

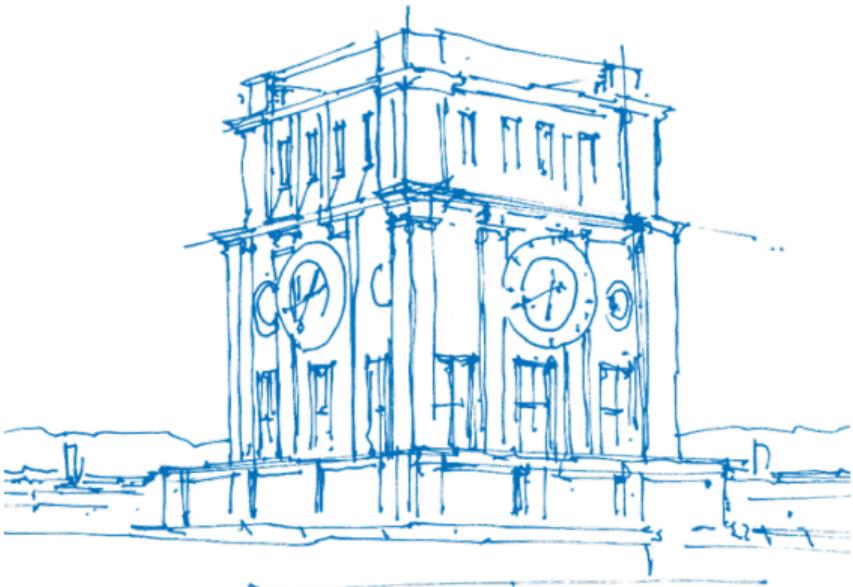
# Grundlagen: Betriebssysteme und Systemsoftware

## Tutorübung

**Mario Delic**

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

Übungswoche 8



TUM Uhrenturm

send()      recv()

## Aufgabe 2

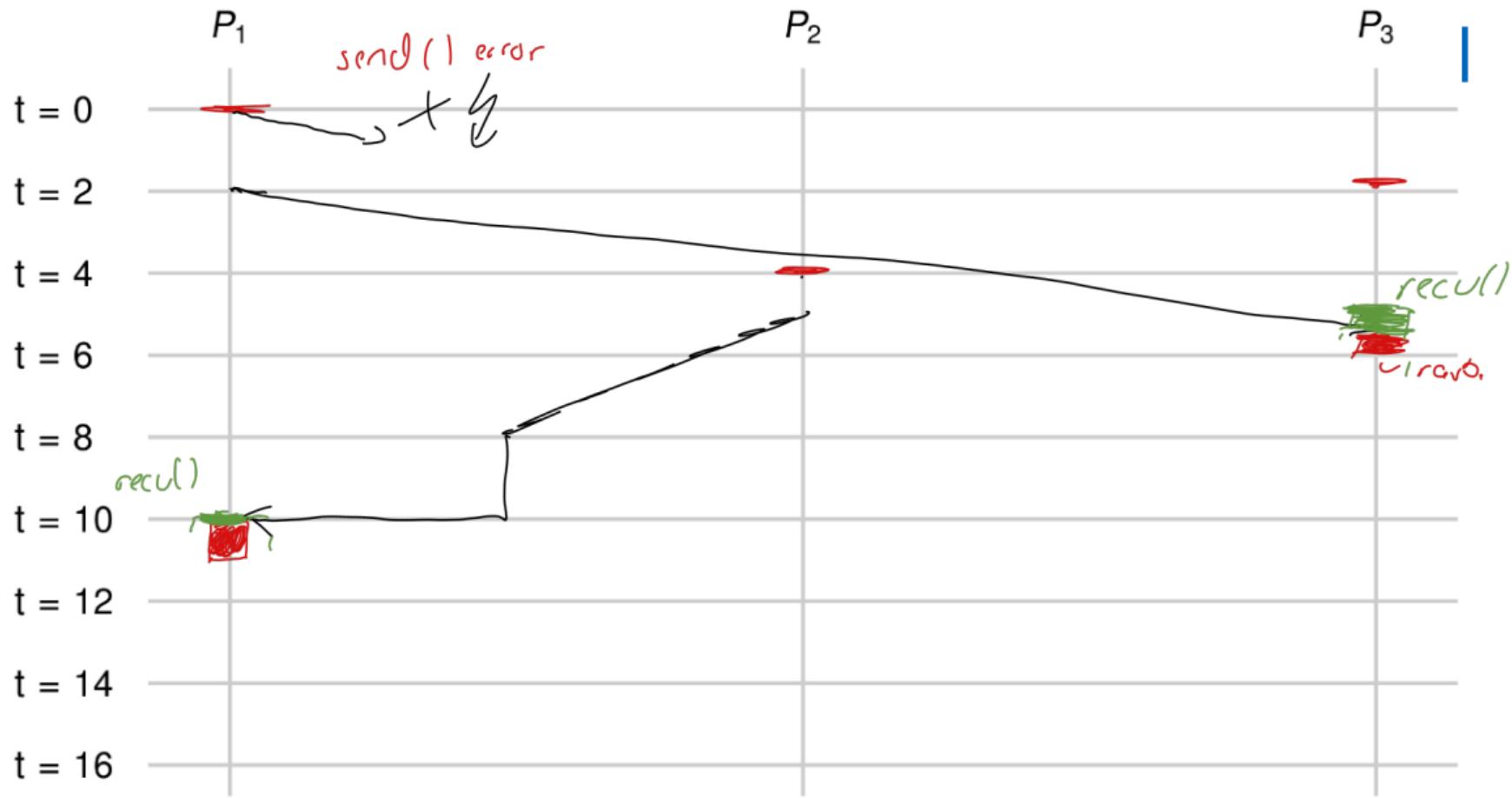
### Modellierung von Kommunikation

$\hookrightarrow = \text{write}()$        $\hookleftarrow = \text{read}()$

- Eine Nachricht kann nur abgeschickt werden, wenn der Empfänger existiert.
- Nachrichtenlaufzeiten = 3 ZE (Mindestzeit zw. erfolgreichem send und receive).
- Bei erfolgreichem receive wird 1 ZE verarbeitet.      *recv()*
- Sonstige Operationen sind overheadfrei (send, Verarbeitung von Bestätigungen...).

#### a) **Asynchron:**

- send wartet nicht auf eine Bestätigungsmessage vom Empfänger.
- Der Sender ist nach erfolgreichem send 'fertig'.
- Der Empfänger ist nach Verarbeiten der Nachricht 'fertig'.



int fd = open() (loc)

## Aufgabe 2

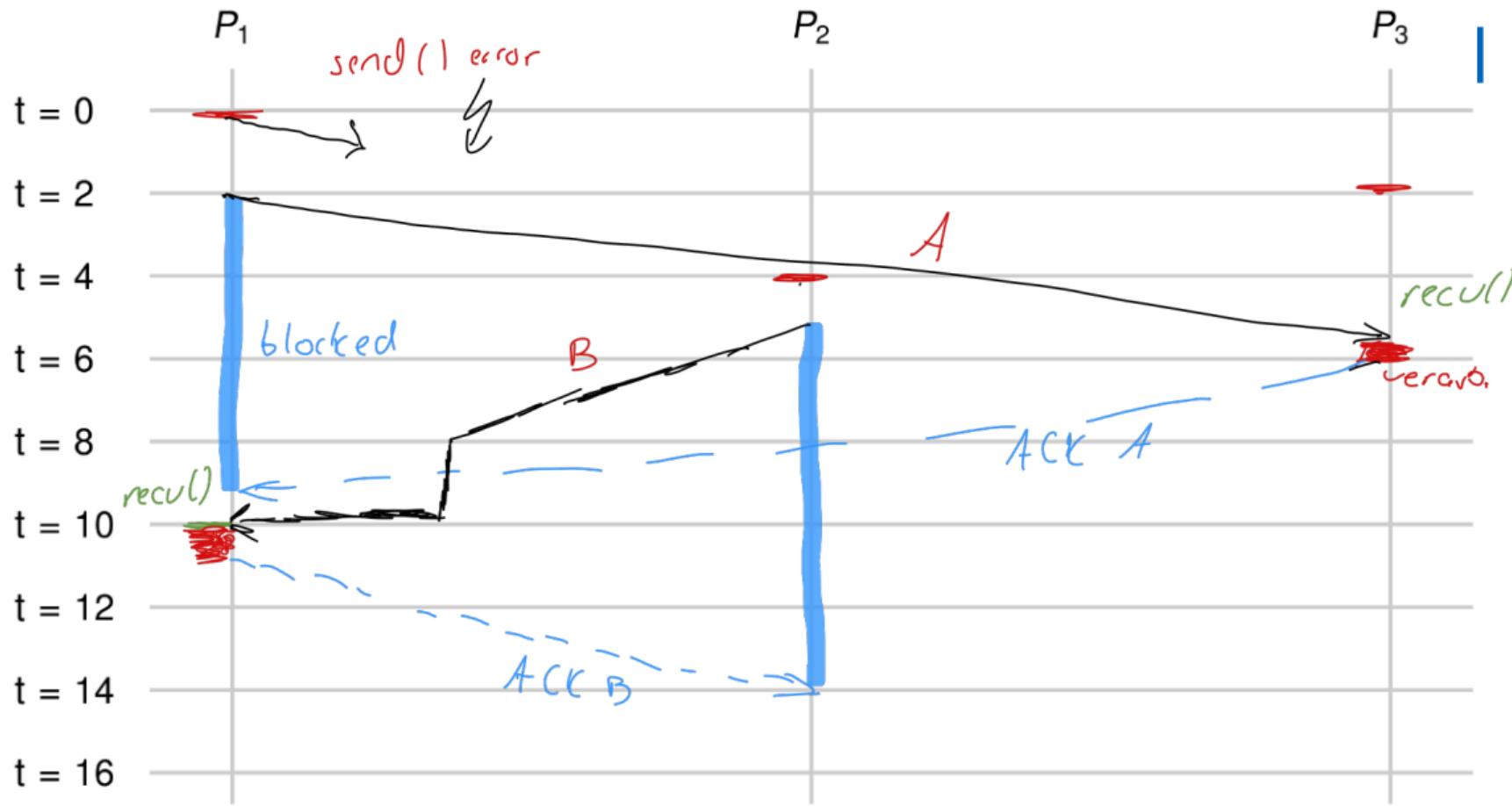
Modellierung von Kommunikation int sd = socket() (int)

- Eine Nachricht kann nur abgeschickt werden, wenn der Empfänger existiert.
- Nachrichtenlaufzeiten = 3 ZE (Mindestzeit zw. erfolgreichem send und receive).
- Bei erfolgreichem receive wird 1 ZE verarbeitet.
- Sonstige Operationen sind overheadfrei (send, Verarbeitung von Bestätigungen...).

b) **Synchron:**

ACK

- send wartet auf eine Bestätigungs Nachricht vom Empfänger.
- send blockiert nach erfolgreichem senden bis zum Erhalt dieser Nachricht.
- Der Empfänger sendet nach der Verarbeitung die Bestätigungs Nachricht.
- Der Empfänger ist nach Senden dieser Nachricht 'fertig'.
- Der Sender ist nach Erhalt der Bestätigungs Nachricht 'fertig'.



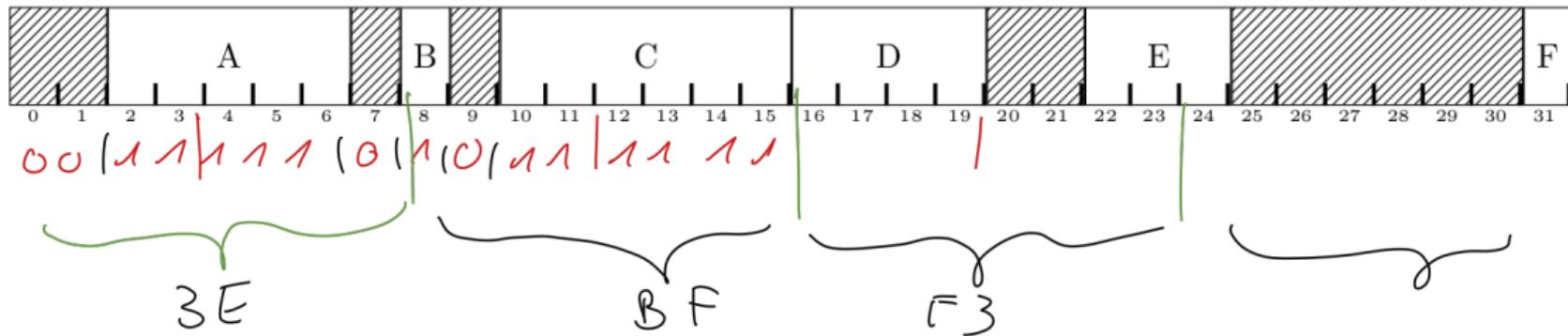
# Aufgabe 3

## Bitmap

Jedes Bit codiert einen Block: 1 = belegt, 0 = frei.

Speicherbelegung → (Binärdarstellung →) Hexdarstellung.

Bekanntes Schema: vier Bits zu einer Hex-Ziffer zusammenfassen.



# Aufgabe 3

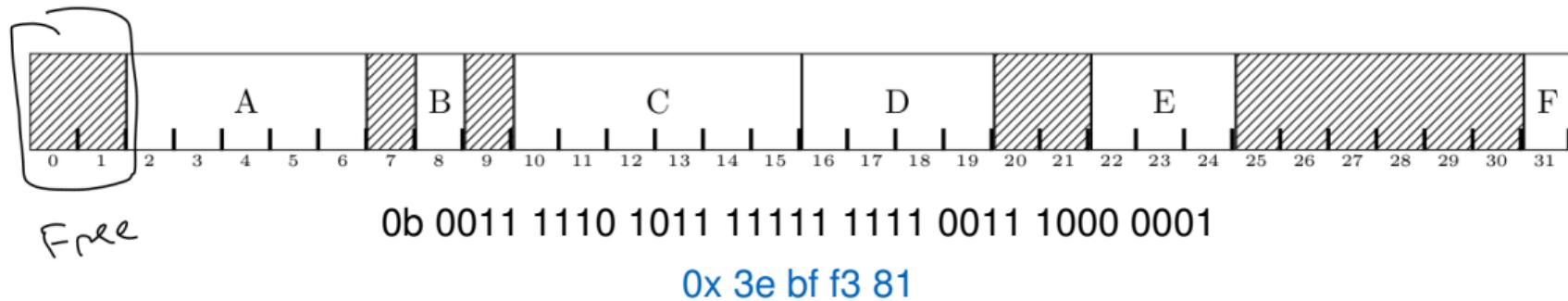
## Bitmap

Jedes Bit codiert einen Block: 1 = belegt, 0 = frei.

in bmp  
- inodes  
Ob bmp  
- datablocks

Speicherbelegung → (Binärdarstellung →) Hexdarstellung.

Bekanntes Schema: vier Bits zu einer Hex-Ziffer zusammenfassen.

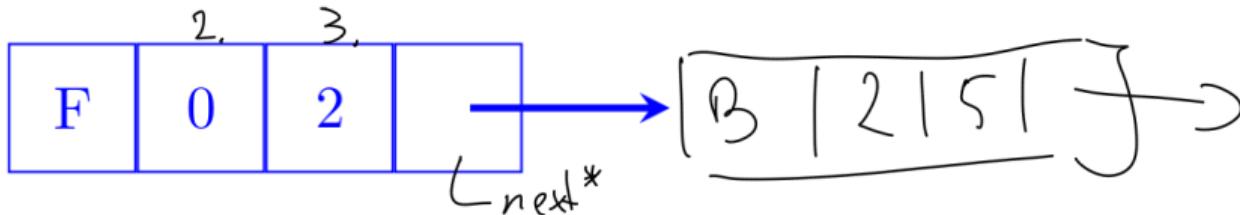


## Aufgabe 3

### Verkettete Liste

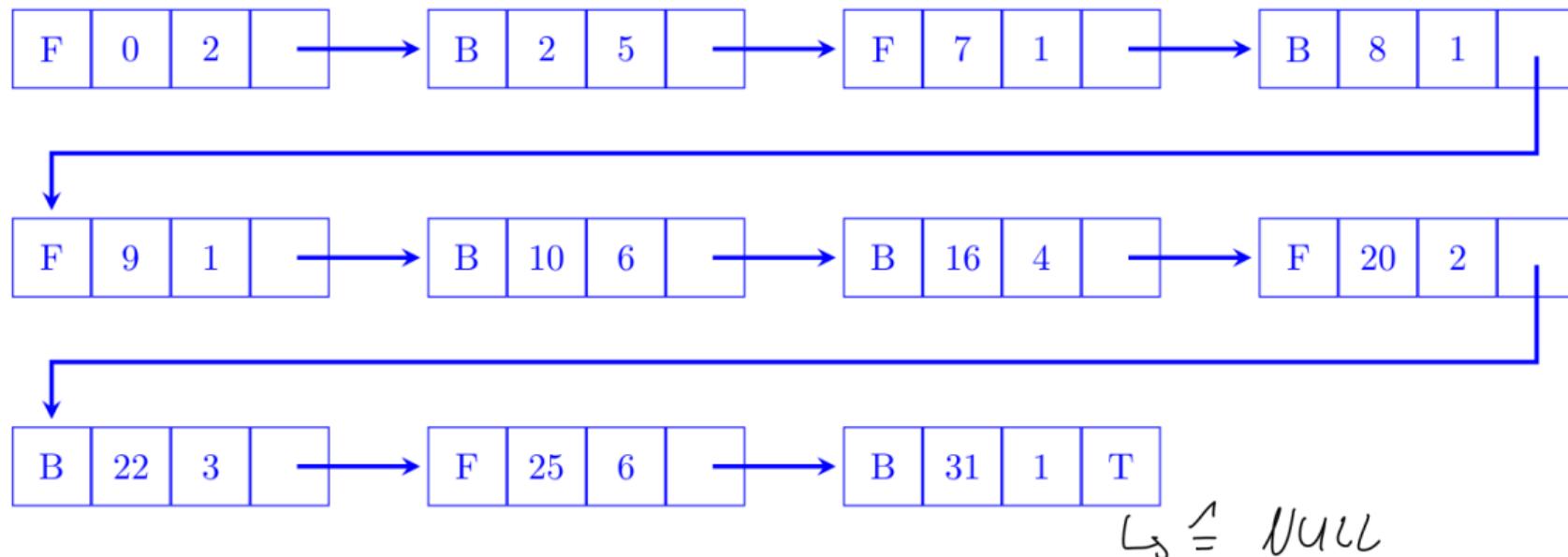
Vier Elemente pro Listeneintrag:

1. **F** (frei) oder **B** (belegt).
2. **Startadresse** dieses Speicherbereichs.
3. **Länge** dieses Speicherbereichs.
4. **Next-Pointer** auf Speicherbereich (modelliert durch Pfeil).



# Aufgabe 3

## Verkettete List



# Die 'Fit'-Verfahren (informell)

- **First-Fit:**

Suche immer von Anfang an und belege den ersten, passenden Speicherplatz.

- **Next-Fit:**

Suche immer von der Stelle aus, in der zuletzt Eingefügt wurde (inklusive die Stelle selbst '! ) und belege den ersten, passenden Speicherplatz.

- **Best-Fit:**

Durchsuche den gesamten Speicher und belege den Speicherplatz, bei dem der Größenunterschied am geringsten ist (Differenz *Speicherplatz – Anfrage* so gering wie möglich).

- **Worst-Fit:**

Durchsuche den gesamten Speicher und belege den Speicherplatz, bei dem der Größenunterschied am höchsten ist (Differenz *Speicherplatz – Anfrage* so hoch wie möglich).

## Aufgabe 4

## First-Fit

1.      2.      3      4.      5.

Anfrage	Liste freier Speicherbereiche				
	100 KiB	400 KiB	250 KiB	200 KiB	50 KiB
1. 30 KiB	70 KiB				
2. 60 KiB	10 KiB				
3. 120 KiB		280 KiB			
4. 20 KiB		260 KiB			
5. 100 KiB		160 KiB			
6. 10 KiB	0 KiB				
7. 250 KiB			0 KiB		

## Aufgabe 4

## Next-Fit

	1.	2.	3	4.	5.
Anfrage	Liste freier Speicherbereiche				
1. 30 KiB	100 KiB	400 KiB	250 KiB	200 KiB	50 KiB
2. 60 KiB	70 KiB				
3. 120 KiB	10 KiB				
4. 20 KiB		280 KiB			
		260 KiB			
		160 KiB			
		150 KiB			
250 KiB			0 KiB		

# Aufgabe 4

## Best-Fit

Anfrage	Liste freier Speicherbereiche				
	100 KiB	400 KiB	250 KiB	200 KiB	50 KiB
30 KiB	(				20
60 KiB	40	)			
120 KiB				80	
20 KiB	(				0
100 KiB			150		
10 KiB	30				
250 KiB	(	150			

## Aufgabe 4

### Worst-Fit

Anfrage	Liste freier Speicherbereiche				
	100 KiB	400 KiB	250 KiB	200 KiB	50 KiB
30 KiB		370			
60 KiB		310			
120 KiB		190			
20 KiB			230		
100 KiB			130		
10 KiB				190	
250 KiB					