Klausur Betriebssysteme und Sicherheit, 22.07.2025

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

1	2	3	4	5	6	$\sum$
15	15	15	15	15	15	90

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

Aufgabe 1 – Kryptographie und digitale Zertifikate

15 Punkte

Eine Blockchiffre kann in verschiedenen Betriebsarten zum Verschlüsseln verwendet werden. Eine der einfachsten Betriebsarten ist Electronic Code Book (ECB).

a) Beschreiben Sie das Vorgehen zur Verschlüsselung eines Klartextes (Plaintext) mit einer Blockchiffre im ECB-Modus sowie zur Entschlüsselung des resultierenden Geheimtextes (Ciphertext). Begründen Sie dabei, ob bzw. warum Blockchiffren symmetrische oder asymmetrische Verfahren darstellen.

3 P

b) Gegeben ist ein Geheimtext der mit einer 8-Bit-Blockchiffre im ECB-Modus verschlüsselt wurde. Tabelle 1 zeigt den Geheimtext und Klartext einer Nachricht der auf die gleiche Weise verschlüsselt wurde wie der gegebene Geheimtext.

Tabelle 1: Geheimtext und Klartext dargestellt als Hex bzw. ASCII-Zeichen

Geheimtext	63	65	6E	79	79	2B	69	62	7A	2B	6A	75	76	66	78	6C	2B
	73	79	62	74	2B	64	68	6E	6B	2B	71	72	61	2B	77	72	67	2B
	6D	68	2B	6F	65	68	70	75									
Klartext	p	r	a	l	l	-	v	o	m	-	w	h	i	s	k	y	-
	f	l	o	g	-	q	u	a	x	-	d	e	n	-	j	e	t	-
	z	u	-	b	r	u	c	h									

Führen Sie einen Known-Plaintext-Angriff durch. Entschlüsseln Sie hierzu mithilfe des bekannten Klartexts folgende Nachricht und erläutern Sie Ihr Vorgehen dabei. Der Geheimtext ist Hexadezimal angegeben.

Geheimtext	73	72	66	67	76	61	6E	2B	79	72	61	67	72
Klartext													

4 P

c) Erläutern Sie die wesentlichen Unterschiede zwischen den Betriebsarten ECB und CBC (Cipher Block Chaining) im Bezug auf Sicherheit und Performanz.

2 P

Digitale Zertifikate spielen eine zentrale Rolle für die sichere Internetkommunikation. Beantworten Sie die folgenden Fragen.

d) Nennen Sie zwei Faktoren, die für die Sicherheit eines digitalen Zertifikats eine Rolle spielen. **2 P**

e) Wie kann man kryptographisch überprüfen, ob ein digitales Zertifikat gültig ist? Welcher Schlüssel kommt dabei zum Einsatz und von wem stammt dieser? **2 P**

f) Was ist der Zweck von Zertifikaten im Web, beispielsweise beim TLS-Handshake? **1 P**

g) Welche Risiken bestehen, wenn man einem gefälschten Zertifikat vertraut? **1 P**

Aufgabe 2 – Sicherheitslücken und Datenanonymisierung **15 Punkte**

Gegeben sei der C-Code in Abbildung 1.
Nehmen Sie an, dass keine zusätzlichen Schutzmechanismen, insbesondere keine vom Compiler bereitgestellten Sicherheitsfunktionen, aktiv sind.
Hinweis zur Funktionsweise von `strncmp`*:* Die Funktion `strncmp(const char* str1, const char* str, size_t num)` vergleicht bis zu `num` Zeichen von `str1` mit denen von `str2` und gibt den Wert 0 zurück, wenn sie gleich sind. Die Funktion beginnt mit dem Vergleich des ersten Zeichens jeder Zeichenkette. Wenn sie gleich sind, wird mit den folgenden Paaren fortgefahren bis sich die Zeichen unterscheiden, bis ein abschließendes Null-Zeichen erreicht wird oder bis `num` Zeichen in beiden Zeichenketten übereinstimmen.

```
1  #include <stdio.h>
2
3  int main() {
4          char password[8] = "secret";
5          char input[8];
6
7          printf("Enter password:");
8          gets(input);
9          if (strncmp(password, input, 8) == 0) {
10                 printf("access granted\n");
11         } else {
12                 printf("access denied\n");
13         }
14         return 0;
15 }
```

Abbildung 1: C-Code.

a) Was ist ein Buffer Overflow? Kann es es in Abb. 1 zu einem Buffer Overflow kommen? Falls ja, beschreiben Sie warum. Falls nein, begründen Sie Ihre Antwort. **2 P**

b) Können Sie eine Eingabe konstruieren, so dass `strncmp` in Zeile 9 den Wert 0 zurückliefert, ohne dass Sie das Passwort kennen? Falls ja, beschreiben Sie das allgemeine Vorgehen. Falls nein, begründen Sie Ihre Antwort. **2 P**

c) Was könnte man in Abb. 1 hinzufügen, um die Schwachstelle zu beseitigen? Geben Sie Pseudocode an und nennen Sie wo (Zeilennummer) dieser hinzugefügt werden soll. **2 P**

d) Nennen und erklären Sie zwei Schutzmechanismen, die gegen Buffer-Overflow-Angriffe wirken. **2 P**

Mit k-Anonymität kann die Offenlegung von Informationen kontrolliert und Identität verschleiert werden. In Tabelle 2 finden Sie ein Beispiel für einen Datensatz mit sensiblen Krankheitsdaten, der nach dem Prinzip der k-Anonymität anonymisiert wurde. Nehmen Sie an, dass die Krankheit das sensitive Attribut ist.

Tabelle 2: Beispiel für k-Anonymität

Ort	**Geschlecht**	**Krankheit**	**Alter**
Deutschland	Männlich	Muskelzerrung	20–59
Deutschland	Männlich	Muskelzerrung	20–59
Deutschland	Männlich	Krebs	20–59
Deutschland	Männlich	Krebs	20–59
Deutschland	Weiblich	Haarausfall	20–59
Deutschland	Weiblich	Haarausfall	20–59
Deutschland	Weiblich	Krebs	20–59
Deutschland	Weiblich	Krebs	20–59
Frankreich	Männlich	Karies	20–79
Frankreich	Männlich	Karies	20–79
Frankreich	Männlich	Grippe	20–79
Frankreich	Männlich	Grippe	20–79
Frankreich	Weiblich	Grippe	20–79
Frankreich	Weiblich	Grippe	20–79
Frankreich	Weiblich	Haarausfall	20–79
Frankreich	Weiblich	Haarausfall	20–79
Polen	Männlich	Vergiftung	20–69
Polen	Männlich	Vergiftung	20–69
Polen	Männlich	Krebs	20–69
Polen	Männlich	Krebs	20–69
Polen	Weiblich	Muskelzerrung	20–69
Polen	Weiblich	Muskelzerrung	20–69
Polen	Weiblich	Vergiftung	20–69
Polen	Weiblich	Vergiftung	20–69

e) Erklären Sie, wie das Prinzip der k-Anonymität dabei helfen soll Datenschutz zu erreichen. Welchen Wert hat k? Wie viele Äquivalenzklassen gibt es? **3 P**

f) Was müssten Sie an den Werten des Datensatzes ändern, um ein $k = 8$ zu erreichen? **1 P**

g) Was versteht man unter dem *Privacy-Utility Tradeoff* in Bezug auf Datenanonymisierung? **1 P**

h) Was bedeutet ℓ-Diversity? Welchen Wert hat ℓ in dem Beispiel? **2 P**

Aufgabe 3 – Synchronisation

15 Punkte

Es seien die folgenden Codeabschnitte A und B gegeben, die in zwei unabhängigen Threads ausgeführt werden. Die Initialisierung wird vor der Ausführung der beiden Threads einmal durchgeführt.

Initialisierung

```
int value_1 = 100;
int value_2 = 100;
```

Codeabschnitt A

```
value_1 = value_1 - 100;
value_2 = value_2 + 100;
```

Codeabschnitt B

```
value_2 = value_2 - 50;
value_1 = value_1 + 50;
```

a) Zeigen Sie anhand der folgenden Definition, dass zwischen den Codeabschnitten A und B eine Wettlaufsituation vorliegt.

> **Definition** Wettlaufsituation:
> Gegeben seien mehrere Prozesse, die sich unabhängig voneinander um die zeitweise exklusive Nutzung derselben Betriebsmittel bewerben. Eine Wettlaufsituation liegt vor, wenn die parallele Ausführung zweier Codeabschnitte in diesen Prozessen zu einem anderen Zustand führen kann als die sequenzielle Ausführung dieser beiden Codeabschnitte.

5 P

b) Verwenden Sie Semaphore um die Wettlaufsituation aufzulösen. Die Semaphore sollen so verwendet werden, dass die beiden Codeabschnitte A und B maximal parallel ausgeführt werden können. Tragen Sie Ihre Lösung in die grau hinterlegten Lücken im Quelltext ein.

HINWEIS: Falls Sie ihre Antwort korrigieren möchten, können Sie das Ersatzdiagramm am Ende der Aufgabe nutzen. **Streichen Sie in diesem Fall ungültige Lösungen deutlich durch!**

5 P

Initialisierung

```
int value_1 = 100;
int value_2 = 100;
```

Codeabschnitt A

```
value_1 = value_1 - 100;
value_2 = value_2 + 100;
```

Codeabschnitt B

```
value_2 = value_2 - 50;
value_1 = value_1 + 50;
```

Alternativ kann die Wettlaufsituation, wie unten gezeigt, mittels Schlossvariablen gelöst werden:

Initialisierung

```
int value_1 = 100;
int value_2 = 100;


bool req_a = false;
bool req_b = false;
```

Codeabschnitt A

```
req_a = true;
while (req_b == true) {
}
value_1 = value_1 - 100;
value_2 = value_2 + 100;
req_a = false;
```

Codeabschnitt B

```
req_b = true;
while (req_a == true) {
}
value_2 = value_2 - 50;
value_1 = value_1 + 50;
req_b = false;
```

c) Im Zusammenhang mit Schlossvariablen wird deren Lebendigkeit diskutiert. Nennen Sie die Kriterien für Lebendigkeit.

3 P

d) Zeigen Sie anhand der Lebendigkeitskriterien, dass diese Implementierung nicht lebendig ist. Nennen Sie ein Kriterium, das nicht erfüllt ist, und erklären Sie, warum dieses Kriterium nicht erfüllt ist.

HINWEIS: Beachten Sie, dass die Codeabschnitte A und B nicht wiederholt ausgeführt werden.

2 P

Ersatzdiagramm (Streichen Sie ungültige Lösungen deutlich durch):

Initialisierung

```
int value_1 = 100;
int value_2 = 100;
```

Codeabschnitt A

```
value_1 = value_1 - 100;
value_2 = value_2 + 100;
```

Codeabschnitt B

```
value_2 = value_2 - 50;
value_1 = value_1 + 50;
```

Aufgabe 4 – Speicher

15 Punkte

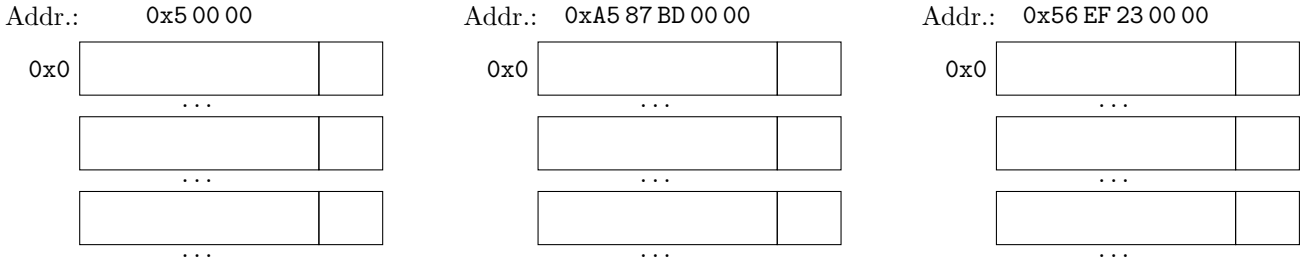
In einem 40-Bit-System wird eine 2-stufige Seitentabelle zur Verwaltung der virtuellen Adressräume verwendet, wobei die Indizes der zwei Stufen jeweils 12 Bit groß sind. Die Rechte werden mittels eines *present*-Bits (p) und eines *writable*-Bits (w) repräsentiert, die bei einem Zugriff auf beiden Stufen überprüft werden.

a) Nach dem Start eines neuen Prozesses mit leerem Adressraum werden die folgenden Speicherzugriffe ausgeführt:

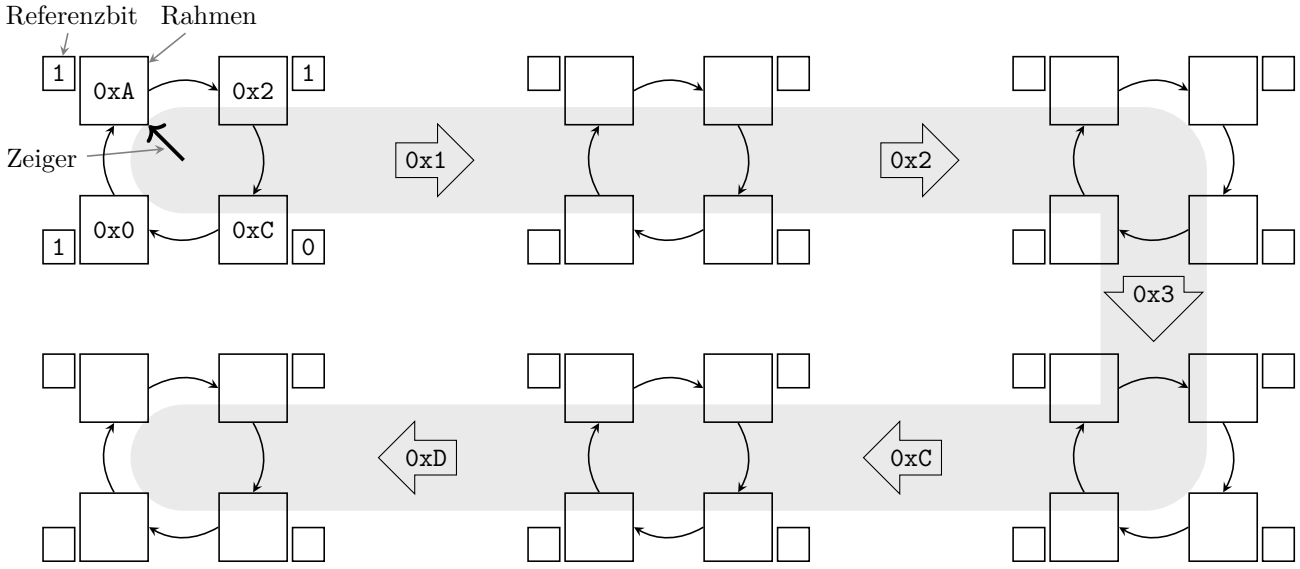
- Schreibend auf `0x6 32 23 41`
- Schreibend auf `0xAB C6 32 45 61`
- Lesend auf `0xAB C4 51 34 51`

Alle Seiten liegen dabei in Regionen, für die der Prozess die erforderlichen Zugriffsrechte hat, d.h. alle Zugriffe sollen gültig sein. Dem System stehen freie Rahmen an den physischen Adressen `0xA 20 00 00`, `0xC 64 00 00` und `0x80 86 00 00` zur Verfügung.

Welchen Zustand hat das System nach diesen Zugriffen? Tragen Sie im nachstehenden Schema die entsprechenden Zeilenindizes, Rahmennummern sowie Rechtebits ein. Der Einsprungpunkt für die Adressübersetzung, also die Tabelle der ersten Stufe, liegt an Adresse `0x5 00 00`. Zwei weitere Tabellen stehen an den im Schema angegebenen Adressen bereit. **8 P**



b) Für die Seitenverdrängung nutzt das verwendete Betriebssystem den Clock-Algorithmus. Gegeben sei folgende Zugriffsreihenfolge auf Seiten im System: `0x1`, `0x2`, `0x3`, `0xC`, `0xD`
Notieren Sie alle Änderungen und Ersetzungen gemäß dem Clock-Algorithmus, die sich aus dieser Zugriffsreihenfolge ergeben. Starten Sie dabei vom ersten ausgefüllten Zustand. **5 P**



c) Nennen Sie *eine* Ursache für Thrashing (Seitenflattern) und geben Sie *kurz* an, welches Problem durch Thrashing hervorgerufen wird.

2 P

Aufgabe 5 – Unix **15 Punkte**

Gegeben ist ein UNIX-artiges System mit den Benutzern `manager`, `datenanalyst` und `entwickler`.
Im System existieren im Verzeichnis `/etc/` zwei Dateien, auf die die Benutzer mit bestimmten Einschränkungen zugreifen können müssen.
Um diese Einschränkungen darzustellen, wurde die folgende Rechtevergabe in Form von *Capabilities* erarbeitet:

`entwickler`:	(`daten.csv`; read)	(`deploy`; read, write, execute)
`manager`:	(`daten.csv`; read)	(`deploy`; read, execute)
`datenanalyst`:	(`daten.csv`; read, write)	(`deploy`; execute)

a) Überführen Sie die gegebenen Berechtigungen in Unix-Zugriffsrechte. Beachten Sie, dass es im System **keine** weiteren Benutzer gibt. Nutzen Sie für Ihre Lösung die unten stehende Tabelle und geben Sie sowohl die eingeführten Gruppen als auch die Benutzerzuordnung zu den Gruppen an. **6 P**

Datei	**Rechte-Bits**	**Besitzer**	**Gruppe**
`daten.csv`			
`deploy`			

Gruppe	**Benutzer**

Die Dateien liegen auf einer SSD mit einer Blockgröße von 8 KiB wie folgt im unten skizzierte Dateisystem. Das Dateisystem nutzt eine Blockgröße von 8 KiB. Die dargestellten Blöcke enthalten entweder Inodes (IB), Superblöcke (SB), Verzeichnisblöcke (VB) oder eine Allokations-Bitmap für Blöcke (BA). Weitere Blöcke (z.B. für Dateiinhalte) existieren, sind aber nicht dargestellt. Einträge in den Inodeblöcken haben dabei die Form:

`<Inode-Nummer>` → `<Liste der Datenblöcke> / <Dateigröße>`

ID	0	1	2	3	4
Typ	SB	BA	IB	VB	VB

Inhalt	free: 23
	root: `I0`
		`0-6`	`I0` → `4 / 32`
			`I1` → `3 / 32`
			`I2` → `5,6 / 15000`
				`data.csv` → `I2`	`etc` → `I1`

b) Welche Blöcke des Dateisystems müssen in welcher Reihenfolge gelesen werden, wenn aus der Datei `/etc/data.csv` 6 KiB ab einem Offset von 6 KiB eingelesen werden sollen? Nehmen Sie dabei an, dass die Inhalte eines Blockes immer vollständig gelesen werden (also kein Block ein zweites Mal gelesen werden muss). **6 P**

c) Im Folgenden werden 2000 Bytes Daten an die Datei `/etc/data.csv` angehangen. Geben sie den aktualisierten Eintrag für Inode `I2` in der folgenden Form an:
`I2` → `<Liste der Datenblöcke> / <Dateigröße>` **3 P**

Aufgabe 6 – Scheduling

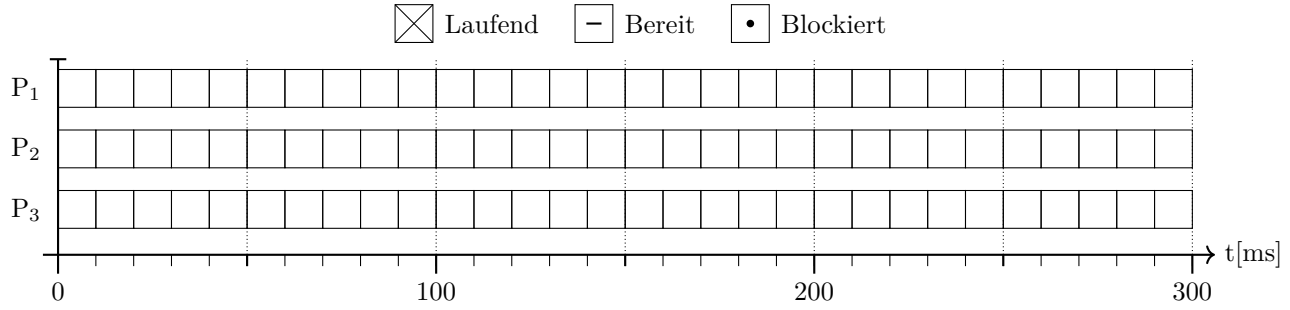
15 Punkte

Ein Einprozessor-Betriebssystem verwaltet drei Prozesse P_1, P_2 und P_3. Die Prozesse treffen in dieser Reihenfolge im System ein und sind alle zum Zeitpunkt $t = 0$ rechenbereit. Die Prozesse wiederholen sich unendlich lange. Nach jedem CPU-Stoß führen die Prozesse einen E/A-Stoß durch. Die CPU-Stöße (in ms) und E/A-Stöße (in ms) der Prozesse sind in der Tabelle 3 angegeben.

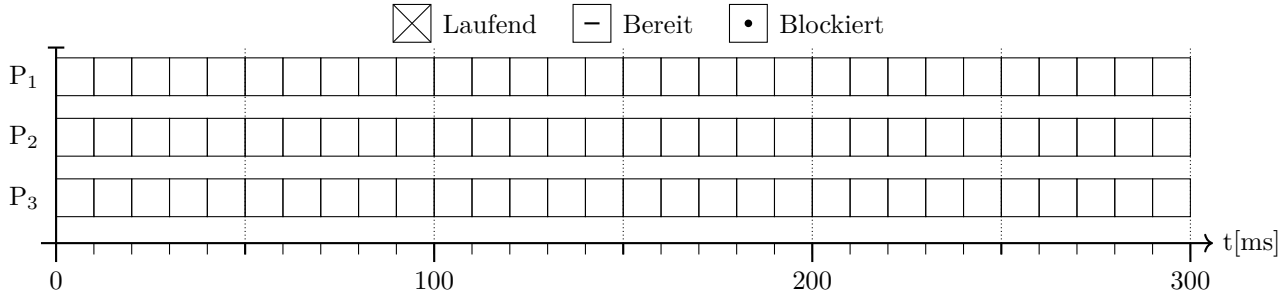
Prozesse	P_1	P_2	P_3
CPU-Stöße	30	40	50
E/A-Stöße	50	40	30

Tabelle 3: Prozesslaufzeiten

a) Zeichnen Sie in das folgende Gantt-Diagramm ein, wie die drei Prozesse P_1, P_2 und P_3 bearbeitet werden, wenn das Scheduling nach der „Virtual Round Robin“-Strategie mit einer Zeitscheibe von 30 ms vorgenommen wird. Nehmen Sie an, dass die Prozesse die in der Tabelle 3 gegebenen Laufzeiten haben und der Scheduler auf Basis dieses Wissens entscheiden kann. **6 P**



<u>Ersatzdiagramm</u> (Streichen Sie ungültige Lösungen deutlich durch):



b) Bei einem der Schedulingaufrufe in einem Betriebssystem mit präemptivem Scheduling wird Prozess A durch Prozess B verdrängt. Welche Schritte müssen dafür im Betriebssystem durchgeführt werden? Wählen Sie die dazu nötigen Schritte aus der unten stehenden Liste aus und notieren Sie die zugeordneten Buchstaben in der korrekten Reihenfolge.

HINWEIS: Die Auswahl falscher Schritte führt zu Punktabzug innerhalb dieser Teilaufgabe! **6 P**

a) Offene Dateien von A 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 B finden

g) Prozessinformationen von A löschen

h) Bankalgorithmus ausführen

i) Adressraum umschalten (Register CR3 schreiben)

j) Elternprozesse von A beenden

c) In dem unten stehenden Diagramm sind die drei Prozesszustände der kurzfristigen Einplanung (*short-term scheduling*) gegeben. Ergänzen Sie die Abbildung um alle erlaubten Übergänge zwischen den Zuständen in Form von Pfeilen.

Markieren Sie, in welchen der Zustände sich die Prozesse A und B von Teilaufgabe (b) vor dem beschriebenen Schedulingaufruf befinden und welche Übergänge sie im Rahmen der Teilaufgabe (b) vollführen. **3 P**