

Betriebssysteme, Kapitel 4.3

1. Speichermanagement mit virtuellem Adressraum

Ihr Rechner hat einen 32-Bit byte-adressierbaren virtuellen Adressraum. Die Grösse der Seiten ist 4 KB. Wie viele Seiten virtuellen Speichers existieren?

$32\text{Bit} \Rightarrow 2^{32}$ Adressen, pro Adresse 1 Byte $\Rightarrow 2^{32}$ Bytes

$4\text{KB} = 2^{12} = 4096\text{Bytes}$

$2^{32} / 2^{12} = 2^{20} = 1'048'576$ Seiten

2. Speichermanagement mit Seitenfehler und Schutzattributen

Ein segmentierter Speicher hat seitenorientierte Segmente. Jede virtuelle Adresse enthält eine 2 Bit Segmentnummer, eine 2 Bit Seitennummer und einen 11 Bit Offset innerhalb der Seite. Der Hauptspeicher enthält 32 KB, aufgeteilt in 2 KB Seiten. Jedes Segment hat ein Attribut read-only, read-execute, readwrite oder read-write-execute. Wenn ein Schutzfehler passiert, spielt es keine Rolle, ob die Seitentabelle oder Seite sich im Speicher befindet. Die Seitentabelle und die Schutzattribute sind wie folgt gesetzt:

Segment 0		Segment 1		Segment 2		Segment 3	
read-only		read-execute		read-write-execute		read-write	
Virtuelle Seite	Seite	Virtuelle Seite	Seite	Virtuelle Seite	Seite	Virtuelle Seite	Seite
0	9	0	auf Disk	Seitentabelle nicht im Hauptspeicher		0	14
1	3	1	0			1	1
2	auf Disk	2	15			2	6
3	12	3	8			3	auf Disk

Es finden folgende Zugriffe auf den virtuellen Speicher statt:

Zugriff	Segment	Seite	Offset innerhalb Seite
1	fetch data	0	1
2	fetch data	1	1
3	fetch data	3	2047
4	store data	0	1
5	store data	3	1
6	store data	3	0
7	execute	1	3
8	fetch data	0	2
9	fetch data	2	0
10	execute	3	0

Geben Sie für die 10 Zugriffe jeweils an, auf welche physikalische Adresse zugegriffen wird, bzw. welcher Fehler auftritt.

fetch data -> mindestens lesen

store data -> Daten speichern

1) $3 * 2048 + 1 = 6'145$

*Segment 0, Seite 1 => physischer Speicher an Addr. 3, pro Addr 2kb => $3 * 2048 + \text{Offset } 1$*

Lesezugriff auf die physikalische Adresse 6'145.

2) $0 * 2048 + 10 = 10$

Lesezugriff auf die physikalische Adresse 10

3) $3, 3 \Rightarrow$ auf Disk => Page Error

Seitenfehler / Page Error, Seite nicht vorhanden

- 4) 0,1 => read only => Schutzfehler
Schutzfehler
- 5) $1 * 2048 + 2 = 2'050$
Schreibzugriff auf die physikalische Adresse 2'050
- 6) $14 * 2048 + 14 = 28'686$
Schreibzugriff auf die physikalische Adresse 28'686
- 7) $8 * 2048 + 100 = 16'484$
Ausführbarer Code aus physikalischer Adresse 16'484 lesen & ausführen
- 8) 0, 2 => auf Disk => Page Error
Seitenfehler / Page Error, Seite nicht vorhanden
- 9) 2, 0 nicht bekannt => Segmentfehler
Segmentfehler
- 10) 3, 0 nur read-write jedoch nicht execute => Schutzfehler
Schutzfehler

3. Speichermanagement mit Seitenfehler

Ein virtueller Speicher hat eine Seitengrösse von 1024 Worten, 8 virtuellen Seiten und 4 physikalischen Seiten. Gegeben ist die Seitentabelle:

Virtuelle Seite	Physikalische Seite
0	3
1	1
2	Nicht im Hauptspeicher
3	Nicht im Hauptspeicher
4	2
5	Nicht im Hauptspeicher
6	0
7	Nicht im Hauptspeicher

a) Listen Sie alle virtuellen Adressen auf, die einen Seitenfehler erzeugen.

Virtuelle Seiten 2, 3, 5 und 7

Seitengrösse: 1024

*Page 2: Start-Addr.: $2 * 1024 = 2048$ / End-Addr.: $3 * 1024 - 1 = 3071$*

*Page 3: Start-Addr.: $3 * 1024 = 3072$ / End-Addr.: $4 * 1024 - 1 = 4095$*

*Page 5: Start-Addr.: $5 * 1024 = 5120$ / End-Addr.: $6 * 1024 - 1 = 6143$*

*Page 7: Start-Addr.: $7 * 1024 = 7168$ / End-Addr.: $8 * 1024 - 1 = 8191$*

Bei folgenden Adressen gibt es einen Seitenfehler: 2048 – 4095, 5120 – 6143 und 7168 – 8191

b) Welche physikalischen Adressen entsprechen den virtuellen Adressen 0, 3728, 1023, 1024, 1025, 7800 und 4096?

Virt.	Phys.
0	3072
3728	Fehler
1023	4095
1024	1024
1025	1025
7800	Fehler
4096	2048

4. Interrupts

Beschreiben Sie in wenigen Sätzen den Unterschied zwischen Wichtigkeit und Dringlichkeit. Inwiefern hat das mit Interrupts etwas zu tun.

Wichtig: Wie wichtig die Ausführung ist, Zeitpunkt ist relativ

Dringlichkeit: Wie dringend etwas ausgeführt werden muss, je später umso schlechter

Interrupts: primär dringend! Wichtigkeit ist je nach Priorität unterschiedlich.

5. Interrupts und Semaphore

Ein Betriebssystem könnte Semaphore folgendermassen implementieren: Wenn die CPU eine P(s) oder V(s) Operation auf ein Semaphor ausführen möchte, setzt sie die Priorität so, dass alle Interrupts ausgeschaltet (disabled) werden. Dann erst holt sie das Semaphor, modifiziert es und gibt dann die Interrupts wieder frei.

Funktioniert diese Methode:

a) bei einer single core CPU, die alle 100 msec eine Prozessumschaltung vornimmt?

Ja

Prozessumschaltung stört uns nicht

b) bei einer double core CPU, die auf gemeinsamen Speicher, in dem das Semaphor liegt, zugreift?

Nein

es kann nur der Interrupt von einer CPU gesperrt werden und nicht über mehrere

6. Blockgrösse beim Dateisystem

Studien haben ergeben, dass die meisten Files nur wenige KB gross sind. Die überwiegende Mehrheit aller Files ist kleiner als 8 KB. Andererseits belegen die 10% grössten Files etwa 95% der ganzen Disk. Welche Schlüsse in Bezug auf die wünschbare Blockgrösse lassen sich daraus ableiten?

Grosse Blockgrösse: Schnelle Zugriffszeiten, ggf. „Nutzdaten-Menge Verlust“

Kleine Blockgrösse: Bessere Auslastung der Nutzgrösse, jedoch langsamer Zugriff

⇒ *Die Mehrheit aller Files ist kleiner als 8KB*

Es ist zu empfehlen das 8KB Blockgrösse verwendet wird.

Für die kleinen Dateien wird nicht so viel Platz „verschenkt“, und für die grossen Dateien ist der Zugriff noch +/- schnell.

Es gibt jedoch keine definitive Lösung, je nach Anforderung kann ggf. eine andere Blockgrösse besser sein. Z.B. schneller Zugriff auf die kleinen Dateien, vs. Schneller Zugriff auf die grossen Dateien und Speicherplatzverlust für die kleinen Dateien ist relativ.