OPERATING SYSTEMS PROJECT PHASE-1 REPORT

Software Engineering Issues:

I've faced the below software engineering issues while developing the project:

- The flow of control between the programs is the toughest thing I've faced.
- Designing of the project initially became difficult because of the unclarity of my understanding. But once I became familiar with it, it became easier to write it.

Approach:

I took 5 classes:

1. SYSTEM
2. MEMORY
3. CPU
4. ERROR_HANDLER
5. LOADER

I used Flow charts to keep track of the flow of the program and then with the help of pseudo code, I developed a skeleton of the program. The actual implementation is as follows:

LOADER:

I implemented file reading and storing in the memory operations using this class. I used BuffredReader class to read the instructions from the input file. After reading and strong them into an array, I stored them into Memory array directly.

MEMORY:

The memory class has the main memory where the values are stored and fetched from. It has three consitions. READ, WRIT and DUMP. READ us used to read input from a memory location, WRIT is used to store values into the memory and DUMP is used to preview the contents of the memory.

CPU:

The CPU class has all the main logic that has to be performed. Here I used Nested If-else statements to check for the opcode and perform the corresponding operation. In this class, I used variety of predefined methods like Interger.parseInt(), Binary.parseBinary(), etc. for converting strings to integers, strings to binary, etc. It will make use of the PC value to fetch the instruction from the memory and perform corresponding operation using Opcode.

ERROR_HANDLER:

All the exceptions from all the classes are handled in this class. The error code is passed as an argument to the method of this class which outputs the corresponding error code and terminates the program. It handles various types of errors like Invalid trace value erroe, infinite loop error, program size too large error, Memory size too large error etc.

SYSTEM:

All the classes are invoked and called from the SYSTEM class. This is like the driver of the entire project.

List of utilities used:

- BufferedReader
- Java.util.scanner
- Java.io

Break down of time spent:

I have spent approximately 78 hours in developing the entire project. Out of that time, most of the time (around 25) was spent in printing the trace file, dump file and checking infinite loop condition. CPU class took 20 hours to write, ERROR_HANDLER took around 10 hours and the rest of the item was taken by SYSTEM and MEMORY classes.

Bulk complexities:

The CPU class has around 700 lines of code.

Memory class has around 30 lines of code

ERROR_HANDLER has around 30 lines of code

LOADER class has around 70 linesof code.

SYSTEM class has around 30 lines of code.

Choice of programming language:

I have choosed java programming language because we can trace the program easily and its easy to manage. It has many advantages because of its Object Oriented Principles. I'm also strong in java programming which made me choose this programming language to develop the project.