ENCE464 - Profiling

GProf is a Unix profiling system. It uses a combination of compile-time instrumentation (built into GCC), and run-time sampling to build reports on the execution details of a program.

OVERVIEW

- Check out the example code from https://eng-git.canterbury.ac.nz/are111/ence464-profile-lab
- Read & make sure you understand what it is doing
- Which do you expect to take more time - determining $\sqrt{x}$, or calculating the x^2 ?
- Build & run the 'profile_example' code.
  - This will produce a new file called 'gmon.out'.
  - Use `gprof` to determine where the majority of the time is being spent
    - `make profile_example.prof`

TODO

- Add gprof instrumentation to the naive implementation of the assignment, and generate a report for it.
- Read the documentation about gprof statistical sampling, and confirm that you are able to combine multiple execution runs into a single report: https://sourceware.org/binutils/docs/gprof/Sampling-Error.html#Sampling-Error

FURTHER READING

- GProf supports a range of different options, see https://www.tutorialspoint.com/unix_commands/gprof.htm for full details
- There is a slightly different profiling tool available on Linux called 'perf'. This requires kernel-level instrumentation to do statistical profiling, and unfortunately requires elevated permissions that are not available on the lab machines. If you have full access to a Linux system, try using the following commands to get the 'perf' report for the program.

```
sudo perf record ./profile_example
perf report
```