Quang Tran

qxt110430

SUMMARY

The project requirement is to implement a simple CPU-memory architecture. The purpose is to understand how CPU and memory works together, that is how the CPU fetches and executes instructions and data from the memory. Also, students need to understand interrupt handling and apply two types of interrupts: interrupt by timer and interrupt by system call.

To implement the project, I first started to code in Java. After struggling to perform inter-process communication using Java library, I decided to change to C and immediately saw the differences and how easy it is to program IPC in C using fork command and pipes. The implementation basically works as the following steps:

* The CPU is the parent process; creates memory process as its child. They communicate through two pipes.
* The memory process initializes its memory storage immediately after created. It sends a confirmation signal to the parent.
* Receiving the initialization signal from its child, CPU starts fetching instructions and data. First, it sends a READ\_COMMAND signal saying that “I am about to read something from you”. The second piece of data the CPU sends is the address it wants to read. In case of writing into memory, CPU sends WRITE\_COMMAND signal. If error occurs and the CPU decides to stop, it sends CRASH\_COMMAND to the memory process and wait for the child to stop.
* The memory process after receiving commands from its parent will perform the action needed and sends the value at address to its parent in case of READ\_COMMAND.
* The two processes communicate back and forth to perform the whole program.

Unfortunately, I started to the project late and was strictly restricted by the deadline. Working with Java took me about 3 days off the project without any progress. Linghuang, our TA, helps me a lot to understand the requirement and how each instruction should be performed. Since I started to learn C from the beginning, reading from input files and storing information in integer array took me significant amount of time. After that, implementation was straight-forward even though it is extremely difficult to decode and follow the program because the program is written in very low-level language and it is very long. Overall, I enjoy the project because I actually learnt something. Coding and debugging took me about 10 hours after fully understanding the requirement, so the project is moderate even though I exceeded the deadline for some personal reasons. I hope that we have more projects of this type in the future.