

Solution Final Examination System Architecture – April 15, 2003

Aufgabe/Question 1 (Zum Aufwärmen/Warm up, 3 + 3 + 1 + ... + 1 Punkte/marks)

1. „Zählen Sie **drei Systemgrößen** auf, die anzeigen, dass ein **Kernel-Level-Thread** im **Benutzermodus** rechnet!“
*“Enumerate **three system entities** which show, that a **kernel-level thread** is running in **user mode**.”*

- a) Value of instruction pointer register (program counter)
- b) Mode bit in processor status word
- c) Value of stack pointer register
- d) Empty kernel stack

2. „Zählen Sie 3 verschiedene Gründe dafür auf, dass man eine *Task* (Prozess) **ganz** oder **teilweise** vom Hauptspeicher in den Sekundärspeicher (z.B. Platte) auslagert.“
*“State **3 different reasons** for **swapping** -moving **part or all of a task** (process)- from main memory to secondary storage (e.g. disk).”*

- a) Address space of task large than physical memory (partition)
- b) Another suspended task waiting outside preempts one of the “in-memory tasks”
- c) Midterm-scheduler decides that one inside task has exhausted its memory-time slice
- d) Working sets of the current tasks have changed significantly (to prevent thrashing)

Einige der folgenden Aussagen sind korrekt, einige inkorrekt. Unterstreichen Sie „korrekt“, wenn die Aussage korrekt ist, unterstreichen Sie „inkorrekt“, wenn die Aussage inkorrekt ist!
Some of the following statements are correct, some are incorrect. Underline “korrekt” if the statement is correct; underline “inkorrekt” if the statement is incorrect.

3. „**Unterbrechungen** dienen in erster Linie dazu, die **Prozessorauslastung** zu verbessern, da die meisten externen Geräte schneller als der Prozessor sind.“
*“**Interrupts** are provided primarily to improve **processor utilization**, since most external devices are faster than the processor.”*

korrekt

inkorrekt

4. „In einem System **ohne virtuellen Speicher** ist es möglich, dass Teile einer Applikation während ihrer Abarbeitung aus- und eingelagert werden können.“
*“In a system **without virtual memory**, it is possible, that parts of an application can be swapped out and in during its execution.”*

korrekt

inkorrekt

2

5. „Wenn der Hauptspeicher in eine **feste Anzahl von Partitionen** unterteilt wird, nennt man dieses Belegungsschema **festgeschriebene Partitionen (fixed partitioning)** .“
“*If main memory is divided into a **fixed number of partitions**, this allocation scheme is called **fixed partitionion**.*”

korrekt

inkorrekt

6. „Die Menge an Seiten, die von einer Applikation **innerhalb einer bestimmten Zeitspanne referenziert** werden, heißt **Residentmenge (resident set)** .“
“*The set of pages that an application is **accessing over a specific period of time** is called **resident set**.*”

korrekt

inkorrekt

7. „Ein Ablaufplaner (*scheduler*) optimiert den **Durchsatz** , wenn seine Strategie die Zahl der beendeten Applikationen pro Zeiteinheit maximiert.“
“*A scheduler optimizes the **throughput**, if its policy maximizes the number of completed applications per unit of time.*”

korrekt

inkorrekt

8. „Ein System besitzt **Datenvertraulichkeit (data confidentiality)** , wenn es **geheime Information** geheim halten kann.
“*A system achieves **data confidentiality** if its **secret information** is kept secret.*”

korrekt

inkorrekt

Aufgabe/Question 2

(2 + 2 + 4 + 4 Punkte/marks)

1. „Angenommen, Sie müssten einen Webserver mit mehreren Threads implementieren. Wenn in dem System der einzige Weg von einer Datei zu lesen der normale, **blockierende read Systemaufruf** ist, welches der beiden unten angegebenen **Threadmodelle** würden Sie dann verwenden? Geben Sie hierfür eine knappe, ausreichende Begründung an!“
“*Suppose you have to implement a web server with multiple threads. If in the system, the only way to read from a file is the normal, **blocking read system call**, which one of the following two **thread models** would you use? Give a short, but sufficient explanation.*”

a) Pure User-Level Threads

b) Kernel-Level Threads

If a user-level thread is executing a blocking-read, then the kernel scheduler will blocks the complete webserver-task, i.e. no additional web-requests can be executed.

2. „Wenn eine **Anwendung (application)** aus nur zwei Kernel-Level Threads besteht, macht es dann Sinn, eine **Barriere** zu benutzen, um beide Threads zu synchronisieren? Warum oder warum nicht?“
“*If an **application** has only two kernel-level threads, does it make sense to use a **barrier** to synchronize them? Why or why not?*”

A barrier can be used for $n \geq 2$ threads, it's an efficient solution, because in each thread you only have to do one system-call for synchronization purpose, especially if both threads operate in phases and neither thread may enter the next phase until both have finished the current one.

3. „Nennen Sie **vier** Schedulingalgorithmen, die für **interaktive Systeme** geeignet sind!“
 “Name **four** scheduling algorithms, being suitable for **interactive systems**.”

- a) **Round-Robin (with optimized time-slice for interactive requests)**
- b) **Multilevel-Feedback**
- c) **Preemptive Priority Scheduling**
- d) **Highest Response Ratio**
- e) **Fair Share Scheduling**

4. „Erläutern Sie so knapp wie möglich, aber gleichwohl so ausführlich wie nötig den folgenden Implementierungsvorschlag! Für welches **Nebenläufigkeitsproblem** (*concurrency problem*) sind beide angegebenen Assemblerrouninen vorgesehen? Für welches **Threadmodell** ist diese Variante gedacht? Diskutieren Sie, ob alle Anforderungen an eine Lösung dieses Problems erfüllt werden!“
 “*Explain as succinctly as possible, but as detailed as necessary, the implementation listed below. For which **concurrency problem** are these two assembler routines provided? For what **thread model** these routines are possible solution variants? Discuss whether all requirements for a solution of that problem are met.*”

```
mutex_lock:
    TSL REGISTER, MUTEX      | copy mutex to register, atomically set mutex to 1
    CMP REGISTER, #0        | was mutex zero?
    JZE ok                  | if it was zero, mutex was unlocked, so return
    Call thread_yield()     | mutex is busy; schedule another thread
    JMP mutex_lock          | try again later
ok: RET                    | return to caller; enter CR

mutex_unlock:
    MOVE MUTEX, #0         | set mutex to 0
    RET                    | return to caller
```

Nebenläufigkeits problem (*concurrency problem*): Mutual exclusion

Threadmodell (*thread model*): Both thread models (or only user level threads, due to Tanenbaum)

Diskussion (*disussion*):

- a) **Mutual exclusion is provided**
- b) **No assumptions about speed or number of CPUs are made**
- c) **No thread running outside the critical section may block other competing threads**
- d) **However, bounded waiting may happen if the scheduler is not fair**

Aufgabe 3 / Question 3 :**(2 + 1 + 3 + 6 Punkte/marks)**

1. „Bekanntlich verfolgen sowohl Unix als auch Windows den „Vogel-Strauss“-Algorithmus (*ostrich-algorithm*), d.h. tun nichts, um **Verklemmungen** (*deadlocks*) zu **vermeiden** oder zu **verhindern**. Finden Sie Argumente, die dieses Vorgehen rechtfertigen!“
“It’s well known that Unix and Windows use the so called ostrich-algorithm, i.e. both system do nothing to prevent or avoid deadlocks. Find arguments justifying this procedure.”

If deadlocks do not occur pretty often compared to other system crashes, there is no need for the overhead and the restrictions for deadlock avoidance or deadlock prevention.

2. „Ein System hat **sechs** Bandlaufwerke, um die **n single-threaded Anwendungen** (*single-threaded applications*) konkurrieren. Jede Anwendung braucht bis zu **zwei beliebige** Laufwerke. Für welche Werte für **n** ist das System **verklemmungsfrei** (*deadlock free*)?“
“A system has six tape drives, with n single-threaded applications competing for them. Each application may need up to two arbitrary drives. For which values of n is the system deadlock free?”

n < 6

3. „Beim **Round-Robin**-Scheduling mit fester Zeitscheibenlänge werden alle bereiten (*ready*) Kernel-Level Threads durch **genau einen Eintrag** in einer zyklischen Liste verwaltet. Wie würde es sich auf das **Scheduling auswirken**, wenn ein Kernel-Level Thread in dieser Liste **zweifach** geführt wird? Zu **welchen Problemen** kann diese Maßnahme führen?“
“Round-robin schedulers normally maintain a circular list of all ready kernel-level threads, with each thread occurring once in that list. What consequences to scheduling behavior will occur, when a kernel-level thread appears twice in that list? What problems may result from this approach?”

As long as the thread does no blocking system call (thread is running or ready), it will get more processor cycles, thus it will finish earlier. This approach can be used to give important threads a bigger share of the CPU. A problem may arise if the thread is blocking, then both entries have to be removed.

4. „Welche der folgenden **Attribute** sind **task-spezifisch**, welche **thread-spezifisch**? Markieren Sie entweder mit Task oder mit Thread!“
“Which of the following attributes are task specific, which are thread specific? Mark with task or thread.”

List of Open Files	Task
Address space	Task
Instruction pointer	Thread
Global variables	Task
Registers	Thread
User stack	Thread

5. „Für welche **Zwecke** sind **Pop-Up Threads** nützlich?“
 “For what **purpose** are **pop-up threads** useful?”

For a multi-threaded server with a rarely used service-function, whereby a small delay between request and the service can be tolerated. Instead of having a waiting thread for this rarely used procedure, you delay an incoming request as long as the pop-up-thread has been created to execute this procedure.

Aufgabe/Question 4

(3 + 2 + 2 + 5 Punkte/marks)

1. „Erklären Sie, wie es zu **zwei Seitenfehlern** im Zuge **einer einzigen Speicherreferenz** kommen kann!“
 “Explain how **two page faults** can occur for a **single memory reference**.”

The referenced object spans two pages

Neither page-table nor page in a multilevel-page table system are mapped

2. „Wenn eine Seite (*page*) gleichzeitig zu zwei Adressräumen gehört, können dann für den einen Adressraum **Lese- und Schreibzugriffe** auf diese Seite erlaubt sein, während für den anderen **nur Lesezugriffe** erlaubt sind? **Begründen** Sie Ihre Antwort!“
 “If a page belongs to two address spaces, is it possible, that one address space has **read- and write-access** to that page, whereas the other one can **only read** from that page? **Explain** your answer.”

It is possible. Assuming that segmentation is not present, the protection information must be in the page table. If each address space has its own page table, each one also has its own protection bits. They can be different.

3. „Ein **TLB-Eintrag** auf einer VAX enthält **kein Referenzbit** (R-bit). Wie haben es die Systemarchitekten von VMS gleichwohl geschafft, einen effizienten, referenzbasierten Seitenersetzungsalgorithmus zu implementieren?“
 “A **TLB-entry** on a VAX does not contain a **referenced bit**. How did the system architects of VMS manage to implement an efficient reference based paging algorithm?”

The referenced bit is not needed in the TLB. The mere presence of a page in the TLB means the page has been referenced; otherwise it would not be there. Thus the bit is completely redundant. However, when the entry is written back to main memory, the reference bit in the software-page-table is set.

4. „Erklären Sie, warum ein System mit virtuellem Speicherkonzept einen höheren *Multiprogramming-Grad* ermöglicht als ein System ohne virtuellen Speicher!“
 “Explain, why a system with a virtual memory allows a higher degree of multiprogramming than a system without virtual memory.”

In a virtual memory system address spaces no longer have to be mapped completely, only the needed parts have to be in memory, the rest may stay on disk. Thus more space in main memory is available for other tasks.

5. „Wenn, wie in MULTICS, **Segmentierung und Paging** benutzt werden, muss zunächst der Segmentdeskriptor geladen werden, dann der Seitendeskriptor. Funktioniert auch der TLB mit einer zweistufigen Suche? **Begründen** Sie Ihre Antwort!“
*“When **segmentation and paging** are both being used, as in MULTICS, first the segment descriptor must be looked up, then the page descriptor. Does the TLB also work this way, with two levels of lookup? Justify your answer.”*

No, the search key uses both the segment number and the virtual page number, so the exact page can be found in a single match.

Aufgabe 5/ Question 5:

(2 + 2 + 2 + 6 Punkte/marks)

1. „Was bedeutet „**Geräteunabhängigkeit**“? Welche Vor- oder Nachteile hat ein System mit dieser Eigenschaft?“
*“What is “**device independence**”? What advantages or what disadvantages has a system with that property?”*

Device independence means that files and devices are accessed the same way, independent of their physical nature. Systems that have one set of calls for writing to a file, but a different set of calls for writing on the console (terminal) do not exhibit device independence (e.g. no straightforward I/O-redirection).

2. „Ein mögliches Dateiattribut stellt die **Satzlänge (record length)** dar. Wann und weswegen kümmert sich ein Betriebssystem jemals um dieses Attribut?“
*“One potential file attribute is the **record length**. When and why does an operating system ever care about this?”*

The operating system cares about record length when files can be structured as records with keys at a specific position within each record and it is possible to ask for a record with a given key. In that case, the system has to know how big the records are so it can search each one for the key.

3. „Einige Betriebssysteme stellen den Systemaufruf **rename** zur Verfügung, um einer Datei einen neuen Namen geben zu können. Gibt es überhaupt einen Unterschied zwischen diesem Aufruf und dem Kopieren der Datei auf einen neue, gefolgt von der Löschung der alten Datei?“
*“Some operating systems provide a system call **rename** to give a file a new name. Is there any difference at all between using this call to rename a file, and just copying the file to a new file with the new name, followed by deleting the old one?”*

Yes, the **rename system call does not change the creation time or the time of last modification, but creating a new file causes it to get the current time as both creation time and the time of last modification. Also, if the disk is full, the copy might fail.**

4. „In einem älteren Unixdateisystem mit nur der „Wurzelpartition“ (*root partition*) sei dessen **Liste der freien Plattenblöcke** durch einen **Systemabsturz komplett zerstört**. Gibt es irgendeinen **Ausweg**, sich aus diesem Unglück zu befreien, oder ist dieses Dateisystem unwiederbringlich verloren? **Begründen** Sie ihre Antwort!“
*“In an older Unix file system with just the root partition the **list of free disc blocks** is **completely destroyed** due to a **system crash**. Is there any way to **recover** from this disaster, or is this file system lost forever? **Explain** your answer.”*

No, the file system is still available, i.e. all disk blocks – occupied blocks = free blocks. Temporarily mark all disk blocks free. Then touch each directory and file, then remove each touched block from the free list, the remainder is the current free list.

5. „Wie hätten Sie das **TENEX-System modifiziert**, um den **berühmten Schlüsselwortfehler** (*password flaw*) zu vermeiden?“
“*How would you have **modified the TENEX system** to avoid the **famous password flaw**?”*”

Use string compare instead of character compare.

6. „Um ein System nach einer **erfolgreichen Virusattacke** wiederherzustellen, kann man manchmal die unten angegebenen Anweisungen lesen. Nennen Sie **zwei schwerwiegende Fehler** in diesen Anweisungen!“
“*Sometimes one sees the following instructions for recovering from a **successful virus attack**. Name **two serious errors** in these instructions.*”
- Boot the infected system
 - Back up all files to an external medium
 - Run the program **fdisk** to format the disk
 - Reinstall the operating system from the original CD-ROM
 - Reload the files from the external medium.

First, running the `fdisk` program from the hard disk is a mistake. That program may have been infected and it may infect the boot sector. `fdisk` has to be run from the original CD-ROM or from a write-protected floppy-disk.

Second, the restored files may have been infected, too. Putting them back without cleaning them, may just reinstall the virus.