Project 4: BackItUp! (Deadline: 04/27)

Program Description

This assignment should be programmed entirely in C and must compile and run on the lab's linux environment or similar. Copying & pasting code from anywhere is considered cheating and results in an automatic F.

WARNING: This assignment involves the creating and overwriting of files. It is very important that you are careful with your testing! It is very easy to lose your program! Your goal for this assignment is to create a file backup system called BackItUp!

1. *BackItUp* should create a directory *.backup/* if one does not already exist (Unix Tip: files and directories that start with a '.' are hidden and in order to see them in the terminal you must use the *ls -a* command). (4 pts)

2. *BackItUp* should then create a copy of all the regular files in the current working directory in the *.backup/* directory. All the backup files should have *.bak* appended to their name. (4 pts)

3. If a backup file already exists, your program should compare the last modification times of the original file and the backup file and determine if a backup is required. If the existing *.bak* backup file is older, then your program should overwrite it. Print out a warning when doing so. Otherwise, you should not overwrite the file, and you should notify the user that the file is already the most current version. (4 pts)

4. Your program should allocate a new thread to copy each file. Therefore, if the current working directory contains five files, then your program should create five threads to backup each file. Of course, your program should not terminate until all the threads have finished. (5 pts)

5. You must recursively handle subdirectories. In each subdirectory, you must again spawn one thread per file. (3 pts)

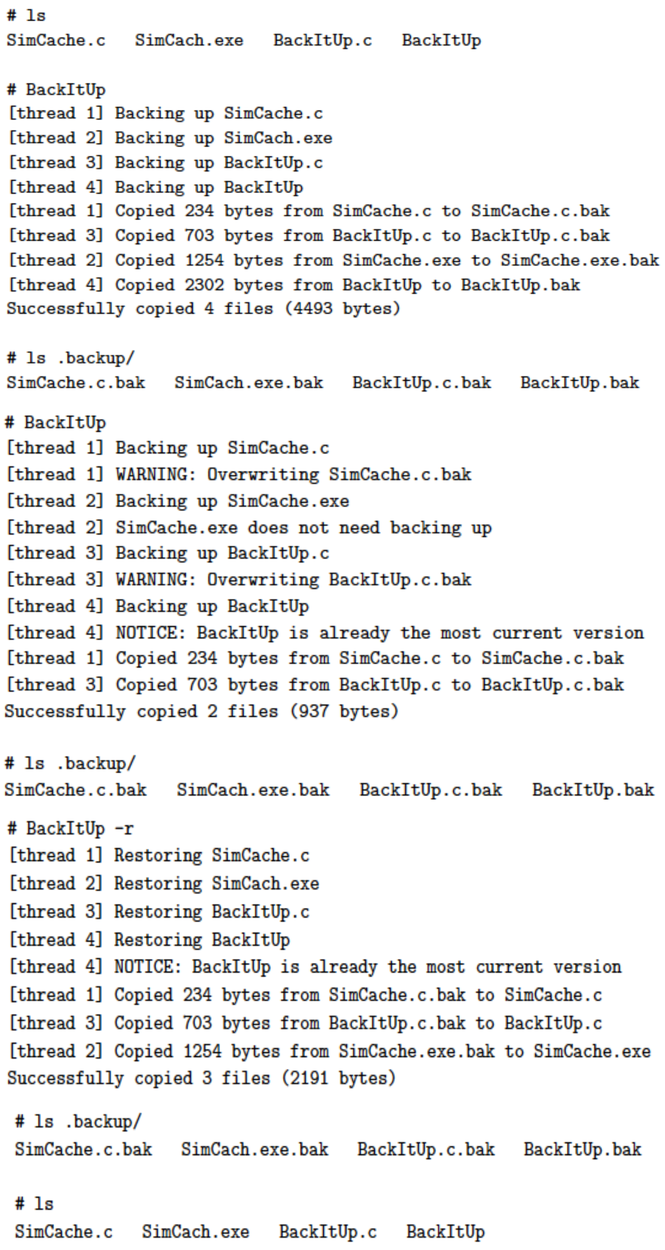
6. If the user invokes *BackItUp* with the optional $-r$ (restore) argument, then it should restore all backup files in the *.backup* directory by copying them to the current working directory (this is the part you have make sure you don't overwrite your existing version of *BackItUp*). Again, it should not restore any files that have a later modification time than the backed up copy. Like before, the restore function should create a separate thread for each file being copied. (5 pts)

7. Notes: You will want to check out the following Unix system calls for C: *stat()*, *opendir()*, and *readdir()*.

In the figure below, you can see the output of *BackItUp* in the following sample interaction.

Deliverables

- You must create a *Makefile* to compile your source. (4 pts)
- You output should be similar to those in the sample interaction. (4 pts)



```
# ls
SimCache.c   SimCach.exe   BackItUp.c   BackItUp

# BackItUp
[thread 1] Backing up SimCache.c
[thread 2] Backing up SimCach.exe
[thread 3] Backing up BackItUp.c
[thread 4] Backing up BackItUp
[thread 1] Copied 234 bytes from SimCache.c to SimCache.c.bak
[thread 3] Copied 703 bytes from BackItUp.c to BackItUp.c.bak
[thread 2] Copied 1254 bytes from SimCache.exe to SimCache.exe.bak
[thread 4] Copied 2302 bytes from BackItUp to BackItUp.bak
Successfully copied 4 files (4493 bytes)

# ls .backup/
SimCache.c.bak   SimCach.exe.bak   BackItUp.c.bak   BackItUp.bak

# BackItUp
[thread 1] Backing up SimCache.c
[thread 1] WARNING: Overwriting SimCache.c.bak
[thread 2] Backing up SimCache.exe
[thread 2] SimCache.exe does not need backing up
[thread 3] Backing up BackItUp.c
[thread 3] WARNING: Overwriting BackItUp.c.bak
[thread 4] Backing up BackItUp
[thread 4] NOTICE: BackItUp is already the most current version
[thread 1] Copied 234 bytes from SimCache.c to SimCache.c.bak
[thread 3] Copied 703 bytes from BackItUp.c to BackItUp.c.bak
Successfully copied 2 files (937 bytes)

# ls .backup/
SimCache.c.bak   SimCach.exe.bak   BackItUp.c.bak   BackItUp.bak

# BackItUp -r
[thread 1] Restoring SimCache.c
[thread 2] Restoring SimCach.exe
[thread 3] Restoring BackItUp.c
[thread 4] Restoring BackItUp
[thread 4] NOTICE: BackItUp is already the most current version
[thread 1] Copied 234 bytes from SimCache.c.bak to SimCache.c
[thread 3] Copied 703 bytes from BackItUp.c.bak to BackItUp.c
[thread 2] Copied 1254 bytes from SimCache.exe.bak to SimCache.exe
Successfully copied 3 files (2191 bytes)

# ls .backup/
SimCache.c.bak   SimCach.exe.bak   BackItUp.c.bak   BackItUp.bak

# ls
SimCache.c   SimCach.exe   BackItUp.c   BackItUp
```

Figure 1: An example output of the sample interaction using *BackItUp*.

- Please submit in the blackboard your source code before 11:59pm on the due date.
- Please include all files necessary for a successful build as a zip file.
- The name of your submission should be *cs460-project4-TeamID.zip*.

Grading: Your submission will be graded by compiling and running it.

- Please make sure your source code can compile. Absolutely no credit if it does not compile.
- Please don't include the binary files. Do a make clean before submission.
- Please don't leave out any files!
- Please don't modify any files you don't need to!
- Please don't send us the meta-information from your revision control system!