

Lösungsskizzen zur Abschlussklausur

Systemsoftware (SYS)

2.7.2009
MSc Christian Baum

Aufgabe 1 (4+2 Punkte)

Nennen Sie die beiden grundsätzliche Kategorien von **Echtzeitbetriebssystemen** und beschreiben Sie die Unterschiede. Nennen Sie auch vier typische Einsatzgebiete von Echtzeitbetriebssystemen und ordnen Sie jedes Einsatzgebiet einer der beiden Kategorien zu.

Aufgabe 2 (2+2+2 Punkte)

- Es existieren zwei grundsätzliche Konzepte, um Schreibzugriffe auf den Cache durchzuführen. Welche beiden Konzepte sind das?
- Beschreiben die beiden Konzepte. Die Unterschiede sollen dabei erkennbar sein.
- Was sind die Vor- und Nachteile der beiden Konzepte?

Aufgabe 3 (2 Punkte)

Was ist die Kernaussage der **Anomalie von Laszlo Belady**?

Aufgabe 4 (2+2+2 Punkte)

- Welche zwei **Gruppen von Ein- und Ausgabegeräten** gibt es bezüglich der kleinsten Übertragungseinheit.
- Was charakterisiert jede der beiden Gruppen?
- Nennen Sie für jede Gruppe zwei Geräte-Beispiele.

Aufgabe 5 (6+2 Punkte)

- Nennen Sie die drei Möglichkeiten, die es gibt, damit eine Anwendung Daten von Ein- und Ausgabegeräten lesen kann. Was sind die Unterschiede, Vor- und Nachteile?
- Was halten Sie davon, dass Programme direkt auf Speicherstellen zugreifen? Ist das eine gute Idee? Begründen Sie ihre Antwort.

Aufgabe 6 (1+2+1 Punkte)

- Was ist ein **Scheduler** und was sind seine Aufgaben?
- Die existierenden **Schedulingverfahren** können in zwei grundsätzliche Klassen unterteilt werden. Welche sind das und in was unterscheidet diese?
- Was ist der Grund für die Existenz des sogenannten **Leerlaufprozesses** und wie funktioniert er?

Aufgabe 7 (6+6+6 Punkte)

Auf einem Einprozessorrechner sollen sechs Prozesse verarbeitet werden.

Prozess	CPU-Laufzeit (ms)	Ankunftszeit
A	3	0
B	2	3
C	5	4
D	3	5
E	2	9
F	5	10

- 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:

- **Harte Echtzeitsysteme**

- Zeitvorgaben müssen unbedingt eingehalten werden.
- Verzögerungen können unter keinen Umständen akzeptiert werden.
- Verzögerungen führen zu katastrophalen Folgen und hohen Kosten.
- Ergebnisse sind nutzlos wenn sie zu spät erfolgten.
- Einsatzbeispiele: Reaktorsteuerung, ABS, Flugzeugsteuerung.

- **Weiche Echtzeitsysteme**

- Gewisse Toleranzen sind erlaubt.
- Verzögerungen führen zu geringen, akzeptablen Kosten.
- Einsatzbeispiele: Telefonanlage, Parkschein- oder Fahrkartenautomat, Multimedia-Anwendungen wie Audio/Video on Demand.

Name:

Vorname:

Matr.Nr.:

Aufgabe 2)

Punkte:

a) **Write-Back** und **Write-Through**

b)

- **Write-Back:** Schreibzugriffe werden nicht direkt an die nächst tiefere Speicherebene weitergegeben. Inkonsistenzen zwischen den Daten im Cache und auf dem zu cachenden Speicher entstehen. Die Daten werden erst zurückgeschrieben, wenn der betreffende Datenblock aus dem Cache verdrängt wird.
- **Write-Through:** Schreibzugriffe werden sofort an die nächst tiefere Speicherebene weitergegeben.

c)

- **Write-Back:**
Vorteil: Höhere System-Geschwindigkeit.
Nachteil: Daten gehen beim Systemausfall verloren.
- **Write-Through:**
Vorteil: Datenkonsistenz ist gesichert.
Nachteil: Geringere System-Geschwindigkeit.

Name:

Vorname:

Matr.Nr.:

Aufgabe 3)

Punkte:

- Laszlo Belady zeigte in seiner Anomalie, dass unter sehr ungünstigen Umständen FIFO bei einem größeren Speicher sogar zu mehr Zugriffsfehlern (Miss) führt, als bei einem kleinen Speicher.
- Ursprünglich ging man davon aus, dass eine Vergrößerung des Speichers immer zu weniger oder schlechtestenfalls gleich vielen Zugriffsfehlern führt.
- Bis zur Entdeckung von Belady's Anomalie hielt man FIFO für eine gute Ersetzungsstrategie.

Name:

Vorname:

Matr.Nr.:

Aufgabe 4)

Punkte:

a) **Zeichenorientierte Geräte und Blockorientierte Geräte**

b)

- **Zeichenorientierte Geräte:** Bei der Ankunft/Anforderung jedes einzelnes Zeichens wird immer mit dem Prozessor kommuniziert.
- **Blockorientierte Geräte:** Die Datenübertragung wird erst bei Vorliegen eines kompletten Blocks (z.B. 1-4 kB) angestoßen.

c)

- **Zeichenorientierte Geräte:**
Maus, Tastatur, Drucker, Terminals, Magnetbänder
- **Blockorientierte Geräte:**
Festplatten, CD-/DVD-Laufwerke, Disketten-Laufwerke

Name:

Vorname:

Matr.Nr.:

Aufgabe 5)

Punkte:

a)

- **Busy Waiting:** Der Prozess sendet die Anfrage an das Gerät und wartet in einer Endlosschleife, bis die Daten bereit stehen. Leicht zu implementieren. Belastet den Prozessor und behindert die gleichzeitige Abarbeitung mehrerer Programme.
- **Interrupt-gesteuert:** Der Prozess initialisiert die Aufgabe und wartet auf einen Interrupt (Unterbrechung) durch den notwendigen Interrupt-Controller. Die CPU ist während des Wartens nicht blockiert. Zusätzliche Hardware notwendig.
- **Direct Memory Access:** Ein zusätzlicher DMA-Baustein überträgt die Daten direkt zwischen Speicher und Controller ohne Mithilfe der CPU. Vollständige Entlastung der CPU. Hoher Hardware-Aufwand.

b)

- Würden Prozesse direkt auf die Speicherstellen zugreifen, würde dies in einem Multitasking-System zu großen Problemen wegen der fehlenden Datensicherheit führen.
- Besser als der direkte Zugriff auf die Speicherstellen ist eine Abstraktion der Prozesse von den verwendeten Speichertechnologien und ihren gegebenen Ausbaumöglichkeiten.

Name:

Vorname:

Matr.Nr.:

Aufgabe 6)

Punkte:

a)

- Der Scheduler eines Betriebssystems legt Ausführungsreihenfolge der Prozesse im Zustand **bereit** fest.
- Die Entscheidung, welcher Prozess wann an der Reihe ist, ist vom Scheduling-Algorithmus abhängig.

b)

- **Nicht-präemptives Scheduling** (nicht-verdrängendes Scheduling): Ein Prozess, der vom Scheduler die CPU zugewiesen bekommen hat, behält die Kontrolle über diese bis zu seiner vollständigen Fertigstellung. Eine vorzeitige Entziehung der CPU durch den Scheduler ist nicht vorgesehen. Problematisch ist dabei, dass ein Prozess die CPU so lange belegen kann, wie er möchte und andere, vielleicht dringendere Prozesse für lange Zeit nicht zum Zuge kommen.
- **Präemptives Scheduling** (verdrängendes Scheduling): Einem Prozess kann die CPU vor seiner Fertigstellung wieder entzogen werden, um diese anderen Prozessen zuzuteilen. Der Prozess pausiert so lange in seinem aktuellen Zustand, bis ihm wieder die CPU vom Scheduler zugeteilt wird.

c)

- Bei modernen Betriebssystemen erhält die CPU zu jedem Zeitpunkt einen Prozess.
- Wenn kein Prozess im Zustand **bereit** und damit ausführungsbereit ist, kommt der **Leerlaufprozess** (*Idle Task*) zum Zug.
- Der Leerlaufprozess ist immer aktiv und hat die niedrigste Priorität.
- Durch den Leerlaufprozess muss der Scheduler nie den Fall berücksichtigen, dass kein aktiver Prozess existiert.
- Auf modernen Prozessor-Architekturen versetzt der Leerlaufprozess die CPU in einen stromsparenden Modus.

Name:

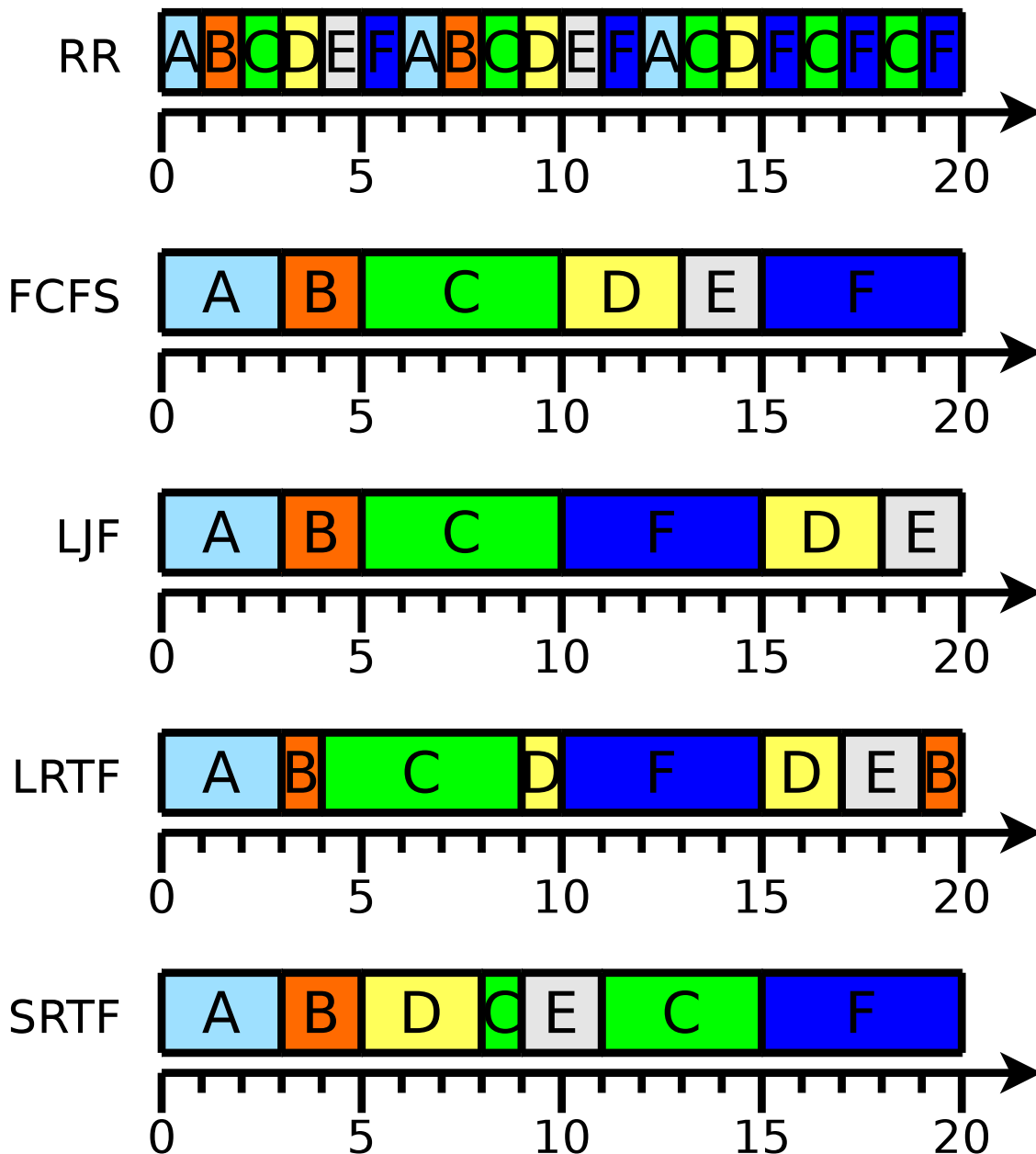
Vorname:

Matr.Nr.:

Aufgabe 7)

Punkte:

a)



Name:

Vorname:

Matr.Nr.:

Aufgabe 7)

Punkte:

b) Laufzeit (Turnaround Time) der Prozesse

	A	B	C	D	E	F
Round Robin	13	8	19	15	11	20
First Come First Served	3	2	6	8	6	10
Longest Job First	3	2	6	13	11	5
Longest Remaining Time First	3	17	5	12	10	5
Shortest Remaining Time First	3	2	11	3	2	10

$$\begin{aligned}
 \text{Round Robin} & \quad \frac{13+8+19+15+11+20}{6} = 14, \overline{3} \text{ ms} \\
 \text{First Come First Served} & \quad \frac{3+2+6+8+6+10}{6} = 5, \overline{8\overline{3}} \text{ ms} \\
 \text{Longest Job First} & \quad \frac{3+2+6+13+11+5}{6} = 6, \overline{6} \text{ ms} \\
 \text{Longest Remaining Time First} & \quad \frac{3+17+5+12+10+5}{6} = 8, \overline{6} \text{ ms} \\
 \text{Shortest Remaining Time First} & \quad \frac{3+2+11+3+2+10}{6} = 5, \overline{1\overline{6}} \text{ ms}
 \end{aligned}$$

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

	A	B	C	D	E	F
Round Robin	10	6	14	12	9	15
First Come First Served	0	0	1	5	4	5
Longest Job First	0	0	1	10	9	0
Longest Remaining Time First	0	15	0	9	8	0
Shortest Remaining Time First	0	0	6	0	0	5

$$\begin{aligned}
 \text{Round Robin} & \quad \frac{10+6+14+12+9+15}{6} = 11 \text{ ms} \\
 \text{First Come First Served} & \quad \frac{0+0+1+5+4+5}{6} = 2, \overline{5} \text{ ms} \\
 \text{Longest Job First} & \quad \frac{0+0+1+10+9+0}{6} = 3, \overline{3} \text{ ms} \\
 \text{Longest Remaining Time First} & \quad \frac{0+15+0+9+8+0}{6} = 5, \overline{3} \text{ ms} \\
 \text{Shortest Remaining Time First} & \quad \frac{0+0+6+0+0+5}{6} = 1, \overline{8\overline{3}} \text{ ms}
 \end{aligned}$$