CPSC 457 S17 - Assignment 1

Due date: **Wednesday, May 24, 2017 at 11:59pm**.
Individual assignment. No group work allowed.
Weight: 6% of the final grade.

Link to the assignment web page:

https://sites.google.com/site/pfederlcpsc457spring2017/assignments/assignment-1

Q1 - Written question (5 marks)

Assume that a CPU pipeline has three stages, as shown in class, and each stage is handled by a separate unit, namely, fetch unit, decoding unit, and execute unit. For every instruction, the fetch unit takes 10 nsec, the decoding unit takes 0.5 nsec, and the execute unit takes 1 nsec. How many instructions per second can this CPU execute on average? Is it possible to improve the performance of this CPU? If so, how? If not, why?

Q2 - Written question (5 marks)

What are the benefits of using a virtual machine from the operating system's perspective and from the user's perspective?

Q3 - Written question (5 marks)

Define interrupts and traps and then describe the differences. Explain why they are handled in kernel mode instead of user mode.

Q4 - Written question (5 marks)

On the assignment page you will find a C++ program called `readFile.cpp`. This program reads in a text file, specified on the command line, and outputs it's contents to standard output. It's functionality is very similar to the `cat` utility program. In order to answer this question, you will need to download this program, compile it and run it in a Linux environment.

Once you have the C++ program running, you will compare it's performance to the `cat` program. To this end you will use the `time` program, which can time the execution of another executable. For example, if you compile the C++ program into an executable called `readFile`, you can run the following commands to time both programs:

```
time ./readFile sample.txt
time cat sample.txt
```

You can find the `sample.txt` file on the assignment web page. Answer the following questions:

- What are the outputs of the time commands? Copy/paste this from the shell.
- How much time did the C++ program and `cat` spend in the kernel mode and user mode, respectively?
- Why is the `cat` program faster than the C++ program?

You can get more information about the `cat` and `time` utilities from their manual pages, which are accessible using the following commands:

```
man cat
man time
```

Q5 - Programming question (5 marks)

Can you improve the performance of the `readFile.cpp` program from the previous question so that it is closer to the performace of the `cat` utility program? Write your solution in a file called `myCat.cpp` and submit it together with your report. In the report, include the timings of your program and compare it to the timings you obtained for `readFile.cpp` and `cat`.

Submission

You should submit two files for this assignment:

- Answers to the written questions combined into a single file, called either `report.txt` or `report.pdf`. Do not use any other file formats.
- Your solution to Q5 in a file called `myCat.cpp`.

Since D2L will be configured to accept only a single file, you will need to submit an archive, eg. `assignment1.tar.gz`. To create such an archive, you could use a command similar to this:

```
tar czvf assignment1.tar.gz myCat.cpp report.pdf
```

General information about all assignments:

1. Due time: All assignments are due at 23:59 on the due date listed on the assignment. Late assignments or components of assignments will not be accepted for marking without approval for an extension beforehand. What you have submitted in D2L as of the due date is what will be marked.
2. Extensions may be granted for reasonable cases, but only by the course instructor, and only with the receipt of the appropriate documentation (e.g. a doctor's note). Typical examples of reasonable cases for an extension include: illness or a death in the family. Cases where extensions will not be granted include situations that are typical of student life, such as having multiple due dates, work commitments, etc. Forgetting to hand in your assignment on time is not a valid reason for getting an extension.
3. After you submit your work to D2L, make sure that you check the content of your submission. It's your responsibility to do this, so make sure that you submit your assignment with enough time before it is due so that you can double-check your upload, and possibly re-upload the assignment.
4. All assignments should include contact information, including full name, student ID and tutorial section, at the very top of each file submitted.
5. Assignments must reflect individual work. Group work is not allowed in this class nor can you copy the work of others. For further information on plagiarism, cheating and other academic misconduct, check the information at this link: http://www.ucalgary.ca/pubs/calendar/current/k-5.html.

6. You can and should submit many times before the due date. D2L will simply overwrite previous submissions with newer ones. It’s better to submit incomplete work for a chance of getting partial marks, than not to submit anything.
7. Only one file can be submitted per assignment. If you need to submit multiple files, you can put them into a single container. The container types supported will be ZIP and TAR. No other formats will be accepted.
8. Assignments will be marked by your TAs. If you have questions about assignment marking, contact your TA first. If you still have questions after you have talked to your TA then you can contact your instructor.