Linux System Administration and Server Configuration

Lecture-3

Basic Linux Commands

Move Command

- The "mv (move)" command can renames file without making a copy of it.

- rename a file: keeping it in the *current directory*

```
$ mv   file1   file2
```

Moving multiple files

- If a list of arguments is provided and the final argument in the sequence is the name of an existing directory, mv moves all of the other items into that directory

```
    mv  file1   file2   file3    test/
```

Move command

- Move a file named *file3*, without changing its name, from the current directory to an existing subdirectory dir1 of the current directory:

- *$ mv   file3   dir1/file3*

Displaying Contents of File on Screen

- "gedit" command :
  - Creates and displays the file.
- "cat" command :
  - Displays the content of a file.
  - cat filename
  - cat file1 file2 file 3
  - cat *.txt
- "vi" command
  - Strong editor in shell.

Displaying Contents of File on Screen

- “more”/ “less”
- Shows one page at a time.
- Called pagers.
- Press “q” to return.
- Press “h” for help for “more”.
  - “space” for next “screen size” lines.
  - “z” : shows next “screen size” lines.
  - “=” : displays current line number.
  - “b” : backwards screen sized lines.
  - “s” : skips line.
  - “f” : forwards screen sized lines.

Displaying Contents of File on Screen

  - "ctrl-L" : redraw screen.
  - ":f" : displays file name and line number.
- Press "h" for help for "less".
  - "e" : forward N lines.
  - "y" : backward 1 lines.
  - "f/z" : forward one window.
  - "b/w" : backward one window.
  - "d" : forward half of one window.
  - "u" : backward half of one window.

Displaying using Head / Tail

- "head" command :
  - Displays first 10 lines of the file.
  - "-v" : prints file name as header + first 10 lines.
  - "-n" : prints first n lines.
  - If two file name provided, displays 10 lines from each with a vertical gap.
- "tail" command :
  - Displays last 10 lines of the file.
  - "-n" : prints last n lines.
  - "-v" : prints file name as header + last 10 lines.
  - "-f" : follow.

Searching the Content of File

- Search
  - First "less filename"
  - After that "/search text" : forward search
  - "?search text" : backward search
  - "&search text" : show the lines with search text
- Redirection
  - Output redirection with replacement : >
  - Output redirection with append : >>
  - Input redirection with replacement : <
  - Output and Input redirection together :

```
$command  <  in.txt   >  out.txt
```

Searching the Content of File

- "grep(global regular expression print)" :
  - grep pattern filename
  - Normally grep is case sensitive.
  - "-i" : removes case sensitivity.
  - "-w" : show the lines with matching pattern.
  - "-x": show the line is whole line is matched.
  - "grep 'word1 word2' filename"
  - "grep –r 'a*' a.txt"

Searching the Content of File

- “**wc**” :
  - Word Count
  - Print “newline”, “word”, “character” count
  - “-w” : counts words
  - “-l” : counts lines
  - “-m” : counts character
  - “-L” : prints length of the longest line.

Piping

- Used when output of one command is used as input of another command.
  - $command | command
  - ls | sort
  - cat a.txt | sort
  - cat a.txt | wc
  - head a.txt | sort
  - ls | head | sort –r > c.txt

Copy a file

- The "cp (copy)" utility makes a copy of a file.
- This utility can copy any file, including text and executable program (binary) files.
- 'cp' can be used a make a backup copy of a file or a copy to experiment with.
- cp's basic syntax is
  - cp [option] name new_name

Copy a file

- The cp command line use the following syntax to specify source and destination files:
  - cp   source   dest
- The source file is the name of the file the cp will copy.
- The destination file is the name cp assigns to the resulting (new) copy of the file.

Copy a file

- By default cp only copies files and NOT directories
- If a file with the same name (or a directory as assigned to the copy of a directory) already exists, it will be overwritten
- The owner, group and permissions for the copy become the same as those of the file with the same name that it replaced.
- When a copy is made of a file or directory, the copy must have a different name than the original if it is to be placed in the same directory as the original.

Copy a file

- However, the copy can have the same name if it is made in a different directory

- Thus, for example, a file in the current directory (i.e., the directory in which the user is currently working) named *file1* could be copied with the same name into another directory, such as into */home/john/*, as follows:
  - `cp   file1   /home/john/file1`

Copy multiple files

- Any number of files can be simultaneously copied into another directory by listing their names followed by the name of the directory.
- cp is an *intelligent* command and knows to do this when only the final argument (i.e., piece of input data) is a directory
- Thus, for example, the following would copy the files named *file2*, *file3* and *file4* into a directory named *dir1*:

```
$ cp file2  file3   file4  dir1
```

Copying Directories

- Directories are not copied by default
- To make it more difficult for users to accidentally overwrite existing directory
- The *-r* (i.e., *recursive*) option, which can also be written with an upper case R, allows directories including all of their contents to be copied

```
$ cp   -r   dir2    dir3
```

Copying Directories

- The *-r* (i.e., *recursive*) option, which can also be written with an upper case R, allows directories including all of their contents to be copied

```
$ cp   -r   dir2    dir3
```

END