

Aufgabenblatt 1

...Koniev...

Abgabetermin: 14.05.2017 23:55 Uhr

Aufgabe 1.2: Betriebssysteme (1,5 Punkte)

a) Welche drei Aufgabenbereiche und sieben Einzelfunktionen besitzt ein Betriebssystem? (0,5 Punkte)

Grobe Aufteilung in drei Aufgabenbereiche:

1. Bereitstellung von Hilfsmitteln für Benutzerprogramme
2. Vernachlässigung der genauen Benutzerkenntnis von HW-Eigenschaften und spezieller SW-Komponenten, wie z.B. Gerätetreiber
3. Koordination und Vergabe der zur Verfügung stehenden Betriebsmittel an mehrere, gleichzeitig arbeitende Benutzer

Einzelfunktionen eines Betriebssystems:

1. Unterbrechungsverarbeitung (interrupt handling)
2. Verteilung (dispatching): Prozessumschaltung
3. Betriebsmittelverwaltung (resource management): Belegen, Freigeben und Betreiben von Betriebsmitteln, Werkzeuge zur Prozesssynchronisation
4. Programmallokation (program allocation): Linken von Teilprogrammen, Laden und Verdrängen von Programmen in/aus dem Hauptspeicher
5. Dateiverwaltung (file mangement):

- Organisation des Speicherplatzes in Form von Dateien auf Datenträgern
 - Bereitstellung von Funktionen zur Speicherung, Modifikation und Wiedergewinnung der gespeicherten Informationen
6. Auftragsteuerung (job control): Festlegung der Reihenfolge, in der die eingegangenen Aufträge und deren Bestandteile bearbeitet werden sollen
7. Zuverlässigkeit (reliability):
- Funktionen zur Reaktion auf Störungen und Ausfälle der Rechnerhardware sowie auf Fehler in der Software
 - Korrektheit, Robustheit und Toleranz (ständig betriebsbereit unter der Aufrechterhaltung einer Mindestfunktionsfähigkeit)

Grundlegende Betriebssystemfunktionen:

1. Dateiverwaltung (filemanagement)
 - Organisation des Speicherplatzes in Form von Dateien auf Datenträgern
 - Bereitstellung von Funktionen zur Speicherung, Modifikation und Wiedergewinnung der gespeicherten Informationen
2. Auftragsteuerung (jobcontrol)
 - Festlegung der Reihenfolge, in der die eingegangenen Aufträge und deren Bestandteile bearbeitet werden sollen
3. Zuverlässigkeit (reliability)
 - Funktionen zur Reaktion auf Störungen und Ausfälle der Rechnerhardware sowie auf Fehler in der Software
 - Korrektheit, Robustheit und Toleranz (ständig betriebsbereit unter der Aufrechterhaltung einer Mindestfunktionsfähigkeit)

b) Was ist der Unterschied zwischen Mechanismen und Policies? Nennen Sie je zwei Beispiele für diese. (0,5 Punkte)

Wichtige Unterscheidung zwischen Mechanismen und Policies:
Ein Mechanismus ist quasi analog zu einer Interface in Java, in dem es sagt was für Funktionalität angefordert wird; Policies sind eher wie dies implementiert wird, analog zur Implimentation einer Interface in Java.

1. Mechanismus: Wie wird eine Aufgabe prinzipiell gelöst?
2. Policy: Welche Vorgaben/Parameter werden im konkreten Fall eingesetzt?

Beispiele:

1. Zeitscheibenprinzip

- Existenz eines Timers zur Bereitstellung von Unterbrechungen \Rightarrow Mechanismus
- Entscheidung, wie lange die entsprechende Zeit für einzelne Anwendungen / Anwendungsgruppen eingestellt wird \Rightarrow Policy
- Es werden Daten, die in der Nähe gespeichert sind, mit in den Cache gebracht bzw. räumliche Lokalität wird ausgenutzt \Rightarrow Mechanismus
- Entscheidung, wie viele auf jeder Seite des ausgewählten Dateis genommen werden und was sie ersetzen sollten \Rightarrow Policy

2. Trennung wichtig für Flexibilität

- Policies ändern sich im Laufe der Zeit oder bei unterschiedlichen Plattformen \Rightarrow Falls keine Trennung vorhanden, muss jedes Mal auch der grundlegende Mechanismus geändert werden
- Wünschenswert: Genereller Mechanismus, so dass eine Policiesveränderung durch Anpassung von Parametern umgesetzt werden kann

c) Nennen Sie die fünf häufigsten Designstrukturen für Betriebssysteme, und beschreiben Sie eine davon im Detail. (0,5 Punkte)

Häufige Designstrukturen für Betriebssysteme:

<pre>main() int x = 5;</pre>

«Stelle 1»

1. Monolithische Systeme
2. Geschichtete Systeme
3. Virtuelle Maschinen
4. Exokern
5. Client-Server-Systeme mit Mikrokern

Monolithische Systeme: Häufigste Realisierungsform

1. Große Menge von Funktionen mit wohl-definierten Schnittstellen für Parameter und Ergebnisse \Rightarrow Alle Funktionen bilden den Objektcode
2. Hauptprogramm: Ruft die Dienstfunktionen auf
3. Dienstfunktionen: Führen die Systemaufrufe durch
4. Hilfsfunktionen: stellen Mechanismen bereit, die von diesen benötigt werden, z.B. Kopieren von Daten aus einem Programm

Aufgabe 1.1: Logischer Adressraum (0,5 Punkte)

In der Vorlesung haben Sie das Layout des logischen Adressraums kennengelernt. Nachfolgend finden Sie den Code für ein einfaches Programm. Skizzieren Sie für die vier markierten Stellen jeweils den Aufbau des Stacks.

sum(0) n = 0 return 0;
sum(1) n = 1 return sum(0) + 1;
sum(2) n = 2 return sum(1) + 2;
sum(3) n = 3 return sum(2) + 3;
sum(4) n = 4 return sum(3) + 4;
sum(5) n = 5 return sum(4) + 5;
main() int x = 5;

«Stelle 2»

sum_direkt(5) return 15;
main() int x = 5;

«Stelle 3»

<pre>main() int x = 5; ergebnis = 15; ergebnis = 15; return 0;</pre>
--

«Stelle 4»

Quelle:

O. Kao: Systemprogrammierung 2017 «Folie 1»