

Problem Session 1: Processes and Scheduling

SOLUTION

Wednesday, April 1, 2020

1. How many lines of output does the following function print? Give your answer as a function of n .

```
void foo(unsigned n){
    for(unsigned i = 0; i < n; i++){
        fork();
    }
    printf("hello\n");
}
```

Solution: 2^n

Note that `fork` create a new process with the same code and the same state as the parent process, so each iteration through the loop the total number of processes doubles. At the end of the function, each process prints one line, so the total number of lines printed is equal to the total number of processes (2^n).

2. Consider the following program:

```
void f(){
    printf("2");
}

int main(){
    int check = 0;

    if(fork() == 0){
        check = 1;
    }

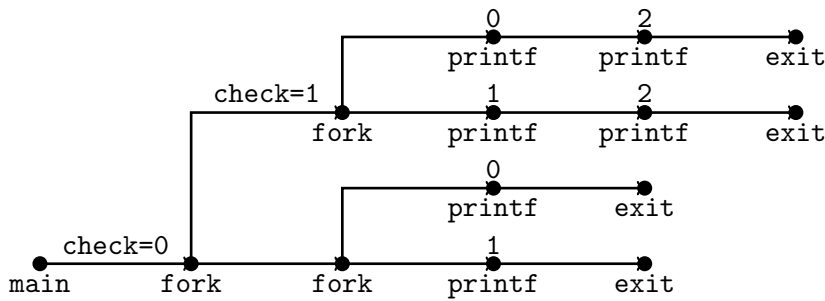
    if(fork() == 0){
        printf("0");
    } else {
        printf("1");
    }

    if(check){
        f();
    }
    exit(0);
}
```

Which of the following outputs are possible:

- (a) 112002
- (b) 211020
- (c) 102120
- (d) 122001
- (e) 100212

Solution: (a), (c), and (e) are possible. (b) and (d) are not. To see this, draw the process graph:

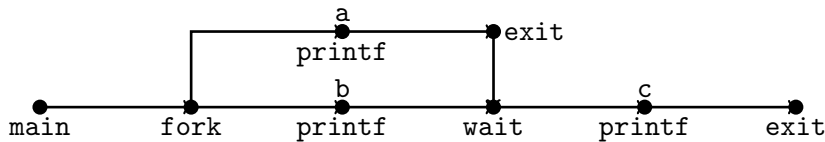


3. What are the possible output sequences from the following program?

```

int main(){
    if(fork() == 0){
        printf("a");
        exit(0);
    } else {
        printf("b");
        wait();
    }
    printf("c");
    exit(0);
}
  
```

Solution: The possible outputs of this program are abc and bac. To see this, draw the process diagram:

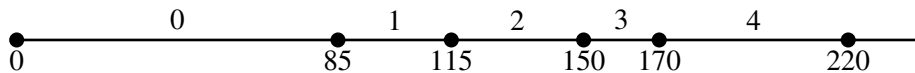


4. Given the following jobs, compute the latency and response time for each job, along with the average response time, for FIFO, STCF, and RR scheduling algorithms. Assume a time slice of 10 for RR.

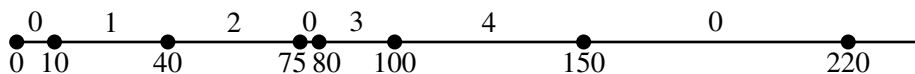
Job	Length	Arrival Time	FIFO		STCF		RR	
			Latency	Response	Latency	Response	Latency	Response
0	85	0	85	0	220	0	220	0
1	30	10	105	75	30	0	70	0
2	35	15	135	100	60	25	110	5
3	20	80	90	70	20	0	55	10
4	50	85	135	85	65	15	120	15
Average:			90	66	79	8	115	6

Solution: See table above. To see this, complete the timeline schedule for this set of jobs for each of the scheduling algorithms as follows:

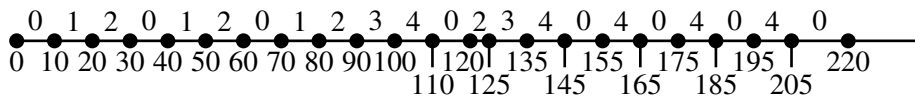
First in, First out (FIFO):



Shortest Time-to-Completion First (STCF):



Round Robin (RR):



5. Consider a set of three jobs, A, B, and C, running concurrently on a computer system:

- Job A arrives first at time 0 and uses the CPU for 50ms before finishing.
- Job B arrives at time 1. Job B loops five times; for each iteration of the loop, B uses the CPU for 2ms and then does I/O for 8ms.
- Job C arrives at time 2. Job C is identical to Job B except for the arrival time.

Assuming there is no overhead to doing a context switch, identify when A, B, and C will finish for each of the following scheduling algorithms:

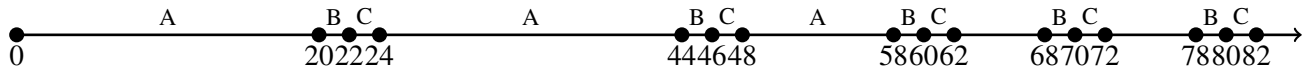
- RR with a 1ms time slice
- RR with a 20 ms time slice
- Multilevel feedback queue with four levels with a time slice of 10 in the highest priority queue, 20 in the next, 40 in the next, and 80 in the lowest priority queue. Priorities reset every 200ms.

Solution: To solve this problem, first complete the timeline schedule for this set of jobs for each of the scheduling algorithms. Then compute the completion time.

Round Robin (RR) with 1ms time slice: Assuming that there is no contention for I/O resources, Job A completes at time 70. Job B finishes on the CPU at time 53 and finishes I/O at time 61. Job C finishes on the CPU at time 54 and finishes I/O at time 62.



Round Robin (RR) with 20ms time slice: Assuming that there is no contention for I/O resources, Job A completes at time 58, Job B finishes on the CPU at time 80 and finishes I/O at time 88, and Job C finishes on the CPU at time 82 and finishes I/O at time 90.



Multilevel feedback queue: Assuming that there is no contention for I/O resources, Job A completes at time 70, Job B finishes on the CPU at time 52 and finishes I/O at time 60, and Job C finishes on the CPU at time 54 and finishes I/O at time 62.

