CS586 – Distributed Computing
Instructor: Panagiota Fatourou

1st Exercise

Deadline: November 14, 2011

Consider the implementation of a multi-writer m-component snapshot object presented in Figure 1. This implementation supports the execution of only one SCAN operation at each point in time (i.e., the implementation is not correct in case more than one SCAN operations are concurrently active at any point in time; such snapshot implementations are called *single-scanner*).

shared registers seq, V[∞][m]; // *initially, seq = 0, V[1][j] = ⊥ και V[i][j] = null, 1≤j≤m, ∀i>1*

```
void UPDATE(int j , value v) {
// UPDATE της συνιστώσας j με την τιμή v
    int lseq;

    lseq = seq;
    V[lseq][j] = v;
}
```

```
value *SCAN(void) {
    value a[m];
    int lseq, j, k;

    lseq = seq + 1;
    seq = lseq;
    for j = 1 to m do
        k = lseq;
        repeat {
            k = k – 1;
            a[j] = V[k][j];
        } until (a[j] ≠ NULL);
    return a;
}
```

Figure 1: A single scanner multi-writer m-component snapshot implementation from registers

The implementation employs a two-dimensional array V[∞][m] which has an infinite number of rows and m columns (one column for each snapshot component).

A SCAN S starts executing by incrementing a register, called seq. We remark that register seq is written only by SCAN operations, and it is read by any other process. Since there is just one active SCAN at each point in time, the sequence of values stored in seq is increasing; let each SCAN be characterized by the sequence number it writes in seq.

An UPDATE operation U on component A_j with value v, starts by reading the value of register seq. It then writes v in one of the positions of array V, namely V[lseq][j] (i.e., this position is on column j and on row lseq, where lseq is the value read by U in seq). Notice that in this way, each SCAN operation determines (by writing in seq) the row of V where some of the UPDATE operations should write their values.

Consider a SCAN S that writes s in seq. The UPDATE operations that write in one of the positions of row s have started their execution after the beginning of S and have ended their execution before the beginning of the SCAN following S.

A SCAN S that writes s in seq, reads, for each j, the array registers in column j starting from row s-1 towards row 0. S returns for component j the first value ≠ NULL it reads in any of these registers.

Prove that the implementation of Figure 1 is a correct implementation of a single-scanner multi-writer m-component snapshot. More specifically, for each execution of the implementation:

(1) explain how linearization points should be placed for the operations of the execution;

(2) prove that the linearization point of each operation is within the execution interval of the operation, and

(2) prove that SCAN operations calculate (and return) consistent vectors.

Important Notice: It is crucial that only one SCAN can be active at any point in time.