VO 182.022 27. Oktober 2010

Prüfung Betriebssysteme

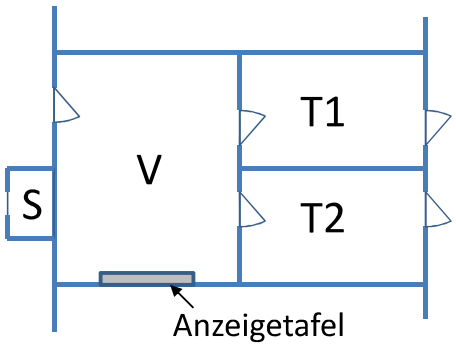
KNr. MNr. Zuname, Vorname

Ges.)(100) 1.)(30) 2.)(20) 3.)(50) Zusatzblätter:

Bitte verwenden Sie nur dokumentenechtes Schreibmaterial!

1 Synchronisation mit Semphoren (30)

In einem Fitnesscenter kann man seine Fitness testen lassen. Dazu gibt es ein Testzentrum mit folgenden Einrichtungen (siehe Abbildung):



- einen Schalter (S), bei dem man sich ein Anmeldeformular abholt bzw. das ausgefüllte Formulare abgibt und damit auch eine Identifikationsnummer (ID) für den Test bekommt. Beim Schalter darf sich immer nur maximal eine Person aufhalten.
- einen Vorbereitungsraum (V), in dem sich maximal 6 Personen aufhalten dürfen, um sich auf den Test vorzubereiten.
- zwei Testräume (T1 und T2), in denen sich jeweils maximal eine Person aufhalten darf, um den Fitnesstest zu absolvieren.
- Weiters gibt es eine Anzeigetafel, die die IDs der beiden Testteilnehmer anzeigt, die sich als nächstes für das Eintreten in einen der beiden Testräume bereithalten sollen. Diese Reihenfolge kann von der Reihenfolge der Anmeldung beim Schalter S abweichen. Die Aktualiserung der Anzeigetafel erfolgt mit folgendem Codestück:

```
P(Anzeige)
update_Anzeige()
V(Anzeige)
```

Schreiben Sie einen Prozess für einen Sportler *Sp*, der unter Einhaltung der obigen Einschränkungen einen Fitnesstest durchführt. Verwenden Sie *Semaphore* zur Synchronisation und vermeiden Sie unnötige Einschränkungen der Parallelität mehrerer Sportler. *Sp* führt folgende Schritte durch:

- *Sp* holt sich mit der Funktion `get_form()` ein Anmeldeformular vom Schalter, füllt dieses aus (`fill_form()`), gibt das Forumlar ab und bekommt eine ID (Der Aufruf der Funktion `submit_form()` dient zum Abgeben des Formulars und retourniert die ID).

- *Sp* tritt in den Vorbereitungsraum ein (`enter_V()`) und ruft abwechselnd `prepare()` und `check_board()` auf, die die Vorbereitung auf den Test bzw. das Ablesen der Anzeigetafel realisieren. An `check_board()` übergibt der Prozess die ID von *Sp*. Der Returnwert der Funktion gibt an, ob und für welchen Raum die Nummer von *Sp* auf der Anzeigetafel steht: (Wert 0: ID steht nicht auf der Anzeigetafel, Wert 1: *Sp* soll sich für das Eintreten in T1 bereithalten, Wert 2: *Sp* soll sich für das Eintreten in T2 bereithalten).
  Achten Sie darauf, dass Ihre Lösung das gleichzeitige Ablesen der Tafel durch mehrere Sportler erlaubt.

- *Sp* wartet bis der ihm zugewiesene Testraum frei wird, betritt dann den Testraum (Aufruf von `enter_T()` mit T1 oder T2 als Parameter), führt den Test durch (Aufruf von `fitness_test()`), und verlässt den Testraum durch die Ausgangstür, am Plan rechts (Aufruf von `exit_T()`).

Lösen Sie die Synchronisationsaufgabe mit *Semaphoren* und geben Sie Initialisierungen für die Semaphore an. Achten Sie bei der Implementierung von *Sp* darauf, dass die Lösung die Ausführung und Synchronisation einer "beliebigen" Anzahl von Sportlern erlaubt.

(a) Initialisierungen

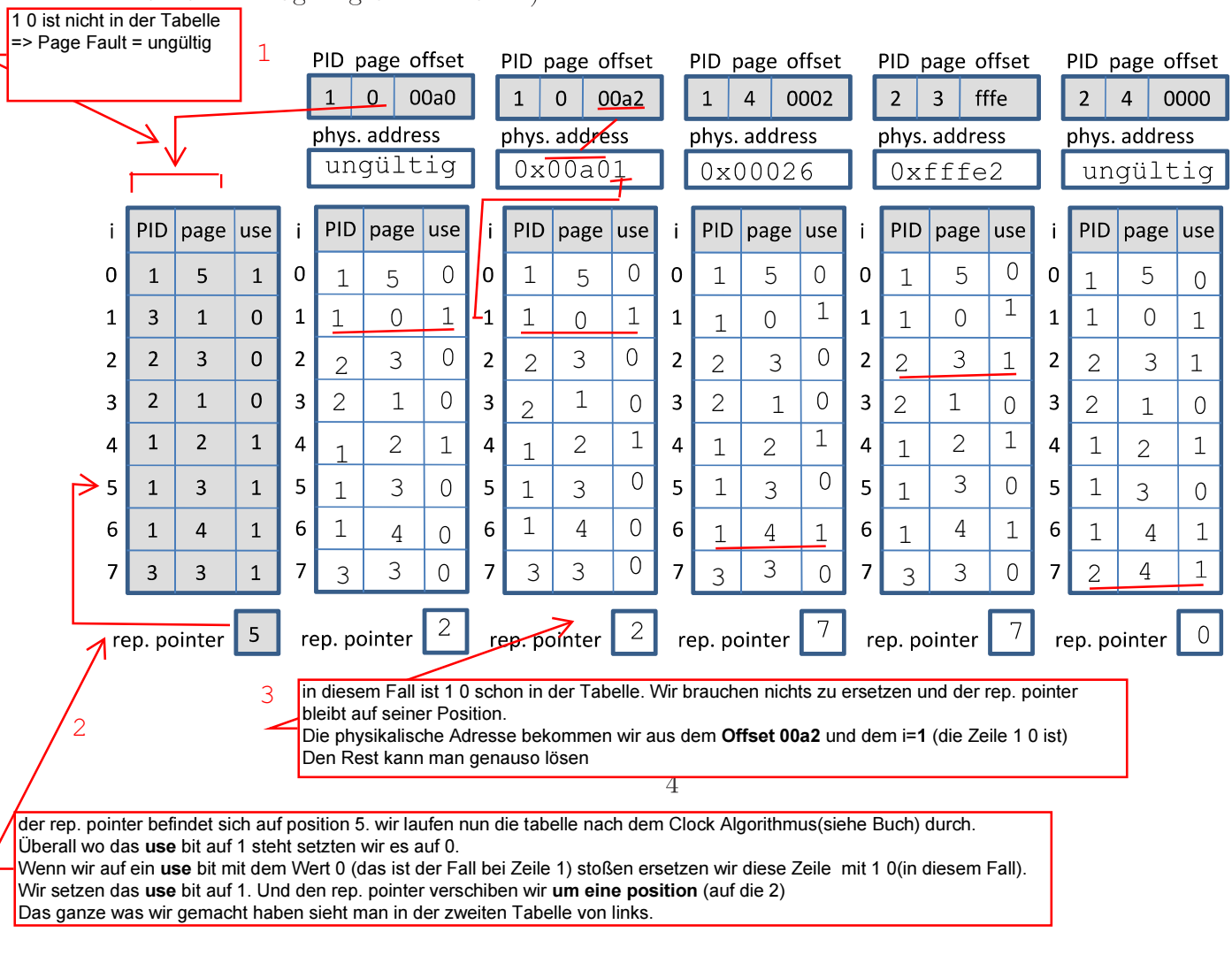
(b) Code für *Sp*

2 Memory Management (20)

Was ist eine Inverted Page Table?

Die Page Table eines kleinen Computersystems ist als Inverted Page Table (IPT) mit acht Einträgen realisiert. Beim Auftreten von Page Faults wird in diesem Computersystem der Clock Algorithmus als globale Page Replacement Strategie eingesetzt.

Im zu lösenden Beispiel ist eine Ausgangsbelegung der Page Table (links), sowie eine Folge von Zugriffen auf den virtuellen Speicher, charakterisiert durch Prozessnummer (PID), adressierte Seite (page) und Offset, gegeben. Simulieren Sie die Ausführung der Zugriffsfolge von links nach rechts und tragen Sie für jeden Zugriff auf den virtuellen Speicher (a) die entsprechende physikalische Adresse, (b) den Zustand der IPT und (c) den Wert des Replacement Pointers (rep. pointer) für den Clock Algorithmus nach dem Speicherzugriff an (Sie brauchen bei jedem Schritt nur die Werte in der IPT anzugeben, die sich beim aktuellen Zugriff geändert haben).



3 Fragen zu Betriebssystemen (50)

Zeichnen Sie das Zustandsdiagramm eines Prozesses, beginnend mit der Entstehung bis zur Beendigung des Prozesses. Geben Sie die Namen der Zustände an und beschreiben Sie kurz jeden Zustand und Zustandsübergang. (5)

Welche Strategien zur Vorbeugung gegen Deadlocks bzw. zur Vermeidung von Deadlocks gibt es? Beschreiben Sie diese kurz. (5)

Bei der Deadlock-Vermeidung spricht man von einem *Safe State* bzw. einem *Unsafe State*. Erklären Sie die Bedeutung dieser Begriffe. (4)

Erklären Sie den Begriff des *Multithreading*. Geben Sie Probleme bei der Verwendung von Threads an. (3)

Nennen Sie die Arten von Optimierungszielen, die ein Scheduler beim Prozess-Scheduling verfolgen kann und geben Sie jeweils zwei Beispiele an. (4)

Was versteht man unter einem *Monitor* zur Prozesssynchronisation? Nennen Sie die wichtigsten Komponenten und Eigenschaften eines Monitors. (4)

Was versteht man unter dem Begriff *Relocation*? (2)

Beim Paging ist ein optimaler Seitenersetzungsalgorithmus bekannt. Beschreiben Sie diesen und geben Sie an, warum er in der Praxis nicht verwendet wird. (3)

Welche Möglichkeiten kennen Sie, mit denen in Paging-Systemen Speicherschutz realisiert werden kann? (4)

Was versteht man unter der Working-Set Strategie? Beschreiben Sie deren Funktionsweise und wie diese Strategie zur Optimierung eines Paging-Systems eingesetzt werden kann. (5)

Nennen Sie Möglichkeiten, wie die Blockbelegung von Dateien auf einer Festplatte repräsentiert werden kann. Geben Sie Vor- und Nachteile der angegebenen Organisationsformen an. (4)

Erklären Sie die Begriffe *absoluter Pfadname* und *relativer Pfadname* und geben Sie jeweils ein Beispiel an. (4)

Die Implementierung einer Zugriffsmatrix kann in der Form von *Access Control Lists* oder *Capability Lists* erfolgen. Erklären Sie diese Begriffe. (4)