CSCI 315 Operating Systems Design
Fall 2015 - Prof. Felipe Perrone
Activity 2

---

1) Consider Peterson's solution to the critical-section problem. The processes share the following data items:

```
int turn;
boolean flag[2];
```

The structure of process P_i is as follows:

```
do {
  flag[i] = TRUE;
  turn = j;
  while (flag[j] && turn == j);

  critical section

  flag[i] = FALSE;

  remainder section
} while (TRUE);
```

a) How many processes can use this solution?

b) What is the purpose of setting `flag[i]` to `TRUE`?

c) Can we say that only one process will be in its critical section, at any time? Explain.

d) What happens if more than one process executes `turn = j;` "at the same time" in the same CPU.

e) Does this solution cover the three requirements of **mutual exclusion**, **progress**, and **bounded waiting**? Present arguments for each one of these.

2) Consider the TestAndSet approach to build a *lock* with which to solve the critical section problem.

```
boolean TestAndSet(boolean *target) {
  boolean rv = *target; *target = TRUE; return rv;
}
```

a) Discuss whether a *software library's implementation* of TestAndSet is an effective approach to use in creating a solution to the critical section problem.

b) Show how one can use TestAndSet to implement a solution to the critical section problem.

c) Show how one can use CompareAndSwap to implement a solution to the critical section problem.