

# Betriebssystemtechnik

*Adressräume: Trennung, Zugriff, Schutz*

## III. Betriebssystemarchitektur

Wolfgang Schröder-Preikschat / Volkmar Sieh

3. Mai 2023



## Einleitung

- Architektur

- Aufbaustruktur

## Architekturformen

- Monolith

- Mikrokern

- Makrokern

- Exokern

## Dualität

- Betriebssystemstruktur

## Zusammenfassung



Ar·chi·tek'tur<sup>1</sup> < f.; -, -en > (lat. *architectura*)

1. Baukunst [als wissenschaftliche Disziplin]
2. a) [mehr oder weniger] kunstgerechter Aufbau und [künstlerische] Gestaltung von Bauwerken  
b) Konstruktion, Struktur des Aufbaus
3. Gesamtheit von Erzeugnissen der Baukunst (besonders eines Volkes, Bereichs, Stils, einer Zeit); Baustil

Prinzipien von Architektur: *venustas, firmitas, utilitas* [12]

- Anmut, Schönheit, Wunsch
- Solidität, Stabilität, Wesentlichkeit
- Zweckmäßigkeit, Nützlichkeit, Funktion, Gut

---

<sup>1</sup><http://www.duden.de/rechtschreibung/Architektur>

Ar·chi·tek'tur<sup>1</sup> < f.; –, –en > (lat. *architectura*)

1. Baukunst [als wissenschaftliche Disziplin]
2. a) [mehr oder weniger] kunstgerechter Aufbau und [künstlerische] Gestaltung von Bauwerken  
b) Konstruktion, Struktur des Aufbaus
3. Gesamtheit von Erzeugnissen der Baukunst (besonders eines Volkes, Bereichs, Stils, einer Zeit); Baustil

Prinzipien von Architektur: *venustas, firmitas, utilitas* [12]

- Anmut, Schönheit, Wunsch
  - Solidität, Stabilität, Wesentlichkeit
  - Zweckmäßigkeit, Nützlichkeit, Funktion, Gut
- ↪ die insbesondere auch für Betriebssysteme Gültigkeit besitzen...

---

<sup>1</sup><http://www.duden.de/rechtschreibung/Architektur>

Entlehnung in die Informatik erstmalig mit IBM System/360 [1]:

*The term **architecture** is used here to describe the attributes of a system as seen by a programmer, i.e.,*

- *the conceptual structure and functional behavior, as distinct from the organization of the data flow and controls,*
- *the logical design,*
- *and the physical implementation.*



Entlehnung in die Informatik erstmalig mit IBM System/360 [1]:

*The term **architecture** is used here to describe the attributes of a system as seen by a programmer, i.e.,*

- *the conceptual structure and functional behavior, as distinct from the organization of the data flow and controls,*
- *the logical design,*
- *and the physical implementation.*

Attribute, die **nichtfunktionale Eigenschaften** in den Fokus stellen,

- aus *nichtfunktionalen Anforderungen* (an ein Produkt) resultieren,
  - Zuverlässigkeit, Leistung, Effizienz
  - Betriebs-/Angriffssicherheit (*safety/security*)
  - Portierbarkeit (Plattformunabhängigkeit), Übertragbarkeit
  - Aussehen, Handhabung, Benutzbarkeit
- Auswirkungen auf die technische Umsetzung eines Systems haben.



- Rechnerarchitektur



- **Rechnerarchitektur** ist bestimmt durch ein *Operationsprinzip* für die Hardware und die *Struktur* ihres Aufbaus aus ihren Betriebsmitteln



- **Rechnerarchitektur** ist bestimmt durch ein *Operationsprinzip* für die Hardware und die *Struktur* ihres Aufbaus aus ihren Betriebsmitteln
  - **Operationsprinzip** definiert das funktionelle<sup>2</sup> Verhalten der Architektur durch Festlegung einer *Informationsstruktur* und einer *Kontrollstruktur*

---

<sup>2</sup>Impliziert die Spezifikation der Funktion, die es zu erfüllen gilt.



- **Rechnerarchitektur** ist bestimmt durch ein *Operationsprinzip* für die Hardware und die *Struktur* ihres Aufbaus aus ihren Betriebsmitteln
  - **Operationsprinzip** definiert das funktionelle<sup>2</sup> Verhalten der Architektur durch Festlegung einer *Informationsstruktur* und einer *Kontrollstruktur*
    - **Informationsstruktur** ist bestimmt durch die Typen und Repräsentationen der Informationskomponenten und der auf sie anwendbaren Operationen

---

<sup>2</sup>Impliziert die Spezifikation der Funktion, die es zu erfüllen gilt.



- **Rechnerarchitektur** ist bestimmt durch ein *Operationsprinzip* für die Hardware und die *Struktur* ihres Aufbaus aus ihren Betriebsmitteln
  - **Operationsprinzip** definiert das funktionelle<sup>2</sup> Verhalten der Architektur durch Festlegung einer *Informationsstruktur* und einer *Kontrollstruktur*
    - **Informationsstruktur** ist bestimmt durch die Typen und Repräsentationen der Informationskomponenten und der auf sie anwendbaren Operationen
    - **Kontrollstruktur** ist bestimmt durch die Spezifikation der Algorithmen für die Interpretation und Transformation der Informationskomponenten

---

<sup>2</sup>Impliziert die Spezifikation der Funktion, die es zu erfüllen gilt.



- **Rechnerarchitektur** ist bestimmt durch ein *Operationsprinzip* für die Hardware und die *Struktur* ihres Aufbaus aus ihren Betriebsmitteln
  - **Operationsprinzip** definiert das funktionelle<sup>2</sup> Verhalten der Architektur durch Festlegung einer *Informationsstruktur* und einer *Kontrollstruktur*
    - **Informationsstruktur** ist bestimmt durch die Typen und Repräsentationen der Informationskomponenten und der auf sie anwendbaren Operationen
    - **Kontrollstruktur** ist bestimmt durch die Spezifikation der Algorithmen für die Interpretation und Transformation der Informationskomponenten
  - **Struktur** ist gegeben durch Art/Anzahl der (Hardware-) Betriebsmittel und die sie verbindenden Kommunikationseinrichtungen
    - sowie die dafür definierten Kommunikations- und Kooperationsregeln

---

<sup>2</sup>Impliziert die Spezifikation der Funktion, die es zu erfüllen gilt.



- **Rechnerarchitektur** ist bestimmt durch ein *Operationsprinzip* für die Hardware und die *Struktur* ihres Aufbaus aus ihren Betriebsmitteln
  - **Operationsprinzip** definiert das funktionelle<sup>2</sup> Verhalten der Architektur durch Festlegung einer *Informationsstruktur* und einer *Kontrollstruktur*
    - **Informationsstruktur** ist bestimmt durch die Typen und Repräsentationen der Informationskomponenten und der auf sie anwendbaren Operationen
    - **Kontrollstruktur** ist bestimmt durch die Spezifikation der Algorithmen für die Interpretation und Transformation der Informationskomponenten
  - **Struktur** ist gegeben durch Art/Anzahl der (Hardware-) Betriebsmittel und die sie verbindenden Kommunikationseinrichtungen
    - sowie die dafür definierten Kommunikations- und Kooperationsregeln
- **Betriebssystemarchitektur** versteht sich ähnlich und ist insbesondere nicht losgelöst von Rechnerarchitektur zu sehen
  - es fehlt jedoch eine spezifische Definition für den Betriebssystembereich
  - der Begriff „Architektur“ wurde/wird hier unreflektiert übernommen

---

<sup>2</sup>Impliziert die Spezifikation der Funktion, die es zu erfüllen gilt.



## Definition (in Anlehnung an [5])

*Eine Betriebssystemarchitektur ist bestimmt durch ein Operationsprinzip für die Systemsoftware und die Struktur ihres Aufbaus aus den einzelnen Systemprogrammen.*



## Definition (in Anlehnung an [5])

*Eine Betriebssystemarchitektur ist bestimmt durch ein Operationsprinzip für die Systemsoftware und die Struktur ihres Aufbaus aus den einzelnen Systemprogrammen.*

- das Operationsprinzip manifestiert sich in der Rechnerbetriebsart



## Definition (in Anlehnung an [5])

*Eine Betriebssystemarchitektur ist bestimmt durch ein Operationsprinzip für die Systemsoftware und die Struktur ihres Aufbaus aus den einzelnen Systemprogrammen.*

- das Operationsprinzip manifestiert sich in der Rechnerbetriebsart
  - für die Informations- und Kontrollstruktur verschieden ausgelegt sind
    - Universal- vs. Spezialbetrieb, Zeitmultiplex- vs. Echtzeitbetrieb
    - Stapel- vs. Dialogbetrieb, Ein- vs. Mehrprogramm- vs. Mehrzugangsbetrieb



## Definition (in Anlehnung an [5])

*Eine Betriebssystemarchitektur ist bestimmt durch ein Operationsprinzip für die Systemsoftware und die Struktur ihres Aufbaus aus den einzelnen Systemprogrammen.*

- das Operationsprinzip manifestiert sich in der Rechnerbetriebsart
  - für die Informations- und Kontrollstruktur verschieden ausgelegt sind
    - Universal- vs. Spezialbetrieb, Zeitmultiplex- vs. Echtzeitbetrieb
    - Stapel- vs. Dialogbetrieb, Ein- vs. Mehrprogramm- vs. Mehrzugangsbetrieb
  - die Betriebsarten „benutzen“ [11] spezifische Module
    - die abstrakte Datentypen implementieren und verwenden
    - die in einen logisch-hierarchisch Zusammenhang angeordnet sind



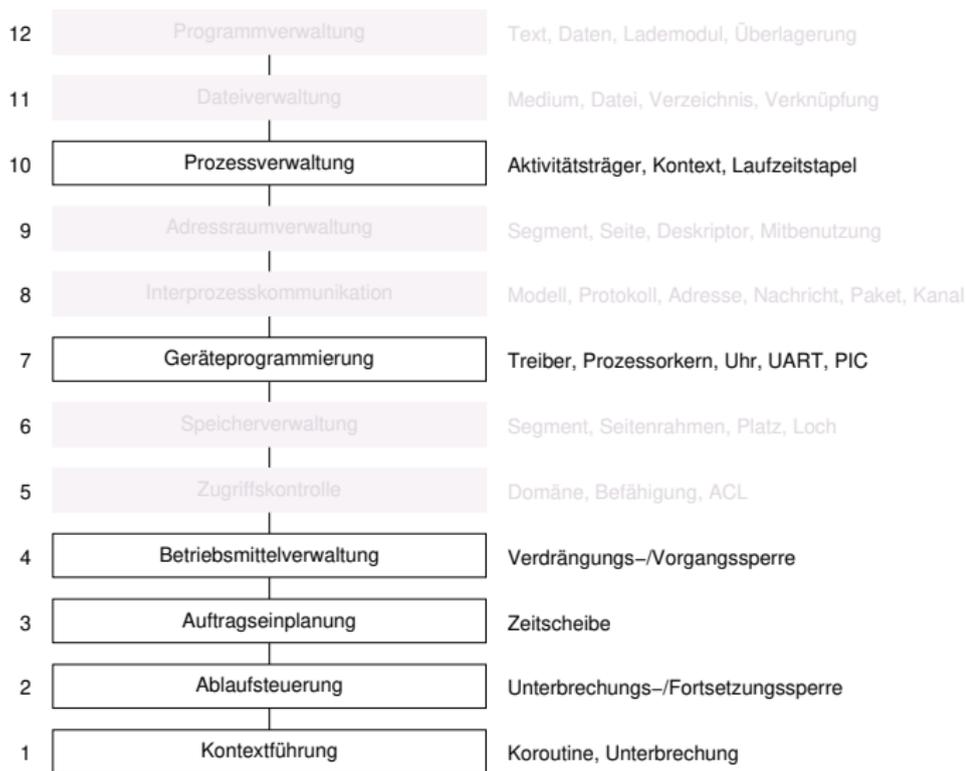
## Definition (in Anlehnung an [5])

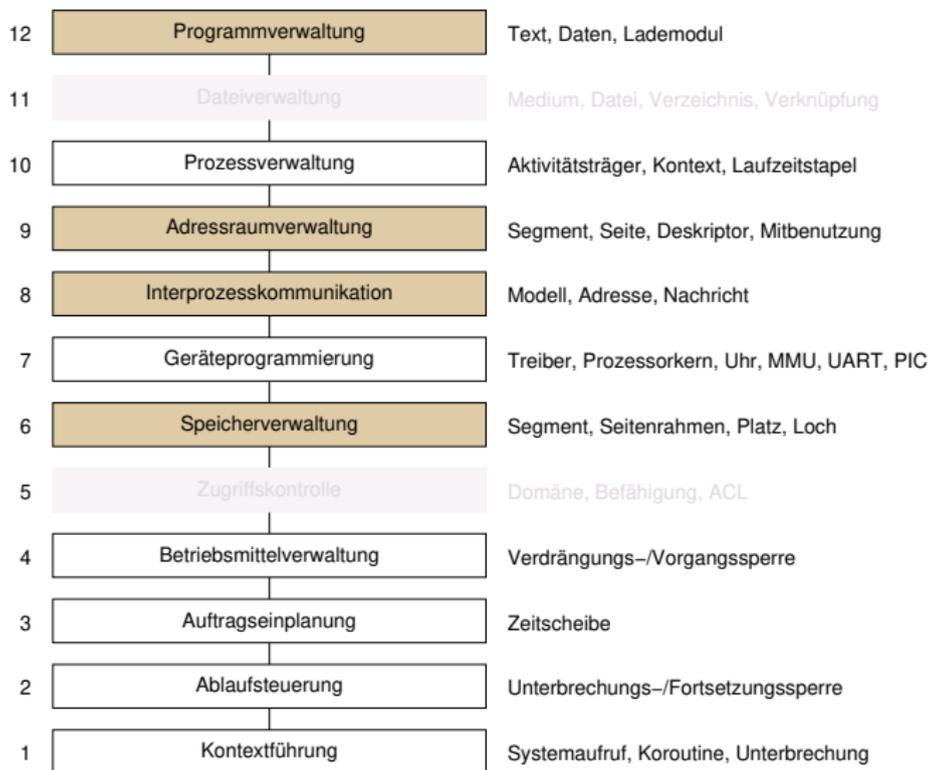
*Eine Betriebssystemarchitektur ist bestimmt durch ein Operationsprinzip für die Systemsoftware und die Struktur ihres Aufbaus aus den einzelnen Systemprogrammen.*

- das Operationsprinzip manifestiert sich in der Rechnerbetriebsart
  - für die Informations- und Kontrollstruktur verschieden ausgelegt sind
    - Universal- vs. Spezialbetrieb, Zeitmultiplex- vs. Echtzeitbetrieb
    - Stapel- vs. Dialogbetrieb, Ein- vs. Mehrprogramm- vs. Mehrzugangsbetrieb
  - die Betriebsarten „benutzen“ [11] spezifische Module
    - die abstrakte Datentypen implementieren und verwenden
    - die in einen logisch-hierarchisch Zusammenhang angeordnet sind
- wesentliche Struktur gebende Betriebsmittel dabei sind:
  - Art realer/virtueller Prozessor, gemeinsamer/verteilter Speicher
  - Anzahl ein-, mehr-, vielkerniger Uni-/Multiprozessor

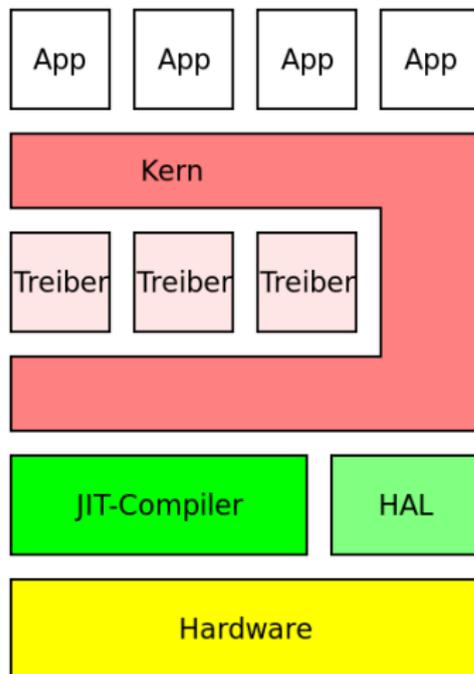








# Beispiel: Struktur JITTY



Einleitung

Architektur

Aufbaustruktur

Architekturformen

Monolith

Mikrokern

Makrokern

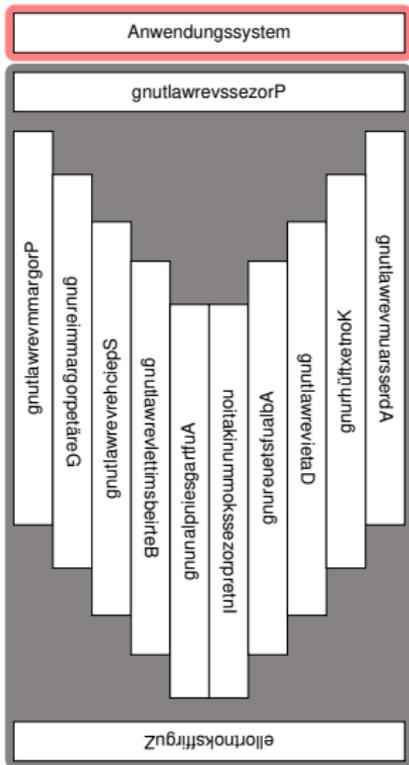
Exokern

Dualität

Betriebssystemstruktur

Zusammenfassung



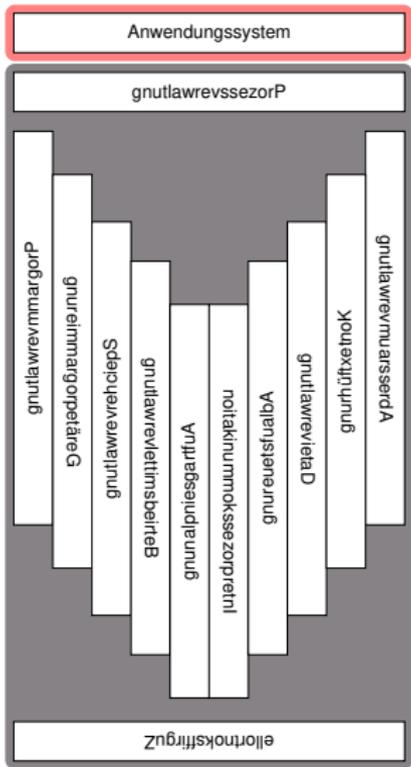


Mo·no·lith (grch. *monólithos*)

1. großer Steinblock
2. Stein aus einem Guss
3. monumentales Kunstwerk aus nur einem Stein







Mo·no·lith (grch. *monólithos*)

1. großer Steinblock
2. Stein aus einem Guss
3. monumentales Kunstwerk aus nur einem Stein

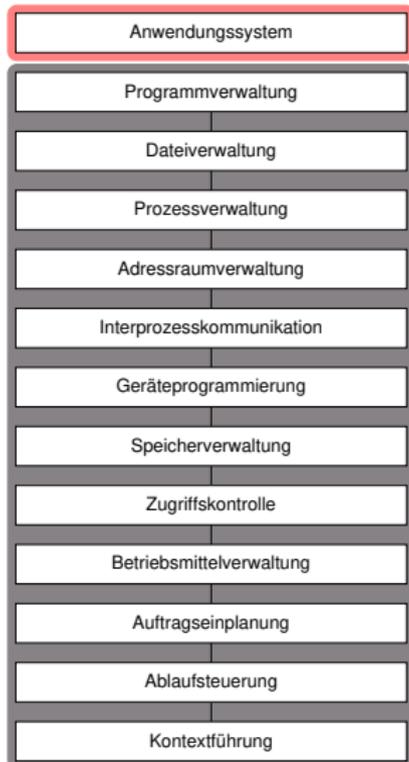
Architekturform des *Absolutismus*<sup>3</sup>

- ungetrennte Systemfunktionen
  - einheitlicher Adressraum
  - keine Fehlereingrenzung
  - derselbe Arbeitsmodus: privilegiert
- bestenfalls schwache Modularität
  - hohe räumliche Verflechtung
  - unaufwändige Interaktion im System
  - prozedur-/prozessorientierter Aufbau

<sup>3</sup>Eigene Machtvollkommenheit, ohne Mitwirkung ständischer Institutionen.



# Monolithisches schichtenstrukturiertes Betriebssystem



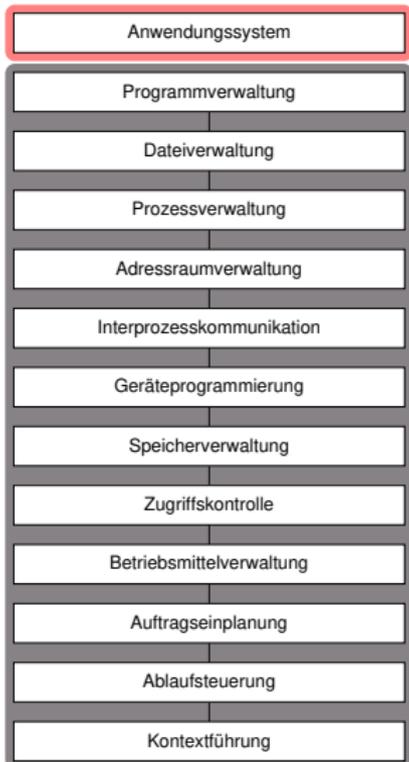
Architekturform des *Historismus*<sup>4</sup> [2]

- ungetrennte Systemfunktionen
  - einheitlicher Adressraum
  - keine Fehlereingrenzung
  - derselbe Arbeitsmodus: privilegiert

<sup>4</sup>Rückgriff auf und Nachahmung älterer Stilrichtungen.



# Monolithisches schichtenstrukturiertes Betriebssystem



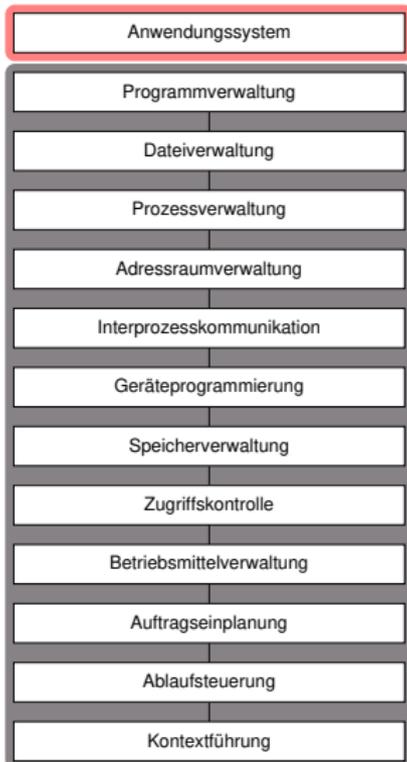
Architekturform des *Historismus*<sup>4</sup> [2]

- ungetrennte Systemfunktionen
  - einheitlicher Adressraum
  - keine Fehlereingrenzung
  - derselbe Arbeitsmodus: privilegiert
- starke logische Modularität
  - schwache räumliche Verflechtung
  - unaufwändige Interaktion im System
  - prozedur-/prozessorientierter Aufbau

<sup>4</sup>Rückgriff auf und Nachahmung älterer Stilrichtungen.



# Monolithisches schichtenstrukturiertes Betriebssystem

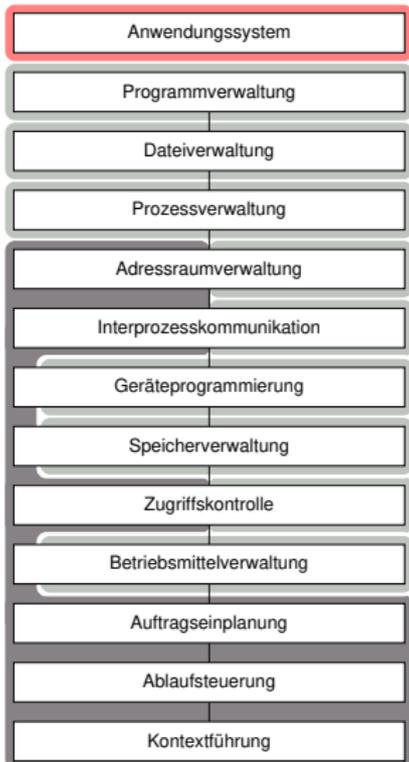


Architekturform des *Historismus*<sup>4</sup> [2]

- ungetrennte Systemfunktionen
    - einheitlicher Adressraum
    - keine Fehlereingrenzung
    - derselbe Arbeitsmodus: privilegiert
  - starke logische Modularität
    - schwache räumliche Verflechtung
    - unaufwändige Interaktion im System
    - prozedur-/prozessorientierter Aufbau
  - wohldefinierte innere Struktur
    - Programmhierarchie
    - „Benutzthierarchie“
    - funktionale Hierarchie
- ↪ vgl. auch [10]

<sup>4</sup>Rückgriff auf und Nachahmung älterer Stilrichtungen.

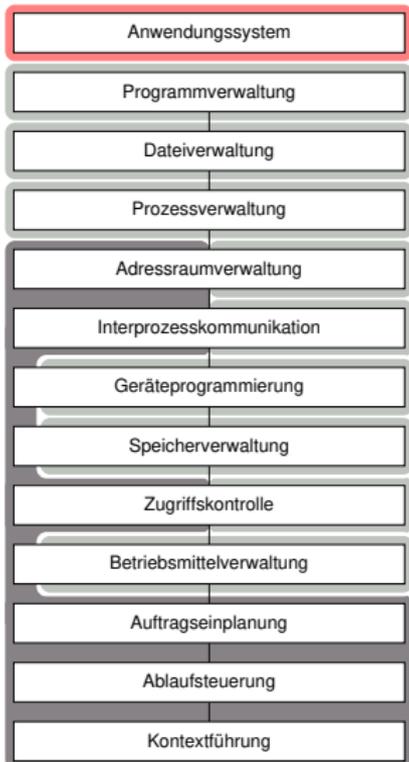




- Architekturform der *Moderne*<sup>5</sup> [8]
- getrennte Systemfunktionen
    - in separierten Adressräumen
    - starke Fehlereingrenzung
    - verschiedene Arbeitsmodi
      - privilegierter Kern (dunkel)
      - unprivilegierte Systemprozesse (hell)

<sup>5</sup>Umbruch in allen (Lebens-)Bereichen gegenüber der Tradition.

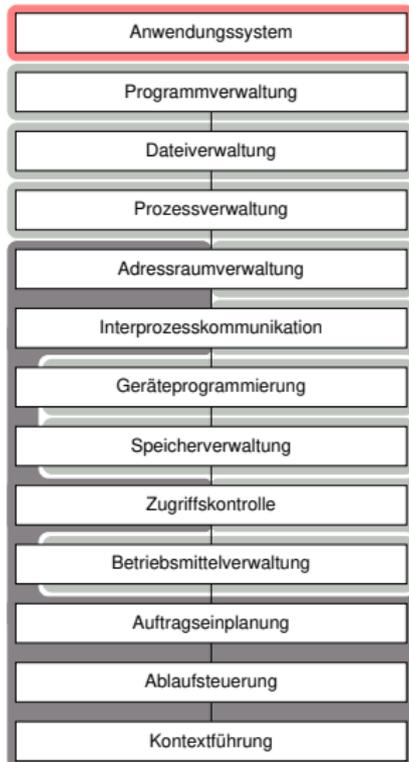




- Architekturform der *Moderne*<sup>5</sup> [8]
- getrennte Systemfunktionen
    - in separierten Adressräumen
    - starke Fehlereingrenzung
    - verschiedene Arbeitsmodi
      - privilegierter Kern (dunkel)
      - unprivilegierte Systemprozesse (hell)
  - starke reale und logische Modularität
    - schwache räumliche Verpflechtung
    - aufwändige Interaktion im System
- Prozesse IPC (hell)  
Kern Systemaufruf (dunkel)

<sup>5</sup>Umbruch in allen (Lebens-)Bereichen gegenüber der Tradition.





- Architekturform der *Moderne*<sup>5</sup> [8]
  - getrennte Systemfunktionen
    - in separierten Adressräumen
    - starke Fehlereingrenzung
    - verschiedene Arbeitsmodi
      - privilegierter Kern (dunkel)
      - unprivilegierte Systemprozesse (hell)
  - starke reale und logische Modularität
    - schwache räumliche Verpflechtung
    - aufwändige Interaktion im System
      - Prozesse IPC (hell)
      - Kern Systemaufruf (dunkel)
  - mittelkörnige Struktur
    - Granularität auf Adressraumbene
    - Kernadressraum als Monolith

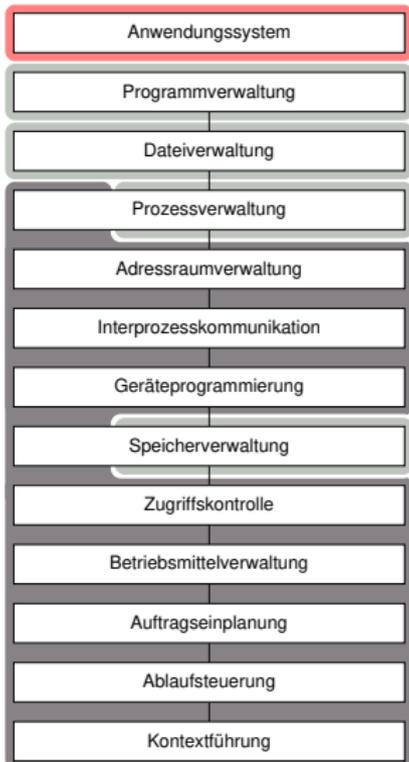
<sup>5</sup>Umbruch in allen (Lebens-)Bereichen gegenüber der Tradition.



- Arch.-form des *Utilitarismus*<sup>6</sup> [13]
- teils getrennte Systemfunktionen
    - in separierten Adressräumen
    - schwache Fehlereingrenzung
    - verschiedene Arbeitsmodi
      - privilegierter Hybridkern (dunkel)
      - unprivilegierte Systemprozesse (hell)

<sup>6</sup>Unterbegriff zu Konsequentialismus: Der Zweck heiligt die Mittel.

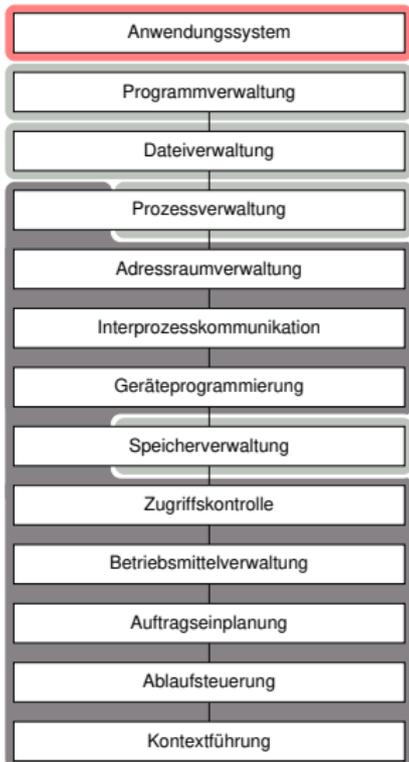




- Arch.-form des *Utilitarismus*<sup>6</sup> [13]
- teils getrennte Systemfunktionen
    - in separierten Adressräumen
    - schwache Fehlereingrenzung
    - verschiedene Arbeitsmodi
      - privilegierter Hybridkern (dunkel)
      - unprivilegierte Systemprozesse (hell)
  - eingeschränkte Modularität
    - bedingte räumliche Verpflechtung
    - unaufwändige Interaktion im System
      - Prozesse IPC (hell & dunkel)
      - Hybridkern Systemaufruf (dunkel)

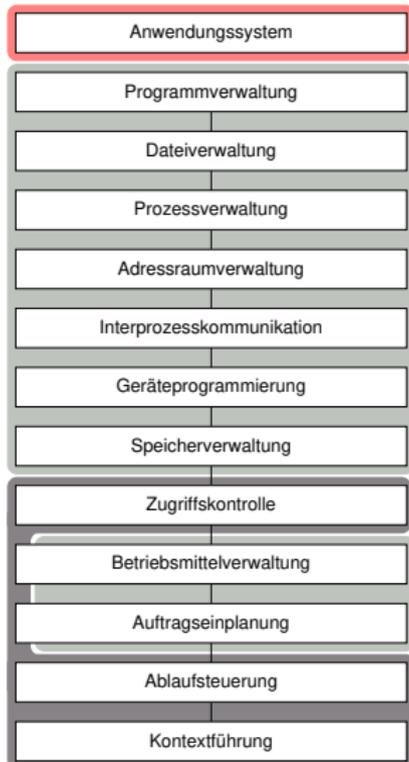
<sup>6</sup>Unterbegriff zu Konsequentialismus: Der Zweck heiligt die Mittel.





- Arch.-form des *Utilitarismus*<sup>6</sup> [13]
- teils getrennte Systemfunktionen
  - in separierten Adressräumen
  - schwache Fehlereingrenzung
  - verschiedene Arbeitsmodi
    - privilegierter Hybridkern (dunkel)
    - unprivilegierte Systemprozesse (hell)
- eingeschränkte Modularität
  - bedingte räumliche Verpflechtung
  - unaufwändige Interaktion im System
    - Prozesse IPC (hell & dunkel)
    - Hybridkern Systemaufruf (dunkel)
- grobkörnige Struktur
  - Granularität auf Adressraumbene
  - Hybridkernadressraum als Monolith

<sup>6</sup>Unterbegriff zu Konsequentialismus: Der Zweck heiligt die Mittel.

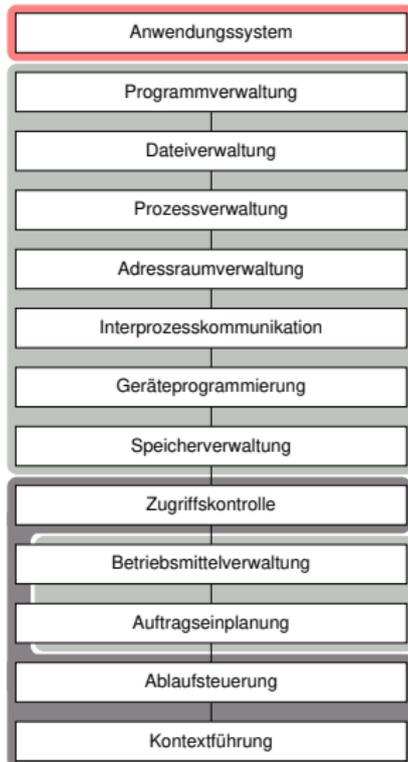


## Architekturform der *Avantgarde*<sup>7</sup> [3]

- Trennung von Belangen
  - Betriebsmittelverwaltung
    - durch Bibliotheksbetriebssysteme (hell)
  - Betriebsmittelschutz
    - durch den Exokern (dunkel)

<sup>7</sup>Fortschrittsorientiert, Radikalität gegenüber bestehenden Verhältnissen.



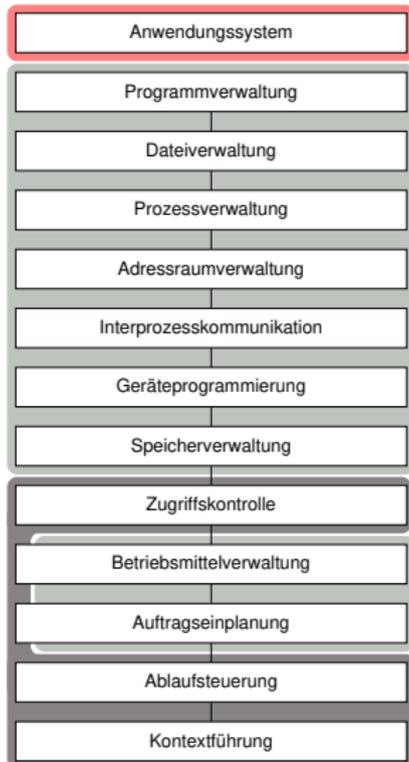


## Architekturform der *Avantgarde*<sup>7</sup> [3]

- Trennung von Belangen
  - Betriebsmittelverwaltung
    - durch Bibliotheksbetriebssysteme (hell)
  - Betriebsmittelschutz
    - durch den Exokern (dunkel)
- konsequentes Freilegen. . .
  - von Betriebsmittelbindungen
  - aller Betriebsmittelbelegungen
  - der realen Betriebsmittelnamen
  - des Widerrufs von Betriebsmitteln

<sup>7</sup>Fortschrittsorientiert, Radikalität gegenüber bestehenden Verhältnissen.





## Architekturform der *Avantgarde*<sup>7</sup> [3]

- Trennung von Belangen
  - Betriebsmittelverwaltung
    - durch Bibliotheksbetriebssysteme (hell)
  - Betriebsmittelschutz
    - durch den Exokern (dunkel)
- konsequentes Freilegen. . .
  - von Betriebsmittelbindungen
  - aller Betriebsmittelbelegungen
  - der realen Betriebsmittelnamen
  - des Widerrufs von Betriebsmitteln
- drei Grundtechniken:
  1. sichere Betriebsmittelbindung
  2. sichtbarer Betriebsmittelwideruff
  3. imperativer Betriebsmittelentzug

<sup>7</sup>Fortschrittsorientiert, Radikalität gegenüber bestehenden Verhältnissen.



Einleitung

Architektur

Aufbaustruktur

Architekturformen

Monolith

Mikrokern

Makrokern

Exokern

Dualität

Betriebssystemstruktur

Zusammenfassung



# Strukturanalogie zwischen den Architekturformen

---

*Many operating system designs can be placed into one of two very rough categories, depending upon how they implement and use the notions of process and synchronization.*



*Many operating system designs can be placed into one of two very rough categories, depending upon how they implement and use the notions of process and synchronization.*

- *One category, the “**Message-oriented System,**” is characterized by a relatively small, static number of processes with an explicit message system for communicating among them.*



*Many operating system designs can be placed into one of two very rough categories, depending upon how they implement and use the notions of process and synchronization.*

- *One category, the “**Message-oriented System,**” is characterized by a relatively small, static number of processes with an explicit message system for communicating among them.*
- *The other category, the “**Procedure-oriented System,**” is characterized by a large, rapidly changing number of small processes and a process synchronization mechanism based on shared data.*



# Strukturanalogie zwischen den Architekturformen

*Many operating system designs can be placed into one of two very rough categories, depending upon how they implement and use the notions of process and synchronization.*

- *One category, the “**Message-oriented System,**” is characterized by a relatively small, static number of processes with an explicit message system for communicating among them.*
- *The other category, the “**Procedure-oriented System,**” is characterized by a large, rapidly changing number of small processes and a process synchronization mechanism based on shared data.*

*These two categories are duals of each other and a system which is constructed according to one model has a direct counterpart in the other.*



# Strukturanalogie zwischen den Architekturformen

*Many operating system designs can be placed into one of two very rough categories, depending upon how they implement and use the notions of process and synchronization.*

- *One category, the “**Message-oriented System,**” is characterized by a relatively small, static number of processes with an explicit message system for communicating among them.*
- *The other category, the “**Procedure-oriented System,**” is characterized by a large, rapidly changing number of small processes and a process synchronization mechanism based on shared data.*

*These two categories are duals of each other and a system which is constructed according to one model has a direct counterpart in the other. The principal conclusion is that neither model is inherently preferable, and the main consideration for choosing between them is *the nature of the machine architecture* upon which the system is being built, *not the application* which the system will ultimately support. (Lauer/Needham, [7])*



# Strukturanalogie zwischen den Architekturformen

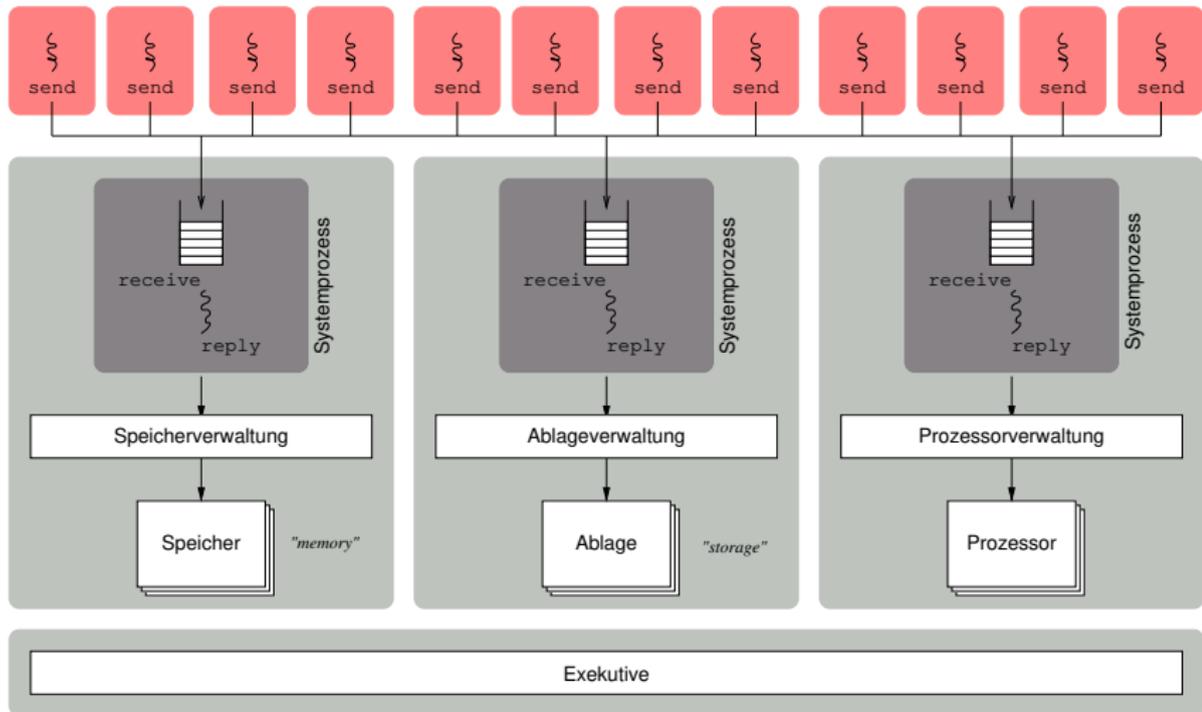
*Many operating system designs can be placed into one of two very rough categories, depending upon how they implement and use the notions of process and synchronization.*

- *One category, the “**Message-oriented System,**” is characterized by a relatively small, static number of processes with an explicit message system for communicating among them.*
- *The other category, the “**Procedure-oriented System,**” is characterized by a large, rapidly changing number of small processes and a process synchronization mechanism based on shared data.*

*These two categories are duals of each other and a system which is constructed according to one model has a direct counterpart in the other. The principal conclusion is that neither model is inherently preferable, and the main consideration for choosing between them is *the nature of the machine architecture* upon which the system is being built, *not the application* which the system will ultimately support. (Lauer/Needham, [7])*

↳ **beachte:** dieser Sicht wird aber auch widersprochen [4, S. 435] **!!!**





## Charakteristik:

- kleine und relativ statische Anzahl großer Prozesse  $\rightsquigarrow$  [4]: Teams
  - steht und fällt mit dem Prozessmodell und dessen Implementierung



## Charakteristik:

- kleine und relativ statische Anzahl großer Prozesse  $\rightsquigarrow$  [4]: Teams
  - steht und fällt mit dem Prozessmodell und dessen Implementierung
- expliziter Satz von Nachrichtenkanälen zwischen diesen Prozessen
  - typischerweise langfristige Bindung  $\rightsquigarrow$  [4]: kurzfristig, verbindungslos
  - Nachrichtenversenden (*message passing*) als Interaktionsmittel



## Charakteristik:

- kleine und relativ statische Anzahl großer Prozesse  $\rightsquigarrow$  [4]: Teams
  - steht und fällt mit dem Prozessmodell und dessen Implementierung
- expliziter Satz von Nachrichtenkanälen zwischen diesen Prozessen
  - typischerweise langfristige Bindung  $\rightsquigarrow$  [4]: kurzfristig, verbindungslos
  - Nachrichtenversenden (*message passing*) als Interaktionsmittel
- relativ begrenzter Umfang von direkt gemeinsam benutzten Daten
  - im Haupt- oder Arbeitsspeicher  $\rightsquigarrow$  [4]: außerhalb von Teams



## Charakteristik:

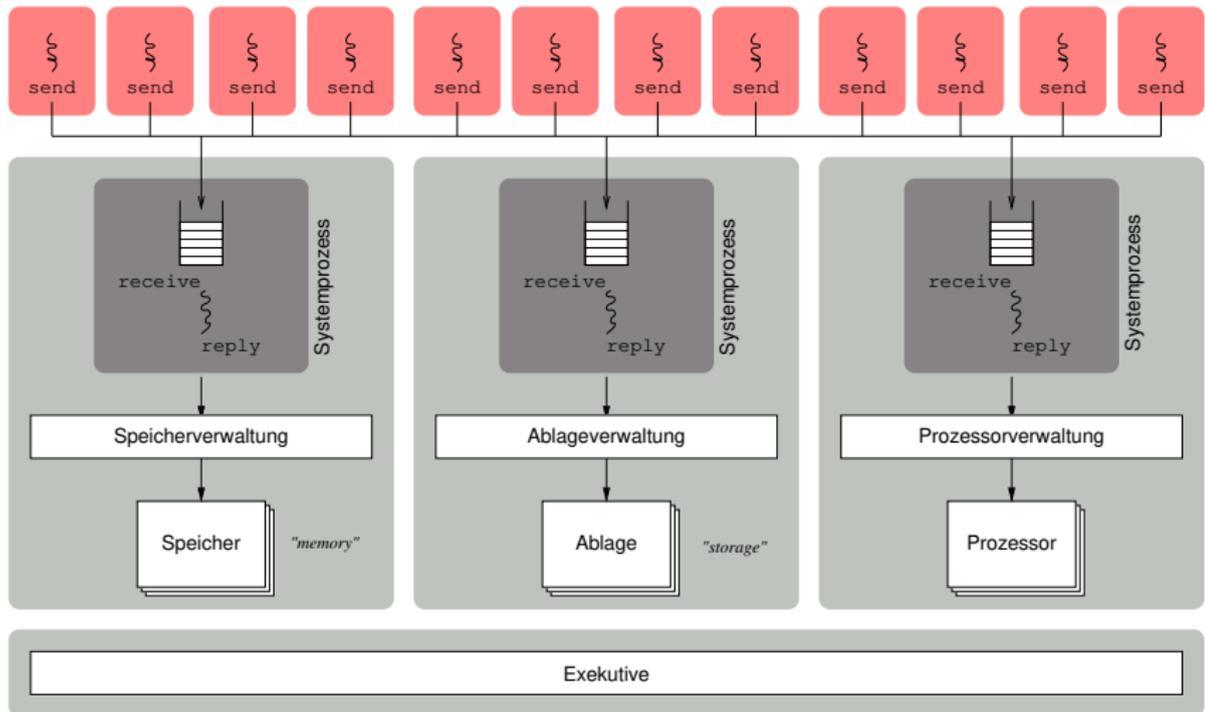
- kleine und relativ statische Anzahl großer Prozesse  $\rightsquigarrow$  [4]: Teams
  - steht und fällt mit dem Prozessmodell und dessen Implementierung
- expliziter Satz von Nachrichtenkanälen zwischen diesen Prozessen
  - typischerweise langfristige Bindung  $\rightsquigarrow$  [4]: kurzfristig, verbindungslos
  - Nachrichtenversenden (*message passing*) als Interaktionsmittel
- relativ begrenzter Umfang von direkt gemeinsam benutzten Daten
  - im Haupt- oder Arbeitsspeicher  $\rightsquigarrow$  [4]: außerhalb von Teams
- eine Identifizierung von Adressraum oder Kontext mit den Prozessen
  - jeder Prozess neigt dazu, in einem relativ statischen Kontext zu laufen
  - Adressräume stehen in Übereinstimmung mit Prozessen  $\rightsquigarrow$  [4]: Teams
  - Prozesse überschreiten Schutzgrenzen eher selten
    - wenn, dann nur, um kurz die Exekutive oder den Kern zu betreten



## Charakteristik:

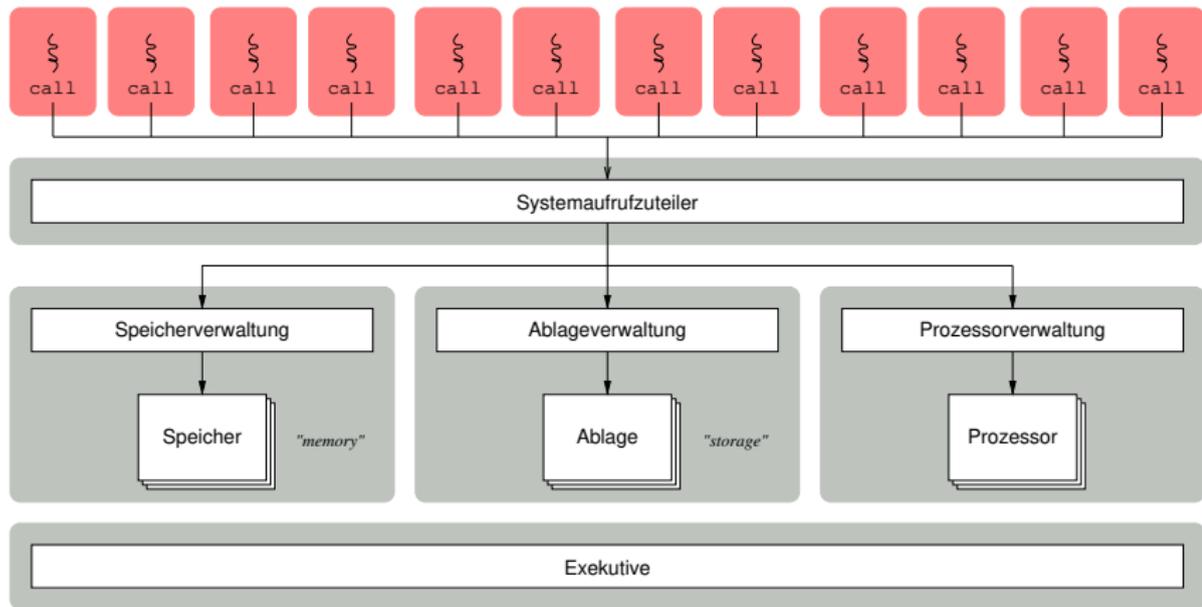
- kleine und relativ statische Anzahl großer Prozesse  $\leadsto$  [4]: Teams
    - steht und fällt mit dem Prozessmodell und dessen Implementierung
  - expliziter Satz von Nachrichtenkanälen zwischen diesen Prozessen
    - typischerweise langfristige Bindung  $\leadsto$  [4]: kurzfristig, verbindungslos
    - Nachrichtenversenden (*message passing*) als Interaktionsmittel
  - relativ begrenzter Umfang von direkt gemeinsam benutzten Daten
    - im Haupt- oder Arbeitsspeicher  $\leadsto$  [4]: außerhalb von Teams
  - eine Identifizierung von Adressraum oder Kontext mit den Prozessen
    - jeder Prozess neigt dazu, in einem relativ statischen Kontext zu laufen
    - Adressräume stehen in Übereinstimmung mit Prozessen  $\leadsto$  [4]: Teams
    - Prozesse überschreiten Schutzgrenzen eher selten
      - wenn, dann nur, um kurz die Exekutive oder den Kern zu betreten
- ↳ als prägend zeigt sich, Prozesse mit Systemressourcen zu assoziieren
- ↳ Anwendungsbedürfnisse erhält das System in Nachrichten kodiert





im-/explizite Koordinierung der Systemprozesse durch die Exekutive





## Charakteristik:

- große Anzahl sehr kleiner ((teil-)aufgabenspezifischer) Prozesse
  - die mittels Prozeduraufrufe das ganze System durchstreifen



## Charakteristik:

- große Anzahl sehr kleiner ((teil-)aufgabenspezifischer) Prozesse
  - die mittels Prozeduraufrufe das ganze System durchstreifen
- kurzfristige/rasche Erzeugung und Beseitigung der Prozesse
  - da keine expliziten Kommunikationskanäle auf-/abgebaut werden müssen



## Charakteristik:

- große Anzahl sehr kleiner ((teil-)aufgabenspezifischer) Prozesse
  - die mittels Prozeduraufrufe das ganze System durchstreifen
- kurzfristige/rasche Erzeugung und Beseitigung der Prozesse
  - da keine expliziten Kommunikationskanäle auf-/abgebaut werden müssen
- Kommunikation über direkt gemeinsam benutzte Daten
  - Verriegelung (*interlocking*) von Daten im Haupt-/Arbeitsspeicher
  - Zugriff und Schutz globaler Daten über prozedurale Schnittstellen



## Charakteristik:

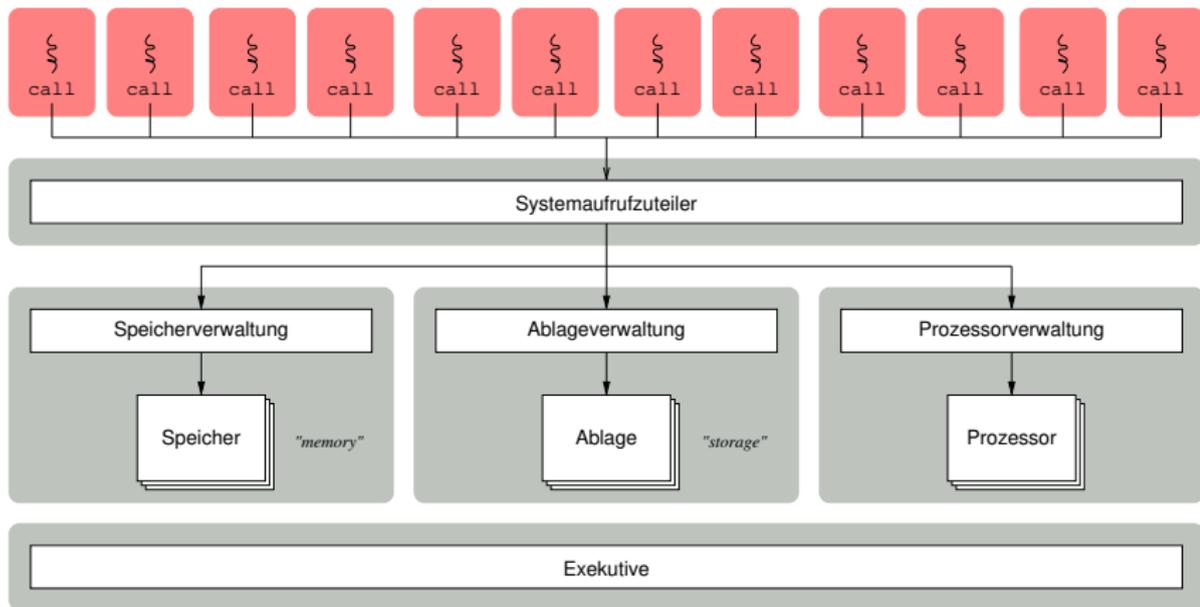
- große Anzahl sehr kleiner ((teil-)aufgabenspezifischer) Prozesse
  - die mittels Prozeduraufrufe das ganze System durchstreifen
- kurzfristige/rasche Erzeugung und Beseitigung der Prozesse
  - da keine expliziten Kommunikationskanäle auf-/abgebaut werden müssen
- Kommunikation über direkt gemeinsam benutzte Daten
  - Verriegelung (*interlocking*) von Daten im Haupt-/Arbeitsspeicher
  - Zugriff und Schutz globaler Daten über prozedurale Schnittstellen
- eine Identifizierung des Ausführungskontextes jeder Funktion
  - genauer: die sich gerade in Ausführung befindet



## Charakteristik:

- große Anzahl sehr kleiner ((teil-)aufgabenspezifischer) Prozesse
    - die mittels Prozeduraufrufe das ganze System durchstreifen
  - kurzfristige/rasche Erzeugung und Beseitigung der Prozesse
    - da keine expliziten Kommunikationskanäle auf-/abgebaut werden müssen
  - Kommunikation über direkt gemeinsam benutzte Daten
    - Verriegelung (*interlocking*) von Daten im Haupt-/Arbeitsspeicher
    - Zugriff und Schutz globaler Daten über prozedurale Schnittstellen
  - eine Identifizierung des Ausführungskontextes jeder Funktion
    - genauer: die sich gerade in Ausführung befindet
- ↳ Systemressourcen repräsentieren sich als Datenstrukturen
  - global oder für mehrere Prozesse gemeinsam
- ↳ Anwendungen sind mit Prozessen assoziiert
  - deren Bedürfnisse sind in Aufrufe an systemseitige Prozeduren kodiert





- i.d.R. explizite Koordinierung der Systemprozesse durch die Exekutive
  - Systemprozesse rekrutieren sich implizit aus den Anwendungsprozessen
    - sie bilden den „Fortsatz“ der Ausführungsstränge auf Anwendungsebene



## Nachrichtenorientierung

- sequentieller Prozess

## Prozedurorientierung

- ein-/wechselseitiger Ausschluss



## Nachrichtenorientierung

- sequentieller Prozess
- Nachrichten
  - Kanal
  - Anschluss (*port*)

## Prozedurorientierung

- ein-/wechselseitiger Ausschluss
- Prozeduren
  - Name
  - Einstiegsadresse



## Nachrichtenorientierung

- sequentieller Prozess
- Nachrichten
  - Kanal
  - Anschluss (*port*)
- Anforderung
  - Beauftragung
    - Antwort (unverzögert)
    - Antwortzusage (verzögert)
  - Empfang
  - Beantwortung

## Prozedurorientierung

- ein-/wechselseitiger Ausschluss
- Prozeduren
  - Name
  - Einstiegsadresse
- Aktivierungen
  - Aufruf
    - Vordergrundauführung
    - Hintergrundauführung
  - Einsprung
  - Rücksprung



## Nachrichtenorientierung

- sequentieller Prozess
- Nachrichten
  - Kanal
  - Anschluss (*port*)
- Anforderung
  - Beauftragung
    - Antwort (unverzögert)
    - Antwortzusage (verzögert)
  - Empfang
  - Beantwortung
- Zuteilung
  - Selektionsanweisung

## Prozedurorientierung

- ein-/wechselseitiger Ausschluss
- Prozeduren
  - Name
  - Einstiegsadresse
- Aktivierungen
  - Aufruf
    - Vordergrundauführung
    - Hintergrundauführung
  - Einsprung
  - Rücksprung
- Bindung
  - Prozedurdeklaration



## Nachrichtenorientierung

- sequentieller Prozess
- Nachrichten
  - Kanal
  - Anschluss (*port*)
- Anforderung
  - Beauftragung
    - Antwort (unverzögert)
    - Antwortzusage (verzögert)
  - Empfang
  - Beantwortung
- Zuteilung
  - Selektionsanweisung
- selektiver Nachrichtenempfang

## Prozedurorientierung

- ein-/wechselseitiger Ausschluss
- Prozeduren
  - Name
  - Einstiegsadresse
- Aktivierungen
  - Aufruf
    - Vordergrundausführung
    - Hintergrundausführung
  - Einsprung
  - Rücksprung
- Bindung
  - Prozedurdeklaration
- Bedingungsvariable



Einleitung

Architektur

Aufbaustruktur

Architekturformen

Monolith

Mikrokern

Makrokern

Exokern

Dualität

Betriebssystemstruktur

Zusammenfassung



- Einleitung
  - Definition „Betriebssystemarchitektur“:
    1. ein Operationsprinzip für die Systemsoftware
    2. Struktur ihres Aufbaus aus den einzelnen Systemprogrammen
  - das Operationsprinzip manifestiert sich in der Rechnerbetriebsart
- Architekturformen
  - monolithisch, Zusammenschluss aller Systemfunktionen
  - mikrokernbasiert, Trennung der Systemfunktionen
  - makrokernbasiert, Trennung zusammengesetzter Systemfunktionen
  - exokernbasiert, Trennung von Verwaltung und Schutz

↪ ohne/mit wohldefinierter innerer (hierarchischer) Struktur
- Dualität
  - nachrichtenorientiertes System: monolithisch, mikro-/makrokernbasiert
  - prozedurorientiertes System: monolithisch, exokernbasiert

↪ keines der beiden Modelle ist von Natur aus vorzuziehen



## ■ Einleitung

### ■ Definition „Betriebssystemarchitektur“:

1. ein Operationsprinzip für die Systemsoftware
2. Struktur ihres Aufbaus aus den einzelnen Systemprogrammen

### ■ das Operationsprinzip manifestiert sich in der Rechnerbetriebsart

## ■ Architekturformen

- monolithisch, Zusammenschluss aller Systemfunktionen
- mikrokernbasiert, Trennung der Systemfunktionen
- makrokernbasiert, Trennung zusammengesetzter Systemfunktionen
- exokernbasiert, Trennung von Verwaltung und Schutz

↪ ohne/mit wohldefinierter innerer (hierarchischer) Struktur

## ■ Dualität

- nachrichtenorientiertes System: monolithisch, mikro-/makrokernbasiert
- prozedurorientiertes System: monolithisch, exokernbasiert

↪ keines der beiden Modelle ist von Natur aus vorzuziehen



## ■ Einleitung

### ■ Definition „Betriebssystemarchitektur“:

1. ein Operationsprinzip für die Systemsoftware
2. Struktur ihres Aufbaus aus den einzelnen Systemprogrammen

### ■ das Operationsprinzip manifestiert sich in der Rechnerbetriebsart

## ■ Architekturformen

### ■ monolithisch, Zusammenschluss aller Systemfunktionen

### ■ mikrokernbasiert, Trennung der Systemfunktionen

### ■ makrokernbasiert, Trennung zusammengesetzter Systemfunktionen

### ■ exokernbasiert, Trennung von Verwaltung und Schutz

↪ ohne/mit wohldefinierter innerer (hierarchischer) Struktur

## ■ Dualität

### ■ nachrichtenorientiertes System: monolithisch, mikro-/makrokernbasiert

### ■ prozedurorientiertes System: monolithisch, exokernbasiert

↪ keines der beiden Modelle ist von Natur aus vorzuziehen



## ■ Einleitung

### ■ Definition „Betriebssystemarchitektur“:

1. ein Operationsprinzip für die Systemsoftware
2. Struktur ihres Aufbaus aus den einzelnen Systemprogrammen

### ■ das Operationsprinzip manifestiert sich in der Rechnerbetriebsart

## ■ Architekturformen

### ■ monolithisch, Zusammenschluss aller Systemfunktionen

### ■ mikrokernbasiert, Trennung der Systemfunktionen

### ■ makrokernbasiert, Trennung zusammengesetzter Systemfunktionen

### ■ exokernbasiert, Trennung von Verwaltung und Schutz

↪ ohne/mit wohldefinierter innerer (hierarchischer) Struktur

## ■ Dualität

### ■ nachrichtenorientiertes System: monolithisch, mikro-/makrokernbasiert

### ■ prozedurorientiertes System: monolithisch, exokernbasiert

↪ keines der beiden Modelle ist von Natur aus vorzuziehen



- [1] AMDAHL, G. M. ; BLAAUW, G. A. ; BROOKS, JR., F. P.:  
Architecture of the IBM System/360.  
In: *IBM Journal* (1964), Apr., S. 87–101
- [2] DIJKSTRA, E. W.:  
The Structure of the “THE”-Multiprogramming System.  
In: *Communications of the ACM* 11 (1968), Mai, Nr. 5, S. 341–346
- [3] ENGLER, D. R. ; KAASHOEK, M. F. ; O’TOOLE, J. :  
Exokernel: An Operating System Architecture for Application-Level Resource Management.  
In: [6], S. 251–266
- [4] GENTLEMAN, W. M.:  
Message Passing Between Sequential Processes: The Reply Primitive and the Administrator Concept.  
In: *Software—Practice and Experience* 11 (1981), Nr. 5, S. 435–466
- [5] GILOI, W. K.:  
*Rechnerarchitektur*.  
Berlin Heidelberg : Springer-Verlag, 1993. –  
ISBN 3–540–56355–5



- [6] JONES, M. B. (Hrsg.):  
*Proceedings of the 15th ACM Symposium on Operating System Principles (SOSP '95)*.  
ACM Press, 1995 . –  
ISBN 0-89791-715-4
- [7] LAUER, H. C. ; NEEDHAM, R. M.:  
On the Duality of Operating System Structures.  
In: LANCIAUX, D. (Hrsg.) ; Institut de Recherche en Informatique et Automatique (IRIA) (Veranst.): *Proceedings of the Second International Symposium on Operating Systems Theory and Practice* Institut de Recherche en Informatique et Automatique (IRIA), 1978, S. 3–19 (as reprint in SIGOPS Operating Systems Review, ACM, Bd. 13 Nr. 2, 1979)
- [8] LIEDTKE, J. :  
On  $\mu$ -Kernel Construction.  
In: [6], S. 237–250
- [9] LIONS, J. :  
*A Commentary on the Sixth Edition UNIX Operating System*.  
The University of New South Wales, Department of Computer Science, Australia :  
<http://www.lemis.com/grog/Documentation/Lions, 1977>



- [10] PARNAS, D. L.:  
On a 'Buzzword': Hierarchical Structure.  
In: ROSENFELD, J. L. (Hrsg.): *Information Processing 74, Proceedings of the IFIP Congress 74*.  
New York, NY, USA : North-Holland Publishing Company, 1974. –  
ISBN 0-7204-2803-3, S. 336-339
- [11] PARNAS, D. L.:  
Some Hypothesis About the "Uses" Hierarchy for Operating Systems / TH  
Darmstadt, Fachbereich Informatik.  
1976 (BSI 76/1). –  
Forschungsbericht
- [12] POLLIO GEN. „VITRUV“, M. V.:  
*De Architectura Libris Decem*.  
Primus Verlag, 1996 (Original 27 v. Chr.)
- [13] SINGH, A. :  
*Mac OS X Internals: A Systems Approach*.  
Addison-Wesley, 2006

