



Musterlösung

14. 09. 2010

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 14 Blättern: Einem Deckblatt, 12 Aufgabenblättern mit insgesamt 5 Aufgaben und einem Konzeptblatt.
The examination consists of 14 pages: One cover sheet, 12 sheets containing 5 questions, and one draft page.
- Es sind keinerlei Hilfsmittel erlaubt!
No additional materials are 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 mehreren bzw. widersprüchlichen Lösungen werden mit 0 Punkten bewertet.
Make sure to clearly mark your final solution to each question. Questions with multiple or contradicting answers are void (0 points).

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 / Question 1: Warmup

- a) Geben Sie das Ergebnis der folgenden Rechenoperation in hexadezimaler Darstellung an. („>>“ ist ein Bitshift-Operator, „&“ ein bitweises „und“.)

2 pt

What is the result of the following arithmetic operation in hexadecimal notation? (“>>” is a bitshift operator, “&” is a bit-wise “and”).

$$(0x0ABC \gg 4) \& 0x060$$

- b) Welche der folgenden Aussagen sind korrekt, welche sind inkorrekt?

5 pt

Which of the following statements are correct, which are incorrect?

korrekt/ correct	inkorrekt/ incorrect	
---------------------	-------------------------	--

- | | | |
|--------------------------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <input type="checkbox"/> | <input type="checkbox"/> | Die CPU-Instruktion, die Interrupts ausschaltet, ist privilegiert.
<i>The CPU-instruction that turns off interrupts is privileged.</i> |
| <input type="checkbox"/> | <input type="checkbox"/> | Multitasking ist nur auf Mehrprozessorsystemen möglich.
<i>Multitasking is only possible on multiprocessor systems.</i> |
| <input type="checkbox"/> | <input type="checkbox"/> | Beim One-to-One-Modell teilen sich alle Threads in einem Adressraum den selben Stack.
<i>In the one-to-one model, all threads within an address space share the same stack.</i> |
| <input type="checkbox"/> | <input type="checkbox"/> | Das Hauptproblem von physisch indizierten (<i>physically indexed</i>) Caches ist das Auftreten von Mehrdeutigkeiten.
<i>The main problem of physically indexed caches is the occurrence of ambiguities.</i> |
| <input type="checkbox"/> | <input type="checkbox"/> | Interrupts werden vom Betriebssystem ausgelöst, wenn ein Benutzerprogramm eine unzulässige Instruktion ausgeführt hat.
<i>Interrupts are raised by the operating system when a user program executed an illegal instruction.</i> |

- c) Erläutern Sie kurz die Unterschiede zwischen einem System mit monolithischem Kern und einem mikrokernbasierten System. Nennen Sie einen Nachteil eines mikrokernbasierten Systems gegenüber einem System mit monolithischem Kern.

3 pt

Briefly explain the differences between a system with a monolithic kernel and a microkernel-based system. State one disadvantage of a microkernel-based system compared to a monolithic system.

- d) Nennen Sie zwei Möglichkeiten, wie bei einem Systemaufruf Parameter an den Betriebssystemkern übergeben werden können.

2 pt

List two possibilities of how parameters can be passed to the kernel when a system call is performed.

**Total:
12 pt**

Aufgabe 2: Prozesse / Question 2: Processes

- a) Skizzieren Sie einen gebräuchlichen Aufbau des virtuellen Adressraums eines Prozesses. 2 pt

Depict a common layout of the virtual address space of a process.

- b) Erörtern Sie Vor- und Nachteile des Many-to-One-Modells gegenüber dem One-to-One-Modell. 2 pt

Discuss advantages and disadvantages of the many-to-one model compared to the one-to-one model.

- c) Betrachten Sie die untenstehende Tabelle und das zugehörige Gantt-Diagramm in Abbildung 2. Welches aus der Vorlesung bekannte Scheduling-Verfahren wurde verwendet, um den dargestellten Ablauf zu erreichen? 2 pt

Consider the table below and the corresponding Gantt chart in Figure 2. Which of the scheduling strategies introduced in the lecture was used to obtain the depicted schedule?

Prozess/process	Burstlänge/burst length	Ankunftszeit/arrival time
P_0	2	0
P_1	3	0
P_2	1	3
P_3	2	5

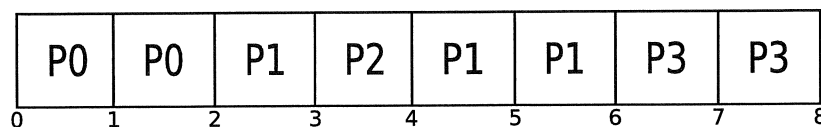


Abbildung 2: Gantt-Diagramm / Gantt chart

- d) Berechnen Sie für obigen Ablauf die Turnaround-Zeit für jeden der vier Prozesse. 2 pt

Calculate the turnaround-time of each of the four processes in the above schedule.

- e) Berechnen Sie die durchschnittliche Wartezeit unter obigem Ablauf. 2 pt

Calculate the average waiting time for the above schedule.

- f) In einem Mehrprozessor-System kann man entweder eine gemeinsame Ready-Queue für alle Prozessoren oder eine lokale Ready-Queue pro Prozessor verwenden. Diskutieren Sie die Vor- und Nachteile der beiden Möglichkeiten.

2 pt

In a multiprocessor system, one can either use a shared ready queue for all processors, or a local ready-queue per processor. Discuss advantages and disadvantages of both approaches.

**Total:
12 pt**

Aufgabe 3: Synchronisation/Question 3: Synchronization

- a) Betrachten Sie ein Programm bestehend aus $N > 4$ Threads. Das Programm enthalte einen Codeabschnitt, der von genau vier Threads gleichzeitig ausgeführt werden muss. Die Funktion `wait_for_others()` (siehe nächstes Blatt) wird von jedem Thread vor dem Eintritt in diesen Abschnitt aufgerufen, funktioniert allerdings nicht korrekt. Beschreiben Sie, wie es passieren kann, dass weniger als vier Threads den entsprechenden Codeabschnitt betreten.

4 pt

Consider a program consisting of $N > 4$ threads. The program contains a code section that must be executed by exactly four threads. The function `wait_for_others()` (see next page) is executed by each thread before it enters that code section, but does not work correctly. Depict a scenario in which less than four threads enter the code section.

- b) Finden und erläutern Sie ein weiteres Problem, das bei der Verwendung der Funktion `wait_for_others()` auftreten kann.

3 pt

Find and explain an additional problem that can occur when using the function `wait_for_others()`.

- c) Abbildung 3 zeigt einen fehlerhaften resource-allocation graph. Markieren Sie alle Fehler in der Abbildung und erklären Sie (direkt im Bild) kurz, was daran falsch ist. Nehmen Sie an, dass Ressourcen weder gemeinsam genutzt noch entzogen werden können.

2 pt

Figure 3 shows a resource-allocation graph that contains errors. Mark all errors in the figure and briefly explain (next to your markings) what is wrong. Assume that resources can neither be shared nor preempted.

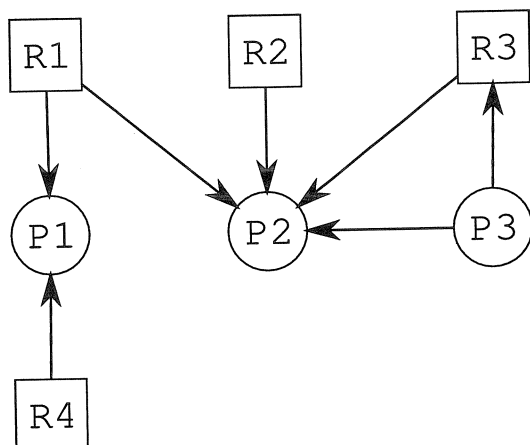


Abbildung 3: RAG mit Fehlern / RAG with errors

- d) Transformieren Sie den in Abbildung 4 dargestellten resource-allocation graph in einen wait-for graph.

3 pt

Transform the resource-allocation graph depicted in Figure 4 into a wait-for graph.

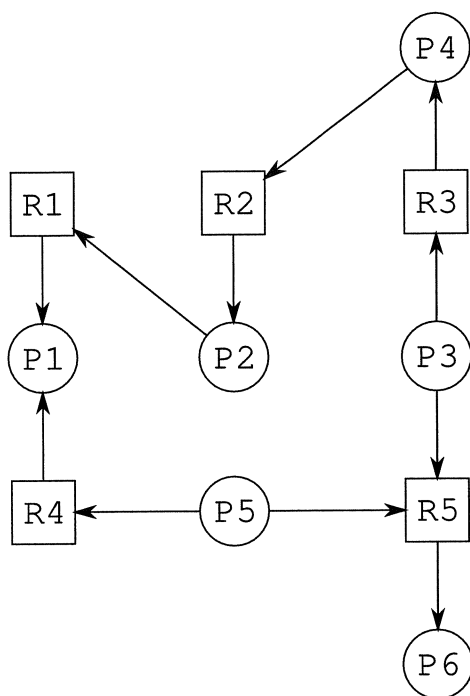


Abbildung 4: Resource-allocation graph

Total:
12 pt

Aufgabe 4: Speicher / Question 4: Memory

- a) Bei x86-Prozessoren enthält das sogenannte CR3-Register die Basisadresse der äußeren Seitentabelle (das sog. page directory). Wird vom Betriebssystem eine neue Adresse in dieses Register geladen, so invalidiert die Hardware automatisch den TLB. Warum?

2 pt

x86 processors use a register called CR3 to hold the base address of the outer page table (called page directory). When the operating system loads a new address into this register, the hardware automatically invalidates the TLB. Why?

- b) Durch welche Änderung am TLB könnte diese Invalidierung vermieden werden?

1 pt

What modification of the TLB could serve to avoid this invalidation?

- c) Erklären Sie die Begriffe demand paging und pre-paging. Bei welchem Verfahren ist mit mehr "verschwendetem" Speicher zu rechnen? Warum kann dieses Verfahren dennoch von Vorteil sein?

3 pt

Explain the terms demand paging and pre-paging. Which method is expected to "waste" more memory? Why can this method nevertheless be advantageous?

- d) Durch welche Technik können die mit dem `fork()`-Systemaufruf verbundenen Kosten (in Form von Rechenzeit) deutlich gesenkt werden?

1 pt

Which technique can be used to significantly reduce the costs (in terms of CPU time) of the `fork()` system call?

- e) Welche Seite wählt der optimale Seitenersetzungsalgorithmus zur Ersetzung aus?

1 pt

What page is selected for replacement by the optimal page replacement algorithm?

- f) Erklären Sie, wie bei der Verwendung von Segmentierung eine logische Adresse auf eine physische Adresse umgesetzt wird (Bild + Erklärung).

4 pt

Explain how a logical address is mapped to a physical address when segmentation is used (figure + explanation).

**Total:
12 pt**

Aufgabe 5: Dateisysteme / Question 5: File Systems

- a) Welche Zugriffsrechte haben Besitzer, Gruppe und andere an der Datei `exam.tex` auf einem Unix-Dateisystem, nachdem untenstehendes Kommando erfolgreich ausgeführt wurde?

2 pt

State the respective access rights of owner, group, and others to the file `exam.tex` on a Unix file-system after the command below has been successfully executed.

```
chmod 740 exam.tex
```

- b) Erläutern Sie den Unterschied zwischen mandatory und advisory file locking.

2 pt

Explain the difference between mandatory and advisory file locking.

- c) Erklären Sie den Unterschied zwischen einem Hardlink und einem symbolischen Link.

2 pt

Explain the difference between a hard link and a symbolic link.

- d) Moderne Dateisysteme erlauben es in der Regel, dass Dateien aus beliebig auf der Festplatte verteilten Blöcken bestehen können. Warum kann es dennoch von Vorteil sein, Blöcke für eine Datei möglichst "am Stück" zu allozieren?

2 pt

Modern file systems usually allow files to consist of blocks that are arbitrarily distributed on the hard disk. Nevertheless, it might be advantageous to allocate blocks for a file contiguously. Why?

- e) Nennen Sie zwei Beispiele für zeichenorientierte Geräte (*character devices*).

1 pt

List two examples of character devices.

- f) Wofür steht das Akronym DMA? Erklären Sie kurz, wie DMA funktioniert und wozu man es verwendet.

3 pt

What is the meaning of the acronym DMA? Briefly explain how DMA works and what it is used for.

**Total:
12 pt**

Lösungen

Aufgabe 1

a) **Lösung:** 0x020

b) korrekt, inkorrekt, inkorrekt, inkorrekt, inkorrekt

c) **Lösung:**

Monolith: Gesamte Funktionalität (einschl. Treiber) in einem einzelnen Binary, d.h. keine "harte" Isolation zwischen Komponenten. **(1 P)**

Mikrokern: Nur kleine, privilegierte Code-Basis, alle Dienste/Treiber laufen isoliert und unprivilegiert, Kommunikation über Nachrichten. **(1 P)**

Nachteile eines Mikrokerns:

- Geringere Performance durch mehr Kommunikationsoverhead **(1 P)** .
- Hoher Aufwand bei Portierung von Monolith **(0.5 P)** aber insgesamt max. **(1 P)** für Nachteil.

d) **Lösung:**

- Über Register
- Auf dem Stack
- Über einen dedizierten Speicherbereich, dessen Anfangsadresse in einem Register übergeben wird (**(-0.5 P)** wenn nicht genannt wird wie man an die Adresse kommt).

Aufgabe 2

a) **Lösung:**

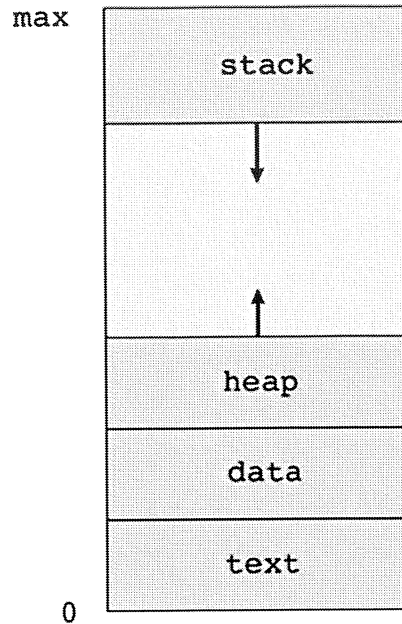


Abbildung 1: **(0.5 P)** für jeden korrekten Bereich

b) **Lösung:**

- + Schnelles Threadmanagement **(0.5 P)**
- + UL-Scheduling **(0.5 P)**
- + Leicht portierbar auf 1:1 oder M:N **(0.5 P)**
- Profitiert nicht von Mehrprozessor-Systemen **(0.5 P)**
- Blockierender Systemaufruf eines Threads blockiert alle anderen Threads der Anwendung. **(0.5 P)**
- Profiling/Debugging schwierig **(0.5 P)**

Maximal **(2 P)**

c) **Lösung:** Präemptives **(1 P)** SJF **(1 P)**

d) **Lösung:**

$$P_0: 2 - 0 = 2 \text{ **(0.5 P)**}$$

$$P_1: 6 - 0 = 6 \text{ **(0.5 P)**}$$

$$P_2: 4 - 3 = 1 \text{ **(0.5 P)**}$$

$$P_3: 8 - 5 = 3 \text{ **(0.5 P)**}$$

e) **Lösung:**

P_0, P_2 : 0 **(0.5 P)**

P_1 : 3 **(0.5 P)**

P_3 : 1 **(0.5 P)**

Durchschnitt: $\frac{0+3+0+1}{4} = \frac{4}{4} = 1$ **(0.5 P)**

f) **Lösung:**

Zentral:

- + Gleichmäßige Auslastung aller Prozessoren **(1 P)**
- Queue wird zum Flaschenhals **(1 P)**

Pro CPU:

- + "Automatische" Prozessor-Affinität **(1 P)**
- Load-Balancing nötig **(1 P)**

Maximal **(2 P)**

Aufgabe 3

a) **Lösung:**

Assume that threads T_1, \dots, T_k , $1 \leq k < 4$ execute `wait_for_others` until they are preempted right after line 18 **(1 P)**, just before putting themselves on the `sleep_list`. Further assume that threads T_{k+1}, \dots, T_3 execute `wait_for_others` until they put themselves to sleep in line 20. T_4 will find `count == 4` in line 10 and wake up all the threads that are currently on the `sleep_list` **(1 P)**, i.e., threads T_{k+1}, \dots, T_3 . The k other threads (T_1, \dots, T_k) are not (yet) on the list **(1 P)** and thus are not woken up by T_4 . As a consequence, $4 - k < 4$ threads will now enter the critical section **(1 P)**.

Scenario in violation of `mutex`: **(1 P)**.

b) **Lösung:**

Continuing the setting from part a), we have `count = 0` and eventually k threads on the `sleep_list` **(1 P)**. The next three threads increase `count` to 3 and enlarge the `sleep_list` to $k + 3$ threads **(1 P)**. The fourth thread can then wake up all $k + 3$ sleeping threads and enter the critical section, allowing for $4 + k > 4$ threads in the critical section **(1 P)**.

More than 4 threads can enter the critical section (without scenario) **(1 P)**.

If a thread T_i gets preempted right after line 19, and if the `sleep_list` is woken up before T_i has executed line 20, T_i will remain blocked forever **(2 P)**: The wakeup signal will be ignored as T_i is not sleeping, waking up all sleepers will remove them (including T_i) from the list, so that no other thread can wake up T_i **(1 P)**.

The *sleep_list* is not (explicitly) purged (**1 P**), so threads remain in the list and might be woken up at a later time – possibly when they are waiting for an unrelated event (**1 P**).

If $N \geq 8$, a second group of 4 threads can become ready and enter the critical section before the previous group has left it, thus allowing > 4 threads in the critical section (**2 P**).

If *append()* is not thread-safe (**1 P**), the *sleep_list* can become corrupted (**1 P**).

Threads may sleep forever if less than 4 threads ever want to enter the critical section (**1 P**).

The order in which the threads enter the critical section is undefined (**1 P**).

Scheduling effects might cause $n \neq 4$ to actually execute code in the critical section concurrently, even if exactly 4 threads are admitted (**0 P**). (Scheduling effects are independent of the mechanism used to synchronize threads.)

```

1      List sleep_list;
2      Semaphore mutex = 1;
3
4      int count = 0;
5
6      function wait_for_others ()
7      {
8          mutex.wait ();
9          count++;
10         if ( count == 4 )
11         {
12             count = 0;
13             wake_all( sleep_list );
14             mutex.signal ();
15         }
16         else
17         {
18             mutex.signal ();
19             sleep_list.append( myself );
20             sleep ();
21         }
22     }
23 }
```

c) **Lösung:**

R_1 is invalidly shared among $\{P_1, P_2\}$; at least one of its outgoing edges must be removed or reversed (**1 P**).

P_3 cannot request P_2 / P_3 cannot be assigned to P_2 ; there must not be any edges among processes (or resources) in a RAG (**1 P**).

d) **Lösung:**

To transform the RAG into a wait-for-graph, one must replace all pairs of edges (P_i, R_j) , (R_j, P_k) with a single edge (P_i, P_k) and remove all resource nodes (and their attached edges) from the RAG.

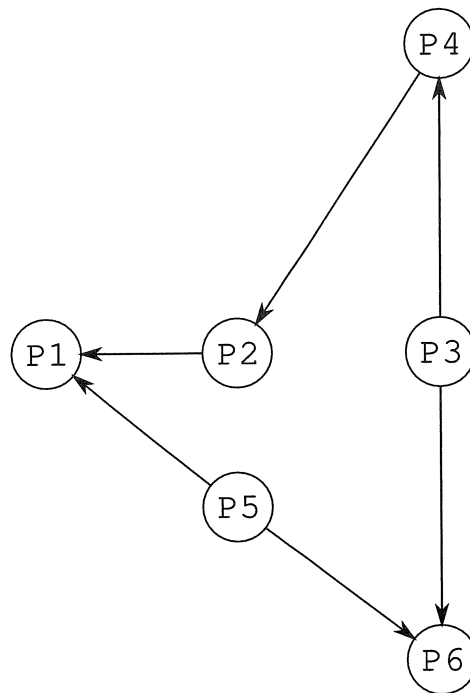


Abbildung 5: Wait-for-graph

Correct graph: **(3 P)** .

Invalid edge/edge direction/node: **(-0.5 P)** .

Aufgabe 4

- a) **Lösung:** Weil der TLB sonst die (Virt \rightarrow Phys) Abbildungen des "alten" Adressraums zurückliefern würde (und somit der Prozess im "neuen" Adressraum auf falsche Daten zugreifen würde). **(2 P)**
 (Formaler: Der TLB kann als virtuell indexierter, virtuell getaggter Cache betrachtet werden. Wird er beim Adressraumwechsel nicht invalidiert, treten Mehrdeutigkeiten (Ambiguities) auf.)

- b) **Lösung:** Unterstützung von ASIDs (address space identifier)

- c) **Lösung:**

Demand paging: Seiten werden bei Bedarf, also erst beim Auftreten eines Seitenfehlers, eingelagert. **(0.5 P)**

Pre-paging: Seiten werden "auf Verdacht" eingelagert, in der Hoffnung, dass sie zu einem späteren Zeitpunkt benötigt werden. **(0.5 P)**

Pre-paging verschwendet potentiell mehr Speicher **(0.5 P)** (da Seiten eingelagert werden können, die gar nicht gebraucht werden). Dennoch kann pre-paging von Vorteil sein, da dadurch die Zahl der auftretenden Seitenfehler reduziert und somit die Leistung verbessert werden kann **(1.5 P)** . Effizienterer Datentransfer von der Festplatte: **(1 P)** , falls als einziger Vorteil genannt, **(0.5 P) Bonus**, falls zusätzlich zur reduzierten PF-Rate.

- d) **Lösung:** copy-on-write (1 P)
- e) **Lösung:** Die Seite, auf die am längsten nicht zugegriffen werden wird. (1 P) (Der Algorithmus müsste also die Zukunft kennen und ist somit nur in der Theorie nutzbar.)
- f) **Lösung:**

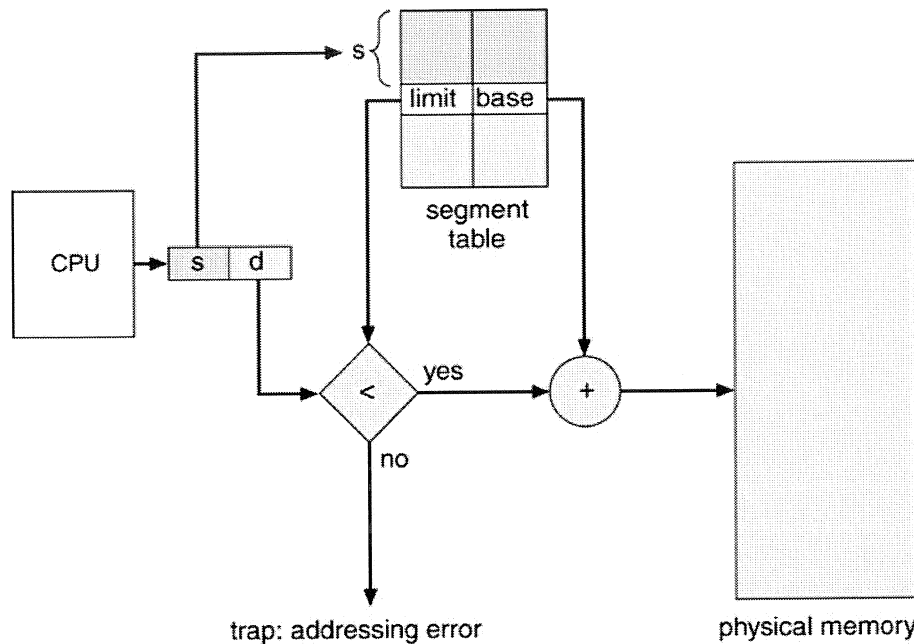


Abbildung 6: Segmentierung (2 P)

- Logische Adresse wird aufgeteilt in Segmentnummer s und Offset/Displacement d . (0.5 P)
- Die Segmentnummer dient als Index in die Segmenttabelle, dort wird der Segmentdeskriptor (base und limit) an der entsprechenden Stelle geladen. (0.5 P)
- limit wird mit dem Offset verglichen. Ist der Offset größer, so ist der Zugriff ungültig, da die resultierende Adresse außerhalb des Segments liegen würde. In diesem Fall wird eine Ausnahme ausgelöst. (0.5 P)
- Ist der Offset kleiner als das limit, so ist der Zugriff gültig, und die Basisadresse wird zum Offset d addiert. Das Resultat ist die physische Adresse, die an den Speichercontroller weitergegeben werden kann. (0.5 P)

Aufgabe 5

a) **Lösung:**

Besitzer rwx
Gruppe r--
Andere ---

Falsch beschriebene Rechte: **(-0.5 P)**

b) **Lösung:**

Mandatory: Das Betriebssystem stellt sicher, dass kein Prozess eine Datei öffnen kann, auf die bereits ein anderer Prozess ein Lock hält. **(1 P)**

Advisory: Das Betriebssystem erlaubt den gleichzeitigen Zugriff von mehreren Prozessen auf eine Datei, die Prozesse müssen sich ggf. selbst um korrekte Synchronisation kümmern. **(1 P)**

c) **Lösung:**

Hardlink: Ein (weiterer) Name für eine Datei, d.h. ein Verzeichniseintrag, der eine (schon vorhandene) Inode referenziert. **(1 P)**

Symlink: Eine spezielle Datei vom Typ "Link", die den Namen der Zieldatei enthält. **(1 P)**

d) **Lösung:**

Mechanische Festplatten erlauben vergleichsweise hohe Transferraten, wenn Blöcke sequentiell (also auf derselben Spur) gelesen werden. Die Latenz für wahlfreie Zugriffe ist dem gegenüber aber relativ hoch, da in diesem Fall der Schreib-/Lesekopf erst positioniert werden muss (seek time). **(1 P)**

Unter Annahme der Lokalitätseigenschaft (sequentieller Zugriff oder wahlfreie Zugriffe auf dieselbe Spur) **(0.5 P)** können daher zeitaufwendige Positionierungen des Schreib-/Lesekopfes vermieden werden. **(0.5 P)**

e) **Lösung:** Z.B. Tastatur **(0.5 P)** und Maus **(0.5 P)** .

f) **Lösung:**

Direct Memory Access **(0.5 P)**

Das Betriebssystem programmiert den DMA-Controller **(0.5 P)** , der die Daten vom Gerät in den Hauptspeicher überträgt (oder umgekehrt). **(0.5 P)** . Das Ende des Kopiervorgangs wird durch einen Interrupt signalisiert. **(0.5 P)**

DMA erlaubt den Datentransfer zwischen Hauptspeicher und einem E/A-Gerät, ohne die CPU zu belasten. **(1 P)**