# UNIT 3: Ordinary File Handling

UNIX and Shell Programming: BCAC691

Dr. Tumpa Banerjee
Assistant Professor, Department of MCA

# $cat$ command

- Cat command is responsible for various file-related operations such as user to view concatenate create copy merge and manipulate file content.
- Basic syntax of cat command: $\$\ cat\ [option]\ [file]$
- Option: represents various command line options
- File: the name of the files to be processed

# *cat* command

- View the content of a file
- `$ cat filename1`
- It will display the content of a file on the terminal
- View the content of multiple files
- `$ cat file1 file2 file3`
- It will consecutively display the content of the files on the terminal

# $cat$ command

- View Contents of a File preceding with Line Numbers
- Adding the $-n$ option to cat introduces line numbers, making it convenient to identify and reference specific lines within the file.
- $\$\,cat\ \ -n\ filename$

- Create a file and add content
- create a new file or overwrite an existing file with new content, using ‘cat’ with the output redirection (`>`):
- $\$\,cat > note1.txt$

# $cat$ command

- ‘cat’ can concatenate multiple files into a single file.
- $\$\, cat\ file2\ file2\ \ > merged\_file$
- it can add the content of one file to another, ‘cat’ can be used along with the append (>>) operator:
- $\$\, cat\ file1\ \ >>\ \ file2$
- The ‘tac’ command is the reverse of ‘cat’ and is used to display the content of a file in reverse order.
- $\$\, tac\ filename$

# *cp* command

- the cp command is used to duplicate files or entire directories from one location of the file system to another.
- Syntax $ *cp source_file dest_file*
- if dest_file does not exist, it is created.
- If dest_file already exists, it is overwritten without any warning.
- Copy files to a Directory in Linux
- $ *cp file*1 *file*2 *file*3 *dest_directory*

# $cp$ command

- Options for $cp$ command

| option | description |
|---|---|
| $-i$ | Interactive copying with a warning before overwriting the destination file |
| $-b$ | Create a backup of the destination file in the same folder with a different name and format |
| $-f$ | Forces copying even if the user lacks writing permission delete destination file if necessary |
| $-r\ or\ -R$ | Copies directory structure recursively |
| $-p$ | Preserve file characteristics |

# $rm$ command

- The $rm$ (remove) command deletes one or more files.
- it normally operates silently and should be used with caution.
- A file once deleted cannot be recovered. $rm$ won’t normally remove a directory but it can remove files from one.
- $rm\ file1\ file2\ file3$
- $rm\ dir1/file1$

# $rm$ command

- Options for $rm$ command

| Option | Description |
|---|---|
| $-i$ | Ask the user for confirmation before removing each file |
| $-R$ | A thorough recursive search for all directories  and files within these subdirectories |
| $-f$ | Force removal |

# $mv$ command

- The $mv$ command rename(moves) files. It has two distinct function
  - ➢ It rename a file(or directory)
  - ➢ It moves a group of file to a different directory
- $mv$ does not create a copy of the file; it merely renames it. No additional space is consumed during renaming.
- $mv\ file1\ file2$
- A group of files can be moved to a directory.

# $more$ command

- The ‘more’ command in Linux is a useful tool for viewing text files in the command prompt, particularly when dealing with large files like log files.
- It displays the content one screen at a time, allowing users to scroll through the text easily.
- $more\ [-options]\ [-num]\ [+/pattern][linenum][filename]$

# $more$ command

- [-options]: any option that you want to use in order to change the way the file is displayed. Choose any one from the followings: (‘-d’, ‘-l’, ‘-f’, ‘-p’, ‘-c’, ‘-s’, ‘-u’)

| Option | Description |
|---|---|
| $-d$ | Prompt navigation help |
| $-f$ | Disable line wrapping |
| $-p$ | Clear and display |
| $-c$ | Overlapping text |
| $-s$ | Squeeze blank lines |

# *more* command

- **[-num]**: type the number of lines that you want to display per screen.

- **[+/pattern]**: replace the pattern with any string that you want to find in the text file.

- **[+linenum]**: use the line number from where you want to start displaying the text content.

- **[file_name]**: name of the file containing the text that you want to display on the screen.

# $lp$ (line printing) command

- UNIX is a multiuser system, and no user has direct access to the printer. One has to spool(line up) a job along with others in a print queue.
- Spooling ensures the orderly printing of jobs and relieves the user from the necessity of administering the print resources.
- The spooling facility is provided by the $lp$ command.
- $lp\, note.txt$

# $wc$ counting lines, words and characters

- $wc$ is a counting program, counts lines, words and characters
- Syntax: $wc\ filename$
- It will display four things:
  - ➢ Line count
  - ➢ Word count
  - ➢ Character count
  - ➢ Filename

# $wc$ command

- $wc$ specify three options to make a specific count.

| Options | Descriptions |
|---|---|
| $-l$ | Number of lines |
| $-w$ | Number of words |
| $-c$ | Number of characters |

# $cmp$ comparing two files

- $cmp$ command compare two files byte by byte, and the location of the first mismatch is echoed to the screen.
- $cmp\ file1\ file2$

# $comm$ command

- $comm$ command find similarity between two files
- It takes two sorted files and lists the differing entries in different columns.
- $comm\ file1\ file2$

# $diff$ command

- $diff$ tells which lines in one file need to be changed to make the two files identical.
- $diff\ file1\ file2$

# Compressing and archiving files

- Large and infrequently used files need to be compressed to conserve disk space
- compression is often required before sending it as an email attachment
- Unix file system comes with the following compression and decompression utilities
- *gzip or gunzip*
- *bzip2 or bunzip2*
- *Zip or unzip*

# Compressing and archiving files

- The extension acquired by the compressed file name is given in parenthesis
- The degree of compression that can be achieved depends on the type of file its size and the compression program used
- Large text files compress more but GIF and gpeg image file compress very little because they hold data in compressed form

# Compressing and archiving files

- You also need to group a set of files into a file called archive
- tar command can peek back an entire directory structure into an archive
- Archiving with completion bring down the file size, therefore tar often used with $gzip$ and $bzip$ for creating a compressed archive
- zip handles both function itself

# $gzip$ and $bzip$ Compressing and decompressing

- $gzip$ is a very popular program that works with one or more filenames. It provides extension $.gz$ to the compressed filename and remove the original file.
- $\$\,wc\ -c\ libc.html$
- $\$\ gzip\ libc.html$
- $\$\,wc\ libc.html.gz$
- Use $-\mathrm{l}$ option with original and compressed filename.
- $\$\ gzip\ -l\ libc.htm.gz\ use-guide.ps.gz$

# Decompressing a $gzip$ file

- Decompression requires restoring original file
- Two options available
- $gzip$ with $-d$ option
- $gunzip$ option
- Extension $.gz$ is optional here
- $gunzip\ libc.html$
- $gzip -d\ libc.html.gz$
- $-r$ option is used for recursively compressing all files in subdirectory

# $tar$ the archival program

- $tar$ is used for creating a disk archive that contain group of files or an entire directory structure
- Options of $tar$ command

| Options | Description |
|---|---|
| $-c$ | Create an archive |
| $-x$ | Extract files from archive |
| $-t$ | Display files in archive |
| $-f$ | Specify the archive |

# Creating an archive

- to create an archive, we need to specify the name of the archive(-f), the copy or write option(-c) and the filename as arguments.
- use -v option to display the progress bar while tar works.
- $\$\, tar\ \ -cvf\ \ filename.tar\ filename3$
- $\$\, gzip\ file\_arch$

# Extracting files from archive ($-x$)

- tar uses the -x option to extract files from and archive. can use it right away on a .tar file
- $\$\,tar\ \ -xvf\ progs.tar$

# viewing the archive($-t$)

- to view the content of the archive, using the -t (table of contents) option.
- it does not extract files but simply shows their attributes in a form that you will see more often later.
- `$ tar −tvf archive.tar`

# Compressing and Archiving

- Zip does two thing at a time
- Syntax: $zip\ file.zip\ file1\ file2$
- Syntax: $zip\ \ -r\ archive.zip$
- Syntax: $unzip\ file.zip$