Simulation Notes: Example 1

Craig A. Struble

October 19, 2000

1 Introduction

In *CS3204: Operating Systems*, three simulations for the second programming project were covered. This document covers the first of those three simulations, using the table formats used in class. In this example, I have split up the input files by `simulate` commands and comment on each set of events up to the simulate command.

2 Example

We start by initializing the simulation with 10 units of memory, a start time of 0, a time quantum of 0.050, a context switch overhead of 0.010, handling up to 10 active processes, and using a 4 level multi-level feedback queue. Three system processes are created, with identifiers 1, 2, and 3. The memory and amount of time to execute follows the process identifiers.

```
init 10 0.0 0.050 0.010 10 4
create 0 1 3 0.100
create 0 2 2 0.200
create 0 3 1 0.070
simulate 0.70
```

The sequence of events that occur are as follows:

Time	Event	Queue	Mem Remaining	Time Remaining
0.0	1A	(1) 1	7	1 0.100
0.0	2A	(1) 1 2	5	2 0.200
0.0	3A	(1) 1 2 3	4	3 0.070
0.01	1S	(1) 2 3		
0.06	1TQ	(1) 2 3		1 0.050
0.07	1CS	(1) 2 3		
		(2) 1		
0.07	TE			

We see at time 0 all three system processes are created. Once the system has finished creating the system processes (handling system commands), the scheduler is called to dispatch a process to the CPU. The first process in the ready list is process 1. This process is scheduled to start running at time 0.01, adding the context switch overhead to the system time. Process 1 runs for a full time quantum. At time 0.06 process 1 is interrupted and a context switch initiated. The context switch completes at time 0.07. Process 1 has its priority reduced and is placed back into the ready list. Simulation time ends at 0.07.

The next set of commands create a child of process 2 with id 4.

```
create 2 4 2 0.150
simulate 0.020
```

The sequence of events generated from these commands is

Time	Event	Queue	Mem Remaining	Time Remaining
0.08	2S	(1) 3	4	
		(2) 1		
0.08	4A	(1) 3 4	2	4 0.0150
		(2) 1		
0.09	TE		2 0.190	

The system time is initially 0.07 when the simulation starts again. The system checks its command queue and finds it empty, so the scheduler is called to dispatch a process. Process 2 is at the head of the queue, so it is scheduled to run at 0.08, incorporating the context switch cost. Once process 2 starts executing, it checks its command queue. The command to create a child process is found, so process 4 is created at 0.08 (process commands do not advance system time). At 0.09, simulation time expires, so the parser takes over again and starts reading the input file.

The following commands appear in the input file. Process 3 requests to create a child process with id 5, and then waits for that process to terminate. Waiting for a child process will block process 3.

```
create 3 5 1 0.050
wait 3 5
simulate 0.070
```

The `create` and `wait` commands are placed in process 3's command queue.

The sequence of events is in the table below.

Time	Event	Queue	Mem Remaining	Time Remaining
0.13	2TQ	(1) 3 4	2	2 0.150
		(2) 1		
0.14	2CS	(1) 3 4		
		(2) 1 2		
0.15	3S	(1) 4		
		(2) 1 2		
0.15	5A	(1) 4 5	1	5 0.050
		(2) 1 2		
0.15	3W			3 0.070
0.16	3CS			
0.16	TE			

The simulation starts again at 0.09, and the system starts by checking for system commands. None exist, so process 2 continues to execute for its time quantum. At time 0.13, process 2 completes its time quantum and a context switch is initiated. At time 0.14, the context switch completes and process 2 is placed in level 2 (having its priority reduced) of the multilevel feedback queue. Since there are no commands for the system to handle, the scheduler is called to dispatch another process. Process 3 is the next process in the multilevel feedback queue, so a context switch is started and process 3 starts executing at time 0.15. The commands in process 3's command queue are executed. First, a child process with id 5 is created at time 0.15 (remember, process commands do not advance times). Following the creation of process 5, process 3 waits for its newly created child to terminate. Waiting blocks process 3, so a context switch is initiated. The context switch completes at 0.16, and process 3 is placed into a blocked state. Some indicator (e.g., a flag in the parent or child) is set to recognize that process 3 is waiting. Simulation time expires at 0.16.

The file contains one final `simulate` command to complete the simulation execution.

```
simulate 0.84
```

The remaining sequence of events is in the following table.

Time	Event	Queue	Mem Remaining	Time Remaining
0.17	4S	(1) 5	1	
		(2) 1 2		
0.22	4TQ			4 0.100
0.23	4CS	(1) 5		
		(2) 1 2 4		
0.24	5S	(1)		
		(2) 1 2 4		
0.29	5F			5 0.000
0.30	5CS	(1) 3	2	
		(2) 1 2 4		
0.31	3S	(1)		
0.36	3TQ			3 0.020
0.37	3CS	(2) 1 2 4 3		
0.38	1S	(2) 2 4 3		
0.43	1F			1 0.000
0.44	1CS		5	
0.45	2S	(2) 4 3		
0.50	2TQ			2 0.100
0.51	2CS	(2) 4 3		
		(3) 2		
0.52	4S	(2) 3		
		(3) 2		
0.57	4TQ			4 0.050
0.58	4CS	(2) 3		
		(3) 2 4		
0.59	3S	(3) 2 4		
0.61	3F		6	3 0.000
0.62	3CS	(3) 2 4		
0.63	2S	(3) 4		
0.68	2TQ			2 0.050
0.69	2CS	(3) 4		
		(4) 2		
0.70	4S	(4) 2		
0.75	4F			4 0.000
0.76	4CS		8	
0.77	2S			
0.82	2F		10	2 0.000
0.83	2CS			
1.00	TE			

The simulation starts at system time 0.16. The system checks and finds no system commands, so the scheduler is called to dispatch a process, in this case process 4. A context switch is started and completes at 0.17, at which time process 4 starts execution. Process 4 runs for its time quantum, which completes at 0.22. The context switch for process 4 completes at time 0.23, and the process is placed in level 2 of the feedback queue. The scheduler is called to dispatch another process, process 5 in this example. Process 5 starts execution at time 0.24 and runs until completion (which happens to also be the full amount for a time quantum) at time 0.29. The context switch back to the operating system completes at time 0.30. Because process 5 has completed, process 3 stops waiting and is placed back in the ready list at the priority that it had when it was blocked. Process 5 is then placed in the terminated list. The remainder of the time is spent scheduling and executing processes until they complete. After time 0.83, the operating simply spends the rest of the time in an idle state. Simulation time expires at 1.00.