Chapter 3: Text Utilities

The cat* command

- The `cat` command merges two or more files into a single file:

```
$cat file1 file2 > file3
```

- Can also be used to display the contents of a file to the terminal:

```
$cat file1
```

*concatenate

The more & less commands

- While the `cat` command can be used to display the contents of a file to the terminal, it doesn't "pause" the display
- Use the `more` command to view a file one page at a time
- The `less` command uses many of the navigation keys used by the **vi** text editor in addition to ones used with `more`

The file command

- Only text files should be displayed, not binary files
- Displaying binary file can cause terminal corruption
- Either logout or run the reset command to fix terminal corruption
- To view file contents type, use the file command:

```
$file filename
```

The split command

- The split command will break large files into smaller files
- Useful for file transfer when large files create problems
- Syntax:

```
split [OPTION]...[INPUT [PREFIX]]
```

- `INPUT` is a file or stdin

The split command

- By default the new files will be named with a prefix of x and a suffix of aa, ab, etc.
- For example, the first file would be called xaa, the second file would be called xab, etc
- The `-d` option splits files to have a numeric suffix instead of a default alphabetic suffix
- The file names will start with the PREFIX, if specified; if not specified, then "x" is used

The nl command

- The `nl` command will number the lines of its output
- By default will only number lines that are not blank
- To number every line use:

```
$nl -ba
```

The head command

- The `head` command views the beginning of a file or output of another command
- Displays 10 lines by default
- Use `-n` to specify a different number of lines to display:

```
$head -n -30
```

The tail command

- The `tail` command views the end of a file or output of another command
- Displays 10 lines by default
- Use `-n` to specify a different number of lines to display

```
$tail-n -25
```

- Use `-f` to "follow" file changes

The paste & join commands

- The `paste` command merge the lines of one or more files, line by line, using a tab delimiter by default
- Use `-d` to specify a different delimiter
- The `join` command merges files, matching the values of fields to determine which lines to combine
- Use `-t` to specify a different delimiter

The expand & unexpand commands

- The `expand` command will convert tab characters into spaces
- The `unexpand` command will convert space characters into tabs
- Useful to make the content of files consistent
- Use -t to specify a tab stop position

The cut command

- The `cut` command extracts fields of information from a text file

```
$head -1 /etc/passwd | cut -d: -f1,5,6,7
root:root:/root:/bin/bash
```

- Default field separator is space or tab
- Use -d to change field separator

The sort command

- The `sort` command displays a file sorted on a specific field of data
- To specify the fields to sort from first to last, you can use one or more `-k` options

The sort command

- Use the "n" value to perform numeric sorts

```
$sort -t',' -k1n os.csv
1970,Unix,Richie
1970,Unix,Thompson
1987,Minix,Tanenbaum
191,Linux,Torvalds
```

The sort command

- Use the "r" value to reverse sort order
- Use the -u option to remove duplicate lines

The uniq command

- The `uniq` command removes duplicate lines from sorted documents
- Use `-c` to get a count of each line

The fmt command

- The `fmt` command does very simple text formatting
- The most common purpose is to format a text file with a maximum line width
- Use `-w` option to specify the width of each line

The pr command

- The `pr` command prepares a file for printing by
  - breaking the file into "pages“
  - displaying header information at the top of each page
- Use `-l` to specify lines per page
- Use `-d` to double space
- Use `-o` to specify a margin

The od command

- The `od` command performs an *octal dump* of data by default
- Used to display the contents of a file when it contains non-printable characters

The od command options

Single Option	Option with Argument	Meaning
-a	-t a	Named characters, ignoring high bit
-b	-t o1	Octal bytes
-c	-t c	ASCII characters or backslash escapes
-d	-t u2	Unsigned decimal 2-byte units
-f	-t fF	Floats
-i	-t dI	Decimal integers
-l	-t dL	Decimal longs
-o	-t o2	Octal 2 byte units
-s	-t d2	Decimal 2 byte units
-x	-t x2	Hexadecimal 2 byte units

The tr command

- The `tr` command can be used to *tr*anslate from one set of characters to another

```
$tr 'a-z' 'A-Z' < alpha-
first.txt.original
A IS FOR APPLE
B IS FOR BEAR
C IS FOR CAT
D IS FOR DOG
E IS FOR ELEPHANT
```

The sed command

- The *s*tream *ed*itor command (`sed`) can be used to modify text

```
$sed 's/Apple/Animal/' alpha-first.txt
A is for Animal
B is for Bear
C is for Cat
D is for Dog
```

- Use `–i` to modify the original file

The sed command

- Replaces first occurrence only be default
- Use `-g` to modify all occurrences
- Use `-i` to perform case insensitive matches

The sed command

- Use "i\" to insert
- Use "a\" to append

The sed command

- Use "d" to delete lines
- Use "c\" to change lines

The sed command

- Use `-e` to provide multiple changes