Threads

Thread oder **Leichtgewichtigen Prozess** ist Teil eines Prozesses, der einen unabhängigen Kontrollfluss repräsentiert.

• Ein **Prozess** kann aus **mehreren Threads** bestehen, somit mehrere voneinander unabhängige, **nebenläufige Kontrollflüsse**

BEISPIEL

- Ein Textverarbeitungsprogramm wird gestartet. Somit existiert auf dem Computersystem ein Textverarbeitungs-Prozess.
 - Dieser Prozess startet intern mehrere Threads, die jeweils bestimmte Aufgaben übernehmen:
 - Thread 1: realisiert den Texteditor. Er reagiert also auf Eingaben des Users mit Tastatur oder Maus.
 - Thread 2: realisiert eine "Alle 10 Minuten automatisch im Hintergrund speichern"-
 - Thread 3: realisiert die automatische Rechtschreibprüfung in kleinen zeitlichen Abständen
- Das Thread-Konzept wird von Betriebssystemen unterstützt und können deshalb auf Kernel-Ebene realisiert sein
- Unterstützt Betriebssystem keine Threads, so kann auf der **User-Ebene** ein spezielles Programm Bereitstellung übernehmen
- Wird die Java-Laufzeitumgebung auf einem Betriebssystem installiert, laufen Threads auf beiden Ebenen
- Bei Systemen ohne jegliche Thread-Unterstützung ist ein Prozess praktisch gleichzusetzen mit einem Thread.

Wenn ein Prozess nur einen Thread besitzt, ist dieser betrachtete Prozess praktisch gleichzusetzen mit seinem Thread.

Bei **Thread-Kontextwechsel** auf der CPU wird aktiver Thread A durch einen anderen Thread B ersetzt.

Vorteile Nachteil

- Thread-Kontextwechsel leichter als Prozess-Kontextwechsel.
- Threads eines Prozesses haben Zugriff auf alle Betriebsmittel, welche Prozess zugeordnet sind.
- Der Anwendungsprogrammierer kann die Funktionalität der Gesamtanwendung in unterschiedliche Threads aufteilen, welche jeder für sich einfacher zu implementieren ist.
- Threads eines Prozesses haben Zugriff auf Betriebsmittel, welche Prozess zugeordnet sind.
- Anwendungsprogrammierer müssen über Kenntnisse bei der Programmierung von Threads verfügen.

Revision #1 Created 24 September 2022 16:21:45 by Merith Holtmann Updated 2 October 2022 19:22:32 by Merith Holtmann