**What is a race condition?**

Race means that 2 processes are running parallel, which could cause trouble with the program.

**Disabling Interrupts**

1. Because context switches!
2. Long interrupt possible -> because the process could take over!

**Peterson’s Solution**

Process 0: wants to go into the critical region

Saves the other process in a variable, sets itself as interested, points itself as the loser but continues cause the other one isn’t interested yet

***Meanwhile*** Process 1: wants to go into the critical region

Saves the other process in a variable, overrides loser and stops cause the other one is interested

Process 0: Leaves critical region and sets his self as not interested   
 anymore after doing his processes

Process 1: sees that the other one isn’t interested anymore and can continue to   
 the end now until it leaves the critical region

**strict alternation**

Process 0: the last one becomes the loser because he has to wait for the other process

int i;

int j = 0;

void enter\_region(int process)

{

j++;

loser = process;

while (i == process && j >= 1) ;

}