Verteilte Systeme

Bettina Schnor
Institut für Telematik
Medizinische Universität Lübeck
schnor@itm.mu-luebeck.de

70er Jahre:	Timesharing	(1 Rechner für viele Benutzer)
80er Jahre:	PC	(1 Rechner pro Benutzer)
90er Jahre:	parallel/distributed computing	(viele Rechner pro Benutzer)

Verteilte Systeme	
Betriebssysteme	Rechnernetze

Inhalt

1. Betriebssysteme
2. Rechnernetze
3. Verteilte Anwendungen
4. Studium – Was nun?

1. Betriebssysteme

Schichtenmodell eines Rechensystems

Kommandointerpreter (Shell), Compiler, Editor, Spiele, ...
X Window System
Betriebssystem
Maschinenprogrammebene
Mikroprogrammebene
physikalisches Gerät

Definition 1: Die Ressourcen (Betriebsmittel) eines Rechensystems sind alle Hard- und Softwarekomponenten, die für die Programmausführung relevant sind.

Beispiele: Prozessor (CPU), Hauptspeicher, Ein-/Ausgabegeräte, Daten, ausführbarer Code, ...

Definition 2: Ein Betriebssystem bezeichnet alle Programme eines Rechensystems, die die Ausführung der Benutzerprogramme, die Verteilung der Ressourcen auf die Benutzerprogramme und die Aufrechterhaltung der Betriebsart steuern und überwachen.

Betriebsarten

1. Ein–/Mehrbenutzersystem

2. Einprogramm–/Mehrprogrammbetrieb
   (single/multi programming)

3. Stapelbetrieb / Dialogbetrieb
   (batch/interactive mode)

Workstation: Mehrprogrammbetrieb, Dialogbetrieb

Parallelrechenzentrum: Mehrprogrammbetrieb, Stapelbetrieb

Aufgaben eines Betriebssystems

1. Prozeßverwaltung (Erzeugen, Beenden, Zuteilen von Rechenzeit (Scheduling), Prozeßkommunikation, Synchronisation),
2. Hauptspeicherverwaltung (Zuteilung des verfügbaren physikalischen Speichers an Prozesse),
3. Verwaltung des Dateisystems (Lokalisieren einer Datei auf dem Hintergrundspeicher (Platte, Floppy), Verwaltung der freien Bereiche auf dem Hintergrundspeicher, Zugriffskontrolle),
4. Geräteverwaltung,
5. Monitoring, Accounting, Auditing

Definition 3: Ein **Prozeß** ist ein in Ausführung befindliches Programm bestehend aus Programmtext (Binaries), Daten, Speicherplatz für Umgebungsvariablen (z.B. Programmargumente, Rücksprungadressen) und Verwaltungsinformationen (Prozeß-ID, Befehlszähler, ...)).

Beispiel: Prozeßkommunikation und Synchronisation

Aufgabe: Drucke aus der Postscript-Datei `folien.ps` die Seiten 2-4 aus!

1. Lösung:

```
> psselect 2-4 folien.ps > f.ps
> lp f.ps
```

2. Lösung: Eine Pipe zwischen Prozeß A und B (i.Z. A | B) lenkt die Ausgabe von A auf die Eingabe von B um.

```
> psselect 2-4 folien.ps | lp
```


Aufgabe: Suche in der Datei "Weihnachtspost", die alle Briefe an den Weihnachtsmann enthält, nach dem Begriff "Weihnachtswuensche"!

1. Lösung:

```
> grep Weihnachtswuensche Weihnachtspost
```

Aufgabe: Suche in der Datei "Weihnachtspost", die alle Briefe an den Weihnachtsmann enthält, nach dem Begriff "Weihnachtswuensche", und lasse die gefundenen Zeilen seitenweise anzeigen!

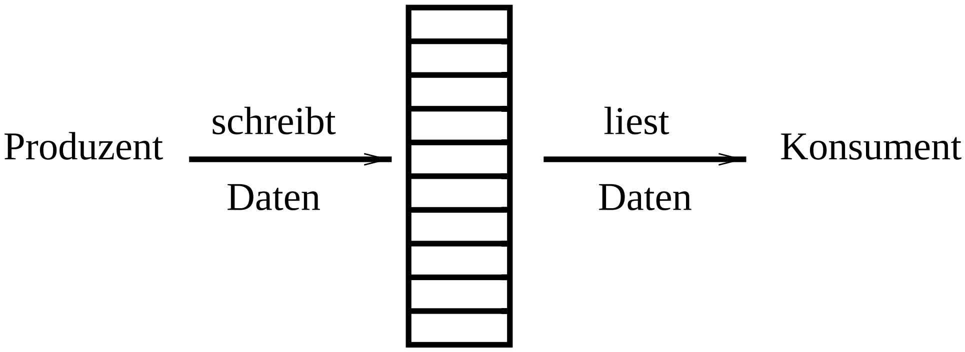
2. Lösung:

```
> grep Weihnachtswuensche Weihnachtspost | less
```

Pipe–Realisierung

MS-DOS: Prozeß A ("Produzent") und Prozeß B ("Konsument") arbeiten sequentiell. Die Daten werden in einer Datei zwischengespeichert.

UNIX: Produzent und Konsument arbeiten "nebenläufig".

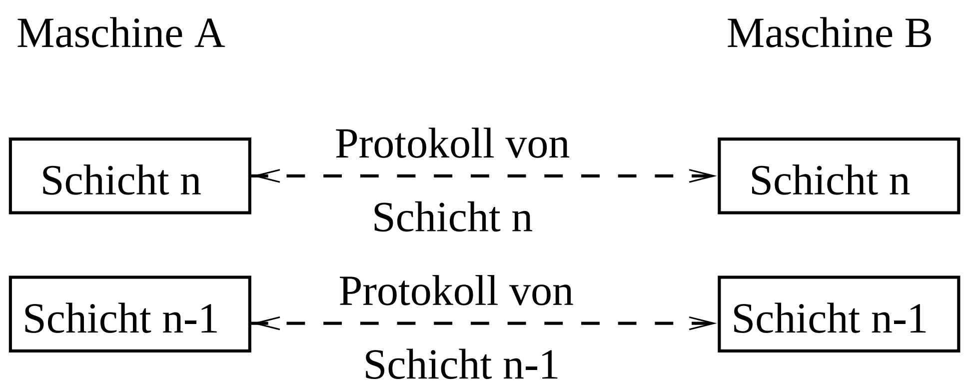


2. Rechnernetze

Aufgaben:

- Primitiven zum Senden und Empfangen von Nachrichten,
- Beachtung der ggf. unterschiedlichen Datendarstellung auf Maschinen unterschiedlicher Hersteller,
- Aufteilen der zu übermittelnden Daten in Pakete,
- Adressierung und Wegwahl,
- Fehlerkontrolle: Treffen alle Pakete einer Nachricht beim Empfänger ein?
- Flußkontrolle: Sender und Empfänger müssen sich im Arbeitstempo synchronisieren, um z.B. den Überlauf von Sende- bzw. Empfangsbuffer zu verhindern.
- Umwandlung der digitalen Daten in elektrische bzw. Lichtsignale,
- fehlerfreie Übertragung eines Pakets (Bitkette) zwischen benachbarten Rechnern.

Strukturierung der Kommunikationssoftware in Schichten:



Ein Prozeß auf Schicht n auf Rechner A kommuniziert mit einem Partnerprozeß auf Schicht n auf Rechner B. Die Regeln, nach denen die Kommunikation abläuft, werden das Protokoll von Schicht n genannt.

Die Kommunikation der Partnerprozesse aus Schicht n wird durch den Dienst der Schicht $n-1$ realisiert. Der Dienst beinhaltet eine Menge von Funktionen, die Prozessen aus Schicht n von der Schicht $n-1$ angeboten werden.

Die Schichten und Protokolle bilden die Netzarchitektur.

Beispiel: Internet Protocol Stack

1969 Arpanet

Geldgeber: (Defense) Advanced Research Projects Agency (DoD (Department of Defense), DoE (Department of Energy) und NASA)

Verbindung der Vielzahl verschiedener Netze $\Longrightarrow$ Internet Protocol (IP) (Mitte der 70er Jahre)

rlogin/rsh	Telnet	FTP	HTTP	DNS	NFS
TCP				UDP	
IP					
Sicherungsschicht					
Bitübertragungsschicht					

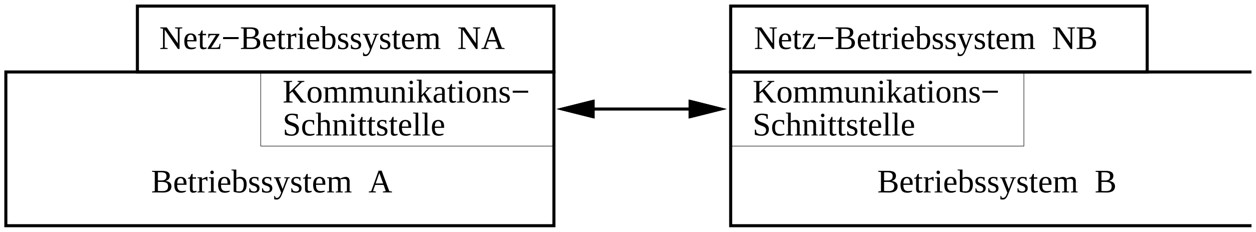
rlogin (remote login)
rsh (remote shell)
Telnet (login session auf entferntem Rechner)
FTP (file transfer program)
HTTP (Hypertext Transfer Protocol (WWW))
DNS (Domain Name Service)
NFS (Network File System)

Wer leitet Pakete durch's Netz? — IP!

Wer vernichtet unzustellbare Pakete? — IP!!

Wer läßt Pakete im Netz verschwinden? — IP!!!

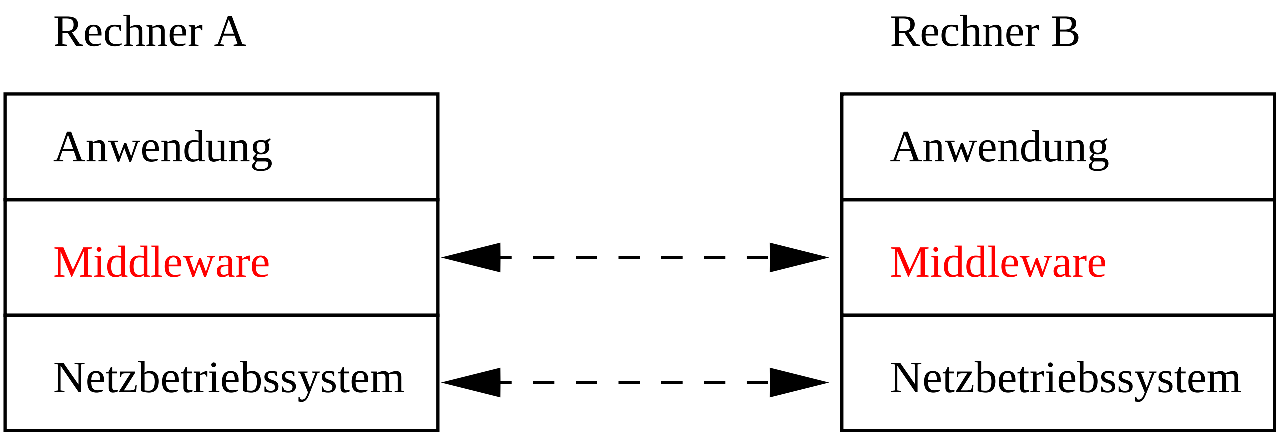
Definition 4: Ein Netzbetriebssystem ist ein Betriebssystem, das um einen Kommunikationsmechanismus erweitert wurde.



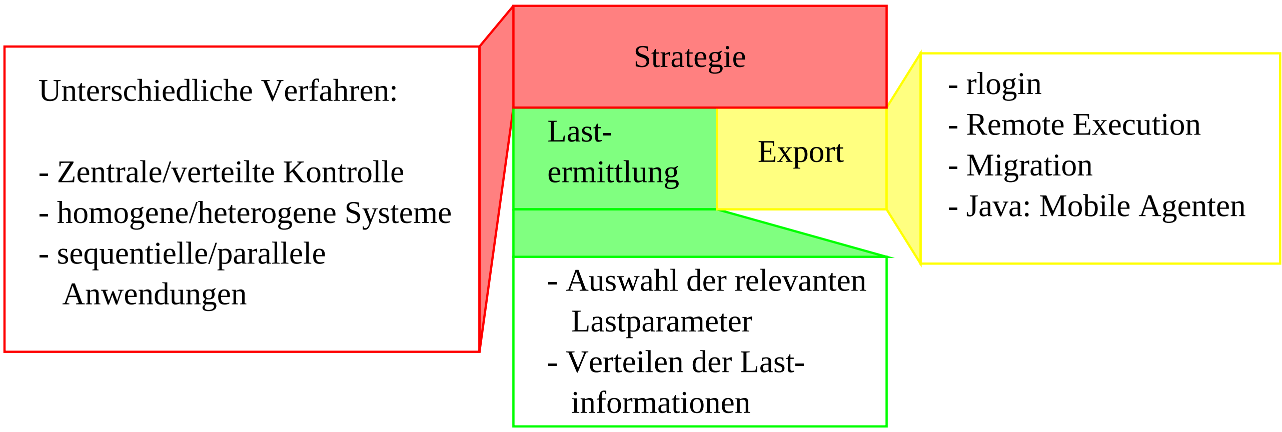
Beispiele für Netzbetriebssysteme: UNIX-Derivate, Windows/NT

3. Verteilte Anwendungen

Ziel: Systemunterstützung für parallele und verteilte Anwendungen auf Workstation-Clustern für Lastverteilung und Fehlertoleranz



Lastverteilung



Beispiele paralleler Anwendungen auf Workstation-Clustern

- genetische Algorithmen zur Optimierung (Axmann, TU Braunschweig):
  - Optimierung von Satellitenkonstellationen
  - Parallelisierungswerkzeug PVM, 70 Workstation.
- 1995 und 1997 Gordon Bell Prize Sieger in der Price/Performance Kategorie:
  Simulation von Schallwellen in Flöten auf einem Cluster von 20 HP Workstations (Skordos, MIT).
  Modellierung von Emulsionen auf 28 DEC Alpha 266 MHz,
  4.7 Gf/s bzw. 10.8 Gf/s/$M (Phan-Thien und Lee, University of Sidney, Tullock, Los Alamos National Lab)
- *Toy Story* wurde auf "Farm" aus 118 Sun-Workstations erstellt.
- *Titanic* auf 350 SGI CPUs und 200 DEC Alpha CPUs unter Linux.

Fehlertoleranz

1. Fehlertolerante Anwendungen

- Checkpointing/Restart
- Prozesse/Server replizieren
- "Wachhund"

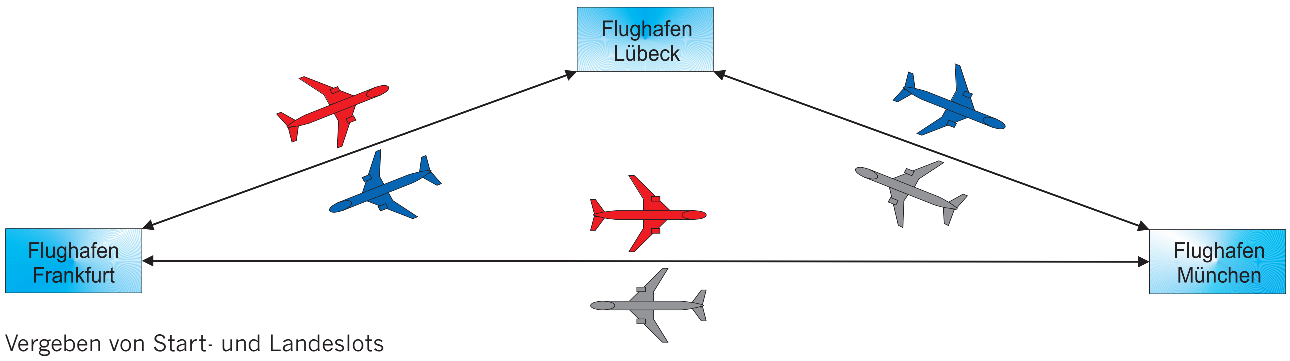
Fehlertolerante Wachhunde

- Synchronisation zwischen Wachhunden: Wer entscheidet?
  $\Longrightarrow$ Synchronisationsprotokolle (Master-Slave, Majority Consensus, ...)
- Datenkonsistenz: Besitzen alle Wachhunde aktuelle Informationen?

Aktuell: Bewertung von Synchronisationsprotokollen unter Benutzung von Multicast-Technologien (Gruppenkommunikation) (z.B. IP-Multicast)

Verteiltes Airport Control System

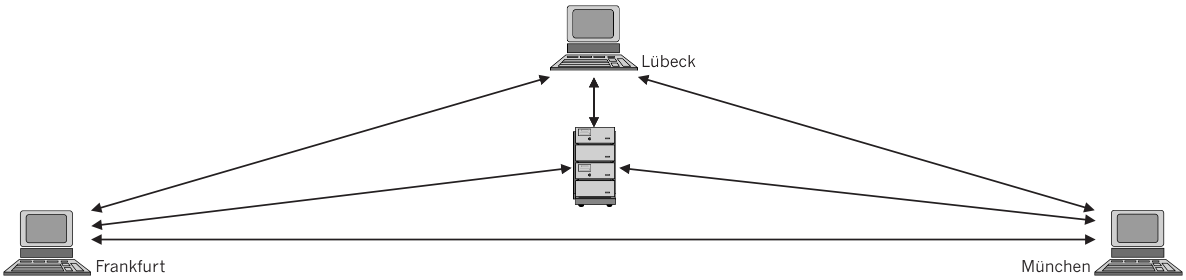
Praktikum „Betriebs- und Kommunikationssysteme“



Vergeben von Start- und Landeslots

Flugumlenkung im Falle von Störungen, z.B. Schlechtwetterfronten, Ausfall des Kontrollrechners, Notfälle an Bord, Ausfall des Zielflughafens

Globale Übersicht über alle Objekte im deutschen Flugraum



Implementation in **Java**

Das **Airport Control System** (ACS) ermöglicht eine globale Sicht und die Notfallbehandlung

Kommunikation zwischen entfernten Rechnern über entfernte Methodenaufrufe mittels **CORBA** (Common Object Request Broker Architecture)

4. Studium – Was nun?

Fallbeispiel 1: Marc G.

Arbeitgeber: EDV-Berater bei der FSS GmbH EDV FINANZSYSTEME UND SERVICES, Hannover

Aufgabengebiet: Projektarbeit bei Kunden wie z.B. dvg Hannover (das Rechenzentrum der nord- und ostdeutschen Sparkassen), Nord/LB, Bankgesellschaft Berlin.

1) Evaluierung, Einfuehrung und Betreuung des Entwicklungswerkzeuges Sniff+ der Fa. TakeFive für die Serverplattform Sun Solaris.

2) Konzeption und Realisierung von Schnittstellen zwischen Unix-basierten Anwendungen und "bestandsführenden" Systemen wie z.B. dem Großrechner der dvg.

Aufteilung der Arbeitszeit: Konzeption 40 %, Betreuung 40 %, Entwicklung 20 %

Fallbeispiel 2: Mathias B.

Arbeitgeber: Volkswagen Bank GmbH,
Abteilung: Softwareentwicklung Direktbank (etwa 20 Personen).

Die Volkswagen Bank direct ist eine Direktbank, die in erster Linie Guthaben-Konten anbietet, aber auch Sparpläne, Sparbriefe, kleine Verbraucherkredite usw.. Die Kontaktaufnahme durch den Kunden geht z.Zt. noch über Telefon.

Ziel: Internetbanking mit Unterstützung eines Solution-Providers für Banksysteme realisieren.

Aufgabengebiet:
1) Koordination der technischen Realisierung im Zusammenspiel mit den durch den Fachbereich der Bank gestellten Anforderungen.

2) Technische Tests der Programme des Solution-Providers und Betreuung der Fachbereiche bei den fachlichen Tests.

3) Konzeption und teilweise Realisierung der Schnittstelle zum "Altsystem" (IBM-Mainframe).

Mathias B. ist seit 1,5 Jahren Mitarbeiter bei der Volkswagen Bank GmbH. Seine Arbeitszeit teilt sich auf in

1. ca. 5-6 Wochen Entwicklung
2. Besprechungen mit Fachbereichen und Solution-Provider, Erstellung von Konzepten, Testkonzeption und -durchfhrung sowie Dokumentation und etwas Betreuung des Systems im laufenden Betrieb

Werkzeuge: Microsoft C++, Java, Javascript, Oracle DB's unter AIX, Microsoft SQL-Server, IBM MVS, SAP usw.