

联机情报检索语言IRL/1*

姚卿达

(数学力学系)

IRL/1是以实现电子计算机辅助管理企业各类情报为主而设计的一种联机情报检索语言(Information Retrieval Language),它是属于信息管理中的自立语言系统(Self-Contained Language System)。而不同于目前国际上流行的ALGOL-60、FORTRAN等用于描述数值计算过程的算法语言。

IRL/1将数据描述功能与数据管理功能合在一起考虑,力求方便、有效。IRL/1的数据类型有:简单算术型、行型以及结构型,适于描述情报信息,其结构型的层阶结构方式类似于COBOL语言的记录,但IRL/1的结构型数据不仅考虑描述能力,还兼顾了数据管理的需要。IRL/1提供了十四种操作语句,允许扩充。其程序结构较简单,由一个可以独立使用的语句或一组复合子句序列所组成,用于实现情报文档的定义(建立)、存贮、检索以及编辑管理等作业。IRL/1具有“非过程式”特性,以解释方式运行程序。

IRL/1是近年来我们在情报检索系统的研究和实践基础上提出来的,它的基本部分已在配有磁盘、实时时钟、多路通讯接口以及二十六个终端的DJS-130机上实现。在操作系统管理之下,允许多个用户同时地进行各类管理与检索作业。系统已在实际运行中收到了预期的效果。

在IRL/1的语法表示中,除了使用熟知的BNF外,还引入花括号以及上、下角标表示:

1、花括号{、}中,总是并列地给出几个语法成分,使用时选择其中之一。

2、花括号或括号后面出现的下角标和上角标,用于表示括号内成分能够重复的最小和最大次数。

一、基本符号

语法:〈基本符号〉 ::= 〈字母〉|〈数字〉|〈定义符〉

〈字母〉 ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|
Q|R|S|T|U|V|W|X|Y|Z

〈数字〉 ::= 0|1|2|3|4|5|6|7|8|9|

● 本文得到李宏新、徐满昌、欧贵文、张瑛等同志的关心与协助。

- 〈定义符〉 ::= 〈句首定义符〉 | 〈消错符〉 | 〈分隔符〉 | 〈括号〉 | 〈语句终结符〉 | 〈运算符〉 | 〈控制符〉 | 〈说明符〉 | 〈累加变量符〉 | 〈其他符〉
- 〈句首定义符〉 ::= OPEN | CLOSE | RETRIEVE | DEFINE | LOAD | REPLACE | INSERT | DELETE | TYPE | CHANGE | CLEAR | COPY | SEEK | COMPLEX
- 〈运算符〉 ::= 〈算术运算符〉 | 〈逻辑运算符〉 | 〈比较运算符〉
- 〈算术运算符〉 ::= + | - | * | /
- 〈逻辑运算符〉 ::= V | ^
- 〈比较运算符〉 ::= *GT* | *LT* | *GE* | *LE* | *NE*
- 〈分隔符〉 ::= . | , | : | ; | " | ' | { | }
- 〈括号〉 ::= (|)
- 〈语句终结符〉 ::= @
- 〈消错键〉 ::= ' | <
- 〈控制符〉 ::= \ | =
- 〈说明符〉 ::= { REAL | R } | { INTEG | I } | { ALPHA | A } | { WORD | W }
- 〈累加变量符〉 ::= AC₀ | AC₁ | AC₂ | AC₃ | AC₄ | AC₅ | AC₆ | AC₇
- 〈其他符〉 ::= ALL | MAX | MIN | PTP | PTR | LPT | DISK | TTI | TTO | READ | WRITE | MOVE | PRINT | ACC | TO

说明:

字母不分大写与小字,通常只用大写,消错符“'”用于消去前一输入的字符,而“<”则清除前面全部输入的字符,控制符“\”(回车)表示一行输入结束,终结符“@”表示作业提交完毕或本语句输入结束,“=”(换行)只作为输入或输出控制符,“\”“=”以及“ ”(空格)不印字符。

二、数据类型与结构

IRL/1的数据包括简单算术型数据、行型数据以及结构型数据。

1、简单算术型具有整数(INTEG)和实数(REAL)两种。

语法: 〈数〉 ::= 〈整数〉 | 〈实数〉

〈整数〉 ::= 〈无符号整数〉 | +〈无符号整数〉 | -〈无符号整数〉

〈无符号整数〉 ::= 〈数字〉 | 〈无符号整数〉 〈数字〉

〈实数〉 ::= 〈无符号数〉 | +〈无符号数〉 | -〈无符号数〉

〈无符号数〉 ::= 〈无符号整数〉 | 〈十进制小数〉 | 〈无符号整数〉 〈十进制小数〉

〈十进制小数〉 ::= 〈.无符号整数〉

说明：整数与实数具有通常意义。

2、行型数据具有字符行和机器字。

语法：〈字符行〉 ::= 〈字符〉 | 〈字符行〉 〈字符〉 | 〈空〉

〈字符〉 ::= 〈字母〉 | 〈数字〉 | 〈分隔符〉

〈机器字〉 ::= 〈无符号整数〉 | 〈位行〉

〈位行〉 ::= * 〈零壹组〉 * | 〈空〉

〈零壹组〉 ::= 0 | 1 | 〈零壹组〉 0 | 〈零壹组〉 1

说明：字符行是由字母、数字和分隔符所组成的任一序列。

机器字用来作指针。

位行是一组二进位。

3、结构型数据包括情报模块(以IM表示)和备注记录。情报模块是具有层次结构的数据，其结构方式以下列IM格式说明所确定：

语法：〈IM格式说明〉 ::= 0 1 〈 〉₁ⁿ 〈IM名字〉 (〈说明子句〉) ↘
〈IM体〉

〈IM体〉 ::= 〈 〉₀ⁿ 0 2 〈 〉₁ⁿ 〈辅助名标志〉 ↘ { 〈 〉₀ⁿ 0 3 〈 〉₁ⁿ

〈辅助名标识符〉 (〈说明子句〉) ↘ }₁ⁿ 〈 〉₀ⁿ 0 2 〈 〉₁ⁿ 〈条件标志〉 ↘

{ 〈 〉₀ⁿ 0 3 〈 〉₁ⁿ 〈条件描述词标识符〉 (〈说明子句〉) ↘ }₁ⁿ 〈 〉₀ⁿ 0 2 〈 〉₁ⁿ 〈内容标志〉 ↘

{ 〈 〉₀ⁿ 0 3 〈 〉₁ⁿ 〈描述词标识符〉 (〈说明子句〉) }₁ⁿ 〈 〉₀ⁿ 0 2 〈 〉₁ⁿ 〈备注标志〉 ↘

〈 〉₀ⁿ 0 3 〈 〉₁ⁿ 〈备注事项标识符〉 (〈说明子句〉) ↘ 〈 〉₀ⁿ 0 3 〈 〉₁ⁿ 〈备注指针标识符〉 (〈说明子句〉) ↘

〈 〉₀ⁿ 0 3 〈 〉₁ⁿ 〈同名模块链标识符〉 (〈说明子句〉) ↘

〈IM名字〉 ::= 〈标识符〉

〈标识符〉 ::= 〈字母〉 | 〈标识符〉 〈字母〉 | 〈标识符〉 〈数字〉

〈说明子句〉 ::= 〈类型说明〉 { , 〈字符个数〉 | , 〈字符个数〉 ·
〈字符个数〉 | 〈空〉 }

〈类型说明〉 ::= 〈说明符〉

〈字符个数〉 ::= 〈无符号整数〉

〈辅助名标志〉 ::= 〈位行〉 | 〈字符行〉

〈内容标志〉 ::= 〈位行〉 | 〈字符行〉

〈备注标志〉 ::= 〈位行〉 | 〈字符行〉

〈条件标志〉 ::= 〈位行〉 | 〈字符行〉

〈备注记录〉 ::= 〈头标〉 ↘

〈链指针〉 ↘

〈备注〉↘

〈尾标〉↘

〈头标〉 ::= 〈位行〉 | 〈字符行〉

〈尾标〉 ::= 〈位行〉 | 〈字符行〉

〈链指针〉 ::= 〈机器字〉

〈备注〉 ::= 〈字符行〉

说明:

1) IM格式说明与备注记录中出现的标识符以四个字符为限,超过时只取前四个,同一IM格式说明中,不允许出现相同的标识符。

2) IM格式说明与备注记录中的各个标志,由编译程序给出。

3) IM包含两方面内容:一是IM格式说明(或叫IM的型);一是IM的数据(或叫IM的值)。如无特别声明,IM指的是数据,或者叫存贮模块。

4) IM格式说明定义了IM的构造方式,一个IM由若干数据项所组成,每个数据项有个层号,用两位整数表示,层号大小反映了各数据项之间的从属关系。层号为01者仅有一项,是最高项,作为IM名字;若一个数据项可以再分为若干低层的数据项来表示,则称为组项,在IM结构中,凡层号为02者都是组项,各组项以特定位行或字符行标记的,不可再分的数据项称为初等项,每个初等项有个以标识符表示的名字。初等项相当于简单变量,其值可以是算术型数据或行型数据,取值类型以及长度由相应的说明子句所给出。长度由字符个数来量度,对于实型量来说,用〈字符个数〉·〈字符个数〉指明整数部分与小数部分的字符个数。

5) IM格式中的名字项以及内容描述项是必须有的,而辅助名、条件描述以及备注等组项是任选项,对于某一类具体文档,可以省去一些任选项。

6) IM格式说明是存取情报和加工情报的依据,在存贮器里存放的各个情报模块IM都是按IM格式说明形成的数据块,即存贮模块。存贮模块的最高项(IM名字之值)叫做检索词。

具有相同检索词的IM,称为同名模块,同名的各模块,通过同名模链而连结在一起。IM中出现的辅助名、描述词以及条件描述词都必须是标准化的,描述词表由用户根据管理对象(某类情报)自行建立。

4、在定义IM格式时,辅助名个数 K 、条件描述词个数 m 以及内容描述词个数 p 必须是确定的,因此情报模块是定长的。为了便于表示某些特殊的情报,采用可变长备注记录来弥补定长之不足。备注记录是有头、尾标志的任意字符序列,其解释权属于用户。备注记录与IM的连接是通过M的备注指针与备注记录中的链指针来实现。

三、文档、情报库

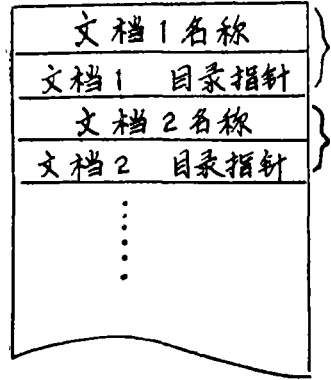
1、文档

文档是一系列情报模块和备注记录按特定规则在存贮器上的积集,文档具有符

号名字, 由目录、检索表、辅助检索表, 非标准词表、情报模块及其备注记录集合等几部分组成。

文档目录格式如下:

- 文档名称
- 建档时间
- 检索表指针(检索表盘区号)
- 辅助检索表指针(辅助检索表盘区号)
- 非标准词表指针(非标准词表盘区号)
- 打开标志(标志文档是否已打开?)
- 链单元 1 (指示检索表在内存当前位置)
- 链单元 2 (指示辅助检索表在内存当前位置)
- 链单元 3 (指示非标准词表在内存当前位置)
- 密码(特殊位行或字符行)



IM格式说明

检索表是检索词与相应模块存放盘区地址的对照表; 辅助检索表是辅助名与相应模块存放盘区地址对照表; 非标准词表是非标准检索词与同义检索词的对照表。

检索表、辅助检索表以及非标准词表均为有序结构, 对于整型数(或数字串)表示的词按大小顺序, 对于字符行表示的词按字典排列次序。

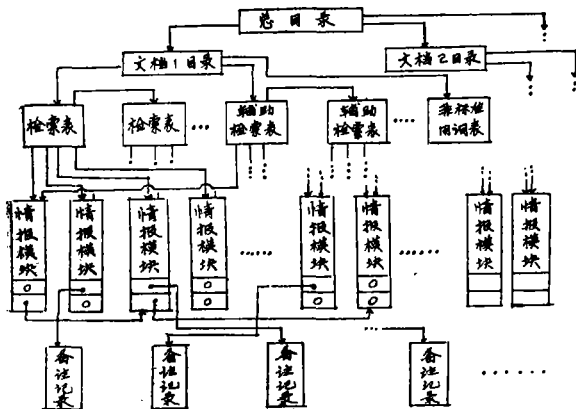
情报模块与备注记录在存贮器上随机存放, 通过目录、检索表、情报模块中的备注指针以及备注记录中的链指针, 使文档各部分组成一个整体。

2、情报库

情报库是一个或若干个文档通过总目录而组织起来的一个数据库。

总目录是各个文档的标题信息所组成的线性表(见上图)。

情报库结构示意图(见下图)。



情报库结构示意图

四、表达式

语法: <表达式>:: = <算术表达式> | <检索条件式> | <比较条件式>

<算术表达式>:: = | <项> <加法运算符> <项> | <算术表达式> <加法运算符> <项>

<项>:: = <无符号数> | <数值性初等项> | <累加变量> | (<算术表达式>) | <项> <乘法运算符> <项>

- 〈数值性初等项〉 ::= 〈标识符〉
 〈累加变量〉 ::= 〈整型累加变量〉 | 〈实型累加变量〉
 〈整型累加变量〉 ::= $AC_0 | AC_1 | AC_2 | AC_3$
 〈实型累加变量〉 ::= $AC_4 | AC_5 | AC_6 | AC_7$
 〈加法运算符〉 ::= $+ | -$
 〈乘法运算符〉 ::= $* | /$
 〈检索条件式〉 ::= 〈检索项〉 | 〈检索条件式〉 \vee 〈检索项〉
 〈检索项〉 ::= 〈检索因子〉 | 〈检索项〉 \wedge 〈检索因子〉
 〈检索因子〉 ::= { 〈检索词〉 | 〈辅助检索词〉 } { (〈条件〉) | 〈空〉 }
 〈条件〉 ::= 〈条件描述项〉 { , 〈条件描述项〉 }ⁿ
 〈条件描述项〉 ::= 〈条件比较值〉 | 〈比较运算符〉 〈条件比较值〉
 〈条件比较值〉 ::= 〈数〉 | 〈位行〉 | 〈字符行〉 | 〈条件描述词〉
 〈比较运算符〉 ::= $*GT* | *LT* | *GE* | *LE* | *NE*$
 〈检索词〉 ::= 〈无符号数〉 | 〈字符行〉
 〈辅助检索词〉 ::= 〈无符号数〉 | 〈字符行〉
 〈条件描述词〉 ::= 〈数〉 | 〈位行〉 | 〈字符行〉
 〈比较条件式〉 ::= 〈比较项〉 | 〈比较条件式〉 〈逻辑运算符〉 〈比较项〉
 〈比较项〉 ::= 〈初等项〉 〈比较运算符〉 〈初等项比较值〉
 〈初等项〉 ::= 〈标识符〉
 〈初等项比较值〉 ::= 〈数〉 | 〈字符行〉 | 〈位行〉

说明:

1、关于算术表达式:

算术表达式是数值性初等项(以标识符表示)、累加变量和数用算术运算符和圆括号连接起来的式子,它描述一个数值计算规则。本语言不允许用乘方和初等函数。

算术表达式中引入累加变量,目的在于方便地实现不同IM之间的运算联系和统计分析,累加变量共八个,以 AC_0, AC_1, \dots, AC_7 表示,在多用户情形,每个用户均有这样八个累加变量,对它们,不必说明即可在表达式中加以使用。

算术表达式中出现的初等项,是取当前模块(即当前正在处理的情报模块)中相应的初等项之值参加运算。

2、关于检索条件式

检索条件式由若干检索词(或辅助检索词)附加相应的比较条件,用逻辑运算符 \vee (逻辑加)以及 \wedge (逻辑乘)连结起来的式子,用于检索语句中,描述求取情报模块的规则。

检索条件式中出现的检索词和辅助检索词必须是在检索表或辅助检索表中已有定义的,对于同义检索词则必须是在非标准词表中已有定义的。书写式子时,辅助检索词前面必须加“·”,以和检索词有所区别。检索条件式的运算规则如下:

第一、检索因子处理优先。根据检索因子中的检索词或辅助检索词，找出相应的IM，并验证条件，求得该因子的值(一个IM或空)；

第二、∧优先；

第三、∨优先；

第四、同一优先级的逻辑运算，自左向右依次处理。

检索条件式中的检索因子的数目，不许超过编译程序的规定。

条件描述项中出现的比较运算符的意义如下：

GT (greater then)表示大于

LT (less then) 表示小于

= (equal) 表示等于

GE (greater then or equal)表示大于或等于

LE (less then or equal)表示小于或等于

NE (not equal)表示不等于

条件描述项的数目与次序必须与该文档IM格式说明中条件描述词个数与次序一致，而且比较值的类型与相应条件描述词类型也必须一致。条件描述项之间用逗号“，”分开，当某一条件不比较时，相应项空着，当所有条件不比较时，可以省去整个括号内的条件。例如检索因子。

〈检索词〉(〈条件描述项〉，，)表示只比较第一个条件，其余两个不作比较。

条件描述项是检索词和辅助检索词的补充，处理同名模块时，往往要附加条件来加以区分。附加条件与检索词之间隐含着逻辑乘的关系。

3、关于比较条件式

比较条件式是寻找合乎条件的IM的式子，用在寻找语句之中。

五、语 句

IRL/1的语句属于非过程式，每个语句具有句首定义符及语句信息两部分，其一般形式是：

〈句首定义符〉〈语句信息〉@

@是语句结束号。每个语句完成情报管理中的一个联机作业，句首定义符指出“做什么”，语句信息提供有关的信息和参数。

为了增加语言灵活性，还引入复合形式的语句，它通过一系列运算符来描述某些处理细节。

语句共十四种，即

〈语句〉::=〈定义语句〉|〈打开语句〉|〈关闭语句〉|〈插入语句〉|〈装入语句〉|〈检索语句〉|〈寻找语句〉|〈更新语句〉|〈删除语句〉|〈显示语句〉|〈密码改变语句〉|〈文档清除语句〉|〈复制语句〉|〈复合语句〉

1、定义语句

语法: <定义语句> ::= DEFINE <定义模块> ②

<定义模块> ::= <文档名> | <密码> | <文档类型> | <记录格式>
{ * <存贮分配信息> | <空> }

<文档名> ::= <标识符>

<密码> ::= <位行> | <字符行>

<文档类型> ::= <检索表方式> · <辅助检索表方式>

<记录格式> ::= <IM格式说明>

<存贮分配信息> ::= <无符号整数>

<检索表方式> ::= I | A

<辅助检索表方式> ::= I | A

说明:

1) 定义语句用于定义一个文档,它根据定义模块所提供的信息,生成标题信息挂在总目录上。并建立文档目录,加工IM格式说明,分配盘区——给出一个空载盘图,搭好检索表、辅助检索表和非标准词表的框架,为装入情报模块以及检索词等作好准备。此外,还为用户开辟工作模块,供加工情报模块时使用。

2) 密码是由建档人任意指定的位行或字符行。

3) 文档类型指出检索表、辅助检索表的结构方式, I 表示检索词类型为整型,按大小顺序排列, A 表示检索词类型为字符行型,按字典方式排列,非标准词表的结构方式与检索表一致。

4) 存贮分配信息给出情报模块总数或预计值,供分配盘区与搭框架用,若未指定,则按标准方式分配,由编译程序加以考虑。

5) 记录格式定义了文档的IM格式,以及各数据项的标识符、类型和取值范围,作为存贮和加工IM的依据。

2、打开语句

语法: <打开语句> ::= OPEN <文档名> | <密码> ②

说明: 此语句用于打开一个文档,使处于可用状态。执行时首先核对密码,然后在内存分配一个区域,将检索表(或其中一部分,或其索引)调入内存,若检索表是分级的,则将首级调入内存。

3、关闭语句

语法: <关闭语句> ::= CLOSE <文档名> | <密码> ②

说明: 此语句用于关闭一个文档,将已打开文档的检索表调入盘区,释放内存空间。

打开与关闭必须配对使用,一个文档打开之后,必须用关闭语句先关闭才能再次打开。

4、插入语句

语句: <插入语句> ::= <模块插入语句> | <非标准词插入语句>

〈模块插入语句〉 ::= INSERT〈文档名〉┌{〈情报模块〉|〈空〉}{┌
└〈备注模块〉|〈空〉}②

〈非标准词插入语句〉 ::= INSERT〈文档名〉┌井〈检索词〉{┌〈非标准
词〉}┐₁②

说明：1) 插入语句用于终端上进行离散插入情报模块及相应的检索词、辅助检索词和非标准词。

2) 插入过程中，若遇同名检索词，则沿着指针找到同名模块链的链尾，将链尾修改指向新的同名模块。若迂同名的辅助检索词，则不是拉链，而是在辅助检索表中列出若干指针来解决同名问题。

3) 插入语句在文档已打开状态下进行。在插入过程中不允许其它作业对此文档的访问(后面那些具有编辑功能的语句也类似处理)。

5、装入语句

语法：〈装入语句〉 ::= LOAD〈文档名〉┌〈密码〉┌〈输入设备〉②

〈输入设备〉 ::= PTR|TTR

说明：

1) 装入语句用于批量装入情报模块，事先将情报模块与备注记录穿孔记录在纸带，然后挂在输入设备上，与本语句配合使用。

2) 输入设备PTR表示纸带输入机，TTR表示电传纸带输入机。

6、检索语句

语法：〈检索语句〉 ::= RETRIEVE〈文档名〉┌〈密码〉┌〈检索条件式〉②

说明：检索语句根据检索条件式进行检索运算，找出合乎条件式的情报模块。若检索成功，则将检索结果按特定的格式输出，若检索失败，则输出无此情报的信息。

执行时，首先核对密码，打开文档，然后按条件式检索，若文档已打开，则直接进入检索。

本语句是用户经常地、连续地使用的重复语句，为了方便操作和加快作业速度，在文档已打开的情况下，可以省写句首定义符、文档名和密码，只给检索条件式即可。

7、寻找语句

语法：〈寻找语句〉 ::= 〈找最大值语句〉|〈找最小值语句〉|条件寻找语句〉

〈找最大值语句〉 ::= SEEK〈文档名〉┌〈密码〉┌MAX┌〈数值性初等项〉②

〈找最小值语句〉 ::= SEEK〈文档名〉┌〈密码〉┌MIN┌〈数值性初等项〉②

〈条件寻找语句〉 ::= SEEK〈文档名〉┌〈密码〉┌ALL┌〈比较条件式〉②

说明：寻找语句不是按检索词或辅助检索词去检索模块，而是在同一文档的所有情报模块中，对指定的某一初等项，经过比较运算，找出包含最大值或最小值，或合乎某一条件的模块出来。

在比较运算中。除了相等比较之外，其余比较只限于数值性初等项。

8、更新语句

语法：〈更新语句〉 ::= 〈模块更新语句〉 | 〈数据项更新语句〉

〈模块更新语句〉 ::= REPLACE 〈文档名〉 ⊖ 〈检索因子〉 ⊖ { 〈情报模块〉 | 〈空〉 } { - 〈备注模块〉 | 〈空〉 } @

〈数据项更新语句〉 ::= REPLACE 〈文档名〉 ⊖ 〈检索因子〉 ⊖ * 〈初等项标识符〉 : 〈更新值〉 @

〈更新值〉 ::= 〈整数〉 | 〈实数〉 | 〈字符行〉 | 〈机器字〉

说明：模块更新语句将情报模块、或备注模块、或两者进行更新，模块更新时，尚需核对检索词与辅助检索词，不一致时，同时更改检索表与辅助检索表中有关部分。

数据项更新只允许对初等项进行。

更新语句在文档打开情况下进行，更新完毕需关闭文档。

对同名模块中之一更新时，必须附加条件以指明其中那一块。

9、删除语句

语法：〈删除语句〉 ::= 〈模块删除语句〉 | 〈非标准词删除语句〉

〈模块删除语句〉 ::= DELETE 〈文档名〉 ⊖ 〈检索因子〉 @

〈非标准词删除语句〉 ::= DELETE 〈文档名〉 ⊖ * 〈非标准词〉 @

说明：模块删除语句将检索因子所指定的情报模块及其备注记录删除，此外，所涉及的检索词、辅助检索词和非标准词亦作删除和整理。

对于同名模块，若未给出条件，则将所有同名者删除。模块删除后，释放相应盘区。

非标准词删除语句将该词删去，并整理非标准词表。

删除语句在文档已打开的状态下进行，删除完毕需关闭文档。

10、显示语句

语法：〈显示语句〉 ::= TYPE @ | TYPE 〈文档名〉 @ | TYPE 〈文档名〉] 〈检索词〉 : 〈显示幅度〉 @

〈显示幅度〉 ::= 〈无符号整数〉

说明：显示语句用于在终端上显示(打印)文档目录信息或一组情报模块。若语句参数全空，显示总目录。若只给文档名，则显示文档目录，若给出文档名、检索词及显示幅度，则显示出以该检索词为中心，按显示幅度大小，将前面与后面若干检索词及其相应情报模块显示出来，不论何种情况，密码不予显示。

11、密码改变语句

语法: <密码改变语句> ::= CHANG<文档名>┐<密码>; <更新密码>①
 <更新密码> ::= <位行>|<字符行>

说明: 对已定义的密码, 以新换旧, 新密码类型与长度必须与旧密码一致。

12、文档清除语句

语法: <文档清除语句> ::= CLEAR<文档名>┐<密码>①

说明: 此语句将该文档的标题目录、检索表、辅助检索表、非标准表、情报模块及其备注记录, 乃至为同户开辟的工作模块等全部信息清除, 释放在贮空间。

13、文档复制语句

语法: <文档复制语句> ::= COPY<文档名>┐<密码>┐<复制设备>①
 <复制设备> ::= DISK<驱动器号>|LPT|PTP|TTO

说明: 将一个文档的全部内容按一定格式复制(输出)到复制设备上。

复制设备有DISK(磁盘), LPT(行式打印机)、PTP(纸带输出机)、TTO(电传输机)。

14、复合运算语句

语法: <复合运算语句> ::= COMPLEX<文档名>┐<密码>┐<运算子句>ⁿ①
 <运算子句> ::= <读模块语句>|<写模块语句>|<打印语句>|<送值语句>|<计算语句>|<累加语句>

<读模块语句> ::= READ<检索因子>↘

<写模块语句> ::= WRITE{<检索因子>|<空>}↘

<计算语句> ::= COMPUTE<算术表达式>┐TO┐<变量>↘

<送值语句> ::= MOVE{<变量>|<直接变量>}┐TO┐<变量>↘

<打印语句> ::= PRINT{<变量>|<输出行><变量>|<直接量>}↘

<累加语句> ::= ACC┐ALL┐<数值性初等项>┐TO┐<变量>↘

<变量> ::= <数值性初等项>|<累加变量>

<直接量> ::= <数>|<字符行>|<位行>

<输出行> ::= *<不包括*的任意字符行>*

说明:

1) 读模块语句将检索因子指定的一个情报模块读入到当前模块。

2) 写模块语句将当前模块写入检索因子指定的模块之中, 若未给检索因子, 则写入为用户开辟的工作模块之中。

3) 计算语句将表达式的值计算, 并送入指定变量, 类型以指定变量(结果变量)为准。

4) 送值语句将变量或直接量的值送入结果变量, 并以结果变量类型为准进行转换。

5) 打印语句将变量或直接变量的值印出, 为了印刷清楚醒目, 可以在印值之

前先印一个字符行。

6) 累加语句对同一文档各情报模块的某一初等项, 求其总和, 送入结果变量之中。

六、程序

IRL/1的程序就是一个可以单独使用的语句, 或是一系列运算符组成的复合运算语句。未引入控制转向及启行语句, 复合运算语句中各运算符句, 按顺序解释执行。

七、交互功能

IRL/1是联机检索语言, 以会话形式进行各种作业, 在作业进行之前, 有接通终端会话, 作业过程由用户提交作业、系统输出错误信息、结果信息、完成标志, 也可以给出提问或提示。

有关交互会话内容和规定, 由编译程序和操作系统考虑。

八、扩充功能

根据管理各类情报的需要, 允许扩充IRL/1语句, 或增加语句功能。本文未提供输出格式说明以及格式输出语句, 可另行扩充。

九、实现

IRL/1的基本部分已在配有磁盘和多个终端设备的DJS-130机上实现。已运行的系统在数据结构、语句类型与格式等方面作了简化, 功能较弱, 有待改进和扩充。

动态

我校教师代表团訪問香港大专院校

今年二月十九日至三月四日, 我校教师代表团一行十二人, 团长为黄焕秋副校长, 团员包括数学、化学、物理、外语、中文各系负责人, 科研处、东南亚历史研究所负责人等, 前往香港参观访问香港大学、中文大学、理工学院, 考察教学、科学研究和管理的情况与经验, 交换关于开展学术交流的意见, 受到三所大专院校领导人和教师的热情接待。

代表团参观了三所大专院校的各学系、文理科的各个研究中心、教育科技中心(包括视听中心、教研媒介资源中心)、电子计算中心、工业中心、实验室、图书馆、博物馆以及校外课程部。代表团广泛会见了三院校的知名学者、教授、教师, 参加了一系列的情况介绍会与座谈会, 互相交换了关于加强学术交流的意见。代表团团员还和三院校的一些教授、学者同行之间进行了互访与晤谈。通过代表团这次访问活动, 加深了彼此之间的友谊与了解, 促进了我校与香港大专院校学术交流关系的建立与发展。