VO 182.022 24. Januar 2011

Prüfung Betriebssysteme

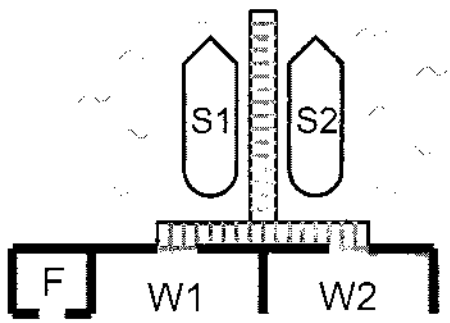
KNr. MNr. Zuname, Vorname

Ges.)(100) 1.)(30) 2.)(25) 3.)(45) Zusatzblätter:

Bitte verwenden Sie nur dokumentenechtes Schreibmaterial!

1 Synchronisation mit Semphoren (30)

Sie sollen einen Softwareentwickler beim Schreiben einer Simulationssoftware für eine Schiffsanlegestelle (siehe Abbildung) unterstützen. Diese Software simuliert die Aktionen von Schiffen und Passagieren durch Prozesse, die mit Semaphoren synchronisiert werden.



An der Schiffsanlegestelle gibt es Anlegeplätze für zwei Schiffe, S1 und S2, zwei Wartezonen, W1 und W2, die jeweils einem Schiff zugeordnet sind, und einen Fahrkartenschalter F.

Personen, die mit einem der Schiffe, S1 oder S2, verreisen, lösen am Schalter F eine Fahrkarte für ihr Schiff und gehen dann in die entsprechende Wartezone. Dort warten sie, bis sie an die Reihe kommen und gehen dann über den Steg auf ihr Schiff, um ihre Schiffsreise anzutreten.

Schiffe kommen jeweils leer am Steg an, warten dann, bis die erlaubte Anzahl von Passagieren an Bord ist, und verlassen dann den Steg wieder.

Ergänzen Sie in den Codestücken die fehlenden Semaphoroperationen, um die Prozesse unter folgenden Rahmenbedingungen zu synchronisieren.

- Ein Schiff kann N Passagiere aufnehmen. Schiffe fahren leer zur Anlegestelle, nehmen dort N Passagiere auf und fahren dann wieder ab.
- Für die Durchgänge von den Warteräumen zum Steg gilt, dass jeder Durchgang von maximal zwei Personen gleichzeitig passiert werden kann.
- Die Einstiegstelle der Schiffe ist sehr schmal. Passagiere können die Schiffe daher nur einzeln besteigen.

- Aus Sicherheitsgründen dürfen sich zu jedem Zeitpunkt maximal K Personen auf dem Steg befinden.
- Beim Anlegen bzw. Ablegen eines Schiffes dürfen sich keine Passagiere am Steg befinden.
- Schiffe und Passagiere sollen unter Einhaltung der angegebenen Regeln möglichst unabhängig und möglichst ohne Einschränkungen der Parallelität agieren können.

(a) Initialisierungen (vor Start der Prozesse)

(b) Prozesse für Schiffe

Bedeutung der verwendeten Funktionen:

`anlegen(S)`	Schiff S legt am Steg an
`abfahren(S)`	Schiff S legt vom Steg ab

```
/** Schiff S1 **/

anlegen(S1)

abfahren(S1)
```

```
/** Schiff S2 **/

anlegen(S2)

abfahren(S2)
```

(c) Prozesse für Passagiere

Bedeutung der verwendeten Funktionen:

`fahrkarte_kaufen()`	der Passagier kauft eine Fahrkarte
`warteraum_betreten(W)`	der Passagier betritt den Warteraum W
`steg_betreten()`	der Passagier betritt und geht auf dem Steg
`schiff_betreten(S)`	der Passagier betritt das Schiff S

```
/**  Passagier fuer S1  **/
fahrkarte_kaufen()
warteraum_betreten(W1)
steg_betreten()
schiff_betreten(S1)
```

```
/**  Passagier fuer S2  **/
fahrkarte_kaufen()
warteraum_betreten(W2)
steg_betreten()
schiff_betreten(S2)
```

2 Page Replacement (25)

Gebeben ist ein Arbeitsspeicher mit vier Frames, dessen Seiten mit unterschiedlichen Ersetzungsstrategien ersetzt werden sollen: mit OPT, LRU, bzw. mit dem Clock Algorithmus. Geben Sie in den Tabellen für jeden Algorithmus die Speicherinhalte für jeden Frame nach jedem Zugriff der angegebenen Seitenzugriffsfolge an. Die Seitenzugriffsfolge ist für alle Algorithmen gleich. Sie ist jeweils in der Kopfzeile der Tabelle gegeben. Geben Sie in den Spalten die Speicherbelegung nach dem entsprechenden Seitenzugriff an und kennzeichnen Sie in der letzten, mit *PF* markierten Zeile das Auftreten von Page Faults. Der Arbeitsspeicher ist am Beginn leer.

OPT-Strategie:

	A	B	C	D	E	F	C	D	E	C	F	G	H	E	D
0															
1															
2															
3															
PF															

LRU-Strategie:

	A	B	C	D	E	F	C	D	E	C	F	G	H	E	D
0															
1															
2															
3															
PF															

Clock-Algorithmus:

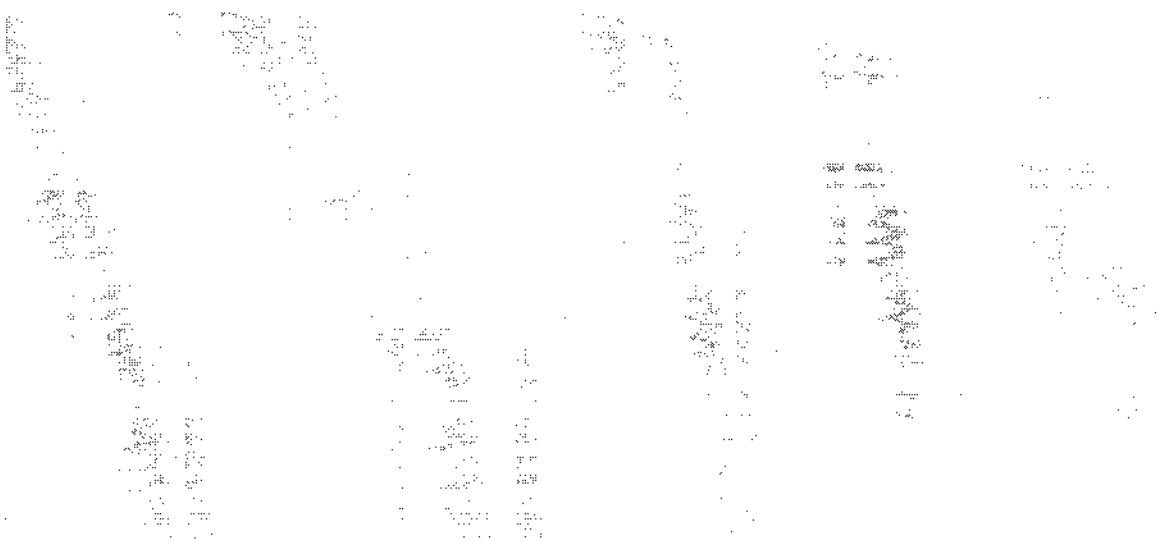
	A	B	C	D	E	F	C	D	E	C	F	G	H	E	D
0															
1															
2															
3															
PF															

Vergleichen und diskutieren Sie die Anzahl der bei den Seitenersetzungsstrategien beobachteten Page Faults.

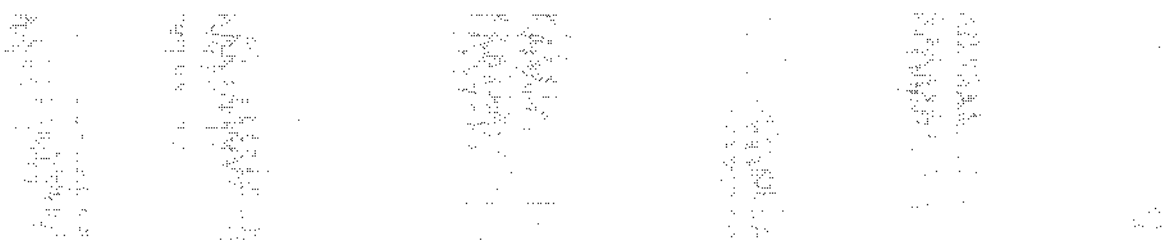
3 Fragen zu Betriebssystemen (45)

Erklären Sie die drei Begriffe *Deadlock*, *Lifelock* und *Starvation*. (5)

Worin liegt der grundlegende Unterschied zwischen *Prozessen* und *Threads*? Welcher Vorteil ergibt sich aus der Einführung von Threads für den Benutzer und worauf muss der Benutzer achten? (4)



Was versteht man unter *Virtual Memory Management*? Welche Vorteile bietet es? (5)



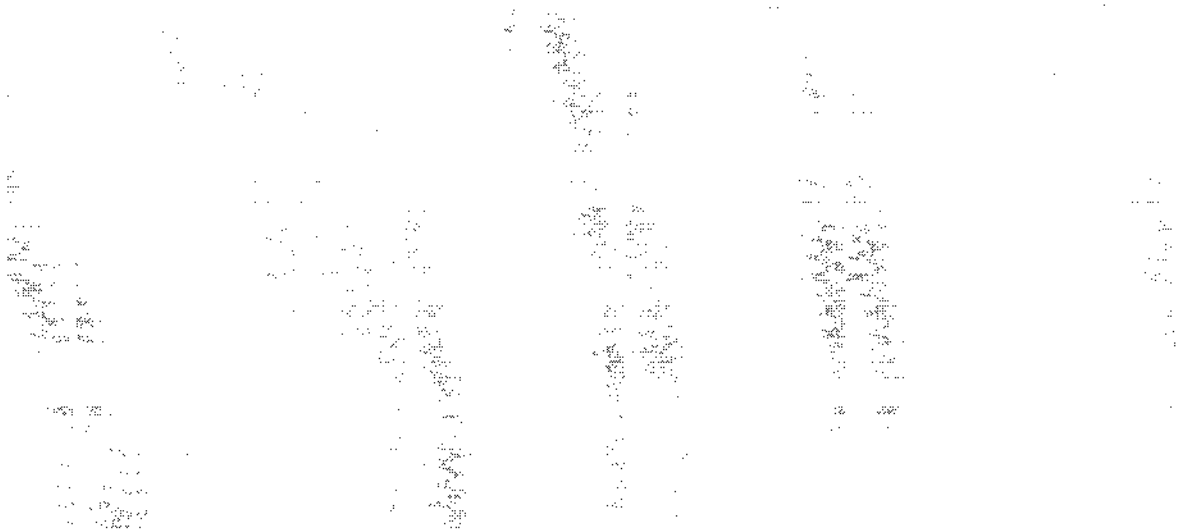
Beschreiben Sie Aufgabe und Funktion eines *Translation Lookaside Buffers*? Worauf hat man bei der Betriebssystemimplementierung bei einem Process Switch zu achten, wenn man einen Translation Lookaside Buffer verwendet? (4)

Was versteht man unter *Thrashing*? (2)

Wie funktioniert *Round Robin* Scheduling? Welchen wichtigen Parameter gibt es bei diesem Verfahren? Wie wird man diesen Parameter günstiger Weise wählen? (4)

Was versteht man unter *Buffering*? Welche Vorteile bietet es, wo liegen seine Grenzen und worauf hat man bei der Verwendung von Puffern bei der Betriebssystemimplementierung zu achten? (5)

Wie ist der Inhalt einer (mechanischen) Festplatte typischer Weise organisiert? (4)



Wie ist ein *i-node* aufgebaut? Welche Informationen enthält er? (4)

Was beschreibt das Modell von Bell und LaPadula? Geben Sie die vom Modell geforderten Eigenschaften an. (4)

Nennen Sie Design Prinzipien für die Konstruktion von sicheren Systemen. Geben Sie für jede Regel ein Beispiel an. (4)