 University of Karlsruhe System Architecture Group Frank Bellosa, Gerd Liefländer		
<b>Nachname / Last name</b>	<b>Vorname / First name</b>	<b>Matrikelnummer / Matriculation number</b>

System Architektur / *System Architecture*  
 Klausur / *Examination*  
 WS 2006/2007, 29.03.2007

- Bitte tragen Sie zuerst auf dem Deckblatt Ihren Namen, Vornamen und Ihre Matrikelnummer ein, auf den sonstigen Blättern nur noch Ihre Matrikelnummer, auch auf dem Konzeptblatt. *Please enter your last name, first name, and matriculation number on this page and your matriculation number on all other pages (including the draft page).*
- Die Prüfung dauert 60 Minuten und besteht aus 5 Aufgaben auf 11 Seiten und einem Konzeptblatt. *You have 60 minutes to complete your answers. The examination consists of 5 questions on 11 pages. You have received one additional blank page for drafts, etc.*
- Die Prüfung wäre mit mindestens 20 Punkten von 60 erreichbaren Punkten bestanden. *You pass the examination by obtaining at least 20 marks out of the possible 60 marks.*
- Es sind keinerlei Hilfen erlaubt! *No additional means are 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 more draft pages please notify one of the supervisors.*
- Bitte machen Sie eindeutig klar, was Ihre endgültige Lösung zu den jeweiligen Teilaufgaben ist. Teilaufgaben mit mehreren Lösungen oder mit widersprüchlichen Teilen werden mit 0 Punkten bewertet. *Make sure that it is absolutely clear what your final solution is for each question. Questions with multiple solutions or with contradicting parts are void: 0 marks.*

Die folgende Tabelle wird von uns ergänzt! *The table below is completed by us!*

Aufgabe / <i>Question</i>	1	2	3	4	5	Total
Erreichbare Punkte / <i>Possible marks</i>	12	12	12	12	12	60
Erreichte Punkte / <i>Obtained marks</i>						
Note / <i>Grade</i>						

<b>Matrikelnummer/Matriculation number</b>	
--	--

**Aufgabe 1 / Question 1 (Zum Aufwärmen/Warm up, 2 + 2 + 2 + 1 + 5 Punkte/marks)**

1. „Hinsichtlich der **Speicherverwaltung** werden **statische** und **dynamische** Bereiche (*regions*) unterschieden. Zählen Sie unter der Rubrik ‚**static**‘ **zwei statische Adressbereiche** und unter der Rubrik ‚**dynamic**‘ **zwei dynamische Adressbereiche** von Linux-Anwendungen auf!“  
*“Concerning memory management we discriminate **static** from **dynamic** regions. Enumerate two static address regions of a Linux application in the column ‘static’ and two dynamic address regions in the column ‘dynamic’.”*

**static**

**dynamic**

.....  
 .....

.....  
 .....

2. „Zählen Sie **vier verschiedene POSIX** oder **Linux Systemaufrufe** der **Thread Management API** auf.“  
*“Enumerate **four different POSIX** or **Linux system calls** of the **thread management API**.”*

.....  
 .....

.....  
 .....

3. „Zählen Sie **vier verschiedene Kerndatentypen** (*kernel data types*) auf.“  
*“Enumerate **four different kernel data types**.”*

.....  
 .....

.....  
 .....

Einige der folgenden Aussagen sind korrekt, einige sind 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.*

4. „Um in den Kern einzutreten, braucht man einen **privilegierten Befehl**.“  
*“To enter the kernel you need a **privileged instruction**.”*

korrekt

inkorrekt



<b>Matrikelnummer/Matriculation number</b>	
--	--

**Aufgabe 2 / Question 2****(4 + 3 + 5 Punkte/marks)**

1. „Gegeben sei ein **Einprozessorsystem** mit **nicht verdrängender Prioritätsplanungsstrategie** (*non-preemptive priority scheduling policy*) und **statischen Prioritäten**. **Unterstreichen** Sie von den folgenden Problemen all diejenigen, die auftreten können.“

*“Assume a **single-processor system** with a **non-preemptive priority scheduling policy** and **static priorities**. **Underline** all of the following problems that can occur.”*

Prozesse können aushungern / *processes can starve*

Durch Prioritätsumkehr können niedrigst priore Prozesse aushungern / *lowest priority processes can starve due to priority inversion*

Die Auswahlfunktion arbeitet ineffizient bei vielen bereiten Prozessen / *the selection function works inefficiently if many processes are ready*

Die Verwendung von Spinlocks zur Sicherung kritischer Abschnitte kann zum Systemstillstand führen / *Using spin locks to guard critical sections can lead to a system life-lock*

2. „Welche **unerwünschten** Auswirkungen kann die **unbedarfte** Verwendung des **Test-And-Set-Befehls** zur **Synchronisation von Anwendungen** auf **Mehrprozessorsystemen** haben?“

*“What **undesired** effects can be caused by **carelessly** using the **Test-And-Set instruction** for **synchronizing applications in multi-processor systems**?”*

<b>Matrikelnummer/Matriculation number</b>	
--	--

**Fortsetzung von Aufgabe 2 / Question 2 continued**

**(5 Punkte/marks)**

3. „Analysieren Sie die folgende Programmskizze des **allgemeinen Erzeuger/Verbraucher-Problems mit begrenztem Puffer** (*bounded buffer*) in gewohnter Weise! **Verbessern** Sie das Programm mit den gegebenen Semaphoren so, dass es eine **effektive Lösung** wird.“  
 “*Analyze the following program draft of the general producer-consumer problem with a bounded buffer in the usual way. Improve the program with the given semaphores to achieve an effective solution.*”

```

/* global data declarations */
item BUFFER[100];          /* bounded buffer of length 100 */
int i=0; int j=0;          /* indices to fetch & store */
bin_sem plock=1;          /* Only 1 producer can access buffer */
bin_sem cclock=1;         /* Only 1 consumer can access buffer */
count_sem free=100;       /* controls # of free buffer slots */
count_sem full=0;         /* controls # of occupied buffer slots */

producer()
{
    item x;
    x = produce();          /* can be time consuming */
    p(free);
    p(plock);
    BUFFER[i] = x;         /* put product into buffer */
    i=(i+1)%100;
    v(plock);
    v(full);
}

consumer()
{
    item x;
    p(full);
    p(cclock);
    x = BUFFER[j];        /* fetch product from buffer */
    v(cclock);
    j=(j+1)%100;
    v(free);
    consume(x);           /* can be time consuming */
}

```

<b>Matrikelnummer/Matriculation number</b>	
--	--

**Aufgabe 3 / Question 3****(1 + 2 + 2 + 2 + 5 Punkte/marks)**

1. „Erläutern Sie die **Auswirkung** eines blockierenden Systemaufrufs in einem PULT!“  
*“Explain the **implication** of a blocking system call within a PULT.”*
  
2. „Erläutern Sie, wie die **K42** Systemarchitekten die obige Auswirkung gemildert haben!“  
*“Explain how the system architects of **K42** improved the implication above.”*
  
3. „Erläutern Sie den Begriff **Seitendiebstahl** (*page stealing*)! **Wo** und **wann** kann Seitendiebstahl vorkommen?“  
*“Explain the notion **page stealing**. **Where** and **when** can page stealing occur?”*
  
4. „Kann das folgende System aus zwei Prozessen A und B sowie den zwei exklusiven Betriebsmitteln R und S verklemmen? **Begründen** Sie Ihre Meinung!“  
*“Can the following system consisting of the two processes A and B and the two exclusive resources R and S enter a deadlock? **Exemplify** your choice.”*  
**A: allocate(R), allocate(S), release(S), release(R)**  
**B: allocate(S), release(S), allocate(R), release(R)**

Matrikelnummer/Matriculation number	
-------------------------------------	--

Fortsetzung von Aufgabe 3 / Question 3 continued

(5 Punkte/marks)

5. „Gegeben sei ein Einprozessorsystem mit drei exklusiven Betriebsmitteln R, S und T sowie vier Prozessen A(1), B(2), C(3) und D(4) (die Zahlen in Klammern seien die Prozessprioritäten). Vervollständigen Sie den untenstehenden Ablaufplan, wobei **strikte verdrängende Prioritätsplanungsstrategie** und das Betriebsmittelvergabeprotokoll **Prioritätsvererbung** angenommen werden. Beachten Sie, dass die Prozesse erst zu den fett markierten Zeitpunkten (also **0, 3, 6** und **7**) bereit werden. Tragen Sie gemäß der Vorlage an allen Stellen, an denen sich die Prioritäten der Prozesse ändern, die jeweils neuen Prioritätswerte ein.“

“Assume a single-processor system with three exclusive resources R, S, and T, and four processes A(1), B(2), C(3), and D(4), where the numbers in parentheses denote the priorities of these processes. Complete the schedule below according to **strict priority driven scheduling with preemption** and the resource allocation protocol **priority inheritance**. Pay attention: Processes become ready only at the bold-marked points in time (i.e., **0, 3, 6, and 7**). Whenever priorities will change, add the new value in the schedule below.”

Operationsfolge pro Prozess, wobei eine Rechenphase “c” jeweils eine Zeiteinheit, alle anderen Operationen jeweils 0 Zeiteinheiten dauern und stets am Ende einer Rechenphase c ausgeführt werden sollen:

Sequence of operations per process; each compute phase “c” takes exactly one time unit, all other operations need 0 time units and will take place at the end of a compute phase c.

**D:** c, allocate(S), c, allocate(T), c, release(S), c, release(T), c, exit

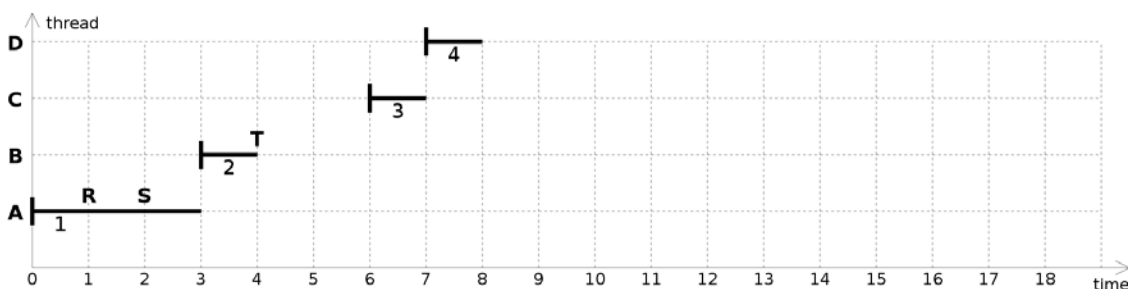
**C:** c, c, exit

**B:** c, allocate(T), c, allocate(R), c, release(R), c, release(T), c, exit

**A:** c, allocate(R), c, allocate(S), c, c, release(S), c, release(R), c, c, exit

Wenn Sie in der unten angeführten Skizze das Ereignis „allocate(X)“ für  $X \in \{R, S, T\}$  markieren wollen, benutzen Sie den Großbuchstaben **X**, für das Ereignis „release(X)“ benutzen Sie **x**.

Use a capital **X** if you want to mark the event “allocate(X)” with  $X \in \{R, S, T\}$  in the chart below; for the event “release(X)” use a lowercase **x**.



Hinweis:

Haben Sie sich verschrieben, finden Sie auf dem Konzeptblatt noch ein weiteres Diagramm.

Hint:

In case you made too many mistakes, there is another diagram on the draft page.

Matrikelnummer/ <i>Matriculation number</i>	
---	--

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

1. „Erläutern Sie, warum beim Threadwechsel zwischen Kernel-Level Threads (KLTs) innerhalb der **Prozedur** `THREAD_SWITCH` der **Befehlszeiger** (*instruction pointer*) **nicht gerettet** werden muss, während dies beim **Makro** `THREAD_SWITCH` **nötig** ist.“  
“Describe why you **do not have to save the instruction pointer** when switching between two kernel-level threads (KLTs) using the **procedure** `THREAD_SWITCH`, whereas this is **necessary** when using a **macro** `THREAD_SWITCH`.”
2. „Warum ist der **fork()** Systemaufruf in modernen Linux-Implementierungen nicht mehr so aufwändig wie er dies in frühen Unix Versionen war? **Erläutern** Sie **ausführlich**, was das System für die **neue Task bereitstellen** muss.“  
“Why is the system call **fork()** in modern Linux implementations no longer as complex as it was in early Unix versions. **Explain in detail** what the system has to **provide** for the **new task**.”
3. „Diskutieren Sie die **Orthogonalität** der folgenden beiden Kommunikationsvarianten: **verbindungsorientierte** bzw. **lose IPC** und **direkte** bzw. **indirekte IPC!**“  
“Discuss the **orthogonality** of the following two communication variants: **connection-oriented** versus **connectionless IPC** and **direct** versus **indirect IPC**.”

<b>Matrikelnummer/Matriculation number</b>	
--	--

**Fortsetzung von Aufgabe 4 / Question 4 continued**

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

4. „Geben Sie möglichst präzise an, in welchen Adressbereichen (*regions, sections*) die jeweiligen Variablen des folgenden C++ Programms in einem modernen Linux-System abgespeichert werden.“

*“Indicate as precisely as possible in which address regions (sections) the variables of the following program will be stored.”*

```
static long A[4];
unsigned B[4] = {1, 2, 3, 4};

long mult(int j, long *X, long *Y) {
    for (int i = 0; i < j; i++) {
        X[i] = Y[i] * Y[i];
    }
}

void main(void) {
    long TEMP[4];
    mult(4, &TEMP[0], &A[0]);
    for (int i = 0; i < 4; i++) {
        printf("%ld\n", TEMP[i]);
    }
}
```

5. „Beim Seitentausch auf Verlangen (*demand paging*) ist es mitunter ratsam, **Kacheln** mit **Nullen zu füllen**, bevor sie der Anwendung zugänglich gemacht werden. **Welche Seiten** sollten „genullte“ Kacheln erhalten und **warum „nullt“** man die Kacheln überhaupt?“

*“With paging on demand it can be advisable to **zero-fill page frames** before making them accessible to the application. **Which pages** should be backed by zeroed frames and **why** should you **zero-fill** them at all?”*

<b>Matrikelnummer/Matriculation number</b>	
--	--

**Aufgabe 5 / Question 5****(1 + 1 + 4 + 6 Punkte/marks)**

1. „Welche Angaben enthält ein Verzeichniseintrag (*directory entry*) im **Linux EXT2-Dateisystem**?“

“Which fields does a directory entry of the **Linux file system EXT2** contain?”

.....

2. „Welches der folgenden **Dateiattribute** ist **nicht** Bestandteil des Dateikopfes (*inode*) einer Linux EXT2 Datei?“ (Zutreffendes unterstreichen.)

“Which of the following **file attributes** is **not** part of an *inode* of a Linux EXT2 file?”  
(Underline appropriately.)

Dateilänge / *file length*

Anzahl symbolischer Links / *number of symbolic links*

Zugriffsrechte / *accessrights*

Eigentümer / *owner*

3. „Erläutern Sie so knapp und präzise wie möglich wie Sie erreichen können, dass Sie stets den aktuellsten und konsistenten Inhalt einer **mehrfach geöffneten B\*-Indexsequentiellen Datei** erhalten, egal ob Sie sequentiell oder über einen Schlüssel zugreifen.“

“Describe as short but as precisely as possible how you can achieve that you always get the most current and consistent content of a **multiply opened B\*-indexed sequential file**, regardless of whether you are accessing the file sequentially or via a key.”



<b>Matrikelnummer/Matriculation number</b>	
--	--

