



TECHNISCHE  
UNIVERSITÄT  
DRESDEN

Fakultät Informatik Institut für Systemarchitektur, Professur für Betriebssysteme

# BETRIEBSSYSTEME UND SICHERHEIT

*Einführung*

<https://tud.de/inf/os/studium/vorlesungen/bs>

HORST SCHIRMEIER

# Inhalt

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

## Literatur

Silberschatz, Chap. 1,  
„Introduction“

Tanenbaum, Kap. 1,  
„Einführung“

# Inhalt

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

## Literatur

Silberschatz, Chap. 1,  
„Introduction“

Tanenbaum, Kap. 1,  
„Einführung“

# Zweigeteiltes Modul

- **Betriebssysteme**

- Horst **Schirmeier**
- Professur für Betriebssysteme
- **Di 09:20** HSZ/HS4 + YouTube-Stream
- Folien und Aufzeichnungen auf der Webseite zur Vorlesung



- ... und **Sicherheit**

- Florian **Tschorsch**
- Professur für Privacy and Security
- **Fr 09:20** HSZ/AUD, **kein Streaming**
- Folien und Aufzeichnungen in OPAL



# Übungsaufgaben

- Wöchentliches **Übungsblatt** auf der Webseite
  - praktisches Anwenden von Vorlesungswissen
  - aber auch darüber hinaus (Selbststudium)
  - Kein Einreichen, sondern aktive Teilnahme an Übungen
- Aktuell: *Blatt 00 – Einführung zu UNIX*
  - Ziel: Erste Erfahrung sammeln, z.B. mit Hilfe der Linux-Entwicklungsumgebungs-VM von der Webseite

# Übungen

- ab **2. Vorlesungswoche** Übungstermine
  - gemeinsames Entwickeln von Lösungen, teilweise auch zu Klausuraufgaben
  - 9 Termine zur Auswahl (einer davon auf Englisch, ein anderer online/BBB)
  - **Einschreibung über OPAL ab 18:00**, limitierte Teilnehmerzahlen
  - zusätzlich: in Gruppe „Betriebssysteme und Sicherheit“ einschreiben

**TutorInnen-Team:** André, Antonia, Eric, Robert, Robin, Tianhao, Till/Leonie

# Kommunikation

- Interaktion **während der Vorlesung**:
  - Hand heben :-)
  - oder Matrix (z.B. über <https://matrix.tu-dresden.de> oder <https://matrix.fachschaften.org>):  
**#bus-vorlesung:tu-dresden.de**
  - oder anonym + online über <https://frag.jetzt>  
(Raum-Code: **94066703**)
- Jederzeit:
  - Matrix: **#bus-vorlesung:tu-dresden.de**
  - Forum in OPAL



# Feedback

- Bitte nicht erst am Veranstaltungsende!
- Kommentare/Anregungen zu Organisation, Vorlesung, Übung?
  - persönlich
  - per PN (Matrix)
  - per eMail
  - ... oder über den „Anonymen Briefkasten“



## BETRIEBSSYSTEME UND SICHERHEIT

Dozent	Prof. Dr.-Ing Horst Schirmeler Prof. Dr. Florian Tschorsch
Modul	INF-B-380, INF-LE-EUI, IST-05-PF-HS
Umfang und Art	4 SWS Vorlesung, 2 SWS Übung (Deutsch)
Turnus	Wintersemester
	<b>Start: 15.10.2024</b>
Zeit und Ort	Teil <b>Betriebssysteme</b> : Hybrid-Vorlesung Dienstag, 09:20 Uhr ( <a href="#">↗ HSZ/HS4</a> ), parallel im <a href="#">YouTube-Kanal des Dozenten</a> Interaktion im Hörsaal, per <a href="#">TUD Matrix</a> (#bus-vorlesung:tu-dresden.de) oder <a href="#">frag.jetzt</a>
	Teil <b>Sicherheit</b> : Vorlesung Freitag, 09:20 Uhr ( <a href="#">↗ HSZ/HS3</a> ) ( <b>kein Livestream</b> ), Aufzeichnungen in <a href="#">OPAL</a>
Klausur	TBD, i.d.R. am Ende des Prüfungszeitraums ( <a href="#">&gt; weitere Klausurhinweise</a> , <a href="#">&gt; Klausurvorbereitung</a> )
Wiederholungsklausur	TBD ( <a href="#">&gt; weitere Klausurhinweise</a> , <a href="#">&gt; Klausurvorbereitung</a> )
Feedback	per Mail, <a href="#">↗ über Matrix</a> (#bus-vorlesung:tu-dresden.de), oder über den <a href="#">anonymen Briefkasten</a>
Evaluation	<a href="#">↗ 2023/24</a> (Teil "Betriebssysteme")



# Klausur

- Nach Ende des Semesters (Ende Februar/Anfang März?)
  - keine **formale** Voraussetzung
  - Inhaltlich relevant ist der **Vorlesungs- und Übungsstoff**.
  - **Durchführung:** 90 Minuten Klausur
- Gilt für:
  - Bachelor-Studium Informatik und Medieninformatik
  - Diplom Informatik und Informationssystemtechnik
  - ***Alle anderen bitte melden!***
- Geplant: Termine für Probeklausur-Vorrechnen und Fragen

# Lernziele

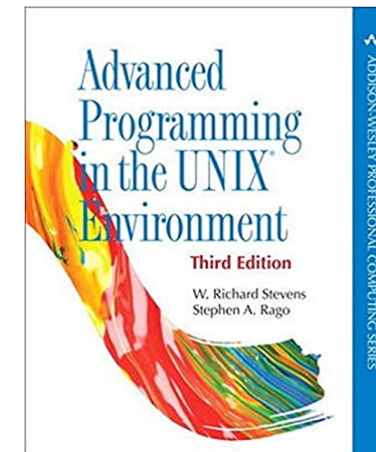
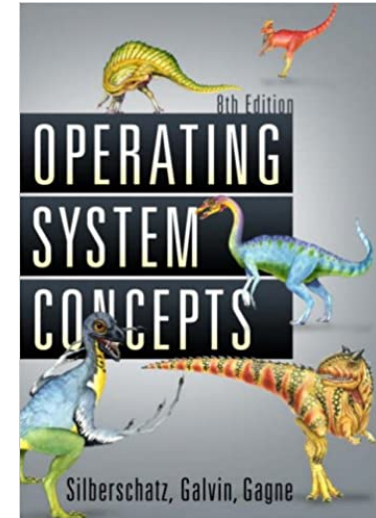
- Grundlagenwissen über **Betriebssysteme** erwerben
  - Funktionsweise und Struktur
  - Algorithmen und Implementierung
- Erfahrung mit **systemnaher Programmierung** sammeln
  - Übungsaufgaben in C unter UNIX
- Verständnis der Vorgänge in einem **Rechnersystem**
- Aktuelle **Trends** und **Herausforderungen**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

# Ausblick: Stoff von BuS

- Kontrollflussabstraktionen: Prozesse, *Threads*
- Prozessorzuteilung
- Kooperation und Konkurrenz von Kontrollflüssen
  - Synchronisation, *Deadlocks*
- Verwaltung und Virtualisierung des Hauptspeichers
- Ein- und Ausgabe
- Dateisysteme
- Sicherheit
- Multiprozessorsysteme

# Empfohlene Literatur

- [1] A. Silberschatz et al.  
*Operating System Concepts (8th Ed.)*.  
Wiley, 2008. ISBN-10: 0470128720
- [2] A. Tanenbaum, H. Bos  
*Moderne Betriebssysteme (4. Aufl.)*.  
Pearson, 2016. ISBN 978-3-86894-270-5
- [3] J. Gustedt  
*Modern C*.  
Manning, 2019. ISBN 9781617295812  
<https://modernc.gforge.inria.fr/>
- [4] R. Stevens et al.  
*Advanced Programming in the UNIX Environment (3rd Ed.)*,  
Addison-Wesley, 2013. ISBN-10: 0321637739



# Inhalt

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

# 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 **residenter 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 darunterliegende reale Maschine].“*

Tanenbaum [2]

# Vielfalt der Anforderungen

## **High Performance Computing**

Minimale Kommunikations-  
latenzen



## **Arbeitsplatz- systeme**

Intuitive Benutzer-  
oberfläche



## **Sichere Systeme**

Zugriffsschutz



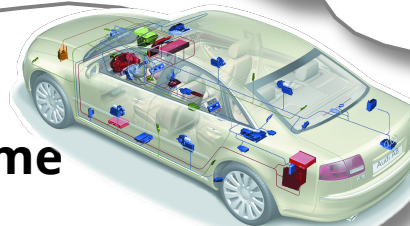
## **Echtzeitsysteme**

Vorhersagbares  
Zeitverhalten



## **Eingebettete und automotive Systeme**

Minimaler  
Speicherplatzbedarf



# Zwischenfazit

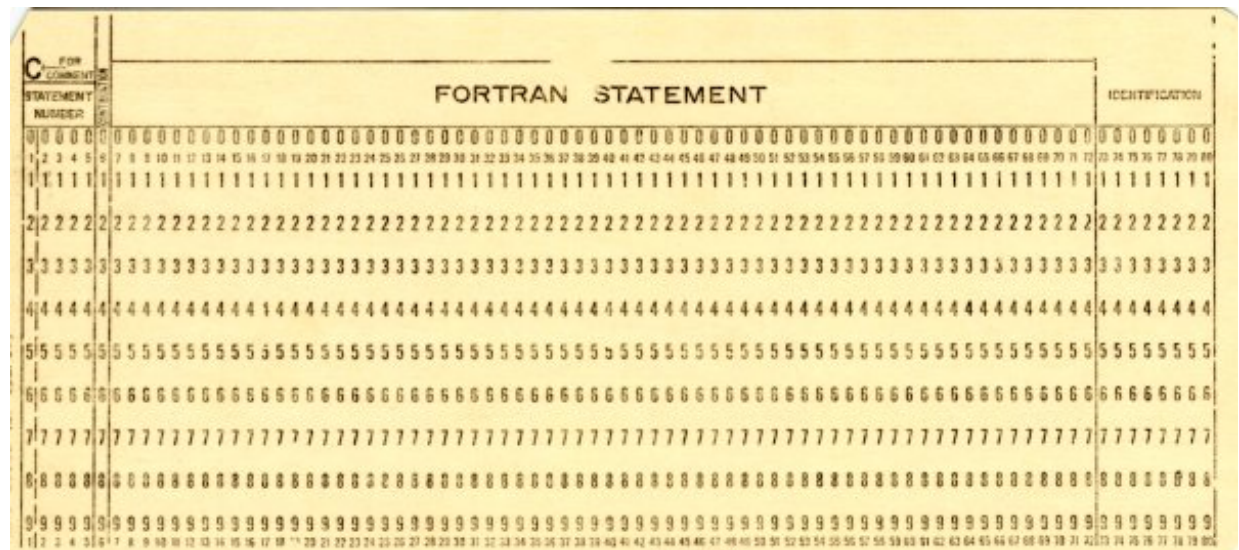
- Es gibt viele Auslegungen des Begriffs „Betriebssystem“.
- Festhalten kann man:
  - Das Betriebssystem **dient den Anwendern bzw. deren Anwendungsprogramm(en)** und nie dem Selbstzweck.
  - Es muss die Hardware genau kennen und den Anwendungen **geeignete Abstraktionen zur Verfügung stellen**.
- Hardware 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 typischen Anwendungen betrachten**.

# Inhalt

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

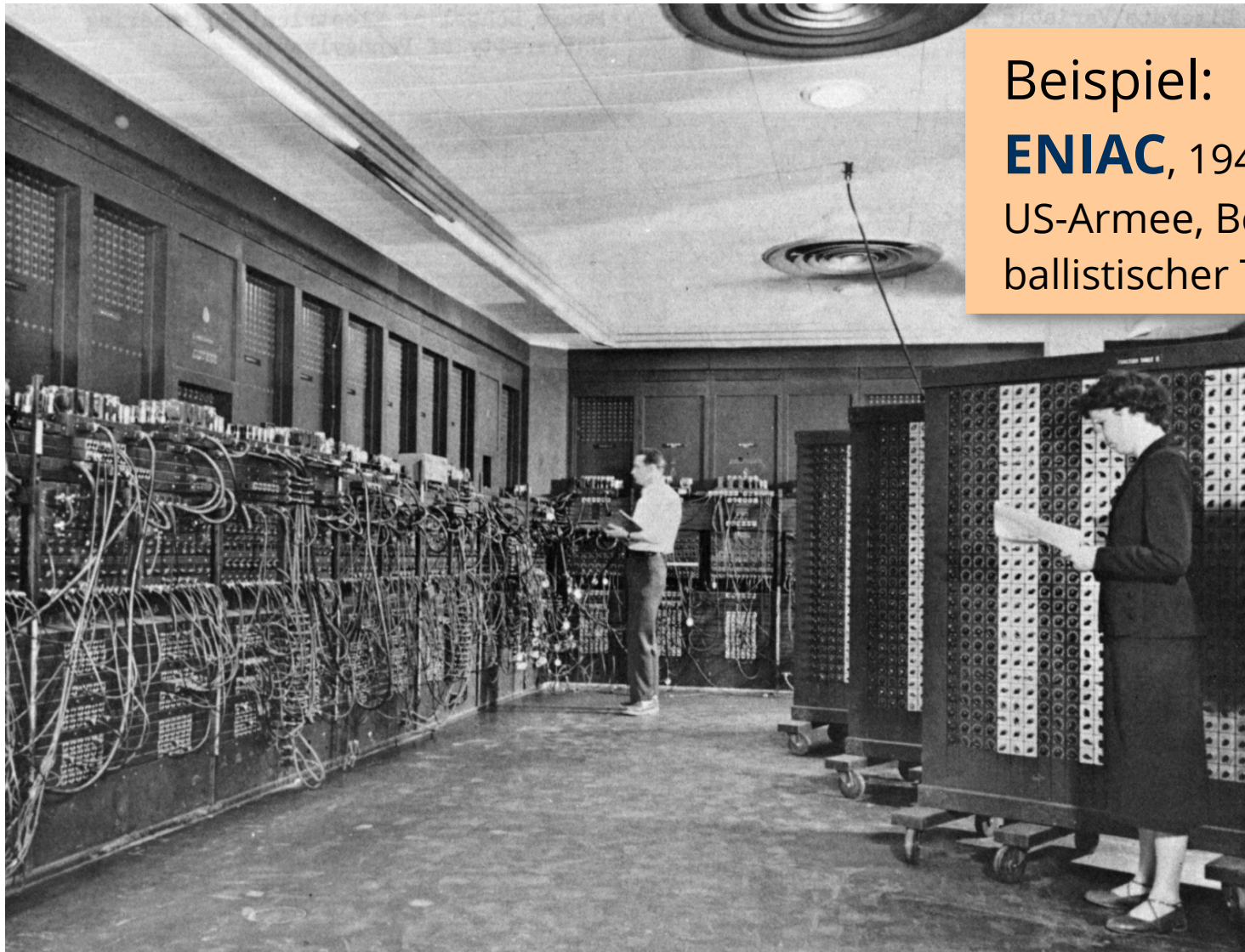
# Am Anfang stand die Lochkarte

- 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)

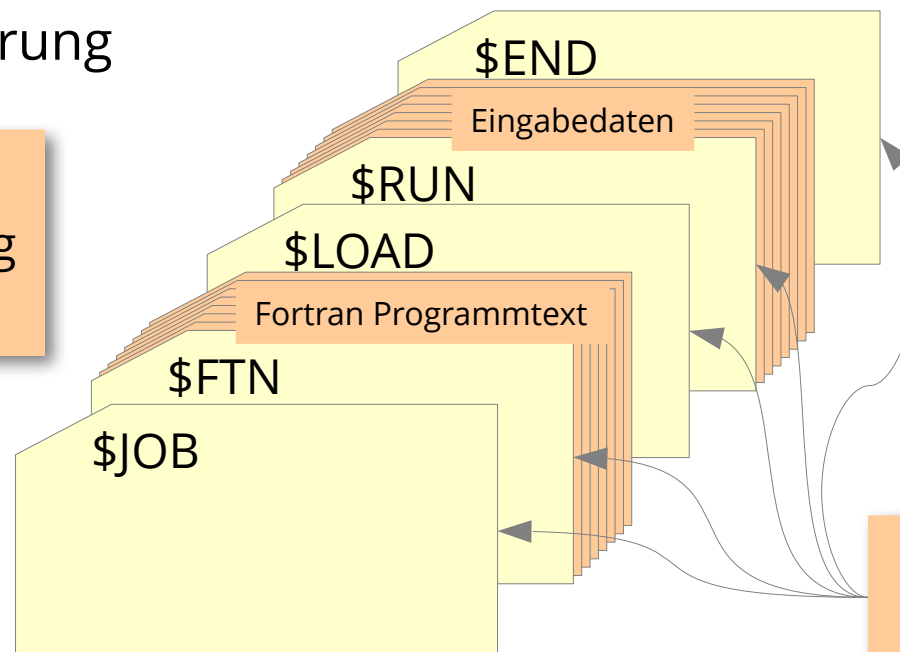
- Programmierung
  - i.d.R. in Maschinencode
  - Eingabe über Lochkartenleser, Ausgaben über Drucker
  - Fehleranzeige durch Kontrolllämpchen
- Rechnerzeituteilung auf Papierterminkalender
  - **Rechnerzeitverschwendung** durch zu großzügige Reservierung oder Abbruch wegen Fehler
- Minimale Auslastung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

**Monitor**

Arbeitsspeicher

Gerätetreiber

Sequentielle  
Job-Steuerung

Steuersprach-  
interpreter

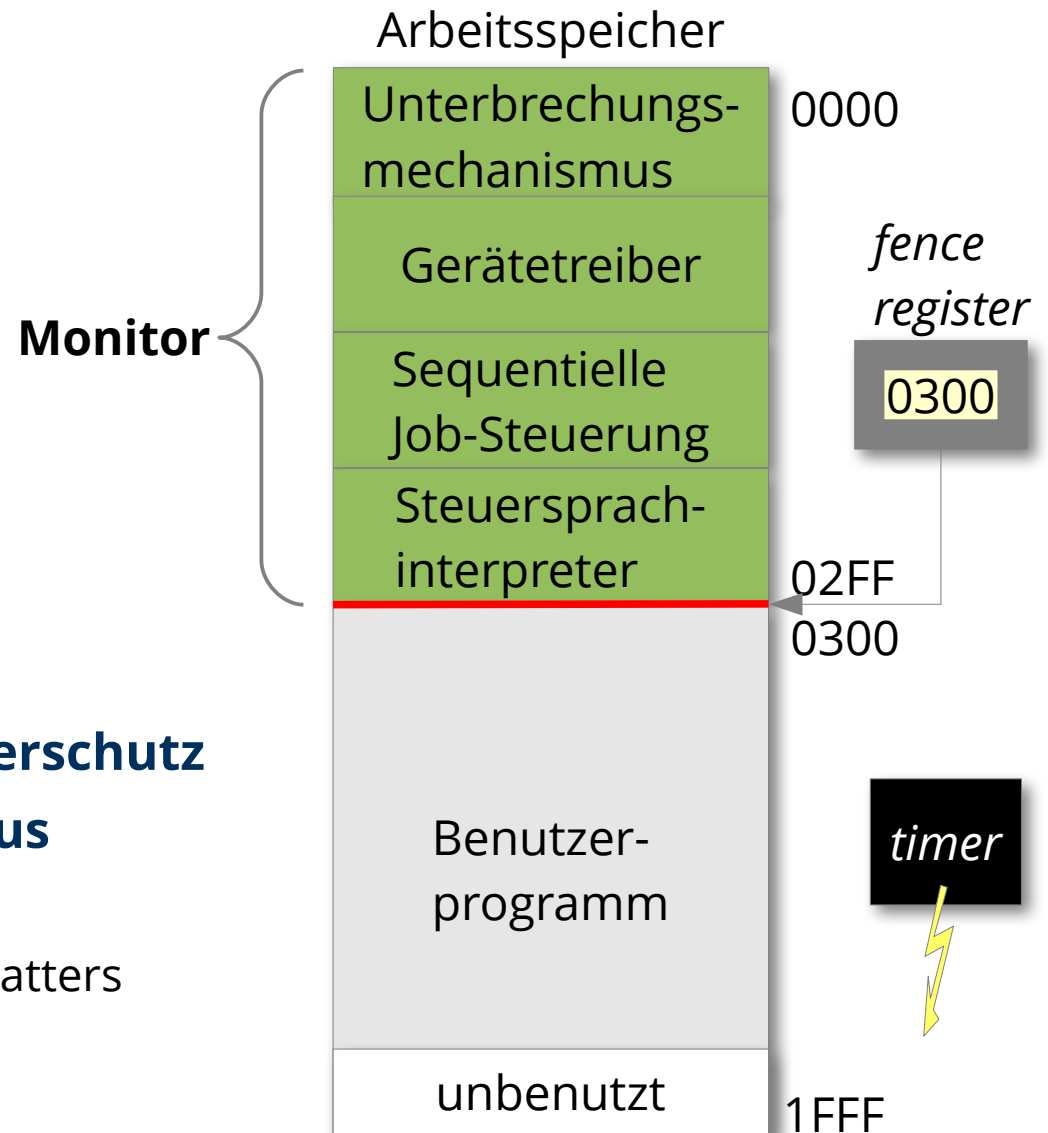
Benutzer-  
programm-  
bereich

- **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)

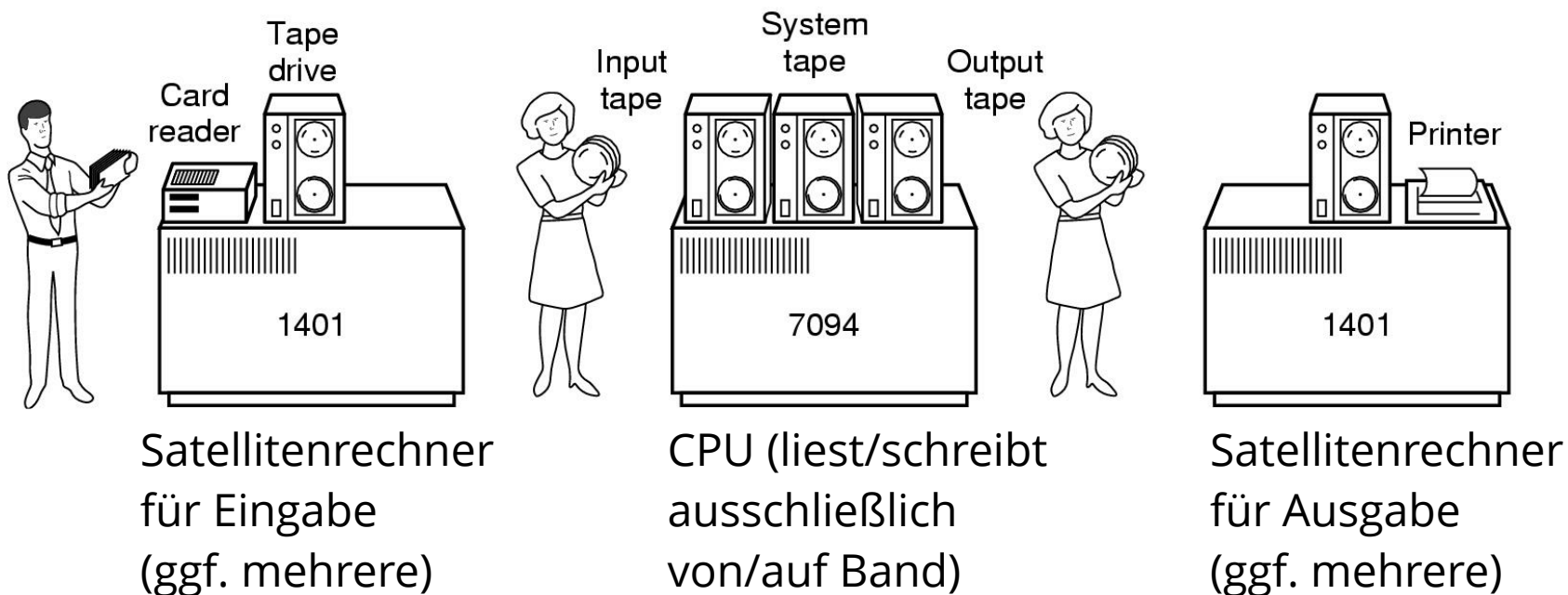
## Lösungen:

- 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



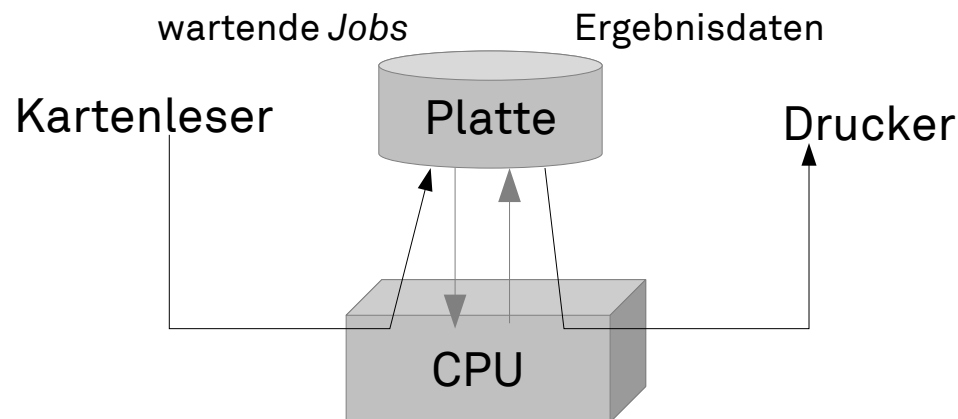
# Der Ein-/Ausgabe-Flaschenhals

- **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**

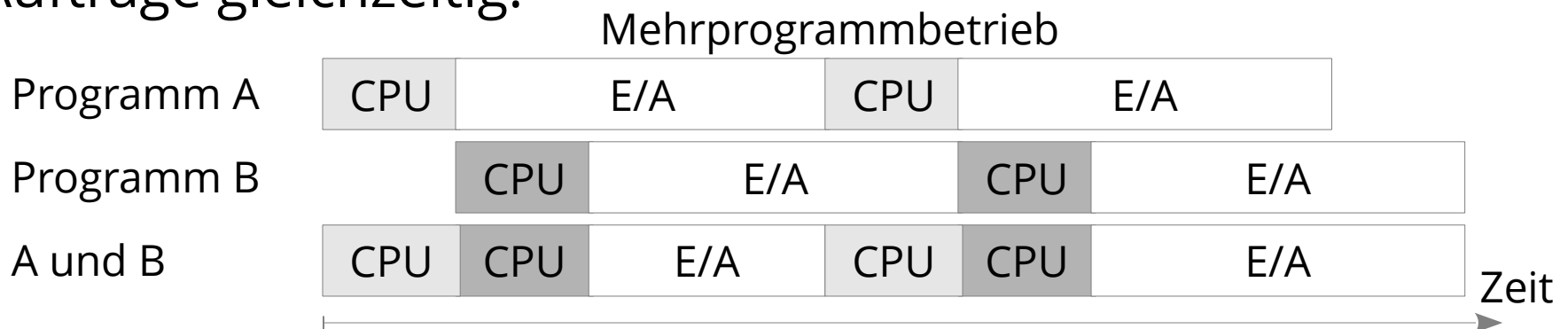


# Mehrprogrammbetrieb (ab 1965)

- 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.

Einprogrammbetrieb

## 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*)

A und B

CPU

CPU

E/A

CPU

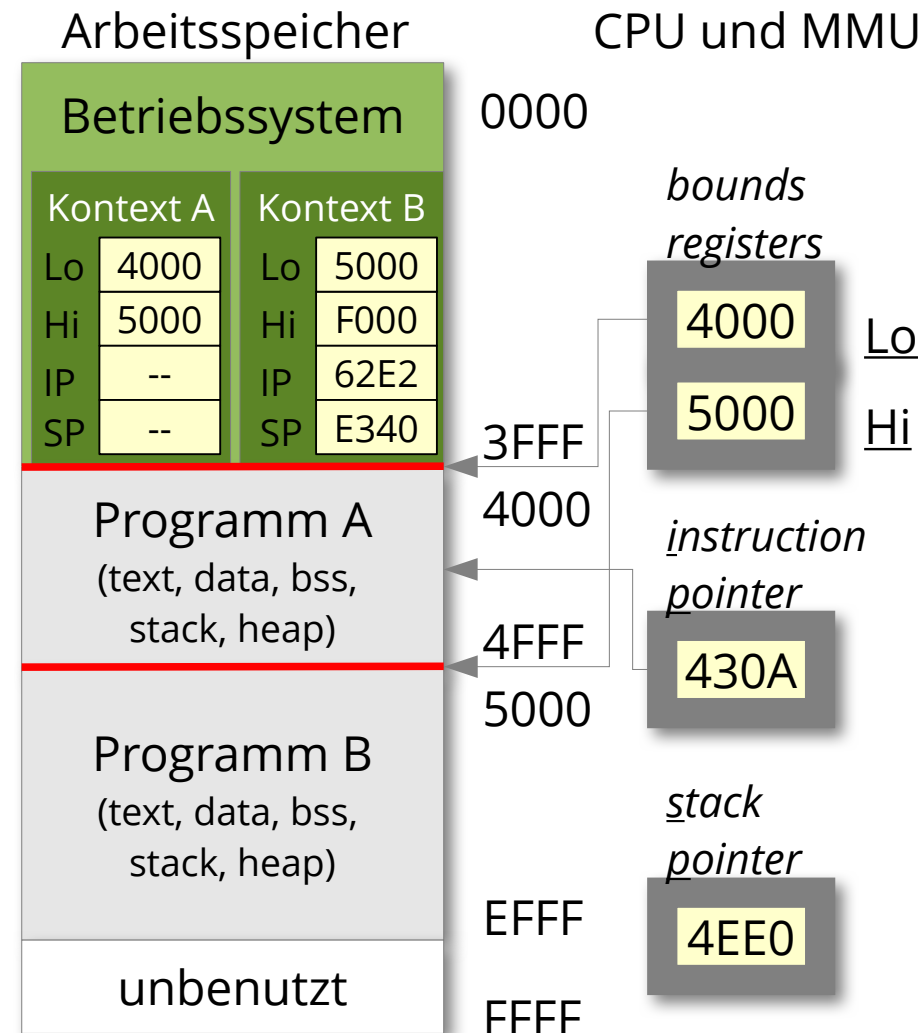
CPU

E/A

Zeit

# 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)

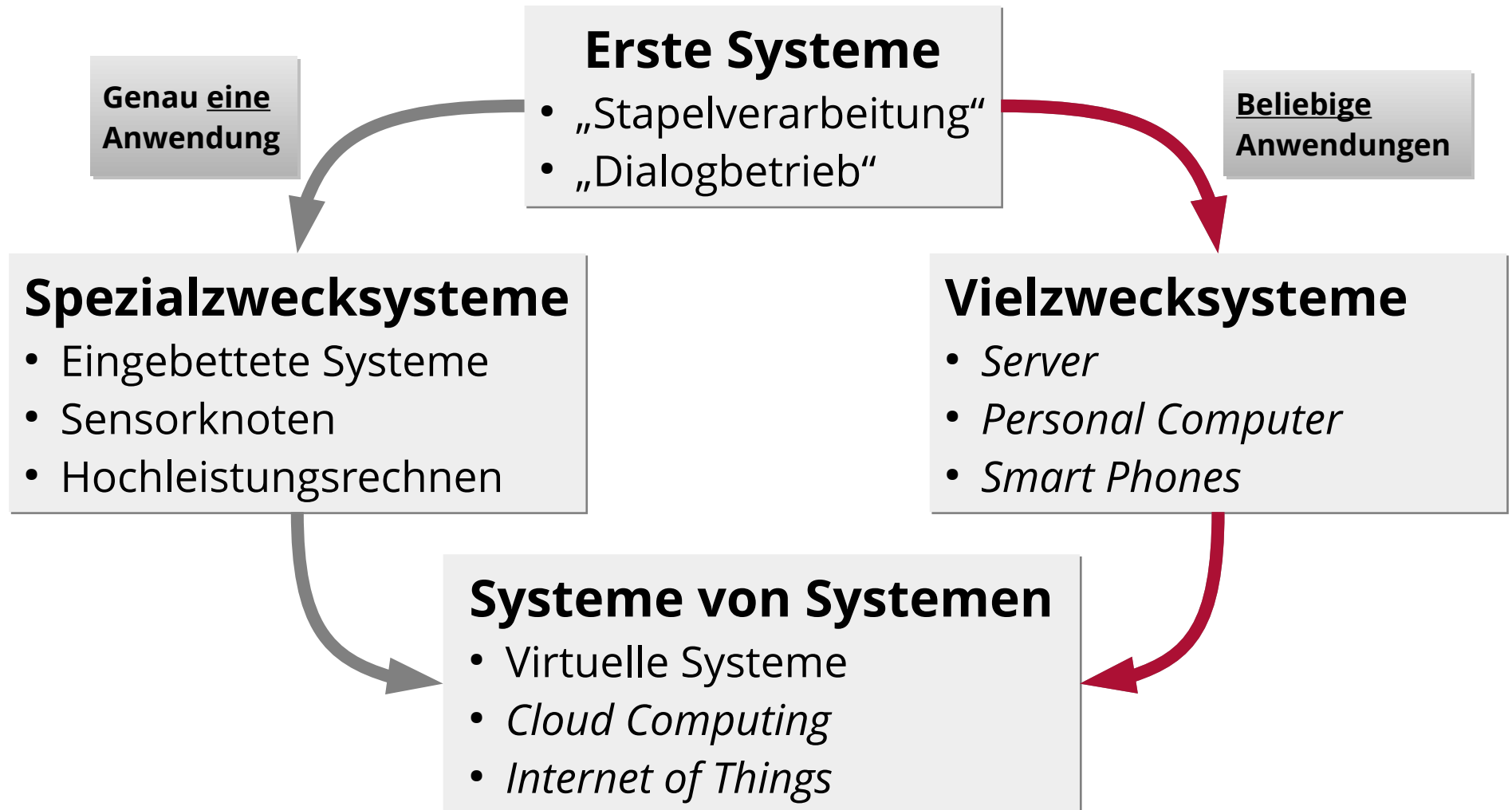
- **Neue Ein- und Ausgabegeräte** erlauben interaktive Software
  - Tastatur, Monitor, später Maus
- *Time-Sharing*-Betrieb
  - ermöglicht akzeptable **Antwortzeiten** für interaktive Nutzer
  - Zeitgeber-Unterbrechungen sorgen für **Verdrängung** (zu) lang laufender Prozesse
- Systemprogramme erlauben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



Quelle: DIGITAL Computing Timeline

# Weitere Entwicklung der Systeme

(Hardware, Betriebssysteme und Anwendungen „Hand in Hand“)



# Next Up

- Systemabstraktionen im Überblick
  - Prozesse
    - CPU-Zuteilung
    - Synchronisation und *Deadlocks*
    - Interprozesskommunikation
  - Speicherverwaltung
    - Arbeitsspeicher
    - Hintergrundspeicher
- **Nicht vergessen:**
  - **Übungs-Anmeldung** in OPAL
  - **Übungsblatt 0** und Linux-VM