

CPSC 457 – Assignment 1

Question 1

- a) If the fetch unit takes 10 nsec, the decoding unit takes 0.5 nsec, and the execute unit takes 1 nsec, then each instruction takes $10 + 0.5 + 1 = 11.5$ nsec to execute each instruction, and this CPU can execute 86, 956, 521 instructions on average per second when the three stages are not parallelized.
- b) If the fetch unit takes 10 nsec, the decoding unit takes 0.5 nsec, and the execute unit takes 1 nsec, then each instruction takes 10 nsec to execute, as they can only be executed as fast as the slowest stage in the CPU cycle. The CPU can execute 100, 000, 000 instructions per second on average if the three stages are running in parallel.

Question 2

- a) Benefits of a virtual machine from the perspective of the operating system include that each Virtual Machine running on the same piece of hardware is completely isolated from one another, and this keeps the host system safe from unsafe software or actions performed within the virtual machine.
- b) Benefits of a virtual machine from the perspective of the user includes room for error, as virtual machines can be vehicle for research since they can easily be deleted and created if anything happens to go wrong. Also, with virtual machines one computer can run several operating systems or versions of operating systems without the need to buy the hardware for each. System consolidation and the money saved on system hardware is another benefit of virtual machines for the user because one expensive piece of hardware can emulate several operating systems as if they though have their own processor and hardware.

Question 3

- a) Interrupts: An interrupt is a signal from the interrupt controller that notifies the CPU that an external device or application has completed its operation. Once the CPU has been notified, it can handle the interrupt by executing the appropriate interrupt handler from kernel mode.
- b) Traps: A trap is a special instruction that switches from user mode to kernel mode and invokes a condition predefined by the OS. Traps are triggered by an exceptional condition from user mode and will pause an application to execute the kernel routine. Once it is done, user mode and the application are restored.

c) Differences between interrupts and traps:

- Interrupts are signals generated by external events, such as I/O and the timer, that are delivered to the CPU. Traps are internal events invoked by error conditions or system calls.
- Interrupts are asynchronous with the CPU while traps are synchronous with the activity of the CPU,
- Time of interrupts is unknown while traps occur as the result of a machine instruction and can be predictable.

d) Why must interrupts and traps be handled in kernel mode instead of user mode?

Interrupts and traps need to be handled in kernel mode instead of user mode because user mode does not allow for access to hardware and has a limited instruction set, whereas kernel mode has full access to both. Interrupts and traps both need to access hardware, being the CPU, to send a signal to, and both need to invoke a kernel procedure, being an appropriate interrupt handler for interrupts, and a kernel routine configured by the OS for traps.

Question 4

a) Outputs of the `time` command for `countLines` and `wc -l`.

	<code>countLines</code>	<code>wc -l</code>
Real	0m2.005s	0m0.112s
User	0m0.125s	0m0.000s
Sys	0m1.828s	0m0.047s

b) The C++ program, `countLines`, spent 1.828 seconds in kernel mode, and 0.125 seconds in user mode.

The `wc -l` utility spent 0.047 seconds in kernel mode, and 0 seconds in user mode.

c) The '`wc -l`' utility is much faster than the `countLines` program because it makes fewer system calls than `countLines` does. `countLines` is very slow because it reads the file byte by byte and makes very many system calls.

Question 5

	<code>myWC</code>	<code>countLines</code>	<code>wc -l</code>
Real	0m.078s	0m1.956s	0m0.100s
User	0m0.000s	0m0.031s	0m0.000s
Sys	0m0.031s	0m1.922s	0m0.016s

`myWC` is closer to the performance of the '`wc`' utility now because of the reduced amount of system calls compared to `countLines`.