CPSC457W2019- Assignment 1

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

Professor: Pavol Federl, pfederl@ucalgary.ca, ICT 742

Lectures: TuTh 9:30am - 10:45am, ST 145

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.

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

- If the stages are not parallelized, they would be running one at a time (sequentially). This means one instruction must finish before the next starts. Therefore, the total time taken by one instruction is the time per instruction for the CPU.
- For one instruction, the time is: fetch + decode + execute = 6nsec + 4nsec + 2nsec = 12nsec. To get the instructions per sec, we must do 1/12nsec = 1/(12E-9) sec = **83333333.3333** instructions per sec.

b) How many instructions per second can this CPU execute on average if all stages are operating in parallel?

- In parallel, the pipeline is as fast as the slowest step. Therefore, for one instruction, the time is: fetch = 6nsec. To get the instructions per sec, we must do 1/(6E-9) sec = **166666666.667** instructions per sec.

Q2 - Written question (5 marks)

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

a) for a company;

- A company could also use virtual machines to easily manage and revoke licenses, as opposed to having them be associated with a physical system. A company could also forego several high-performing systems and have one system run multiple virtual machines instead.

b) for a programmer;

- a programmer may want to test their program on an OS that their computer does not have installed, i.e. Windows or Linux. If they use a virtual machine, they won't need to buy a new system, or dual boot their existing one.

c) for a regular user; and

- They may want to see if a file they've received is a virus. They can run the virus on a virtual machine (mostly) without fear of their computer getting bricked by a malicious file.

d) for a system administrator.

- A system administrator might use virtual machines to be able to remotely access user systems that all run on the same server. They could also have backup images for a corrupted/new user system.

Q3 - Written question (5 marks)

a) Define interrupts.

- An interrupt is something generated to interrupt the CPU. It is used to invoke a kernel routine defined by the OS. Once an interrupt happens, it switches the CPU to kernel mode. Once it's done, CPU switches back. Generally, interrupts refer to hardware interrupts, where exceptions/traps refer to software interrupts.

b) Define traps.

- A trap is an intentional software interrupt, used to access kernel mode to execute a predefined routine. They are used by OS's to implement system calls.

c) Describe the differences between interrupts and traps.

- Traps are generated on purpose, to be able to use a kernel routine. Interrupts are generally unintentional. Traps are synchronous to the CPU, whereas interrupts are asynchronous. Interrupts (if referring to hardware interrupts) are generated externally, like I/O or timers, whereas traps occur via special instructions.

d) Explain why interrupts and traps are handled in kernel mode instead of user mode.

- System calls inherently cannot be executed in user mode, for one. To be able to get the origin of the interrupt, or to execute the routine the trap was for, system calls are needed. Therefore, the CPU must switch to kernel mode to do these calls.

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:

a) What are the outputs of the time commands? Copy/paste this from the terminal output to your report.

```
arebe@DESKTOP-578UJC1 /cygdrive/c/Users/arebe/OneDrive/OneDocs/School/Winter
2018/CPSC457/Assignments/Assignment 1
$ time ./countLines.exe romeo-and-juliet.txt
4853 romeo-and-juliet.txt

real    0m0.827s
user    0m0.062s
sys     0m0.703s

arebe@DESKTOP-578UJC1 /cygdrive/c/Users/arebe/OneDrive/OneDocs/School/Winter
2018/CPSC457/Assignments/Assignment 1
$ time wc -l romeo-and-juliet.txt
4853 romeo-and-juliet.txt

real    0m0.078s
user    0m0.000s
sys     0m0.015s
```

b) How much time did the C++ program and ‘wc’ spend in the kernel mode and user mode, respectively?

- The C++ program spent 0.703 seconds in kernel mode and 0.062 seconds in user mode.
- The ‘wc’ program spent 0.015 seconds in kernel mode and 0.000 seconds in user mode.

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 romeo-and-juliet.txt file from the assignment web page.

- The wc program uses fopen() and getc(), whereas the C++ program uses open() and read(). I’m not sure if this difference would make wc that much faster.
- The C++ program also goes character by character to search for a ‘\n’ character, using read(). Read() uses a system call, and system calls are slow.
- Proportionally, the C++ program spends much more time in user than wc does. Entering and exiting kernel mode is slow, as one must save and restore registers and other values.

Q5 - Programming question (5 marks)

myWc.cpp is included with this file in the same folder (hopefully).

Compared to the times of the wc command, myWc performed almost the same. This is especially true if we compare it to the timing of countLines, to which both wc and myWC outperformed by ~25 times.

Here are the outputs to the time command.

```
arebe@DESKTOP-578UJC1 /cygdrive/c/Users/arebe/OneDrive/OneDocs/School/Winter
2018/CPSC457/Assignments/Assignment 1
$ time ./countLines.exe romeo-and-juliet.txt
4853 romeo-and-juliet.txt

real    0m0.794s
user    0m0.109s
sys     0m0.702s

arebe@DESKTOP-578UJC1 /cygdrive/c/Users/arebe/OneDrive/OneDocs/School/Winter
2018/CPSC457/Assignments/Assignment 1
$ time ./myWc.exe romeo-and-juliet.txt
4853 romeo-and-juliet.txt

real    0m0.034s
user    0m0.000s
sys     0m0.015s

arebe@DESKTOP-578UJC1 /cygdrive/c/Users/arebe/OneDrive/OneDocs/School/Winter
2018/CPSC457/Assignments/Assignment 1
$ time wc -l romeo-and-juliet.txt
4853 romeo-and-juliet.txt

real    0m0.031s
user    0m0.000s
sys     0m0.015s

arebe@DESKTOP-578UJC1 /cygdrive/c/Users/arebe/OneDrive/OneDocs/School/Winter
2018/CPSC457/Assignments/Assignment 1
$ /Assignment 1
$
```