



Hinweise zur Personalisierung:

- Ihre Prüfung wird bei der Anwesenheitskontrolle durch Aufkleben eines Codes personalisiert.
- Dieser enthält lediglich eine fortlaufende Nummer, welche auch auf der Anwesenheitsliste neben dem Unterschriftenfeld vermerkt ist.
- Diese wird als Pseudonym verwendet, um eine eindeutige Zuordnung Ihrer Prüfung zu ermöglichen.

Grundlagen Betriebssysteme und Systemsoftware

Klausur: IN0009 / Midterm

Datum: Mittwoch, 11. Dezember 2019

Prüfer: Prof. Dr.-Ing. Jörg Ott

Uhrzeit: 18:30 – 19:15

Bearbeitungshinweise

- Diese Klausur umfasst **8 Seiten** mit insgesamt **4 Aufgaben**.
Bitte kontrollieren Sie jetzt, dass Sie eine vollständige Angabe erhalten haben.
- Die Gesamtpunktzahl in dieser Prüfung beträgt 45 Punkte.
- Das Heraustrennen von Seiten aus der Prüfung ist untersagt.
- Als Hilfsmittel sind zugelassen:
 - ein **analoges Wörterbuch** Deutsch ↔ Muttersprache **ohne Anmerkungen**
- Mit * gekennzeichnete Teilaufgaben sind ohne Kenntnis der Ergebnisse vorheriger Teilaufgaben lösbar.
- **Es werden nur solche Ergebnisse gewertet, bei denen der Lösungsweg erkennbar ist.** Auch Textaufgaben sind **grundsätzlich zu begründen**, sofern es in der jeweiligen Teilaufgabe nicht ausdrücklich anders vermerkt ist.
- Bei Multiple Choice Fragen müssen alle Antworten korrekt angekreuzt sein. Es ist mindestens eine Antwort korrekt.
- Schreiben Sie weder mit roter/grüner Farbe noch mit Bleistift.
- Schalten Sie alle mitgeführten elektronischen Geräte vollständig aus, verstauen Sie diese in Ihrer Tasche und verschließen Sie diese.

Hörsaal verlassen von _____ bis _____ / Vorzeitige Abgabe um _____

Aufgabe 1 Generische Fragen (12 Punkte)

0 a)* Übersetzen Sie die folgenden physischen Adressen (p_x) zu logischen Adressen (v_x), mittels Basisadressierung.

1

Umrechnungsvorschrift: $p_x := f(b_x, v_x) = b_x + v_x$

$b_0 = 0x1000$

$b_1 = 0x2000$

$b_2 = 0x3000$

$b_3 = 0x4000$

$b_4 = 0x5000$

Physische Adressen (p_x):

0x000000001FF0, b_0 :

0x123456789ABCDEF, b_4 :

0x09875A67B, b_3 :

0 b)* Finden und erläutern Sie die Schwachstelle die ein Angreifer in diesem Programm ausnutzen könnte.

1

2

```
1 #include <stdio.h>
2 int main(void) {
3     char buf[1024];
4     gets(buf);
5     int i = atoi(buf);
6     printf("Int: %d", i);
7 }
```

0 c)* Finden erläutern und korrigieren Sie die Schwachstelle die ein Angreifer in diesem Programm ausnutzen könnte.

1

2

```
1 #include <stdio.h>
2 int main(void) {
3     char buf[1024];
4     scanf("%1024s", buf);
5     int i = atoi(buf);
6     printf("Int: %d", i);
7 }
```

d)* Beschreiben Sie kurz den Unterschied zwischen einem monolithischen und einem Mikrokern-basierten Betriebssystem.



e)* Was ist ein Livelock?



f)* Erläutern Sie die Begriffe *Waise (Orphan)* und *Zombie* im Kontext von Prozessen unter Linux. Ist ein *Dämonen-Prozess* ein Waise oder ein Zombie?



g)* Worin besteht der Unterschied zwischen externer und interner Fragmentierung von Speicherbereichen?



h)* Beschreiben Sie 2 mögliche Nachteile/Probleme von Parallelisierung.



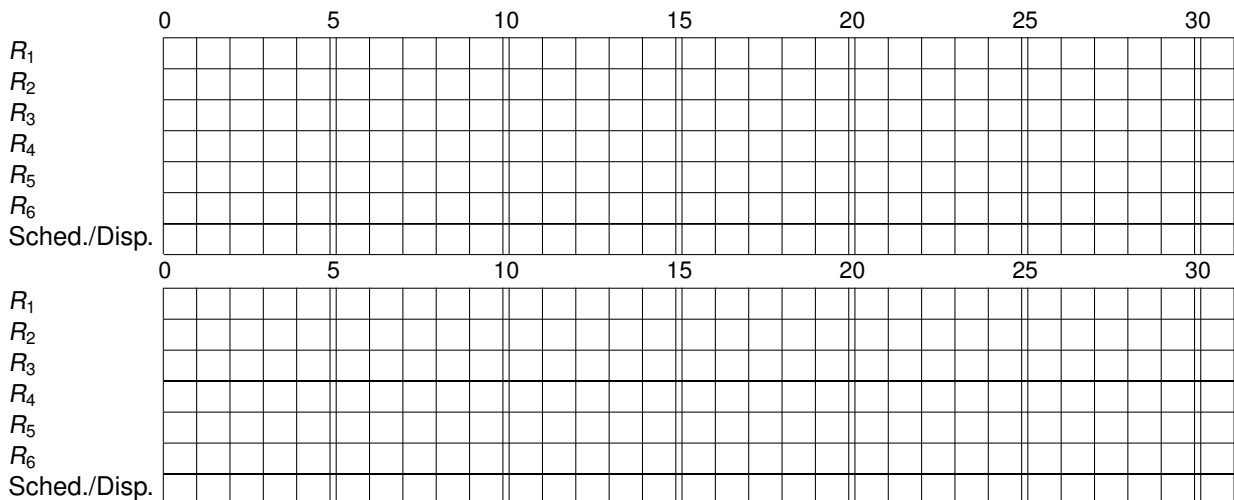
Aufgabe 2 Scheduling (15 Punkte)

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- a)* Für das Scheduling in einem Echtzeitsystem werde die präemptive **Earliest Deadline First** Strategie verwendet. Hierbei gibt jeder Prozess einen Zeitpunkt an, zu dem seine Ausführung abgeschlossen sein muss. Konnte ein Prozess bis zu seiner Deadline nicht vollständig abgeschlossen werden, wird er zur Deadline abgebrochen und aus dem System entfernt, wobei der Kontrollfluss an den Scheduler übergeht. Stehen nach der Earliest Deadline First Strategie mehrere Prozesse zur Auswahl, so wird die CPU dem Prozess mit der kleineren ID zugewiesen (z.B. R_1 und R_3 stehen zur Auswahl $\Rightarrow R_1$ wird ausgewählt). Das Scheduling benötige stets eine Zeiteinheit, ein Kontextwechsel dauere eine weitere Zeiteinheit. Für die Realtimeprozesse R_1 bis R_6 gelten folgende Deadlines, Ankunftszeiten im System und benötigte Zeiteinheiten zur vollständigen Ausführung:

Prozess	Ankunftszeit	Benötigte Rechenzeit	Deadline
R_1	0	4	7
R_2	3	1	9
R_3	8	4	30
R_4	16	5	24
R_5	8	3	11
R_6	21	2	30

Vernachlässigen Sie den initialen Kontextwechsel, sprich der erste Prozess beginne zum Zeitpunkt 0 direkt mit der Ausführung.

Nutzen Sie den zweiten Vordruck, falls Sie Fehler im ersten Vordruck korrigieren möchten. Machen Sie klar erkenntlich, welche Lösung gewertet werden soll (sonst keine Wertung möglich)!



- 0
- 1
- b) Wie könnte die **Earliest Deadline First** Strategie verändert werden, sodass das Verhungern von system-relevanten Prozessen vermieden wird?

c)* Bei der **Round Robin** Strategie ist eine geeignete Wahl des Zeitquantums von herausragender Bedeutung. Geben Sie jeweils einen Vorteil der Wahl eines kurzen und eines langen Zeitquantums an und geben Sie die Prozessklassen an, die hiervon profitieren.



Kurzes Zeitquantum:

Langes Zeitquantum:

d)* Geben Sie zwei unterbrechende und zwei nicht unterbrechende Schedulingstrategien an (nicht Earliest Deadline First).



Unterbrechend:

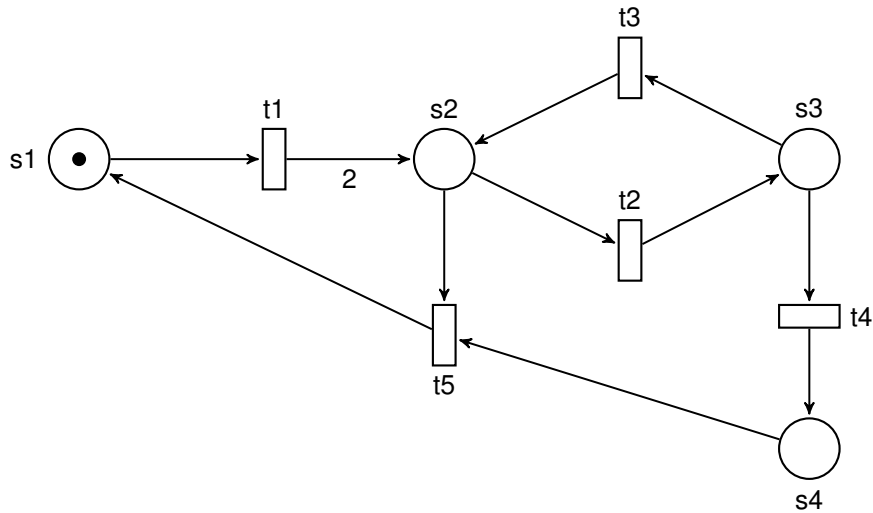
Nicht unterbrechend:

e)* Grenzen Sie die Aufgaben des Schedulers und Dispatchers voneinander ab. Gehen Sie dabei insbesondere auf die Bedeutung des Process Control Block (PCB) ein.



Aufgabe 3 Petrinetze (10 Punkte)

Gegeben sei folgendes **natürliczzahliges** Petrinetz:



- 0
- 1
- 2
- 3
- 4
- 5
- 6

a)* Geben Sie den Erreichbarkeitsgraphen zum gegebenen Petrinetz an.

- 0
- 1

b)* Wie können Sie Verklemmungen anhand eines Erreichbarkeitsgraphens erkennen?

c) Welche Eigenschaften erfüllt das Petrinetz? Begründen Sie Ihre Antwort.

0
1
2

d)* Fügen Sie eine Transition in das Petrinetz ein oder verändern Sie eine bestehende Transition, sodass der Erreichbarkeitsgraph nicht verändert wird.

0
1

Aufgabe 4 Multiple Choice (8 Punkte)

Kreuzen Sie richtige Antworten an

Kreuze können durch vollständiges Ausfüllen gestrichen werden

Gestrichene Antworten können durch nebenstehende Markierung erneut angekreuzt werden



a) Was trifft auf Synchronisation zu?

- TSL (Test and Set Lock) funktioniert auf Einprozessorsystemen.
- Ein kritischer Bereich darf nur aus maximal drei Befehlen bestehen.
- Mehrprozessorsysteme lassen sich durch Unterbrechungssperren synchronisieren.
- Eine Instruktion ist immer unteilbar.

b) Ein Mutex hat folgende Eigenschaften:

- Die Verwendung von Mutexes garantiert Fairness.
- Durch Mutexes kann man Verklemmungen erkennen.
- Durch Mutexes kann man Verklemmungen verhindern.
- Mutexes helfen bei der Synchronisation.

c) Welche Eigenschaften haben asynchrone oder synchrone Kommunikation?

- Bei synchroner Kommunikation wartet der Sender auf eine Bestätigung durch den Empfänger.
- Bei asynchroner Kommunikation wartet der Sender auf eine Bestätigung durch den Empfänger.
- Bei asynchroner Kommunikation wartet der Sender nicht auf eine Bestätigung durch den Empfänger.
- Bei synchroner Kommunikation wartet der Sender nicht auf eine Bestätigung durch den Empfänger.

d) Kleines Kopfrechnen mit Binär- und Dezimalpräfixen.

- 4096 MiB sind 4 GiB
- 1024 GB sind 1 TB
- 10 MB sind 0,0001 TB
- 2 KiB sind 2000 Byte

