Homework #1 -- UNIX Utilities

Purpose: This homework is intended to give you some practice with the utilities you will use to compile and test your programs.

Preparations: To keep your typing to a minimum, set a ‘‘soft link’’ to the public directory from your home directory as follows:

ln -s ../public public **<-- I will give the path “...” in class**

Keep in mind that UNIX is case-sensitive and double-check your typing. Also, unlike Windows, UNIX uses a forward slash to separate components of a file name. To verify that you entered the command correctly, list the contents of the public directory as follows:

ls public

Among other things, you should see subdirectories named **HW1**.

Now copy the contents of the **HW1** directory to a subdirectory in your home directory:

cp -r public/HW1 HW1

Don’t forget the **“-r”** option. Finally, change directories to your **HW1** subdirectory. Be sure the following files are in your **HW1** subdirectory:

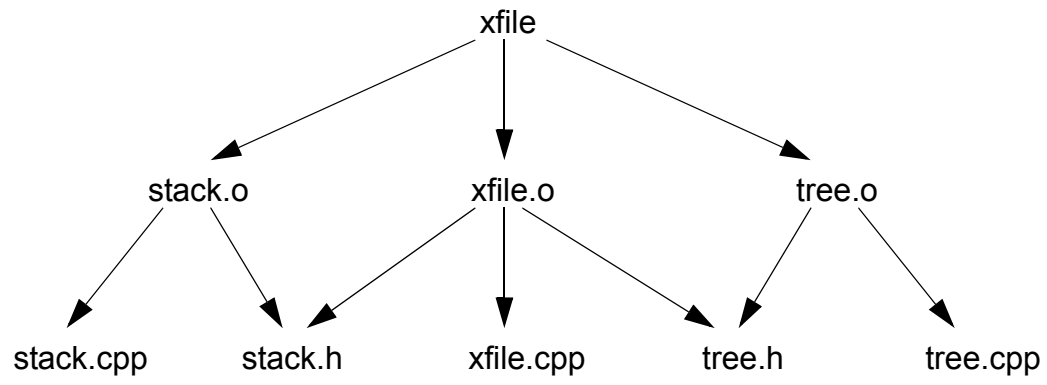
Makefile	**queue.h**	**stack.h**
hw1.cpp	**queue.cpp**	**stack.cpp**
hw1.dat		

The files also be available in the Pilot dropbox. You will then have to upload the files to unixapps1 using an appropriate FTP application.

---

1. Analysis of Make:

a. Derive a **Makefile** from the dependency graph shown below.



b. Draw a dependency graph for the **Makefile** in your **HW1** subdirectory.

2. Computer Exercise:

Use the utility **script** to produce a transcript file named **typescript** which contains the output of your computer session. To start the transcript, just execute **script** with no parameters. Enter commands as usual. To end the transcript, just type **Control-D** (hold down the control key and press d). Your transcript should contain the following items:

a. Using **make**, compile and link **hw1**.

b. Run **hw1**, redirecting standard input from file hw1.dat (e.g. hw1 < hw1.dat).

c. Using the utility **touch**, update file **stack.cpp**

touch stack.cpp

d. Compile and link **hw1** again, using **make** of course.

e. Type control-d to terminate the typescript

f. Download your transcript file and submit the file to the homework 1 dropbox.

3. In 2d, you were asked to recompile and relink program hw1. You should have noticed that Queue.cpp and hw1.cpp were not re-compiled. Using the dependency graph from 1.b, explain why.

4. Using the various tools discussed in class, write a C++ program that computes the greatest common divisor of a pair of integers. If you search for a solution on the web, you will several variations of algorithms to solve this problem. This problem is also described in exercise 1.11 on page 63 of your textbook.

Your program should take input from the keyboard and produced an single integer value representing the greatest common divisor. You may assume the user provides a valid pair of nonzero positive values. A samples of the behavior of your program is shown below:

```
gcd
12 15
3
```

When you are done, submit a makefile and gcd.cpp to the homework1 dropbox.