Lab Session 1

Submission deadline: Feb 4, 11:59pm

Please submit your lab results through CatCourses, including Makefile and a short report (up to one page).

Task 1: Learning how to write a Makefile

Learn how to write Makefile from this simple but instructive tutorial (http://nuclear.mutantstargoat.com/articles/make/).

In this task, you will write a Makefile for a program. You can find the program from CatCourses (see Files/lab_materials/lab1/task1).

You need to put the .h file in an "include" directory, the source code in a "src" directory. You need to hide .o files in an "obj" directory. Also, include a rule for cleaning up your source and object directories if you type `make clean`.

Task 2: **Using gettimeofday to measure performance**

gettmeofday is a common method to measure program execution time in Linux/Unix environment. Read the Linux man page for gettimeofday (http://linux.die.net/man/2/gettimeofday).

In this task, you will measure the execution time of a specific loop in a sample program. You can find the program from CatCourses (see Files/lab_materials/lab1/task2). Pick up three different input and report the execution time of the loop.

Task 3: **Performance profiling with gprof**

gprof is a common tool to profile your program. Here is a simple but instructive tutorial for gprof (http://web.eecs.umich.edu/~sugih/pointers/gprof_quick.html).

In this task, you will have two sub-tasks to profile a program with gprof. Download the program from Catcourses (Files/lab_materials/lab1/task3).

* To untar the program tarball

```
tar –xzvf task3.tgz
```

* To compile the program, simply run Make

* To run the program, simply run lulesh2.0

Task 3.1:

Use gprof to find the top 6 most time-consuming routines. The execution time of a routine refers to the self-call time, not including the time in the children routines. You will have to change the Makefile in order to use gprof.

Task 3.2:

Explain the call graph of these 6 routines, including their parents and children routines.