1. Racecondition bedeutet das Scenario, dass 2 Prozesse gleichzeitig laufen und gleichzeitig auf einen Wert zugreifen. Zb 2 Programme, die beide den Wert einer Variable um 1 erhöhen laufen genau gleichzeitig ab; Startwert 0. Beide lesen gleichzeitig den Wert 0 ein, erhöhen ihn um 1 und speichern diesen wieder. Eigentlich sollte es so ablaufen: Programm 1 liest 0 ein, erhöht den Wert um 1. Erst danach liest Programm 2 den Wert ein – sprich 1, erhöht ihn um 1 und speichert ihn -> also ist 2 das eigentliche – und gewünschte Ergebnis.
2. You cant achieve Mutual Exclusion via disabling interrupts on a multi-core machine because on every core there runs one of the 2 prozesses simultanes. If you disable interrups, nothing will change.
3. If you give user processes the power to disable interrupts, one prozess may get stuck on ist own without enableing interrupts again. Your machine is now successfully dead.
4. So, prozess 1 calls enter\_\_region so he set interested[0] to true. Then the second prozess sets interested[1] to true, but gets stuck in the while because the loser is this prozess and the other prozess is still interested. Then the first prozess calls leave region and sets interested[0] to false. The second prozess will not be able to enter the critical region.
5. Both prozesses enter the critcal region. Interested on both is set to true. Lets say the first is a liiiitle slower at the while than the second, so the looser will be prozess 0 ( the first one ). He will get stuck in the while loop cuz the other prozess is still interested and the looser is himself. The second will pass the while cuz he is not the looser and enters crit region.
6. The strict alternation will fail if only 1 prozess exists and enters the crit zone twice and waits for the second non-existing prozess to switch the „run“ variable.
7. Im not sure which variable you ment by \tt turn but i guess looser. This var indicates which prozess is the looser so he gets stuck in the while and has to wait until the other prozess left the crit region.
8. Siehe file