

## Filter Lock

Prove that the filter lock (given below) provides mutual exclusion for  $n$  threads. Assume sequential consistency.

```
volatile int level[n] = {0,0, ..., 0};
```

```
volatile int victim[n];
```

```
void lock(){  
    for (int l=1; l<n; l++){  
        level[tid] = l;  
        victim[l] = tid;  
        while (( $\exists k \neq tid$ ) (level[k] >= l && victim[l] == tid)) {};  
    }  
}
```

```
void unlock() { level[tid] = 0; }
```

## Measure Cache Misses

Consider the false sharing benchmark of Assignment n.2. Verify that the increase in time you observe when increasing the number of threads actually depends on the number of cache misses. This can be showed by instrumenting the benchmark with the PAPI library, that allows you to read the performance counters provided by your system.

*Hint:* Read the PAPI\_L1\_DCM counter inside each thread (i.e., inside the OMP parallel section) and report the maximum among the number of cache misses observed by each thread, as it is already done for the time.