

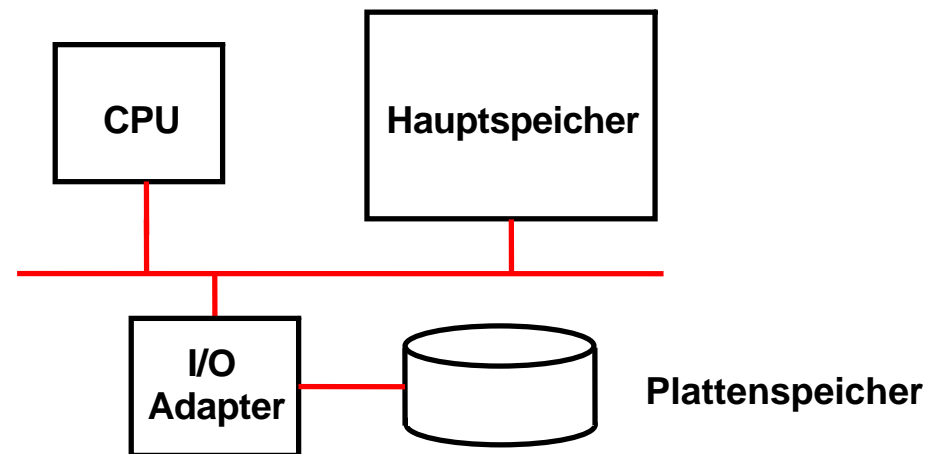
**Enterprise Computing  
Einführung in das Betriebssystem z/OS**

**Prof. Dr. Martin Bogdan  
Prof. Dr.-Ing. Wilhelm G. Spruth**

**WS2012/13**

**Verarbeitungsgrundlagen Teil 1**

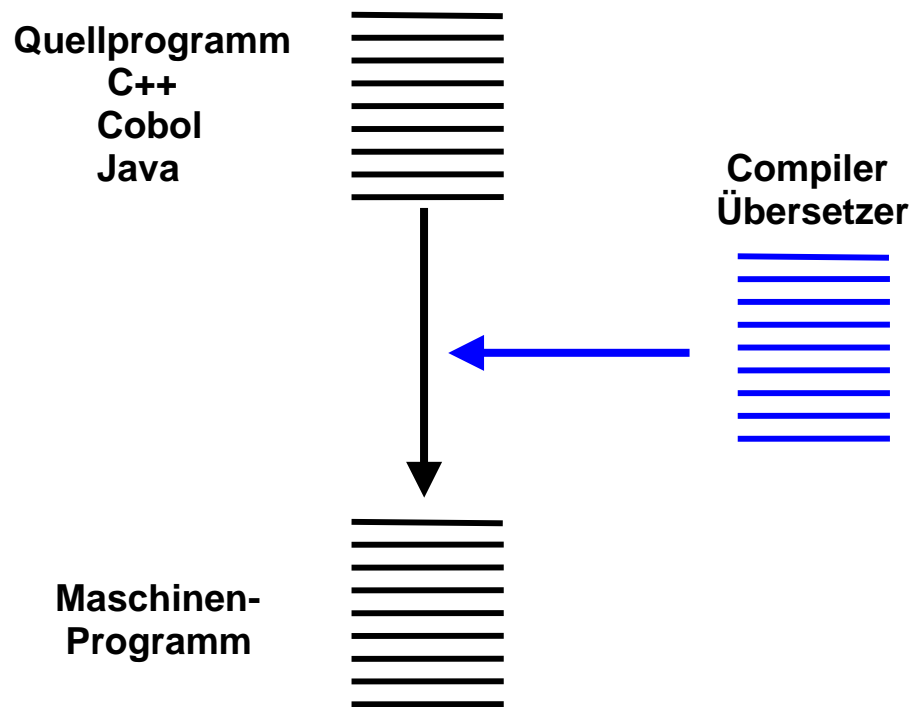
**Multiprogrammierung**



**Ein moderner Rechner besteht aus einer CPU, welche ein Programm ausführt, einem Hauptspeicher (Main Storage), welcher das auszuführende Programm und temporäre Daten enthält, und einem Anschluss (I/O Adapter) für einen (oder viele) Plattenspeicher, auf dem ausführbare Programme in Bibliotheken (Libraries) und Daten in Dateien und Datenbanken gespeichert sind.**

**Ein auszuführendes Programm besteht aus einer Folge von Maschinenbefehlen (Machine Instructions). Das Format der Maschinenbefehle wird durch die Rechnerarchitektur bestimmt. Architekturen wie x86 (Pentium), PowerPC, Sparc, Itanium und System z (Mainframe) haben unterschiedliche Formate der Maschinenbefehle.**

**Ein Anwendungsprogramm, auch Benutzerprogramm genannt (application program, user program) wird heutzutage fast immer in einer höheren Programmiersprache wie Java, C++, Cobol, PL/1, ADA geschrieben, und durch einen Compiler in ein Maschinensprache-Programm übersetzt, welches aus einer Folge von Maschinenbefehlen besteht.**



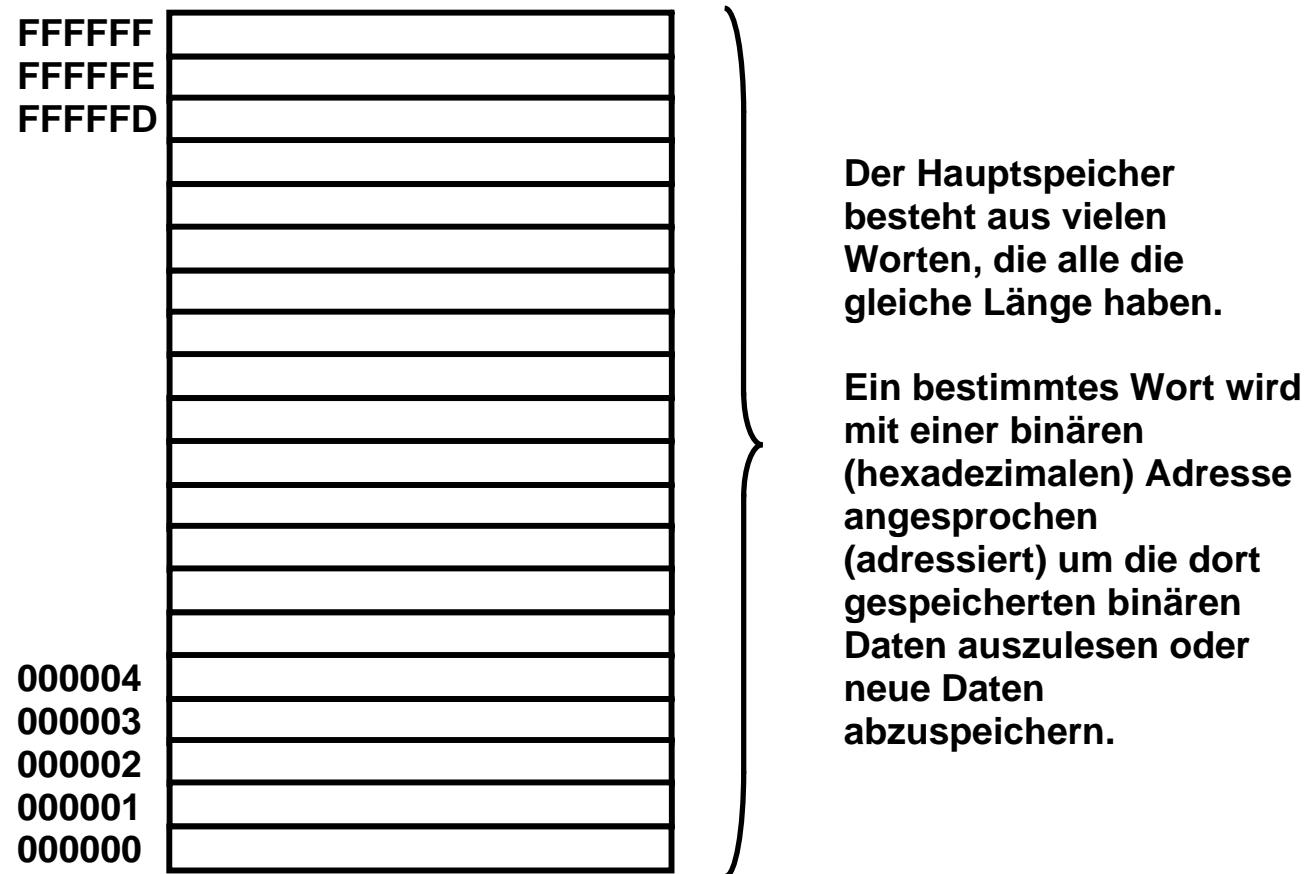
Benutzerprogramme (andere Bezeichnung Anwendungsprogramm, engl.: user programm, application programm) werden in der Regel in einer höheren Programmiersprache wie C++, Cobol, PL/1, Java usw. geschrieben.

Ein Compiler ist ein Programm, dessen Ausführung ein Quellprogramm (Source Programm) als Input Daten benutzt, um daraus ein Maschinenprogramm, bestehend aus einzelnen Maschinenbefehlen, zu erstellen. Andere Bezeichnungen: Object Programm oder Binaries. Der Aufruf eines Compilers bewirkt eine Übersetzung (Compilation) eines Quellprogramms in ein Maschinenprogramm.

Meistens benutzt man ein weiteres Programm, einen **Linker**, um das übersetzte Maschinenprogrammen mit anderen, bereits früher übersetzten Programmen, zu einem ausführbaren Programm (executable program) zu verknüpfen. Ein Benutzerprogramm kann aus einzelnen Teilen (Modulen) bestehen, die einzeln geschrieben und übersetzt werden. Die übersetzten Maschinenprogramm Module werden in einem als Programmbibliothek (Library) bezeichneten Verzeichnis auf dem Plattenspeicher untergebracht.

Das Betriebssystem stellt zahlreiche weitere Maschinenprogramm Module zur Verfügung, die mit Hilfe des Linkers ebenfalls mit einem neu erstellten Maschinenprogramm verknüpft werden. Beispiele sind Ein/Ausgabe (I/O) Routinen oder die Socket Library für Intersystem Communication. Unter dem z/OS Betriebssystem stehen viele solcher Maschinenprogramme in einer mit dem Parameter SYS1 gekennzeichneten Library.

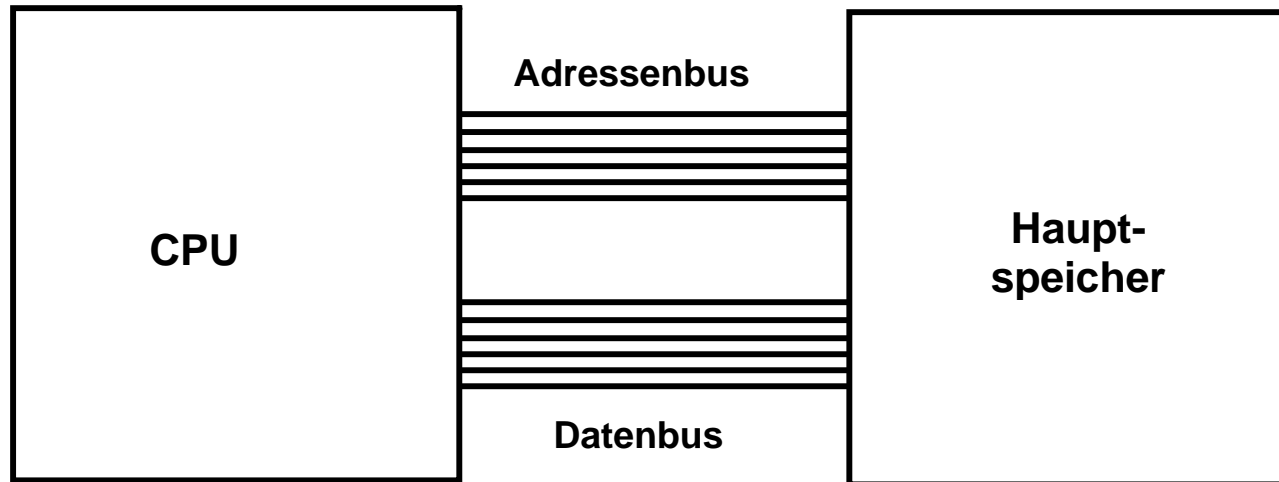
Ein **Loader** ist ein Maschinenprogramm, welches ein ausführbares Programm aus einer Library ausliest und in den Hauptspeicher lädt.



**Der Hauptspeicher (main Storage) besteht aus lauter gleich langen Wörtern, die mit binären bzw. hexadezimalen Ziffern angesprochen (adressiert), gelesen und gespeichert werden.**

**Die Wortlänge eines Hauptspeichers ist entweder 8, 12, 16, 14, 32, 36 oder 48 Bit. Moderne Rechner benutzen fast ausschließlichschließlich eine Wortlänge von 8 Bit ( ein Byte).**

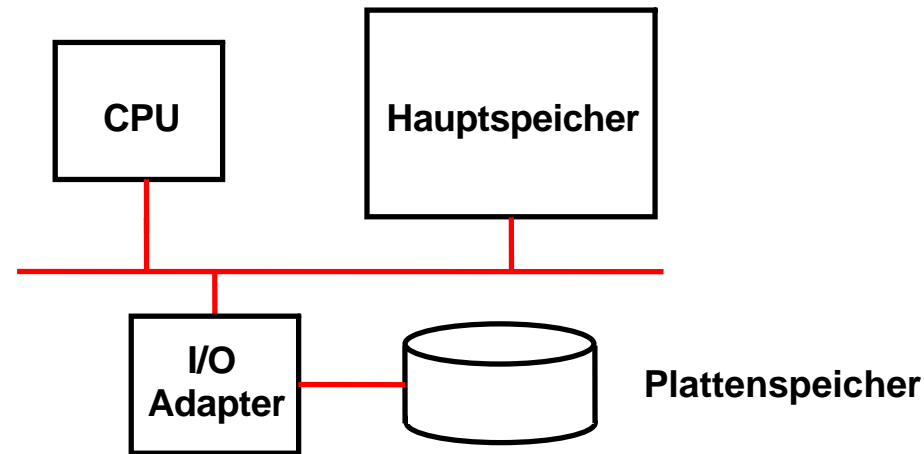
**Im Hauptspeicher werden sowohl Maschinenprogramme als auch Daten gespeichert.**



Wir unterscheiden zwischen 32 Bit und 64 Bit Architekturen. Bei einer 32 Bit Architektur ist die CPU mit dem Hauptspeicher mit einem Adressenbus und einem Datenbus verbunden. Der Adressenbus besteht aus 32 Leitungen, um eine von  $2^{32}$  Adressen an den Hauptspeicher zu übergeben. Bei einer 64 Bit Architektur sind es 64 Leitungen, um eine von  $2^{64}$  Adressen an den Hauptspeicher zu übergeben. Der Datenbus besteht aus 8 Leitungen, auf denen jeweils 1 Byte (8 Bit) zwischen Hauptspeicher und CPU übertragen wird.

Die Maschinenbefehle gehen von dieser Struktur aus, die wir als das logische Erscheinungsbild einer CPU bezeichnen. Die physische Implementierung eines Rechners weicht in der Regel von der logischen Sicht ab. So besteht z.B. ein Datenbus häufig aus 256 Leitungen, und es werden gleichzeitig 8 Bytes (256) Bits zwischen CPU und Hauptspeicher übertragen. Ein moderner Mainframe Rechner (z196) hat einen physischen Hauptspeicher mit einer maximalen Größe von 4 TByte ( $2^{42}$  Byte), für dessen Adressierung 42 Leitungen des Adressenbusses ausreichen.

Anmerkung: Bei der Mainframe Architektur verfügt die ehemalige 32 Bit Architektur nur über 31 Bit Adressen, und adressiert damit einen Hauptspeicher mit einer maximalen Größe von  $2^{31}$  Adressen und  $2^{31}$  Byte Speicherkapazität.



**Im Gegensatz zum Hauptspeicher ist der Zugriff auf einen Plattenspeicher sehr viel komplizierter. Für den Zugriff setzt die CPU ein spezielles Maschinenprogramm, einen I/O (Input/Output) Driver, ein.**

**Eine moderne CPU führt größenordnungsmäßig eine Milliarde (  $10^9$  ) Maschinenbefehle pro Sekunde aus. Die Zugriffszeit zum Hauptspeicher beträgt größenordnungsmäßig wenige 100 Picosekunden (  $10^{-9}$  bis  $10^{-10}$  s, logische Sicht ). Die Zugriffszeit zum Plattenspeicher beträgt größenordnungsmäßig 10 Millisekunden (  $10^{-2}$  s ). Dieser Unterschied in der Zugriffszeit hat eine sehr großen Einfluss auf die Struktur moderner Betriebssysteme.**

Adresse FF..FF

**Benutzer Bereich  
(User Space)**

enthält  
Anwendungs-  
programme  
(Application  
Programs)

**Systembereich  
(Supervisor Space)**

enthält  
Überwacher  
(Supervisor,  
Kernel)

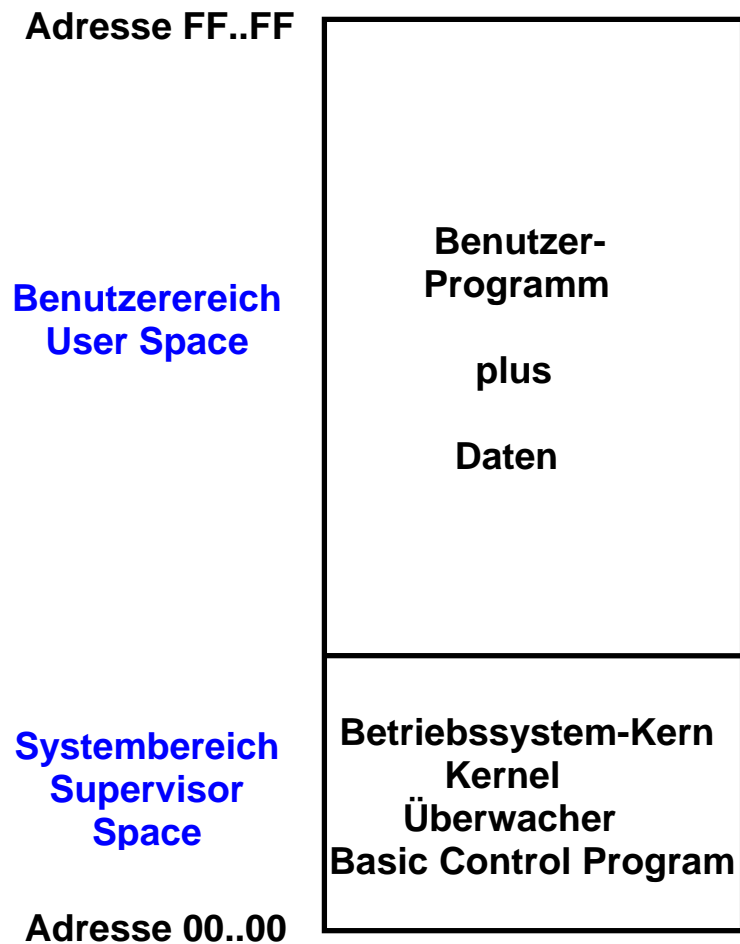
Adresse 00..00

10 KByte  
bei kleinen,  
> 10 Gbyte  
bei großen  
Rechnern

## Aufteilung des Hauptspeichers in einen System- und einen Benutzerbereich

Ein Betriebssystem enthält Programmroutinen, die von Benutzerprogrammen immer wieder gebraucht werden. Ein Beispiel ist der I/O Driver (Input/Output Driver), der das Lesen oder Schreiben von Daten vom/zum Plattenspeicher bewirkt..

Der Code des Betriebssystems befindet sich auf dem Plattenspeicher. Beim Hochfahren eines Rechners wird ein Teil des Betriebssystems in den Hauptspeicher geladen. Diesen Teil bezeichnet man als Betriebssystem-Kern, Kernel oder Überwacher (Supervisor). z/OS benutzt auch die Begriffe Nucleus oder „Basic Control Program“ (BCP).



Wir bezeichnen die Menge aller Hauptspeicheradressen als Adressenraum (Address Space). In dem hier dargestellten einfachsten Fall wird der Adressenraum des Hauptspeichers in 2 Teile aufgeteilt: einen Teil für den Überwacher und einen zweiten Teil für das auf dem Rechner laufende Benutzerprogramm (Anwendungsprogramm) und seine Daten. Das ursprüngliche DOS Betriebssystem für den PC hatte diese Struktur.

Der gesamte Adressenraum des Hauptspeichers gliedert sich in einen Überwacher Adressenraum (Supervisor Address Space) und einem Benutzer-Adressenraum (User Address Space).

Damit ein fehlerhaftes oder böswilliges Benutzerprogramm nicht unbefugt Programme oder Daten im Überwacher Adressenraum modifizieren kann, läuft der Rechner in jedem Augenblick entweder im Überwacherstatus (Supervisor State) oder im Benutzerstatus (User State, auch als Problem State bezeichnet).

Ein Programm, das im Überwacherstatus läuft, kann auf den gesamten Adressenraum des Hauptspeichers zugreifen. Ein Programm, das im User Modus läuft, hat Zugriffsrechte nur für den User Adressenraum.



# Prozess

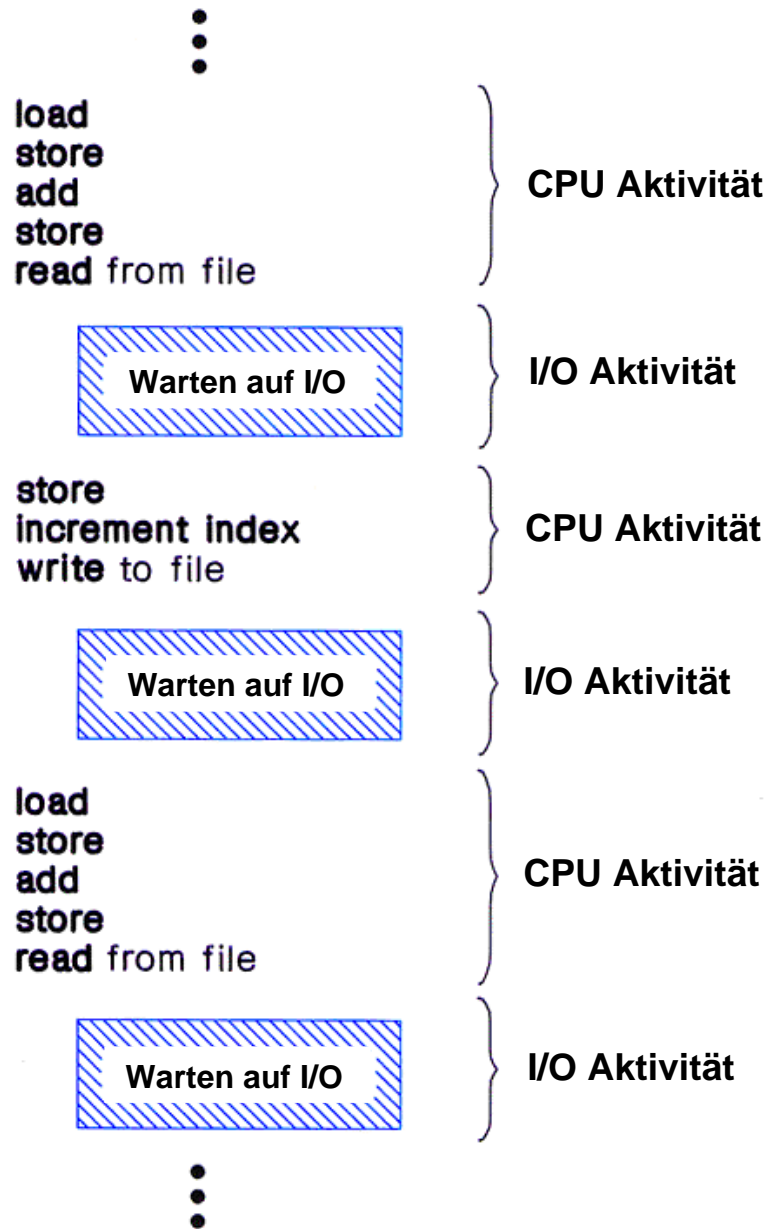
Ein Maschinen-Programm lässt sich mit den Noten eines Musikstückes vergleichen. Ein Konzert ist die Aufführung eines Musikstückes entsprechend den verwendeten Musiknoten. Ein **Prozess** ist die Ausführung eines Maschinenprogramms auf einem Computer.

Der Begriff „Prozess“ ist ein zentrales Konzept in der modernen Datenverarbeitung. Ein Prozess ist die Ausführung von einem Programm (oder mehreren Programmen) um ein für einen Benutzer relevantes Problem zu bearbeiten. Beispiele für Prozesse sind die monatliche Lohn- und Gehaltsabrechnung in einem Unternehmen, die Verbuchung bei der Überweisung eines Geldbetrages, eine URL Abfrage im Internet oder die Stücklistenenerstellung beim Bau eines Automobils.

Ein Benutzer Prozess läuft innerhalb des Benutzer Adressenraums. Systemprozesse bearbeiten irgendwelche Teilaufgaben des Betriebssystems.

Die Ausführung eines Prozesses bewirkt in der Regel, dass Daten abgeändert werden, die auf einem Plattenspeicher permanent gespeichert sind. Ein Beispiel sind Kontodaten bei der Überweisung eines Geldbetrages. Hierzu lädt der Prozess die Daten vom Plattenspeicher in den Hauptspeicher, modifiziert sie und schreibt das Ergebnis auf den Plattenspeicher zurück.

Eine Transaktion ist eine spezielle Art eines Prozesses, bei dem garantiert wird, dass alle Änderungen von Daten auf dem Plattenspeicher entweder vollständig oder garnicht, nicht aber teilweise, erfolgen.



## Struktur eines Benutzerprogramms

Ein Benutzerprogramm besteht aus einer Folge von

- Gruppen von Maschinenbefehlen
- I/O Operationen (Zugriff auf Plattenspeicher-Daten).

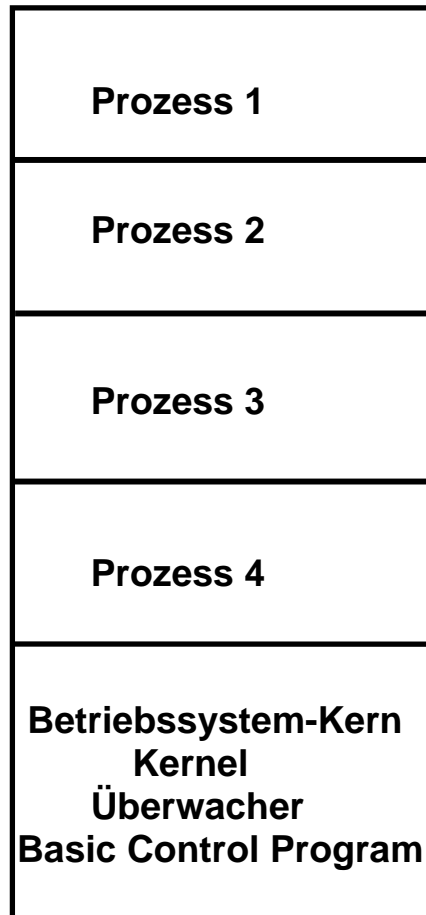
Die Programmausführung ist eine abwechselnde Folge von CPU und I/O Aktivitäten.

Beim Start einer I/O Operation muss das Anwendungsprogramm in der Regel warten, bis die I/O Operation abgeschlossen ist (z.B. die vom Plattenspeicher gelesenen Daten im Hauptspeicher eingetroffen sind).

Der Zugriff auf einem Plattenspeicher benötigt etwa 10 Millisekunden. Während dieser Zeit könnte die CPU etwa 10 Millionen Maschinenbefehle ausführen. Es wäre unökonomisch, wenn die CPU während der I/O Aktivität warten müsste.

# Multiprogrammierung

Adresse FF..FF



Adresse 00..00

Hierzu teilt man des Benutzer Adressenraum in mehrere Teilbereiche (Regions) auf und ordnet einem Prozess jeweils eine eigene Region zu.

Jetzt befinden sich die Programme (Code) mehrerer Prozesse gleichzeitig im Hauptspeicher. Wir nehmen in diesem Beispiel an, dass der Rechner eine einzige CPU hat. Wenn nun ein Prozess eine I/O Operation ausführt, wird er in einen Wartezustand versetzt. Ein anderer Prozess, dessen Programme und Daten sich ausführungsbereit in seiner Region des Benutzer Adressenraums befinden, wird nun von der CPU ausgeführt, bis er ebenfalls eine I/O Operation ausführen will. Jetzt beginnt die CPU einen dritten Prozess auszuführen usw.

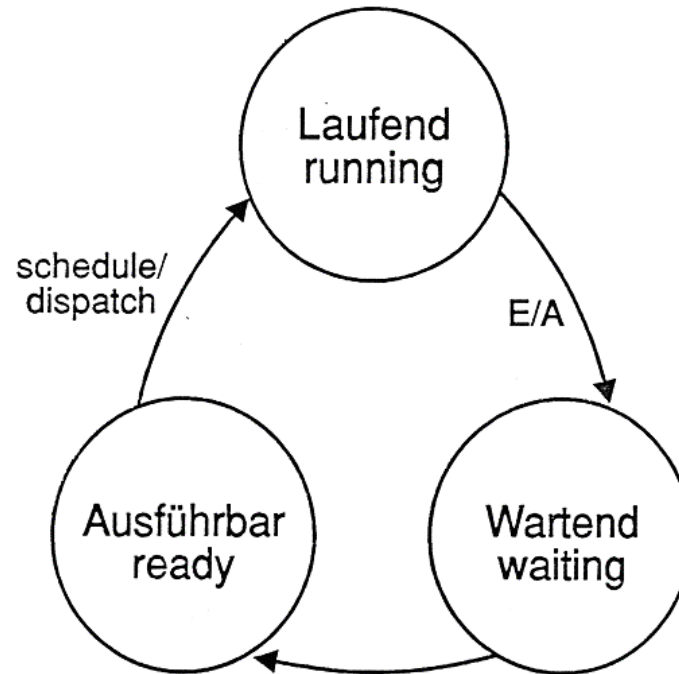
In einem Mainframe Rechner werden auf diese Art typischerweise tausende oder zehntausende von Prozessen parallel ausgeführt. Diese Art der Datenverarbeitung, bei der mehrere Prozesse ineinander geschachtelt parallel verarbeitet werden, wird als „Multiprogrammierung“ (Multiprogramming) bezeichnet.

# Zustand von Prozessen

Wenn ein Prozess eine Plattenspeicher I/O Operation startet wird er in den Zustand „wartend“ (waiting) versetzt. Wenn die I/O Operation abgeschlossen ist, könnte er wieder von der CPU ausgeführt werden. Da die CPU vermutlich gerade mit einem anderen Prozess beschäftigt ist, wird er statt dessen in den Zustand „ausführbar“ (ready) versetzt. Irgendwann wird die CPU den Prozess weiter verarbeiten. Ein Prozess, der durch die CPU verarbeitet wird, wird als „Laufend“ (running) bezeichnet.

Jeder Prozess rotiert wiederholt durch die Zustände laufend, wartend und ausführbar. Moderne Rechner haben meistens mehr als eine CPU, z.B. 80 bei einem z196 Mainframe. In der Regel verarbeitet jede CPU einen anderen Prozess. In diesem Fall können sich mehrere Prozesse im Zustand „laufend“ befinden.

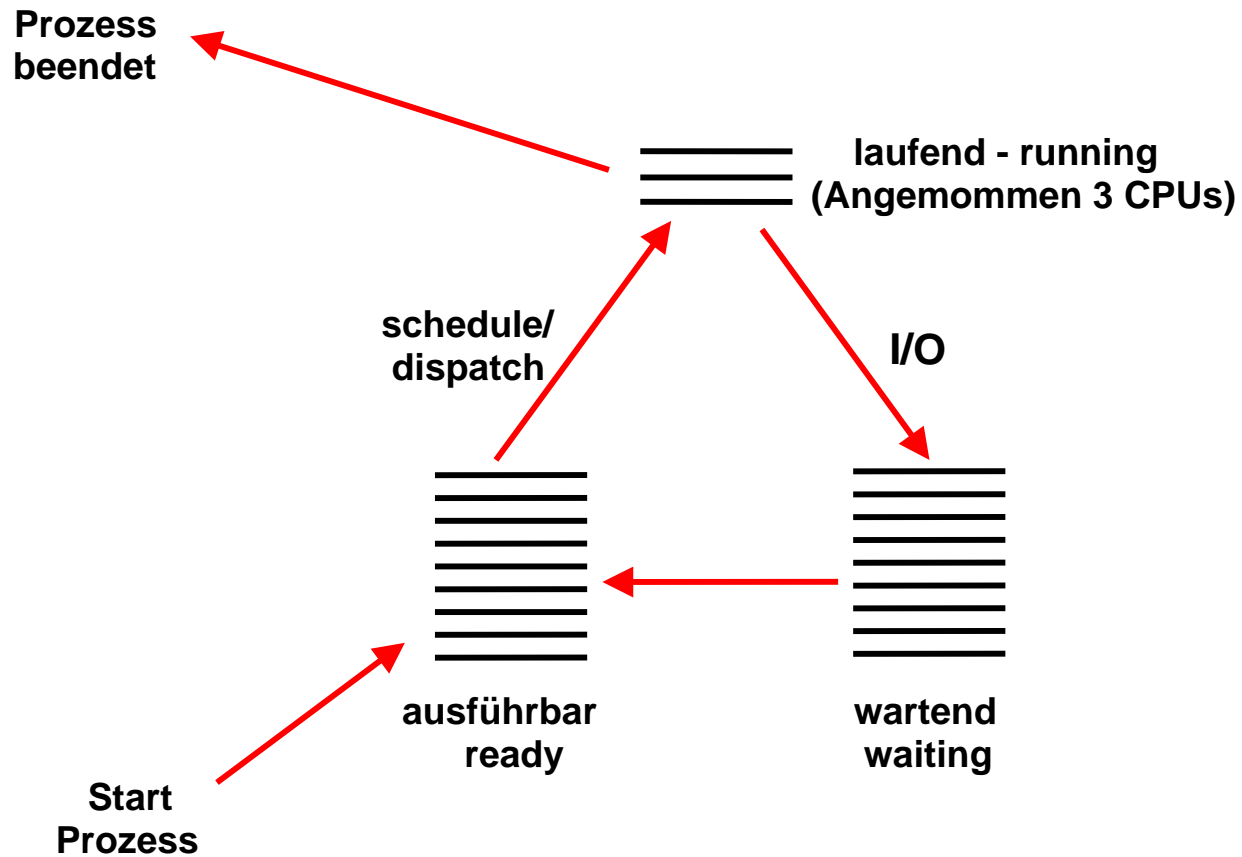
- Ein Prozess befindet sich im Zustand laufend, wenn er von einer der verfügbaren CPUs ausgeführt wird.
- Ein Prozess befindet sich im Zustand wartend, wenn er auf den Abschluss einer I/O Operation wartet.
- Ein Prozess befindet sich im Zustand ausführbar, wenn er darauf wartet, dass die Scheduler Komponente des Überwachers ihn auf einer frei geworden CPU in den Zustand laufend versetzt.



Jeder Prozess wird durch einen Bereich im Hauptspeicher (Task Control Block, TCB) gekennzeichnet und repräsentiert. Die Scheduler Komponente des Überwachers unterhält drei Warteschlangen für wartende, ausführbare und laufende Prozesse. In diese Warteschlangen werden die TCBs der Prozesse beim Wechsel des Zustandes eingeordnet.

Der Scheduler des Überwachers selektiert aus der Warteschlange ausführbarer Prozesse jeweils einen Kandidaten wenn immer eine CPU frei wird.

In einem Rechner laufen immer zahlreiche Prozesse gleichzeitig ab. Die Prozesse befinden sich immer in einem von drei Zuständen. Hierfür werden drei Warteschlangen benutzt, in denen sich die TCBs der Prozesse befinden.



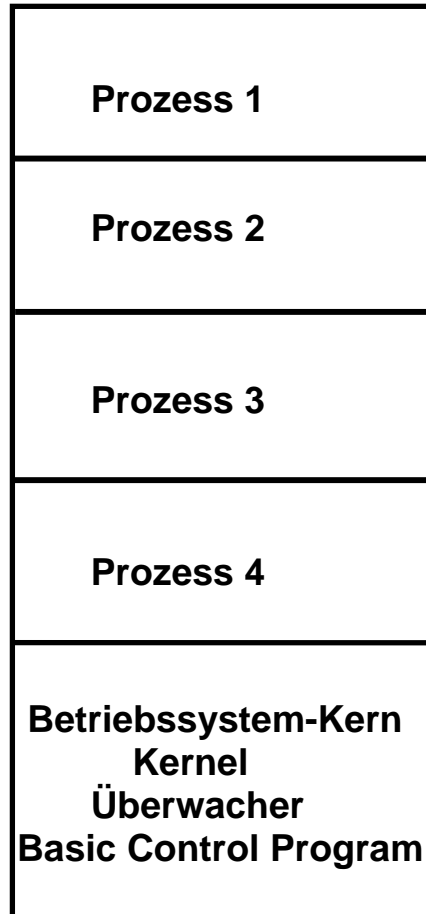
## Zustand von Prozessen

Jede Linie entspricht einem Prozess, dargestellt durch seinen TCB (Task Control Block).

Der Scheduler des Betriebssystem Kernels verwaltet Task Control Block (TCB) Warteschlangen.

Ausführbare Prozesse verfügen über Ressourcen, besonders über Platz im Hauptspeicher (Programm Code, Daten)

Adresse FF..FF



Adresse 00..00

Die Aufteilung des Benutzer Adressenraum in mehrere Regions für mehrere Prozesse erzeugt mehrere Probleme:

- Es muss verhindert werden, dass fehlerhafte Programme eines Prozesses auf die Region eines anderen Prozesses zugreift.
- Bei Beedigung eines Prozesses wird die benutzte Region des Benutzer Adressenraums freigegeben. In ihr kann jetzt ein neuer Prozess gestartet werden.

Die einzelnen Prozesse benötigen aber unterschiedlich viel Speicherplatz. Die Regions haben deshalb eine unterschiedliche Größe. Die Größe der freigewordenen Region entspricht nicht notwendigerweise den Bedürfnissen des neuen Prozesses. Eine komplexe Speicherverwaltungskomponente (als Teil des Überwachers) ist erforderlich.

Zur Lösung dieses Problems führte IBM 1970 die virtuelle Speichertechnik (Virtual Storage) für eine neue Serie von Mainframe Rechnern (System /370) ein.