CPSC457 Spring 2020 - Assignment 1

Due date is posted on D2L.
Individual assignment, no group work allowed.
Weight: 15% of the final grade.

Q1 - Written question (4 marks)

Consider a CPU with a 5 stage instruction cycle. For every instruction, the five stages take 3ns, 6ns, 1ns, 10ns and 5ns, respectively. Answer the following:

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

Q2 - Written question (4 marks)

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

a) from a company's perspective;
b) from a programmer's / developer's perspective;
c) from a regular user's perspective; and
d) from a system administrator's perspective.

Q3 - Written question (4 marks)

a) Define interrupts.
b) Define traps.
c) What are some key differences between hardware interrupts and traps?
d) Why are interrupts handled in kernel mode instead of user mode?

Q4 - Written question (6 marks)

On the assignment page you will find a C++ program called `simple_wc.cpp`. This program reads characters from standard input, counts the number of characters, words and lines in the input, and outputs the counts to standard output. Its functionality is identical to the `'wc'` utility program when used without any parameters. In order to answer this question, you will first need to download `simple_wc.cpp` and then compile it:

```
$ g++ simple_wc.cpp –o simple_wc
```

You can then run it:

```
$ ./simple_wc < a-tale-of-two-cities.txt
   16272  138883  804335
```

The output above indicates that the file `a-tale-of-two-cities.txt` contains 16272 lines, 138883 words, and 804335 characters. You can verify that the `'wc'` utility produces the same results, although the spacig between the number might be slightly different:

```
$ wc < a-tale-of-two-cities.txt
 16272 138883 804335
```

Once you have the C++ program running, you will compare its performance to the 'wc' program. 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 ./simple_wc < a-tale-of-two-cities.txt
$ time wc < a-tale-of-two-cities.txt
```

Answer the following questions:

a) What are the outputs of the time commands? Copy/paste this from the terminal output to your report.
b) How much time did the C++ program and 'wc' spend in the kernel mode and user mode, respectively?
c) 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 a-tale-of-two-cities.txt file from the assignment web page.

Q5 - Programming question (15 marks)

Improve the simple_wc.cpp program from the previous question so that its performace matches the 'wc' utility program as closely as possible. Write your solution in a file called myWc.cpp or myWc.c and submit it together with your report.

Requirments:

- Your program must read input from standard input.
- Whether you use C or C++, you are only allowed to use the read() system call wrapper to read standard input. You cannot use any other APIs, such as mmap(), fopen(), fread(), fgetc(), or C++'s streams. Hint: use read() with a buffer size of 1MiB.
- Your program must be able to handle input of size up to 2^63-1 bytes.
- For the definition of what 'word' means, read the man page for wc.
- Your program must run on linux.cpsc.ucalgary.ca server.

Q6 - Written question (2 marks)

Measure the performance of your program from Q5 and compare it to the timings you obtained for simple_wc.cpp and wc in Q4. Answer the following:

a) Is your program from Q5 faster than simple_wc.cpp? Why do you think that is?
b) Is your program faster or slower than 'wc' and why?

Justify your answers by using the strace utility with the '-c' command line option. Include the output of strace in your report.

Submission

You will submit two files for this assignment to D2L:

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

General information about all assignments:

1. All assignments are due on the date listed on D2L. Late sumissions will be not be marked.
2. Extensions may be granted only by the course instructor.
3. After you submit your work to D2L, verify your submission by re-downloading it.
4. You can 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. Please bear in mind that you cannot re-submit a single file if you have already submitted other files. Your new submission would delete the previous files you submitted. So please keep a copy of all files you intend to submit, and resubmit all of them every time.
5. 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.
6. All programs you submit must run on `linux.cpsc.ucalgary.ca`. If your TA is unable to run your code on the Linux machines, you will receive 0 marks for the relevant question.
7. Assignments must reflect individual work. 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.
8. Here are some examples of what you are not allowed to do for individual assignments: you are not allowed to copy code or written answers (in part, or in whole) from anyone else; you are not allowed to collaborate with anyone; you are not allowed to share your solutions with anyone else; you are not allowed to sell or purchase a solution. This list is not exclusive.
9. We will use automated similarity detection software to check for plagiarism. You submission will be compared to other students (current and previous), as well as to any known online sources. Any cases of detected plagiarism or any other academic misconduct will be investigated and reported.