

UNIX SYSTEMS PROGRAMMING



By

Dr. Trilok Nath Pandey

Dept. of C.S.E

S 'O' A, Deemed to be University

ITER, Bhubaneswar.

Previous Class

- File and directory commands
- cat
- cp
- mv
- rm

Today's Agenda

- chmod
- cmp
- diff
- top
- ps

File & Directory commands

- File permissions
- [SysPgm@labserver ~] \$ ls -al ↵
- The output list preceded by the words
- “total <# of blocks>”
- indicates that total <# of blocks> is occupied by the files in the disk.
- First two entries are . and ..

File & Directory commands

- First column of list identifies the type and permissions associated with each file.
- Type is identified by first character
 - – it is hyphen for regular files and d for directories.
- Next 9 characters take the values r (read), w (write), x (execute)
 - – tells about file permissions for three types of users User/Owner (u), Group (g) and Others (o)
- Each [u – g – o] one of them has three permissions
- To read (r) - 4, To write (w) – 2, To execute (x) – 1
 - – *A hyphen in any position means that you don't have that particular permission.*

File & Directory commands

- ***chmod*** command
- used to set the three permissions for all three categories of users (u – g – o) for the file
- Only owner of the file can set the access permissions
- **Syntax**
- *chmod [OPTION]... MODE[MODE]... FILE...*
- *chmod [OPTION]... OCTAL-MODE FILE...*

<i>Options</i>	<i>Purpose</i>
-R	change files and directories recursively

File & Directory commands

- Chmod Command

Command	Explanation
chmod u+x will	Execute permission is added to the owner for the file will
chmod ugo+x will	Execute permission is added to all for the file will
chmod o-x will	Execute permission is removed from the others for the file will
chmod g+r+w will	Read and write permission are added to the group for the file will

File & Directory commands

- Chmod Command

Command	Explanation
chmod 777 will	Read, write and execute permissions to all for the file <i>will</i>
chmod 774 will	Owner and group have all three permissions and others have only read permission for the file <i>will</i>
chmod 400 will	Read permission to the owner for the file <i>will</i> is set. Group and others don't have any access to the file.

File & Directory commands

- `cmp`
- **cmp** command in Linux/UNIX is used to compare the two files byte by byte and helps you to find out whether the two files are identical or not.
- When `cmp` is used for comparison between two files, it reports the location of the first mismatch to the screen if difference is found and if no difference is found *i.e* the files compared are identical.
- `cmp` displays no message and simply returns the prompt if the files compared are identical.
- Syntax: `cmp <file name1> <file name2>`
- Example:
- `[SysPgm@labserver ~] $ cmp file1 file2 ↵`

File & Directory commands

- diff
- Compare the contents of two ASCII files line by line.
- If the difference is found then it will also list differences along with line numbers.
- it tells us which lines in one file have to be changed to make the two files identical.
- Diff command can also be used to compare the contents of two directories.
- Syntax: `diff <file name1> <file name2>`
- Example:
- `[SysPgm@labserver ~] $ diff file1 file2 ↵`

File & Directory commands

- diff
- The important thing to remember is that **diff** uses certain **special symbols** and **instructions** that are required to make two files identical. It tells you the instructions on how to change the first file to make it match the second file.
- **Special symbols are:**
 - a : add
 - c : change
 - d : delete

File & Directory commands

- Let's take a look at what this output means. The first line of the **diff** output will contain:
- Line numbers corresponding to the first file,
- A special symbol and
- Line numbers corresponding to the second file.
- Like in our case, **0a1** which means **after** lines 0(at the very beginning of file) you have to add **Tamil Nadu** to match the second file line number 1. It then tells us what those lines are in each file preceded by the symbol:
- Lines preceded by a **<** are lines from the first file.
- Lines preceded by **>** are lines from the second file.
- Next line contains **2,3c3** which means from line 2 to line 3 in the first file needs to be changed to match line number 3 in the second file. It then tells us those lines with the above symbols.
- The three dashes (**"—"**) merely separate the lines of file 1 and file 2.

File & Directory commands

- As a summary to make both the files identical, first add *Tamil Nadu* in the first file at very beginning to match line 1 of second file after that change line 2 and 3 of first file i.e. *Uttar Pradesh* and *Kolkata* with line 3 of second file i.e. *Andhra Pradesh*. After that change line 5 of first file i.e. *Jammu and Kashmir* with line 5 of second file i.e. *Uttar pradesh*.
- `diff myfile3 myfile4`
- Here above output **3d2** means delete line 3rd of first file i.e. *Telangana* so that both the files **sync up** at line 2.

File & Directory commands

- top
- – List top CPU processes. It provides a dynamic real-time view of the running systems.
- Syntax: top
- Example:
- [SysPgm@labserver ~] \$ top ↵

File & Directory commands

- Top

Header	Description
PID	Shows task's unique process id.
PR	Stands for priority of the task.
VIRT	Total virtual memory used by the task.
USER	User name of owner of task.
%CPU	Represents the CPU usage.
TIME+	CPU Time, the same as 'TIME', but reflecting more granularity through hundredths of a second.
SHR	Represents the Shared Memory size (kb) used by a task.
NI	Represents a Nice Value of task. A Negative nice value implies higher priority, and positive Nice value means lower priority.
%MEM	Shows the Memory usage of task.

File & Directory commands

- ps
- The **ps** (process status) command is one of the most frequently used commands in Linux. Usually it is used to get the more and detailed information about a specific process or all processes.
- ***Specifying options with the ps command***
- The **ps** command accepts options in three styles.
- **BSD UNIX style:** - In this style, options are supplied without any leading dash (such as "**aux**").
- **AT & T Unix style:** - In this style, options are supplied with a leading dash (such as "**-aux**").
- **GNU Linux style:** - In this style, options are supplied with double leading dashes (such as "**--sort**").

THANK YOU