**File Display Commands**

**cat - concatenate a file**

Display the contents of a file with the concatenate command, cat.

**Syntax**

cat [options] [file]

**Common Options**

-n precede each line with a line number

-v display non-printing characters, except tabs, new-lines, and form-feeds

-e display $ at the end of each line (prior to new-line) (when used with -v option)

***Examples***

% cat filename

You can list a series of files on the command line, and cat will concatenate them, starting each in turn,

immediately after completing the previous one, e.g.:

% cat file1 file2 file3

**more, less, and pg - page through a file**

more, less, and pg let you page through the contents of a file one screenful at a time. These may not

all be available on your Linux system. They allow you to back up through the previous pages and

search for words, etc.

**Syntax**

more [options] [+/pattern] [filename]

less [options] [+/pattern] [filename]

pg [options] [+/pattern] [filename]

**Options**

more less pg Action

-c -c -c clear display before displaying

-i ignore case

-w default default don’t exit at end of input, but prompt and wait

-lines -lines # of lines/screenful

+/pattern +/pattern +/pattern search for the pattern

**Internal Controls**

more displays (one screen at a time) the file requested

<space bar> to view next screen

<return> or <CR> to view one more line

q to quit viewing the file

h help

b go back up one screenful

/word search for word in the remainder of the file

See the man page for additional options

less similar to more; see the man page for options

pg the SVR4 equivalent of more (page)

-------------------------------------------------------------------------------

**echo - echo a statement**

The echo command is used to repeat, or echo, the argument you give it back to the standard output

device. It normally ends with a line-feed, but you can specify an option to prevent this.

**Syntax**

echo [string]

**Common Options**

-n don’t print <new-line> (BSD, shell built-in)

\c don’t print <new-line> (SVR4)

\0n where n is the 8-bit ASCII character code (SVR4)

\t tab (SVR4)

\f form-feed (SVR4)

\n new-line (SVR4)

\v vertical tab (SVR4)

***Examples***

% echo Hello Class or echo "Hello Class"

To prevent the line feed:

% echo -n Hello Class or echo "Hello Class \c"

where the style to use in the last example depends on the echo command in use.

The \x options must be within pairs of single or double quotes, with or without other string characters.

-------------------------------------------------------------------------------

**head - display the start of a file**

head displays the head, or start, of the file.

**Syntax**

head [options] file

**Common Options**

-n number number of lines to display, counting from the top of the file

-number same as above

***Examples***

By default head displays the first 10 lines. You can display more with the "-n number", or

"-number" options, e.g., to display the first 40 lines:

% head -40 filename or head -n 40 filename

-------------------------------------------------------------------------------

**more**

Browses/displays files one screen at a time.

* Use h for help
* spacebar to page
* b for back
* q to quit
* /string to search for string

Example:

more sample.f

-------------------------------------------------------------------------------

**tail - display the end of a file**

tail displays the tail, or end, of the file.

**Syntax**

tail [options] file

**Common Options**

-number number of lines to display, counting from the bottom of the file

***Examples***

The default is to display the last 10 lines, but you can specify different line or byte numbers, or a

different starting point within the file. To display the last 30 lines of a file use the -number style:

% tail -30 filename