CEng 536 Advanced Unix
Fall 2016
HW1
Due: 28/10/2016

In this homework you will write a utility called `synchronize` which gets two directory paths as parameters and recursively check for timestamps and sizes of files under these directories and try to find the set of operations to make these two directories identical.
There will be two modes of operation:

1. In the **union** mode, the purpose is to synchronize two directories such that missing files of a directory is updated with the other and common files will be updated with the most recent (written later) one. Resulting file list will be the union of the both directories.

2. In the **intersect** mode, the purpose is to synchronize two directories in a minimal common set. Only files existing in both directories will be kept and others will be erased. When timestamps are different for two files the older one will overwrite the recent one. (You can think this process as the backtracking to a previous synchronize situation)

Your command syntax will be as follows:
`synchronize [-i] dir1 dir2`

Default mode of operation is **union** and '`-i`' switches to the **intersect** mode of operation.
Your output will be list of `cp`, `rm` and `ln` command lines to make the synchronization.
For example if `ls -ulR` output for the directory `a` and `b` is:

```
./a:
total 4
-rw-r--r--    1 onur  users   1515 Jun 27 01:13 projects.html
-rw-r--r--    1 onur  users   3081 Jun 27 01:12 projects.log
-rw-r--r--    1 onur  users  55344 Jun 27 01:12 projects.pdf
drwxr-xr-x    2 onur  users   4096 Mar 16  2003 zzz

./a/zzz:
total 0
-rw-r--r--    1 onur  users   5819 Sep 29 17:25 hw1.tex
-rw-r--r--    1 onur  users    234 Sep 29 17:00 hw2.tex

./a/xxx:
-rw-r--r--    1 onur  users   5819 Sep 29 17:25 project.tex
-------------------------------------------------------------
```

```
./b:
total 4
-rw-r--r--    1 onur  users   1515 Jun 27 01:15 projects.html
-rw-r--r--    1 onur  users   3081 Jun 26 01:10 projects.log
drwxr-xr-x    2 onur  users   4096 Mar 16  2003 zzz

./b/zzz:
total 0
-rw-r--r--    1 onur  users   5819 Sep 29 17:25 hw1.tex
lrwxrwxrwx    1 onur  users   5819 Sep 29 17:30 hw2.tex -> ./hw1.tex
```

In `synchronize ./a ./b` case, it should output:

```
cp -p ./b/projects.html ./a
cp -p ./a/projects.log ./b
cp -p ./a/projects.pdf ./b
ln -sf ./hw1.tex ./a/zzz/hw2.tex
cp -rp ./a/xxx ./a
```

The order of the commands are not important. In `synchronize -i ./a ./b` case, it should output:

```
cp -p ./a/projects.html ./b
cp -p ./b/projects.log ./a
rm ./a/projects.pdf
rm ./b/zzz/hw2.tex
cp -p ./a/zzz/hw2.tex ./b/zzz
rm -rf ./a/xxx
```

When two timestamps are equal but the file sizes or types are different you should assume larger file has a more recent update timestamp and regular files are always more recent than symbolic links.
You can assume that you will only encounter regular file-symbolic link situation. No special files, socket files, named files and directory/regular file conflicts in the given arguments. Similarly only consider timestamps of regular files, not directories.
You will write your program in C. You will/may be using the following system calls/library functions:
`open(), close(), lseek(), read(), fcntl()`
`stat(), fstat(), lstat(), symlink(), readlink()`
`utime(), chdir(), getcwd(), opendir(), readdir(), closedir()`
So you are basically expected to learn/know these functions and use in a simple Unix application.

Submission details will be announced later. Please ask all questions to:
`news://news.ceng.metu.edu.tr/metu.ceng.course.536/`