EECS 2031 SOFTWARE TOOLS, SUMMER 2014
LAB 6

PRZEMYSLAW PAWLUK

1. Objective

Cellphones and other electronic devices must be off while you are in the lab.

Write an ANSI-C program called `lab3.c` that reads lines from standard input, parses them, and classifies the input according to the context set out above and the requirements specified below. Test that your program correctly implements the required functionality

Short Reference of useful Unix commands

First, open a command line window/unix prompt by starting an xterm.

1.1. **Commands related to directories/folders.**

- `ls` lists current directory
- `cd <name>` change current directory to named directory, `cd ..` to move up in the hierarchy, `cd ~` to go back to your home directory
- `pwd` prints location of current directory, i.e., the path where you currently are.
- `mkdir <name>` creates new subdirectory with the given name in current directory

Hint: you can use the tab key for autocompletion of file and directory names. The up/down cursor keys can be used to scroll through the history of commands.

1.2. **Commands related to compiling files.**

- `gcc -o <name> <name>.c` compiles the named source file `"name.c"` into the executable `"name"`. That executable can then be started by typing `"name"`.
- `cat <name>` prints the contents of a file to the terminal.

Note that you will need to re-compile your program before you can test changes.

2. What to do

First, create a few subdirectories:

- create a subdirectory called ”lab6” in 2031
- change the current directory to the newly created directory
- if you print the current directory with `pwd`, the system should show you now something like `/cse/home/ZZZcseZZZ/2031/lab4` (with your CSE account name instead of `ZZZcseZZZ`)

Then create a simple ANSI-C program and compile it.

2.1. Idea.

HeapSort is a method of sorting that uses binary tree. Your task is to use heap sort to sort an input file containing records in the format from the previous lab exercises (shown below). Build a `struct` that will allow you to realize this task.

```
timestamp userID newBallance
```

There is a single Space character separating the 3 pieces of information. The fields are defined as follows:

- The timestamp is an integer with the number of seconds since 00:00, Jan 1, 1970 UTC, which conforms to the standard specification of time in Unix/Linux systems.
- The `userID` is a string, which conforms to the rules for the naming of C variables. It will be at most 31 characters long.
- The `newBallance` as a floating point number (`.2f`) representing new balance on the account.

2.2. Requirements.

1. The program should read file called `input.txt`
2. The program should have a flag (#define FIELD) that takes the value 1 or 2
   - if *SORT==1* then we sort based on userId
   - if *SORT==2* then we sort based on balance
3. The program should print the sorted records to the file `sorted.txt`

Assuming that the program is started with `lab6`, given the following input

```
3600 godzilla1 300
36000 godzilla2 299
36001 godzilla5 2000
36002 godzilla8 0
36003 godzilla10 -10
36004 innocent 69
```

for `SORT=1` your program should create the following output:

```
36004 innocent 69
3600 godzilla1 300
36000 godzilla2 299
36001 godzilla5 2000
36002 godzilla8 0
36003 godzilla10 -10
```

for `SORT=2` your program should create the following output:

```
36003 godzilla10 -10
36002 godzilla8 0
36004 innocent 69
36000 godzilla2 299
3600 godzilla1 300
36001 godzilla5 2000
```

2.3. **Submission.** Submit your work using the submit command. Submit entire folder lab6

```
submit 2031 lab6 lab6
```