



Technische
Universität
Braunschweig

Institut für Betriebssysteme
und Rechnerverbund



Betriebssysteme – Übung

10T: Speicherverwaltung und virtueller Speicher

Signe Rüscher, Wintersemester 2019/2020

Übersicht

- **10.1 Speicherverwaltung**
- 10.2 Virtueller Speicher
- 10.3 Ersetzungsstrategien
- 10.4 Scheduling-Verfahren für Schreib-/Lesekopf

Speicherverwaltung

First Fit

- Block wird in *ersten* freien Bereich eingelagert, in den er passt
- Für jede Anfrage Start der Überprüfung bei Adresse 0

Next Fit

- Block wird in *nächsten* freien Bereich eingelagert, in den er passt
- Für jede Anfrage Start der Überprüfung nach dem zuletzt eingelagerten Block

Speicherverwaltung

Best Fit

- Block wird in den freien Bereich eingelagert, in den er am *besten* passt
- Suche nach der kleinsten passenden Lücke

Worst Fit

- Block wird in freien Bereich eingelagert, in den er am *schlechtesten* passt
- Suche nach der größten passenden Lücke

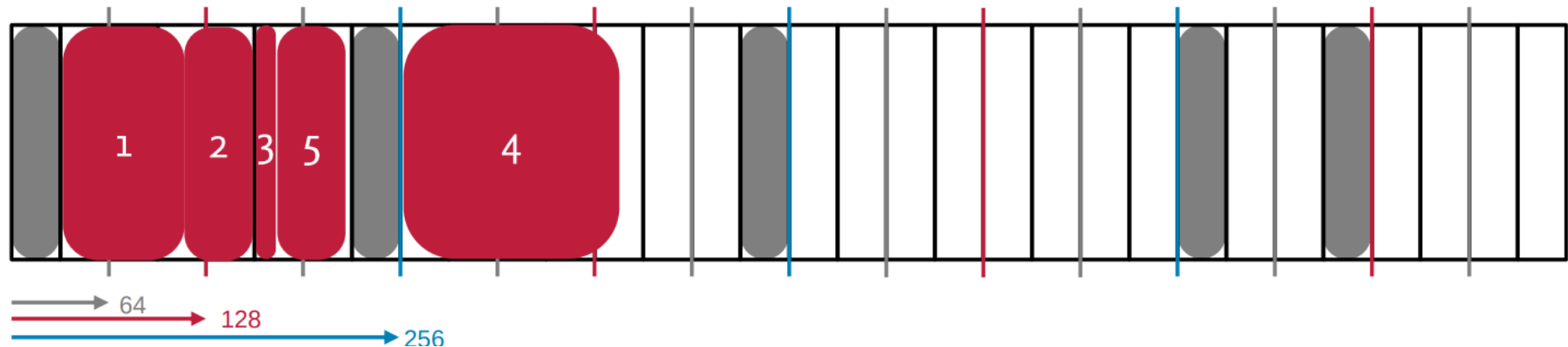
Speicherverwaltung

Buddy

- Größe des Speicherbereichs: 2^K
- Zu allozierender Block wird auf nächsthöhere Zweierpotenz (2^X) aufgerundet, mindestens 32 Byte
- Verfügbarer Speicherbereich wird so lange in gleich große Blöcke halbiert, bis ein Block der Größe 2^X zur Verfügung steht
- Halbieren/Überprüfen immer erst in Richtung der kleineren Adressen
 - z.B. 1024B = 2 * 512B : erst linken Block weiter halbieren
- Wenn passender Block gefunden, wird gesamter Speicher alloziert
- Kleinstmögliche Blockgröße ist 32 Byte

First-Fit

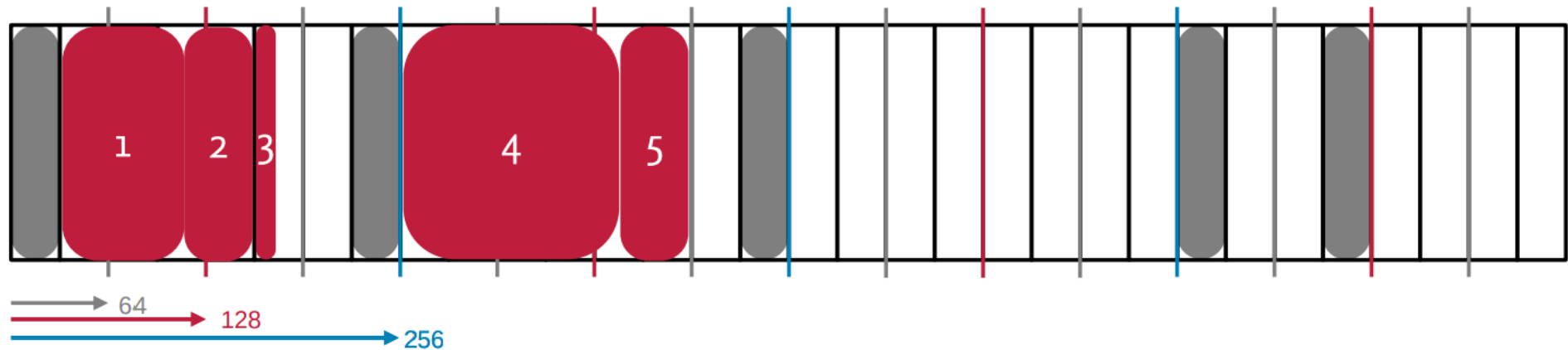
a) Tragen Sie nach dem **First-Fit**-Verfahren den Anfang des zugewiesenen Speichers ein und skizzieren Sie die Belegung.



Aufruf	First-Fit
1. malloc(88)	32
2. malloc(48)	120
3. malloc(12)	168
4. malloc(144)	256
5. malloc(40)	180

Next-Fit

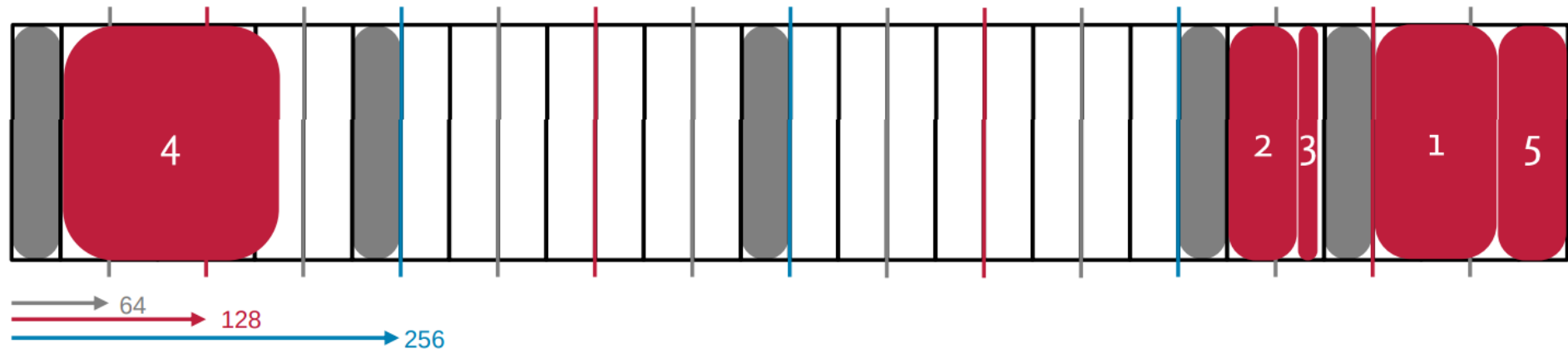
b) Tragen Sie nach dem **Next-Fit**-Verfahren den Anfang des zugewiesenen Speichers ein und skizzieren Sie die Belegung.



Aufruf	Next-Fit
1. malloc(88)	32
2. malloc(48)	120
3. malloc(12)	168
4. malloc(144)	256
5. malloc(40)	400

Best-Fit

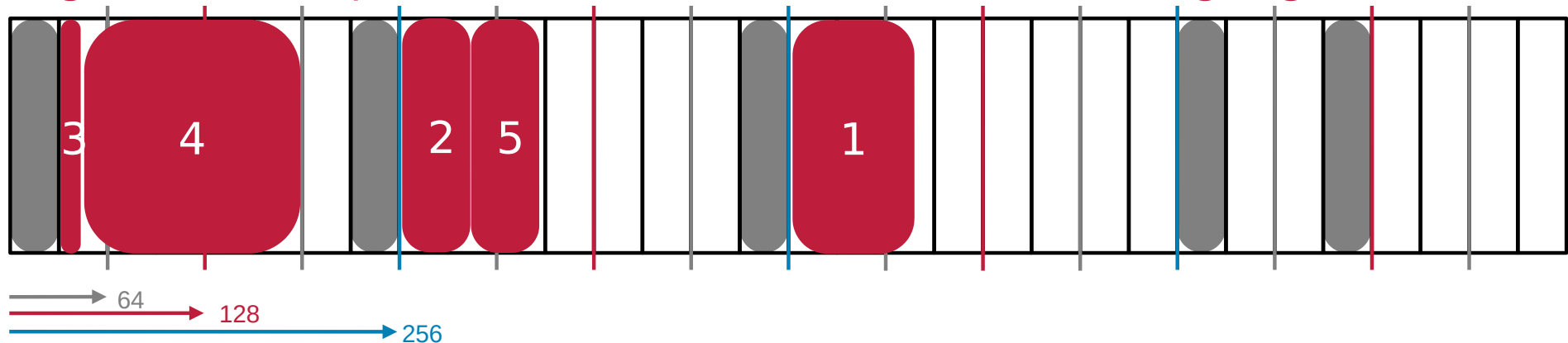
c) Tragen Sie nach dem **Best-Fit**-Verfahren den Anfang des zugewiesenen Speichers ein und skizzieren Sie die Belegung.



Aufruf	Best-Fit
1. malloc(88)	896
2. malloc(48)	800
3. malloc(12)	848
4. malloc(144)	32
5. malloc(40)	984

Worst-Fit

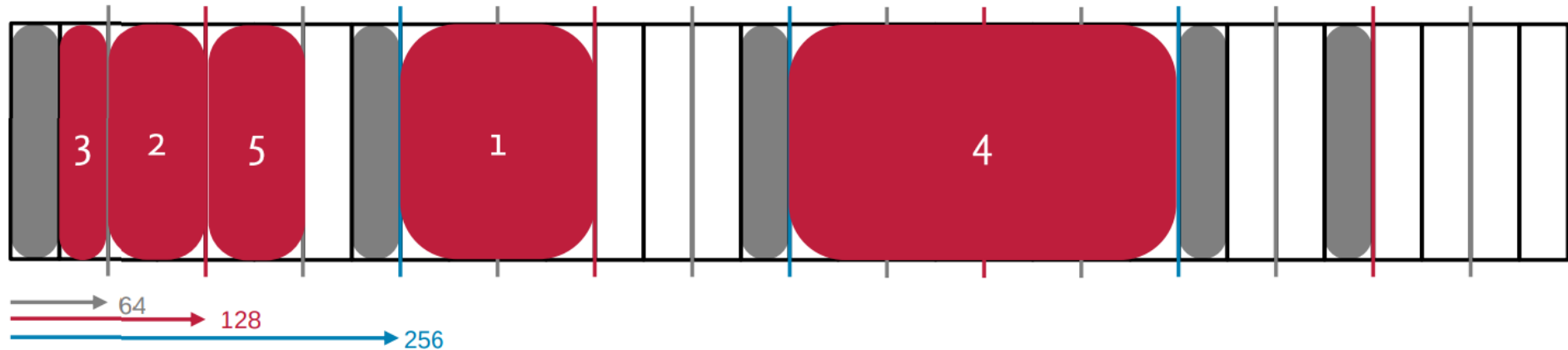
d) Tragen Sie nach dem **Worst-Fit**-Verfahren den Anfang des zugewiesenen Speichers ein und skizzieren Sie die Belegung.



Aufruf	Worst-Fit
1. malloc(88)	512
2. malloc(48)	256
3. malloc(12)	32
4. malloc(144)	44
5. malloc(40)	304

Buddy

e) Tragen Sie nach dem **Buddy**-Verfahren den Anfang des zugewiesenen Speichers ein und skizzieren Sie die Belegung.



Aufruf	Alloziert	Buddy
1. malloc(88)	128	256
2. malloc(48)	64	64
3. malloc(12)	32	32
4. malloc(144)	256	512
5. malloc(40)	64	128

10.2 a) Virtueller Speicher

Erklären Sie die Idee hinter Kachelverwaltung (Paging).

10.2 a) Virtueller Speicher

Erklären Sie die Idee hinter Kachelverwaltung (Paging).

- Ermöglicht virtuelle Speicherverwaltung: Prozess hat 'unbegrenzten' virtuellen Speicher
- Verwendung von nicht zusammenhängendem Speicher
- Auslagerung des Hauptspeichers in den Hintergrundspeicher
- Vermeidet externe Fragmentierung
- Gemeinsame Verwendung von Speicherbereichen zwischen mehreren Prozessen

Virtueller Speicher: Anmerkungen

Realisierung

Traditionell:

- Adressumsetzung in HW realisiert

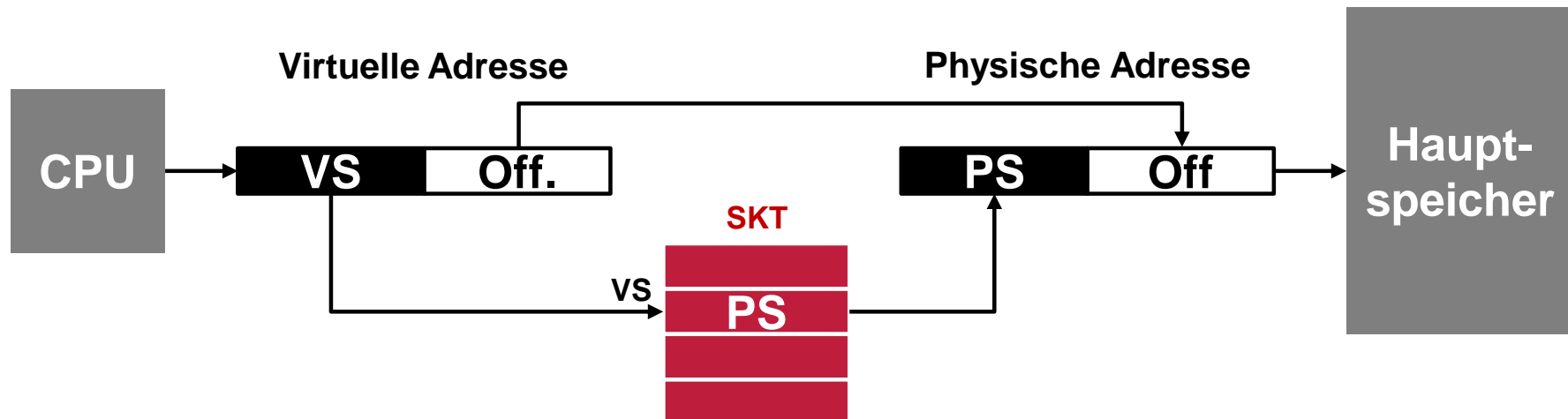
Heute:

- Hardware-gestützte Implementierung im Betriebssystem
- SKT liegt in einem definierten Bereich des Hauptspeichers

Virtueller Speicher: Anmerkungen

Prinzip

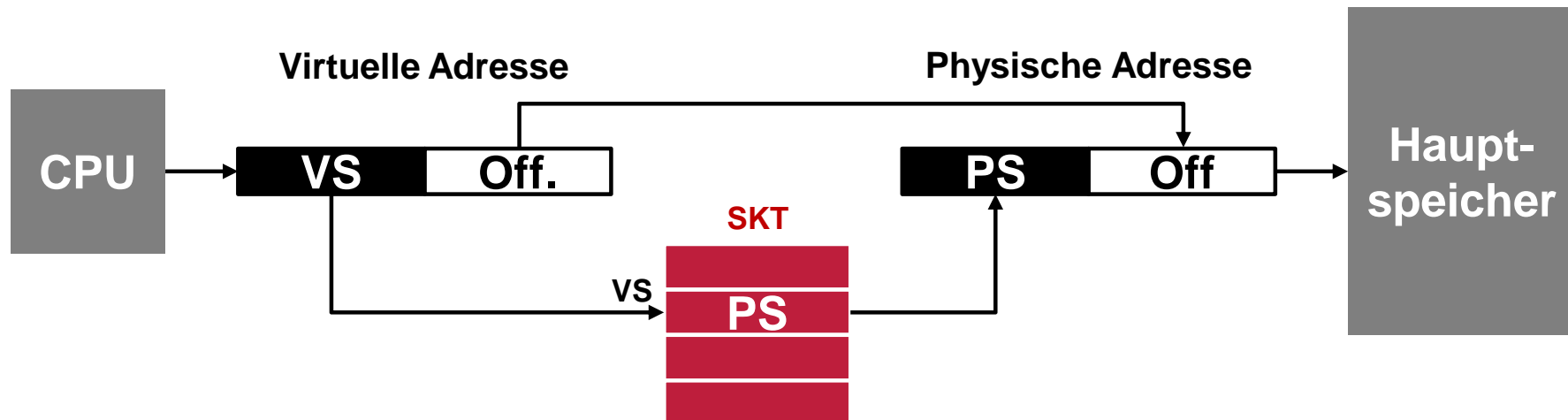
- Von der CPU generierte (virtuelle) Adressen teilen sich auf in:
 - **Virtuelle Seitennr. (VS)** – Adressiert die Seiten-Kachel-Tabelle (SKT)
 - **Seiten-Offset (Off)** – Adressiert Einheiten innerhalb von Seiten



Virtueller Speicher: Anmerkungen

Prinzip

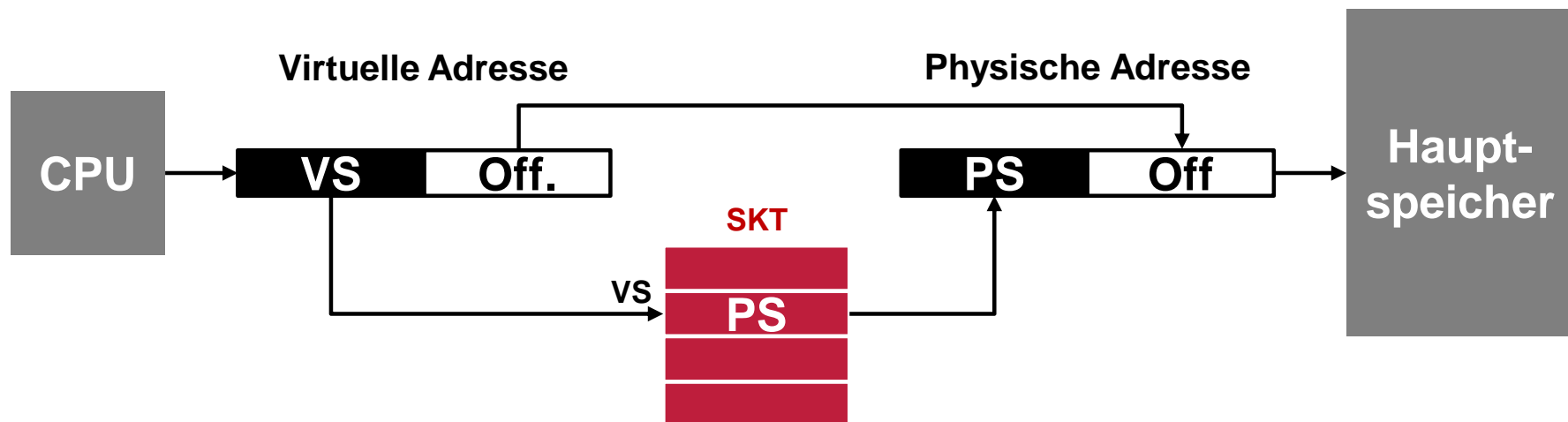
- Von der CPU generierte (virtuelle) Adressen teilen sich auf in:
 - **Virtuelle Seitennr. (VS)** – Adressiert die Seiten-Kachel-Tabelle (SKT)
 - **Seiten-Offset (Off)** – Adressiert Einheiten innerhalb von Seiten
- Hauptspeicher adressiert durch:
 - **Physische Seitennr. (PS)** – Adressiert physische Seite
 - Seiten-Offset (Off)



Virtueller Speicher: Anmerkungen

Prinzip

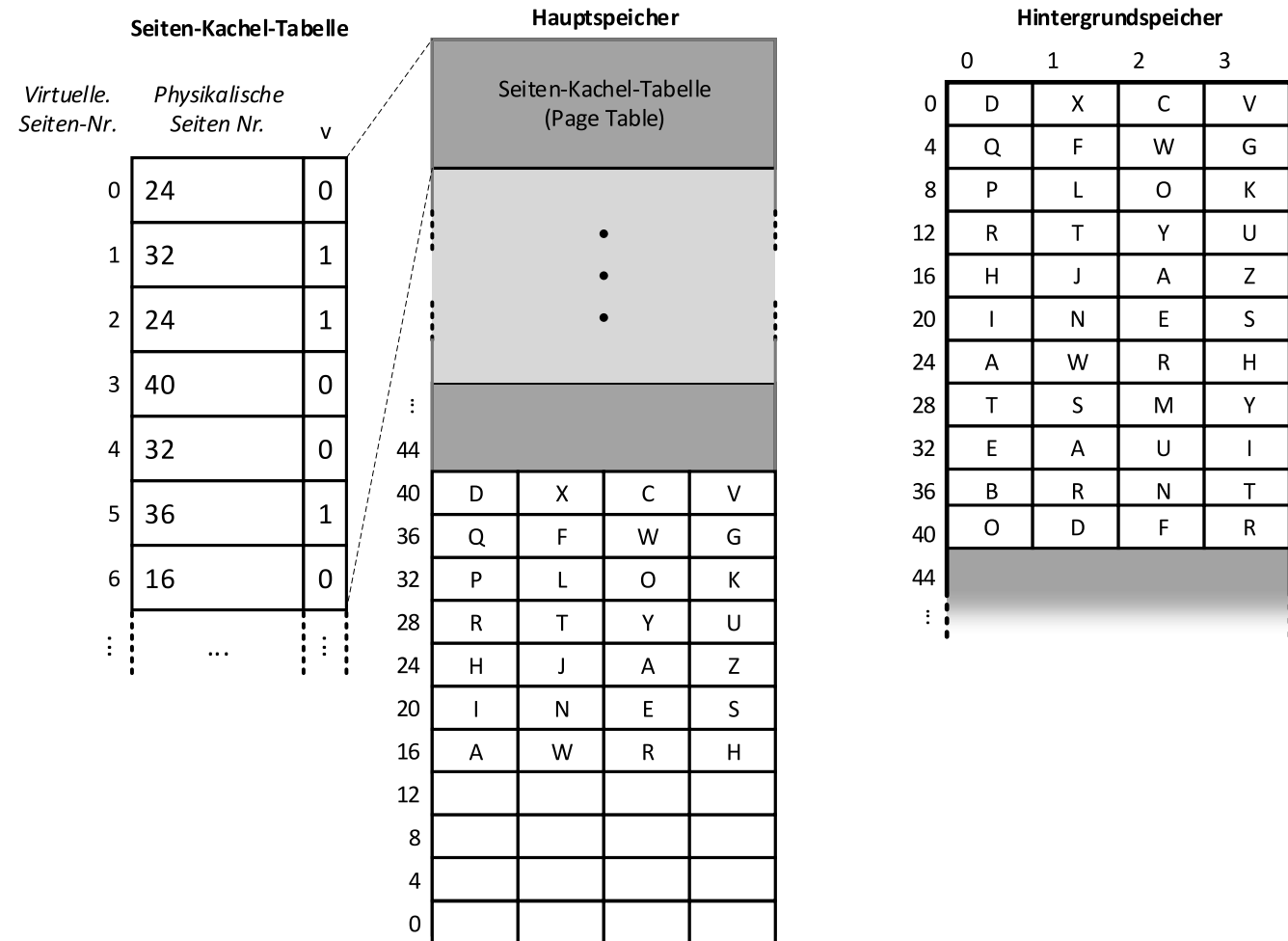
- Seiten-Kachel-Tabelle (SKT):
 - enthält die **physische Seitennummer (PS)** sowie verschiedene Flags
 - Valid Bit (v, Präsenzbit): gibt an, ob Seite im Hauptspeicher vorhanden ist
 - übersetzt die virtuelle Seitennummer in physische Seitennummer



10.2 b) Virtueller Speicher

Bestimmen Sie:

- Inhalt der gelesenen Daten
- Seitenfehler
- eingelagerte Seiten



10.2 b) Virtueller Speicher

Seiten-Nr.	Offset	Seitenfehler	Gelesen
1	0		
2	3		
3	1		
4	2		
3	3		
4	3		

Seiten-Kachel-Tabelle		
Virtuelle. Seiten-Nr.	Physikalische Seiten Nr.	v
0	24	0
1	32	1
2	24	1
3	40	0
4	32	0
5	36	1
6	16	0
⋮	...	⋮

Hauptspeicher

Seiten-Kachel-Tabelle (Page Table)				
		•		
		•		
		•		
⋮				
44				
40	D	X	C	V
36	Q	F	W	G
32	P	L	O	K
28	R	T	Y	U
24	H	J	A	Z
20	I	N	E	S
16	A	W	R	H
12				
8				
4				
0				

	0	1	2	3
0	D	X	C	V
4	Q	F	W	G
8	P	L	O	K
12	R	T	Y	U
16	H	J	A	Z
20	I	N	E	S
24	A	W	R	H
28	T	S	M	Y
32	E	A	U	I
36	B	R	N	T
40	O	D	F	R
44				
:				

10.2 b) Virtueller Speicher

Seiten-Nr.	Offset	Seitenfehler	Gelesen
1	0	nein	P
2	3		
3	1		
4	2		
3	3		
4	3		

Seiten-Kachel-Tabelle

Virtuelle. Seiten-Nr.	Physikalische Seiten Nr.	v
0	24	0
1	32	1
2	24	1
3	40	0
4	32	0
5	36	1
6	16	0
...

Hauptspeicher

Seiten-Kachel-Tabelle (Page Table)			
•			
•			
•			
:			
44			
40			
D	X	C	V
36	Q	F	G
32	P	L	O
28	R	T	Y
24	H	J	A
20	I	N	E
16	A	W	R
12			
8			
4			
0			

Hintergrundspeicher

	0	1	2	3
0	D	X	C	V
4	Q	F	W	G
8	P	L	O	K
12	R	T	Y	U
16	H	J	A	Z
20	I	N	E	S
24	A	W	R	H
28	T	S	M	Y
32	E	A	U	I
36	B	R	N	T
40	O	D	F	R
44				
:				

10.2 b) Virtueller Speicher

Seiten-Nr.	Offset	Seitenfehler	Gelesen
1	0	nein	P
2	3	nein	Z
3	1		
4	2		
3	3		
4	3		

Seiten-Kachel-Tabelle		
Virtuelle. Seiten-Nr.	Physikalische Seiten Nr.	v
0	24	0
1	32	1
2	24	1
3	40	0
4	32	0
5	36	1
6	16	0
⋮	...	⋮

Hauptspeicher

Seiten-Kachel-Tabelle
(Page Table)

•
•
•

:

44				
40	D	X	C	V
36	Q	F	W	G
32	P	L	O	K
28	R	T	Y	U
24	H	J	A	Z
20	I	N	E	S
16	A	W	R	H
12				
8				
4				
0				

	0	1	2	3
0	D	X	C	V
4	Q	F	W	G
8	P	L	O	K
12	R	T	Y	U
16	H	J	A	Z
20	I	N	E	S
24	A	W	R	H
28	T	S	M	Y
32	E	A	U	I
36	B	R	N	T
40	O	D	F	R
44				
:				

10.2 b) Virtueller Speicher

Seiten-Nr.	Offset	Seitenfehler	Gelesen
1	0	nein	P
2	3	nein	Z
3	1	ja	
4	2		
3	3		
4	3		

Seiten-Kachel-Tabelle		
Virtuelle. Seiten-Nr.	Physikalische Seiten Nr.	v
0	24	0
1	32	1
2	24	1
3	0	1
4	32	0
5	36	1
6	16	0
⋮	...	⋮

Hauptspeicher

Seiten-Kachel-Tabelle
(Page Table)

•
•
•

:

44				
40	D	X	C	V
36	Q	F	W	G
32	P	L	O	K
28	R	T	Y	U
24	H	J	A	Z
20	I	N	E	S
16	A	W	R	H
12				
8				
4				
0	O	D	F	R

	0	1	2	3
0	D	X	C	V
4	Q	F	W	G
8	P	L	O	K
12	R	T	Y	U
16	H	J	A	Z
20	I	N	E	S
24	A	W	R	H
28	T	S	M	Y
32	E	A	U	I
36	B	R	N	T
40	O	D	F	R
44				
:				

10.2 b) Virtueller Speicher

Seiten-Nr.	Offset	Seitenfehler	Gelesen
1	0	nein	P
2	3	nein	Z
3	1	ja	D
4	2		
3	3		
4	3		

Seiten-Kachel-Tabelle

Virtuelle. Seiten-Nr.	Physikalische Seiten Nr.	v
0	24	0
1	32	1
2	24	1
3	0	1
4	32	0
5	36	1
6	16	0
...

Hauptspeicher

Seiten-Kachel-Tabelle (Page Table)				
•				
•				
•				
:				
44	D	X	C	V
40	Q	F	W	G
36	P	L	O	K
32	R	T	Y	U
28	H	J	A	Z
24	I	N	E	S
20	A	W	R	H
16				
12				
8				
4				
0	O	D	F	R

Hintergrundspeicher

	0	1	2	3
0	D	X	C	V
4	Q	F	W	G
8	P	L	O	K
12	R	T	Y	U
16	H	J	A	Z
20	I	N	E	S
24	A	W	R	H
28	T	S	M	Y
32	E	A	U	I
36	B	R	N	T
40	O	D	F	R
44				
:				

10.2 b) Virtueller Speicher

Seiten-Nr.	Offset	Seitenfehler	Gelesen
1	0	nein	P
2	3	nein	Z
3	1	ja	D
4	2	ja	
3	3		
4	3		

Seiten-Kachel-Tabellen		
Virtuelle. Seiten-Nr.	Physikalische Seiten Nr.	v
0	24	0
1	32	1
2	24	1
3	0	1
4	4	1
5	36	1
6	16	0
⋮	...	⋮

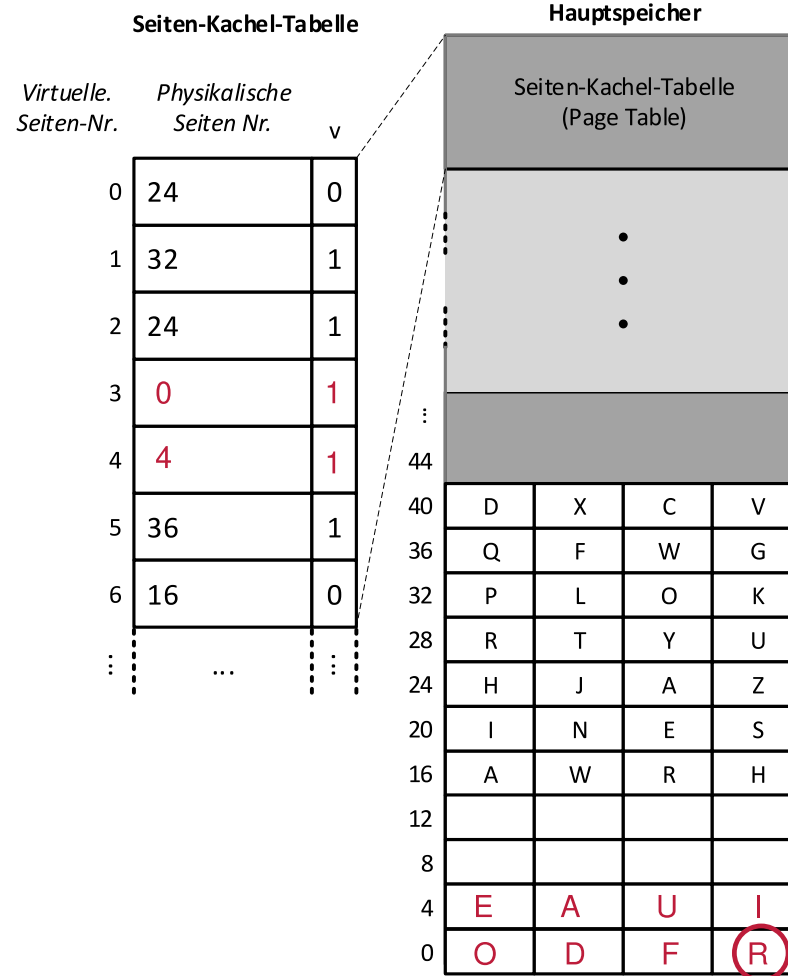
[illegible]

	0	1	2	3
0	D	X	C	V
4	Q	F	W	G
8	P	L	O	K
12	R	T	Y	U
16	H	J	A	Z
20	I	N	E	S
24	A	W	R	H
28	T	S	M	Y
32	E	A	U	I
36	B	R	N	T
40	O	D	F	R
44				
:				



10.2 b) Virtueller Speicher

Seiten-Nr.	Offset	Seitenfehler	Gelesen
1	0	nein	P
2	3	nein	Z
3	1	ja	D
4	2	ja	U
3	3		
4	3		

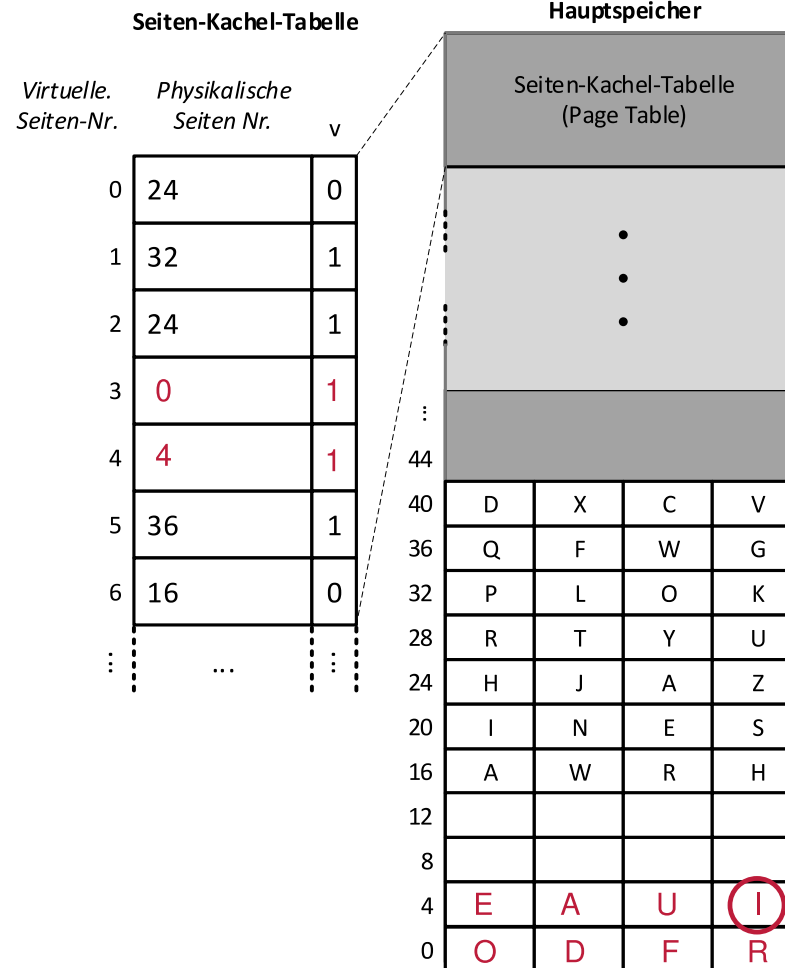


Hintergrundspeicher

	0	1	2	3
0	D	X	C	V
4	Q	F	W	G
8	P	L	O	K
12	R	T	Y	U
16	H	J	A	Z
20	I	N	E	S
24	A	W	R	H
28	T	S	M	Y
32	E	A	U	I
36	B	R	N	T
40	O	D	F	R
44				
...				

10.2 b) Virtueller Speicher

Seiten-Nr.	Offset	Seitenfehler	Gelesen
1	0	nein	P
2	3	nein	Z
3	1	ja	D
4	2	ja	U
3	3	nein	R
4	3		

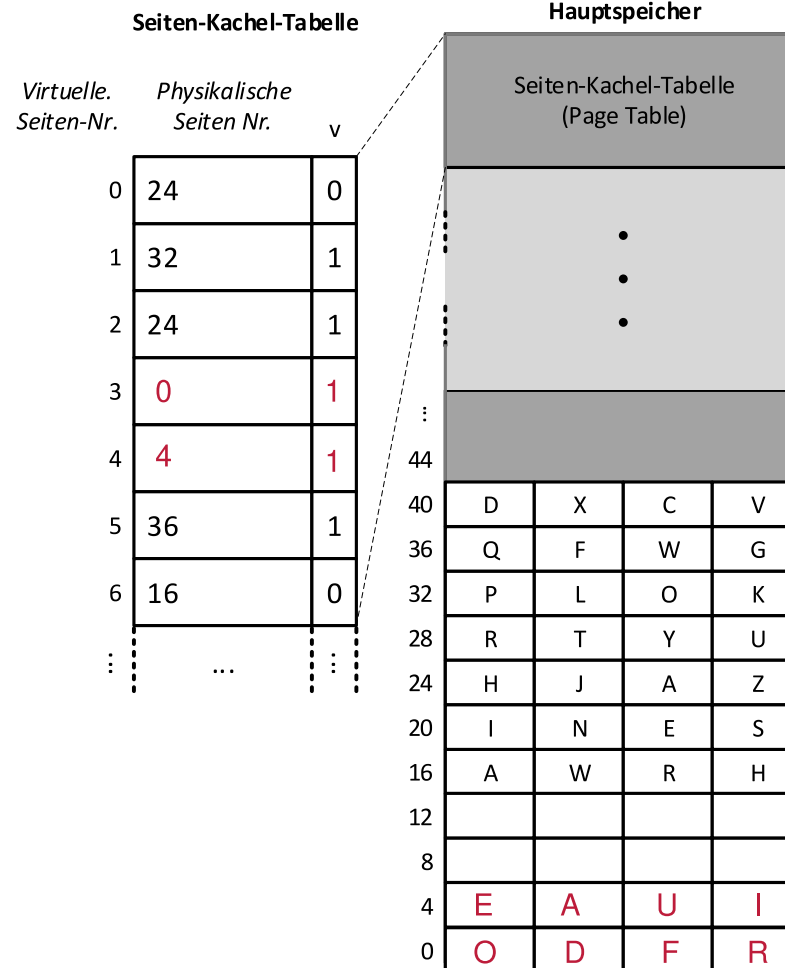


Hintergrundspeicher

	0	1	2	3
0	D	X	C	V
4	Q	F	W	G
8	P	L	O	K
12	R	T	Y	U
16	H	J	A	Z
20	I	N	E	S
24	A	W	R	H
28	T	S	M	Y
32	E	A	U	I
36	B	R	N	T
40	O	D	F	R
44				
...				

10.2 b) Virtueller Speicher

Seiten-Nr.	Offset	Seitenfehler	Gelesen
1	0	nein	P
2	3	nein	Z
3	1	ja	D
4	2	ja	U
3	3	nein	R
4	3	nein	I



Hintergrundspeicher

	0	1	2	3
0	D	X	C	V
4	Q	F	W	G
8	P	L	O	K
12	R	T	Y	U
16	H	J	A	Z
20	I	N	E	S
24	A	W	R	H
28	T	S	M	Y
32	E	A	U	I
36	B	R	N	T
40	O	D	F	R
44				
...				

10.2 c) Virtueller Speicher

Gegebene Daten:

- Kopiervorgang vom Hintergrundspeicher in RAM: 10 ms
- RAM Zugriffszeit: 10 ns
- Kein gleichzeitiger Zugriff auf Daten und SKT

10.2 c) Virtueller Speicher

Gegebene Daten:

- Kopiervorgang vom Hintergrundspeicher in RAM: 10 ms
- RAM Zugriffszeit: 10 ns
- Kein gleichzeitiger Zugriff auf Daten und SKT

Zugriffszeiten:

- Ohne Seitenfehler:
 $10 \text{ ns (SKT)} + 10 \text{ ns (Daten)} = 20 \text{ ns}$

10.2 c) Virtueller Speicher

Gegebene Daten:

- Kopiervorgang vom Hintergrundspeicher in RAM: 10 ms
- RAM Zugriffszeit: 10 ns
- Kein gleichzeitiger Zugriff auf Daten und SKT

Zugriffszeiten:

- Ohne Seitenfehler:

$$10 \text{ ns (SKT)} + 10 \text{ ns (Daten)} = 20 \text{ ns}$$

- Mit Seitenfehler:

$$10 \text{ ns (SKT)} + 10 \text{ ms (Einlagern)} + 10 \text{ ns (SKT aktualisieren)} + 10 \text{ ns (SKT)} \\ + 10 \text{ ns (Daten)} = 10.000.040 \text{ ns}$$

10.2 c) Virtueller Speicher

Seiten-Nr	Offset	Seitenfehler	Gelesen	Zeit
1	0	nein	P	20 ns
2	3	nein	Z	20 ns
3	1	ja	D	10 ms + 40 ns
4	2	ja	U	10 ms + 40 ns
3	3	nein	R	20 ns
4	3	nein	I	20 ns

Zugriffszeiten:

- ohne Seitenfehler: 20 ns
- mit Seitenfehler:
10.000.040 ns
- Summe: 20.000.160 ns

10.2 d) Virtueller Speicher

Gegebene Daten:

- 32-Bit virtueller Adressraum aufgeteilt in:
 - 11 Bit Seiten-Offset
 - 21 Bit virtuelle Seiten-Nr.
- Byte-weise Adressierung

10.2 d) Virtueller Speicher

Gegebene Daten:

- 32-Bit virtueller Adressraum aufgeteilt in:
 - 11 Bit Seiten-Offset
 - 21 Bit virtuelle Seiten-Nr.
- Byte-weise Adressierung

1. Wie viel Speicher umfasst eine Seite?

- Größe Seiten-Offset x Adressierungseinheit

$$\rightarrow 2^{11} \cdot 1 \text{ Byte} = 2 \text{ KiB}$$

10.2 d) Virtueller Speicher

Gegebene Daten:

- 32-Bit virtueller Adressraum aufgeteilt in:
 - 11 Bit Seiten-Offset → adressiert einen Bereich von 2 KiB
 - 21 Bit virtuelle Seiten-Nr.
- Byte-weise Adressierung

2. Wie viele Seiten hat die SKT?

10.2 d) Virtueller Speicher

Gegebene Daten:

- 32-Bit virtueller Adressraum aufgeteilt in:
 - 11 Bit Seiten-Offset → adressiert einen Bereich von 2 KiB
 - 21 Bit virtuelle Seiten-Nr.
- Byte-weise Adressierung

2. Wie viele Seiten hat die SKT?

- Abgeleitet aus Anzahl der möglichen virtuellen Seitennummern
- 2^{21} Seiten = 2.097.152 Seiten (je Prozess)

10.2 d) Virtueller Speicher

Gegebene Daten:

- 32-Bit virtueller Adressraum aufgeteilt in:
 - 11 Bit Seiten-Offset → adressiert einen Bereich von 2 KiB
 - 21 Bit virtuelle Seiten-Nr. → 2.097.152 Seiten (je Prozess)
 - Byte-weise Adressierung
3. Wie groß ist der Speicherbedarf für die SKT von 135 Prozessen, wenn eine SKT-Zeile 32 Bit breit ist?

10.2 d) Virtueller Speicher

Gegebene Daten:

- 32-Bit virtueller Adressraum aufgeteilt in:
 - 11 Bit Seiten-Offset → adressiert einen Bereich von 2 KiB
 - 21 Bit virtuelle Seiten-Nr. → 2.097.152 Seiten (je Prozess)
- Byte-weise Adressierung

3. Wie groß ist der Speicherbedarf für die SKT von 135 Prozessen, wenn eine SKT-Zeile 32 Bit breit ist?

- $2^{21} \cdot 135 \cdot 32 \text{ Bit}$
 $= 2.097.152 \cdot 135 \cdot 4 \text{ Byte}$
 $= 1.132.462.080 \text{ Byte}$
 $= 1.080 \text{ MiB} \approx 1,05 \text{ GiB}$

10.2 d) Virtueller Speicher

Gegebene Daten:

- 32-Bit virtueller Adressraum aufgeteilt in:
 - 11 Bit Seiten-Offset → adressiert einen Bereich von 2 KiB
 - 21 Bit virtuelle Seiten-Nr. → 2.097.152 Seiten (je Prozess)
 - Byte-weise Adressierung
4. Wiederholen Sie Schritte 1. bis 3. für einen virtuellen Adressraum von 64-Bit (SKT-Adressen = 44 Bit, Seiten-Offset = 20 Bit, SKT-Zeilenlänge = 55 Bit). Was fällt auf?

10.2 d) Virtueller Speicher

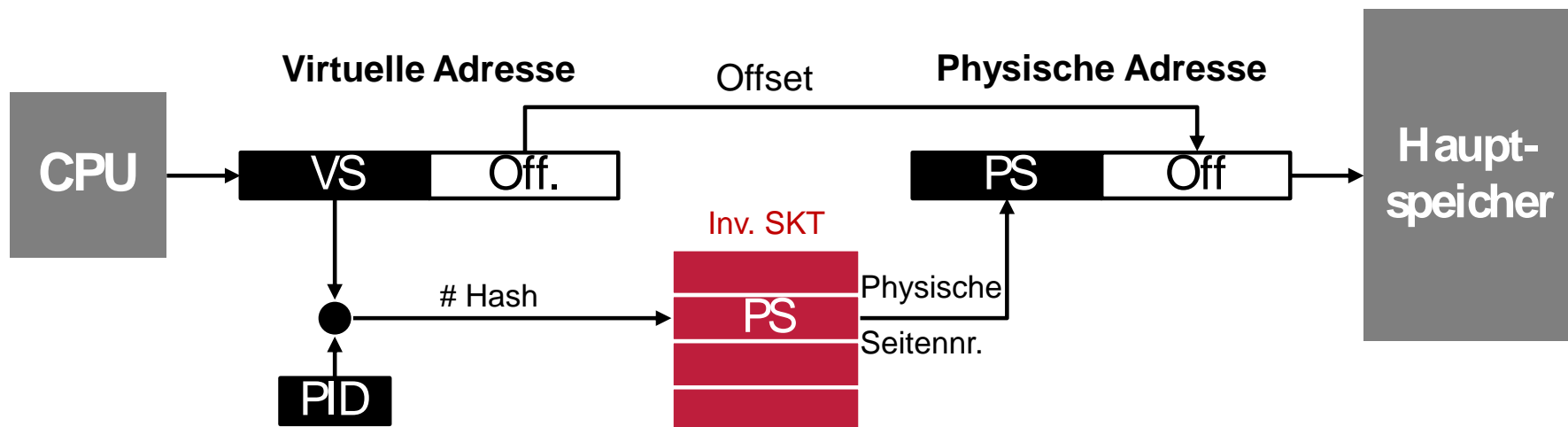
Gegebene Daten:

- 64-Bit virtueller Adressraum aufgeteilt in:
 - 20 Bit Seiten-Offset → adressiert einen Bereich von 1 MiB
 - 44 Bit virtuelle Seiten-Nr. → $2^{44} = 16$ Teraseiten (je Prozess)
 - Byte-weise Adressierung
4. Wiederholen Sie Schritte 1. bis 3. für einen virtuellen Adressraum von 64-Bit (SKT-Adressen = 44 Bit, Seiten-Offset = 20 Bit, SKT-Zeilenlänge = 55 Bit). Was fällt auf?
- $2^{44} \cdot 135 \cdot 55$ Bit
 - = $16 \cdot 135 \cdot 55$ Terabit
 - = 14,85 Petabyte

10.2 d) Virtueller Speicher - Anmerkungen

Invertierte Seiten-Kachel-Tabelle

- Inv. SKT enthält nur die Umsetzung für physische Adressen
 - Einträge in der SKT müssen gesucht werden
 - Üblich ist eine Realisierung als Hash-Tabelle
- Hash wird aus **Prozess ID (PID)** und **virtueller Seitennr. (VS)** gebildet



Übersicht

- 10.1 Speicherverwaltung
- 10.2 Virtueller Speicher
- **10.3 Ersetzungsstrategien**
- 10.4 Scheduling-Verfahren für Schreib-/Lesekopf

10.3 Ersetzungsstrategien

Gegeben:

- eine fiktive Seiten-Kachel-Tabelle (SKT) mit 3 Kacheln
- eine Referenzfolge: 3, 2, 5, 4, 2, 6, 2, 4, 3, 4

Bestimmen Sie Ablauf und Anzahl der Ersetzungen für folgende Ersetzungsstrategien:

- First-in, First-out (FIFO)**
- Least-Recently-Used (LRU)**
- Second-Chance (Clock)**

a) First In, First Out (FIFO)

- Älteste Seite wird ersetzt
- Kontrollzustand: Alter der Seite pro Kachel

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3									
	Kachel 2										
	Kachel 3										
Kontroll- zustände	Kachel 1	0									
	Kachel 2										
	Kachel 3										

a) First In, First Out (FIFO)

- Älteste Seite wird ersetzt
- Kontrollzustand: Alter der Seite pro Kachel

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3								
	Kachel 2		2								
	Kachel 3										
Kontroll- zustände	Kachel 1	0	1								
	Kachel 2		0								
	Kachel 3										

a) First In, First Out (FIFO)

- Älteste Seite wird ersetzt
- Kontrollzustand: Alter der Seite pro Kachel

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3							
	Kachel 2		2	2							
	Kachel 3			5							
Kontroll- zustände	Kachel 1	0	1	2							
	Kachel 2		0	1							
	Kachel 3			0							

a) First In, First Out (FIFO)

- Älteste Seite wird ersetzt
- Kontrollzustand: Alter der Seite pro Kachel

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4						
	Kachel 2		2	2	2						
	Kachel 3			5	5						
Kontroll- zustände	Kachel 1	0	1	2	0						
	Kachel 2		0	1	2						
	Kachel 3			0	1						

a) First In, First Out (FIFO)

- Älteste Seite wird ersetzt
- Kontrollzustand: Alter der Seite pro Kachel

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4					
	Kachel 2		2	2	2	2					
	Kachel 3			5	5	5					
Kontroll- zustände	Kachel 1	0	1	2	0	1					
	Kachel 2		0	1	2	3					
	Kachel 3			0	1	2					

a) First In, First Out (FIFO)

- Älteste Seite wird ersetzt
- Kontrollzustand: Alter der Seite pro Kachel

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4				
	Kachel 2		2	2	2	2	6				
	Kachel 3			5	5	5	5				
Kontroll- zustände	Kachel 1	0	1	2	0	1	2				
	Kachel 2		0	1	2	3	0				
	Kachel 3			0	1	2	3				

a) First In, First Out (FIFO)

- Älteste Seite wird ersetzt
- Kontrollzustand: Alter der Seite pro Kachel

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4	4			
	Kachel 2		2	2	2	2	6	6			
	Kachel 3			5	5	5	5	2			
Kontroll- zustände	Kachel 1	0	1	2	0	1	2	3			
	Kachel 2		0	1	2	3	0	1			
	Kachel 3			0	1	2	3	0			

a) First In, First Out (FIFO)

- Älteste Seite wird ersetzt
- Kontrollzustand: Alter der Seite pro Kachel

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4	4	4		
	Kachel 2		2	2	2	2	6	6	6		
	Kachel 3			5	5	5	5	2	2		
Kontroll- zustände	Kachel 1	0	1	2	0	1	2	3	4		
	Kachel 2		0	1	2	3	0	1	2		
	Kachel 3			0	1	2	3	0	1		

a) First In, First Out (FIFO)

- Älteste Seite wird ersetzt
- Kontrollzustand: Alter der Seite pro Kachel

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4	4	4	3	
	Kachel 2		2	2	2	2	6	6	6	6	
	Kachel 3			5	5	5	5	2	2	2	
Kontroll- zustände	Kachel 1	0	1	2	0	1	2	3	4	0	
	Kachel 2		0	1	2	3	0	1	2	3	
	Kachel 3			0	1	2	3	0	1	2	

a) First In, First Out (FIFO)

- Älteste Seite wird ersetzt
- Kontrollzustand: Alter der Seite pro Kachel

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4	4	4	3	3
	Kachel 2		2	2	2	2	6	6	6	6	4
	Kachel 3			5	5	5	5	2	2	2	2
Kontroll- zustände	Kachel 1	0	1	2	0	1	2	3	4	0	1
	Kachel 2		0	1	2	3	0	1	2	3	0
	Kachel 3			0	1	2	3	0	1	2	3

Anzahl der Ersetzungen: 8

a) First In, First Out (FIFO) – Anmerkungen

- Einfache Implementierung
 - Zeiger auf die älteste Kachel wandert immer um eine Kachel pro Ersetzung weiter
- Es kann zu Anomalien kommen:
 - Erhöhung der Kacheln führt evtl. zu mehr Ersetzungen
- Nicht besonders effizient

b) Least Recently Used (LRU)

- Seite mit längster Dauer seit Zugriff wird ersetzt
- Kontrollzustand: Rückwärtsabstand (Dauer)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3									
	Kachel 2										
	Kachel 3										
Kontroll- zustände	Kachel 1	0									
	Kachel 2										
	Kachel 3										

b) Least Recently Used (LRU)

- Seite mit längster Dauer seit Zugriff wird ersetzt
- Kontrollzustand: Rückwärtsabstand (Dauer)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3								
	Kachel 2		2								
	Kachel 3										
Kontroll- zustände	Kachel 1	0	1								
	Kachel 2		0								
	Kachel 3										

b) Least Recently Used (LRU)

- Seite mit längster Dauer seit Zugriff wird ersetzt
- Kontrollzustand: Rückwärtsabstand (Dauer)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3							
	Kachel 2		2	2							
	Kachel 3			5							
Kontroll- zustände	Kachel 1	0	1	2							
	Kachel 2		0	1							
	Kachel 3			0							

b) Least Recently Used (LRU)

- Seite mit längster Dauer seit Zugriff wird ersetzt
- Kontrollzustand: Rückwärtsabstand (Dauer)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4						
	Kachel 2		2	2	2						
	Kachel 3			5	5						
Kontroll- zustände	Kachel 1	0	1	2	0						
	Kachel 2		0	1	2						
	Kachel 3			0	1						

b) Least Recently Used (LRU)

- Seite mit längster Dauer seit Zugriff wird ersetzt
- Kontrollzustand: Rückwärtsabstand (Dauer)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4					
	Kachel 2		2	2	2	2					
	Kachel 3			5	5	5					
Kontroll- zustände	Kachel 1	0	1	2	0	1					
	Kachel 2		0	1	2	0					
	Kachel 3			0	1	2					

b) Least Recently Used (LRU)

- Seite mit längster Dauer seit Zugriff wird ersetzt
- Kontrollzustand: Rückwärtsabstand (Dauer)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4				
	Kachel 2		2	2	2	2	2				
	Kachel 3			5	5	5	6				
Kontroll- zustände	Kachel 1	0	1	2	0	1	2				
	Kachel 2		0	1	2	0	1				
	Kachel 3			0	1	2	0				

b) Least Recently Used (LRU)

- Seite mit längster Dauer seit Zugriff wird ersetzt
- Kontrollzustand: Rückwärtsabstand (Dauer)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4	4			
	Kachel 2		2	2	2	2	2	2			
	Kachel 3			5	5	5	6	6			
Kontroll- zustände	Kachel 1	0	1	2	0	1	2	3			
	Kachel 2		0	1	2	0	1	0			
	Kachel 3			0	1	2	0	1			

b) Least Recently Used (LRU)

- Seite mit längster Dauer seit Zugriff wird ersetzt
- Kontrollzustand: Rückwärtsabstand (Dauer)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4	4	4		
	Kachel 2		2	2	2	2	2	2	2		
	Kachel 3			5	5	5	6	6	6		
Kontroll- zustände	Kachel 1	0	1	2	0	1	2	3	0		
	Kachel 2		0	1	2	0	1	0	1		
	Kachel 3			0	1	2	0	1	2		

b) Least Recently Used (LRU)

- Seite mit längster Dauer seit Zugriff wird ersetzt
- Kontrollzustand: Rückwärtsabstand (Dauer)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4	4	4	4	
	Kachel 2		2	2	2	2	2	2	2	2	
	Kachel 3			5	5	5	6	6	6	3	
Kontroll- zustände	Kachel 1	0	1	2	0	1	2	3	0	1	
	Kachel 2		0	1	2	0	1	0	1	2	
	Kachel 3			0	1	2	0	1	2	0	

b) Least Recently Used (LRU)

- Seite mit längster Dauer seit Zugriff wird ersetzt
- Kontrollzustand: Rückwärtsabstand (Dauer)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4	4	4	4	4
	Kachel 2		2	2	2	2	2	2	2	2	2
	Kachel 3			5	5	5	6	6	6	3	3
Kontroll- zustände	Kachel 1	0	1	2	0	1	2	3	0	1	0
	Kachel 2		0	1	2	0	1	0	1	2	3
	Kachel 3			0	1	2	0	1	2	0	1

Anzahl der Ersetzungen: 6

b) Least Recently Used (LRU) – Anmerkungen

- Effizienter als FIFO (aber nicht immer)
- Aufwendige Implementierung
 - bei jedem Zugriff: Zähler für jede Kachel inkrementieren
 - bei Ersetzung: Kachel mit höchstem Zähler suchen und ersetzen

c) Clock (Second Chance)

- Heuristik für LRU Algorithmus
 - Relativ einfache Implementierung
 - Umlaufzeiger markiert die Kachel, die als nächstes ersetzt wird
 - Bei jedem Zugriff reihum: Kontrollzustand/Referenzbit nur von einer Kachel aktualisieren
 - Bei Ersetzung: Kontrollzustand kontrollieren
- wenn **0**: Kachel ersetzen
- wenn **1**:
- Kontrollzustand auf 0 setzen
 - Zeiger bewegen
 - erneut Ersetzung versuchen

c) Clock (Second Chance)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3									
	Kachel 2										
	Kachel 3										
Kontroll- zustände	Kachel 1	1									
	Kachel 2	0									
	Kachel 3	0									
	Zeiger	2									

c) Clock (Second Chance)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3								
	Kachel 2		2								
	Kachel 3										
Kontroll- zustände	Kachel 1	1	1								
	Kachel 2	0	1								
	Kachel 3	0	0								
	Zeiger	2	3								

c) Clock (Second Chance)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3							
	Kachel 2		2	2							
	Kachel 3			5							
Kontroll- zustände	Kachel 1	1	1	1							
	Kachel 2	0	1	1							
	Kachel 3	0	0	1							
	Zeiger	2	3	1							

c) Clock (Second Chance)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4						
	Kachel 2		2	2	2						
	Kachel 3			5	5						
Kontroll- zustände	Kachel 1	1	1	1	1						
	Kachel 2	0	1	1	0						
	Kachel 3	0	0	1	0						
	Zeiger	2	3	1	2						

c) Clock (Second Chance)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4					
	Kachel 2		2	2	2	2					
	Kachel 3			5	5	5					
Kontroll- zustände	Kachel 1	1	1	1	1	1					
	Kachel 2	0	1	1	0	1					
	Kachel 3	0	0	1	0	0					
	Zeiger	2	3	1	2	2					

c) Clock (Second Chance)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4				
	Kachel 2		2	2	2	2	2				
	Kachel 3			5	5	5	6				
Kontroll- zustände	Kachel 1	1	1	1	1	1	1				
	Kachel 2	0	1	1	0	1	0				
	Kachel 3	0	0	1	0	0	1				
	Zeiger	2	3	1	2	2	1				

c) Clock (Second Chance)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4	4			
	Kachel 2		2	2	2	2	2	2			
	Kachel 3			5	5	5	6	6			
Kontroll- zustände	Kachel 1	1	1	1	1	1	1	1			
	Kachel 2	0	1	1	0	1	0	1			
	Kachel 3	0	0	1	0	0	1	1			
	Zeiger	2	3	1	2	2	1	1			

c) Clock (Second Chance)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4	4	4		
	Kachel 2		2	2	2	2	2	2	2		
	Kachel 3			5	5	5	6	6	6		
Kontroll- zustände	Kachel 1	1	1	1	1	1	1	1	1		
	Kachel 2	0	1	1	0	1	0	1	1		
	Kachel 3	0	0	1	0	0	1	1	1		
	Zeiger	2	3	1	2	2	1	1	1		

c) Clock (Second Chance)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4	4	4	3	
	Kachel 2		2	2	2	2	2	2	2	2	
	Kachel 3			5	5	5	6	6	6	6	
Kontroll- zustände	Kachel 1	1	1	1	1	1	1	1	1	1	
	Kachel 2	0	1	1	0	1	0	1	1	0	
	Kachel 3	0	0	1	0	0	1	1	1	0	
	Zeiger	2	3	1	2	2	1	1	1	2	

c) Clock (Second Chance)

Ref.- Folge		3	2	5	4	2	6	2	4	3	4
Haupt- speicher	Kachel 1	3	3	3	4	4	4	4	4	3	3
	Kachel 2		2	2	2	2	2	2	2	2	4
	Kachel 3			5	5	5	6	6	6	6	6
Kontroll- zustände	Kachel 1	1	1	1	1	1	1	1	1	1	1
	Kachel 2	0	1	1	0	1	0	1	1	0	1
	Kachel 3	0	0	1	0	0	1	1	1	0	0
	Zeiger	2	3	1	2	2	1	1	1	2	3

Anzahl der Ersetzungen: 7

Übersicht

- 10.1 Speicherverwaltung
- 10.2 Virtueller Speicher
- 10.3 Ersetzungsstrategien
- **10.4 Scheduling-Verfahren für Schreib-/Lesekopf**

10.4 Scheduling-Verfahren für Schreib-/Lesekopf

Fiktive Festplatte mit 64 Spuren (0-63)

- Zugriffe auf Spuren: 55, 57, 35, 27, 19, 48, 40, 42, 2, 53
- Der Lese-/Schreibkopf befindet sich zu Beginn an Spur 29
- Suchzeit: 2 ms

Vorgehen

1. Die Zugriffsfolge in eine zum Scheduling-Verfahren passende Reihenfolge bringen
2. Zeit für die Umstellung des Lese-/Schreibkopfes berechnen
3. Zugriffszeitpunkt berechnen

a) First In, First Out (FIFO)

- Zugriffsfolge: 55, 57, 35, 27, 19, 48, 40, 42, 1, 53
- Start-Spur: 29
- Suchzeit: 2 ms

Spur	55	57	35	27	19	48	40	42	1	53
Suchzeit in ms										
Zugriffszeitpunkt										

a) First In, First Out (FIFO)

- Zugriffsfolge: 55, 57, 35, 27, 19, 48, 40, 42, 1, 53
- Start-Spur: 29
- Suchzeit: 2 ms

Spur	55	57	35	27	19	48	40	42	1	53
Suchzeit in ms	52	4	44	16	16	58	16	4	82	104
Zugriffszeitpunkt										

a) First In, First Out (FIFO)

- Zugriffsfolge: 55, 57, 35, 27, 19, 48, 40, 42, 1, 53
- Start-Spur: 29
- Suchzeit: 2 ms

Spur	55	57	35	27	19	48	40	42	1	53
Suchzeit in ms	52	4	44	16	16	58	16	4	82	104
Zugriffszeitpunkt	52	56	100	116	132	190	206	210	292	396

- Einfach
- Jedoch sehr ineffizient!

b) Shortest Seek Time First (SSTF)

- Zugriffsfolge: 55, 57, 35, 27, 19, 48, 40, 42, 1, 53
- Start-Spur: 29
- Suchzeit: 2 ms

Spur	27	19	35	40	42	48	53	55	57	1
Suchzeit in ms										
Zugriffszeitpunkt										

b) Shortest Seek Time First (SSTF)

- Zugriffsfolge: 55, 57, 35, 27, 19, 48, 40, 42, 1, 53
- Start-Spur: 29
- Suchzeit: 2 ms

Spur	27	19	35	40	42	48	53	55	57	1
Suchzeit in ms	4	16	32	10	4	12	10	4	4	112
Zugriffszeitpunkt										

b) Shortest Seek Time First (SSTF)

- Zugriffsfolge: 55, 57, 35, 27, 19, 48, 40, 42, 1, 53
- Start-Spur: 29
- Suchzeit: 2 ms

Spur	27	19	35	40	42	48	53	55	57	1
Suchzeit in ms	4	16	32	10	4	12	10	4	4	112
Zugriffszeitpunkt	4	20	52	62	66	78	88	92	96	208

→ Sehr effizient

→ Erfordert jedoch die Kenntnis der kompletten Zugriffsfolge

c) Elevator

- Zugriffsfolge: 55, 57, 35, 27, 19, 48, 40, 42, 1, 53
- Start-Spur: 29
- Suchzeit: 2 ms

Spur	27	19	1	35	40	42	48	53	55	57
Suchzeit in ms										
Zugriffszeitpunkt										

c) Elevator

- Zugriffsfolge: 55, 57, 35, 27, 19, 48, 40, 42, 1, 53
- Start-Spur: 29
- Suchzeit: 2 ms

Spur	27	19	1	35	40	42	48	53	55	57
Suchzeit in ms	4	16	36	68	10	4	12	10	4	4
Zugriffszeitpunkt										

c) Elevator

- Zugriffsfolge: 55, 57, 35, 27, 19, 48, 40, 42, 1, 53
- Start-Spur: 29
- Suchzeit: 2 ms

Spur	27	19	1	35	40	42	48	53	55	57
Suchzeit in ms	4	16	36	68	10	4	12	10	4	4
Zugriffszeitpunkt	4	20	56	124	134	138	150	160	164	168

- Sehr effizient
- Erfordert die Kenntnis der kompletten Zugriffsfolge
- Minimiert Richtungswechsel und ist daher materialschonend