# Aufgabe 5.1)

a)

Wird P0 zuerst gescheduled, so wird er nicht in die innere while-Schleife kommen, da flag[1] false ist. Wird direkt danach P1 gescheduled, so kommt auch dieser nicht in die innere while-Schleife, da P0 sein Flag noch nicht umgesetzt hat. Folglich können beide gleichzeitig in der criticalSection operieren.

* Verletzung der Mutual Exclusion

b)

P0 arbeitet zuerst seinen Programmcode ab und setzt sein flag[0] auf TRUE. Direkt danach wird P1 gescheduled und arbeitet seinen Code ab. Allerdings kommt er nicht mehr über die innere while-Schleife hinaus, da ja schon P0 sein Flag auf TRUE gesetzt hat. Anschließend sind also beide Flags auf TRUE und beide Prozesse bleiben in der inneren while Schleife hängen.

* Verletzung von Progress

c)

P0 läuft zuerst durch, da turn global auf 0 gestetzt wird und somit P1 nicht die innere while Schleife verlassen kann. Erst nachdem P0 turn auf 1 gesetzt hat nach der criticalSection, kann P1 in diesen Bereich eintreten. Anschließend muss P0 in der inneren while Schleife darauf warten, dass P1 die criticalSection verlässt und turn wieder auf 0 setzt. Hier entsteht aber das Problem, dass P0 bzw P1 unnötigerweise auf den anderen Prozess warten muss, falls der eine zunächst gar nicht in die criticalSection eintreten möchte, dieser aber an der Reihe ist.

* Verletzung von Bounded Waiting

# Aufgabe 5.2)

Es ergibt sich ein Problem, wenn zwei oder mehr Reporter warten, da so bei ungünstigem Scheduling (Der erste Reporter nimmt sich den ersten Politiker, der ankommt und der zweite den nächsten usw.) die Wartezeit länger als nötig sein könnte.

Zur Lösung kann man einen weiteren Mutex (hier: wartend) einfügen, der den Wartevorgang nur für einen Reporter zugänglich macht. Dadurch wird verhindert, dass eine unnötig lange Wartezeit entsteht.

void politicianArrives () {

signal ( politicianReady );

wait ( reporterReady );

}

void reporterArrives () {

mutex\_lock ( wartend )

wait ( politicianReady );

wait ( politicianReady );

mutex\_unlock ( wartend );

EnterElevator ();

signal ( reporterReady );

signal ( reporterReady );

}

# Aufgabe 5.3)

b)

1) Es ist nicht sichergestellt, dass der erste Wagen der ankommt auch der erste Wagen ist, der von einem Besucher bestiegen wird.

2)Es kann passieren, dass mehr als zwei Besucher den Wagen betreten. Dies ist der Fall, da zuerst der Wagen von einem Besucher betreten wird, bevor die Anzahl der freien Sitze reduziert wird.

d)

Nein, dies geht nicht mit Semaphoren, da Semaphore immer auf etwas bestimmtes warten. Entweder wartet man dann auf einen VIP oder einen Besucher. Wenn nun aber kein VIP kommt, dann wird solange gewartet, bis einer kommt.