

CPSC457W2019 - Assignment 1

Due date: **Friday, February 1, 2019 at 11:30pm.**

Individual assignment, no group work allowed.

Weight: 5% of the final grade.

Q1 - Written question (5 marks)

Assume that a CPU cycle 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 6 nsec, the decoding unit takes 4 nsec, and the execute unit takes 2 nsec.

- How many instructions per second can this CPU execute on average if the stages are not parallelized?
- How many instructions per second can this CPU execute on average if all stages are operating in parallel?

Q2 - Written question (5 marks)

Describe one benefit of using virtual machines for each of the following:

- for a company;
- for a programmer;
- for a regular user; and
- for a system administrator.

Q3 - Written question (5 marks)

- Define interrupts.
- Define traps.
- Describe the differences between interrupts and traps.
- Explain why interrupts and traps 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 `countLines.cpp`. This program reads in a text file, specified on the command line, counts the number of lines in the file and outputs the result to standard output. Its functionality is very similar to the `'wc'` utility program when used with the `'-l'` option. In order to answer this question, you will need to download `countLines.cpp` program, then compile it:

```
g++ countLines.cpp -o countLines
```

and then run it in a Linux environment, eg.:

```
./countLines romeo-and-juliet.txt
```

Once you have the C++ program running, you will compare its performance to the command line utility `'wc -l'`. To this end you will use the `time` command, which measures the execution of another executable. Run the following commands to time both programs:

```
time ./countLines romeo-and-juliet.txt
time wc -l romeo-and-juliet.txt
```

Answer the following questions:

- What are the outputs of the time commands? Copy/paste this from the terminal output to your report.
- How much time did the C++ program and 'wc' spend in the kernel mode and user mode, respectively?
- Why is the 'wc' program faster than the C++ program?

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

```
man wc
man time
```

You can download the `romeo-and-juliet.txt` file from the assignment web page.

Q5 - Programming question (5 marks)

Improve the `countLines.cpp` program from the previous question so that its performance is closer to the 'wc' utility program. Write your solution in a file called `myWc.cpp` or `myWc.c` 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 `countLines.cpp` and `wc`.

Whether you use C or C++, you are only allowed to use the following low level file APIs: `open()`, `close()` and `read()`. You cannot use any other APIs, such as `fopen()`, `fread()` or C++'s streams.

Your program must run on the CPSC Linux machines. If your TA is unable to run your code on the Linux machines, you will receive 0 marks for the question.

Submission

You should submit two files for this assignment:

- Answers to the written questions combined into a single file called `report.pdf`. Do not use any other file format.
- Your solution to Q5 in a file called `myWc.cpp` or `myWc.c`.

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 myWc.cpp report.pdf
```

General information about all assignments:

1. All assignments are due on the date listed on D2L. 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 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.
9. All programs you submit must run on the CPSC Linux machines. If your TA is unable to run your code on the Linux machines, you will receive 0 marks for the relevant question.