Assignment 18

State the filter algorithm. List its properties and prove each them.

SHARED:

Array level[n] = {0}
Array TTW[n] = doesn't matter

Code for process i

```
for (L = 1 to n) {
        level[i] = L
        TTW[L] = i
        while (exist k != i: level[k] >= L AND TTW[L] == i) {}
}
<CS>
level[i] = 0
```

Mutual exclusion:

Assume by contradiction that the filter algorithm doesn't provide mutual exclusion. We can said that processes i and j such that i<j can enter in the CS at the same time.
Assuming the process i is in the CS, we have the level[i] = n. Thus, when j want to enter in the CS, he first have to pass all the level. But, recall level[i] = n, when j enter in the while loop, he will be blocked here since exist k != j (which is i) and yes level[k] >= L since his level is n. Adding TTW[L] = j since the process j state it just before to enter in the while loop. Then, there is no way for j to enter in the CS when I already here, contradicting our assumption and proving the claim.

■

Starvation free:

Assuming for the sake of the contradiction that a process k is stuck forever in the entry code. By the assumption that k is stuck in the entry code, we have at least level[k] = 1 and TTW[1] = k such that by examining the code:
Write$_k$(level[k]=1) $\rightarrow$ Write$_k$(TTW[1]=k) $\rightarrow$ Read$_k$(exist j != k: level[j] >= 1 AND TTW[1] == k)
Then, let's now show that level[j] will obviously be 0 in a finite time.
We doing this by assuming that exist at most n processes interested. First, let's focus by the process which arrived first (process 0). Then, since he arrived first,

he will not be block in the while loop since there is no k such that level[k] >= L. And then, this process will enter in the CS. Since we cannot assume that a process is block in the CS, the process will enter in the exit code and turn level[0] to 0. By doing this the second process which arrived after the process 0 will enter in the CS before everyone else since he will gets running time (by examination of the code and the level of the at his maximum). Repeating this until process k, we are now sure that k will enter in the CS contradicting that he will stuck forever in the entry code.

■