

Aufgabe/Question 1 (Zum Aufwärmen/Warm up, 4 + 3 + 1 + ... + 1 Punkte/marks)

1. „Zählen Sie **vier Operationsziele** (*objectives*) eines **Rechenservers** (*compute server*) auf!“
 “Enumerate *four objectives of a compute server.*”

a) **Performance**

b) **Robustness**

c) **Scalability**

d) **Security**

2. „Geben Sie für jeden der folgenden drei Posix-ähnlichen Systemaufrufe (*system calls*) einen Grund dafür an, dass er möglicherweise fehlschlägt: `fork`, `exec` und `unlink`.“
 “For each of the three following Posix like system calls, give a condition that may cause it to fail: *fork, exec, and unlink.*”

`fork`: **All entries in the task table are full**

`exec`: **The file to be executed is not executable**

`unlink`: **A component of the path prefix is not a directory**

3. „Im **Einprogrammbetrieb** benötigt man keine **blockierenden Systemaufrufe**.“
 “With *single programming* you don't need *blocking system calls.*”

korrekt

inkorrekt

4. „Ein **Seitenfehler** (*page fault*) ist eine **Ausnahme** (*exception*) und keine **Unterbrechung** (*interrupt*).“
 “A *page fault* is an *exception*, not an *interrupt.*”

korrekt

inkorrekt

5. „**Multi-threaded Anwendungen** (*multi-threaded applications*), deren Threads nur **nebenläufig**, aber **nicht echt parallel** ausgeführt werden, können nur in Einprozessorsystemen vorkommen.“
 “*Multi-threaded Applications*, whose threads can only be executed *concurrently*, but *not really in parallel*, can only occur within *single-processor systems.*”

korrekt

inkorrekt

6. „Das **Klienten-Dienstgeber-Modell** (*client server model*) kann auch in lokalen Systemen erfolgreich eingesetzt werden.“

“*The client-server model can be used successfully in local systems, too.*”

korrekt

inkorrekt

7. „**Peterson’s Lösung** des Problems des wechselseitigen Ausschließens funktioniert, wenn das Scheduling **nicht preemptiv** ist.“

“*Peterson’s solution to the mutual exclusion problem works when scheduling is non preemptive.*”

korrekt

inkorrekt

8. „Vom **Kern angebotene Semaphore** können zur Synchronisation **innerhalb** von multi-threaded Anwendungen verwendet werden, egal ob diese mittels **reinen User-Level Threads** oder mittels **Kernel-Level Threads** implementiert sind.

“*Semaphores offered by the kernel can be used to synchronize within multi-threaded applications, whether these applications are implemented by pure user-level threads or by kernel-level threads.*”

korrekt

inkorrekt

Aufgabe/Question 2

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

1. „Welche **Threadattribute** würden Sie im **Threadkontrollblock TCB** eines **harten Echtzeitsystems** installieren?“

“*Which thread attributes would you include in the thread control block TCB of a hard real-time system?*”

Priority

Deadline

Resource Amount

Expected Execution Time

2. „Zählen Sie **zwei** geeignete **Schedulingstrategien** für Echtzeitsysteme auf und erläutern Sie deren **Wirkungsweise!**“

“*Enumerate two appropriate scheduling policies for real-time systems and explain how they work.*”

EDF, order tasks according to non decreasing deadlines, execute the task with the earliest deadline first

Rate monotonic, order the tasks according to non decreasing period lengths, execute the task with the smallest period length

3. „Zählen Sie **vier** charakteristische Merkmale von **Echtzeitsystemen** auf!“
 “Enumerate *four* characteristics of *real-time systems*.”

- a) **Some deterministic behavior (periodical, or sporadic tasks with deterministic execution times)**
- b) **Meet deadlines, assure quality of service**
- c) **Some sort of user control concerning loading the system**
- d) **Reliability**

4. „Welcher **Typ** von **Echtzeitapplikationen** kann in WindowsNT unterstützt werden? Durch welche Maßnahmen erfolgt diese Unterstützung?“
 “What *type* of *real-time applications* is supported by WindowsNT and how is this support implemented?”

Type of application: soft-real time applications (i.e. multi-media)

Implementation of support: define a disjoint priority schema for the real-time tasks, i.e. no non real-time task can ever preempt a real-time task

5. „Was verstehen Sie im Zusammenhang mit **Prioritätsinvertierung** unter einem **Betriebsmittelzuordnungprotokoll** (BZP) (*resource allocation protocol*)“? Erläutern Sie ein konkretes BZP und zählen Sie dessen charakteristische Eigenschaften auf!“
 “Connected with *priority-inversion* what is a *resource allocation protocol* (RAP)? Explain a concrete RAP and enumerate its characteristic properties.”

Non-preemptive critical sections, i.e. whenever a task holds a resource, it is no longer preempted.

This simple protocol

- bounds priority inversion,**
- prevents from deadlocks,**
- but does not use the resources efficiently and**
- blocks high priority tasks even if these won't ever use a critical resource.**
- It does not need a priori knowledge of resource usage.**

Aufgabe 3 / Question 3 :

(6 + 6 Punkte/marks)

1. „Gegeben sei die folgende Programmskizze für das „Interaktionsproblem begrenzter Puffer“ (*bounded buffer problem*). Untersuchen Sie die **Einsatzfähigkeit** (korrekte Funktion, Auftreten von Verklemmungen, Fairneß etc.) dieser Lösung! “
 “Given the following program draft to the bounded buffer problem. Analyze this proposal's *fitness for use* (correct function, potential occurrence of deadlocks, fairness etc.).”

```

semaphore e = n;           // number of empty buffers
semaphore f = 0;          // number of filled buffers
semaphore b = 1;         // controlling adding to or
                          // taking from buffer

cobegin

producer: loop
    produce;
    P(b);P(e); add to buffer ; V(b); V(f);
end of loop

consumer: loop
    P(f);P(b); take from buffer; V(b);V(e);
    consume;
end of loop

coend

```

Korrektheit/correctness: **Not correct**

Verklemmung/deadlock: **Deadlock when the producer has filled the buffer threetimes, and the consumer has passed P(f), but before passing P(b), the producer executes p(b) again. Thus the consumer is awaiting V(b), and the producer is awaiting V(e). However, both events will never occur.**

Fairness: **Not unfair, but still not correct**

Sonstiges/other: **Proposal is inefficient due to the number of superfluous semaphores.**

2. „Ein System hat **zwei Anwendungen** und **drei identische Betriebsmittel** (*resources*). Jede Anwendung braucht maximal zwei Betriebsmittel. Ist eine **Verklemmung** (*deadlock*) möglich? Begründen Sie Ihre Antwort!“
“A system has two applications and three identical resources. Each application needs a maximum of two resources. Is deadlock possible? Explain your answer.”

Deadlock free, because at least one of both applications can get all of its needed resources, the other one is awaiting, until the first application will terminate.

3. „Was passiert, wenn **zwei CPUs** in einem **Multiprozessorsystem** versuchen, zur **exakt der selben Zeit** auf **exakt das gleiche physische Speicherwort** zuzugreifen, so lange dieses Speicherwort noch in **keinem** der Prozessorcaches steht?“
“What happens if two CPUs in a multiprocessor system attempt to access exactly the same physical memory word at exactly the same instant of time, as long as this memory word is not yet located in any of the processor caches?”

Reading can be done in parallel (on some busses), reading-writing will be serialized by the bus-arbiter.

4. „Erläutern Sie, wie es bei **Verwendung von Spinlocks** in einem **Mehrprozessorsystem** zum „**Cache-Line Thrashing**“ kommen und wie man diese **vermeiden** kann!“
“Explain how cache line thrashing can occur using spinlocks in a multiprocessor system and how you can avoid this.”

Suppose you have more than 1 thread spinning due to a closed spin-lock, then on each CPU you're doing the TAS-instruction in vain, i.e. each CPU is writing to its cache-line, thus invalidating the the cache-line in all other CPUs (even though always the same value (e.g. 1) is written. This decreases the bus capacity.

Instead of you can use a normal read on the spinlock, and in case that this variable is closed, no longer the cache-line entries are changed until one CPU unlocks the spin-lock.

Aufgabe/Question 4

(2 + 6 + 7 Punkte/marks)

1. „Ein imaginärer Computer mit 32-Bit-Adressen benutzt eine **zweistufige Seitentabelle**. Virtuelle Adressen werden in ein 9-Bit-Feld für die oberste Seitentabelle, 11 Bit für die zweite Seitentabelle und einen „*Offset*“ unterteilt. Wie **groß** sind die **Seiten** und wie **viele Seiten** sind im Adressraum?“

*“An imaginary computer with 32-bit addresses uses a two-level page table. Virtual addresses are split into a 9-bit top-level page table field, an 11-bit second-level page table field, and an offset. How **large** are the pages and how **many** are there in the address space?”*

page size = 2^{12} bytes = 4 KB

number of pages = 2^{20}

2. „Zählen Sie **drei Zeitpunkte** im Leben einer Anwendung auf, zu denen das Betriebssystem Arbeit hinsichtlich von **Seitentausch** (*paging*) zu verrichten hat. **Erläutern** Sie die anfallende Arbeit des Betriebssystems.“

*“Enumerate **three times** in the life of an application when the operating system has work to do relating to **paging**. **Explain** the corresponding work of the OS.”*

a) Loading the application, i.e. establishing all mapping-data (page-table etc.)

b) Starting of the application, i.e. loading the initial working set

c) Transition from one working-set to another one during execution.

d) Terminating the application, i.e. freeing the pages and all the meta data

3. „Erläutern Sie „Belady’s-Anomalie“ und bei welchen Verfahren tritt diese Anomalie auf?“

“Explain “Belady’s anomaly” and with which policies this anomaly can be observed?”

It can happen that in a larger RAM more page-faults occur, e.g. FIFO-Paging

5. „Erläutern Sie die **Vorteile** von „Superseiten“ (*super pages*)!“
 “*Explain the advantages of super pages.*”

You need fewer TLB-entries, you may need fewer page-tables, if there is some locality within that super page, there will be fewer page-faults.

Aufgabe 5/ Question 5:

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

1. „Erläutern Sie, wie das **Pufferüberlaufproblem in C** wirkungsvoll für **Attacken** auf ein System ausgenutzt werden kann!“
 “*Explain how the buffer overflow problem in C can be used effectively to attack a system.*”

Since there is no overflow control for arrays in C, you can use a local character array on the stack to overwrite the return address of the main program with another start address that is doing harm to the system. Suppose the main program has super user rights, the system will crash sooner or later or becomes open for all kinds of attacks in alter attack.

2. „Stellen Sie die **Rechte**, die im folgenden Unix-Verzeichnislisting gezeigt werden, als **Zugriffskontrolllisten** dar!“
 “*Express the permissions shown in the following Unix directory listing as access control lists.*”

```
-rw-r--r--  2 gmw  users    908  May 05 1:45  PPP-Notes
-rwxr-xr-x  1 asw  devel    432  May 06 2:35  progr1
-rw-rw----  1 asw  users   50094 May 12 4:55  project.t
-rw-r----- 1 asw  devel   13128 May 15 10:30 splash.gif
```

File	ACL
PPP-Notes	gmw:RW; *:R
progr1	asw: RWX; devel: RX;*:R
project.t	asw: RW; users: RW
splash.gif	asw: RW; devel: R

3. „Entwerfen Sie für ein Mehrprogrammiersystem **Dateinutzungskontingente pro Benutzer!**“
 “*Design file usage quotas per user in a multiprogramming system.*”

1. Limit the amount of space in the file-system (i.e. disk blocks for data-bas well as for meta-blocks for files and directories) a user can get.
2. You may also want to limit the of the number of open files per user, in order to use file tables.

4. „In welcher der **vier Ein-/Ausgabe-Softwareschichten** (*layers*) wird jeder der folgenden Punkte bearbeitet?“

*“In which of the **four I/O-software layers** is each of the following done?”*

- 5.1 Berechnung der Spur, des Sektors und des Kopfes beim Lesen von der Platte/ *computing the track, sector, and head for a disk read:*

Device Driver

- 5.2 Schreiben von Kommandos in die Gerätereister/ *writing commands to the device registers:*

Device Driver

- 5.3 Prüfung, ob der Benutzer das Gerät verwenden darf/ *checking whether the user is permitted to use the device:*

Device independent Software

- 5.4 Konvertierung von Binär-Integer-Zahlen nach ASCII beim Drucken/ *Converting binary integers to ASCII for printing:*

User-level Software