

CEng 536 Advanced Unix

Fall 2011

HW1

Due: 16/10/2011

In this homework you will write a utility called `dirsinc` which gets a directory path and a target directory as parameters and synchronize content of the directory on the target in a shadow directory. The tool is a simplified local version of `rshapshot` tool. Your purpose is to take backups of your workspace directory on a different location on disk. But instead of overwriting target, you want to keep multiple versions in an efficient manner. In order to preserve space, you will copy only modified (since last backup) files while hard linking the unmodified ones.

A sample senario: Assume source directory `hw1` contains `a.txt`, `src/b.c` and `src/t.c`. When you first call `dirsinc hw1 /tmp/hw1`, the directory content will be recursively copied to `/tmp/hw1` keeping the timestamps¹. The content on target will be (`ls -lRi` output):

```
/tmp/hw1/:
total 8
33761 -rw-r--r-- 1 onur onur 1009 Oct  4 14:38 a.txt
26710 drwxr-xr-x 2 onur onur 4096 Oct  4 14:39 src
```

```
/tmp/hw1/src:
total 12
33762 -rw-r--r-- 1 onur onur    4 Oct  4 14:39 b.c
33763 -rw-r--r-- 1 onur onur 7799 Oct  4 14:39 t.c
```

Then assume you edited and saved `b.c`, now it is a newer file from the target copy. In your next call to `dirsinc hw1 /tmp/hw1`, first it will move `/tmp/hw1` to `/tmp/hw1.0` and create a new `/tmp/hw1` with the current content of `hw1` directory. However `a.txt` and `t.c` will be hard link to previous content while `b.c` will be the new file. In this way, if you repeat this process, you will have `/tmp/hw1`, `/tmp/hw1.2`, `/tmp/hw1.1`, `/tmp/hw1.0` will have snapshots of the original directory in different times. Thanks to magic of hard linking erasing any snapshot directory will not affect the actual file. The contents after second call to `dirsinc` on target will be:

```
/tmp/hw1:
total 8
33761 -rw-r--r-- 2 onur onur 1009 Oct  4 14:38 a.txt
26710 drwxr-xr-x 2 onur onur 4096 Oct  4 14:51 src
```

```
/tmp/hw1/src:
total 12
33766 -rw-r--r-- 1 onur onur   11 Oct  4 14:51 b.c
33763 -rw-r--r-- 2 onur onur 7799 Oct  4 14:39 t.c
```

Please note that the timestamps of directories are not relevant, they are created at each call, not linked. Your task is:

- Create target directory if it does not exist.
- If it exists, rename it incrementing the last integer (if none start from 0)
- Recursively traverse all files in source directory and for each file:
 - if it is changed or does not exist on previous target copy and save access modes and timestamps (preserving owner and other attributes are not necessary) on target
 - if it does not changed, create a hard link from previous target file to new target directory.
- For each directory during traversal, create a directory with same name on target and recurse.

¹see `utime` man page.

For changed/unchanged decision, use a heuristic. If the file size is changed or last modification time changes the file is changed. If none of them changed, file is assumed unchanged even content is different. Your implementation should treat symbolic links, special files and named pipes as the same on target. In other words when you synchronize a symbolic link, a symbolic link is created/hard linked on target instead of copying the content. Similarly special files and named pipes should be realized as themselves. You are not allowed to call any external unix program or library other than standard C library or standard C++ library in your implementation. You don't need to handle all errors properly like file cannot be read, access permission errors etc. Assume the source directory is fully readable and target directories are writable.

You will write your program in C. You will/may be using the following system calls/library functions: `opendir()`, `closedir()`, `readdir()`, `getcwd()`, `chdir()`, `perror()`, `lstat()`, `link()`, `symlink()`, `mknod()`, `mkfifo()`

Submission details will be announced later. Please ask all questions to:

news://news.ceng.metu.edu.tr:2050/metu.ceng.course.536/

<https://cow.ceng.metu.edu.tr/News/thread.php?group=metu.ceng.course.536>