

# Systemsoftware und Betriebssysteme

Marcel Waldvogel

## Übersicht

- Grundlagen, Komponenten
- Funktionen: Verwaltung, Koordination
- Verwandtschaft
- Beispiel: Unix
  
- Auswahl
- Portierbarkeit
- Applikationsgrundlage

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 2

## Definition Systemsoftware

- Programme, welche zusammen mit dem Betriebssystem die grundsätzlich Nutzung eines Rechners ermöglichen oder vereinfachen
  - Editor
  - Übersetzer (Compiler, Assembler)
  - Binder (Linker)
  - Lader (Loader)
  - Debugger
  - Datei-, Speicherverwaltung
  - ...

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 3

# Aufgaben

## ■ Dateisystem (File System)

- Bereitstellung, Verwaltung (Sekundärspeicher)

## ■ Betriebsmittel: Überwachung, Verwaltung

- Hardware: CPU, Memory, VM, Peripherie
- Software: Programme, Dateien, Datenbanken

## ■ Jobabwicklung

- Steuerung der Ausführungsreihenfolge
- Zeitliche Überlappung

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 4

# Aufgaben (2)

## ■ Auslastung, Performance

- Prioritäten

## ■ Fehlerbehandlung für Hard-, Software

- Datenschutz (Zugriffsrechte)
- Datensicherheit (gegen Verlust)

## ■ Kommunikation, lokal und verteilt

- Prozess <-> Benutzer
- Prozess <-> Prozess

## ■ Steuerung gleichzeitiger Abläufe

- Mehrere Prozessoren; verteilt über Netzwerk

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 5

# Aufgaben bei Multi\*

## ■ Multiuser, Multitasking, Multiprocessing

## ■ Einheitliche Sicht des Systems

## ■ Trennung und Isolation der Benutzer, Tasks/Prozesse, Prozessoren

- Ausnahmen explizit und einvernehmlich verlangt und festgelegt
- Beispiele?

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 6

# Quellcode bis Ausführung

- Editor
  - Textdatei
- Assembler
  - Quellprogramm zu Objektcode
- Binder (Linker)
  - Verknüpfen von Objektmodulen
- Lader (Loader)
  - Laden und Einrichten im Speicher

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 7

# Assembler

- Aufgaben
  - Syntaxprüfung
  - Pseudoinstruktionen (Assemblerdirektiven)
  - Symbolische Referenzen (Labels)
  - Maschinencode
- Vorwärtsreferenzen
  - Unterprogramm
  - Bedingte Ausführung
  - Daten

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 8

# Auflösung von Vorwärtsreferenzen

- Zwei-Phasen-Übersetzung
  - Phase 1
    - Scanner tokenisiert
    - Aufbau Symboltabelle
  - Phase 2
    - Parser überprüft Syntax
    - Maschinencode
- oder
- Fixup-Tabelle

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 9

# Linker und Lader

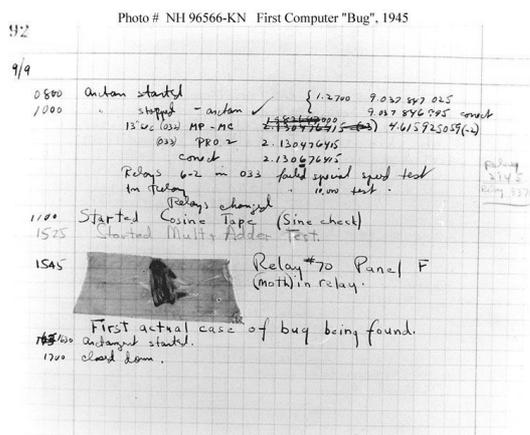
## ■ Aufgaben

- Modularisierung
- Mischen von Code und Daten
- Auflösung externer Referenzen
  - Direkt im Code; mittels Adresstabelle
- Initialisierung der Module

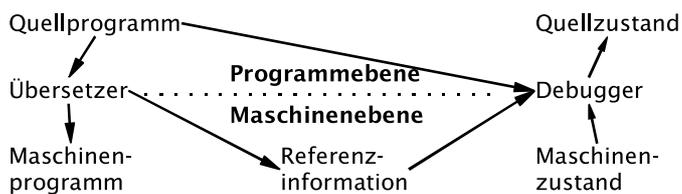
## ■ Objektformat

- Vorspann
- Code
- Globale Daten
- Bindeinformation
  - Exportliste, Importliste, Referenzliste
- Programmsymbole

# Bugs



# Debugger



# Betriebssystem

## ■ Wichtigste Aufgaben

- Kommunikation mit Benutzer mittels Control/Command Language
- Bereitstellung und Zuteilung der Betriebsmittel
- ...

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 13

# Kommunikationsbeispiele

## ■ Filesystem

- z.B. Kommunikation zwischen Assembler, Linker und Lader
- Baumartig, hierarchisch
- Operationen auf Dateien
- Tools für Aufbau, Navigation, Modifikation
- Dateien: Strukturiert oder Bytestrom

## ■ Benutzer

- Spezielle Ein-/Ausgabegeräte
- Oftmals im Filesystem abgebildet

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 14

# Prozesse

## ■ Prozess = Programm + Zustand, Ablauf

- Zeitliche Komponente

## ■ Objekte mit Ressourcen (OS)

## ■ Verarbeitung

- Parallel
- Sequentiell
- Überlappend (Zeitscheiben, kooperativ)

## ■ Zustände

- Running, runnable, sleeping
- Übergänge?

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 15

# Weight Watchers

- **Zustandsübergänge teuer**
  - Schwer- vs. leichtgewichtige Prozesse (LWP, Thread) mit viel, wenig Zustandsinfo

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 16

# Prozesskoordination

- **OS: Hauptsächlich Synchronisation**
- **Abhängigkeiten zwischen nebenläufigen, kooperierenden Prozessen**
  - Ressourcen teilen (I/O, Datenstrukturen, Speicher, ...)

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 17

# Synchronisation

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

Marcel Waldvogel, IBM Zurich Research Laboratory, Universität Konstanz, 15.10.2001, 18