

1.3.2 Client/Server-Modell

Jede Betriebssystemfunktion stellt einen bestimmten Dienst wie z.B. das Lesen einer Datei zur Verfügung. Der Serverteil dieses Dienstes erledigt den Hardwarezugriff und muss deshalb im Kernel Mode laufen. Er befindet sich also im Kernel. Im Clientteil befinden sich alle Funktionen, die im User Mode laufen können und nicht die erweiterten Rechte des Kernels brauchen. Der Clientteil bildet die Schnittstelle zu den Applikationen und ruft über einen Systemaufruf den Serverteil auf.

Die Interaktion zwischen Anwendungsprogramm und Betriebssystem spielt sich in groben Zügen wie folgt ab: Das Betriebssystem erhält eine Anfrage einer Anwendung (z. B. für das Lesen einer Datei). Der Clientteil empfängt und bearbeitet die Anfrage und verlangt mit einem Systemaufruf die entsprechenden Daten auf der Festplatte vom Serverteil.

Hinweis

▷ Im hier erläuterten Zusammenhang liegen Client und Server auf dem lokalen Computer und sind nicht mit dem Client/Server-Modell für Netzwerkdienste zu verwechseln.

1.4 Merkmale

Nachfolgend werden wichtige Eigenschaften erläutert, die moderne Betriebssysteme aufweisen.

1.4.1 Multiuser-Betrieb

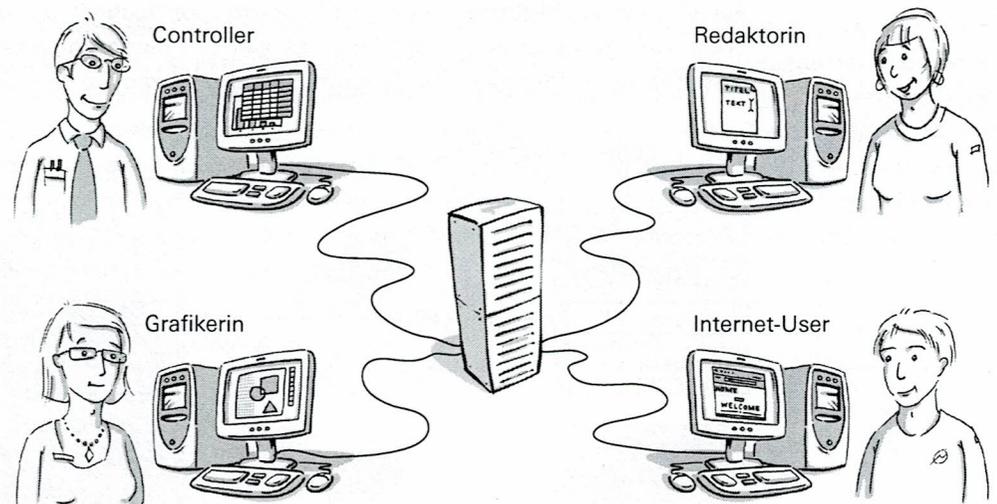
Bezüglich der Systemnutzung kann zwischen **Einbenutzerbetrieb**^[1] und **Mehrbenutzerbetrieb**^[2] unterschieden werden. Ein Computer kann also entweder nur von einem einzelnen Benutzer **gleichzeitig** verwendet werden, oder es steht ein Rechner zur Verfügung, auf den mehrere Benutzer gleichzeitig Zugriff haben. Ein System, das den Multiuser-Betrieb beherrscht, kann flexibler und vielfältiger eingesetzt werden und bietet durch die Rechteverwaltung von verschiedenen Benutzern mehr Sicherheit. Es ist jedoch auch komplexer aufgebaut und braucht mehr Ressourcen.

Der Mehrbenutzerbetrieb ist die Technologie der klassischen Mainframes wie z. B. IBM AS-400™ oder DEC-VAX™. Diese Maschinen werden von Unix-ähnlichen Betriebssystemen gesteuert. Grundsätzlich unterstützen heute aber auch alle gängigen Betriebssysteme für PCs den Multiuser-Betrieb. **Linux** hat dieses Konzept auf den PC übertragen, sodass auf einer Maschine gleichzeitig mehrere Benutzer angemeldet sein können. So kann sich auch eine einzelne Person als verschiedene Benutzer, d. h. in verschiedenen Rollen anmelden, z. B. um als Administrator ein Programm zu installieren. **Windows** bietet ebenfalls einen Multiuser-Betrieb, allerdings sind dessen Möglichkeiten noch nicht so ausgereift und werden weniger konsequent eingesetzt als bei Linux/Unix.

[1] Engl. Fachbegriff: Single User Mode.

[2] Engl. Fachbegriff: Multi User Mode.

[1-2] Multiuser-Betrieb



1.4.2 Multiprozessor-Betrieb

Die meisten PCs, die im geschäftlichen oder privaten Bereich zum Einsatz kommen, sind mit einem einzigen Prozessor ausgestattet, der die Arbeit für alle anfallenden Aufgaben übernimmt. Unter den Servern und Workstations sind aber auch Maschinen mit mehreren Prozessoren anzutreffen. Diese arbeiten im Multiprozessor-Betrieb und sind deutlich teurer als «Single-Prozessor-Systeme». Dafür können sich die Prozessoren die anfallende Arbeit aufteilen. Wie gut diese Arbeitsteilung funktioniert, hängt allerdings stark vom jeweiligen Betriebssystem und den eingesetzten Anwendungsprogrammen ab. Einerseits bringt der Multiprozessor-Betrieb also mehr Rechenleistung, andererseits ist die Rechenleistung nicht direkt proportional zur Anzahl Prozessoren. Mit anderen Worten: Zwei Prozessoren leisten nicht zweimal mehr als ein einziger Prozessor, drei Prozessoren leisten nicht dreimal mehr als ein einziger Prozessor, etc.

1.4.3 Multitasking-Betrieb

Multitasking bedeutet, dass mehrere Aufgaben (Tasks) bzw. Prozesse (auch unterschiedlicher Programme) gleichzeitig abgearbeitet werden. So kann beispielsweise ein Anwender in einem Textverarbeitungsprogramm Dokumente bearbeiten, während eine Datenbank im Hintergrund Daten sortiert. Der Vorteil ist offensichtlich, dafür wird ein gerechter Zeitzuteilungsmechanismus für die verschiedenen Prozesse benötigt und eine umfangreiche Verwaltung, damit die Prozesse nach einem Unterbruch problemlos weiterlaufen können. Schliesslich soll eine Aufgabe genau an der Stelle fortgesetzt werden können, an der sie unterbrochen wurde. Die dazu benötigten Informationen werden vom Betriebssystem im so genannten **Process Control Block (PCB)** gespeichert.

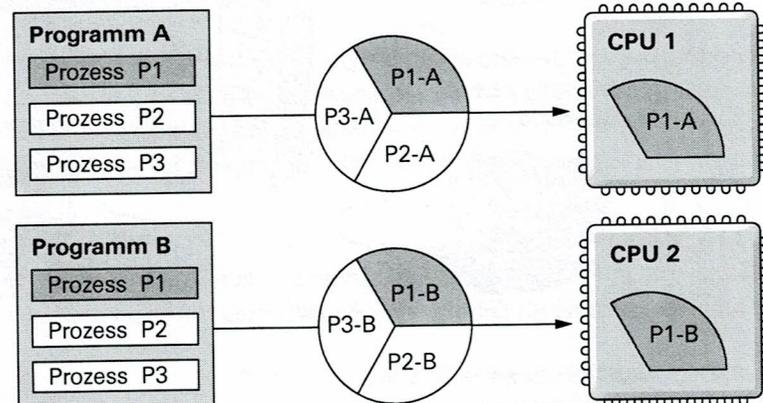
Beispiele von Daten, die für jeden Prozess im PCB gespeichert werden:

- Prozessorregister
- Offene Dateien
- Aktuelles Verzeichnis
- Status der Peripheriegeräte

Echtes Multitasking

Für ein **echtes Multitasking** sind Multiprozessor-Rechner mit **mehreren Prozessoren** nötig. Das Betriebssystem legt fest, wie viel Zeit einem Prozess zur Verfügung gestellt wird. Die Anwendungen bzw. Programme haben darauf keinen Einfluss.

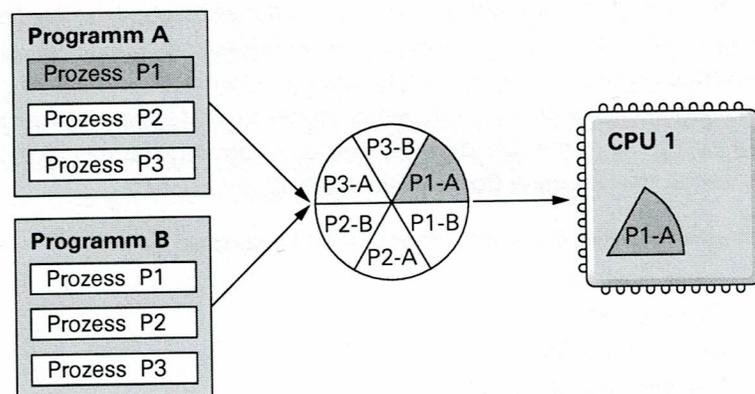
[1-3] Echtes Multitasking



Unechtes Multitasking

Beim **unechten Multitasking** werden die einzelnen Prozesse zeitlich versetzt auf **einem Prozessor** ausgeführt. Alle modernen Einprozessor-Systeme laufen im unechten Multitasking-Betrieb. Die Prozessorzeit wird hierbei in mehrere Zeitabschnitte (auch «Zeitscheiben») eingeteilt. Ist ein Zeitabschnitt abgelaufen, entzieht das Betriebssystem dem aktuellen Prozess den Zugriff auf den Prozessor und gibt ihn an den nächsten Prozess weiter. Dieses Verfahren wird **preemptives Multitasking** genannt und von allen modernen Betriebssystemen eingesetzt. Es verhindert, dass ein Programm selbstständig bestimmt, wann es die Kontrolle über den Prozessor abgibt. Beim **kooperativen Multitasking** bestimmt der Prozess selber, wann er die Kontrolle abgibt.^[1] Dieses Verfahren kann allerdings dazu führen, dass ein schlecht programmiertes, «abgestürztes» oder «böses» Programm alle anderen Prozesse blockiert, weil es die Kontrolle nie übergibt. Deshalb wird es heute kaum mehr eingesetzt.

[1-4] Unechtes Multitasking

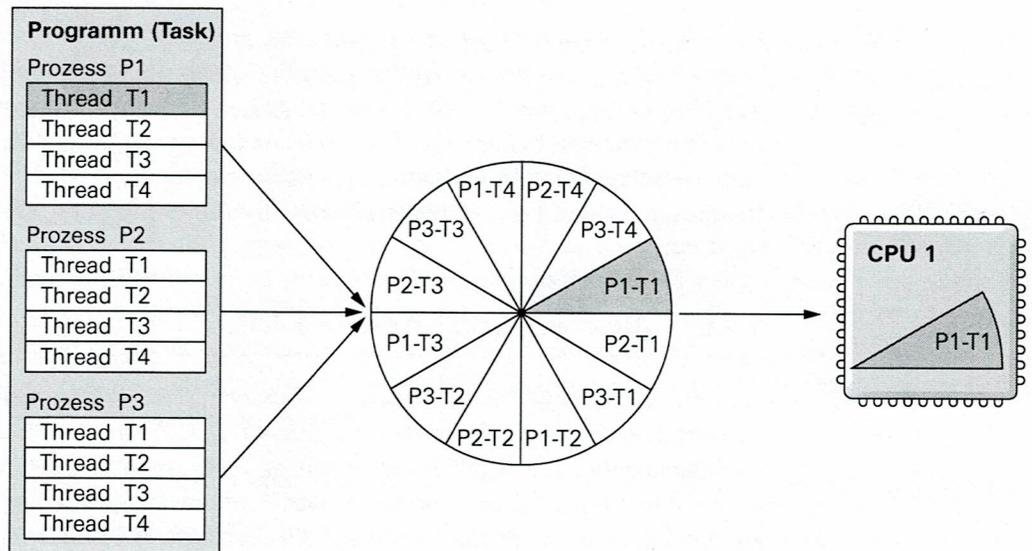


[1] Dieses Verfahren wird z. B. bei MS Windows 3.11 eingesetzt.

1.4.4 Multi-Threading

Multi-Threading ist eine Weiterentwicklung des Multitasking-Verfahrens. Dabei wird ein einzelner Prozess in mehrere simultan laufende Teilprozesse bzw. Threads unterteilt. Multi-Threading kann quasi als **Multitasking innerhalb eines Tasks** aufgefasst werden. Multi-Threading erlaubt beispielsweise, dass der Benutzer einen Text erfasst, während im Hintergrund die Silbentrennung und Rechtschreibprüfung laufen. Die Silbentrennung und Rechtschreibprüfung entsprechen also einzelnen «Prozessfäden» im Task der «Textverarbeitung». Gleichzeitig können weitere Tasks wie z. B. ein Download von MP3-Dateien oder eine Datenbankabfrage im Gang sein.

[1-5] Multi-Threading



Vorteile eines Multi-Threading-Betriebssystems:

- Threads sind so konzipiert, dass das Umschalten zwischen den einzelnen Threads viel schneller geht als zwischen autonom laufenden Prozessen, da weniger interner Verwaltungsaufwand entsteht.
- Auf einem Multiprozessor-System kann die Verarbeitungsgeschwindigkeit beträchtlich gesteigert werden, weil der Hauptprozess Teilaufgaben abgeben kann und so entlastet wird.

Nachteil eines Multi-Threading-Betriebssystems:

- Da Threads denselben Speicherbereich benutzen, können sie sich gegenseitig beeinflussen, stören oder gar zum Absturz bringen.

Die aktuellen Betriebssysteme arbeiten nach dem Multi-Threading-Verfahren. Damit dessen Vorteile aber effektiv zum Tragen kommen, muss Multi-Threading auch vom Anwendungsprogramm unterstützt werden. Auf diese Weise können Threads den Arbeitsspeicher gemeinsam nutzen, d. h. auf die gleichen Adressen zugreifen.