

微型、小型计算机 应用软件彙编

Application Software



中国科学院成都计算机
应用研究所情报室主编

《微型、小型计算机应用软件汇编》

目 录

数据处理与采集

数据处理系统设计及应用实例.....	(1)
微型机用于高炉称重系统中的数据处理方式.....	(15)
MDR—Z80 机随机信号处理方面的应用.....	(23)
微处理机数据采集处理系统软件包设计.....	(27)
用微计算机做结构的模态试验.....	(31)
微电脑在银行储蓄数据处理中的应用.....	(35)
OB法的计算机处理研究.....	(42)
介绍一个实时数据压缩程序.....	(46)
多元线性回归方法的TI—59程序及应用.....	(61)
疲劳载荷数据处理的山峰谷法及其在微计算机上的实现.....	(67)
微型机数据采集系统的应用软件.....	(70)
一种实用显微密度的计算机系统.....	(76)

图形、图象处理

TRS—80微型机通用曲线输出程序包.....	(81)
用微型计算机绘制材料的特性曲线.....	(87)
BCM—S68000微机的绘图程序.....	(90)
用BASIC语言控制绘图仪.....	(94)
一种简单的微机图象显示修改系统.....	(97)
一个微机控制小型图象采集系统.....	(99)
字幕图象的电脑控制.....	(101)
以BCM—80为基础的汉字处理软件.....	(104)

网络、通信

PERT微型计算机软件系统.....	(107)
--------------------	---------

微处理机在扩展群同步码中的应用.....	(120)
利用微型电子计算机诊断通信设备故障的方案.....	(126)
包裹细分机分栋系统的程序设计.....	(136)
微型计算机在长途自动电话测试中应用.....	(141)
微机在长途电话自动计费上的应用.....	(144)
微处理机用于用户电报交换机的计费设备.....	(148)
中文电报翻译软件的设计与分析.....	(150)
电子计算机在邮袋分栋中的应用.....	(153)
广播发射机微型机监控装置.....	(156)
采用LTS——80一位机控制的电台音周自动切换装置.....	(159)
微型计算机控制的工业电视系统.....	(165)

机械、轻、化工

MC——APT简明会话型微机辅助编程系统.....	(168)
用微型计算机进行数控机床程序坐标变换.....	(177)
冒口计算的程序编制.....	(181)
TP——801微计算机在橡胶制品硫化工艺上的应用.....	(187)
带微处理机的丝茧含水率测试仪.....	(189)
TRS——80微机在多相反应催化剂评选装置中的的应用.....	(192)
乙糖煮制过程的微型计算机控制系统.....	(195)

交通运输

单板计算机用于城市交通信号管理.....	(200)
电子轨道衡用计算机系统.....	(201)
介绍计算交流轨道电路的一种程序.....	(206)
天文导航微型计算机软件系统.....	(209)

电子、电路

计算印刷电路板密度的程序.....	(216)
在TRS——80微机上用网络拓扑理论对稳态交流电路的分析.....	(220)
用单板机产生任意波形.....	(226)
单板机对波形存贮器的开发应用.....	(231)

水利、电力

用BASIC语言编制调洪演算程序.....	(237)
-----------------------	---------

MZ—80B微型计算机在故县水库洪水预报中的应用	(241)
微型计算机在水利灌溉中应用	(246)
应用于梯级水电站远方集中监控系统的微处理机与CRT实时时钟联调试验	(252)
计算弧门启闭机高度指示器刻度值的BASIC程序	(258)
用于大坝观测数据采集和处理的微计算机系统的在线应用软件	(262)
高压配电线路等值电阻计算程序	(266)
PC—1500 计算机在评价汽轮机叶片振动特性中的应用	(273)

农 业

应用微型电子计算机合理选配农业机械的基本方法	(279)
up控制的农机物资管理供应系统	(284)
用计算机优选作物抗灾稳产种植比例的方法探讨	(290)
电子计算机在家蚕商种上的应用	(293)

共用程序

二—十进制浮点算术运算程序	(297)
Z80微型机四字节浮点运算程序——加减法和乘法子程序	(308)
微型机浮点多字节解释系统及其子程序库的编制方法	(311)
RAM测试程序	(329)
M6800并行接口测试程序	(337)
TRS—80BASIC II 状态下内存与磁带直接交换信息的程序以及JKL命令	(354)
实现两个十六位数相乘的小程序	(360)
介绍一种实用的数制转换软件编程方法	(361)
微机现场保护和系统的重入	(363)
硬件系统实时检查程序与人机联系程序	(365)
逐步聚类分析程序	(368)
仪器固件程序模仿	(375)

数据处理与采集

文献名称: 数据处理系统设计及应用实例

作者: 云南大学数学系软件组

使用机型: 小型、微型机

使用语言: BASIC

用途: 数据处理

部份内容:

基本概念

一、域

由一个或多个数字位组成,是文件中不可分割、可使用的最小单位。域的长度是按位计算。

二、记录

是有关域按一定形式排列起来的集合。记录的长度规定为有关域长度之和。一个记录通常由三部分组成:第一部分是关键字,它是记录中的一个域,用来唯一地标识一个记录。第二部分是数据,由若干域组成。第三部分是控制部分,指出记录的长度或指出下一个记录的位置。

三、块记录

将两个或多个记录连在一起,组成一块记录,读写时以块记录为单位进行读写。

四、文件

若干个相同结构的记录的集合叫做文件。

例如某工厂职工情况登记表构成一个文件。

此文件中每个职工情况是一个记录,每一个记录由五个域组成,职工号、姓名、职业、性别、婚否,分别是五个域的域名。

五、文件的类型

文条的类型很多,这里仅就几种常用的文件类型给预介绍。常用的文件类型有顺序文件、索引顺序文件、随机文件和倒排文件。下面依次对这四种文件的组织和处理进行介绍。

顺序文件

顺序文件是最简单而又经常使用的一种文件形式。

一、顺序文件的组织

记录是连续地存放在存贮设备上,除了第一个记录以外,每个记录有唯一的前任记录,除了最后一个记录外,每个记录有唯一的后继记录。若记录的关键字是有序排列则构成有序顺序文件;若记录的关键字是无序排列则构成无序顺序文件。

例如,下面文件是有序顺序文件组织。

职工号	姓名	职业	性别	婚否
2	王五	车工	女	婚
5	李四	钳工	男	婚
29	张三	电工	男	婚
36	赵六	车工	女	否

二、顺序文件的处理

文件的处理包括两个内容：一是检索即查询一个记录，二是修改，而修改又包括删去一个记录，改变一个记录中的某些域值和搜入一个新记录。

什么样的文件组织就决定了什么样的文件处理方法。顺序文件组织就决定了按顺序处理文件的方法。顺序文件处理在时间上和空间上都是效率低的。例如要检索和修改一个记录，都要存取它前面的所有记录。怎样提高顺序文件处理的效率呢？批处理方法是比较好的方法。

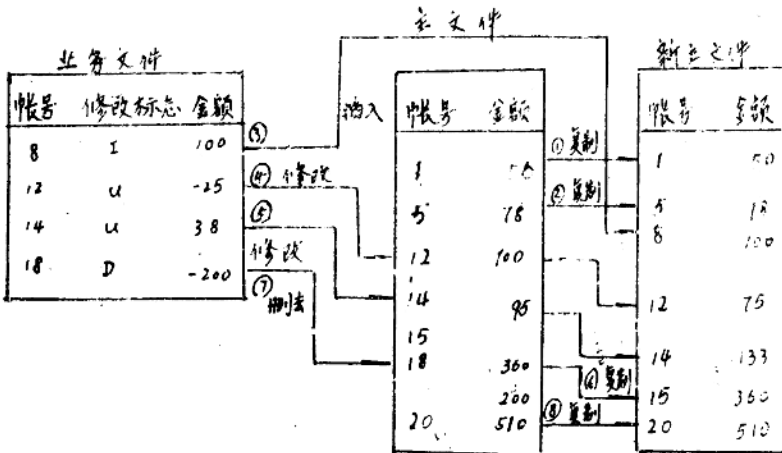
通常进行处理时要建立三个文件：一个是主文件即要修改的原始文件；一个是业务文件，将所有要修改的记录集中在一起形成一个暂时的业务文件；第三个是新主文件，

当业务文件足够大时，用业务文件去修改主文件以形成一个新主文件。

注意：这三个文件都要求按关键字排序。

下面通过一个例子来看批处理的全过程。

例1. 一个银行的账目文件，其中主文件记着各储蓄户存款的金额，每个储蓄户是一个记录，每个记录有二个域：账号域、金额域，以账号域为关键字，记录按关键字从小到大排列。每天顾客来银行取款和存款集中在一个业务文件中，业务文件中每个记录有三个域：账号、修改标记、金额，当业务文件足够大时，就用业务文件去修改主文件，得到一个新主文件，下一次新主文件当作主文件又称为被修改的文件。



上面业务文件中: I 表示插入一个记录;

U 表示修改一个记录;

D 表示删去一个记录。

-25表示取款25元, 38表示存款38元(即正数表示存款数, 负数表示取款数)。

下面用DJS-130扩展BASIC给出上面例子的算法。

创建主文件

```
14 OPEN FILE ( 1 , 1 ),
'MASTER'
15 PRENT 'N' = ' , 'M = '
20 INPUT N,M
25 IF N = -9999 THEN 45
30 PRINT
35 WWRITE FILE(1),N,M
40 GOTO 20
45 CLOSE FILE(1)
50 END
```

创建业务文件

```
5 DIM C$(1)
10 OPEN FILE(2,1),
'TRANSACTION'
15 PRINT 'N1 = ' , 'C$ = ' , 'M1 = '
20 INPUT N1,C$,M1
25 IF N1 = -9999 THEN 45
30 PRINT
35 WRITE FILE(2),N1,C$,M1
40 GOTO 20
45 CLOSE FILE(2)
50 END
```

以业务文件去修改主文件程序

```
6 DIM C$(1)
10 OPEN FILE(1,3), 'MAaTER'
15 OPEN FILE(2,3),
'TRANSACTION'
```

```
20 OPEN FILE(3,1), 'NEW'
25 IF EOF(1) THEN 850
35 READ FILE(1),N,M IF
    EOF(2) THEN 800
40 READ FILE(2),N1,C$,M1
45 IF N >= N1 THEN 70
50 WRITE FILE(3),N,M
55 IF EOF(1) THEN 860
60 READ FILE(1),N,M
65 GOTO 45
70 IF N=N1 THEN 110
75 IF C$='I' THEN 90
80 PRINT 'ERROR RECORD'
    N1,C$,M1
85 STOP
90 WRITE FILE(3),N1,M1
95 IF EOF(2) THEN 800
10 GOTO 40
110 IF C$='D' THEN 25
115 IF C$='U' THEN 130
120 GOTO 80
130 LET M=M+M1
135 WRITE FILE(3),N,M
140 GOTO 25
805 WRITE FILE(3),N,M IF
    EOF(1) THEN 90
810 READ FILE(1), N,M
815 GOTO 800
850 IF EOF(2) THEN 900
855 READ FILE(2),N1,C$,M1
860 IF C$ <> 'I' THEN GOTO 80
865 WRITE FILE(3),N1,M1
870 GO TO 850
900 CLOSE FILE(1)
905 CLOSE FILE(2)
910 CLCSE FILE(3)
915 END
```

索引顺序文件

假若我们需要在顾客信用文件中，查询一个顾客的信用情况，即查看上月该顾客是否付清欠账，以决定本月是否能让该顾客欠账购买物品，这就需要在售货点终端设备上询问该顾客信用情况，且要求回答要快，否则顾客将不耐烦，然而顺序文件不能适应这种在线询问，因为它费时太长，怎样解决这个问题呢？索引顺序文件能较好地解决这个问题。

问题。

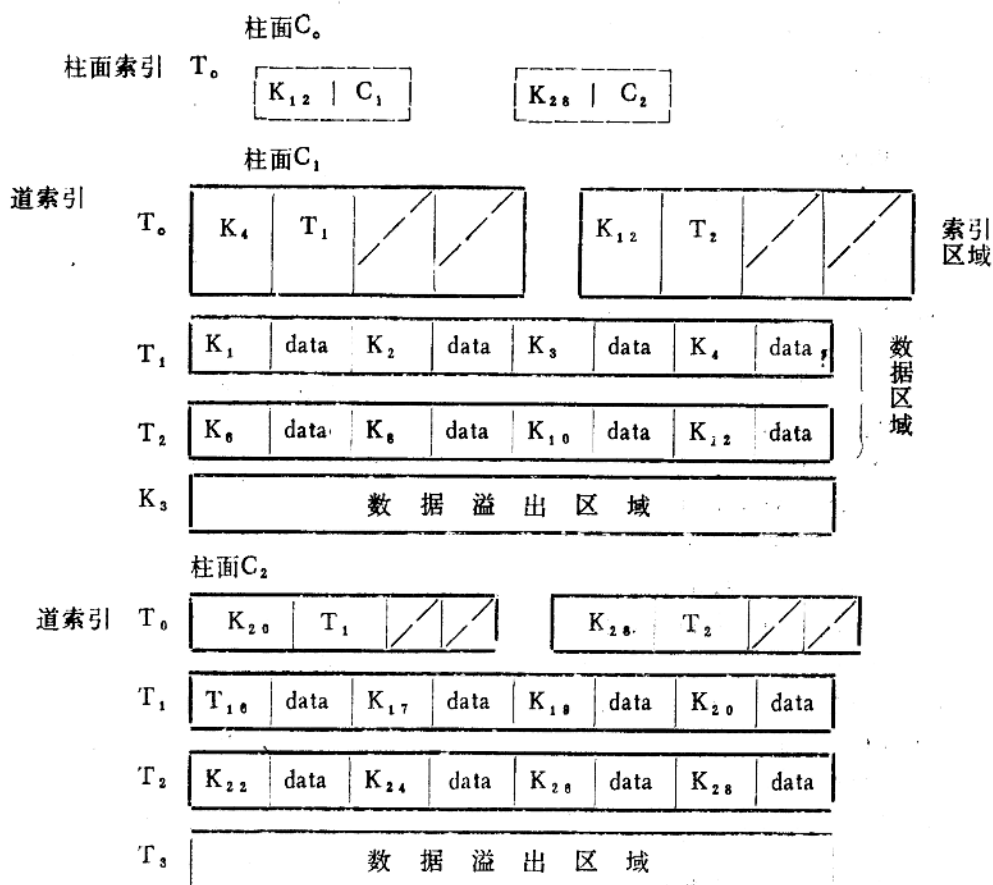
一、索引顺序文件的组织

索引顺序文件是一个具有索引的顺序文件，索引使得能较快地存取一个记录。

一个索引文件通常由三个区域组成：

一是索引区域包括柱面索引、道索引、二是数据区域；三是数据溢出区域。

假设每个框面有四道，每道存四个记录，如图2所示。



为简化起见，假设：

每个记录有两个域：关键字Kay和数据data。

每个溢出记录有三个域：关键字Key、数据data和指针point。

每个块记录由两个记录组成，且占一道。相邻二个块记录，分配在同一柱面上，以减少查找时间。

道索引有二个索引记录，每个索引记录有四个域。

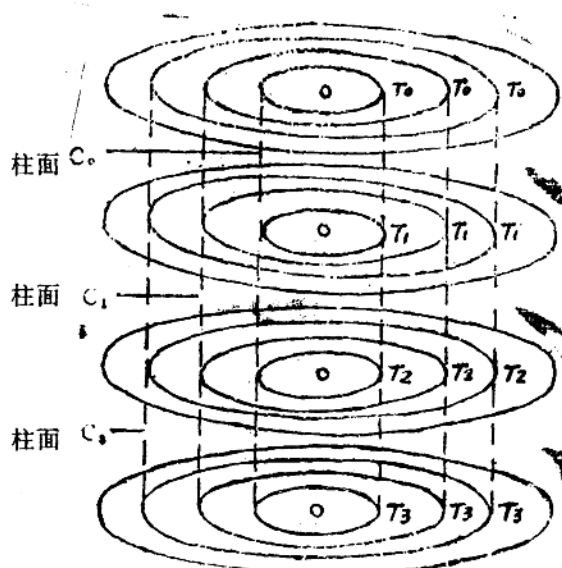


图 2

第一域 Key域放该块记录中最大的 Key。

第二域 放该块第一个记录的地址。

第三域 放该道第一个溢出记录的key°

第四域 放该道最后一个溢出记录的地址。

假设一个索引顺序文件建立时,情况如图 所示,这时数据区已充满记录,溢出区是空白,索引记录中的第三、四域是空的。

二、索引顺序文件的处理

1. 检索

找 Key=17 的记录

(1) 将柱面索引读入内存

查柱面索引 $K_{17} > K_{12}, K_{17} \leq K_{28}$ 。因此,知道 K_{17} 在柱面 C_2 中。

(2) 将柱面 C_2 的道索引读入内存

道 索 引 T_0

K_{20}	T_1	/	/	K_{28}	T_2	K_{28}	$T_3/1$
----------	-------	---	---	----------	-------	----------	---------

T_1

K_{10}	data	K_{17}	data	K_{19}	data	K_{20}	data
----------	------	----------	------	----------	------	----------	------

T_2

K_{22}	data	K_{24}	data	K_{26}	data	K_{28}	data
----------	------	----------	------	----------	------	----------	------

数据区		溢出区	
Key	PTR	Key	PTR

第一域 第二域 第四域 第四域

查道索引 $K_{17} < K_{20}$, 知道 K_{17} 在第一道。

(3) 将柱面 C_2 第一道的块记录读入内存。

顺序检索该块记录

$K_{17} < K_{10}, K_{17} = K_{17}$

(4) 读出该记录

2. 插入记录

(1) 插入关键字为 K_{28} 的一个记录。

首先读柱面索引到内存, $K_{28} > K_{12}$, $K_{28} \leq K_{28}$ 知道要插在柱面 C_2 中。其决读 C_2 的道索引到内存, 比较 $K_{28} > K_{20}$, $K_{28} \leq K_{28}$, 则知道要插入柱面 C_2 中的第二道, 这时将该道中关键字大于 25 的记录顺序往后移, 使关键字 K_{28} 的记录溢出, 存放到溢出区, 此时将第二道索引记录的第一域改为 K_{28} , 第三域添写 K_{28} , 第四域写 $T_3/1$ 。

这时柱面 C_2 各磁道情况如下:

T ₃	K ₂₈	data	/		
----------------	-----------------	------	---	--	--

(2) 插入K₁₈: 柱面C₂各磁道情况如下:

T ₀	K ₁₈	T ₁	K ₂₀	T ₃ /2	K ₂₈	T ₂	K ₂₈	T ₃ /1	
T ₁	K ₁₆	data	K ₁₇	data	K ₁₈	data	K ₁₉	data	
T ₂	K ₂₂	data	K ₂₄	data	K ₂₅	data	K ₂₆	data	
T ₃	K ₂₈	data	/	K ₂₀	data	/			

(3) 插入K₂₃, 柱面C₂各磁道情况如下

T ₀	K ₁₈	T ₁	K ₂₀	T ₃ /2	K ₂₅	T ₂	K ₂₈	T ₃ /3	
T ₁	K ₁₆	data	K ₁₇	data	K ₁₈	data	K ₁₉	data	
T ₂	K ₂₂	data	K ₂₃	data	K ₂₄	data	K ₂₅	data	
T ₃	K ₂₈	data	/	K ₂₀	data	/	K ₂₅	data	T ₃ /1

3. 删去 只需要找到要删去的记录位置, 在该记录的关键字处作删去标记, 并不真正移动数据或修改指针。

经过多次搜入记录和删除记录后, 文件结构变动很大, 大量进入溢出区, 而基本区又浪费很多空间, 所以通常要周期地整理索引顺序文件, 即把所有记录读入内存, 重新排列, 复制成一个新的索引顺序文件, 使它尽量填满数据区, 而空出溢出区。

实例, 某另售商店为顾客建立了信用文件, 当顾客在店欠账购买物品时, 售货员首先在终端设备上查询该顾客的信用情况, 即查看上月该顾客欠账多少, 以决定本月是否继续给顾客欠账购买物品。

用DJS-130机磁盘BASIC语言写出程序

key	point			
15	1	1	Wang	0
60	6	2	Chan	-100
122	11	3	Li	-50
200	16	4	Zhou	-10
		5	Feng	2
		6	Liu	0
		7	Andy	0
		8	B.A.	0
		9	Red	0
		10	Ann	0
		11	29	-13
		12	hel	-123
		13	joe	-100
		14	plli	-70
		15	Tom	0
		16	jok	0
		17	ipw	0
		18	irvr	-70
		19	son	-50
		20		
		21		
		22		
		23		

1、创建索引

```
20 OPEN FILE ( O, O ),  
    "DIRECTORY"  
25 LET I = 1  
30 PRINT "KEY = ", "POINT = "  
40 INPUT K, P  
50 IF K = -9999 THEN 90  
60 PRINT  
70 WRITE FILE ( O, I ), K, P  
75 LET I = I + 1  
80 GOTO 40  
90 CLOSE FILE ( O )
```

2、创立主文件

```
10 DIM A$( 1 )  
15 LET I = 1  
20 OPEN FITE(1,0), "MASTER"  
30 PRINT "NUMBER = " I "NAME  
    IS", "CHARGE = "  
40 INPUT N, A$, C  
50 IF N = -9999 THEN 90  
60 PRINT  
70 WRITE FILE ( 1, I ), N, A$, C  
75 LET I = I + 1  
80 GOTO 40  
90 CLOSE FILE ( 1 )
```

3、检索程序

```
10 OPEN FILF ( O, O ),  
    "DIRECTORY"  
20 LET I = 1
```

```
30 PRINT "ENTER EMPLOYEE  
    NUMBER = "  
40 INPUT N  
45 IF N = -9999 THEN 230  
50 IE EOF ( O ) THEN 100  
60 READ FILE ( O, I ), K, P  
70 IF N <= K THEN 120  
80 LET I = I + 1  
90 GOTO 50  
100 PRINT "EMPTOYEE NUMBER  
    ISNOT FOUND"  
110 GOTO 20  
120 OPTN FILE ( 1, O ),  
    "MASTER"  
130 IF EOF ( 1 ) THEN 180  
140 READ FILE ( 1, P ), N1, A$, C  
150 IF N = N1 THEN 210  
160 LET P = P + 1  
170 GOTO 130  
180 PRINT "THERE IS NOT THE  
    EMPLOYEE NUMBER"  
190 CLOSE FILE ( 1 )  
200 GOTO 20  
210 PRINT "EMPLOYEE RECORD  
    IS", N1, A$, C  
220 GOTO 190  
230 PRINT "SYSTEM IS STOP  
    FOT NUMBER = ", N  
240 CLOSE  
250 STOP  
260 END
```

随机文件

随机文件的组织比索引顺序文件简单，可以在线检索，也可以进行批处理，文件中的记录不要求排序。

随关文件适用于关键字分布范围广，数字相差比文件记录个数大得多的情况。

一、文件的组织和处理

用杂凑法组织文件，其思想是根据关键字来计算记录地址，即建立一个函数 $f(x)$ ，将

关键字K代入f(x)中得

$$f(K) = A$$

f(x)是哈希函数，A是地址，k是关键字。

哈希函数完成将关键字k转换成地址A的作用。

这里指出可能重复，即 $K_i \neq K_j$ 但

$$f(K_i) = f(K_j)$$

这就是不同的关键字映射成相同的地址空间，这叫冲突，由上可见，用杂凑法组织随机文件要解决两个问题：

1. 寻找哈希函数，实现关键字到地址的转换。要求哈希函数映射的地址，尽量均匀，且冲突尽量少。

2. 确定解决冲突的方法。

构造哈希函数常用的方法有：

(1) 平方取中法；(2) 除留余数法；(3) 折叠法；(4) 数字分析法。

解决冲突的方法有：

(1) 开放地址法；(2) 链接法。

这里仅就解决冲突的两个方法给读者进行介绍，

十个职员记录是

先看开放地址法。

0023427

例：有十个职员的记录使他们的职工号作为关键字，

1002343

记录的其它域暂竟略去。

1023672

首先用除留余数法构造哈希函数：

0983261

$$A = f(K) = 10 + (K - [K/10] \times 10 + 1) \times 2$$

0834673

0552364

其中符号[]表示取[]中数的最小整数。

0780026

例 $[3456/100] = [34.56] = 34$

0000235

将上面每个记录中的职工号（关键字）代入哈希

0926789

函数中求得地址：

1002568

职工号	地址
0023427	26
1002343	18
1023672	16
0983261	14
0834673	18
----->f(K)<-----	
0052364	20
0780026	24
0000235	22
0926789	30
1002568	28

然后将内存的14到30号单元开辟为基本区，用来存放记录，再将内存的31到35号单元开

辟为溢出区，用来存放发生冲突时基本区又放不下的记录。下面是经过哈希函数的转换，记录在内存中的分布情况。

由职工号经过哈希函数 $f(K)$ 转换成地址的过程中我们发现职工号 现1002343和0834673都转换成同一地址18，即发生冲突了。下面就用开放地址法来解决这种冲突。

用开放地址法解决冲突是这样的：

对 $K_i \neq K_j$

但 $f(K_i) = f(K_j) = A$

那么记录 $R(K_i)$ 放在A单元中

$R(K_j)$ 放在A+1单元中。

若A+1单元中有记录就放在A+2单元中，……直到基本区顶都有记录就回头放，若基本区已放满，就将记录 $R(K_j)$ 顺序放在溢出区。所以职工号为0834763的记录就放在09号单元。

用开放地址法解决冲突的缺点是：

(1) 对频繁搜入和删除的文件是不理想的，因为产生溢出的可能性大。

(2) 对物理地删除一个记录是困难的。

因此下面介绍另一种解决冲突的方法—链接法。

链接法解决冲突的过程如下：

首先将映射成同一地址的关键字组成一个等价类，每个等价类存储成一个链接表，表内每个记录的格式是：

关键字域	其他域	链接指针
------	-----	------

如果在一个关键字集合里使用某一哈希函数产生了M个等价类，则这个文件由M个独立链接表组成

这里还要用到一个叫做等值表的一维数组，其中每个单元的地址是哈希函数映射的地址，该单元中放相应链接表最后搜入记录的地址，如果某个单元中没有地址，则其值为0。

例：一位来访者到某宾馆找某旅客，只知道该旅客姓名，不知道他住几楼几号房间，就可到查询终端处，将查找姓名在终端键盘上敲入，终端屏幕立即显示出该旅客住几楼几号房间。

下面由随机组织文件来进行该系统的设计。

查询旅客的方法：是这样的例如要查姓名叫COW的旅客住几楼几号房间，首先由哈希函数ASC(COW)实现关键字转变成地址67，再以67单元的内容为地址（即103）将其内容取出得：

103	Can	209	102
-----	-----	-----	-----

接着用COW和关键字Can进行比较看是否相等，这时不相等就根据103单元的第三部分，链接指针102，将102单元内容取出得：

内存		24	0780026
14	0983261	25	
15		26	0023427
16	1023672	27	
17		28	1002568
18	1002343	29	
19	0834673	30	0925789
20	0052364	31	
21		32	
22	0000235	33	
23		34	
		35	

102	COW	208	0
-----	-----	-----	---

再用COW和关键字COW比较，这时相等了，就将102单元内容打印出来，这样就知道该旅客住208房间。

源程序如下：(CROMEMCO 16K BASIC)

```

3  CREATE "EQ"
4  CREATE "MA"
5  OPEN/1/"EQ"
10 OPEN/2/"MA"
15  A=100, B=65
20  FOR I=0 TO 25
22  A$="0", M=0, L=0
23  PUT/1, B+I/M
25  PUT/2, A+I/A$(-1), M, L
30  NEXT I
35  READ N$(-1), M
37  IF M=9999 THEN 100
40  I=ASC(N$(-1))
42  GET/1, I/A1,
43  IF A1=0 THEN 70
45  PUT/1, I/A
55  PUT/2, A/N$(-1), M, A1
63  A=A+1
65  GOTO 35
70  PUT/1, I/A
75  PUT/2, A/N$(-1), M, L
78  A=A+1
80  GOTO 35
100 CLOSE
101 OPEN/"EQ", OPEN/2/"MA"
103 PRINT "WHAT IS NAME OF YOUR FIREND?"
105 INPUT N$(-1)
115 GET/1, ASC(N$(-1))/A
120 GET/2, A/N1$(-1), M, L
125 IF N$=N1$ THEN PRINT N$, M: GOTO 103
130 IF L=0 THEN PRINT "NOT FIND": GOTO 103
135 A=L: GOTO 120
200 DATA "AW" 101, "BEN", 207, "COW", 208, "CAN", 209
205 DATA "IA", 303, "IAM", 309, "IBN", 317, "IOY", 418
10

```

210 DATA "AA", 202

999 DATA "0", 9999

倘若要删去某旅客或者登记某旅客只须在Data语句里进行修改就可以。凡修改了Data语句的内容,必须重新运行本存序让它重新建立等值表和链接表,若没有修改Data语句,调出本程序后,只须由101语句开始执行即可进行查找。

倒排文件

仅只有一个主关键字或仅只有一个索引,对文件中的记录的检索是不方便的。

例如,对职工文件

记录号	职工号	职务	性别	婚否
1	29	程序员	男	婚
2	05	维修员	男	婚
3	02	程序员	女	婚
4	33	穿孔员	男	未
5	31	维修员	男	婚
6	43	维修员	女	未
7	17	程序员	男	未
8	48	穿孔员	女	未

提出下列询问:

1. 列出所有男的未婚职工的名单。
2. 列出所有女的已婚程序员的名单。
3. 列出所有男的未婚程序员的名单。

以上这些信息在文件中都有,询问应该是允许的。但是询问的域名象职务、性别和婚否都不是主关键字,并且关于一些域名也没有建立索引。因而回答上述问题只能按主关键字顺序查找文件中每一个记录。这样做算法复杂,效率很低。为提高查找效率,下面介绍倒排文件。

一、组织

倒排文件组织由两部分组成

1. 原始文件:如上述的职工文件表。
2. 索引表:由次关键字值和指向具有该值的记录号组成。

上述职工文件的索引表如下:

职业索引表

穿孔员	维修员	程序员
04	02	03
08	05	07
	06	01

性别索引表

男	女
2	3
7	6
1	8
5	
4	

婚否索引表

婚	未
3	7
2	4
1	8
5	6

二、处 理

1. 检索

列出所有男的未婚职工的名单。

(1) 由性别索引表和婚否索引表中查得:

男性记录号集合: 2、7、1、5、4。

未婚记录集合: 7、4、8、6。

(2) 取这两个集合的交集得 7、4。即是满足询问条件的记录号, 再由职工文件中将 7 号记录、4 号记录打印出来, 即得所要的名单: 职工号 17 和 33 者是男的未婚的职工。

列出上述职工文件中所有女的已婚程序员名单。

(1) 由性别索引表中得女性记录号集合是 3、6、8。

由婚否索引表中得已婚记录号集合是 3、2、1、5。

由职务索引表中得程序员记录集合是 3、7、1。

(2) 由这三个集合求交集得: 3

(3) 由职工文件中将 3 号记录印出, 得知职工号是 2 者是女的, 已婚的程序员。

2. 检索算法

询问男的未婚职工有那些? 请按①键。

询问女的已婚的程序员有那些? 请按②键。

(1) 创建职工文件及职务、性别、婚否索引表。

程序清单如下: (DJS-130 磁盘 BASIC)

```

5  DIM d$(5), S$(5), m$(3)
10 OPEN FILN(0, 0), "MASTER"
15 PRINT "ENTER RECORD NUMBER"
20 INPUT N
25 FOR I=1 TO N
30 INPUT N1, D$, S$, M$
35 WRITE FILE(O, I), N1, D$, S$, D$
40 NEXN I
45 CLOSE FILE(O)
50 OPEN FILE(1, 0), "DATY"
```



```

55 PRINT*ENTER DUTY RECORD NUMBER"
60 INPUT N
65 FOR I = 1 TO N
70 INPUT P,M,P1
75 WRITE FILE(1,I)P,M,P1
80 NEXT I
85 CLOSE FILE(1)
90 OPEN FILE(2,O),"SEX"
90 PRINT*ENTER SEX RECORD NUMBER"
100 INPUT N
105 FOR I = 1 TO N
110 INPUT M,F
115 WRITE FILE(2,I),M,F
120 NEXT I
125 CLOSE FILE(2)
130 OPEN FILE(3,O),"MARITAL"
135 PRINT*ENTER MARITAL RECORD NUMBER"
140 INPUT N1
145 FOR I = 1 TO N1
150 INPUT Y,N
155 WRITE FILE(3,I),Y,N
160 NEXT I
165 CLOSE FILE ( 3 )
170 END

```

(二) 检索程序

```

5 OPET FILE(O,O),"MASTER"
10 PRINT*PROBLEM NUMBER = , "
15 INPUT N
20 IF N>2 THEN 35
25 IF N<0 THEN 35
30 ON N GOTO 45,190
35 PRINT*INQUIRED NUMBER ERRE"
40 GOTO 10
45 OPEN FILE(2,0),"SEX"
50 LET I = 1
55 DIM M(10)
60 IF EOF(2) THEN 100
65 READ FILE(2,I),M1,F
70 IF M1 = 0 TEHN 90

```