Juan Luis de Reiset
UCID: 30050167

CPSC 457 W2019 - Assignment 1

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?

We know that the fetch unit takes 6 nsec, the decoding unit takes 4 nsec and the execution unit takes 2 nsec. Then to execute each instruction takes 6+4+2 = 12 nsec. Given that each second has 1*10^9 nsec, hence the machine can execute **83,333,333** instructions per seconds. (When the stages are not in parallel)

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

Again, fetch unit takes 6, nsec, decoding unit takes 4 nsec and execution unit takes 2 nsec. Since this time the three stages are in parallel the execution time of the instructions is bounded to the slowest performing unit which in this case is the Fetching unit, thus each instruction takes 6 nsec to execute. Hence the machine can execute **166,666,666** instructions per second. (when the stages are in parallel).

Q2 - Written question (5 marks)

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

a) for a company;

Once companies start to expand, they typically need more physical equipment. As they add operations or start new projects they tend to add a new server. Buying, running and maintaining the servers eventually becomes very expensive. Virtual machines are a practical alternative to setting up and running numerous servers helping reduce the cost for the company.

b) for a programmer;

As a programmer one can run unsafe programs in a virtual machine rather than in the main OS in order to prevent greater damage done to main OS.

.

c) for a regular user;

-The user is able to try different OSes, in creative or even damaging ways, without buying additional expensive equipment.

d) for a system administrator.

-The system administrator may benefit from using a virtual machines since, they allow for reducing cost of setting a server for the company and all maintenances are done in one (more powerful) server instead of multiple (less powerful) servers.

Q3 - Written question (5 marks)

Juan Luis de Reiset
UCID: 30050167

a) Define interrupts.

In system programming, an interrupt is a signal to the processor emitted by software or hardware indicating an event that needs to be handled immediately. An interrupt alerts the processor to a high-priority condition that requires the interruption of the current code the processor is executing.

b) Define traps.

A trap is special instruction that switches from user mode to kernel mode and invokes a condition predefined by the Operating System. Traps are triggered by some unusual condition from user mode and will pause an application to execute the kernel routine. Once it is finished, user mode and the application are restored.

c) Describe the differences between interrupts and traps.

1. The difference between interrupts and traps is that interrupts are signals generated by external events, which includes the computer timer and I/O, that are delivered to the CPU. Traps are internal events invoked by error conditions or system calls.
2. Additionally Interrupts are asynchronous with the CPU while traps are synchronous with the activity of the CPU.
3. Traps occur as a the result of a machine instructions, which means they are predictables while interrupts are produced by unexpected events.

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

User mode does not allow for access to hardware and has a limited instruction set, whereas kernel mode has full access to both. This way the interrupts and traps are able to send a signal to the CPU and invoke a kernel procedure in an appropriate way.

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
```

Juan Luis de Reiset
UCID: 30050167

Answer the following questions:

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

	countLines	wc -1
Real	0m0.005s	0m0.002s
User	0m0.002s	0m0.001s
Sys	0m0.002s	0m0.001s

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

- The C++ program spend 0.002 seconds in kernel mode and in user mode as well
- ‘wc -l’ spend 0.002s on kernel mode and didn't spend any time on user mode

c) Why is the `'wc'` program faster than the C++ program?

- ‘wc’ program is faster than the C++ program (countLines.cpp) because it makes few system calls. countlines compares every char of the line to search for a ‘\n’ (ascii for end of line), that way the program is forced to do many calls to the system.

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`.

	myWc	countLines	wc -l
Real	0m0.003s	0m0.005s	0m0.002s
User	0m0.002s	0m0.002s	0m0.001s
Sys	0m0.001s	0m0.002s	0m0.001s

During the execution of myWc.cpp we see that the overall time spend on kernel is less than the countLines c++ program. This values show to be closer to wc -l.

Juan Luis de Reiset
UCID: 30050167

Submission

You should submit two files for this assignment:

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`.

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
```