

1 Generelles

Charakterisieren Sie das BIOS und nennen Sie die wichtigsten Aufgaben.
Wofuer steht die Abkuerzung ueberhaupt?



Basisbetriebssystem

Ausgaben von einem 1. Betriebssystem, dem *BIOS*



Beispiel eines BIOS ROM Chips

EEPROM BIOS Chip (Fa. „winbond“)
verbaut auf dem Motherboard „ASRock Z77 Extreme 11“



Quelle: http://www.techpcmag.com/reviews/motherboards/Z77_Extreme11/1/MC

Aufgaben des BIOS (Basic Input Output System)

- POST = Power On Self Test → Suche nach und Zugriff auf wichtigsten Hardware-Komponenten (Tastatur, Arbeitsspeicher, Grafikkarte, Festplatten, CD-Laufwerk) und evtl. Test (Arbeitsspeicher)
- Sequentialle Überprüfung aller gefundener Laufwerke, ob Sie einen Bootsektor (MBR = Master Boot Record) enthalten
- Konfiguration der Hardware (BIOS-Menü), z.B. Prozessor- oder Arbeitsspeicher-Taktfrequenz

::Einführung

Richtig oder falsch: Ältere Betriebssysteme wie MS-DOS benutzten Basisfunktionen des BIOS auch nach dem Laden des eigentlichen BS.

Richtig. Neuere Varianten tun dies nicht mehr, sodass man gar den Chip des BIOS während des laufenden Betriebs entfernen kann.

Entwicklung des BIOS



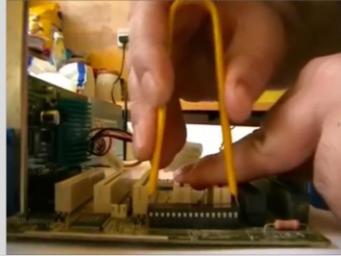
Früher vs. Heute

- Ältere Betriebssysteme wie MS-DOS benutzten Basisfunktionen des BIOS auch nach Laden des eigentlichen BS
- z.B. für Ausgabe von Text, Verarbeitung von Tastatureingaben, Zugriff auf Festplatte

"MS-DOS (PC DOS), which was the dominant PC operating system from the early 1980s until the mid 1990s, relied on BIOS services for disk, keyboard, and text display functions.

"MS Windows NT, Linux, and other protected mode operating systems in general ignore the abstraction layer provided by the BIOS and do not use it after loading, instead accessing the hardware components directly."

Entfernen des BIOS Chip nach Start des Betriebssystems



Quelle: <https://www.youtube.com/watch?v=3RzqjgjtvY>

::Einführung

Erläutern Sie den Begriff des **UEFI** wofür steht diese Abkürzung, woher kommt es und welche Vorteile bietet es im Gegensatz zum BIOS? Erläutern Sie ebenfalls kurz was man unter dem Begriff *Firmware* versteht. Welche Nachteile bietet das Booten mittels UEFI?

Unified Extensible Firmware Interface (UEFI)

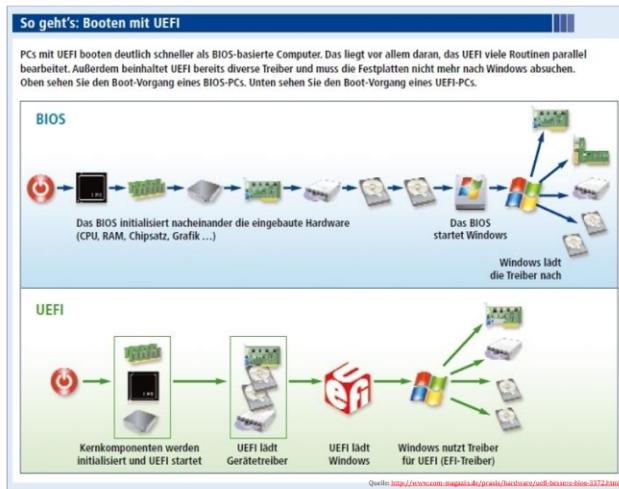


Woher kommt es?	Vorteile
Unified Extensible Firmware Interface Vereinheitlichte Erweiterbare Firmware*-Schnittstelle  <ul style="list-style-type: none"> ▪ als Ersatz/Nachfolger des BIOS gedacht ▪ geht zurück auf eine Initiative von Intel, um Beschränkungen des alten BIOS zu überwinden <p>*Firmware = Software, die in Geräte eingebettet ist</p>	<ul style="list-style-type: none"> ▪ Unterstützung von Laufwerken mit mehr als 2.2 TB → erlaubt also z.B. Booten auch von einer 3TB Platte ▪ Schnelleres Booten <ul style="list-style-type: none"> ▪ Paralleles Laden der Gerätetreiber ▪ UEFI Treiber werden vom BS wiederverwendet ▪ Netzwerkzugriff schon zur Bootzeit → z.B. Suche nach neuen UEFI-BIOS Updates <small>↳ Nachteil: Hackangriffe möglich</small> ▪ UEFI-Shell: Eigene Kommandozeile ▪ Bootmanager zur Auswahl des zu startenden BS ist schon integriert

::Einführung

Skizzieren Sie den **Bootvorgang** mittels UEFI und grenzen Sie diesen von dem mittels des klassischen BIOS ab.

Bootvorgang mit UEFI



::Einführung

Erläutern Sie die BS *Dienstleistung* der **Abstraktion** von konkreter Hardware.
Wieso ist so etwas sinnvoll und gibt es hierbei Ausnahmen?

2. Dienstleistung des Betriebssystem



Abstraktion von der konkreten Hardware

Ob Webcam, Maus, Drucker, Grafikkarte, etc. von Firma X oder Y ist aus Sicht der Programme egal



Somit: Gerätetreiber = Teil des Betriebssystems

Bsp: MS-DOS & fehlende Abstraktionsschicht für ...

... Sounderzeugung

Soundkarte bei MS-DOS = Teil der Hardware, die direkt von jedem Programm angesprochen werden musste



z.B. Soundblaster Karte 2.0 (1991)

::Einführung

Welche Dienstleistungen stellt ein Betriebssystem im Bereich der **Speichermedien** bereit?

3. Dienstleistung des Betriebssystem



Dateisysteme als Abstraktionsschicht vom Speichermedium

- Dateisysteme: FAT32, NTFS, ReFS (Resilient File System → Win8), ext2, ext3, ext4, ReiserFS, Reiser4, Tux3, etc.
- Egal ob Festplatte, CD, DVD, Blu-Ray, USB-Stick, SD-Card, Netzlaufwerk, etc. die Zugriffsweise ist immer gleich



Bildquellen:
USB-Stick: http://www.images-amazon.com/images/I/31V5gyv230UL._SL160_.JPG
CD: <https://en.wikipedia.org/wiki/CD-ROM>
SSD: https://en.wikipedia.org/wiki/Solid-state_drive

Dateisystem-Rechteverwaltung

Wer darf auf Datei X wie zugreifen?



Vorlesung Betriebssysteme

Prof. Dr. D. Säring © FH Wedel
2019

20

::Einführung

Welche Dienstleistungen stellt ein BS im Bereich der **Prozessverwaltung** bereit? Erläutern Sie ebenfalls diesen Begriff.



4. Dienstleistung des Betriebssystem

Prozessverwaltung

- Starten/Stoppen von Prozessen („Programme“)
- Hilfe bei Vermeidung möglicher Konflikte parallel arbeitenden Threads („Programmfäden“)
- Zuordnung: Rechenkern / Rechenzeit
- Zuteilung von Speicher und anderer beschränkter Betriebsmittel (z.B. gleichzeitiger Zugriff mehrerer Prozesse auf die gleiche Festplatte)
- Überwachen von Speicherzugriffsverletzungen

Beispiel: Speicherzugriffsverletzung

```
#include <iostream>
int main()
{
    int *a = new int[5];
    // a[0] = 2*0
    // a[1] = 2*1
    // a[2] = 2*2
    // a[3] = 2*3
    // a[4] = 2*4

    for (int i=0; i<=50; i++)
    {
        a[i] = 2*i;
        std::cout << a[i] << std::endl;
    }
    return 0;
}
```

Prozess- und Betriebsmittelverwaltung

Abstraktion von der Hardware
Benutzeroberfläche



Zwischenfazit: Betriebssystem als Dienstleister!

Vorlesung Betriebssysteme

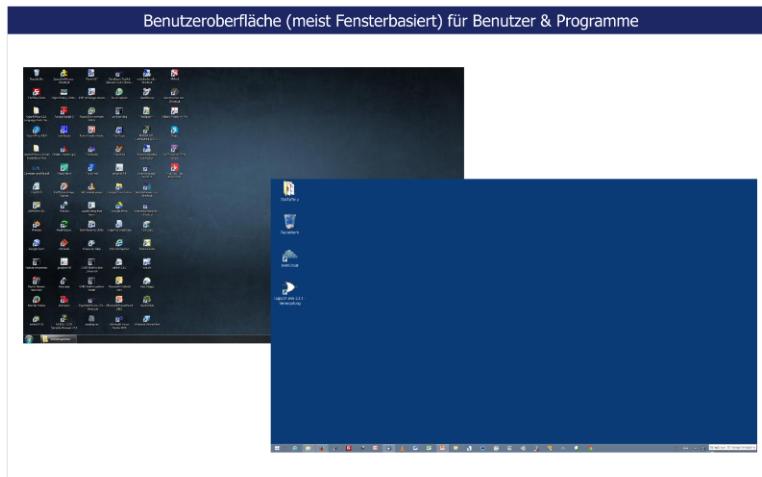
Prof. Dr. D. Säring © FH Wedel
2019

21

::Einführung

Nennen und beschreiben Sie alle wesentlichen Dienstleistungen die ein Betriebssystem zur Verfügung stellt.

1. Dienstleistung des Betriebssystem



2. Dienstleistung des Betriebssystem



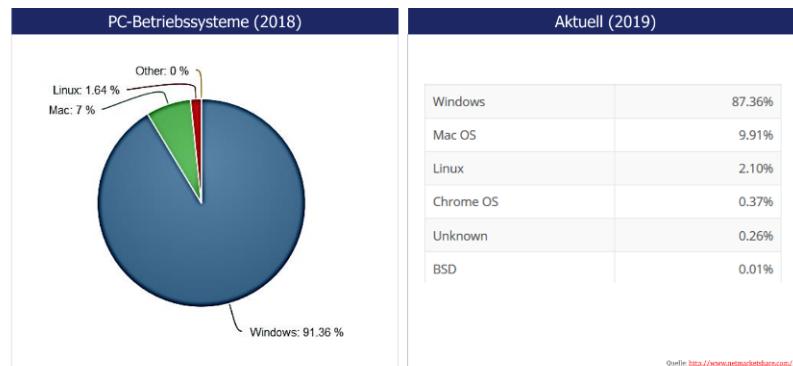
3. Dienstleistung des Betriebssystem



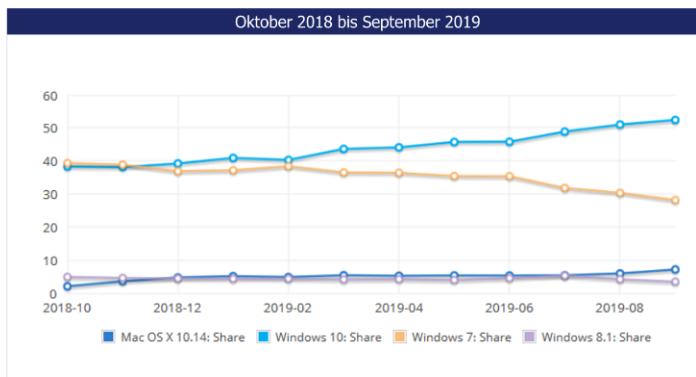
::Einführung marked

Beschreiben Sie grob die Marktanteile der PC-Betriebssysteme.

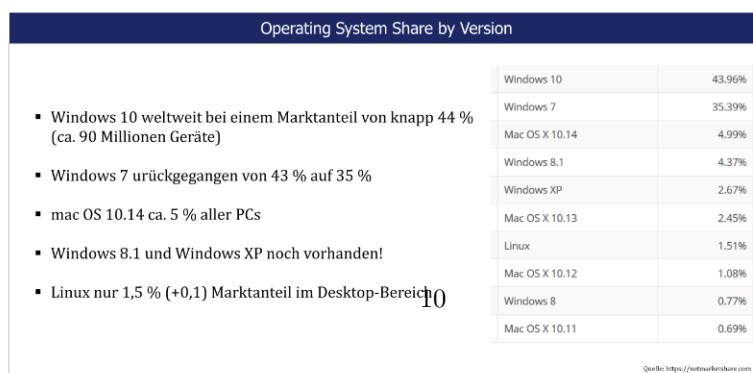
PC-Betriebssysteme



Marktanteile der führenden Betriebssysteme I



Marktanteile der führenden Betriebssysteme II



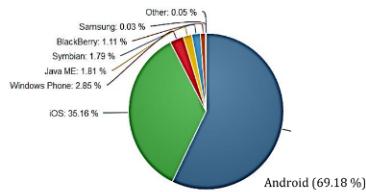
::Einführung

Beschreiben Sie grob die Marktanteile der **mobilen-Betriebssysteme**.

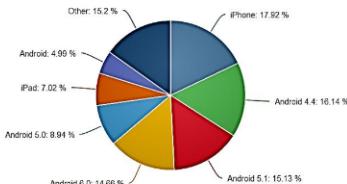
Mobile-Betriebssysteme



Mobile-Betriebssysteme (2018 / 2019)



Versionen



Vorlesung Betriebssysteme

Prof. Dr. D. Säring © FH Wedel
2019

26

Einführung und Überblick



Lernziele: Einführung und Überblick

- Wozu benötigt man ein Betriebssystem bzw. welche Aufgaben erfüllt ein Betriebssystem?
- Welche Arten von Betriebssystemen gibt es?
- Was ist der „Kernel“?



Vorlesung Betriebssysteme

Prof. Dr. D. Säring © FH Wedel
2019

22

::Einführung

Nennen Sie die Besonderheiten von **Server-Betriebssystemen**.

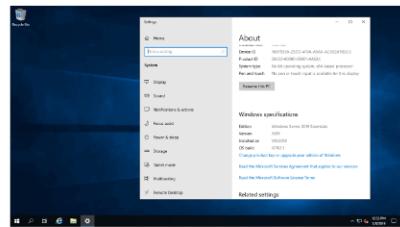
Server-Betriebssysteme



Bei Servern stehen andere Aufgaben im Vordergrund

- **Multi-User Fähigkeit:**
mehrere Benutzer sollen gleichzeitig den Rechner verwenden können
- Bereitstellung bestimmter **Dienste:**
 - **Fileserver**
 - **eMail-Server:** z.B. Microsofts „Exchange Server“ (Groupware) kann nur auf Microsoft Server BS installiert werden
 - **Rechteverwaltung beim Zugriff auf Ressourcen**
in einem Firmennetzwerk (Loginrechte auf PCs, Druckerzugriff) z.B. Microsofts „Active Directory“ in Windows Server BS
 - **Webserver** z.B. Microsofts „Internet Information Server (IIS)“

Beispiel: MS Server 2019



Quelle: <https://www.server-magazin.de>

::Einführung

Nennen Sie die Besonderheiten von **Echtzeitbetriebssystemen**. Und nennen Sie einige Beispiele.

Echtzeitbetriebssysteme (Real Time Operating Systems)

Wieder andere Rahmenbedingungen & Anforderungen

- oft leistungsschwächere Hardware als im Desktop PC / Server -Bereich (weniger Speicher, weniger Rechenleistung, ...)
- Echtzeit-Betriebssystem muss kompakter sein / Ressourcen-Bedarf muss sehr gering sein
- Garantieren von
 - stabilem Betrieb
 - sofortiger Abarbeitung von Prozessen
 - vorhersagbaren Laufzeiten von Prozessen stehen im Vordergrund
(Beispiel: Radarbasierter Notbremsassistent)
- geringere BS-Lizenzkosten für eingebettete Systeme

Fahrzeuge = Komplexe Rechensysteme



„Sie müssen das System neu starten, damit ihre Änderungen wirksam werden“ ... *

Quellen:
Auto: <http://www.autogesetz.de/medien/mediendatenbank/das-schweizer-autos-des-jahres-2011-dor-peugeot-308-tecnisch>
Fahrzeug: http://de.wikipedia.org/wiki/High-speed_train
Flugzeug: <http://schulmeister.de/2011/09/21/der-grindelwald-air-link-eine-alte-nostalgie-kennst-du-nach-einer-idee-von-michael-stappert/> (Hochschule Darmstadt)

Vorlesung Betriebssysteme

Prof. Dr. D. Säring © FH Wedel
2019

33

Echtzeitbetriebssysteme (II)

Beispiele von Echtzeitbetriebssystemen



RTLinux



Quellen:
RTLinux: <http://www.rtlinux.org/>
QNX: <http://devkit.qnx.com/qnx/qnx.html>
Windows Embedded: <http://www.microsoft.com/windows/embedded/developers.aspx>
Wie funktioniert RTLinux? → <http://www.opensourcefor.com/2010/12/getting-started-with-rtlinux/>

Vorlesung Betriebssysteme

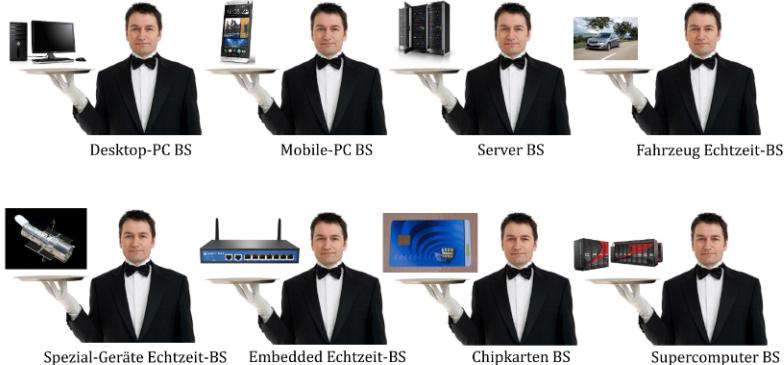
Prof. Dr. D. Säring © FH Wedel
2019

34

::Einführung

Fassen Sie die verschiedenen Arten von Betriebssystemen zusammen.

Vielfalt der Betriebssysteme



Vorlesung Betriebssysteme

Prof. Dr. D. Säring © FH Wedel

37

Weitere Betriebssysteme



Router-BS



Quelle: <http://wellimanded.wordpress.com/2011/12/29/you-can-fine-tune-a-computer-and-5-on-the-main-router/>

SmartCard (Prozessorkarten vs. Speicherchipkarten) BS

BasicCard, CombOS, CardOS, JCOP, MTCOS, MultOS, SECCOS, Sicrypt, STARCOS, TCOS, etc.



Zum Beispiel Speichern eines Private Keys auf der SmartCard und direkte Verschlüsselung von Daten auf der Karte mittels des Karten-Prozessors

Quelle: <http://www.kioskenet.info/definition/lexikon/Smartcard-smartcard.html>

Supercomputer BS

Cray Linux Environment (CLE), früher "UNICOS" skaliert auf bis zu mehr als 500.000 Prozessor-Kerne



Quelle: <http://www.cray.com/products/Computing/XE.aspx>

Verteilte Betriebssysteme

d.h. Virtualisierung von verteilten Ressourcen wie Speicher, Prozesse, etc.
z.B. LinuxPMI = Linux Process Migration Infrastructure



Quelle: <http://lxswx.org/trac/>

Vorlesung Betriebssysteme

Prof. Dr. D. Säring © FH Wedel
2019

36

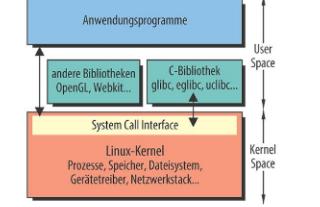
::Einführung

Was ist ein **Kernel** und woraus bestehen seine wesentlichen Aufgaben?

Zusammenfassung - Kernel: Software Kern des Betriebssystems, ist selbst ein Programm welches Dienste bereitstellt.

Aufgaben des BS-Kern (Kernel)



Typische BS-Kern Aufgaben	Weitere nicht BS-Kern Aufgaben
<ul style="list-style-type: none"> ▪ Abstraktion von der Hardware ▪ Speicherverwaltung ▪ Prozessverwaltung insbesondere IPC = Inter-Prozess-Communication ▪ Geräteverwaltung <i>• Wichtig: "GUI/Desktop nicht Teil d. kernels."</i> ▪ Dateisystemzugriff 	<ul style="list-style-type: none"> ▪ Grafische Benutzeroberfläche z.B. in Linux: Fenstermanager KDE, Gnome, Unity (Ubuntu) ▪ Shell / Kommandozeile ▪ BS-Tools

Aber: Liste der Aufgaben die vom BS-Kern erfüllt werden hängt vom BS-Kern Typ ab ...

::Einführung

Nennen Sie die Hauptarten von BS-Kernel.



Klassen von Kernel

BS-Kerne lassen sich anscheinend klassifizieren					...
Kernel name	Used in	Creator	Executable format (also see section below)	Type	
Agnix [3] ↗	?	?	?	?	
Amiga Exec	AmigaOS	Commodore International	HUNK	Exokernel (atypical) [2]	
Amiga Exec SG (2nd Generation)	AmigaOS 4	Hyperion Entertainment	ELF/HUNK	Exokernel (atypical)	
DragonFly BSD kernel	DragonFly BSD	Matt Dillon	ELF, others - platform dependent	hybrid	
FreeBSD kernel	FreeBSD, Debian GNU/kFreeBSD, Gentoo/kFreeBSD, Orbis OS	The FreeBSD Project	ELF, others - platform dependent	monolithic	
	GNU/Hurd (Arch)	GNU			

SunOS kernel	SunOS	Sun Microsystems	a.out	monolithic
Solaris kernel	Solaris, OpenSolaris, GNU/kOpenSolaris (Nexenta OS)	Sun Microsystems	ELF (32-bit only until Solaris 7 in 1999)	monolithic
Trix kernel	Trix, GNU/kTrix	Massachusetts Institute of Technology	a.out	monolithic
Windows NT kernel [12] ↗	Windows NT, 2000, XP, 2003, Vista, Windows 7, Windows 8	Microsoft	PE, others?	hybrid
XNU (Darwin kernel) [13] ↗	Mac OS X, iOS, OpenDarwin, PureDarwin GNU/Darwin	Apple Computer	Mach-O	hybrid
SPARTAN kernel [14] ↗	HeilenOS	Jakub Jermar	ELF	microkernel

Quelle: http://en.wikipedia.org/wikia/Komparison_of_operating_system_kernels

Vorlesung Betriebssysteme

Prof. Dr. D. Säring © FH Wedel

2019

43

::Einführung

Beschreiben Sie die *Philosophie*, *Nachteile* und die *Vorteile* eines **monolithischen Kernel BS**.

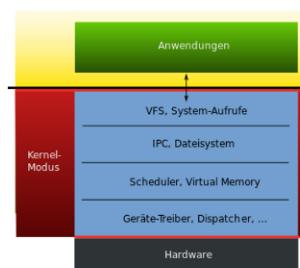


Die 3 Hauptarten von BS-Kernel



Monolithische Kernel BS

- **Philosophie:**
 - Sämtliche BS Komponenten in einem großen Programm -> riesiger BS-Kern
- **Nachteile:**
 - Wartbarkeit: schwierig weiter zu entwickeln / zu warten
 - Stabilität: ein einziger Fehler im BS-Kern kann zum Systemabsturz führen
- **Vorteile:**
 - Performanz: typischerweise sehr schnell
 - ↳ keine Einschränkungen wegen Trafik zwischen einz. Komponenten
- **Beispiele:** Unix und Linux



Vorlesung Betriebssysteme

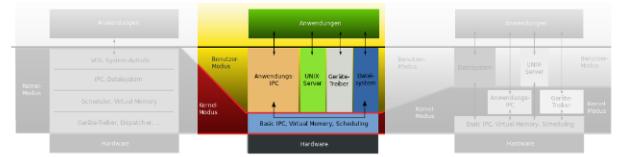
Prof. Dr. D. Säring © FH Wedel
2019

44

::Einführung

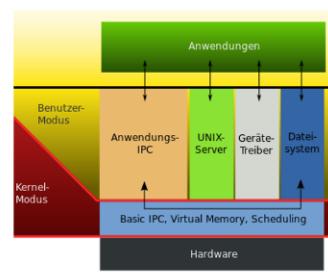
Beschreiben Sie die *Philosophie*, *Nachteile* und die *Vorteile* eines **Mikrokernels** BS .

Die 3 Hauptarten von BS-Kernel



Mikrokernell BS

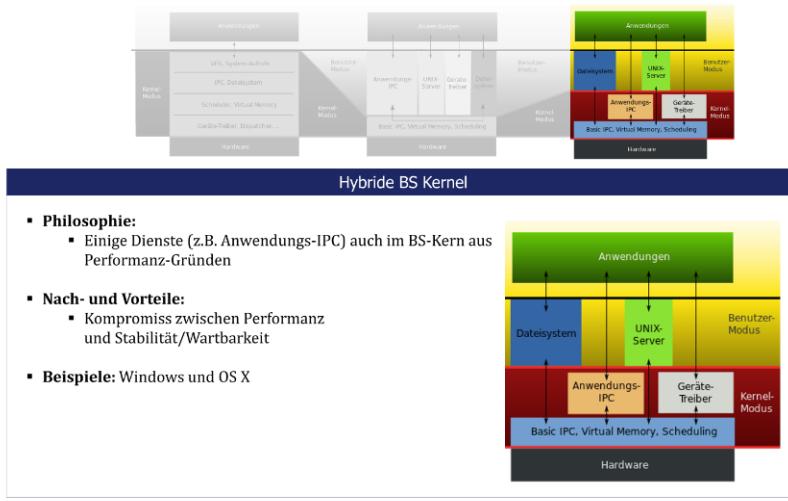
- **Philosophie:**
 - Möglichst wenige BS Komponenten im BS-Kern
-> kleiner BS-Kern
- **Nachteile:**
 - Performanz: Hoher Kommunikationsoverhead zwischen den Komponenten
- **Vorteile:**
 - Wartbarkeit: einfacher zu warten
 - Stabilität: Fehler in einzelnen Komponenten (z.B. Gerätetreiber) führen oft zum Absturz der Komponente, aber nicht zum Systemabsturz
- **Beispiele:** Minix, Mach und L4Linux (L4 Mikrokernell)



::Einführung

Beschreiben Sie die *Philosophie*, *Nachteile* und die *Vorteile* eines **hybridenBS Kernels**.

Die 3 Hauptarten von BS-Kernel



::Einführung

Fassen Sie noch einmal die verschiedenen Klassen von Kernels zusammen und grenzen Sie diese voneinander ab.

Klassen von Kernel

BS-Kerne lassen sich anscheinend klassifizieren					...
Kernel name	Used in	Creator	Executable format (also see section below)	Type	
Agnix [3] ↗	?	?	?	?	
Amiga Exec	AmigaOS	Commodore International	HUNK	Exokernel (atypical) [2]	
Amiga Exec SG (2nd Generation)	AmigaOS 4	Hyperion Entertainment	ELF/HUNK	Exokernel (atypical)	
DragonFly BSD kernel	DragonFly BSD	Matt Dillon	ELF, others - platform dependent	hybrid	
FreeBSD kernel	FreeBSD, Debian GNU/kFreeBSD, GentooFreeBSD, Orbis OS	The FreeBSD Project	ELF, others - platform dependent	monolithic	
GNU/Hurd (Arch)	GNU				

Quelle: http://en.wikipedia.org/wikia/Comparisons_of_operating_system_kernels

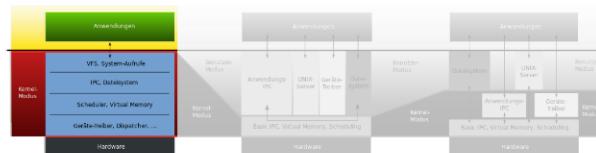
Vorlesung Betriebssysteme

Prof. Dr. D. Säring © FH Wedel 2019

43

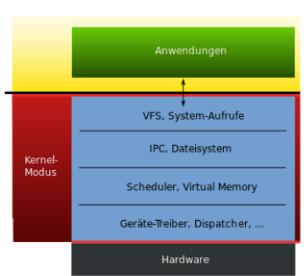


Die 3 Hauptarten von BS-Kernel



Monolithische Kernel BS

- Philosophie:**
 - Sämtliche BS Komponenten in einem großen Programm -> riesiger BS-Kern
- Nachteile:**
 - Wartbarkeit: schwierig weiter zu entwickeln / zu warten
 - Stabilität: ein einziger Fehler im BS-Kern kann zum Systemabsturz führen
- Vorteile:**
 - Performanz: typischerweise sehr schnell
 - ↳ keine Einbußen wegen Traffic zwischen einer Komponenten
- Beispiele:** Unix und Linux



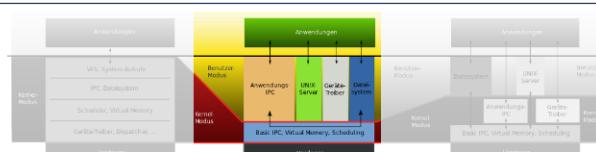
Vorlesung Betriebssysteme

Prof. Dr. D. Säring © FH Wedel 2019

44

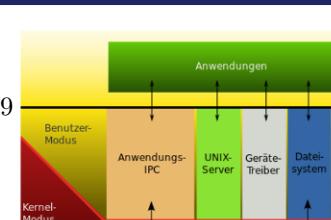


Die 3 Hauptarten von BS-Kernel



Mikrokern BS

- Philosophie:**
 - Möglichst wenige BS Komponenten im BS-Kern -> kleiner BS-Kern
- Nachteile:**
 - Performanz: Hoher Kommunikationsoverhead zwischen den Komponenten
- Vorteile:**
 - Wartbarkeit: einfacher zu warten
 - Stabilität: Fehler in einzelnen Komponenten



19

Kernell-Modus

::Einführung

Nennen Sie die wesentlichen Vorteile von **UEFI** gegenüber **BIOS** !
IDENTIFIER*i*

Wiederholung (I)



Lernkontrolle

- Nennen Sie die wesentlichen Vorteile von UEFI gegenüber BIOS
 - Unterstützung von Laufwerken mit mehr als 2.2 TB
 - Schnelleres Booten
 - Paralleles Laden der Gerätetreiber
 - UEFI Treiber werden vom BS wiederverwendet
 - Netzwerkzugriff schon zur Bootzeit
 - UEFI-Shell: Eigene Kommandozeile
 - Bootmanager zur Auswahl des zu startenden BS ist schon integriert
- Nennen Sie die wesentlichen „Dienstleistungen“ eines Betriebssystems!
 - Prozess- und Betriebsmittelverwaltung
 - Abstraktion der Hardware
 - Bereitstellung einer Benutzeroberfläche
 - Umsetzung von Dateisystemen



::Einführung

Betriebssysteme::Lernkontrolle

Nennen Sie die wesentlichen Dienstleistungen eines Betriebssystems

Wiederholung (I)



mitglieder: <http://www.ukw.edu.onlines.de>

Lernkontrolle

- Nennen Sie die wesentlichen Vorteile von UEFI gegenüber BIOS!
 - Unterstützung von Laufwerken mit mehr als 2.2 TB
 - Schnelleres Booten
 - Paralleles Laden der Gerätetreiber
 - UEFI Treiber werden vom BS wiederverwendet
 - Netzwerkzugriff schon zur Bootzeit
 - UEFI-Shell: Eigene Kommandozeile
 - Bootmanager zur Auswahl des zu startenden BS ist schon integriert
- Nennen Sie die wesentlichen „Dienstleistungen“ eines Betriebssystems!
 - Prozess- und Betriebsmittelverwaltung
 - Abstraktion der Hardware
 - Bereitstellung einer Benutzeroberfläche
 - Umsetzung von Dateisystemen



::Einführung Betriebssysteme::Lernkontrolle

Welche der Kernel Aufgaben gehört **NICHT** zu den Aufgaben eines BS-Kerns?

Wiederholung (II)



mitglieder: <http://www.ukw.edu.onlines.de>

Lernkontrolle

- Welche der folgenden Aufgaben gehört **NICHT** zu den Aufgaben eines BS-Kerns?
 - Abstraktion von der Hardware
 - Dateisystemzugriff
 - Grafische Benutzeroberfläche
 - Geräteverwaltung
- Beschreiben Sie die drei Hauptarten von BS-Kernel und gehen Sie dabei auf die Vor- und Nachteile ein!
 - Monolithische Kernel BS
 - (+) Performanz
 - (-) Wartbarkeit und Stabilität
 - Mikrokernel BS
 - (+) Wartbarkeit und Stabilität
 - (-) Performanz
 - Hybride BS Kernel
 - Kompromiss zwischen Performanz und Stabilität/Wartbarkeit



::Einführung Betriebssysteme::Lernkontrolle marked

Beschreiben Sie die drei Hauptarten von BS-Kernel und gehen Sie dabei auf die Vor- und Nachteile ein!

Wiederholung (II)



Lernkontrolle

- Welche der folgenden Aufgaben gehört **NICHT** zu den Aufgaben eines BS-Kerns?
 - Abstraktion von der Hardware
 - Dateisystemzugriff
 - Grafische Benutzeroberfläche
 - Geräteverwaltung
- Beschreiben Sie die drei Hauptarten von BS-Kernel und gehen Sie dabei auf die Vor- und Nachteile ein!
 - Monolithische Kernel BS
 - (+) Performanz
 - (-) Wartbarkeit und Stabilität
 - Mikrokern BS
 - (+) Wartbarkeit und Stabilität
 - (-) Performanz
 - Hybride BS Kernel
 - Kompromiss zwischen Performanz und Stabilität/Wartbarkeit



::Einführung Betriebssysteme::Lernkontrolle

Definieren Sie den Begriff des **Prozesses**. Was versteht man in diesem Zusammenhang unter der der *Quasiparallelität*?
missing IDENTIFIER;



Prozessmodell

Aspekt der Laufzeit

- Prozesse wechseln sich mit der Nutzung der CPU ab
(Beispiel: Warteschlange)
- Geschwindigkeit des Programmablaufs ist daher nicht einheitlich und selten reproduzierbar
(Beispiel: Audioprozess mit zeitlicher Verzögerung)
 - Spezialfall: Echtzeitbetriebssysteme



Quelle: die-besten-100.de

Allgemein

- Prozesse sind eine Aktivität, das auf einem Programm basiert, Ein- und Ausgaben sowie einen Zustand besitzt
- Prozesse teilen sich die CPU, wobei die Nutzung durch eine Schedulingstrategie bestimmt wird
- *Anmerkung:* Ein Programm, das zweimal ausgeführt wird läuft in unterschiedlichen Prozessen ab

... mal schauen wie ein Prozessmodell umgesetzt wird!

Prozessmodell

Aspekt der Laufzeit

- Prozesse wechseln sich mit der Nutzung der CPU ab
(Beispiel: Warteschlange)
- Geschwindigkeit des Programmablaufs ist daher nicht einheitlich und selten reproduzierbar
(Beispiel: Audioprozess mit zeitlicher Verzögerung)
 - Spezialfall: Echtzeitbetriebssysteme



Quelle: die-besten-100.de

Allgemein

- Prozesse sind eine Aktivität, das auf einem Programm basiert, Ein- und Ausgaben sowie einen Zustand besitzt
- Prozesse teilen sich die CPU, wobei die Nutzung durch eine Schedulingstrategie bestimmt wird
- *Anmerkung:* Ein Programm, das zweimal ausgeführt wird läuft in unterschiedlichen Prozessen ab

... mal schauen wie ein Prozessmodell umgesetzt wird!

::ProzesseUndThreads

Was versteht man unter dem sog. **Prozessmodell**?
 Erläutern Sie in diesem Zusammenhang grob, was man unter einer *Schedulingstrategie* versteht.

Prozessmodell



Aspekt der Laufzeit
<ul style="list-style-type: none"> ▪ Prozesse wechseln sich mit der Nutzung der CPU ab (Beispiel: Warteschlange) ▪ Geschwindigkeit des Programmablaufs ist daher nicht einheitlich und selten reproduzierbar (Beispiel: Audioprozess mit zeitlicher Verzögerung) <ul style="list-style-type: none"> ▪ Spezialfall: Echtzeitbetriebssysteme
Allgemein
<ul style="list-style-type: none"> ▪ Prozesse sind eine Aktivität, das auf einem Programm basiert, Ein- und Ausgaben sowie einen Zustand besitzt ▪ Prozesse teilen sich die CPU, wobei die Nutzung durch eine Schedulingstrategie bestimmt wird ▪ <i>Anmerkung:</i> Ein Programm, das zweimal ausgeführt wird läuft in unterschiedlichen Prozessen ab



Quelle: die-besten-100.de

... mal schauen wie ein Prozessmodell umgesetzt wird!

Vorlesung Betriebssysteme

Prof. Dr. D. Säring © FH Wedel
2019

54

Prozessmodell



Aspekt der Laufzeit
<ul style="list-style-type: none"> ▪ Prozesse wechseln sich mit der Nutzung der CPU ab (Beispiel: Warteschlange) ▪ Geschwindigkeit des Programmablaufs ist daher nicht einheitlich und selten reproduzierbar (Beispiel: Audioprozess mit zeitlicher Verzögerung) <ul style="list-style-type: none"> ▪ Spezialfall: Echtzeitbetriebssysteme
Allgemein
<ul style="list-style-type: none"> ▪ Prozesse sind eine Aktivität, das auf einem Programm basiert, Ein- und Ausgaben sowie einen Zustand besitzt ▪ Prozesse teilen sich die CPU, wobei die Nutzung durch eine Schedulingstrategie bestimmt wird ▪ <i>Anmerkung:</i> Ein Programm, das zweimal ausgeführt wird läuft in unterschiedlichen Prozessen ab



Quelle: die-besten-100.de

... mal schauen wie ein Prozessmodell umgesetzt wird!

Vorlesung Betriebssysteme

Prof. Dr. D. Säring © FH Wedel
2019

54

::ProzesseUndThreads

Beschreiben Sie den Begriff der **Prozesstabellen**. Welche Daten werden hierbei festgehalten?



Prozesstabellen

Prozesstabellen	Prozesskontrollblock (PCB)
<p>speichert PCBs aller Prozesse</p> <p>Assozierte Daten</p> <p>Metadaten, die BS mit Prg assoziiert:</p> <ul style="list-style-type: none"> ▪ Prozessnummer (PID) ▪ Prozessbesitzer ▪ Priorität ▪ Prozessrechte ▪ Befehlszähler ▪ Scheduling-Informationen ▪ ... <p>Reservierte Ressourcen</p> <ul style="list-style-type: none"> ▪ Speicher <ul style="list-style-type: none"> ▪ Heap (Datenspeicher) ▪ „Call Stack“ (Liste der aufgerufenen Funktionen) ▪ File Handles ▪ CPU Register 	<p>Der Prozesskontrollblock (Prozesskontext oder Task Control Block) ist eine Datenstruktur, die vom BS verwendet wird, um die assoziierten Daten für einen Prozess zu speichern</p> <p>htop Kommando</p> <pre> Tasks: 122, 169 thr: 5 running CPU: 1.1% user, 0.4% nice, 0.4% system, 0.1% interrupt, 0.0% softirq Mem: 3.9G used, 2.0G free, 0.0M swap Swap: 0.0M total, 0.0M used, 0.0M free PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ Command 4907 dseiring 20 0 318M 34560 1004 S 0.3 0.1 0:00.01 /usr/bin/dseiring 4945 dseiring 20 0 30360 29728 21528 S 3.0 0.1 0:01.25 gnome-terminal 779 root 20 0 288M 6556 6564 S 1.2 0.0 0:00.05 /usr/lib/policykit-1/polkitd --no-daemon 3432 dseiring 20 0 274M 37328 28532 S 0.6 0.0 0:07.40 /usr/libexec/gdm-x-demon -daemonize 4518 dseiring 20 0 742M 37328 28532 S 0.6 0.0 0:07.40 /usr/libexec/gdm-x-demon -daemonize 469 messenger 20 0 40136 3374 2744 S 0.0 0.0 0:00.11 dbus-daemon -system --fork 776 root 20 0 333M 2336 1936 S 0.6 0.0 0:00.01 /usr/libexec/gdm-x-demon -no-daemon 3137 root 20 0 233M 6476 6396 S 0.6 0.0 0:00.17 /usr/lib/powerd/upowerd 4133 dseiring 20 0 1182M 79824 34956 S 0.0 0.0 0:21.49 compiz 4125 dseiring 20 0 1182M 79824 34956 S 0.0 0.0 0:21.49 /usr/lib/xorg/Xorg -n vncr 3829 dseiring 20 0 347M 10444 1004 S 0.0 1.0 0:00.19 /usr/bin/dbus-daemon -daemonize 3876 dseiring 20 0 477M 20976 19444 S 0.0 2.0 0:00.53 /usr/lib/dbus/dbus-ut-gtk3 3880 dseiring 20 0 139M 4768 4456 S 0.0 0.5 0:00.23 /usr/bin/vtrootd 2895 dseiring 20 0 139M 4768 4456 S 0.0 0.5 0:00.23 /usr/bin/vtrootd 770 root 20 0 348M 8484 7752 S 0.0 0.0 0:00.00 NetworkManager 3392 dseiring 20 0 122M 2812 2564 S 0.0 0.0 0:00.00 /usr/libexec/dbus-engine-simple 3887 dseiring 20 0 477M 24976 19444 S 0.0 2.5 0:00.18 /usr/lib/dbus/dbus-ut-gtk3 3352 dseiring 20 0 122M 6112 5904 S 0.0 0.0 0:00.01 thermald -no-daemon --ibus-enable 3353 dseiring 20 0 122M 6112 5904 S 0.0 0.0 0:00.01 thermald -no-daemon --ibus-enable 3326 root 20 0 122M 6112 5904 S 0.0 0.0 0:00.19 thermald -no-daemon --ibus-enable 3927 dseiring 20 0 13648 7040 6592 S 0.0 0.0 0:00.01 /usr/libexec/dbus-engine-simple 1 root 20 0 13648 7040 6592 S 0.0 0.0 0:00.01 /sbin/init 378 root 20 0 19480 1788 1788 S 0.0 0.2 0:00.78 upstart-udev-bridge - daemon 588 root 20 0 10380 2636 2636 S 0.0 0.3 0:00.03 /usr/libexec/udisksd 495 root 20 0 10380 2636 2636 S 0.0 0.3 0:00.03 /usr/libexec/bluetoothd F1 1111 F2 1111 F3 1111 F4 1111 F5 1111 F6 1111 F7 1111 F8 1111 F9 1111 F10 1111 </pre>

Vorlesung Betriebssysteme

Prof. Dr. D. Söring © FH Wedel
2019

::ProzesseUndThreads marked

Was versteht man unter einem *Prozesskontrollblock*?missing IDENTIFIER_i

Prozesstabelle



Prozesstabelle	Prozesskontrollblock (PCB)
speichert PCBs aller Prozesse	Der Prozesskontrollblock (Prozesskontext oder Task Control Block) ist eine Datenstruktur , die vom BS verwendet wird, um die assoziierten Daten für einen Prozess zu speichern
Assozierte Daten <ul style="list-style-type: none"> Metadaten, die BS mit Prg assoziiert: <ul style="list-style-type: none"> ▪ Prozessnummer (PID) ▪ Prozessbesitzer ▪ Priorität ▪ Prozessrechte ▪ Befehlszähler ▪ Scheduling-Informationen ▪ ... Reservierte Ressourcen <ul style="list-style-type: none"> ▪ Speicher <ul style="list-style-type: none"> ▪ Heap (Datenspeicher) ▪ „Call Stack“ (Liste der aufgerufenenFunktionen) ▪ File Handles ▪ CPU Register 	

Vorlesung Betriebssysteme

Prof. Dr. D. Säring © FH Wedel

55

Prozesskontrollblock (PCB) in Linux



Anfang der Datenstruktur: Zeile 1282 in sched.h	Ende der Datenstruktur: Zeile 1712 in sched.h
<pre>1280 } 1281 1282 struct task_struct { 1283 volatile long state; /* -1 unrunnable, 0 runnable, >0 stopped */ 1284 void *stack; 1285 atomic_t nr_threads; 1286 unsigned long flags; /* per process flags, defined below */ 1287 unsigned int ptrace; 1288 1289 #ifdef CONFIG_SMP 1290 struct list_head wake_wake_entry; 1291 int oncpu; 1292 atomic_long_t wake_nr_ticks; 1293 unsigned long wake_flip; /* wake_flip_decay_ts */ 1294 unsigned long wake_flip_decay_ts; 1295 1296 int wake_cpus; 1297 1298 int on_rq; 1299 1300 int prio, static_prio, normal_prio; 1301 unsigned int rtime; /* relative time */ 1302 const struct sched_class *sched_class; 1303 struct sched_entity se; 1304 struct list_head se_rq; 1305 #endif /* CONFIG_SMP */ 1306 #endif /* CONFIG_SCHED */ 1307 struct task_group *sched_task_group; 1308 1309 struct sched_dl绽位list dl;</pre>	<pre>1675 int curr_ret_stack; 1676 /* Return stack addresses for return function tracing */ 1677 struct Trace_ret_stack *ret_stack; 1678 /* time stamp for the scheduler */ 1679 unsigned long tracing_timeStamp; 1680 1681 /* Number of functions that haven't been traced 1682 * because of depth overrun. 1683 */ 1684 atomic_t trace_overrun; 1685 /* Pause for the tracing */ 1686 atomic_t trace_stop; /* tracing_graph_pause */ 1687 #endif 1688 #endif /* CONFIG_SCHED */ 1689 /* state flags for use by tracers */ 1690 #define _TRACE_INNER 1 1691 #define _TRACE_OUTER 2 1692 #define _TRACE_DEPTH 4 1693 #define _TRACE_RECURSION 8 1694 #define _TRACE_BLOCKING 16 1695 #define _TRACE_HZ 32 1696 #define _TRACE_SCHED 64 1697 struct seq_file *seq_file; 1698 struct seq_file *seq_file_memp;</pre>

Vorlesung Betriebssysteme

Prof. Dr. D. Säring © FH Wedel

56

::ProzesseUndThreads

Erläutern Sie den Begriff der **Prozesserzeugung**. Wann ist so etwas sinnvoll, welche Möglichkeiten der Erzeugung gibt es, welche Befehle werden verwendet?

Prozesserzeugung



Prozesserzeugung

- Erzeugung von Prozessen können verursacht werden durch:
 - Initialisierung des Systems
 - Systemaufruf zum Erzeugen eines Prozesses durch einen anderen Prozess
 - Benutzeranfrage, einen neuen Prozess zu erzeugen
 - Initiierung einer Stapelverarbeitung
- Prozesse werden unterschieden in **Vordergrund** und **Hintergrundprozesse** (u.a. Daemons)
- Prozesserzeugung ist immer dann hilfreich, wenn Arbeit auf mehrere in Verbindung stehende aber unabhängig voneinander interagierende Prozesse aufteilbar ist! (Beispiel: BigData Processing)
- Ein Prozess wird immer durch einen anderen Prozess mit Hilfe eines Systemaufrufs erzeugt (`fork / createProcess`)



... und wie beenden wir Prozesse?

::ProzesseUndThreads

Wie kann ein Prozess beendet werden? Welche unterschiedlichen Arten gibt es, erläutern Sie dies am Beispiel vom BS Linux.

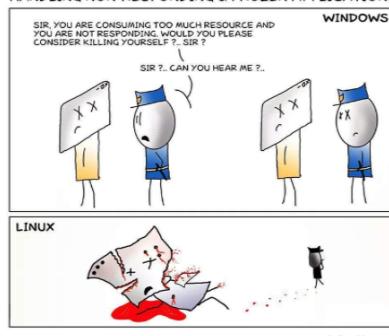
Prozessbeendigung



Prozessbeendigung

- Ein Prozess **terminiert** aufgrund einer der folgenden Bedingungen:
 - Normales Beenden (freiwillig)
 - Beenden durch Fehler (freiwillig)
 - Beenden durch schwerwiegenden Fehler (unfreiwillig)
 - Beenden durch anderen Prozess (unfreiwillig) [z.B. `kill -9` da selbst ein Programm]
- **Freiwilliges Beenden** z.B. nach vollständiger Erfüllung der Aufgabe erfolgt durch einen Systemaufruf (`exit / exitProcess`)
- **Unfreiwilliges Beenden** z.B. durch Programmierfehler die durch das System erkannt werden oder durch einen Befehl eines anderen Prozesses (`kill / terminateProcess`)

HANDLING NON-RESPONDING & FROZEN APPLICATIONS



Quelle: reddit.com

::ProzesseUndThreads