Exercises-Find

Part One

Description

This exercise practices simple forms of the **find** command. **find** is used to output paths beneath a set of directories that match as set of options, called predicates. **find**'s options are called predicates for good reason: each options describes a quality of a unix object and **find** only outputs a path if the object it refers to has qualities that match each of the predicates. An example should clarify this:

```
find dir1 dir2 -name "foo" -type d
```

This **find** command has two directories to look under, **dir1** and **dir2**, and two predicates. The first indicates that the name of the object must be **foo**. The second indicates that the object must be a directory. Thus **find** will only output the paths to directories named **foo** that are beneath **dir1** or **dir2**.

find has many predicates. In this class, we will only learn the two simplest and most useful: **-name** and **-type**:

-name "pattern"	here *pattern* is a wildcard pattern. You should always quote this pattern!
-type x	here *x* is a file type. The only file types we will learn in this class are regular file (*f*) and directory(*d*) (and, later, symbolic link (l)

For example, if the output of the command

```
find . -name "A*"
```

is

```
./Amy
./work/Apple
./hw/Answers
```

this means that there are three objects beneath the current directory whose name starts with **A**. Note, however, that you do not know whether these objects are regular files or directories.

(Note: if you had not quoted the wildcard pattern in the **find** command above, only the first object would have been found. Can you explain why?)

The general form of a **find** command is

```
find list-of-directories [ list-of-predicates ]
```

Exercises

Connect to the directory find beneath the class public directory on hills. Then write **find** commands to do the following:

1. Output the paths to regular files beneath **usr**
2. Output the paths to directories beneath **practice**
3. Output the paths to regular files beneath **practice** whose names are **tree**
4. Output the paths to directories beneath **practice** whose names are **tree**
5. Output the paths to directories beneath **usr** whose names are five characters long
6. Output the paths to objects beneath **practice** whose names start with **p**
7. Output the paths to objects beneath **usr** whose names have a **.** in them
8. Output the paths to everything beneath the current directory. Put the output in a file named **find.out** in your home directory. Then examine the contents of **find.out** using **more.**
9. Last, output the paths to all objects beneath **practice**. Can you use this output to create a directory

tree of **practice**? How can you know what is a file and what is a directory?

Part Two

Description

The **locate** command is useful for finding data that has been on the filesystem for at least a day. It does not allow you to list objects by attributes - it only searches a list of all data objects on the filesystem produced periodically (usually overnight). Thus you can only search for patterns in file *paths*, not by what type of data it is.

Since **locate** does not examine current files, rather it examines a list of files produced previously, it is much faster than **find**. The pattern you use to search for the filenames can be used to restrict the output.

```
      locate [options] pattern
```

The default (no options) is to interpret the pattern as a wildcard pattern. If the pattern does not contain wildcard characters, leading and trailing asterisks are added - thus **locate pattern** is really **locate '*pattern*'**

You can alter the type of pattern used by use of the **--regex** (regular expression) or -**-regexp** (extended regular expression) options.

Procedure

1. Output all file paths that contain the string **asmt02.script**
2. Output all file paths where the filename part (the last path segment) is **asmt02.script**
3. Create a new file named **asmt02.script** in your current directory and re-issue the command for 2. Does your new file show up?
4. Output all file paths beneath **gboyd**'s home directory (**/users/gboyd/**) whose filename part is **asmt02.script**
5. Output all file paths that look like configuration files (whose name ends in **.conf**)
6. Limit the search of 5. to paths beneath the system directory **/etc**

Answers

Part One

1. **find usr -type f**
2. **find practice -type d**
3. **find practice -type f -name "tree"**
4. **find practice -type d -name "tree"**
5. **find usr -type d -name "?????"**
6. **find practice -name "p*"**
7. **find usr -name "*.*"**
8. **find . > ~/find.out**
   more ~/find.out
9. **find practice** You can almost draw the directory tree from this output. If a directory isn't empty, you can tell it is a directory since it has something in it. You cannot, however, distinguish a regular file from an empty directory.

Part Two

1 **locate asmt02.script**

2 **locate "*/asmt02.script"**

3 **no notes**

4 **locate "/users/gboyd/*/asmt02.script"**

5 **locate "*/*.conf"**

6 **locate "/etc/*/*.conf"**

This document was produced with free software: OpenOffice.org on linux.