

Nachname/*Last name*

Vorname/*First name*

Matrikelnr./*Matriculation no*

Nachklausur 18. 09. 2012

- 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 13 Blättern: Einem Deckblatt und 12 Aufgabenblättern mit insgesamt 5 Aufgaben.
The examination consists of 13 pages: One cover sheet and 12 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 die Nummer der virtuellen Speicherseite zu berechnen, auf die der Zeiger `addr` zeigt. Verwenden Sie ausschließlich Bitoperationen (`<<`, `>>`, `&`, `|`, `^`, `~`). Nehmen Sie eine Seitengröße von 4 KiB an.

1 pt

Write one valid line of C-code to calculate the virtual page number that the pointer `addr` points to. Use only bit operators (`<<`, `>>`, `&`, `|`, `^`, `~`). Assume a page size of 4 KiB.

```
page_num =
```

- b) Schreiben Sie eine gültige Zeile C-Code, um den Offset der Adresse zu berechnen, auf die der Zeiger `addr` zeigt (innerhalb der virtuellen Seite `page_num`). Verwenden Sie ausschließlich Bitoperationen (`<<`, `>>`, `&`, `|`, `^`, `~`). Nehmen Sie eine Seitengröße von 4 KiB an.

1 pt

Write one valid line of C-code to calculate the offset of the address that pointer `addr` points to (within the virtual page `page_num`). Use only bit operators (`<<`, `>>`, `&`, `|`, `^`, `~`). Assume a page size of 4 KiB.

```
offset =
```

- c) Erklären Sie, was jeweils in den folgenden Ausdrücken deklariert wird. Geben Sie außerdem an, wieviel Speicherplatz (in Byte) die deklarierte Struktur auf einem 64-Bit-System jeweils benötigt.

2 pt

Explain what is declared by each of the following statements. Furthermore, state how much memory (in bytes) each declared structure consumes on a 64-bit system.

```
void *(*v)(void * v);
```

```
void *v[8];
```

d) Was berechnet der folgende C-Code? Nehmen Sie an, dass *a* vom Typ `uint32_t` ist.

1 pt

*What does the following C-code calculate? Assume that *a* has the type `uint32_t`.*

```
for( uint32_t i = 0; a & 1; ++i )
    a >>= 1;
return i;
```

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

7 pt

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

korrekt/ <i>correct</i>	inkorrekt/ <i>incorrect</i>	
----------------------------	--------------------------------	--

- | | | |
|--------------------------|--------------------------|--|
| <input type="checkbox"/> | <input type="checkbox"/> | Beim Lottery Scheduling ist garantiert, dass jeder teilnehmende Faden nach endlicher Wartezeit eine CPU-Zeitscheibe zugeteilt bekommt.
<i>Lottery scheduling guarantees that each participating thread receives a CPU time slice after a finite waiting time.</i> |
| <input type="checkbox"/> | <input type="checkbox"/> | Ausnahmen werden vom Betriebssystem ausgelöst, wenn ein Benutzerprogramm eine unzulässige Instruktion ausgeführt hat.
<i>Exceptions are raised by the operating system when a user program executed an illegal instruction.</i> |
| <input type="checkbox"/> | <input type="checkbox"/> | Beim Ausführen vieler I/O-Operationen in kurzer Zeit lässt sich durch Einsatz von Polling anstelle von Interrupts ein höherer Durchsatz erreichen.
<i>When executing many I/O-operations in a short timeframe, the use of polling instead of interrupts leads to a higher throughput.</i> |
| <input type="checkbox"/> | <input type="checkbox"/> | Wird ausschließlich Paging mit fester Seitengröße verwendet, um virtuellen Speicher zu implementieren, kann im physischen Adressraum externe Fragmentierung auftreten.
<i>When using only paging with a fixed page size to implement virtual memory, there can be external fragmentation in the physical address space.</i> |
| <input type="checkbox"/> | <input type="checkbox"/> | <code>malloc</code> ist ein Systemaufruf, mit dem Prozesse Speicher vom Betriebssystem anfordern können.
<i>malloc is a system call that processes can use to request memory from the operating system.</i> |
| <input type="checkbox"/> | <input type="checkbox"/> | Im Kernadressraum gibt es keine Seitenfehler.
<i>There are no page faults in the kernel address space.</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> |

**Total:
12.0pt**

Aufgabe 2: Koordination von Prozessen / Assignment 2: Process Coordination

Betrachten Sie das Code-Beispiel in Listing 2. Stellen Sie sich ein System vor, in dem zwei Threads parallel die Funktion `worker_thread()` ausführen.

Consider the code example in Listing 2. Imagine a system where two threads simultaneously execute the function `worker_thread()`.

```
1 int data[SIZE]; // SIZE be defined above, its actual value does not matter
2 int pos = 0;
3
4 void worker_thread ( void )
5 {
6     while( pos < SIZE ) {
7         int entry = get_data_from_somewhere();
8         data[pos] = entry; // enter new entry
9         pos++;
10    }
11 }
```

- a) Beschreiben Sie einen Ablauf, in dem ein Thread auf Speicher jenseits der Grenzen des Arrays zugreift.

2 pt

Describe a flow of execution in which one thread accesses memory beyond the bounds of the array.

- b) Beschreiben Sie einen Ablauf, der zum Verlust eines Datums führt.

2 pt

Describe a flow of execution that leads to a data entry being lost.

- c) Wie nennt man ein solches Fehlverhalten von Software? Warum treten solche Fehler manchmal auf, manchmal nicht?

1 pt

How do we call such errors in software? For what reason do they sometimes occur, sometimes not?

- d) Was verstehen wir unter *kritischen Abschnitten* im Kontext von Betriebssystemen? Beschreiben Sie den Begriff kurz.

1 pt

What are critical sections in the context of operating systems? Briefly explain that term.

- e) Enthält der Code im Listing 2 (auf der vorherigen Seite) kritische Abschnitte? Falls ja, geben Sie die Zeilennummern des Codes in den kritischen Abschnitten an.

2 pt

Does the code in Listing 2 contain critical sections? If yes, give the line numbers of the code comprising critical sections.

- f) Nennen und erläutern Sie kurz die drei notwendigen Bedingungen für eine gültige Lösung des Problems kritischer Abschnitte.

3 pt

Enumerate and briefly explain the three requirements for a valid solution of the critical section problem.

- g) Nennen Sie einen Mechanismus, mit dem man einen Mutex für ein Multiprozessorsystem implementieren kann. Ist dieser nur im Betriebssystem-Kern oder auch im User-Space nutzbar? Begründen Sie kurz, warum.

1 pt

Give a mechanism that can be used to implement a mutex for a multiprocessor system. Can this mechanism be used only in the OS kernel or in user-space as well? Briefly state why.

**Total:
12.0pt**

Aufgabe 3: Hintergrundspeicher / *Assignment 3: Secondary Storage*

- a) Tragen Sie in der folgenden Tabelle jeweils die Oktalzahl ein, die das jeweilige Zugriffsrecht/Spezielle Flag repräsentiert.

3 pt

In the following table, enter the octal representation of the respective access right/special flag.

SGID	
Write	
Execute	
Sticky	
Read	
SUID	

- b) Sie sollen die Zugriffsrechte für ein geteiltes Verzeichnis `work` einer Arbeitsgruppe `project` festlegen. Alle Gruppenmitglieder sollen in `work` Dateien auflisten, lesen, schreiben und anlegen können, ansonsten soll aber niemand Zugriff auf das Verzeichnis haben. Die korrekten Rechte sollen auch automatisch für neu angelegte Dateien gesetzt werden.

2 pt

Geben Sie in Oktalnotation die Werte für die Zugriffsrechte und Benutzermaske sowie je eine kurze Erklärung an!

You need to set access rights for a shared directory `work` for your work group `project`. Allow all group members to list, read, write and create files, while disallowing access from everyone else. Access rights to newly created files should be set correctly automatically.

In octal notation, provide the values for the access rights and user mask as well as a short explanation.

`chmod _____ work`

`umask _____`

- c) Unter welchen Umständen ist eine Free-List eine bessere Datenstruktur als eine Bitmap, um freie Blöcke eines Dateisystems zu verwalten?

1 pt

Under what circumstances is a free list a better choice than a bitmap for tracking free blocks in a file system?

- d) In C/Unix kann man Dateihandles offener Dateien an Funktionen übergeben. Nehmen Sie an, Sie schreiben eine Funktion, die ein offenes Dateihandle als Argument übergeben bekommt. Ist es einfach möglich, den Dateinamen der offenen Datei zu bestimmen? Wie bzw. Warum?

2 pt

In C/Unix you can pass file handles of open files to functions. Assume you are writing a function that receives an open file handle as an argument. Is it easily possible to find out the file name of the open file? How or Why?

- e) Erklären Sie für die folgenden Daten, wo/wie diese in einem Unix Dateisystem gespeichert werden!

4 pt

Explain where/how the following data is stored in a Unix file system!

File name:

Name of containing Directory:

File size:

Number of Symbolic Links:

**Total:
12.0pt**

Aufgabe 4: Speicher / Assignment 4: Memory

Betrachten Sie ein System, das mittels einer zweistufigen, hierarchischen Seitentabelle virtuelle in physische Speicheradressen übersetzt. Jede Stufe besteht aus 1024 Einträgen. Sowohl der virtuelle als auch der physische Adressraum sind 4 GiB groß, wobei jede Seite 4096 Bytes umfasst.

Consider a system that translates virtual addresses to physical addresses using two-level, hierarchical page tables. Every level comprises 1024 entries. The size of both the virtual and the physical address spaces is 4 GiB. The page size is 4096 bytes.

- a) Zerlegen Sie die virtuelle Adresse `0xD4008A40` wie es für eine Adressübersetzung im gegebenen System notwendig ist. Geben Sie dabei für jeden Teil an, wozu er dient und wie viele Bits er umfasst.

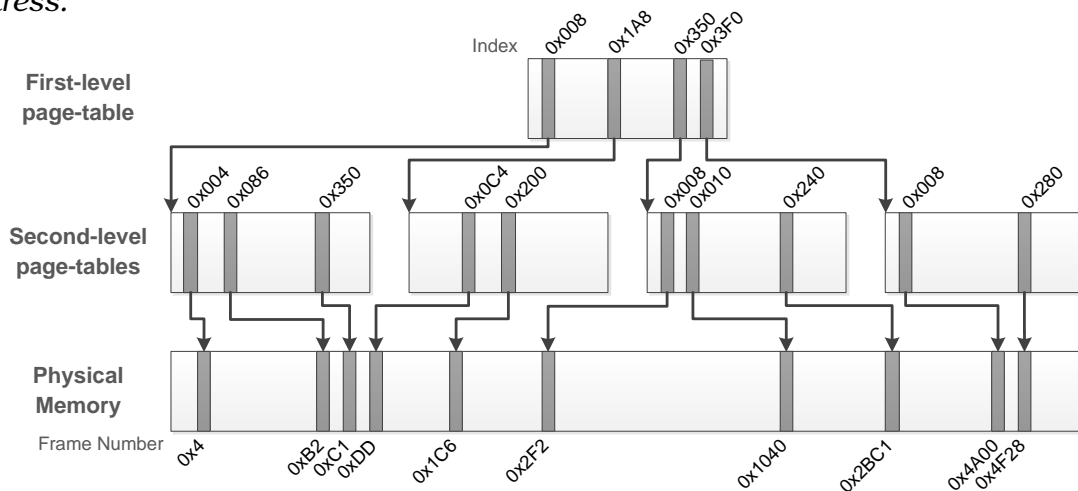
3 pt

Split the virtual address `0xD4008A40` into the parts that are necessary for an address translation in the given system. For every part, denote its purpose and state its size in bits.

- b) Führen Sie mit Hilfe der gegebenen Seitentabellenhierarchie eine manuelle Übersetzung der aus Aufgabe a) bekannten Adresse durch und geben Sie die entsprechende physische Adresse an.

1 pt

Based on the given page table hierarchy, perform a manual address translation of the virtual address from question a) and calculate the corresponding physical address.



Seitentabellenhierarchie / Page table hierarchy

- c) Wie kann in einem System mit virtuellem Speicher und Demand Paging die Anzahl von Seitenfehlern durch das Hinzufügen von Pre-Paging reduziert werden?

1 pt

How can pre-paging reduce the number of page faults in a system with virtual memory and demand paging?

- d) Ist es möglich, dass es durch Pre-Paging zu mehr Seitenfehlern kommt? Wenn ja, wie? Wenn nicht, was verhindert dies?

1 pt

Is it possible that pre-paging leads to more page faults? If yes, how? If not, what prevents this?

- e) Erklären Sie kurz, was man unter interner und externer Fragmentierung versteht. Geben Sie jeweils ein Beispiel an, wo diese auftreten können.

3 pt

Shortly explain the terms internal and external fragmentation. For each one, give an example where it may occur.

Intern/Internal:

Extern/External:

- f) In der Vorlesung wurde eine Methode zur Reduktion von externer Fragmentierung vorgestellt. Um welches Verfahren handelt es sich? Eignet sich dieses Verfahren, um externe Fragmentierung im virtuellen Adressraum eines laufenden Prozesses zu verringern? Begründen Sie Ihre Antwort!

1 pt

In the lecture, a method has been presented that is capable of reducing external fragmentation. How is this method called? Is this technique suitable to reduce external fragmentation in the virtual address space of a running process? Justify your answer!

- g) Bei Copy-On-Write werden die gleichen physischen Seiten (z.B. von Programmbibliotheken) in die Adressräume mehrerer Prozesse eingeblendet. Um die Speicherisolation zwischen den Prozessen zu gewährleisten, wird bei Schreibvorgängen auf diese Seiten durch das Betriebssystem eine private Kopie angelegt. Wie erkennt das Betriebssystem solche Schreibvorgänge?

1 pt

With copy-on-write, the same page frames (e.g. from libraries) are mapped into the virtual address spaces of multiple processes. To guarantee that the memory isolation between the processes remains intact, the operating system creates a private copy of pages on write operations. How does the operating system detect such write operations?

- h) Für welche Speicherbereiche ist es typischerweise nötig, das Caching durch den Prozessor zu unterbinden?

1 pt

Which memory areas typically require that caching through the processor is disabled?

**Total:
12.0pt**

- c) Gegeben seien die untenstehenden fünf Prozesse auf einem Uniprozessorsystem mit den angegebenen Ankunftszeiten (0 = Start), Burst-Zeiten und Prioritäten (1 = höchste Priorität). Vervollständigen Sie die untenstehenden Scheduling-Pläne für die Strategien Non-Preemptive Priority, PSJF und RR. Ein Kasten im Zeitplan stellt eine Zeitscheibe dar.

3 pt

Consider the five processes given below on a uniprocessor system, with given arrival times (0 = start), burst times, and priorities (1 = highest priority). Complete the scheduling plans given below, for the policies Non-Preemptive Priority, PSJF, and RR. A box in the scheduling plan represents one time slice.

Process	Arrival Time	Burst-Time	Priority
1	1.5	8	1
2	0	2	4
3	0.5	4	4
4	2.4	1	3
5	2.6	5	2

Non-Preemptive Priority

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

PSJF

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

RR

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

- d) Berechnen Sie für den obigen *Non-Preemptive-Priority-Plan* die durchschnittliche Wartezeit.

1.5 pt

Calculate, for the above non-preemptive priority plan, the average waiting time.

- e) Berechnen Sie für den obigen *PSJF-Scheduling-Plan* die durchschnittliche Turn-around-Zeit.

1.5 pt

Calculate, for the above PSJF scheduling plan, the average turnaround time.

f) Gegeben sei ein Uniprozessorsystem, auf welchem zehn I/O-bound Threads und ein CPU-bound Thread laufen. Die I/O-bound Threads stoßen einmal pro Millisekunde an CPU-Berechnungen eine I/O-Operation an, und jede dieser I/O-Operationen benötige 10 Millisekunden zur erfolgreichen Bearbeitung. Die Kosten eines Kontextwechsels betragen ferner 0,1 Millisekunden, und alle Threads seien langlebig. Berechnen Sie den CPU Overhead in Prozent bei Benutzung eines Round-Robin Schedulers in den folgenden Fällen:

- (a) Eine Zeitscheibe betrage 1 Millisekunde
- (b) Eine Zeitscheibe betrage 10 Millisekunden

(Angaben in Bruchform genügen)

3 pt

Consider a uniprocessor system running ten I/O-bound threads and one CPU-bound thread. Assume that each I/O-bound thread issues an I/O operation once for every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context switching overhead is 0.1 millisecond and that all threads are long-running. What is the CPU overhead due to context switching in percent when using a round-robin scheduler in the following cases:

- (a) The time quantum is 1 millisecond*
- (b) The time quantum is 10 milliseconds*

(Stating results as fractions is sufficient)

**Total:
12.0pt**