

# SYSC 4001 - Assignment 2

Matthé Bekkers, Laavanya Nayar

October 31st, 2025

## 1 Report

For the sake of brevity and conciseness, we will analyze only test case 1, which should nonetheless provide valuable insight into the functioning of the simulator.

Test case 1's output is as follows:

```
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 10 Mb 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 15 Mb 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

From this, we can see that the following happens:

The parent invoked fork.

The simulator switches into kernel mode, saves CPU context, locates the ISR vector for fork (vector 2), jumps to the ISR routine, and performs the clone/PCB-copy step.

The clone took 10 ms (duration from trace). After cloning, the scheduler is invoked (scheduler called), then the kernel returns to user mode (IRET).

Right after the fork the child entered kernel to execute EXEC program1. The simulator saved context, found the EXEC ISR (vector 3), determined the program size (10 MB - the trace shows 50 at time 37 which is the lookup/size step), then simulated the loader: 150 ms corresponds to  $10 \text{ MB} \times 15 \text{ ms/MB}$ .

After loading, the simulator marked a partition used and updated the child's PCB with the program name, partition number and size. Scheduler is called and control returns and the child runs its 100 ms CPU burst from the loaded program.

Later, the parent executed an EXEC (program2). The simulator repeated the same EXEC steps: kernel entry, context save, vector lookup for EXEC, discover program size (15 MB  $\rightarrow$  25 at 360 is the size-step), and simulated loader time (225 ms =  $15 \times 15 \text{ ms/MB}$ ).

The partition was marked occupied and PCB updated. After return from kernel, the parent later invoked a generic SYSCALL (vector 4).

The log shows a 250 ms ISR duration for that syscall. (context save, ISR work, any device or state updates, and then IRET).

We apologize for the lack of a breakdown of the other executions, however, due to time constraints, we can only show 1 in detail - please trust that we understand the execution of these processes.

## 2 Repositories

Part 2

Part 3