

Prozesse

- Betriebssystem-Abstraktionen zur Ausführung von Programmen
- Eigentümer von Ressourcen
- differenzierte Prozessmodelle: definieren konkrete Prozesseigenschaften

Prozessmanagement

- Komponente eines Betriebssystems, die Prozessmodell dieses Betriebssystems implementiert
- wichtige Aufgaben: Prozesserschöpfung u. -beendigung (u. Scheduling)
- wichtigste Datenstrukturen: Prozessdeskriptor, Prozessdeskriptortabelle

Prozessdeskriptor (PCB) Buchführung über sämtliche zum Management eines Prozesses notwendigen Informationen

- Prozessidentifikation
- Rechtemanagement
- Speichermanagement
- Prozessormanagement
- Kommunikationsmanagement

Prozessdeskriptortabelle enthält: Prozessdeskriptoren aller momentan existierenden Prozesse

Threads

- Betriebssystem-Abstraktionen für sequentielle, nebenläufige Aktivitäten
- sind Gegenstand des Scheduling

Multithread-Prozessmodell

- vollständige Beschreibung einer ablaufenden Aktivität
- Dazu gehören insbesondere
 1. das ablaufende Programm
 2. zugeordnete Betriebsmittel (Prozessor/Speicher/Kommunikation)
 3. Rechte
 4. prozessinterne parallele Aktivitäten (Threads) und deren Bearbeitungszustände

Threaddeskriptor (TCB)

- ein TCB enthält lediglich:
 1. Threadzustand (aktiv, bereit, blockiert, ...)
 2. Ablaufkontext, falls nicht aktiv (Programmzähler, Stackpointer, Prozessorregister)
- enthält nicht: Beschreibung der Ressourcen (Speicherlayout, Rechte)

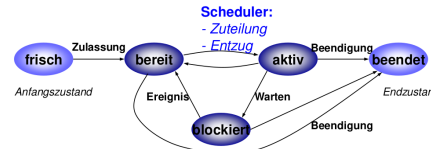
Thread-Typen

- Kernel Level Threads (KLTs): Kenntnis über Threads: hat Betriebssystem, genauer: der Betriebssystem-Kern(el)
- User Level Threads (ULTs): Kenntnis über Threads: existiert nur auf Benutzer-Ebene (user level)
- das Betriebssystem [genauer: der Betriebssystem-Kern(el)] weiß nicht, dass Threads existieren

Scheduling: Grundaufgabe Entscheidung: Welche Threads erhalten wann u. wie lange einen Prozessor/Prozessorkern zugeteilt?

Zustandsmodelle

- Threads können verschiedene Zustände annehmen (Beispiel 3/5-Zustandsmodell)
- bereit, aktiv, blockiert, frisch, beendet



Scheduling: Notwendigkeit u. Sinn

- im Allgemeinen: Anzahl der Aktivitäten \gg Anzahl der Prozessoren
- nicht alle können gleichzeitig arbeiten
- eine Auswahl muss getroffen werden
- Auswahlstrategie: Schedulingstrategie, algorithmus
- die Betriebssystemkomponente: Scheduler

Scheduling-Strategien

- abhängig vom Einsatzfeld eines Betriebssystems
 - Echtzeitsysteme: Einhaltung von Fristen
 - interaktive Systeme: Reaktivität
- wichtige Strategien:
 - FCFS
 - SRTN
 - Round Robin (ohne u. mit Prioritäten)
 - EDF
 - ratenmonotones Scheduling

Privilegierungsebenen

- sind typischerweise "kernel mode" und "user mode"
- steuern Rechte ...
 - zur Ausführung privilegierter Prozessorinstruktionen
 - zur Konfiguration des Arbeitsspeicherlayouts
 - zum Zugriff auf Arbeitsspeicherbereiche
 - zum Zugriff auf E/A-Geräte
- ermöglichen: Durchsetzung von Regeln: "Nur ein im 'kernel mode' ablaufender Prozess hat Zugriff auf ..."

Kommunikation u. Synchronisation

- Austausch von Daten zwischen Prozessen = Kommunikation (Inter-Prozess-Kommunikation, IPC)
- Abweichende Geschwindigkeiten von Sender und Empfänger: behandelt durch Synchronisation

kritischer Abschnitt und wechselseitiger Ausschluss

- in einem kritischen Abschnitt darf sich stets nur ein Thread aufhalten
- notwendig: wechselseitiger (gegenseitiger) Ausschluss
- realisiert durch Entry- u. Exit-Code [z.B. die Semaphore-Operationen belegen (P) und freigeben (V)]

Mechanismen zur Synchronisation

- Semaphore (binäre Semaphore u. mehrwertige Semaphore) (ansehen!)
- (Hoar'sche) Monitore

Mechanismen zur Kommunikation

- Shared Memory (gemeinsamer Speicher)
- Botschaften
- Fernaufrufe
- Systemaufrufe

Notwendigkeit des Ereignismanagement

- in Betriebssystemen laufen sehr viele Aktivitäten parallel ab
- dabei entstehen immer wieder Situationen, in denen auf unterschiedlichste Ereignisse reagiert werden muss, z.B.
 - Timerablauf
 - Benutzereingaben (Maus, Tastatur)
 - Eintreffen von Daten von Netzwerken, Festplatten, ...
 - Einlegen/stecken von Datenträgern
 - Aufruf von Systemdiensten
 - Fehlersituationen

Umgangsformen mit Ereignissen

- "busy waiting"
- "periodic testing"
- Unterbrechungen (Interrupts)

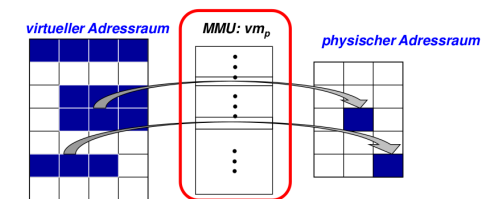
Programmiermodelle zum Umgang mit Interrupts

- Prozeduren (→ inline Prozeduraufrufmodell)
- IPC-Operationen (→ IPC-Modell)
- Threads (→ pop-up Thread Modell)

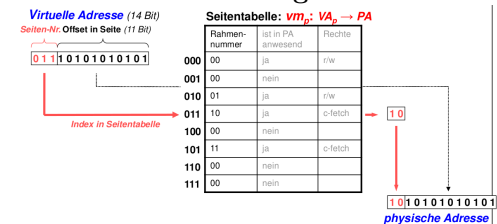
Interrupts auf Anwendungsebene

- notwendig: Event Service Routines (ESRs)
- Beispiel: UNIX/Linux-Signalbehandlung

Virtuelle Prozessadressräume und physischer Adressraum, Abbildungen



Seitenabbildungstabellen



Seitentableneinträge

Rahmennummer im Arbeitsspeicher (PA)	anwesend	benutzt	verändern	Schutz	Caching
--------------------------------------	----------	---------	-----------	--------	---------

- anwesend: Indikator, ob Seite im Arbeitsspeicher liegt ("presentBit")

- benutzt: Indikator, ob auf die Seite zugegriffen wurde (used/referencedBit)
- verändert: Indikator, ob Seite schmutzig ist ("dirty/modifiedBit")
- Schutz: erlaubte Zugriffsart je Privilegierungsebene (access control list")
- Caching: Indikator, ob Inhalt der Seite gecached werden darf

Seitenaustauschalgorithmen

- Optimale Strategie: Auslagerung derjenigen Arbeitsspeicherseite, deren ...
 - nächster Gebrauch am weitesten in der Zukunft liegt ("Hellseher")
 - Auslagerung nichts kostet (man müsste man wissen, ob eine Seite seit ihrer Einlagerung verändert wurde)
- im Folgenden: einige Algorithmen, die sich diesem Optimum annähern:
 - First-In, First-Out (FIFO)
 - Second-Chance
 - Least Recently Used (LRU)
 - Working Set / WSClock

i-Node i-Node: Metainformationen über genau eine Datei



Verzeichnis Verzeichnis = Menge von Paaren (Name, i-Node-Index)

Superblock

- Superblock: Einstiegspunkt eines Dateisystems
- enthält: Schlüsselparameter
 - Name des Dateisystems
 - Typ (NTFS, Ext *, HFS, ...) → Layout der Metadaten
 - Größe und Anzahl Sektoren
 - Ort und Größe der i-Node-Tabelle
 - Ort und Größe der Freiliste
 - i-Node-Nummer des Wurzelverzeichnisses

Hardware-Prinzipien

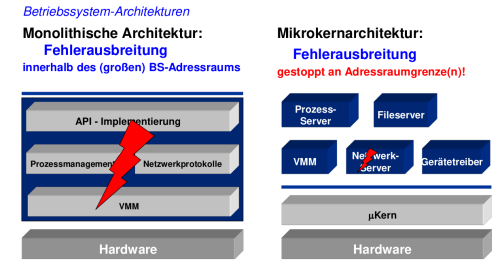
- Controller-Register
 - in E/A-Adressräumen
 - im Arbeitsspeicher (Memory Mapped E/A)
 - Isolation, Robustheit, Sicherheit

- Interruptsystem: asynchrone Benachrichtigungen

Software-Prinzipien Gerätemanager (Treiber)

- Auftragsmanagement
- ISRs

Betriebssystem-Architekturen



Sicherheit

- Schutz des Betriebssystems
- z.B. gegen Angriffe von außen

SELinux-Ansatz

- Sicherheitspolitiken in SELinux
- neue Betriebssystem-Abstraktion
 - absolute Kontrolle über kritische Funktionen des Betriebssystems
 - spezifiziert durch Regelmenge
 - implementiert durch die SELinux-Sicherheitsarchitektur

Robustheit

- Ziel: Tolerierung unvorhergesehener Fehler und Ausfälle
- Mikrokernelarchitekturen: robuster als Makrokernelarchitekturen
- Gründe:
 - Fehlerisolation
 - Möglichkeiten zur Fehlerbehebung (z.B. Micro-Reboot)



