Solution System Architecture, September 11, 2009

Aufgabe 1 / *Question 1*(Zum Aufwärmen/*Warm up*, 2 + 2 + 2 + 1 + … 1 Punkte/*marks*)

1. „Unterstreichen Sie diejenigen der folgenden **Linux-Systemaufrufe**, die eine **neue Aktivität** erzeugen!“
   *“Underline those of the following **Linux system calls** that create a **new activity**.”*

   execve() **clone()** **fork()** pthread_create()

2. „Unterstreichen Sie diejenigen der folgenden **POSIX-Funktionen**, die zur **Synchronisation** von Threads der **gleichen Task** benutzt werden können!“
   *„Underline those of the following **POSIX functions** that can be used to **synchronize threads** of the **same task**.”*

   wait() exit() **pthread_join()** **pthread_exit()**

3. „Zählen Sie **zwei** der drei **Mutextypen** auf, die von POSIX angeboten werden!“
   *“Enumerate **two** of the three **types of mutexes** that are offered by POSIX.”*

   Mutex with priority ceiling **Mutex with priority inheritance**

4. „**Gerätetreiber** sind **typische Komponenten** eines **Mikrokerns**.“
   *“**Device drivers** are **typical components** of a **micro kernel**.”*

   korrekt **inkorrekt**

5. „Je **größer** die **Seitenkachel** (*page frame*) ist, desto **größer** wird der **externe Verschnitt** (*external fragmentation*).“
   *“The **bigger** the **page frame** the **larger** will be the **external fragmentation**.”*

   korrekt **inkorrekt**

6. „Wenn eine Applikation im **Nutzermodus** (*user mode*) eine **privilegierte Operation** ausführen will, dann wird eine **Unterbrechung** (*interrupt) auftreten.*“
   *“When an application tries to **execute** a **privileged operation** in **user mode**, an **interrupt** will occur.”*

   korrekt **inkorrekt**

7. „Angenommen, eine Festplatte würde Daten in den Hauptspeicher mit der **Datenrate 8 MB/s** übertragen. Heutige Prozessoren sind so schnell, dass sie in der Lage sind, jeden **32 Bit-Block per Polling** anzunehmen.“
   *“Assume that your hard disk can transfer data to RAM with a **data rate of 8 MB/s**. Today’s processors are so fast that they could **poll each 32 bit-block**.”*

   korrekt inkorrekt

8. Eine **zentrale Bereitwarteschlange** *(ready queue)* **erhöht** die **Skalierbarkeit** von **Mehrprozessorsystemen.**“
*“A **centralized ready queue improves** the **scalability** of **multi processor-systems**.”*

korrekt **inkorrekt**

9. „In einem **Mehrprozessorsystem** führt ein **Interprozessorsignal** zu einer **Unterbrechung** (*interrupt*) des signalisierten Prozessors.“
*“On a **multi-processor system**, an **interprocessor-signal** leads to an **interrupt** of the processor that has been signaled.”*

korrekt inkorrekt

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

1. „Welche **Art** von **Daten** einer **Applikation** wird typischerweise auf ihrer **Halde (***heap*) abgebildet (*mapped*)?”
*“What **kind** of **data** of an **application** typically will be mapped to its **heap**?”*

Dynamic objects

2. „Zählen Sie **typische Anwendungen** eines **Handheld** (z.B. PDA)! Vergleichen Sie dessen Betriebssystem mit dem eines Personal Computers (PC)! Welche **wesentlichen Unterschiede** bestehen zwischen diesen beiden Betriebssystemen?“
*“Enumerate **typical applications** of a **handheld** (e.g. PDA). Compare its operating system with an OS of a personal computer (PC). What will be the **major differences** between both operating systems?”*

Handheld applications: **Web-Browser, Terminplaner, Email, Notizbuch, Adressenbuch, Spiele, Wecker, Uhr, …**

Major OS differences:
OS of PDA is a single user system (single-programming), due to slow CPUs and relatively small RAM, need for performance, i.e. no virtual memory, no need for extensibility and only really needed device-drivers are installed

OS of a PC must be extensible, has virtual memory, many loadable modules for all those device drivers, that can be added to the system

3. „Welche **Adressbereiche** (*regions*) und **Datenstrukturen müssen** bei der Erzeugung eines **Kernel-Level Threads** im **Nutzer**- bzw. im **Kernadressraum** angelegt werden?“
*“In the **user**- respectively **kernel-address space**, what **regions** and **data structures have to be** installed when **creating** a **kernel-level thread**?”*

Nutzeradressraum/user-address space: **User-stack**

Kernadressraum/kernel-address space: **Thread-Control Blco (TCB)**
Kernel Stack (often part of TCB)

4. „Der unten angegebne Pseudoprogrammcode soll die Synchronisation an einem zyklischen Puffer aus N Elementen zwischen mehreren Produzentenprozessen und einem Konsumentenprozess übernehmen. **`freeBuffer`** und **`fullBuffer`** sind Zählsemaphore (*counting semaphore*). Geben Sie eine Ausführungssequenz an, die zu einem Fehler führt."
   "The pseudo program code below is intended to synchronize buffer control between multiple producing processes and a single consuming process. `freeBuffer` and `fullBuffer` are semaphores. Describe an execution sequence which causes an error."

```
/* global data declarations */
current free := 0; /*pointer to free buffer element */
current full := 0; /*pointer to full buffer element */
freeBuffer := N;   /* number of free elements */
fullBuffer := 0;   /* number of filled elements */
lockBuffer := 1;   /*for mutual exclusion at buffer */

Producers:
LOOP
produce item;
p(freeBuffer);
p(lockBuffer);
COPY (buffer[current free], item);
current free := (current free + 1) MOD N;
v(lockBuffer);
v(fullBuffer);
END;
Consumer:
LOOP
p(fullBuffer);
COPY (item, buffer[current full]);
current full := (current full + 1) MOD N;
v(freeBuffer);
consume item;
END;
```

Fehlerhafte Sequenz/sequence causing an error:

Producer 1 is preempted after the COPY, then producer2 runs. Both copy to the same location inside the buffer, whereas the second buffer element is not filled. In our case the buffer element of producer1 is lost. Thus the consumer will consume first product of producer 2 and afterwards and empty buffer element

5. „**Ergänzen** Sie die **fehlenden Teile** zum obigen Code, so dass er das Problem korrekt löst! (Monitore sind nicht erlaubt)."
   "**Add** the **missing parts** to the above code to correct the problem. (Monitors are not allowed)."

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

1. „Was ist ein **Kern-Ablaufplaner** (*kernel scheduler*)? **Beschreiben** Sie so **präzise und vollständig** wie **möglich** seine Aufgabe! “
   *“What is a **kernel-scheduler**? **Describe** as **precisely** and **completely** as **possible** its task.”*

   The kernel-scheduler preemptively or non-preemptively selects the next activity to run on the CPU according to some scheduling policy. It might also decide how long an activity is allowed to run. The kernel-scheduler tries to optimize some performance measures, such as turnaround time, response time, or device utilization.

2. Erklären Sie, **warum** manche **Kernablaufplaner** (*kernel scheduler*) **E/A-intensive** Prozesse gegenüber **CPU-intensiven** Prozessen **bevorzugen**!“
   *“Explain **why** some **kernel-schedulers favor I/O-bound** processes versus **CPU-bound** processes.”*

   Many I/O devices are much slower than the CPU, i.e. favoring I/O intensive processes might help to reduce their long turnaround-times, whereas the CPU-intensive processes have only marginally longer turnaround times. Additionally, favoring I/O-bound processes might help to improve the response-time of the system.

3. „Erklären Sie warum man **temporäres Ausmaskieren** von **Unterbrechungen** (*interrupts*) in einem **Einprozessor**- aber **nicht** auf einem **Mehrprozessorsystem** als Basis für den **wechselseitigen Ausschluss** von **Kernoperationen** verwenden kann!“
   *“Explain why **temporarily masking interrupts** can be used as a basis on which to build **mutual exclusion** for a **single-processor**- but not for a **multi-processor-system**.”*

   On a single-processor-system, masking of interrupts ensures that the code execution is not interrupt by external events, thus ensuring atomicity.
   However, interrupt masking only affects a single processor. Interrupts as well as other kinds of kernel invocations (syscalls, exceptions) might still occur on the other processors in the system.

4. „Ein **Pure-User-Level Thread** (PULT) ruft eine Funktion **`f()`** auf.
   a) In welchem **PULT-Zustand** (*PULT state*) befindet sich der PULT **während** der **Ausführung** von **`f()`**?
   b) Unterstreichen Sie diejenigen der unten angegebenen **Taskzustände**, in welche die zugehörige Task während der Ausführung von **`f()`** **kommen kann**.“

   “A pure-user-level thread** (PULT) calls a function **`f()`.
   *a) In what **PULT-state** is the PULT **during** the **execution** of **`f()`**?*
   *b) Underline those of the below mentioned **task states** into which the corresponding task **can come** during the execution of **`f()`**. ”*

   a) Running

   b) Task-Zustand/*task/state*:

   Running **Ready** **Blocked** **Swapped-Out**

5. „Was ist ein **APIC** und zu **welchem Zweck** wird er **installiert**?
*"What is an **APIC** and for **what purpose** is it **installed**."*

An APIC is a programmable HW-unit that supports the CPU when handling interrupts. You can program the APIC to enable a priority driven scheduling or a RR scheduling of pending interrupts. You can mask a couple of interrupts, thus you can change the standard interrupt handling behaviour of the system.

Aufgabe 4 / *Question 4* (6 + 1 + 2 + 3 Punkte/*marks*)

1. „Gegeben sei ein System mit **zweistufigen Seitentabellen** und einem **Hardware gesteuerten TLB**. Geben sie **detailliert** und in der **richtigen Reihenfolge** alle einzelnen Schritte an, die der **Prozessor** durchführt, um **ein Byte** von der **logischen Adresse lx** in ein **Prozessorregister** zu **laden** unter der Annahme, dass der **TLB** zu Beginn **keine gültigen TLB-Einträge** enthalte. Gehen Sie dabei insbesondere darauf ein, woher der Prozessor die Informationen darüber erhält, an welcher Stelle im Speicher die benötigten Daten liegen. Nehmen Sie der Einfachheit halber an, dass der Prozessor über keinen Pufferspeicher(Cache) verfüge und sämtliche Daten aus dem Hauptspeicher (*RAM*) angefordert werden müssen. Nehmen Sie weiter an, dass sämtliche benötigte Daten im Hauptspeicher vorhanden (und nicht etwa ausgelagert) sind."
*"Assume a system with **two-level page-tables** and a **hardware-controlled TLB**. Enumerate in **detail** and in the **correct sequence** all individual steps that the **processor** is executing in order to **load a byte** from the **logical address lx** into a **processor register** assuming that the **TLB** initially does **not** contain **any valid TLB entry**? Regard carefully how the processor gets the necessary information where inside the memory the needed data are located. For reasons of simplicity you can assume that the processor does not have a cache, i.e. all data have to be fetched from RAM. Further assume that all needed data are in RAM, i.e. they are not paged out."*

   a) **For logical address lx there is no TLB-entry, i.e. a TLB-miss will be raised, handled by the HW-controlled TLB.**
   b) **This HW-controlled TLB acc4esses the primary page-table, whose starting address is in the page-table base register of the processor. The offset in the primary page-table will be calculated using the high ordered bits of lx. Result: physical address of the secondary page-table**
   c) **Again the HW-controlled TLB accesses the secondary page-table, this time using the low ordered bits of lx to get the corresponding offset in this secondary page-table. Result: physical address y, the logical address lx will be mapped to.**
   d) **Performing the corresponding TLB-entry at least containing lx and y, but in many cases other control-bits will be part of this TLB entry, too.**
   e) **Load the data stored a y into the register.**

2. „„Ein **Zugriff** auf den Hauptspeicher benötige **40ns**. Die Wahrscheinlichkeit, dass zu einer logischen Adresse lx ein **Eintrag im TLB** vorhanden ist, betrage **98%.** Berechnen Sie die **effektive Zugriffszeit**. (Vernachlässigen Sie dabei die zur Adressberechnung und zum Nachschlagen im TLB benötigte Zeit und berücksichtigen Sie nur die für Speicherzugriffe benötigte Zeit.)"
"An **access** to main memory (RAM) takes **40ns.** The probability that there is a TLB entry for a logical address lx is 98%. Calculate the **effective access time**. Neglect the times needed to calculate the address and for accessing the TLB, but include only the time needed for RAM acceses.)

(you need at least 1 access to RAM in case of a TLB hit, otherwise 3 accesses to RAM)
EAT = 40 (0.98 + 0.02 * 3) = 40 *1.04 = 41.6

3. „Was wird durch ein **Referenzbit** angezeigt? Für welchen Zweck kann man dieses Referenzbit verwenden?“
*“What does a **reference-bit** indicate? For what purpose can you use the reference-bit?”*

The reference bit indicates whether the corresponding page has been referenced at least once since this bit has been controlled the last time.

Controlling this control-bit might help the paging algorithm to distinguish between pages that have been referenced in the very past and those that have not been referenced, i.a. any LRU-like paging algorithm can be supported.

Aufgabe 5 / *Question 5* (1 + 1 + 3 + 4 + 3 Punkte/*marks*)

1. „Erläutern Sie den Begriff **logische Blockgröße** eines **Dateisystems**!“
*“Explain the term **size** of a **logical block** of a **file system**.”*

A logical block of a file system is the minimal portion of contiguous storage medium on which the file system is implemented.

2. **“**Welche **Randbedingungen** können einen Systemarchitekten dazu bewegen, sich für **große** bzw. für **kleine** logische Blockgrößen beim Dateisystementwurf zu entscheiden.
*“Designing its new file system, what **constraints** can a system architect **influence** that he opts for **small logical blocks**.”*

Whenever the system has to support many small files, you better chose small logical blocks. Those small files will induce few or only moderate internal fragmentation, i.e. the disk will be used more effectively.
Whenever the majority of the files are moderate or even large, you better chose large logical blocks, because the internal fragmentation only affects the last block of the files. Furthermore the data transfer of one large block is better than that one of multiple small blocks, furthermore you have fewer meta data of a file when using large blocks.

3. „Zählen Sie **vier verschiedene Dateisysteme** auf!“
*“Enumerate **four different file systems**.”*

BtrFS, EXT4, XFS, ZFS, Reiser

4. „Zählen Sie **vier verschiedene Metadaten** von **Dateisystemen** auf!“
*“Enumerate **four different meta data** of **file systems**.”*

Block size(s), inode-table, super-block, link to root, journal, special containers for very small files, directory

5. „Es gibt zwei Arten von Verweisen auf Dateien: **Hard-Links** und **Symbolic Links**.
a) Beschreiben Sie kurz den Unterschied zwischen den beiden Arten!

b) Was passiert in beiden Fällen mit der Zieldatei, wenn der Link gelöscht wird?
"There are two kinds of references to a file: ***hard-links*** *and* ***symbolic-links****.*
a) Describe in short the difference between both kinds of links.
b) What happens to the target-file when the link will be deleted?"

a) A hard link is another file-name, i.e. a directory-entry. The number of current hard-links is mapped to the reference-counter, i.e. any new hard-link increments the reference counter. A hard-link is restricted to the file-system partition where the file is located.
A soft link is a new file that contains a pathname to the file, you can even have a soft link to another file system or even to a file that is currently deleted or on an unmounted part of the file system.

b) Whenever you try to delete a file, that has a reference counter greater than 1, you only decrement the reference counter. Deleting a file is deleted until the reference counter is zero. Whenever you delete a soft link nothing happens to the target file