

Klausur Betriebssysteme und Sicherheit, 18.07.2023

— Bearbeitungszeit: 90 Minuten — Prüfer: Prof. Dr. Schirmeier, Dr. Köpsell —

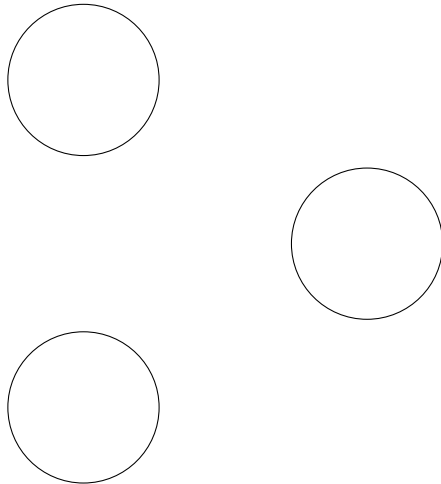
1	2	3	4	5	6	Σ
15	15	15	13	14	18	90

Alle Aussagen sind so ausführlich wie nötig, aber so knapp wie möglich zu begründen.

Aufgabe 1 – Scheduling

15 Punkte

- a) Ergänzen Sie im vorgegebenen Diagramm die drei Prozesszustände der kurzfristigen Einplanung (*short-term scheduling*) und deren Übergänge. Nummerieren Sie die Kanten und beschreiben Sie zu jedem Zustandsübergang kurz, durch welches Ereignis er ausgelöst wird. **4 P**



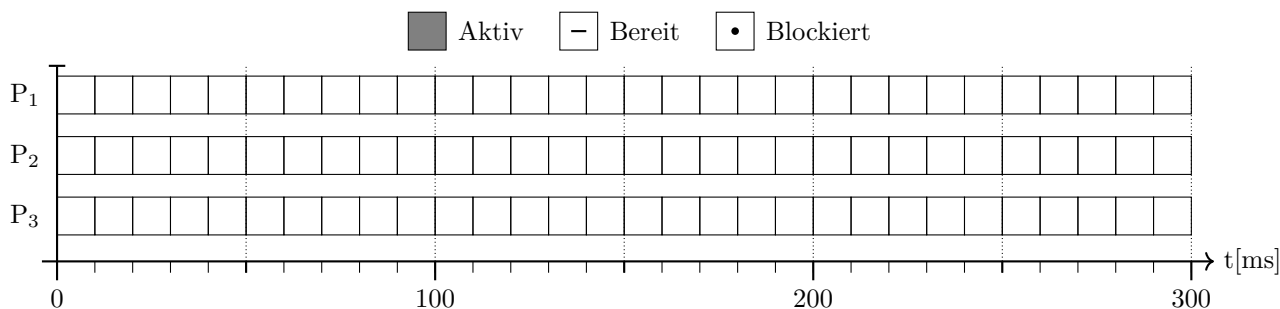
- b) Erklären Sie stichpunktartig die Vorgehensweise des Scheduling-Verfahrens *Shortest Process Next* (SPN). Erläutern Sie kurz, wie das Betriebssystem den „Shortest Process“ ermitteln kann, wenn die Programme nicht mit Zeitschranken annotiert sind. **2 P**

- c) In einem Uniprozessorsystem verwaltet das Betriebssystem die drei Prozesse P_1 , P_2 und P_3 , die zum Zeitpunkt $t = 0$ alle rechenbereit sind. Jeder Prozess hat drei CPU-Stöße, deren Dauer (in ms) in folgender Tabelle aufgeführt wird:

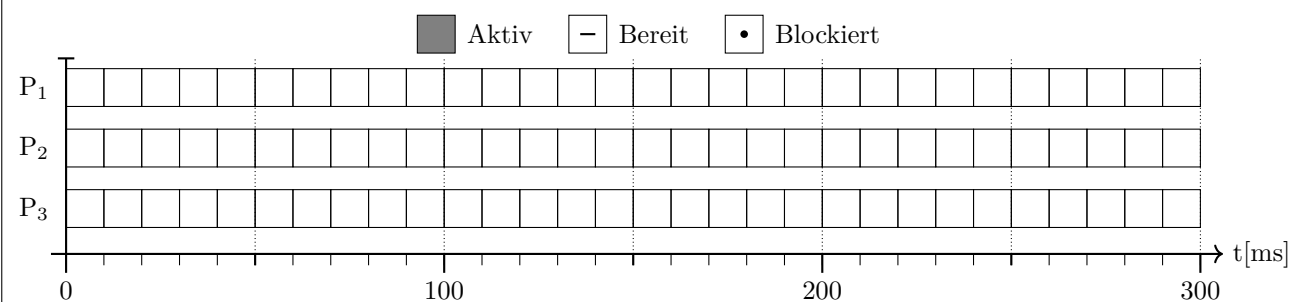
Prozesse	CPU-Stöße		
P_1	40	20	20
P_2	60	60	30
P_3	10	20	10

Die E/A-Stöße in dem System dauern jeweils genau 10 ms.

Zeichnen Sie in das folgende leere Gantt-Diagramm ein, wie die drei Prozesse unter Anwendung der Strategie *Shortest Process Next* (SPN) abgearbeitet werden würden. **5 P**



Ersatzdiagramm (Streichen Sie ungültige Lösungen deutlich durch):



- d) Welche „Gefahr“ besteht grundsätzlich bei Verwendung von SPN, und wodurch wird diese bei *Highest Response Ratio Next* (HRRN) vermieden? **2 P**

- e) Wir betrachten nun ein Echtzeitsystem. Begründen Sie, ob folgende periodische Tasks mit dem Verfahren *Earliest Deadline First* (EDF) auf einem Uniprozessorsystem einplanbar sind. Es gilt $\text{Periodenende} = \text{Zeitschranke}$.

Tasks	Periodenlänge	Bearbeitungszeit
T_1	$p_1 = 5\text{s}$	$t_1 = 2\text{s}$
T_2	$p_2 = 3\text{s}$	$t_2 = 1\text{s}$
T_3	$p_3 = 4\text{s}$	$t_3 = 1\text{s}$

2 P

Aufgabe 2 – Sicherheit

15 Punkte

- a) Was verstehen Sie unter dem Recht auf informationelle Selbstbestimmung? Warum ist es im Hinblick auf die freie Entwicklung der Persönlichkeit wichtig? **2 P**
- b) Die Verarbeitung personenbezogener Daten hat transparent zu erfolgen. Welche Rechte bzw. Pflichten werden daraus abgeleitet? Nennen und erläutern Sie diese kurz. **2 P**
- c) Im Zuge der Risikoanalyse wurden mögliche Gefährdungen ermittelt. Beschreiben Sie in knapper Form die Bewertung dieser Gefährdungen mit Hilfe der Risikomatrix! Welche Vorteile bietet die Verwendung der Risikomatrix im Vergleich zur Anwendung der Risikoformel? **4 P**
- d) Was besagt das Prinzip von Kerkhoffs? Geben Sie zwei Argumente an, die begründen, warum dieses Prinzip sinnvoll ist! **2 P**

e) Was ist jeweils der Vorteil symmetrischer bzw. asymmetrischer Kryptographie? Nennen und erläutern Sie diese kurz. **2 P**

f) Um die jeweiligen Vorteile symmetrischer und asymmetrischer Kryptographie zu nutzen, werden hybride Systeme eingesetzt. Betrachten Sie den Fall, dass Teilnehmer A an Teilnehmer B eine verschlüsselte Nachricht mit einem hybriden System schicken will und beschreiben Sie in knapper Form den Ablauf (welche Schritte führt A aus, welche Schritte führt B aus; welche Schlüssel werden verwendet, was wird mit dem symmetrischen bzw. dem asymmetrischen System verschlüsselt)! **3 P**

Aufgabe 3 – Betriebssystemgrundlagen

15 Punkte

a) Nennen Sie drei Aufgaben, die ein Betriebssystem erfüllt.

3 P

b) In einem System – bestehend aus einer CPU, Hauptspeicher, einer Festplatte und einer Netzwerkkarte – erreicht ein Netzwerkpaket die Netzwerkkarte. Welche zwei Hardware-Mechanismen werden benötigt, damit die Netzwerkkarte das Paket an das Betriebssystem melden und dieses weiter verarbeiten kann? Erklären Sie jeweils kurz, wofür der Mechanismus benötigt wird.

4 P

c) Aufgrund des in Teilaufgabe (b) auf dem System eingetroffenen Netzwerkpaketes entscheidet sich das Betriebssystem von dem aktuell laufenden Prozess „vsChiffre“ auf den wartenden Prozess „Feuerfuchs“ zu wechseln. Wählen Sie die dazu nötigen Schritte aus der unten stehenden Tabelle aus und notieren Sie die zugeordneten Buchstaben in der korrekten Reihenfolge.

HINWEIS: Die Auswahl falscher Schritte führt zu Punktabzug innerhalb dieser Teilaufgabe!

- a) Offene Dateien von „vsChiffre“ schließen
- b) Scheduler-Informationen aktualisieren
- c) Freie Rahmen finden/schaffen und als belegt markieren
- d) Hardwarezustand (Register, ...) sichern
- e) Hardwarezustand (Register, ...) wiederherstellen
- f) Prozessinformationen von „Feuerfuchs“ finden
- g) Prozessinformationen von „vsChiffre“ löschen
- h) Bankalgorithmus ausführen
- i) Adressraum umschalten (Register CR3 schreiben)
- j) Elternprozesse von „vsChiffre“ beenden

6 P

d) Mit der Aktivierung des Prozesses „Feuerfuchs“ aus Teilaufgabe (c) wird eine verschlüsselte Verbindung via HTTPS zu einer Website aufgebaut. Dafür benötigt der Prozess Zufallszahlen, um einen Sitzungsschlüssel für die zugehörige TLS-Verbindung zu generieren. Der Prozess verwendet dabei die Schnittstelle `/dev/random` des Betriebssystems. Nennen Sie zwei Möglichkeiten, wie das Betriebssystem echte Zufallszahlen generieren kann, um diese dem Prozess zur Verfügung zu stellen.

2 P

Aufgabe 4 – Synchronisation und Verklemmung

13 Punkte

Gegeben seien drei parallele Threads, die der Vorlesung entsprechend definierte binäre Semaphore vom Typ `sem_t` nutzen. Zudem geben die Threads über die Funktion `puts` Zeichenketten an den Nutzer aus. Die Semaphore sind wie folgt initialisiert:

```
1 /* Initialisierung der sem_t-Variablen */
2 sem_t m(1);
3 sem_t s1(1);
4 sem_t s2(0);
5 sem_t s3(1);
```

T₁

```
6 s1.wait();
7 m.wait();
8 puts("Romanes");
9 m.signal();
10 s2.signal();
11 s1.signal();
```

T₂

```
12 s2.wait();
13 m.wait();
14 puts("eunt");
15 m.signal();
16 s2.signal();
```

T₃

```
17 s3.wait();
18 m.wait();
19 puts("domus");
20 m.signal();
21 s3.signal();
```

- a) Der vorliegende Code nutzt Semaphore zur Synchronisation. Warum kann er dennoch verschiedene Ausgaben erzeugen? Nennen Sie alle möglichen Ausgaben. **4 P**

- b) Können bei der Ausführung dieses Codes Verklemmungen auftreten? Begründen Sie anhand der aus der Vorlesung bekannten Bedingungen für eine Verklemmung. **5 P**

- c) Der Code soll stets „Romanes eunt domus“ ausgeben. Welche Änderungen sind in Bezug auf Semaphore im Code nötig? **3 P**

- d) Was ändert sich bezüglich der Aussagen in (a) und (b), wenn Spinlocks an Stelle von Semaphore zum Einsatz kommen? Begründen Sie Ihr Antwort. **1 P**

Aufgabe 5 – Unix

14 Punkte

Ein UNIX-Prozess führt folgenden Code aus:

```
1 int counter1 = 1;
2 int counter2 = 1;
3 char buf[2];
4
5 int main(){
6     int file1 = open("datei1.txt", O_CREAT|O_WRONLY);
7     int file2 = open("datei2.txt", O_CREAT|O_WRONLY);
8
9     int pid = fork();
10
11     for(int i = 0; i < 3; i++) {
12         if(pid == 0) {
13             write(file1, "a", 1);
14             counter1++;
15         }
16         else {
17             write(file2, "b", 1);
18             counter2++;
19         }
20     }
21
22     if(pid == 0) {
23         int sum = counter1 + counter2;
24         printf("Counter: %d", sum);
25     }
26     return 0;
27 }
```

Die Dateien `datei1.txt` und `datei2.txt` existieren vor Programmausführung nicht, sondern werden durch das Öffnen angelegt. Alle Systemaufrufe laufen korrekt zu Ende, das heißt es treten keine Laufzeitfehler auf.

HINWEIS:

Der beispielhafte Aufruf von `printf("x: %d", y)` gibt „x: <Variable y>“ auf der Konsole aus, wobei <Variable y> mit der zu diesem Zeitpunkt in Variable y gespeicherten Zahl zu ersetzen ist.

a) Was steht nach Ablauf des Programms in den Dateien `datei1.txt` und `datei2.txt`? Falls es mehrere Varianten gibt, nennen Sie diese und begründen Sie Ihre Lösung. **2 P**

b) Was gibt das Programm auf der Konsole aus? Begründen Sie Ihre Antwort. **2 P**

c) In welchem Segment (Text, Data, BSS, Heap oder Stack) befinden sich die Variablen `counter1` und `counter2`? Begründen Sie ihre Antwort. **2 P**

-
- d) Wie heißt die Hardware-Einheit, welche dafür sorgt, dass Prozesse nur auf ihren eigenen Speicher zugreifen können? Welche Software konfiguriert diese Einheit? **2 P**

In einem UNIX-System existieren die folgenden Nutzer mit unten stehender Gruppenzuordnung:

Nutzer	Gruppen
werner	staff, nutzer
kalle	praktikant, nutzer

Weiterhin existiert das Verzeichnis `results` mit folgenden Zugriffsrechten:

```
drwx r-x --- werner nutzer results
```

In diesem Verzeichnis befinden sich folgende Dateien:

```
rwX r-x --- werner nutzer createFile1
rws r-x --- werner nutzer createFile2
rwX r-x --- kalle nutzer printResult
rw- r-- --- werner nutzer result0.txt
```

Das Programm `createFile1` versucht eine Datei names `result1.txt` im Ordner `results` anzulegen. Das Programm `createFile2` versucht `result2.txt` im Ordner `results` anzulegen. Das Programm `printResult` versucht den Inhalt der Datei `result0.txt` auszugeben. Die Datei `result0.txt` hat den Inhalt „2.0“.

- e) Beschreiben und begründen Sie kurz, was die nachfolgenden Aktionen bewirken. **6 P**

– `kalle` ruft `createFile1` auf.

– `kalle` ruft `createFile2` auf.

– `kalle` ruft `printResult` auf.

Aufgabe 6 – Speicher und Seitenersetzung

18 Punkte

- a) Nennen und erläutern Sie kurz zwei Nachteile von segmentbasierter Adressierung im Vergleich zu seitenbasierter Adressierung. **2 P**

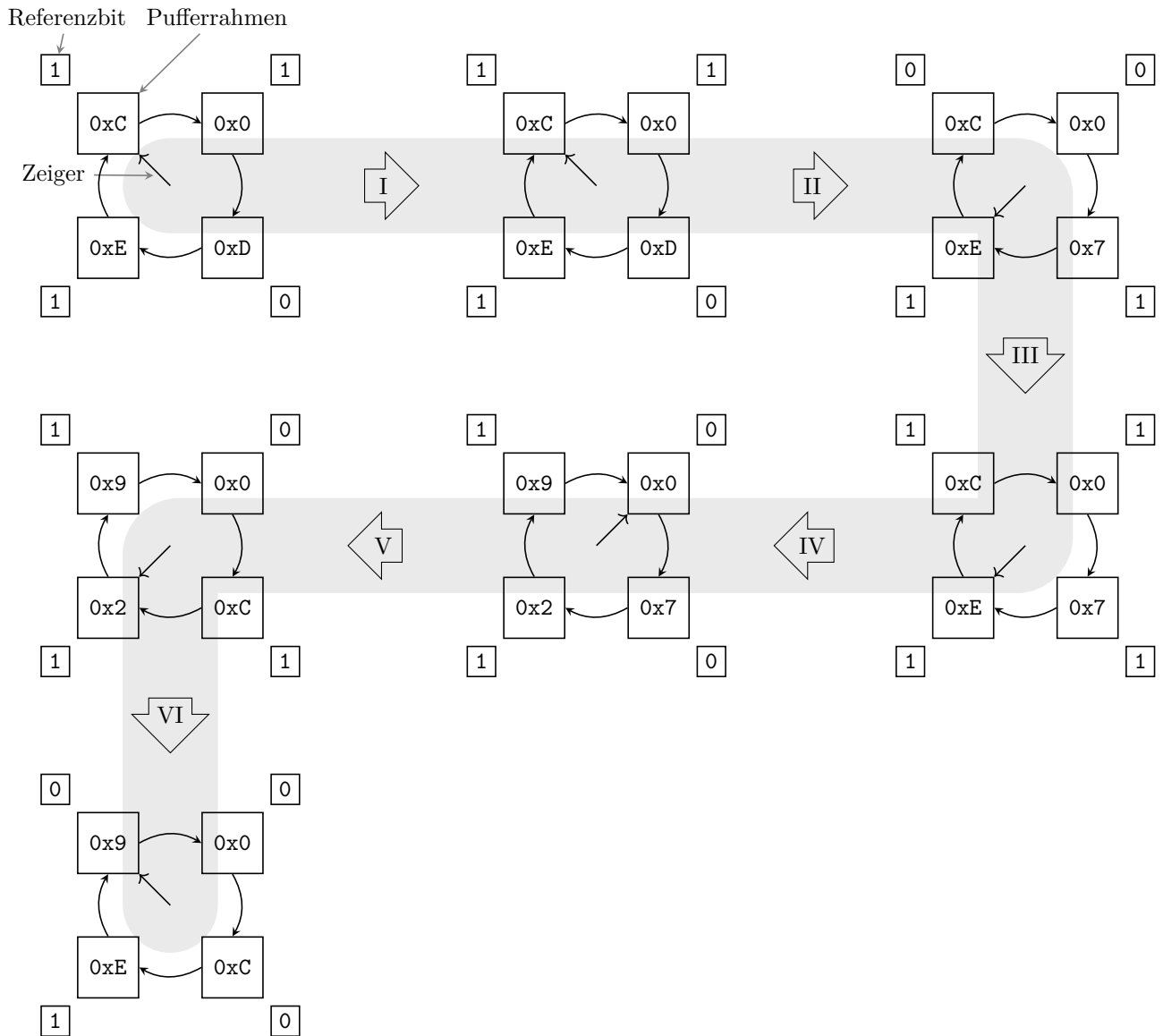
Ein 64-Bit-System verwendet vierstufige Seitentabellen, um virtuelle Adressen in physische Adressen zu übersetzen. Die Tabellen der verschiedenen Stufen enthalten jeweils 8192 Einträge, wobei ein Eintrag 8 Byte benötigt. In diesem System ist eine Seite 4 KiB groß.

- b) Wie viele Bit umfasst der Index für eine Stufe in diesem System? Begründen Sie Ihre Antwort. **1 P**

- c) Wie groß ist der Offset in diesem System? Begründen Sie Ihre Antwort. **1 P**

- d) Auf dem System läuft aktuell ein Prozess, der 128 MiB des virtuellen Speichers verwendet. Berechnen Sie den minimal notwendigen Speicherplatzbedarf der zugehörigen Seitentabellen. Die Angabe von 2er-Potenzen mit passender Einheit genügt. Geben Sie den Lösungsweg an. **6 P**

In dem System werden folgende Zustände im *Clock*-Algorithmus beobachtet.



e) Geben Sie für jede der römischen Zahlen eine Seitenreferenzfolge an, die zu dem entsprechenden Zustandsübergang geführt haben kann. Jede Seitenreferenzfolge besteht aus mindestens einer Seitenreferenz.

8 P

I	II	III
IV	V	VI