Assignment 1

Satyaki Ghosh
10077685
T #01

1. On average, this CPU can execute 100,000,000 instructions per second. This is because the Fetch unit takes 10nsec to execute and since that is the bottleneck we can't perform more than 1 instruction every 10nsec (10^{-8} sec). Thus, 1 sec / (10^{-8}sec / instruction) = 100,000,000 instructions every second. We can improve the performance of the CPU by lowering the execute time of the Fetch unit.

2. The benefits of using a VM are:
   - The host system is protected from the VMs
   - VMs are isolated and protected from each other – can run conflicting applications in separate VMs
   - Multiple different OSs or versions can be running on the same computer concurrently
   - Perfect vehicle for OS research and development
   - System consolidation – can potentially save a lot of money by buying one single large server and partitioning it, instead of buying multiple smaller ones

3. Interrupts are external events delivered to the CPU. They can originate from the I/O, timer or user input. They are asynchronous with the current activity of the CPU. The time when a interrupt occurs is not known and unpredictable.
   Traps are internal events i.e. system calls, error conditions, etc. They run synchronously with the current activity of the CPU. They occur during the execution of a machine instruction.
   Traps and Interrupts both put the CPU in kernel mode. They both save the current state of the CPU, invoke a kernel procedure (defined by the OS) and they both resume the original operations when they are done. They are handled in kernel mode instead of user mode because they both invoke kernel procedures which can only be called from the kernel mode.

4. Using 'time ./countLines romeo-and-juliet.txt', these times were found:

   ```
   4853 romeo-and-juliet.txt
   real    0m0.057s
   user    0m0.002s
   sys     0m0.052s
   ```

   Using 'wc -l time ./countLines romeo-and-juliet.txt', these times were found:

   ```
   17 ./countLines
   4853 romeo-and-juliet.txt
   4870 total
   real    0m0.003s
   user    0m0.000s
   sys     0m0.002s
   ```

The C++ program spent 0.052s in kernel mode and 0.002s in user mode. The ‘wc’ spent 0.002s in kernel mode and 0s in user mode. The ‘wc’ program runs faster than the C++ program because ‘wc’ reads each word in the file and finds the ‘/n’ delimiter instead of reading each and every single char, which is what the C++ program does.

5. I improved the performance of countLines by modifying the code to read whole lines instead of reading single chars and the modified code is in file ‘myWc.cpp’.
   Using ‘time ./myWc romeo-and-juliet.txt’, these times were found:

```
4853 romeo-and-juliet.txt
real    0m0.003s
user    0m0.001s
sys     0m0.002s
```

   Using ‘time ./countLines romeo-and-juliet.txt’, these times were found:

```
4853 romeo-and-juliet.txt
real    0m0.057s
user    0m0.002s
sys     0m0.052s
```

   Using ‘wc -l time ./countLines romeo-and-juliet.txt’, these times were found:

```
17 ./countLines
4853 romeo-and-juliet.txt
4870 total
real    0m0.003s
user    0m0.000s
sys     0m0.002s
```