

EDV-Terminologie (Folge 3)

Assembler (dt. Maschinensprache)

Der Assembler ist eine spezielle → Programmiersprache, in der jeder Befehl eindeutig einem → Maschinenbefehl der → Zentraleinheit zugeordnet ist. Assemblerprogramme sind deshalb maschinenabhängig und können nicht ohne weiteres von einem Maschinentyp zu einem anderen portiert werden (→ Portierung). Der Begriff des Assemblers wird auch verwendet, um einen bestimmten → Compiler zu bezeichnen, der in Assemblersprache geschriebene Programme in den → Object Code übersetzt.

Binärzahl

Eine aus den Ziffern 0 und 1 zusammengesetzte Dualzahl (→ Dualsystem) nennt man auch Binärzahl.

Datei

Eine Datei (engl. File) ist im Sprachgebrauch der Datenverarbeitung die Zusammenfassung von Datensätzen in einem gemeinsamen Speicherbereich. Normalerweise werden Dateien auf → externen Speichern angelegt.

Man unterscheidet verschiedene Dateitypen:

- sequentielle Datei
- Random-Datei
- ISAM-Datei
- ausführbare Datei

Directory (dt. Inhaltsverzeichnis)

Der Begriff „Directory“ wird normalerweise im Zusammenhang mit → externen Speichern verwendet. Er bezeichnet hier einen bestimmten Speicherbereich, der ein Verzeichnis enthält, welche → Dateien sich wo auf dem externen Speicher befinden. Meistens sind zusätzlich die Größe der Dateien und das Erstellungsdatum sowie weitere Informationen z.B. über die Zugriffsberechtigung mit im Directory gespeichert. Der Aufbau und die Verwaltung des Directory wird vom → Betriebssystem durchgeführt.

Dualsystem

Das von Leibniz entwickelte Dualsystem kennt nur zwei Ziffern, die Ziffer 0 und die Ziffer 1. Jede Zahl wird durch diese zwei Ziffern dargestellt. Das System erkennt man unschwer, wenn man sich die Dualzahlen von 0 bis 8 anschaut:

Dezimalzahl	Dualzahl	
0	0	(0)
1	1	(1)
2	10	($2^1 + 0$)
3	11	($2^1 + 1$)
4	100	($2^2 + 0 + 0$)
5	101	($2^2 + 0 + 1$)
6	110	($2^2 + 2^1 + 0$)
7	111	($2^2 + 2^1 + 1$)
8	1000	($2^3 + 0 + 0 + 0$)

File

→ Datei

Hexadezimalsystem

Das hexadezimale Zahlensystem unterscheidet sich von dem gebräuchlichen dezimalen Zahlensystem dadurch, daß die Basis nicht die Zahl 10 sondern die Zahl 16 ist. Daraus ergibt sich, daß eine zweite Stelle nicht nach Erreichen der Zahl 9 eingeführt werden muß, sondern erst nach Erreichen der Zahl 15.

Zu diesem Zweck war es notwendig, die im Dezimalsystem gebräuchlichen Zahlzeichen 0,1,2,3,4,5,6,7,8 und 9 um 6 weitere Zahlzeichen zu erweitern, die in einer Stelle die Werte 10 bis 15 repräsentieren. Dazu werden die Buchstaben A-F verwendet, wobei der Buchstabe A den Wert 10 und der Buchstabe F den Wert 15 darstellt.

Die Umrechnung von Hexzahlen in Dezimalzahlen soll an einem Beispiel erläutert werden:

Die Zahl „3F“ in hexadezimaler Schreibweise stellt folgenden Wert im Dezimalsystem dar:

Das Zahlzeichen „F“ in der niederwertigsten Stelle repräsentiert den Wert 15. Das Zahlzeichen 3 in der höherwertigen Stelle repräsentiert den Wert 3 multipliziert mit der Basis des Zahlensystems 16, also 48. Dezimal ausgedrückt ist „3F“ also der Wert $48 + 15 = 63$ zugeordnet (3 mal die Basis 16 + F).

Den Sinn des hexadezimalen Systems erkennt man bei einer Gegenüberstellung mit dem → Binärsystem, das intern in den meisten Rechnern verwendet wird. Das Binärsystem kann die Ziffern von 0-15 mit genau vier Stellen ($2^4 = 16$) darstellen. Einer Hexziffer sind also immer genau vier Binärziffern zugeordnet. Dadurch kann man die Umrechnung von einem Zahlensystem ins andere vornehmen, ohne daß Überträge zwischen einzelnen Stellen zu berücksichtigen sind. Dies erleichtert die Konvertierung von einem Zahlensystem in das andere. Würde man Binärzahlen in das Dezimalsystem konvertieren wollen, würden immer etwa 3,5 Stellen einer Dezimalziffer entsprechen. Dies bedeutet, daß Überträge zwischen den Ziffern zu berücksichtigen sind, die die Konvertierung erschweren.

Index-Datei

Index-Dateien werden als Zusatzdateien bei bestimmten Formen der Dateioorganisation (z.B. → ISAM) angelegt. Eine Index-Datei enthält immer nur einen Teil des Datensatzes, den sogenannten Schlüssel. Zusätzlich ist in dieser Datei dann bei dem entsprechenden Schlüsselwert die dazugehörige laufende Satznummer z.B. einer → Random-Datei gespeichert. Dieses Organisationsprinzip ermöglicht das schnelle Auffinden von Informationen.

Interface**(dt. Schnittstelle)**

Eine Schnittstelle ist immer dann erforderlich, wenn zwei Datenverarbeitungsgeräte miteinander verbunden werden sollen. Dabei kann es sich um eine interne Schnittstelle (z.B. zur Koppelung der → Zentraleinheit mit dem → Speicher) oder um eine externe Schnittstelle (z.B. → Zentraleinheit/→ Drucker) handeln.

Es gibt → serielle und → parallele Schnittstellen.

ISAM**Indexed sequential access method****(dt. indexierte sequentielle Zugriffsmethode)**

Es handelt sich um ein spezielles Dateizugriffsverfahren, das die Datensätze in Form von → Bäumen ablegt, so daß der schnelle Zugriff über einen → Schlüssel möglich ist. Unterschied zur → Random-Datei ist, daß bei jener der Zugriff über eine fortlaufende Satznummer erfolgt, bei einer ISAM-Datei aber ein beliebiger Schlüssel Verwendung findet. So kann, wenn das Feld "Name" als Schlüssel definiert ist, der Zugriff z.B. auf den Satz „Müller" erfolgen, ohne daß die Nummer des Satzes bekannt sein muß, in dem der Name "Müller" gespeichert ist.

Object Code**(dt. Objektprogramm oder Maschinenprogramm)**

Im Gegensatz zum → Source Code handelt es sich beim Objektprogramm um eine Zusammenstellung von Bitkombinationen (→ Bit), die bei der Konstruktion der Maschine vom Hersteller als Befehle festgelegt worden sind. Jede Zentraleinheit verwendet einen anderen → Befehlssatz. Diese Befehle sind für den Menschen nicht unmittelbar zu lesen. Erst unter Zuhilfenahme von speziellen Programmen (→ Disassembler) kann der Object Code in einen Quellcode (→ Source Code) zurückverwandelt werden. Wichtig hierbei ist jedoch, daß eine Rückübersetzung nur für in → Assembler geschriebene Programme zu einem eindeutigen Ergebnis führt.

In einer höheren Programmiersprache (→ Cobol, Pascal, Fortran, PL/1, C, Basic) geschriebene Programme aus denen ein Object-Code erzeugt worden ist können nicht eindeutig in ein Programm in der

entsprechenden Programmiersprache zurückübersetzt werden.

Der Object Code stellt also gewissermaßen eine Reduktion der im Quellcode enthaltenen Anweisungen auf die minimale Informationsmenge dar, die notwendig ist, um die im Programm codierten Funktionen für den Computer eindeutig darzustellen.

parallele Schnittstelle

Im Gegensatz zur → seriellen Schnittstelle werden bei der parallelen Schnittstelle alle → Bit's eines Wortes gleichzeitig und nicht nacheinander übertragen. Es sind also mindestens so viele Verbindungsleitungen notwendig, wie Bit's in einem Wort enthalten sind. Parallele Verbindungen benötigen deshalb mehr Verbindungsleitung als serielle Verbindungen. Vorteilhaft ist jedoch, daß die Übertragung unter Verwendung von parallelen Schnittstellen auch wesentlich schneller abläuft als mit seriellen Schnittstellen, da in einem Übertragungszyklus bei einer 8 bit breiten Parallel-Schnittstelle die 8-fache Informationsmenge einer seriellen Schnittstelle, die immer nur ein bit überträgt, übermittelt werden kann.

Parallele Schnittstellen werden daher hauptsächlich für Verbindungen innerhalb eines Computers eingesetzt (z.B. für die Verbindung von externen Speichern (→ Speicher, externer) mit internem Speicher (→ Speicher, interner) während die serielle Übertragung bei größeren Strecken Verwendung findet.

Quellcode

→ Source Code

Random-Datei

In einer Random-Datei kann auf jeden einzelnen Datensatz sofort zugegriffen werden. Dieser sogenannte wahlfreie Zugriff setzt im Gegensatz zum sogenannten sequentiellen Zugriff (→ sequentielle Datei) keine bestimmte Reihenfolge beim Abruf der Datensätze voraus.

Relative Datei

Die Bezeichnung relative Datei ist eine andere Bezeichnung für eine → Random-Datei, die im wesentlichen im Zusammenhang mit der Programmiersprache → Cobol verwendet wird.

RS-232

→ serielle Schnittstelle

Schnittstelle

→ Interface

Sedezimalsystem

→ Hexadezimalsystem