

Systemprogrammierung

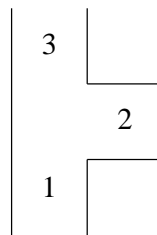
WS 2004/2005

Übungsblatt 6

Abgabe der Lösungen: ab 29.11.2004 in den Übungen

Aufgabe 1 (6 Punkte): Semaphore

In einer 30 km/h Zone befindet sich eine Kreuzung. Die Kreuzung sieht wie folgt aus:



In dieser Zone gilt die Vorfahrtsregel “Rechts vor Links”. Autos erscheinen vollkommen zufällig aus einer der drei Straßenrichtungen. An der Kreuzung ist die Vorfahrtsregel einzuhalten.

Schreiben sie eine Funktion `auto`, die das passieren eines Autos an dieser Kreuzung repräsentiert. Die Funktion wird als eigener Thread ausgeführt, und läuft solange, bis das Auto die Kreuzung überquert hat. Falls ein Auto wegen der “Rechts vor Links”-Regel nicht in die Kreuzung einfahren kann, geht es in einen Wartezustand über. Erst wenn alle Autos, die von rechts kommen, die Kreuzung überquert haben, fährt das Auto los.

Die Funktion `fahren()` ist vordefiniert. In ihr findet der simulierte Fahrvorgang des Autos statt. Weiterhin ist ein Array namens `int nr_wait[]` vorgegeben. In diesem Array können Sie die Anzahl der wartenden Autos je nach Straßenrichtung speichern. Die wartenden Autos werden mit einem Semaphoren-Array `semaphore waiting[]` verwaltet. Ein neu entstandenes Auto wird in diesem Array je nach Straßenrichtung gespeichert und fährt dann los, wenn kein Auto in der rechts liegenden Straße wartet und kein Auto die Kreuzung gerade überquert. Das Array `nr_wait` wird so adressiert dass `nr_wait[1]` die wartenden Autos aus der Straße 1 speichert. Genauso soll auch das Semaphoren-Array `semaphore waiting[]` adressiert werden. Benutzen Sie weitere Semaphore je nach Notwendigkeit.

Es kann immer nur ein Fahrzeug die Kreuzungsmittte zur Zeit überqueren. Dies ist auch von dem Programm zu beachten. Die Vorfahrt wird nur nach der Regel “Rechts vor Links” gesteuert.

Hinweis: Gehen Sie davon aus, dass stets Semaphore mit assoziierten Warteschlangen verwendet werden. Die Funktionen `signal(s)` und `wait(s)` stehen Ihnen zur Verfügung.

Aufgabe 2 (6 Punkte): Semaphore

In einem nicht näher spezifizierten Urwald gibt es eine Hängebrücke über einen Fluss. Die Brücke ist so schmal, dass keine zwei Personen aneinander vorbei gehen können. Die Brücke ist die einzige Verbindung zwischen den beiden Dörfern Obawe (O) und Metani (M). Früher war das Verkehrsaufkommen zwischen den beiden Dörfern so gering, dass man einfach die Brücke betreten konnte. Falls jemand entgegen kam, so kehrte man um, wenn man weniger als die Hälfte der Brücke passiert hatte. Inzwischen ist dies aber nicht mehr möglich, weil zu viele Personen zwischen Obawe und Metani hin und her pendeln. Versuchen Sie, das Betreten der Brücke mit den in der Vorlesung kennengelernten Hilfsmitteln zu koordinieren. Beachten Sie, dass sich mehrere Personen auf der Brücke befinden dürfen. Beschreiben Sie zuerst die Brücke mit Hilfe geeigneter Variablen und Semaphoren. Personen, die die Brücke überqueren wollen, rufen die Funktion `überquereBrücke (int Richtung)` auf. `Richtung` gibt die beabsichtigte Richtung der Personen (0 oder 1) an. Mittels der Funktion `gehen()` passieren Sie die Brücke. Schreiben Sie die Funktion `überquereBrücke(int Richtung)`, wobei Sie die Lösung ausführlich kommentieren.

Aufgabe 3 (8 Punkte): Petri-Netze

Ein Getränke-Automat verkauft Getränkedosen der Sorten Cola und Wasser, die einen einheitlichen Preis haben. Das Gerät zeigt an, welche der zwei Magazine noch Waren haben. Sofern genügend bezahlt worden ist, kann durch Druck auf einen von zwei Knöpfen eine Dose aus dem entsprechenden Magazin ausgegeben werden. Beachten Sie insbesondere die notwendigen Tätigkeiten, die Magazine gefüllt zu halten, da jedes Magazin nur einen beschränkten Vorrat an Dosen aufnehmen kann. Während des Nachfüllens kann natürlich keine Dose verkauft werden. Das Herausgeben von Wechselgeld wird nicht betrachtet. Erlaubt sind Münzen der Werte 50 Cent, 1 Euro und 2 Euro. Eine Dose kostet genau 2,00 Euro.

Spezifizieren Sie mit einem Petri-Netz die Abläufe im Automaten. Beschriften Sie Stellen und Transitionen mit sinnvollen Bezeichnern und erläutern Sie Ihr Modell. Nennen Sie explizit, welche Annahmen Sie für Ihr Modell getroffen haben. Gehen Sie davon aus, dass jede Transition schaltet, sobald sie schalten kann.