

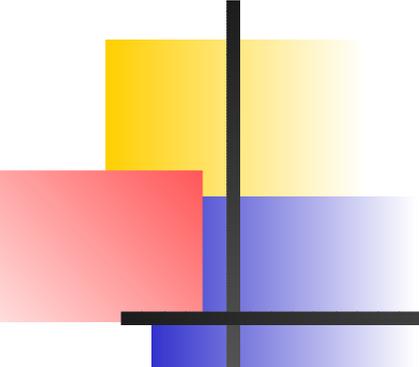
Konzepte von Betriebssystem-Komponenten

UNIX

Christian Kollee

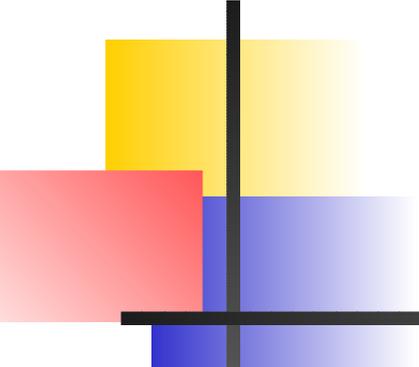
`sichkoll@cip.informatik.uni-erlangen.de`

Friedrich-Alexander-Universität Erlangen-Nürnberg



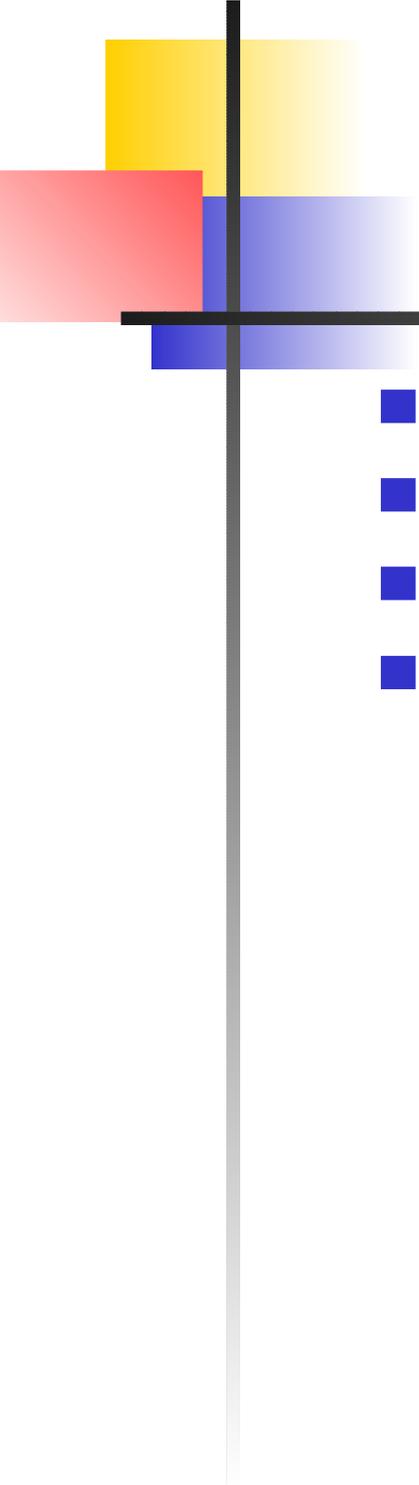
Überblick

- UNIX ist eine Betriebssystemfamilie
- über 30 Jahre alt
- Multiuser/Multiprogramming
- in C geschrieben
- verschiedenste Versionen
- standardisiert: POSIX



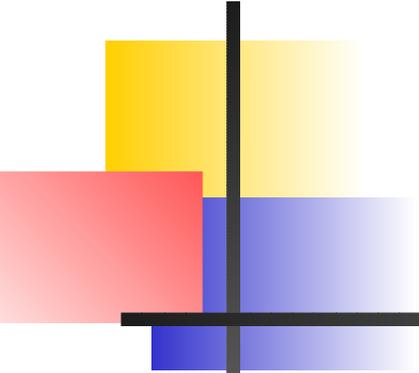
Der UNIX-Kern - Ein Stückchen Code

- ursprünglich: klein und effizient
- monolithischer Ansatz
 - + effiziente Kommunikation innerhalb des Kerns
 - fehlende Modularität
- Modularisierungsversuche
 - defekte oder korruptierte Module
 - fehlende Schutzmaßnahmen



Prozesse (1) - Der Aufbau

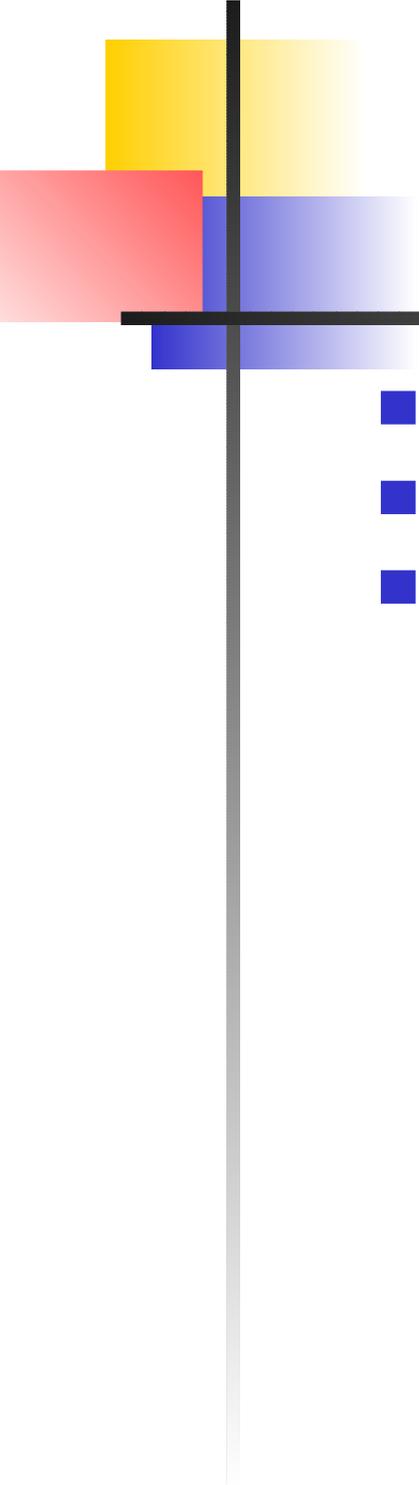
- text section: übersetzter Programmtext
- data section: globale und statische Variablen
- process stack: temporäre Variablen
- program counter: zeigt auf den nächsten auszuführenden Befehl



Prozesse (2) - Die Erzeugung

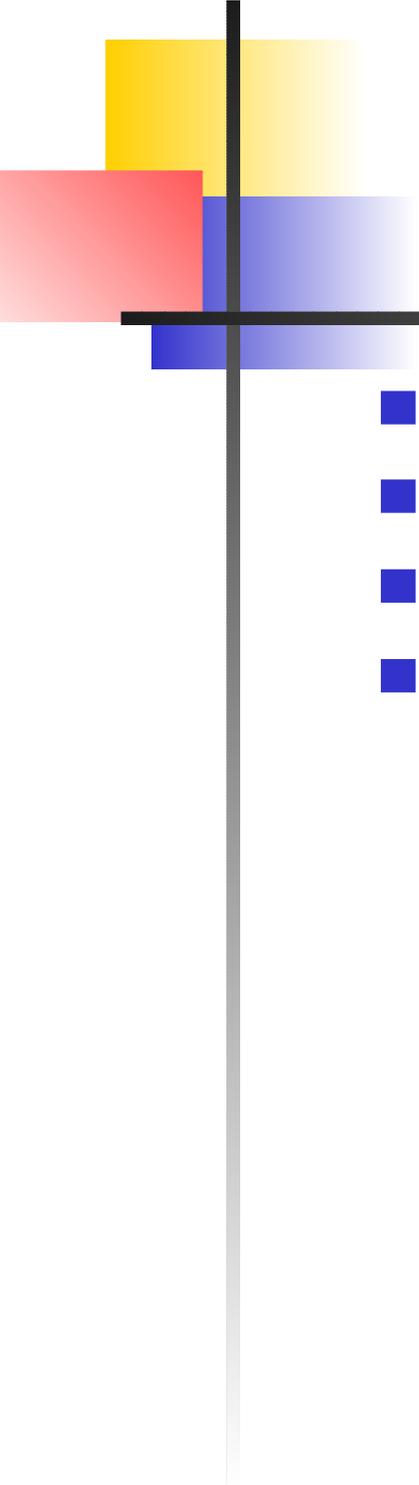
- fork-System Call
- Erstellung eines PID
- Prozesse arbeiten unabhängig voneinander
- evtl. `execve`-System Call

```
...  
void main(int argc, char *argv[]) {  
    int pid;  
    pid = fork();  
    if (pid < 0) { /* Fehler! */  
        ... }  
    else if (pid == 0) { /* Code fr das Kind */  
        ... }  
    else { /* Code fr den Vater */  
        ... }  
}
```



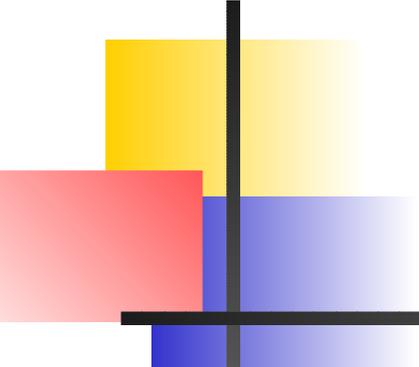
Prozesse (3) - Das Ende

- eigenständige Beendigung durch *exit()*
- Rückgabe des Exitstatus (Integer 0 bis 255)
- Beendigung durch Signal von einem Mitglied der Prozessgruppe
 - Nachfahren
 - Vorfahren



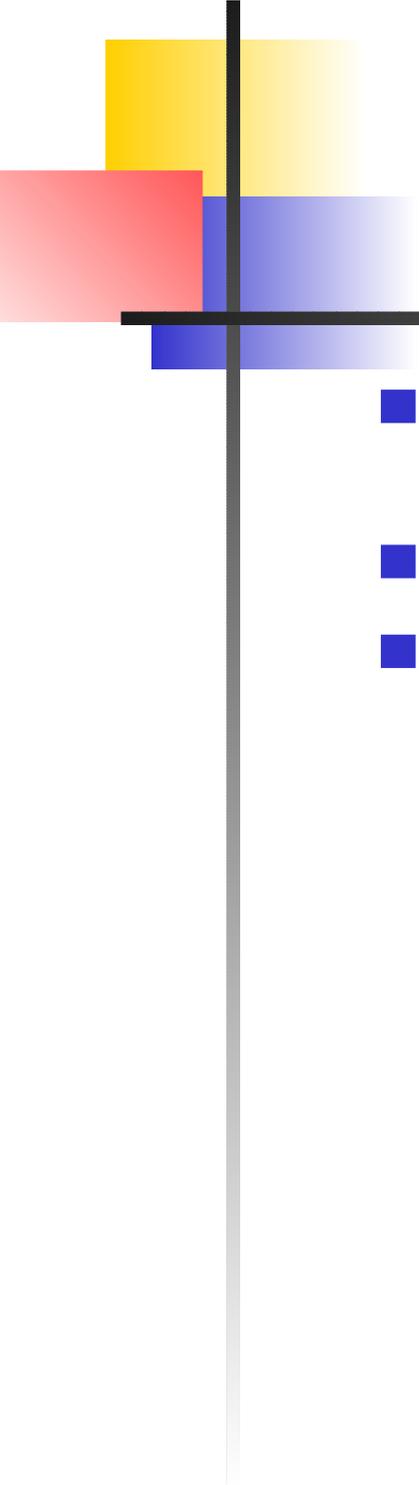
Prozesse (4) - Interprozesskommunikation

- Sockets
- Message Queues
- Signale
- (Shared Memory)



Prozesse (5) - Semaphoren

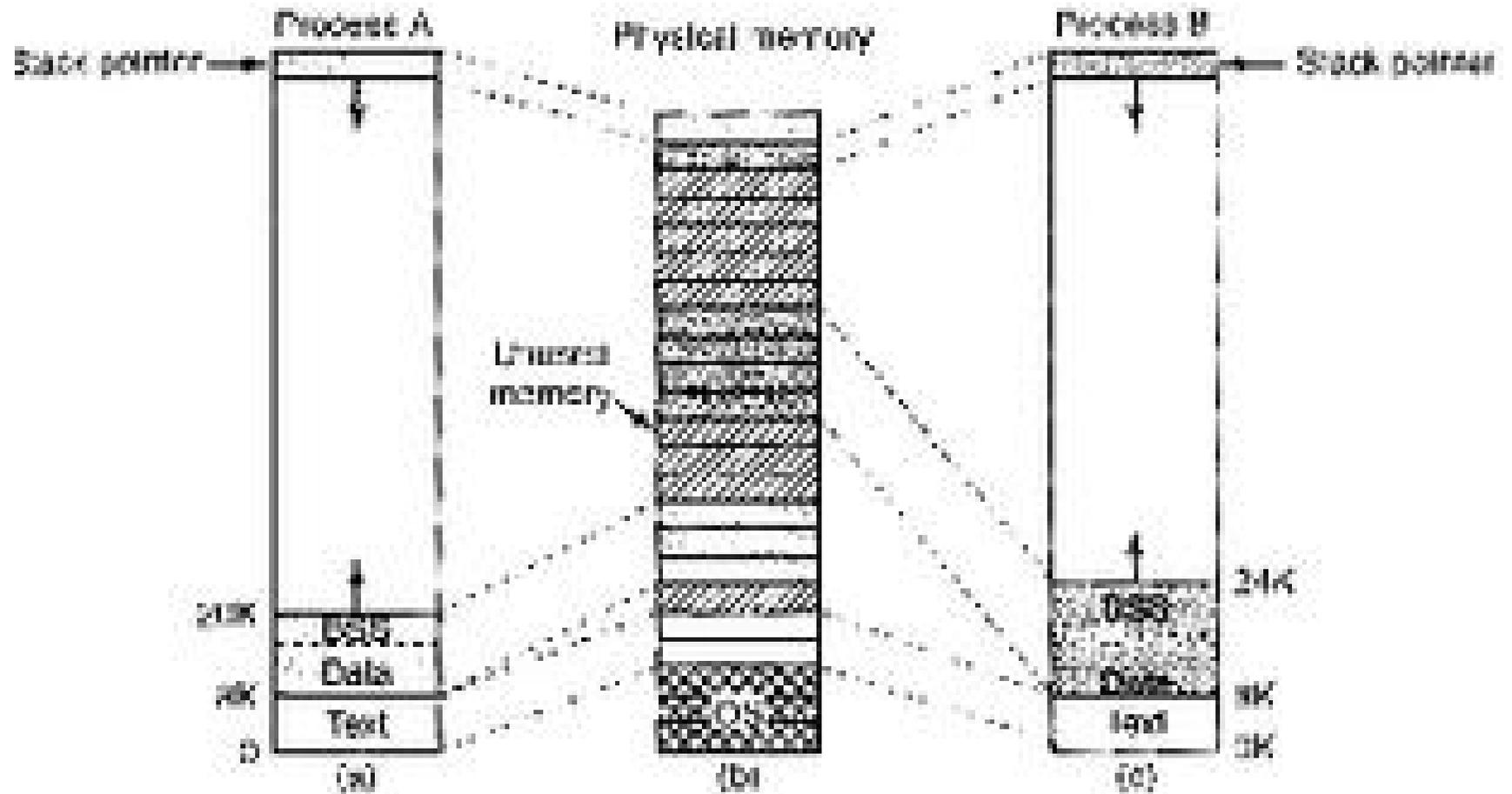
- 1965 von Dijkstra vorgeschlagen
- Verhinderung von
 - Erzeuger-Verbraucher-Problem
 - Leser-Schreiber-Problem
- UNIX stellt Funktionen zur Verfügung

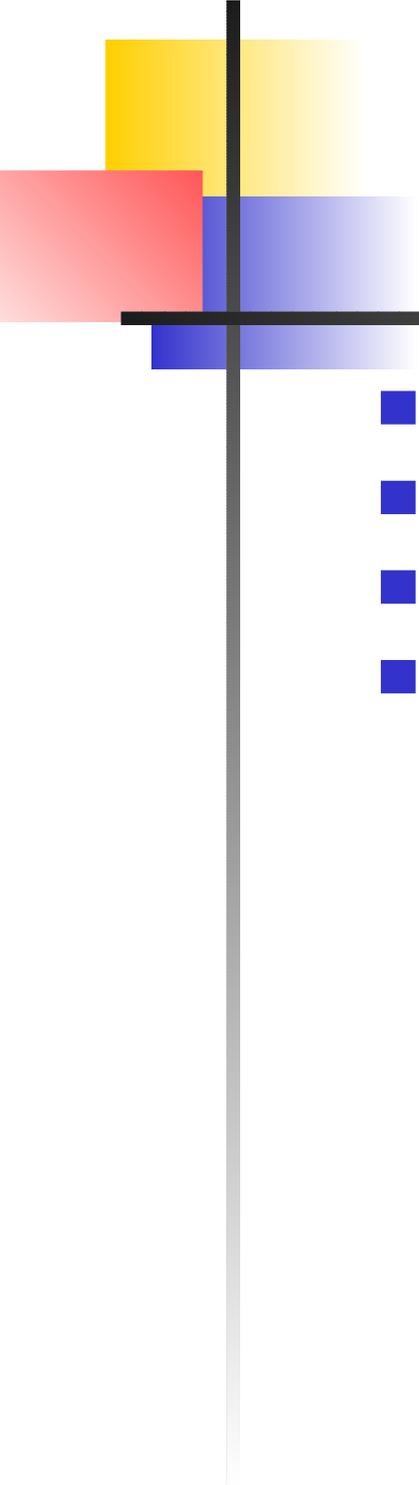


Speicherverwaltung (1) - Shared Text Segments

- Multiprogramming: evtl. mehrmals das gleiche Programm im Speicher
- Textsegmente sind identisch
- nur ein Textsegment im Speicher (virtuelle Adressen)

Speicherverwaltung (2) - Shared Text Segments

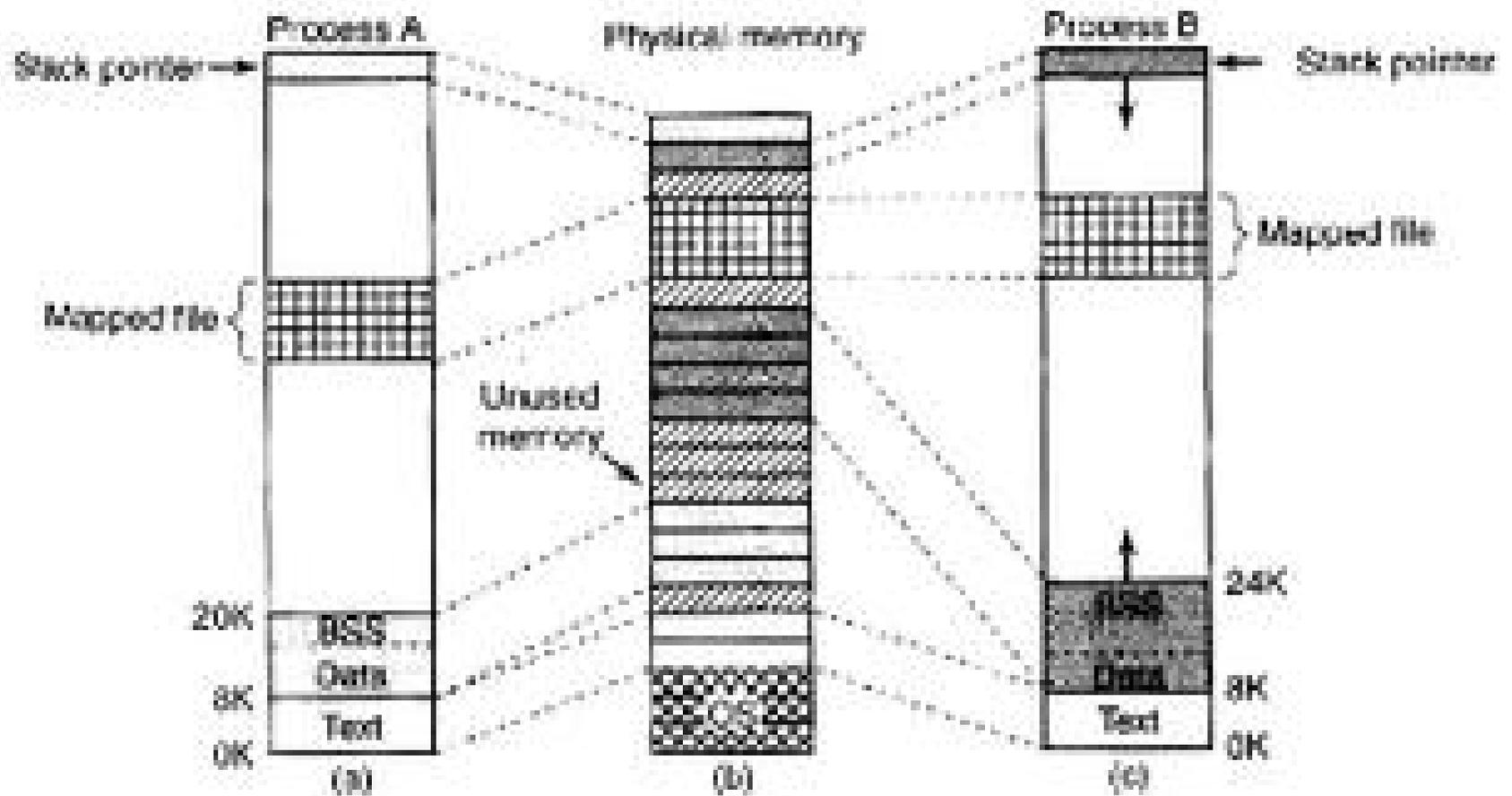


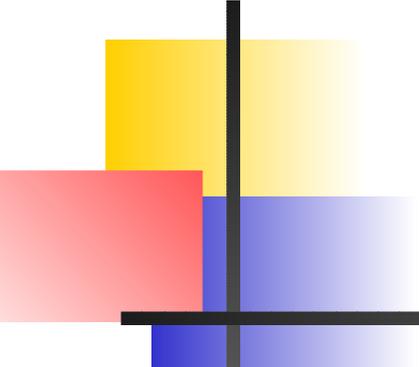


Speicherverwaltung (3) - Memory-Mapped-Files

- Datei wird in den physikalischen Speicher geladen
- Abbildung auf den virtuellen Adressraum eines Prozesses
- einfacher Zugriff (wie auf normalen Speicher)
- Möglichkeit für zwei Prozesse Speicher zu teilen

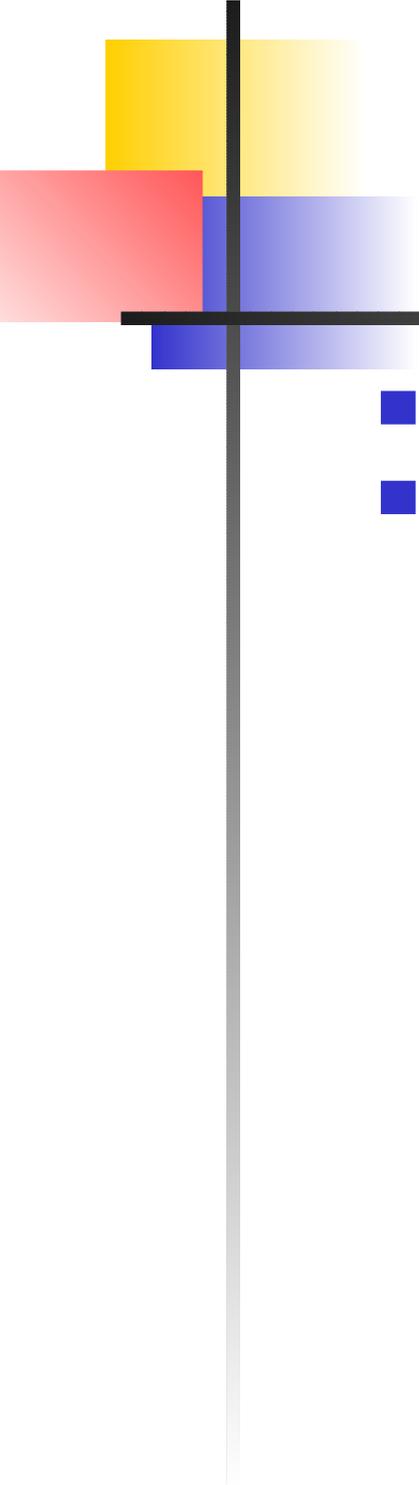
Speicherverwaltung (4) - Memory-Mapped-Files





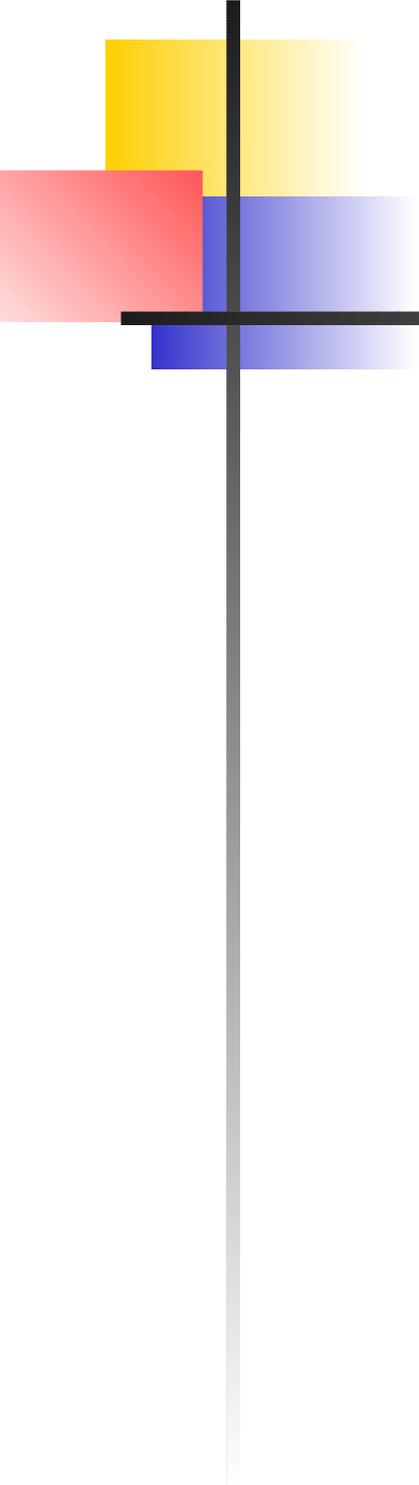
Input/Output - Devices

- Blockdevices
 - wahlfreier Zugriff
 - Festplatte, Diskette
- Characterdevices
 - sequentieller Zugriff
 - Tastatur, Maus



Anekdoten

- Warum wurde UNIX geschrieben?
- Wie entstand der Name UNIX?



Welche eurer Frage darf ich
zuerst beantworten?