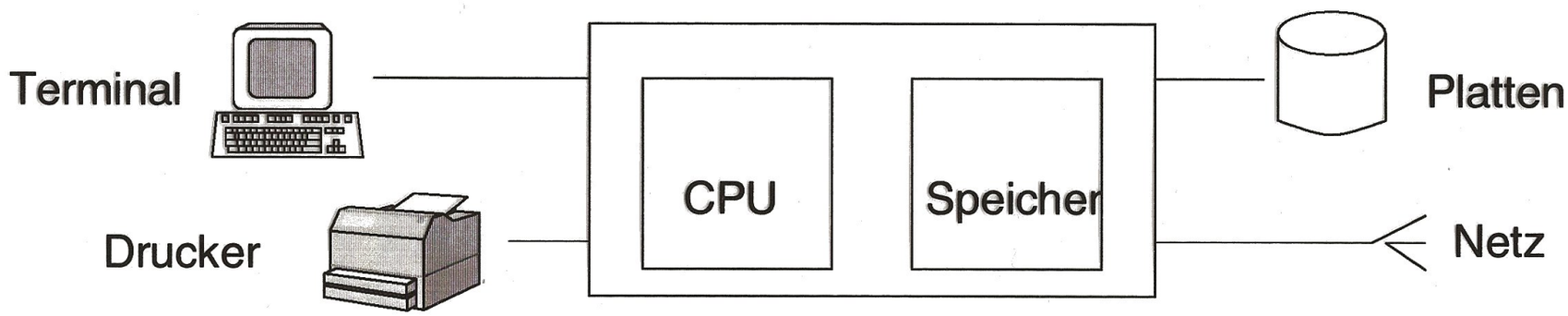
Work in progress

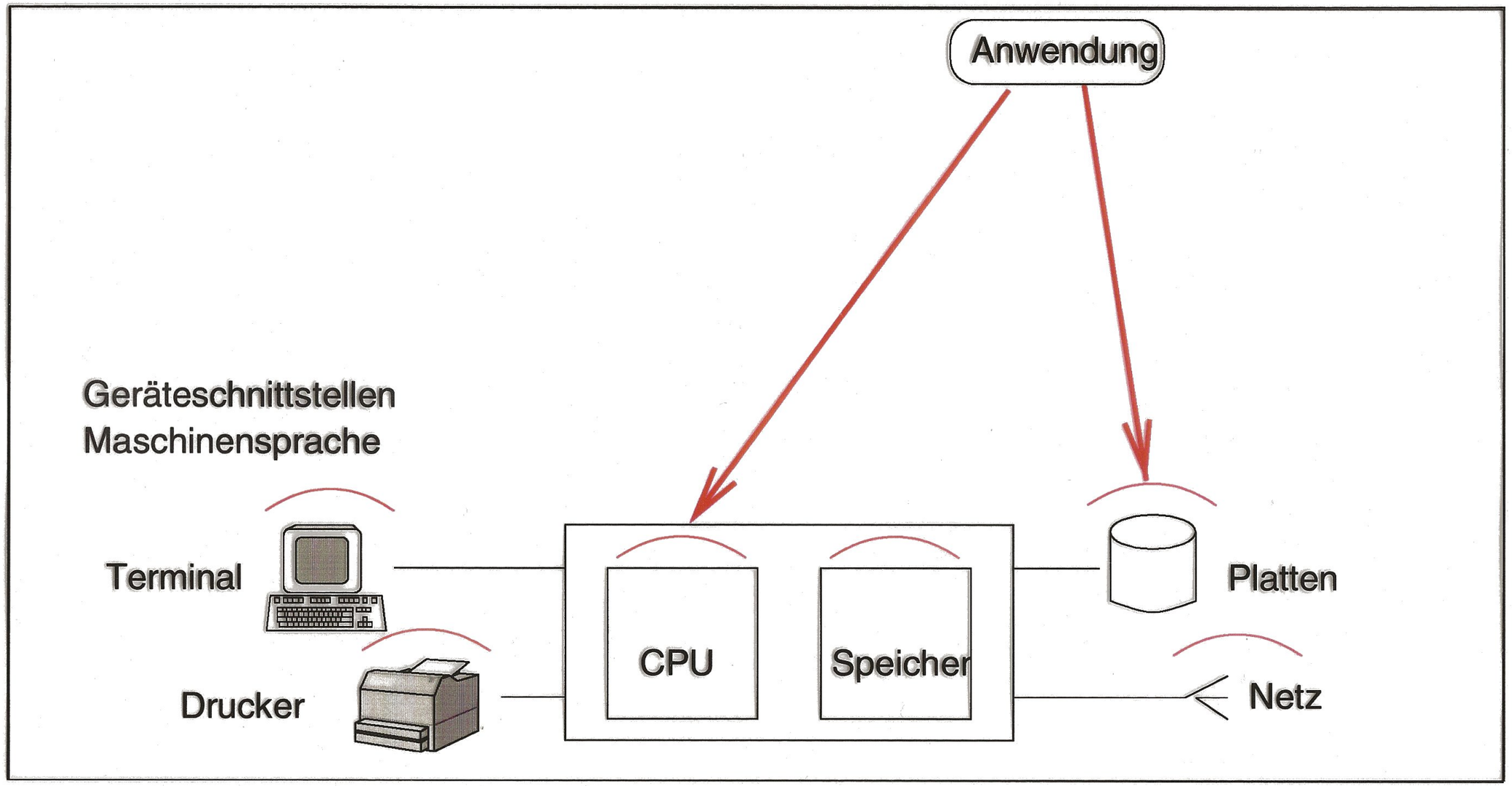
Überblick Betriebssysteme

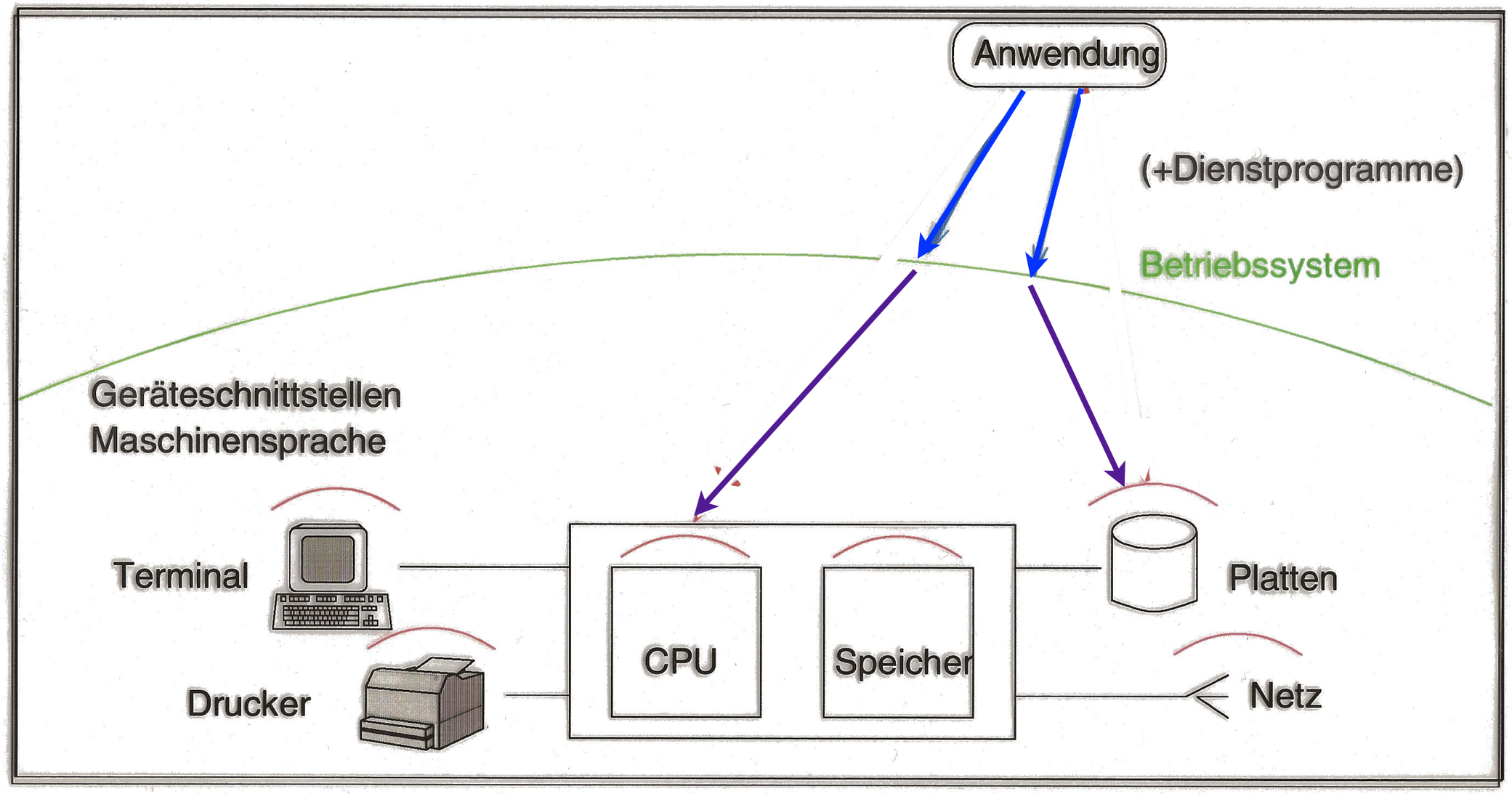
Ute Bormann, TI2

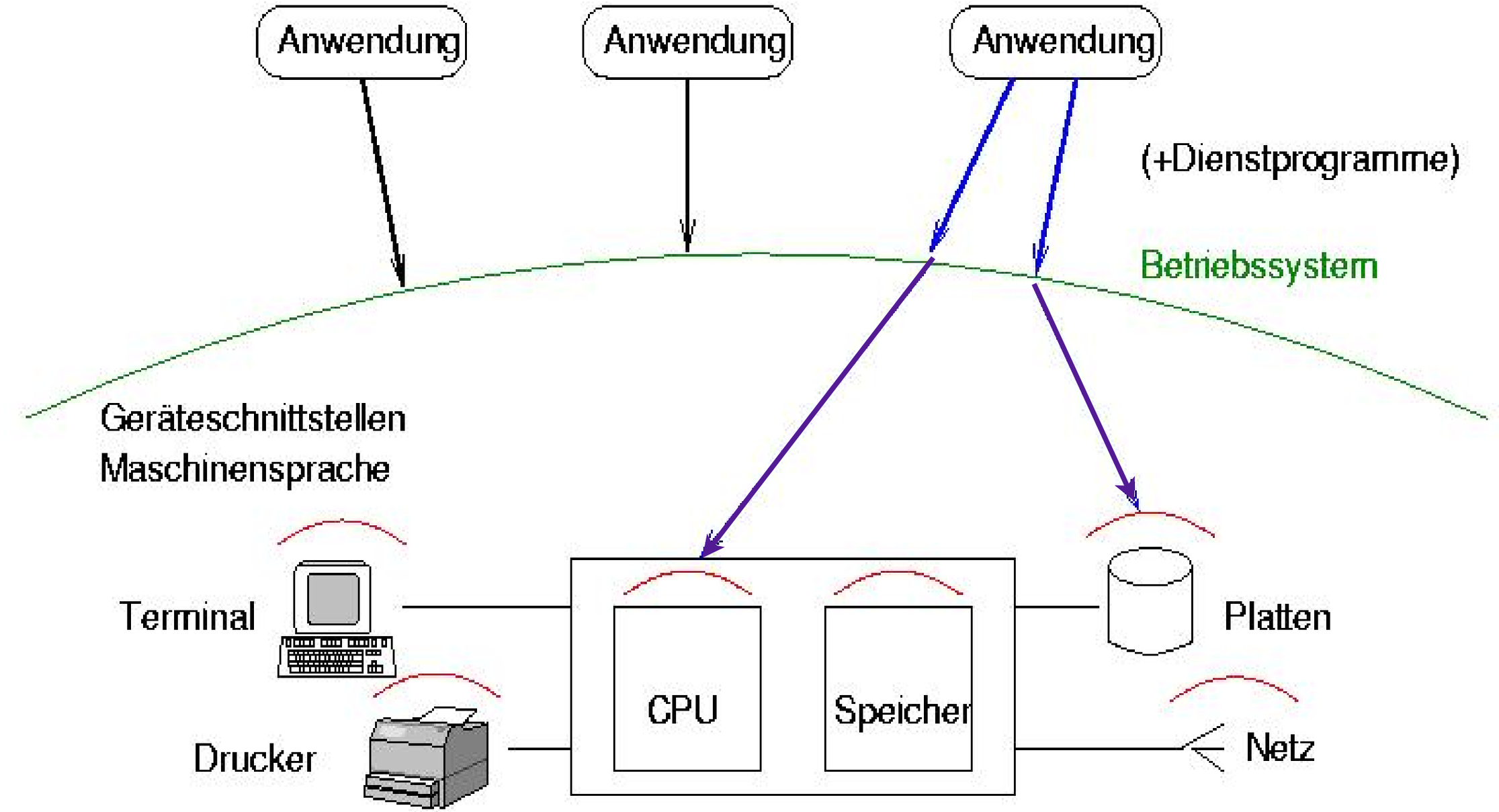
2023-10-13

Anwendung




Anwendung
Geräteschnittstellen
Maschinensprache
Terminal
Drucker
CPU
Speicher
Platten
Netz


Anwendung
(+Dienstprogramme)
Betriebssystem
Geräteschnittstellen
Maschinensprache
Terminal
Drucker
CPU
Speicher
Platten
Netz


Anwendung
Anwendung
Anwendung
(+Dienstprogramme)
Betriebssystem
Geräteschnittstellen
Maschinensprache
Terminal
Drucker
CPU
Speicher
Platten
Netz

Aufgaben eines Betriebssystems

- Abstraktion von Geräteeigenschaften
  - Geräteunabhängige Schnittstelle zu den Anwendungen
    ⇒ virtuelle Maschine
  - Geräteüberwachung und -steuerung
  - Datenhaltung

Aufgaben eines Betriebssystems

- Abstraktion von Geräteeigenschaften
  - Geräteunabhängige Schnittstelle zu den Anwendungen
    ⇒ virtuelle Maschine
  - Geräteüberwachung und -steuerung
  - Datenhaltung
- Unterstützung des Mehrbenutzerbetriebs
  - Betriebsmittelverwaltung
  - Zuteilungsstrategien
  - Kostenabrechnung
  - Schutz

  ⇒ komplex und hardwarenah

Aufgaben eines Betriebssystems

- Abstraktion von Geräteeigenschaften
  - Geräteunabhängige Schnittstelle zu den Anwendungen
    ⇒ virtuelle Maschine
  - Geräteüberwachung und -steuerung
  - Datenhaltung
- Unterstützung des Mehrbenutzerbetriebs
  - Betriebsmittelverwaltung
  - Zuteilungsstrategien
  - Kostenabrechnung
  - Schutz

  ⇒ komplex und hardwarenah
- Jedoch: Dienstprogramme auslagern

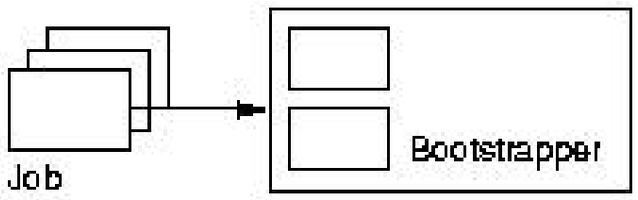
„Randbedingungen“ des Betriebssystem-Entwurfs

⇒ sehr hohe Qualitätsanforderungen

Zuverlässigkeit:	• Korrektheit • Sicherheit • Verfügbarkeit • Fehlertoleranz • Robustheit
Benutzerfreundlichkeit:	• Verständlichkeit • Angemessenheit • Vernünftiges Fehlerverhalten
Wartbarkeit und Flexibilität:	• Testbarkeit • Erweiterbarkeit • Adaptierbarkeit • Portabilität
Leistungsfähigkeit:	• Effektivität • Effizienz
Kosten	

Geschichtlicher Überblick

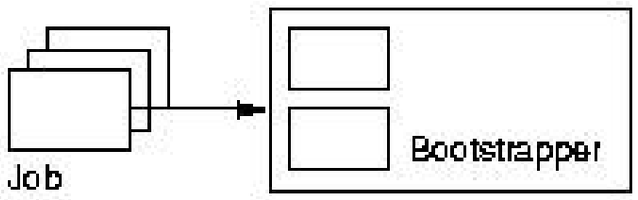
- Einfacher Stapelbetrieb (Batch) ≈ 1955/60



\+ spezielle Karten zur Aktivierung von Compiler, Linker, …

Geschichtlicher Überblick

- Einfacher Stapelbetrieb (Batch) ≈ 1955/60



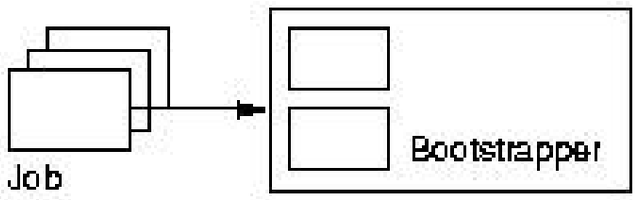
\+ spezielle Karten zur Aktivierung von Compiler, Linker, …

- Mehrfachstapelbetrieb ≈ 1962
  - Bessere Ausnutzung der Ressourcen (Hardware war teuer)

  ⇒ komplexere Verwaltungssoftware nötig

Geschichtlicher Überblick

- Einfacher Stapelbetrieb (Batch) ≈ 1955/60



\+ spezielle Karten zur Aktivierung von Compiler, Linker, …

- Mehrfachstapelbetrieb ≈ 1962
  - Bessere Ausnutzung der Ressourcen (Hardware war teuer)

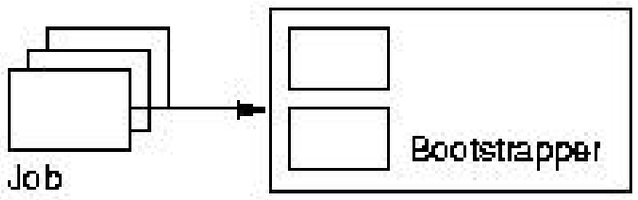
  ⇒ komplexere Verwaltungssoftware nötig

- Dialogbetrieb (Timesharing) ≈ 1965
  - Interaktiver Zugang zum Rechner ⇒ schnellere Antwortzeiten
  - sehr komplexe Betriebssysteme entstanden (IBM 0S/360, Multics, …)

  ⇒ nur von wenigen Experten beherrschbar, fehlerhaft…

Geschichtlicher Überblick

- Einfacher Stapelbetrieb (Batch) ≈ 1955/60



\+ spezielle Karten zur Aktivierung von Compiler, Linker, …

- Mehrfachstapelbetrieb ≈ 1962
  - Bessere Ausnutzung der Ressourcen (Hardware war teuer)

  ⇒ komplexere Verwaltungssoftware nötig

- Dialogbetrieb (Timesharing) ≈ 1965
  - Interaktiver Zugang zum Rechner ⇒ schnellere Antwortzeiten
  - sehr komplexe Betriebssysteme entstanden (IBM 0S/360, Multics, …)

  ⇒ nur von wenigen Experten beherrschbar, fehlerhaft…

⇒ Vereinfachungen:

- Unix (ab 1970)
- PCs (ab 1981, zunächst kein Mehrbenutzerbetrieb)

Seither im wesentlichen drei „Erweiterungen“

- Netwerkbetriebssystem ≈ 1985
  - Vernetzte PCs
  - Spezielle Diensterbringer (Client-Server-Konzept)

Seither im wesentlichen drei „Erweiterungen“

- Netwerkbetriebssystem ≈ 1985
  - Vernetzte PCs
  - Spezielle Diensterbringer (Client-Server-Konzept)

- Verteiltes Betriebssystem ≈ 1985
  - Multiprozessorsysteme
  - Transparente Verteilung von Aufträgen auf Prozessoren

Seither im wesentlichen drei „Erweiterungen“

- Netwerkbetriebssystem ≈ 1985
  - Vernetzte PCs
  - Spezielle Diensterbringer (Client-Server-Konzept)

- Verteiltes Betriebssystem ≈ 1985
  - Multiprozessorsysteme
  - Transparente Verteilung von Aufträgen auf Prozessoren

- Echtzeitbetriebssystem
  - insbesondere für eingebettete Systeme

Warum Unix?

- Universelles Betriebssystem
  (Time-Sharing- und Netzwerkbetriebssystem)
- Herstellerunabhängig
- Für sehr unterschiedliche Rechnertypen verfügbar
- Source-Code verfügbar (in C), vergleichsweise übersichtlich
- Im akademischen Bereich seit langem stark verbreitet
- Ideale Entwicklungsumgebung für Software
- Basis für viele Weiterentwicklungen

Warum Unix?

- Universelles Betriebssystem
  (Time-Sharing- und Netzwerkbetriebssystem)
- Herstellerunabhängig
- Für sehr unterschiedliche Rechnertypen verfügbar
- Source-Code verfügbar (in C), vergleichsweise übersichtlich
- Im akademischen Bereich seit langem stark verbreitet
- Ideale Entwicklungsumgebung für Software
- Basis für viele Weiterentwicklungen

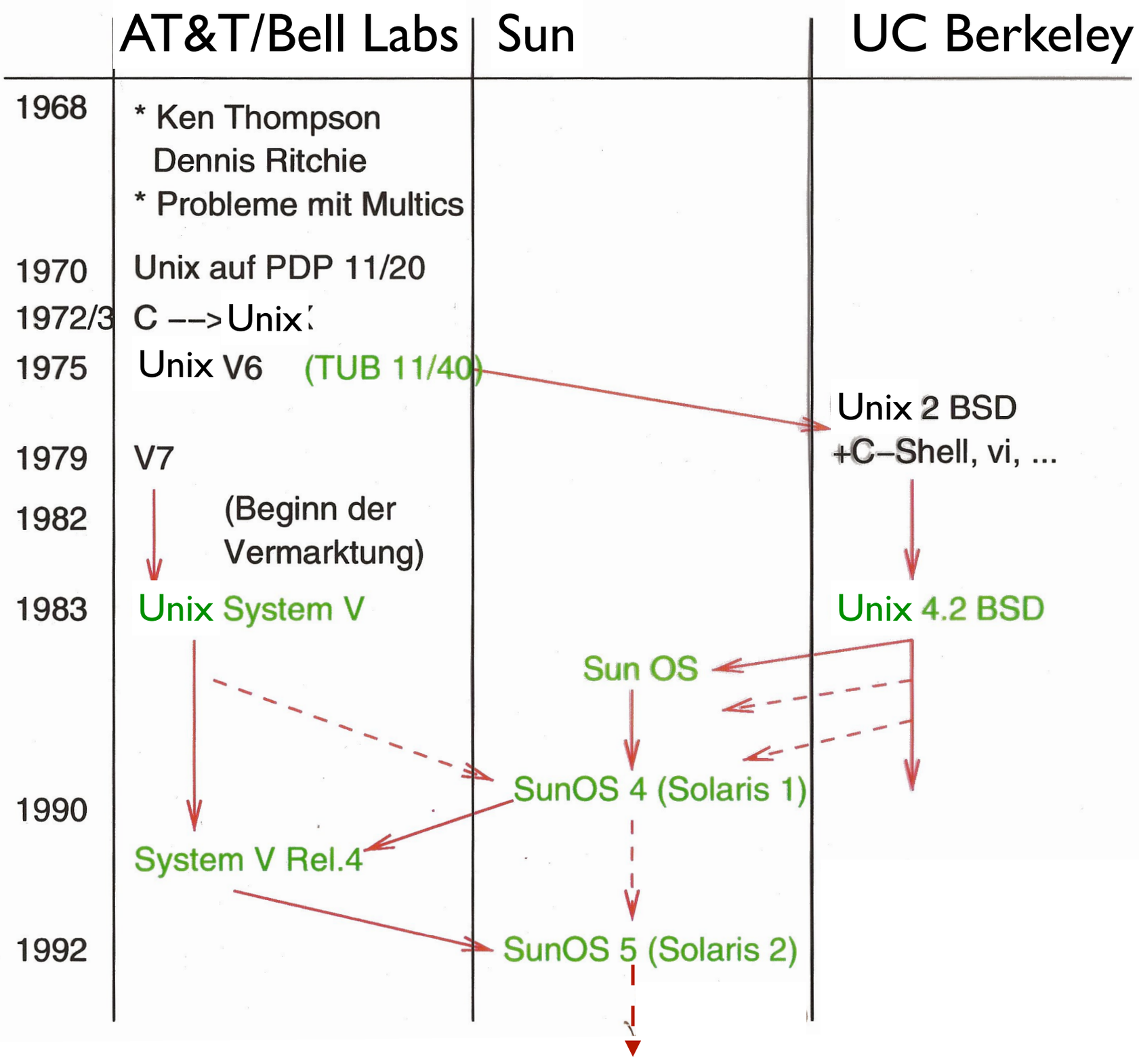
Probleme?

- Unsicher?
- Echtzeitunterstützung?
- Reales System (nicht alles didaktisch entwickelt)

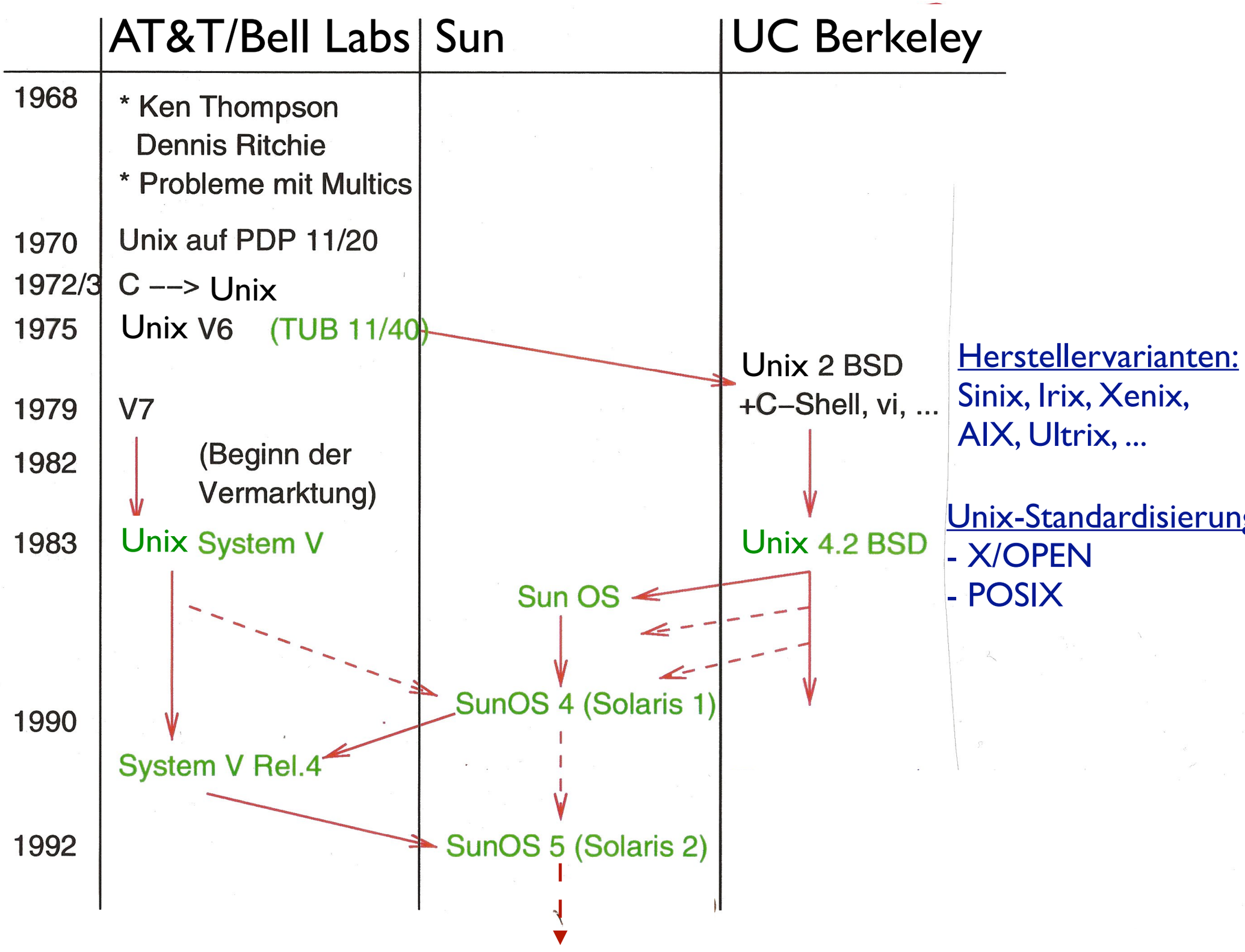
Die Geschichte von Unix (vereinfacht)

	AT&T/Bell Labs	Sun	UC Berkeley
1968	* Ken Thompson Dennis Ritchie * Probleme mit Multics		
1970	Unix auf PDP 11/20		
1972/3	C --> Unix		

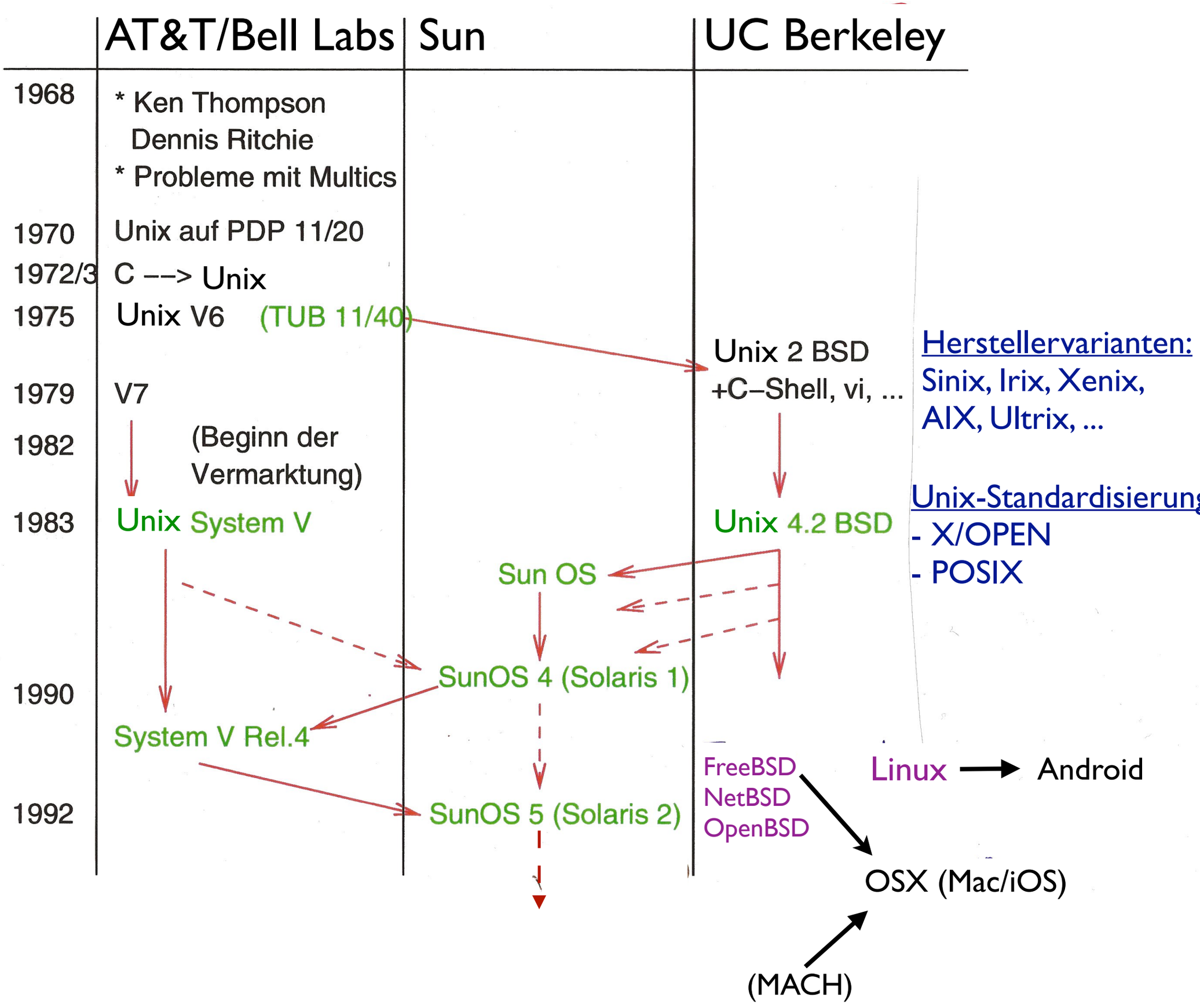
Die Geschichte von Unix (vereinfacht)



Die Geschichte von Unix (vereinfacht)



Die Geschichte von Unix (vereinfacht)



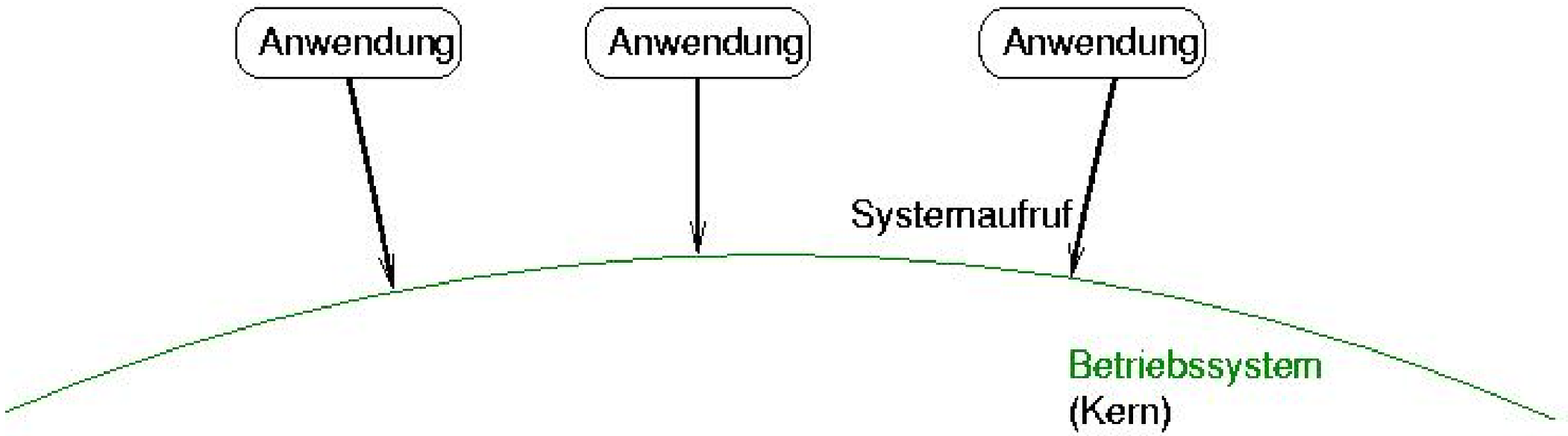
Zentrale Betriebssystemkonzepte

- Prozesse

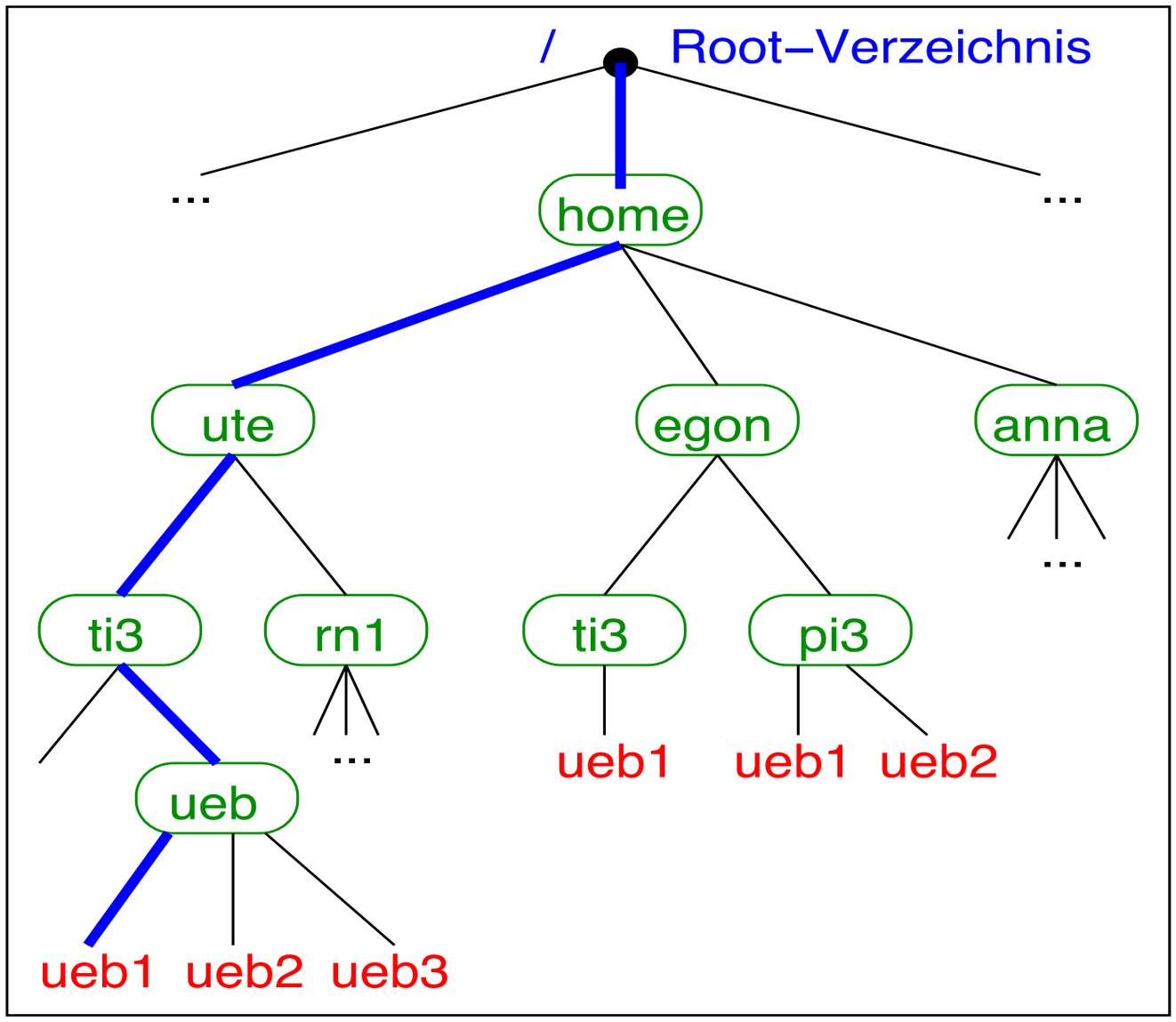
  Mehrere Programme in Ausführung
  ⇒ Verwaltung erforderlich:

  – Identifikation
  – Verarbeitungszustand
  – Datensätze

⇒ Kontrollfaden ⇒ Prozess



- Dateien (Files)
  - langlebige Datenobjekte
  - über eindeutige Namen identifizierbar
  - in Verzeichnissen organisierbar
  - von Prozessen aus zugreifbar/modifizierbar



- Kommandointerpreter (klassisch „Shell“ → Prozess)
  - „Nutzungsschnittstelle“
    ⇒ Absetzen von Aufträgen an das Betriebssystem

- Kommandointerpreter (klassisch „Shell“ → Prozess)
  - „Nutzungsschnittstelle“
    ⇒ Absetzen von Aufträgen an das Betriebssystem
  - Beispiel:

```
$ date
Mon Oct 8 15:37:19 MEST 2022
$
```

- Kommandointerpreter (klassisch „Shell“ → Prozess)
  - „Nutzungsschnittstelle“
    ⇒ Absetzen von Aufträgen an das Betriebssystem
  - Beispiel:

```
$ date
Mon Oct 8 15:37:19 MEST 2022
$ ls
org skript ueb
$ …
```

- Kommandointerpreter (klassisch „Shell“ → Prozess)
  - „Nutzungsschnittstelle“
    ⇒ Absetzen von Aufträgen an das Betriebssystem
  - Beispiel:

```
$ date
Mon Oct 8 15:37:19 MEST 2022
$ ls
org skript ueb
$ …
```

  - Mittlerweile natürlich vornehmlich grafische Nutzungsschnittstellen

Zusammenfassung

- Aufgaben von Betriebssystemen
- Kleiner geschichtlicher Überblick
- Beispiel: Unix
- Wichtige Betriebssystem-Konzepte

Überblick Betriebssysteme – Fragen

1. Welche zwei Hauptaufgaben hat ein Betriebssystem?
2. Was ist ein *Prozess*?
3. Welche Aufgabe hat ein *Kommando-Interpreter* (z.B. in Unix die Shell)?