CSCI 3323 (Principles of Operating Systems), Fall 2021

Homework 3b

Credit: 10 points.

1 Reading

(No real reading for this assignment.)

2 Programming Problems

For this assignment you won't actually be writing code, but you'll be doing something on a computer, so I'm calling it a programming homework. Turn it in in the same ways as for the written homeworks, by putting it in your course "TurnIn" folder on Google Drive. I think for most people it will be easier to use one of the department Linux machines, but you can use anything that provides suitable environment.

1. (10 points) One of the problems in Homework 3a asks why access time to an array can depend on whether it's accessed row by row or column by column. It's interesting to try to watch this in action, so I wrote a little program to try it out, allowing users to vary the size of the array. I tried the program on several different computers and for several different sizes and observed that it did matter, and how much it mattered varied too. Your mission for this problem is just to compile and run this program on at least two different computers and several different problem sizes and report results.

   First about the program:

   - It doesn't do anything very meaningful, but I tried to, well, rig it to keep the compiler's optimizer from deciding none of the calculations would have any measurable effect and so could be skipped. (Don't laugh — it's allowed, and I think I've observed it.)
   - It takes command-line arguments that let you specify access by rows or columns, dimensions (rows and columns), and how many times to repeat the whole calculation (so that for small arrays the total calculation will be big enough to bother timing).
   - I do access by rows and access by columns in separate executions because I wanted to start each of the two methods with a clean slate.

   I spent probably too much time trying this out on different computers and with different problem sizes, but I observed many interesting things:

   - Some computers were faster than others (no surprise); some had more memory than others.
   - For all of them access by row was faster, but how much faster varied depending on array size; the most meaningful comparison I thought was ratio of time for access by row to time for access by column, and I observed basically two patterns.

   What I want you to do is this:

- **Download and compile the program**:

  You'll need files twoD-pgm.c. timer.h.

  I recommend compiling with `-O`, since to me that seems only reasonable for code whose performance is of interest. (I thought it was safest and most fair to recompile when switching to a different computer, but that may not necessary.)

- **Do performance experiments:**

  Try it out on several different computers, preferably ones that will be different in some meaningful way — we replace classroom/lab machines one room at a time, so one of the Xena machines is probably much like another, but a Xena will likely be different from a Janus. Groups include:

  - Xena00 through Xena21 (CSI 388, or remote access).
  - Janus00 through Janus24 (CSI 257, or remote access).
  - Atlas00 through Atlas09 (CSI 270A, or remote access).
  - Computers in CSI 270L (also remote access).
  - Pandora00 through Pandora08 (server room, remote access). Note that these computers have more memory so can be tried with bigger arrays.
  - Dione (server room, remote access). Note that this one is much older and so noticeably slower but uses a different memory architecture and also has more memory than the classroom computers.
  - Your own machine, but note that the function to report time is Linux only, so this may not work.

  (I'm not sure I'd recommend Dias01 through Dias05 because they're virtual, but that might be interesting too.) For most meaningful results, you should try to make sure you're the only person using a machine. (Linux command `w` will tell you.)

  Also try out at least two array sizes on each computer. I recommend trying increasing sizes until you try something too big for the machine (as reported by the program). Some things to note:

  - Does ratio of time to access by rows to time for access by columns time stay about the same, or does it change for some problem size? (I found that it changed.)
  - Does ratio of time to access by rows to time for access by columns time vary between machines? (I found that it did, though there were some similarities.)
  - Does execution time seem to increase linearly with number of repeats?

  Don't feel like you have to spend hours on this, but I think you can't get meaningful information unless you let each experiment run at least 5 seconds and some run for up to 5 minutes or so. (For once a good use of multitasking?)

  You can get full credit if you try at least two computers and half a dozen problems (counting access by rows and access to columns as a single problem). If you do more I'll give up to 2.5 extra points of extra credit.

- **What to turn in:**

  Turn in reports of your experiments in whatever form you think will be reasonably readable. A simple text-based or word-processed table may be easiest.

3 Essay and pledge

Include with your assignment the following information.

For programming assignments, please put it a separate file. (I strongly prefer plain text, but if you insist you can put it in a PDF — just no word-processor documents or Google Drive links please.) For written assignments, please put it in your main document.

3.1 Pledge

This should include the Honor Code pledge, or just the word "pledged", *plus* at least one of the following about collaboration and help (as many as apply). Text *in italics* is explanatory or something for you to fill in; you don't need to repeat it!

- I did not get outside help *aside from course materials, including starter code, readings, sample programs, the instructor.*
- I worked with *names of other students* on this assignment.
- I got help with this assignment from *source of help — ACM tutoring, another student in the course, etc. (Here, "help" means significant help, beyond a little assistance with tools or compiler errors.)*
- I got help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.. (Here too, you only need to mention significant help — you don't need to tell me that you looked up an error message on the Web, but if you found an algorithm or a code sketch, tell me about that.)*
- I provided help to *names of students* on this assignment. *(And here too, you only need to tell me about significant help.)*

3.2 Essay

This should be a brief essay (a sentence or two is fine, though you can write as much as you like) telling me what if anything you think you learned from the assignment, and what if anything you found interesting, difficult, or otherwise noteworthy.