# BASIC COMMANDS

## Creating, Removing, Copying, Moving files & Directories

**Creating a file in Linux Using cat command:**

* cat (Concatenate) command is used to create a file and to display and modify the contents of a file.
* **To create a file**

**# cat > filename (say ktfile)**

Hello World

**Ctrl+d (To save the file)**



**To display the content of the file**

# cat filename (say ktfile)



**To append the data in the already existing file**

**# cat >> <filename> # cat >> ktfile**

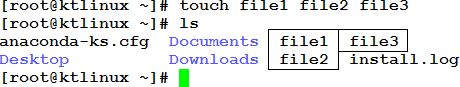
**Ctrl+d** (to save the changes)



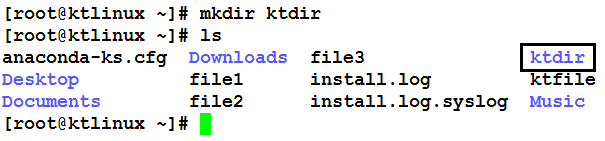
**Creating multiple files at same time using touch command #touch <filename> <filename> <filename>**

**#touch file1 file2 file3**

Note: to check the files use **# ls** command



**Creating a Directory: #mkdir <dir name> #mkdir ktdir**



**Making multiple directories inside a directory**

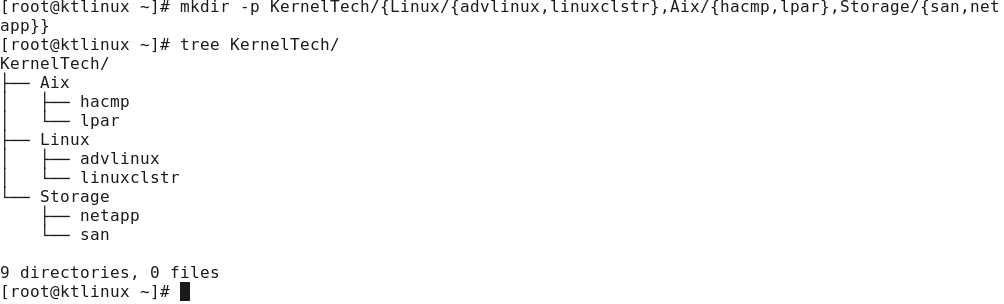
Let us make some directories according to the following architecture in one command.

**Tech**

**Linux Aix Storage**

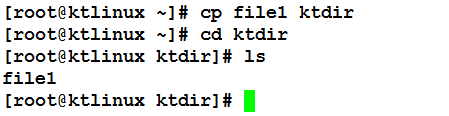
advlinux linuxclstr hacmp lpar san netapp

**#mkdir –p KernelTech/{Linux/{advlinux,linuxclstr},Aix/{hacmp,lpar},Storage/{san,netapp}}**

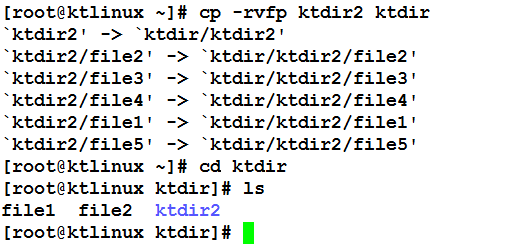
Check it by using **tree** command or **ls –R** command

**Copying files into directory**

**#cp <source filename> <destination directory in which to paste the file> #cp file1 ktdir**

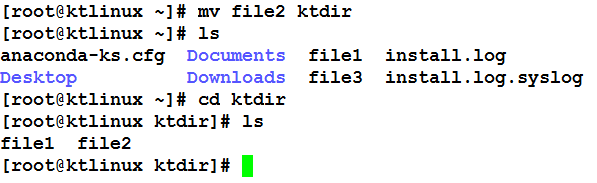


**Copying directories from one location to other # cp –rvfp <dir name> <destination name> #cp –rvfp ktdir2 ktdir**



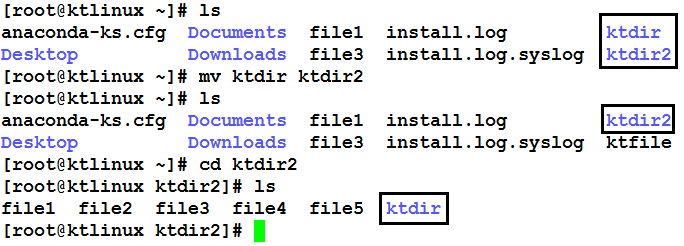
**Moving files from one location to other (cut and Paste)**

**#mv <filename> <Destination directory> #mv file2 ktdir**

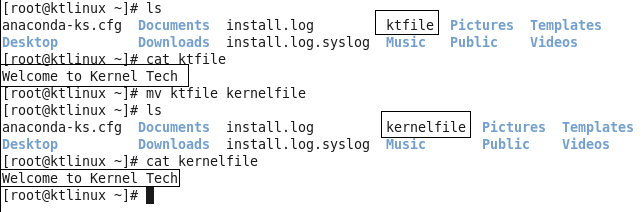


**Moving a Directory from one location to other #mv <dir name> <destination dir name>**

**#mv ktdir ktdir2**

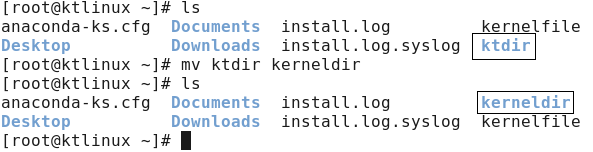


**Renaming a File**

**#mv <old name> <new name> #mv ktfile kernelfile**

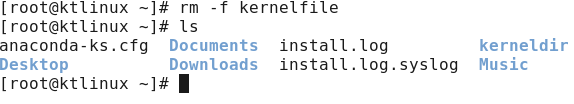
**Renaming a Directory**

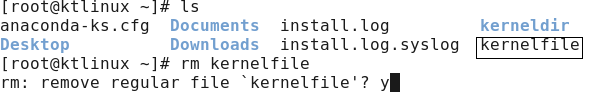
* The procedure and command for renaming the directory is exactly same as renaming a file.

**#mv old name new name #mv ktdir kerneldir**

**Removing a File**

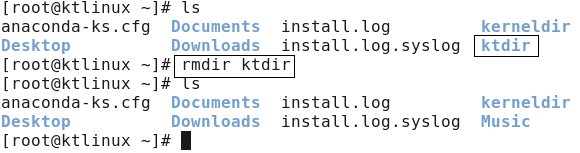
**#rm filename or #rm –f filename (without prompting)**





**Without prompting:**

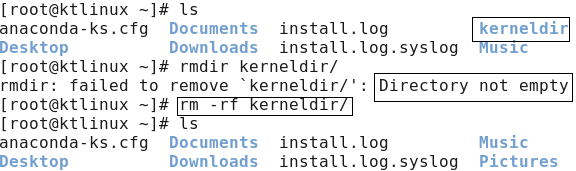
**Removing an Empty directory #rmdir dirname**



**Removing a directory with files or directories inside**

A dir which is having some contents inside it cannot be removed by **rmdir** command. There are two ways to delete the directory with contents.

1. Remove the contents inside the directory and then run **rmdir** command
2. Run **#rm –rf dirname** (where **r** stands for recursive and **f** stands for forcefully.



## VIM EDITOR

**VI Visual display editor**

**VIM Visual display editor improved**

**This is command mode editor for files. Other editors in Linux are emacs, gedit vi editor is most popular**

**It has 3 modes:**

1. **Command Mode**
2. **Insert mode (edit mode)**
3. **extended command mode**

**Note:** When you open the vim editor, it will be in the command mode by default.

In the command mode the cursor’s can be used as h/l/k/j to move cursor left/right/up/down **Insert Mode:**

**i**

**I**

**a A**

**o O**

**To begin insert mode at the cursor position**

**To insert at the beginning of line**

**To append to the next word’s letter To Append at the end of the line**

**To insert a new line below the cursor position To insert a new line above the cursor position**

**Command Mode:**

**gg**

**G**

**w b nw nb u**

**To go to the beginning of the page**

**To go to end of the page**

**To move the cursor forward, word by word To move the cursor backward, word by word To move the cursor forward to n words (5W) To move the cursor backward to n words (5B) To undo last change (word)**

**U**

**Ctrl+R yy nyy p**

**P**

**dw x dd ndd**

**/**

**To undo the previous changes (entire line)**

**To redo the changes To copy a line**

**To copy n lines (5yy or 4yy)**

**To paste line below the cursor position To paste line above the cursor position**

**To delete the word letter by letter (like Backspace) To delete the world letter by letter (like DEL Key) To delete entire line**

**To delete n no. of lines from cursor position(5dd) To search a word in the file**

**Extended Mode: ( Colon Mode)**

**Extended Mode is used for save and quit or save without quit using “Esc” Key with “:”**

**Esc+:w To Save the changes**

**Esc+:q To quit (Without saving)**

**Esc+:wq To save and quit**

**Esc+:w! To save forcefully**

**Esc+wq! To save and quit forcefully**

**Esc+:x To save and quit**

**Esc+:X To give password to the file and remove password Esc+:20(n) To go to line no 20 or n**

**Esc+: se nu To set the line numbers to the file Esc+:se nonu To Remove the set line numbers**

**To open multiple files in vim editor**

#vim –o file1 file2

To switch between files use **Ctrl +w**

**Listing files and directories:**

#ls list the file names

#ls -l long listing of the file

#ls –l filename to see the permissions of a particular file

#ls -al shows the files in ascending order of modification.

#ls p\* All the files start with p.

#ls ?ample Files with any first character and has ample #ls -ld l\* Directory listing only

#ls –ld directory name to see the permissions of a particular directory #ls [ae]\* First character of the filename must be a or e.

# ls [!ae]\* ! Symbol complements the condition that follows. The characters must not be a or e.

#ls [a-m][c-z][4-9] list all the files in specific range

**Types of Files:**

**Symbol**

**-**

**d l b c**

**Type of File**

**Normal file Directory**

**Link file (shortcut)**

**Block file (Harddisk, Floppy disk) Character file (Keyboard, Mouse)**

## Symbolic Link

**There are two types of Links:-**

**1**

**2**

**3**

**Soft Link**

**Size of link file is equal to no. of characters in the name of original file Can be created across the Partition Inode no. of source and link file is different**

**if original file is deleted, link is broken and data is lost**

**SHORTCUT FILE**

**Hard link**

**Size of both file is same**

**Can't be created across the partition Inode no. of both file is same**

**If original file is deleted then also link**

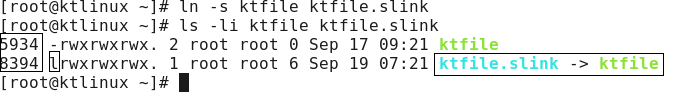
**will contain data BACKUP FILE**

**4**

**5**

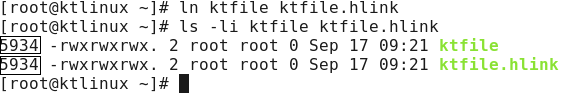
**Creating a soft link:**

**# ln –s <source file> <destination>**



**Creating a Hard link:**

**#ln <source file> <Destination>**



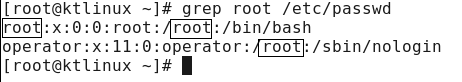
## Regular Expressions, Pipelines & I/O Redirections Grep:

Grep stands for **Global Regular Expression Print.** It is used to pick out the required expression from the file and print the output. If grep is combined with another command it can be used to pick out the selected word, phrase from the output of first command and print it.

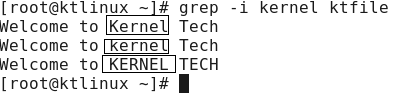
**Examples of Grep:**

Let us pick the information about **root** from the file **/etc/passwd** (/etc/passwd contains information about all the users present in the system)

**#grep root /etc/passwd**

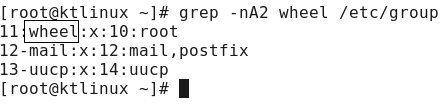


**To avoid case sensitivity of the word (i.e. the word may be uppercase of lowercase) use -i #grep –i kernel ktfile** (lets grep the word **kernel** whether upper of lower case in the file **ktfile**)

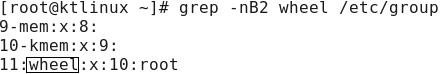


**To display a word and 2 lines after the word:**

**#grep –nA2 wheel /etc/group**



**To display a word and 2 lines after the word: #grep -nB2 wheel /etc/group**



**To display the things except the given word: #grep –v kernel ktfile**



**To display the searched word in color #grep --color root /etc/passwd**

**Combining grep with other commands**

**# cat ktfile | grep –I kernel (pipe | is used to combine to commands) #ls –l |grep –I ktfile**

**# ifconfi g |grep –I eth0**

**Like this we can combine grep with many commands which we will see in later chapters**

# Filter Commands:

* Filter commands are used to filter the output so that the required things can easily be picked up. The commands which are used to filter the output are

## #less #more #head #tail #sort #cut #sed

* **less:-**

The **less** command is used to see the output line wise or page wise. Ex: less /etc/passwd



**Note: -**press **Enter** key to scroll down line by line (or)

Use **d** to go to next page Use **b** to go to previous page

Use **/** to search for a word in the file

Use **v** to go vi mode where you can edit the file and once you save it you will back to less command

## more:-

**more** is exactly same like **less Ex:** #more /etc/passwd

**Note: -**press **Enter** key to scroll down line by line (or) Use **d** to go to next page

Use **/** to search for a word in the file

Use **v** to go vi mode where you can edit the file and once you save it you will back to more command

## head:

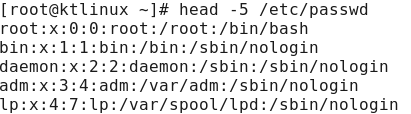
It is used to display the top **10 lines** of the file.

**Ex:# head /etc/passwd**



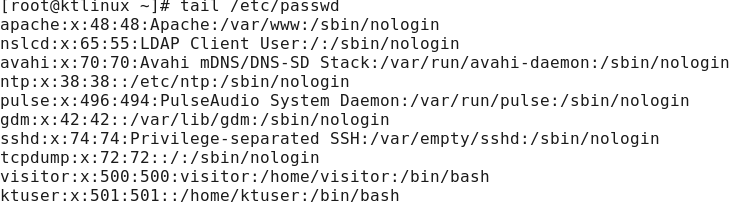
**To display the custom lines**

**#head -n /etc/passwd (where n can be any number)**



## tail:

It is used to display the **last 10** lines of the file #tail /etc/passwd



**To display the custom lines**

#tail -n /etc/passwd (where n can be any number)



## Sort:

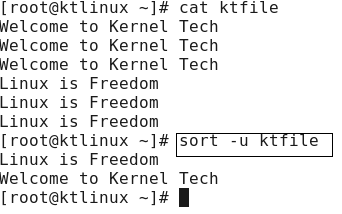
**It is used to sort the output in numeric or alphabetic order #sort filename**



**To sort the file according to numbers #sort –d ktfile or #sort –h ktfile**

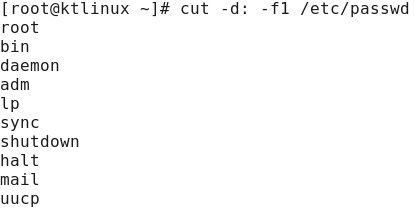


**To remove the duplicate entries from the output #sort –u ktfile**



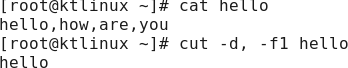
## cut command:

**The cut command is used to pick the given expression (in columns) and display the output. # cut -d -f filename** (where d stands for delimiter ex. : , “ “ etc and f stands for field)



**To delimit spaces and print the field**

**#cut –d “ “ –f1 filename**

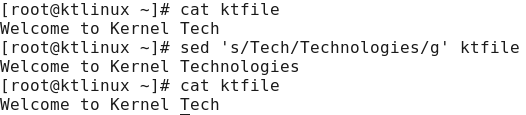
**To delimit commas and print the field #cut –d, -f1 filename**

## sed command:

**sed** stands for **stream editor**, which is used to search a word in the file and replace it with the word required to be in the output

**Note**: it will only modify the output, but there will be no change in the original file.

**#sed ‘s/searchfor/replacewith/g’ filename**



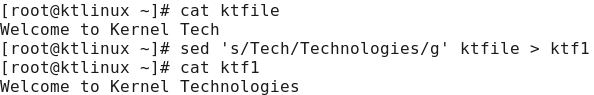
## I/O Redirection:

Redirection is a process where we can copy the output of any command(s), file(s) into a new file. There are two ways of redirecting the output into a file.

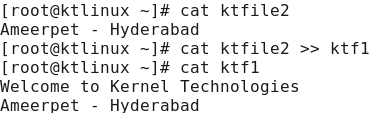
Using **>** or **>> filename** after the command, and Using **tee** command

**Let’s see the > and >> option first Syn: command > new file**

**Note: if the given name of the file is not available a new file will be created automatically. If the file already exists then it will overwrite contents of that file.**



**Appending another output in same the same file**

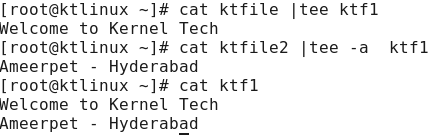
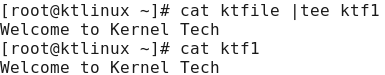


Likewise there are many options where we can use redirections

**Ex:**

**Copying contents of two files in a new file #cat file1 file2 > file3**

## Using tee command:



**The above options of redirections will not display any output, but directly save the output in**

**a file. Using tee command will not only redirect the output to new file but it will also display the output.**

**Syn: cat <filename> | tee <new file name>**

**Note: if the given name of the file (newfile) is not available a new file will be created automatically. If the file already exists then it will overwrite contents of the file.**

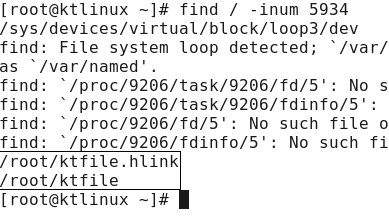
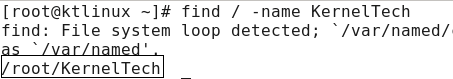
**#cat ktfile |tee ktf1**

**Appending data in the same file using tee command**

**Syn: cat filename |tee –a filename2 #cat ktfile1 | tee –a ktf1**

**Find command:**

**find** command is used to find the files or directory’s path, it is exactly like the find option in



windows where you can search for a file.

**Syntax: find / (under root) –option filename Options that can be used with find command:**

**Option**

**-name**

**-inum**

**-type**

**-user**

**-group**

**Usage**

For searching a file with its name

For searching a file with particular inode number For searching a particular type of file

For files whose owner is a particular user For files belonging to particular group

**Finding a File with name**

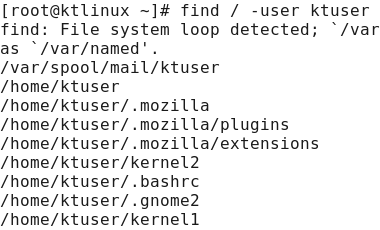
**#find / -name Kernel Tech**

**Finding a file with its inode number**

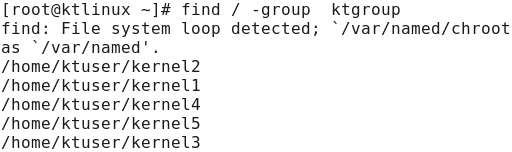
**#find / -inum 5934**

**.**

**Finding the files, whose owner is a user called “ktuser” #find / -user ktuser**



**Finding the files whose group is “ktgroup” #find / -group ktgroup**



# File Permissions:

**Permissions are applied on three levels:-**

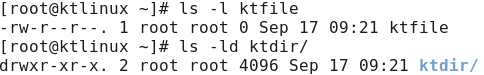
* Owner or User level
* Group level
* Others level

**Access modes are of three types:-**

* r read only
* w write/edit/delete/append
* x execute/run a command

**Access modes are different on file and directory:**

**Files**



**Filetype+permission, links, owner, group name of owner, size in bytes, date of modification, file name**

**Permissions**

**r w x**

**Open the file**

**Write, edit, append, delete file To run a command/shell script**

**Directory**

**'ls' the contents of dir Add/Del/Rename contents of dir To enter into dir using 'cd'**

**Permission can be set on any file/dir by two methods:- 1 Symbolic method (ugo)**

**2 Absolute methods (numbers)**

1. **Symbolic method (ugo):**

* Symbolic mode: General form of symbolic mode is:

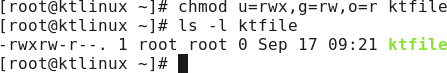
**# chmod [who] [+/-/=] [permissions] file**

who  To whom the permissions to be assigned

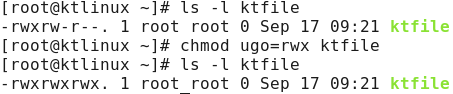
**User/owner (u); group (g); others (o)**

**Example: -**

**Assigning different permissions to the file (user=rwx, group=rw and others=r)**

#chmod u=rwx,g=rw,o=r ktfile (where ktfile is the name of the file)

**Assigning full permission to the file i.e. rwx to all**

#chmod ugo=rwx <file name>

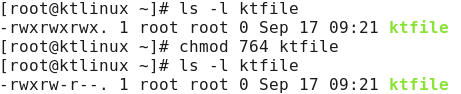
**Likewise you can add or remove permissions from any file for anyone (user group or other)**

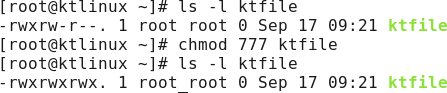
* **#chmod u+x ktfile (Adding execute permission to user only)**
* **#chmod go-wx ktfile (Removing write and execute permissions from group and other)**
* **#chmod go+wx ktfile (Adding write and execute permissions from group and other)**
* **#chmod go=r ktfile (Giving only read permission to group and other)**

1. **Absolute Method (numbers)**

In Absolute method we use numbers instead of using symbols i.e.

* **Read=4**
* **Write=2**
* **Execute=1**

**Assigning different permissions to the file (user=rwx, group=rw and others=r) #chmod 764 ktfile (where 7 means rwx i.e. 4+2+1, rw=6 i.e. 4+2 and 1 indicates x)**

**Assigning full permission to the file i.e. rwx to all #chmod 777 ktfile**

**Likewise you can give different permissions according to your requirement**

**Removing all permissions from others**

#chmod 770 ktfile (where **0** indicates **no** permissions)

**Note: All the above permissions and procedure is same for files and directories.**

**Umask:**

When we create any file using touch, cat or vi commands they get created with default file permissions as stored in umask **(User file creation mask)**.umask is a 4 digit octal number which tells Unix which of the three permissions are to be denied rather than granted. Umask will decide that what should be the default permissions for a file and directory when it is created.

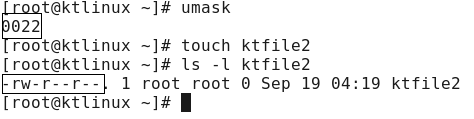
**The default umask value is 0022 #umask**



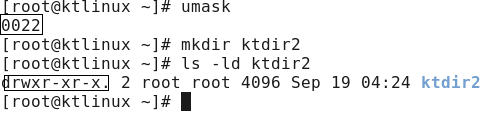
**Calculation of default permissions for file and directory, basing upon the umask value**

**Note:** For a file by default it cannot have the execute permission, so the maximum full permission for a file at the time of creation can be **666** (i.e. 777 -111 = 666), whereas a directory can have full permissions i.e. **777**

* **The full permission for the file 666**
* **Minus the umask value -022**
* **The default permission for file is 644 (rw-,r--,r--)**



* **The full permission for the directory 777**
* **Minus the umask value - 022**
* **The default permission for file is 755 (rwx, r-x, r-x)**

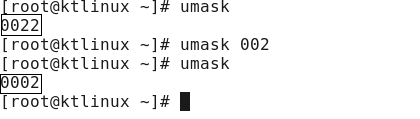


**Modifying the umask value:**

**#umask 002**

The Modified default Permission for a **file** will be **666-002=664** i.e. **rw,rw,r,** and for the

**directory** it will be **777-002=775** i.e. **rwx,rwx,r-x.**



**Note: Create a file and a directory and check for the default permissions.**

**These were the few things amongst the basics; keep working to furnish your basics. After All, “if the foundation is good then only the building can stand still”**