

Musterlösung

26. 03. 2012

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 und 13 Aufgabenblättern mit insgesamt 5 Aufgaben.
The examination consists of 14 pages: One cover sheet and 13 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 multiple, 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) Schreiben Sie eine gültige Zeile C-Code um der Variablen `b` den vierten Wert des Arrays `a` zuzuweisen. Benutzen Sie für die Zuweisung den Pointer `ptr`. Benutzen Sie nicht den Index-Operator `[]`.

1 pt

Write a valid line of C-Code to assign the fourth value of Array `a` to variable `b`. Use the pointer `ptr` for the assignment. Do not use the index-operator `[]`.

```
int a[32], b, *ptr;
ptr = a;
```

- 1.0P für `b = *(ptr + 3);`
- 0.5P für `b = *(ptr + 4);`
- -0.5P bei jedem Syntaxfehler

- b) Erklären Sie, was jeweils in den folgenden Ausdrücken deklariert wird.

3 pt

Explain what is declared in the following statements, respectively.

`int (*a)();` Function pointer to a function that returns an integer.

`int *b();` Function that returns a pointer to an integer.

`int c(int d());` Function that returns an integer and takes a function pointer as a parameter. The function that is pointed to returns an integer.

- c) Was tut der folgende Assmbler-Befehl auf einer IA-32 Architektur? Wofür wird er semantisch benutzt?

1 pt

What does the following assembly command do on an IA-32 architecture? What is it used for semantically?

```
subl $16, %esp
```

Lösung:

Subtract 16 bytes from stack pointer. Semantically this is an allocation of 16 bytes on the stack.

- d) Nennen Sie 2 privilegierte IA-32 Instruktionen und erklären Sie jeweils, warum diese privilegiert sein müssen (Eine Instruktion und ihre inverse Instruktion zählen als dieselbe Instruktion).

2 pt

Give 2 privileged IA-32 instructions and explain why each one of them needs to be privileged (An instruction and its inverse instruction count as the same instruction).

Lösung:

- *CLI/STI - IRQ off/on - e.g., Timer triggers via interrupt, if user could disable interrupts, she could interfere with scheduling*
- *PUSHF/POPF - Push/Pop Flags - e.g., to set CPL, if user could set CPL, she could get kernel-mode privileges.*

e) Erklären Sie den Begriff: preemption.

1 pt

Explain the term: preemption.

Lösung:

Forcefully take away a resource from a process.

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

4 pt

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

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

- | | | |
|-------------------------------------|-------------------------------------|---|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | open und close sind privilegierte CPU Instruktionen.
<i>open and close are privileged CPU instructions.</i> |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | Die MMU muss bei jedem Load/Store Zugriff eines User Level Prozesses Virtuelle Adressen in Physische Adressen übersetzen.
<i>The MMU needs to translate virtual addresses to physical addresses on every load/store access in a user level process.</i> |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | Wenn wir schnellen, günstigen, großen, kompakten, energieeffizienten und nicht-flüchtigen Speicher hätten, bräuchten wir keine Cache-Hierarchy.
<i>If we had fast, cheap, large, compact, energy efficient, and non-volatile memory, there would be no need for a cache hierarchy.</i> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | Belady's Anomalie kann auftreten, wenn die LRU Seitenersetzungsstrategie genutzt wird.
<i>Belady's Anomaly may occur when using the LRU page replacement strategy.</i> |

**Total:
12 pt**

Aufgabe 2: Prozesse und Threads / *Assignment 2: Processes and Threads*

- a) Nennen Sie vier verschiedene Metriken/Kriterien, welche zur Beurteilung eines Scheduling-Verfahrens herangezogen werden können.

2 pt

Name four different metrics/criteria that can be used to estimate the quality of a scheduling policy.

Lösung:

- *Processor utilization*
- *Application Throughput*
- *Turnaround time*
- *Waiting time*
- *Response time*
- *...*

- b) Diskutieren Sie Vor- und Nachteile einer langen gegenüber einer kurzen Zeitscheibenlänge.

2 pt

Discuss pros and cons of long versus short time slice lengths.

Lösung:

The longer a timeslice is, the fewer context switches per time-unit are required. As context switches require a certain overhead (entering the kernel, selecting a new process to run, switching address spaces if necessary, etc.), reducing the number of context switches can help to improve the throughput. However, making time slices too long will negatively affect the responsiveness of the system: Each process will run for a relatively long time until it will be preempted, and consequently ready processes have to wait longer until they are allowed to run again.

- c) Gegeben seien ein Lottery-Scheduler auf einem Uniprozessorsystem mit der Prozessliste $\{P_1, P_2, P_3\}$. P_1 habe 2 Tickets, P_2 habe 5 Tickets, P_3 habe 1 Ticket. Alle Prozesse sind permanent lauffähig.

Vervollständigen Sie den untenstehenden Plan (10 Kreuze). Die erste Zeile (TIME) gibt die Zeit an. Eingeplant wird jeweils beim Wechsel der Zeiteinheit. Um die Lotterie deterministisch zu gestalten, enthält die zweite Zeile (LOTTERY) das vorgegebene Lotterie-Ergebnis zum jeweiligen Anfang der Zeiteinheit.

3 pt

Consider a lottery scheduler on a uniprocessor system with process list $\{P_1, P_2, P_3\}$. P_1 has 2 lottery tickets, P_2 has 5 lottery tickets, and P_3 has 1 ticket.

Complete the scheduling plan given below (10 crosses). The first row (TIME) states the time. Scheduling occurs at time unit changes. In order to ensure deterministic results, the second row (LOTTERY) contains a pre-defined lottery result for each time unit start.

<i>TIME</i>	0	1	2	3	4	5	6	7	8	9
<i>LOTTERY</i>	4	5	0	0	1	3	2	4	1	6
P_1										
P_2										
P_3										

Lösung:

<i>TIME</i>	0	1	2	3	4	5	6	7	8	9
<i>LOTTERY</i>	4	5	0	0	1	3	2	4	1	6
P_1			X	X	X				X	
P_2	X	X				X	X	X		X
P_3										

- d) Lottery-Scheduling kann insofern als fair bezeichnet werden, als es im Durchschnitt jedem Prozess proportional zu den Tickets Zeit zuteilt. Nehmen Sie an, Sie möchten auf einem Prozessor eine im Hintergrund ablaufende Kernel-Kompilierung (Prozess P_4) gleichzeitig mit einer Videokonferenzanwendung (Prozess P_5) ausführen lassen. P_4 habe permanenten Rechenbedarf, während P_5 nie die volle Zeitscheibenallokation ausnutzt. Bleibt Lottery-Scheduling fair, wenn Sie den beiden Prozessen dieselbe Anzahl Tickets zuteilen? Warum/Warum nicht?

2 pt

Lottery scheduling can be termed fair insofar, as it will allocate, on average, processing time to each process in proportion to its respective number of tickets. Suppose you want to execute, on a single processor, a kernel compilation job in background (process P_4) together with a video conferencing application (P_5). P_4 requires the processor at any time, while P_5 never eats up its time slice completely. Will Lottery Scheduling stay fair (according to previous definition) if you allocate the same number of tickets to both P_4 and P_5 ? Why / why not?

Lösung:

No. While both processes will win the same number of lotteries on average, P_3 will receive only a fraction of the processing time of P_4 .

- e) Erklären Sie kurz das Scheduling-Verfahren: Multi-Level Feedback Queue (MLFB) mit variablen Zeitscheibenlängen.

3 pt

Briefly explain the scheduling policy: Multi-Level Feedback Queue (MLFB) with variable time slices.

Erklären Sie kurz, ob MLFB ein geeignetes Verfahren in den folgenden Fällen ist:

Briefly discuss whether MLFB is a good scheduler in the following cases:

In einem System, in welchem die Kosten für Kontextwechsel hoch sind.

In a system where context switching costs are high.

In einem System, in welchem Aushungern (Starvation) vermieden werden sollte.

In a system where starvation should be avoided.

Lösung:

MLFB: Multiple queues with individual priorities and time slice. Typically, the lower the prio, the larger the timeslice. Interactive processes have high-prio and small time-slice. CPU-bound processes that eat up all CPU time go to a lower-priority queue with higher time slice.

MLFB is a good choice when context switching costs are high as it only requires frequent switches with short jobs but avoids switching contexts with long-running jobs.

In MLFB, many short-running jobs may starve lower-priority jobs. However, this can be avoided by moving processes that have been waiting too long into higher-priority queues.

**Total:
12 pt**

Aufgabe 3: Koordination von Prozessen / Assignment 3: Process Coordination

a) Was sind die 4 notwendigen Bedingungen für einen Deadlock?

2 pt

What are the 4 necessary conditions for a deadlock?

Lösung:

- *Mutual exclusion (+0.5P)*
- *Hold and wait (+0.5P)*
- *No preemption (+0.5P)*
- *Circular wait (+0.5P)*

b) Gegeben sei ein System mit 2 Threads, die 3 Semaphoren zum Schutz kritischer Abschnitte verwenden. Thread i führe jeweils die Funktion `thread_i()` aus Listing 2 (auf der nächsten Seite) aus. In dem System gebe es keine weitere Aktivität. Geben Sie für jede der Bedingungen für Deadlocks an, ob diese beim Zugriff auf die Semaphoren in diesem System erfüllt ist und begründen Sie jeweils Ihre Antwort.

5 pt

Given a system with 2 threads that use 3 semaphores to protect critical sections. Each thread i executes function `thread_i()` in Listing 2 (on the following page) respectively. There is no other activity in the system.

For each condition for deadlocks, state whether it is fulfilled or not regarding the use of semaphores in this system, and explain your answer.

Können Deadlocks in diesem System auftreten? Begründen Sie Ihre Antwort.

Can deadlocks occur in this system? Explain.

Lösung:

- *Mutual exclusion: Yes (+0.5p), critical sections are exclusive; the semaphores are used to protect them. (+0.5P)*
- *Hold and wait: Yes (+0.5P), critical sections are nested / threads wait() on semaphores to acquire them as locks after they have wait()ed on other semaphores. (+0.5P)*
- *No preemption: Yes (+0.5P), threads leave critical sections only voluntarily / no one besides the involved threads can signal the semaphores (+0.5P).*
- *Circular wait. No (+0.5P); the order of resource allocation (wait()s on semaphores) does not lead to cycles in the RAG / there is a resource ordering policy in place / both threads wait() on semaphores in the order S2, S3. (+0.5P)*

Deadlocks cannot occur in this system (+0.5P) because one of the necessary conditions for a deadlock is not fulfilled (+0.5P).

Listing 2:

```
Semaphore S1, S2, S3;
void thread_1()
{
    wait( S1 );
    wait( S3 );
}
void thread_2()
{
    wait( S3 );
    wait( S2 );
}
```

```

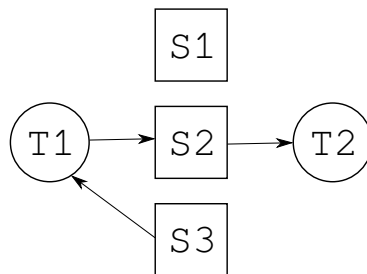
signal( S1 );           signal( S3 );
wait( S2 ); //X        //X
signal( S3 );          signal( S2 );
signal( S2 );
}                       }
    
```

- c) Zeichnen Sie den Resource Allocation Graph (RAG) für die Threads aus Listing 2 mit den Semaphoren als Ressourcen, zu dem Zeitpunkt, an dem die Threads an den mit X markierten Stellen in Ihrer Ausführung unterbrochen wurden.

3 pt

Draw the Resource Allocation Graph (RAG) for the threads from Listing 2 with the semaphores as resources, at the moment the threads have been interrupted at the position marked with an X.

Lösung:



- d) Welche Hardware-Mechanismen werden typischerweise auf Einprozessorsystemen zur Implementierung von Synchronisation im Betriebssystem verwendet, welche auf Mehrprozessorsystemen?

1 pt

Which hardware mechanisms are typically used on single-processor systems to implement synchronization in the operating system? Which mechanisms are used on multi-processor systems?

Lösung:

- *Uniprocessor: Disabling interrupts (+0.5P).*
- *Multiprocessor: Atomic instructions / inter-processor interrupts (+0.5P, one is sufficient)*

- e) Warum greift `testAndSet` auf den Hauptspeicher und nicht auf ein weiteres Register zu? Begründen sie Ihre Antwort mit einem Szenario, wofür `testAndSet` typischerweise eingesetzt wird.

1 pt

Why does `testAndSet` access main memory and not another register instead? Explain using a scenario where `testAndSet` is typically used.

Lösung:

The register set is private to a thread and CPU-local (+0.5P). `testAndSet` is typically used for synchronization between several CPUs, thus it needs to be visible across threads and across CPUs (+0.5P).

**Total:
12 pt**

Aufgabe 4: Speicher / *Assignment 4: Memory*

Vergleichen Sie die zwei Szenarien reine Segmentierung und reines Paging (mit einer Seitengröße von 4 KiB). Betrachten Sie hierzu das Beispiel eines Programms mit 15 KiB Text-Segment, 5 KiB nur lesbaren Daten und 4 KiB Stack.

Compare the two scenarios pure segmentation and pure paging (with 4 KiB page size). For this purpose, consider the example of a program with 15 KiB text segment, 5 KiB read-only data, and 4 KiB stack segment.

- a) Der physische Speicher sei zunächst leer. Dann wird das beschriebene Programm geladen. Die Segmente werden hierzu in der gegebenen Reihenfolge hintereinander im physischen Speicher abgelegt (beginnend bei Adresse 0). An welchen physischen Adressen beginnen Text-Segment, Daten und Stack in den beiden Szenarien?

3 pt

Assume the physical memory is empty at first. Then, the program we described above is loaded. The segments are placed in physical memory consecutively in the given order (starting at address 0).

What are the physical starting addresses of text segment, data, and stack in the two scenarios respectively?

Segment	Startadresse mit Segmentierung <i>Starting address with segmentation</i>	Startadresse mit Paging <i>Starting address with paging</i>
Text Segment	0 KiB	0 KiB
Daten (<i>data</i>)	15 KiB	16 KiB
Stack Segment	20 KiB	24 KiB

+0.5P for every correct start address (i.e., per table entry).

- b) Welche Arten von Fragmentierung wurden in der Vorlesung vorgestellt?

1 pt

Which kinds of fragmentation have been introduced in the lecture?

Lösung: *Internal fragmentation (+0.5P) and external fragmentation (+0.5P).*

- c) Tritt in der Speicherbelegung aus Teilaufgabe a) Fragmentierung auf? Wenn ja, welche Art und in welchem Umfang?

2 pt

Is there fragmentation in the memory allocation from question a)? If yes, state the type and the amount of fragmentation.

Segmentierung / *Segmentation:*

Lösung: *no fragmentation (+1P).*

Paging:

Lösung: *Internal fragmentation (+0.5P) of 4 KiB in total (+0.5P).*

d) Komplettieren Sie die folgende einstufige Pagetable für den Start-Zeitpunkt des oben beschriebenen Programms gemäß den folgenden Kriterien:

- Ein Prozess nutzt den virtuellen Adressraum im Bereich von 0 bis 3 GB. Text-Segment und Daten werden am Anfang, der Stack am Ende dieses Bereichs platziert.
- Zugriffsrechte seien so restriktiv wie möglich.
- Das Betriebssystem verwende on-demand paging für den Stack und pre-paging für den Programm-Code und Daten. Physische Seiten werden in der Reihenfolge der Pagetable-Einträge zugewiesen, beginnend bei Adresse 0.
- Markieren Sie Einträge, deren Inhalt nicht verwendet wird, mit einem Bindestrich (-). Markieren Sie Bits als gesetzt (1), nicht gesetzt (0), oder beliebig (-).

4 pt

Complete the following single-level pagetable for the start time of the program described above, according to these guidelines:

- *A process utilizes the virtual address space from 0 to 3 GB. Text and data are placed at the beginning, the stack at the end of that range.*
- *Access rights have to be as constraint as possible.*
- *The operating system uses on-demand paging for the stack and pre-paging for program code and data. Physical pages are assigned in the order of the pagetable entries, beginning at address 0.*
- *Mark unused entries with a dash (-); mark bits as set (1), reset (0), or arbitrary (-).*

Index	Physical Frame Number	read	write	execute	valid
0	0	1	0	1	1
1	1	1	0	1	1
2	2	1	0	1	1
3	3	1	0	1	1
4	4	1	0	0	1
5	5	1	0	0	1
6	-	-	-	-	0
...	...				
786430	-	-	-	-	0
786431	-	-	-	-	0
...	...				

- text has assigned frame numbers 0 to 3. (+0.5)
- data has assigned frame numbers 4 and 5. (+0.5)
- stack has no assigned page number. (+0.5)
- text and data have the valid bits set. (+0.5)
- stack has its valid bit unset. (+0.5)
- pt entry 6 has - - - - 0. (+0.5)
- text has permissions r-x. (+0.5)
- data has permissions r- -. (+0.5)

- e) Welches Ereignis geschieht beim ersten Zugriff eines Prozesses auf seinen Stack, wenn das Betriebssystem on-demand paging verwendet?

2 pt

Which event happens the first time a process accesses its stack if the operating system uses on-demand paging?

Wie reagiert das Betriebssystem darauf, damit der Prozess anschliessend weiterläuft?

How does the operating system react to this event, in order that the process continues to run afterwards?

Lösung:

The CPU/MMU generates a **pagefault** (+0.5P). The operating system then

- *assigns physical memory / a physical page for the page just accessed, (+0.5P)*
- *updates the pagetable of the process, setting the valid bit of the faulting page, (+0.5P)*
- *and continues the execution of the process by restarting the faulting instruction. (+0.5P)*

**Total:
12 pt**

Aufgabe 5: Hintergrundspeicher / Assignment 5: Secondary Storage

- a) Sie erstellen eine Datei *a*, einen Hardlink *b* auf diese Datei und einen symbolischen Link auf *a*, den Sie *c* nennen. Was sind die Dateirechte von *a*, *b* und *c*, in *rwX* Notation, nach allen folgenden Shell-Befehlen?

3 pt

You create a file a, a hardlink b to this file, and a symbolic link to a, called c. What are the file permissions of a, b, and c in rwX notation, after executing all of the following shell commands?

```
chmod 670 a
chmod 123 b
chmod 456 c
```

Datei/File	Dateirechte in rwX Notation/File permissions in rwX notation
a	--X -W- -WX
b	--X -W- -WX
c	r-- r-X rW-

- b) Was ändert sich an *b* und *c*, wenn *a* gelöscht wird?

1 pt

How do b and c change when a is deleted?

Lösung:

- *b* Link counter is decremented
- *c* No change, only link target disappears

- c) Was ist anonymer Speicher? Nennen Sie zwei mögliche Quellen.

2 pt

What is anonymous memory? Enumerate two possible sources.

Lösung:

Anonymous memory is memory that is not backed by a file. E.g., the heap or the stack segment of a process.

- d) Wie wird benannter (nicht-anonymer) Speicher beim Swapping behandelt?

1 pt

How is named (non-anonymous) memory treated when swapping?

Lösung:

When swapping, named memory may be dropped if it has not been modified. Otherwise it is written back to the original file instead of being written to the swap space.

- e) Mit dem Programm `fsck` können unter Unix Dateisystemfehler gefunden und behoben werden. Geben Sie für die folgenden Fehlermeldungen an, was `fsck` im inkonsistenten Dateisystem gesehen haben könnte um auf den Fehler zu schließen.

2 pt

Unix file systems can be checked and fixed with the `fsck` program. What could `fsck` have seen to generate the following error messages in an inconsistent file system.

Der Inode Linkzähler ist 1, er sollte 2 sein.

The inode link count is 1, it should be 2.

Lösung:

A second directory entry has been found to an inode with link count 1.

Der Frei-Bitmap Eintrag für Block 1234 ist 0 (frei), er sollte 1 sein (belegt).

The free-bitmap entry for block 1234 is 0 (free), it should be 1 (allocated).

Lösung:

An inode entry points to a block which is not marked to be allocated.

- f) Wir haben in der Vorlesung drei Dateiallokationsstrategien kennengelernt: (a) Indexed allocation, (b) chained allocation und (c) contiguous allocation. Geben Sie für die folgenden Szenarien an, wie gut die Strategien jeweils geeignet sind und warum.

3 pt

We have described three file allocation strategies in class:

(a) Indexed allocation, (b) chained allocation, and (c) contiguous allocation. Describe how well the three strategies perform in each one of the following scenarios. Explain why.

Schneller wahlfreier Zugriff auf sehr große Dateien.

Fast random access to very large files.

Lösung:

Contiguous allocation: *Am besten, sehr einfache Index-Berechnung ohne indirekte Zugriffe*

Indexed allocation: *Sehr gut, einfaches Finden des richtigen Blocks mithilfe des Indizes.*

Chained allocation: *Schlecht, lineares durchsuchen der Kette zum Finden des korrekten Blocks.*

Ausnutzung der Festplattenkapazität (Speichern möglichst vieler Datei-Bytes).

Utilization of the disk capacity (storage of as many file-bytes as possible).

Lösung:

Chained allocation: *Sehr gut, keine externe Fragmentierung, kleine Datenstrukturen.*

Indexed allocation: *Sehr gut, keine externe Fragmentierung, minimal größere Datenstrukturen als bei Chained allocation.*

Contiguous allocation: *Schlecht, entweder externe Fragmentierung oder compaction nötig.*

**Total:
12 pt**