

Interprozess- kommunikation (IPC)

Marcel Waldvogel

Ziele

- **Prozesse**
 - Entstehung
 - Lebensweg, -weise
 - Eigenschaften
- **Kommunikation**
 - Gründe
 - Dienste
 - Anwendungen

Was ist IPC?

- **Inter**
- **Prozess**
 - Was ist ein Prozess?
 - Eigenschaften
- **Kommunikation**
 - Kommunikationsarten
 - Kommunikationswege
 - Kommunikationsdienste

Prozesse

- **Prozess = Programm + Zustand, Ablauf**
 - Zeitliche Komponente
- **Objekte mit Ressourcen (OS)**
- **Verarbeitung**
 - Parallel
 - Sequentiell
 - Überlappend (Zeitscheiben, kooperativ)
- **Entstehung**
 - Biologie!
 - Programmausführung im Prozess

Weight Watchers

- **Zustände**
 - Running, runnable, sleeping
 - Übergänge?
- **Zustandsübergänge teuer**
 - Schwer- vs. leichtgewichtige Prozesse (LWP, Thread) mit viel, wenig Zustandsinfo

Prozesskoordination

- **EVA: Daten und Berechnung**
 - Datenerhalt über Prozesslaufzeit hinaus
- **OS: Hauptsächlich Synchronisation**
- **Abhängigkeiten zwischen nebenläufigen, kooperierenden Prozessen**
 - Ressourcen teilen (I/O, Datenstrukturen, Speicher, ...)
- **Koordination bedingt Kommunikation**

Kommunikation

■ Arten

- Synchron, asynchron (S, A)
- Lokal, entfernt/remote (L, R)

■ Wege

- Nachrichten versenden
- Entfernter Funktionsaufruf
- Gemeinsame Datenstrukturen
- Abgelegte Informationen
- Direkter Einfluss

■ Dienste

■ Gegenseitiges Einverständnis!

- Zugangskontrolle

Kommunikationsdienste

■ Nachrichten, Datenströme

- Message Queues (A; L, R)
- Sockets (A; R) und Pipes (A; L)

■ Entfernter Funktionsaufruf

- Remote Procedure Call (S; R)

■ Gemeinsame Datenstrukturen

- Shared Memory (S; L, R)

■ Informationen

- Dateisysteme, Datenbanken (A; L, R)

■ Direkte Manipulationen

Nachrichten in Warteschlangen

■ Beliebteste Kommunikation

- Internet
- Basisdienst für viele Mehrwertdienste

■ Beispiele

- Lokal?
- Entfernt?

■ Abarbeitung in Folge

- evt. priorisiert
- mit/ohne Erhalt von Meldungsgrenzen
- Warteschlange pro Kanal oder global

Remote Procedure Call

- **Dienst**
 - Entfernte Funktionsausführung
- **Semantik**
 - Lokalem Funktionsaufruf möglichst ähnlich
- **Implementation**
 - Nachrichtenpaar
- **Unterschiede?**
 - "Stubs"
 - Marshalling
 - Prozesskontext
- **Anwendungen**
 - X11, Mikrokern, NFS, AFS, Datenbanken; SunRPC, Java RMI, CORBA

Gemeinsame Datenstrukturen

- **Derselbe Speicherbereich sichtbar von mehreren Prozessen**
- **Shared Memory**
- **Implementation**
 - Lokal
 - Entfernt: Nachrichtendienst und viele Tricks
- **Probleme**
 - Basisadresse, Pointer
 - Synchronisation

Synchronisation

- **Interrupts sperren**
- **Verdrängungssperre (Context Switch Block)**
- **Sperrvariablen**
- **Atomare Operationen**
- **Algorithmische Lösung**
- **Semaphore**
- **Deadlock (Verklemmung)**

Direkte Manipulation

■ Fehlersuche, Debugging

- Inspektion und Modifikation von Variablen
- Steuerung der Programmausführung
- Passive Beobachtung ("execution trace")

■ Fehlerumgehung

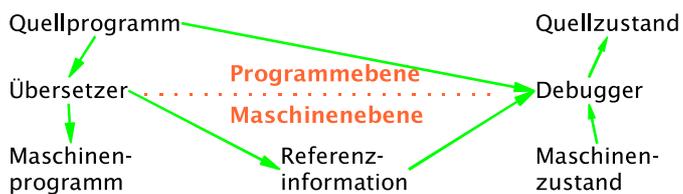
- Zugriffsumgehung
- Lehre: Keine künstlichen, nicht erzwingbaren Rechte!

■ Emulation



Bugs damals und heute

Debugger



Persistenz der Information

■ Dilemma

- Kurzlebige Prozesse
- Daten oft lange wertvoll

■ Abhilfe

- Datenbanken
- Dateisysteme (File system)