ARCHITEKTUREN

Einführung

- eine Floskel zur Komplexität von Betriebssystemen:

> Q: "What is an elephant?"
> A: "A mouse with an operating system."

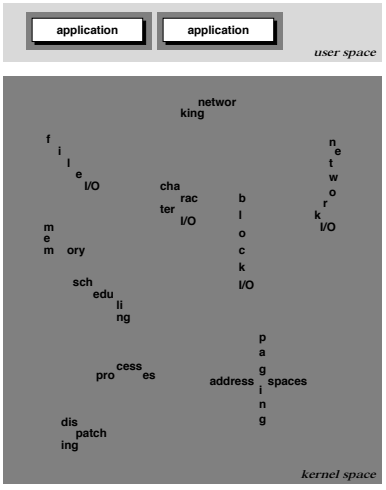
- verschiedene **Architekturformen** haben sich gebildet ...
  - monolithische Systeme
  - schichtenstrukturierte Systeme
  - *Client/Server* Systeme
  - {mikro,nano,piko}kernbasierte Systeme
- ... die für unterschiedliche **Einsatzbereiche** gedacht sind

Monolithen (1)

- ein System ohne wirkliche Struktur
  - ein einziges großes Programm als **Prozedursammlung**
    - uneingeschränkte Aufruffolgen
    - kaum bzw. keine Datenlokalitäten
    - alles in einem Adreßraum
- der „Alptraum" jedes Systementwicklers
  - Nebeneffekte sind unvermeidbar

> It has been known for such systems to reach a point where the number of bugs is constant; any attempt to fix n problems introduces n new ones in exchange.
>
> Lister

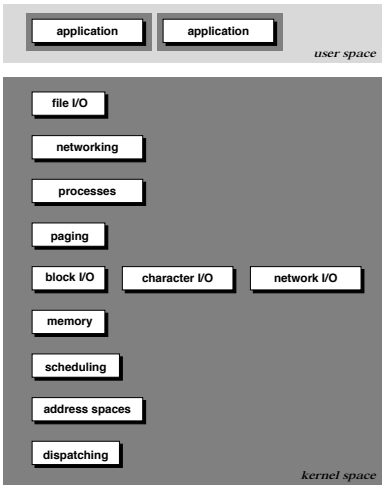
Monolithen (2)



Schichtenstrukturierte Systeme (1)

- **Moduln** werden in **Schichten** angeordnet
  - eindeutige und zyklenfreie Benutztrelation
  - die Korrektheit von Schicht n hängt ab von Schicht $n-1$
- tiefere Schichten
  - kapseln Hardware-Abhängigkeiten
  - sorgen für Zuverlässigkeit
  - sind leistungsoptimiert
  - implementieren bevorzugt Mechanismen
- höhere Schichten
  - sind anwendungsorientiert
  - bedeuten funktionale Anreicherungen
  - implementieren bevorzugt Strategien

Schichtenstrukturierte Systeme (2)



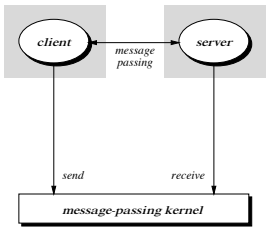
Client/Server Systeme (1)

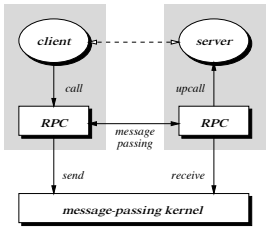
- Moduln sind gleichberechtigte Instanzen
  - Kommunikation über **Nachrichten**
    - nicht über Prozeduraufrufe
  - bidirektionler Nachrichtenfluß
    - vom „aufrufenden“ zum „aufgerufenen“ Modul (*request*)
    - vom „aufgerufenen“ zum „aufrufenden“ Modul (*response*)
  - ein Kommunikationskern bildet die Grundlage
    - das ideale Umfeld für einen Mikrokern
- die Komponenten werden durch **Prozesse** implementiert

Client:	dienstanfordernder (Benutzer-) Prozeß
Server:	diensterbringender (System-) Prozeß

  - ggf. isolierte Prozeßadreßräume
- typische Beispiele:
  - *X server*
  - *file server*, *disk server*, *I/O server*
  - *network server*
  - *external pager*

Client/Server Systeme (2)

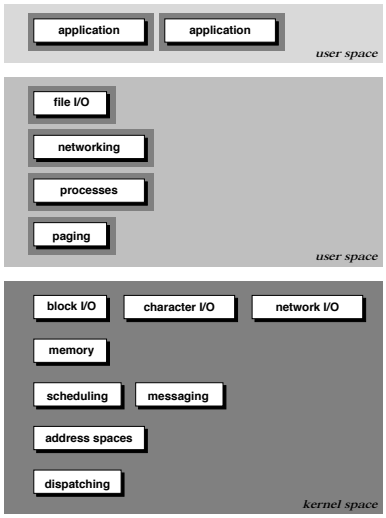




Mikrokerne (1)

- „mikro“ im Sinne der **Funktionalität**
  - die Kernfunktionalität umfaßt typischerweise:
    - Adreßraumverwaltung
    - Prozeß-Scheduling
    - Interprozeßkommunikation
    - Ereignispropagation (an Prozesse)
      - Traps bzw. Interrupts
    - ggf. Gerätetreiber
  - andere Funktionen werden auf Server-Ebene erbracht
    - Client/Server System
- „makro“ im Sinne der **Größe**
  - Megabytes an Speicherbedarf sind nicht unselten

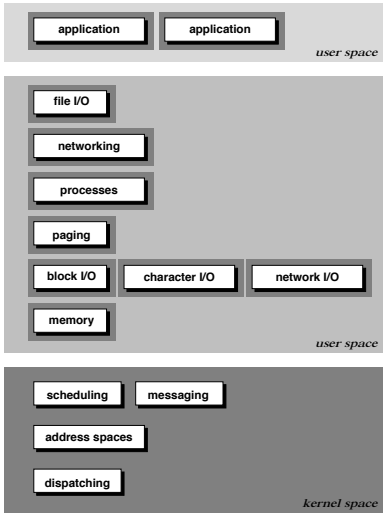
Mikrokerne (2)



Nanokerne (1)

- bieten an **Ereignisse**, **Prozesse** und **Adreßräume**
- der Kern implementiert ein Prozeßkonzept
  - feder-, leicht-, und/oder schwergewichtige Prozesse
  - Scheduling
  - Adreßraumzuordnung
- andere Funktionen laufen auf Benutzerebene ab
  - ggf. auf Server-Basis
  - Client/Server System

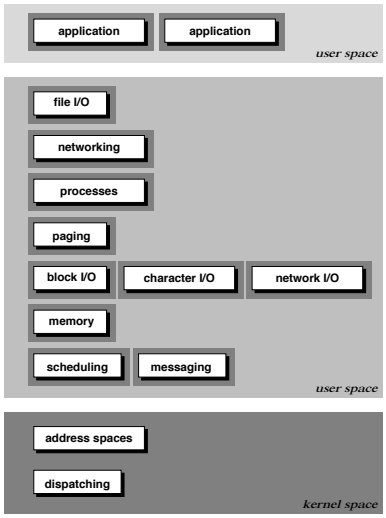
Nanokerne (2)



Pikokerne (1)

- **virtuelle Prozessoren** werden eingeführt
  - der Kern arbeitet als „Melder“ (*dispatcher*) …
    - propagiert Ereignisse hinein in Benutzeradreßräume
  - … ist eine „dünne Schicht“ oberhalb der Hardware
    - implementiert Mechanismen, keine Strategien
    - separiert Schutz von Verwaltung
- Strategien werden außerhalb des Kerns implementiert
  - auf Benutzerebene (*user level*)
  - innerhalb isolierter Adreßräume
    - als Server-Prozeß
    - als Bibliotheksbetriebssystem

Pikokerne (2)



Femtokerne

?

Zusammenfassung

- die **künstliche Grenze** bildet die gemeinsame Basis
  - die Lage der Grenzlinie liefert den Unterschied
  - ist erforderlich/tolerierbar für bestimmte Anwendungen …
    - Mehrbenutzersysteme
  - … unvorteilhaft/unakzeptabel für andere Anwendungen
    - Hochleistungs(parallel)rechner
    - tiefste eingebettete (parallele/verteilte) Systeme
- {Mikro,Nano,Piko}kerne sind nicht notwendigerweise klein
  - Betriebssystemfunktionen wurden zur Benutzerebene migriert
    - Strategien sind aus dem Kern herausgezogen worden
  - neue Kernfunktionen müssen dazu eingeführt werden
    - zur Interaktion zwischen Betriebssystem und seiner Umwelt
- die Erhöhung der **Flexibilität** ist zu Lasten der **Komplexität**