COLBY COLLEGE
Department of Mathematics
and Computer Science

**CS357** **Allen B. Downey**
**Spring 1998**

## Assignment 2: Caching

**Due:** Tuesday 24 February

The purpose of this assignment is to get you to explore the use of memory caches on real machines, and to give you some practice with UNIX utilities like awk, grep and xgraph.

### Find a partner

1. If you have some experience with UNIX, find a partner who doesn't. If you don't, find someone who does.

### Get the program and compile it

1. From the class web page, get the "Cache-testing program", unpack it and compile it.
2. Read the code. It's actually C, not C++, but for codes like this it almost doesn't matter. There are some system calls and things here that we will probably want to use later, so keep this program around for future reference.
3. Run the program on one of the SGI workstations, redirecting the output into a file. For example, `cache > data` will put the output in a program named data. The program takes a few minutes to run.

### Start poking around the data

1. Use more or emacs to read the data file. If you want to know how many lines there are, use wc.
2. You can use awk to extract particular lines from the file. For example:

   ```
   awk '$2 == 4096' < data
   ```

   will print only the lines where the second column is 4096.

   ```
   awk '$2 == 4096 {print $4}' < data
   ```

   selects the same lines as the previous command, but instead of printing the whole line, it only prints the fourth column.

3. You can pipe the output of one command into another. For example

```
awk '$2 == 4096 {print $4}' < data | wc
```

   Counts the number of lines (and words and characters) that have 4096 in the second column.

**Use xgraph to plot some data**

1. We have a program called xgraph that does some simple plotting. You may find it useful for exploring this data set. If you have another graphing program you prefer, though, feel free to use it instead.
2. There should be a man page for xgraph available. You may want to print it. You can use man with the `-t` option to do that.

**Figure out the cache system**

1. Make whatever plots you need to infer the cache size and the block size of the cache. If you can, try to infer the replacement policy (associative, direct-mapped, something else?).
2. Run the program again on iris12, collecting a new data set. Compare the cache structure on the two machines.

**What to turn in**

I would like a single report from each pair, with both names on it, and an indication of which partner is the UNIX expert and which is the UNIX neophyte.

The report should include whatever graphs you used, and it should explain howyou used the graphs (or any other means of exploratory data analysis) to infer cache characteristics. It should compare the cache structure on the SGI workstations with that of the Power Challenge (iris12).

Report data sparingly. Do not overwhelm me with uninterpreted graphs. Use more words than numbers. Label the axes on all graphs. Write on the graphs. Explain everything.