



Klausur
Grundlagen der Betriebssysteme/Technische Informatik I

Datum und Uhrzeit: 06.10.2018 10:00 Uhr Bearbeitungszeit: 90 Minuten
Institut: Institut für Verteilte Systeme Prüfer: Prof. Dr. Franz J. Hauck

Vom Prüfungsteilnehmer auszufüllen:

Name: _____ Vorname: _____ Matrikelnummer: _____
Studiengang: _____ Abschluss: _____

Hiermit erkläre ich, dass ich prüfungsfähig bin. Sollte ich nicht auf der Liste der angemeldeten Studierenden aufgeführt sein, dann nehme ich hiermit zur Kenntnis, dass diese Prüfung nicht gewertet werden wird.

Unterschrift des Prüfungsteilnehmers

Optionales Codewort für den Aushang

Hinweise zur Prüfung:

- Bitte Vollständigkeit der Klausur prüfen! (insgesamt 8 Aufgaben auf 10 Seiten)!
- Lösungen bitte nur auf Aufgabenblätter und nicht mit Rot- oder Bleistift schreiben!
- Als Schmierzettel bitte Rückseiten verwenden! Lösungen, die nicht direkt bei der Aufgabe stehen, bitte deutlich kennzeichnen und referenzieren!
- Codewort dient zur zusätzlichen Bekanntgabe inkl. erreichter Punktzahl.

Barcode

Erlaubte Hilfsmittel:

Ein beidseitig handbeschriebenes DIN A4 Blatt.

Vom Prüfer auszufüllen:



Aufgabe	1	2	3	4	5	6	7	8	Σ
Punkte	14	11	10	13	11	15	4	12	90
Erreicht									
Zeichen									

Note: _____

Unterschrift Prof. Dr. Franz J. Hauck

Korrektur-Status:

1 1 2 3 2 1 2 3 1 2 4 1 2 3 4 5 1 2 3 6 1 2 3 4 5 6 7 8 7 1 2 8 1 2 3 4 5

Aufgabe 1: Zahlendarstellung

(14 Punkte)



Ihr IEEE 754 Gleitkommaformat hat einen 32 Bit Aufbau der Form: 1 Bit Vorzeichen s , 8 Bit Exponent e , 23 Bit Mantisse m , mit einer Bias von 127. Die Berechnung des Wertes erfolgt mit der Formel $(-1)^s \cdot 2^{e-127} \cdot 1, m$.

- 1.) Stellen Sie die Zahl $+520,5_{10}$ in diesem Binärformat dar.

$$({}_4P)$$
[illegible]

- 2.) Stellen Sie die Zahl $-BA_{16}$ in diesem Binärformat dar.

 $(4P)$ [illegible]

- 3.) Addieren Sie die beiden zerlegt dargestellten Zahlen. Verwenden Sie den Additionsalgorithmus für IEEE 754 Zahlen. Geben Sie die Binärdarstellung der Summe an: (6 P)

$$(6P)$$

$$s_1 = 0_2, e_1 = 130_{10} = 1000\ 0010_2, m_1 = 1100\ 0000\ 0000\ 0000\ 0000\ 000_2$$

$$s_2 = 0_2, e_2 = 132_{10} = 1000\ 0100_2, m_2 = 0011\ 1000\ 0000\ 0000\ 0000\ 000_2$$

[illegible]

Aufgabe 2: Architektur

(11 Punkte)

Sie erinnern sich an unseren Spielprozessor. Er hat eine kleine Menge von Arbeitsregistern R0 bis R2 sowie die üblichen Register eines Prozessors (Programmzähler, Condition-Code-Register).

- 1.) Der Programmzähler verweist im Speicher auf den Befehl `MOV 3A, R0`, der den Inhalt der Speicherzelle 3A in das Register R0 übernimmt. Beschreiben Sie am Beispiel dieser Instruktion, wie unser Spielprozessor im allgemeinen eine Instruktion bearbeitet. Lassen Sie dabei eine mögliche Unterbrechung außen vor. (6 P)

- 2.) Was passiert bei Teilaufgabe 1, wenn während der Befehlsbearbeitung eine Unterbrechungsbehandlung angefordert wird? Beschreiben Sie auch, wie eine solche Behandlung wieder zur vorherigen Befehlsfolge zurückkehren kann. (5 P)

Aufgabe 3: Scheduling

(10 Punkte)

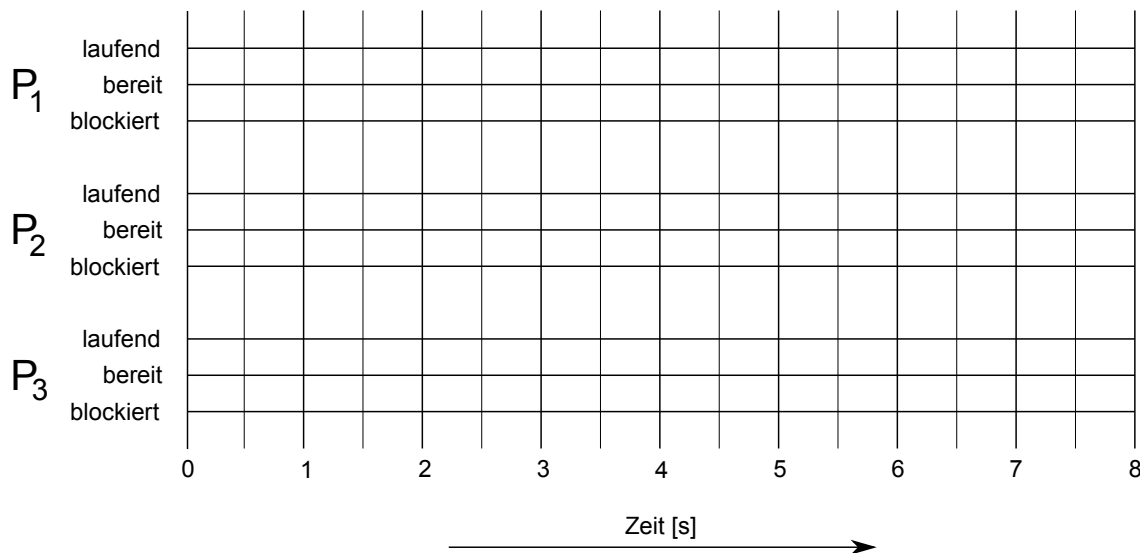


Gegeben sind drei Prozesse P_1 , P_2 und P_3 . Sie kommen zu unterschiedlichen Startpunkten ins System und haben unterschiedliches Laufverhalten (Rechenbedarf, Blockierungen):

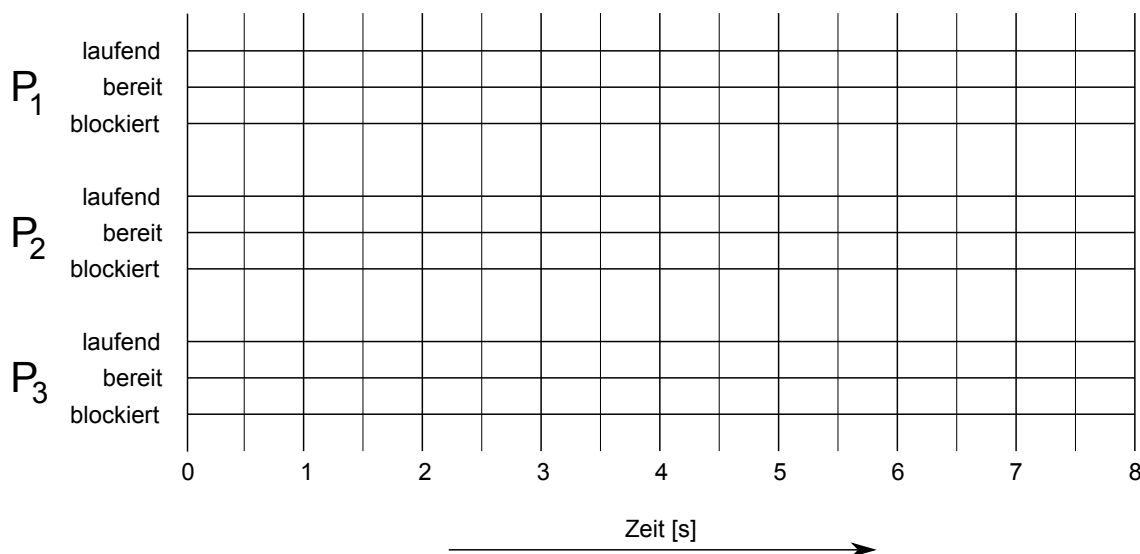
- P_1 : Start bei $t = 0s$, läuft $2,0s$, blockiert für $0,5s$, läuft noch einmal $1,0s$ und terminiert
- P_2 : Start bei $t = 0,5s$, läuft $1,0s$, blockiert für $0,5s$, läuft noch einmal für $1,0s$ und terminiert
- P_3 : Start bei $t = 1,0s$, läuft $1,5s$ ohne Blockierung und terminiert

Tragen Sie die Prozesszustände in folgende Zeitdiagramme ein. Markieren Sie mit einem Balken auf der jeweiligen Achse, so dass zu jedem Zeitpunkt (x-Achse) ersichtlich ist, in welchem Zustand sich jeder Prozess befindet.

- 1.) Tragen Sie die Prozesszustände für die **nicht-präemptive** Strategie Highest-Priority-First ein! P_1 hat die höchste, P_2 die nächst niedrige und P_3 die niedrigste Priorität. (5 P)



- 2.) Tragen Sie die Prozesszustände für die Round-Robin-Strategie mit einer Zeitscheibe von $1,5s$ ein! (5 P)



Aufgabe 4: Prozessverwaltung

(13 Punkte)

- 1.) Bei der Strategie Highest-Priority-First gibt es das Problem der Prioritätsinversion. Was versteht man darunter? (5 P)

- 2.) Wie kann man Prioritäteninversion verhindern? (2 P)

- 3.) Was versteht man unter nebenläufigen Prozessen? (3 P)

- 4.) Was versteht man unter Koordinierung? Geben Sie ein Beispiel für Koordinierung. (3 P)

Aufgabe 5: Dateisysteme

(11 Punkte)

- 1.) Welchen Zweck hat die FAT im gleichnamigen Dateisystem? Erläutern Sie im Detail was darin gespeichert wird. (5 P)

- 2.) In einem Unix-Dateisystem speichert der Inode so genannte Metadaten für unterschiedliche Elemente des Dateisystems ab. Nennen Sie wenigstens vier unterschiedliche Typen von Elementen und damit von Inodes. (4 P)

- 3.) Um welche Funktionalität erweitern ACLs die klassische Unix Rechteverwaltung? (2 P)

Aufgabe 6: Seitenadressierung

(15 Punkte)

Sie haben ein System mit reiner Seitenadressierung vor sich. Die Größe der Seitenkachel-tabelle (SKT) ist nicht beschränkt. Das System hat einen TLB. Ein Anwendungsprozess führt einen lesenden Speicherzugriff aus. Die SKT ist eingelagert, die Seite, von der gelesen wird, jedoch nicht.

Vervollständigen Sie die nun ablaufenden Schritte und geben Sie bei **allen** Schritten an, ob dieser in Hard- oder Software abläuft (nicht zutreffendes streichen, z.B: ~~HW~~/SW für Software). (3 P)

- 1.) HW/SW: Das Seitenkachel-tabellen-Basisregister wird zur logischen Seitennummer der logischen Adresse addiert und ergibt die Adresse des betreffenden SKT-Eintrags
- 2.) HW/SW: Der SKT-Eintrag wird gelesen. Welche Informationen enthält der Eintrag? (2 P)

- 3.) HW/SW: Parallel zu den beiden vorherigen Schritten läuft im TLB welcher Schritt ab? Was ist in diesem Fall das Ergebnis dieses Schritts? (2 P)

- 4.) HW/SW: Wie erkennt das System, dass die Seite ausgelagert ist? (1 P)

- 5.) HW/SW: Was passiert unmittelbar nach dieser Erkennung? (1 P)

- 6.) HW/SW: Der Prozess wird blockiert und die Seite auf eine frei Kachel eingelagert.

- 7.) HW/SW: Was muss dann gemacht werden, bevor Schritt 8 durchgeführt werden kann? (2 P)

- 8.) HW/SW: Der Prozess wird deblockiert und wiederholt damit die Leseoperation

- 9.) Die ersten drei Schritte dieser Aufgabe wiederholen sich.

- 10.) HW/SW: Nachdem die Seite diesmal eingelagert ist, passiert was nach Schritt 3 und vor Schritt 11? (1 P)

- 11.) HW/SW: Die Speicherzelle wird gelesen. Was passiert derweil im TLB? (3 P)

Aufgabe 7: Speicherbelegung

(4 Punkte)

Sie haben einen Speichers gegeben, der in gleich große Blöcke eingeteilt ist. Der Speicher wird nur blockweise vergeben. Bereits belegte Blöcke sind grau hinterlegt.

Die folgenden Speicherbereiche sollen nun **zusammenhängend** und in dieser Reihenfolge belegt werden:

- A: 3 Block
- B: 2 Blöcke
- C: 1 Block
- D: 4 Blöcke

- 1.) Tragen Sie in das Diagramm den jeweiligen Buchstaben der Belegung (also z.B. A) in den zugeteilten Block ein. Verwenden Sie zur Zuteilung den **First-fit**-Algorithmus, der hier von vorne beginnt. Geben Sie zusätzlich explizit an, falls Speicherbereiche (A-D) nicht im Speicher untergebracht werden können. Markieren Sie verbleibende freie Blöcke mit einem Kreuz.



(2P)

- 2.) Verwenden Sie alternativ den **Best-fit**-Algorithmus und tragen Sie die Belegung in das folgende Diagramm ein. Geben Sie zusätzlich explizit an, falls Speicherbereiche (A-D) nicht im Speicher untergebracht werden können. Markieren Sie verbleibende freie Blöcke mit einem Kreuz.



(2P)

Aufgabe 8: Ein Kessel Bunter

(12 Punkte)



- 1.) Bestimmen Sie die binäre Darstellung von -68 im Zweierkomplement. Verwenden Sie eine Breite von 8 Bit. (2P)

A rectangle is divided into two trapezoids by a diagonal line. The line starts at the top-left corner and extends to the bottom-right corner, creating two trapezoidal regions.

[illegible]

- 2.) Wieviel sind 8 MB in Bytes? (1 P)

A rectangle is divided into two trapezoids by a diagonal line. The line runs from the top-left corner to the bottom-right corner.

- 3.) Wieviel sind 2 KiB in Bytes? (1 P)

- 4.) Was versteht man unter Paravirtualisierung? (5 P)

- 5.) Wie werden Folgeaufträge in einem Treiber im Interrupt-Betrieb gestartet, wenn die zugehörigen Prozesse bereits blockiert sind? (3 P)



Zusatzblatt zu Aufgabe ____:

GdBS/TI1 2018

