CS1101: Lecture 5

Basic File Manipulation II

Dr. Barry O'Sullivan
`b.osullivan@cs.ucc.ie`

Course Homepage
`http://www.cs.ucc.ie/˜osullb/cs1101`

Lecture Overview

- Basic File Manipulation Commands
  - `mkdir`
  - `rmdir`
- Two More Abbreviations
  - ?
  - *
  - Example using Wildcards
- More Basic File Manipulation Commands
  - `file`
  - `more`
  - `cp`
  - `mv`
  - `rm`
  - `lpr`
  - `lpq`

Basic File Manipulation Commands

Command	Action
`cd`	moves you to another directory
`pwd`	tells you which directory you are in
`mkdir`	creates a directory
`rmdir`	removes a directory
`ls`	lists the contents of a directory
`file`	displays the type of a file
`more`	displays the contents of a file
`cp`	copies a file/directory
`mv`	moves a file/directory
`rm`	removes a file/directory
`lpr`	prints a file
`lpq`	checks the printer status

Please note that UNIX is case-sensitive!!!

Making New Directories

- The `mkdir` command creates directories;
- It syntax is: `mkdir <directory>` where `<directory>` is the path of the directory you want to make;
- Suppose your home-directory is `/home/fred`. You want to create a directory there called `cs1101`:

```
wisdom.ucc.ie:%> cd
wisdom.ucc.ie:%> mkdir cs1101
wisdom.ucc.ie:%>
```

- Alternatively, you could simply use the absolute path without changing directories:

```
wisdom.ucc.ie:%> mkdir /home/fred/cs1101
wisdom.ucc.ie:%>
```

- Alternatively, you could simply use the abbreviation for your home-directory:

```
wisdom.ucc.ie:%> mkdir ~/cs1101
wisdom.ucc.ie:%>
```

Removing Directories

- `rmdir` does the opposite of `mkdir` – it removes directories;
- Its syntax is `rmdir <directory>` where `<directory>` is the path to the directory you want to remove;
- Example, to remove a directory called cs1101 in your home-directory you could issue the following command:

  ```
  wisdom.ucc.ie:%> rmdir ~/cs1101
  wisdom.ucc.ie:%>
  ```

- Directories must be empty before they can be deleted using this command.

What Files are in This Directory?

- The `ls` command tells you what files (and directories) are in a directory.
- The syntax of `ls` is `ls <directory>` where `directory` is the (optional) name of the directory you want to look at;
- Example:

  ```
  wisdom.ucc.ie:%> ls /var
  adm       dt        mail      opt
  audit     log       news      preserve
  cron      lp        nis       sadm
  wisdom.ucc.ie:%>
  ```

- If you don't specify a directory, `ls` lists the contents of the current directory:

  ```
  wisdom.ucc.ie:%> ls
  mail      cs1101    letters
  wisdom.ucc.ie:%>
  ```

Seeing Hidden Files

- Files starting with `.` are *hidden*;
- Hidden files are not normally displayed – however, `ls -a` does (the a stands for "all");
- Example:

  ```
  wisdom.ucc.ie:%> ls
  mail      cs1101    letters
  wisdom.ucc.ie:%>
  wisdom.ucc.ie:%>  ls -a
  .         ..        .cshrc
  mail      cs1101    letters
  wisdom.ucc.ie:%>
  ```

Full Information on Files

- `ls -l` (lowercase "L") provides more information about your files – contents of the directory, who owns a file, its size, when it was last modified, etc
- Example:

```
wisdom.ucc.ie:%> ls -l
total 3
drwxr-x---   6 osullb   system       512 Oct  5 16:33 bin
-rw-r--r--   1 osullb   system       873 Feb 18  2000 iimb
drwxr-x---   2 osullb   system       512 Aug 10  1999 nsma
wisdom.ucc.ie:%>
```

Two More Abbreviations: Wildcards

- There are two wildcard characters: `*` and ?
- These can be used to describe a pattern for file/directory names that you want to access;
- `*` represents any number of characters in a file/directory name
- ? represents a single character in a file/directory name

Examples: Wildcards

- `states` is a directory containing files with names of US states:
- Example 1:

```
wisdom.ucc.ie:%> ls ~/states/m*
maine    michigan   mississippi montana
maryland minnesota  missouri    massachu
wisdom.ucc.ie:%>
```

- Example 2:

```
wisdom.ucc.ie:%> ls ~/states/m*a
minnesota montana
wisdom.ucc.ie:%>
```

- Example 3:

```
wisdom.ucc.ie:%> ls ~/states/a?????a
arizona   alabama
wisdom.ucc.ie:%>
```

- Note Alaska wasn't found!

What Sort of Files Are These?

- The `file` command tells you information about the "type" of a file
- Its syntax is `file <file>` where `<file>` is the name of the file whose type you want to know;
- Example:

```
wisdom.ucc.ie%> file mytextfile.txt
mytextfile.txt: ascii text
wisdom.ucc.ie%>
wisdom.ucc.ie%> file myprogram
myprogram: Sun demand paged SPARC execut
wisdom.ucc.ie%>
```

Viewing the contents of files

- The `more` command displays the contents of files:
- Its syntax is `more <file>` where `<file>` is the name of the file whose contents you want to see;
- Example:

```
wisdom.ucc.ie%> more mytextfile.txt
This is the first line of the file
This is the second line of the file
wisdom.ucc.ie%>
```

Copying Files and Directories

- The `cp` command is used;
- The syntax is:

```
cp <source-file> <destination-file>
cp <source-file> <destination-directory>
cp <source-directory> <destination-direc
```

- After using `cp` both the `<source-file>` and `<destination-file>` exist;
- Examples:

```
cp georgia georgia.copy
cp ~/myletter ~/myletter.2
cp ~/myletter ~/letters
```

Moving Files and Directories

- The `mv` command is used;
- The syntax is:

```
mv <source-file> <destination-file>
mv <source-file> <destination-directory>
mv <source-directory> <destination-direc
```

- After using `mv` the `<source-file>` does not exist any longer;
- Examples:

```
mv georgia ukraine
mv ~/myletter ~/mynewletter
mv ~/myletter ~/letters
```

Removing Files

- The `rm` command is used to remove files;
- The syntax is:

```
rm <file-list>
```

- `<file-list>` is a list of files;
- Examples

```
rm myletter
rm myletter myletter2 myletter3
```

- **Once a file is removed, you cannot get it back!!!**
- Use the following version for prompts:

```
rm -i <file-list>
```

- Directories can also be removed;

Printing Files

- Use `lpr` to print out files;
- Syntax:

```
lpr <file>
```

- This command can be used to print postscript files to a postscript-enabled printer;
- To print to a named printer:

```
lpr -P<printer-name> <file>
```

- `lpq` tells you what jobs are on the printer;
- Usage:

```
lpq
lpq -P<printer-name>
```