

	University of System Architecture Group Gerd Liefländer	Tutors: Uwe Dannowski Joshua LeVasseur Espen Skoglund
---	--	--

<i>Nachname/Last name</i>	<i>Vorname/First name</i>	<i>Matrikelnummer/ Matriculation number</i>

System Architektur (Architecture)

Zwischenklausur (Voluntary Examination)

WS 2003/2004 am 19. December 2003

- Bitte tragen Sie zuerst auf dem Deckblatt Ihren Namen, Vornamen und Ihre Matrikelnummer ein, auf den sonstigen Blättern nur noch ihre Matrikelnummer, auch auf den Konzeptblättern. *Please enter your last name, first name and matriculation number on this page and your matriculation number on all other pages (including used and unused draft pages).*
- Die Prüfung dauert 60 Minuten und besteht aus 5 Aufgaben auf 11 Seiten und zwei Konzeptblättern. *You have 60 minutes to complete your answers. The examination consists of 5 questions on 11 pages. You have received two additional blank pages 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 subquestion. Subquestions with multiple solutions or with contradicting parts are void: 0 marks.*

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

<i>Aufgabe/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/Grade:						

Matrikelnummer/Matriculation number	
--	--

Aufgabe1/Question 1 (Zum Aufwärmen/Warm up, 3 + 3 + 1 + ... + 1 Punkte/marks)

1. „Zählen Sie **drei konkrete Anwendungen** auf, in denen man einen **m:n-Kommunikationskanal** verwenden kann, wobei $m, n > 1!$ “

“Enumerate **three concrete applications** for which you can use an **m:n-channel** ($m, n > 1$)”.

a)

.....

b)

.....

c)

.....

2. „Geben Sie **drei verschiedene** konkrete Anwendungen des **Timeout-Mechanismus** an!“

“Enumerate **three different** concrete applications of the **time-out** mechanism.”

a)

b)

c)

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. „Auf einem **Mehrprozessorsystem** mit **Mehrprogrammbetrieb** (*multi programming*) können **gleichzeitig mehrere** „**Reine User-Level-Threads**“ (*PULTs*) im Threadzustand „**Rechnend**“ (*running*) sein.“

“On a **multi-processor** with **multi programming** multiple **pure-user-level-threads** may be in the thread state “**Running**” at the same time.”

korrekt

inkorrekt

Matrikelnummer/Matriculation number	
--	--

Fortsetzung von Aufgabe 1 / Question 1 continued: (1+1+1+1+1 Punkte/marks)

4. „Ein **Anwendungssystem** aus $n > 1$ Kernel-Level-Threads kann verklemmen, wenn diese Threads **geschachtelte Monitore** verwenden.“
 “An **application system** of $n > 1$ kernel-level-threads may deadlock, if its threads use **nested monitors**.”

korrekt

inkorrekt

5. „Annahme: Eine Applikation benutzt eine **Semaphore S**, um den **exklusiven** Zugriff ihrer Threads auf ein Betriebsmittel R zu **synchronisieren**. Wenn in einem der Threads, die das Betriebsmittel R benutzen, ein **P(S)** Aufruf fehlt, so ist die **exklusive Nutzung** des Betriebsmittels R **nicht mehr garantiert**.“
 “Suppose an application is using a **semaphore S** to **synchronize** its threads for **exclusive** access to a resource R. If a **P(S)** is missing in a thread using the resource R, the **exclusive usage** of this resource R is **no longer guaranteed**.”

korrekt

inkorrekt

6. „Jeder **blockierende Systemaufruf** (*system call*) bewirkt einen **Taskwechsel**.“
 “Every **blocking system call** invokes a **task switch**.”

korrekt

inkorrekt

7. „Annahme: Außer einer Anwendung (*application*) existiert ansonsten **keine Last** auf dem System. Diese Applikation aus reinen „User-Level-Threads“ kann in einem **Mehrprozessorsystem nicht schneller** als auf einem Einprozessorsystem abgearbeitet werden.“
 “Suppose besides the application there is **no other load** on the system. This application with pure user-level-threads can **not be executed faster on a multiple-processor system than on a single-processor-system**.”

korrekt

inkorrekt

8. „In **Java** werden zur Synchronisation von Threads **Monitore** angeboten.“
 “Java offers **monitors** to synchronize threads.”

korrekt

inkorrekt

Matrikelnummer/Matriculation number	
--	--

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

1. „Diskutieren Sie die Implikationen, wenn die zentrale Umschaltfunktion **thread_switch** als **Makro** implementiert wird?“

*“Discuss the implications if the central switch function **thread_switch** is implemented as a macro.”*

.....

.....

.....

.....

2. „Erklären Sie die **Hauptunterschiede** zwischen **Kernel-Level-(managed) Threads** und **User-Level-(managed) Threads**“

*“Explain the **main differences** between **kernel-level-(managed)-threads** and **user-level-(managed)-threads**.”*

Matrikelnummer/Matriculation number	
--	--

Fortsetzung von Aufgabe 2 / Question 2 continued:

(6 Punkte/marks)

3. „Sie sollen das **allgemeine Produzenten/Verbraucherproblem** ($p, c \geq 1$) lösen, wobei unabhängig voneinander mal Produzenten, mal Konsumenten schneller vorankommen können. Welches Threadmodell werden Sie verwenden? Begründen Sie Ihre Wahl? Skizzieren Sie eine effiziente Lösung!“

“Solve the general producer/consumer problem whereby independently of each other, producers or consumers may outrun the other. What thread-model will you apply? Reason your choice. Outline an efficient solution.”

Thread-Model(l):

Begründung (*reason*):

.....

.....

Skizze (outline):

Matrikelnummer/Matriculation number	
--	--

Aufgabe 3 / Question 3 :

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

1. „Skizzieren Sie inklusive aller Übergangsfunktionen das Threadzustandsdiagramm von Java!“

“Outline the thread state diagram of Java including all state- transition functions.”

2. „Beschreiben Sie so knapp wie möglich, gleichwohl so ausführlich wie nötig das **Lotterie-Planungsverfahren!**“

*“Describe as short as possible, but as completely as necessary **lottery scheduling.**”*

.....

.....

.....

.....

3. „Beschreiben Sie so knapp wie möglich das Prinzip: „**Trennung von Strategie und Mechanismus**“, und geben Sie hierfür ein einleuchtendes Beispiel an!“

*“Describe as short as possible the meaning of the principle: “**separation of policy and mechanism**”, and give a clear example.”*

.....

.....

.....

.....

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

Fortsetzung von Aufgabe 3 / *Question 3 continued:*

(4 Punkte/*marks*)

4. „Das Leser-/Schreiberproblem kann auf verschiedene Arten formuliert werden, wenn man berücksichtigt, welche Art von Prozessen (Leser oder Schreiber) zu welchem Zeitpunkt gestartet werden kann. Beschreiben Sie sorgfältig zwei verschiedene Lösungsvarianten des Problems. Schildern sie für jede Variante, was passiert, wenn ein Leser oder Schreiber bereit wird, auf die Datenbank zuzugreifen, und was passiert, wenn ein Prozess terminiert.“

“The reader-/writer-problem can be formulated in several ways with regard to the ordering in which category of processes (reader or writer) can be started. Carefully describe two different variations of the problem. For each variation, specify what happens when a reader or writer becomes ready to access the data base, and what happens when a process terminates.”

4.1

4.2

Matrikelnummer/Matriculation number	
--	--

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

1. „Ein Rechner hat acht Bandlaufwerke, um die n Prozesse konkurrieren. Jeder Prozess braucht bis zu 2 Bandlaufwerke. Für welche n kann es **keine Verklemmung** (*deadlock*) geben?“
*“A computer has eight tape drives, with n processes competing for them. Each process may need up to two tape drives. For which values of n is the system **deadlock free**?”*

.....

2. „In einem elektronischen Überweisungssystem laufen Tausende von identischen Prozessen. Jeder Prozess liest die Überweisungssumme und die beiden Kontonummern von einem Eingabegerät. Anschließend sperrt er beide Konten, überweist das Geld und gibt dann die Konten wieder frei. Die Verklemmungsfahr ist in diesem System sehr hoch. Lassen Sie sich ein Protokoll einfallen, um Verklemmungen zu verhindern. Dabei darf kein Konto wieder freigegeben werden, bevor die Überweisung beendet ist. (Mit anderen Worten, eine Lösung, die ein Konto sperrt und es sofort wieder freigibt, falls das zweite Konto gesperrt ist, gilt nicht.)“

“In an electronic funds transfer system, there are thousands of identical processes that work as follows. Each process reads an input line specifying an amount of money, the account to be credited, and the account to be debited. Then it locks both accounts and transfers the money, releasing the locks when done. With many processes running in parallel, there is a very real danger that having locked account x it will be unable to lock the other account y , because y has been locked by a process now waiting for x . Devise a protocol that avoids deadlocks. Do not release the lock of an account until you have completed the transaction. (In other words, solutions that lock one account and then release it immediately if the other lock is already locked are not allowed.)”

.....

.....

.....

.....

.....

.....

.....

.....

Matrikelnummer/Matriculation number	
--	--

Fortsetzung von Aufgabe 4 / Question 4 continued:

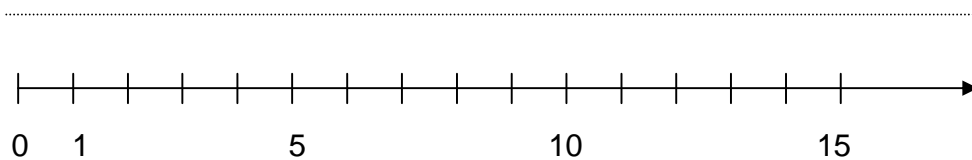
(6 Punkte/marks)

3. „Fünf Prozesse, P1 bis P5, kommen gleichzeitig in einem Einprozessorsystem an. Die Ausführungszeiten, Sollzeitpunkte und Prioritäten (5 ist die höchste Priorität) sind durch folgende Tabelle gegeben. Stellen Sie für jede der drei folgenden Umschaltstrategien den Ablauf durch ein Gantt-Diagramm dar.“

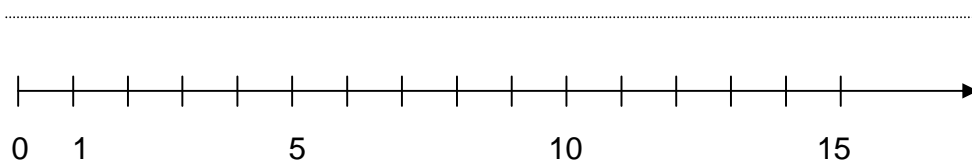
“Five processes, P1 ... P5 arrive at a single-processor system at the same time. The execution times, the deadlines, and the priorities are given in the following table. Show the resulting schedule for the following three scheduling policies in the appropriate Gantt diagram.”

Process	Execution Time	Deadline	Priority
P1	5	16	4
P2	4	15	1
P3	1	1	2
P4	2	3	5
P5	3	14	3

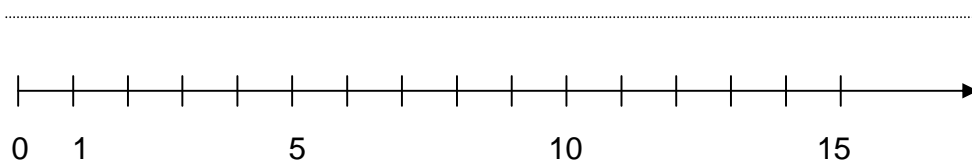
3.1 Shortest Job First (SJF):



3.2 Priority based Scheduling:



3.3 Earliest Deadline First (EDF):



Matrikelnummer/Matriculation number	
--	--

Aufgabe 5/ Question 5:**(2 + 3 + 1 + 6 Punkte/marks)**

1. „Erläutern Sie den Begriff: **Erweiterbarer Kern!** Geben Sie ein **konkretes Beispiel** für einen erweiterbaren Kern an!“

“*Explain the notion: **Extensible Kernel**. Give a **concrete example** for an extensible kernel.*”

.....

.....

.....

2. „Zählen Sie wenigstens 3 verschiedenartige Bibliotheksfunktionen auf, die **keinen Systemcall aufrufen**.“

“*Enumerate at least three different library functions that do **not invoke a system call**.*”

.....

.....

.....

3. „Worin liegt der wesentliche Unterschied zwischen einem **Systemcall** und **sonstigen Ausnahmen (exception)**?“

“*What is the main difference between a **system call** and **other exceptions**?*”

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

Fortsetzung von Aufgabe 5 / Question 5 continued: (6 Punkte/marks)

4. „Ergänzen sie die unten angegebene Programmschablone so, dass sie eine **atomare** und **orthogonale** Lösung einer **starken Zählersemaphore** (*strong counting semaphore*) für ein **Einprozessorsystem** darstellt.“

“Complete the following program pattern to represent an **atomic** and **orthogonal** solution of a **strong counting semaphore**.”

```

module export P2,V2
  import BLOCK, UNBLOCK
  type countsem = record

    QWT: list of Threads = empty      {waiting threads}

  end

P2(S: countsem)
{

}

V2(S: countsem)
{

}

}
end atomic orthogonal strong counting semaphore.

```

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

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