

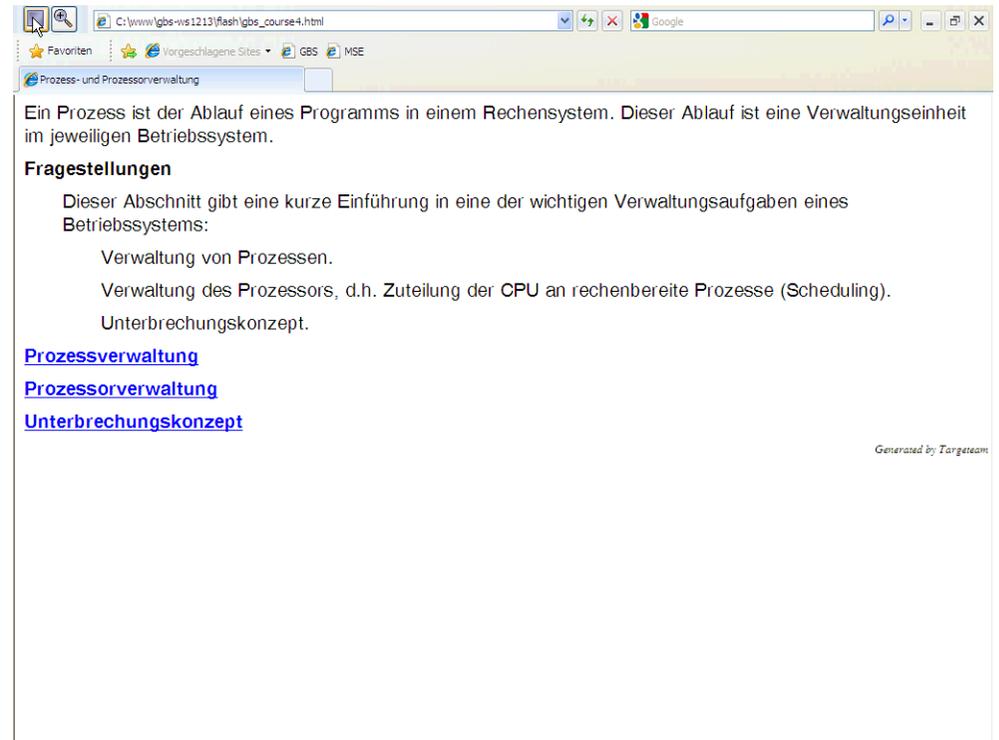
Script generated by TTT

Title: Grundlagen_Betriebssysteme (21.11.2012)

Date: Wed Nov 21 13:15:25 CET 2012

Duration: 45:09 min

Pages: 7



Ein Prozess ist der Ablauf eines Programms in einem Rechner. Dieser Ablauf ist eine Verwaltungseinheit im jeweiligen Betriebssystem.

Fragestellungen

Dieser Abschnitt gibt eine kurze Einführung in eine der wichtigen Verwaltungsaufgaben eines Betriebssystems:

- Verwaltung von Prozessen.
- Verwaltung des Prozessors, d.h. Zuteilung der CPU an rechenbereite Prozesse (Scheduling).
- Unterbrechungskonzept.

[Prozessverwaltung](#)

[Prozessorverwaltung](#)

[Unterbrechungskonzept](#)

Generated by Targeteam



Dieser Abschnitt behandelt das Prozesskonzept, Datenstrukturen zur Beschreibung des aktuellen Prozesszustandes sowie Dienste zum Aufruf von Systemfunktionen.

Prozesse repräsentieren eine Aktivität; sie haben ein Programm, Eingaben, Ausgaben und einen Zustand.

[Prozesskonzept](#)

[Dispatcher](#)

[Arbeitsmodi](#)

[Systemaufrufe](#)

[Realisierung von Threads](#)

Generated by Targeteam



Dienste der Prozessverwaltung

Die Prozesse werden durch das Betriebssystem verwaltet.

Auslösende Ereignisse für die Erzeugung eines Prozesses

Initialisierung des Systems.

Systemaufruf zum Erzeugen eines Prozesses durch einen anderen Prozess.

Benutzeranforderung zum Starten eines neuen Prozesses (Start einer Applikation).

Auslösung eines Stapelauftrags (Batch Job).

Formen der Terminierung von Prozessen

Normale Beendigung (freiwillig).

Vorzeitige Beendigung bei einem durch den Prozess selbst erkannten Fehler (freiwillig).

Vorzeitige Beendigung bei einem katastrophalen Fehler, erkannt durch das BS (unfreiwillig).

Terminierung durch einen anderen Prozess (unfreiwillig).

Prozess-Auswahl, Strategien zur Prozessorzuteilung: **Scheduling**.

Prozessor-Anbindung; **Dispatching**.

Generated by Targeteam



Prozesskontrollblock



Jeder Prozess muss als eine **Verwaltungseinheit** beschrieben sein. Ein Prozess wird durch seinen Prozess-Kontext und dieser durch den Prozesskontrollblock (PCB) beschrieben. Ein **PCB** (process control block) enthält i.d.R. folgende Informationen:

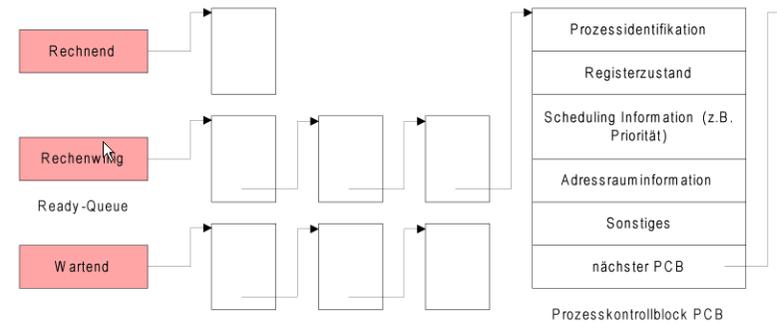
- eindeutiger Name, z.B. fortlaufende Nummerierung des Prozesses (z.B. pid in Unix)
- Name des Benutzers, dem der Prozess zugeordnet ist
- der momentane Prozesszustand (wartend, rechnend, rechenwillig, ...)
- falls der Prozess rechnend ist, Angabe der zugeordneten CPU
- falls der Prozess wartend ist, eine Spezifikation des Ereignisses, auf das der Prozess wartet (z.B. Adresse eines Semaphors).
- die Ablaufpriorität des Prozesses
- die Inhalte der programmierbaren Register (die Anzahl ist abhängig von der jeweiligen CPU-Architektur), z.B. Kellerpointer.
- die Inhalte der Register, in denen die Anfangsadresse und Länge der prozessspezifischen Speicherabbildungstabellen enthalten sind (virtuelle Adressierung).
- das Programmstatuswort (PSW). Das PSW enthält weitere Informationen, die die CPU über den Prozess kennt, z.B. Ablaufcodes, Condition Codes
- PCB unter Linux ist durch die Struktur `task_struct` spezifiziert; definiert unter `include/linux/sched.h`.

Generated by Targeteam



Die Prozesse werden in Zustandslisten verwaltet, die als verkettete Liste der PCBs realisiert sind.

für E/A-Geräte (z.B. Platte, Terminal) existiert i.d.R. jeweils eine eigene Warteschlange, die die PCBs der wartenden Prozesse enthält.



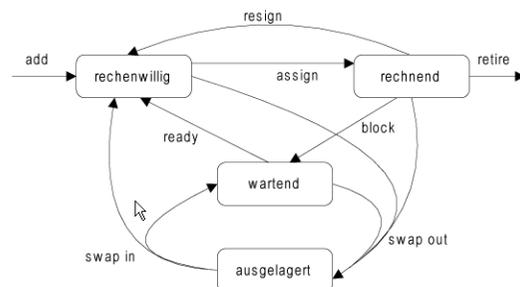
Generated by Targeteam



Zustandsmodell



Das Prozess-Zustandsmodell unterscheidet neben den bereits vorgestellten Zuständen **rechenwillig**, **rechnend**, **wartend** auch den Zustand **ausgelagert**. Letzterer Zustand tritt ein, wenn der Adressraum aufgrund Speichermangels aus dem Arbeitsspeicher auf den Hintergrundspeicher verlagert wird ("swapping").



Zustandsübergänge sind:

- add**: ein neu erzeugter Prozess wird zu der Menge der rechenwilligen Prozesse hinzugefügt;
- assign**: als Folge eines Kontextwechsels wird dem Prozess die CPU zugeordnet;
- block**: aufgrund eines EA-Aufrufs oder einer Synchronisationsoperation wird der Prozess wartend gesetzt;
- ready**: nach Beendigung der angestoßenen Operation wechselt der Prozess in den Zustand rechenwillig; er bewirbt sich erneut um die CPU;
- resign**: dem rechnenden Prozess wird die CPU entzogen; er bewirbt sich anschließend erneut um die CPU;
- retire**: der aktuell rechnende Prozess terminiert;
- swap out**: der Prozess wird auf die Festplatte ausgelagert;



Prozesse erzeugen andere Prozesse: parent und child. Vater

kann auf Beendigung von Kind warten, oder parallel weiterlaufen.

Prozesserzeugung: 2 Vorgehensweisen

Windows NT: Vaterprozess veranlasst eine Reihe von Systemaufrufen, die das Kind entstehen lassen.

Unix: Vater kloniert sich mit Systemaufruf `fork()`; Kind ändert selbst seine Aufgabe.

Unix Prozesserzeugung

Linux unterstützt den Systemaufruf `clone()` zur Erzeugung neuer Thread-Kopien.

Generated by Targeteam