

## Phase 2 Documentation

Github repository link: <https://github.com/mclaurind/CMSC312-OS-Simulator.git>

Commit to grade: src - before multithreading

Phase 2 12/05

### OS Simulator Methods & Classes

#### OS class

##### **public static void main (String [] args)**

User is asked to choose a program and to input how many processes to spawn for the program. Then the scheduler will run, using the round robin scheduling algorithm. After the scheduler executes, a list of the child processes made will be displayed, all in a terminated state (cascading termination).

With multithreading:

4 Scheduler threads would be initialized and have a thread number associated with it to set threads and the processes assigned to them.

##### **public static pcb generateProcess (String programName)**

Parses program files and returns a process

#### Scheduler class

##### **public synchronized void roundRobinScheduler (ArrayDeque<pcb> new Queue, int processTotal, int remMemory)**

Implements a round robin scheduling algorithm. A process will go through the new, wait, and ready queues, and ultimately run until its cycles are completed in relation to the time quantum. Also checks processes against memory limits and throws random interrupts while a process is simulating.

##### **public void run()**

Thread number for each thread will be assigned to a process to be simulated with a round robin scheduler.

public static Log(String activity)

Prints live

#### CSHandler Class

Implements semaphores to handles critical sections in processes.

##### **public void waitSem(Semaphore semaphore, pcb process, int permit)**

Makes processes wait for permit

##### **public void signalSem(Semaphore semaphore, pcb process, int permit)**

Signifies processes in wait queue that a permit is available

Semaphore class (nested in CSHandler)

##### **public void wakeup(pcb process)**

Removes a process from semaphore's wait queue

##### **public void block(pcb process)**

Places processes wanting to execute their critical section's code in the semaphore's wait queue

## Phase 2 Documentation

### **IODevice**

Has three IO devices, keyboard, mouse, and monitor which are chosen at random to interrupt simulating processes for a randomized amount of cycles.

### **public boolean shouldInterrupt()**

A simulating process will have a 50% of being interrupted during any of its instructions.

### **pcb Class**

Process control block for a process. Updated to include memory of a process, as well as PIDs for child processes.

### **public string toString()**

Prints the pcb for all parent and child processes.

### **cpu Class**

2 cpu objects will be made and be assigned two schedulers (round robin and priority) in Phase 3.

### **ProcessState enum**

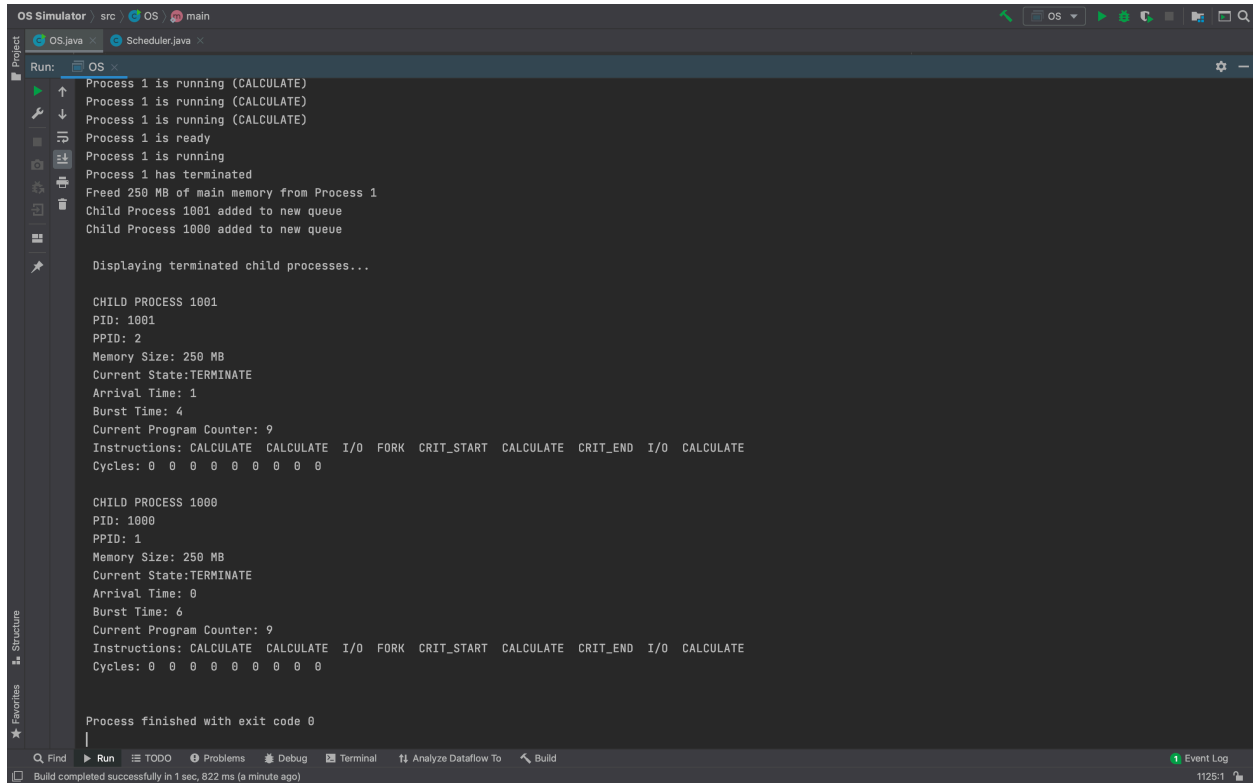
Contains the states of a process.

## Phase 2 Documentation

### Example outputs and multithreading

#### Before multithreading

Terminated children are displayed, as well as when processes are in their critical section and when they exit so another process can execute its critical section code. Additionally, it's shown when a process exceeds the memory amount and when it does finally become a simulating process.



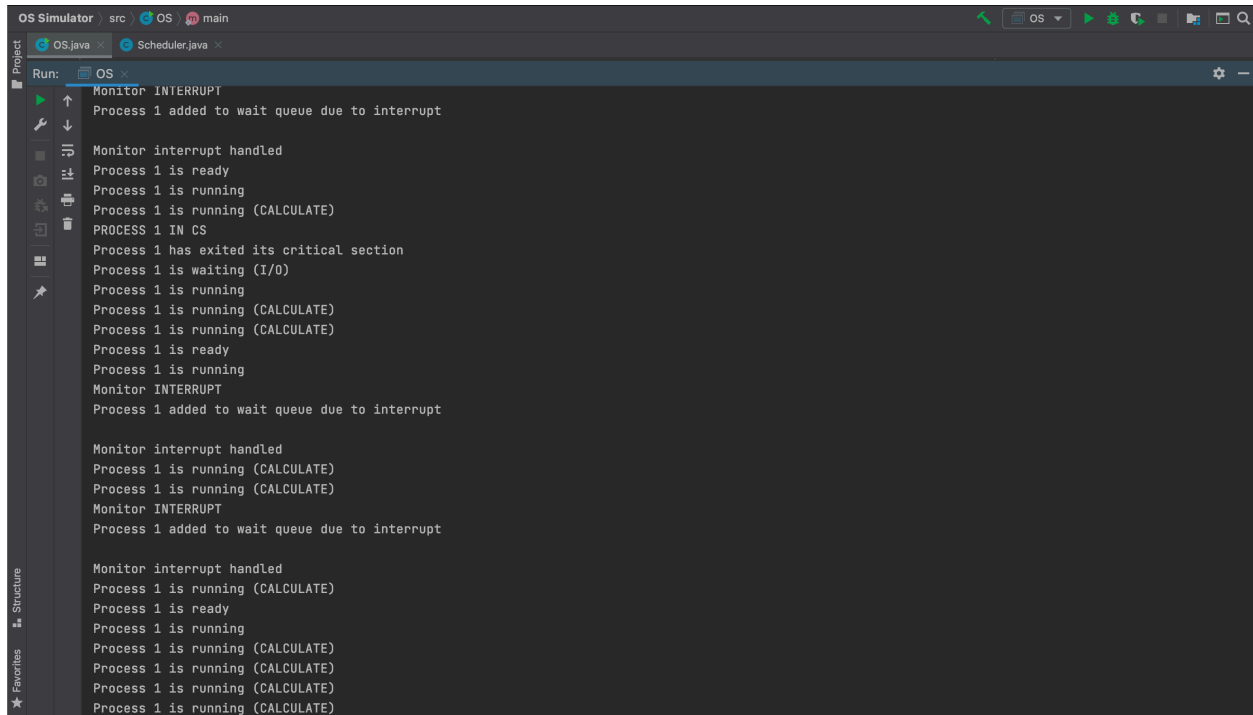
```
OS Simulator  src  OS  main
OS.java  Scheduler.java
Run: OS
Process 1 is running (CALCULATE)
Process 1 is running (CALCULATE)
Process 1 is running (CALCULATE)
Process 1 is ready
Process 1 is running
Process 1 has terminated
Freed 250 MB of main memory from Process 1
Child Process 1001 added to new queue
Child Process 1000 added to new queue

Displaying terminated child processes...

CHILD PROCESS 1001
PID: 1001
PPID: 2
Memory Size: 250 MB
Current State:TERMINATE
Arrival Time: 1
Burst Time: 4
Current Program Counter: 9
Instructions: CALCULATE  CALCULATE  I/O  FORK  CRIT_START  CALCULATE  CRIT_END  I/O  CALCULATE
Cycles: 0 0 0 0 0 0 0 0 0

CHILD PROCESS 1000
PID: 1000
PPID: 1
Memory Size: 250 MB
Current State:TERMINATE
Arrival Time: 0
Burst Time: 6
Current Program Counter: 9
Instructions: CALCULATE  CALCULATE  I/O  FORK  CRIT_START  CALCULATE  CRIT_END  I/O  CALCULATE
Cycles: 0 0 0 0 0 0 0 0 0

Process finished with exit code 0
|
Build completed successfully in 1 sec, 822 ms (a minute ago)
```



```
OS Simulator  src  OS  main
OS.java  Scheduler.java
Run: OS
Monitor INTERRUPT
Process 1 added to wait queue due to interrupt

Monitor interrupt handled
Process 1 is ready
Process 1 is running
Process 1 is running (CALCULATE)
PROCESS 1 IN CS
Process 1 has exited its critical section
Process 1 is waiting (I/O)
Process 1 is running
Process 1 is running (CALCULATE)
Process 1 is running (CALCULATE)
Process 1 is ready
Process 1 is running
Monitor INTERRUPT
Process 1 added to wait queue due to interrupt

Monitor interrupt handled
Process 1 is running (CALCULATE)
Process 1 is running (CALCULATE)
Monitor INTERRUPT
Process 1 added to wait queue due to interrupt

Monitor interrupt handled
Process 1 is running (CALCULATE)
Process 1 is ready
Process 1 is running
Process 1 is running (CALCULATE)
Process 1 is running (CALCULATE)
Process 1 is running (CALCULATE)
Process 1 is running (CALCULATE)
```

## Phase 2 Documentation

```
Run: OS
Process 7 exceeds OS memory limit and will stay in new queue.
Keyboard INTERRUPT
Process 4 added to wait queue due to interrupt

Keyboard interrupt handled
Process 7 exceeds OS memory limit and will stay in new queue.
Process 4 is running (CALCULATE)
Process 4 is ready
Process 5 is running
Process 7 exceeds OS memory limit and will stay in new queue.
Process 5 is running (CALCULATE)
Process 7 exceeds OS memory limit and will stay in new queue.
Process 5 is running (CALCULATE)
Process 7 exceeds OS memory limit and will stay in new queue.
Process 5 has terminated
Freed 250 MB of main memory from Process 5
Process 6 is running
Process 7 successfully loaded into main memory.
Process 6 is running (CALCULATE)
Process 6 is ready
Process 4 is running
Keyboard INTERRUPT
Process 4 added to wait queue due to interrupt

Keyboard interrupt handled
Keyboard INTERRUPT
Process 4 added to wait queue due to interrupt

Keyboard interrupt handled
Keyboard INTERRUPT
Process 4 added to wait queue due to interrupt
```

### After multi-threading

The OS freezes and the different threads seems to be split among only one process. So, I will have to go back and look at my scheduler for phase 3.

```
OS Simulator  src Scheduler OS.java OS Scheduler.java run
Project Project OS.java Scheduler.java
Run: OS OS
PID: 2
Memory Size: 250 MB
Current State: NEW
Arrival Time: 1
Burst Time: 2
Current Program Counter: 0
Instructions: CALCULATE CALCULATE I/O FORK CRIT_START CALCULATE CRIT_END I/O CALCULATE
Cycles: 99 30 16 0 0 18 0 9 4

ROUND ROBIN SCHEDULER ACTIVITY LOG
*****
Process 1 successfully loaded into main memory.
Process 2 successfully loaded into main memory.
Process 1 is running
Mouse INTERRUPT
Process 1 added to wait queue due to interrupt
THREAD THREAD THREAD THREAD THREAD0
THREAD THREAD THREAD THREAD THREAD1

Mouse interrupt handled
Mouse INTERRUPT
THREAD THREAD THREAD THREAD THREAD2
Process 1 added to wait queue due to interrupt

Mouse interrupt handled
THREAD THREAD THREAD THREAD THREAD3
Child process 1000 created for Process 1 (FORK)
Process 1 has entered its critical section
Process 1 sent to semaphore wait queue
Process 1 has exited its critical section

IntelliJ IDEA 2020.3.4 available
Update...

Build completed successfully in 2 sec, 154 ms (moments ago) 63-2 LF UTF-8 4 spaces
```