

# 1 Lezione del 22-10-25

## 1.1 Mutua esclusione

Analizziamo il problema della mutua esclusione studiando il seguente pseudocodice C, implementante una semplice struttura *stack*:

```
1 T stack[n];
2 int top = -1;
3
4 // inserisci in cima
5 void insert(T y) {
6     top++;
7     stack[top] = y;
8 }
9
10 // estrai dalla cima
11 T extract() {
12     T temp = stack[top];
13     top--;
14     return temp;
15 }
```

Se le variabili *stack* e *top* si trovano in memoria condivisa, o in altre la struttura *stack* creata si trova in memoria condivisa, potremmo incorrere in situazioni dove più operazioni sullo *stack* vengono iniziate contemporaneamente (le funzioni che operano sullo *stack* vengono chiamate contemporaneamente), e lo scheduler interlaccia le operazioni in un modo che rende lo *stack* inconsistente.

Ad esempio, con 2 processi  $p_1$  e  $p_2$ , il primo chiamante *insert()* e il secondo chiamante *extract()* potremmo avere la timeline di esecuzione:

```
1 t0: top++;           // p1
2 t1: temp = stack[top]; // p2
3 t3: top--;           // p2
4 t4: stack[top] = y;   // p1
```

con conseguenze chiaramente disastrose!