More About Files

Referencing Files on the Command Line

In this lab, we will continue learning about files, including:

- how to reference them as arguments to commands
- how to specify them as input to and output from commands

Run the following commands to set up for this lab:

```
% mkdir lab04

% cd lab04

% touch firstFile sampleFile nextFile onefile twofile threefile

% ls
```

File Name Meta-characters

We will learn about how to build name patterns to refer to a set of files as an argument to supply to a command. First, some basic background on useful metacharacters:

- the star/asterisk - * - refers to zero, one, or more characters in a file name, and these characters can be any characters that are valid in a file name
- the question mark - ? - refers to one and only one character in a file name, but this character can be any character that is valid in a file name
- a bracket pair - [ ] - refers to one and only one character in a file name, and this character must be one of those specified within the bracket pair
- the caret - ^ - used to exclude a set of characters specified within a bracket pair, i.e., the character must not be one of those specified within the bracket pair

To illustrate the use of these metacharacters, execute the following commands and explore why you got the results you did:

```
% ls *F*

% ls *f*

% ls *r*

% ls *[Ff]*

% ls *[n-p]*

% ls ????ile

% ls *e?ile

% ls [^s]*
% ls *[^s]*
```

Types of Command Line Input and Output

The term for the set of inputs and outputs we can work with on the command line is called *standard i/o*. *I/O* stands for input/output. There are three types of standard i/o:

- standard input (*stdin*) – this is the input that is available to a program run from a command line
- standard output (*stdout*) – this is one of the outputs that is available to a program run from a command line, and reports information generated when the program is running correctly
- standard error (*stderr*) – this is one of the outputs available to a program run from a command line, and reports error conditions encountered by the program

Standard Input

A program can use data supplied by standard input by "reading" it. By default, the program will wait for you to type in data if the program you run needs data supplied by standard input. You can type in data, including the Enter key, and indicate you have no more data to enter by pressing control-D.

Controlling Command Line Input and Output

You can control text accepted and generated by a program as follows:

- redirection of standard output by replacement - > *filename* – this metacharacter specifies that the output of the command shall be placed into a file named *filename* as follows: if the file exists, its contents will be replaced by the output of the command, and if the file does not exist, the file will be created and will contain the output of the command
- redirection of standard output by append - >> *filename* – this metacharacter specifies that the output of the command shall be placed into a file named *filename* as follows: if the file exists, the output of the command shall be added to the end of the file while retaining the existing contents, and if the file does not exist, the file will be created and will contain the output of the command
- redirection of both standard output and standard error by replacement - >& *filename* – this works just like redirecting just standard output by replacement
- redirection of both standard output and standard error by append - >>& filename – this works just like redirecting just standard output by append
- standard input redirection - < *filename* – this metacharacter specifies that the contents of *filename* shall be used as the input of the command, rather than waiting for the user to type in input
- pipe - | – this metacharacter specifies that the output of the command on the left side of the pipe shall be used as the input of the command on the right side of the pipe

Try the following on the command line:

Standard Output and Standard Error

```
% ls > fileList
% cat fileList
% date >> fileList
% pwd >> fileList
% cat fileList
% ls fileNotFound > f.out
% cat f.out
% ls fileNotFound >& f.err
% cat f.err
```

Standard Input

```
% sort > sorted
```

The shell will wait for your input. Type in five words or sentences, ending each by pressing the Enter key. Then, press control-D. Examine the program's output by:

```
% cat sorted
```

Try redirecting standard input by:

```
% cat < fileList
% sort < fileList
```

Pipes

```
% cat fileList | sort
% man sleep | sort
```