The University of the West Indies
Department of Computing
COMP3101 Operating Systems

Group Project

(to be attempted by a group of up to 4 students)

Project Description

You are required to develop a simulation of processes executing on a uni-processing computer system that uses 32-bit instructions. The instruction set comprises three simple instructions:

> COMPUTE - This instruction requires 5 ticks of the CPU.
>
> INPUT - This instruction requires 1 tick of the CPU then causes an event wait on event number 1.
>
> OUTPUT - This instruction requires 1 tick the causes an event wait on event number 2.

The simulator shall allow a user to load and run simple programs. A program is a text file with one instruction per line. Program execution is merely simulated since no operands are used with the instructions. As such, the simulator simply reads each instruction and pretends to execute it for the stipulated number of ticks.

The Toy System allocates memory in variable sized portions. Memory allocation strategies are described in the Settings section. The processor is scheduled on a round-robin basis wth the quantum being one of the parameters that a user can set.

The simulator can be used to approximate system performance under various scenarios. In its present form, the simulator covers rudimentary aspects of process management and control, and memory management. Further work can be done to incorporate other aspects of operating systems.

You can use any programming language you wish for this assignment. However, you should be prepared to illustrate good programming practices in an examination of your code.

In addition to the specified functionality a number of points can be earned for innovation. For example, you might implement advanced memory management such as virtual memory, or additional interrupts for devices.

Settings

On starting up, the simulator shall prompt the user for the following settings. These settings may also be loaded from a configuration file.

Setting	Description
memory s, u	Specifies the amount of RAM to be used for the simulation in words. Parameters are: s the amount of RAM reserved for the OS u the amount of memory available for user processes For example, the setting *memory 64,128* sets up the simulator to run with 64 words of system memory and 128 words available for user programs.
degreem n	The degree of multiprogramming. For example, the command *degreem 5* specifies that a maximum of 5 processes may be active at any time.
context x	The number of ticks it takes the processor to switch from one process to the next.
allocation mode	Specifies the memory allocation strategy to be used. Parameter values are: 1. First Fit. Place the job in the first available memory locations. 2. Best Fit. Place the job in the smallest hole that can accommodate it. 3. Next Fit. Place the job in the first available hole after the point where the last allocation was made.

Simulator Commands

Once the simulator has been set up the user shall be prompted to enter commands. Each command shall be processed as follows:

Command	Simulator Response
stop	Stop the simulation and display summary statistics.
show jobs	Display all active processes with the status of each process.
show queues	Display the contents of each queue (Ready Queue, Blocked Queue). The display shall show for each process: • Size of the image (number of words) • Status (ready, running, blocked) • Program counter • CPU ticks consumed thus far • Event being awaited (for a blocked process)
show memory	Display a map of memory showing starting location of each job.
tick n	Run the simulation for a specified number of ticks. During each tick the running process receives one cycle of the processor and an appropriate action is taken. Actions can include:

Command	Simulator Response
	• Timeout – when a process has received its quantum of ticks it is moved to the Ready Queue and the first process in the Ready Queue is given the processor; • System call – if the running process makes a system call it is placed in the Blocked Queue for a period and the first process in the Ready Queue is given time on the processor. • I/O Event – if an event that a process is waiting on occurs then the process is moved to the Ready Queue.
admit *progname*	Admit the specified program as a job. If the degree of multiprogramming has been achieved then the simulator shall issue a message to say so and will not allow the job to be admitted. Otherwise, the job is admitted and placed in the Ready Queue.
interrupt n	An interrupt event has occurred. Any process that is waiting on the event will be moved from the Blocked Queue into the Ready Queue.

For the sake of simplicity, it is assumed that the tick command is the only command that causes the processor to consume a tick, or number of ticks. You are free to enforce constraints in this regard. For example, it might be useful to allow only one system changing action (admit or interrupt) to occur between ticks.

Summary Statistics

At the end of a simulation, the simulator shall display summary statistics. All times are recorded as numbers of ticks. Space is reported as number of words. The statistics shall include:

- Configuration setting values (format as you wish)
- Number of jobs completed
- Response time for each completed job measured from the time each job is admitted to the time it first gets to use the processor. For example, if process A is admitted at tick 3 and first gets the processor at tick 6 then the response time is 6-3=3 ticks.
- Turnaround time for each job measured as the time a process leaves the system minus the time it was admitted into the system. For example, if process A is admitted at tick 1 and completes at tick 10 then the turnaround time is 10-1=9 ticks.
- Average response time for all completed processes.
- Average turnaround time for all completed processes.
- Overall CPU utilization rate expressed as the total time the processor spends executing user programs divided by the total simulation time. For example, if the

total simulation takes 50 ticks of which 10 ticks were consumed by the processor in switching processes then the CPU utilization rate is (50-10) / 50 = 80%.

- Average memory utilization rate measured as the average percentage of memory occupied by user programs in each tick. For example, if the simulation is run for 4 ticks and the amount of memory occupied at the start of each tick is 10%, 20%, 20%, and 30% then the average memory utilization is (10+20+20+30/4) or 20%.

Project Submission and Demonstration

All groups must be registered by September 15, 2024.

Project demonstrations will be held in accordance with the schedule we agree on during Week 13 of the semester. Each group of students will be required to make a 10 minute presentation and to field up to 5 minutes of questions from the assessors.

Grading Guidelines

The project will be marked out of 100 points as follows:

- 15 points for demonstration of the simulator's response to changed settings.
- 40 points for functionality based on the percentage of system functions that are demonstrated to work correctly.
- 5 points for usability of the software including helpful prompts and messages;
- 5 points for the overall quality of the presentation;
- 5 points for quality of the software (modular design, clean code, etc.);
- 10 points for innovation. Please point this out for assessment.

A total of 80 points can be scored as described above. The remaining 20 points are to be earned through Peer Assessment. Each student will appraise the contribution of the other students in the group. Each student will receive a final score equal to:

$$g + [p/12*20] * g/80$$

where:

g is the group score

p is the student's peer assessed score

All the best!