



Betriebssysteme (BS)

Einführung

<http://ess.cs.tu-dortmund.de/DE/Teaching/SS2018/BS/>

Olaf Spinczyk

olaf.spinczyk@tu-dortmund.de

<http://ess.cs.tu-dortmund.de/~os>





Inhalt

- Organisatorisches
- Was ist ein Betriebssystem?
- Ein Blick in die Geschichte
 - Serielle Verarbeitung und Stapelbetrieb
 - Mehrprogramm- und Dialogbetrieb

Literatur

Silberschatz, Kap. 1,
„Einführung“

Tanenbaum, Kap. 1,
„Einführung“



Inhalt

- **Organisatorisches**
- Was ist ein Betriebssystem?
- Ein Blick in die Geschichte
 - Serielle Verarbeitung und Stapelbetrieb
 - Mehrprogramm- und Dialogbetrieb

Literatur

Silberschatz, Kap. 1,
„Einführung“

Tanenbaum, Kap. 1,
„Einführung“



Lernziele

- Grundlagenwissen über **Betriebssysteme** erwerben
 - Funktionsweise und Struktur
 - Algorithmen und Implementierung
- Erfahrung mit **systemnaher Programmierung** sammeln
 - Übungsaufgaben in C unter UNIX
- Verständnis der Vorgänge in einem **Rechnersystem**
- Aktuelle **Trends** und **Herausforderungen** kennen
 - Zumindest ein paar wichtige ...



Voraussetzungen

- Grundkenntnisse aus Vorlesungen
 - DAP1
 - Rechnerstrukturen
- Neugier
- Fähigkeit zum selbständigen Arbeiten
 - Teamfähigkeit
 - Beherrzigung der Regeln guter wissenschaftlicher Praxis



Organisation

- **Vorlesung („VL“)**
(1,5 Std. wöchentlich, Mo. 12:15-13:45)
 - Vorstellung und detaillierte Behandlung des Lehrstoffs
- **Tafelübung („TÜ“)**
(1,5 Std. **zweiwöchentlich**:
Gruppen mit Bezeichnung „gerade KW“ / „ungerade KW“ alternierend;
Dienstag: 10:15-11:45 oder 14:15-15:45 oder 16:15-17:45 oder
Mittwoch: 08:30-10:00 oder 10:15-11:45 oder 14:15-15:45)
 - Vertiefung der VL sowie Besprechung der Übungsaufgaben
- **Rechnerübung/Help-Desk („RÜ“)**
(6,5 Std. wöchentlich, Mo. 14:15-16:45, Di. 10:15-11:45, Mi. 14:15-16:45)
 - Hilfestellung beim Bearbeiten der Übungsaufgaben am Rechner
- **Vor-/Nacharbeit**
(N Std. wöchentlich, $0 < N < 158,5$)

in der Summe: **3 SWS**



Dozenten, Übungsleiter

- Vorlesung:
 - Olaf Spinczyk
- Übung:
 - Daniel Friesel (1)
 - Henri Gründer (2)
 - Sebastian Lukas Hauer (3)
 - Boguslaw Jablkowski (4)
 - Matthias Jakobs (5)
 - Tim Katzke (6)
 - Nicolas Lenz (7)
 - David Nagy (8)
- Rechnerübung/Help-Desk:
 - Sami Ansawi (9)





Tafelübungen

- Anmeldung über Veranstaltungsw Webseite
 - mit Matrikelnummer und Uni-Mailadresse
 - Auswahl aus 6 Übungszeiten
 - Angaben von „Cliques“ möglich
- Prioritäten-Verfahren bis **13.04. um 14 Uhr.**
 - danach: FCFS (Nachzügler, Gruppe wechseln, ...)
- Automatische Zuordnung zu passender Übungsgruppe
 - Kalenderwoche und Übungsleiter legt der Optimierer fest

Wann?	Priorität	Beliebtheit (# Prio 1)
Di : 10:15 - 11:45	1 - höchste ▾	<div><div></div></div> 66
Di : 14:15 - 15:45	1 - höchste ▾	<div><div></div></div> 63
Di : 16:15 - 17:45	1 - höchste ▾	<div><div></div></div> 10
Mi : 08:30 - 10:00	1 - höchste ▾	<div><div></div></div> 10
Mi : 10:15 - 11:45	1 - höchste ▾	<div><div></div></div> 45
Mi : 14:15 - 15:45	1 - höchste ▾	<div><div></div></div> 16
Speichern		

Stand: heute 10:10



Übungsaufgaben

- Theoriefragen und praktische Programmieraufgaben
- Vorstellung der neuen Aufgaben in der TÜ
- Bearbeitung in Dreiergruppen
(Gruppenmitglieder sollten in derselben TÜ angemeldet sein)
 - Kein Kopieren von anderen Gruppen!
 - Wer ein **Plagiat** abgibt, erhält **keine Studienleistung**
- Abgabe bis **Di. 08:00** bzw. **Do. 08:00** in der Woche nach der TÜ
(mind. eine Woche Bearbeitungszeit)
- Vorstellung der Lösung in der folgenden TÜ



Leistungskontrolle/Anforderungen

- Studienleistung
 - Erfolgreiche Bearbeitung *aller* Aufgaben (A0-A5)
 - Wer 1-3 Aufgaben nicht schafft, darf in den „**Last Chance Test**“
- Prüfung
 - Klausur nach Ende des Semesters (9.8. und/oder 27.9.)
 - Studienleistung ist Voraussetzung für die Teilnahme
 - Relevant ist der Vorlesungs- und Übungsstoff!
- Wen betrifft was?
 - Bachelor-Studium Informatik u. Angewandte Informatik
 - Lehramt (5 CP/Bachelor, 4 CP/Master)
 - Schüler-Studierende
 - Alle **anderen** bitte melden



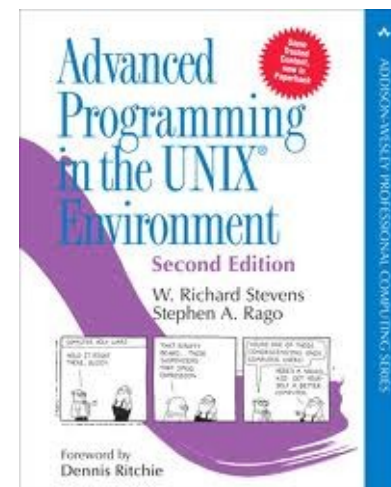
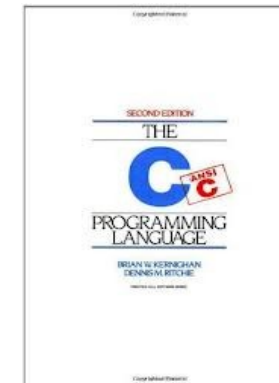
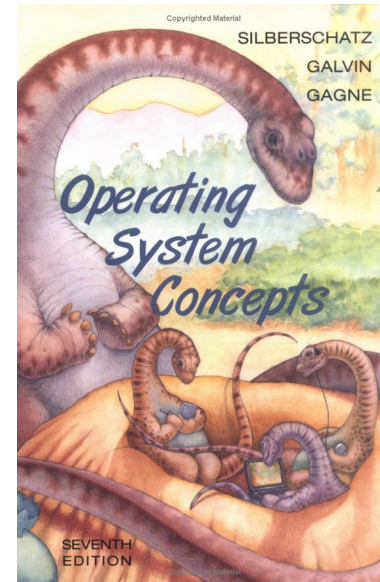
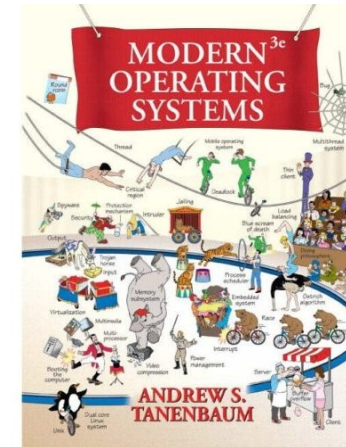
Ausblick: Stoff von BS

- Kontrollflussabstraktionen
 - Fäden, Prozesse
- Prozessorzuteilung
- Kooperation und Konkurrenz von Kontrollflüssen
 - Synchronisation, Verklemmungen
- Verwaltung und Virtualisierung des Hauptspeichers
- Ein- und Ausgabe
- Dateisysteme
- IT-Sicherheit
- Multiprozessorsysteme



Empfohlene Literatur

- [1] A. Silberschatz et al.
Operating System Concepts.
Wiley, 2004. ISBN 978-0471694663
- [2] A. Tanenbaum.
Modern Operating Systems (2nd ed.).
Prentice Hall, 2001. ISBN 0-13-031358-0
- [3] B. W. Kernighan, D. M. Ritchie.
The C Programming Language.
Prentice-Hall, 1988.
ISBN 0-13-110362-8 (paperback)
0-13-110370-9 (hardback)
- [4] R. Stevens.
Advanced Programming in the UNIX Environment,
Addison-Wesley, 2005. ISBN 978-0201433074





Inhalt

- Organisatorisches
- **Was ist ein Betriebssystem?**
- Ein Blick in die Geschichte
 - Serielle Verarbeitung und Stapelbetrieb
 - Mehrprogramm- und Dialogbetrieb

Literatur

Silberschatz, Kap. 1,
„Einführung“

Tanenbaum, Kap. 1,
„Einführung“



Los geht's ...

Quizfrage:

Was ist ein Betriebssystem?



Definitionen (1)

*„Ein Computer ist, wenn er genau betrachtet wird, nur eine Ansammlung von Plastik und Metall, das zur Leitung von Strom benötigt wird. Dieser „Industriemüll“ kann somit nicht ausschließlich das sein, was wir unter einem modernen Computer verstehen, etwas, das dem **Computer „Leben“ einhaucht** und ihn zu dem Werkzeug unseres Jahrhunderts macht.*

*Es ist das Betriebssystem, das die **Kontrolle** über das Plastik und Metall (Hardware) übernimmt und anderen Softwareprogrammen (Excel, Word, . . .) eine **standardisierte Arbeitsplattform** (Windows, Unix, OS/2) schafft.“*

Ewert et al., Literatur zu „Freehand 10“



Definitionen (2)

„**Be'triebs-sys-tem** Programmbündel, das die **Bedienung eines Computers** ermöglicht.“

Universalwörterbuch Rechtschreibung

„Summe derjenigen Programme, die als **residentialer Teil** einer EDV-Anlage für den Betrieb der Anlage und für die Ausführung der Anwenderprogramme erforderlich ist.“

Lexikon der Informatik

„Die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften der Rechenanlage die Grundlage der möglichen Betriebsarten des digitalen Rechensystems bilden und insbesondere die **Abwicklung von Programmen** steuern und überwachen.“

DIN 44300



Definitionen (3)

*„Ein Programm das als **Vermittler** zwischen Rechnerbenutzer und Rechnerhardware fungiert. Der Sinn des Betriebssystems ist eine Umgebung bereitzustellen, in der Benutzer bequem und effizient Programme ausführen können.“*

Silberschatz [1]

*„Eine **Softwareschicht**, die alle Teile des Systems verwaltet und dem Benutzer eine Schnittstelle oder **virtuelle Maschine** anbietet, die leichter zu verstehen und zu programmieren ist [als die nackte Hardware].“*

Tanenbaum [2]



Vielfalt der Anforderungen

**High Performance
Computing**

Minimale Kommunikations-
latenzen



**Arbeitsplatz-
systeme**

Intuitive Benutzer-
oberfläche



Sichere Systeme
Zugriffsschutz

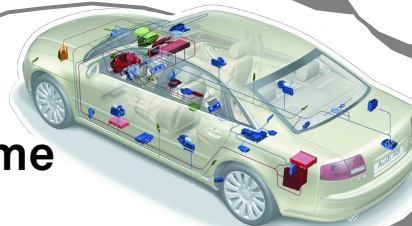


Echtzeitsysteme
Vorhersagbares
Zeitverhalten



**Eingebettete und
automotive Systeme**

Minimaler
Speicherplatzbedarf





Zwischenfazit

- Es gibt viele Auslegungen des Begriffs „Betriebssystem“
- Festhalten kann man ...
 - Das Betriebssystem dient den Anwendern bzw. deren Anwendungsprogramm(en) und nie dem Selbstzweck.
 - Es muss die Hardware genau kennen und den Anwendungen geeignete Abstraktionen zur Verfügung stellen.
- Hardware und Anwendungsanforderungen bestimmen die Dienste des Betriebssystems
 - Struktur und Funktionsweise ergeben sich entsprechend
 - Um zu verstehen, welche Hardwareabstraktionen Betriebssysteme heute anbieten, muss man ihre Entwicklungsgeschichte im Zusammenhang mit der Hardwareentwicklung und typischer Anwendungen betrachten.



Inhalt

- Organisatorisches
- Was ist ein Betriebssystem?
- **Ein Blick in die Geschichte**
 - Serielle Verarbeitung und Stapelbetrieb
 - Mehrprogramm- und Dialogbetrieb

Literatur

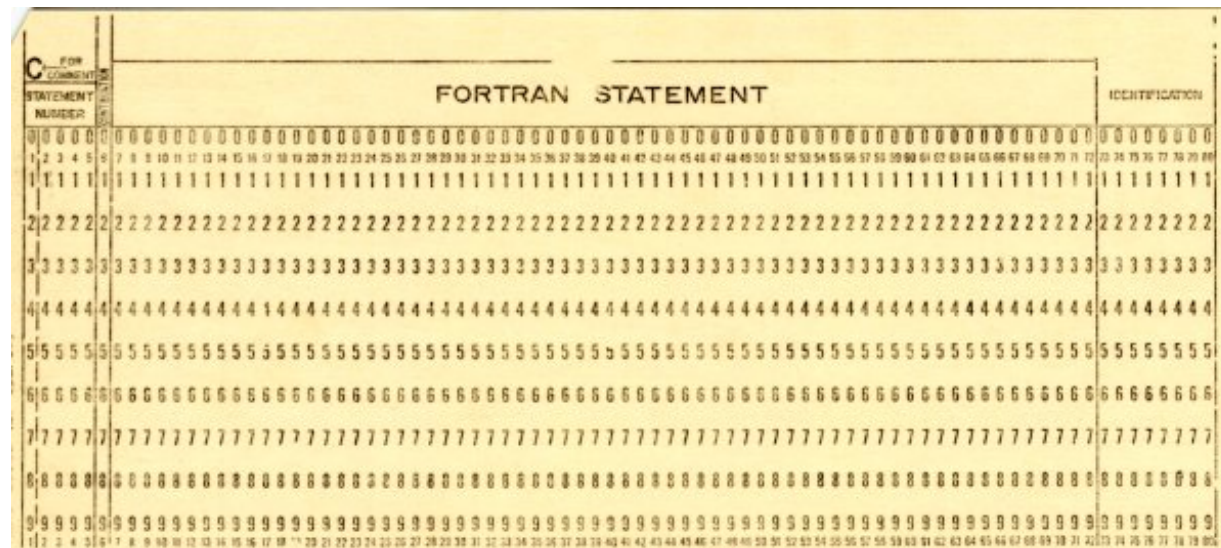
Silberschatz, Kap. 1,
„Einführung“

Tanenbaum, Kap. 1,
„Einführung“



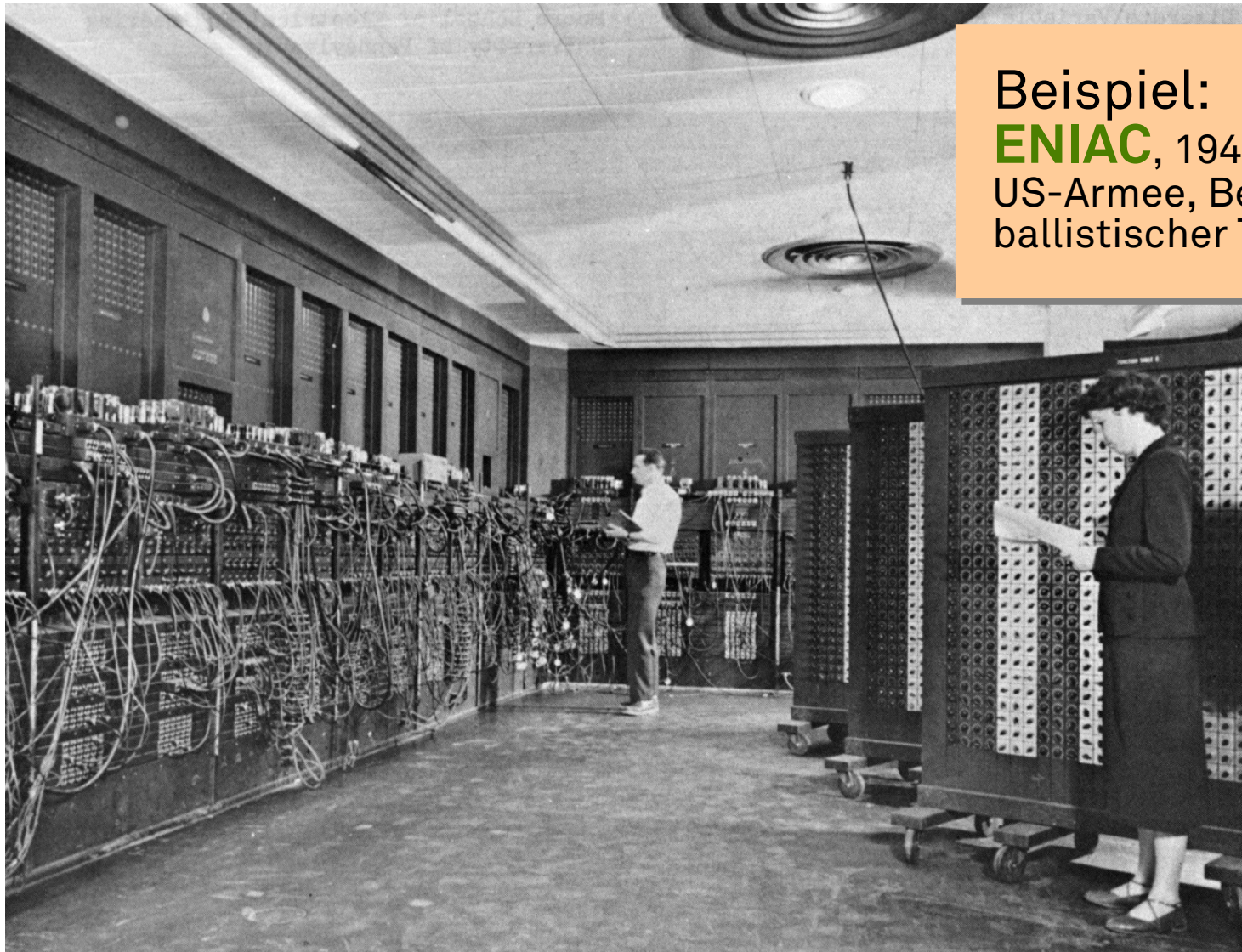
Am Anfang stand die Lochkarte

- Es gibt sie schon seit 1725 zur Webstuhlsteuerung
- Herman Hollerith nutzte sie 1890 für eine Volkszählung
 - aus seiner Firma und zwei weiteren ging später IBM hervor
- Sie wurde bis in die 70er Jahre als vielseitiger Speicher eingesetzt.





Erste elektronische Universalrechner



Beispiel:
ENIAC, 1946
US-Armee, Berechnung
ballistischer Tabellen



Erste elektronische Universalrechner



Beispiel:
ENIAC, 1946
US-Armee, Berechnung
ballistischer Tabellen

Ein Rechenmonster!

- Größe: 10m x 17m x 2,7m
- Gewicht: 27t
- Leistung: 174kW (> 17.000 Röhren!)
- Preis: 468.000 \$
- Geschwindigkeit: 500 Additionen pro Sekunde



Serielle Verarbeitung (ab 1945)

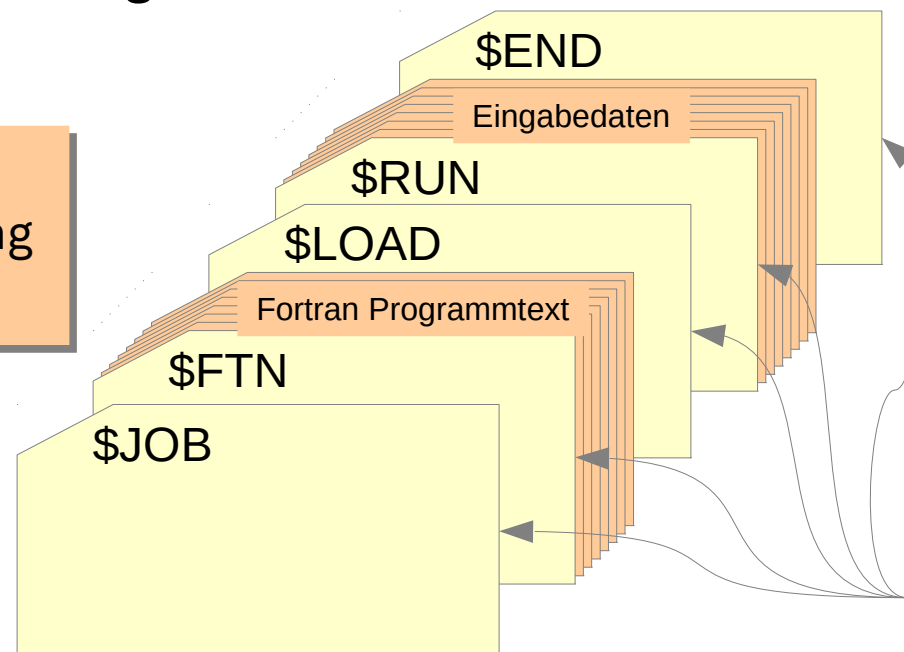
- Programmierung
 - i.d.R. in Maschinencode
 - Eingabe über Lochkartenleser, Ausgaben über Drucker
 - Fehleranzeige durch Kontrolllämpchen
- Rechnerzeituteilung auf Papierterminkalender
 - **Rechnerzeitverschwendung** durch zu großzügige Reservierung oder Abbruch wegen Fehler
- Minimale Auslastung der CPU
 - Die meiste Zeit verbrauchten **langsame E/A Geräte** (Lochkarten, Drucker)
- Erste Systemsoftware in Form **Programmbibliotheken**
 - Binder, Lader, *Debugger*, Gerätetreiber, ...



Einfache Stapelsysteme (ab 1955)

- Verringerten die Häufigkeit manueller Betriebseingriffe
- Die ersten Betriebssysteme: „**residente Monitore**“
 - Interpretation von *Job*-Steuerbefehlen
 - Laden und Ausführen von Programmen
 - Geräteansteuerung

Ein Stapel Lochkarten zur Übersetzung und Ausführung eines FORTRAN Programms



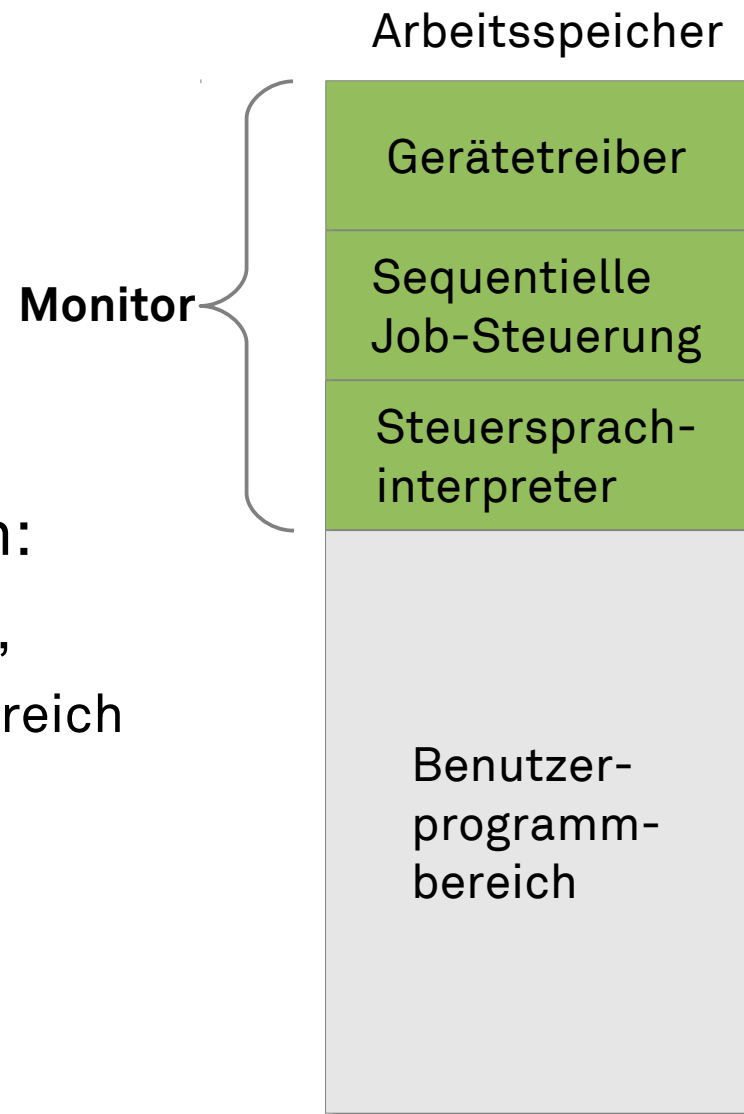
NEU: „Steuerkarten“
(engl. *control cards*)



Einfache Stapelsysteme (ab 1955)

Der Monitor bleibt dauerhaft im Speicher während er ein Anwendungsprogramm nach dem anderen ausführt.

- **Probleme** durch fehlerhafte Anwendungen:
 - Programm terminiert nicht,
 - schreibt in den Speicherbereich des residenten Monitors
 - Greift auf den Kartenleser direkt zu und interpretiert Steuerbefehle als Daten

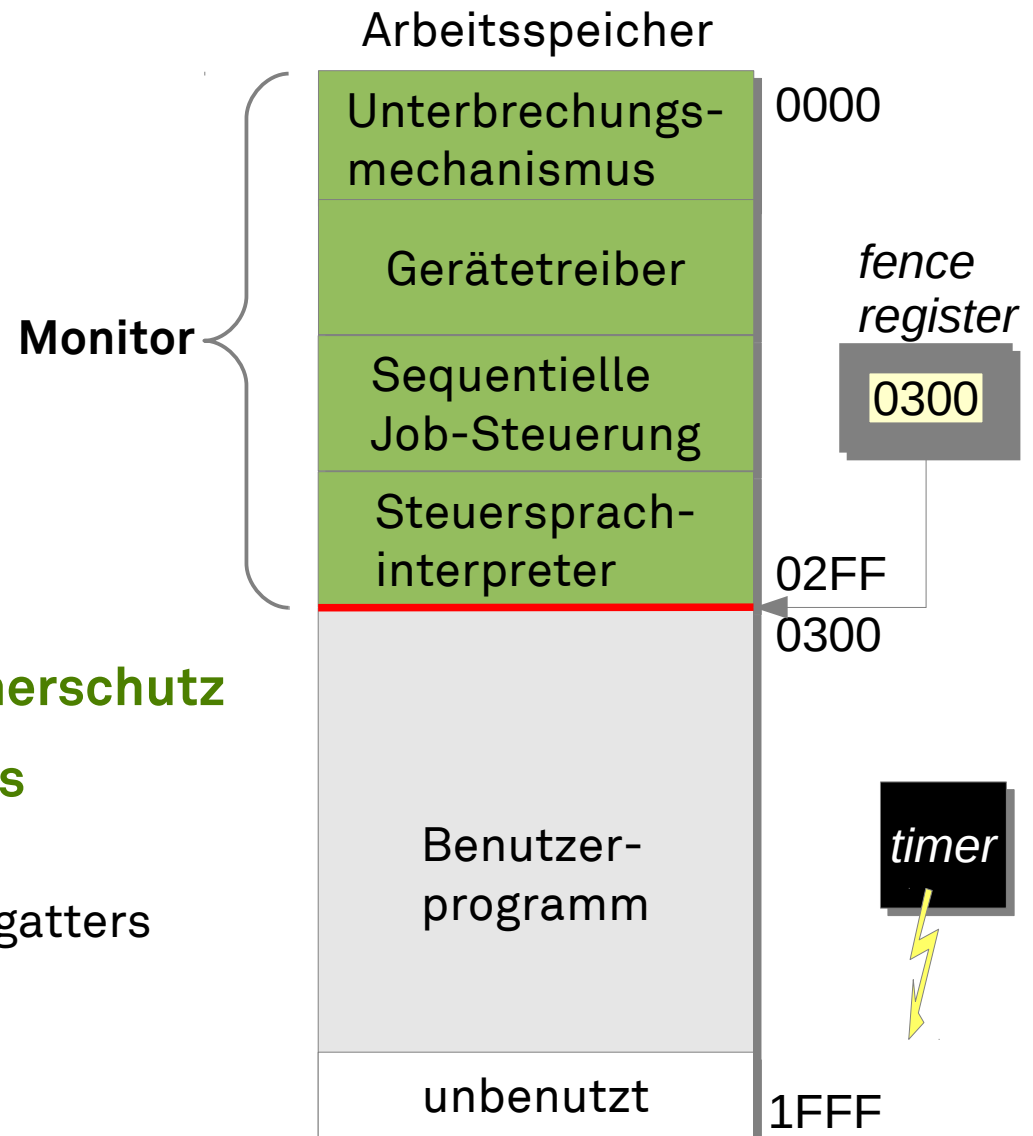




Einfache Stapelsysteme (ab 1955)

Lösungen:

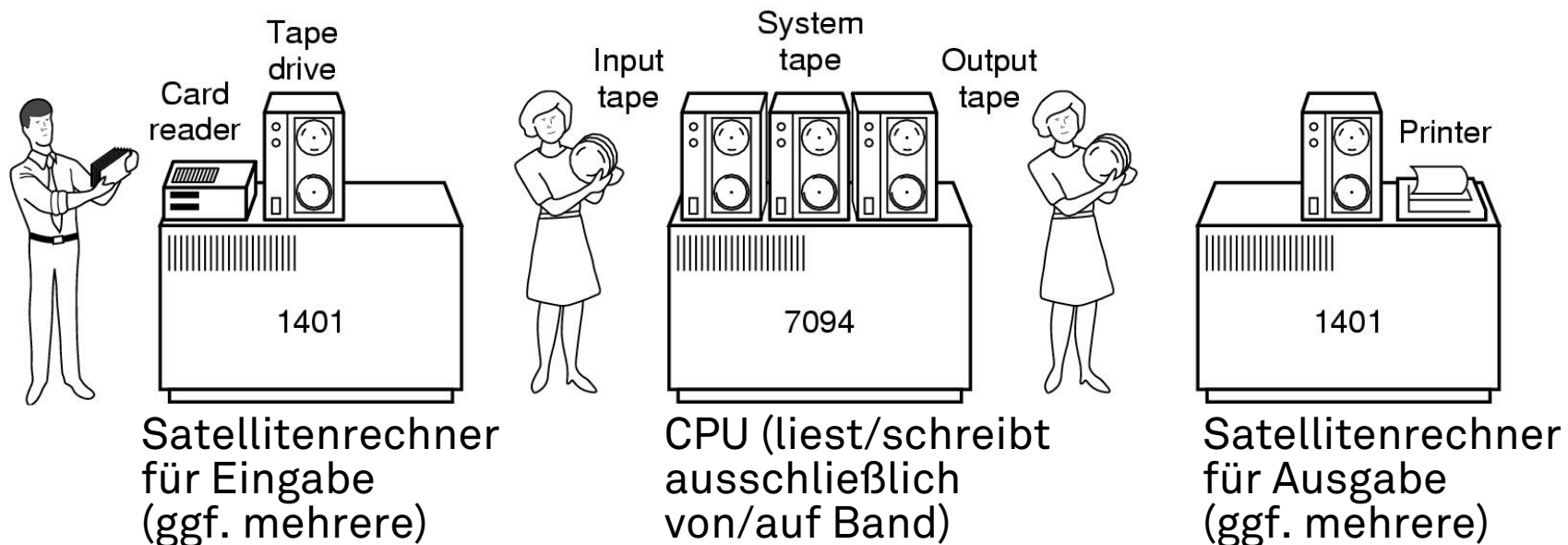
- Zeitgeberbaustein (*timer*) liefert **Unterbrechungen** (*interrupts*)
- **Fallen** (*traps*) für fehlerhafte Programme
 - Schutzgatterregister (engl. *fence register*) realisiert primitiven **Speicherschutz**
 - **Privilegierter Arbeitsmodus** der CPU (supervisor mode)
 - Deaktivierung des Schutzgatters
 - Ein-/Ausgabe





Der Ein-/Ausgabe-Flaschenhals

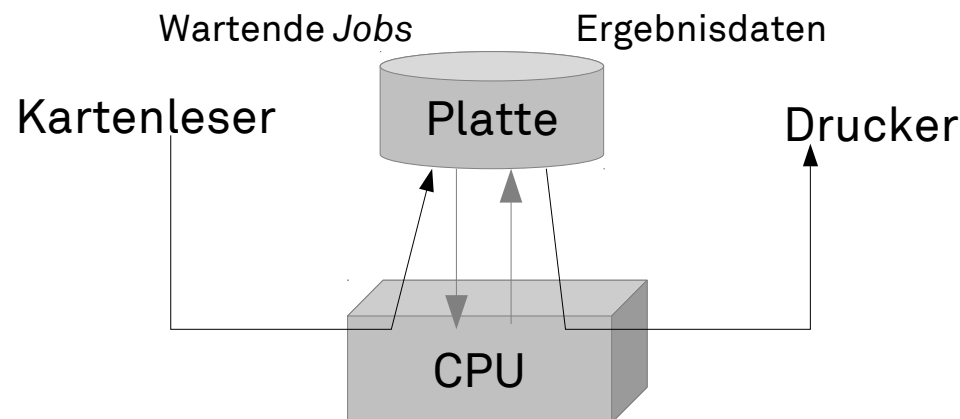
- **Problem:** CPU ist schneller als Kartenleser und Drucker
 - kostbare Rechenzeit wird durch (aktives) Warten verschwendet
- **Lösung 1: *Off-line processing***
 - dank Bandlaufwerken
 - Parallelisierung von Ein-/Ausgaben durch mehrere **Satellitenrechner**





Der Ein-/Ausgabe-Flaschenhals

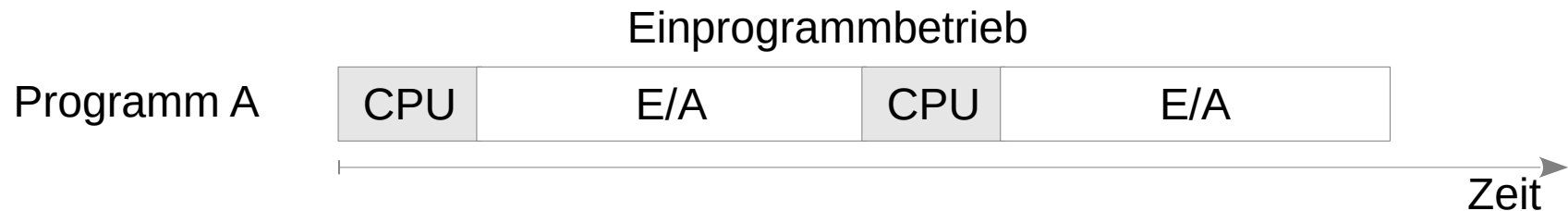
- **Problem:** CPU ist schneller als Kartenleser und Drucker
 - kostbare Rechenzeit wird durch (aktives) Warten verschwendet
- **Lösung 2: Spooling**
 - dank Plattenlaufwerken (wahlfreier Zugriff), **Direct Memory Access** und **Unterbrechungen**
 - Berechnungen und Ein-/Ausgaben werden dabei parallelisiert.
 - Regeln für **Prozessorzuteilung**



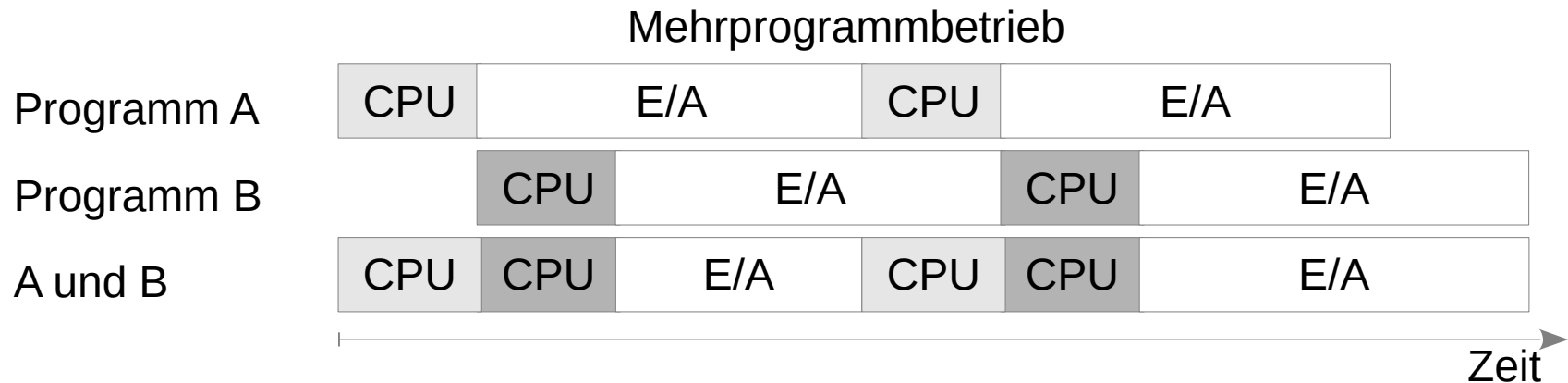


Mehrprogrammbetrieb (ab 1965)

- Trotz *spooling* nutzt ein einzelnes Programm die CPU nicht effizient.
 - **CPU-Stöße** (*CPU bursts*) und **E/A-Stöße** (*I/O bursts*), bei denen die CPU warten muss, wechseln sich ab.



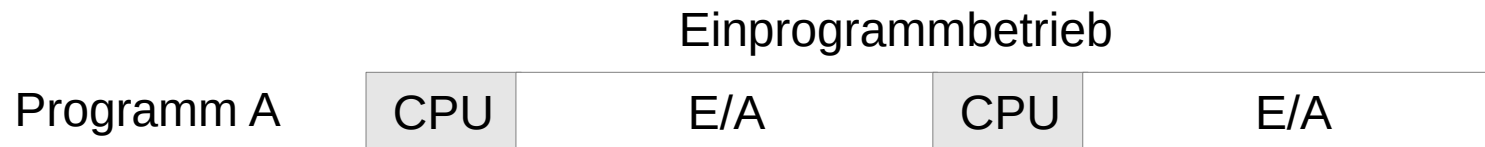
- Beim Mehrprogrammbetrieb bearbeitet die CPU mehrere Aufträge gleichzeitig:



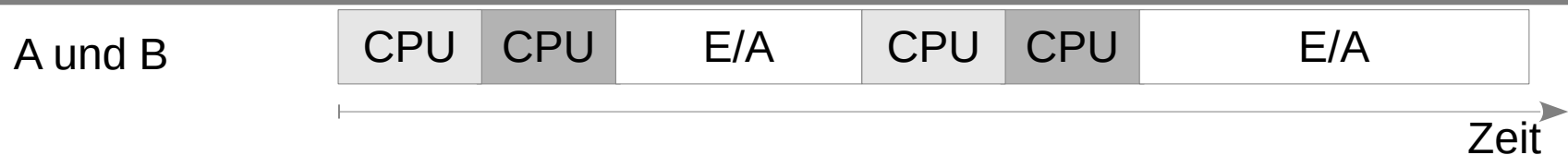


Mehrprogrammbetrieb (ab 1965)

- Trotz *spooling* nutzt ein einzelnes Programm die CPU nicht effizient.
 - **CPU-Stöße** (*CPU bursts*) und **E/A-Stöße** (*I/O bursts*), bei denen die CPU warten muss, wechseln sich ab.



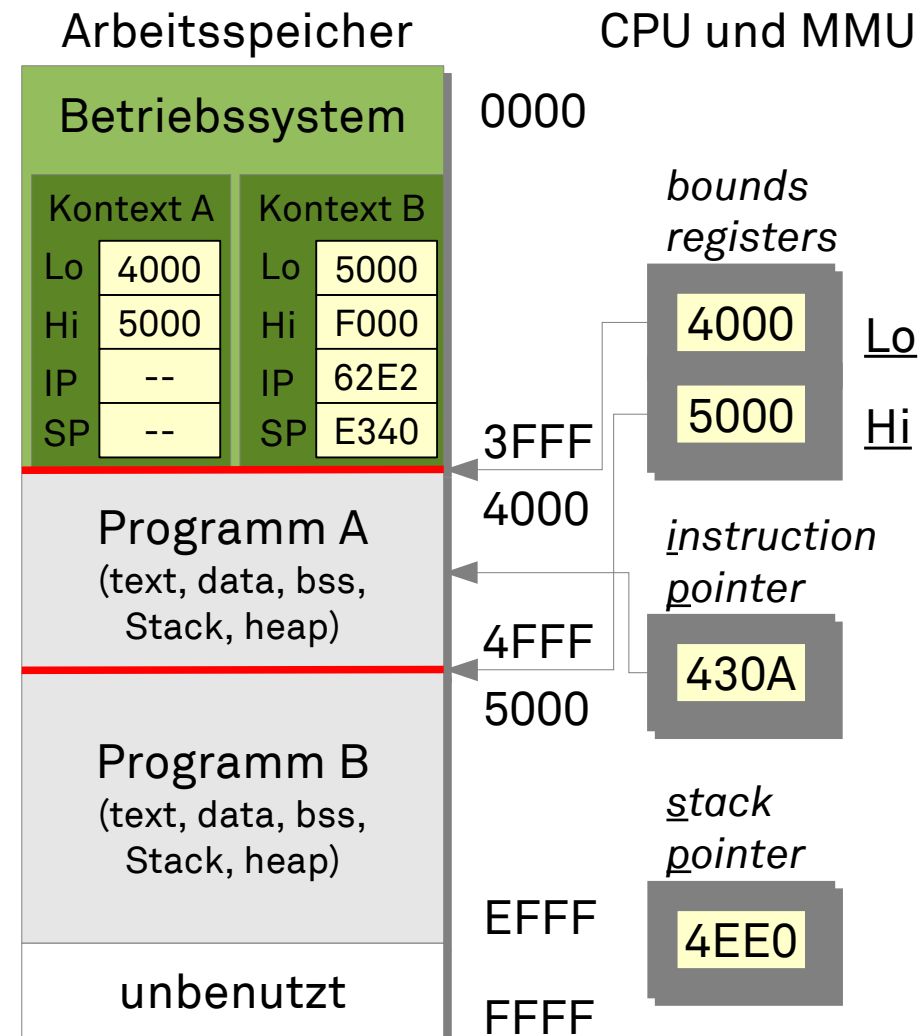
- **Das Betriebssystem wird immer komplexer:**
 - Umgang mit nebenläufigen E/A Aktivitäten
 - **Verwaltung des Arbeitsspeichers** für mehrere Programme
 - Interne Verwaltung von Programmen in Ausführung („**Prozesse**“)
 - **Prozessorzuteilung** (*scheduling*)
 - Mehrbenutzerbetrieb: **Sicherheit** und Abrechnung (*accounting*)





Mehrprogrammbetrieb (ab 1965)

- **Speicherverwaltung:**
 - Den zu startenden Programmen muss dynamisch freier Speicher zugewiesen werden.
- **Speicherschutz:**
 - Einfaches Schutzgatter reicht nicht mehr, um einzelne Programme zu isolieren. Lösung: einfache MMU („Memory Management Unit“)
- **Prozessverwaltung:**
 - Jedes „Programm in Ausführung“ besitzt einen Kontext. Beim Prozesswechsel muss dieser ausgetauscht werden.





Dialogbetrieb (ab 1970)

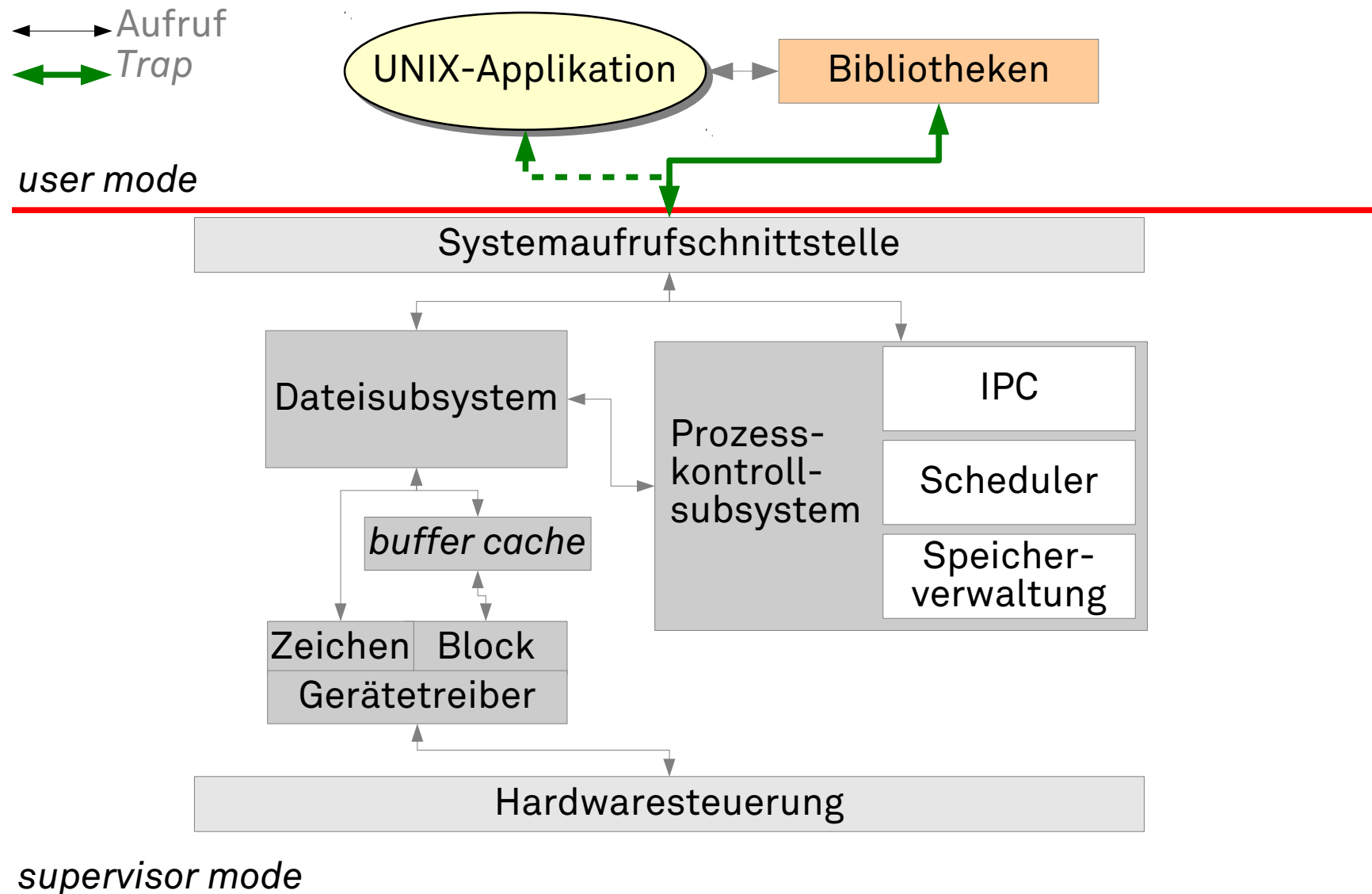
- Neue Ein- und Ausgabegeräte erlauben interaktive Software
 - Tastatur, Monitor, später Maus
- *Time-Sharing* Betrieb
 - ermöglicht akzeptable Antwortzeiten für interaktive Nutzer
 - Zeitgeber-Unterbrechungen sorgen für Verdrängung (zu) lang laufender Prozesse
- Systemprogramme erlauben auch interaktive SW-Entwicklung.
 - *Editor, Shell, Übersetzer, Debugger*
- Platten und Dateisysteme erlauben jederzeit Zugriff auf Programme und Daten



Quelle: DIGITAL Computing Timeline

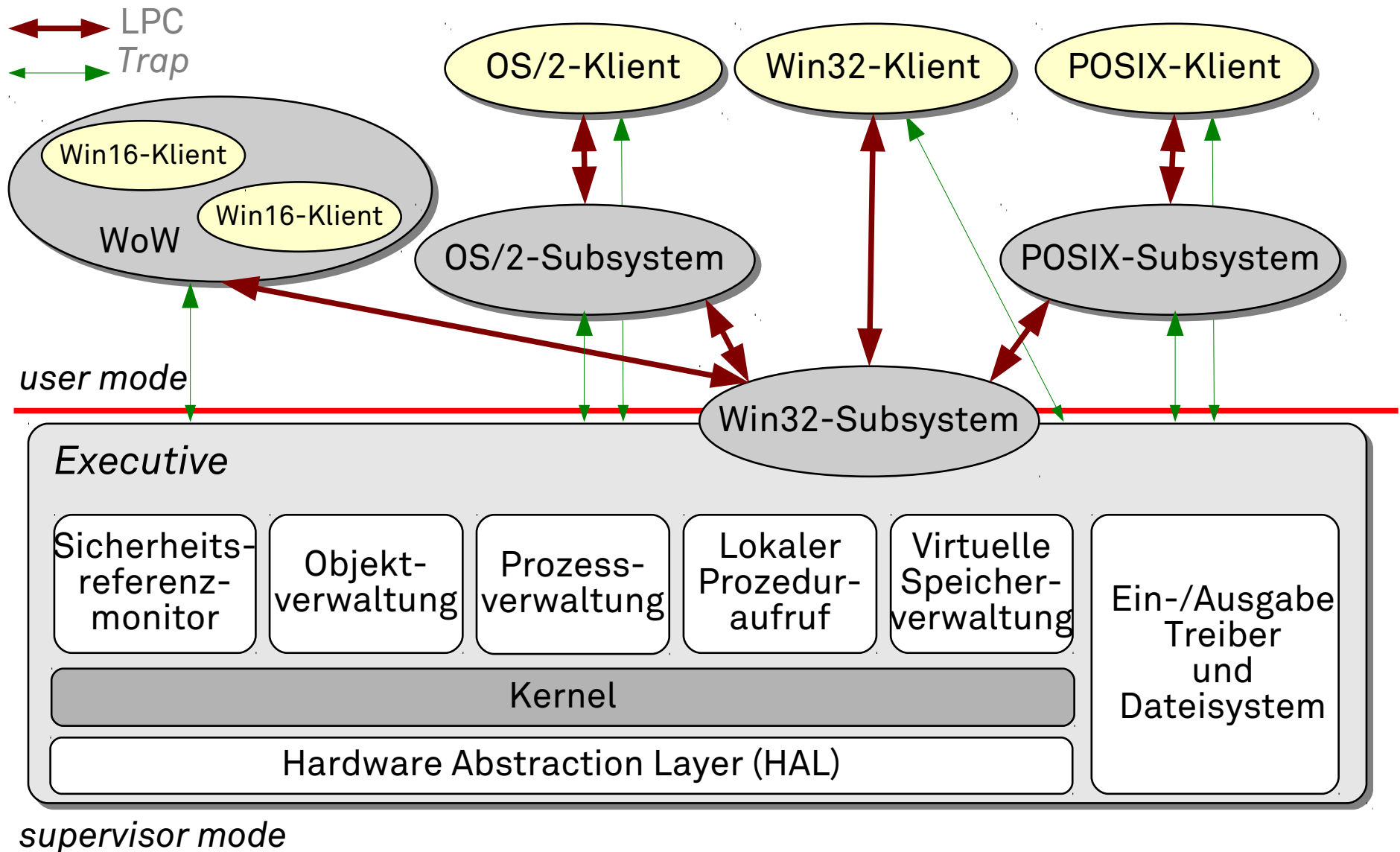


UNIX-Systemstruktur





Windows-Systemstruktur





Weiter geht's nächste Woche mit ...

- Systemabstraktionen im Überblick
 - Prozesse
 - CPU-Zuteilung
 - Synchronisation und Verklemmungen
 - Interprozesskommunikation
 - Speicherverwaltung
 - Arbeitsspeicher
 - Hintergrundspeicher
- und der ersten (praktischen) Übungsaufgabe
- **Übungsanmeldung nicht vergessen!**