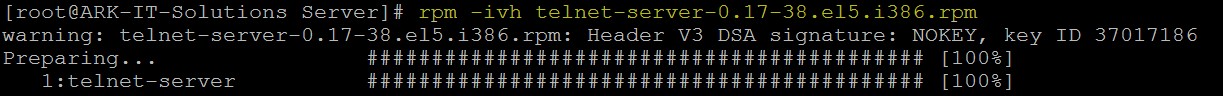
-a - all

-h - hashes ‘#’ s

# rpm –import <key file name> - to import the rpm license key

##### Drawbacks of RPM:

1. Distribution packages
2. Architecture specific

Dependency

# rpm –import <key file path> - to import license key



# rpm –ev <package name> - to delete/erase rpm



# rpm –qa |grep telnet - to query the rpm



# rpmbuild <file name> - to RPM from source code

***Source Code:*** source code is the external packages which we are writing some scripts OR software packages.

***Note:*** when we are installing some of the packages using package manager it will ask you to install dependency packages, installing all the packages manually will eat most of your time. To resolve these types of dependencies we have to configure YUM server.

## YUM: yellowdog updater modified

|  |  |
| --- | --- |
| 3down voteaccepted | If everything was installed via the yum package manager, you can undo previous commands:  # Get all yum commands previously run  yum history list all  # Get the details of the command  yum history info [entry number]  # Undo each command top-down  yum history undo [entry number] |

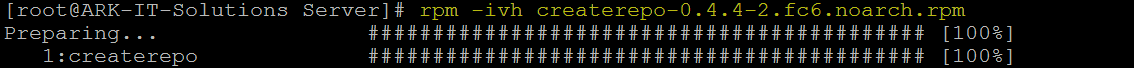
Some of the advantages of YUM include

* Automatic resolution of software dependencies.
* Command-line and graphical versions. YUM can install or upgrade software by using either the command-line version (yum command) or one of two graphical programs:
* Adding and removing software.
* Package updater that only shows software updates available from RHN.
* Multiple software locations at one time. YUM can be configured to look for software packages in more than one location at a time.
* Ability to specify particular software versions or architectures.

YUM downloads software from repositories located over the network, either on the local network or over the Internet. The files, including the RPM package files, in these repositories are organized in a specific way so that they can be found by the YUM client.

Configure YUM server we have to install createrepo RPM in server

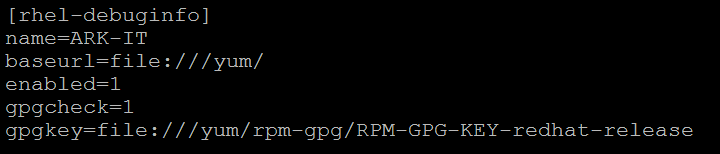
# rpm -ivh createrepo-0.4.4-2.fc6.noarch.rpm



Copy all the RPM packages to one location local path (in this case I have copied to /yum location)

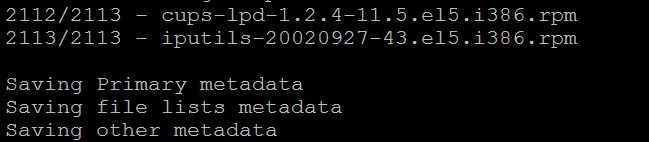
Edit the file

# vi /etc/yum.repos.d/rhel-debuginfo.repo

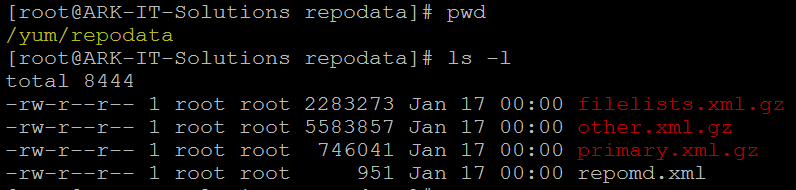


Save the file and exit Create a repository

# createrepo –v /yum/ - to create a repository database

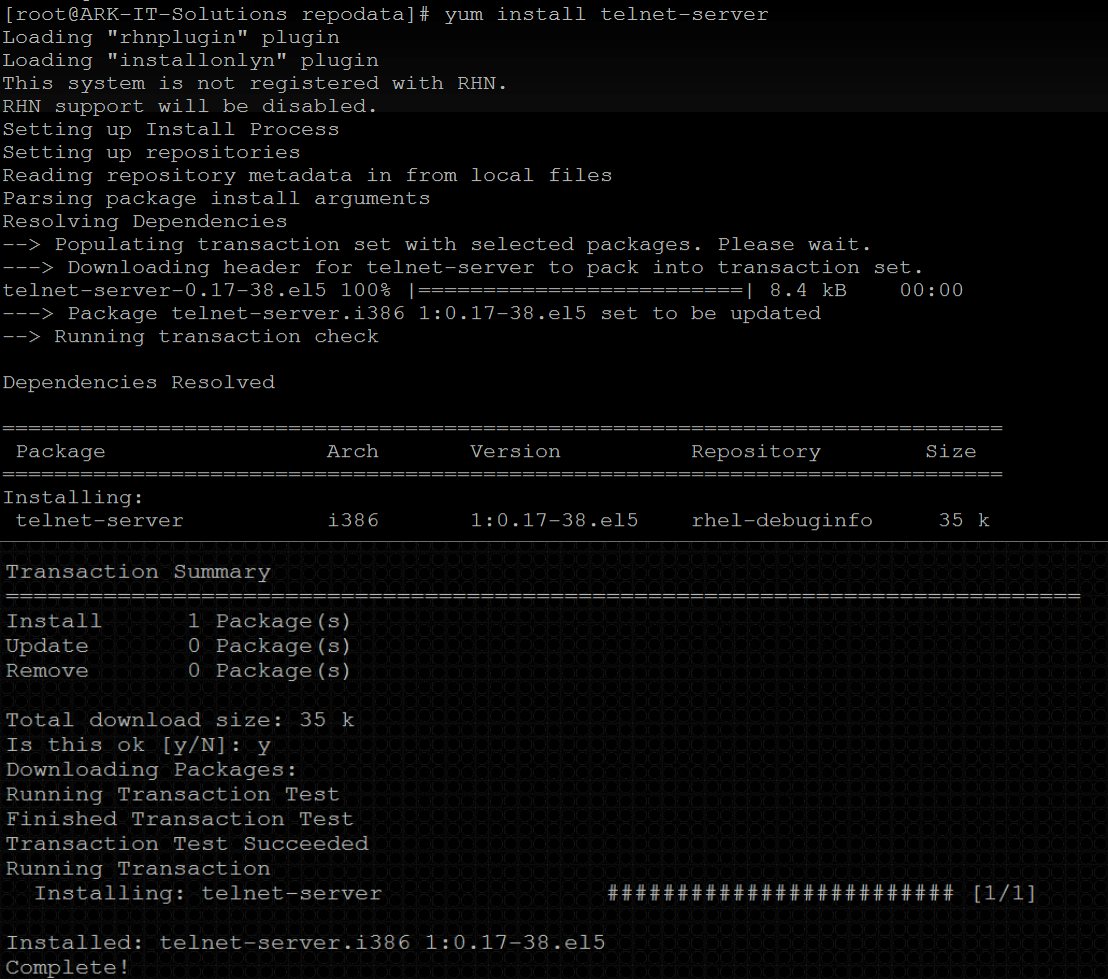


Repo data path (repository files)



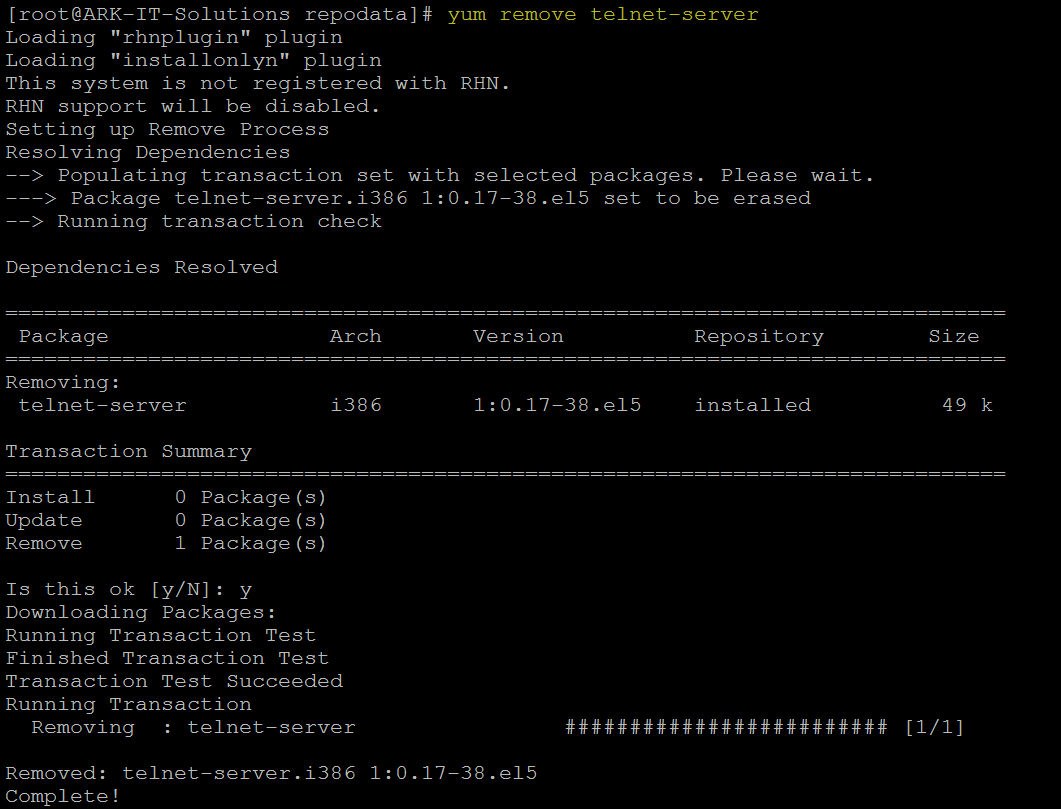
# yum list all - to check packages are fetching from YUM server

# yum install <package name> - to install packages



Like above all the packages and there dependencies will be installed.

# yum remove <package name> - to remove package its dependencies



# yum clean all - it will clean all the cache data of yum server

# yum update <package name> - to update/upgrade mentioned package

If you want make this YUM as centralized server for all the local domain servers. We have to share this using FTP and HTTP protocol.

##### SERVER SIDE

[rhel-debuginfo] name=ARK-IT baseurl=ftp://yum/ enabled=1 gpgcheck=1

gpgkey=ftp://yum/rpm-gpg/RPM-GPG-KEY-redhat-release

##### CLIENT SIDE

[rhel-debugino] Name=ARK-IT

Baseurl=ftp://SERVERIP/PATH Enabled=1

Gpgcheck=0

**21. Networking**

A computer network or data network is a telecommunications network that allows computers to exchange data. In computer networks, networked computing devices pass data to each other along data connections. Data is transferred in the form of packets.

##### Configuration Files

|  |  |
| --- | --- |
| **File** | **Description** |
| /etc/resolve.conf | List DNS servers for internet domain name resolution. |
| /etc/hosts | Lists hosts to be resolved locally (not by DNS). |
| /etc/nsswitch.conf | List order of host name search. Typically look at local files, then NIS server, then DNS server. |
| /etc/sysconfig/network-scripts/ifcfg-device | Specify TCP network information. |

# mii-tool - Media in dependent interface tool it will check for the connectivity, speed of device.

# system-config-network - to configure the IP Address in CLI and GUI

# setup - to configure the IP and DNS, it will work in CLI and GUI

# ifconfig - to see the IP details

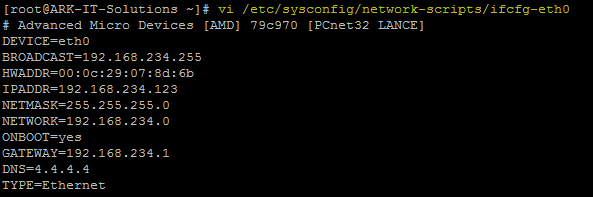
# ifdown <ethernet> - it will bring down the interface

# ifup <Ethernet> - it will bring UP the interface

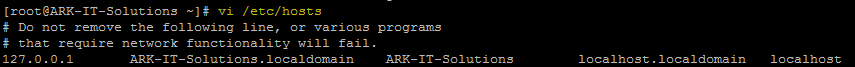
To Assign Temporary IP Adress

# ifconfig <device name> <IP Address> <Netmask> <Gateway> - this will assign a IP temporarily after reboot it will not be there.

# vi /etc/sysconfig/network-scripts/ifcfg-eth0 – to assign permanent IP Address



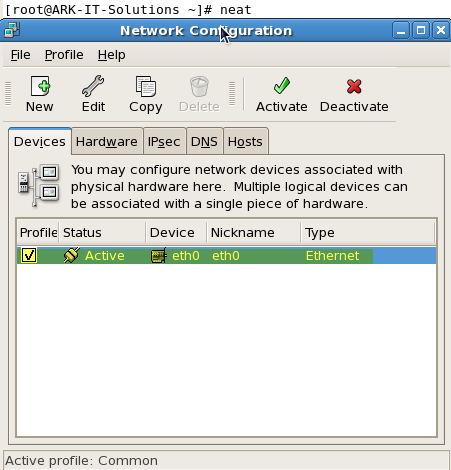
# vi /etc/hosts - to add entry in this file follow as below



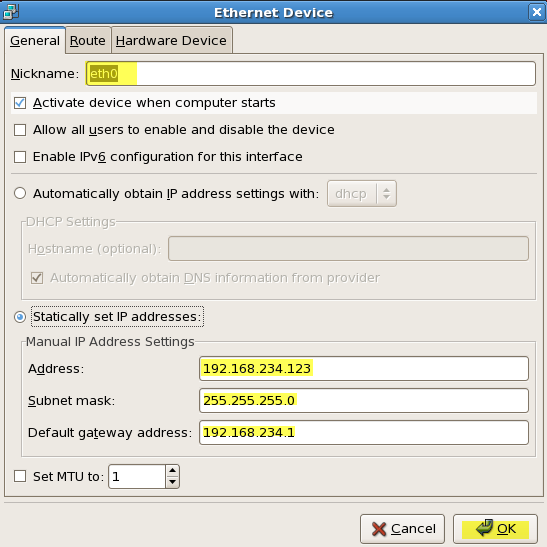
# hostname - to add temporary Hostname and see the hostname

## Using NEAT Command

# neat - it is used to assign the IP Address in GUI mode. (It will not work in CLI

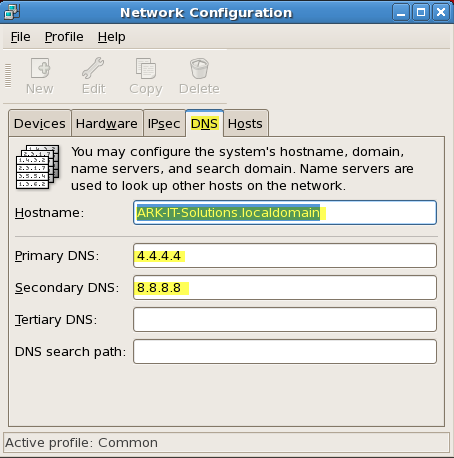


Double Click on the Device name



Nickname: Provide the device name Address: <Enter IP Address> Subnet mask: <Enter Netmask> Gateway: <Enter Gateway>

Click on OK



Using DNS tab you can provide Hostname: <Enter FQDN>

Primary DNS: <Enter Primary DNS Address> Secondary DNS: <Enter Primary DNS Address>

Using neat we can also create virtual network devices

# service network restart - to restart the network service

# service network reload - to reload the network configuration settings

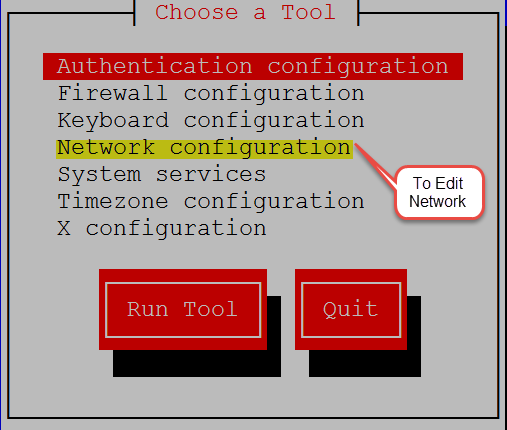
# service network start/stop – to start and stop the service

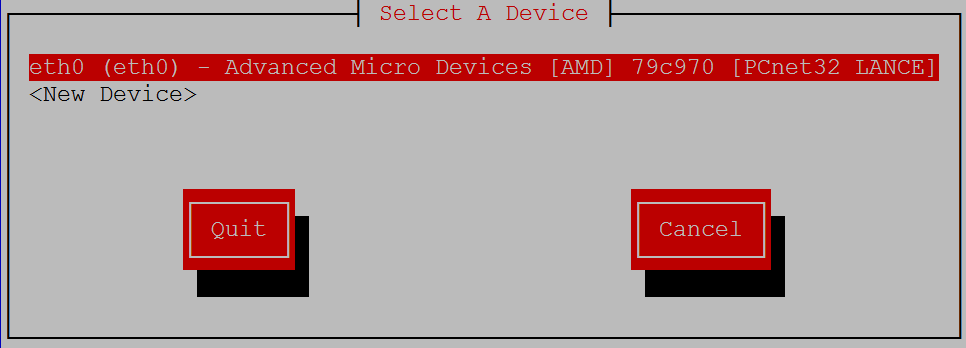
## Using Setup Tool

Using Setup command you can configure

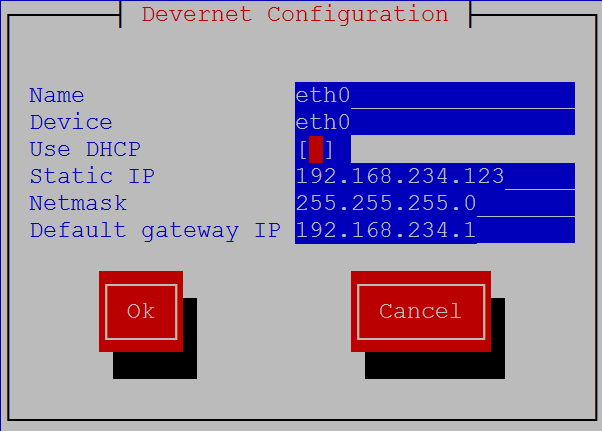
1. Authentication Configuration
2. Firewall Configuration
3. Keyboard configuration
4. Network configuration
5. System services
6. Time zone configuration
7. X configuration

In this topic we are going to talk about Network configuration

# setup - to enter into the setup tool 

Select Network Configuration and hit Enter (click on Run Tool) 

After entering into the Network configuration all the Network devices (Ethernet cards) you can able to see. Select the card which you want to modify the settings and hit Enter.



Provide all the required information such as IP Address, Netmask and Gateway Address. Click on OK

Then restart/reload your network service so that it will affect the new configuration changes

**22. Kick Start Installation and Configuration**

#### What is Kickstart?

Installing Red Hat based systems from physical or virtual media is easy enough when there are only one or two hosts to install. Beyond this, when an administrator needs to setup several Linux systems, Kickstart can provide a relatively easy way to accomplish mass deployment and can be totally automated. It is easy to have several different Kickstart configurations ready to go, each having a different deployment configuration.

Most large companies use this technique to deploy a Red Hat base server image that can then be customized as required. What makes Kickstart even easier is that an administrator can configure Kickstart with nothing more complex than a web server, installation media and at least one Kickstart configuration file. **The only other requirement is an active DHCP server so that new clients waiting to be kick started can reach the Kickstart server, having obtained a valid DHCP address. Kickstart can also use other installation mediums, such as NFS and even local media if desired.**

Just we can say un-attended installation process, this can be configured using

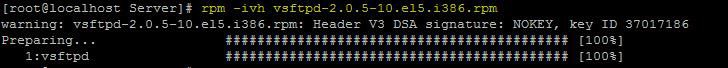
* 1. NFS
  2. FTP
  3. HTTP Prerequisites: -
     + Required installation media
     + Configure FTP site to share packages
     + Configure DHCP server

To copy all the packages from installation media mount the DVD to mount point

# mount /dev/dvd /media

Install FTP packages

# rpm -ivh vsftpd-2.0.5-10.el5.i386.rpm



Copy the data from /media to /var/ftp/pub/

# cp –Rv /media/\* /var/ftp/pub

Start the ftp service

# service vsftpd restart; chkconfig vsftpd on



Stop firewall and SELinux security for now.

# service iptables stop

# vi /etc/selinux/config

In 6th line change from enforcing to disabled.

#### SELINUX=disabled

Save & exit

Install and start the DHCP service

# rpm -ivh dhcp-3.0.5-3.el5.i386.rpm



After installing the dhcp package DHCP sample configuration file will be available in

/usr/share/doc/dhcp\*/dhcpd.conf

Copy above sample file to /etc/dhcpd.conf then modify as per your requirement.

# cat dhcpd.conf.sample > /etc/dhcpd.conf Edit the main config file now

# vi /etc/dhcpd.conf

# cat /etc/dhcpd.conf

#--------------------------Configuration File Starting----------------# ddns-update-style interim;

ignore client-updates;

subnet 192.168.234.0 netmask 255.255.255.0 {

# --- default gateway

option routers 192.168.0.1;

option subnet-mask 255.255.255.0;

option nis-domain "domain.org"; option domain-name "domain.org"; option domain-name-servers 192.168.1.1;

option time-offset -18000; # Eastern Standard Time

# option ntp-servers 192.168.1.1;

# option netbios-name-servers 192.168.1.1;

# --- Selects point-to-point node (default is hybrid). Don't change this unless

# -- you understand Netbios very well

# option netbios-node-type 2;

range dynamic-bootp 192.168.234.28 192.168.234.50;

default-lease-time 21600;

max-lease-time 43200;

# we want the nameserver to appear at a fixed address host ns {

next-server localhost.localdomain; hardware ethernet 00:0C:29:07:8D:6B; fixed-address 192.168.234.123;

}

}

# --------------------- Config file END -------------#

# service dhcpd restart



Copy the below sample configuration file

In this case i have created a file in /var/ftp/pub/rhel5.cfg

#sample KickStart File Start install

text

url --url=ftp://192.168.234.132/pub/ key --skip

lang en\_US.UTF-8 keyboard us

network --onboot yes --device eth0 --bootproto dhcp reboot

rootpw redhat

firewall --disabled

authconfig --enableshadow --enablemd5 selinux --disabled

timezone America/New\_York

bootloader --location=mbr --append="rhgb quiet"

clearpart --all

part / --fstype ext3 --size=4000 part /boot --fstype ext3 --size=100 part swap --size=2000

%packages --nobase

@core

%post

#END

Then boot the station1 with installation media (RHEL5/6)



When it will boot with CD/DVD enter

Boot: linux ks=ftp://192.168.234.132/pub/rhel5.cfg Installation will continue and it will complete automatically.

**23. Quota Management**

Disk quotas are commonly used by ISPs, by Web hosting companies, on FTP sites, and on corporate file servers to ensure continued availability of their systems.

Quotas are used to limit a user’s or a group of user’s ability to consume disk space. This prevents a small group of users from monopolizing disk capacity and potentially interfering with other users or the entire system.

#### Soft limit

this is the maximum amount of space a user can have on that partition. If you have set a grace period, this will act as an alarm. The user will then be notified she is in quota violation. If you have set a grace period, you will also need to set a hard limit. A grace period is the number of days a user is allowed to be above the given quota. After the grace period is over, the user must get under the soft limit to continue. By default grace period have seven days limits.

#### Hard limit

hard limits are necessary only when you are using grace periods. If grace periods are enabled, this will be the absolute limit a user can use. Any attempt to consume resources beyond this limit will be denied. If you are not using grace periods, the soft limit is the maximum amount of available space for each user.

#### Grace Periods

Linux has provided the default of seven days for both inode and block usage. That is, a user may exceed the soft limit on either resource for up to seven days. After that, further requests by that user to use files will be denied.

To do quota we will use edquota, repquota and quotacheck tools to create modify and report quota management.

Quota can be applied to users and groups, block size and inode number. We have to install/verify quota packages is installed.

# rpm –qa |grep quota - to verify quota rpm is installed or not



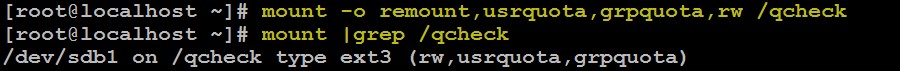
Now open /etc/fstab to open quota

/dev/sdb1 /qcheck ext3 defaults,usrquota,grpquota 1 2



Either reboot or remount the file system to enable quota

# mount –o remount,usrquota,grpquota,rw /qcheck



In this case /qcheck is the mount point.

To enable or create a quota file we have to run

# quotacheck –cugm /qcheck

quotacheck command options

-v scans and prints verbose

-c performs a new scan

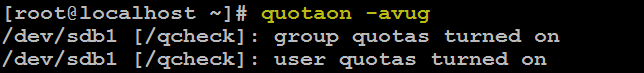
-g scan quota for groups

-m remount the scanned file system

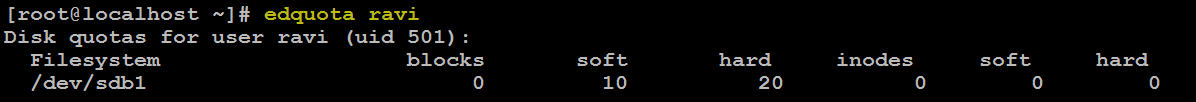
-u scan quota for users

-a Check all quota-enabled, locally-mounted FS Switch on the quota using below command

# quotaon –avug



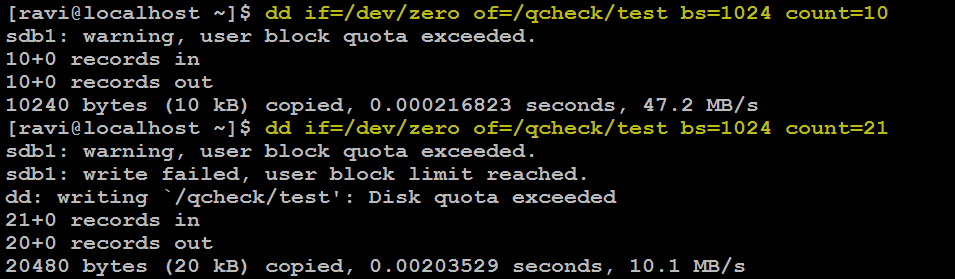
Now quota is on. Add quota soft and hard limits to any user



Now verify writing some dump data (dd command). Login as a ravi user then

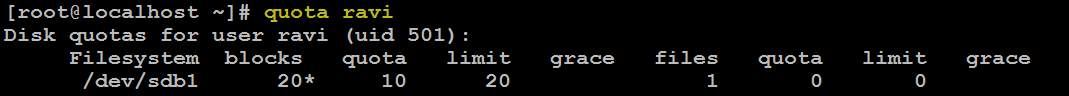
$ dd if=/dev/zero of=/qcheck/test bs=1024 count=10

$ dd if=/dev/zero of=/qcheck/test bs=1024 count=21



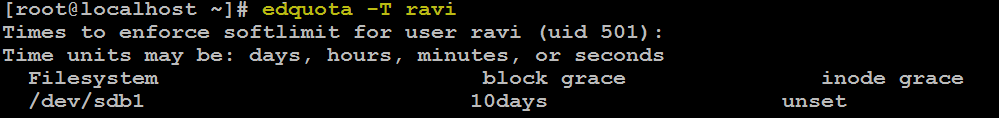
In above screenshot it is showing that warning and exceeded limits for user ravi. To verify the quota settings

# quota –u <user name>



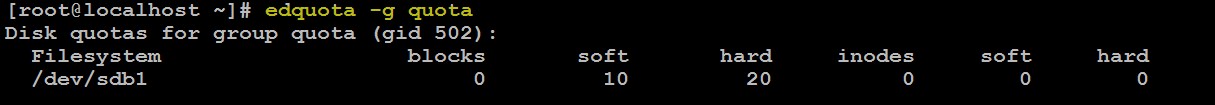
To increase a grace period for a user

# edquota –T <user name>



Enable the quota for a group.

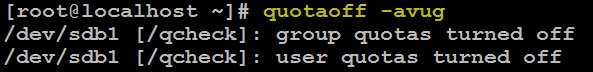
# edquota –g <group name>



In this case quota is group name Removing quota

Switchoff the quota

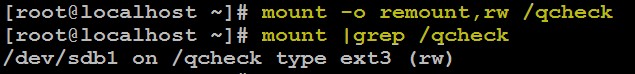
# quotaoff –avug



Then clear the entry in /etc/fstab Remount the partition using



# mount –o remount,rw <mountpoint>



It will clear the quota

**Remote Management**

Remote management can be done using CLI and GUI modes, just we will discuss here about most common tools.

GUI

1. VNC (real/tight)
2. Webmin
3. Rdesktop
4. Xming

CLI

1. Putty

**VNC (Virtual Network Computing)** Servers enables remote desktop access for Linux systems similar to **MSTSC** in windows. Generally Linux administrators doesn’t prefer to use windows access, But sometimes we required to have access remote desktop of Linux. In that case we need to install vnc server on our Linux system.

# Step 1: Install Required Packages

Most of Linux servers doesn’t have desktop installed on their system. So make sure you have installed else use following command to install it.

Yum groupinstall

For CentOS/RHEL 6:

# yum groupinstall "Desktop"

For CentOS/RHEL 5:

# yum groupinstall "GNOME Desktop Environment"

Now install few required packages for **vnc-server**

# yum install pixman pixman-devel libXfont

# Step 2: Install VNC Server

After installing required packages, lets install **vnc-server** in your system. **vnc-server** is available under default yum repositories.

# yum install vnc-server

On **CentOS/RHEL 6**, you will see that **tigervnc-server** package will be installed.

# Step 3: Create User for VNC

Lets’ create few users for connecting through vnc. You can also use existing system users by connecting through vnc, In that case we only need to set **vncpasswd** for that account.

# useradd ravi

# passwd ravi

# useradd ravi1

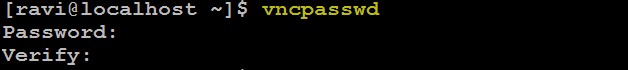
# passwd ravi2

Now set the vnc password for all accounts need to connect through vnc.

# su - ravi

$ vncpasswd

$ exit



# su – ravi1

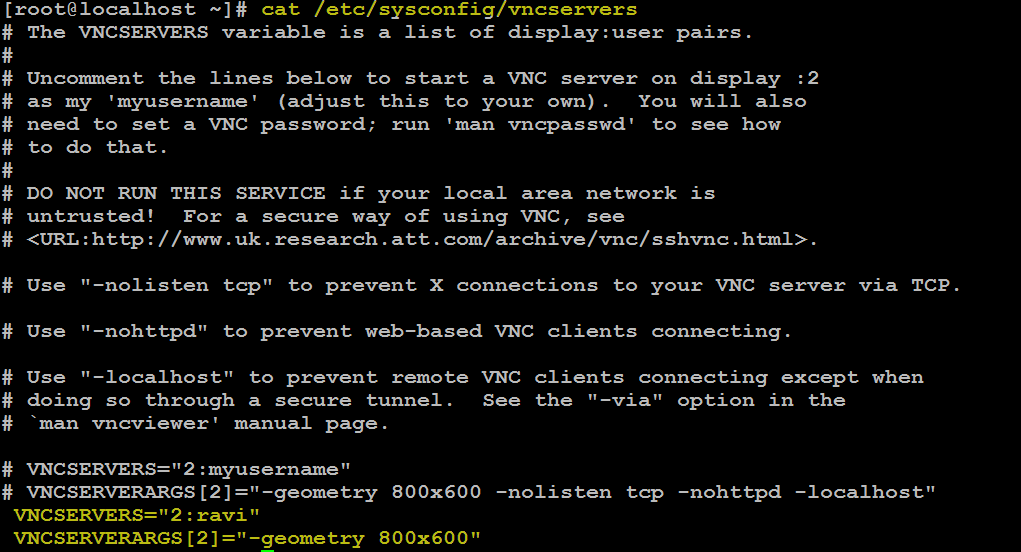
$ vncpasswd

$ exit

# Step 4: Configure VNC Server for Users

# Now edit /etc/sysconfig/vncservers configuration file and add the following to the end of the file.

VNCSERVERS="1:user1 2:user2" VNCSERVERARGS[1]="-geometry 800x600" VNCSERVERARGS[2]="-geometry 1024x768"



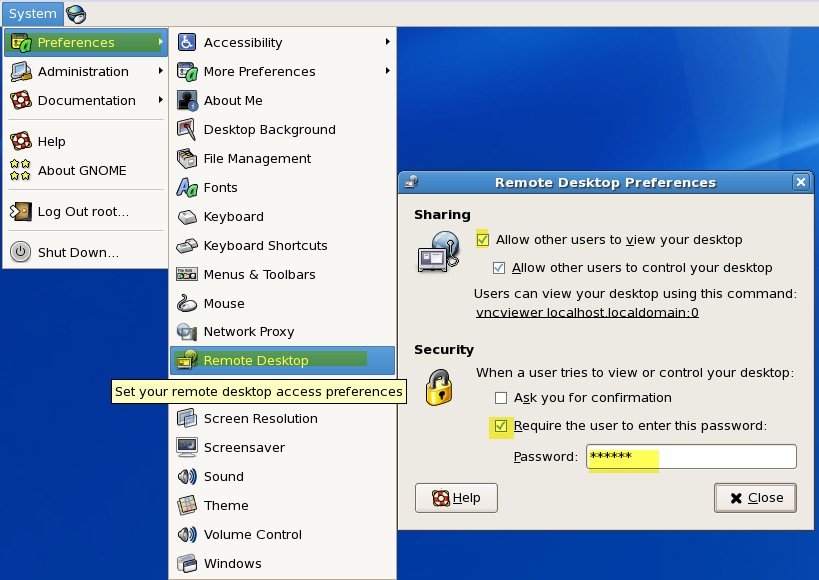
**Where VNCSERVERS is the list of users need to connect, VNCSERVERARGS defined the screen size. Like user ravi have a 800×600 screen size on his client.**

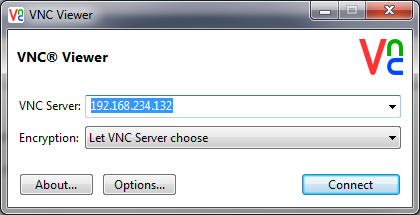
Now start vnc-server service using following command and check the output

# **service vncserver start**

# 

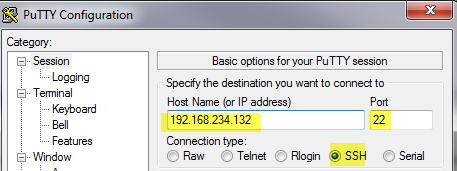
# Step 5: Connect VNC Server using VNC Viewer

If you’re not able to connect to your VNC server then just enable the remote desktop allow from GUI Login into the server GUI mode then go to SystemPreferences Remote Desktop Sharing Allow 

Provide the sharing and keep password for the session. Now try

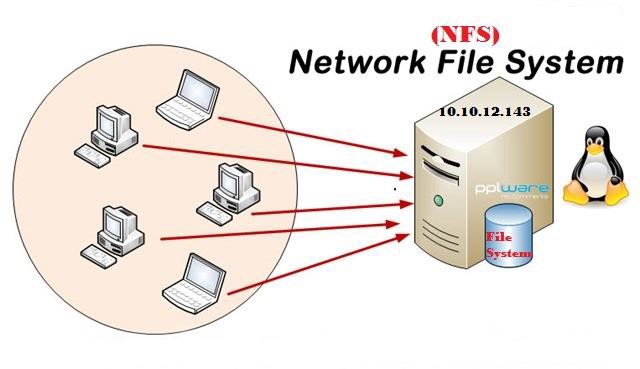
You will get the VNC Session connected

To connect using Putty down load the putty software and run



**25. Network file system (NFS)**

**(NFS) is a distributed file system protocol** originally developed by Sun Microsystems in 1984, **allowing a user on a client computer to access files over a** network much like local storage is accessed.

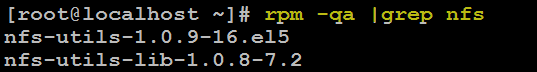


**Service Profile:** nfs and portmap

**Configuration File:** /etc/exports

**Port Number:** 2049 portmap: 111 **Log File:** /var/log/messages **Versions available:** Version 2, 3 & 4

To configure NFS share we have to install below packages



Nfs-utils are installed in my system, if there is no nfs-utils in your server please install

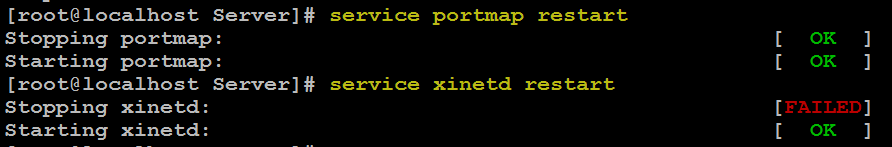
# yum install nfs\*

# yum install xinet\*

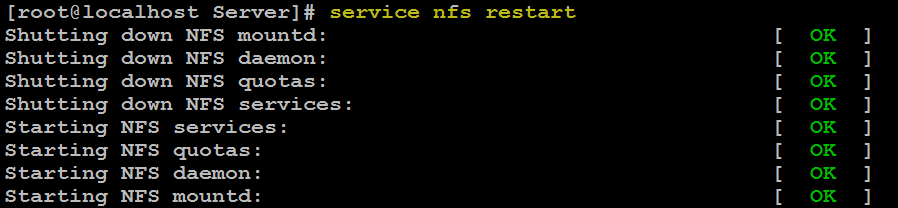
Start NFS, Portmap and Xinetd services

# service portmap restart

# service xinetd restart

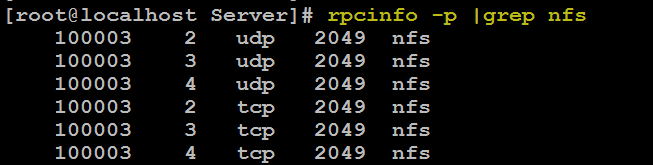


# service nfs restart



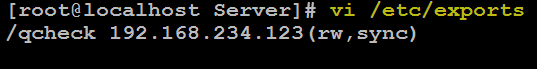
Verify the port nfs is working

# rpcinfo -p |grep nfs



Share the directory or disk using NFS

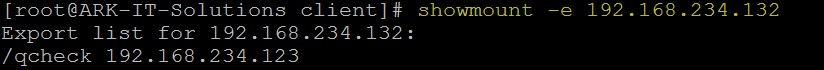
To sharing the Directory/Disk we have edit the /etc/exports and add the entries



#### CLIENT SIDE

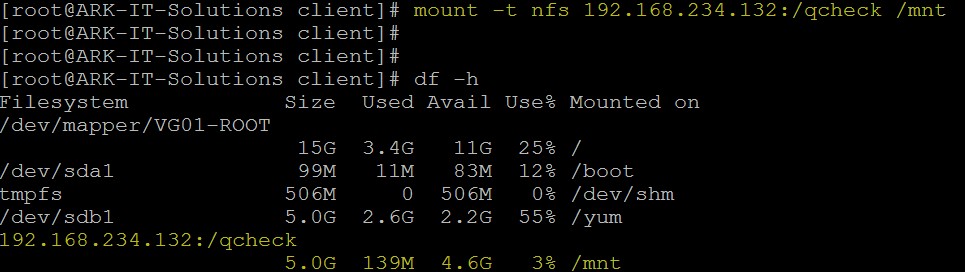
Go to Client machine and check it out share is accessible

# showmount -e 192.168.234.132



Now mount the share to your local mount point

# mount -t nfs 192.168.234.132:/qcheck /mnt



Mounting NFS share as permanent, we have to edit /etc/fstab, if you want restart and verify the mount point



# mount –a - To refresh the mount points

To remove nfs share we have to un-mount the share from client machine

# umount /mnt



### Remove entry from /etc/fstab file:

Here are the most common NFS export techniques and options:

|  |  |
| --- | --- |
| /home/nfs/ 192.168.234.123(rw,sync) | export /home/nfs directory for host with an IP address 192.168.234.123 with read, write permissions, and synchronized mode |
| /home/nfs/ 192.168.234.0(ro,sync) | export /home/nfs directory for network 192.168.234.0 with netmask 255.255.255.0 with read only permissions and synchronized mode |
| /home/nfs/ 192.168.234.123(rw,sync) 192.168.234.124 (ro,sync) | export /home/nfs directory for host with IP 192.168.234.123with read, write permissions, synchronized mode, and also export /home/nfs directory for another host with an IP address 192.168.234.124 with read only permissions and synchronized mode |
| /home/nfs/ 192.168.234.123 (rw,sync,no\_root\_squash) | export /home/nfs directory for host with an IP address 192.168.234.123with read, write permissions, synchronized mode and the remote root user will be treated as a root and will be able to change any file and directory. |
| /home/nfs/ \*(ro,sync) | export /home/nfs directory for any host with read only permissions and synchronized mode |
| /home/nfs/ \*.linuxcareer.com(ro,sync) | export /home/nfs directory for any host within linuxconfig.org domain with a read only permission and synchronized mode |
| /home/nfs/ foobar(rw,sync) | export /home/nfs directory for hostname foobar with read, write permissions and synchronized mode |

### 26.2.1. Reinstalling the Boot Loader

**In many cases, the GRUB boot loader can mistakenly be deleted, corrupted, or replaced by other operating systems.**

The following steps detail the process on how GRUB is reinstalled on the master boot record:

* Boot the system from an installation boot medium.
* Type **linux rescue** at the installation boot prompt to enter the rescue environment.
* Type **chroot /mnt/sysimage** to mount the root partition.
* Type **/sbin/grub-install /dev/hda** to reinstall the GRUB boot loader, where **/dev/hda** is the boot partition.
* Review the **/boot/grub/grub.conf** file, as additional entries may be needed for GRUB to control additional operating systems.
* Reboot the system.

Mount –o remount rw /