Part 3: Design and Implementation of an API Simulator: fork/exec

When executing each instruction, a set of boilerplate code is run every time. First, the operating system switches into kernel mode, then the context switch occurs, then it goes to the vector table to find the ISR for the vector for that instruction, and then puts the address into the PC. After executing the instruction, it calls the scheduler and returns from the interrupt (called IRET in this simulation.) The differences between each instruction depends on what the instruction is; The CPU instruction initiates a CPU burst after returning from the interrupt. The SYSCALL and END_IO instructions initiate a system call and I/O ISR respectively. The fork and exec system calls are quite a bit more complicated however.

When fork is called, the parent process is duplicated and the copy then finds and claims a large enough partition for itself. The process's trace file is a duplicate of the parent's after the fork, until the IF_PARENT or ENDIF instruction. This copy usually (but not necessarily always) contains an exec function call, which will be discussed next.

Usually when exec is called, it is done in a forked copy, to not overwrite the original (unless that's what you want I suppose.) The exec will load that program into memory, overwriting the child's slot on the PCB. The new process will need to find a new partition if it's too large however. The PCB is updated, and then the loaded program will start running. Afterwards, the control is given back to the parent and continues from there.

```
-----Given Example 1-----
0, 1, switch to kernel mode // Running fork
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 // Running exec

time: 24; current trace: FORK, 10
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|   1 |        init |               5 |    1 | running |
|   0 |       init |             6 |    1 | waiting |

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 // Executing program1

time: 247; current trace: EXEC program1_1, 50
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|   1 |  program1_1 |             4 |   10 | running |
|   0 |       init |             6 |    1 | waiting |
+------------------------------------------------------+

347, 1, switch to kernel mode // Running exec for program2
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 // Executing program2

time: 620; current trace: EXEC program2_1, 25
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|  1 |  program2_1 |           3 |   15 | running |
+------------------------------------------------------+
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
------------------------------
```

```
-------My  Example 1-------
0, 1, switch to kernel mode // Running fork
...
29, 1, switch to kernel mode // Running exec

time: 29; current trace: FORK, 15
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|  1 |       init |            5 |    1 | running |
|  0 |       init |            6 |    1 | waiting |
+------------------------------------------------------+
...
336, 1, IRET
337, 50, CPU Burst // Executing program1

time: 337; current trace: EXEC program1_4, 60
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|  1 |  program1_4 |           3 |   15 | running |
|  0 |       init |            6 |    1 | waiting |
+------------------------------------------------------+
...
400, 564, SYSCALL ISR (ADD STEPS HERE)
964, 1, IRET
965, 1, switch to kernel mode // Running second fork
...
999, 10, CPU Burst // CPU burst before exec
```

```
time: 999; current trace: FORK, 20
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|   2 |  program1_4 |               2 |   15 | running |
|   0 |        init |               6 |    1 | waiting |
|   1 |  program1_4 |               3 |   15 | waiting |
+------------------------------------------------------+
...
1121, 1, IRET
1122, 1, switch to kernel mode // Running exec

time: 1122; current trace: EXEC program2_4, 15
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|   2 |  program2_4 |               5 |    5 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------------+
...
1135, 564, SYSCALL ISR (ADD STEPS HERE) // Executing program2
1699, 1, IRET
------------------------------

-------My Example 2-------
0, 1, switch to kernel mode // Running fork
...
53, 1, IRET
54, 1, switch to kernel mode // Running exec 1

time: 54; current trace: FORK, 40
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|   1 |        init |               5 |    1 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------------+
...
```

```
556, 1, IRET
557, 5, CPU Burst // Executing program 1

time: 557; current trace: EXEC program1_5, 30
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|   1 |  program1_5 |               1 |   30 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------------+
...
595, 1, IRET
596, 1, switch to kernel mode // Running second fork

time: 596; current trace: FORK, 20
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|   2 |  program1_5 |               0 |   30 | running |
|   0 |        init |               6 |    1 | waiting |
|   1 |  program1_5 |               1 |   30 | waiting |
+------------------------------------------------------+
...
938, 1, IRET
939, 1, switch to kernel mode // Running second exec

time: 939; current trace: EXEC program2_5, 20
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|   2 |  program2_5 |               2 |   20 | running |
|   0 |        init |               6 |    1 | waiting |
|   1 |  program1_5 |               1 |   30 | waiting |
+------------------------------------------------------+
...
952, 211, SYSCALL ISR (ADD STEPS HERE) // Executing program 2
1163, 1, IRET
1164, 1, switch to kernel mode // Back to the first parent
```

```
...
1346, 1, IRET
1347, 1, CPU Burst // Executing program 3

time: 1347; current trace: EXEC program3_5, 10
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|   1 |  program3_5 |                4 |   10 | running |
+------------------------------------------------------+
------------------------------
```