

KNr.

MNr.

Zuname, Vorname

Ges.) (100)

1.) (25)

2.) (25)

3.) (25)

4.) (25)

Zusatzblätter:

Bitte verwenden Sie nur dokumentenechtes Schreibmaterial!

1 Synchronisation (25)

Gegeben sei ein System mit einer beliebigen Anzahl von parallel laufenden Prozessen vom Typ A und vom Typ B. Weiters ist in dem System ein Gerät vom Typ G 4-fach vorhanden. Die zu produzierende und konsumierende Ressource vom Typ R ist initial 0-mal vorhanden. Ein Prozess vom Typ A benötigt für die Abarbeitung einer Iteration exklusiven Zugriff auf 3 Geräte vom Typ G und produziert dabei eine Ressource vom Typ R. Ein Prozess vom Typ B benötigt für die Abarbeitung exklusiven Zugriff auf ein Gerät vom Typ G und konsumiert zusätzlich zwei Ressourcen vom Typ R.

Synchronisieren Sie den Arbeitsablauf der Prozesse mit Semaphoren. Achten Sie auf Vermeidung von Deadlocks und ermöglichen sie maximale Parallelität. Verwenden Sie möglichst wenige Synchronisationskonstrukte. Die Verwendung von globalen Variablen ist verboten.

Verwenden Sie folgende Funktionen für Operationen auf Semaphoren:

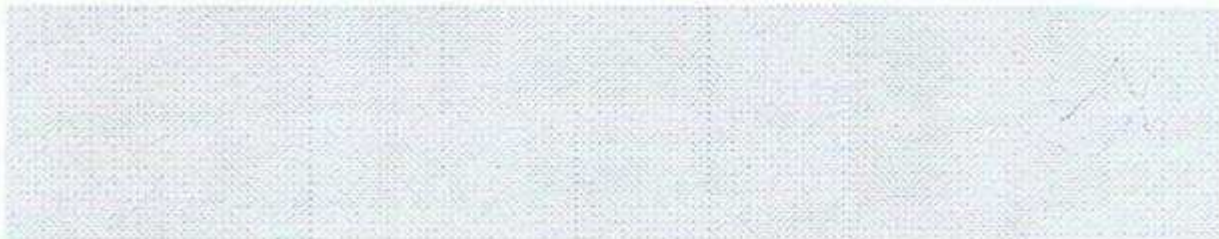
$\text{initS}(\text{Sem}, \text{init})$ legt einen Semaphor mit dem angegebenen symbolischen Namen *Sem* an und initialisiert ihn mit der Zahl *init*.

$\text{P}(\text{Sem})$ implementiert ein *wait* auf dem Semaphore, und

$\text{V}(\text{Sem})$ implementiert ein *signal* auf dem Semaphore.

a) Initialisierungen

Initialisieren Sie die notwendigen Semaphore.

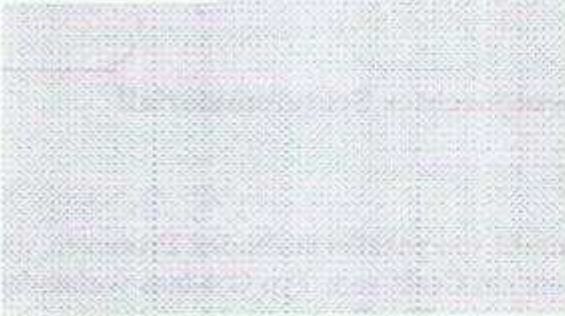


b) Entwerfen Sie die Prozesse vom Typ A und B

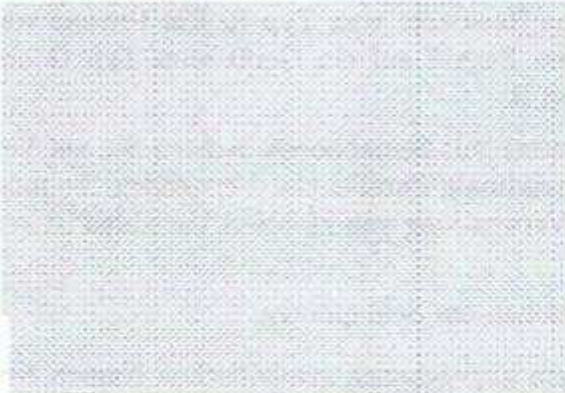
Die Prozesse A und B benötigen in der Funktion *do_the_work_A()* bzw. *do_the_work_B()* jeweils exklusiven Zugriff auf die jeweilige Anzahl an Geräten G.

Prozess Typ A:

```
do forever() {
```



```
do_the_work_A();
```



```
}
```

Prozess Typ B:

```
do forever() {
```



```
do_the_work_B();
```



```
}
```


2 Scheduling (25)

2.1 Uniprocessor Rate Monotonic Scheduling (10)

Gegeben ist nebenstehendes Taskset. Alle Tasks sind periodisch, wobei die Deadlines mit dem Ende der jeweiligen Periode gleichzusetzen sind. Der Overhead für den Taskwechsel ist vernachlässigbar.

Task	Ausführungszeit	Periodendauer
A	2	7
B	2	6
C	1	9
D	1	7

Ermitteln Sie für dieses Taskset die notwendige und die hinreichende Bedingung für das Rate Monotonic Scheduling (RMS) Verfahren. Berechnen Sie die Zahlenwerte überschlagsmäßig ($\sqrt[3]{2} \approx 1,41$ $\sqrt[4]{2} \approx 1,26$ $\sqrt[5]{2} \approx 1,19$ $\sqrt[6]{2} \approx 1,15$ $\sqrt[7]{2} \approx 1,12$).

$$\sum C_i/T_i = \frac{2}{7} + \frac{2}{6} + \frac{1}{9} + \frac{1}{7} = 0,84 \leq 1 \quad \text{notwendige Bedingung erfüllt}$$

$$\sum C_i/T_i \leq n \cdot (2^{1/n} - 1) \Rightarrow 0,84 \leq 4 \cdot (2^{1/4} - 1) \Rightarrow 0,84 \leq 4 \cdot 0,19$$

$$\Rightarrow 0,84 \leq 0,76 \quad \text{hinreichende Bedingung erfüllt nicht}$$

Ist die notwendige Bedingung erfüllt? ☒ Ja ☐ Nein

Ist die hinreichende Bedingung erfüllt? ☐ Ja ☒ Nein

Versuchen Sie das Taskset mit dem RMS Verfahren zu schedulen. Verwenden Sie dazu die nachstehenden Vorlagen. Tragen Sie bei jeder Vorlage die aktiven Taskzeiten ein und bezeichnen Sie deutlich eventuelle Deadlineverletzungen. Kreuzen Sie an, ob das Scheduling erfolgreich war. Eine Vorlage dient als Ersatz, streichen Sie gegebenenfalls eine falsch ausgefüllte Vorlage deutlich durch.

Scheduling nach dem RMS-Verfahren:

Erfolgreich: ☒ Ja ☐ Nein

A			A	A				A	A				A	A			
B	B	B					B	B					B	B			B
C					C						C						C
D				D					D							D	
	0		5	5	5	5	5	5	5	5	5	5	5	5	5	5	5

Ersatzvorlage: Scheduling nach dem RMS-Verfahren:

Erfolgreich: ☐ Ja ☐ Nein

A																	
B																	
C																	
D																	
	0		5						10						15		

2.2 Uniprocessor Round Robin Scheduling (7)

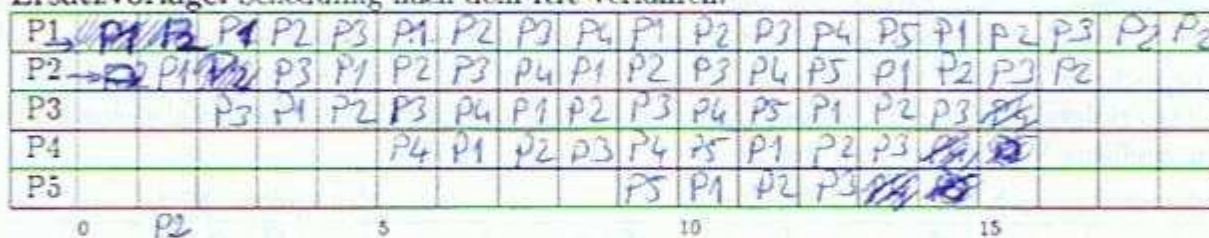
Scheduling nach dem Round Robin Verfahren: Versuchen Sie das Taskset nach dem *Round Robin* (RR) Verfahren zu schedulen. Fügen Sie zuerst *Tasks mit abgelaufener Zeitscheibe vor neu ankommenden Tasks* in die Ready Queue. Verwenden Sie die nachstehenden Vorlagen. Tragen Sie in die Vorlage die aktiven Taskzeiten ein und bezeichnen Sie deutlich (mit einem Pfeil \rightarrow) die Arrival Time der Tasks. Eine Zeitscheibe entspricht einer *Service Time* von eins. Eine der Vorlagen dient als Ersatz, streichen Sie gegebenenfalls eine falsch ausgefüllte Vorlage deutlich durch.

Process	Arrival Time	Service Time
P1	0	5
P2	1	7
P3	3	4
P4	6	2
P5	10	1

Scheduling nach dem RR-Verfahren:



Ersatzvorlage: Scheduling nach dem RR-Verfahren:



2.3 Scheduling allgemein (8)

Welche Scheduling-Kriterien kennen Sie? Vervollständigen Sie nachstehende Tabelle.

	User-oriented	System-Oriented
Performance	Response Time, Turnaround Time Deadlines	Throughput, Processor Utilization
Other	Predictability	Fairness, Priorities, Resource Balance

3 Deadlock (25)

Deadlock-Bedingungen (6)

Erklären Sie die Deadlock-Bedingungen *Hold and Wait* und *No Preemption*.

Hold and Wait:

Prozesse können Ressource halten, während sie auf weitere Ressourcen warten.

No Preemption:

Zugewiesene Ressourcen werden den Prozessen nicht weggenommen

Banker's Algorithmus (6)

Wozu wird der Banker's Algorithmus verwendet?

Banker's Algorithmus wird zur Vermeidung von Deadlock benutzt.

Was bedeutet es, wenn der Banker's Algorithmus einen *Safe State* bzw. einen *Unsafe State* diagnostiziert?

Safe State: sicheren Zustand

Es existiert eine Folge von Schritten, mit der der Algorithmus alle Prozesse als finished markiert. In den Endzustand bringt.

Unsafe State: unsicheren Zustand

Es existiert keine solche Schrittfolge.

Deadlock-Erkennung (13)

In einem Computersystem laufen fünf Prozesse, die gemeinsame Ressourcen aus vier Ressourcenkategorien verwenden. Die vorhandenen Ressourcen sind durch den Vektor $R = (6, 5, 9, 6)$ gegeben. Die unten gegebenen Matrizen beschreiben die aktuellen Ressourcenanforderungen und -allokation der fünf Prozesse. Führen Sie den Algorithmus zur Deadlock-Erkennung durch, um festzustellen, ob im gegebenen System ein Deadlock vorliegt.

$$Q = \begin{pmatrix} 2 & 0 & 6 & 4 \\ 0 & 1 & 3 & 0 \\ 3 & 0 & 1 & 1 \\ 1 & 4 & 3 & 2 \\ 2 & 0 & 0 & 1 \end{pmatrix} \quad A = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 4 & 2 \\ 3 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad V = (1, 2, 4, 0)$$

~~Q~~

$$Q = \begin{pmatrix} 2 & 0 & 6 & 4 \\ 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 1 \\ 1 & 4 & 3 & 2 \\ 2 & 0 & 0 & 1 \end{pmatrix} \quad A = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 3 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad V = (2, 2, 8, 2)$$

$$Q = \begin{pmatrix} 2 & 0 & 6 & 4 \\ 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 1 \\ 1 & 4 & 3 & 2 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad A = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 3 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad V = (2, 3, 8, 2)$$

Deadlock

$$Q = \begin{pmatrix} \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \end{pmatrix} \quad A = \begin{pmatrix} \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \end{pmatrix} \quad V = (\text{shaded}, \text{shaded}, \text{shaded}, \text{shaded})$$

$$Q = \begin{pmatrix} \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \end{pmatrix} \quad A = \begin{pmatrix} \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \end{pmatrix} \quad V = (\text{shaded}, \text{shaded}, \text{shaded}, \text{shaded})$$

$$Q = \begin{pmatrix} \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \end{pmatrix} \quad A = \begin{pmatrix} \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \end{pmatrix} \quad V = (\text{shaded}, \text{shaded}, \text{shaded}, \text{shaded})$$

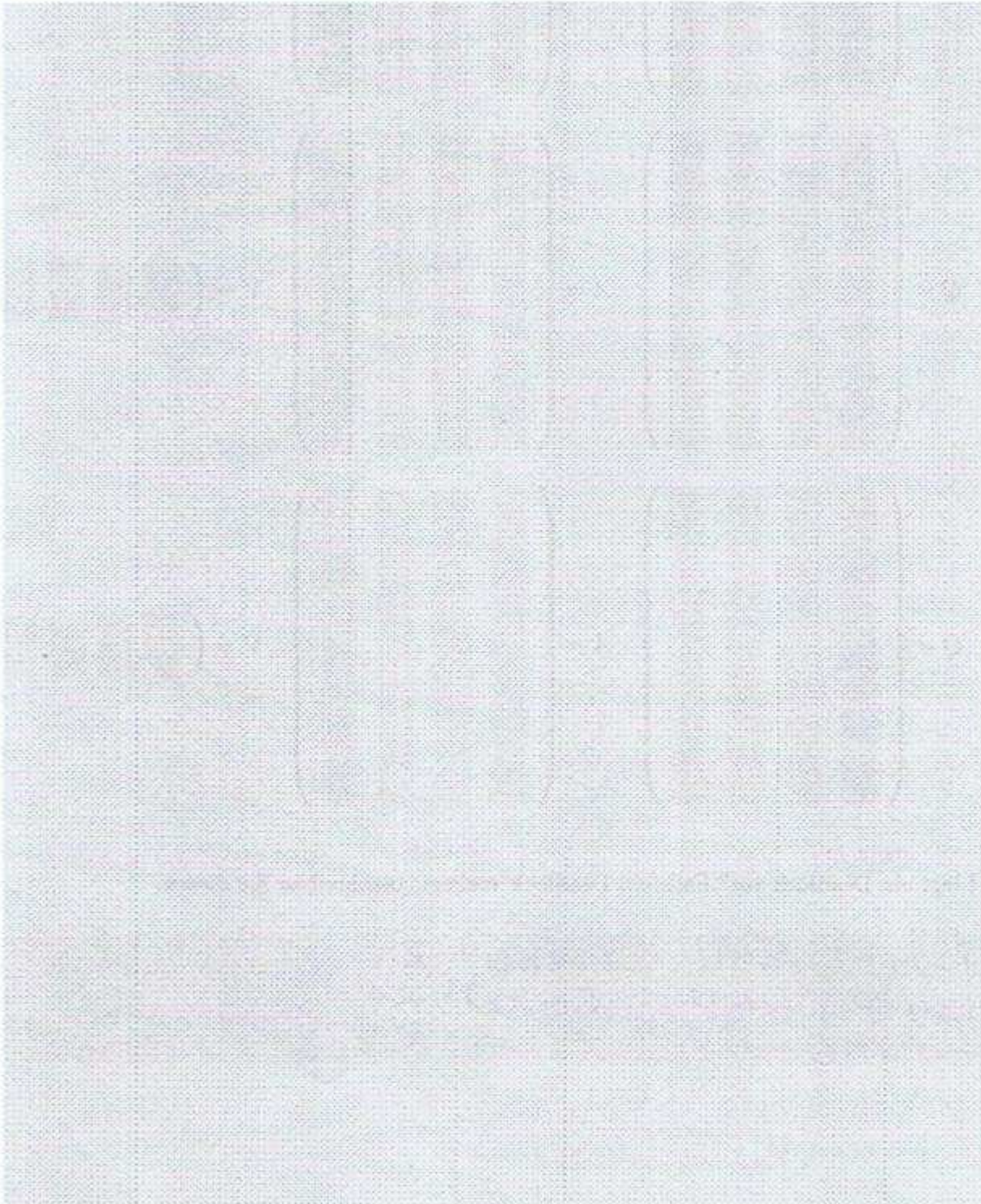
Liegt ein Deadlock vor? Falls ein Deadlock vorliegt, beschreiben Sie diesen.

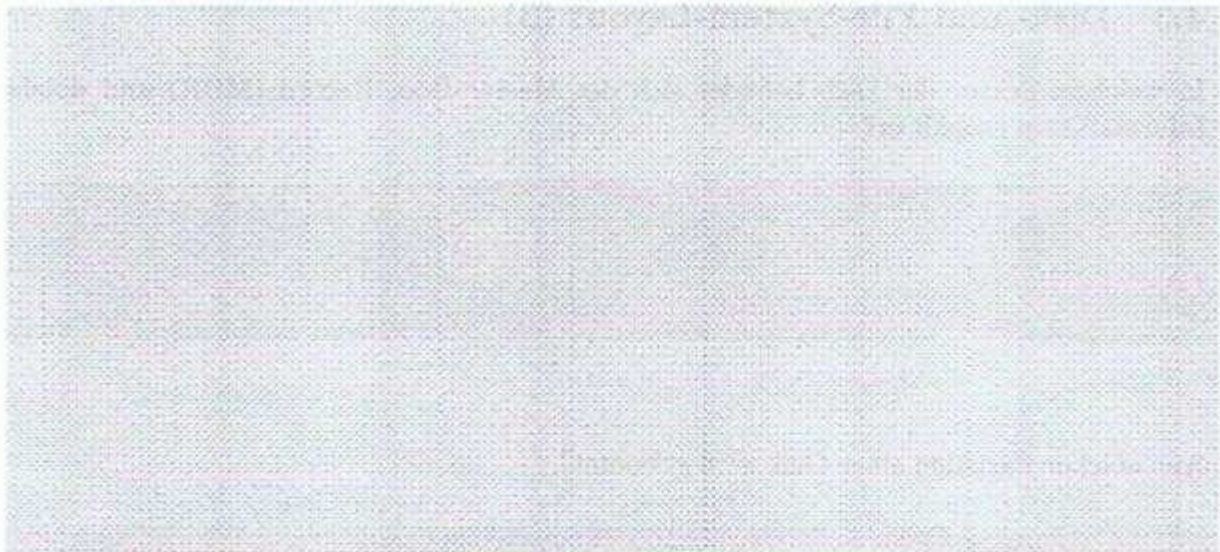
für P1 fehlt 2 Einheit R4
 für P3 fehlt 1 Einheit R1
 für P4 — 1 — 1 — 1 — R2

4 Filemanagement (25)

4.1 Allokation (12)

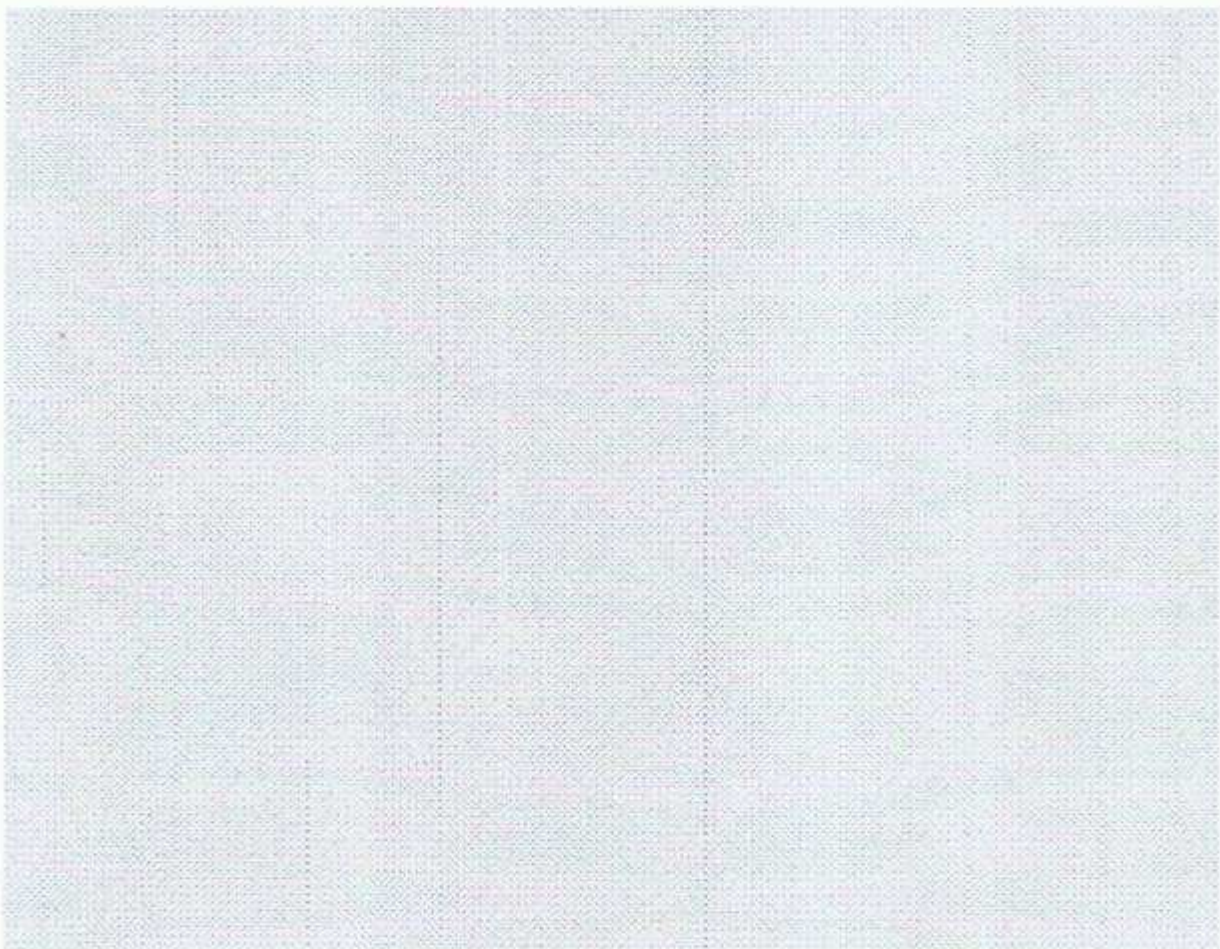
Welche Strategien zur Block-Allokation kennen Sie zur Implementierung einer Datei im sekundären Speicher? Erklären Sie diese und erläutern Sie Vor- und Nachteile!





4.2 Datei-Organisation und -Zugriff (10)

Nennen und erklären Sie fünf fundamentale Arten der Datei-Organisation!



4.3 Disk- und File-System-Layout (3)

In welchem Sektor der Disk befindet sich der Master Boot Record (MBR) und welche Informationen enthält er?



Aus welcher Partition einer Disk wird gebootet?

