

Chap: 12 manipulating Text

* Cat

→ cat is short for Concatenate and is one of the most frequently used linux command line utilities.

→ It is often used to read and print files as well as simply viewing file contents.

→ To view file, use the following command.

```
$ cat <filename>
```

→ However main purpose of this command is to combine multiple file together.

→ The tac command prints the line of a file in reverse order.

→ Each line remains the same.

```
$ tac file
```

```
$ tac file1 file2 > newfile
```

Command	usage
cat file1 file2	concatenate multiple files and display the output
cat file1 file2 > newfile	combine multiple files and save the output into newfile
cat file >> existingfile	Append a file to the end of existing file
cat >file	Any subsequent line typed will go into the file, until CTRL-D is typed.
cat >> file	Any subsequent line are appended to the file, until CTRL-D is typed

x echo

→ echo simply displays (echos) text.

\$ echo string.

→ echo can be used to display a string on standard output or to place in a new file (using the > operator) or append to an already existing file (using >> operator).

→ The `-e` option, along with the following switches, is used to enable special character sequence, such as newline character or horizontal tab.

- `\n` represent newline

- `\t` represents horizontal tab.

Command	usage
<code>echo string > newfile</code>	The specified string is placed in a new file
<code>echo string >> existingfile</code>	The specified string is appended to the end of an already existing file.
<code>echo \$variable</code>	The contents of specified environment variable are displayed.

* working with large file.

→ For working with large file if you want to see that file in terminal you can use less command.

```
$ less somefile
```

```
$ cat somefile | less
```

* head

→ head reads the first few lines, of each named file (10 by default) and display it on standard output. you can give different number of lines.

For Example

```
$ head -n 5 | etc | default | gnyb
```

you can also say

```
$ head -5 | etc | default | gnyb
```

* tail

→ tail prints the last few lines of each named file and display it on standard output.

→ tail is especially useful when you are troubleshooting any issue using log files.

```
$ tail somefile.log.
```

→ If you want to see 15 line then

```
$ tail -n 15 somefile.log
```

or

```
tail -15 somefile.log
```

→ To continually monitor new output in a growing log file

```
$ tail -f somefile.log.
```