Betriebssystemtechnik

Adressräume: Trennung, Zugriff, Schutz

III. Betriebssystemarchitektur

Wolfgang Schröder-Preikschat / Volkmar Sieh

SS 2024

Gliederung

Architektur im Allgemeinen

Ar·chi·tek'tur[1] $<$ *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

[1] `http://www.duden.de/rechtschreibung/Architektur`

Architektur im Allgemeinen

Ar·chi·tek'tur[1] $< f.; -, -en >$ (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...

[1] `http://www.duden.de/rechtschreibung/Architektur`

Architektur im Speziellen: Rechner I

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.*

Architektur im Speziellen: Rechner I

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.

Architektur im Speziellen: Rechner II

vgl. [5, S. 21]

- Rechnerarchitektur

Architektur im Speziellen: Rechner II

vgl. [5, S. 21]

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

Architektur im Speziellen: Rechner II

vgl. [5, S. 21]

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

[2] Impliziert die Spezifikation der Funktion, die es zu erfüllen gilt.

Architektur im Speziellen: Rechner II

vgl. [5, S. 21]

- Rechnerarchitektur ist bestimmt durch ein *Operationsprinzip* für die Hardware und die *Struktur* ihres Aufbaus aus ihren Betriebsmitteln
  - Operationsprinzip definiert das funktionelle[2] 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

[2] Impliziert die Spezifikation der Funktion, die es zu erfüllen gilt.

Architektur im Speziellen: Rechner II

vgl. [5, S. 21]

- Rechnerarchitektur ist bestimmt durch ein *Operationsprinzip* für die Hardware und die *Struktur* ihres Aufbaus aus ihren Betriebsmitteln
  - Operationsprinzip definiert das funktionelle[2] 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

[2] Impliziert die Spezifikation der Funktion, die es zu erfüllen gilt.

Architektur im Speziellen: Rechner II

vgl. [5, S. 21]

- Rechnerarchitektur ist bestimmt durch ein *Operationsprinzip* für die Hardware und die *Struktur* ihres Aufbaus aus ihren Betriebsmitteln
  - Operationsprinzip definiert das funktionelle[2] 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

[2] Impliziert die Spezifikation der Funktion, die es zu erfüllen gilt.

Architektur im Speziellen: Rechner II

vgl. [5, S. 21]

- Rechnerarchitektur ist bestimmt durch ein *Operationsprinzip* für die Hardware und die *Struktur* ihres Aufbaus aus ihren Betriebsmitteln
  - Operationsprinzip definiert das funktionelle[2] 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

[2] Impliziert die Spezifikation der Funktion, die es zu erfüllen gilt.

Architektur im Speziellen: Betriebssystem III

Definition (in Anlehnung an [5])

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

Architektur im Speziellen: Betriebssystem III

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

Architektur im Speziellen: Betriebssystem III

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

Architektur im Speziellen: Betriebssystem III

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

Architektur im Speziellen: Betriebssystem III

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

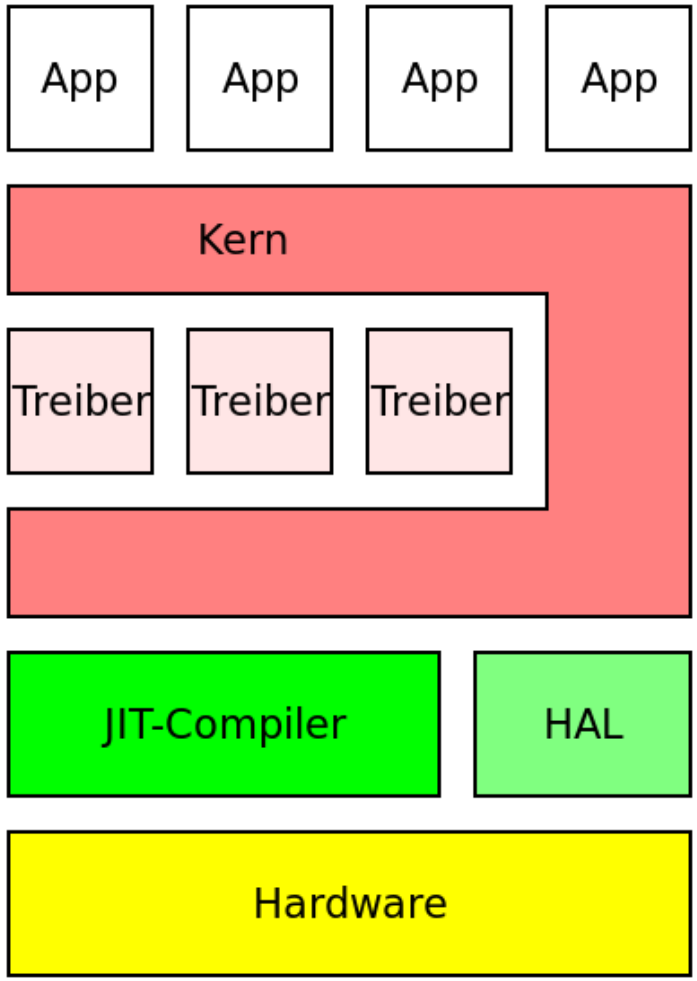
  Anzahl ein-, mehr-, vielkerniger Uni-/Multiprozessor

Allgemeine Struktur des Aufbaus

Spezielle Struktur des Aufbaus von StuBS

Spezielle Struktur des Aufbaus von StuBS$_{ml}$

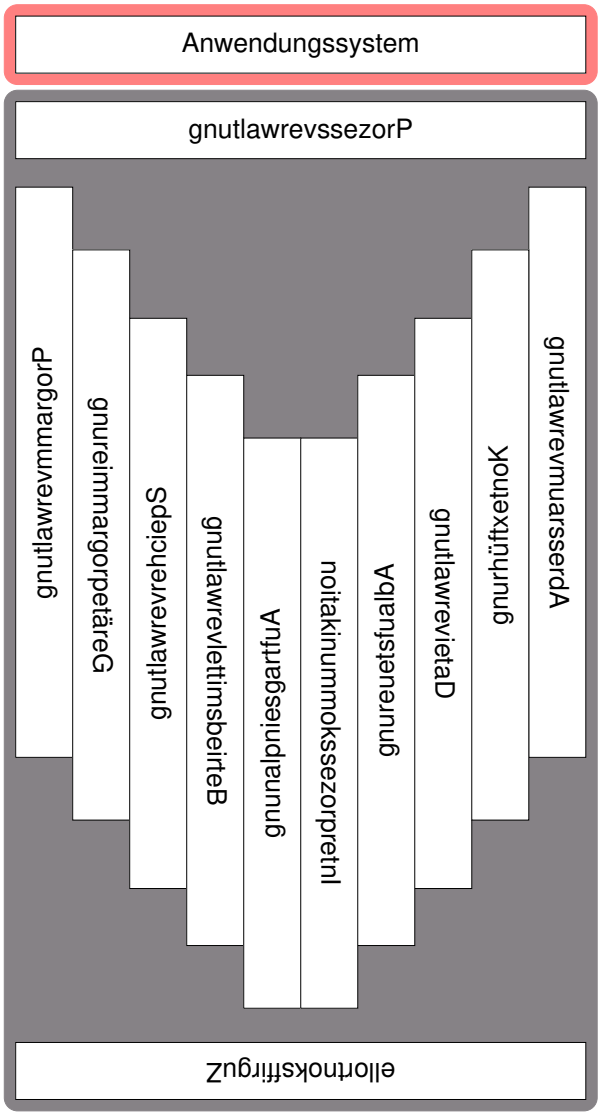
Beispiel: Struktur JITTY



Gliederung

Monolithisches Betriebssystem

bspw. [9]

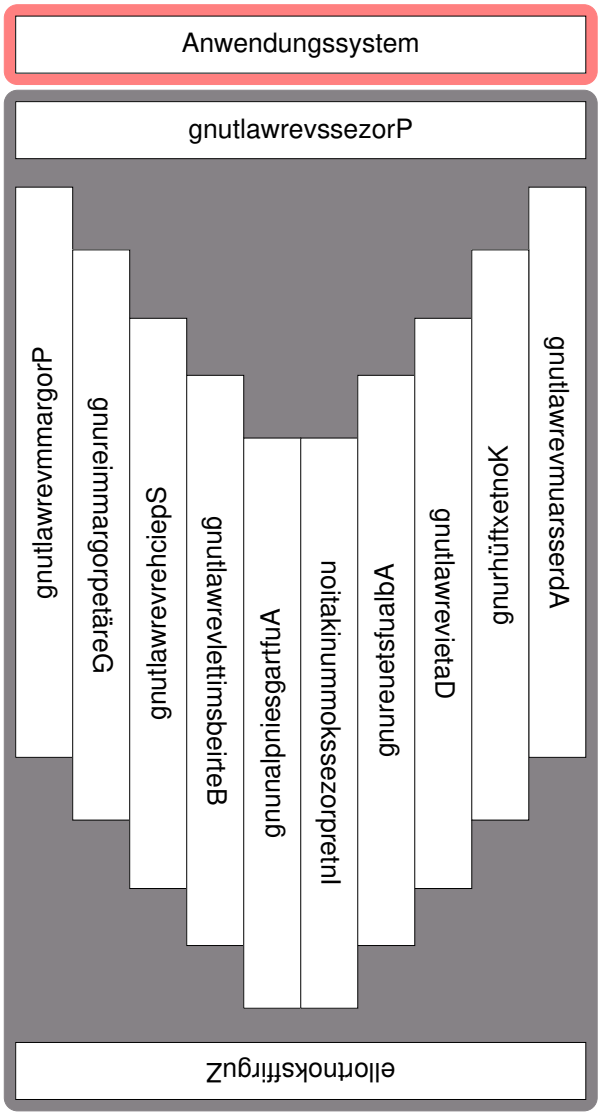


Mo·no·lith (grch. *monólithos*)

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

Monolithisches Betriebssystem

bspw. [9]



Mo·no·lith (grch. *monólithos*)

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

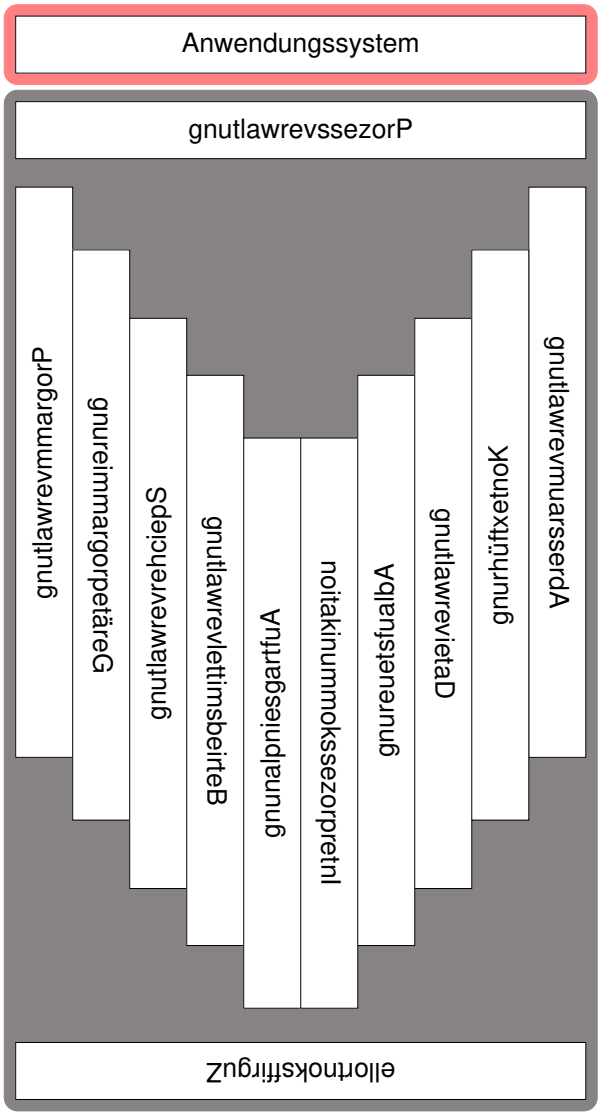
Architekturform des *Absolutismus*[3]

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

[3] Eigene Machtvollkommenheit, ohne Mitwirkung ständischer Institutionen.

Monolithisches Betriebssystem

bspw. [9]



Mo·no·lith (grch. *monólithos*)

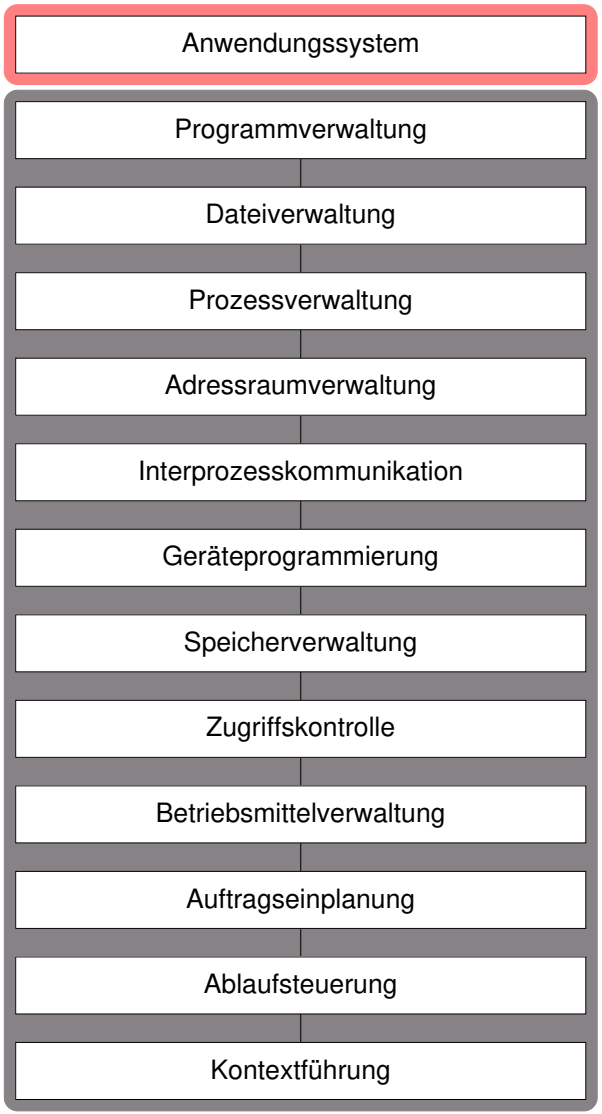
1. großer Steinblock
2. Stein aus einem Guss
3. monumentales Kunstwerk aus nur einem Stein

Architekturform des *Absolutismus*[3]

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

[3] Eigene Machtvollkommenheit, ohne Mitwirkung ständischer Institutionen.

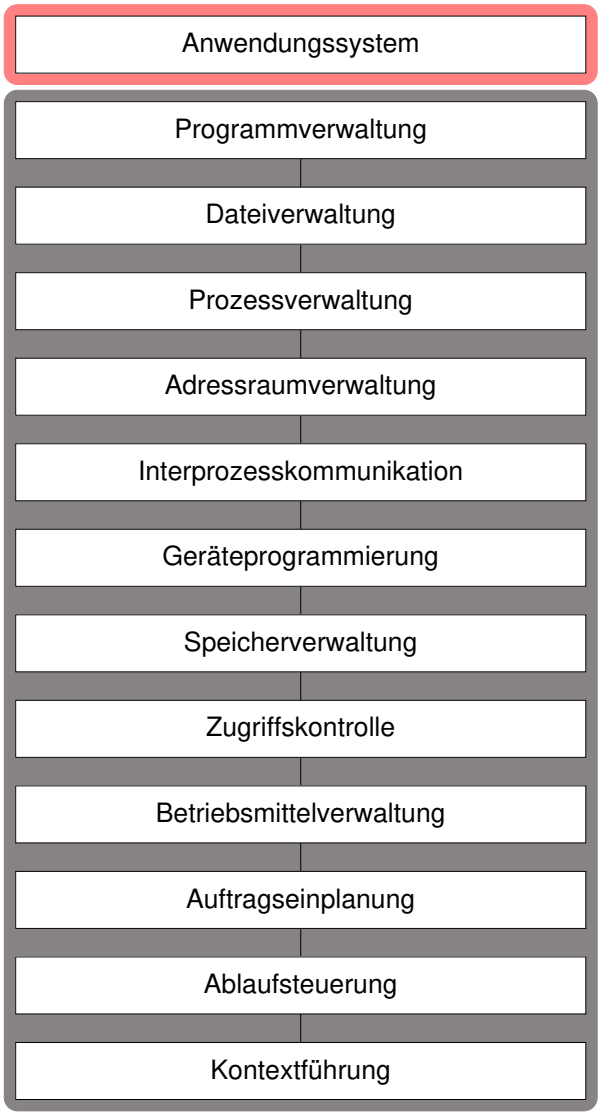
Monolithisches schichtenstrukturiertes Betriebssystem



- Architekturform des *Historismus*[4] [2]
- ungetrennte Systemfunktionen
  - einheitlicher Adressraum
  - keine Fehlereingrenzung
  - derselbe Arbeitsmodus: privilegiert

[4] Rückgriff auf und Nachahmung älterer Stilrichtungen.

Monolithisches schichtenstrukturiertes Betriebssystem

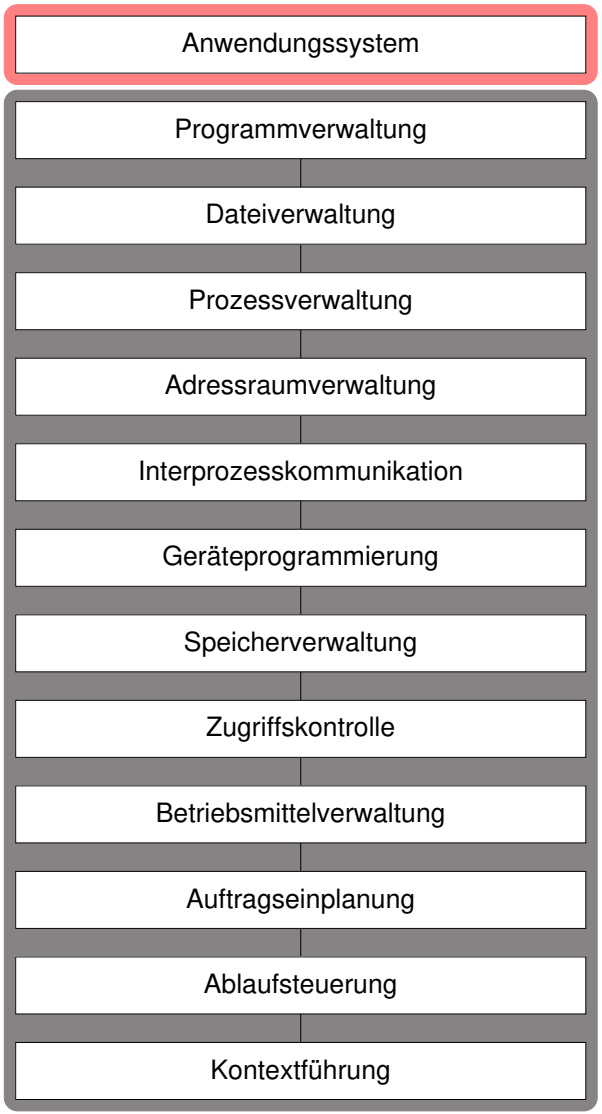


Architekturform des *Historismus*[4] [2]

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

[4] Rückgriff auf und Nachahmung älterer Stilrichtungen.

Monolithisches schichtenstrukturiertes Betriebssystem



Architekturform des *Historismus*[4] [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]

[4] Rückgriff auf und Nachahmung älterer Stilrichtungen.

Mikrokernbasiertes Betriebssystem

Architekturform der *Moderne*[5] [8]

- getrennte Systemfunktionen
  - in separierten Adressräumen
  - starke Fehlereingrenzung
  - verschiedene Arbeitsmodi
    - privilegierter Kern (dunkel)
    - unprivilegierte Systemprozesse (hell)

[5] Umbruch in allen (Lebens-)Bereichen gegenüber der Tradition.

Mikrokernbasiertes Betriebssystem

Architekturform der *Moderne*[5] [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)

[5] Umbruch in allen (Lebens-)Bereichen gegenüber der Tradition.

Mikrokernbasiertes Betriebssystem

Architekturform der *Moderne*[5] [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

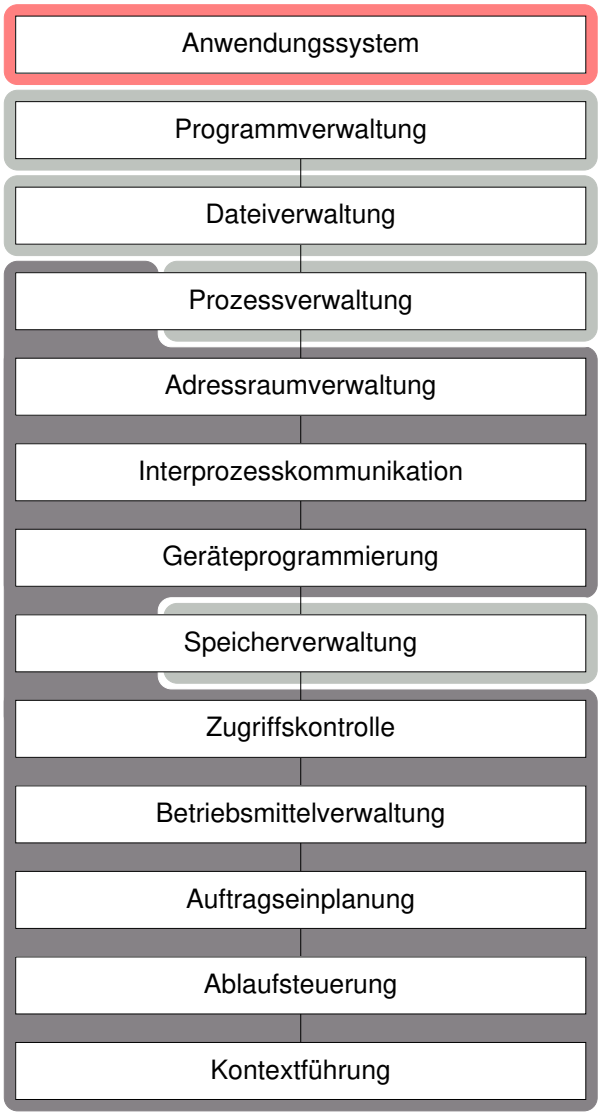
  Prozesse IPC (hell)

  Kern Systemaufruf (dunkel)
- mittelkörnige Struktur
  - Granularität auf Adressraumebene
  - Kernadressraum als Monolith

[5] Umbruch in allen (Lebens-)Bereichen gegenüber der Tradition.

Makrokernbasiertes Betriebssystem

auch: Hybridkern



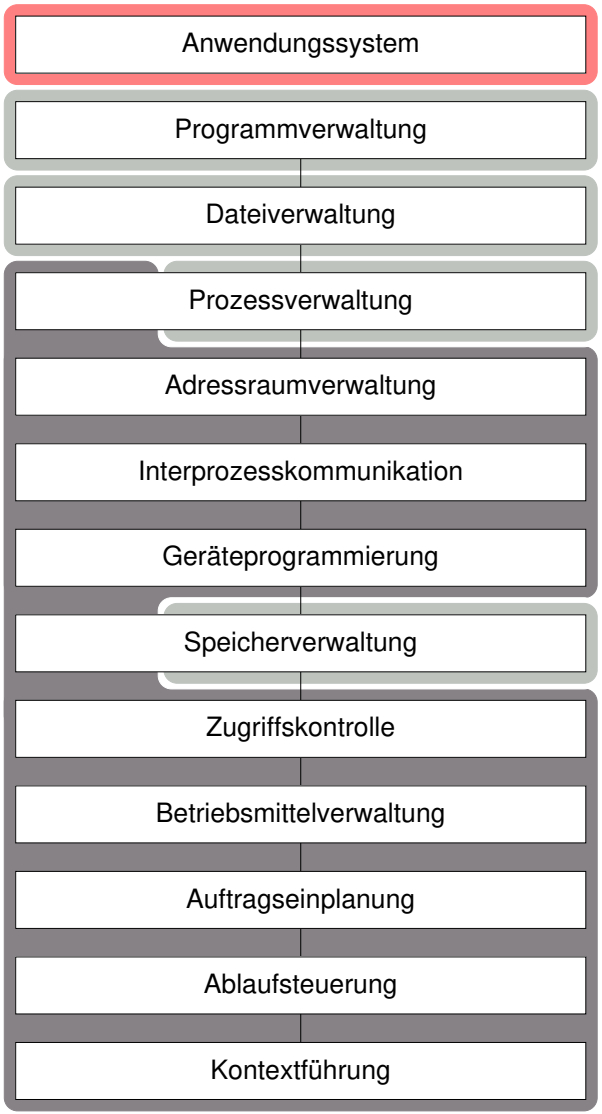
Arch.-form des *Utilitarismus*[6] [13]

- teils getrennte Systemfunktionen
  - in separierten Adressräumen
  - schwache Fehlereingrenzung
  - verschiedene Arbeitsmodi
    - privilegierter Hybridkern (dunkel)
    - unprivilegierte Systemprozesse (hell)

[6] Unterbegriff zu Konsequentialismus: Der Zweck heiligt die Mittel.

Makrokernbasiertes Betriebssystem

auch: Hybridkern



Arch.-form des *Utilitarismus*[6] [13]

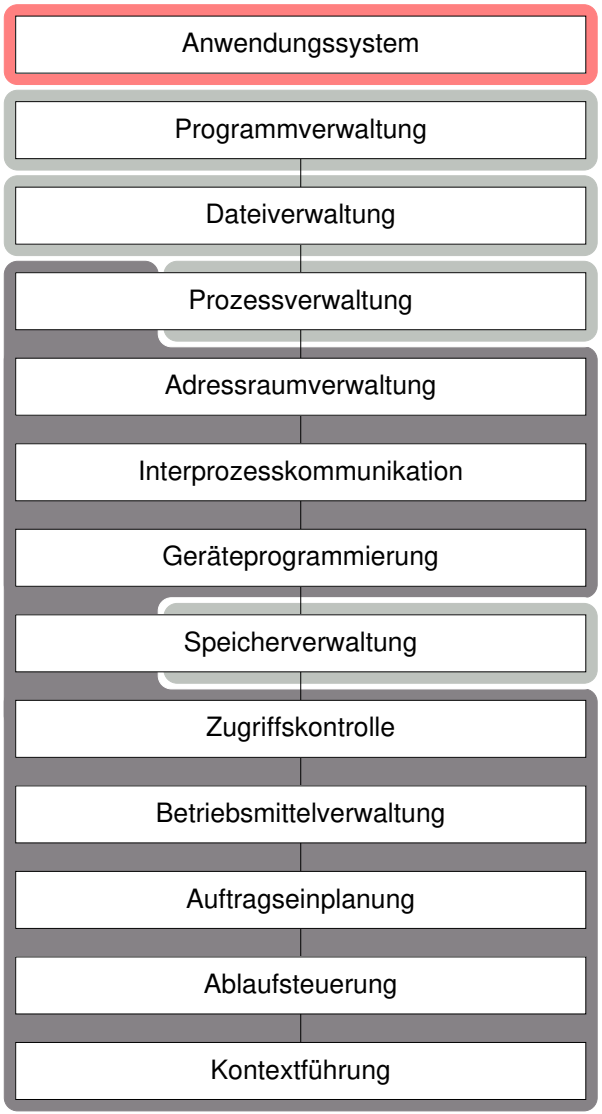
- teils getrennte Systemfunktionen
  - in separierten Adressräumen
  - schwache Fehlereingrenzung
  - verschiedene Arbeitsmodi
    - privilegierter Hybridkern (dunkel)
    - unprivilegierte Systemprozesse (hell)
- eingeschränkte Modularität
  - bedingte räumliche Verpflechtung
  - unaufwändige Interaktion im System

Prozesse IPC (hell & dunkel)
Hybridkern Systemaufruf (dunkel)

[6] Unterbegriff zu Konsequentialismus: Der Zweck heiligt die Mittel.

Makrokernbasiertes Betriebssystem

auch: Hybridkern



Arch.-form des *Utilitarismus*[6] [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

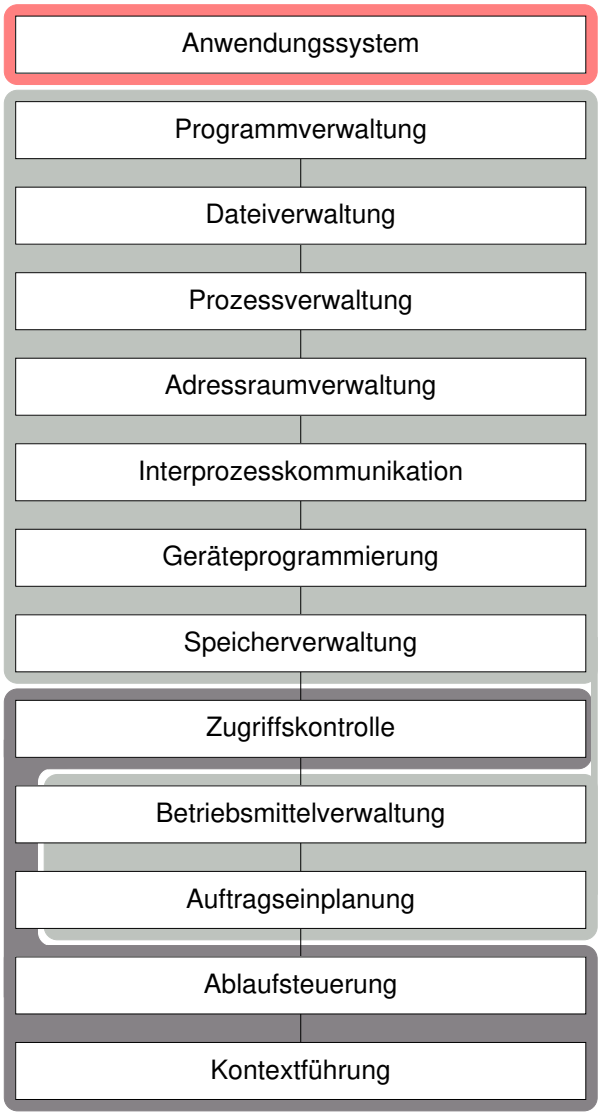
    Prozesse IPC (hell & dunkel)

    Hybridkern Systemaufruf (dunkel)
- grobkörnige Struktur
  - Granularität auf Adressraumebene
  - Hybridkernadressraum als Monolith

[6] Unterbegriff zu Konsequentialismus: Der Zweck heiligt die Mittel.

Exokernbasiertes Betriebssystem

„Hardwaremultiplexer“



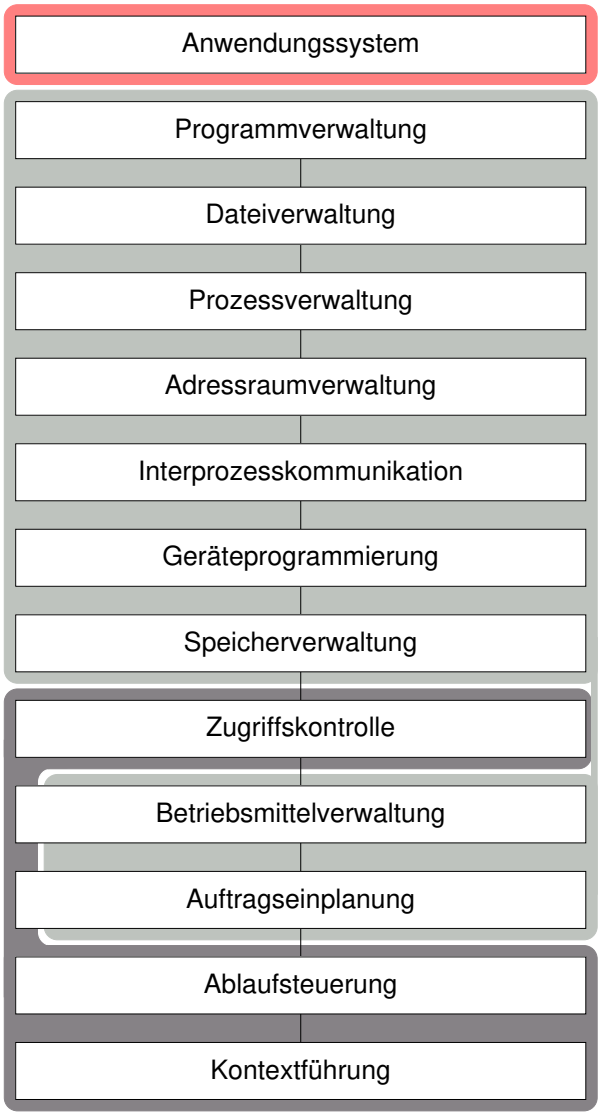
Architekturform der *Avantgarde*[7] [3]

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

[7] Fortschrittsorientiert, Radikalität gegenüber bestehenden Verhältnissen.

Exokernbasiertes Betriebssystem

„Hardwaremultiplexer“



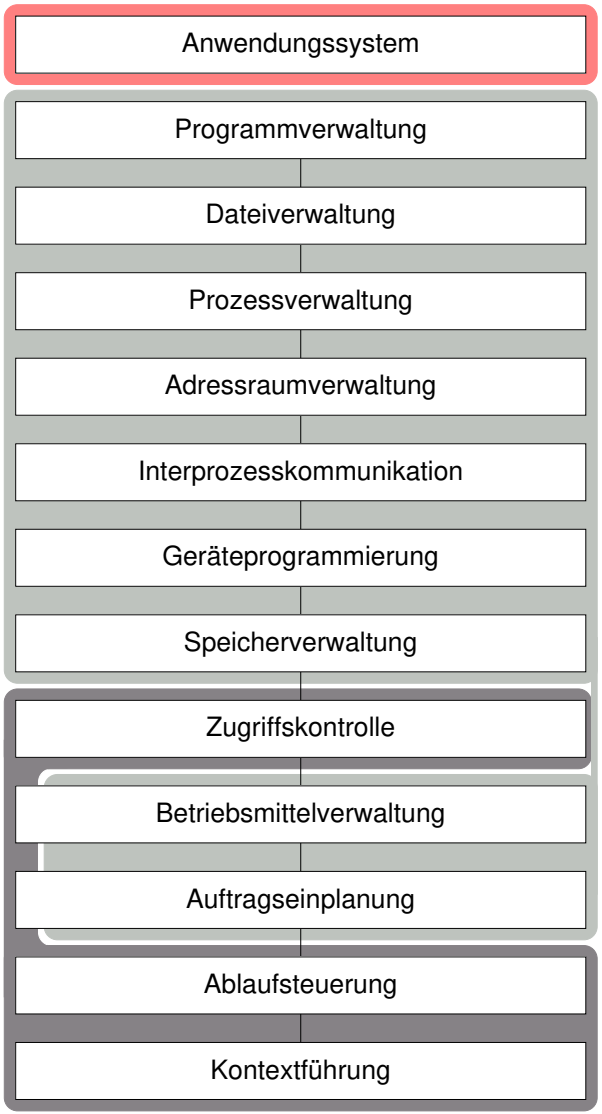
Architekturform der *Avantgarde*[7] [3]

- 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

[7] Fortschrittsorientiert, Radikalität gegenüber bestehenden Verhältnissen.

Exokernbasiertes Betriebssystem

„Hardwaremultiplexer“



Architekturform der *Avantgarde*[7] [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 Betriebsmittelwiderruf
  3. imperativer Betriebsmittelentzug

[7] Fortschrittsorientiert, Radikalität gegenüber bestehenden Verhältnissen.

Gliederung

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.

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.*

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.*

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

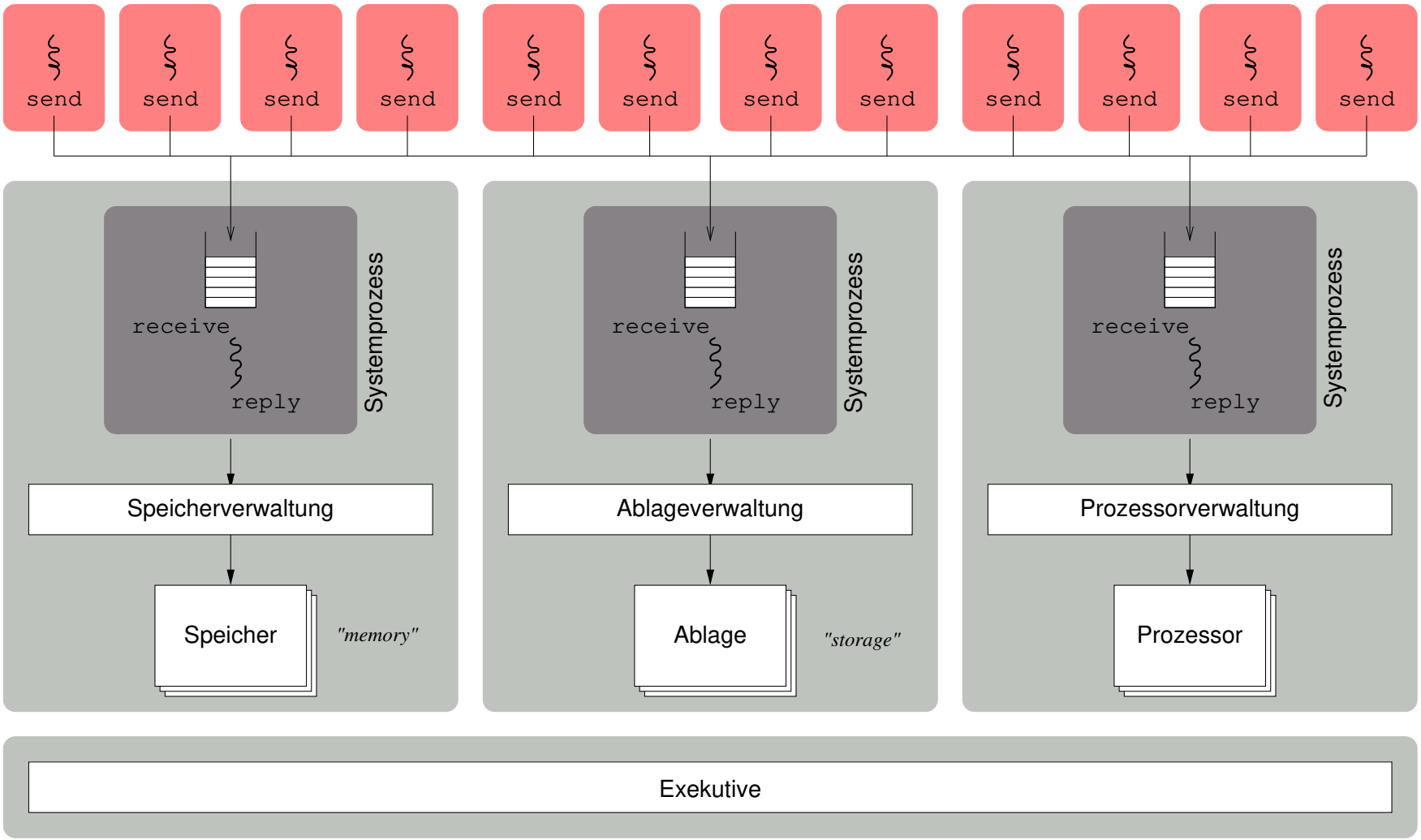
Many operating system designs can be placed into one of two very rough categories, depending upon how they implement and use the notions of process and synchronization.

- *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] **!!!**

Nachrichtenorientiertes System I



Nachrichtenorientiertes System II

vgl. [7] vs. [4, S. 444]

Charakteristik:

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

Nachrichtenorientiertes System II

vgl. [7] vs. [4, S. 444]

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

Nachrichtenorientiertes System II

vgl. [7] vs. [4, S. 444]

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

Nachrichtenorientiertes System II

vgl. [7] vs. [4, S. 444]

Charakteristik:

- kleine und relativ statische Anzahl großer Prozesse ⇝ [4]: Teams
  - steht und fällt mit dem Prozessmodell und dessen Implementierung
- expliziter Satz von Nachrichtenkanälen zwischen diesen Prozessen
  - typischerweise langfristige Bindung ⇝ [4]: kurzfristig, verbindungslos
  - Nachrichtenversenden (*message passing*) als Interaktionsmittel
- relativ begrenzter Umfang von direkt gemeinsam benutzten Daten
  - im Haupt- oder Arbeitsspeicher ⇝ [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 ⇝ [4]: Teams
  - Prozesse überschreiten Schutzgrenzen eher selten
    - wenn, dann nur, um kurz die Exekutive oder den Kern zu betreten

Nachrichtenorientiertes System II

vgl. [7] vs. [4, S. 444]

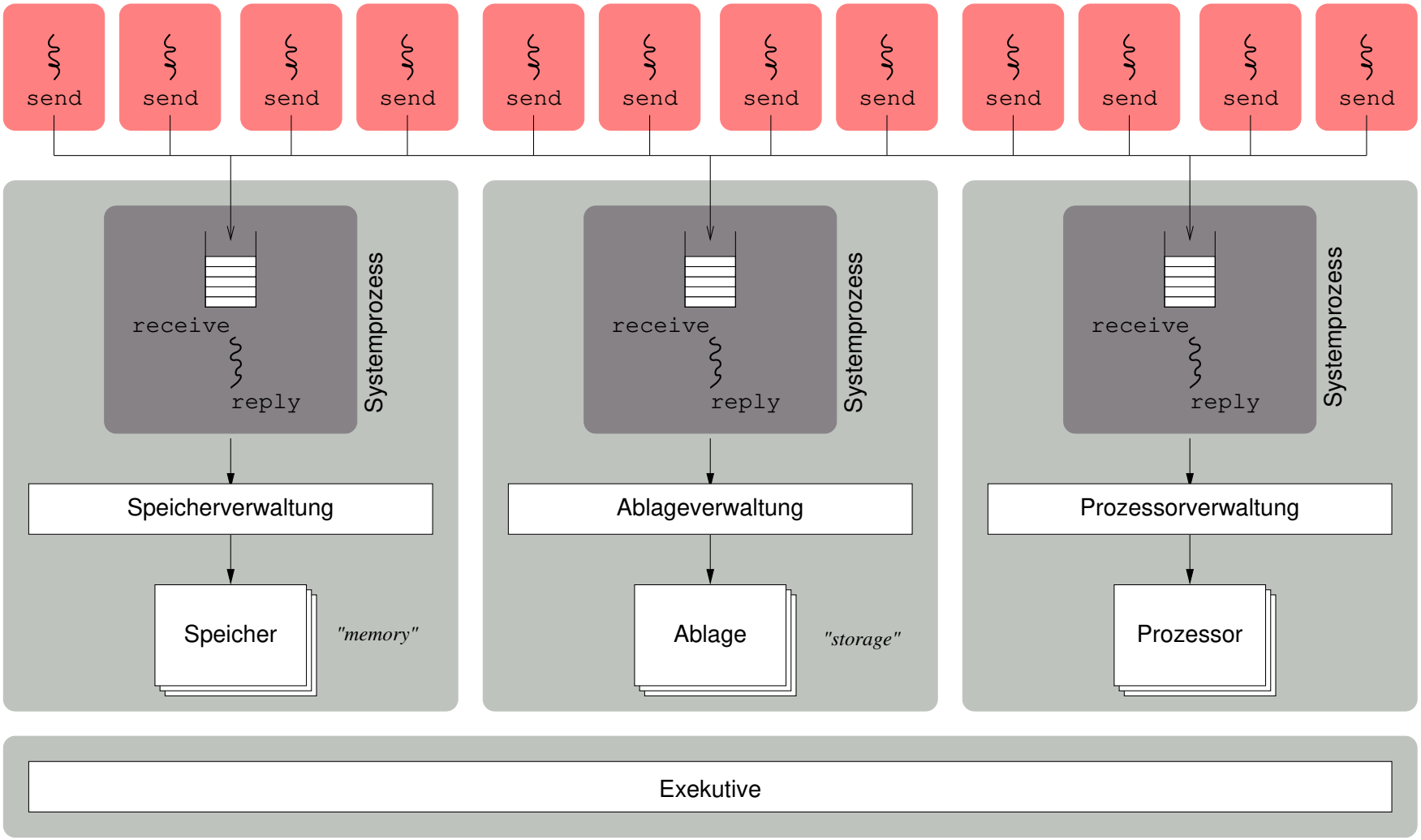
Charakteristik:

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

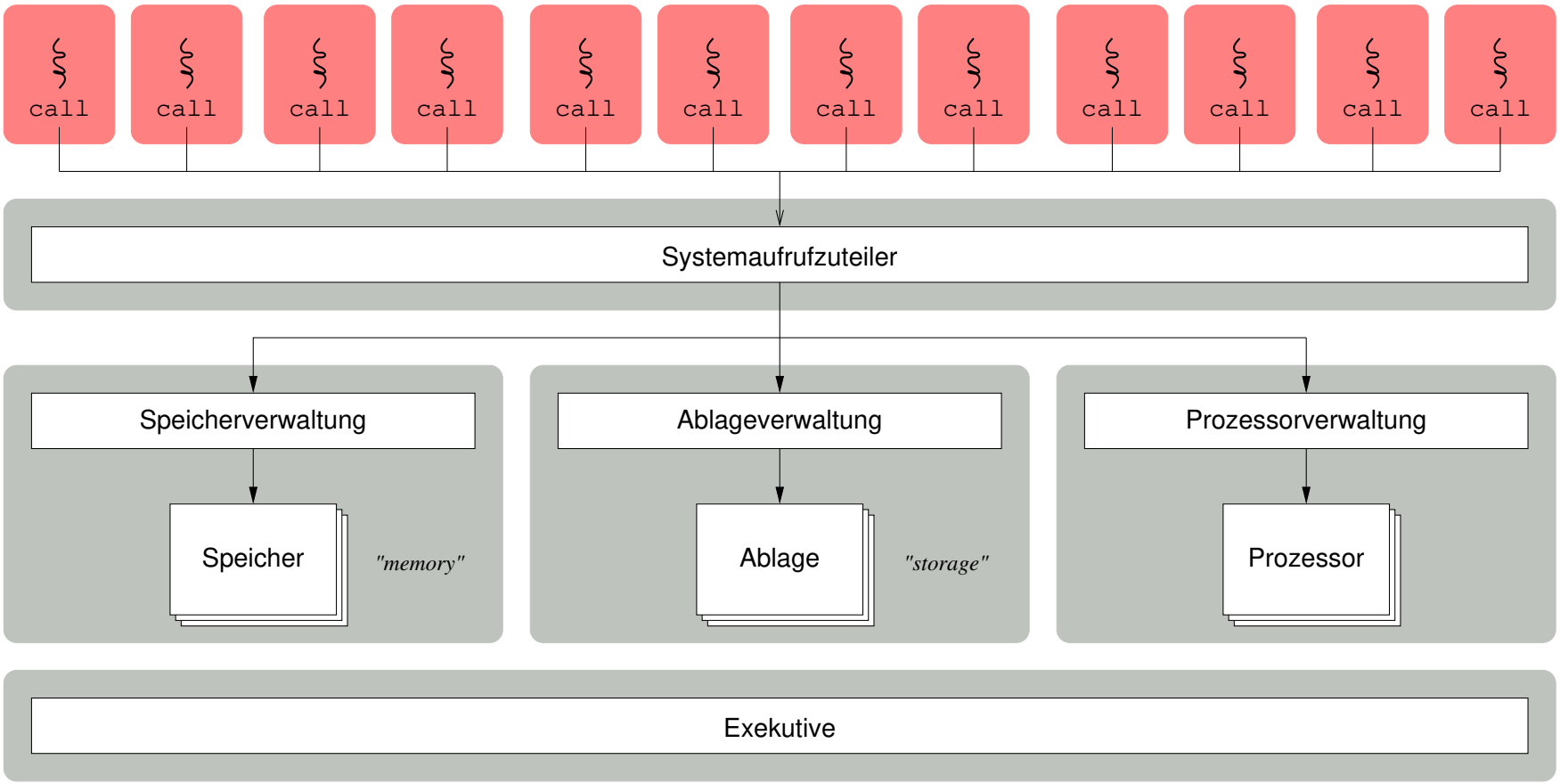
↪ Anwendungsbedürfnisse erhält das System in Nachrichten kodiert

Nachrichtenorientiertes System III



- im-/explizite Koordinierung der Systemprozesse durch die Exekutive

Prozedurorientiertes System I



Prozedurorientiertes System II

vgl. [7]

Charakteristik:

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

Prozedurorientiertes System II

vgl. [7]

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

Prozedurorientiertes System II

vgl. [7]

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

Prozedurorientiertes System II

vgl. [7]

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

Prozedurorientiertes System II

vgl. [7]

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

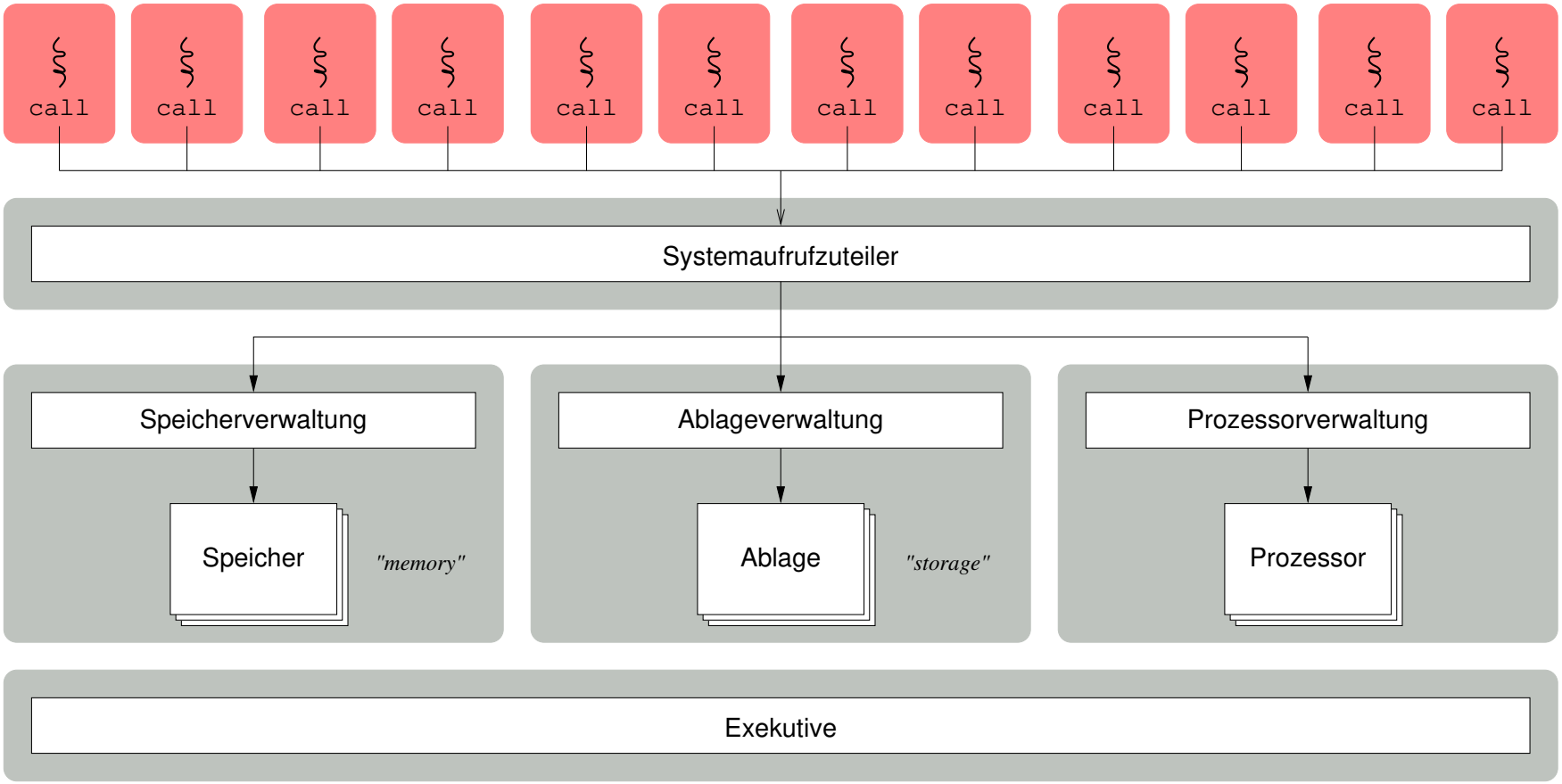
↪ Systemressourcen repräsentieren sich als Datenstrukturen

- global oder für mehrere Prozesse gemeinsam

↪ Anwendungen sind mit Prozessen assoziiert

- deren Bedürfnisse sind in Aufrufe an systemseitige Prozeduren kodiert

Prozedurorientiertes System III



- 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

Gegenüberstellung

vgl. [7, S. 12]

Nachrichtenorientierung

- sequentieller Prozess

Prozedurorientierung

- ein-/wechselseitiger Ausschluss

Gegenüberstellung

vgl. [7, S. 12]

Nachrichtenorientierung

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

Prozedurorientierung

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

Gegenüberstellung

vgl. [7, S. 12]

Nachrichtenorientierung

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

Prozedurorientierung

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

Gegenüberstellung

vgl. [7, S. 12]

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
    - Vordergrundausführung
    - Hintergrundausführung
  - Einsprung
  - Rücksprung
- Bindung
  - Prozedurdeklaration

Gegenüberstellung

vgl. [7, S. 12]

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

Gliederung

Resümee

- 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 zusammengeschlossener 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

Resümee

- 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 zusammengeschlossener 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

Resümee

- 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 zusammengeschlossener 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

Resümee

- 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 zusammengeschlossener 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

Literaturverzeichnis I

[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

Literaturverzeichnis II

[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 μ-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

Literaturverzeichnis III

[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