Date of submission (Individual): Dec. 1
Colin Calhoun (Student 1) - 101307947
James Noel (Student 2) - 101306496

1. EXTERNAL PRIORITIES

This algorithm schedules processes based on their priority—which in this case is their PID number—where the lower PID has the higher priority. When a process' I/O begins, it enters the wait queue and completes its I/O before entering the ready queue to proceed with execution. The simulator logs the states of each process as well as the memory status when a process starts running.

Trace 1:
1, 2, 0, 5, 0, 0

The simulator schedules one process with a PID of 2, memory size of 2 Mb, arrival time of 0, execution time of 5, and there is no I/O. The process goes from the NEW state to READY at time = 0, READY to RUNNING at time = 0, and RUNNING to TERMINATED at time = 5.

```
PID 1 starts running at 0
----------------------------------------
Partition: 1, Size: 40 , Availability: Free
Partition: 2, Size: 25 , Availability: Free
Partition: 3, Size: 15 , Availability: Free
Partition: 4, Size: 10 , Availability: Free
Partition: 5, Size: 8 , Availability: Free
Partition: 6, Size: 2 , Availability: used by pid 1
----------------------------------------------------
Total memory used:   2
Total free memory:   98
Total usable memory:   2
----------------------------------------------------
```

Using the above memory status as an example, the simulator logs which PID starts running and at what time. Then it records the occupation of each memory partition, with its size and which process is occupying it—in this case PID 1 occupies partition 6 which is a size of 2 Mb. It then tracks the total memory used by all processes at the moment in time, total free memory, and total usable memory (not internal fragments.

Trace 2:
4, 20, 5, 20, 0, 0

The simulator schedules one process with a PID of 4, memory size of 20 Mb, arrival time of 5, execution time of 20, and there is no I/O. The process goes from the NEW state to READY at time = 5, READY to RUNNING at time = 5, and RUNNING to TERMINATED at time = 25.

Trace 3:
10, 20, 8, 8, 0, 0

The simulator schedules one process with a PID of 10, memory size of 20 Mb, arrival time of 8, execution time of 8, and there is no I/O. The process goes from the NEW state to READY at time = 8, READY to RUNNING at time = 8, and RUNNING to TERMINATED at time = 16.

Trace 4:
0, 5, 2, 100, 0, 0

The simulator schedules one process with a PID of 0, memory size of 5 Mb, arrival time of 2, execution time of 100, and there is no I/O. The process goes from the NEW state to READY at time = 2, READY to RUNNING at time = 2, and RUNNING to TERMINATED at time = 102.

Traces 5-6:
0, 1, 0, 10, 5, 1
2, 1, 0, 8, 3, 2

The simulator schedules one process with a PID of 0, memory size of 1 Mb, arrival time of 0, execution time of 10, an I/O frequency of 5ms, and an I/O time of 1ms. The process goes from the NEW state to READY at time = 0, READY to RUNNING at time = 0, RUNNING to WAITING at time = 5ms to handle I/O, WAITING to READY at time = 6 when I/O completes, READY to RUNNING at time = 6, and RUNNING to TERMINATED at time = 10.

Traces 7-8:
9, 1, 0, 15, 7, 4
4, 3, 0, 10, 4, 2

The simulator schedules one process with two I/O operations, the logic is the same as the trace above but with different values.

Traces 9-17: Traces with three processes, involving I/O

Example for one of them:

```
1, 2, 0, 5, 0, 0
2, 10, 0, 4, 2, 1
```

```
+------------------------------------------------+
|Time of Transition |PID | Old State | New State |
+------------------------------------------------+
|                 0 |  1 |       NEW |     READY |
|                 0 |  2 |       NEW |     READY |
|                 0 |  2 |     READY |   RUNNING |
|                 2 |  2 |   RUNNING |   WAITING |
|                 3 |  2 |   WAITING |     READY |
|                 3 |  2 |     READY |   RUNNING |
|                 4 |  2 |   RUNNING |TERMINATED |
|                 4 |  1 |     READY |   RUNNING |
|                 9 |  1 |   RUNNING |TERMINATED |
+------------------------------------------------+
```

Same logic as the traces above but with two processes

Traces 18-20: Traces with three processes, involving I/O

Example for one of them:

```
1, 2, 0, 5, 0, 0
2, 10, 0, 4, 2, 1
```

```
+------------------------------------------------+
|Time of Transition |PID | Old State | New State |
+------------------------------------------------+
|                 0 |  1 |       NEW |     READY |
|                 0 |  2 |       NEW |     READY |
|                 0 |  3 |       NEW |     READY |
|                 0 |  3 |     READY |   RUNNING |
|                 2 |  3 |   RUNNING |   WAITING |
|                 3 |  3 |   WAITING |     READY |
|                 3 |  3 |     READY |   RUNNING |
|                 3 |  3 |   RUNNING |TERMINATED |
|                 3 |  2 |     READY |   RUNNING |
|                 5 |  2 |   RUNNING |   WAITING |
|                 6 |  2 |   WAITING |     READY |
|                 6 |  2 |     READY |   RUNNING |
|                  7 |  2 |   RUNNING |TERMINATED |
|                  7 |  1 |     READY |   RUNNING |
|                 12 |  1 |   RUNNING |TERMINATED |
+------------------------------------------------+
```

Same logic as the traces above but with three processes.

2. ROUND ROBIN WITH q = 100 ms

This algorithm schedules processes by allowing them to run for a certain amount of time, in this case 100ms. If a process' quantum runs out, it enters the ready queue and the next process in the ready queue gets its turn. When a process' I/O begins, it enters the wait queue and completes its I/O before entering the ready queue to proceed with execution. The simulator logs the states of each process as well as the memory status when a process starts running.

Traces 1-4: One process with no I/O
10, 20, 0, 150, 0, 0

```
+------------------------------------------------+
|Time of Transition |PID | Old State | New State |
+------------------------------------------------+
|                  0 | 10 |       NEW |     READY |
|                  0 | 10 |     READY |   RUNNING |
|                 99 | 10 |   RUNNING |     READY |
|                 99 | 10 |     READY |   RUNNING |
|                150 | 10 |   RUNNING |TERMINATED |
+------------------------------------------------+
```

The simulator schedules one process that will get interrupted by the quantum at least once, at time == 99 the process enters the ready queue, and then begins running until termination.

Traces 5-8: One process with I/O:

Example for one of them:
10, 20, 0, 300, 150, 50

```
+------------------------------------------------+
|Time of Transition |PID | Old State | New State |
+------------------------------------------------+
|                 0 | 10 |       NEW |     READY |
|                 0 | 10 |     READY |   RUNNING |
|                99 | 10 |   RUNNING |     READY |
|                99 | 10 |     READY |   RUNNING |
|               150 | 10 |   RUNNING |   WAITING |
|               200 | 10 |   WAITING |     READY |
|               200 | 10 |     READY |   RUNNING |
|               298 | 10 |   RUNNING |     READY |
|               298 | 10 |     READY |   RUNNING |
|               349 | 10 |   RUNNING |TERMINATED |
+------------------------------------------------+
```

The simulator schedules one process that will get interrupted by the quantum at least once, and has an I/O every 150 seconds, so it should occur once—as shown at time = 150 ms.

Traces 9-20: Two processes with no I/O

Example for one of them:
10, 20, 0, 200, 0, 0
11, 20, 50, 200, 0, 0

```
+------------------------------------------------+
|Time of Transition |PID | Old State | New State |
+------------------------------------------------+
|                 0 | 10 |       NEW |     READY |
|                 0 | 10 |     READY |   RUNNING |
|                50 | 11 |       NEW |     READY |
|                99 | 10 |   RUNNING |     READY |
|                99 | 11 |     READY |   RUNNING |
|               199 | 11 |   RUNNING |     READY |
|               199 | 10 |     READY |   RUNNING |
|               300 | 10 |   RUNNING |TERMINATED |
|               300 | 11 |     READY |   RUNNING |
|               400 | 11 |   RUNNING |TERMINATED |
+------------------------------------------------+
```

The simulator schedules two processes that will get interrupted by the quantum at least once, they both enter the ready queue after being interrupted 100ms after they first start running.