Script **generated by TTT**

Title: Grundlagen_Betriebssysteme (30.11.2012)

Date: Fri Nov 30 08:31:35 CET 2012

Duration: 89:57 min

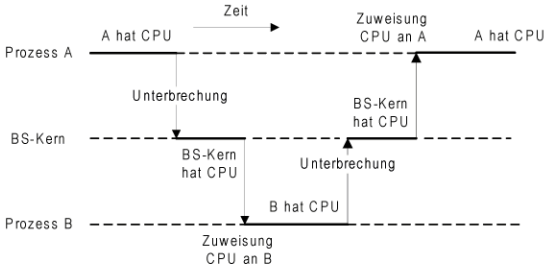
Pages: 23

Prozessorverwaltung

Eine wesentliche Aufgabe der Prozessorverwaltung besteht darin zu entscheiden, welcher der um den bzw. die Prozessor(en) konkurrierenden Prozesse (bzw. Threads) zu einem Zeitpunkt an den bzw. die Prozessor(en) gebunden wird. Dazu steht die BS-Komponente ***Scheduler*** zur Verfügung.

Prozessablauf besteht aus einer Sequenz von alternierenden CPU- und E/A-Perioden.

Zeitliche Verschränkung der Prozessbearbeitung bei einer CPU.



Kriterien

Scheduling-Strategien

Beispiel Unix Scheduling

Thread Scheduling

Mehrschichtiges Scheduling

Echtzeit Scheduling

Thread Scheduling

Die Prozessorzuteilung von Threads hängt von deren Art der Realisierung ab.

User-Threads

Realisierung der Threads im Benutzeradressraum ⇒ Kern hat keine Kenntnis bzgl. der Threads. BS-Scheduler wählt nur Prozess aus. Laufzeitsystem des Prozesses wählt rechenwilligen Thread des Prozesses aus; es kann ein beliebiges Scheduling-Verfahren für Prozesse verwendet werden.

Java Virtual Machines verwenden unterbrechendes Prioritäten-Scheduling für Threads;

10 ist die höchste und 1 die niedrigste Priorität;

Kernel-Threads

Realisierung der Threads im Systemadressraum ⇒ BS-Scheduler wählt den nächsten auszuführenden Thread aus.

a) Ist ausgewählter Thread demselben Prozess zugeordnet wie der vorher rechnende Thread ⇒ geringer Kontextwechsel.

b) Ist ausgewählter Thread nicht demselben Prozess zugeordnet wie der vorher rechnende Thread ⇒ aufwendiger Kontextwechsel.

Einlagern eines rechenwilligen Prozesses von der Platte in den Arbeitsspeicher ist aufwendig; deshalb Mehr-Schichten Scheduling.

Short-Term-Scheduler (CPU Scheduler)

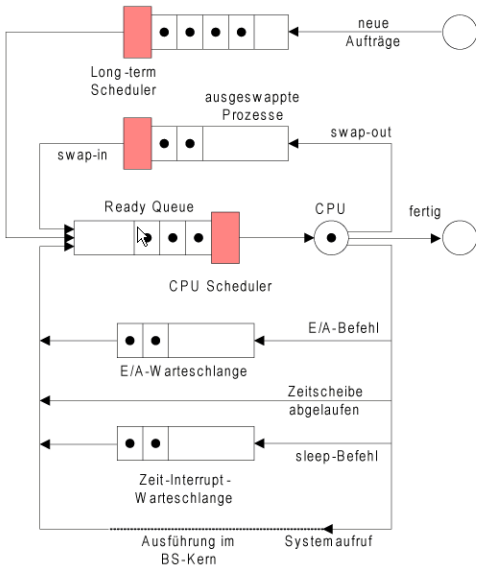
Auswahl eines geeigneten Prozesses aus der Ready-Queue; wird häufig aufgerufen; Verfahren siehe oben.

Long-Term-Scheduler

Auswahl rechenwilliger neuer Aufträge (meist Jobs aus dem Hintergrundbetrieb (batch)) und Einlagerung in den Arbeitsspeicher; Einfügen der Prozesse in die Ready-Queue.

Graphische Darstellung

Graphische Darstellung



Prozesse in Multimedia-Anwendungen führen oft zeitkritische Operationen aus. Bzgl. des Scheduling existieren zwei gegensätzliche Ziele:

a) ein ***unkritischer Prozess*** sollte nicht dauerhaft blockiert werden, weil zeitkritische Prozesse eine Ressource gänzlich auslasten.

b) ***zeitkritische Prozesse*** dürfen nicht durch Scheduling-Verfahren (Round-Robin oder Prioritäten) am zeitkritischen Fortschritt gehindert werden ⇒ Einhaltung von Zeitvorgaben.

Zuordnung von Kenngrößen zu zeitkritischen Prozessen

Bereitzeit (ready time): frühestmöglicher Ausführungsbeginn einer Aktivität.

Frist (deadline): spätester Zeitpunkt für die Beendigung einer Aktivität.

Ausführungszeit: worst-case Abschätzung für das zur vollständigen Ausführung einer Aktivität notwendige Zeitintervall.

Earliest Deadline First (EDF)

Rate-Monotonic Scheduling

Der Prozessor wird immer dem Prozess mit der am nächsten in der Zukunft liegenden Frist zugeordnet. Es existieren die beiden Varianten: nicht-unterbrechend und unterbrechend.

nicht-unterbrechend

Eine Prozessorzuordnung bleibt bis der Prozess eine blockierende Systemfunktion aufruft oder freiwillig die CPU abgibt. Neue eintreffende Prozesse mit kürzeren Fristen werden erst beim nächsten Scheduling berücksichtigt.

unterbrechend

Diese Variante führt einen Kontextwechsel durch, wenn ein Prozess mit einer kürzeren Frist rechenwillig wird.

Rate-Monotonic Scheduling

Rate-Monotonic Scheduling (RMS) ist für periodische Prozesse; RMS ordnet Prioritäten in Abhängigkeit von der Periode zu.

1) Prozesse mit der höchsten Frequenz (kleinste Periode) erhalten die höchste Priorität.

2) Prozesse mit der geringsten Frequenz (längste Periode) erhalten die niedrigste Priorität.

Auswahl der Prozesse anhand ihrer Priorität.

hochfrequente Prozesse werden minimal verzögert.

Zerstückelung niederfrequenter Prozesse, da sie häufig wegen hochfrequenter Prozesse unterbrochen werden.

Echtzeit Scheduling

Prozesse in Multimedia-Anwendungen führen oft zeitkritische Operationen aus. Bzgl. des Scheduling existieren zwei gegensätzliche Ziele:

a) ein ***unkritischer Prozess*** sollte nicht dauerhaft blockiert werden, weil zeitkritische Prozesse eine Ressource gänzlich auslasten.

b) ***zeitkritische Prozesse*** dürfen nicht durch Scheduling-Verfahren (Round-Robin oder Prioritäten) am zeitkritischen Fortschritt gehindert werden ⇒ Einhaltung von Zeitvorgaben.

Zuordnung von Kenngrößen zu zeitkritischen Prozessen

Bereitzeit (ready time): frühestmöglicher Ausführungsbeginn einer Aktivität.

Frist (deadline): spätester Zeitpunkt für die Beendigung einer Aktivität.

Ausführungszeit: worst-case Abschätzung für das zur vollständigen Ausführung einer Aktivität notwendige Zeitintervall.

Earliest Deadline First (EDF)

Rate-Monotonic Scheduling

Die optimale Ausnutzung und Auslastung aller Geräte eines Rechensystems legt Mehrprogrammbetrieb nahe. Zerlegung der Ausführungsphasen eines Programms in viele Einzelteile;

- Aufbrechen der Ausführung eines Prozesses in mehrere Phasen: u.a. Rechnen, E/A, Synchronisieren mit Partner.
- Die Ausführung der Programme wird in der Regel mehrfach unterbrochen.

Motivation

Unterbrechungsarten

Behandlung externer Unterbrechungen

Konflikte

Ursachen für Unterbrechungen

zugeteilte Prozessorzeit ist aufgebraucht;

benötigte Ressourcen stehen aktuell nicht zur Verfügung;

ein E/A-Gerät meldet sich zurück;

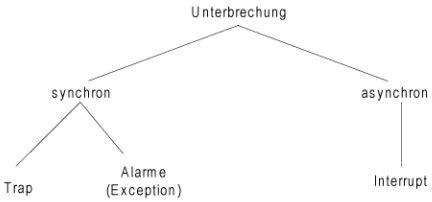
ein Fehler tritt auf, z.B. Division durch 0;

Systemaufruf (wurde bereits als spezielle Unterbrechung eingeführt);

Bei einer Unterbrechung wird ein gerade aktiver Prozess unterbrochen und eine ***Unterbrechungsbehandlung*** durchgeführt.

Es ist erforderlich, bei der Unterbrechung den CPU Status des gerade aktiven Prozesses für die spätere Fortsetzung zu speichern.

Forderung: Eine Unterbrechung muss so ***kontrolliert*** erfolgen, dass ein definierter Prozessstatus festgehalten werden kann.

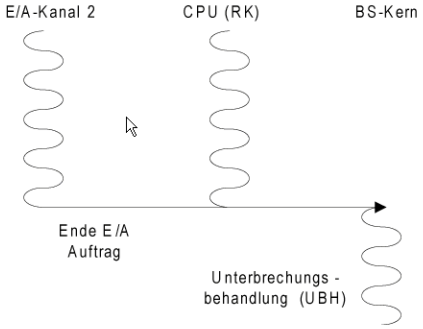


Externe, asynchrone Unterbrechungen

Interne, synchrone Unterbrechungen

Dies sind Unterbrechungen (Interrupts), die von außerhalb des zu unterbrechenden Prozesses ausgelöst werden, z.B. E/A-Kanal-"Endmeldung".

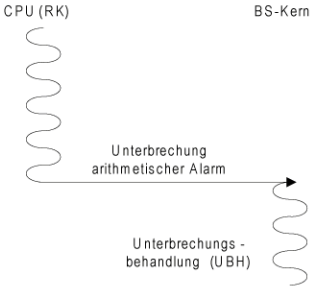
Der Ablauf im Rechnerkern (RK) wird unterbrochen und eine Unterbrechungsanfangsbehandlung des Betriebsystemkerns wird aktiviert.



Interne, synchrone Unterbrechungen

Dies sind Unterbrechungen (Alarme, Exceptions), die durch den zu unterbrechenden Prozess selbst ausgelöst werden, z.B. Division durch 0.

Der Ablauf im RK wird unterbrochen und eine Unterbrechungsanfangsbehandlung des Systemkerns wird aktiv.



Beispiele von internen Unterbrechungen

Speicherschutzalarm: Prozess greift auf Speicherbereich zu, der nicht vorhanden ist oder auf den er nicht zugreifen darf.

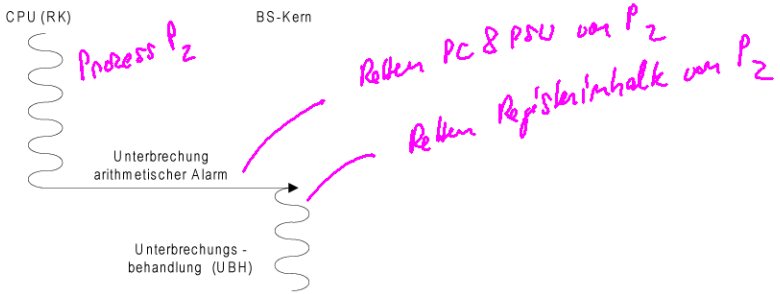
Befehlsalarm: dem Operationscode des Maschinenbefehls ist keine Operation zugeordnet.

Seitefehltalarm: Seite der virtuellen Adresse ist nicht im Arbeitsspeicher.

arithm. Alarm: arithm. Operation kann nicht ausgeführt werden, z.B. Division durch 0.

Interne, synchrone Unterbrechungen

Der Ablauf im RK wird unterbrochen und eine Unterbrechungsanfangsbehandlung des Systemkerns wird aktiv.



Beispiele von internen Unterbrechungen

Speicherschutzalarm: Prozess greift auf Speicherbereich zu, der nicht vorhanden ist oder auf den er nicht zugreifen darf.

Befehlsalarm: dem Operationscode des Maschinenbefehls ist keine Operation zugeordnet.

Seitefehltalarm: Seite der virtuellen Adresse ist nicht im Arbeitsspeicher.

arithm. Alarm: arithm. Operation kann nicht ausgeführt werden, z.B. Division durch 0.

Systemaufruf: kontrollierter Übergang in das Betriebssystem.

Ablauf

Geräte-Controller meldet Unterbrechung über spezielle Interrupt-Leitung an CPU.

CPU prüft im Befehlszyklus nach jeder Befehlsausführung, ob eine Unterbrechung gemeldet wurde.

Falls Unterbrechung vorliegt: sichern u.a. des aktuellen Befehlszählers, des Programmstatusworts und Sprung zu einer Unterbrechunganfangsbehandlung, die an festgelegter Speicheradresse steht.

Routine untersucht Unterbrechungsursache, die vom Controller über Datenleitung gemeldet wird (Unterbrechungsnummer).

über Unterbrechungsnummer erfolgt die Auswahl der benötigten Unterbrechungsbehandlungsroutine; Nummer ist i.a. Index in eine Tabelle, dem ***Unterbrechungsvektor*** .

Vektor enthält Speicheradresse der Unterbrechungsbehandlungsroutine.

Konflikte bei Unterbrechungen treten z.B. in folgenden Situationen auf:

(1) während einer Unterbrechungsbehandlung treten weitere Unterbrechungen auf;

(2) es treffen gleichzeitig mehrere Unterbrechungswünsche ein.

Beispiel

Mögliche Konfliktlösungen

Integration der Unterbrechungsbehandlung in den Befehlszyklus der CPU

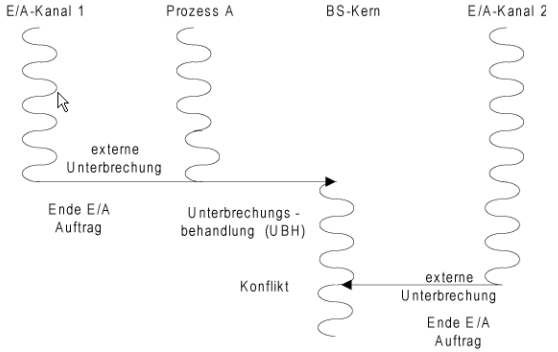
prüfen ob interne Unterbrechung aufgetreten,

falls ja, Behandlung der Unterbrechung

sonst : prüfen ob externe Unterbrechung mit höherer Priorität. Wenn ja wähle eine mit höchster Priorität.

Bei Unterbrechung: sichere alten Zustand, stelle neuen Zustand her und führe ersten Befehl der Unterbrechungsbehandlungsroutine aus.

E/A-Kanal 1 und E/A-Kanal 2 erledigen beide Aufträge für Prozess A.



Andere Unterbrechungen nicht zulassen, d.h. ***Maskierung*** von Unterbrechungen; anstehende Unterbrechung ignorieren oder vorläufig zurückstellen; Problem: u.a. Rechtzeitigkeit der Unterbrechungsbehandlung.

Interne Unterbrechungen erfolgen stets sofort und geben der zugehörigen Unterbrechungsbehandlung dieselbe Priorität, wie sie der unterbrochene Ablauf hatte.

Externe Unterbrechungen erhalten ***Prioritäten*** z.B. (0,...,31) zugeordnet. Die aufgerufene Unterbrechungsbehandlung erhält die Priorität (Ablaufpriorität) der auslösenden Unterbrechung.

Behandlung neu eintreffender externer Unterbrechung, falls ***Priorität höher als Ablaufpriorität*** ; andernfalls zurückstellen.

Konflikte

Konflikte bei Unterbrechungen treten z.B. in folgenden Situationen auf:

(1) während einer Unterbrechungsbehandlung treten weitere Unterbrechungen auf;

(2) es treffen gleichzeitig mehrere Unterbrechungswünsche ein.

Beispiel

Mögliche Konfliktlösungen

Integration der Unterbrechungsbehandlung in den Befehlszyklus der CPU

prüfen ob interne Unterbrechung aufgetreten,

falls ja, Behandlung der Unterbrechung

sonst : prüfen ob externe Unterbrechung mit höherer Priorität. Wenn ja wähle eine mit höchster Priorität.

Bei Unterbrechung: sichere alten Zustand, stelle neuen Zustand her und führe ersten Befehl der Unterbrechungsbehandlungsroutine aus.

- Prof. J. Schlichter
  - Lehrstuhl für Angewandte Informatik / Kooperative Systeme, Fakultät für Informatik, TU München
  - Boltzmannstr. 3, 85748 Garching

    Email: schlichter@in.tum.de

    Tel.: 089-289 18654

    URL: http://www11.informatik.tu-muenchen.de/

Fragestellungen

Dieser Abschnitt beschäftigt sich mit den Adressräumen für Programme und deren Abbildung auf den physischen Arbeitsspeicher einer Rechenanlage:

Programmadressraum vs. Maschinenadressraum.

Direkte Adressierung, Basisadressierung.

Virtualisierung des Speichers; virtuelle Adressierung, insbesondere Seitenadressierung.

Speicherabbildungen

Dieser Abschnitt behandelt einige Mechanismen zur Abbildung von Programmadressen auf Maschinenadressen des Arbeitsspeichers.

Einführung

Die unmittelbare Nutzung des physischen Adressraums (Arbeitsspeichers) bei der Anwendungsentwicklung ist nicht empfehlenswert. Probleme sind folgende:

- Kenntnisse über Struktur und Zusammensetzung des Arbeitsspeichers notwendig.
- Kapazitätsengpässe bei der Arbeitsspeichergröße.

⇒ deshalb Programmerstellung unabhängig von realer Speichergröße und -eigenschaften.

Maschinenadressraum

Arbeitsspeicher = Folge von fortlaufend nummerierten Bytes (ab 0). ***Maschinenadresse*** = Nummer des Bytes.

Programmadressraum

Programmadressen = Adressen im Programm (z.B. für Variable).

Die Menge der zulässigen Programmadressen ist der ***Programmadressraum*** . Dieser ist prozessspezifisch, d.h. Programmadressen haben nur Programm-lokale Bedeutung z.B. als Sprungziele.

Speicherabbildung

Abbildung der Programmadressen auf Maschinenadressen (durch CPU).

direkte Adressierung,

Basisadressierung,

Seitenadressierung und

Segment-Seitenadressierung.