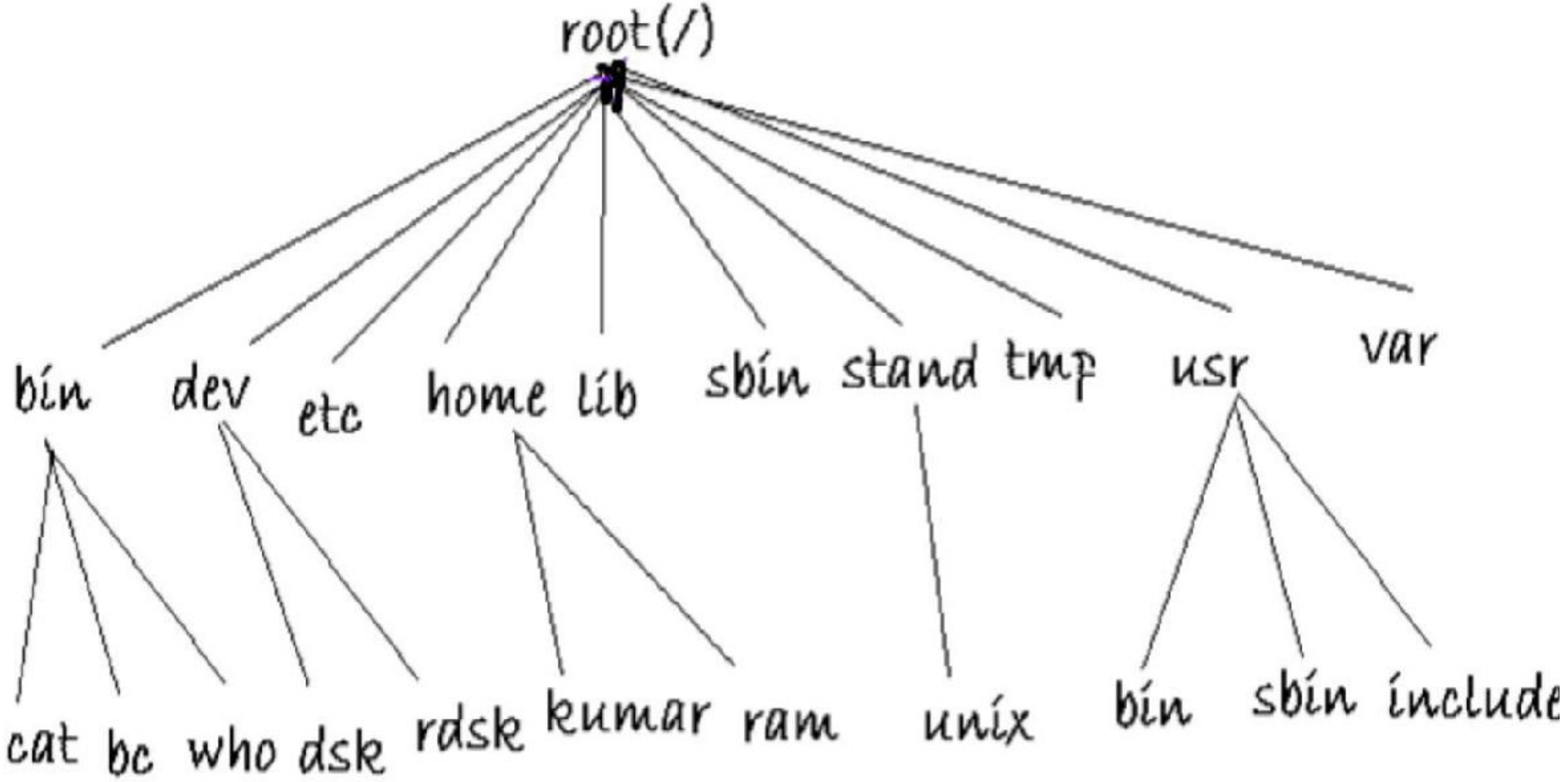
UNIX FILE SYSTEM

- Collection of control structures and Data blocks that occupy the space defined by a partition and allow for the storage and management of data.
- Common File Types
- Ordinary Files :Regular Files
- Directory Files: table of contents, that stores a list of files/directories.
- Device Files :For every device there is a device file used by kernel to interact with the devices.
- Hard Link:It is an another Reference to Files.
- **Symbolic Link:** Its link to other files



The UNIX File System Tree

UNIX FILE SYSTEM

- The UNIX file system contains following directories within the different system files resides.
- **/usr : The binaries,shared libraries,shared documentation etc.**
- **/bin: contains all commonly UNIX commands.**
- **/etc: contains configuration files of system.**
- **/dev: contains all device files.**
- **/lib: contains all library files in binary form.**
- **/home: different users are housed here.**
- **/tmp: contains all temporary files.**
- **/var: is the variable part of the system. Contains all your print job,**
- **incoming & outgoing mails.Stores the Log file of the system.**
- **/stand: holds the commands usable only by System Administer**

UNIX FILE SYSTEM

- / :root directory, top level directory
- /bin : user commands like cp, cron, cmp etc..
- /usr/sbin: admin commands
- /dev : logical device files of all hardware devices
- /devices: physical device files
- /etc : System configuration files and user database
- /tmp : to store temporary files
- /usr : the binaries, shared libraries, shared documentation etc.
- /var: stores the log Files

File Path name

- A sequence of file names, separated by slashes (/), that describes the path, the system must follow to locate a file in the file system.
- **Absolute pathname (start from the /-directory):**

  Eg: /export/home/user1/file1.
- **Relative pathname (start from the current directory)**
- ./test1 (. = current directory)
- ../team03/.profile (.. = parent directory)

cd Change Directory

- **syntax: cd [dir_name]** **pwd**
- Example: /tmp
- $ pwd
- /home/user3
- $ cd memo; pwd
- /home/user3/memo
- $ cd ../..; pwd
- /home
- $ cd /tmp; pwd

mkdir, rmdir – create & remove Directories

- **Syntax: mkdir directory**
- eg: $ mkdir d1
- syntax: rmdir dir
- eg: $rmdir /export/home/user1
- Create multiple directories simultaneously:
- $ mkdir -p   dir1/dir2/dir3
- Remove a directory and all its subdirectories
- $ rmdir -p dir1/dir2/dir3

touch command

- Updates the atime and mtime of a file if it exists. Else create an empty file.
- Example:
- $ ls –l
- -rw-r—r-- 1 user1 group1 100 Sep 5 09:30 test
- $date
- Mon, Sep 5 10:00 2005
- $ touch test file1
- $ ls -l
- -rw-r—r-- 1 user1 group1 100 Sep 5 10:00 test
- -rw-r—r-- 1 user1 group1 100 Sep 5 10:00 file

Managing Files

- Upon completing this module you should be able to understand the following:
- cat
- wc
- cp
- mv
- rm
- more
- tail.
- File Characteristics.

Managing Files

- cat: creates a file ex: $ cat>filename
- cat also used to display the contents of the file
- Eg cat filename display the contents of the filename.
- cat < filename displays the contents of the filename using input redirection operator.
- cat >filename creates a file using output redirection.
- cat >> filename performs append operation to a file.

Managing Files

- mv: perform two functions.
- 1. Renames a file or directory.
- 2. Moves a group of files to different directory.
- ex: $ mv c1 b1 [enter]
- renames c1 to b1
- $ mv c1 c2 abc [enter]
- moves c1, c2 to directory abc if the destination argument is a directory then it moves the files to the corresponding directory.
- **Syntax:**
- **mv [-i]** ***file new_file*** Rename a file
- **mv [-i]** ***file [file...] dest_dir*** Move files to a directory
- **mv [-i]** ***dir [dir...] dest_dir*** Rename or move directories

Managing Files

- cp - Copy Files
- **Syntax:**
- **cp [-i] file1 new_file** Copy a file
- **cp [-i] file [file...] dest_dir** Copy files to a directory
- **cp -r [-i] dir [dir...] dest_dir** Copy directories
- Example:
- cp file1 d1 copies file1 to d1 directory
- cp file2 file3 create a copy of file2 as file3 in the same
- directory

rm - Remove Files

- **Syntax:**
- **rm [-if]** ***filename [filename...]*** Remove files.
- **rm -r[if]** ***dirname [filename...]*** Remove directories.

- Examples:
- rm f1 removes the file f1
- rm –r d1 remove the directory.

Managing Files more - Display the Contents of a File

- **Syntax:**
- **more [*filename]*...** Display files one screen at a time
- Example:
- $ more funfile
- --funfile (20%)--
- Q *or q Quit more*
- Return *One more line*
- Space bar *One more pag*.

Managing Files wc command

- The **wc** command counts the number of lines, words and bytes in a named file:
- **syntax:**
- **wc [-c] [-l] [-w] filename**
- Options:
- -c counts the number of bytes
- -l counts lines
- -w counts words
- Example:
- $wc testfile

file :Knowing the File Types

- Unix provides the file command to determine the type of file,especially of an ordinaryfile.
- $ file archive.zip
- archive.zip :ZIP archive.
- file correctly identifies the basic file types (regular,directory or devices).
- Using the * to signify all files ,this is how files behave on this system having regular files of varying types:
- file *

cmp:Comparing Two Files

- You may often require to know whether two files are identical so that one of them can be deleted.
- There are three commands in the Unix system that can tell you that.
- cmp chap01 chap02
- chap01   chap02 differ: char 9,line 1.
- The two files are compared byte by byte and the location of the first mismatch is echoed on the screen. cmp by default ,does'nt bother about possible subsequent mismatches.
- The –l(list) option gives a detailed list of the byte number and the differing   bytes in octal for each character that differs in both files.

cmp:Comparing Two Files

- cmp –l note1 note2
- 3   143   145
- 6   170   167
- 7   171   170
- 8   172   171
- The above Eg shows four differences in the two files .If two files are identical ,cmp displays no message ,but simply returns the prompt.

- $ cmp chap01 chap02
- $ _

comm What is Common

- Suppose You have two lists of people and you are asked to find out the names available in one and not in the other,or even those common to both.
- comm is the command you need for this work it requires two sorted files and lists the differing entries in different columns.

• $cat file1	$cat file2
• c.k.Shukla	anil aggarwal
• chanchal singhvi	barun dasgupta
• s.n.dasgupta	c.k.shukla
• sumit chakrobarty	lalit chowdary
•	s.n.dasgupta

Comm What is common

- when you run comm, it displays a three columnar output:
- comm file1 file2

```
•          anil Aggarwal
•          barun sengupta
•                     c.k.Shukla
• chanchal  singhvi
•          lalit chowdury
•                      s.n.dasgupta
• sumit chakrobarty
```

Comm What is common

- The first column contains two lines unique to first file, and the second column shows three lines unique to the second file.The third column displays two files common(hence its name) to both files.

- These commands require single column output from comm,and comm can produce it using the options -1,-2 or -3.To drop a particular column ,simply use its column number as an option prefix .

- comm -3 foo1 foo2 Selects lines not common to both files
- comm -13 foo1 foo2 Selects lines present only in second file

ls Listing Directories and Files

- $ls When we type ls command at the shell prompt It is going to list all files and directories in the Present working Directory in the ASCII collating sequence.(Numbers first,uppercase amd then lowercase).

- Directories often contain many files and you may be simply interested in knowing whether a particular file is available.
- ls perl
- Perl:No such file or directory.
- ls comes with a very large number of options .
- Output in Multiple Columns(-x) When you have several files its better to display the filenames in multiple columns
-

Ls command options

- $ls –Fx :These option outputs ls command to identify directories and executable files. Combining with x produces a multicolumn output.

Showing Hidden Files (-a) ls does'nt normally show all files in a directory .There are certain hidden files (fienames begin with a dot)in every directory,especially
In the home directory that normally don't show up in listing.

The –a option (all) lists all hidden files along with the other files.

The –d option of ls force ls to list the attributes of the directory,rather than its contents.
Eg $ ls –ld helpdir progs

.

ls command Options

• -x	Multicolumnar output
• -F	Marks executables with * directories with / and symbolic links with @
• -a	Shows all filenames beginning with a dot including . and ..
• -R	recursive list
• -r	sorts filenames in reverse order
• -t	sorts filenames by last modification time
• -u	sorts filenames by last access times.
• -i	Displays inode Number

ls commands

- Listing Directory contents In the last example ,you specified some ordinary filenames to ls to have a selective listing

```
.   ls  -x helpdir progs
   helpdir:
     forms.odd    graphics.odd   reports.odd
  progs:
         array.pl      name.pl
```

.) ls -xR (Recursive Listing) The recursive option lists all files and subdirectories in a directory tree.

lp Subsystem:Printing a File

- The following lp command prints a single copy of the file rfc822.ps
- $lp rfc822.ps
- request id is pr1-320 (1 file)
- This file is’nt actually printed at the time the command is invoked, but

  later depending on the number of jobs already lined up in the queue.
- lp notifies the request id a combination of the printer name (pr1)

  and the job number [320] which can later be accessed with other commands.

lp options

- lp accepts the above requests because a default printer has been defined

  by the administrator.If there is more than one printer in the system

  you have to use the –d option with the printer name.

```
$lp –d laser chap01.ps
```

The –t[title] option,followed by the title string prints the title on the first page. Eg lp –t "First chapter" chap01.ps.

After the file has been printed ,you can notify the user with –m option.

You can also print multiple copies(-n).

```
$lp –n3  -m chap01.ps
```

Other command of the lp command

- The print queue is viewed with the lpstat command .By viewing this list you can use cancel command to cancel any jobs submitted by you.
- cancel uses the request-id or printer name as argument
- $cancel laser   cancel’s current job on printer Laser.
- $cancel pr1-320   cancels job with request-id pr1-320
- $ You can cancel only those jobs that you own (i,e you have submitted yourself) but the system admin can cancel any job.

OD –Octal Dump Command

- Many files (especially executable) contain non-printing characters and most unix commands don’t display them properly.The file odfile contains some of the charcaters that don’t show up normally.
- $ more odfile
- White space includes a _____ _ _ (Tab) 011 octal value for \t
- The ^G character rings a bell.
- The ^L character skips a page.
- The apparently incomplete first line actually contains a tab(entered by hitting tab).The od command makes these commands visible by displaying the ASCII

octal value of its input(here a file).

OD –Octal Dump Command

- The –b option displays this value for each character separately.
- Here's a trimmed output.
- $od –b odfile
- 00000 127 150 151 164 145 040 163 160 141 143 145 040 151 156 143
-
- 000020 165 144 145 163 040 etc all the character octal value is dumped on to the console.
- Each line displays 16 bytes of data in octal,preceded by offset.
- When the option –bc is combined with od command the output is friendlier.
- It displays the octal value along with the printable charcters.

dos2unix and Unix2dos

- Converting between Dos and unix Sometimes you'll need to move files between Windows and unix system.
- Windows use the same format as Dos where the end of line is signified by two characters CR(\r) and LF(\n).
- Unix Files on the other hand use only LF.
- Here are two lines from a DOS file ,foo viewed on UNIX system with vi editor.
- Line 1^M
- Line 2^M
- There's a ^M representing the CR sequence at the end of each line. An octal dump confirms this.

dos2unix   and Unix2dos

- od –bc foo
- 0000000   114 151 156 145 040   061 015 012 114 151 156 145 040
-                        L   i   n   e   1   \r   \n   L   I   n   e

- 040   062   015   012
-              2   \r   \n.

- The CR-LF combination is represented by octal values 015-012 and the character escape sequence \r and \n.
- Conversion of this file to UNIX is just a simple matter of removing the \r.

dos2unix   and Unix2dos

- For this purpose some UNIX systems feature two utilities –dos2unix and unix2dos for converting files between DOS and unix.
- This is how we use dos2unix to convert this file foo to UNIX format on a solaris system.
- dos2unix foo foo.dos
- The output is written to foo.dos .When you use od again ,you’ll find
- That the CR character is gone.
- od –bc foo.dos

```
•  0000000   114  151  156 145 040   061   012  114 151 156 145 040 062 012
•                  L     i      n    e           1      \n   L    I     n    e          2    \n
```

dos2unix   and Unix2dos

- unix2dos inserts CR before every LF,and thus increase the file size by the number of lines in the file.
- The syntactical form that works for dos2unix would also work for unix2dos.

Diff command

- diff is the third command that can be used to display file differences.
- Unlike its fellow members cmp and comm It also tells you which lines in one file have to be changed to make the two files Identical.
- $diff file1 file2
- or diff file[12]

• 0a1,2	Append after Line 0 of first file
• >anil Aggarwal	this line
• >barun sengupta	and this line
• 2c4	Change line 2 of first file
• < chanchal singhvi	Replacing this line
• ---	with
• > lalit chowdury	this line.

Diff Command

- 4d5 Delete Line 4 of first File
- <sumit chakrobarty containing this line.
- Diff uses certain special symbols and instructions to indicate the changes
- that are required to make two files identical.
- Each Instruction uses an address combined with an action that is applied

to the first file .0a1,2 means appending two lines after line 0 which become line 1 and 2 in the second file.

4d5 deletes Line 4 .

Compressing and Archiving Files

- To conserve Disk space you'll have to compress large and infrequently used files.A file can often be compressed to a fraction of its Original size.

- Every Unix System comes with some or all of the following compression and decompression utilities.
- compress and uncompress(.Z) (obsolete)
- gzip and gunzip(.gz)
- tar command for archival.
- zip and unzip(.zip)

Compressing and Archiving Files

- The Extension acquired by the compressed filename is shown in parenthesis.The degree of the compression that can be achieved depends on the type of file,its size and the compression program used.
- We wont consider compress in this syllabus because it is superseded by other programs.
- Apart from compressing, You'll also require to group a set of files into a single file,called an archive.The tar and zip commands can pack an entire directory structure into an archive.
- In the next section we'll be discussing these compression and archival utilities.

Compression and Archival Files

- We'll start with gzip a very popular program,that works with one or more filenames.It provides the extension .gz to the compressed filename and removes the original file.
- In the following example we run gzip on the file libc.html. We also note the file size ,both before and after compression using wc –c which counts characters.

```
$wc –c libc.html
3875302 libc.html
$gzip libc.html
$wc –c libc.html.gz
 788096  libc.html.gz
```

Compress and Archive File

- We can repeat same set of commands by using postscriptfile.
- wc -c UserGuide.ps
- 372267 User_Guide.ps   Before Compression
- 128341   User_Guide.ps.gz   After compression
- How much compression did we actually achieve for both files? Use the –l option with the compressed or orginal filenames as argument
- $gzip –l libc.html.gz   User_Guide.ps.gz
- Compressed uncompr   ratio   uncompressed name
- 788096   3875302   79.6   libc.html
- 128341   372267   65.5   User_guide.ps
- 916437   4247569   78.4   (totals)

Uncompressing a gzipped File(-d)

- To restore the original and uncompressed file ,You have two options.Use either gzip –d or gunzip with one or more filenames as arguments the .gz extension is optional yet again.
- For Eg
- gunzip libc.html.gz Recieves libc.html

  gunzip libc.html.gz User_guide.ps.gz Works with multiple Files.

  gzip -d libc.html Same __.gz assumed

  .) You can now browse the files with their respective viewers.
-

Compressing a gzipped File(-d)

- Recursive Compression (-r) You can descend a directory structure and compress all files found in subdirectories.You need the –r option ,and the arguments to gzip must comprise at least one directory.
- gzip -r progs
- This option can be used for decompression also. To decompress all files
- In the directory you need to use gunzip –r progs or gzip –dr progs.
- Writing to Terminal(-c)
-

tar – Archival Program

- For creating a disk archive that contains a group of files or an entire directory structure we need to use tar.
- For this minimal use of tar we need to know these key options.
- -c Creates an archive.
- -x   Extracts Files from archive.
- -t   Display files in archive
- -f arch Name the archive arch
- Only one of these key options can be used at a time.We'll also learn to use gzip and gunzip to compress and decompress the archive created with tar.

Creating an Archive (-c)

- To create an archive,we need to specify the name of the archive (with –f) the copy or write operation(-c) and the filename as arguments.
- Eg of tar command
- $ tar –cvf archive.tar   libc.html User_Guide.ps
- a libc.html   3785k
- a   User_Guide.ps 364k
- By convention we use the .tar extension ,so you'll remember to use the same tar command for extraction.
-

tar – Archival Program

- Tar command also behaves recursively to back up one or more directories.
- In the following example,tar fills the archive progs.tar with three directory structures.
- tar -cvf   progs.tar   c_progs java_progs   shell_scripts.
- We'll soon use the same tar command to extract files from this archive .
- Let's see how we can compress archive .
- <u>Using gzip with tar</u>
- If the created archive is very big ,you may like to compress it with gzip.
- gzip archive.tar.
- This creates a tar –gzipped file archive.tar.gz . This can be sent over the ftp or Internet.

Extracting Files from Archive (-x)

- tar uses the –x option to extract files from an archive. You can use it right away on a .tar file,the one we just used to archive three dimensions.
- tar –xvf progs.tar
- But to extract files from a .tar.gz file(like archive .tar.gz) you must first use gunzip to decompress the archive and then run tar:
- $gunzip   archive.tar.gz   Retrieves archive.tar
- $tar   -xvf archive.tar   Extracts file
- libc.html ,3875302   bytes
- User_Guide.ps   372267   bytes
- You'll now find the two fles in the current directory

Viewing Files from Archive(-x)

- Selective Extraction is also possible .Just follow the above command line with one or more filenames that have to be extracted.

  tar –xvf archives.tar User_guide.ps Extracts only User_guide.ps

  The above command extracts only a single file from the archive.
- tar –tvf archive.tar
- _rw_r__r__ 102/10 3875302 Aug 24 19:49 2002 libc.html
- _rw_r__r__ 102/10 372267 Aug 24 19:48 2002 Usr_guide.ps

Zip and Unzip :Compresing and Archiving

- Zip command generally does'nt compress as much as gzip but it combines the compressing function of gzip with the archival function of tar.
- So instead of using two commands to compress a directory structure ,you can use only one –zip. All the letters of the alphabet are available as it options but we'll consider only few of them.

- Eg $zip archive.zip libc.html User_Guide.ps
- Recursive Behaviour(-r) zip uses the –r option .It descends the directory tree structure in the same way tar does except that it also compresses files.

Zip and Unzip :Compresing and Archiving

- $cd ~
- $ zip -r sumit_home.zip   .   (Compresing and archiving home directory files)
- Using unzip Files are restored with the unzip command which in its simplest

form uses the compressed filename as argument.

$unzip archive.zip .

But if the uncompressed file exists on disk,then unzip makes sure that it's doing the right thing by seeking user confirmation.

replace libc.html ? [y]es, [n]o, [A]ll, [N]one, [r]ename:y

You can respond with y or n.You can also rename the file   (r) to prevent overwriting or direct unzip to perform the decompression on the remaining

Files non-Interactively.

Zip and Unzip :Compresing and Archiving

- Viewing the Archive(-v) You can view the compressed archive with the -v option .The List shows both the compressed and uncompressed size of each file in the archive along with the percentage of compression achieved.

```
$unzip –v archive.zip
Archive:  archive.zip
```