

Musterlösung

13. 09. 2013

Alle Punkteangaben ohne Gewähr!

- Bitte tragen Sie zuerst auf dem Deckblatt Ihren Namen, Ihren Vornamen und Ihre Matrikelnummer ein. Tragen Sie dann auf den anderen Blättern (auch auf dem Konzeptblatt) Ihre Matrikelnummer ein.
Please fill in your last name, your first name, and your matriculation number on this page and fill in your matriculation number on all other pages (including the draft page).
- Die Prüfung besteht aus 15 Blättern: Einem Deckblatt und 15 Aufgabenblättern mit insgesamt 5 Aufgaben.
The examination consists of 15 pages: One cover sheet and 15 sheets containing 5 assignments.
- Es sind keinerlei Hilfsmittel erlaubt!
No additional material is allowed.
- Die Prüfung gilt als nicht bestanden, wenn Sie versuchen, aktiv oder passiv zu betrügen.
You fail the examination if you try to cheat actively or passively.
- Wenn Sie zusätzliches Konzeptpapier benötigen, verständigen Sie bitte die Klausuraufsicht.
If you need additional draft paper, please notify one of the supervisors.
- Bitte machen Sie eindeutig klar, was Ihre endgültige Lösung zu den jeweiligen Teilaufgaben ist. Teilaufgaben mit widersprüchlichen Lösungen werden mit 0 Punkten bewertet.
Make sure to clearly mark your final solution to each question. Questions with contradicting answers are void (0 points).
- Wir werden Punkte abziehen, falls korrekte Antworten auch inkorrekte oder irrelevante Informationen enthalten. Bitte schreiben Sie nicht einfach möglichst viel hin, in der Hoffnung, das richtige Schlagwort zu treffen.
We will take off points if a correct answer also includes incorrect or irrelevant information. Do not write down everything you know in hopes of saying the correct buzz word.

Die folgende Tabelle wird von uns ausgefüllt! *The following table is completed by us!*

Aufgabe	1	2	3	4	5	Total
Max. Punkte	12	12	12	12	12	60
Erreichte Punkte						
Note						

Aufgabe 1: Zum Aufwärmen / *Assignment 1: Warmup*

- a) Die Funktion `strcpy(char * target, char * source)` kopiert eine Zeichenkette. Vervollständigen Sie die folgende Zeile C-Code, um vor dem Kopieren ein genau passendes Speicherstück (`target`) vom Betriebssystem anzufordern.

1 pt

*The function `strcpy(char * target, char * source)` copies a character string. Complete the following line of C code to allocate an exactly fitting chunk of memory (`target`) from the operating system before copying.*

```
target =
```

Lösung:

```
target = malloc(strlen(source) + 1); (+1P)
```

For each syntax error: (-0.5P) – the semicolon is optional

- b) Nennen Sie die **drei** in der Vorlesung besprochenen **Zeitpunkte**, an denen bei der Verarbeitung eines Programms, dessen Programmtext und Programmvariablen an Speicheradressen gebunden werden.

1.5 pt

*Name the **three points in time** during program processing, when program text and program variables are bound to memory addresses, which were discussed in the lecture.*

Lösung:

- *Compile time (Übersetzungszeit) (+0.5P)*
- *Load time (Ladezeit) (+0.5P)*
- *Execution time (Ausführungszeit) (+0.5P)*

- c) Wie kann die CPU erkennen, dass ein nicht privilegierter Prozess versucht eine privilegierte Instruktion aufzurufen? Wie wird das Betriebssystem darüber benachrichtigt?

1 pt

How can the CPU detect that a non-privileged process has tried to execute a privileged instruction? How is the operating system notified about it?

Lösung:

- *The CPU has `mode flag` (i.e. user mode). Now, privileged instructions are prohibited to be executed. (+0.5P)*
- *The CPU generates an exception, which invokes an operating system handler immediately. (+0.5P)*

- d) Beschreiben Sie knapp die Unterschiede in der Struktur von Systemen auf Basis von **monolithischen und Mikro-Kernen**. In welcher Art von Systemen liegt prinzipiell ein größerer Overhead durch IPC vor? Begründen Sie Ihre Antwort kurz.

1.5 pt

*Describe the differences in the structures of systems based on **monolithic and micro kernels** briefly. Which type of systems imposes larger overhead caused by IPC in general? Explain your answer briefly.*

Lösung:

- *micro kernel: user-space (hosting applications and “as many as possible parts of the kernel”), kernel-space (hosting the “kernel essentials”) (+1P)*
- *Micro kernels impose larger IPC overhead, as kernel-internal communication imposes switches between user- and kernel-mode. (+0.5P)*

- e) Beschreiben Sie knapp den Unterschied zwischen **Virtualisierung und Para-Virtualisierung**. Welche Auswirkungen hat dies für das Gast-OS?

1 pt

*Describe the difference between **virtualization and para-virtualization** briefly. What are the implications for the guest OS?*

Lösung:

Either one of the following (+1P)

- *Using virtualization, guest operating systems are provided with (virtual but) identical copies of the underlying hardware. Therefore, the guests do not have to be modified.*
- *Using para-virtualization, guest operating systems are not provided with fully-identical copies of the underlying hardware. Therefore, the guests have to be “aware of their virtualization” and be modified.*

- f) Welche der folgenden Aussagen sind korrekt, welche sind inkorrekt?
 (falsches Kreuz: -1P, kein Kreuz: 0P, korrektes Kreuz: 1P)

6 pt

*Which of the following statements are correct, which are incorrect?
 (incorrectly marked: -1P, not marked: 0P, correctly marked: 1P)*

korrekt inkorrekt
correct incorrect

<input checked="" type="checkbox"/>	<input type="checkbox"/>	In SMPs müssen Caches durch geeignete Protokolle synchronisiert werden. <i>Caches of SMPs have to be synchronized using suitable protocols.</i>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	RAID 0+1 bietet höhere Ausfallsicherheit als RAID 1+0. <i>RAID 0+1 is more fault tolerant than RAID 1+0.</i>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	In RAID2 und RAID3 sollte die Rotation der Platten synchronisiert sein, um möglichst hohe Performanz zu bieten. <i>The rotation of disks should be synchronized in RAID2 and RAID3 to foster high performance.</i>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bei Nutzung des One-to-One-Threading-Modells kann das Betriebssystem I/O-Operationen nicht überlappen, falls es Prozessen nur eine synchrone I/O-API bietet. <i>When using the one-to-one threading model, the operating system cannot overlap I/O operations if it offers only a synchronous I/O API to processes.</i>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Bei IPC mittels synchronem Message Passings blockiert ein <code>send</code> -Aufruf solange bis der Empfänger die <code>receive</code> -Operation abgeschlossen hat. <i>In IPC with synchronous message passing, a <code>send</code> call blocks until the receiver has completed the <code>receive</code> operation.</i>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Der TLB kann nicht in Hardware implementiert werden. <i>The TLB cannot be implemented in hardware.</i>

**Total:
12.0pt**

Aufgabe 2: Prozesse und Ablaufplanung / Assignment 2: Processes and Scheduling

- a) Erklären Sie den Begriff **Zombie** im Kontext von Prozessen. Geben Sie insbesondere an, wie ein Zombie entsteht und wie man ihn entfernt.

1.5 pt

*Explain the term **zombie** in the context of processes. State specifically how it is created and how to get rid of it.*

Lösung:

- *Zombie: Stub of a (child) process in the process table after the (child) process has terminated. (+0.5P)*
- *Parent processes need to collect the exit status of their terminated child processes using the `wait` or `waitpid` syscalls to get rid of zombies. (Note: Zombies exist for precisely that reason, to keep a child process' exit status.) (+1P)*

Erklären Sie den Begriff **Orphan** im Kontext von Prozessen.

0.5 pt

*Explain the term **orphan** in the context of processes.*

Lösung:

- *A process whose (original) parent process has died (+0.5P), whereby the child is left orphaned.*

- b) Nennen Sie ein **Threading-Modell**, welches das Debugging von Anwendungen begünstigt, und eines, welches es erschwert. Begründen Sie Ihre Antworten.

2 pt

*Name a **threading model** which favors application debugging and one that complicates it. Explain your answers.*

Lösung:

- *Favors: one-to-one (+0.5P). A debugger can interact with a single User-Level Thread without interfering with concurrent threads (of the same application). (+0.5P)*
- *Complicates: many-to-one (+0.5P). When a debugger interacts with a User-Level Thread it interferes strongly with concurrent threads (of the same application), e.g. if one thread is halted, any other threads have to be halted too. (+0.5P)*

- c) Erklären Sie knapp die Rollenverteilung zwischen **CPU-Scheduler** und **Dispatcher**.

1 pt

*Explain briefly the separation of concerns between **CPU scheduler** and **dispatcher**.*

Lösung:

- *CPU Scheduler: Policy welcher Prozess als nächstes läuft. (+0.5P)*
- *Dispatcher: Mechanismus der den tatsächlichen Prozesswechsel durchführt. (+0.5P)*

- d) Gegeben seien die untenstehenden 3 Prozesse auf einem Uniprocessorsystem mit den angegebenen Ankunftszeiten (Start-Zeit = 1.0) und Burst-Zeiten. Vervollständigen Sie die untenstehenden Ablaufpläne für die Strategien **Round Robin (RR)** und **Shortest-Job-First (SJF)**. Ein Kasten im Zeitplan entspricht einer Zeitscheibe. Hinweise: Rechts neben den Ablaufplänen ist etwas Platz für Notizen. Zum Zeitpunkt 11 sind alle Prozesse beendet.

2 pt

Consider the 3 processes given below on a uniprocessor system, with the given arrival times (start time = 1.0) and burst times. Complete the schedules given below according to the policies **Round Robin (RR)** and **Shortest Job First (SJF)**. A box in the schedule represents one time slice/quantum.

Notes: right next to the schedules is some free space for notes. All processes are finished at time 11.

Prozess Process	Ankunfts-Zeit Arrival time	Burst-Zeit Burst time
P_1	1.9	5.0
P_2	1.0	4.0
P_3	2.6	1.0

Lösung:

Zeit Time	Ablaufplan RR Schedule RR	Ablaufplan SJF Schedule SJF
1	2	2
2	1	2
3	2	2
4	3	2
5	1	3
6	2	1
7	1	1
8	2	1
9	1	1
10	1	1

- RR correct (+1P)
- SJF correct (+1P)

Welcher der beiden Ablaufpläne hat eine **geringere durchschnittliche Turnaround-Zeit**? Und wieviel geringer? Geben Sie ihren Rechenweg an. Ergebnisse in Bruchform sind ausreichend.

2 pt

Which of the two schedules has a **lower average turnaround time**? How much lower is it? Provide your calculations. Results in fraction form are sufficient.

Lösung:

RR:

Process	Arrival time	Finish time	Turnaround time
P_1	1.9	11.0	9.1
P_2	1.0	9.0	8.0
P_3	2.6	5.0	2.4

$$T_{P_1}^T = 9.1, T_{P_2}^T = 8.0, T_{P_3}^T = 2.4$$

$$T_{avg}^T = \frac{T_{P_1}^T + T_{P_2}^T + T_{P_3}^T}{3} = \frac{9.1 + 8.0 + 2.4}{3} = \frac{195}{30} = 6\frac{1}{2} \quad (+0.5P)$$

SJF:

Process	Arrival time	Finish time	Turnaround time
P_1	1.9	11.0	9.1
P_2	1.0	5.0	4.0
P_3	2.6	6.0	3.4

$$T_{P_1}^T = 9.1, T_{P_2}^T = 4.0, T_{P_3}^T = 3.4$$

$$T_{avg}^T = \frac{T_{P_1}^T + T_{P_2}^T + T_{P_3}^T}{3} = \frac{9.1 + 4.0 + 3.4}{3} = \frac{165}{30} = 5\frac{1}{2} \quad (+0.5P)$$

- Geringer: SJF (+0.5P)
- Wieviel: $6\frac{1}{2} - 5\frac{1}{2} = 1$ (+0.5P)

e) Welche Art von Prozessen wird von **Virtual Round Robin** gegenüber **Robin Robin** gefördert? Begründen Sie ihre Antwort knapp.

1 pt

Which kind of process is supported by **Virtual Round Robin** in contrast to **Robin Robin**? Explain your answer briefly.

Lösung:

- I/O-bound processes are prioritized over CPU-bound processes. (+0.5P)
- When an I/O has completed, the blocked process is moved to an auxiliary queue which gets preference over the main ready queue. (+0.5P)

- f) Nennen und beschreiben Sie 4 in der Vorlesung besprochene Kriterien, die ein Scheduler optimieren kann.

2 pt

Name and describe 4 criteria which a scheduler can optimize, which have been discussed in the lecture.

Lösung:

Throughput # of processes that complete per time unit (+0.5P)

Turnaround time time from submission to completion (+0.5P)

Response time time from request to first response (+0.5P)

CPU utilization keep the CPU as busy as possible (+0.5P)

Waiting time time each process waits in ready queue (+0.5P)

**Total:
12.0pt**

Aufgabe 3: Prozesskoordination und -kommunikation / Assignment 3: Process Coordination and Communication

- a) Erklären Sie **Direct Communication** und **Indirect Communication** im Kontext von Prozesskommunikation.

2 pt

*Explain **direct** and **indirect communication** in the context of Process Communication.*

Lösung:

Direct communication:

- *Processes must name each other explicitly: $send(P, message), receive(P)$ (+0.5P)*
- *Links are established automatically, exactly one link is associated with exactly one pair of communicating processes. (+0.5P)*

Indirect communication:

- *Messages are directed to and received from mailboxes. (+0.5P)*
- *Link established only if processes share a common mailbox. A link may be associated with many processes. (+0.5P)*

- b) Beschreiben Sie das **Producer-Consumer-Problem**.

2 pt

*Describe the **producer-consumer problem**.*

Lösung:

- *At least one producer produces data that is consumed by at least one consumer. Concurrent access to shared data may result in data inconsistency. (+1P)*
- *Maintaining data consistency requires mechanisms to ensure the orderly execution of cooperating processes. (+1P)*
 - *There must be no race conditions in data accesses. (+0.5P)*
 - *If a bounded buffer is used, the producer does not overflow the buffer, consumer does not underflow the buffer. (+0.5P)*

- c) Erklären Sie knapp was **Critical Sections** sind.

0.5 pt

*Explain briefly what **critical sections** are.*

Lösung:

- *Sections of programs where concurrent activities or threads access shared data. (0.5P)*

- d) Betrachten Sie das folgende Code-Beispiel. Nehmen Sie an, dass **zwei Producer** und ein **Consumer** in einem System existieren. Sonst gibt es keine weitere Aktivität in diesem System. Das System hat einen präemptiven Scheduler.

*Have a look at the following code sample. Assume that **two producers** and **one consumer** exist in a system. There is no other activity in this system. The system schedules pre-emptively.*

```
1 Item buffer[SIZE]; // assume SIZE > 1
2 unsigned int in = 0, out = 0;
3 Semaphore sem1 = sem_init(SIZE), sem2 = sem_init(0);
```

```
4 void producer() {
5     for (;;) {
6         /* ... create new item */
7         wait(sem1);
8         buffer[in] = item;
9         in = (in + 1) % SIZE;
10        signal(sem2);
11    }
12 }
```

```
13 void consumer() {
14     for (;;) {
15         wait(sem2);
16         item = buffer[out];
17         out = (out + 1) % SIZE;
18         signal(sem1);
19         /* ... work with item */
20     }
21 }
```

Liegt in dem Code eine **Race-Condition** vor? Wenn ja, nennen Sie die fehlerhaften Zeilen im Code.

1 pt

*Does a **race condition** exist in this code? If yes, refer to the incorrect lines in the code.*

Lösung:

Yes. The two producers may race in accessing the variable `in` in lines 8 and 9. (+1P)

Beschreiben Sie einen Ablauf, der dazu führt, dass der Consumer ein **ungültiges Item** aus dem Puffer nimmt.

2 pt

*Describe a flow of execution that leads to the consumer remove an **invalid item** from the buffer.*

Lösung:

- *Producer 1 (P1) and Producer 2 (P2) execute line 8 in any order. (the second valid item overwrites the first valid item in the first element of the buffer.) (+0.5P)*
- *P1 executes line 9 first. (a buffer fill size of 1 is signaled.) (+0.5P)*
- *P2 executes line 9 second. (an incorrect fill size of 2 is signaled.) (+0.5P)*
- *Consumer (C) takes an invalid item from the buffer the second time it hits line 16. (+0.5P)*

- e) Wie können Sie das Problem aus der vorherigen Teilaufgabe mit **genau einem weiteren Semaphore** lösen? **1.5 pt**

*How can you solve the problem in the previous task by adding **exactly one semaphore**?*

Lösung:

- *New semaphore Semaphore sem3 = sem_init(1); (+0.5P)*
- *wait(sem3); between lines 7 and 8 (or between lines 6 and 7) and signal(sem3); between lines 9 and 10 (or between lines 10 and 11). (1P)*

- f) Erklären Sie knapp, wann sich ein System im Kontext von Ressourcenzuteilung in einem **unsicheren Zustand** befindet? **1 pt**

*Explain briefly: when is a system considered in an **unsafe state** in the context of resource allocation?*

Lösung:

A system is in an unsafe state, if there exists no schedule such that the maximal resource requests of all processes can be satisfied. (+1P)

- g) Erklären sie jeweils knapp **Deadlock Prevention** und **Deadlock Avoidance**. **2 pt**

*Briefly explain **deadlock prevention** and **deadlock avoidance**.*

Lösung:

- *Deadlock Prevention: Negate at least one of the required deadlock conditions (+1P)*
- *Deadlock Avoidance: Decide if system stays in safe state on every resource request (+1P)*

**Total:
12.0pt**

Aufgabe 4: Speicher und Caches / Assignment 4: Memory and Caches

Betrachten Sie ein System, welches virtuelle Adressen mittels **dreistufiger Seitentabellen** auf physische Adressen übersetzt. **Virtuelle Adressen** sind **32 Bit** lang, und **physische Adressen** sind **64 Bit** lang. Die Seitengröße beträgt 4 KiB. Jede Seitentabelle besteht aus 8 Byte langen Einträgen. Die erste Stufe besteht aus nur 4 Einträgen, und die tieferen beiden aus je 512 Einträgen. Der Speicher ist byte-adressierbar.

Hinweis: Ergebnisse in Form von Zweierpotenzen sind ausreichend.

Consider a system that uses **three-level page-tables** to translate virtual addresses to physical addresses. **Virtual addresses** are **32 bit** long, and **physical addresses** are **64 bit** long. The page size is 4 KiB. Each page-table consists of 8 bytes long entries. The first level consists of only 4 entries, and the lower two consist of 512 entries each.

Note: Results in form of powers of two are sufficient.

- a) In welche vier Teile muss eine virtuelle Adresse zerlegt werden, um eine Adressübersetzung durchführen zu können? Aus wie vielen Bits besteht jeder dieser Teile unter den oben genannten Voraussetzungen?

3 pt

Into which four parts must a virtual address be split so that it can be translated into a physical address? How many bits are required for each part under the assumptions stated above?

Lösung:

- 1-st level index (aka page directory table), 2 bits
- 2-nd level index (aka page directory), 9 bits
- 3-rd level index (aka page table), 9 bits
- entry level index (aka offset), 12 bits

+0,5P für Nennung von lvl 1 bis 3 (zusammen, nicht jeweils),
+0,5P für Nennung von offset/entry level,
jeweils +0,5P für korrekt angegebene bit-Länge.

- b) Wie groß ist der adressierbare virtuelle Speicher (in GiB)?

1 pt

How large is the addressable virtual Memory (in GiB)?

Lösung:

- $2^{32} \text{ Byte} = 4 \text{ GiB}$ (+1P)

- c) Wieviel Speicher benötigt die gesamte Seitentabelle **eines Prozesses** mindestens, falls diesem **1 GiB Speicher** zugewiesen wurde? Ignorieren Sie den durch die erste Stufe belegten Speicher.

3 pt

How much memory does the entire page table of a process consume at least if the process has been allocated 1 GiB of memory? Ignore the consumption of the first level.

Lösung:

- # of pages = # of page table entries = $\frac{1GiB}{4KiB} = \frac{2^{30}}{2^2 \times 2^{10}} = 2^{18}$ (+1P)
- min. # of page tables = $\frac{2^{18}}{2^9} = 2^9$ (+1P)
- min. # of page directories = $\frac{2^9}{2^9} = 1$ (+0.5P)
- memory consumption = $2^9 \times 4KiB + 1 \times 4KiB = 2^{10} \times 2KiB + 4KiB = 2MiB + 4KiB$ (+0.5P)

- d) Der TLB mit 4096 Einträgen basiert auf einem **8-Wege Satz-assoziativem Cache**. Aus wievielen Sätzen besteht der TLB?

0.5 pt

The TLB contains 4096 entries and is based on a 8-way set associative cache. How many sets does the TLB consist of?

Lösung:

- $\frac{4096}{8} = 512$ sets (+0.5P)

Welche Stufen der Seitentabelle werden durch den TLB beschleunigt?

0.5 pt

Which levels of the page table are speed up by the TLB?

Lösung:

- All three (+0.5P)

Welche Bits dienen als **Set-Index** und welche als **Tag**, wenn eine Teilfolge der Adressbits als Set-Index genutzt wird?

1 pt

Which bits serve as the set index and which as the tag, if a sub-sequence of the address bits is used as set index?

Lösung:

31	21	20	12
Tag		Index	

- Index: identifies the set in the TLB
- Tag: identifies the entry in the set

+0,5P für Tag und richtige Bit-Grenzen +0,5P für Index und richtige Bit-Grenzen

- e) Bestimmen Sie die **effektive Speicherzugriffszeit** unter der Annahme, dass ein Lookup in unserem TLB 20 ns und ein Speicherzugriff 100 ns dauert und dass der TLB eine durchschnittliche Trefferrate von 75% hat. Nehmen Sie an, dass das System keine weiteren Caches außer dem TLB besitzt. Geben Sie den Rechenweg in nachvollziehbarer Form an, nicht nur das Ergebnis.

3 pt

Calculate the **effective memory access time**, assuming that a lookup in our TLB takes 20 ns, a memory access takes 100 ns, and that the TLB has an average hit ratio of 75%. Assume that the system uses no caches apart from the TLB. Include all calculation steps in your answer, not just the result.

Lösung:

- *Grundformel:* $0.75 \times Hit + 0.25 \times Miss$ (+0.5P)
- *Hit:* $20ns + 100ns = 120ns$ (+1P)
- *Miss:* $20ns + 100ns + 100ns + 100ns + 100ns = 420ns$ (+1P)
- *Ergebnis:* $195ns$ (+0.5P)

**Total:
12.0pt**

Aufgabe 5: Hintergrundspeicher und Dateisysteme / Assignment 5: Secondary Storage and File Systems

- a) Wir haben in der Vorlesung drei Dateiallokationsstrategien kennengelernt: (a) **Indexed Allocation**, (b) **Chained Allocation** und (c) **Contiguous Allocation**. Sortieren Sie die Strategien nach der Reihenfolge ihrer Eignung in den folgenden Szenarien. Begründen Sie für jede Strategie kurz, warum sie sich eignet/nicht eignet.

3 pt

*We have discussed three file allocation strategies in the lecture: (a) **indexed allocation**, (b) **chained allocation**, and (c) **contiguous allocation**. Order the strategies based on how well they perform in each of the following scenarios. Provide a short explanation why each strategy performs well/bad.*

Schnelles Auslesen aufeinanderfolgender Blöcke einer Datei.

Fast reading of consecutive blocks of a file.

Lösung:

- (a) Contiguous allocation: Am besten, da Blöcke einer Datei direkt zusammenhängen. (+0.5P)*
- (b) Indexed allocation: Schlechter, da Blöcke einer Datei nicht direkt zusammenhängen müssen. (+0.5P)*
- (c) Chained allocation: Noch schlechter, da nächste Blockadressen erst berechnet werden können nachdem ein aktueller Block geladen wurde. (+0.5P)*

Effiziente Datenspeicherung durch wenig Metadaten. Gehen Sie davon aus, dass der Speicher unfragmentiert ist.

Efficient storage of data in terms of few metadata. Assume that the memory is defragmented.

Lösung:

- (a) Contiguous allocation: Am besten, da nur eine Anfangsadresse und eine Längeninformation für die Datei gespeichert werden muss. (+0.5P)*
- (b) Chained allocation: kleine Datenstrukturen nur Pointer am Ende von Blöcken. (+0.5P)*
- (c) Indexed allocation: größere Datenstrukturen als bei Chained allocation (Index-Struktur und Block-Pointer). (+0.5P)*

- b) Welche Metadaten einer Datei sind notwendig, wenn **Chained Allocation** zur Reservierung von Plattenspeicher verwendet wird?

1 pt

*Which metadata of a file are necessary if **chained allocation** is used to allocate disk space?*

Lösung:

- *Anfangsadresse des ersten Blocks (+0.5P)*
- *Jeweils die Adressen des nächsten Blöcke (+0.5P)*

- c) Gegenüberstellung von absoluten und relativen Pfadnamen: geben Sie ein Beispiel an, dass absolute Pfadnamen für den Zugriff auf eine Datei weniger Festplattenzugriffe erfordern als relative, und ein weiteres Beispiel für den umgekehrten Fall. Gehen Sie davon aus dies sei ihr **Current Working Directory**: /home/user1/

2 pt

*Comparison of absolute and relative pathnames: provide an example in which more disc accesses are needed to access a file when using absolute pathnames instead of a relative ones, and provide another example to illustrate the opposite case. Assume the following is your **current working directory**: /home/user1/*

Lösung:

Absolute faster than relative path:

- */home/user1/somefile \longleftrightarrow ../../../../home/user1/somefile (+1P)*

Relative faster than absolute path:

- *somedir/somefile \longleftrightarrow /home/user1/somedir/somefile (+1P)*

- d) Sie haben ein **RAID1** aus **zwei Festplatten**. Die Platten rotieren mit 1200 RPM, bestehen aus Spindeln von je 8 Platten, besitzen 16 Köpfe, haben eine Sektorgröße von 512 Bytes und haben 50 Sektoren auf jeder Spur. Für beide Platten beträgt die durchschnittliche Zeit eine Spur zu fokussieren 10ms und die durchschnittliche Verzögerung um zu einem Sektor auf einer ihrer Spuren zu rotieren 5ms. Gehen Sie davon aus, dass in einer Festplatte zu einer Zeit nur ein Kopf aktiv (lesen oder schreiben) sein kann.

*You have **two hard disk drives** in a **RAID1** setup. The disks spin at 1200 RPM, have spindles of 8 platters, 16 heads, a sector size of 512 bytes and hold 50 sectors per track. The average time to seek a track on each disk is 10ms, and the average delay to rotate to a sector on their tracks is 5ms.*

Assume that only one head per hard disk drive can be active (reading or writing) at a time.

Geben Sie die maximale Übertragungsrate an mit der Sequenzen von zufällig ausgewählten Sektoren beschrieben werden können. Nehmen Sie an, dass sich jeder Sektor einer zufälligen Sequenz auf einer anderen Spur befindet.

2 pt

Provide the maximum transfer rate at which sequences of randomly selected sectors can be written. Assume that each sector in the random sequence is on a different track.

Lösung:

- Each transfer writes 512 bytes to both disks, which takes $\frac{1}{\frac{1200rpm}{60s} * 50sectors} = 1ms$. (+1P)
- Each transfer involves an average seek time of 10ms and an average rotational delay of 5ms. (+0.5P)
- Therefore, we service the sequences at $\frac{1}{1ms+10ms+5ms} * 512 \text{ bytes} = 32.000B/s$. (+0.5P)

Wie würde sich die maximale Übertragungsrate verändern, wenn beide Platten in einem **RAID0** angeordnet wären? Begründen Sie Ihre Antwort kurz.

1 pt

*How would the maximum transfer rate change if both disks were arranged in a **RAID0**? Explain your answer briefly.*

Lösung:

- Each transfer writes 512 bytes to one disk only, therefore we can service two writes in parallel. (+0.5P)
- Therefore, we service the sequences at $2 * 32.000B/s = 64.000B/s$. (+0.5P)

- e) Zählen Sie **drei verschiedene RAID-Architekturen** auf und nennen Sie die Eigenschaften durch welche Redundanz und Performanz verbessert werden. Kombinierte Architekturen, wie z.B. RAID 0+1, sind ausgeschlossen.

3 pt

Enumerate **three different RAID architectures** and name the features by which redundancy and performance are improved.

Combined architectures, such as RAID 0+1, are excluded.

Lösung:

RAID	Redundancy	Performance
RAID	Redundanz	Performanz
RAID0	n/a	Block-level striping and non-synchronized spindles offers parallel reads and writes
RAID1	Mirroring	Mirrored blocks and non synchronized spindles offers parallel reads only
RAID2	Error-correcting code on extra disks	Bit-level striping and synchronized spindles restricts to one parallel read/write, but offers high throughput
RAID3	Byte-level parity on extra parity disk	Byte-level striping and synchronized spindles restricts to one parallel read/write, but offers high throughput
RAID4	Block-level parity on extra parity disk	Block-level striping and non-synchronized spindles offers parallel reads
RAID5	Distributed block-level parity	Block-level striping and non-synchronized spindles offers parallel reads

**Total:
12.0pt**