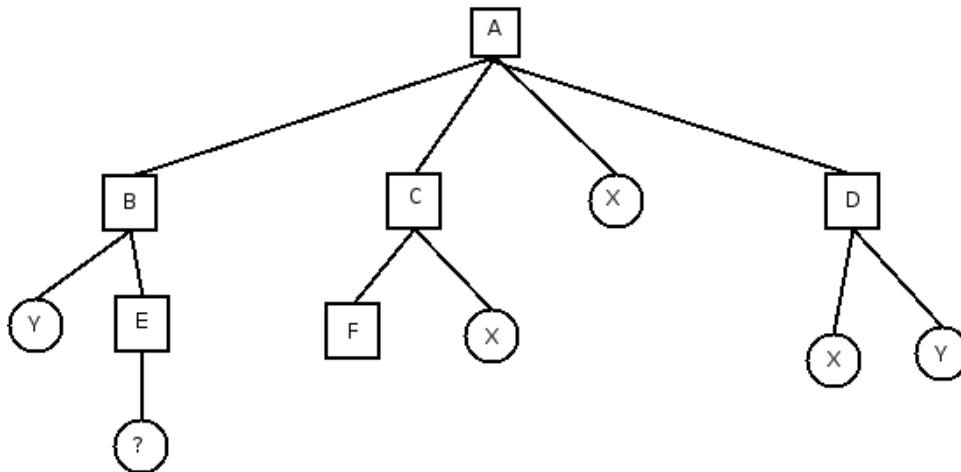


Stat 133, Fall '05
Homework 3: Shell Tools
Due Tuesday, 27 Sep

Problem Solving with Shell Tools

A window-system/desktop with interactive file rollers and other graphical interfaces to the data stored on your computer can be very handy and easy to use. But sometimes the scale or complexity of the problem makes these easy to use interfaces cumbersome and time consuming, as they involve many repetitive mouse and button clicks. Other times, the problem needs to be performed automatically, i.e. without manual intervention. These are the times when shell tools can be real time savers and can automate a tedious procedure. The following exercises will hopefully demonstrate the usefulness of these tools.

1. Create a file and directory structure in the top-level directory of your SCF account that matches the diagram shown here, where the squares represent directories and the circles files. For the file marked “?” you are to find and copy a specific file in the **s133/data/** area. Use two different approaches to find your personal file:
 - Look for a file name that contains your SCF login, e.g. s133aw.
 - Look for a file whose contents contain your actual name, e.g. Deborah Nolan. This would be the name that you typed in when you signed up for your SCF account.
2. It's often a good idea to make back ups of your data, and sometimes it is necessary for these copies to preserve symbolic links, devices, attributes, permissions, ownerships etc. For example, if you have a big project that you want to be able to work on on your laptop and on the computer in your office, then it would be important to synchronize these files and directories on a regular basis in order to keep your two work environments identical.



Search the web using Google (www.google.com) to find how to use a synchronization program called `rsync`. Read their examples, and then recursively transfer all the files in **A** to another directory called **Archive** in your `s133xx` home directory. Be sure to preserve all of the permissions, attributes, etc in your file transfer.

3. At other times, we need to ship many files over the internet, and it can be far more efficient to do this if we first wrap them up into a single file. One such file, called a *tar ball*, is made using the `tar` program. Read the documentation and create a tar ball of all the files in the directory **A**. Be sure to give the file a name with a `.tar` file type extension.
4. In the directory `/class/u/s133/s133aw` you will find a directory called **bare**. This directory contains another 10 subdirectories named **part1** through **part10**. In each of these there are many files of email messages, i.e. one file for each email message. There are two kinds of files in these directories, representing two kinds of email, spam and regular email, and these files are named differently if they are spam or regular mail.
 - Determine the naming convention for these files.
 - Find the total number of spam message files in the **bare** directory. Use two different approaches to do this. The commands `ls`, `find`, `grep` and `wc` may prove useful. Be sure not to include the directories in your count. It is often a good idea to do something two different ways

in order to check that you have done it correctly. If the two answers do not match then one (or both) of the approaches is incorrect.

- Now determine how many spam files there are in each **part1-part10** directory. Do this with one compound command. For example, you could create a list, one line for each spam file, where you only “print” the directory that the file is in. Then the `sort` and `uniq` commands can help you determine how many duplicates that you have.

Place your shell commands in a **plain text** file called **myCode.txt** in your top-level directory on your stat account. We will check the time stamp on the file to determine that it has been last modified prior to 10 p.m. on the due date.