

Name:

Korrektorexemplar
Matrikelnummer:

Aufgabe 1: Rechnerarchitektur und Betriebssystemgrundlagen

(4 Punkte)

(a) (2 Punkte)

Erläutern Sie den Aufbau der Von-Neumann-Architektur in Textform oder mit Hilfe einer Zeichnung.

todo: Bild einfügen.

(b) (2 Punkte)

Erklären Sie die beiden Abarbeitungsarten von Unterbrechungen in Textform oder mit Hilfe einer Zeichnung: sequentiell und geschachtelt

Sequentiell: Verbieten weiterer Unterbrechungen während der Unterbrechungs- behandlung (Unterbrechungssperre setzen, disable interrupt).

Geschachtelt: Klassifikation von Unterbrechungen in Prioritätsklassen (statisch), Unterbrechungen höherer Priorität dürfen die Bearbeitung von Unterbrechungen geringerer Priorität unterbrechen.

Punkte	
--------	--

Name:

Korrektorexemplar
Matrikelnummer:**Aufgabe 2: Parallelität****(18 Punkte)****(a) (8 Punkte)**

Gegeben sei folgender Pseudocode unter Verwendung von `parbegin/parend` (Listing 1).
Zeichnen Sie einen Abhängigkeitsgraphen, der diese Implementierung widerspiegelt.

```
A;  
parbegin  
    D;  
    begin  
        B;  
        R;  
        parbegin  
            E;  
            Q;  
            Y;  
        parend  
    end  
    F;  
parend  
parbegin  
    H;  
    G;  
parend  
X;
```

Listing 1: Pseudocode mit `parbegin/parend`

Entsprechender Abhängigkeitsgraph:

Punkte	
--------	--

Name:

Korrektorexemplar
Matrikelnummer:

(b) (2 Punkte)

Erläutern Sie die Bedeutung des Befehls `pardo`.

F 2-43

(c) (4 Punkte)

Nennen Sie vier Abstraktionsebenen, auf denen Parallelität bzw. Nebenläufigkeit realisiert werden können. Ordnen Sie die Ebenen ihrer Granularität entsprechend von grob- bis feingranular.

F2-44

(d) (4 Punkte)

Beschreiben Sie die Bedeutung der folgenden Prozessbezeichnungen:

Eltern-Prozess:

Kind-Prozess:

Zombie:

Daemon:

Daemon: F2-48

Punkte	
--------	--

Name:

Matrikelnummer:

Aufgabe 3: Scheduling

(10 Punkte)

Konstruieren Sie basierend auf den Shortest Remaining Time Next- und Highest Response Ratio Next-Verfahren für die Prozesse aus Tabelle 1 jeweils einen gültigen Schedule (soweit wie möglich) in den Tabellen 2 und 5. Tragen Sie anschließend jeweils Warte- und Antwortzeiten der beiden Verfahren in die Tabellen 3 und 6 ein.

Prozesse	Laufzeit (Dauer)	Ankunftszeit
P1	3	1
P2	2	3
P3	9	4
P4	7	7
P5	6	12

Tabelle 1: Aperiodische Prozesse

Zeit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CPU															
Zeit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
CPU															

Tabelle 2: Schedule nach dem Shortest Remaining Time Next-Verfahren

	P1	P2	P3	P4	P5
Wartezeit					
Antwortzeit					

Tabelle 3: Warte- und Antwortzeiten für das Shortest Remaining Time Next-Verfahren

Punkte	
--------	--

Name:

Korrektorexemplar

Matrikelnummer:

Prozesse	Laufzeit (Dauer)	Ankunftszeit
P1	3	1
P2	2	3
P3	9	4
P4	7	7
P5	6	12

Tabelle 1: Aperiodische Prozesse (Kopie von Seite 5)

Zeit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CPU															

Zeit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
CPU															

Tabelle 5: Schedule nach dem Highest Response Ratio Next-Verfahren

	P1	P2	P3	P4	P5
Wartezeit					
Antwortzeit					

Tabelle 6: Warte- und Antwortzeiten für das Highest Response Ratio Next-Verfahren

SJN:

Punkte	
--------	--

Name:

Korrektorexemplar

Matrikelnummer:

Zeit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CPU	1	1	1	2	2	3	3	-	-	-	-	-	-	3	5
			2	3	3		4	4	4	4	4	5	5	5	4
												4	4	4	

Zeit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
CPU	5	5	5	5	5	4									
	4	4	4	4	4										

	P1	P2	P3	P4	P5
Wartezeit	0	1	2	14	3
Antwortzeit	3	3	11	21	9

Nachfolgend MLFB:

Zeit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CPU	1	1	1	2	3	2	4	3	3	4	4	5	5	5	3
t=1			2	3											
t=2					2	3	3	4	4						
t=4										3	3	3,4	3,4	3,4	4,5
FIFO															

Zeit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
CPU	3	3	3	4	4	4	4	5	5	5	3	3			
t=1															
t=2															
t=4	4,5	-	4,5	5	5	5	5								
FIFO				3	-	-	3	3	3	3					

	P1	P2	P3	P4	P5
Wartezeit	0	2	15	9	7
Antwortzeit	3	4	24	16	13

Bewertung: Linear mit korrektem Anteil

Punkte	
--------	--

Name:

Korrektorexemplar
Matrikelnummer:

SJN: 3 P. Verfahren, 1 P. Warte- und Antwortzeiten

HRRN: 5 P. Verfahren, 1 P. Warte- und Antwortzeiten

Punkte	
--------	--

Name:

Korrektorexemplar
Matrikelnummer:**Aufgabe 4: Koordination****(15 Punkte)**

- (a) **(3 Punkte)** Die Prozesse P1 und P2 werden parallel ausgeführt. Jede Zeile ist dabei eine **atomare** Anweisung. Welche Werte können die Variablen x und y nach der Ausführung annehmen?

1 Ausgangszustand:

2 $x=1; y=2;$

3

4 P1:

5 $x = x + x + 4;$

6 $y = x + y;$

7

8 P2:

9 $y = 9;$

10 $x = x + 4;$

todo

- (b) **(2 Punkte)** Was versteht man unter *Kooperation* im Zusammenhang mit Koordination?
Kooperation: Prozesse arbeiten an einer gemeinsamen Aufgabe und müssen interagieren, z.B. sich synchronisieren.

- (c) **(2 Punkte)** Was versteht man unter *Konkurrenz* im Zusammenhang mit Koordination?
Konkurrenz: Prozesse stehen im Wettbewerb um gemeinsame Betriebsmittel, die exklusiv belegt werden müssen.

- (d) **(8 Punkte)**

In einem Prüfungsamt einer Universität gibt es vier Schalter. Aus diesem Grund dürfen sich im Prüfungsamt maximal vier Studenten gleichzeitig aufhalten.

Vervollständigen Sie unter Verwendung der *in der Vorlesung* vorgestellten Konstrukte `signal`, `wait`, `lock` und `unlock` den Pseudocode des Prozesses Student um sicher-

Punkte	
--------	--

Name:

Korrektorexemplar

Matrikelnummer:

zustellen, dass die maximale Anzahl von Studenten im Prüfungsamt nicht überschritten wird.

```
// definieren Sie hier ggf. eigene globale Variablen
// (gelten fuer alle Prozesse vom Typ Student)

    int student;

// definieren Sie hier ggf. Initialanweisungen
// (Funktion wird einmalig ausgefuehrt)
init(){

    student = 0;

}

// Prozess Student
// (wird beim "start" eines "neuen" Studenten ausgefuehrt)
proc_student(){

    lock( L );
    while ( student >= 4 ){
        unlock( L );
        wait( C );
        lock( L );
    }
    student++;
    unlock( L );

    //Student betritt Pruefungsamt
    enterPA();
    //Student verlaesst Pruefungsamt

    lock( L );
    student--;
    unlock( L );

    signal( C );

}
```

Punkte	
--------	--

Name:

Korrektorexemplar
Matrikelnummer:

Aufgabe 5: Speicherverwaltung

(19 Punkte)

(a) (3 Punkte)

Abbildung 1 zeigt die Speicherbelegung für drei aufeinanderfolgende Anforderungen.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
				4					6								2					8						
				4					6								2					8						
				4					3					3					2					8				
				4					3					3					2					8				

} t = 0

} t = 1

} t = 2

} t = 3

Abbildung 1: Entwicklung der Speicherbelegung. Belegter Speicher ist grau hinterlegt.

Nennen Sie *alle* Ihnen aus der Vorlesung bekannten Auswahlstrategien, die zu der in Abbildung 1 dargestellten Entwicklung der Speicherbelegung führen.

Lösung

[Korrektur]

- First Fit
- Next Fit
- Worst Fit
- Buddy Verfahren

Von den angegebenen Strategien passt nur Worst Fit. Es kann nicht First Fit sein, da $t = 2$ sonst ebenfalls in den zweiten freien Block gelegt worden wäre. Es kann auch nicht Next Fit sein, da an sonst $t = 3$ in den ersten freien Block gekommen wäre. Für das Buddyverfahren müssten die Speicherstücke in 2er Potenzen und aligned verteilt werden.

(b) (6 Punkte)

Wie sind die Metriken *interner Verschnitt*, *externer Verschnitt* und *Auslastung* definiert?

Interner Verschnitt:

Externer Verschnitt:

Auslastung:

Punkte	
--------	--

Korrektorexemplar

Name:

Matrikelnummer:

(c) (4 Punkte)

Es sei folgende Sequenz von Speicheranforderungen gegeben:

```
t = 1:  alloc(3);
t = 2:  alloc(4);
t = 3:  alloc(7);
t = 4:  alloc(3);
```

Wenden Sie die Auswahlstrategien **Next Fit** (Abb. 2) und **Worst Fit** (Abb. 3) bei dieser Sequenz in den nachfolgenden Abbildungen soweit wie möglich an. Tragen Sie dabei auch jeweils die Größe des noch verbleibenden freien Speichers ein. Falls eine Allokation nicht bearbeitet werden kann, notieren Sie dieses. Belegter Speicher ist in allen Vorlagen bereits grau hinterlegt.

[Korrektur]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
3					4				3					11							}	t = 0						
3					4				3					11							}	t = 1						
2	1				4				3					11							}	t = 2						
2	1				4				3					9					2		}	t = 3						
2	1				4				3					9					2		}	t = 4						

Abbildung 2: Best Fit: Belegter Speicher ist grau hinterlegt.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
3					4				3					11							}	t = 0							
3					4				3					4		7							}	t = 1					
3					4				3					4		2	5							}	t = 2				
alloc(9) nicht möglich																												}	t = 3
3					4				3					4		2	2			3				}	t = 4				

Abbildung 3: Worst Fit: Belegter Speicher ist grau hinterlegt.

(d) (6 Punkte) Gegeben sei ein physikalischer Speicher mit vier Kacheln. Auf folgende Seiten wird nacheinander zugegriffen:

$$R = 1, 2, 3, 4, 2, 5, 3, 2, 2, 0, 1, 3, 1, 1, 0$$

Wenden Sie das Seitentauschverfahren **Least-Recently-Used (LRU)** unter Angabe der Seitenzugriffe jeder Seite zu jedem Zeitpunkt an. Machen Sie dabei sowohl erfolgreiche

Punkte	
--------	--

Korrektorexemplar

Name:

Matrikelnummer:

Zugriffe auf belegte Kacheln als auch Seitenzugriffsfehler kenntlich.

Tragen Sie die Lösung in folgender Tabelle ein:

Referenz	1	2	3	4	2	5	3	2	2	0	1	3	1	1	0
Kachel 1	1zf														
Kachel 2	-														
Kachel 3	-														
Kachel 4	-														
Stapel top	1														
1	-														
2	-														
3	-														

Bitte verwenden Sie die folgenden Kennzeichnungen innerhalb des Kachel-Abschnitts:

- 1z: Kachelzugriff auf Seite 1
- 1zf: Zugriff nach Seitenzugriffsfehler auf Seite 1

Referenz	1	2	3	4	2	2	0	1	3	1	5	3	2	1	0
Kachel 1	1zf						0zf				5zf				0zf
Kachel 2	-	2zf			2z	2z							2z		
Kachel 3	-		3zf					1zf		1z					1z
Kachel 4	-			4zf					3zf			3z			
Zähler 0	-						1								2
Zähler 1	1							2		3					4
Zähler 2	-	1			2	3							4		
Zähler 3	-		1						2			3			
Zähler 4	-			1											
Zähler 5	-										1				

Bewertung: Linear mit korrektem Anteil

Punkte	
--------	--

Name:

Korrektorexemplar
Matrikelnummer:

Aufgabe 6: Sicherheit

(4 Punkte)

(a) **(2 Punkte)**

Erklären Sie die Gemeinsamkeiten und Unterschiede von symmetrischer Kryptographie und Public-Key-Kryptographie.

(b) **(2 Punkte)**

Erläutern Sie, weshalb bzw. auf welche Art und Weise die Verwendung der Datei `/etc/shadow` zu einem Sicherheitszugewinn führt.

Punkte	
--------	--