CSE 332 Final

Güneş ERÇAL, Olcay Taner YILDIZ

I. Question (20 points)

Given the following C program, draw the process tree, with meaningful process numbering. Remember that the return code for fork() is zero for the child process, whereas the nonzero process identifier of the child is returned to the parent.

```
void main(void)
{
 int p[5], q[5];
 int temp1, temp2;

 for (int i = 0; i < 5; i++) {
   temp1 = fork();
   if (temp1 > 0) {
      p[i] = temp1;
   } // end if
   else {
      temp2 = fork();
      if (temp2 > 0) {
         q[i] = temp2;
      } // end inner if
      return; // NOTE THE RETURN CASE
   } // end else
 } // end for
}
```

II. Question (20 points)

Consider the situation that there are $n = 99$ processes $P_1, P_2, P_3, \ldots, P_n$ arriving simultaneously in that same order $P_1, P_2, \ldots, P_n$ with CPU burst times as follows: Each of processes P_1 through P_{33} has a CPU burst time of 99 milliseconds. Processes P_{34} through P_{66} each has a CPU burst time of 66 milliseconds. And processes P_{65} through P_{99} each has a CPU burst time of 33 milliseconds. Consider running the following algorithms on this process sequence: FCFS, SJF, SJRF (pre-emptive), RR with quantum = 1 millisecond and RR with quantum = 50 milliseconds. Compare each of these five algorithms on this process sequence in terms of their average response times. In particular, order the five algorithms on this process sequence from best average response time to worst average response time.

III. Question (15 points)

We are given $n > 3$ processes, $P_0, P_1, P_2, \ldots, P_{n-1}$ with statements $S_1, S_2, \ldots, S_n$ respectively corresponding to each process (such that statement S_i is in the code of process P_{i-1}) with the following constraints:

- There are no constraints on S_0.
- If i is even, then S_i must execute after both statements $S_{\frac{i}{2}}$ and S_{i-1} has completed execution.
- If i is odd, then S_i must execute after statement S_1 has completed execution.

a. Using arrays of semaphores with appropriate initializations and separating into reasonable cases, write the code for the processes in such a way that enforces the above precedence constraints.

b. Consider adding the following additional constraint: If i is prime, then S_i must execute after $S_{(i+3) mod n}$ has completed execution. Using ideas from resource allocation graphs, where each statement is a resource, demonstrate that this additional constraint is guaranteed to create a deadlock situation. Similarly, explain why the original set of constraints does not create deadlock.

IV. Question (20 points)

Consider a reference string S_1 of length 20 formed by repeating the reference string $1, 2, 3, 4, 5$ four times in a row, and another reference string S_2 also of length 20 formed by repeating the reference string $1, 2, 3, 4, 5, 5, 4, 3, 2, 1$ twice in a row. Assuming each page frame is initially empty and that there are 4 available frames in the following page replacement algorithms:

a. How many page faults occur for string S_1 with LRU?
b. How many page faults occur for string S_1 with OPT?
c. How many page faults occur for string S_2 with LRU?
d. How many page faults occur for string S_2 with OPT?

V. Question (15 points)

Let $n > 1$ be given, and say we have $n + 1$ memory partitions $2^n, 2^{n-1}, 2^{n-2}, \ldots, 2, 1$ kilobytes given in that order. Recall that $2^n = \Sigma_{i=0}^{n-1} 2^i + 1$. Consider that $2n$ processes arrive in the following size order: First n processes arrive with sizes in order $2^{n-1}, 2^{n-2}, 2^{n-3}, \ldots, 2, 1$ kilobytes followed by n more processes with sizes in the order $2^{n-1}, 2^{n-2}, 2^{n-3}, \ldots, 2, 2$ kilobytes (note that only the last process size is different: 2 instead of 1). Prove that amongst the algorithms First-Fit, Best-Fit, and Worst-Fit, only Best-Fit will succeed in placing all of the $2n$ processes into the available partitions.

VI. True or False Questions (10 points)

For each of the following questions, 1 through 10, write either "TRUE" if the statement is True, or "FALSE" if the statement is False.

1. One way to ensure that the acyclic directory structure is maintained is by allowing links only to files but not to subdirectories.
2. Given a logical address space of eight pages of 32 words each, there are 10 bits in the logical address.
3. Turning off the timer interrupt must be a protected instruction.
4. Interactive processes do not favor round robin scheduling.
5. The number of queues is a parameter of multilevel feedback queue scheduler.
6. By supporting multiple file structures, operating system can easily handle files, can distinguish them and can use sets of special operations for manipulating files with those structures.
7. The necessary conditions for a deadlock to occur are Mutual exclusion, Hold and Wait.
8. The main problem with SJF policy is starvation.
9. Stack data structure is good for demand-paged environment.
10. Hash table is good for demand-paged environment.