Yasin ATLI 2020400246

REPORT OF THE PROJECT

In the file, I defined a Process struct in Process.h file. A process struct is consist of a string of id(ID of the process), int the_first_arrival_time(arrival time for queue which changes burst to burst), int burst_time(total burst time), int finish_time(process termination time), int priority(priority of the process defined in definition file),string type(type of the process), int array of instruction_times(array of time of instructions of the process's), int f_a_t(first arrival time of the process), int index_of(index of where I left in executing instructions of the process), int type_change_count(count for type change of the process), int total_instr(total instr count for creating instructions of the process) and int rr_time(arriving time of the round robin queue).

Then, I open the instructions.txt file and read the instructions and their cost and store their values index by index at the instruction_time_costs array. After getting the instructions cost's, I create all the processes' instructions cost arrays by using readInstructionTimes which takes processes name and its instr array and fills them. After that, the program reads the definition.txt file. It counts the number of processes and creates a Process array. By reading line by line the definition.txt file, the program creates the defined processes and adds them into the processes array. After all the file is read, we sort the array using qsort by comparing them using compareProcesses method. The method compares processes based on arrival time , type , priority, round robin arrival time and alphabetically. Later, we start execution in a while loop.

Every iteration of the while loop we choose the first element in the queue. If the element's type is platinum we directly execute all the instructions of the process and to know it finished we set it's the_first_arrival_time to INT_MAX for indicating in comparison that it is finished and setting the finish_time as real_time of the finish. In silver and gold processes we execute instr by instr because they are preemptive. If any higher priority comes in the queue after an execution of an instruction, it will be preempted to the higher order process. By incrementing type change count, we change the type of the process. If

the first instruction of the burst is executed and there is not a preemption and there is time left from the time quantum we proceed the next instruction in a while loop until there is a process to preempt or there is no time quantum left. Then, we set the last executed process and proceed in the queue. If the executing process and the last executed process are not the same context switch occurs which in case is 10 ms. The main difference between a silver process and gold process is time quantum. Time quantum for the silver process is 80 ms and for the gold process is 120ms and there is type change count. Type change for silver process is 3 bursts and for gold process is 5 bursts.

The execution of the queue goes like explained above. If all the processes are finished, the_first_arrival_time of the head of the queue must be INT_MAX. We check that condition at every iteration of the queue while loop. After the loop is finished, we compute all the waiting and turnaround time for the processes. Then I compute the average outputs.