

Klausur WS06/07 – Nebenläufigkeit (15 Punkte = 15min)

Prozess: tankendes Auto

fahreInWartebereich();

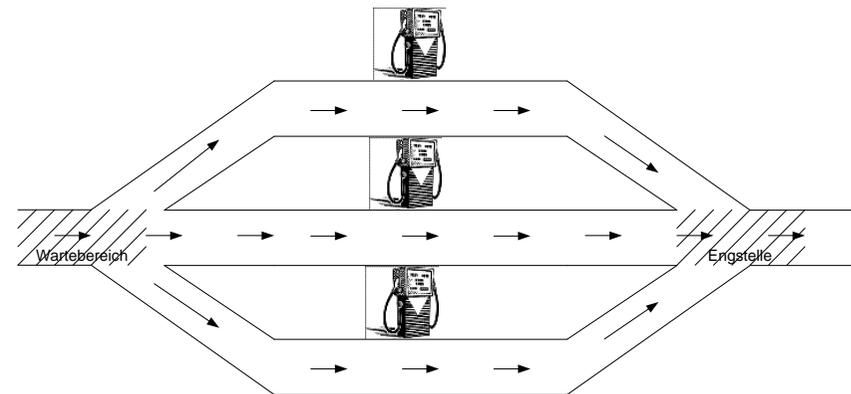
fahreAnZapfsaeule();

tanke();

bezahle();

fahreInEngstelle2();

verlasseEngstelle2();



- Geben Sie die notwendigen Semaphore (mitsamt Initialisierung) an, um das gegebene Problem zu lösen. Beispiel: `semAuto(1)` würde bedeuten, Sie verwenden einen Semaphor `semAuto`, der mit 1 initialisiert ist.
- Ergänzen Sie den folgenden Autoprozess mit passenden `up()` und `down()`-Methoden, um Kollisionen zu vermeiden. Achten Sie darauf, dass es zu keiner Verklemmung kommt. **Anmerkung:** Es muss nicht an jeder freien Stelle Code eingefügt werden. Beispiel: `1: down(semaAuto); up(semaAuto);` bedeutet das Einfügen der beiden Operationen in Zeile 1.

Klausur WS06/07 - Nebenläufigkeit

- c) Aufgrund einer Baustelle ist die Ausfahrt blockiert (siehe Abbildung), so dass die Wartebereich sowohl zur Einfahrt, als auch zur Ausfahrt genutzt werden muss. Ergeben sich notwendige Änderungen im Vergleich zur Lösung der Aufgabe b) und wenn ja welche?

Prozess: *tankendes Auto*

fahreInWartebereich();

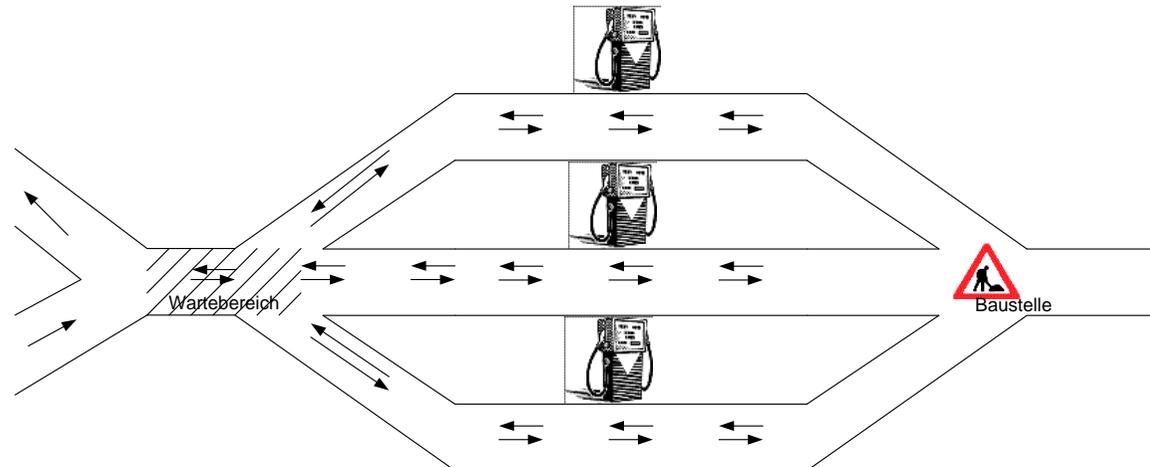
fahreAnZapfsaeule();

tanke();

bezahle();

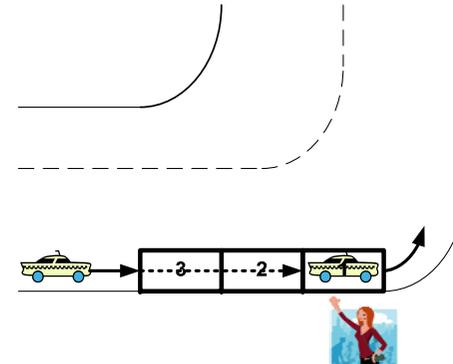
fahreInEngstelle2();

verlasseEngstelle2();



Klausur WS07/08 – Nebenläufigkeit (20 Punkte = 20min)

- Gegeben Sie folgendes Szenario: am Münchner Odeonsplatz gibt es eine Wartebucht für Taxis. Zur Vereinfachung gehen wir davon aus, dass die Wartebucht aus drei Plätzen besteht und immer nur ein Passagier gleichzeitig auf ein Taxi wartet. Passagiere steigen an der ersten Wartebucht ein, die Taxis rücken nach, sobald das Taxi vor ihnen losgefahren ist. Implementieren Sie nun schrittweise eine Prozesssynchronisation, so dass es zu keinen Auffahrunfällen kommt, die Taxis in der Ankunftsreihenfolge auch wieder losfahren, Taxis nur mit Passagier losfahren, Passagiere nicht aus Versehen ein nicht-existentes Taxi betreten und es zu keinen Verklemmungen kommt.



- Notieren Sie die wichtigen Programmabschnitte des Taxiprozesses und des Passagierprozesses. Lassen Sie genügend Platz für spätere Synchronisationsoperationen.
Beispiel: `fahreInErsteWartebucht()` ;
- Geben Sie die zur Synchronisation der Taxis und Passagiere benötigten Semaphore, sowie der Initialwerte an. Gehen Sie dabei davon aus, dass zu Beginn kein Taxi in der Wartebucht und keine wartenden Passagiere vorhanden sind.
Beispiel: `semTaxi(1)` würde bedeuten, Sie verwenden einen Semaphor `semTaxi`, der mit 1 initialisiert ist.
`int i=0`; wenn sie eine ganzzahlige Variable mit Initialisierungswert 1 benutzen wollen.
- Ergänzen Sie den Taxiprozess und Passagierprozess mit passenden `up()` und `down()`-Methoden, um die Aufgabenstellung zu erfüllen.
Beispiel: `down(semTaxi)` ; bedeutet das Anfordern des Semaphors `semTaxi`
Beispiel: `up(semTaxi)` ; bedeutet das Freigeben des Semaphors `semTaxi`
- Der Wartebereich am Odeonsplatz ist begrenzt. Stellen Sie sicher, dass maximal 3 Taxis auf Fahrgäste warten und kein Rückstau entsteht. Die Überprüfung ob der Wartebereich belegt ist, soll dabei so schnell wie möglich erfolgen um den Straßenverkehr nicht zu behindern. Andererseits, sollen die Taxifahrer auf jeden Fall in den letzten Wartepplatz fahren, falls dieser frei ist.