

Aufgabe 1: Schichtenmodell

5 Punkte

- (a) Zeichnen Sie das Schichtenmodell des Mach-Kerns und kennzeichnen Sie die Schnittstellen. (2Punkte)
- (b) Benennen Sie die Vor- und Nachteile des Aufbaus des Mach-Betriebssystems. (2Punkte)
- (c) Welchen Vorteil bietet es, möglichst viele Funktionalitäten in den Kernel-Mode zu verlagern? (1Punkt)

Aufgabe 2: Software-Hardware-Migration

5 Punkte

- (a) In Großrechnern gibt es oft keine I/O-Driver, sondern stattdessen Hardwareeinheiten mit gleichen Schnittstellen. Welche Vor – und Nachteile hat eine solche Software-Hardware-Migration? (2 Punkte)
- (b) Geben Sie ein weiteres Beispiel für eine Software-Hardware-Migration an. (2 Punkte)
- (c) Wenn Sie die Wahl zwischen einer Software- und einer Hardwarelösung hätten, welcher Lösung würden Sie den Vorzug geben? Begründen Sie ihre Antwort. (1 Punkt)

Aufgabe 3: Definitionen

10 Punkte

- (a) Was ist der Unterschied zwischen einer abstrakten und einer virtuellen Maschine? (2 Punkte)
- (b) Was sind Prozesse im Unterschied zu *lightweight*, *heavyweight* sowie *User*- und *Kernel*-Threads? (4 Punkte)
- (c) Was versteht man unter *Apartments* in Windows NT? Wozu werden sie verwendet? (4 Punkte)

Aufgabe 4: Multiprozessorscheduling**20 Punkte**

Angenommen, Sie haben ein 3-Prozessorsystem und versuchen, eine Menge von Threads eines Prozesses besonders effizient zu synchronisieren. Gegeben seien dazu folgende Präzedenzrelationen sowie in Klammern die Joblängen (in Zeiteinheiten):

$B \gg D$, $C \gg F$, $B \gg E$, $E \gg F$, $D \gg G$, $G \gg H$, $F \gg H$, $A \gg C$, $D \gg F$

A(1), B(2), C(3), D(2), E(3), F(2), G(1), H(1)

- (a) Erstellen Sie den zu der Präzedenzrelation zugehörigen *Präzedenzgraphen* (8 Punkte)
- (b) Geben Sie den *kritischen Pfad* in dem Graphen an. (2 Punkte)
- (c) Erstellen Sie jeweils einen *Schedul* für *latest scheduling* sowie für *earliest scheduling* mittels eines Gantt-Diagramms. (8 Punkte)
- (d) Was ist dabei der Unterschied zwischen *Scheduling* und *Dispatching*? (2 Punkte)

Aufgabe 5: Prozess-Synchronisation**20 Punkte**

Sei ein paralleles Programm zur Implementierung eines Produzenten/Konsumenten-Verhältnisses durch den folgenden Pseudocode gegeben. Produzent und Konsument kommunizieren über eine gemeinsame Puffervariable p : $\text{append}(p,v)$ fügt ein Produkt v in den Puffer ein, $\text{take}(p)$ entnimmt ein Produkt aus dem Puffer.

Variable n gibt jeweils die Anzahl Produkte im Puffer an. Variable s und d sind binäre Semaphore. Alle Variablen werden sowohl im Produzenten- als auch im Konsumenten-Thread benutzt, lediglich lokal ist eine Variable, die nur lokal im Konsumenten auftritt.

Das parbegin Statement in der Hauptfunktion $\text{main}()$ lässt einen Produzenten und einen Konsumenten-Thread parallel ablaufen.

<pre> int n = 0 ; Puffer p ; binary_semaphore s = 1 ; binary_semaphore d = 0 ; void producer(){ while (true) { product v = produce(); wait(s); append(p,v); n = n + 1 ; if (n == 1) signal(d); signal(s); } } </pre>	<pre> void consumer(){ int lokal ; wait(d) ; while (true) { product v ; wait(s); v = take(p); n = n - 1 ; lokal = n ; signal(s) ; consume(v) ; if (lokal == 0) wait(d); } } void main(){ n = 0; initialisiere(p) ; parbegin(producer,consumer); } </pre>
---	---

- Die Implementierung nutzt binäre Semaphore. Erklären Sie zunächst deren Funktionsweise: Welchen Effekt hat ein Aufruf von $\text{wait}(s)$ bei $s = 0$ und $s = 1$, welchen Effekt hat ein Aufruf von $\text{signal}(s)$ bei $s = 0$ und bei $s = 1$? (4 Punkte)
- Welche Variable müssen im wechselseitigen Ausschluss behandelt werden? Begründen Sie ihre Antwort. (2 Punkte)
- Markieren Sie die kritischen Abschnitte in dem angegebenen Pseudocode. (2 Punkte)
- Sichert die Implementierung den wechselseitigen Ausschluss zu? Begründen Sie Ihre Aussage. (6 Punkte)
- Ist eine Flußkontrolle vorhanden? Begründen Sie Ihre Aussage. (6 Punkte)

Matrikelnummer:

Aufgabe 6: Verklemmungen**15 Punkte**

Bei parallelen und konkurrenten Aktionen kann es zu Verklemmungen kommen.

(a) Nennen Sie die in der Vorlesung genannten Bedingungen, die notwendig sind, damit es zu Verklemmungen kommen kann. (4 Punkte)

1.

2.

3.

4.

(b) Ein Szenario aus 4 Prozessen, P1, P2, P3 und P4, und 3 Arten von Ressourcen, R1, R2, R3 sei gegeben. **Sei R1 neunfach, R2 dreifach und R3 sechsfach vorhanden.** Kann mit der nachfolgend genannten Belegung eine Verklemmung auftreten? Begründen Sie Ihre Entscheidung. (7 Punkte)

<i>Matrix der maximalen Anforderungen</i>			
	R1	R2	R3
P1	3	1	4
P2	4	2	2
P3	3	2	2
P4	6	1	3

<i>Matrix mit aktueller Ressourcenbelegung</i>			
	R1	R2	R3
P1	2	1	1
P2	0	0	2
P3	1	0	0
P4	6	1	2

- (c) Wie würden Sie nach dem Banker-Algorithmus entscheiden, ob in obiger Situation die Anforderung des Prozesses P3 nach zwei weiteren Ressourcen R2 erfüllt werden darf? Skizzieren Sie knapp das prinzipielle Vorgehen. (4 Punkte)

Aufgabe 7: Speicherverwaltung

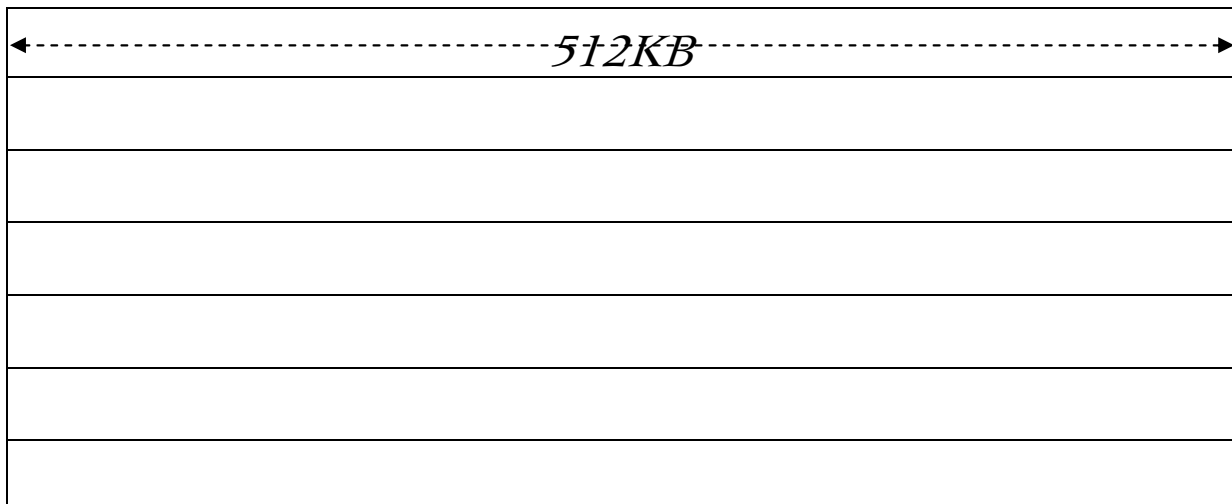
18 Punkte

Die Verwaltung von Heap-Speicher oder anderem, selbst verwalteten Speicher kann mit einfachen Strategien bewältigt werden.

- (a) Folgende freie Speicherstücke seien in der aufgeführten Reihenfolge und genannten Größe gegeben: 8KB, 16KB, 17KB, 9KB, 12KB, 19KB, 18KB und 15KB. Welcher freie Bereich wird jeweils gewählt für eine Anforderung von zunächst (A) 9KB, dann (B) 12KB, anschließend (C) 9KB und zum Schluss (D) 8KB? Beantworten Sie die Frage jeweils für die Strategien *FirstFit*, *NextFit*, *BestFit* und *WorstFit* durch Eintragen der Anforderung in die Tabelle. (6 Punkte)

	<i>8KB</i>	<i>16KB</i>	<i>17KB</i>	<i>9KB</i>	<i>12KB</i>	<i>19KB</i>	<i>18KB</i>	<i>15KB</i>
<i>FirstFit</i>								
<i>NextFit</i>								
<i>BestFit</i>								
<i>WorstFit</i>								

- (b) Angenommen, Sie verwenden das Buddy-System zur Speicherverwaltung Ihres Programm-Heaps. Zu Beginn verfügen Sie über einen freien Speicherblock von 512KB an der Adresse 0. Geben Sie ein Abbild der Speicherbelegung, wenn die folgenden Speicheranforderungen in der angeführten Reihenfolge bedient werden: (A) 129KB, (B) 37KB, (C) 26KB, (D) 31KB, (E) 41KB und (F) 63KB. (10 Punkte)



- (c) Wie groß kann der Verschnitt einer Speicheranforderung für ein Buddy-System im schlimmsten Fall werden? (2 Punkte)

Aufgabe 8: Virtueller Speicher

10 Punkte

- (a) Begründen Sie, warum statt Multi-Level-Tabellen invertierte Seitentabellen genutzt werden. (3 Punkte)
- (b) Was verbirgt sich hinter dem Begriff *Translation Lookaside Buffer*, kurz TLB? (2 Punkte)
- (c) Beschreiben Sie die Funktionsweise eines Assoziativspeichers. (5 Punkte)

Aufgabe 9: B-Baum / B*-Baum**16 Punkte**

Eine entscheidende Rolle bei der Schnelligkeit von Dateisystemen spielt die Organisation der Datenstrukturen.

- a) Zeichnen Sie für folgende Eingabe von Schlüsselwörtern eines Dateisystems

7, 13, 5, 9, 2, 1, 12, 14, 15, 16, 4, 6, 8

einen B-Baum und einen B*-Baum mit $m = 5$. Zeichnen Sie auch Zwischenschritte zu den einzelnen Bäumen! (14 Punkte)

- b) Vergleichen Sie beide Bäume. Was fällt Ihnen auf und warum ist dies so? (2 Punkte)

Aufgabe 10: RAID-Systeme**25 Punkte**

Gegeben sei ein System bestehend aus 8 kleinen und einer großen Festplatte, die unabhängig voneinander ausfallen können. Die Ausfallwahrscheinlichkeit der kleinen Platte betrage p_1 , die der großen p_2 .

- (a) Berechnen Sie die Ausfallwahrscheinlichkeit des Systems für Raid-0. (5 Punkte)
- (b) Berechnen Sie die Ausfallwahrscheinlichkeit des Systems für Raid-1. Nehmen Sie an, dass die große Festplatte einzeln arbeitet und die kleinen Festplatten je paarweise eine Gruppe bilden. (10 Punkte)
- (c) Berechnen Sie die Ausfallwahrscheinlichkeit des Systems für Raid-2. Nehmen Sie an, dass das System einen Fehler korrigieren kann. Der Einfachheit halber darf angenommen werden, dass die Größe der Festplatte keine Rolle spielt bei der Verteilung der Paritätsinformation. (10 Punkte)

Aufgabe 11: Eingebettete Systeme (Bonusaufgabe) 21 Punkte

Echtzeitsysteme bilden eine wichtige Gruppe der eingebetteten Systeme und verlangen durch ihre spezielle Widmung besondere Algorithmen.

(a) Welche drei Eigenschaften muss ein Echtzeitsystem erfüllen? (6 Punkte)

1.

2.

3.

(b) In der Vorlesung wurden für Echtzeitsysteme unter anderem die drei folgenden Strategien zur Erstellung eines Echtzeitschedulingstrategien vorgestellt: (A) *Minimal Deadline First*, (B) *Minimal Processing Time First*, (C) *Rate-Monotonic Scheduling* (RMS).

Erstellen Sie mit diesen drei Strategien für folgende Eingabe einen gültigen Schedule für die ersten 20 Sekunden. Alle Jobs werden zum Zeitpunkt 0 gestartet. (15 Punkte)

Job	Dauer (Sekunden)	Periode (Sekunden)
Task 1	1	4
Task 2	2	6
Task 3	3	10

Matrikelnummer: