Operating Systems

**CS4348**

## Project #1: Exploring Multiple Processes and IPC

**Summary**

**Project Purpose**

The purpose of this project was to explore the concepts of multiprocessing. Also we focused on inter-process communication. Multiprocessing was done by forking of the process into a parent and child process which run independently of each other. The IPC was done in this project with help of the pipe command. The two processes communicated with each other by sending and receiving data through a pipe. Also this project focused on learning the low level concepts important to an operating system. These concepts included processor interaction with memory. Processor instruction behavior, role of registers such as PC, IR, etc., stack processing, procedure calls, system calls, memory protection, I/O and interrupt handling. Apart from this the project also helps us understand the implications of IPC in operating systems such as the UNIX.

**Implementation**

In this project we were to implement a simulation of a simple computer system which consists of a memory and a CPU. This project was done using the C language. The memory and the CPU are simulated by the two processes created in the program using fork command. This CPU communicates with the memory using two pipes (one for data going from CPU to memory and the other vice versa).

The purpose of the CPU is to keep track of the different OS modes (i.e. system or user), execute various operations including arithmetic, logical and I/O commands. The list of the commands are transferred to the CPU by reading a text file and storing the data into the memory array. The CPU then executes the instructions in the user program area. The CPU also handles interrupts by running the interrupt handlers for the different interrupts. This is done by changing the mode from user to system and then running the handler after taking the backup of user SP and PC in the system stack. The SP and PC are then restored after the interrupt handler has run.

The memory process implemented here understands three basic commands. It can read from memory, write to memory or exit the memory process. In the case of reading from the memory the CPU first sends memory the read command, then it sends memory the address to read from and then the memory replies back with the value at that address. Similarly, in case of writing, CPU sends the write command, and then it sends the address and the value to be stored at that address.

**My Experience**

This project was a great learning experience for me. This project not only helped me to have a clear idea as to how IPC works on a computer but also helped me enhance my Unix and C language skills. While doing this project there were a number of challenges I faced. First challenge was how to simulate the working of a simple computer with the help of processes. After studying and discussion with professor, I decided that I need to create two processes for the memory and the CPU and interact between them with help of combination of fork and pipes. My other challenge was to decide how to best use the pipes, I decided here to use two pipes for communication. One for sending data from CPU to memory and other for memory to CPU. Also, here I decided to use three commands for reading, writing and exit. My last challenge was the interrupt handling and the creation of modes. I decided that I would keep a check of time by the number of times an instruction is executed and run the handler after n number of instructions.

This project was an overall an awesome experience. Would like to do more projects like the same in this course.

By-

PULKIT KHEMKA