Prof. Dr. Frank Bellosa
Dipl.-Inform. Marius Hillenbrand
Dipl.-Inform. Marc Rittinghaus

Musterlösung

24. 09. 2014

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 12 Blättern: Einem Deckblatt und 11 Aufgabenblättern mit insgesamt 5 Aufgaben.
  The examination consists of 12 pages: One cover sheet and 11 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) Was sind Zombies und Waisen im Betriebssytem-Kontext? **2 pt**

What are Zombies and Orphans in the context of operating systems?

Lösung:

Zombies: Child exited but no one collected its exit status, yet.
Zombies (windows): Process exited but process context not cleaned up yet. **(1 P)**
Orphan: Parent exited and child does not have a living parent anymore. **(1 P)**

b) Schreiben Sie einen kurzen, syntaktisch korrekten C Codeabschnitt, in welchem D, S, T und ein Array H jeweils in bestimmten Addressraumsegmenten allokiert werden: D im Datensegment, H auf dem Heap, S im Stacksegment und T im Textsegment. **4 pt**

Write a short piece of syntactically correct C code that allocates D, S, T, and an array H in particular address space segments: D in the data segment, H on the heap, S in the stack segment, and T in the text segment.

Lösung:

(1 P) *each*

```c
int D = 23;
int T()
{
        int S;
        int *H = malloc( 42 * sizeof(int) );
}
```

c) Warum kann ein Spinlock vollständig im Benutzermodus implementiert werden? Was benötigt die Implementierung? **1 pt**

Why can a spinlock be fully implemented in user space? What does the implementation require?

Lösung:

A spinlock only requires an atomic swap instruction. Today, CPUs implement such instructions in userspace because memory accesses, such as swapping two memory locations (or a memory location and a register) does not require special privileges. Only saying no privileged instructions: **(0 P)** *as this is always the case for any instruction executed in userspace.*

d) Warum ergibt es keinen Sinn einen Spinlock mithilfe von zwei Registern zu implementieren anstatt Hauptspeicher zu nutzen? **1 pt**

Why does it not make sense to implement a spinlock with two registers instead of using memory?

Lösung:

Every thread has (virtually) its own set of registers.

e) Wie verhindert das Betriebssystem, dass Prozess A auf den Hauptspeicher von Prozess B zugreifen kann? **1 pt**

How does the operating system prevent process A from accessing the memory contents of process B?

Lösung:

Every Process has it's own AS. If you cannot name it, you cannot touch it.
Saying that the CPU will trap attempts to access other AS: **(-1 P)** *you cannot address physical memory*

f) Welche Annahme steht hinter der LRU Seitenersetzungsstrategie? **1 pt**

On which assumption is the LRU page replacement strategy based upon?

Lösung:

The assumption is, that the access pattern of the past is a good projection for the future. **(0.5 P)** *In the LRU case this means, that it assumes, that pages whose reference lies furthest in the past will be accessed again farthest in the future.* **(0.5 P)**

g) Sowohl E/A Geräte, als auch DMA-Controller signalisieren die Fertigstellung eines Transfers mithilfe eines Interrupts. Warum führt die Nutzung eines DMA-Controllers dennoch zu einer besseren Auslastung des Systems? **2 pt**

Both an I/O device and a DMA controller signal the completion of a transfer with an interrupt. Why does the use of a DMA controller still lead to a better utilization of the system?

Lösung:

(1 P) *each for:*

1. *The DMA controller also copies the I/O result from device memory to main memory.*
2. *Less interrupts are needed, when DMA is used, as only the entire transaction needs to be finally signalled.*

Total: 12.0pt

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

a) Nennen Sie die drei in der Vorlesung vorgestellten Threadmodelle und geben Sie für jedes Modell an, wo (Kernel-/User space) Schedulingentscheidungen getroffen werden. **3 pt**

Name the three threading models discussed in the lecture. For each model, also state where (kernel-/user space) scheduling decisions are calculated.

Lösung:

Threading model	*Scheduling decisions in...*
One-to-one (1x1)	*Kernel space*
Many-to-one (Mx1)	*User space*
Many-to-many (MxN)	*Kernel **and** user space*

***0.5 P** for each field in the table*

b) Gegeben seien vier Prozesse auf einem Einprozessorsystem mit den angegebenen Ankunftszeiten (0 = Start) und Burst-Zeiten. Vervollständigen Sie die untenstehenden Scheduling-Pläne für die Strategie FCFS sowie die Strategie RR für die Zeitscheibenlänge 1 Zeiteinheit. Ein Kasten im Zeitplan stellt eine Zeiteinheit dar. **3 pt**

Consider four processes on a uniprocessor system, with given arrival times (0 = start) and burst times. Complete the scheduling plans given below, for the policy FCFS and the policy RR for a timeslice length of 1 units of time. A box in the scheduling plan represents one unit of time.

Process	Arrival Time	Burst-Time
1	2.5	4
2	1.5	7
3	0	4
4	6.5	5

Lösung:

FCFS

3	*3*	*3*	*3*	*2*	*2*	*2*	*2*	*2*	*2*	*2*	*1*	*1*	*1*	*1*	*4*	*4*	*4*	*4*	*4*

RR

3	*3*	*2*	*3*	*1*	*2*	*3*	*1*	*2*	*4*	*1*	*2*	*4*	*1*	*2*	*4*	*2*	*4*	*2*	*4*

***-0.5 P** for each incorrect scheduling decision*

c) Berechnen Sie für den obigen *FCFS-Scheduling-Plan* die Wartezeit aller Prozesse. **1.5 pt**

For the above FCFS scheduling plan, calculate the waiting time of each process.

Lösung:

The waiting time for each process can be calculated as $t_{wait} = t_{finish} - t_{arrival} - t_{burst}$

Process	*Finish time*	*Arrival time*	*Burst time*	*Waiting time*
1	*15*	*2.5*	*4*	*8.5*
2	*11*	*1.5*	*7*	*2.5*
3	*4*	*0*	*4*	*0*
4	*20*	*6.5*	*5*	*8.5*

d) Berechnen Sie für den obigen *RR-Scheduling-Plan* die Tournaround-Zeit aller Prozesse. **1.5 pt**

For the above RR scheduling plan, calculate the tournaround time of each process.

Lösung:

The turnaround time for each process can be calculated as $t_{tournaround} = t_{finish} - t_{arrival}$

Process	*Finish time*	*Arrival time*	*Turnaround time*
1	*14*	*2.5*	*11.5*
2	*19*	*1.5*	*17.5*
3	*7*	*0*	*7*
4	*20*	*6.5*	*13.5*

e) Erklären Sie, warum ein *FCFS-Scheduler* eine schlechte Wahl für Desktopsysteme ist. **1 pt**

Explain why a FCFS-scheduler is a bad choice for desktop systems.

Lösung:

FCFS does not preempt running processes and therefore does not provide good interactivity, which is important for desktop systems. If, for example, a single process runs on the CPU for an extended period of time, other processes, including the desktop environment, are never dispatched.

f) Betrachten Sie einen Webserver mit einem CPU-Kern, der viele Anfragen von mehreren entfernten Clients verarbeiten muss.

Ist SJF in diesem Szenario ein geeignetes Schedulingverfahren? Schlagen sie gegebenenfalls ein anderes Schedulingverfahren vor! Begründen Sie Ihre Antworten! **2 pt**

Consider a web server with a single CPU core, which must process many requests from multiple remote clients.

Is SJF an appropriate scheduling strategy in this scenario? If not, propose a more appropriate scheduling strategy! Justify your answers!

Lösung:

SJF is not appropriate. **(0.5 P)** *In the above scenario, jobs correspond to individual client requests. In order to schedule the shortest job first, the job length (i.e., the time required to process each request) must be known in advance. This is not the case for a typical web server, as the contents of each request, and hence the amount of work required to process each request, are typically unknown in advance. Therefore, SJF cannot be applied in the above scenario. In addition, even if the job lengths are known in advance, SJF would still not be appropriate, since clients will typically send a mixture of short and long requests. Since SJF prioritizes short requests, long requests would likely starve.* **(0.5 P)**

FCFS is a more appropriate strategy, **(0.5 P)** *since it achieves the lowest possible turnaround times given that job lengths are not known in advance and is guaranteed to be starvation-free.* **(0.5 P)**

(One justification per answer is sufficient. We accept scheduling strategies other than FCFS if justified appropriately.) **Total: 12.0pt**

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

a) Was sind *race conditions*? **1 pt**

Explain the term race condition.

Lösung:

Concurrent access to shared data by different threads **(0.5 P)** *can lead to wrong results.* **(0.5 P)**

Alternatively: Computed results depend on the execution order of involved concurrently running threads or processes. **(1 P)**

b) 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.

Lösung:

(a) Mutual exclusion: Only one thread can be in the CS at a time.

(b) Progress:

- *If no process is in the CS one of the processes trying to enter will eventually get in.*
- *Processes that are not trying to enter do not hinder processes that try to enter from getting in.*

If no process is executing in its critical section and there exist some processes that wish to enter their critical sections, then only those processes that are not executing in their remainder sections can participate in deciding which will enter its critical section next, and this selection cannot be postponed indefinitely.

(c) Bounded waiting: Once a thread starts trying to enter the critical section, there is a bound on the number of times other threads get in. A bound must exist on the number of times that other processes are allowed to enter the critical sections after a process has made a request to enter its critical section and before that request is granted.

(d) 0.5pt for each notion, 0.5pt for each correct and matching explanaton.

c) Wie implementiert man typischerweise Synchronisation auf einem Einprozessorsystem? Nennen Sie zwei Beschränkungen bzw. Nachteile dieses Ansatzes. **2 pt**

How do we typically implement synchronization on a single-processor system? State two limitations and disadvantages of that approach.

Lösung:

On a single-processor system, disabling interrupts is sufficient to achieve mutual exclusion. **(1 P)**

That approach has several downsides:

- *We cannot allow user-space to disable interrupts, which restricts this technique to the kernel.* **(0.5 P)** *(Alternative: If we would allow user-space to disable interrupts, the OS cannot prevent processes anymore.)*
- *While the CPU is executing a critical section with interrupts disabled, I/O devices and other CPUs might try to signal the OS via interrupts. These interrupts will be delayed and can potentially be lost.* **(0.5 P)**
- *Alternatively, we also accept: Disabling interrupts does not provide mutual exclusion on multiprocessor systems.* **(0.5 P)**

d) Was ist *Verklemmungsverhinderung*? **1 pt**

What is deadlock prevention*?*

Lösung:

Negate at least one of the required deadlock conditions: Design your system in a way that one (or multiple) of the conditions cannot occur. **(1 P)**

e) Was müssen Sie bei *Verklemmungsverhinderung* beim Entwurf eines Systems beachten? Welche Vorkehrungen müssen Sie zur Laufzeit treffen? Erläutern Sie Ihre Antwort anhand einer konkreten Technik. **2 pt**

What is required when designing a system if you use deadlock prevention*? What provisions are necessary at runtime? Give your answer using an actual technique as example.*

Lösung:

Deadlock prevention needs to be incorporated into a system or application by its developer at design time. It determines how resources can be allocated and thus has an influence on program structure. **(0.5 P)**

For example, with resource ordering a developer has to ensure that control flow always honors the resource hierarchy and never requests a resource with a lower priority than those already held. **(1 P)**

However, deadlock prevention does not demand for any provisions at run time. **(0.5 P)**

f) Was ist *Verklemmungsvermeidung*? **1 pt**

What is deadlock avoidance*?*

Lösung:

Decide if system stays in safe state on every resource request and block the requesting thread if not. **(1 P)**

g) Was müssen Sie bei *Verklemmungsvermeidung* beim Entwurf eines Systems beachten? Welche Vorkehrungen müssen Sie zur Laufzeit treffen? Erläutern Sie Ihre Antwort anhand einer konkreten Technik. **2 pt**

What is required when designing a system if you use deadlock avoidance*? What provisions are necessary at runtime? Give your answer using an actual technique as example.*

Lösung:

Deadlock avoidance, in contrast, does not require changes to program structure besides an upfront claim of the resources it is going to use. **(0.5 P)**

At runtime, though, a deadlock avoidance algorithm performs extensive bookkeeping of active resource claims. **(0.5 P)**

With resource-allocation graphs, for example, the system adds claim edges when starting a process; it turns claim edges into request edges and then into assignment edges, when a resource is requested and granted eventually. **(1 P)**

Total: 12.0pt

Aufgabe 4: Speicher/*Assignment 4: Memory*

a) Betrachten Sie ein System, das mittels einer hierarchischen Seitentabelle virtuelle in physische Speicheradressen übersetzt. Eine Seitentabelle umfasst 512 Einträge, wobei jeder Eintrag 8 Bytes benötigt. Sowohl der virtuelle als auch der physische Adressraum sind 512 GiB groß, wobei jede Seite 4096 Bytes umfasst. Wieviel physischer Speicher würde für eine volle Seitentabellenhierarchie benötigt? Erläutern Sie Ihre Rechnung. **4 pt**

Consider a system that translates virtual addresses to physical addresses using hierarchical page tables. Every page table comprises 512 entries, with each entry having a size of 8 bytes. The size of both the virtual and the physical address spaces is 512 GiB. The page size is 4096 bytes. How much physical memory is required for a full page table hierarchy? Explain your answer.

Lösung:

The number of pages in a 512 GiB address space is $\frac{512\,GiB}{4\,KiB} = 128\,Mi = 2^{27}$*. A page table has* $512 = 2^9$ *entries* $\Rightarrow \frac{2^{27}}{2^9} = 2^{18}$ *page tables* **(0.5 P)** *.*
To form a hierarchy with 512 entries per table we need two extra levels to address the page tables: $2^{18} = (2^9)^2 = 512^2$ **(0.5 P)** *. That gives us* $512 + 1$ *extra page tables for a total of* $2^{18} + 2^9 + 1$ *page tables in three levels* **(0.5 P)** *.*
A single page table is $512 * 8\,bytes = 4\,KiB = 1\,page$ *in size.*
Total: $2^{18} + 2^9 + 1\,pages$ *of physical memory* **(0.5 P)** *.*

b) Nennen Sie einen Vor- und einen Nachteil von großen Seiten (4 MiB) gegenüber kleinen Seiten (4 KiB). **2 pt**

Give one advantage and one disadvantage of huge pages (4 MiB) compared to small pages (4 KiB).

Pro*: Less space for page tables* **(1 P)** *, higher TLB reach* **(1 P)** *, more efficient disk accesses when paging* **(1 P)** *.*
Contra*: Internal fragmentation* **(1 P)** *.*

c) Bewerten Sie folgende Aussage: Der Heap-Allokator *(malloc)* ist Teil des Betriebssystemkerns. Begründen Sie Ihre Antwort. **1 pt**

Evaluate the following statement: The heap allocator (malloc) is part of the operating system kernel. Explain your answer.

Lösung:

The statement is wrong **(0.5 P)** *. A heap allocator resides in user-space as part of a system library. It reserves huge memory chunks from the operating system and satisfies memory requests (malloc) by allocating from these regions* **(0.5 P)** *.*

d) Ein Prozess greift zum ersten Mal auf eine Seite einer in den Speicher eingeblendeten Datei zu. Das Betriebssystem verwendet Demand-Paging. Welche Schritte sind zur Behandlung des Seitenfehlers nötig? **3 pt**

A process accesses a page of a memory-mapped file for the first time. The operating system uses demand-paging. What steps are necessary to handle the page fault?

Lösung:

- *Check if access is in a valid address space region (yes it is)* **(0.5 P)** *.*
- *Get an empty page frame to hold the data for the page of the memory-mapped file. This potentially includes evicting another page from memory* **(0.5 P)** *.*
- *Load contents of requested page from disk into frame* **(0.5 P)** *.*
- *Adapt the page table to reflect the new mapping.* **(0.5 P)** *.*
- *Set the valid bit of the page table entry* **(0.5 P)** *.*
- *Restart the instruction that caused the page fault* **(0.5 P)** *.*

Variation with page cache also valid.

e) Wie löst man das Problem der Bedeutungsgleichheit (Alias) in einem System mit einem virtually-indexed, virtually-tagged Cache? **1 pt**

How do you solve the alias problem in a system with a virtually-indexed, virtually-tagged cache?

Lösung:

Alias problem: different virtual addresses point to the same physical memory location, but the fully-virtual cache would map them to different cache lines.

We can either disable caching for pages the suffer from aliasing, **(0.5 P)** *or we restrict mappings to such virtual addresses that all aliases map to the same cache line in a direct-mapped cache.* **(0.5 P)**

f) Was versteht man unter dem Working Set eines Prozesses? **1 pt**

What is the working set of a process?

Lösung:

The working set is the set of pages that a process accesses in a given time frame **(1 P)** *.*

Total: 12.0pt

Aufgabe 5: I/O, Hintergrundspeicher und Dateisysteme/*Assignment 5: I/O, Secondary Storage, and File Systems*

a) Betrachten Sie ein System, das an ein sehr schnelles Netzwerk angeschlossen ist. Jedes ankommende Paket löst einen Interrupt aus. Welches Problem kann beim Empfang vieler kleiner Pakete auftreten? **2 pt**

Consider a system that is connected to a very fast network. Each arriving packet triggers an interrupt. What problem can arise while receiving many small packets?

Lösung:

Handling an interrupt causes overhead. The CPU needs to save the current execution context and enter the operating system (amongst other CPU-internal activity). In addition, the user-level context has to be restored after processing the interrupt. **(1 P)**

The reception of many small packets can overload the CPU which can result in packets being lost. **(1 P)**

b) Welche alternative I/O-Technik würde diese Situation verbessern und warum? **1 pt**

Which other I/O technique would improve that situation? Why?

Lösung:

Polling, because no overhead of interrupt handling. **(1 P)**

c) Welches ist der Hauptunterschied zwischen Festplatten (HDDs) und Solid State Disks (SDD) in Bezug auf die Zugriffscharakteristik? Was sind die Auswirkungen hiervon für den Entwurf von Dateisystemen? **2 pt**

What is the main difference in access characteristics between hard disk drives (HDDs) and solid state disks (SSDs)? What are the implications on file system design?

Lösung:

SSDs provide high throughput for small requests at random locations **(0.5 P)** *whereas HDDs only achieve peak bandwidth for sequential reads/writes of data placed consecutively.* **(0.5 P)**

(Alternative: SSDs may need to rewrite large blocks internally even when the file system writes only a few small sectors. **(0.5 P)** *)*

File systems typically aimed at allocating files on disk in as few chunks located as closely together as possible to accomodate for that idiosyncracy of HDDs. **(0.5 P)** *For SSDs, that design goal has lost most of its importance.* **(0.5 P)**

d) In einem i-node-basierten Dateisystem biete jeder i-node Platz für bis zu fünf LBA-Zeiger auf 4 KiB große Festplatten-Blöcke. Wie muss man diese Block-Zeiger einsetzen um eine maximale Dateigröße von 2 MiB bei minimalem Speicherverbrauch zu unterstützen? I-nodes belegen 512 B, jeder Block-Zeiger ist 8 B lang. **3 pt**

Consider an i-node based file system that stores up to five LBA pointers in each i-node, each of which refers to a 4-KiB disk block. How do you have to utilize these pointers to support a maximum file size of 2 MiB using a minimum of disk space? Each i-node is 512 B in size, block pointers are 8 B long.

Lösung:

Directly referencing blocks in the i-node would only allow files of size $5{*}4\ KiB = 20\ KiB$. *Thus, indirection is required: The pointers in the i-node refer to disk blocks that contain disk blocks themselves.* **(0.5 P)**

We can determine the required levels of indirection based on the maximum required file size, the length of pointers, and the size of each disk block.

The file system operates with disk blocks of 4 KiB size. Thus, each indirect block can store $\frac{4\ KiB}{64\ bits} = \frac{4096\ B}{8\ B} = 512$ *pointers.* **(0.5 P)**

A single indirect block can address the required $512 * 4\ KiB = 2\ MiB$. *Using the other four pointers for direct blocks saves storeage space for small files.* **(1 P)**

Thus, we require one level of indirection in addressing file blocks. **(0.5 P)**

The five block pointers in the i-node could be used as follows: Four direct blocks (saving space for small files), and one single-indirect block. **(0.5 P)**

e) Wieviel Speicherplatz wird in obigem Dateisystem insgesamt benötigt, um eine 8 KiB große Datei abzulegen? Wieviel für eine 20 KiB große Datei? Beschreiben Sie Ihren Rechenweg. Ignorieren Sie Verzeichniseinträge. **2 pt**

How much disk space in total is required to store a file of 8 KiB size in the file system discussed above? How much for a 20 KiB file? Describe how you arrived at your solution. Ignore directory entries.

Lösung:

An 8 KiB file requires two data blocks (4 KiB each), and 512 B for the i-node, resulting in $2 * 4\ KiB\ + 512\ B = 8.5\ KiB$. **(1 P)**

A 20 KiB file requires 5 data blocks ($20/4 = 5$*), four of which we can refer in direct block pointers. For the fifth data block, we have to allocate an indirect block. Thus, we require* $5 * 4\ KiB$ *for data* $+4\ KiB$ *for the indirect block* $+512\ B$ *for the inode, resulting in* $24.5\ KiB$ *required storage space.* **(1 P)**

f) Können harte Links und/oder symbolische Links verschiedene Dateisysteme überspannen? Warum/warum nicht? **2 pt**

Can hard links and/or symbolic links span different file systems? Why/why not?

Lösung:

Hard links: no. **(0.5 P)** *A hardlink references an inode (by inode number). Inodes are a file system local entity, so inode numbers are unique only within a file system.* **(0.5 P)**

Symbolic links: yes. **(0.5 P)** *Symbolic links reference a VFS path and are thus unrelated to the underlying FS.* **(0.5 P)**

Total: 12.0pt