

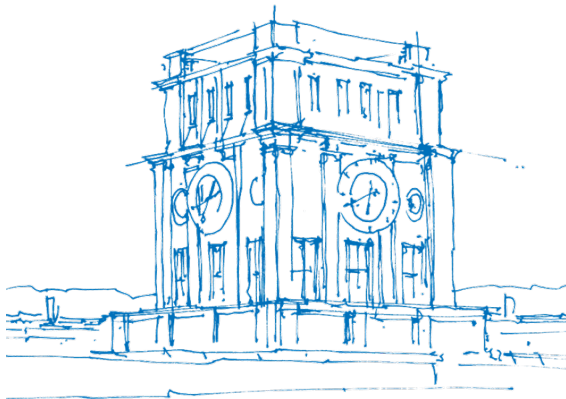
Grundlagen: Betriebssysteme und Systemsoftware

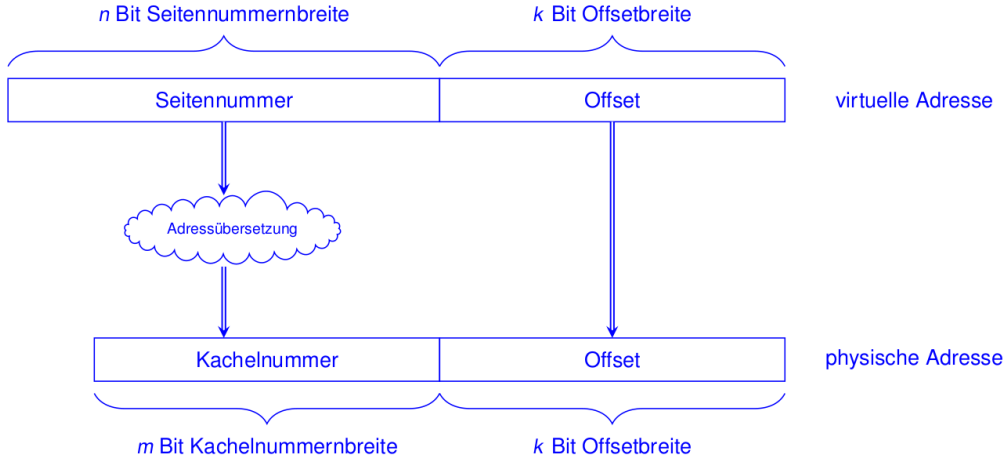
Tutorübung

Mario Delic

Lehrstuhl für Connected Mobility
School of Computation, Information and Technology
Technische Universität München

Übungswoche 10





Über- und Ausblick

• Offset-bits ermitteln:

Über Kachel-/Seitengröße:

4 KiB große Seite $\rightarrow 4 \text{ KiB} = 2^{12}$ Adressierbare **Bytes** $\rightarrow 12$ -bit Offset

Über bekannte Seitennummer-bits:

Adress-bits – Seitennummer-bits = Offset-bits $\rightarrow 24$ -bit Architektur, 13 Seitennummer-bits
 $\rightarrow 24 - 13 = 11$ Bits für Offset

• Seitennummer-bits ermitteln:

Über Anzahl an Seitentabelleneinträgen:

512 Seitentabelleneinträge $= 2^9$ Seitentabelleneinträge $\rightarrow 9$ Bits für Seitennummer

Über bekannte Offset-bits:

Adress-bits – Offset-bits = Seitennummer-bits $\rightarrow 24$ -bit Architektur, 13 Offset-bits $\rightarrow 24 - 13 = 11$ Bits für Seitennummer

$$O = 0 + \#s$$

- **Mehrstufige Seitentabellen:**

Einstufig: Seitennummer \rightarrow indizierte Seitentabelle mit Übersetzungen der v_a zu p_a

Zweistufig: Seitentabelle mit Seitennummern \rightarrow Seitentabelle mit Übersetzungen der v_a zu p_a

- **Anzahl an Blöcken ermitteln:**

Speichergröße \div Blockgröße \rightarrow 2 MiB Speicher, $8 \frac{\text{KiB}}{\text{Block}} : 2^{21} / 2^{13} = 2^8 = 256$ Blöcke

- **Blockgröße ermitteln:**

Speichergröße \div Blockzahl \rightarrow 8 MiB Speicher, 2048 Blöcke: $2^{23} / 2^{11} = 2^{12} = 4 \frac{\text{KiB}}{\text{Block}}$

- **Adressen pro Block (=Elemente pro Struktur):**

Strukturgröße \div Elementlänge \rightarrow Blockgröße $4 \frac{\text{KiB}}{\text{Block}}$, 64-bit Adresslänge $\rightarrow 2^{12} / 2^6 = 2^6$ Elemente

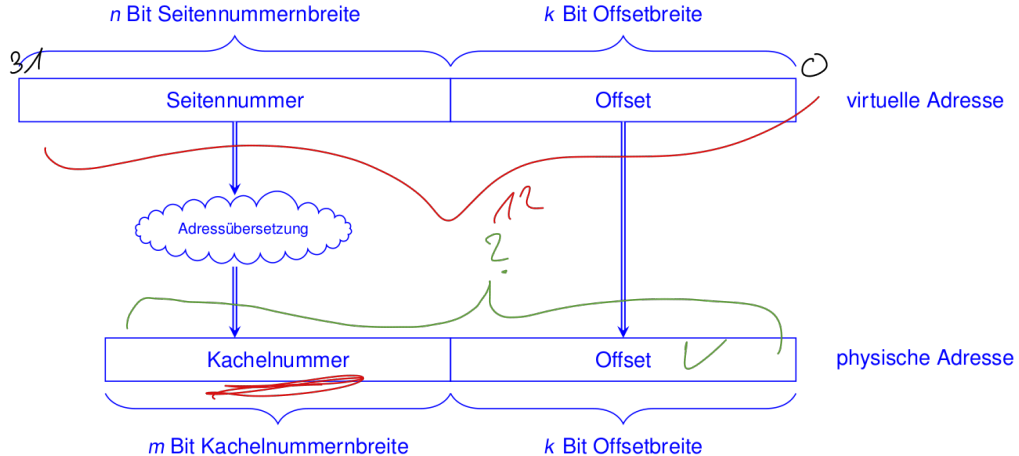
- **Indirect (=mehrstufige) Blöcke:**

Single-Indirect: Eine Adresse \rightarrow Block mit Adressen \rightarrow Datenblöcke

Double-Indirect: Eine Adresse \rightarrow Block mit Adressen \rightarrow Block mit Adressen \rightarrow Datenblöcke

Aufgabe 2

Virtuelle Adressierung



Aufgabe 3

Virtuelle Adressierung

- 12-bit virtuelle Adresse
- 256 Byte physischer Speicher $\rightarrow 2^8$
- 32 Byte Seitengröße (\rightarrow 32 Byte Kachelgröße!) $\rightarrow 2^5 \rightarrow 5 \text{ bits}$

a) Wie viele Bits der virtuellen Adresse entfallen auf die Seitennummer und Kachelnummer, wie viele auf den Offset? Wie viele Seiten können adressiert werden?

$$12 - 5 = 7$$

Aufgabe 2

Virtuelle Adressierung

- 12-bit virtuelle Adresse
 - 256 Byte physischer Speicher
 - 32 Byte Seitengröße (→ 32 Byte Kachelgröße!)
- a) Wie viele Bits der virtuellen Adresse entfallen auf die Seitennummer und Kachelnummer, wie viele auf den Offset? Wie viele Seiten können adressiert werden?

Bits Offset: Exponent der Seitengröße ($\log_2(\text{Seitengröße})$) → 32 Byte = 2^5 Bytes → 5 Offsetbits

- 12-bit virtuelle Adresse
- 256 Byte physischer Speicher
- 32 Byte Seitengröße (→ 32 Byte Kachelgröße!)

a) Wie viele Bits entfallen auf die Seitennummer und Kachelnummer, wie viele auf den Offset? Wie viele Seiten können adressiert werden?

Bits Offset: Exponent der Seitengröße ($\log_2(\text{Seitengröße})$) \rightarrow 32 Byte = 2^5 Bytes \rightarrow 5 Offsetbits

Bits Seitennummer: Adresslänge - Offsetbits \rightarrow 12-bit v.A., 2^5 Bytes Seitengröße

Adressierbare Seiten: $2^{\text{bits}} \rightarrow 2^{\text{v.A.}} = 2^7 = 128$

$256 \text{ B p.M} \rightarrow 2^8 = 256 \text{ B}$; $8 - 5 = 3$

$\log_2(2^8 / 2^5 = 2^3) = 3 \text{-bits}$

Aufgabe 2b

Virtuelle Adressierung

16 bit a; 0 = 12 bit
 $16 - 12 = 4 \rightarrow 2^4$ Permutationen

Seitengröße	Offsetbreite	#Einträge bei _ Bit virtueller Adressbreite:		
		16 Bit	32 Bit	64 Bit
4 KiB = 4096 Byte = 2^{12} Byte	12	2^4	2^{20}	2^{52}
8 KiB = 8192 Byte = 2^{13} Byte	13	2^3	2^{19}	2^{51}
16 KiB = 16384 Byte = 2^{14} Byte	14	2^2	2^{18}	2^{50}

Aufgabe 2c

Virtuelle Adressierung

32-Bit v.a

24-Bit p.a

Seitengröße	Offsetbreite physisch	Offsetbreite virtuell	Seitennummer	Kachelnummer
1 KiB = 1024 Byte = 2^{10} Byte	10	= 10	$32 - 10 = 22$	$24 - 10 = 14$
2 KiB = 2048 Byte = 2^{11} Byte	11	= 11	$32 - 11 = 21$	13
4 KiB = 4096 Byte = 2^{12} Byte	12	= 12	20	12
8 KiB = 8192 Byte = 2^{13} Byte	13	= 13	19	11

Aufgabe 3

$$\rightarrow 2^{16} / 2^3 = 2^{13} = \underline{8} \text{ KiB / Kachel / Seite}$$

Hauptspeichergröße: 64 KiB; 32 Seiten; 8 Kacheln.

16-bit p.c. ; 13 bits offset ; $2^5 \rightarrow 5$ bits

a) Wie lautet die höchste virtuelle Speicheradresse?

$$32 \cdot 8 \text{ KiB} = 2^5 \cdot 2^{13} = 2^{18} = 256 \text{ KiB}$$

$2^3 \rightarrow 3$ bit K. Nr.

4
:
1
64 KiB

b) Wie viele Bit sind für die virtuelle und physische Adresse jeweils breit?

Aufgabe 3

Hauptspeichergröße: 64 KiB; 32 Seiten; 8 Kacheln.

a) Wie lautet die höchste virtuelle Speicheradresse?

2^{16} Bytes / 2^3 Kacheln = 2^{13} Bytes pro Kachel $\rightarrow 2^{13}$ Bytes pro Seite

2^{13} Bytes pro Seite * 2^5 Seiten = 2^{18} Bytes (im virtueller Speicher)

\hookrightarrow Höchstes Offset ist immer Bytes-1, da man bei Byte 0x00 startet: $2^{18} - 1 = 262.144 - 1$
 $= 262.143$

b) Wie viele Bit sind für die virtuelle und physische Adresse jeweils breit?

Aufgabe 3

Hauptspeichergröße: 64 KiB; 32 Seiten; 8 Kacheln.

a) Wie lautet die höchste virtuelle Speicheradresse?

2^{16} Bytes / 2^3 Kacheln = 2^{13} Bytes pro Kachel $\rightarrow 2^{13}$ Bytes pro Seite

2^{13} Bytes pro Seite * 2^5 Seiten = 2^{18} Bytes (im virtueller Speicher)

\hookrightarrow Höchstes Offset ist immer Bytes-1, da man bei Byte 0x00 startet: $2^{18} - 1 = 262.144 - 1 = 262.143$

b) Wie viele Bit sind für die virtuelle und physische Adresse jeweils breit?

Physische Adresse: 64 KiB = 2^{16} adressierbare Bytes = 16 bit.

Alternativ: 2^3 Kacheln = 3 bits Kachel; 2^{13} Bytes pro Kachel/Seite = 13 bits Offset

\hookrightarrow 16 bit Adresse

Virtuelle Adresse: 2^5 Seiten = 3 bits Seite; 2^{13} Bytes pro Kachel/Seite = 13 bits Offset

\hookrightarrow 18 bit Adresse

Aufgabe 3

Hauptspeichergröße: 64 KiB; 32 Seiten (2^5); 8 Kacheln (2^3).

- c) Ermitteln Sie die jeweils angesprochene physische Adresse. Benutzen Sie die Pagetable aus Abbildung 1.

18-bit v.a. ; 13-bit offset
5-bit Seiten Nr.

16-bit p.a. / 13-bit offset
3-bit K. Nr.

Virtuelle Adresse	Seiten nummer	Offset	Kachel nummer	Physische Adresse
0x00559 = ... 0101 0101 1001	0x0	0x559	0x0	= 0x559
0x1208c = 01 601 0 0000 1000 1100	01 0001 = 0x9	0x8c	0x7	1110 0000 1000 1100 = 0xe08c = 0x8001
0x16001 = ...	0x5	0x1	0x4	
0x0a777 = 00 1010 01111 01111 01111	0x5	0x0777	PT	
0x13992 = 01 0011 1001 1001 0010	0x9	0x992	0x7	1111 1001 1001 0010 = 0xf992

Aufgabe 3

Hauptspeichergröße: 64 KiB; 32 Seiten (2^5); 8 Kacheln (2^3).

- d) Ermitteln Sie die jeweils angesprochene virtuelle Adresse. Benutzen Sie die Pagetable aus Abbildung 1.

Physische Adresse			Kachel nummer	Offset	Seiten nummer	Virtuelle Adresse
0x2000 =	0010	0...0	0x 1	0x 0	0x 1	00 0010 0000 0000 0000 = 0x 2000
0x8235 =	1000	0010 0011 0101	0x 4	0x 735	0x B	01 0110, 0x 235... = 0x 16235

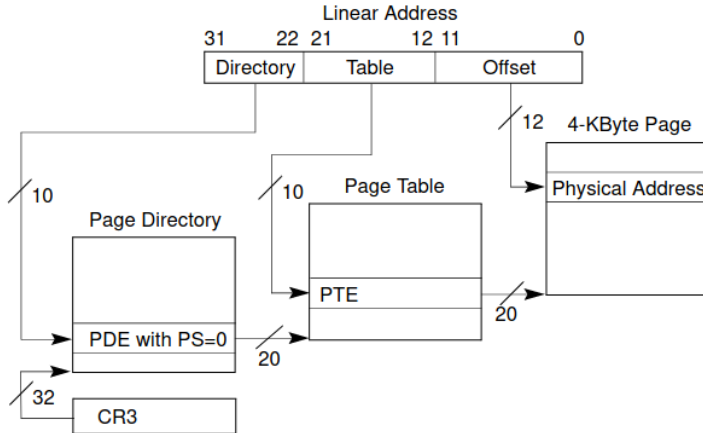


Figure 4-2. Linear-Address Translation to a 4-KByte Page using 32-Bit Paging

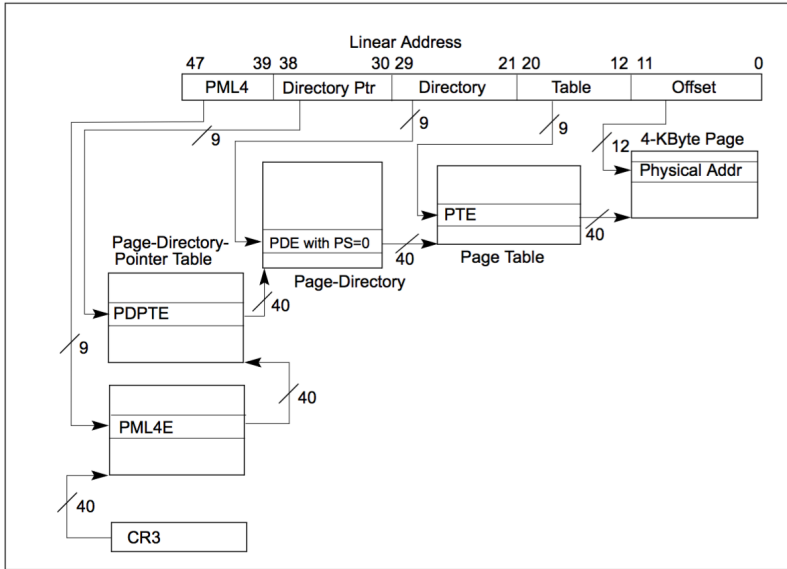


Figure 4-8. Linear-Address Translation to a 4-KByte Page using IA-32e Paging

Aufgabe 4

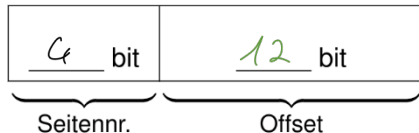
Hauptspeichergröße: 96 KiB; 4 KiB Seiten/Kacheln; 16-bit Architektur; einstufiges Paging.

a) Ergänzen Sie die Abbildung.

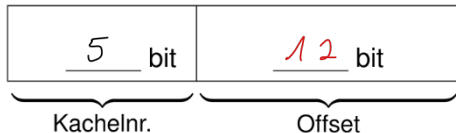
$$2^{16} < 96 \text{ KiB} < 2^{17}$$

$$1. \quad 4 \text{ KiB} = 2^{12} = 12 \text{ bit}$$

Virtuelle Adresse: 16 bit



Physische Adresse: 17 bit



Aufgabe 4

Hauptspeichergröße: 96 KiB; 4 KiB Seiten/Kacheln; 16-bit Architektur; einstufiges Paging.

b) Ergänzen Sie die Abbildung.

16 v.a. \rightarrow 4 SNr
12-bit offset

17 p.a. \rightarrow 5 KNr

Virtuelle Adresse	Seitennummer	Kachelnummer	Physische Adresse
$v_1 = 0x7331$ einfach, da offset vielfaches von 4	0x 7	0x 5	0x 5331
$v_2 = 0x4242$	0x 4	0x C	0x C242

Seitennr.	abgebildet in Kachel:
0x00	
0x01	0x03
0x02	0x04
0x03	0x08
0x04	0x0c
0x05	
0x06	
0x07	0x05
0x08	0x01
0x09	0x06
0x0a	
0x0b	
0x0c	0x00
0x0d	0x02

Aufgabe 4

Hauptspeichergröße: 96 KiB; 4 KiB Seiten/Kacheln; 16-bit Architektur; einstufiges Paging.

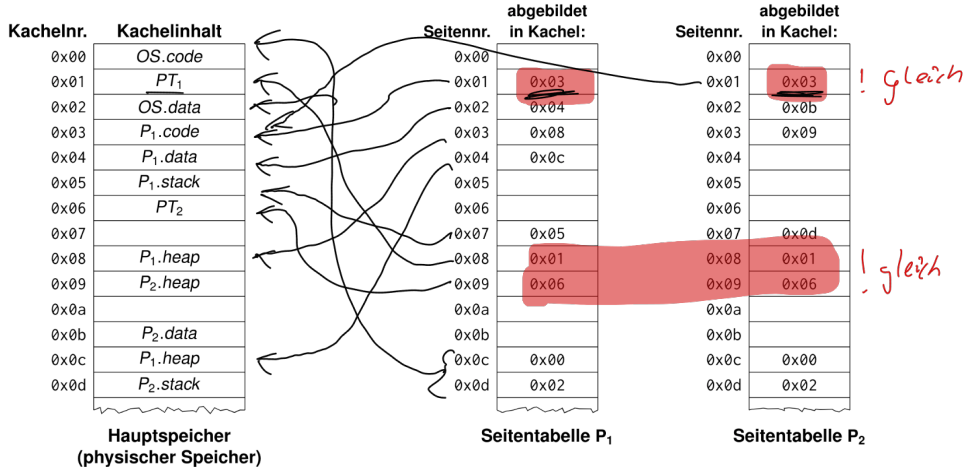
c) Ergänzen Sie die Abbildung.

Virtuelle Adresse	Seitennummer	Kachelnummer	Physische Adresse
$v_1 = 0x7331$	0x <u>7</u>	0x <u>d</u>	0x <u>d331</u>
$v_2 = 0x4242$	0x <u>4</u>	0x <u>PF</u>	0x <u>PF</u>

Seitennr.	abgebildet in Kachel:
0x00	
0x01	0x03
0x02	0x0b
0x03	0x09
0x04	
0x05	
0x06	
0x07	0x0d
0x08	0x01
0x09	0x06
0x0a	
0x0b	
0x0c	0x00
0x0d	0x02

Aufgabe 4

d) Verbinden Sie die Seiten mit den jeweiligen Inhalten.



Aufgabe 4

e) Bestimmen Sie die niedrigste und höchste heap-Adresse von P1.

heap = Seite $0x03$ & $0x04$

↓
 $0x3 \cdot 0x1000 = 0x3000$
(2^{12})

→ $0x5 \cdot 0x1000 - 1 = 0x5000 - 1$
 $= 0x4fff$

Aufgabe 4

e) Bestimmen Sie die niedrigste und höchste heap-Adresse von P1. oder umständlich:

Die erste Seite 0x00 scheint ungenutzt (Byte 0-4095); eine 4 KiB Seite Code (^{bzw. nicht gemappt} $2^{12} = 4096$ Byte); eine 4 KiB Seite Data (weitere 4096 Byte = 8192 Byte). Byte 4096-12287 entfallen auf Code+Data → Heap beginnt bei Seite 3, also bei Byte Nr. 12288.

In Hex:

$$12288 = 2^{13} + 2^{12}$$

1. Koeffizienten: $2^{13 \bmod 4} = 2^1 = \underline{2}$; $2^{12 \bmod 4} = 2^0 = \underline{1} \rightarrow \underline{2 + 1} = 3$

2. Anzahl an Nullen: $13 : 4 = \underline{3}$; $12 : 4 = \underline{3}$; d.h. drei Nullen jeweils. → 2000 + 1000 = 3000
→ 12288 in Hex = **0x3000**.

Der Heap umfasst 2 Seiten, die höchste Adresse ist 1 Byte vor $12288 + 8192 - 1 = 20479$.

Hex: $20480 = 2^{14} + 2^{12} \rightarrow 2^{14 \bmod 4} = 4$; $2^{12 \bmod 4} = 1 \rightarrow 14 : 4 = 3$; $12 : 4 = 3 \rightarrow 4000 + 1000 = 5000 \rightarrow 0x5000 - 0x1 = \text{0x4fff}$

Aufgabe 4

- f) Welche Inhalte aus dem physischem Speicher sind in beiden virtuellen Adressräumen gemapped?

Aufgabe 4

- f) Welche Inhalte aus dem physischem Speicher sind in beiden virtuellen Adressräumen gemapped?

Seitentabellen, Betriebssystemdaten und das Code-Segment von P1 und P2.

- g) Können die Prozesse P1 und P2 Instanz des gleichen Programmes sein? Begründung!

Aufgabe 4

- f) Welche Inhalte aus dem physischem Speicher sind in beiden virtuellen Adressräumen gemapped?

Seitentabellen, Betriebssystemdaten und das Code-Segment von P1 und P2.

- g) Können die Prozesse P1 und P2 Instanz des gleichen Programmes sein? Begründung!

Ja, weil beide Prozesse das gleiche Code-Segment verwenden.

Aufgabe 4

- h) Wo genau ist die Seitentabelle von P2 in Abbildung 3 gespeichert wenn der Prozess rechnet?

Aufgabe 4

h) Wo genau ist die Seitentabelle von P2 in Abbildung 3 gespeichert wenn der Prozess rechnet?

Die Seitentabelle befindet sich im Hauptspeicher (hier Kachel 0x06). Im dem sog. Seitentabellenregister wird die Adresse der Pagetable für den aktuell rechnenden Prozess gespeichert. Bei einem Kontextwechseln wird dieses Register also immer mit der physischen Adresse der Seitentabelle des aktuellen Prozesses gefüllt.

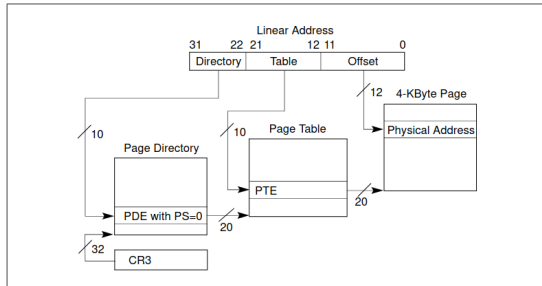


Figure 4-2. Linear-Address Translation to a 4-KByte Page using 32-Bit Paging