

# Tutorübung Grundlagen: Betriebssysteme und Systemsoftware

Moritz Beckel

München, 21. Dezember 2022

Mittwoch 14:15-16:00 Uhr Online (<https://bbb.in.tum.de/mor-6ij-iuw-ypm>)

Zulip-Stream <https://zulip.in.tum.de/#narrow/stream/1296-GBS-Mi-1400-A>

Unterrichtsmaterialien findet ihr hier:

<https://home.in.tum.de/~beckel/gbs>

Lösungen wurden von mir selbst erstellt. Es besteht keine Garantie auf Korrektheit.

# Speicherverwaltung

## Adressraumabbildung

1. Direkte Adressierung (mit Swapping oder Relokation)
2. Basisadressierung
3. Segmentadressierung (Stack, Heap, Data, Text)
4. Paging

# Speicherverwaltung

## Paging

= Adressraum in Seiten aufgeteilt

1. Es müssen nicht alle Seiten im physischen Speicher sein
2. **Seiten** (pages) == virtuell und **Kacheln** (page frames) == physisch
3. Abbildung von virtuelle auf physische Adressen in der **Seitentabelle** (page table)
4. Aufteilung der virtuellen Adresse in virtuelle Seitennummer und **Offset**
5. Seitentabelleneintrag:
  1. **Seitennummer**
  2. **P-Bit** (physische Adresse existent)
  3. **U/S** (user/supervisor)
  4. **R-Bit** (referenced)
  5. **M-Bit** (modify)

# Speicherverwaltung

## Seitenauslagerung

1. **Seitenfehler (page fault)** wird ausgelöst wenn Seite nicht im physischen Speicher (kein P-Bit)
2. Kachel auswählen zu Auslagerung, neue Seite Einlagern, page table anpassen

## Seitenersetzungsstrategien

1. **FIFO-Strategie**
2. **Second Chance / Clock Algorithmus**
3. **LRU Algorithmus** (Least recently used)
  1. **NFU Algorithmus** (Not frequently used)
  2. **Aging**
4. **Working Set-s Algorithmus**

# Speicherverwaltung

## Freispeicherverwaltung

1. Datenstrukturen: Bitmap, verkettete Liste
2. Belegungsstrategien: First Fit, Next Fit, Best Fit, Worst Fit, Buddy Algorithmus

## Fragmentierung

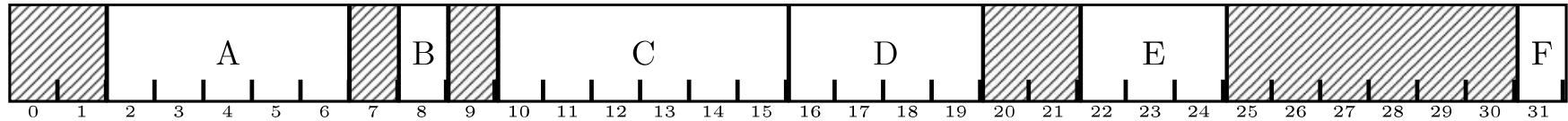
= Verschnitt bei Speicherallokation

1. **Extern** (= genügend Speicher für große Allokation vorhanden, aber nicht als ein Stück, Speicher zu stark Segmentiert, „durchlöchert“)
2. **Intern** (= zu viel Speicher allokiert, da Allokation nur Blockweise)

# Aufgabe 2

a) Fertigen Sie eine **Bitmap** an, welche den **Belegungszustand** korrekt wiedergibt

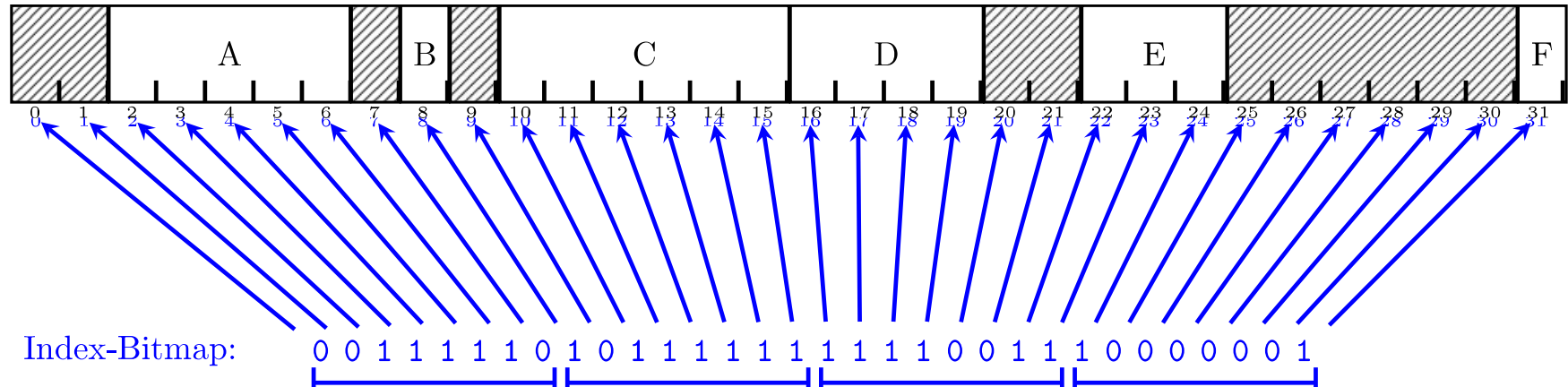
Gegeben sei die folgende Speicherbelegung (32 Blöcke):



# Aufgabe 2

a) Fertigen Sie eine **Bitmap** an, welche den **Belegungszustand** korrekt wiedergibt

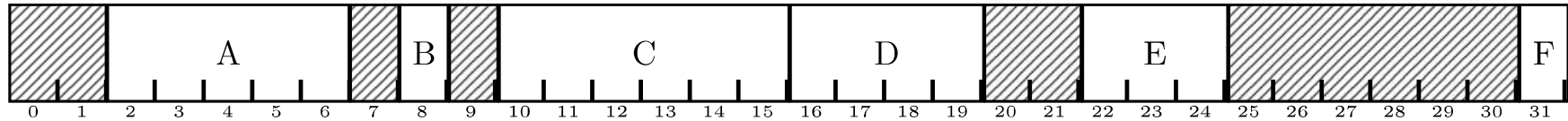
Gegeben sei die folgende Speicherbelegung (32 Blöcke):



## Aufgabe 2

b) Fertigen Sie eine **verkettete Liste** an, welche den **Belegungszustand** korrekt wiedergibt.

Gegeben sei die folgende Speicherbelegung (32 Blöcke):

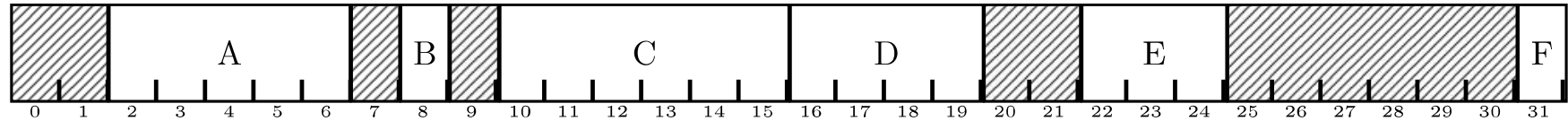




# Aufgabe 2

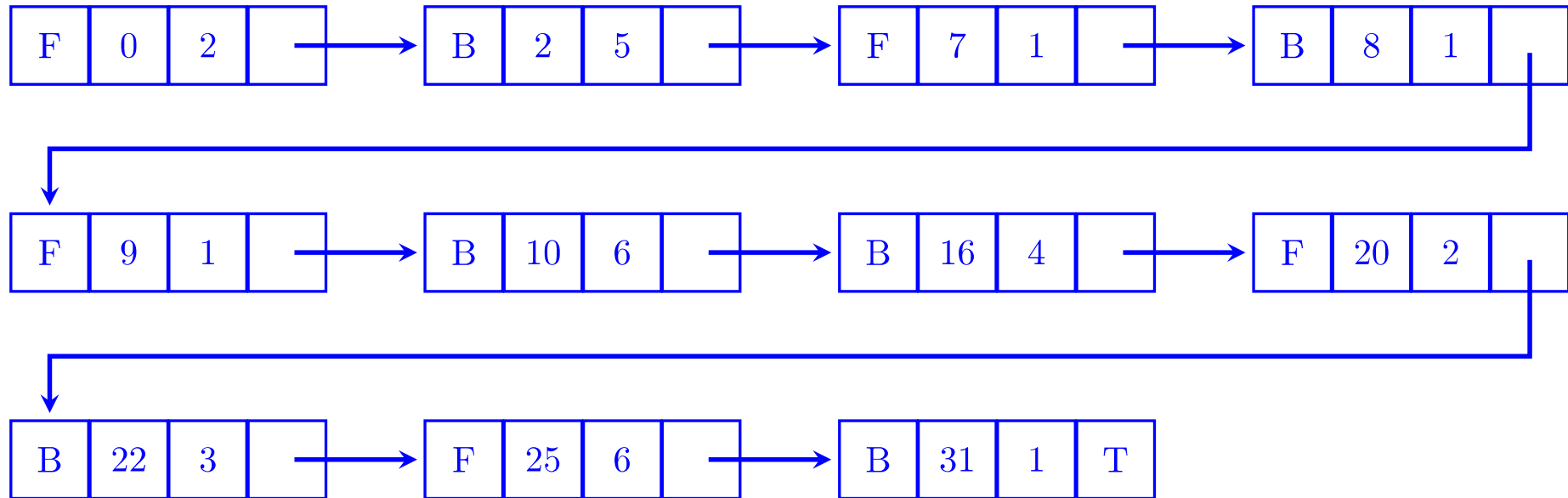
b) Fertigen Sie eine **verkettete Liste** an, welche den **Belegungszustand** korrekt wiedergibt.

Gegeben sei die folgende Speicherbelegung (32 Blöcke):



# Aufgabe 2

## Aufgabe 2



# Aufgabe 3

Halten Sie den **Zustand nach jeder Anfrage** tabellarisch wie folgt für die unten stehenden Verfahren fest:

Anfrage		Liste freier Speicherbereiche				
		100 KiB	400 KiB	250 KiB	200 KiB	50 KiB
30 KiB	<b>70 KiB</b>	400 KiB	250 KiB	200 KiB	50 KiB	
...						

# Aufgabe 3

a) First Fit

Anfrage	Liste freier Speicherbereiche				
	100 KiB	400 KiB	250 KiB	200 KiB	50 KiB
30 KiB					
60 KiB					
120 KiB					
20 KiB					
100 KiB					
250 KiB					

# Aufgabe 3

a) First Fit

Anfrage	Liste freier Speicherbereiche				
	100 KiB	400 KiB	250 KiB	200 KiB	50 KiB
30 KiB	<b>70 KiB</b>	400 KiB	250 KiB	200 KiB	50 KiB
60 KiB	<b>10 KiB</b>	400 KiB	250 KiB	200 KiB	50 KiB
120 KiB	10 KiB	<b>280 KiB</b>	250 KiB	200 KiB	50 KiB
20 KiB	10 KiB	<b>260 KiB</b>	250 KiB	200 KiB	50 KiB
100 KiB	10 KiB	<b>160 KiB</b>	250 KiB	200 KiB	50 KiB
250 KiB	10 KiB	160 KiB	<b>0 KiB</b>	200 KiB	50 KiB

# Aufgabe 3

b) Next Fit

Anfrage	Liste freier Speicherbereiche				
	100 KiB	400 KiB	250 KiB	200 KiB	50 KiB
30 KiB					
60 KiB					
120 KiB					
20 KiB					
100 KiB					
250 KiB					

# Aufgabe 3

b) Next Fit

Anfrage	Liste freier Speicherbereiche				
	100 KiB	400 KiB	250 KiB	200 KiB	50 KiB
30 KiB	<b>70 KiB</b>	400 KiB	250 KiB	200 KiB	50 KiB
60 KiB	<b>10 KiB</b>	400 KiB	250 KiB	200 KiB	50 KiB
120 KiB	10 KiB	<b>280 KiB</b>	250 KiB	200 KiB	50 KiB
20 KiB	10 KiB	<b>260 KiB</b>	250 KiB	200 KiB	50 KiB
100 KiB	10 KiB	<b>160 KiB</b>	250 KiB	200 KiB	50 KiB
250 KiB	10 KiB	160 KiB	<b>0 KiB</b>	200 KiB	50 KiB



# Aufgabe 3

c) Best Fit

Anfrage	Liste freier Speicherbereiche				
	100 KiB	400 KiB	250 KiB	200 KiB	50 KiB
30 KiB					
60 KiB					
120 KiB					
20 KiB					
100 KiB					
250 KiB					

# Aufgabe 3

c) Best Fit

Anfrage	Liste freier Speicherbereiche				
	100 KiB	400 KiB	250 KiB	200 KiB	50 KiB
30 KiB	100 KiB	400 KiB	250 KiB	200 KiB	<b>20 KiB</b>
60 KiB	<b>40 KiB</b>	400 KiB	250 KiB	200 KiB	20 KiB
120 KiB	40 KiB	400 KiB	250 KiB	<b>80 KiB</b>	20 KiB
20 KiB	40 KiB	400 KiB	250 KiB	80 KiB	<b>0 KiB</b>
100 KiB	40 KiB	400 KiB	<b>150 KiB</b>	80 KiB	0 KiB
250 KiB	40 KiB	<b>150 KiB</b>	150 KiB	80 KiB	0 KiB

# Aufgabe 3

d) Worst Fit

Anfrage	Liste freier Speicherbereiche				
	100 KiB	400 KiB	250 KiB	200 KiB	50 KiB
30 KiB					
60 KiB					
120 KiB					
20 KiB					
100 KiB					
250 KiB					

# Aufgabe 3

d) Worst Fit

Anfrage	Liste freier Speicherbereiche				
	100 KiB	400 KiB	250 KiB	200 KiB	50 KiB
30 KiB	100 KiB	<b>370 KiB</b>	250 KiB	200 KiB	50 KiB
60 KiB	100 KiB	<b>310 KiB</b>	250 KiB	200 KiB	50 KiB
120 KiB	100 KiB	<b>190 KiB</b>	250 KiB	200 KiB	50 KiB
20 KiB	100 KiB	190 KiB	<b>230 KiB</b>	200 KiB	50 KiB
100 KiB	100 KiB	190 KiB	<b>130 KiB</b>	200 KiB	50 KiB
250 KiB	Kein geeigneter Speicherplatz vorhanden				

```

A = allocate (13337) ;
B = allocate (24242) ;
C = allocate (8193) ;
D = allocate (13) ;
free ( A ) ;
E = allocate (32768) ;
free ( B ) ;
F = allocate (10000) ;

```

```

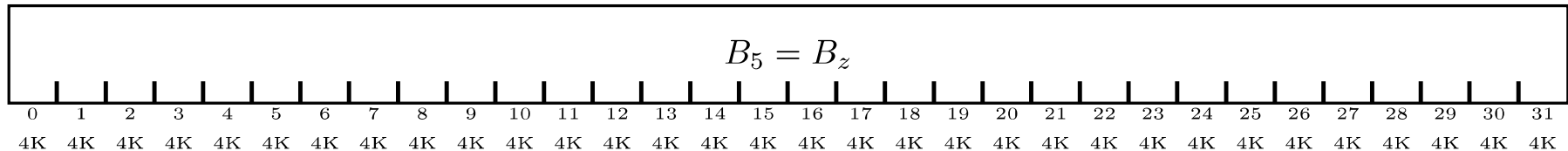
G = allocate (12345) ;
H = allocate (11111) ;
free ( G ) ;
free ( D ) ;
free ( H ) ;
free ( C ) ;
free ( F ) ;
free ( E ) ;

```

## Aufgabe 4

Gegeben sei der **Buddy-Allocator-Algorithmus** gemäß Vorlesung, sowie der Pseudocode des folgenden Programms. Gehen Sie von einer **Blockgröße** von 4096 Bytes (**4KiB**) und einer **Gesamtspeichergröße** von 131072 Bytes (**128KiB**) aus.

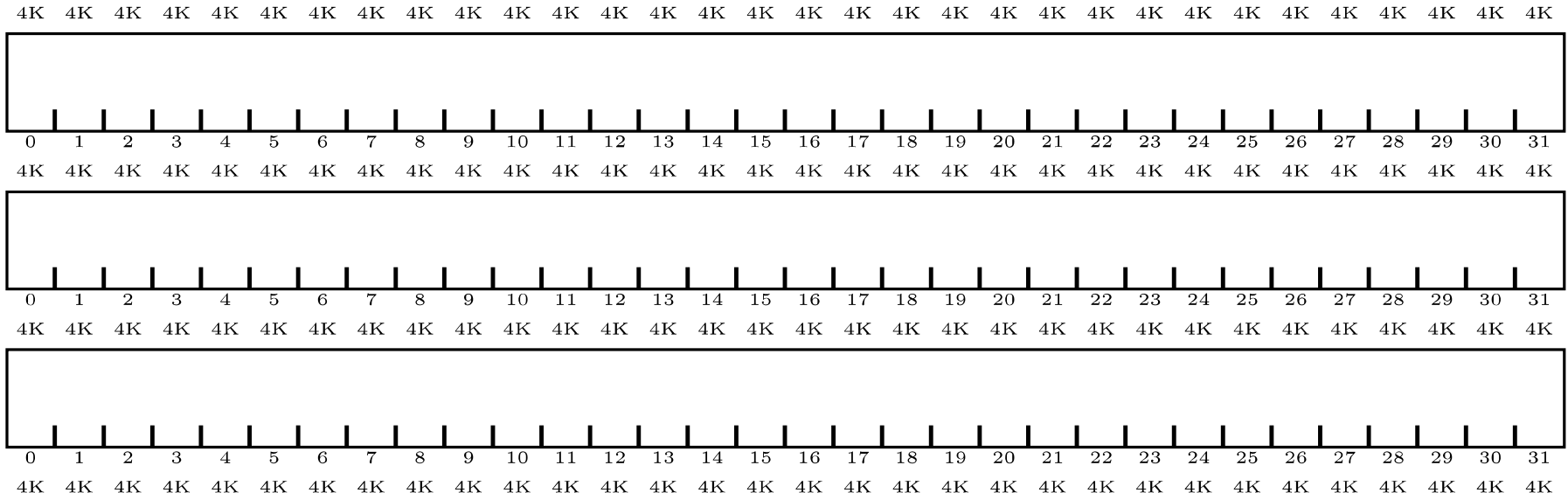
Zeichnen Sie die **Belegung des Speichers** nach der Ausführung jedes Statements aus dem obigen Pseudocode auf. Legen Sie besonderen Wert darauf, Blockgrößen sowie deren Belegungszustand genau zu kennzeichnen. Berechnen Sie außerdem nach jedem Schritt die **Größe des Verschnitts aller Blöcke** zusammen.



# Aufgabe 4

```
A = allocate (13337) ;
B = allocate (24242) ;
C = allocate (8193) ;
D = allocate (13) ;
free ( A ) ;
E = allocate (32768) ;
free ( B ) ;
F = allocate (10000) ;
```

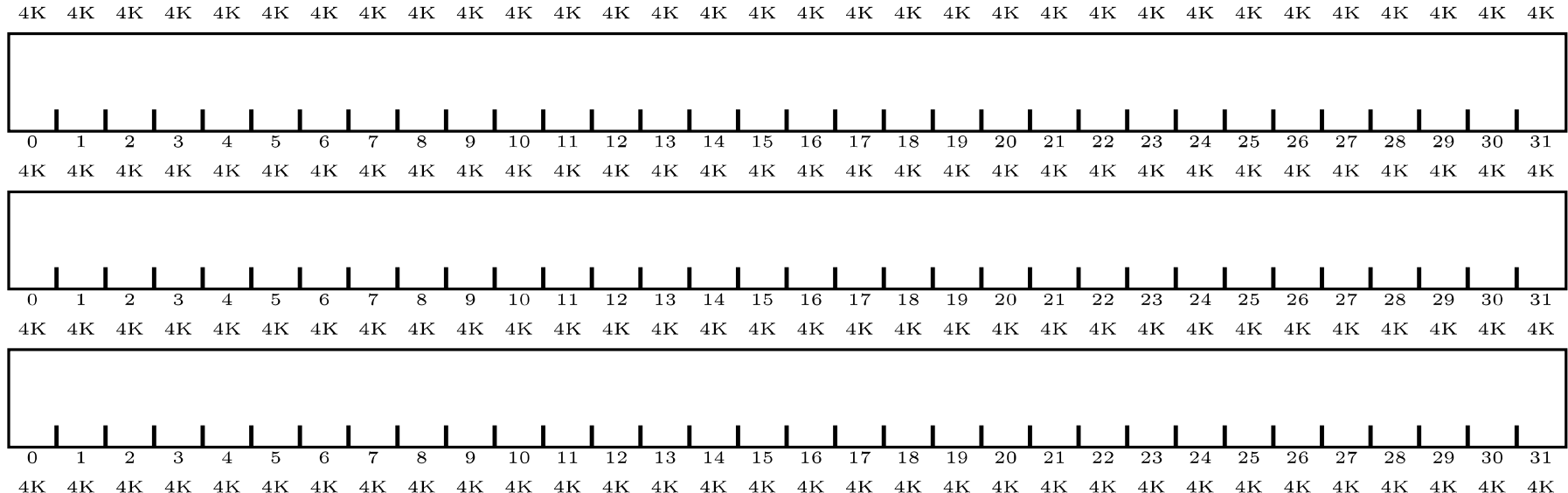
```
G = allocate (12345) ;
H = allocate (11111) ;
free ( G ) ;
free ( D ) ;
free ( H ) ;
free ( C ) ;
free ( F ) ;
free ( E ) ;
```



# Aufgabe 4

```
A = allocate (13337) ;
B = allocate (24242) ;
C = allocate (8193) ;
D = allocate (13) ;
free ( A ) ;
E = allocate (32768) ;
free ( B ) ;
F = allocate (10000) ;
```

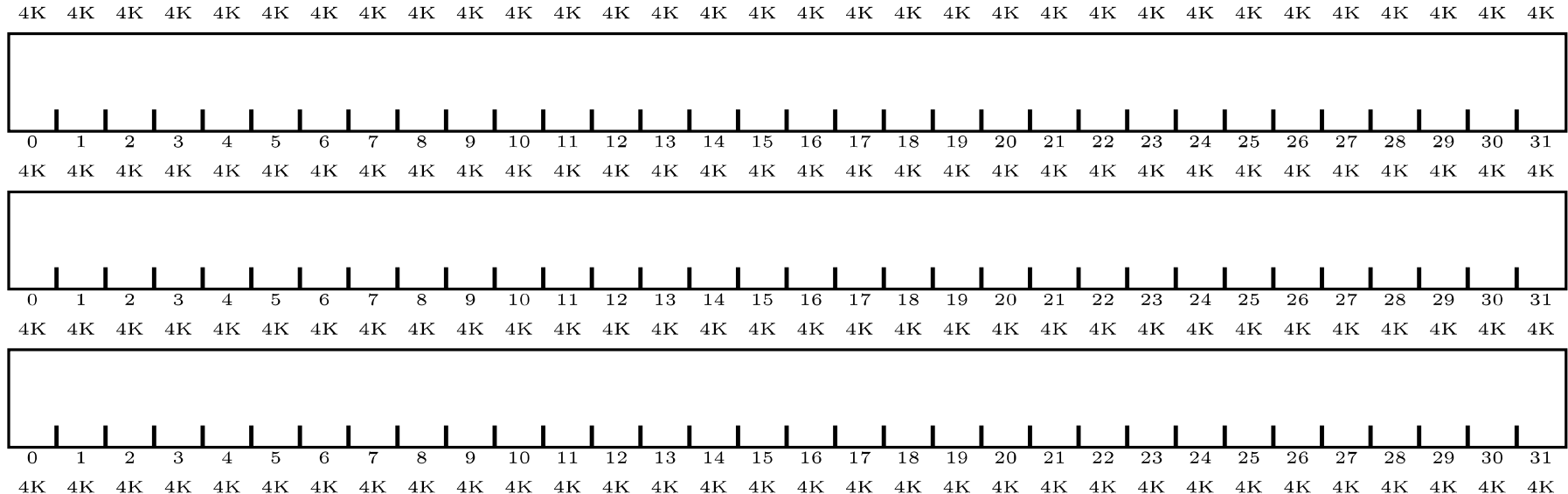
```
G = allocate (12345) ;
H = allocate (11111) ;
free ( G ) ;
free ( D ) ;
free ( H ) ;
free ( C ) ;
free ( F ) ;
free ( E ) ;
```



# Aufgabe 4

```
A = allocate (13337) ;
B = allocate (24242) ;
C = allocate (8193) ;
D = allocate (13) ;
free ( A ) ;
E = allocate (32768) ;
free ( B ) ;
F = allocate (10000) ;
```

```
G = allocate (12345) ;
H = allocate (11111) ;
free ( G ) ;
free ( D ) ;
free ( H ) ;
free ( C ) ;
free ( F ) ;
free ( E ) ;
```

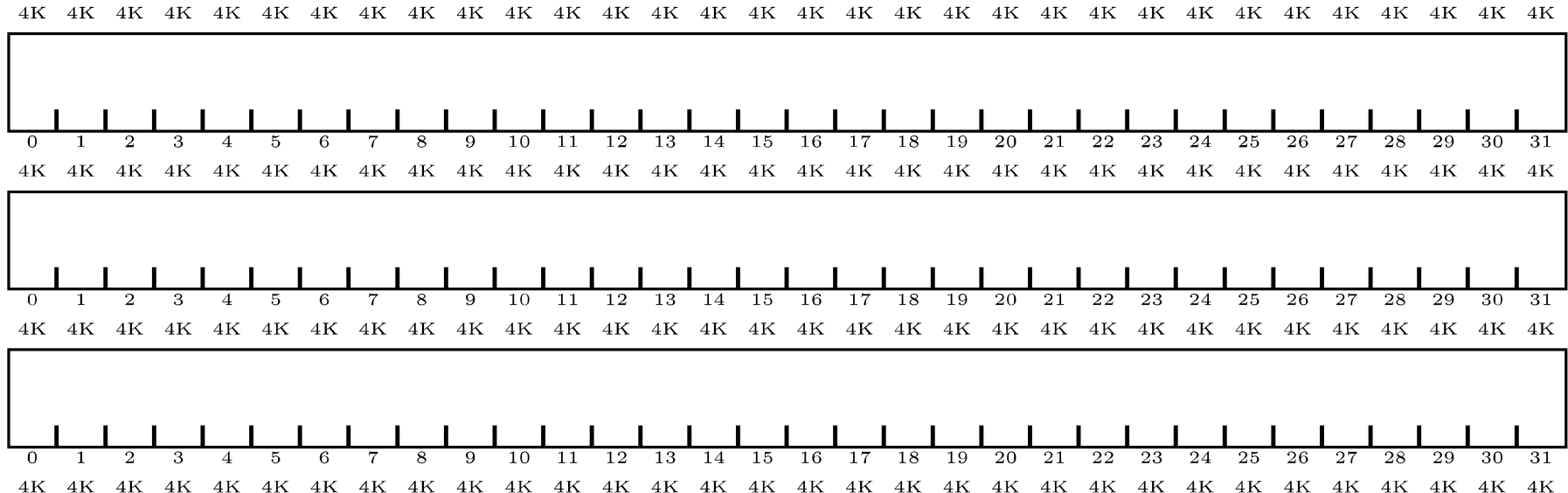




# Aufgabe 4

```
A = allocate (13337) ;
B = allocate (24242) ;
C = allocate (8193) ;
D = allocate (13) ;
free ( A ) ;
E = allocate (32768) ;
free ( B ) ;
F = allocate (10000) ;
```

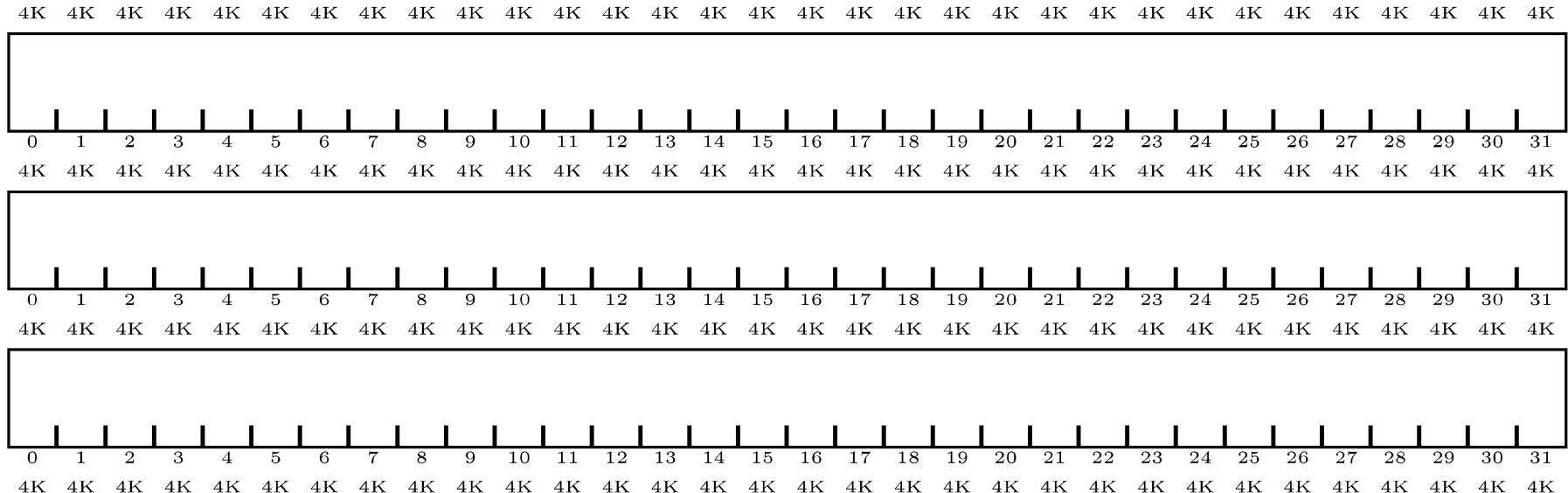
```
G = allocate (12345) ;
H = allocate (11111) ;
free ( G ) ;
free ( D ) ;
free ( H ) ;
free ( C ) ;
free ( F ) ;
free ( E ) ;
```



# Aufgabe 4

```
A = allocate (13337) ;
B = allocate (24242) ;
C = allocate (8193) ;
D = allocate (13) ;
free ( A ) ;
E = allocate (32768) ;
free ( B ) ;
F = allocate (10000) ;
```

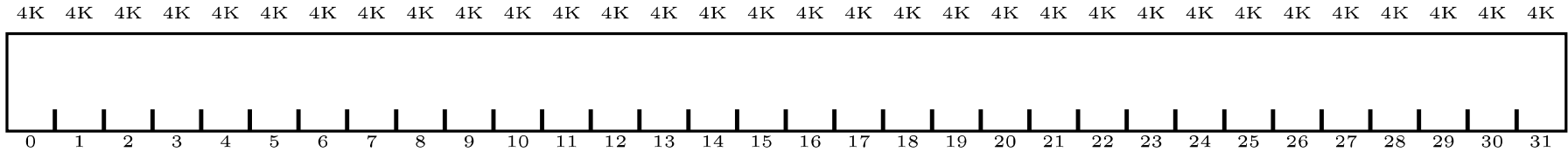
```
G = allocate (12345) ;
H = allocate (11111) ;
free ( G ) ;
free ( D ) ;
free ( H ) ;
free ( C ) ;
free ( F ) ;
free ( E ) ;
```



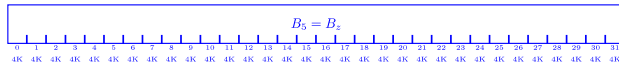
# Aufgabe 4

```
A = allocate (13337) ;
B = allocate (24242) ;
C = allocate (8193) ;
D = allocate (13) ;
free ( A ) ;
E = allocate (32768) ;
free ( B ) ;
F = allocate (10000) ;
```

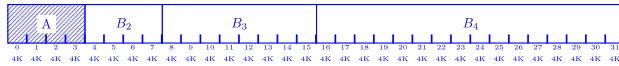
```
G = allocate (12345) ;
H = allocate (11111) ;
free ( G ) ;
free ( D ) ;
free ( H ) ;
free ( C ) ;
free ( F ) ;
free ( E ) ;
```



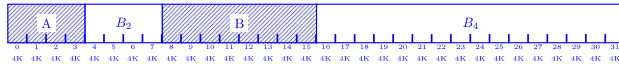
1. Initialzustand: (Verschnitt 0 Bytes)



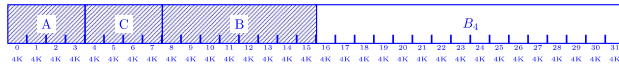
2. Nach **A** = **allocate**(13337): (Verschnitt 3047 Bytes)



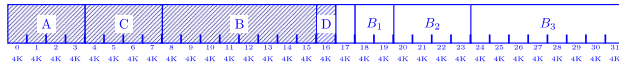
3. Nach **B** = **allocate**(24242): (Verschnitt 11573 Bytes)



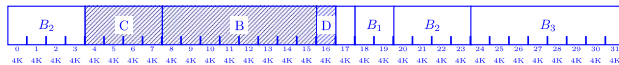
4. Nach **C** = **allocate**(8193): (Verschnitt 19764 Bytes)



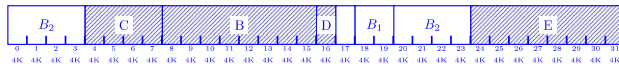
5. Nach **D** = **allocate**(13): (Verschnitt 23847 Bytes)



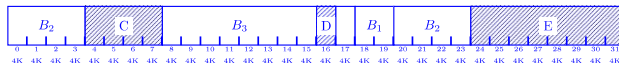
6. Nach **free**(A): (Verschnitt 20800 Bytes)



7. Nach **E** = **allocate**(32768): (Verschnitt 20800 Bytes)



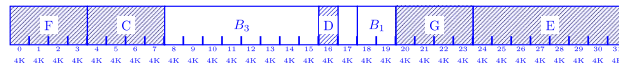
8. Nach **free**(B): (Verschnitt 12274 Bytes)



9. Nach **F** = **allocate**(10000): (Verschnitt 18658 Bytes)



10. Nach **G** = **allocate**(12345): (Verschnitt 22697 Bytes)



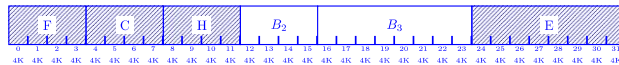
11. Nach **H** = **allocate**(11111): (Verschnitt 27970 Bytes)



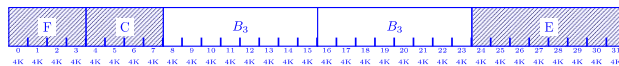
12. Nach **free**(G): (Verschnitt 23931 Bytes)



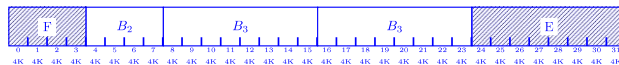
13. Nach **free**(D): (Verschnitt 19848 Bytes)



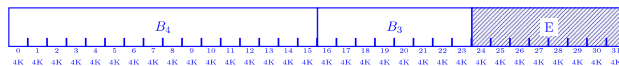
14. Nach **free**(H): (Verschnitt 14575 Bytes)



15. Nach **free**(C): (Verschnitt 6384 Bytes)



16. Nach **free**(F): (Verschnitt 0 Bytes)



17. Nach **free**(E): (Verschnitt 0 Bytes)

