

Operating System and System Administration Tutorial 05

1. Consider two processes that are concurrently executing an instruction i = i * j, where i is initialized to 3 and j is initialized to 2. Assume that the instruction is implemented in machine language as:

```
R1 = i;

R2 = j;

R1 = R1 * R2;

i = R1;
```

- a) Show that without synchronization, the final value of i can be 6. Note that the correct result should be 12.
- b) Use the wait(A) and signal(A) operations of the semaphore to synchronize the two processes. Note that the initial value of semaphore A must be specified.
- c) Illustrate one example of the incorrect solution to part (ii) that results in a deadlock.
- 2. Consider a system consisting of three processes, P0, P1 and P2, each is accessing three semaphores E, F and M. Assume E has value 2, F has 0, and M has 1.

| P0 | P1 | P2 |
|-----------|-----------|-----------|
| wait(E) | wait(F) | wait(F) |
| wait(M) | wait(M) | wait(M) |
| CS | CS | CS |
| signal(M) | signal(M) | signal(M) |
| signal(F) | signal(E) | signal(E) |

- a) Describe what the wait(F) and signal(F) functions do.
- b) Each wait() must be executed *atomically*.
 - Explain what it means by "executed atomically"
 - Describe what might happen if it is NOT executed atomically.
- c) If P0 is scheduled to run, will it block? Why?
- d) If P1 is scheduled to run will it block? Why?
- e) If the three processes run concurrently, list all possible sequence of process terminations. Justify your answer.