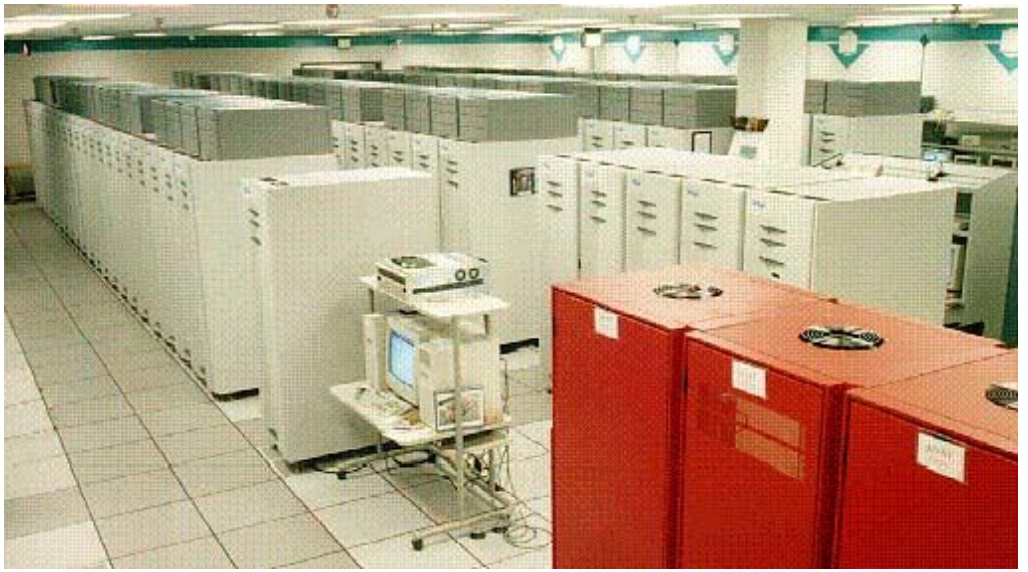
Massiv paralleles Rechnen

- koordinierte dynamische Einplanung von Prozessen desselben Jobs
  - alle Prozesse des parallelen Programms bilden eine ”Bande“ (gang)
  - jede Bande führt ein Programm aus, das mehrere Datenströme verarbeitet Þ single program, multiple data (SPMD)
  - die Bande ist einer bestimmten Anzahl von Prozessoren (fest) zugeteilt
  - Prozesse einer Bande werden als Einheit verwaltet: gang scheduling
- Jobs werden nebenläufig zueinander in eigenen Partitionen ausgef¨uhrt
  - massiv parallele Rechner bestehen aus hunderten/tausenden von Prozessoren
  - Banden teilen sich den Parallelrechner Zeitscheiben-basiert und verdrängend
- Schwachstelle: Skalierbarkeit, Programm-Mix

Systeme 5. Generation

Cray X-MP, 1990



Intel ASCI Red, 1997
4640 Knoten mit Intel Dual-PII Xeon,

Netzwerkbetrieb

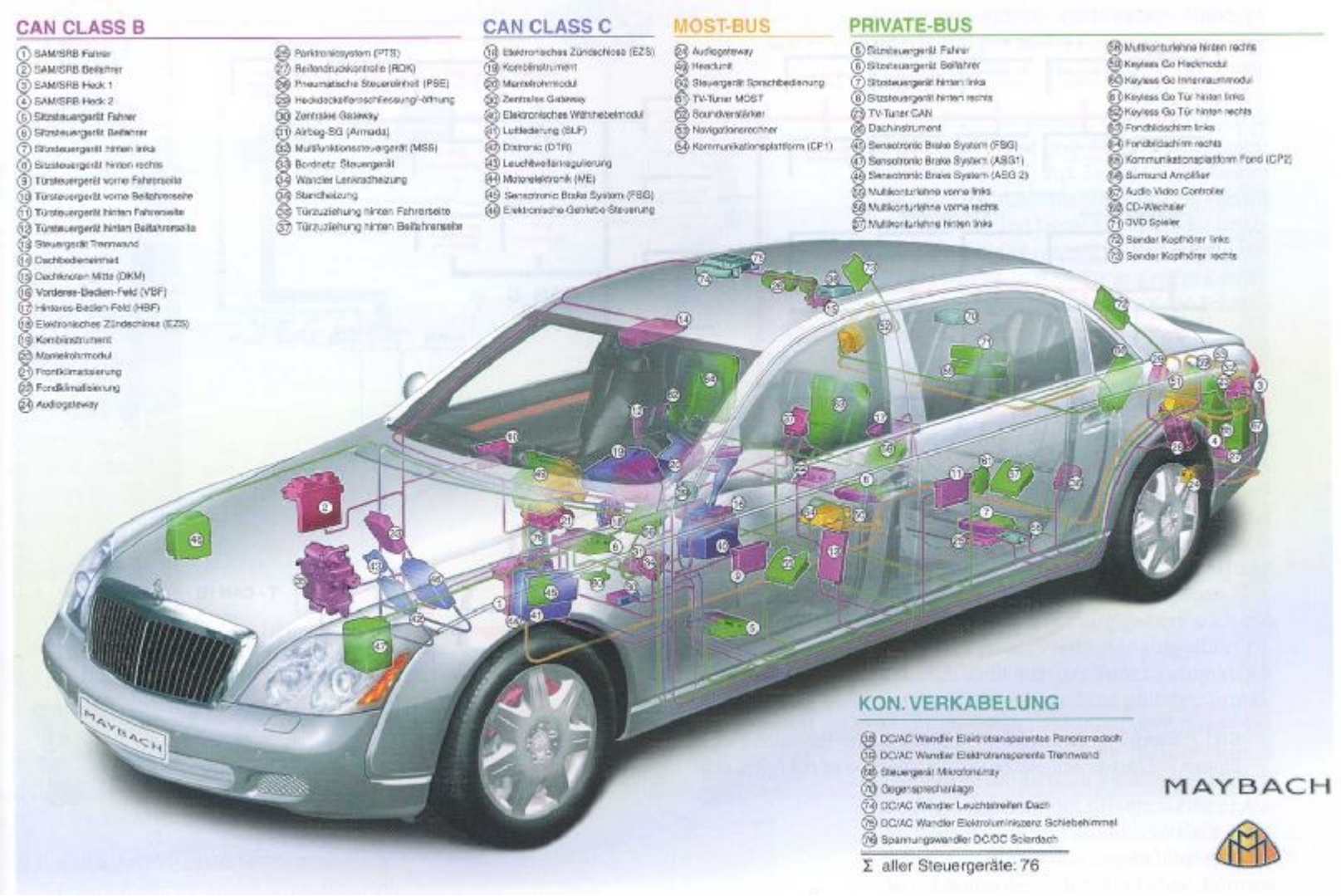
- Benutzer/Programme haben Zugriff auf Betriebsmittel eines Rechnerverbunds
  - das Netzwerk ist in verschiedenerWeise ”transparent“ (d. h., nicht sichtbar):
    - Zugriffs-, Orts-, Nebenläufigkeits-, Replikations-, Fehler-, Migrations-, Leistungs-, Skalierungstransparenz.
- entfernte Betriebsmittel werden über lokale Repräsentanten virtualisiert
- die verteilte Verarbeitung von Programmen wird (ein Stück weit) unterstützt
  - verteilte Kompilierung, verteilt ablaufendes make(1), ftp(1), rsh(1)
  - der Prozedur-Fernaufruf (*Remote Prodcedure Call*) liefert die Illusion eines lokalen Zugriffs
  - Betriebssystemkerne enthalten Kommunikationsprotokolle (TCP/IP)
- Middleware zw. Anwendungsprogramme und Betriebssystem schlägt die Brücke

„Integrationsbetrieb“

- Betriebssystem und Anwendungsprogramm(e) sind mit- bzw. ineinander verwoben, sie bilden eine (in sich geschlossene) Einheit:
  - ”miteinander“ im Sinne der Funktionalität
    - das Betriebssystem ist ”maßgeschneidert“ und ”anwendungsgewahr“
  - ineinander“ im Sinne der Repräsentation
    - das Betriebssystem liegt in Form einer (Quelltext-) Bibliothek vor
    - die Bibliothek wird mit dem Anwendungsprogramm zusammengebunden
- wenn Kompromisslösungen impraktikabel sind $\Rightarrow$ eingebettetes System
  - Jedes in einem Produkt versteckte Rechensystem, wobei das Produkt selbst jedoch kein Rechner ist: Kühlschrank, Mikrowelle, Kochplatte, Backofen, Wasserkocher, Waschmaschine, . . .

Eingebette Systeme

Eingebettete Systeme (2)



Quelle: DaimlerChrysler AG

Einbettbare Betriebssysteme

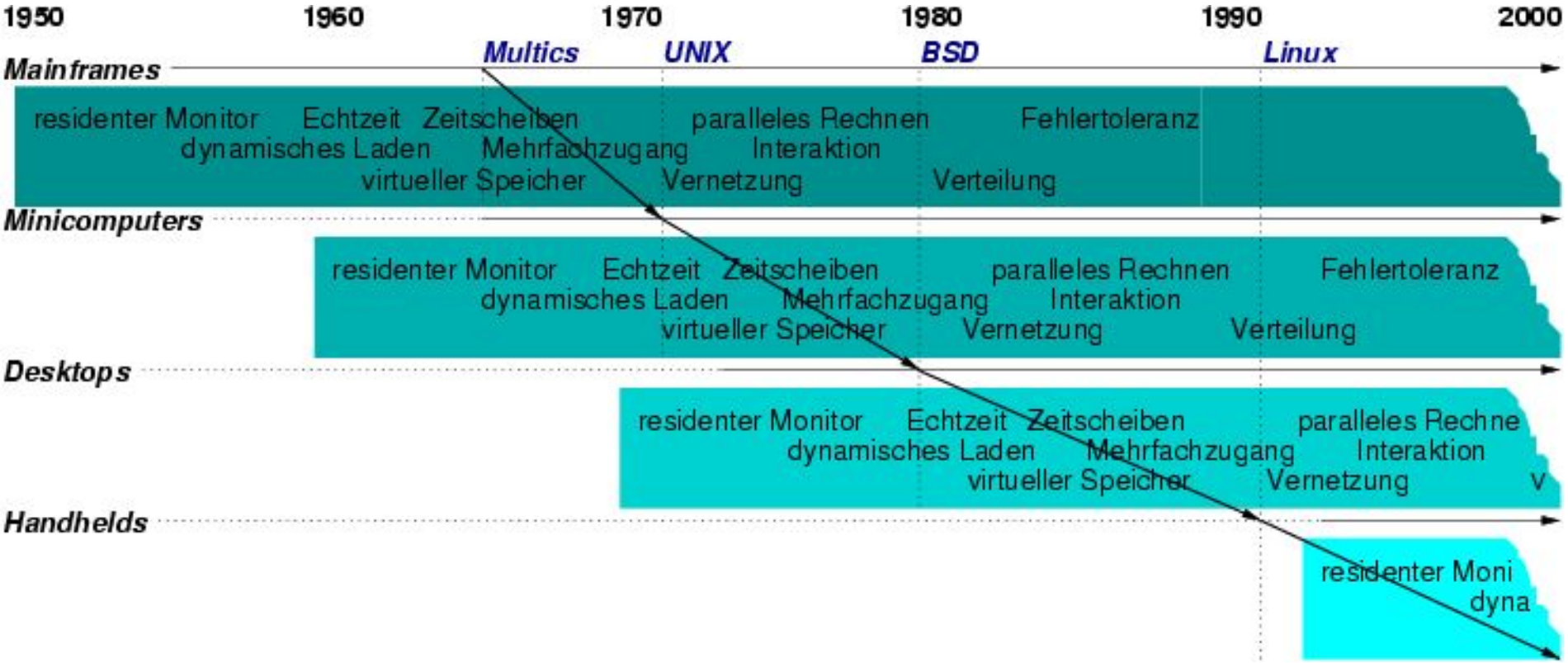
- BlueCat Linux, Embedix, HardHat Linux, Windows CE, Windows NT Embedded

$\Rightarrow$ ”Mist“ vielleicht, aber sicher kein ”Kleinvieh“ ;-)

. . . , C{51, 166, 251}, CMX RTOS, C-Smart/Raven, eCos, eRTOS, Embos, Ercos, Euros Plus, Hi Ross, Hynet-OS, LynxOS, MicroX/OS-II, Nucleus, OS-9, OSE, OSEK {Flex, Turbo, Plus}, OSEKtime, Precise/MQX, Precise/RTCS, proOSEK, pSOS, PURE, PXROS, QNX, Realos, RTMOSxx, Real Time Architect, RTA, RTX{51, 166, 251}, RTXC, Softune, SSXS RTOS, ThreadX, TinyOS, VRTX, VxWorks, . . .

- über 50% des Marktes sind proprietäre Lösungen für eingebettete Systeme

Migration von Konzepten



These: Mit Verfügbarkeit einer bestimmter Hardwarefähigkeiten werden ähnliche Betriebssystemkonzepte umgesetzt.

Stand der Kunst

Linux ”yet another UNIX-like operating system“, was soll’s . . .

- Entwicklungsprozess und -modell sind bedeutsam, der eigentliche ”Kick“

70er-Jahre Technologie — ohne Multics erreicht zu haben

Windows ”new technology“, wirklich?

- vor WNT entwickelte Cuttler VMS (DEC), m. a.W.: WNT = VMS + 1
- mit 94% Marktführer im PC-Sektor — für 2% des Prozessormarktes

MacOS X, ein vergleichsweise echter Fortschritt

- die Symbiose: solides FreeBSD auf solider Mikrokernbasis (Mach)
- Apple bringt PC-Technologie abermals voran — bei < 3% Marktanteil

Zusammenfassung

- größtenteils Systemsoftware entscheidet über die Betriebsart eines Rechners:

  {Stapel-, Off-Line-, Mehrprogramm-,
  Dialog-, Hintergrund-, Teilhaber-,
  Teilnehmer-, Mehrbenutzer-,
  Time-Sharing/Multi-Access-, Multiprozessor-,
  Netzwerk-, Echtzeit-, Integrations-}
  -betrieb

- Betriebssysteme sind Schlüssel zur Hardware
- Betriebssysteme haben sich in ihrer Geschichte großen Wandlungen unterzogen
  - von elementaren Funktionssammlungen hin zu komplexen Softwaregebilden
  - Mainframe-Betriebssysteme ”von damals“ laufen heute auf Handheld

Zusammenfassung (2)

- ein Betriebssystem ist schlechthin das ”Chamäleon“ aller Softwaremaschinen
  - die Anwendung bestimmt maßgeblich Erscheinungsbild und Funktionalität
  - eine für alle möglichen Einsatzbereiche gleich gute Lösung gibt es nicht
  - manche (nicht wenige) Anwendungsfelder dulden keine Kompromisse
- Betriebssysteme sind (nach wie vor — und weiterhin) Schlüsseltechnologie

Betriebssysteme „from scratch“

- Grundbegriffe:
  - Addressraum
    - physikalischer Adressraum
    - logischer Adressraum
    - virtueller Adressraum
  - Speicher
    - Hauptspeicher und Hintergrundspeicher
  - Prozess
    - schwer-, leicht oder federgewichtiger Aktivitätstrager
  - Datei
    - langfristige (permanente) Speicherung von Informationen
  - Koordinations- und Kommunikationsressourcen

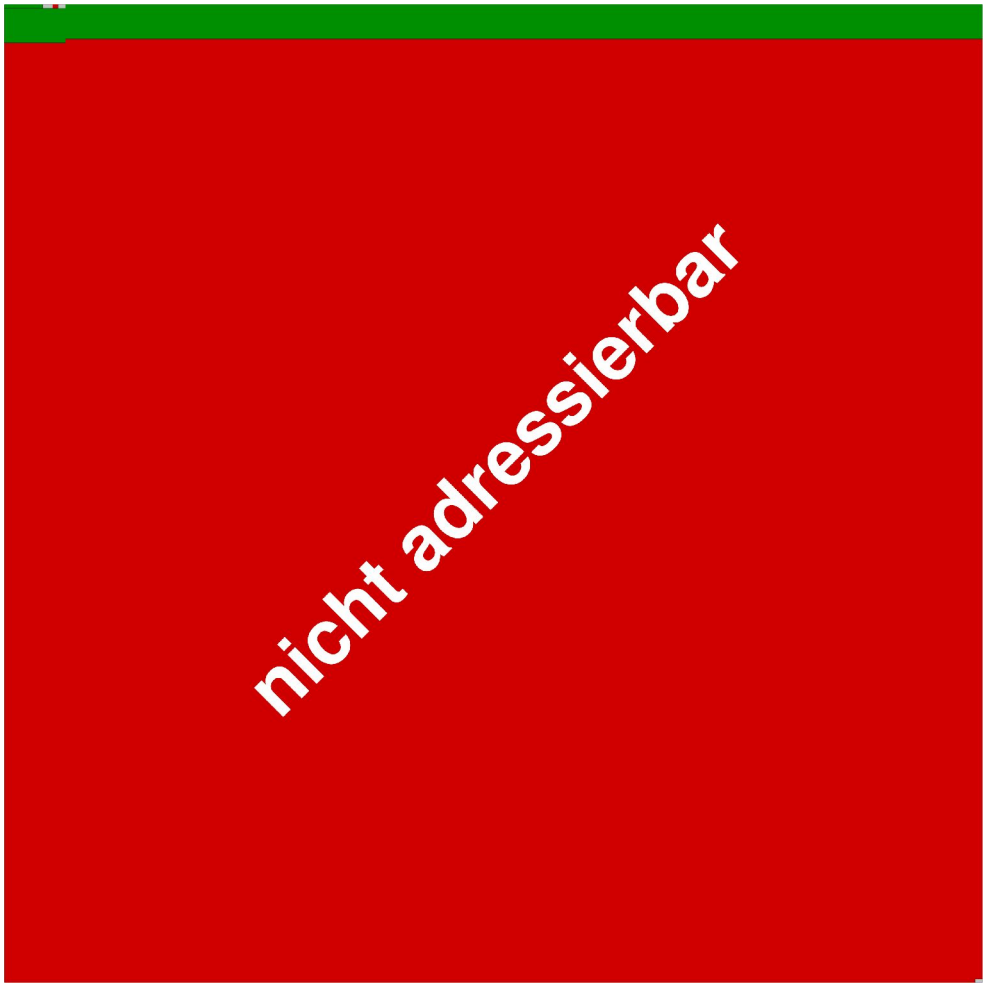
Adressraum

- physikalischer Adressraum $\Rightarrow$ Hardware
  - die Größe entspricht der Adressbreite der CPU: N Bit 2^N Bytes
  - nicht alle Adressen sind gültig und zur Programmspeicherung verwendbar
- logischer Adressraum $\Rightarrow$ Kompilierer, Binder, Betriebssystem
  - definiert einen zusammenhängenden, linear adressierbaren Programmbereich
  - alle Adressen sind gültig und zur Programmspeicherung verwendbar
  - ist typischerweise (sehr viel) kleiner als die Adressbreite der CPU hergibt
- virtueller Adressraum $\Rightarrow$ Betriebssystem
  - ein logischer Adressraum, dessen Größe der Adressbreite der CPU entspricht

Physikalischer Adressraum

Toshiba Tecra 730, 1996

Adressbereich	Belegung
00000000–0009ffff	RAM
000a0000–000c7fff	System
000c8000–000dffff	keine
000e0000–000fffff	System
00100000–090fffff	RAM
09100000–fffdffff	keine
fffe0000–ffffffff	System



Logischer Adressraum

- jedes Programm wird in einem eigenen logischen Adressraum ausgeführt
  - die Anfangs- und Endadressen aller logischen Adressräume sind (meist) gleich
  - logische Adressen sind auf die gültigen physikalischen Adressen abzubilden
- die erforderliche Adressabbildung erfolgt (typischerweise) mehrstufig:
  - Programmadresse $\Rightarrow$ logische Adresse
  - logische Adresse $\Rightarrow$ physikalische Adresse
- im Gegensatz zu physikalischen Adressen sind logische Adressen mehrdeutig

Adressraumsegmentierung

- gebräuchlich ist die logische Unterteilung des Adressraums in drei Segmente:

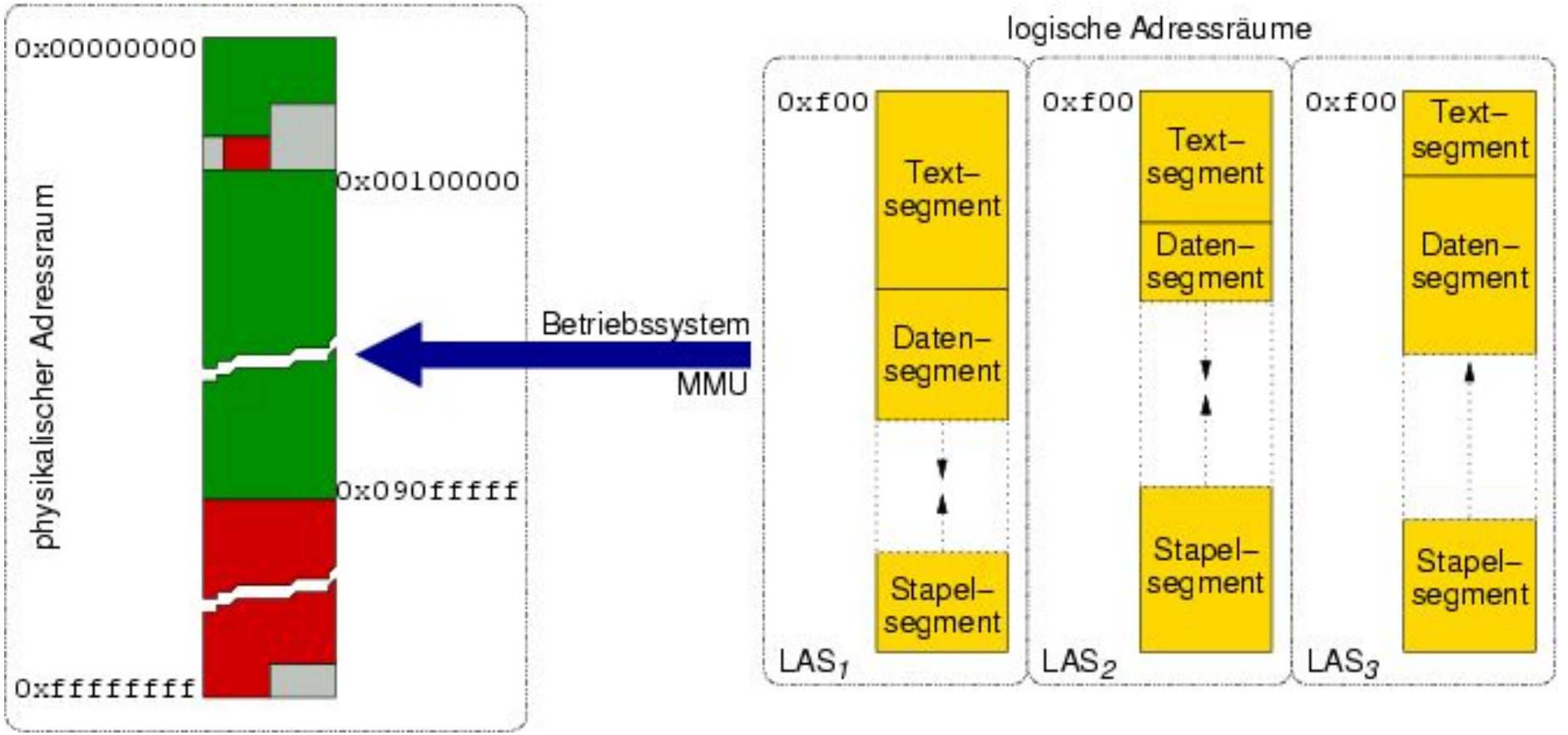
  Textsegment (*text segment*) enthält die Maschinenabweisunge (CPU) und andere Programmkonstanten + statisch/dynamisch

  Datensegment (*data segment*) kapselt initialisierte Daten, globale Variablen und ggf. eine Halde (heap) + statisch/dynamisch

  Stapelsegment (*stack segment*) beherbergt lokale Variablen, Hilfsvariablen und aktuelle Parameter + dynamisch

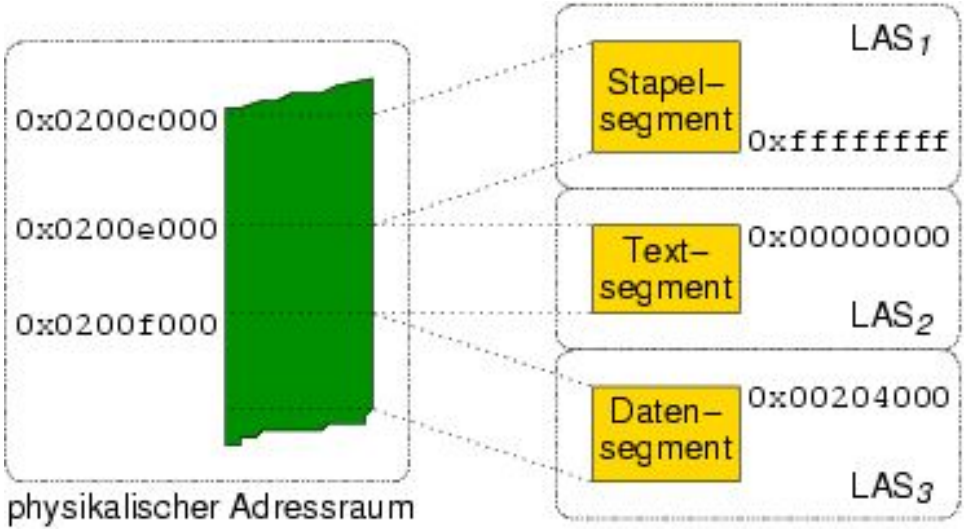
- typischerweise werden mehrere logische Adressräume nebeneinander verwaltet
  - Betriebssystem sorgt für die Adressraumabbildung
  - *memory management unit* (MMU) sorgt für die Adressumsetzung

Adressraumabbildung / MMU (1)



Relokation zur Laufzeit

- Abbildung gleichzeitig vorhandener Adressräume ist disjunktiv, alle Segmente werden überschneidungsfrei im Adressraum angeordnet.

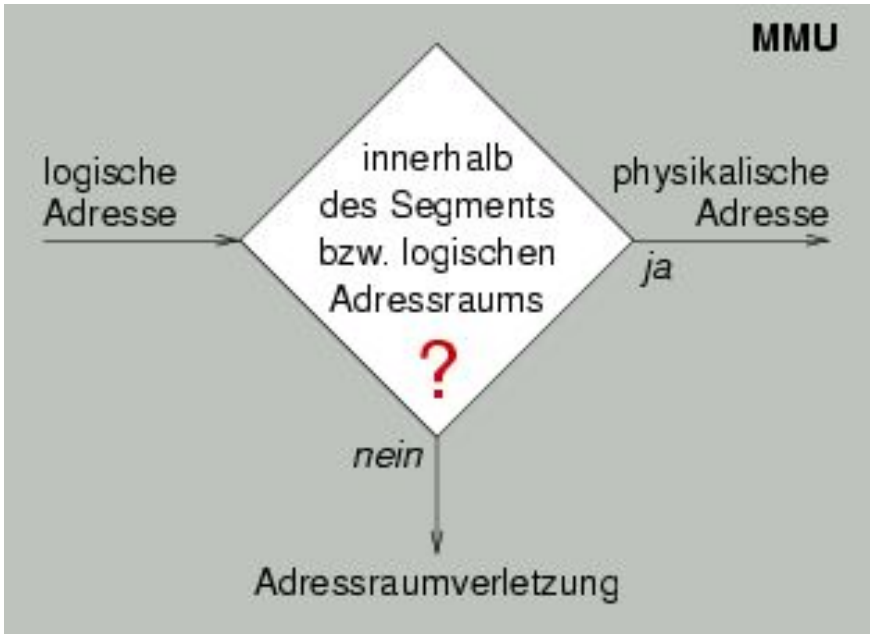


- zur **Ladezeit** lokalisiert BS den passenden Bereich im phys. Adressraum und programmiert MMU
- zur **Laufzeit** setzt die MMU jede logische Adresse in korrespondierende phys. Adresse um

- Ausbrechen aus einem logischen Segment ist zu verhindern.

Isolation der Adressräume

- Die MMU kapselt Adressräume und gewährleistet Zugriffsschutz.



- Adressraumverletzung führt zum Fehler
  - synchrone Programmunterbrechung

  ⇒ segmentation violation
  - Programmabbruch ist die Folge
- Oberste Instanz Betriebssystem

- Betriebssystem verantwortlich für die Integrität **aller** Adressräume
  - korrekte Funktion zwingend erforderlich

Virtueller Adressraum (1)

- die Anzahl gültiger phys. Adressen dimensioniert einen logischen Adressraum
  - in Realität entspricht dies der Größe des gesamten Arbeitsspeichers (RAM)
  - der Speicherbedarf eines Programms kann diese Größe leicht übersteigen
  - umso problematischer: gleichzeitig nebeneinander bestehende Programme
- Überbelegung des Arbeitsspeichers im Mehrprogrammbetrieb ist Normalität
  - Einsatz von Hintergrundspeicher (Platte) entschärft die Engpasssituation:
    - zur Zeit nicht benötigte Programmbereiche liegen auf der Platte
    - nur die zur „Arbeitsmenge“ gehörenden Bereiche sind im RAM
- ein virtueller Adressraum ist dimensioniert durch die Adressbreite der CPU

Adressbreite vs. Adressraumgröße

Adressbreite (Bit)	Adressraumgröße (2^N Bytes)	Dimension	
16	65.536	64	Kilo
20	1.048.576	1	Mega
32	4.294.967.296	4	Giga
48	281.474.976.710.656	256	Tera
64	18.446.744.073.709.600.000	16384	Peta

- Ein Rechner ist nur mit einem Bruchteil des von (künftigen) CPUs adressierbaren Speicher ausgestattet.

Virtueller Adressraum (2)

- die Adressabbildung (logischer Adressräume) ist um eine Stufe zu erweitern:

  Programmadresse $\Rightarrow$ logische Adresse

  logische Adresse $\Rightarrow$ virtuelle Adresse

  virtuelle Adresse $\Rightarrow$ physikalische Adresse

- Zugriffe über virtuelle Adressen können implizit Ein-/Ausgabe zur Folge haben
  - die MMU lässt nur gültige Zugriffe auf den Vordergrundspeicher zu
  - ggf. werden Zugriffe dann partiell interpretiert vom Betriebssystem ($\Rightarrow$ trap)
  - Ergebnis ist die Einlagerung des ”Operanden“ vom Hintergrundspeicher
  - dies kann zur Auslagerung anderer Vordergrundspeicherinhalte führen
- die Ein-/Auslagerung erfolgt typischerweise auf Seitenbasis ($\Rightarrow$demand paging)

Adressraumabbildung / MMU (2)