Server Operating Systems

Lecture 3
Basic File and Directory Mangement

Command Line Control Characters

Control-s	Stops screen output - rarely used
Control-q	Resumes screen output stopped by Control-s - also rare
Control-c	Interrupts current activity - frequently used to abort processes or long display outputs
Control-d	end-of-file or exit. If stuck, try Control-c or Control-d
Control-u	Erases the entire command line - good for mistyped passwords
Control-w	Erases the last word on the command line

`file` - Determining File Type

Use the **`file`** command to determine file type.

Syntax:

```
file filename(s)
```

The output is usually one of:

- *Text*: Any plainly editable file (ASCII).
- *Executable*: Command or a program (/bin/ls).
- *Data*: File created by an application (Word file).

Use it on the following files:

- dir1/coffees/beans
- dante
- /usr/bin/cal

`strings` - display readable characters

Prints out readable characters from an executable or binary file.

Syntax:

```
strings filename
```

Often used by programmers, if they have forgotten what a file contains.

Will display the error messages, for example.

Use it on: /usr/bin/cal

`cat` - Display File Contents

The “concatenate” command.

Syntax:

```
cat filename(s)
```

Used to display the entire contents of a text file.

- Often used for viewing small files (one terminal screen or less).

May need to start/stop scrolling on longer files with Ctrl-Q / Ctrl-S.

Use it on: dante

`more` - Displaying File Contents

Displays the contents of a text file one screen at a time.

Syntax:

```
more filename(s)
```

Press '*h*' while in more to get help on how to navigate through the text file.

- Same commands as when viewing man pages.

The less command offers similar, but extended, functionality.

Use it on: dante

`head` - Displaying Portions of a File

Display the first n lines of one or more text files.

Syntax:

`head [-n]` ***`filename(s)`***

The first 10 lines are displayed if the n option is not given.

Use it on: dante

- How many lines displayed?
- Use it to display the first 20 lines.

`tail` - Displaying Portions of a File

Display the last n lines of a text file.

Syntax:

`tail [-n]` ***`filename(s)`***

- The last 10 lines are shown by default if the n option is not given.

Display all lines from line n to the end of a text file.

`tail [+n]` ***`filename(s)`***

Use it on: dante

`wc` – Word Count Command

Displays *line*, *word*, *byte*, or *character* counts for a text file.

Syntax:

```
wc options filename(s)
```

– Displays line, word and byte count if no option given.

Option	Function
-l	Counts lines
-w	Counts words
-c	Counts bytes
-m	Counts characters

Use it on: dante

`diff` - Comparing Files

The “Difference” command compares two text files displaying the differences between them.

Syntax:

```
diff option file1 file2
```

There are many options, but two important ones are:

-i: ignores case.

-c: performs a detailed comparison and gives three lines of context around the different lines.

Use it to compare the following files: fruit and fruit2

Naming Conventions

- Long file and directory names are not recommended although, combined, they can be as long as 255 characters.
- Alphanumeric characters are recommended along with the non-alphanumeric characters of the dash (-) and underscore (_).
- Other non-alphanumeric metacharacters are allowed but not recommended
- File Names normally contain one extension but can have more than one
- Directory names normally do not contain extensions but can

`touch` - Creating Files

To create a new, empty file, use the touch command.

Syntax:

```
touch filename(s)
```

If the filename exists, touch just updates the modification date/time; otherwise, an empty file is created.

The user must have appropriate permissions to create the file.

`mkdir` - Creating Directories

Use the **`mkdir`** command to create a new directory.

Syntax:

```
mkdir [-p] filename(s)
```

The user must have appropriate permissions to create the directory.

The –p “parent” option creates any directory in the pathname argument that does not exist automatically.

rm - Removing Files

The rm command can be used to remove one or more files from the file system *PERMANENTLY*.

Syntax:

```
rm [-i] filename(s)
```

Typing `rm *.*` is not a good idea!

The “interactive” –i option asks for confirmation before permanently deleting a file.

Removing Directories

The “recursive” option is used with rm to remove directories.

```
rm -r [i] directory_name(s)
```

The directory named is removed including all of its subdirectories and the files in it.

If a directory is empty, rmdir can be used instead.