Carleton University

Department of Systems and Computer Engineering

SYSC 4001 Operating Systems Fall 2025

Assignment 2 – Report

SYSC4001 L2 - 5

Mohamed Cherif Bah 101292844

Nathaniel Baird 101314576

Simulation 1:

INPUTS:

Contents of trace file:	Contents of program1 file:	Contents of program2 file:	Contents of external_files:
FORK, 10 IF_CHILD, 0 EXEC program1, 50 IF_PARENT, 0 EXEC program2, 25 ENDIF, 0	CPU, 100	SYSCALL, 4	program1, 10 program2, 15

OUTPUTS:

Contents of execution file:

```
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, 4, cloning the PCB
17, 0, scheduler called
17, 1, IRET
18, 1, switch to kernel mode
19, 10, context saved
29, 1, find vector 3 in memory position 0x0006
30, 1, load address 0X042B into the PC
31, 50, Program is 10 Mb large
81, 150, loading program into memory
231,7, marking partition as occupied
238,8, 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
610,6, marking partition as occupied
616,4, updating PCB
620, 0, scheduler called
620, 1, IRET
621, 1, switch to kernel mode
622, 10, context saved
632, 1, find vector 4 in memory position 0x0008
633, 1, load address 0X0292 into the PC
634, 250, SYSCALL ISR (ADD STEPS HERE)
884, 1, IRET
```

Contents of system_status file:

```
time: 18; 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: 621; current trace: EXEC program2, 25
+--------------------------------------------------+
| PID |program name |partition number | size |   state |
+--------------------------------------------------+
|   0 |    program2 |               3 |   15 | running |
+--------------------------------------------------+
```

ANALYSIS:

At the start only init exist which is given PID 0 and has a size of 1 mb, it is placed in the smallest free partition that fits which is partition 6.

FORK,10

The trace then starts with a fork which is called by the init, this fork creates a duplicate of the init program (the ISR clones the PCB) which has the PID number 1. The scheduler then gives priority to the child which is running in partition 5 and the parent moves to waiting.

IF_CHILD, 0

EXEC program1, 50

The EXEC call replaces the child program image with program1. The program looks up the size in external files, which is 10mb. The loader time is calculated (10 x 15 = 150). The handler frees up any previous image for the child; it then allocates the smallest partition that fits 10mb and places the child program there (which is partition 4), it then marks the partition occupied, updates the PCB, calls the scheduler, and returns.

IF_PARENT, 0

ENDIF, 0

These lines belong to the parent branch, nothing changes because the parent is not on the CPU, they are structural no ops here.

Program1 runs, CPU 100

The child now executes the CPU burst inside program1, this advances the simulation time only, it does not change the PCB table, so no new system status block is printed.

EXEC program2, 25

Control returns to the top-level trace, and the next EXEC is performed by the process that currently owns the CPU, the handler looks up program2 which is 15mn, the

loader time is 15 x 15 =225, the old image is freed, the smallest partition is chosen for 15 mb which is partition 3. The partition is marked as occupied, the PCB is updated, the scheduler is called, and the handler returns, the system status now shows program2 running in partition 3.

Program 2 runs, SYSCALL 4

Program2 issues a SYSCALL, the kernel boilerplate runs, context is saved, the vector is read, the device service routine runs for the configured delay, then the system returns from interrupt, this does not change the PCB table so there is no new system status block.

Simulation 2:

INPUTS:

Contents of trace file:	Contents of program1 file:	Contents of program2 file:	Contents of external_files:
FORK, 17 IF_CHILD, 0 EXEC program1, 16 IF_PARENT,0 ENDIF, 0 CPU, 205	FORK, 15 IF_CHILD, 0 IF_PARENT, 0 ENDIF, 0 EXEC program2, 33	CPU, 53	program1, 10 program2, 15

OUTPUTS:

Contents of execution file:

```
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, 4, cloning the PCB
17, 0, scheduler called
17, 1, IRET
18, 1, switch to kernel mode
19, 10, context saved
29, 1, find vector 3 in memory position 0x0006
30, 1, load address 0X042B into the PC
31, 16, Program is 10 Mb large
47, 150, loading program into memory
197,7, marking partition as occupied
204,8, updating PCB
212, 0, scheduler called
212, 1, IRET
213, 1, switch to kernel mode
214, 10, context saved
224, 1, find vector 2 in memory position 0x0004
225, 1, load address 0X0695 into the PC
226, 6, cloning the PCB
232, 0, scheduler called
232, 1, IRET
233, 1, switch to kernel mode
234, 10, context saved
244, 1, find vector 3 in memory position 0x0006
245, 1, load address 0X042B into the PC
246, 33, Program is 15 Mb large
279, 225, loading program into memory
504,4, marking partition as occupied
508,6, updating PCB
514, 0, scheduler called
514, 1, IRET
515, 53, CPU Burst
568, 1, switch to kernel mode
569, 10, context saved
579, 1, find vector 3 in memory position 0x0006
580, 1, load address 0X042B into the PC
581, 33, Program is 15 Mb large
614, 225, loading program into memory
839,7, marking partition as occupied
846,3, updating PCB
849, 0, scheduler called
849, 1, IRET
850, 53, CPU Burst
903, 205, CPU Burst
```

Contents of system_status file:

```
time: 18; current trace: FORK, 17
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|   1 |        init |               5 |    1 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------------+

time: 213; current trace: EXEC program1, 16
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|   1 |    program1 |               4 |   10 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------------+

time: 233; 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: 515; 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: 850; current trace: EXEC program2, 33
+------------------------------------------------------+
| PID |program name |partition number | size |   state |
+------------------------------------------------------+
|   1 |    program2 |               3 |   15 | running |
|   0 |        init |               6 |    1 | waiting |
+------------------------------------------------------+
```

ANALYSIS:

At the start only init exist which is given PID 0 and has a size of 1 mb, it is placed in the smallest free partition that fits which is partition 6.

FORK,17

The trace then starts with a fork which is called by the init, this fork creates a duplicate of the init program (the ISR clones the PCB) which has the PID number 1. The scheduler then gives priority to the child which is running in partition 5 and the parent moves to waiting.

IF_CHILD, 0

EXEC program1, 16

The child takes the child path and executes EXEC. The handler looks up program1 in the external files list and finds size 10mb. The loader time is then computed as 10x15 = 150. The old image is freed; the allocator picks the smallest partition that fits 10 mb which is partition 4. The partition is marked as occupied and the PCB is updated. The second system table shows pid 1 running program1 in partition 4 and PID 0 init waiting in partition 6.

IF_PARENT,0

ENDIF,0

This command belongs to the parent branch; nothing changes here because the parent is not on the CPU.

FORK, 15

The running process is PID 1 program1, it calls the FORK command. The kernel creates a new child with PID 2. The scheduler give priority to the new child. The allocator places this child in partition 3 because it fits the size 10mb. PID 2 program1 is running partition 3. PID 1 program1 is waiting in partition 4. PID 0 init is still waiting in partition 6.

Inside program1 both IF_CHILD 0 and IF_PARENT 0 then ENDIF 0 do nothing they are structural and change nothing.

EXEC program2, 33

The process that currently has the CPU is the child with PID 2. It performs EXEC on program2, the handler looks up the size which is 15mb. The loader time is 15 x 15 = 225. The old image is freed, partition 3 still fits 15mb so it is kept and marked occupied, the PCB is updated.

Later the scheduler returns to the parent created by the program1 FORK, that parent also reached the same EXEC command in program2 and executes it, the handler again loads a 15mb image with loader time of 225. The allocator picks the smallest partition that fits 15mb for this parent which is partition 2, while PID 0 init remains in partition 6 waiting.

CPU, 53

This only advances the simulation time.

EXEC program1, 60

The child takes the child path and executes EXEC. The handler looks up program1 in the external files list and finds size 10mb. The loader time is then computed as 10x15 = 150. The old image is freed; the allocator picks the smallest partition that fits 10 mb which is partition 4. The partition is marked as occupied and the PCB is updated. The second system table shows pid 1 running program1 in partition 4 and PID 0 init waiting in partition 6.

Simulation 3:

INPUTS:

Contents of trace file:	Contents of program1 file:	Contents of program2 file:	Contents of external_files:
FORK, 20 IF_CHILD, 0 IF_PARENT, 0 EXEC program1, 60 ENDIF, 0 CPU, 10	CPU, 50 SYSCALL, 6 CPU, 15 END_IO, 6	CPU, 53	program1, 10 program2, 15

OUTPUTS:

Contents of execution file:	Contents of system_status file:
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, 4, cloning the PCB 17, 0, scheduler called 17, 1, IRET 18, 10, CPU Burst 28, 1, switch to kernel mode 29, 10, context saved 39, 1, find vector 3 in memory position 0x0006 40, 1, load address 0X042B into the PC 41, 60, Program is 10 Mb large 101, 150, loading program into memory 251,7, marking partition as occupied 258,8, updating PCB 266, 0, scheduler called 266, 1, IRET 267, 50, CPU Burst 317, 1, switch to kernel mode 318, 10, context saved 328, 1, find vector 6 in memory position 0x000C 329, 1, load address 0X0639 into the PC 330, 265, SYSCALL ISR (ADD STEPS HERE) 595, 1, IRET 596, 15, CPU Burst 611, 1, switch to kernel mode 612, 10, context saved 622, 1, find vector 6 in memory position 0x000C 623, 1, load address 0X0639 into the PC 624, 265, ENDIO ISR(ADD STEPS HERE) 889, 1, IRET	time: 18; current trace: FORK, 20 +----------+ | PID |program name |partition number | size | state | +----------+ | 1 | init | 5 | 1 | running | | 0 | init | 6 | 1 | waiting | +----------+ time: 267; current trace: EXEC program1, 60 +----------+ | PID |program name |partition number | size | state | +----------+ | 0 | program1 | 4 | 10 | running | +----------+

ANALYSIS:

At the start only init exist which is given PID 0 and has a size of 1 mb, it is placed in the smallest free partition that fits which is partition 6.

FORK, 20

The trace then starts with a fork which is called by the init, this fork creates a duplicate of the init program (the ISR clones the PCB) which has the PID number 1. The scheduler then gives priority to the child which is running in partition 5 and the parent moves to waiting.

IF_CHILD, 0

IF_PARENT, 0

Both branches are empty, nothing changes. The child keeps the CPU.

EXEC program1, 60

The child takes the child path and executes EXEC. The handler looks up program1 in the external files list and finds size 10mb. The loader time is then computed as 10x15 = 150. The old image is freed; the allocator picks the smallest partition that fits 10 mb which is partition 4. The partition is marked as occupied and the PCB is updated. The second system table shows pid 1 running program1 in partition 4 and PID 0 init waiting in partition 6.

CPU, 50

These advances time only, no PCB change.

SYSCALL, 6

The CPU traps to vector 6. The device service runs for the configured delay, then returns from the interrupt. No PCB change

CPU, 15

These advances time only, no PCB change.

END_IO, 6

The end of the IO interrupt for device 6, the handler runs and returns. No PCB change.

CPU,10

These advances time only, no PCB change.

Simulation 4:

INPUTS:

Contents of trace file:	Contents of program1 file:	Contents of program2 file:	Contents of external_files:
FORK, 12 IF_CHILD,0 CPU, 30 IF_PARENT,0 EXEC program2, 20 ENDIF,0 CPU, 50	CPU, 50 SYSCALL, 6 CPU, 15 END_IO, 6	CPU, 53	program1, 10 program2, 15

OUTPUTS:

Contents of execution file:	Contents of system_status file:
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, 4, cloning the PCB 17, 0, scheduler called 17, 1, IRET 18, 30, CPU Burst 48, 50, CPU Burst 98, 1, switch to kernel mode 99, 10, context saved 109, 1, find vector 3 in memory position 0x0006 110, 1, load address 0X042B into the PC 111, 20, Program is 15 Mb large 131, 225, loading program into memory 356,7, marking partition as occupied 363,8, updating PCB 371, 0, scheduler called 371, 1, IRET 372, 53, CPU Burst	time: 18; current trace: FORK, 12 +---------- | PID |program name |partition number | size | state +---------- | 1 | init | 5 | 1 | running | 0 | init | 6 | 1 | waiting +---------- time: 372; current trace: EXEC program2, 20 +---------- | PID |program name |partition number | size | state +---------- | 0 | program2 | 3 | 15 | running +----------

ANALYSIS:

Init starts alone with PID 0 in partition 6. A fork clones Init, the child receives PID 1, takes the CPU in partition 5, and the parent moves to waiting in partition 6. The child runs a CPU burst, which only advances time. Later the parent executes exec program2. The handler looks up Program2 as 15 MB, computes loader time as 15 × 15 = 225, frees the old image, and places the new image in the smallest fitting slot, which is partition 3. The PCB is updated, the scheduler is called, and Program2 runs. Program2 then performs its own CPU

burst, which again only advances time. Only the fork and the exec change the system table, so snapshots appear right after those two points.

Simulation 5:

INPUTS:

Contents of trace file:	Contents of program1 file:	Contents of program2 file:	Contents of external_files:
FORK, 10 IF_CHILD, 0 EXEC program1, 12 IF_PARENT, 0 ENDIF, 0 FORK, 8 IF_CHILD, 0 EXEC program2, 18 IF_PARENT, 0 ENDIF, 0	FORK, 12 IF_CHILD, 0 EXEC program2, 16 IF_PARENT, 0 ENDIF, 0 CPU, 20	CPU, 30	program1, 10 program2, 15

OUTPUTS:

Contents of execution file:	Contents of system_status file:

```
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, 4, cloning the PCB
17, 0, scheduler called
17, 1, IRET
18, 1, switch to kernel mode
19, 10, context saved
29, 1, find vector 3 in memory position 0x0006
30, 1, load address 0X042B into the PC
31, 12, Program is 10 Mb large
43, 150, loading program into memory
193,7, marking partition as occupied
200,8, updating PCB
208, 0, scheduler called
208, 1, IRET
209, 1, switch to kernel mode
210, 10, context saved
220, 1, find vector 2 in memory position 0x0004
221, 1, load address 0X0695 into the PC
222, 6, cloning the PCB
228, 0, scheduler called
228, 1, IRET
229, 1, switch to kernel mode
230, 10, context saved
240, 1, find vector 3 in memory position 0x0006
241, 1, load address 0X042B into the PC
242, 16, Program is 15 Mb large
258, 225, loading program into memory
483,4, marking partition as occupied
487,6, updating PCB
493, 0, scheduler called
493, 1, IRET
494, 30, CPU Burst
524, 20, CPU Burst
544, 1, switch to kernel mode
545, 10, context saved
555, 1, find vector 2 in memory position 0x0004
556, 1, load address 0X0695 into the PC
557, 7, cloning the PCB
564, 0, scheduler called
564, 1, IRET
565, 1, switch to kernel mode
566, 10, context saved
576, 1, find vector 3 in memory position 0x0006
577, 1, load address 0X042B into the PC
578, 18, Program is 15 Mb large
596, 225, loading program into memory
821,3, marking partition as occupied
824,10, updating PCB
834, 0, scheduler called
834, 1, IRET
835, 30, CPU Burst
```

```
time: 18; current trace: FORK, 10
+-------------------------------------------------+
| PID |program name |partition number | size |   state |
+-------------------------------------------------+
|   1 |        init |               5 |    1 | running |
|   0 |        init |               6 |    1 | waiting |
+-------------------------------------------------+

time: 209; current trace: EXEC program1, 12
+-------------------------------------------------+
| PID |program name |partition number | size |   state |
+-------------------------------------------------+
|   1 |    program1 |               4 |   10 | running |
|   0 |        init |               6 |    1 | waiting |
+-------------------------------------------------+

time: 229; current trace: FORK, 12
+-------------------------------------------------+
| PID |program name |partition number | size |   state |
+-------------------------------------------------+
|   2 |    program1 |               3 |   10 | running |
|   0 |        init |               6 |    1 | waiting |
|   1 |    program1 |               4 |   10 | waiting |
+-------------------------------------------------+

time: 494; current trace: EXEC program2, 16
+-------------------------------------------------+
| PID |program name |partition number | size |   state |
+-------------------------------------------------+
|   2 |    program2 |               3 |   15 | running |
|   0 |        init |               6 |    1 | waiting |
|   1 |    program1 |               4 |   10 | waiting |
+-------------------------------------------------+

time: 565; current trace: FORK, 8
+-------------------------------------------------+
| PID |program name |partition number | size |   state |
+-------------------------------------------------+
|   1 |        init |               5 |    1 | running |
|   0 |        init |               6 |    1 | waiting |
+-------------------------------------------------+

time: 835; current trace: EXEC program2, 18
+-------------------------------------------------+
| PID |program name |partition number | size |   state |
+-------------------------------------------------+
|   1 |    program2 |               3 |   15 | running |
|   0 |        init |               6 |    1 | waiting |
+-------------------------------------------------+
```

ANALYSIS:

Init starts alone with PID 0 in partition 6 and is running. A fork of duration 10 clones Init; the child gets PID 1 and runs in partition 5, while the parent waits in partition 6. The child then execs Program1. The loader looks up size 10 MB and takes 10 × 15 = 150-time units. The old image is freed, and the child is placed in the best-fit partition 4. Inside Program1 the running process forks again. A new child with PID 2 is scheduled in partition 3, while PID 1 waits in partition 4 and Init still waits in partition 6. That child then execs

Program2. Size 15 MB yields 15 × 15 equals 225 = units. It remains in partition 3 because it fits and becomes the running process. Later the waiting parent of that inner fork resumes at the top level, performs another fork, and finally execs Program2. The 15 MB load again costs 225-time units. It frees partition 4 and moves to the best-fit partition 3. The last table shows PID 1 running Program2 in partition 3, with Init still waiting in partition 6. Only the fork and exec steps change the PCB table. CPU bursts only advance time.