

Interprozesskommunikation

IPC

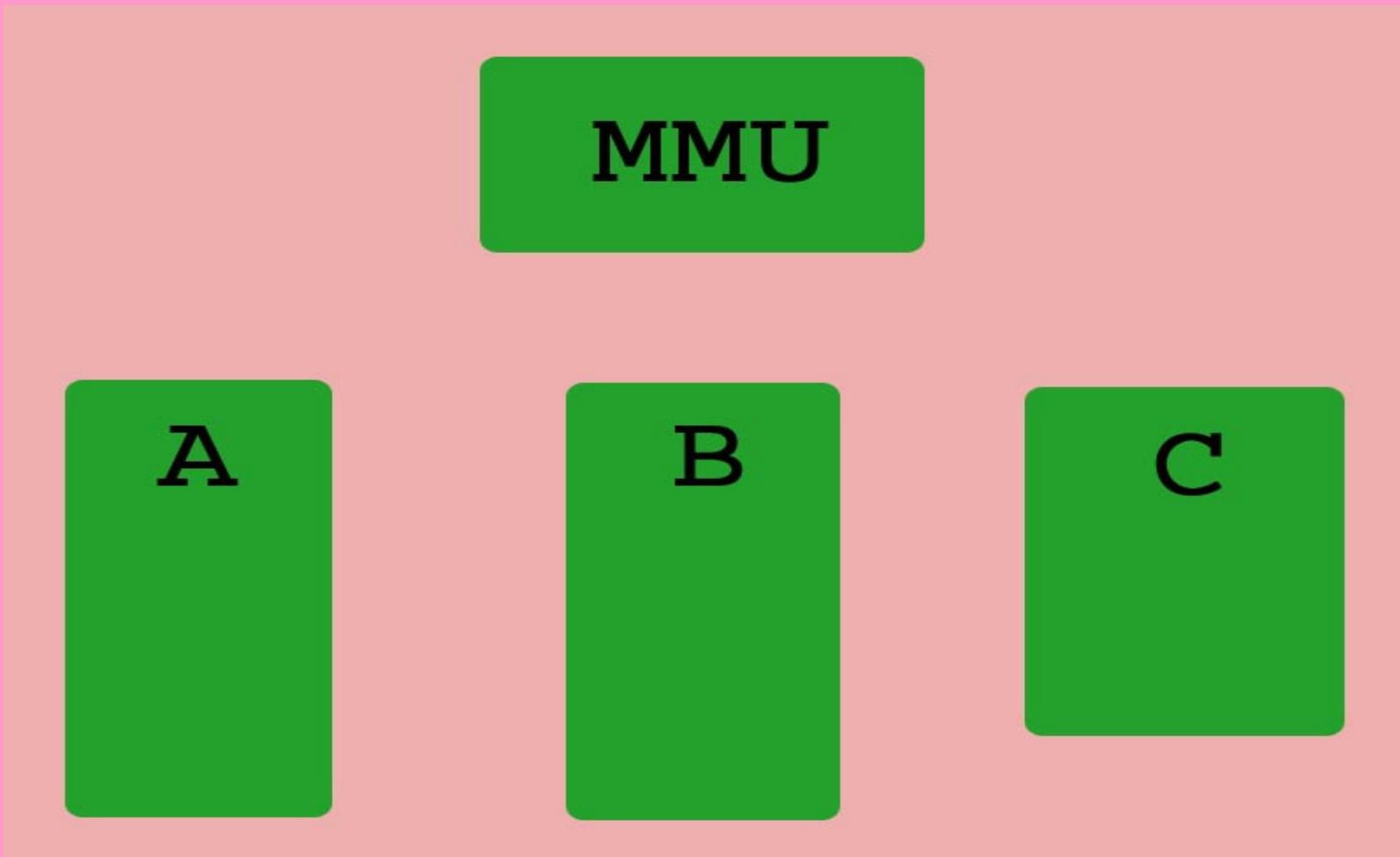
Seminar Konzepte von
Betriebssystem-Komponenten
Denis Koslowski

Gliederung

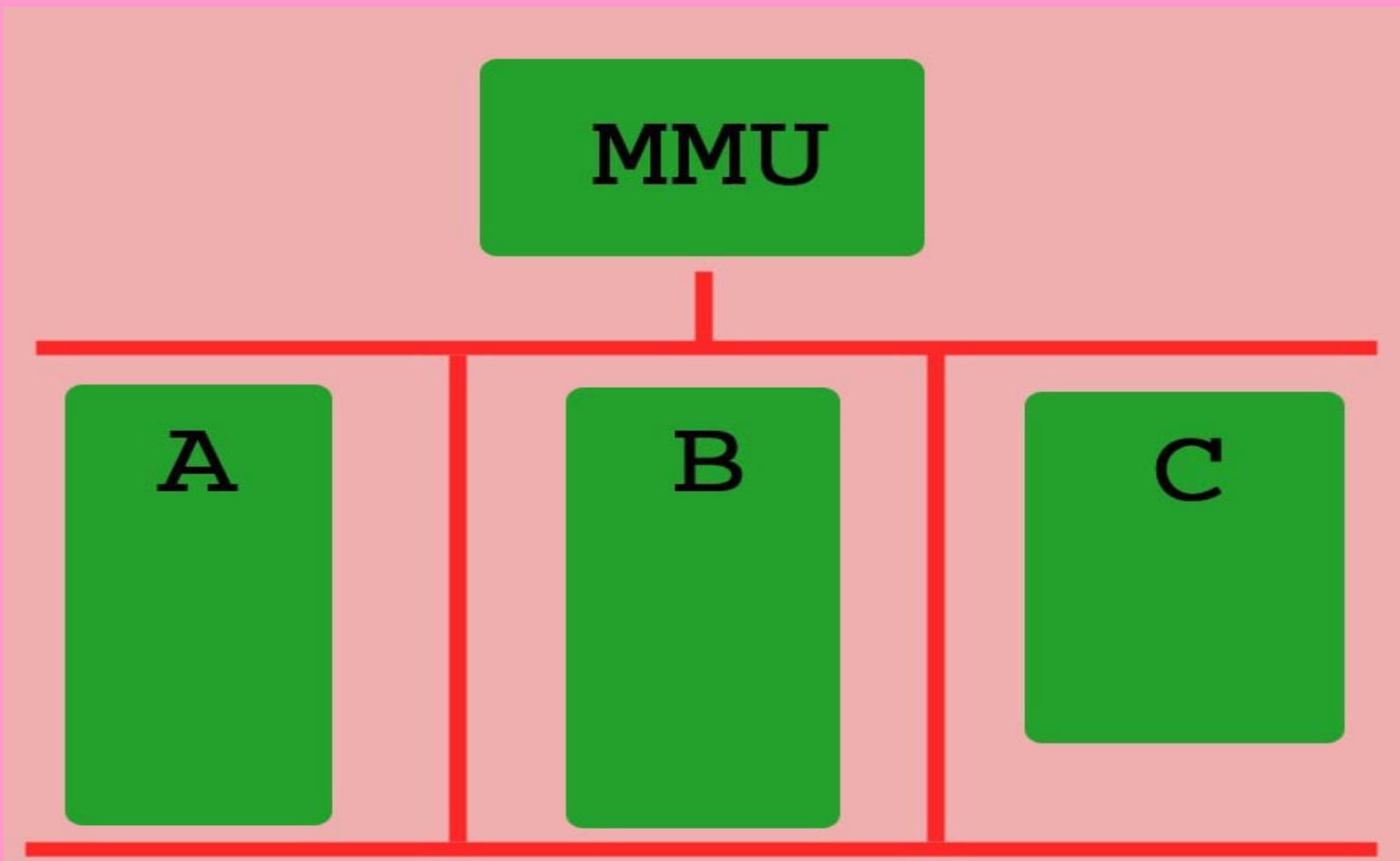
- 1. Was ist IPC?**
- 2. IPC: Datentransfer**
 - 2.1 Begriffe und Konzepte**
 - 2.2 Pipe**
 - 2.3 FIFO-Pipe**
 - 2.4 Shared Memory**
- 3. IPC: Synchronisation**
 - 3.1 Probleme**
 - 3.2 Semaphore**
 - 3.3 Lock Files**
- 4. Zusammenfassung**

1.Was ist IPC?

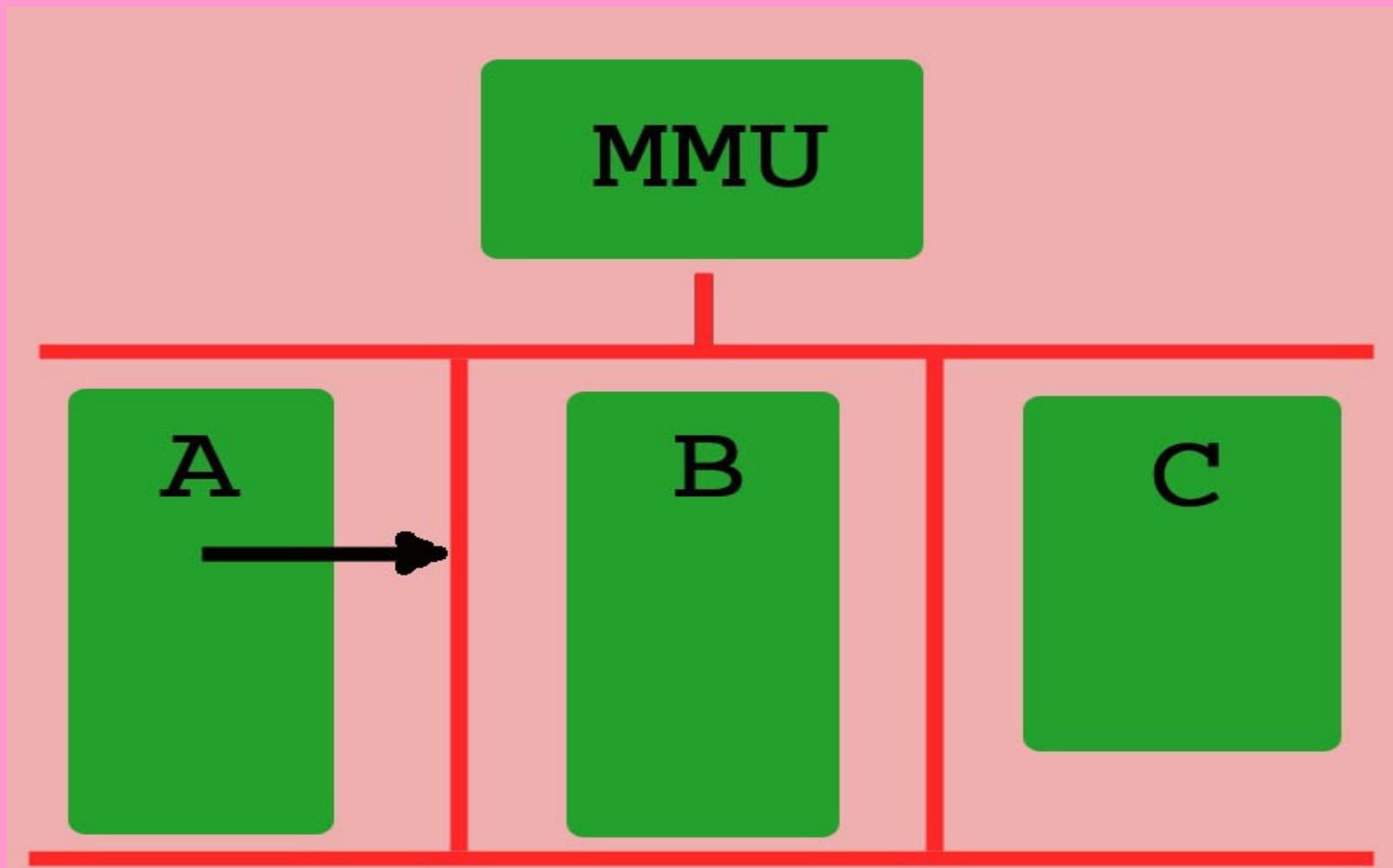
Linux - Multitasking-System



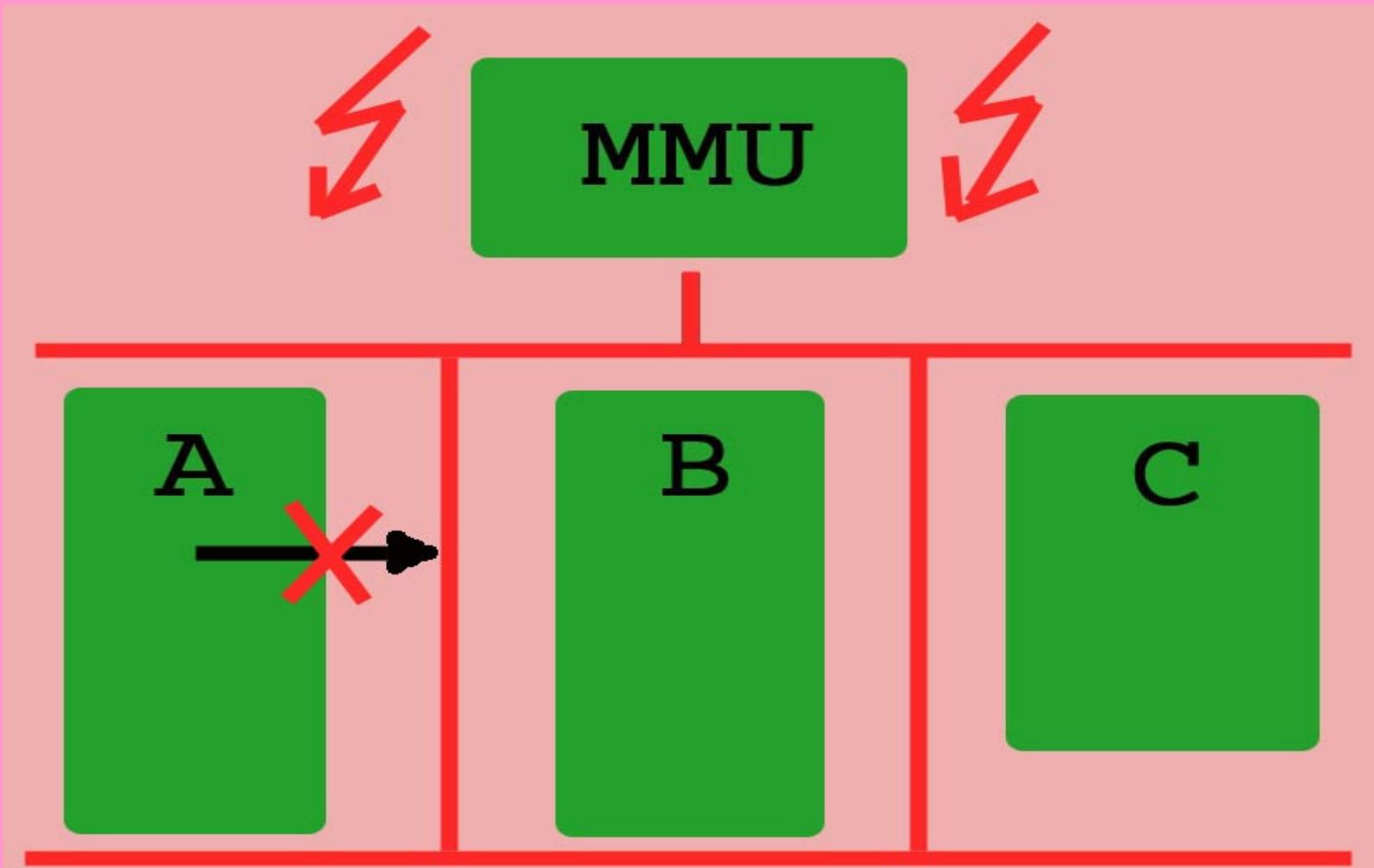
Prozesse wissen nichts von einander



Speicherschutz



Speicherverwaltungseinheit = MMU



**Wann braucht
ein Prozess einen „Tunnel“
zu einem anderen Prozess?**

Es gibt 4 Fälle

1. Fall

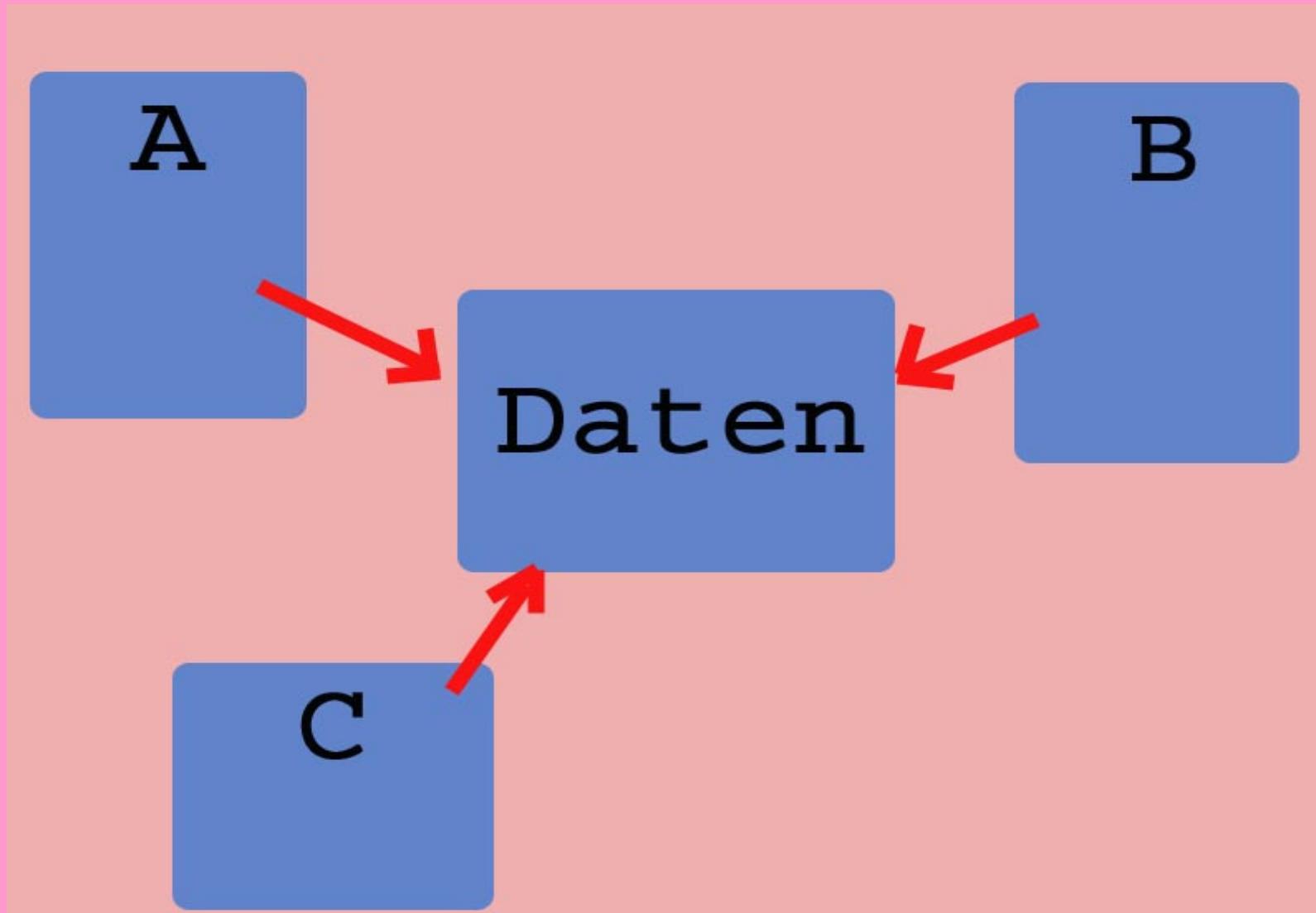
A

B

Daten

C

1. Fall



2. Fall

A

läuft

B

wartet
auf A

2. Fall

A

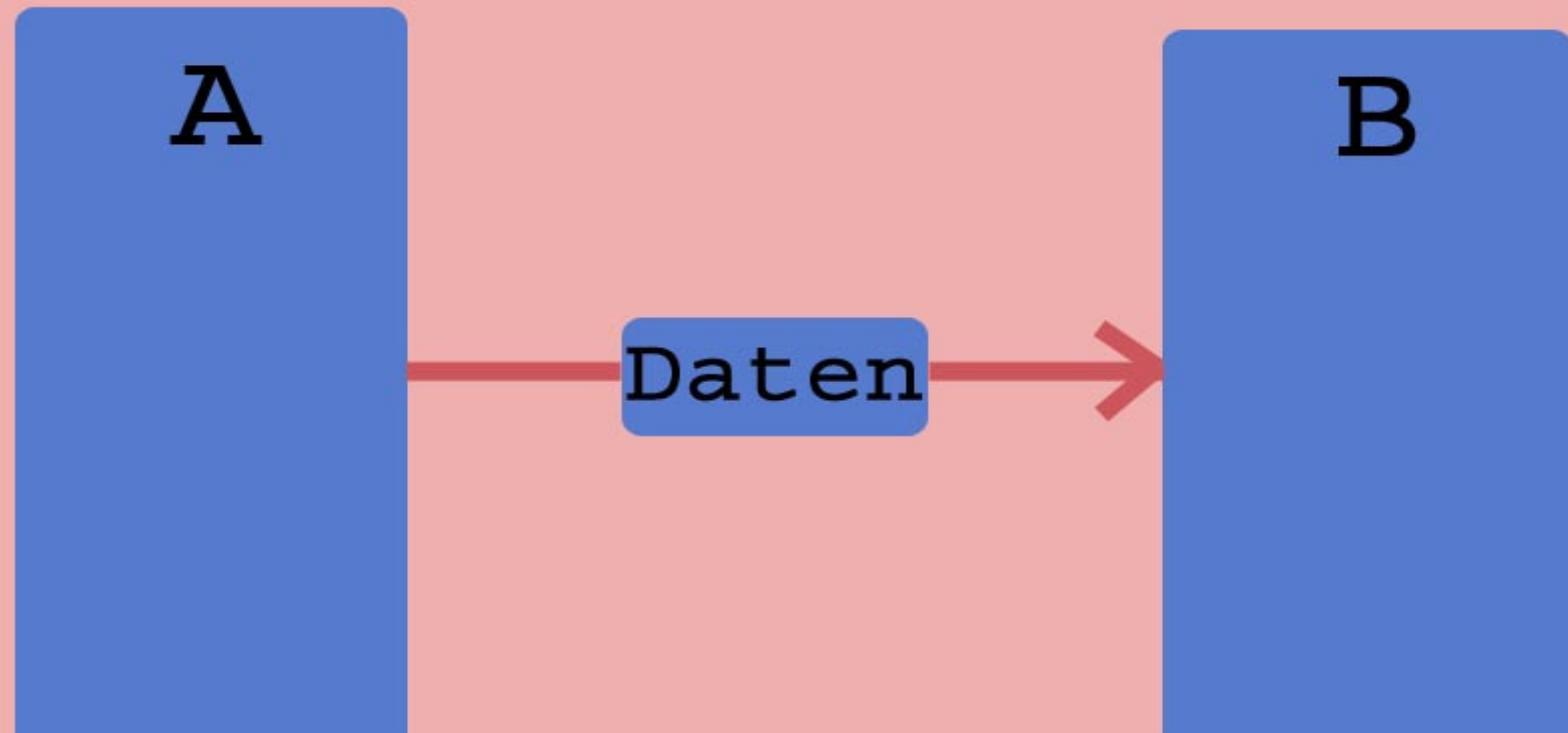
läuft

?

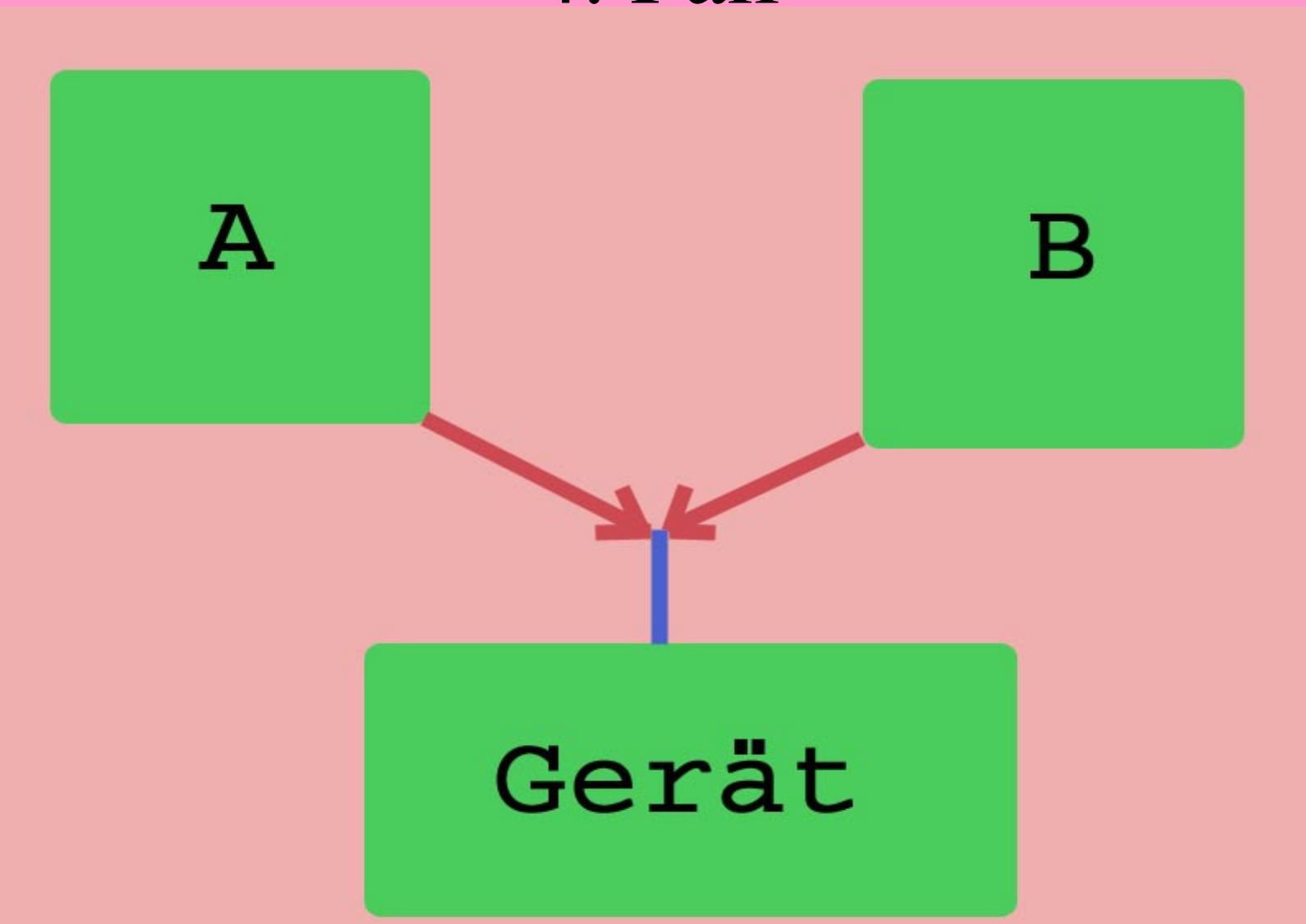
B

wartet
auf A

3. Fall



4. Fall



1.Was ist IPC?

Antwort:

**Mechanismen und Techniken, die den
Datentransfer zwischen Prozessen und die
Synchronisation ihrer Aktivitäten
ermöglichen**

Fragen?

2. IPC: Datentransfer

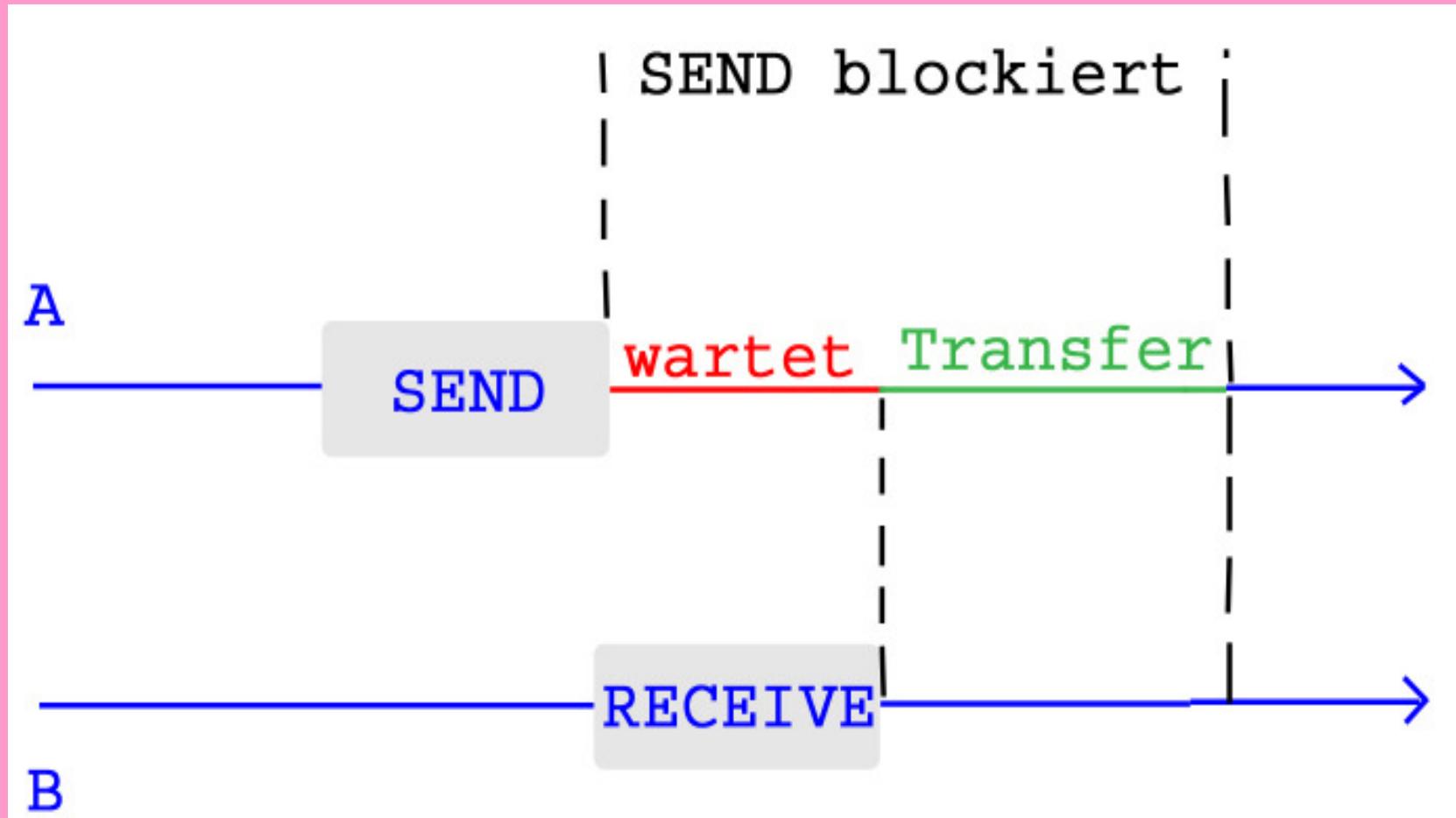
2 Arten:

-gemeinsame Variablen

-gemeinsamer Kommunikationskanal

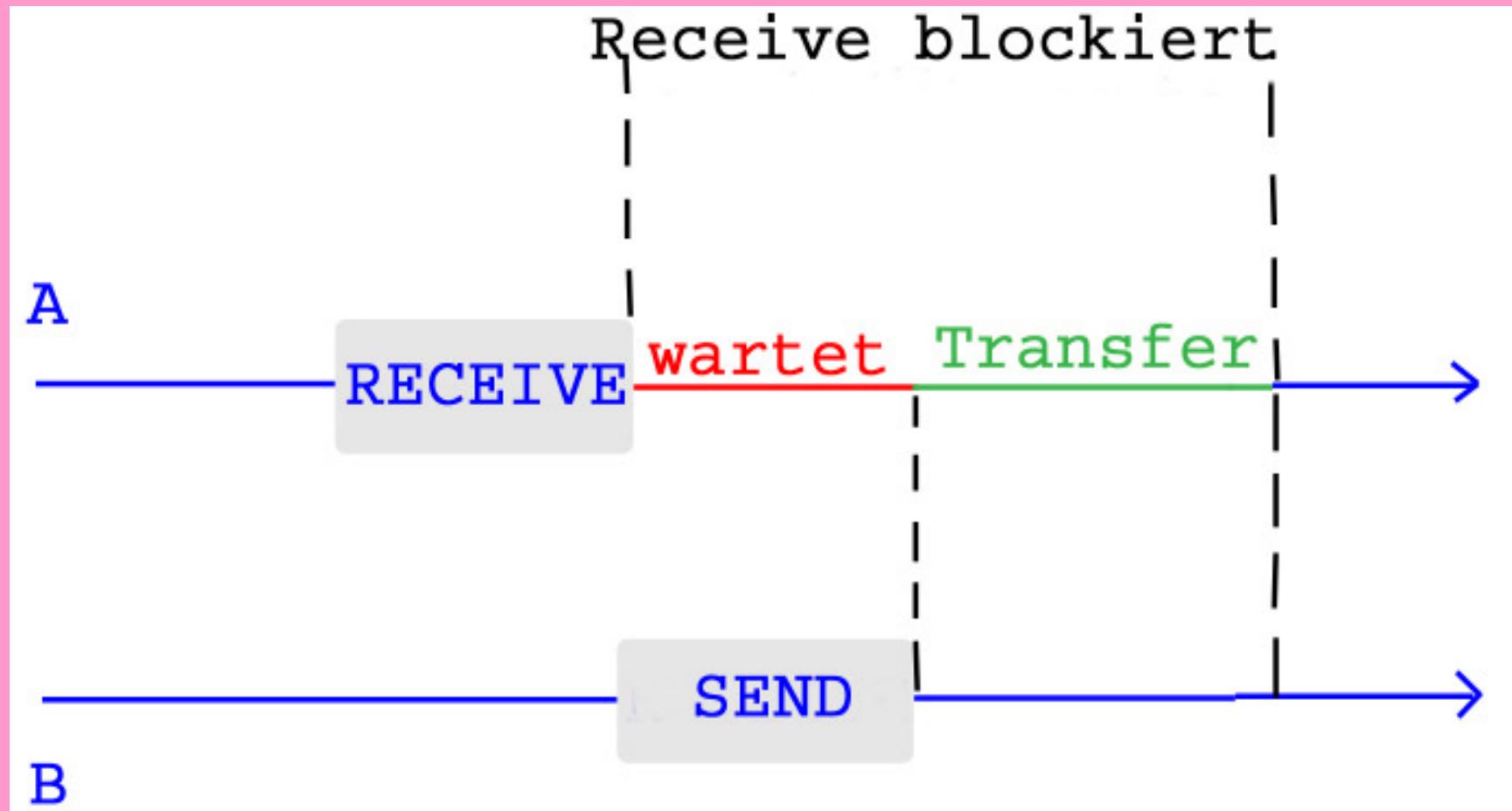
2.1

Synchron und blockierend



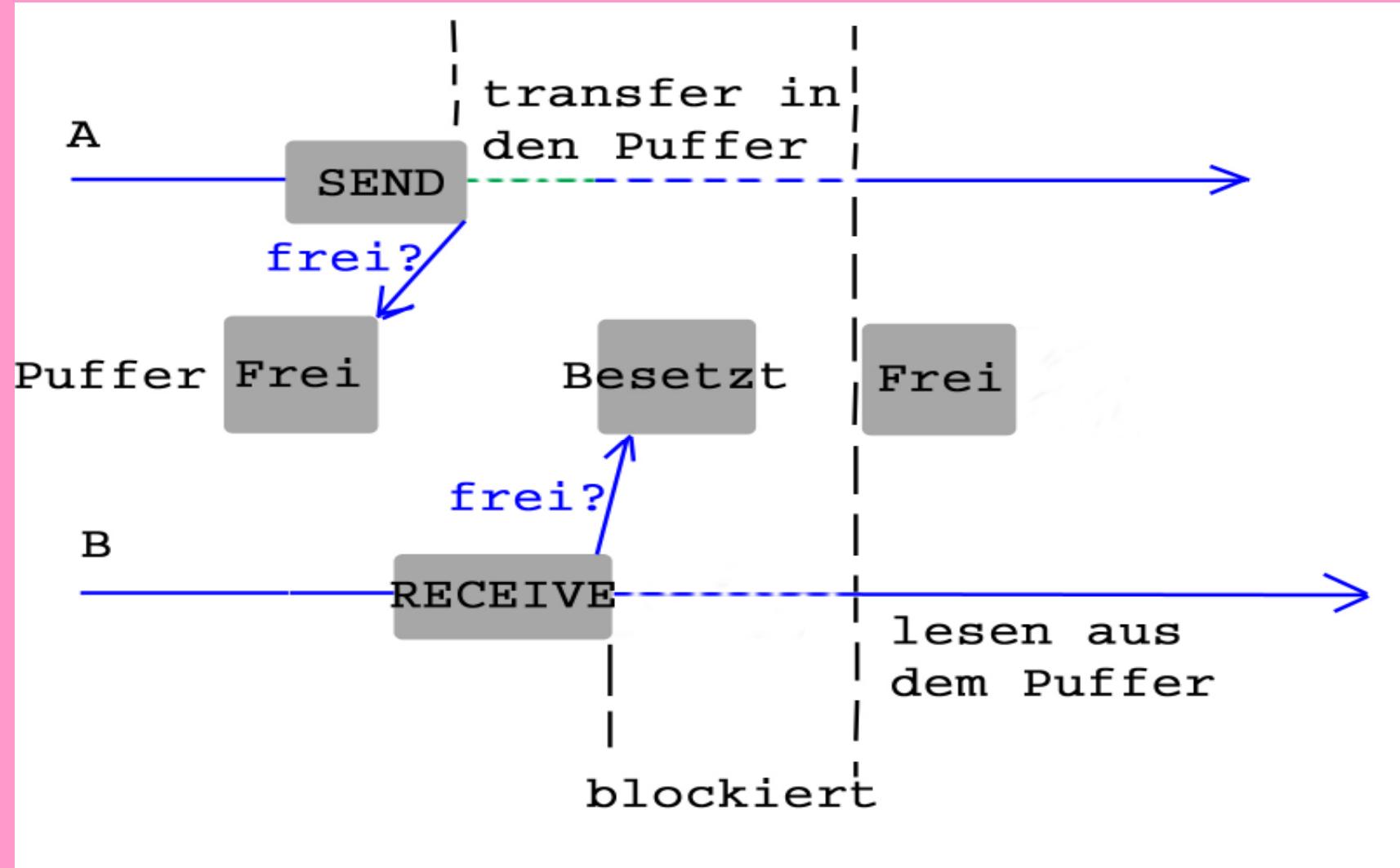
2.1

Synchron und blockierend



2.1

Asynchron und blockierend



2.2 Pipes

Eine Pipe ist unidirektionaler Kommunikationskanal zwischen zwei verwandten Prozessen

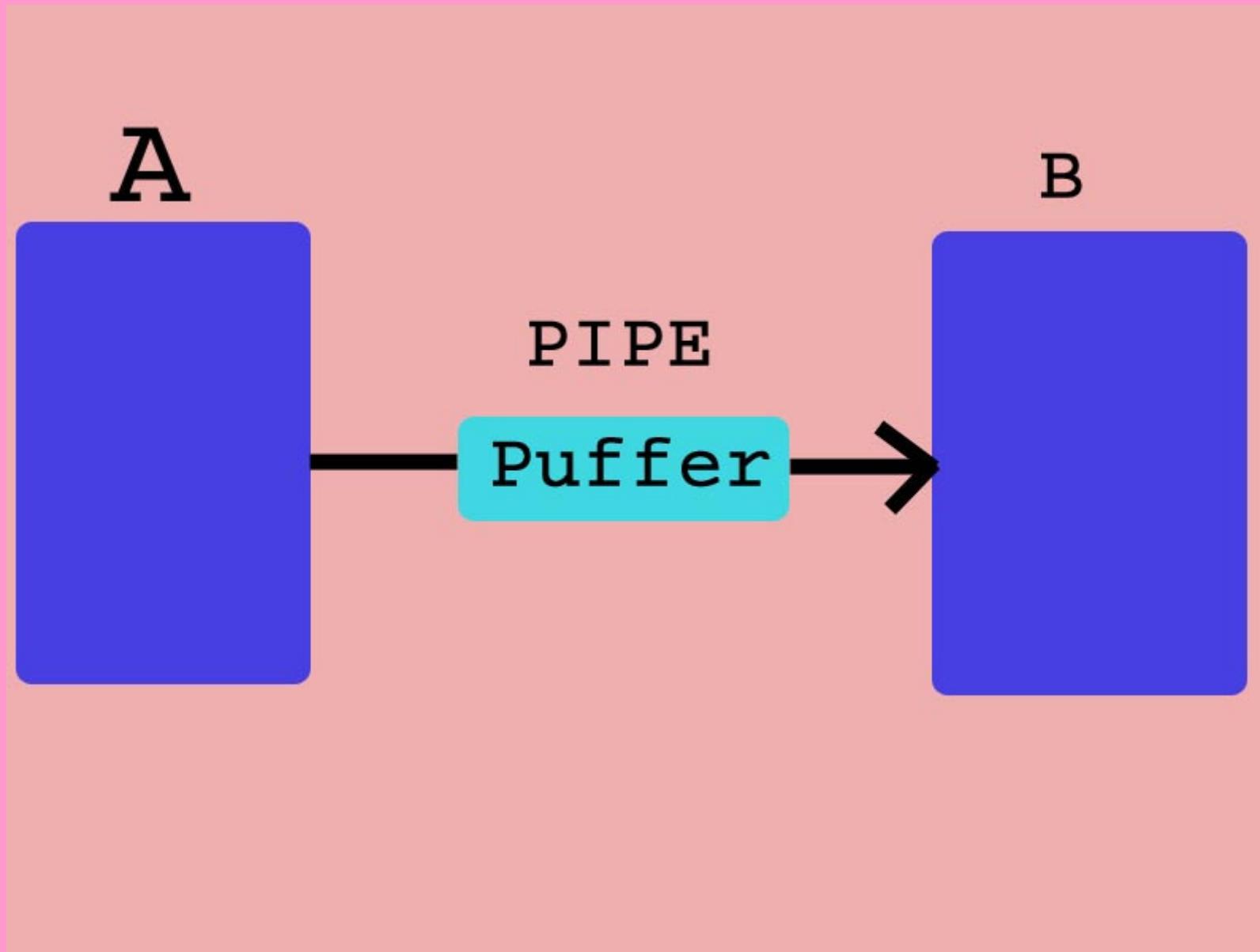
```
you@ host > ps ax | less
```

2.2 Pipes

**Flusskontrolle, da die Kapazität einer
Pipe begrenzt ist**

4-32 Kbyte

Konstante PIPE_BUF in limits.h



2.3 benannte Pipes

FIFO-Pipes

2.3 benannte Pipes

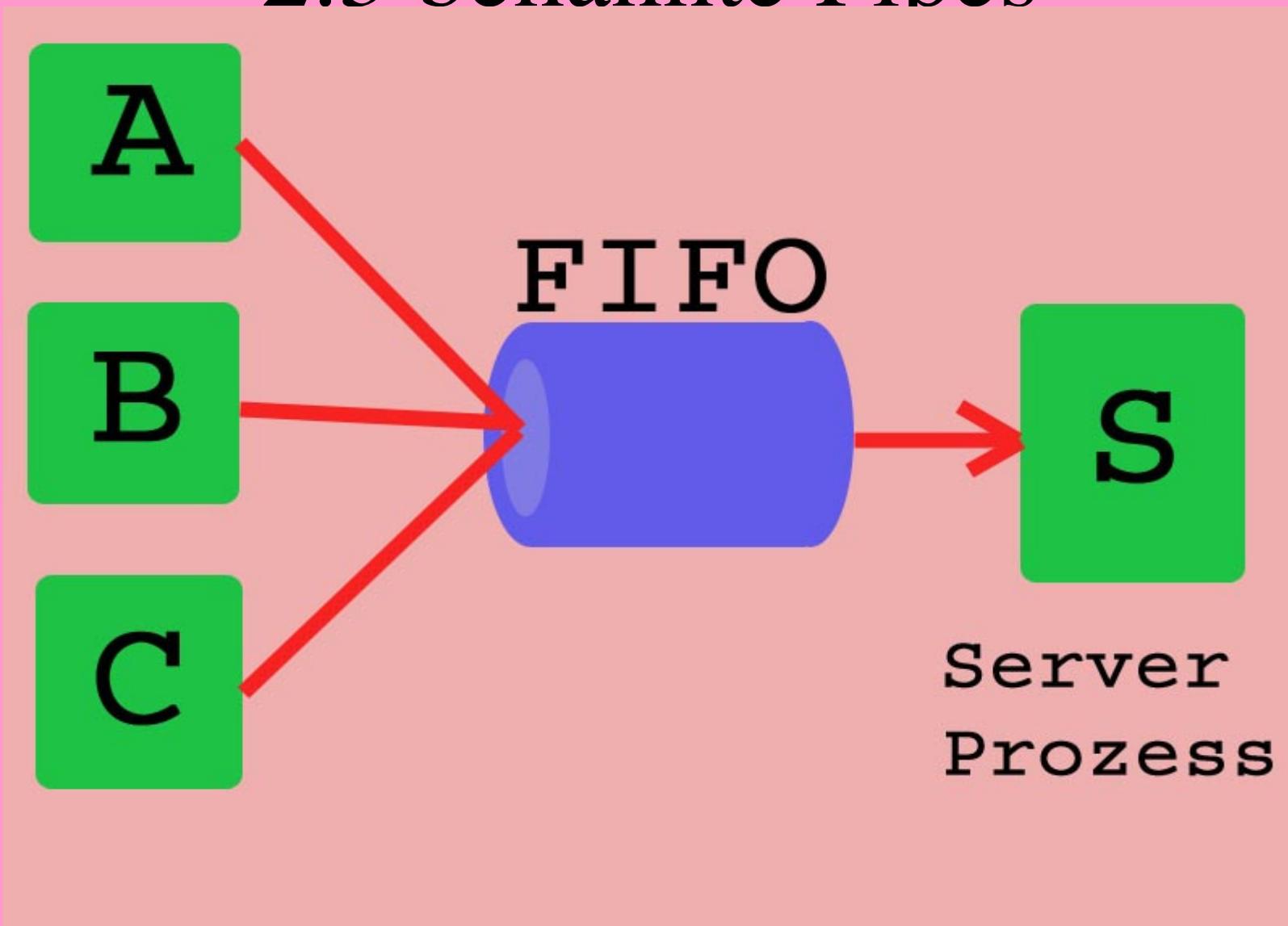
- Kommunikation der völlig fremden Prozessen
- FIFO wird als Datei repräsentiert
- Zugriffsrechte müssen vergeben werden

you@host > mkfifo fifo1

you@host > ls -l

prw-r--r- 1 tot users 0 2005-05-30 10:30 fifo1

2.3 benannte Pipes



2.4 Shared Memory

Gemeinsamer Speicher

2.4 Shared Memory

- gemeinsamer Speicherbereich anlegen
- den Bereich anderen Prozessen bekannt machen
- Zugriffsrechte vergeben
- Synchronisation ist notwendig

3. IPC: Synchronisation

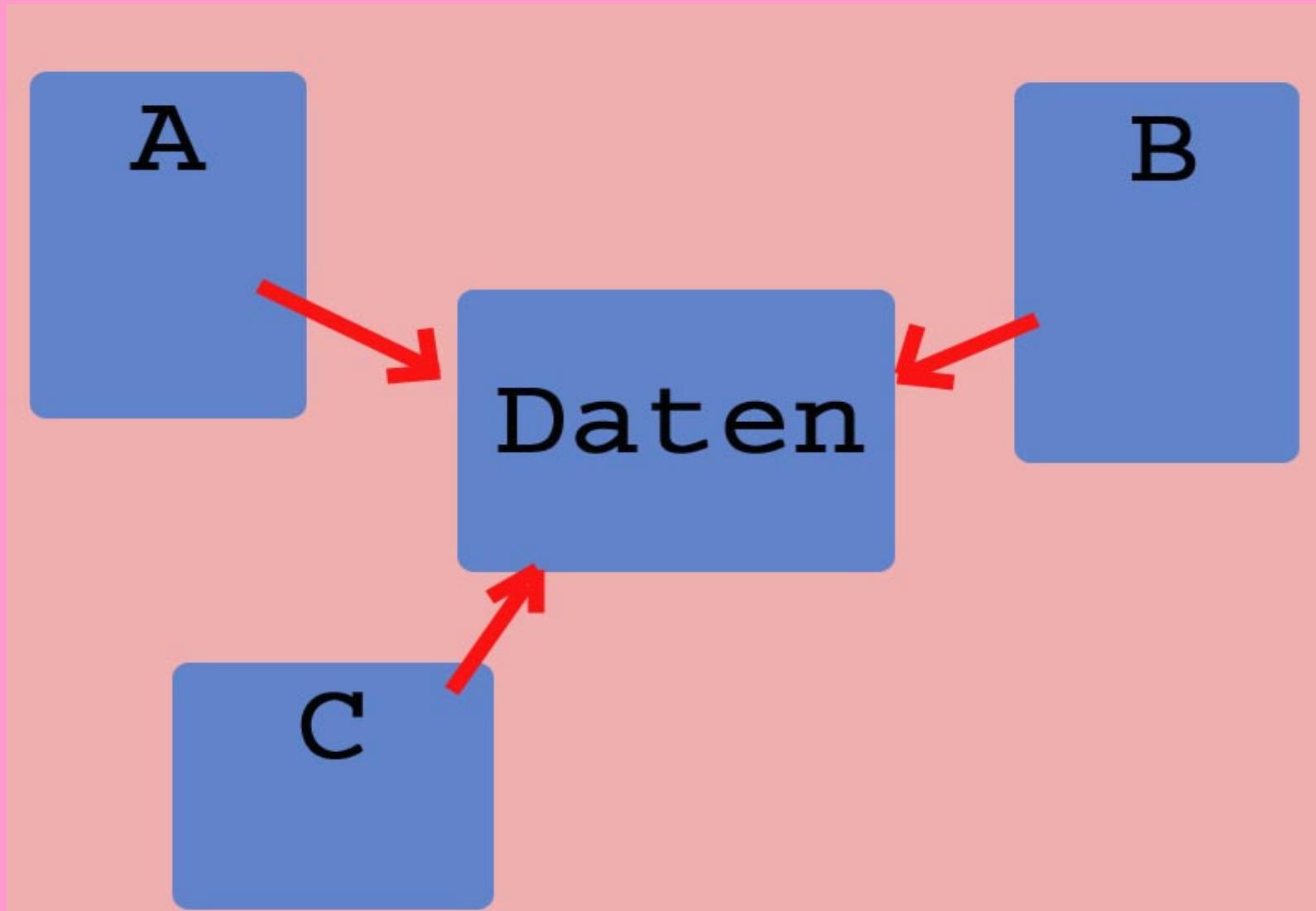
3.1 Probleme

3.1 Probleme

Ziel: eine bestimmte Reihenfolge
der Aktivitäten der Prozesse, Vermeidung
von Deadlocks und Livelocks

Notwendig in den Fällen 1, 2 und 4

1. Fall



2. Fall

A

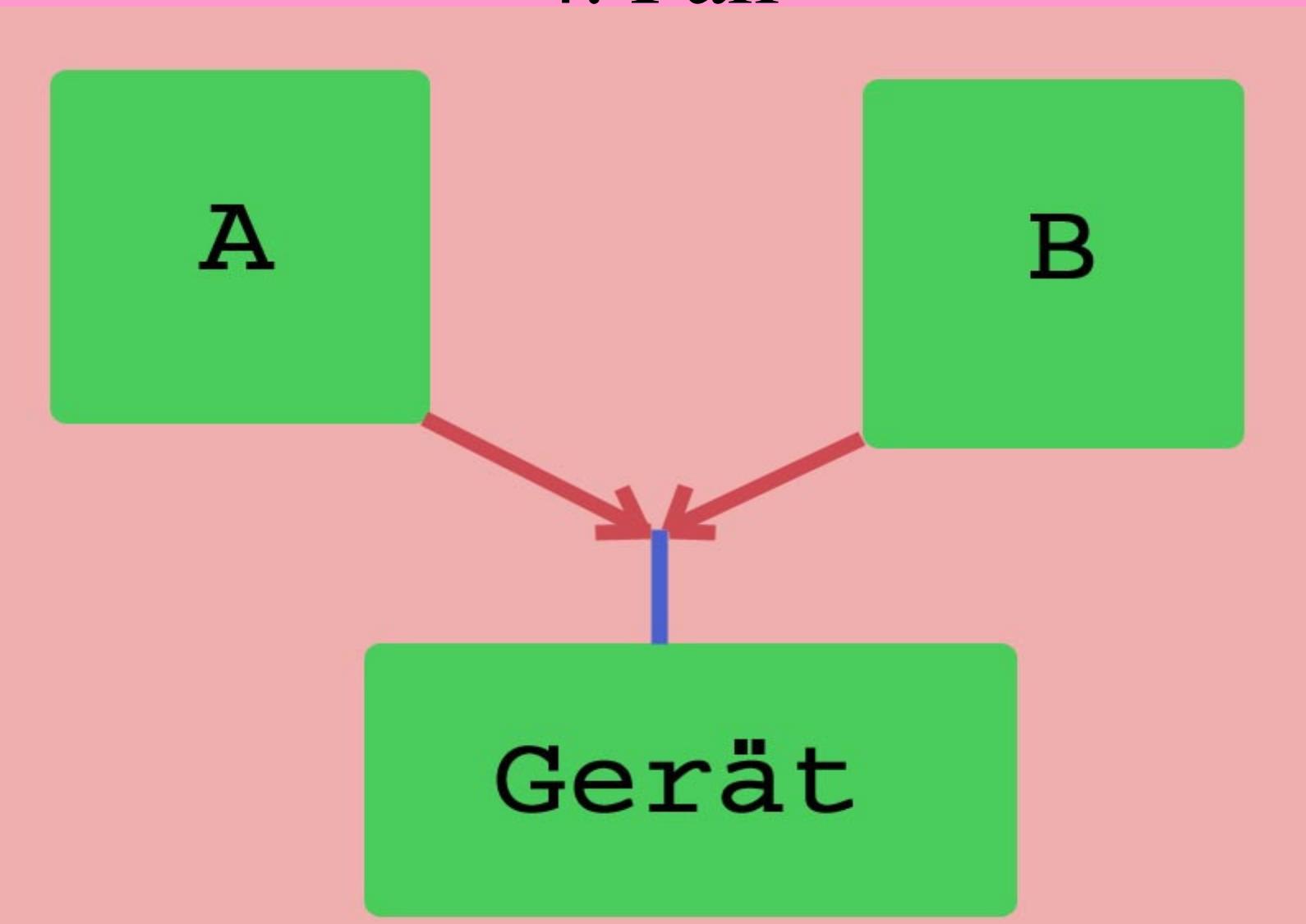
läuft

?

B

wartet
auf A

4. Fall



3.1 Probleme

B: "Ich mache weiter,
nur wenn "A" mir
etwas sagt!"

A

DEADLOCK

B

A: "Ich mache weiter,
nur wenn "B" mir etwas
sagt!"

3.1 Probleme

Lifelock:

- Prozesse blockieren nicht
- machen aber keine Fortschritte

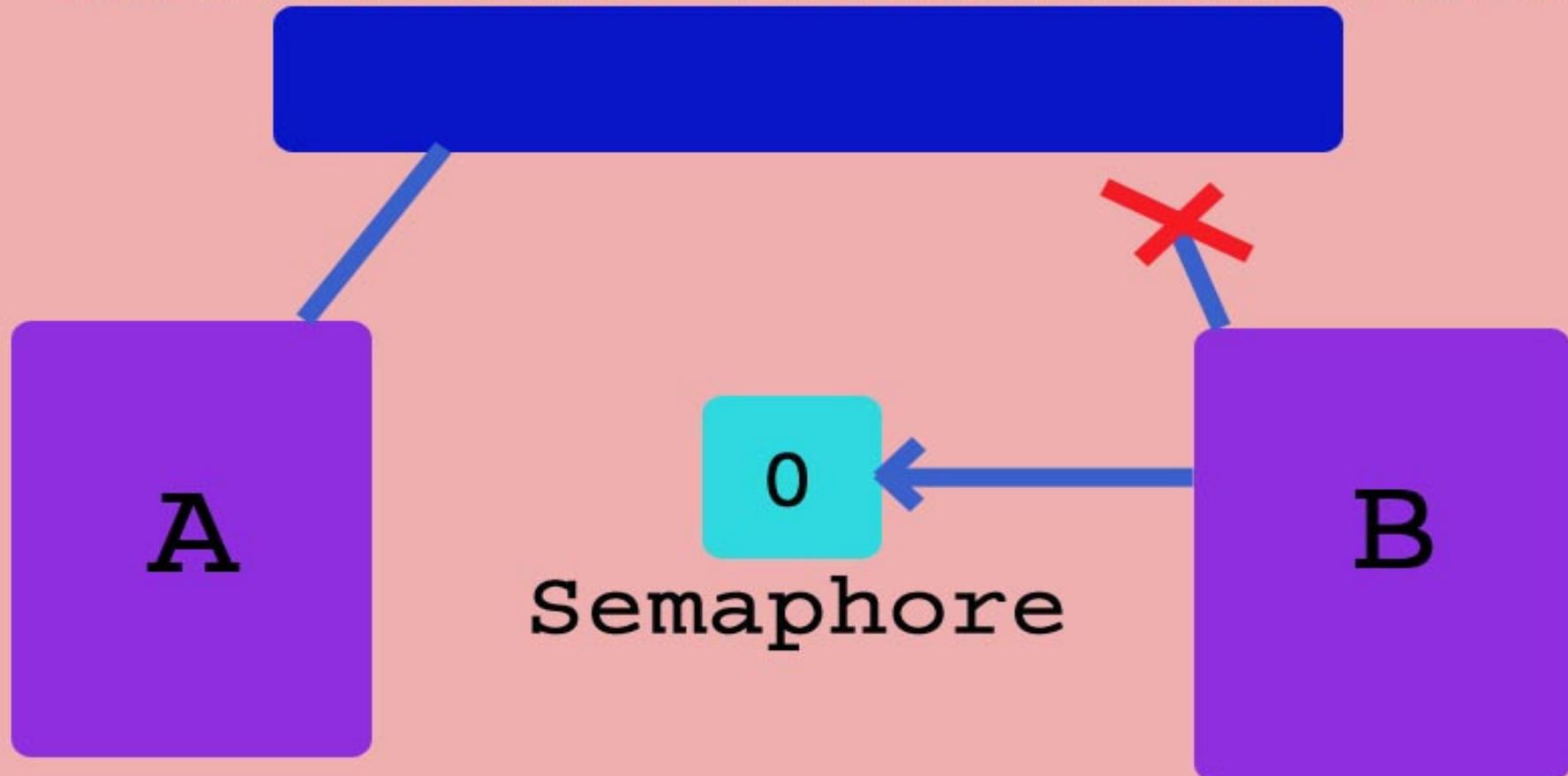
Lifelock ist nicht eindeutig erkennbar

3.2 Semaphore

3.2 Semaphore

- Zugriffsvariablen
- bestimmte Funktionen
- Synchronisation beim Zugriff auf gemeinsam benutzte Betriebsmittel/Datenstrukturen

kritisches Betriebsmittel



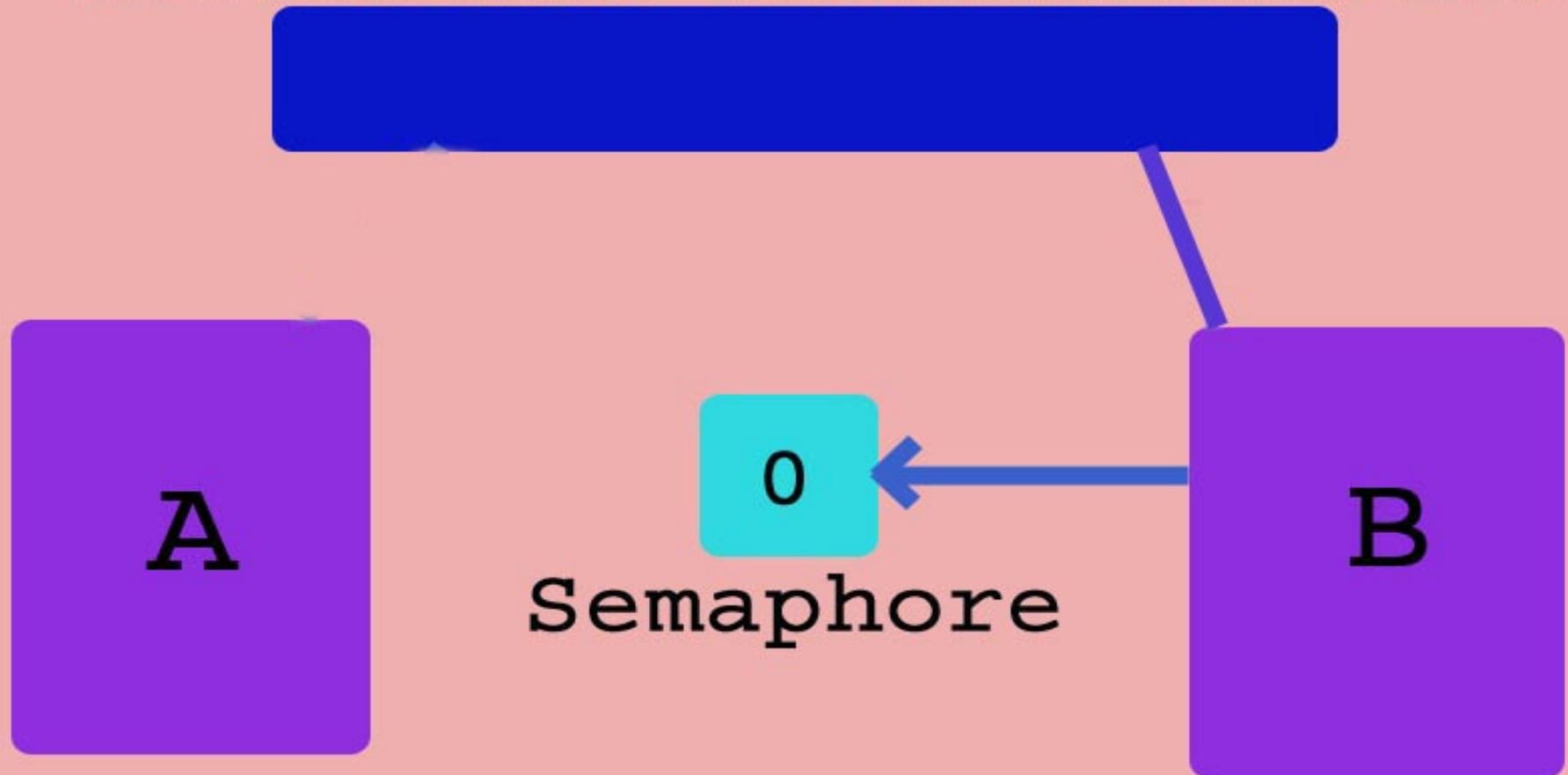
kritisches Betriebsmittel



kritisches Betriebsmittel



kritisches Betriebsmittel



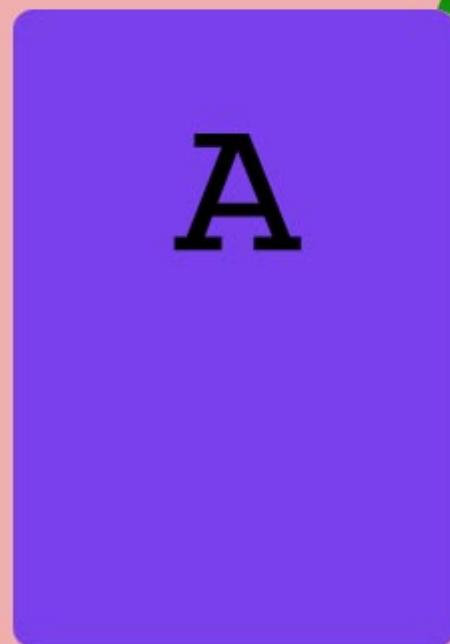
3.3 Lock Files

Sperrdateien

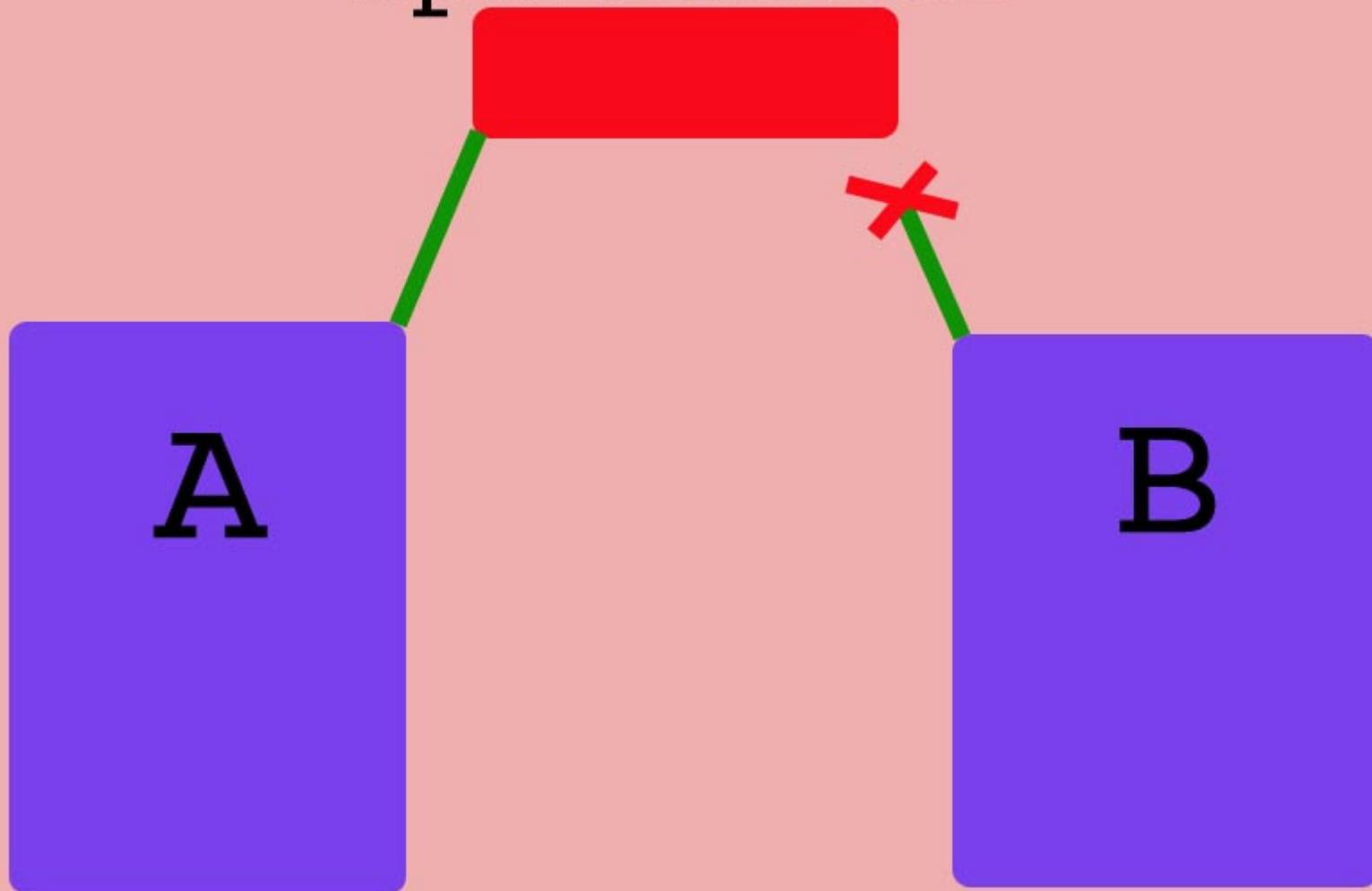
3.3 Lock Files

- eine primitive Form der IPC
- mehrere Prozesse versuchen eine Sperrdatei mit Schreibrechten nur für sich selbst anzulegen
- falls die Sperrdatei schon existiert – warten bis die Datei gelöscht wird

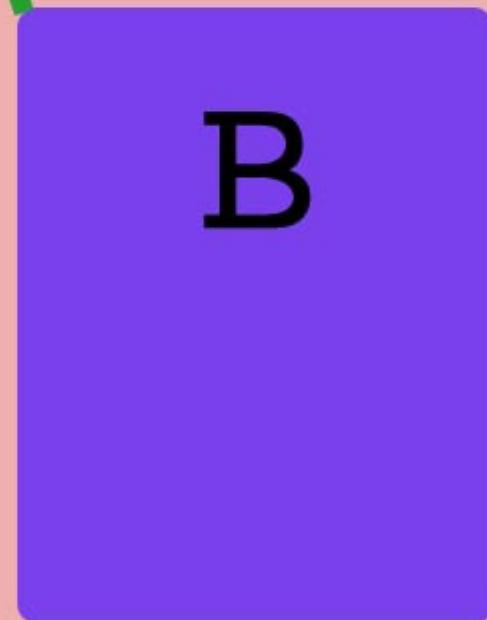
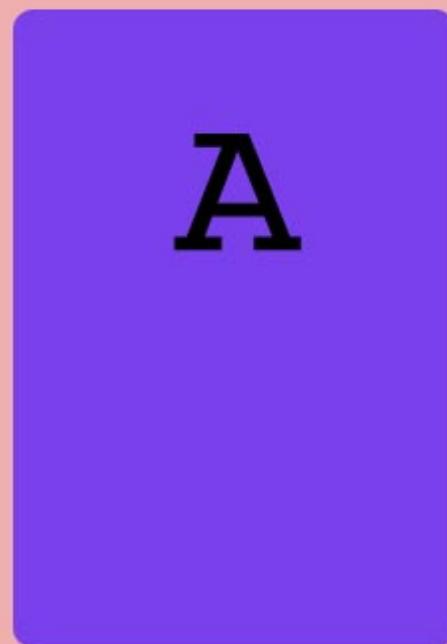
Sperrdatei



Sperrdatei



Sperrdatei



3.3 Lock Files

Nachteile:

- Superuser darf alle Dateien schreiben
- aktives Warten bis die Datei freigegeben ist
- keine Garantie

4. Zusammenfassung

4. Zusammenfassung

IPC ist Transfer von Daten zwischen Prozessen und die Synchronisation dieses Transfers

Der Transfer erfolgt über gemeinsame Speicherbereiche(Variablen) oder über einen gemeinsamen Kommunikationskanal.

Die Kommunikation kann synchron oder asynchron verlaufen.

4. Zusammenfassung

Pipe - unidirektional, verwandte Prozesse

FIFO- unidirektional, fremde Prozesse

Shared Memory - Synchronisation notwendig

Synchronisation – bestimmte Reihenfolge

Lifelock nicht eindeutig erkennbar

Semaphore – Synchronisation beim Zugriff
auf gemeinsam benutzte Betriebsmittel

Sperrdateien – einfach, aber hat viele
Nachteile

Vielen Dank
für Ihre Aufmerksamkeit!