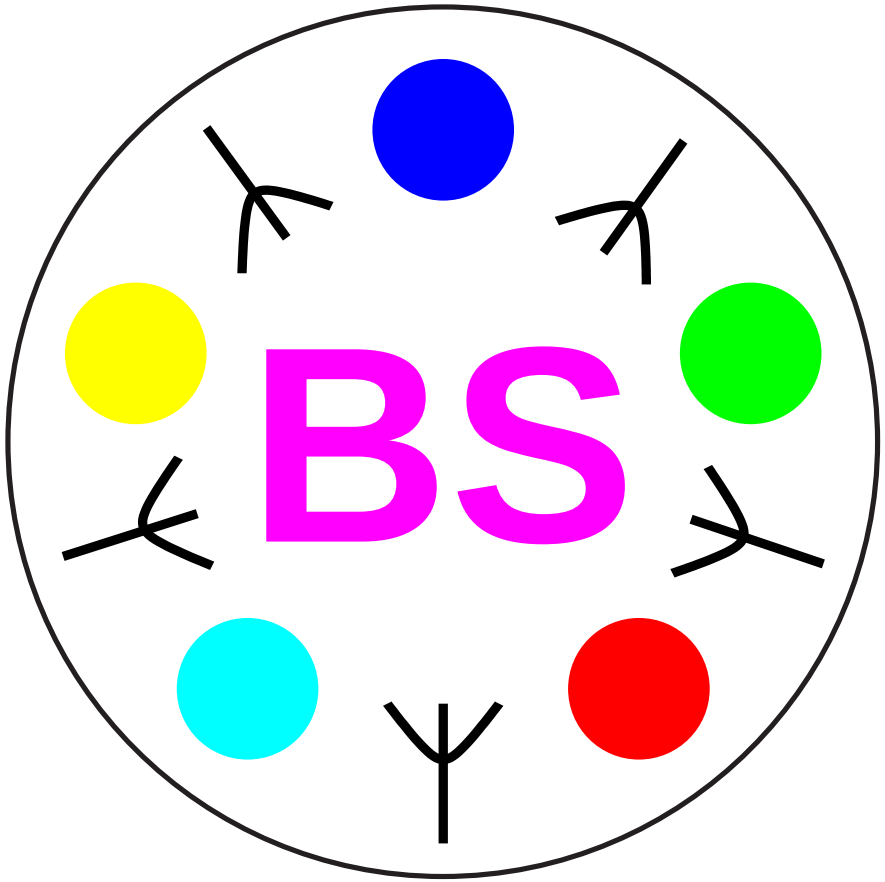
Rechnerarchitektur / Betriebssysteme



Übungen zur Vorlesung

Prof. Dr. Henning Pagnia

(AUF-1) HARDWARE UND INTERRUPTS

(1) Korrekt sind (a) und (c).

(b) ist falsch: System-Calls sind Software-Unterbrechungen.
(c) ist korrekt; die Annahme erfolgt durch den Prozessor, falls das IER-Register die Unterbrechung erlaubt.

(2) (a) ist falsch, da das Main-Memory der Hautpspeicher ist. Diesen als Cache-Speicher zu bauen, ist zu teuer.

(b) und (c) sind korrekt.

(3 (b) ist korrekt.

(a) ist falsch, denn das Stack-Pointer-Register (SP) enthält lediglich die Adresse des Stacks.

(c) ist falsch, denn der Code-Bereich wird beim Programmstart in der Größe festgelegt und wächst dann nicht mehr.

(4) (a) ist korrekt.

(b) ist falsch, denn der First-Level-Cache ist sehr schnell und klein und wird von jeder CPU exklusiv genutzt.

(c) ist falsch, denn bei Hyperthreading-CPUs werden nicht komplette Prozessoren sondern vielmehr nur einige Strukturen, wie z. B. die Register vervielfacht.

(AUF-2) SEMAPHORE: 2-AUS-3-PROBLEM

(a) Alle drei kritischen Abschnitten schließen sich selbst und jeweils gegenseitig aus.

(b) Im folgenden Szenario kommt es (bspw.) zu einer **Verklemmung**:

- Ein Prozess `A` führt `getDiskPrinter()` aus, und verliert nach `disk.p()` den Prozessor.
- Ein Prozess `B` führt `getTapeDisk()` aus, und verliert nach `tape.p()` den Prozessor.
- Ein Prozess `C` führt `getPrinterTape()` aus, und verliert nach `printer.p()` den Prozessor.
- `A` erhält erneut den Prozessor und führt `printer.p()` aus
  $\Rightarrow$ **warten**.
- `B` erhält erneut den Prozessor und führt `disk.p()` aus
  $\Rightarrow$ **warten**.
- `C` erhält erneut den Prozessor und führt `tape.p()` aus
  $\Rightarrow$ **warten**.

Nun liegt ein **zyklischer Wartezustand** vor. $\rightarrow$ **Verklemmung** (*Deadlock*)!

Lösungsansatz:

- Es werden Prioritäten definiert, d.h. die Betriebsmittel dürfen nur einer strengen **Reihenfolge** belegt werden.
  - ◇ **Beispiel:** `tape` vor `disk` vor `printer`.

(AUF-3) PROZESSVERWALTUNG

ausgeführte Schnittstellen-operation	Zustand der Prozessverwaltung						
	rp	ready queue	mutex ctr	mutex queue	resource ctr	resource queue	disk queue
	0	1,2,3	0	–	-1	4	5
mutex.p()	1	2,3	-1	0	-1	4	5
timerItrHandler()	2	3,1	-1	0	-1	4	5
diskItrHandler()	2	3,1,5	-1	0	-1	4	–
resource.v()	2	3,1,5,4	-1	0	0	–	–
writeDisk()	3	1,5,4	-1	0	0	–	2
timerItrHandler()	1	5,4,3	-1	0	0	–	2
resource.v()	1	5,4,3	-1	0	1	–	2
writeDisk()	5	4,3	-1	0	1	–	2,1
diskItrHandler()	5	4,3,2	-1	0	1	–	1
mutex.p()	4	3,2	-2	0,5	1	–	1

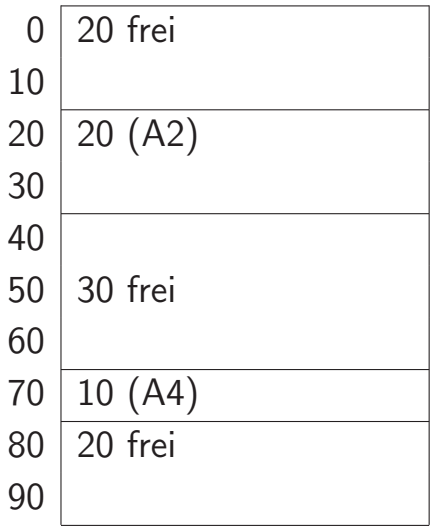
(AUF-4) SPEICHERVERWALTUNG

	Freispeicherliste mit Bereichslängen (1. Element links)							
	First-Fit				Best-Fit			
aktuelle Listen	90	35	60	45	90	35	60	45
Anforderung								
25	**65**	35	60	45	90	**10**	60	45
40	**25**	35	60	45	90	10	60	**5**
30	25	**5**	60	45	90	10	**30**	5
50	25	5	**10**	45	**40**	10	30	5
45	25	5	10		nicht erfüllbar !!!			

(AUF-5) ZUTEILUNG UND -FREIGABE

Es ergeben sich die nachfolgenden Endbelegungen:

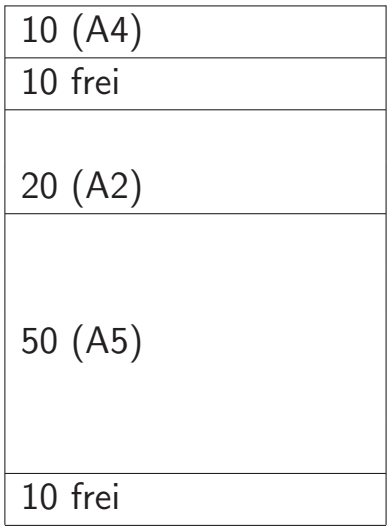
(a) First Fit:



free = 80

(Zustand vor A5,
da A5 nicht erfüllbar)

(b) Best Fit:



free = 90

(AUF-6) LINEARE SEITENTABELLE

Die Wortlänge ist 32 Bit $=$ 4 Byte.

Die Seitengröße ist 4 KiB $= 4 \cdot 2^{10}$ Byte.

Also haben 2^{10} Worte auf eine Seite Platz. Um dies zu adressieren benötigt man die letzten 10 Bit der virtuellen Adresse.

Es verbleiben 22 Bit der virtuellen Adresse zur Adressierung der Seiten. Damit lassen sich 2^{22} Seiten adressieren.

Bei einer Länge von 6 Byte pro Seitentabelleneintrag belegt eine lineare Seitentabelle demnach 24 MiB.

(AUF-7) SEITENTAUSCHSTRATEGIEN

a) Optimale Verdrängung

		Kacheln			
Referenz-reihenfolge	Seitenfehler	K1	K2	K3	am längsten nicht benötigte Seite
1	ja	**1**	-	-	1
5	ja	1	**5**	-	1
2	ja	1	5	**2**	1
4	ja	**4**	5	2	2
5	nein	4	5	2	2
3	ja	4	5	**3**	3
4	nein	4	5	3	3
5	nein	4	5	3	5
2	ja	4	**2**	3	3
4	nein	4	2	3	3
2	nein	4	2	3	2
1	ja	4	**1**	3	1
4	nein	4	1	3	1,4
3	nein	4	1	3	1,3,4

b) First-In-First-Out

		Kacheln			
Referenz-reihenfolge	Seitenfehler	K1	K2	K3	älteste Seite im Hauptspeicher
1	ja	**1**	-	-	1
5	ja	1	**5**	-	1
2	ja	1	5	**2**	1
4	ja	**4**	5	2	5
5	nein	4	5	2	5
3	ja	4	**3**	2	2
4	nein	4	3	2	2
5	ja	4	3	**5**	4
2	ja	**2**	3	5	3
4	ja	2	**4**	5	5
2	nein	2	4	5	5
1	ja	2	4	**1**	2
4	nein	2	4	1	2
3	ja	**3**	4	1	4

c) Least-Recently-Used

		Kacheln			
Referenz-reihenfolge	Seitenfehler	K1	K2	K3	am längsten nicht mehr referenzierte Hauptspeicherseite
1	ja	**1**	-	-	-
5	ja	1	**5**	-	1
2	ja	1	5	**2**	1
4	ja	**4**	5	2	5
5	nein	4	5	2	2
3	ja	4	5	**3**	4
4	nein	4	5	3	5
5	nein	4	5	3	3
2	ja	4	5	**2**	4
4	nein	4	5	2	5
2	nein	4	5	2	5
1	ja	4	**1**	2	4
4	nein	4	1	2	2
3	ja	4	1	**3**	1

d) Second-Chance-Algorithmus

		Kacheln												
Referenz-reihenfolge	Seiten-fehler	K1				K2				K3				Kachel-zeiger
			Ref	C	A		Ref	C	A		Ref	C	A	
		-	0	0	0	-	0	0	0	-	0	0	0	K1
1 r	ja	**1**	**1**	0	0	-	0	0	0	-	0	0	0	K1
5 w	ja	1	1	0	0	**5**	**1**	**1**	0	-	0	0	0	K1
2 w	ja	1	1	0	0	5	1	1	0	**2**	**1**	**1**	0	K1
4 r	ja	1	**0**	0	0									K2
						5	**0**	1	0					K3
										2	**0**	1	0	K1
		4	**1**	0	0	5	0	1	0	2	0	1	0	K2
5 w	nein	4	1	0	0	5	**1**	1	0	2	0	1	0	K2
3 r	ja					5	**0**	1	0					K3
										2	0	**0**	**1**	K1
		4	**0**	0	0									K2
						5	0	**0**	**1**					K3
		4	0	0	0	5	0	0	1	**3**	**1**	0	**0**	K1

		Kacheln												
Referenz-	Seiten-	K1				K2				K3				Kachel-
reihenfolge	fehler		Ref	C	A		Ref	C	A		Ref	C	A	zeiger
4 w	nein	4	**1**	**1**	0	5	0	0	1	3	1	0	0	K1
5 r	nein	4	1	1	0	5	**1**	0	1	3	1	0	0	K1
2 w	ja	4	**0**	1	0									K2
						5	**0**	0	1					K3
										3	**0**	0	0	K1
		4	0	**0**	**1**									K2
		4	0	0	1	**2**	**1**	**1**	**0**	3	0	0	0	K3
4 r	nein	4	**1**	0	1	2	1	1	0	3	0	0	0	K3
2 w	nein	4	1	0	1	2	1	1	0	3	0	0	0	K3
1 r	ja	4	1	0	1	2	1	1	0	**1**	**1**	0	0	K1
4 w	nein	4	1	**1**	1	2	1	1	0	1	1	0	0	K1
3 r	ja	4	**0**	1	1									K2
						2	**0**	1	0					K3
										1	**0**	0	0	K1
		4	0	**0**	1									K2
						2	0	**0**	**1**					K3
		4	0	0	1	2	0	0	1	**3**	**1**	0	0	K1

Ergebnisse

Strategie	Seitenfehler
Optimal	$3 + 4 = 7$
FIFO	$3 + 7 = 10$
LRU	$3 + 5 = 8$
Second-Chance	$3 + 5 = 8$

(AUF-8) SEITENFEHLER

Variante (i):
Bei dieser Variante erfolgt ein *zeilenweiser* Zugriff auf die (zeilenweise) abgespeicherte Matrix. Da eine Matrixseite komplett abgearbeitet wird, muss sie genau einmal eingelagert werden. Daher treten 64 Seitenfehler auf.

Variante (ii):
Die Zugriffsreihenfolge auf die Matrixelemente ist hier:

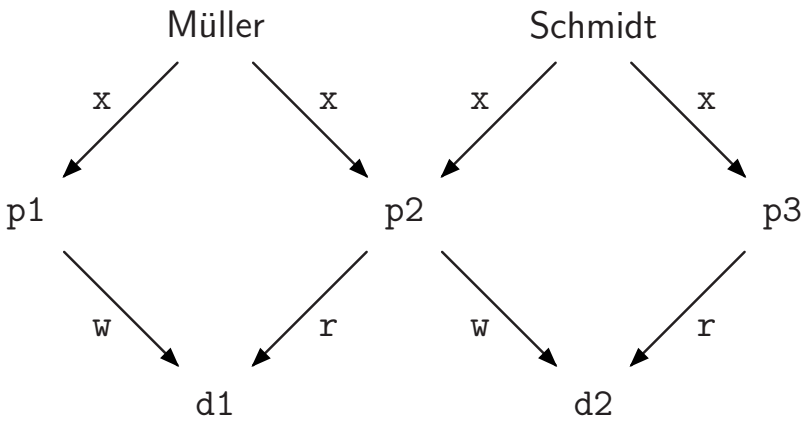
```
m[0,0]   m[1,0]   m[2,0]     ...  m[255,0]
m[0,1]   m[1,1]   m[2,1]     ...  m[255,1]
                  ...
m[0,255] m[1,255] m[2,255]   ...  m[255,255]
```

Jeweils nach der Bearbeitung von 4 Matrixelementen tritt ein Seitenfehler auf. Folglich erhält man $256 * 256/4 = 16384$ Seitenfehler.

(AUF-9) UNIX-DATEISCHUTZ

- Meier und Müller bilden eine Benutzergruppe.

Visualisierung der Zugriffe



Rechtevergabe mit minimalen Rechten

p1: Da Meier p1 besitzt, wird das x-Recht an die restlichen Gruppenmitglieder (= Müller) vergeben.
⇒ Müller erhält das benötigte Ausführungsrecht.

p2: Schmidt besitzt p2. ⇒ Er erhält das x-Recht als Besitzer.
Um auch Müller die Ausführung zu erlauben, muss das
x-Recht an die Allgemeinheit (= Meier und Müller) vergeben werden.
(1)

p3: Schmidt als Eigentümer des Programms erhält das x-Recht.

d1: Als Gruppenmitglied von Meier erhält Müller das r- und das w-Recht. ⇒ Er kann p1 bzw. p2 ausführen.
Die Allgemeinheit (= Schmidt) erhält das r-Recht.
⇒ Schmidt kann p2 ausführen.

d2: Schmidt als Besitzer von d2 erhält das r- und das w-Recht. ⇒ Er kann p2 bzw. p3 ausführen.
Die Allgemeinheit (= Meier und Müler) erhält das w-Recht
⇒ Müller kann p2 ausführen.
(2)

In der folgenden Tabelle sind wegen einer besseren Übersichtlichkeit nur **gewährte** Rechte eingetragen.

Resultierende Zugriffsrechte im Unix-Schema

	Eigentümer (u)			Gruppe (g)			Allgemeinheit (o)		
	r	w	x	r	w	x	r	w	x
	Meier			Müller			Schmidt		
p1						x			
	Schmidt						Meier, Müller		
p2			x						x
	Schmidt						Meier, Müller		
p3			x						
	Meier			Müller			Schmidt		
d1				r	w		r		
	Schmidt						Meier, Müller		
d2	r	w						w	

Zuviel vergebene Rechte

Im Vergleich zu den minimal notwendigen Zugriffsrechten mussten auf Grund des Unix-Schemas folgende Rechte zusätzlich vergeben werden:

(1) Meier erhält das Ausführungsrecht für `p2`.

(2) Meier erhält das Schreibrecht für `d2`.