

Lösungsskizzen zur Abschlussklausur

Systemsoftware (SYS)

Betriebssysteme-orientierter Teil

9. Februar 2009

Name: _____

Vorname: _____

Matrikelnummer: _____

Studiengang: _____

Hinweise:

- Tragen Sie zuerst auf allen Blättern (einschließlich des Deckblattes) Ihren *Namen*, Ihren *Vornamen* und Ihre *Matrikelnummer* ein. Lösungen ohne diese Angaben können nicht gewertet werden.
- Schreiben Sie die Lösungen jeder *Teilaufgabe* auf das jeweils vorbereitete Blatt. Sie können auch die leeren Blätter am Ende der Heftung nutzen. In diesem Fall ist ein Verweis notwendig. Eigenes Papier darf nicht verwendet werden.
- Legen Sie bitte Ihren *Lichtbildausweis* und Ihren *Studentenausweis* bereit.
- Als *Hilfsmittel* sind ein selbstständig, doppelseitig beschriebenes DIN-A4-Blatt und Taschenrechner zugelassen.
- Mit Bleistift oder Rotstift geschriebene Ergebnisse werden *nicht* gewertet.
- Die Bearbeitungszeit dieses Teils der Abschlussklausur beträgt *60 Minuten*.
- Stellen Sie sicher, dass Ihr Mobiltelefon ausgeschaltet ist. Klingelnde Mobiltelefone werden als Täuschungsversuch angesehen und der/die entsprechende Student/in wird von der weiteren Teilnahme an der Klausur **ausgeschlossen!**

Bewertung:

1a)	1b)	2a)	2b)	3)	4)	5a)	5b)	6a)	6b)	6c)
Σ	Note									

Lösungsskizzen zur Abschlussklausur
Systemsoftware (SYS)

9.2.2009
MSc Christian Baun

Aufgabe 1 (6+4 Punkte)

- a) Der Speicher eines Computersystems wird in die drei Kategorien **Primärspeicher**, **Sekundärspeicher** und **Tertiärspeicher** unterschieden. Beschreiben Sie die Merkmale dieser Speichersorten?
- b) Der **Tertiärspeicher** wird ebenfalls in zwei Kategorien unterschieden. Benennen Sie diese beiden Kategorien und beschreiben Sie diese.

Aufgabe 2 (4+6 Punkte)

- a) Was sind die Unterschiede zwischen **Prozessen** und **Threads**?
- b) Was sind die Unterschiede, Vor- und Nachteile zwischen **Kernel-Level-Threads** und **User-Level-Threads**?

Aufgabe 3 (3 Punkte)

Was sind **Race Conditions**, wie können Race Conditions entstehen und wie können Race Conditions verhindert werden?

Aufgabe 4 (4 Punkte)

Was ist der Unterschied zwischen Signalisierung und Sperren?

Aufgabe 5 (3+2 Punkte)

- a) Wie ist die Funktionsweise von **Journaling-Dateisystemen** und was sind die Vorteile von Journaling-Dateisystemen gegenüber Dateisystemen ohne Journal?
- b) Nennen Sie vier Beispiele für Journaling-Dateisysteme.

Aufgabe 6 (6+6+6 Punkte)

Auf einem Einprozessorrechner sollen sechs Prozesse verarbeitet werden.

Prozess	CPU-Laufzeit (ms)	Ankunftszeit
A	4	0
B	5	1
C	2	3
D	4	6
E	5	8
F	5	11

- a) Skizzieren Sie die Ausführungsreihenfolge der Prozesse mit einem Gantt-Diagramm (Zeitleiste) für **Round Robin** (Zeitquantum $q = 1$ ms), **First Come First Served** (FCFS), **Longest Job First** (LJF), **Longest Remaining Time First** (LRTF) und **Shortest Remaining Time First** (SRTF).

ACHTUNG!!! Für Round Robin ist bei allen Prozessen die Ankunftszeit 0. Diese Ausnahme gibt nur für Round Robin! Bei allen anderen Scheduling-Verfahren sind die in der Tabelle angegebenen Ankunftszeiten zu berücksichtigen.

- b) Berechnen Sie die mittleren Laufzeiten der Prozesse.
- c) Berechnen Sie die mittleren Wartezeiten der Prozesse.

Name:

Vorname:

Matr.Nr.:

Aufgabe 1)

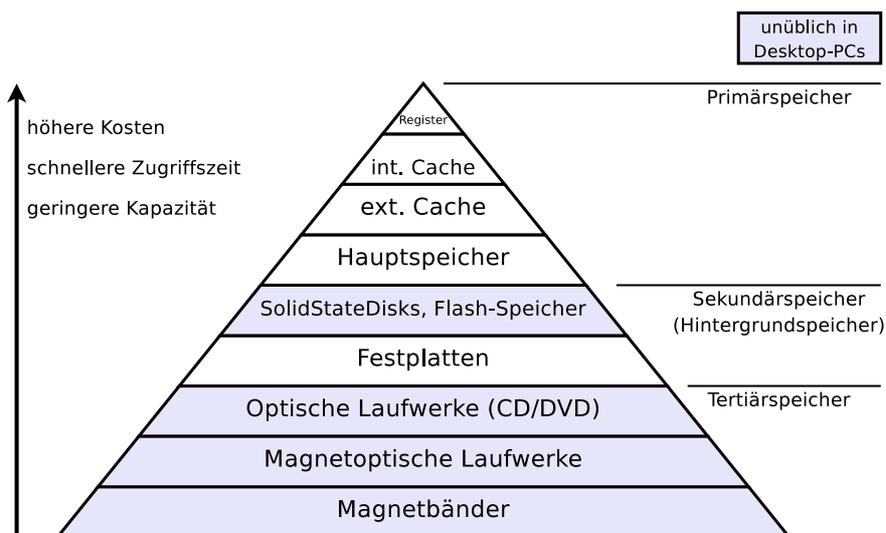
Punkte:

a) Der Speicher eines Computersystems wird in **Primärspeicher**, **Sekundärspeicher** und **Tertiärspeicher** unterschieden.

- Auf den **Primärspeicher** kann der Hauptprozessor direkt zugreifen.
- Der **Sekundärspeicher** wird auch als Hintergrundspeicher bezeichnet und wird über einen Controller angesprochen.
- Primärspeicher und Sekundärspeicher werden auch als **Onlinespeicher** bezeichnet, da sie eine feste Verbindung zum Computer und dadurch geringe Zugriffszeiten auf die Daten haben.
- Der **Tertiärspeicher** ist nicht dauerhaft verfügbar, oder nur über ein Laufwerk mit dem Computer verbunden. Die Hauptaufgabe des Tertiärspeichers ist die Archivierung.

b) Der Tertiärspeicher wird unterschieden in:

- **Nearlinespeicher** wird automatisch und ohne menschliches Zutun dem System bereitgestellt (z.B. Band-Library).
- Beim **Offlinespeicher** werden die Medien in Schränken oder Lagerräumen aufbewahrt und müssen von Hand in das System integriert werden. Streng genommen sind Wechselfestplatten (Sekundärspeicher) auch Offlinespeicher.



Name:

Vorname:

Matr.Nr.:

Aufgabe 2)

Punkte:

a)

- Ein **Thread** ist ein leichtgewichtiger Prozess und eine Aktivität (Programmausführung) innerhalb eines Prozesses.
- Es können mehrere nebenläufige Programmausführungen im gleichen Kontext aktiv sein und ihre Daten gemeinsam nutzen.
- Durch Threads kann ein Programm mehrfach an unterschiedlichen Stellen ausgeführt werden.
- Durch Threads kann ein Programm (Prozess/Task) mehrfach an unterschiedlichen Stellen ausgeführt werden.
- Der Prozessor kann zwischen Threads umschalten, ohne dass dazu aufwendige Kontextwechsel zwischen dem normalen **User Mode** (Benutzermodus) um dem privilegierten **Kernel Mode** (Kernel-Modus) notwendig sind.
- Alle Threads eines Programms arbeiten in dem gleichen Adressraum und besitzen die gleichen Betriebsmittel. Aus diesem Grund können sie direkt miteinander kommunizieren und zusammenarbeiten. Prozesse können dies nicht.
- Durch die Möglichkeit der direkten Kommunikation zwischen den Threads eines Programms sind die Daten, auf denen die Threads arbeiten weniger geschützt, als bei Prozessen mit nur einem Thread.

b)

- Bei Kernel-Level-Threads wird das Scheduling des Betriebssystems verwendet. Die Threads eines Prozesses können auf mehreren Prozessoren verteilt laufen. Bei User-Level-Threads geht das nicht.
- Ein Kernel-Level-Thread der blockiert, blockiert nur sich selbst. Ein User-Level-Thread der blockiert, blockiert den gesamten Prozess.
- Bei Kernel-Level-Threads ist jede Threadoperation ein Systemaufruf. User-Level-Threads sind effizienter.
- Mit User-Level-Threads können Threads auch auf Betriebssystemen ohne Thread-Unterstützung verwendet werden.
- Es kann immer nur ein User-Level-Thread eines Prozesses rechnen, da der Kernel nur den Prozess, aber nicht die User-Level-Threads kennt.

Name:

Vorname:

Matr.Nr.:

Aufgabe 3)

Punkte:

- Eine **Race Condition** (*Wettlaufsituation*) bezeichnet eine Konstellation, bei der das Ergebnis eines Prozesses von der Reihenfolge oder dem zeitlichen Ablauf anderer Ereignisse abhängt.
- Eine Race Conditions ist eine **unbeabsichtigten Wettlaufsituation** zweier Prozesse, die auf die gleiche Speicherstelle schreibend zugreifen wollen.
- Häufiger Grund für schwer auffindbare Programmfehler.
- Problem: Das Auftreten und die Symptome sind von unterschiedlichen Ereignissen und ihrem Verhalten abhängen. Bei jedem Testdurchlauf können die Symptome komplett verschieden sein oder verschwinden.
- Um eine Race Condition zu verhindern, muss bei einem Zugriff auf den Wert der Speicherstelle, diese bis zum Abschluss des Zugriffs gesperrt werden.
- Race Conditions können u.a durch das Konzept der Semaphore vermieden werden.

Name:

Vorname:

Matr.Nr.:

Aufgabe 4)

Punkte:

- Bei der **Signalisierung** wird eine Reihenfolgebeziehung zwischen Prozessen festgelegt.
 - Beispiel: Abschnitt A von Prozess P_1 soll vor Abschnitt B von P_2 ausgeführt werden.
- Bei einer **Sperre** werden kritische Abschnitte gesichert.
 - Die Reihenfolge, in der die Prozesse bei einer Sperre ihre kritische Abschnitte abarbeiten, ist nicht festgelegt!
 - Es soll bei einer Sperre nur sichergestellt werden, dass es keine Überlappung in der Ausführung der kritischen Abschnitte gibt.

Name:

Vorname:

Matr.Nr.:

Aufgabe 5)

Punkte:

a)

- Ein Journaling-Dateisystem führt ein sogenanntes Journal über die Daten, auf die Schreibzugriffe durchgeführt werden sollen.
- Eine zu ändernde Datei behält ihre Gültigkeit, bis die Schreibzugriffe durchgeführt wurden.
- In festen Zeitabständen wird das Journal geschlossen und die Schreiboperationen werden durchgeführt.
- Während der Abarbeitung des Journals wird festgehalten, welche Schreiboperationen bereits erfolgreich durchgeführt wurden.
- Gleichzeitig werden Prüfsummen von Daten von der Änderung erstellt.
- Mit Hilfe des Journals und den Prüfsummen können nach einem Systemabsturz die zu überprüfenden bzw. wiederherzustellenden Dateien schnell identifiziert und repariert werden und das Journal wenn möglich weiter abgearbeitet werden.
- Im schlimmsten Fall gehen Änderungsanforderungen, die im Journal vermerkt waren, verloren. Die Dateien auf dem Medium bleiben aber in einem konsistenten Zustand.
- Das Führen eines Journals führt zu geringen Leistungseinbußen.
- Der Vorteil von Journaling-Dateisystemen ist, dass bei einem Absturz des Betriebssystems nicht alle Daten überprüft, sondern nur die zu dem Zeitpunkt geöffneten Daten repariert werden müssen. Das führt zu großen Zeitersparnissen.

b) ReiserFS, ext3, ext4, JFS, XFS, NTFS, HFSJ, BeFS, ...

Name:

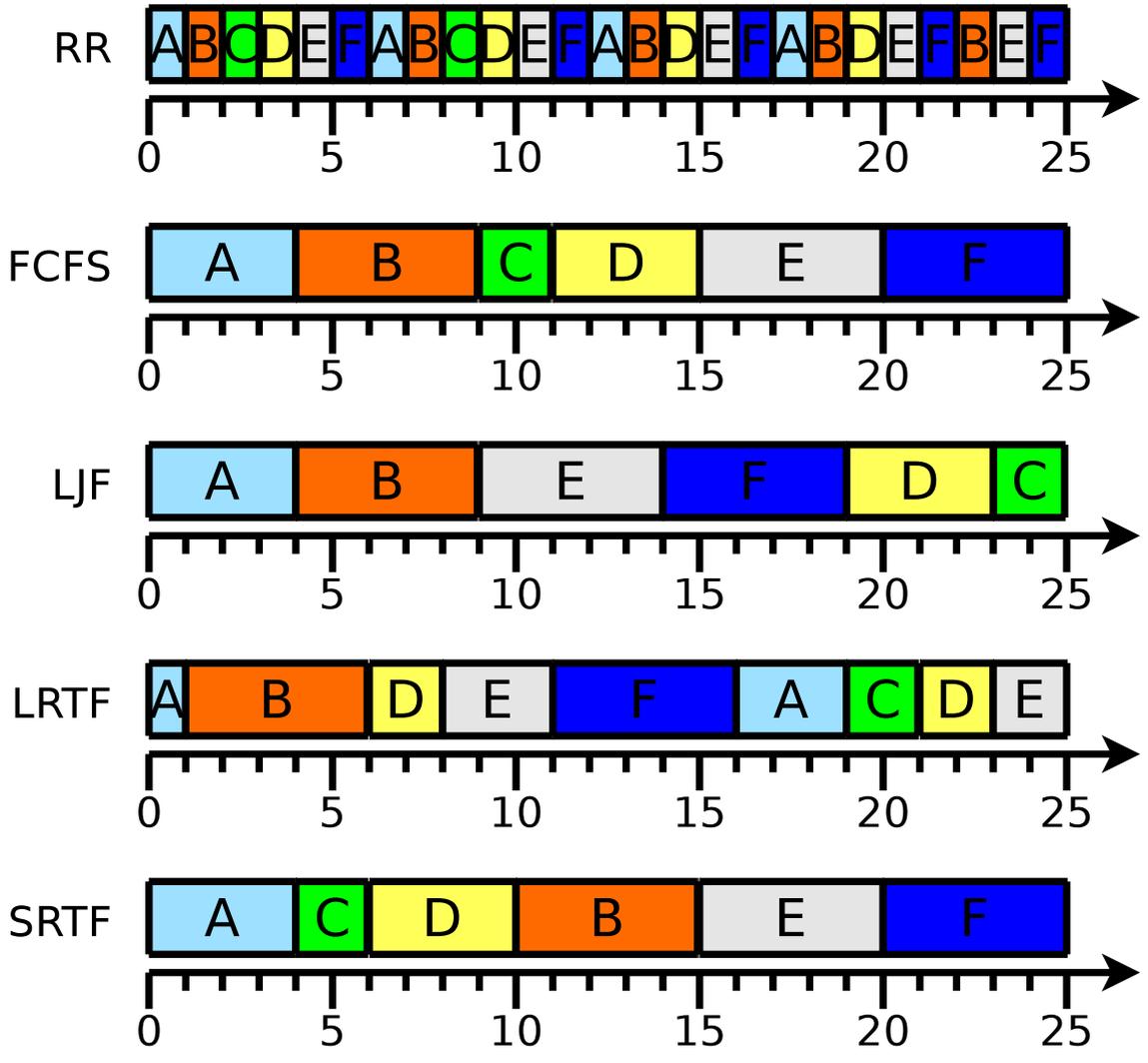
Vorname:

Matr.Nr.:

Aufgabe 6)

Punkte:

a)



Name:

Vorname:

Matr.Nr.:

Aufgabe 6)

Punkte:

b) Laufzeit (Turnaround Time) der Prozesse

	A	B	C	D	E	F
Round Robin	18	23	9	20	24	25
First Come First Served	4	8	8	9	12	14
Longest Job First	4	8	22	17	6	8
Longest Remaining Time First	19	5	18	17	17	5
Shortest Remaining Time First	4	14	3	4	12	14

$$\text{Round Robin} \quad \frac{18+23+9+20+24+25}{6} = 19,8\bar{3} \text{ ms}$$

$$\text{First Come First Served} \quad \frac{4+8+8+9+12+14}{6} = 9,1\bar{6} \text{ ms}$$

$$\text{Longest Job First} \quad \frac{4+8+22+17+6+8}{6} = 10,8\bar{3} \text{ ms}$$

$$\text{Longest Remaining Time First} \quad \frac{19+5+18+17+17+5}{6} = 13,5 \text{ ms}$$

$$\text{Shortest Remaining Time First} \quad \frac{4+14+3+4+12+14}{6} = 8,5 \text{ ms}$$

c) Wartezeit der Prozesse – Zeit in der **bereit**-Liste

	A	B	C	D	E	F
Round Robin	14	18	7	16	19	20
First Come First Served	0	3	6	5	7	9
Longest Job First	0	3	20	13	1	3
Longest Remaining Time First	15	0	16	13	12	0
Shortest Remaining Time First	0	9	1	0	7	9

$$\text{Round Robin} \quad \frac{14+18+7+16+19+20}{6} = 15,6\bar{6} \text{ ms}$$

$$\text{First Come First Served} \quad \frac{0+3+6+5+7+9}{6} = 5 \text{ ms}$$

$$\text{Longest Job First} \quad \frac{0+3+20+13+1+3}{6} = 6,6\bar{6} \text{ ms}$$

$$\text{Longest Remaining Time First} \quad \frac{15+0+16+13+12+0}{6} = 9,3\bar{3} \text{ ms}$$

$$\text{Shortest Remaining Time First} \quad \frac{0+9+1+0+7+9}{6} = 4,3\bar{3} \text{ ms}$$