Profiling Workshop Notes

The GNU profiler is a tool that analyses source codes to determine which parts of the code are computationally intensive for the purpose of optimisation. Effort can be focused on "hot spots" to provide the greatest speed up.

The main gprof documentation web site is:

http://sourceware.org/binutils/docs/gprof/

Commands for Profiling

The command for profiling your code for gprof is:

```
gcc -pg program.c -o program
./program
gprof ./program
```

The command for profiling your code for gcov is:

```
gcc -fprofile-arcs -ftest-coverage program.c -o program
./program
gcov program.c
cat program.c.gcov
```

Practical 1 - Profiling with gprof

If you have not already done so, download the example programs onto your home directory using the command:

```
wget http://grace-head00.uea.ac.uk/grace-docs/dpo_examples.tar
tar -xvf dpo_examples
cd dpo_examples
```

1. Compile `program4.c` with profiling information on;
2. What percentage of program execution time was spent in a) `step` b) `nseq` functions?
3. How many times was `step` called and how many times was `nseq` called?
4. From the call graph, which function calls the `step` function?

Practical 2 - Profiling with gcov

1. Compile `program5.c` for profiling with gcov and create the gcov output file;
2. Which line was not executed at all and why?
3. Which line was executed the most and why?