

KNr.

MNr.

Zuname, Vorname

Ges.)(100)

1.)(35)

2.)(20)

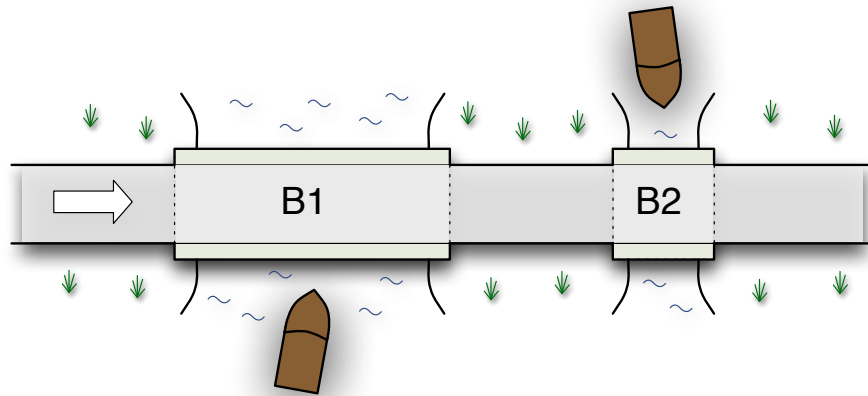
3.)(45)

Zusatzblätter:

Bitte verwenden Sie nur dokumentenechtes Schreibmaterial!

1 Synchronisation mit Semaphoren (35)

In einem System aus parallelen Prozessen soll die Funktion einer Hebebrückenanlage (siehe Skizze) simuliert werden. Eine Einbahnstraße für Autos kreuzt zwei Schiffahrtswege, wobei die Kreuzungen als Hebebrücken (B1 bzw. B2) ausgeführt sind.



Die unten angegebenen Prozesstemplates sind so mit Semaphoroperationen zu ergänzen, dass die parallele Exekution der Prozesse folgende Eigenschaften erfüllt:

- Im Prozesssystem wird jedes Auto und jedes Schiff durch einen Prozess realisiert. Es gibt drei verschiedene Prozesstemplates: A (für Autos), S1 (für Schiffe, die B1 passieren), und S2 (für Schiffe, die B2 passieren).
- Das System unterstützt beliebig viele aktive Kopien von A, S1 und S2.
- Auf keiner Kreuzung dürfen sich gleichzeitig Autos und Schiffe befinden.
- Auf der Kreuzung B1 dürfen maximal 2 Schiffe gleichzeitig passieren. Auf Kreuzung B2 darf sich zu jedem Zeitpunkt maximal 1 Schiff befinden.
- Die maximale Anzahl von Autos, die sich gleichzeitig auf den Brücken befinden dürfen, ist k für B1 und 1 für B2.

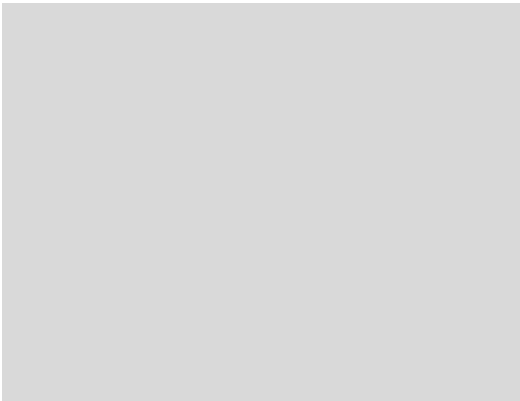
- Der Schiffsverkehr hat Vorrang gegenüber dem Autoverkehr, d.h., Schiffe dürfen nicht durch Autowarteschlangen, die sich vor den Brücken bilden, verzögert werden.
- Autos, die auf das Passieren von B2 warten, halten bereits vor B1, um B1 nicht zu blockieren (d.h., das Straßenstück zwischen B1 und B2 bleibt leer, wenn B2 von einem Schiff passiert wird).
- Das Öffnen und Schließen der Brücken wird zur Vereinfachung durch die Prozesse A, S1 und S2 gesteuert. Der Code dafür ist bereits in den Codefragmenten der Prozesse enthalten und braucht nicht mehr programmiert zu werden.

Code für Prozesse und Initialisierungen

```
/** Schiff S1 **/
```

```
void S1(void)
```

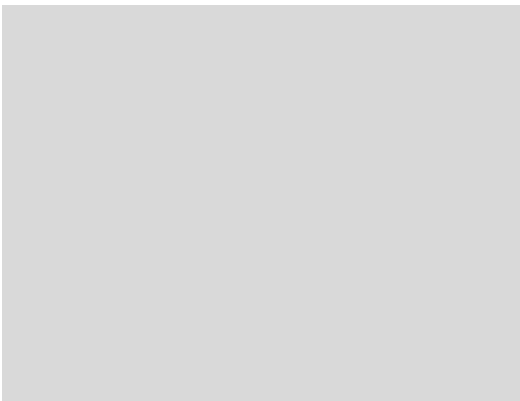
```
{
```



```
/* Schiff passiert B1 */
```

```
if (!B1_open) open_B1();
```

```
passiere_B1();
```

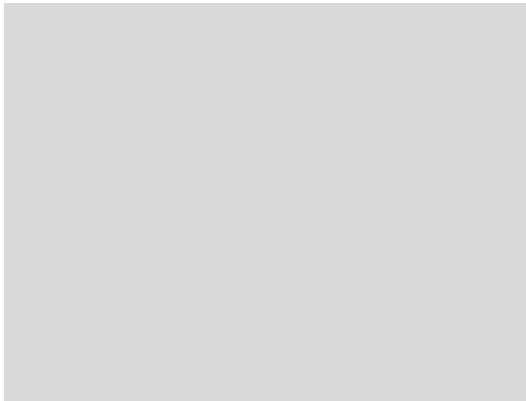


```
}
```

```
/** Schiff S2 **/
```

```
void S2(void)
```

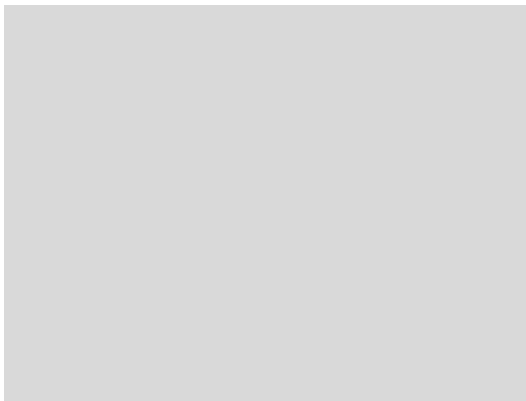
```
{
```



```
/* Schiff passiert B2 */
```

```
if (!B2_open) open_B2();
```

```
passiere_B2();
```

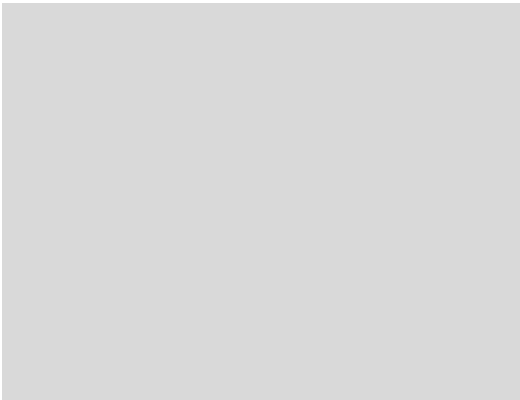


```
}
```

```
/** Auto A **/
```

```
void A(void)
```

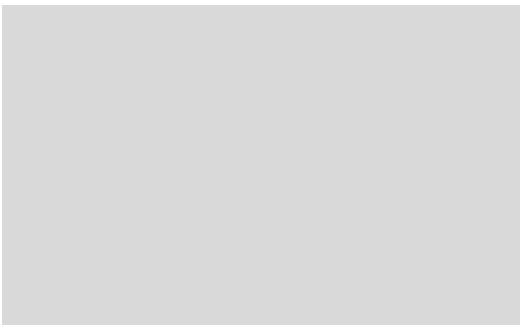
```
{
```



```
/* B1 passieren */
```

```
if (B1_open) close_B1();
```

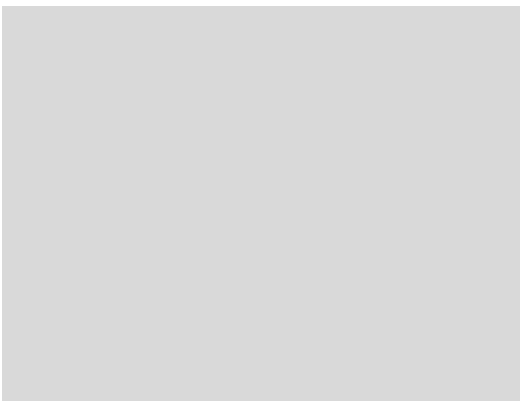
```
passiere_B1();
```



```
/* B2 passieren */
```

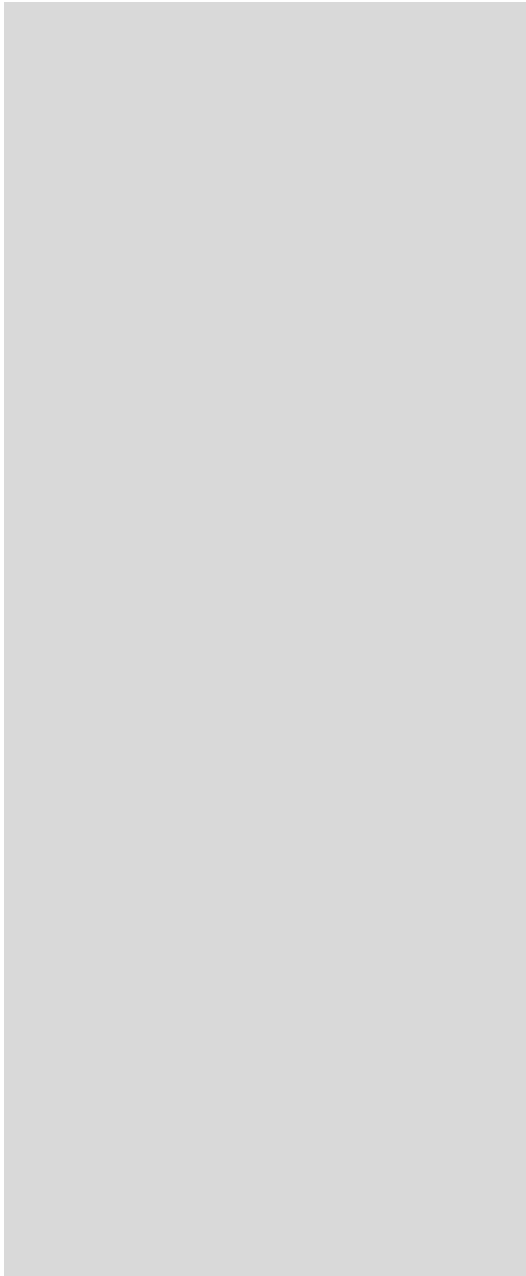
```
if (B2_open) close_B2();
```

```
passiere_B2();
```



```
}
```

```
/** Semaphorinitialisierungen **/
```



2 Memory Management – Working Set (20)

Gegeben ist eine Folge von Speicherreferenzen eines Prozesses. Die Entwicklung des *Working Sets* des Prozesses während der Ausführung soll für unterschiedliche *Window Sizes* (Δ) beobachtet werden. Tragen Sie dazu in der unten stehenden Tabelle für jeden Zeitpunkt das Working Set nach der Referenzierung der angegebenen Speicherseite an.

Referenzierte		Window Size, Δ		
log. Zeit	Seite	2	3	4
1	5			
2	6			
3	7			
4	6			
5	7			
6	8			
7	10			
8	11			
9	13			
10	10			
11	12			
12	13			
13	10			
14	14			
15	14			
16	14			
17	15			
18	14			
19	15			
20	16			

Nehmen Sie an, dass das Lokalitätsverhalten des obigen Prozesses typisch für die Prozesse auf einem Rechner ist. Diskutieren Sie anhand ihrer Tabellenwerte, welchen Wert für Δ Sie bei der Implementierung der Speicherverwaltung für diesen Rechner verwenden würden.

3 Fragen zu Betriebssystemen (45)

Was versteht man unter einem *Process Image*? Erklären Sie, aus welchen Teilen ein Process Image besteht? (4)

Für die Lösung des Problems des geregelten Eintritts in einen kritischen Abschnitt werden drei Eigenschaften gefordert. (a) Nennen Sie diese drei Eigenschaften und erklären Sie deren Bedeutung. (b) Wodurch werden die drei Eigenschaften gewährleistet, wenn Semaphore zum Schutz eines kritischen Abschnitts verwendet werden? (5)

Nennen Sie die Bedingungen für das Eintreten eines Deadlocks und erklären Sie diese. (5)

Bei welchen der folgenden *Scheduling*-Strategien kann es zur *Starvation* kommen: (a) FCFS, (b) Shortest Job First, (c) Round Robin, (d) Priority Scheduling? Begründen Sie jeweils Ihre Antwort. (4)


Was versteht man unter *interner Fragmentierung* und *externer Fragmentierung*? Beschreiben Sie die Begriffe und geben Sie je ein Beispiel an. (3)

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)

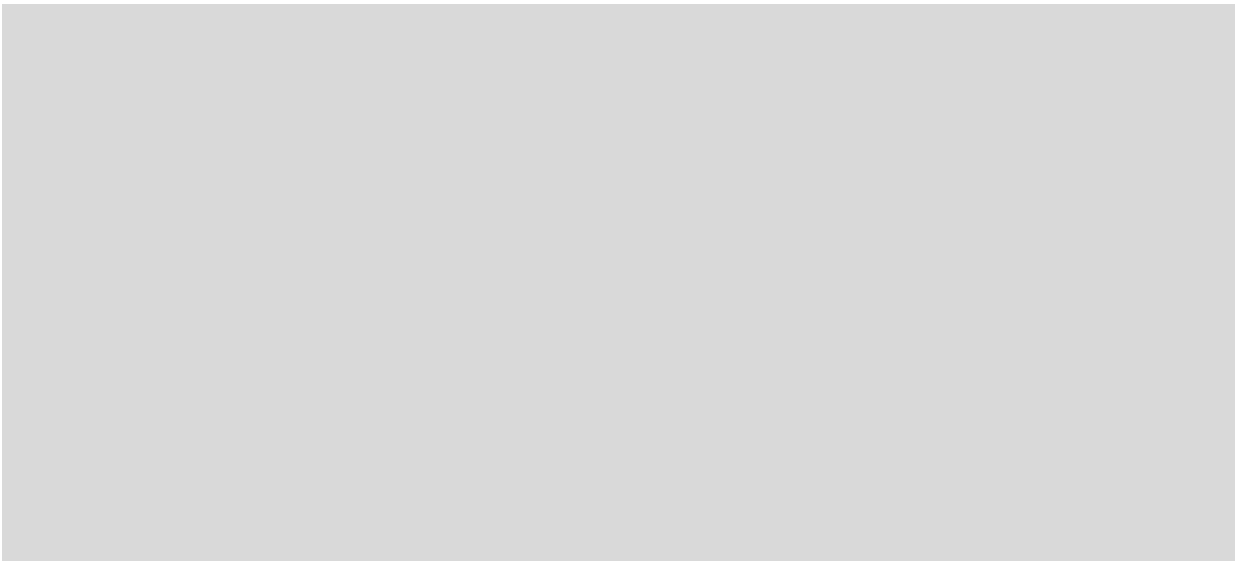
Was versteht man unter *Blocking* bzw. *Non-Blocking I/O*? Beschreiben Sie die beiden Arten, I/O-Operationen durchzuführen. (3)

Erklären Sie die Begriffe *Access Control List* und *Capability List*. Wozu und wie werden diese verwendet? (4)

Beschreiben Sie das typische Layout einer Disk bzw. eines Filesystems. Welche Rolle spielen die einzelnen Teile beim Hochfahren des Betriebssystems? (5)

A large, empty gray rectangular box intended for the user to write their answer to the first question.

Was versteht man unter dem *Domain Name System (DNS)*? Erklären Sie die Aufgabe von DNS und geben Sie ein Beispiel an, um seine Funktionsweise zu illustrieren. (4)

A large, empty gray rectangular box intended for the user to write their answer to the second question.

Beschreiben Sie drei *Naming Strategien*, die in *Verteilten Dateisystemen (DFS)* verwendet werden. (4)

