SYSC 4001 A2 Part 3 Report

Student 1: Pietro Adamvoski (101238885)
Student 2: Jason Huang (101265573)
Github repo part 2: https://github.com/pietrovos/SYSC4001_A2_P2
Github repo Part 3:https://github.com/Dashx2/SYSC_4001_A2_P3

The purpose of this simulation was to demonstrate a simplified operating system that supports interrupts, process creation, process execution, scheduling, and memory management. Using the provided simulator, three test cases were executed to observe how the system responds to SYSCALL, FORK, EXEC, ENDIO, and CPU events. For each test, the simulator produced two output files:

- execution.txt, which logs every interrupt, kernel action, ISR, vector lookup, memory access, and IRET event.
- system_status.txt, which displays snapshots of the system state after IRET of specific interrupts.

These logs were analyzed to evaluate the correctness of interrupt handling, PCB updates, memory allocation, scheduling, and state transitions. Overall, all three tests behaved as expected and accurately reflected the workflow of a real interrupt-driven OS.

Test Scenario 1:

SYSC_4001_A2_P3 > output > test_scenario1 > execution.txt > data

```
0, 1, switch to kernel mode
1, 10, context saved
11, 1, find vector 2 in memory position 0x0004
12, 1, load address 0X0695 into the PC
13, 10, cloning the PCB
23, 0, scheduler called
23, 1, IRET
24, 1, switch to kernel mode
25, 10, context saved
35, 1, find vector 3 in memory position 0x0006
36, 1, load address 0X042B into the PC
37, 50, program is 10Mb large
87, 150, loading program into memory
237, 3, marking partition as occupied
240, 6, updating PCB
246, 0, scheduler called
246, 1, IRET
247, 100, CPU Burst
347, 1, switch to kernel mode
348, 10, context saved
358, 1, find vector 3 in memory position 0x0006
359, 1, load address 0X042B into the PC
360, 25, program is 15Mb large
385, 225, loading program into memory
610, 3, marking partition as occupied
613, 6, updating PCB
619, 0, scheduler called
619, 1, IRET
620, 1, switch to kernel mode
621, 10, context saved
631, 1, find vector 4 in memory position 0x0008
632, 1, load address 0X0292 into the PC
633, 250, SYSCALL ISR (ADD STEPS HERE)
883, 1, IRET
```

SYSC_4001_A2_P3 > output > test_scenario1 > system_status.txt

```
time: 24; current trace: FORK, 10
+------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------+
|   1 |        init |               5 |    1 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------+
time: 247; current trace: EXEC program1, 50
+------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------+
|   1 |    program1 |               4 |   10 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------+
time: 620; current trace: EXEC program2, 25
+------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------+
|   0 |    program2 |               3 |   15 | running |
+------------------------------------------------+
```

The corresponding system_status.txt snapshot at time 24 confirms:

- PID 1 is RUNNING the init program
- PID 0 (the parent) is WAITING

Each process is placed in a 1MB partition
Next, PID 1 calls EXEC program1. The log shows the simulator freeing old memory, loading program1, allocating partition 4, updating the PCB, and returning to user mode. The status snapshot at time 247 shows:

- PID 1 running program1 (10MB)
- PID 0 waiting as init

Finally, after the child finishes, the parent executes EXEC program2. Memory is reloaded, the PCB is updated, and the parent ends up as the only active process. The final snapshot shows:

- PID 0 running program2 (15MB)
- No waiters remaining

Test Scenario 2:

```
0, 1, switch to kernel mode
1, 10, context saved
11, 1, find vector 2 in memory position 0x0004
12, 1, load address 0X0695 into the PC
13, 17, cloning the PCB
30, 0, scheduler called
30, 1, IRET
31, 1, switch to kernel mode
32, 10, context saved
42, 1, find vector 3 in memory position 0x0006
43, 1, load address 0X042B into the PC
44, 16, program is 10Mb large
60, 150, loading program into memory
210, 3, marking partition as occupied
213, 6, updating PCB
219, 0, scheduler called
219, 1, IRET
220, 1, switch to kernel mode
221, 10, context saved
231, 1, find vector 2 in memory position 0x0004
232, 1, load address 0X0695 into the PC
233, 15, cloning the PCB
248, 0, scheduler called
248, 1, IRET
249, 1, switch to kernel mode
250, 10, context saved
260, 1, find vector 3 in memory position 0x0006
261, 1, load address 0X042B into the PC
262, 33, program is 15Mb large
295, 225, loading program into memory
520, 3, marking partition as occupied
523, 6, updating PCB
529, 0, scheduler called
529, 1, IRET
530, 53, CPU Burst
583, 1, switch to kernel mode
584, 10, context saved
594, 1, find vector 3 in memory position 0x0006
595, 1, load address 0X042B into the PC
596, 33, program is 15Mb large
629, 225, loading program into memory
854, 3, marking partition as occupied
857, 6, updating PCB
863, 0, scheduler called
863, 1, IRET
864, 53, CPU Burst
917, 205, CPU Burst
```

SYSC_4001_A2_P3 > output > test_scenario2 > system_status.txt

```
time: 31; current trace: FORK, 17
+------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------+
|   1 |        init |               5 |    1 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------+
time: 220; current trace: EXEC program1, 16
+------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------+
|   1 |    program1 |               4 |   10 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------+
time: 249; current trace: FORK, 15
+------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------+
|   2 |    program1 |               3 |   10 | running |
|   0 |        init |               6 |    1 | waiting |
|   1 |    program1 |               4 |   10 | waiting |
+------------------------------------------------+
time: 530; current trace: EXEC program2, 33
+------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------+
|   2 |    program2 |               3 |   15 | running |
|   0 |        init |               6 |    1 | waiting |
|   1 |    program1 |               4 |   10 | waiting |
+------------------------------------------------+
time: 864; current trace: EXEC program2, 33
+------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------+
|   1 |    program2 |               3 |   15 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------+
```

The system_status snapshots match this behavior:

After the first FORK: PID 1 running, PID 0 waiting

- After EXEC program1: PID 1 running program1 in partition 4
- After second FORK: PID 2 running, PIDs 1 and 0 waiting
- After EXEC program2 by PID 2: PID 2 running program2 in partition 3

Later, PID 1 also executes program2. The simulator allocates a different partition for the second 15MB program image (partition 2), showing correct management of multiple large allocations. The final snapshot shows:

- PID 1 running program2
- PID 0 waiting
- PID 2 terminated after finishing execution

Test Scenario 5:

```
0, 1, switch to kernel mode
1, 10, context saved
11, 1, find vector 5 in memory position 0x000A
12, 1, load address 0X048B into the PC
13, 211, SYSCALL ISR (ADD STEPS HERE)
224, 1, IRET
225, 1, switch to kernel mode
226, 10, context saved
236, 1, find vector 2 in memory position 0x0004
237, 1, load address 0X0695 into the PC
238, 10, cloning the PCB
248, 0, scheduler called
248, 1, IRET
249, 1, switch to kernel mode
250, 10, context saved
260, 1, find vector 3 in memory position 0x0006
261, 1, load address 0X042B into the PC
262, 30, program is 12Mb large
292, 180, loading program into memory
472, 3, marking partition as occupied
475, 6, updating PCB
481, 0, scheduler called
481, 1, IRET
482, 120, CPU Burst
602, 1, switch to kernel mode
603, 10, context saved
613, 1, find vector 2 in memory position 0x0004
614, 1, load address 0X0695 into the PC
615, 8, cloning the PCB
623, 0, scheduler called
623, 1, IRET
624, 1, switch to kernel mode
625, 10, context saved
635, 1, find vector 3 in memory position 0x0006
636, 1, load address 0X042B into the PC
637, 20, program is 12Mb large
657, 180, loading program into memory
837, 3, marking partition as occupied
840, 6, updating PCB
846, 0, scheduler called
846, 1, IRET
847, 120, CPU Burst
967, 1, switch to kernel mode
968, 10, context saved
978, 1, find vector 3 in memory position 0x0006
979, 1, load address 0X042B into the PC
980, 25, program is 12Mb large
1005, 180, loading program into memory
1185, 3, marking partition as occupied
1188, 6, updating PCB
1194, 0, scheduler called
1194, 1, IRET
1195, 120, CPU Burst
```

SYSC_4001_A2_P3 > output > test_scenario5 > system_status.txt

```
time: 249; current trace: FORK, 10
+------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------+
|   1 |        init |               5 |    1 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------+
time: 482; current trace: EXEC program1, 30
+------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------+
|   1 |    program1 |               3 |   12 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------+
time: 624; current trace: FORK, 8
+------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------+
|   1 |        init |               5 |    1 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------+
time: 847; current trace: EXEC program1, 20
+------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------+
|   1 |    program1 |               3 |   12 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------+
time: 1195; current trace: EXEC program1, 25
+------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------+
|   0 |    program1 |               3 |   12 | running |
+------------------------------------------------+

```

The execution logs show three EXEC calls for program1:

- PID 1 loads program1 and receives partition 3
- PID 1 later executes program1 again and receives partition 2
- The parent finally executes program1 and receives partition 1

Even though the same program was used each time, the simulator correctly allocated a different partition for every EXEC and updated the PCB each time. This proves that the memory manager is not simply reusing old memory or assuming only one instance of a program can be loaded.

System status snapshots confirm:

- After FORK, PID 1 runs and PID 0 waits
- After the first EXEC of program1, PID 1 runs program1 while PID 0 waits
- After the second FORK, the new child runs while PID 0 waits
- After the next EXEC of program1, the PCB shows PID 1 running from another partition
- Final EXEC shows PID 0 running program1 with no other active PCBs