Steven Canon-Almagro
10155792
T04

CPSC 457 Assignment 1

Q1

a) How many instructions per second can this CPU execute on average if the stages are not parallelized?
For 1 line of code to execute in this CPU it would take 12 nanoseconds, fetch = 6 nanoseconds, decoding = 4 nanoseconds, and execute = 2 nanoseconds for a total of 12 nanoseconds. Dividing $1x10^9$ by 12 you get an average of 83333333 instructions per second.

b) How many instructions per second can this CPU execute on average if all stages are operating in parallel?
The CPU is operating in parallel it can only execute as fast as its slowest unit which is 6 nanoseconds. Dividing $1x10^9$ by 12 would give you an average of 166666666 instructions per second.

Q2

a) For a company
Business' can save a lot of money by running multiple servers on one very powerful computer instead of having multiple computers running one server on each.

b) For a programmer
A programmer can run multiple OSes on one computer and be able to code for windows or linux depending on what is needed

c) For a regular user
You can run unsafe program and the real computer would be safe from any malicious things that may come from it

d) For a system administrator
This is very helpful for system administrators because instead of needed multiple computers to keep track of the computers in their team he can make multiple virtual machines on one computer than can do the same job and more efficient

Steven Canon-Almagro
10155792
T04

Q3

a) Define interrupts

An Interrupt is usually used for hardware interrupts, like peripheral devices. Interrupts prompts the CPU that an external device or app has completed its tasks. The CPU will then handle the interrupt with the right handler in kernel mode

b) Define traps

A trap is an exception in a user process. Trap occurs (usually) via special instruction, or illegal instructions like dividing by zero. The purpose is to execute predefined routine in kernel mode Operating systems can use traps to implement system calls. Traps allow implementation of a system code/ execute system mode without loss of program continuity.

c) Describe the differences between interrupts and traps

Interrupts are hardware interrupts and traps are invoked through software. When invoking a hardware interrupt all other devices interrupts are disabled but the same can't be said for traps thus allowing an execution of a program without the loss of program continuity.

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

The reason traps and interrupts need to be handles in kernel mode is that hardware isn't accessible in user mode and user mode doesn't have access to the full set of instructions, kernel mode does have full access.

Steven Canon-Almagro
10155792
T04

Q4

a)

time ./countLines romeo-and-juliet.txt

```
real    0m0.233s
user    0m0.051s
sys     0m0.180s
```

time wc -l romeo-and-juliet.txt

```
real    0m0.002s
user    0m0.002s
sys     0m0.001s
```

b) The counting lines program spent 0.180s in kernel mode
The wc command spent 0.001s in kernel mode

c) The wc program is faster then the counting lines program because countingLines.cpp is calling a system call, repeatedly. The program reads one character at a time and compares it to “\n” until there is no characters left in the file. The problem with doing it this way is that read is a system call and system calls need a lot of resources to process to they take a long time to execute so having read execute multiple times is going to make the program very slow.

Q5

Time ./myWc romeo-and-juliet.txt

```
steven.canonalmagro@csx:~/CPSC457/asg/asg1$ time ./myWc romeo-and-juliet.txt
4853 romeo-and-juliet.txt

real    0m0.003s
user    0m0.001s
sys     0m0.002s
```

myWc.cpp is much faster than countLines.cpp but still slower than wc