COP4530 Recitation Fall 2012
Week 9

Objective

1. Profiling code

Sample code can be copied from `~cop4530/fall12/recitation/rect9/` directory. The files are: `main.cpp, main2.cpp, makefile, and README`.

Introduction

Code profiling provides the runtime of your program by time spent in function calls. As a programmer, it may be important to optimize your program's runtime. Utilizing the information provided by a profiler you can figure out which parts of your code need to be optimized.

GNU Profiler

The GNU proiler, `gprof`, is licensed under the GNU public license; therefore freely available. `gprof` is available on the linprog servers.

Profiling code

If you want to use `gprof` to profile your code you must compile with the `–pg` flag.

```
g++ –Wall –pedantic –g –pg
```

Then, you run the program as normal. The runtime of the compiled code will be slower than normal as it records the profiling information. The produced profiling information is stored in a newly created file called `gmon.out`.

Run `gprof` to analyze the profiling information; by default the information is outputted to standard out (you may want to redirect the output).

```
gprof [executable name]
gprof [executable name] > analysis.txt
```

References

1. gprof: http://www.cs.utah.edu/dept/old/texinfo/as/gprof.html