Tze-Yang Chen

Txc172930

CS4348 Project 1 summary

**Part 1 project purpose.**

The purpose of this project is utilizing what we have been discussing in class, and use them to extend our coding experience. This project, java, is using Process programing to achieve the simulation of a CPU. Each different instruction is different into processes. In addition, the two classes, CPU and memory are both used to simulate the CPU. Lastly, This CPU is expected to be a multi-process, IO process between the classes, CPU register processor, stack processing.

**Part 2 Implantation**

This project is implemented into two major class and couple function. The Process uses runtime.exec to generate process for all the instructions. A switch is used for instruction processing.

CPU class

Starter(): starts and resets all the variables (stack pointer, registers, memory size and addresses.)
Main (): sets all the IOs and act as starter point for all the files. This function will also signal the Memory file to launch the file and store it at Memory class.

Fetmem(): read memory, passes read command parameters to Memory to fetch the memory.

Wrimem(): write to a memory, passes write command and parameters to Memory to write to memory.

Pushstack(): push into stack, using preexisting pointer, pushes passed data to stack.

Popstack(): pop out of stack, using preexisting pointer, pop data from stack and returns it.

Iint(): a function to check for interrupt during processing.

outt(): timer handler, pushes state into stacks and stops it.

Processswitch(): process the instructions comes from Memory. Accomplished in a giant switch statement with pre-coded (listed in project description) cases. Each action is coded to be as such.

Memory class

Main() :opens file using args[] from CPU. Opens and saves them in a memory array. Than listens (opens for IO controls from CPU. Parse message and response.

**Part 3 Personal experience**

This project is really time consuming, the hardest part will be IO debugging. The examples are great, but would be better if there are more in-depth if there's any. The switching cases are mostly easy to implement, but sometimes it could be hard to implement.

Output

Sample1.txt
ABCDEFGHIJKLMNOPQRSTUVWXYZ12345678910

Sample2.txt

Sample3.txt

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

A

0

All instruction processed

Sample4.txt

999

998

999

1998

1997

1998

Memory out of Bound, Process Ended.

All instruction processed

Sample5.txt (all random number might be different on different run)

1 64

2 39

3 58

4 14

5 93