

CROMEMCO

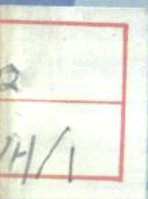
微型计算机

COBOL

程序设计

罗文化 编著

复旦大学出版社



CROMEMCO 微 型 计 算 机

COBOL 程 序 设 计

罗 文 化 编 著

复 旦 大 学 出 版 社

内 容 提 要

本书以ANSI COBOL74为基础,并按照《中华人民共和国国家标准程序设计语言 COBOL》的规定,通俗而详细地介绍了 CROMEMCO 微型计算机 COBOL 语言及其程序设计。为了适应不同读者的需要,本书对程序间通信、位标法的表处理、索引文件和相对文件、以及排序(用算法排序和自动排序)也都作了详细介绍,并有大量的例题。本书可作为大专院校师生、企事业管理人员、计算机程序设计人员学习和使用 COBOL 语言的教材和参考书,也可以作为具有高中以上文化水平的读者自学 COBOL 语言的入门书。

本 COBOL 语言同 IBMPC 和 APPLE II 微型计算机上的 COBOL 语言兼容。

CROMEMCO 微型计算机 COBOL 程序设计

复旦大学出版社出版

新华书店上海发行所发行

复旦大学印刷厂印刷

字数 389 千字 开本 787×1092 1/16 印张 16

1984 年 11 月第一版 1984 年 11 月第一次印刷

印数: 1~35,000

书号: 13253·013 定价: 2.34 元

前 言

微型计算机每年以千万计的数量进入各行各业,对社会影响日益加深,作用越来越大,明显地呈现出了技术革命的重要特征。短短几年来,我国已经有一万多台微型计算机,大大超过了二十多年来大、中、小型计算机的总和。微型计算机更适用于我国各行各业,更容易取得经济效益,有利于我国的技术改造,从而加速四化的过程。

微型计算机已成为计算机发展的主要方向之一,它还具有价廉、多功能、体积小、使用灵活以及容易维护等优点。在计算机日益向研究单位、工厂、机关、学校、商店、服务行业、乃至家庭等领域的普及中,它更具有独特的优越性。因此,微型计算机有着广泛的应用范围和发展前途。

CROMEMCO 微型计算机系统是美国 CROMEMCO 公司近年生产的一种新产品,它已经用于科技计算、工业控制、商业会计、学校教育等各个方面,它的硬件系统采用 Z-80A 大规模集成电路作中央处理机的核心部件,具有158条指令的指令系统,运算速度较快、功能较强,目前已配备的软件系统有 CDOS(Cromemco 磁盘操作系统)、SCREEN(屏幕编辑程序)、EDITOR(文本编辑程序)、Z-80 浮动宏汇编、扩展 BASIC、多用户 BASIC、32K 结构 BASIC、FORTRAN IV、合理的 FORTRAN (RATFOR)又称为结构 FORTRAN、COBOL、DBMS(数据库管理系统)等,还支持 ALGOL,是目前所见八位微型计算机中软件较全、功能较强的微型计算机系统。

本书的内容不完全局限于 CROMEMCO 微型计算机的 COBOL 语言,有一定的通用性,有若干实例程序。因此,本书除可以作为 CROMEMCO 微型计算机 COBOL 语言的教材以外,还可以供初学 COBOL 语言和使用 COBOL 语言的人参考。

由于 CROMEMCO COBOL 编译程序的更新和扩充,CROMEMCO COBOL 编译程序有若干版本,它们之间有些区别,功能不完全相同。本书中所有的实例程序都是用 CROMEMCO COBOL-80 V3.22 编译程序通过编译,其中大多数运行过它们的目标程序,得出了正确的结果。

上海市计算技术研究所裴广生同志审阅了本书初稿,并提出了很好的意见,作者对此谨表谢意。同时也感谢复旦大学计算机科学系李为鑑同志审阅全书,并作了文字修改,并感谢徐广定同志仔细审阅了本书初稿。

本书中不当之处和错误难免,希望读者指正。

编 者

目 录

第一章 绪 论	1
第一节 COBOL 的起源和发展	1
第二节 COBOL 源程序的结构	2
习 题	7
第二章 COBOL 基本概念	8
第一节 字符集.....	8
第二节 数据单位.....	8
第三节 COBOL 字	9
第四节 常 量.....	11
第五节 语法格式.....	13
第六节 标点规则.....	14
第七节 源程序的书写格式.....	14
第八节 流程图符号.....	16
习 题	17
第三章 标识部和环境部	18
第一节 标识部.....	18
第二节 环境部.....	19
习 题	22
第四章 数据部	23
第一节 一般格式.....	23
第二节 文件节.....	23
第三节 工作存贮节.....	53
第四节 连接节.....	54
习 题	55
第五章 过程部	59
第一节 概 述.....	59
第二节 过程部的结构.....	59
第三节 名字的受限.....	61
第四节 MOVE 语句.....	64

第五节	INSPECT 语句	69
第六节	算术语句	73
第七节	GO TO 语句	81
第八节	STOP 语句	82
第九节	ACCEPT 语句	83
第十节	DISPLAY 语句	84
第十一节	PERFORM 语句	85
第十二节	EXIT 语句	93
第十三节	ALTER 语句	94
第十四节	条件表达式和 IF 语句	94
第十五节	OPEN 语句	106
第十六节	READ 语句	107
第十七节	WRITE 语句	108
第十八节	CLOSE 语句	109
第十九节	REWRITE 语句	110
第二十节	STRING 语句	111
第二十一节	UNSTRING 语句	114
第二十二节	DYNAMIC DEBUGGING 语句	119
第二十三节	USE 语句	121
习 题		123
第六章	顺序文件的基本程序设计方法	126
第一节	工资计算程序设计	126
第二节	控制转换处理程序设计	131
实习题一		141
第三节	文件更新处理程序设计	142
实习题二		150
第四节	下标法的表处理程序设计	151
实习题三		159
第七章	程序间通信	161
第一节	概 述	161
第二节	用于被调程序的语言元素	161
第三节	用于调用程序的语言元素	162
实习题四		165
第八章	位标法的表处理	167
第一节	概 述	167
第二节	数据部	167

第三节 过程部.....	172
第四节 表初值的设置.....	190
实习题五	191
第九章 索引文件.....	192
第一节 索引文件的结构和存取方式.....	192
第二节 环境部.....	193
第三节 过程部.....	194
实习题六	205
第十章 相对文件.....	206
第一节 相对文件的结构和存取方式.....	206
第二节 环境部.....	207
第三节 过程部.....	208
实习题七	219
第十一章 排 序.....	221
第一节 概 述.....	221
第二节 用算法排序.....	221
第三节 自动排序.....	226
参考书.....	240
附录 1 CROMEMCO COBOL 结构和特性摘要说明	240
附录 2 文件的组织方式和存取方式.....	241
附录 3 保留字表.....	243
附录 4 COBOL 用的ASCII 码	247
附录 5 I/O 错误处理的一般标志	248
附录 6 COPY 语句	248

第一章 绪 论

第一节 COBOL 的起源和发展

COBOL 是 COMmon BUSiness ORiented Language 的缩写。它是由计算机制造者与用户，特别是由美国政府联合开发的。

为了探讨给商用计算机的程序设计建立通用程序设计语言的必要和可能，于 1959 年 5 月 28、29 日在美国五角大楼 (Pentagon) 召开了一个有私人工商业、政府机关、计算机制造厂商和其他有兴趣部门的代表参加的会议，会议小组同意承担这个课题。这次会议还成立了数据系统语言协会，(CODASYL; CONFERENCE OF DATA SYSTEM LANGUAGES)，进行以不同于任何计算机制造商而对政府部门和私人用户都有利的程序设计语言 COBOL 的开发。

1960 年 4 月，美国政府出版部公布了 CODASYL 提出的第一个 COBOL 60 版本，受到了用户的热烈欢迎和鼓励。COBOL 61 是 COBOL 的第二个版本，1965 年加以修改，得到了用户的普遍承认，为以后 COBOL 的发展打下了基础。

COBOL 是面向过程的事务数据处理用的通用程序设计语言。程序的编写接近于英语，直观性强。它具有结构模块化和自我说明的功能，便于学习和理解，也便于程序交流。它已成为用于数据处理的主要程序设计语言。

其他许多程序设计语言是在实用化以后，再制定标准规格的，而 COBOL 则是先行标准化的，象有排序和制表功能的 COBOL 65，有数据通信功能的 COBOL 69，含有数据库功能的 COBOL 76，含有结构程序设计功能的 COBOL 78 等，都陆续制定了符合各种要求的标准规格。

COBOL 功能模块的划分，随着 COBOL 的发展不断变化，功能不断扩大，功能模块也不断增多。以 1972 年国际标准化组织 (ISO: International Standards Organization) 推荐的 ISO COBOL 72 为例，它分成一个核心和七个功能模块，每个功能模块又按其功能分为 1 到 3 级，如图 1 所示。

核 心	功 能 模 块						
	表 处 理	顺 序 存 取	随 机 存 取	排 序	报 表 编 制	分 段	库
二 级	三 级	二 级	二 级	二 级	二 级	二 级	二 级
	二 级		一 级	一 级	一 级	一 级	一 级
一 级	一 级	一 级	空	空	空	空	空

图1 ISO COBOL 72 模块结构

由图 1 看出, 最小的 COBOL 由核心一级、表处理一级及顺序存取一级构成。

到 1978 年, ISO 推荐的 ISO COBOL 78 (即美国国家标准 X3.23-1974), 已发展到一个核心和十一个功能模块, 如图 2 所示。

核 心	表处理	功 能 模 块									
		顺 序 I-O	相 对 I-O	索 引 I-O	排 序 合 并	报 表 编 制	程 序 分 段	库	排 槽	程序间 通 信	通 信
二 级	二 级	二 级	二 级	二 级	二 级	一 级	二 级	二 级	二 级	二 级	二 级
一 级	一 级	一 级	一 级	一 级	一 级	空	一 级	一 级	一 级	一 级	一 级
一 级	一 级	一 级	空	空	空	空	空	空	空	空	空

图2 ISO COBOL 78 模块结构

由图 2 看出, 有九个功能模块以空作为最低一级, 在所有情况下, 较低一级都是同一功能模块中较高级的真子集, 最小的 COBOL 由核心一级、表处理一级及顺序 I-O 一级构成, 完全标准的 COBOL 由最高级核心和十一个功能模块的最高级构成。

到目前为止, 除了军事部门外, 据不完全统计, 国际上大约有 70% 以上的程序是用 COBOL 语言写的, 使用 COBOL 语言的人最多。

COBOL 能普及和不断发展的原因大致如下:

COBOL 是受到美国政府和最多用户共同支持的程序设计语言, 任何型号的计算机上如没有配备 COBOL, 将被用户怀疑其性能, 从而促使计算机厂商发展 COBOL。

COBOL 得到了 CODASYL 和 ISO 的支持, 使得 COBOL 能用于数据处理的各个领域。

COBOL 不仅适合用户的当前需要, 而且稍加修改, 它就能适应用户的将来需要。

CROMEMCO COBOL 是以美国国家标准程序设计语言 COBOL X3.23-1974 (ANSI-74) 为基础设计的, 关于它的结构和特性摘要说明见附录 1。

第二节 COBOL 源程序的结构

为了使读者一开始就能对 COBOL 源程序结构有个大概的了解, 本节给出一个简单实例的两种型号计算机上的 COBOL 源程序, 实例如下:

要求由卡片输入机输入卡片文件, 计算每个职工的实发工资并记到磁带上, 用日本日立 M 系列计算机上的 COBOL 语言书写源程序。(见左图)

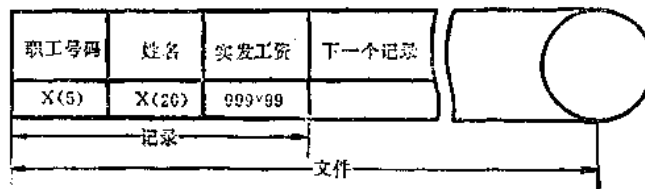
输入文件格式: 工资文件…卡片, 标号省略。(见下图)



职工号码	姓名	基本工资	附加工资	扣除工资	空白
X(5)	X(20)	999v99	99v99	99v99	.X(42)

记录

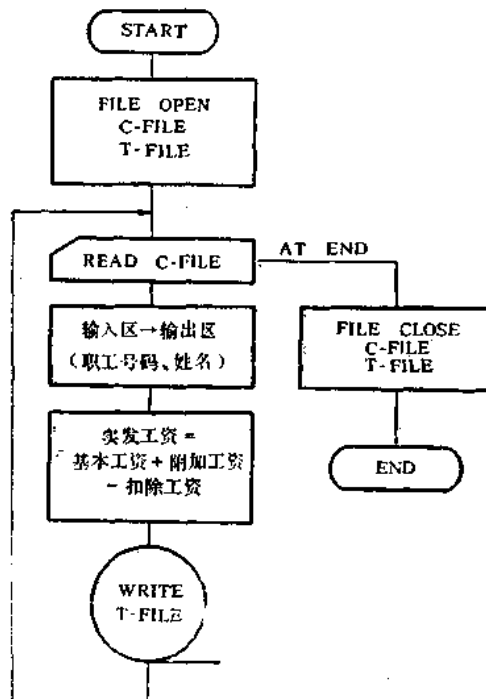
输出文件格式: 实发
文件 工资文件…磁带, 25 个
记录/块, 带标准标号。



实发工资由下式计算:

实发工资 = 基本工资 + 附加工资 - 扣除工资。

flow chart (流程图)



Coding (编码)

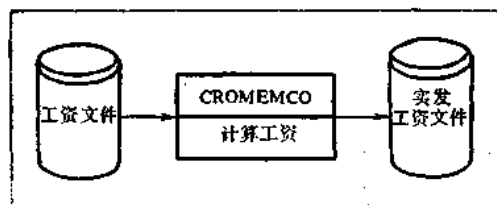
000100 IDENTIFICATION DIVISION.
 000200 PROGRAM-ID. PAYROLL.
 000300 AUTHOR. LO-WEN-HUA.
 000400 INSTALLATION. HITACHI COMPUTATION CENTER TOKYO.
 000500 DATE-WRITTEN. 77.05.26.
 000600 DATE-COMPILED. TODAY.
 000700 SECURITY. NO.
 000800 REMARKS. EXERCISES.
 000900 ENVIRONMENT DIVISION.
 001000 CONFIGURATION SECTION.
 001100 SOURCE-COMPUTER. HITAC-M170.

```

001200 OBJECT-COMPUTER. HITAC-M160.
001300 INPUT-OUTPUT SECTION.
001400 FILE-CONTROL.
001500         SELECT C-FILE ASSIGN SYSOO4-UR-CR-S.
001600         SELECT T-FILE ASSIGN SYSOO5-UT-MT-S.
001700 DATA DIVISION.
001800 FILE SECTION.
001900 FD C-FILE
002000         RECORDING MODE F LABEL RECORD OMITTED
002100         DATA RECORD C-REC.
002200 01 C-REC.
002300     02 C-NO         PICTURE X(5).
002400     02 C-NAME        PICTURE X(20).
002500     02 C-HONKYU      PICTURE 999V99.
002600     02 C-KAKYU       PICTURE 99V99.
002700     02 C-KOJO        PICTURE 99V99.
002800     02 FILLER        PICTURE X(42).
002900 FD T-FILE
003000         RECORDING MODE F LABEL RECORD STANDARD
003100         BLOCK 25 RECORDS DATA RECORD T-REC.
003200 01 T-REC.
003300     02 T-NO         PICTURE X(5).
003400     02 T-NAME        PICTURE X(20).
003500     02 T-SHIKYU      PICTURE 999V99.
003600 PROCEDURE DIVISION.
003700 OPEN-ROUTINE SECTION.
003800 O.   OPEN INPUT C-FILE OUTPUT T-FILE.
003900 MAIN-ROUTINE SECTION.
004000 M.   READ C-FILE AT END GO TO END-ROUTINE.
004100         MOVE C-NO TO T-NO
004200         MOVE C-NAME TO T-NAME
004300         COMPUTE T-SHIKYU = C-HONKYU + C-KAKYU - C-KOJO
004400         WRITE T-REC GO TO M.
004500 END-ROUTINE SECTION.
004600 E.   CLOSE C-FILE T-FILE
004700 STOP RUN.

```

将本实例的源程序用 CROMEMCO COBOL 语言书写, 由于 CROMEMCO 微型计算机没有卡片机和磁带机, 有磁盘机, 故用磁盘代替卡片和磁带。本实例改为“读磁盘文件, 计算每个职工的实发工资并记到磁盘上”。



输入文件格式：工资文件…磁盘，带标准标号。

职工号码	姓名	基本工资	附加工资	扣除工资	下一个记录	
X(5)	X(20)	999v99	99v99	99v99		

记录
 文件

输出文件格式：实发工资文件…磁盘，带标准标号。

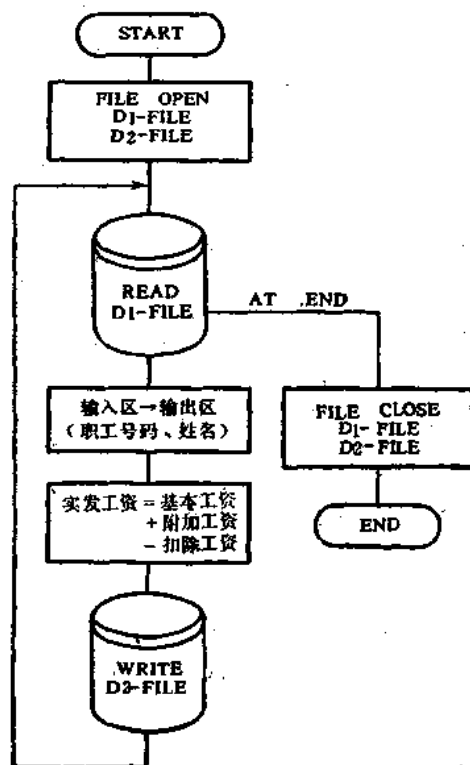
职工号码	姓名	实发工资	下一个记录	
X(5)	X(20)	999v99		

记录
 文件

实发工资由下式计算：

实发工资 = 基本工资 + 附加工资 - 扣除工资

flow chart (流程图)



Coding (编码)

001010 IDENTIFICATION DIVISION.
001020 PROGRAM-ID. SAMPLE.
001030 AUTHOR. LO WEN HUA.
001040 ENVIRONMENT DIVISION.
001050 CONFIGURATION SECTION.
001060 SOURCE-COMPUTER. CROMEMCO SYSTEM THREE.
001070 OBJECT-COMPUTER. CROMEMCO SYSTEM THREE.
001080 INPUT-OUTPUT SECTION.
001090 FILE-CONTROL.
001100 SELECT D1-FILE ASSIGN DISK.
001110 SELECT D2-FILE ASSIGN DISK.
001120 DATA DIVISION.
001130 FILE SECTION.
001140 FD D1-FILE LABEL RECORD STANDARD VALUE OF FILE-ID
001150 "B:D1-FILE.ASM" DATA RECORD D1-REC.
001160 01 D1-REC.
001170 02 D1-NO PICTURE X(5).
001180 02 D1-NAME PICTURE X(20).
001190 02 D1-BPAY PICTURE 999V99.
001200 02 D1-APAY PICTURE 99V99.
001210 02 D1-SPAY PICTURE 99V99.
001220 FD D2-FILE LABEL RECORD STANDARD VALUE OF FILE-ID
001230 "B:D2-FILE.ASM" DATA RECORD D2-REC.
001240 01 D2-REC.
001250 02 D2-NO PICTURE X(5).
001260 02 D2-NAME PICTURE X(20).
001270 02 D2-PAYOFF PICTURE 999V99.
001280 PROCEDURE DIVISION.
001290 1. OPEN INPUT D1-FILE OPEN OUTPUT D2-FILE.
001300 2. READ D1-FILE RECORD AT END CLOSE D1-FILE D2-FILE
 STOP RUN.
001310 MOVE D1-NO TO D2-NO
001320 MOVE D1-NAME TO D2-NAME
001330 COMPUTE D2-PAYOFF = D1-BPAY + D1-APAY - D1-SPAY
001340 WRITE D2-REC GO TO 2.

源程序的简要说明:

第一个源程序中, 001500 行中的 C-FILE 表示工资文件名, SY5004-UR-CR 表示分

配给 C-FILE 的卡片输入机, S 表示 C-FILE 是顺序文件, 001600 行中的 T-FILE 表示实发工资文件名, SY5005-UT-MT 表示分配给 T-FILE 的磁带机, S 表示 T-FILE 是顺序文件。

第二个源程序中, 001100 行中的 D1-FILE 表示工资文件名, DISK 表示分配给 D1-FILE 的磁盘机。001110 行中的 D2-FILE 表示实发工资文件名, DISK 表示分配给 D2-FILE 的磁盘机。

由这个实例的两个源程序可以看出, COBOL 的源程序总是由四个部构成, 它们的严格顺序如图 3 所示, 这种源程序的结构和任何型号计算机上的 COBOL 源程序结构都相同。

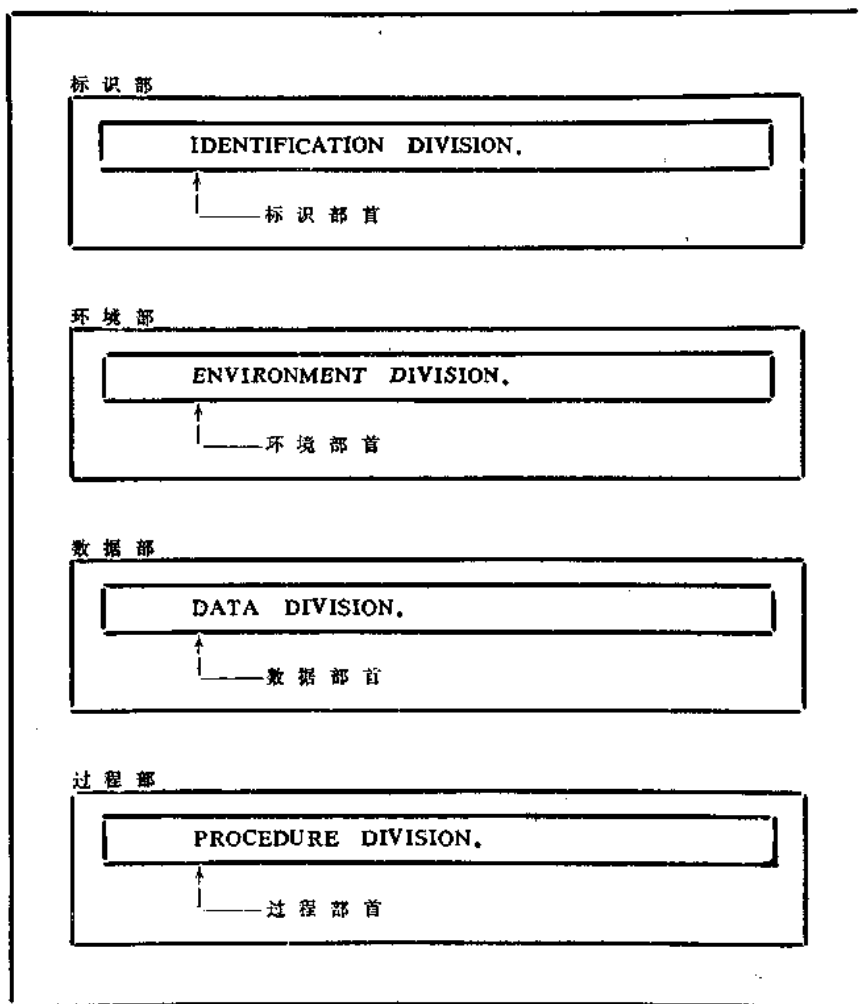


图3 COBOL 源程序的结构

习 题

1. 试说明 COBOL 源程序由哪几个部组成以及它们的排列顺序, 并写出每个部首的英文字。
2. 你学习本章以后, 有哪些问题?

第二章 COBOL 基本概念

第一节 字符集

COBOL 的字符集 (character set) 由下面的 52 个字符构成:

A~Z 字母

 空格

0~9 数字

专用符号:

+	加号	,	分号
-	减号或连字符	.	句号或小数点
*	乘号或星号	"	双引号
=	等号	'	单引号
>	大于号	(	左括号
<	小于号	)	右括号
\$	货币符号 (美元号)	/	除号或斜杠
,	逗号		

在非数值常量、注解行及注解项中, COBOL 字符集可以扩展为包括计算机的整个字符集 (即计算机所能接受的所有字符的集合), 字符是 COBOL 语言的最基本且不可再分的单位。

为了形象起见, 用□表示空格, 但必须注意□并不是字符集中的真正字符。

第二节 数据单位

(一) 数据项 (Data item)

数据项分为初等项和组合项、数据项又简称为项。

1. 初等项 (Elementary item)

初等项由一个或多个字符构成, 它是不能再细分的数据项。

2. 组合项 (Group item)

组合项是可以再细分的数据项, 它由一个或多个初等项构成, 它除了包含初等项外, 还可以包含组合项。组合项最多可以含有 4096 个字符。

(二) 记录 (Record)

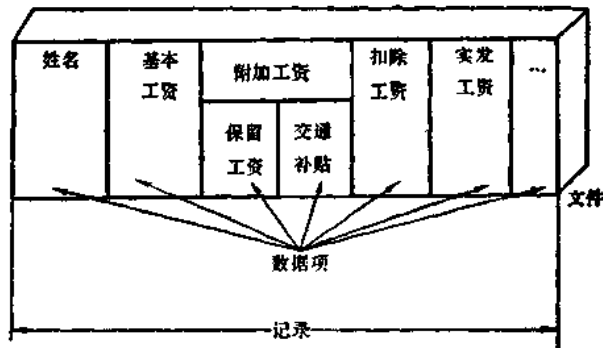
记录由一个或多个逻辑上相关的数据项构成, 它是构成文件的最小逻辑单位。

(三) 文件 (File)

文件是逻辑上相关的记录的一个集合，文件总是按某种组织方式存放在存贮介质（如磁盘）中。

数据项、记录及文件，各有一个名字以供识别，它们的使用者只要给出它们的名字就可以存取它们，不必知道它们所存放的物理位置，所以对它们总是“按名字”存取的。

例 1.



关于文件的组织方式和存取方式，见附录 2。

第三节 COBOL 字

COBOL 字是一个不超过 30 个字符的字符串，它分为保留字和用户定义字，在源程序中，一个 COBOL 字仅仅属于这两种字之一，这两种字是不相交的。

一、保留字 (Reserved word)

保留字是 COBOL 编译程序预先登录的 COBOL 字，它们代表固定的意义，它们可以在 COBOL 源程序中使用。但它们不能作为用户定义字出现在源程序中，使用者不能随便使用，使用时不能写错，保留字可分成下面六种，全部保留字列在附录 3 中。

(一) 基字 (key word)

基字是语法格式中必须出现且下方划线的保留字，有三种：

- ① 动词，如 ADD、MOVE 和 READ 等等。
- ② 在一个语句和描述款格式中必须出现的字，如 TO 和 FROM 等等。
- ③ 有特别功能意义的字，如 SECTION 和 SELECT 等等。

(二) 任选字 (optional word)

在语法格式中，下方没有划线的保留字叫任选字，任选字的出现与否并不改变它在其中出现的源程序的语义，如果出现，也仅仅是为了增加源程序的清晰度和易读性，如 ADVANCING 和 AT 等等，任选字在源程序中是否出现，由用户决定。

(三) 连接符 (Connective)

有三种连接符：

- ① 将数据名、记录名、条件名或段名和它们的限定符相连接的受限连接符 OF 和 IN。
- ② 构造条件使用的逻辑连接字 AND 和 OR。

③ 也将连接两个或多个相邻操作分量的连接符, (分隔符逗号) 和; (分隔符分号) 作为连接符。

(四) 专用寄存器 (Special registers)

某些保留字用来命名和引用专用寄存器, 专用寄存器是编译程序生成的某些存储区, 它们的主要用途是存放特定的信息, 这些专用寄存器包括 LINAGE-COUNTER, LINE-COUNTER, PAGE-COUNTER 以及 DEBUG-ITEM。

(五) 象征常量字 (Figurative Constant word)

某些保留字用来命名和引用特殊的常量值, 如 SPACE 和 HIGH-VALUE 等等, 这些保留字列在象征常量中。

(六) 专用字符字 (Special-character word)

专用字符字是指用作算术运算符和关系运算符的那些保留字。如 + 和 > 等等。

二、用户定义字 (User-defined word)

用户按照字的构成规则定义所需要的字, 不能和保留字相同, 用户定义字数目不限, 需要多少就定义多少。

(一) 用户定义字的构成规则

用户定义字中的每个字符选用 A~Z, 0~9 和 - 等 37 个字符之一, 但 - 不能作为第一个或最后一个字符出现。

例1. 用户定义字

如果用户定义字不唯一时, 在过程部中引用该字时, 必须通过“名字的受限”, 使其唯一, 这将在过程部中介绍。

(二) 用户定义字可用作下列名字

1. 数据名 (Data-name)

数据名是对数据项命名的名字, 它必须以字母开头。

2. 记录名 (Record-name)

记录名是对记录命名的名字, 它必须以字母开头。

3. 文件名 (File-name)

文件名是对文件命名的名字, 它必须以字母开头, 文件名必须唯一。

4. 条件名 (Condition-name)

条件名是对被赋以一个值, 一组值或一个值域命名的名字, 这些值或值域必须落在一个数据项可以具有的值的完全集合中, 数据项本身叫做条件变量, 条件名必须以字母开头。

5. 助忆名 (Mnemonic-name)

助忆名是用来代替一个设备或设备所处状态的名字, 它们在环境部的 SPECIAL-NAMES (专用名) 段中定义, 助忆名必须以字母开头, 助忆名必须唯一。

6. 过程名 (Procedure-name)

过程名包括过程部中的节名和段名, 如果过程名全部由数字构成, 则它最多不能超过 18 位, 过程名的其他情况, 必须满足用户定义字的构成规则。

正	确	错	误
C-REC		- CREC	
REC		- C-REC	
D1-FILE		REC-	
123		D1-FILE-	
OPEN-ROUTINE-1		123-	