

1. Memory Management

Gegeben sei ein Computer mit 32 Bit virtuellem und physikalischem Adressraum, der eine einstufige Seitentabelle verwendet, bei der jede Speicherseite über eine 20 Bit Seitenadresse adressiert wird.

Beantworten Sie die folgenden Fragen:

- (1.1) (2 Punkte) Wie groß ist die Seitentabelle eines Prozesses, wenn in der Seitentabelle jeder Eintrag 32 Bit breit ist, also neben der 20-Bit physikalischen Seitenadresse noch zusätzliche Informationen, wie bspw. Zugriffsrecht enthält? Geben Sie bitte den Rechenweg an.

Lösung:

$$2^{20} \cdot 4\text{Byte} = 4\text{MB}$$

- (1.2) (1 Punkt) Wie groß ist eine Speicherseite in diesem System?

Lösung:

Da 12 Bit von den 32 Bit für das Offset übrigbleiben: $2^{12}\text{Byte} = 4\text{kB}$

- (1.3) (5 Punkte) Wofür könnten die zusätzlichen Bits jedes Eintrags in der Seitentabelle verwendet werden?

Lösung:

zum Beispiel:

- Zugriffsrechte (R, W, X)
- Seite gültig (Valid)
- Seite im Speicher (Present)
- Seite wurde beschrieben (Dirty)

Lösung:

3. In einem I-Node-basierten Dateisystem gelten die folgenden Bedingungen:

- Die Blockgröße in dem Dateisystem beträgt 2 kB.
- Jede I-Node speichere 12 direkte Links auf die ersten Blöcke der Datei, 2 einfach-indirekte Links und einen zweifach indirekten Link auf weitere Dateiblöcke.
- Blöcke, die für die indirekte Verlinkung weiterer Blöcke verwendet werden, speichern ausschließlich Links auf Dateiblöcke und keine weiteren Informationen.
- Jeder Link sei 8 Byte groß.

Beantworten Sie die folgenden Fragen:

(3.1) (3 Punkte) Wie groß kann eine Datei in diesem Dateisystem maximal werden?

Lösung:

- direkte Links: $12 \cdot 2kB = 24kB$
- einfach indirekte Links:
 - Anzahl der Links: $2 \cdot \frac{2kB}{8B} = 2 \cdot 256 = 512$
 - Menge des Speichers: $512 \cdot 2kB = 1MB$
- zweifach indirekter Link:
 - Anzahl der Links: $\frac{2kB}{8B} \cdot \frac{2kB}{8B} = \frac{4M}{8 \cdot 8}$
 - Menge des Speichers: $\frac{4M}{8 \cdot 8} \cdot 2kB = \frac{1GB}{8} = 128MB$

insgesamt: $128MB + 1MB + 24kB$, also etwa 129MB.

(3.2) (2 Punkte) Wie werden Dateien in einem tabellenbasierten Dateisystemen repräsentiert.

Lösung:

(3.3) (2 Punkte) Vergleichen Sie die Dateigrößenbeschränkungen mit tabellenbasierten Dateisystemen.

Lösung:

4. Gegeben sei ein Dateisystems aus 16 Blöcken mit bitmap-basierter Freispeicherverwaltung. deren Freispeicher-Bitmap den folgenden Wert hat:

0x4D87

Belegte Blöcke werden in der Bitmap mit 1 gekennzeichnet. Freie Blöcke werden mit 0 kodiert. Block 0 werde in der Bitmap im niedrigstwertigen Bit (LSB) kodiert.

- (4.1) (2 Punkte) Zeichnen Sie ein, welche Blöcke belegt sind. Die Zahlen entsprechen den Blocknummern (Block 0 ist also das erste Feld):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Lösung:

0x4D87 = (0100 1101 1000 0111)

niedrigstes Bit zuerst: 1110 0001 1011 0010

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
X	X	X					X	X		X	X			X	

- (4.2) (3 Punkte) Schreiben Sie in einer Programmiersprache Ihrer Wahl eine Funktion isEmpty(int bitmap), die möglichst effizient zurückgibt, ob alle 16 Blöcke des Dateisystems, deren 16-Bit Freispeicher-Bitmap als Parameter übergeben wird, frei sind.

Lösung:

```
boolean isEmpty(int bitmap){
    return bitmap == 0;
}
```

(4.3) (2 Punkte) Welche alternative Form der Freispeicherverwaltung haben Sie kennengelernt? Welches Verfahren ist effizienter bei der Suche nach einem freien Block und warum?

[illegible]

5.(5.1) (4 Punkte) Erklären Sie anhand eines minimalen C-Programms, welches den `fork()`-Systemaufruf verwendet, um einen Kindprozess zu erzeugen, was die Aufgabe und Funktionsweise des `fork()`-Systemaufrufs ist.

Lösung:

```

if (fork()){
    // Elternprozess
} else {
    // Kind
}

```

- `fork()` erzeugt zwei (fast) identische Prozesse durch Kopie des Elternprozesses
- Rückgabewert unterschiedlich
- im Kind = 0
- im Elternprozess = PID des Kindes

6. Zielstellungen des Prozess-Schedulings in Betriebssystemen sind in der Regel vom Einsatzzweck des Systems abhängig.

(6.1) (5 Punkte) Was sind die Unterschiede des präemptiven und des kooperativen Multitaskings. In welchen Systemen werden diese i. d. R. eingesetzt?

(6.2) (4 Punkte) Nennen und erklären Sie ein Schedulingverfahren, für jedes der beiden Arten des Multitaskings.

Lösung:

7. Ein Deadlock-Zustand kann auftreten, wenn in einer Menge von Prozessen jeder Prozess aus der Menge auf ein Ereignis wartet, das nur ein anderer Prozess aus der Menge auslösen kann.

(7.1) (3 Punkte) Berechnen Sie mit Hilfe des Bankier-Algorithmus', ob die im Folgenden dargestellte Zuteilung *sicher* ist:

Gegeben sei ein System mit 3 Prozessen A, B, C und einer Ressource, die **10** Mal im System verfügbar ist. Die folgende Tabelle gibt an, wie oft die Ressource von jedem Prozess aktuell verwendet wird und wie oft der Prozess sie maximal verwendet.

Nutzen Sie die Tabellen auf der folgenden Seite, um die einzelnen Schritte darzustellen, die Sie zur Berechnung der Sicherheit der Zuteilung benötigen. Markieren Sie den in jedem Schritt aktivierten Prozess.

	nutzt	max
A	3	5
B	2	4
C	1	5

frei:

	nutzt	max
A		
B		
C		

frei:

	nutzt	max
A		
B		
C		

frei:

	nutzt	max
A		
B		
C		

frei:

	nutzt	max
A		
B		
C		

frei:

	nutzt	max
A		
B		
C		

frei:

(7.2) (1 Punkt) Ist die Zuteilung sicher?

✓ **Ja**

☐ Nein

Lösung:

	nutzt	max
A	3	5
B	2	4
C	1	5

frei: 4	
---------	--

	nutzt	max
A	3	5
→ B	4	4
C	1	5

frei: 2	
---------	--

		nutzt	max
→	A	3	5
	B	0	-
	C	1	5

frej: 6

	nutzt	max								
A	5	5		A	0	-		A	0	-
B	0	-	→	B	0	-	→	B	0	-
C	1	5		C	1	5		C	5	5
frei: 4				frei: 9				frei: 5		
noch eine zweite Lösung, bei der Prozess A zuerst untersucht wird.										

Es gibt