Betriebssysteme

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Lösungen zur Aufgabensammlung

Prof. Dr. J. Wübbelmann /
Prof. Dr. G. Timmer
WS 2021/22

Hinweis:

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Erlaubte Hilfsmittel:

Ein **handgeschriebenes** DINA4-Blatt, keine weiteren Hilfsmittel.

Aufgabe 1

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Ein Prozesswechsel wird im User Mode gehandhabt?

Falsch

Aus einem Zyklus im Ressourcen-Belegungsgraphen folgt unmittelbar ein Deadlock?

Falsch

Aufgabe 1

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Welche Komponente wird zur Beschleunigung der Umrechnung zwischen Virtuellen und Physikalischen Adressen eingesetzt?

MMU mit Cache (TLB)

Aufgabe 2

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Drei Prozesse treffen zu den in der Tabelle angegebenen Zeiten in der Warteschlange eines Schedulers ein. Von jedem Prozess ist die Rechenzeit bekannt. Zusätzlich hat jeder Prozess eine Priorität (0 stellt die höchste Priorität dar).

Prozess	Ankunftszeit [100 ms]	Bedienzeit [100 ms]	Priorität
P_1	0	6	1
P_2	1	5	1
P_3	2	3	0

Aufgabe 2

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Die Prozesse benutzen nur die CPU und werden nie durch E/A oder andere Gründe blockiert. Der Scheduler entscheidet nur aufgrund der zum Scheduling-Zeitpunkt in der Warteschlange vorliegenden Prozesse.

Betrachten Sie die folgenden Scheduling-Strategien und zeichnen Sie für jede der Strategien den zeitlichen Ablauf der Prozessausführung als Balken-Diagramm (Gantt-Chart):

Aufgabe 2

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Shortest Job First (SJF, nicht-präemptiv).

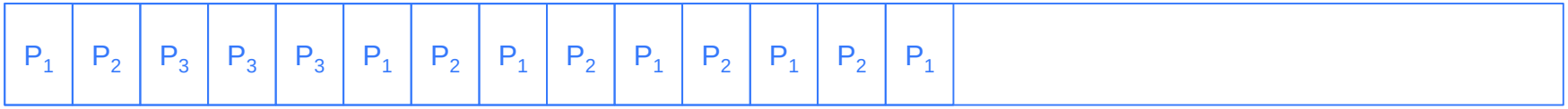
0 6 9

Round Robin (RR) mit einer Zeitscheibenlänge von 3 [x 100 ms]. Die Zeit für den Prozesswechsel kann vernachlässigt werden, die Zeitscheiben sind immer vollständig zu nutzen.

0 3 6 9 12 14

Aufgabe 2

Betrachten Sie nun eine prioritätsbasiertes Scheduling nach dem RR-Verfahren bei einer Zeitscheibenlänge von 1 [x 100 ms]. Zeichnen Sie den zeitlichen Ablauf der Prozessausführung als Balken-Diagramm sofern für jede Prioritätsklasse eine Warteschlange verwaltet wird. (Die Priorität 0 ist die "höchste".)



Aufgabe 3

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

In einem System gibt es 3 gleiche Drucker, auf denen per **`dump()`**-Befehl Dateien ausgegeben werden können. Allerdings müssen **`dump()`**-Aufrufe pro Drucker unter gegenseitigem Ausschluss stattfinden. Um nebenläufige Zugriffe zu synchronisieren, wird ein Semaphor ”**`printer`**“ eingeführt, auf den per **`down()`**-Operation und **`up()`**-Operation zugegriffen werden kann.

Aufgabe 3

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

```
class PrintService {
      semaphor printer(COUNTER);
      bool     is_free[COUNTER];
   public:
      PrintService(void);
      void print(string fname);
};

PrintService::PrintService(){
      for(int i=0; i< COUNTER; i++){
            is_free[i] = TRUE;
      }
}
```

mutex m;

Aufgabe 3

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

```
void PrintService::print(string fname){
        int d = -1;
        printer.down();
        while(! is_free[++d])
        ;
        is_free[d]=FALSE;
        dump(d, fname);
        is_free[d]=TRUE;
        printer.up();
}
```

Aufgabe 3

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Mit welchem Wert muss der Semaphor **`printer`** initialisiert werden?

Mit 3, da es drei gleiche Drucker gibt.

In der Lösung kann es zu Wettbewerbsbedingungen kommen. Was kann passieren?

Ein Thread wird in der **`print()`**-Methode direkt nach der **`while`**-Schleife unterbrochen. Dann ist **`is_free[d]`** immer noch **`TRUE`**.
Dies führt dazu, dass zwei Threads denselben Drucker auswählen. ↯

Aufgabe 3

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Ergänzen Sie die Lösung so, dass durch den Einsatz eines mutex-Objekts die Wettbewerbsbedingung vermieden werden kann.

Jeder Zugriff auf is_free[] sollte durch einen Mutex geschützt werden.

Aufgabe 3

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

```
class PrintService {
      mutex mprinter; …
}
void PrintService::print(string fname){
      int d = -1;
      printer.down();

      mprinter.lock();
      while(! is_free[++d])
      ;
      is_free[d]=FALSE;
      mprinter.unlock();
      dump(d, fname);
      mprinter.lock();
      is_free[d]=TRUE;
      mprinter.unlock();
      printer.up();
}
```

Aufgabe 4

In einem System laufen 4 Prozesse P_1 – P_4. Es existieren 4 Ressourcenklassen R_1-R_4 mit jeweils 4, 5, 3, 5 Instanzen.
Die zurzeit von den Prozessen belegten Ressourcen und die angeforderten Ressourcen können aus den folgenden Tabellen entnommen werden.

Tabelle 1: Belegte Ressourcen

	P1	P2	P3	P4
R1	0	2	0	0
R2	1	0	1	0
R3	1	2	0	0
R4	0	1	1	1

Aufgabe 4

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Tabelle 2: Angeforderte Ressourcen

	P1	P2	P3	P4
R1	2	1	0	1
R2	1	0	1	1
R3	1	0	1	0
R4	0	3	1	1

Liegt eine Verklemmung vor? Begründen Sie Ihre Entscheidung!

Aufgabe 4

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Ressourcenvektor

$$E = (4\ 5\ 3\ 5)$$

Belegungsmatrix

Ressourcen →

$$C = \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{matrix} \begin{pmatrix} 0 & 1 & 1 & 0 \\ 2 & 0 & 2 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

↓ Prozesse

Ressourcenrestvektor

$$A = (2\ 3\ 0\ 2)$$

Aufgabe 4

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Anforderungsmatrix

$$R = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 1 & 0 & 0 & 3 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

Algorithmus:

Suche einen Prozess, dass die Zeile in R ≤ A

$P_4 \Longrightarrow \quad A = (2\ 3\ 0\ 3)$

$P_2 \Longrightarrow \quad A = (4\ 3\ 2\ 4)$

$P_1 \Longrightarrow \quad A = (4\ 4\ 3\ 4)$

$P_3 \Longrightarrow \quad A = (4\ 5\ 3\ 5)$

$\Longrightarrow$ Keine Verklemmung

Aufgabe 5

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Ein Rechner soll einen 32 Bit Adressbus besitzen. Die Seitengröße des virtuellen Speichers betrage 16 KiByte. Sein physikalischer Speicher ist 512 MiByte groß. Ein Eintrag in der Seitentabelle ist 8 Byte groß.
Im System seien 4 Prozesse. Wie viel Prozent des physikalischen Speichers wird von den Seitentabellen der Prozesse belegt, wenn jeder Prozess seine vollständige Tabelle geladen hat?

Aufgabe 5

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Größe Page = 16KiByte = 2^{14} Byte
Virt. Adresse = 32 Bit = Bits Page-Index + 14 Bit Offset $\Longrightarrow$ 18 Bits für Page-Index $\Longrightarrow$ 2^{18} Pages

2^{18} Pages * 8 Byte/Eintrag = $2^{18} * 2^3$ Byte = 2 MiByte für eine Pagetable

4 Prozesse $\Longrightarrow$ insgesamt 8 MiByte für alle Pagetables

8/512 = 1/64 $\Longrightarrow$ ca. 1,5% des RAM werden für alle Pagetables benötigt

Aufgabe 6

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Die Seitentabelle enthält folgende Einträge: Seitenrahmen, Ladezeit der Seite, Zeit des letzten Zugriffs auf die Seite sowie R-, M- und Present- Bits. Die Seitentabelle sieht wie folgt aus:

Index (Seitennummer)	Seiten-rahmen	Lade zeit	Zugriffs -zeit	R	M	P
7	0	80	100	1	0	**0**
6	2	90	125	0	0	**0**
5	3	126	280	1	0	**1**
4	1	100	105	0	0	**0**
3	1	230	270	0	0	**1**
2	2	140	265	0	1	**1**
1	0	110	285	1	1	**1**
0	1	113	130	1	0	**0**

Aufgabe 6

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Welche Seiten sind zurzeit im Speicher und auf welche Seitenrahmen sind sie abgebildet?

Die Seiten mit P=1 sind im Hauptspeicher

- Seite 5: Seitenrahmen 3
- Seite 3: Seitenrahmen 1
- Seite 2: Seitenrahmen 2
- Seite 1: Seitenrahmen 0

Aufgabe 6

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Welche Seite wird bei einem Seitenfehler entfernt

Nur Seiten mit P=1 betrachten (P=0: ausgelagert)

- beim FIFO Algorithmus?
  Seite 1: am längsten geladen
- beim NRU-Algorithmus?
  Seite 3, da die einzige mit P=1, R=0, M=0
- beim Second-Chance Algorithmus?
  FIFO-Liste: 1-5-2-3 (Ladezeitpunkte beachten)
  R-Bit: 1-1-0-0 $\Longrightarrow$ Seite 2
- beim LRU-Algorithmus?
  Seite 2: die am längsten ungenutzte Seite (Zugriffszeitpunkt=265)

Aufgabe 7

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Für ein FAT 12 Dateisystem mit einer Clustergröße von 512 Byte ist folgender Auszug aus der Konfiguration gegeben:

Wurzelverzeichnis:

Name	Erweiterung	Flags	Erster Cluster	Größe/Byte
DAT1	DAT	HS	7	2500
DIR		Verzeichnis	3	
Weitere ...	Einträge ...			

Cluster 4:

Name	Erweiterung	Flags	Erster Cluster	Größe/Byte
PIC	1		10	1024
Frei File	TXT		5	1500
….				

File Allocation Table:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x	x	-1	4	-1	0 6	0 13	8	9	12	11	-1	17	0 -1	15	16	17	-1	-1	0	-1

Aufgabe 7

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

- In das Verzeichnis DIR wird eine Datei FILE.TXT kopiert, die 1500 Byte groß ist. Wie viel Platz belegt die Datei auf dem Datenträger?
- Clustergröße 512 Byte ⟹ 1500 Byte = 3 Cluster ⟹ 3*512 Byte = 1536 Byte

Wie sieht der Verzeichniseintrag für FILE.TXT aus?

Verzeichniseintrag in Cluster 4 (Cluster 3 voll):

Name	Erweiterung	Flags	Erster Cluster	Größe/Byte
PIC	1		10	1024
FILE	TXT		5	1500
….				

… und wie verändert sich die File Allocation Table?

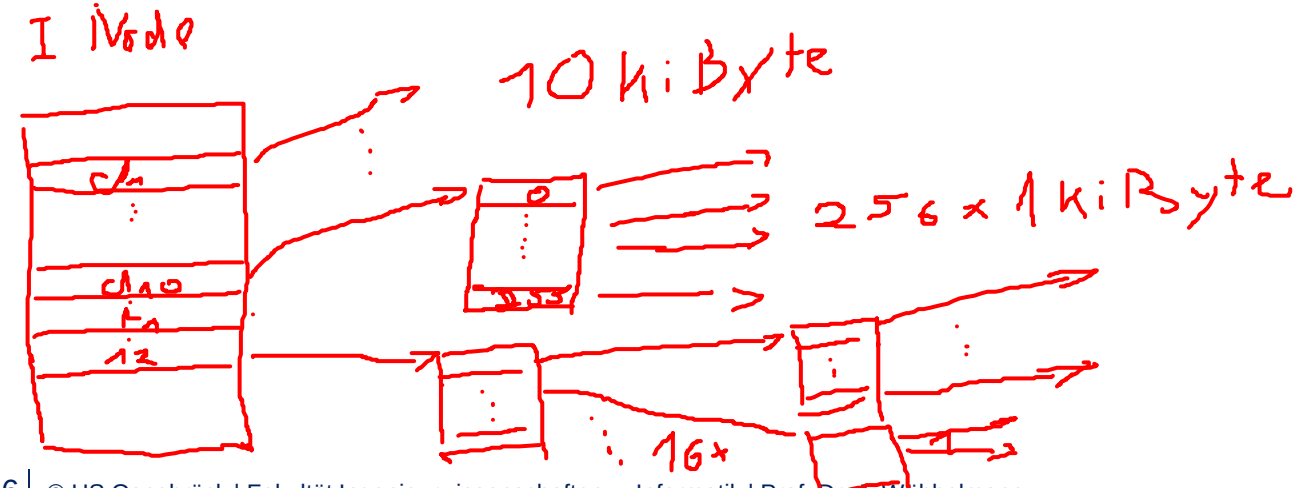
FAT:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x	x	-1	4	-1	6	13	8	9	12	11	-1	17	-1	15	16	17	-1	-1	0	-1

Aufgabe 8

Gegeben ist ein Unix Dateisystem mit einer Blockgröße von 1024 Byte. Blockadressen in Indirektionsblöcken haben eine Länge von 4 Byte.

Auf dieses Dateisystem soll eine Datei von 4208 KiByte gespeichert werden. Wie viel Platz belegt die Datei auf dem Datenträger?



Aufgabe 8

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Blockgröße: 1KiByte
Länge Blockadressen: 4 Byte ⟹ 256 Adr./Block

4208 KiByte = 4208 Blöcke
0. Indirektion = 10 Blöcke
1. Indirektion = 256 Blöcke ⟹ 4208-266 = 3942
2. Indirektion 3942/256 ≈ 16

Insgesamt: 4208 + 1 (1.Indirektion) + (1+16) (2.Indirektion) = 4226 Blöcke = ~~4266~~ 4226 KiByte