

6 Betriebssysteme

Ein *Betriebssystem* ist ein Programm, das dem Benutzer und den Anwendungsprogrammen grundlegende Dienste bereitstellt. Nutzer eines Betriebssystems sind nicht nur Programmierer, sondern auch Personen, die mit dem Rechner umgehen wollen, von dessen Funktionieren aber keine Ahnung haben. Für solche Benutzer präsentiert sich der Rechner über das Betriebssystem. Die Dienste, die es bereitstellt, sind das, was der Rechner in den Augen eines solchen Nutzers *kann*. Erst den grafischen Betriebssystemoberflächen ist es zu verdanken, dass heute jeder einen Rechner irgendwie bedienen kann und dass es auch leicht ist, mit einem bisher unbekannten Programm zu arbeiten, ohne vorher dicke Handbücher zu wälzen.

Genau genommen handelt es sich bei den *grafischen Oberflächen* um Betriebssystemaufsätze. Die elementaren Dienste des Betriebssystems werden, mehr oder minder geschickt, bildlich umgesetzt. Der Rechner präsentiert grafisch einen Schreibtisch (*desktop*), auf dem Akten und Ordner (Dateien und Verzeichnisse) herumliegen. Diese Akten können geöffnet und verändert (*edit*), kopiert (*copy*) oder in einen Papierkorb geworfen werden (*delete*). Man kann die Objekte des Schreibtisches anfassen, verschieben oder aus Ordnern neue Akten herausholen. Mit dieser *Desktop Metapher* kann eine Sekretärin auch ohne vorherige Computerkenntnisse nach kürzester Zeit umgehen.

Leider kam diese Idee des *Desktop* zehn Jahre zu früh. Sie entstand bei Rank Xerox. In demselben Zusammenhang wurde auch das objektorientierte Programmieren, insbesondere die Sprache *Smalltalk*, entwickelt. Das *Office System* von Xerox war Anfang der achtziger Jahre noch zu teuer, die Hardware noch etwas langsam, und die Vorstellung, dass jeder Benutzer seinen eigenen Bildschirm haben sollte, aufgrund der hohen Hardwarekosten noch illusorisch. Rechner schienen zu wichtigeren Zwecken da zu sein als zur Textverarbeitung und zur Büroorganisation. Das Konzept von Xerox wurde dennoch von der Firma *Apple* aufgegriffen. Nach einem anfänglichen Misserfolg mit dem System *Lisa* trat der *Macintosh* seit den frühen 80er Jahren seinen Erfolgzug an.

Unter den Handlungen, die man auf dem Desktop vollführt, können Programmierer einfache Betriebssystemdienste erkennen, die sie bis dato durch Kommandos aufgerufen hatten. Geübte Rechnerbenutzer ziehen oft die unterliegende Kommandosprache dem „Herumfuhrwerken“ mit der Maus vor. Die Kenntnis dieser Kommandosprache ist für sie ohnehin notwendig, weil der *Desktop* nur einen Bruchteil der Betriebssystemdienste widerspiegelt.

6.1 Basis-Software

Direkt nach dem Einschalten eines Rechners müssen bereits die ersten Programme gestartet werden. Diese Programme wollen wir als *Basis-Software* bezeichnen. Dabei handelt es sich noch nicht um Teile eines Betriebssystems. Vielmehr muss der Rechner zuerst initialisiert werden, seine Komponenten werden getestet und erst danach wird zu einem Programm, dem Ladeprogramm gesprungen, welche das *Betriebssystem* lädt.

Ein Teil dieser Basis-Software ist in Festwertspeichern, also in *ROMs* (*read only memory*), gespeichert. Sie wird immer dann aktiviert, wenn der Rechner eingeschaltet oder zurückgesetzt (*reset*) wird. Bei IBM-kompatiblen PCs wird die Grundsoftware *ROM-BIOS* genannt, wir haben das BIOS bereits in Kapitel 1 (S. 58 ff.) besprochen.

Die Basis-Software enthält neben Testroutinen auch grundlegende Programme, um Daten von Diskette, Festplatte oder CD-Laufwerk zu lesen, sowie einfache Ausgaberroutinen, um Meldungen auf dem Bildschirm auszugeben. Diese werden benötigt, denn beim „Hochfahren“ müssen Erfolgs- oder Fehlermeldungen, sowie der Fortschritt des Ladevorganges angezeigt werden. Sind keine Fehler aufgetreten, dann wird von der Festplatte (evtl. auch von einer CD oder einer Diskette) das Betriebssystem geladen. Es können durchaus mehrere Betriebssysteme zur Verfügung stehen. In diesem Fall erscheint ein *Prompt*, also eine Anforderung an den Benutzer, das gewünschte auszuwählen. Dieses wird nun geladen, initialisiert und gestartet.

Als erstes startet ein Betriebssystemkern mit einem so genannten *Kommandointerpreter*. Dies ist ein Programm, das lediglich einen Textbildschirm – z.B. 25 Zeilen zu je 80 Spalten – mit einem Eingabeprompt anzeigt. Hier kann der Benutzer Kommandos eingeben. Diese werden vom Betriebssystem ausgeführt, eventuelle Ausgaben werden auf dem Textbildschirm angezeigt, dann erscheint wieder der Prompt für das nächste Kommando. Die Art und Syntax der Kommandos unterscheidet sich je nach Betriebssystem, dennoch gibt es viele Gemeinsamkeiten. Insbesondere wird der Name jedes ausführbaren Programms als Kommando aufgefasst. Gibt man den Namen einer Datei ein, die ein ausführbares Programm enthält, z.B. `test.exe`, so wird dieses gestartet. Das hört sich einfach an, erfordert aber eine Reihe recht komplizierter Verwaltungstätigkeiten des Betriebssystems, mit denen wir uns in diesem Kapitel befassen wollen.

Die ersten einfachen Betriebssysteme, wie z.B. MS-DOS, konnten nur über einen solchen Kommandointerpreter bedient werden. Der Benutzer tippte den Namen eines Programms ein, das Betriebssystem suchte die entsprechende Programmdatei auf der Festplatte oder auf der Diskette, lud sie in den Hauptspeicher und startete das Programm. Nachdem dieses beendet war, kehrte die Kontrolle zu dem Kommandointerpreter zurück. Erst dann konnte der nächste Befehl gegeben werden. Geriet das Programm in eine Endlosschleife, musste der Rechner neu gestartet werden.

Multitasking Betriebssysteme müssen gleichzeitig viele Kommandos ausführen. Während in einem Fenster ein Web-Browser läuft, im anderen ein Mail-Programm, sorgt vielleicht ein mp3-Abspielprogramm für gute Stimmung. Jedes dieser gleichzeitig ablaufenden Programme ist ein *task* (engl. für *Aufgabe*). Unter *Windows* kann man durch gleichzeitiges Drücken der

Tasten *Strg-Alt-Entf* (*Ctrl-Alt-Del*) alle aktiven Tasks anzeigen lassen. Man erkennt, dass neben den Benutzerprogrammen viele Verwaltungstasks des Betriebssystems aktiv sind. Ein *Multitasking-Betriebssystem* muss also nicht nur die angeschlossenen Geräte verwalten, sondern auch dafür sorgen, dass die vielen gleichzeitig im Hauptspeicher laufenden Programme, die sogenannten *Prozesse*, sich nicht gegenseitig stören, dass sie sich in der Benutzung der CPU abwechseln, dass sie bei Bedarf ein Gerät – etwa einen Drucker – für einen *job* zugeteilt bekommen, diesen aber nicht auf unbestimmte Zeit blockieren.

Multi-user Betriebssysteme müssen zusätzlich noch mehrere Benutzer verwalten und dafür sorgen, dass deren Programme und Daten vor dem Zugriff – und den Augen – der jeweils anderen angemessen geschützt sind.

Ein Betriebssystem übernimmt also umfangreiche Verwaltungstätigkeiten. Es ist andererseits selber ein Programm, benötigt daher ähnliche Ressourcen wie alle anderen. Es muss aber mehr Rechte besitzen, um notfalls ein abgestürztes Programm zu beenden, Speicher wieder freizugeben oder den Zugriff auf eine Ressource zu verweigern. Um zu verhindern, dass Benutzerprogramme sich ähnliche Rechte anmaßen, muss der Prozessor verschiedene *Privilegierungsstufen* vorsehen. Die höchste Stufe steht nur dem *Kern*, d.h. den besonders zentralen und kritischen Teilen des Betriebssystems zur Verfügung. Anwendungsprogramme können nur in der niedrigsten Stufe laufen.

Meist merkt der Benutzer von alledem nichts. Nachdem nämlich sein Rechner hochgefahren ist, wird automatisch ein Betriebssystemaufsatz gestartet, der alle Tätigkeiten des Betriebssystems und alle Ressourcen, wie z.B. Dateien, Geräte und Programme, hinter einer *Benutzeroberfläche* (*user interface*) versteckt. Dem Benutzer wird also ein Schreibtisch mit Schreibwerkzeugen, Telefon, Akten, Papierkorb und Aktenschränken etc. vorgegaukelt, die er mit der Maus durch Anklicken und Verschieben bedient. Der Betriebssystemaufsatz sorgt dafür, dass die Aktionen des Benutzers in entsprechende Programmaufrufe und Dateioperationen des zugrunde liegenden Betriebssystems umgesetzt werden. Benutzeroberflächen sind daher nicht an ein bestimmtes Betriebssystem gebunden. In der Tat hat der Benutzer z.B. bei *Linux* die Qual der Wahl zwischen verschiedenen Benutzeroberflächen, darunter auch einer, die in Aussehen und Funktionsweise dem *Windows-Desktop* entspricht.

Nicht alle Funktionen des Betriebssystems sind über die Bedienoberfläche erreichbar. Im Gegensatz zum Endanwender im Büro, müssen Systemverwalter verstehen, wie das Betriebssystem die Ressourcen – Festplatten, Hauptspeicher, Rechenzeit – verwaltet und mit welchen Kommandos dies gesteuert werden kann. Programmierer müssen wissen, welche System-Kommandos bereitstehen und wie diese in Anwendungsprogrammen eingesetzt werden können.

Einige Betriebssysteme sind nur für PCs verfügbar, dazu gehören das veraltete MS-DOS und verschiedene Varianten von MS-Windows (Windows 3.1, Windows 95/98/ME). Vorwiegend auf PCs werden Windows NT, Windows 2000, Windows XP, OS/2, BeOS und Linux eingesetzt. Diese Systeme sind aber auch auf anderer Hardware (z.B. auf Workstations) verfügbar. MacOS ist ein Betriebssystem speziell für die Macintosh Rechner der Firma Apple.

Workstations werden meistens unter dem Betriebssystem UNIX oder Abkömmlingen dieses Systems wie Linux, Solaris, AiX etc. betrieben. Bekannte Betriebssysteme für Mainframes heißen MVS, VM/SP, CMS und BS 2000.

6.2 Betriebsarten

Je nach Art der vorherrschenden Anwendung kann man verschiedene *Betriebsarten* eines Betriebssystems unterscheiden. Bei einem *Anwender-Betriebssystem* hat man direkten Zugang zum Rechner über Tastatur und Bildschirm und kann beliebige Programme aufrufen. Die einfachsten Anwender-Betriebssysteme bieten *einem* Anwender die Möglichkeit, mit *einem* Computer zu arbeiten. Schwieriger wird es, wenn *mehrere* Anwender gleichzeitig mit einem Computer arbeiten wollen. *Multi-user* Betriebssysteme müssen jedem Anwender im *Zeitmultiplexverfahren* (*timesharing*) den Rechner abwechselnd für kurze Zeit zur Verfügung stellen – man spricht von *Teilnehmerbetrieb*. Im Gegensatz dazu haben beim *Teilhaberbetrieb* die Anwender nur eingeschränkten Zugang zu bestimmten Funktionen, so genannten *Dienstleistungen* des Rechners.

6.2.1 Teilhaberbetrieb

Im Teilhaberbetrieb wird dem Anwender nur die Möglichkeit gegeben, bestimmte vordefinierte Transaktionen auszuführen. Ein Beispiel ist das System einer Bank, die an ihren Schaltern Terminals oder Personal Computer installiert hat, die mit einem Zentralrechner verbunden sind und den Angestellten oder den Kunden bestimmte Transaktionen erlauben, wie z.B. Überweisungen zu tätigen oder Kontoauszüge zu drucken. Andere Beispiele sind Systeme für Flugreservierungen und Reisebuchungen.

Die Anwender sind jeweils mit einem bestimmten Programm, einem *Transaktionsmonitor*, direkt verbunden und können diesem ihre Wünsche mitteilen. Dieses Programm macht aus diesen Wünschen *Transaktionen* und reicht sie an andere Programme weiter, die diese abarbeiten. *Transaktionen* sind jeweils aus kleineren Unteraufträgen bestehende Aufgaben, die entweder komplett oder gar nicht bearbeitet werden müssen.

Ein typisches Beispiel für eine Transaktion ist ein Überweisungsauftrag. Dieser beinhaltet eine Abbuchung vom Sendekonto und eine Gutschrift auf dem Empfangskonto. Es wäre problematisch, wenn nur einer dieser Teilaufträge erledigt würde. Der Transaktionsmonitor garantiert, dass *Transaktionen* komplett durchgeführt werden oder, falls ein Problem auftritt, dass der Zustand zu Beginn der Transaktionsbearbeitung wiederhergestellt wird. Dieses Beispiel wird im Datenbankkapitel ab S. 781 ausführlich diskutiert.

6.2.2 Client-Server-Systeme

Klassische Teilhaber-Systeme werden immer stärker durch verteilte *Client-Server-Systeme* verdrängt. Dabei sind die Anwender nicht mehr direkt mit einem Zentralrechner verbunden, sondern indirekt mit einem oder mehreren Rechnern in einem Netzwerk. Der Anwender arbeitet mit einem Programm, das auf dem Personal Computer an seinem Arbeitsplatz abläuft. Dieses kann Dienstleistungen von einem oder mehreren Rechnern in dem Netzwerk anfor-

dem. Das anfordernde Programm kann man sich als Kunde (engl. *Client*) vorstellen, der von dem Dienstleister (*Server*) einen Dienst anfordert. Diese Betriebsart wird häufig *Client-Server-Betrieb* genannt. Im Unterschied zum Teilhaberbetrieb können von dem Rechner des Anwenders aus viele Dienste und Dienstleister in Anspruch genommen werden, so etwa Datei-Server, Datenbank-Server (siehe dazu auch S. 794 ff), Druck-Server oder Mail-Server.

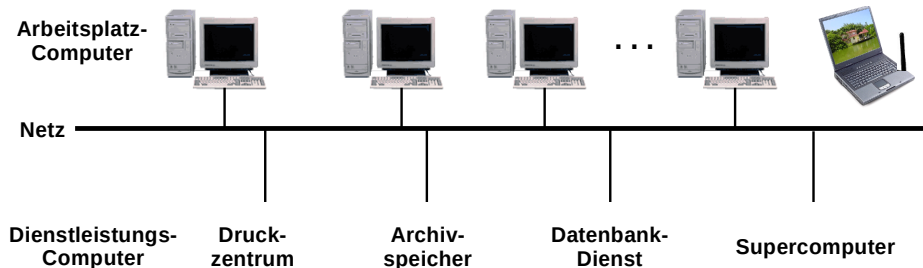


Abb. 6.1: Client-Server-Betrieb

Durch eine solche Aufgabenverteilung werden die Dienstleistungsrechner (*server*) entlastet und können ihre eigentliche Aufgabe besser und schneller erbringen. Die Benutzerschnittstelle und die Überprüfung der Benutzereingaben werden direkt am Arbeitsplatz realisiert, bei dem *client*. Nur diejenigen Benutzerwünsche, die nicht bereits lokal erledigt werden können, werden als Dienstleistungsanforderungen über das Netz an den Server geschickt. Dienstleistungen können durch Anwenderprogramme auf Universalrechnern realisiert werden. In diesem Fall ist dann z.B. nur der Transaktionsmonitor der wesentliche Teil des Betriebssystems dieser Rechner. Dienstleistungsrechner sind von der Notwendigkeit entlastet, direkt mit Anwendern zu kommunizieren und können sich folglich auf andere Dinge konzentrieren wie z.B.:

- garantiertes Antwortzeitverhalten,
- Ausfallsicherheit und Transaktionsbearbeitung,
- Netzzugriffe,
- Datenschutz und Datensicherheit.

Jedes einzelne Betriebssystem in einem verteilten System könnte einfacher sein als ein Universalbetriebssystem, das alle Aufgaben eines Computersystems abdecken muss. In vielen Fällen wird aber auch im Client-Server-Betrieb ein Universalbetriebssystem – häufig *Linux* – auf der Serverseite eingesetzt.

Die verteilte *dezentrale* Rechnerversorgung wirft einige organisatorische Probleme auf. Wie sollen zum Beispiel verschiedene Entwickler auf gemeinsame Daten zugreifen? Muss jedes benutzte Programm auf jedem Rechner als Kopie vorliegen oder kann es zentral gehalten und gemeinsam genutzt werden? Die Lösung konnte nur sein, dass man die verschiedenen Rechner in irgendeiner Weise untereinander verband und so zu einem *Netz* zusammenfasste. Ein oder mehrere leistungsfähige Rechner in diesem Netz speichern zentral zumindest die gemeinsam genutzten Dateien und Programme und spielen so die Rolle eines Datei-Verwalters (engl. *File-Server*). Sie sind mit einer großen Plattenkapazität ausgestattet, während die Rechner der Benutzer im Extremfall nicht einmal eine Festplatte besitzen müssen (*Diskless*