Exercise 5 – Synchronization, Resource allocation, Deadlocks

Some of the questions are taken from Stallings, Operating Systems Internals and Design Principles, fifth edition and Silberschatz et al., Operating System Concepts, seventh edition, as referred.

1 – Stallings 5.3

Consider the following program:

```
const int n = 50;
int tally;
void total()
{
  int count;
  for (count = 1; count <= n; count++)
  {
    tally++;
  }
}
void main()
{
  tally = 0;
  parbegin (total(), total());
  write (tally);
}
```

a) Determine the proper lower bound and upper bound on the final value of the shared variable tally output by this concurrent program. Assume processes can execute at any relative speed and that a value can only be incremented after it has been loaded into a register by a separate machine instruction.

b) Suppose that an arbitrary number of these processes are permitted to execute in parallel under the assumptions of part (a). What effect will this modification have on the range of final values of tally?

2 – Bounded buffer

Consider the bounded buffer producers-consumers problem (defined in section 6.1,6.6.1 in Silberschatz). Describe an implementation of a solution using:

a) Semaphores

b) Message-boxes

Argue about the correctness of your solution.

3 – Silberschatz 7.13

Consider the deadlock situation that could occur in the dining-philosophers problem when the philosophers obtain the chopsticks one at a time. Discuss

how the four necessary conditions for deadlock hold in this setting. Discuss how deadlocks could be avoided by eliminating any one of the four conditions.

HINT: Do not always stick to the chopsticks idea...

4 – Silberschatz 7.5

In a real computer system, neither the resources available nor the demands of processes for resources are consistent over long periods (months). Resources break or are replaced, new processes come and go, new resources are bought and added to the system. If deadlock is controlled by the banker's algorithm, which of the following changes can be made safely (without introducing the possibility of deadlock), and under what circumstances?

a) Increase Available (new resources added).

b) Decrease Available (resource permanently removed from system).

c) Increase Max for one process (the process needs more resources than allowed; it may want more).

d) Decrease Max for one process (the process decides it does not need that many resources).

e) Increase the number of processes.

f) Decrease the number of processes.

5 – Silberschatz 7.7

Consider a system consisting of m resources of the same type being shared by n processes. Resources can be requested and released by processes only one at a time. Show that the system is deadlock free if the following two conditions hold:

a) The maximum need of each process is between 1 and m resources.

b) The sum of all maximum needs is less than m + n.

6 – Bounded waiting mutex

Describe an implementation of a bounded waiting mutex using the *TestAndSet* operation.