

图像处理子系统：管理软件

——IRSA-2 遥感图像分析处理系统发展之三

高朋 李丽

(中国科学院遥感应用研究所)

1986年7月23日收到

一、引言

一个图像处理软件系统,除了包括各种图像处理应用程序之外,为了有效地利用这些应用程序,还应当有管理系统程序,用它对应用程序进行合理的组织和安排。在计算机操作系统的支持下,管理软件和应用程序一起构成了一个图像处理子系统。管理程序的好坏直接影响整个系统的性能。好的管理软件应当能使用户最方便,最有效地使用系统中的各种功能,真正起到用户与处理功能之间的接口作用。在 IRSA-2 图像处理软件系统中,采用的是多级分支清单 (MENU) 结构与命令行 (COMMAND LINE) 结构相结合的系统管理方法,它既可使整个系统使用方便,又能高效率地运行。

二、图像处理子系统管理方法

(一) 多级分支清单

多级分支清单 MENU 结构是一种使用方便的人机交互式系统管理方法。它通过逐级显示系统功能清单,逐步引导用户进入所需要的处理功能。各级系统功能清单由粗到细,将全部系统功能进行分类组合,分支清单中的每一项都对应着字符“A, B, C, D, E, F, G, H”或数字“1, 2, 3, 4, 5, 6, 7, 8”。用户只要简单地按动相应的字符或数字键即可进入下级分支清单,直至达到所希望的处理功能。在该点上,按相应的功能键,管理程序即把控制转给应用程序。处理完成之后,控制回到管理程序,重新显示分支清单。图 1 给出了在彩色监视器上显示一幅图像的操作过程。第一级分支清单称为系统功能清单,如图 1(a) 所示。选择交互功能时,只需按相应的字符键 C, 系统就显示二级功能清单,如图 1(b) 所示。二级清单中也有八项选择,每项选择的意义明确,有很强的提示作用。再回答操作码“1”,系统将进入第三级功能清单,如图 1(c) 所示。若再选择功能“1”,管理程序则把控制传递给图像显示应用程序。应用程序结束后将自动返回管理程序,如图 1(d) 所示。由此可以看出,多级分支清单方式,结构简单,有很强的提示作用,整个过程完全是以人机交互方式进行的。用户可以边学习边使用。

IRSA-2 系统多级分支清单管理系统,管理性能比较完善,具有一定的容错性和一些

SYSTEM FUNCTION MENU

- | | |
|--------------------------|-----------------------------|
| A. ALLOCATE IMAGE FILE | E. GRAPHIC PROCESSING MENU |
| B. CCT MENU | F. GEO - INFO - SYSTEM MENU |
| C. INTERACTIVE MENU | G. COMMAND LINE MODE |
| D. IMAGE PROCESSING MENU | H. RETURN |

ANSWER THE ALPHABET CORRESPONDING TO THE OPERATION YOU WANT TO DO(ANSWER A , B , C , D , E , F , G , H , OR \):

OPERATION CODE = ? C

图(a)

‡ ‡ ‡ INTERACTIVE WITH COMTAL DISPLAY ‡ ‡ ‡

- | | |
|---------------------------|---------------------------|
| 1. DISK IMAGE TO COMTAL | 5. COMTAL IMAGE TO DISK |
| 2. TAPE IMAGE TO COMTAL | 6. COMTAL IMAGE TO TAPE |
| 3. DISK GRAPHIC TO COMTAL | 7. COMTAL GRAPHIC TO DISK |
| 4. TAPE GRAPHIC TO COMTAL | 8. COMTAL GRAPHIC TO TAPE |

OPERATION CODE = ? 1

图(b)

DISK IMAGE TO COMTAL DISPLAY

- | | |
|----------------------------|------------------------|
| 1. DISPLAY MONCHROME IMAGE | 5. DISPLAY COLOR IMAGE |
| 2. | 6. |
| 3. | 7. COMMAND LINE MODE |
| 4. | 8. RETURN |

OPERATION CODE = ? 1

图(c)

DATA SENDED TO COMTAL

STOP

SYSTEM FUNCTION MENU

- | | |
|--------------------------|-----------------------------|
| A. ALLOCATE IMAGE FILE | E. GRAPHIC PROCESSING MENU |
| B. CCT MENU | F. GEO - INFO - SYSTEM MENU |
| C. INTERACTIVE MENU | G. COMMAND LINE MODE |
| D. IMAGE PROCESSING MENU | H. RETURN |

ANSWER THE ALPHABET CORRESPONDING TO THE OPERATION YOU WANT TO DO(ANSWER A , B , C , D , E , F , G , H , OR \):

OPERATION CODE = ? G

图(d)

图 1 图像显示的过程

Fig. 1 The process of Image Display

法,由用户通过终端给出。对每个参数都有相应的提示。

考虑到图像处理子系统面向各种专业背景的用户,而不仅仅是受过专门训练的计算机专业人员,因此,在系统实现时采取了两项保护措施:第一,屏蔽对子系统的中断。在 ECLIPSE S 140 计算机上,中断控制键为 CONTROL/C, CONTROL/B。如果用户在运行时按入此键对,一般情况下程序将被中断,退回到 AOS 操作系统。可是,用户需要的是图像处理子系统,而不是计算机操作系统。因此,在实现 MENU 时,对中断键进行了处理。使系统在接收到中断命令后返回到子系统 MENU,而不是到 AOS 操作系统。如要退出子系统,其唯一途径是通过功能项 STOP。第二,限制用户退出自己的用户目录。每个用户只允许在自己的目录中执行各种操作,绝不允许进入其它的用户目录,特别是超级用户目录,否则将可能破坏整个系统。因此,尽管可以通过功能项 STOP 进入 AOS 操作系统,但却不能退出本用户目录进入其它用户目录。如果用户企图退出自己的用户目录,则系统将控制重新回到 MENU 子系统中。

在 AOS FORTRAN 5 中,实现各级 MENU,或 MENU 与应用程序,或 MENU 与命令行解释程序间控制转移的方法有三种:①子程序调用,②系统功能 CHAIN(链接),③系统功能 SWAP(交换)。子程序调用方式对于少数几个程序是一个很好的方法,但是整个图像处理系统包括上百个应用程序,全部链接在一起,必然会使执行文件过于庞大,使可利用的内存空间减少,甚至可能难于实现。利用程序覆盖技术可以解决执行文件过大的问题,但如果整个系统均利用程序覆盖,覆盖关系复杂,很难进一步扩充 MENU。系统功能 CHAIN 是将控制从当前程序永久地转移到另一个程序,该程序结束后也不返回。系统功能 SWAP 则是将控制暂时转给其它程序,该程序结束后,控制返回到断点,继续向下执行。在这两种控制转移方式之下,每个分支 MENU 和应用程序均以可执行文件形式单独存在,无需链接在一起,因此,很容易修改和扩充。在 MENU 结构的具体实现中,主要采用了这两种系统功能。一般在各级 MENU 之间, MENU 与应用程序之间采用系统功能 SWAP,暂时将控制交给下级程序,当下级程序结束后再将控制收回。它反映出不同的上下级之间的调用关系。而在 MENU 与命令行解释程序的这种平行关系之间,则采用系统功能 CHAIN,完全将控制转移出去。如果这时仍用系统功能 SWAP,则两种方式之间的频繁转换会使 MENU 与命令行解释程序嵌套过深,致使不能很快退出图像处理子系统。

(二) 命令行解释程序设计

多级分支清单结构虽然易于使用,但它的反应速度慢,为了达到所需要的处理功能,至少需要三级选择。特别是当进行多次顺序处理时,两次处理不在同一个清单内,则必须先返回到高级清单,然后再逐级进行,直到进入新的处理功能。为使熟练的用户能高效率地使用图像处理系统,IRSA-2 采用了命令行与 MENU 相结合的管理系统。

在命令行管理方式下,用户按照规定的命令格式,可以直接进入所希望的处理功能。执行处理功能所需要的各种参数,如输入出图像文件名、处理图像的范围和其它有关参数等均在命令行中给出,以替代 MENU 结构下的人机对话输入。

命令行管理方式,是通过命令行解释程序和命令行功能控制块 (CLFCB) 来完成的。

解释程序按照一定的格式接受输入命令行, 对命令行进行分离、解释, 通过功能控制块传递参数, 最后将控制交给相应的应用程序。应用程序运行之后, 从功能控制块获得所需参数, 完成处理功能, 最后返回到命令行解释程序, 准备接收下一个命令行。

1. 命令行功能控制块

命令行功能控制块是一个 256 字的数据块, 用来在管理程序与应用程序之间传递各种参数, 图像文件名和其它辅助文件名。命令行解释程序将从命令行中获得的各种参数按一定的格式写入命令行功能控制块, 然后将控制转给应用程序。应用程序运行之后, 再从命令行控制块中读出这些参数, 图像文件名以及其它辅助文件名, 赋给相应的变量。

命令行功能控制块的分配如图 3 所示。整个数据块分为四个部分: 参数区, 文件名区及数据块控制字。

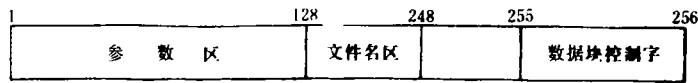


图 3 功能控制块

Fig. 3 The Functional Control Block

参数区 (1—128): 命令行功能控制块的前 128 个字规定为参数区。每个参数均用两个字表示, 最大允许参数量为 64。对任何一个程序, 这一数目都是足够的。

文件名区 (129—248): 包括 120 个字。每 20 个字表示一个文件名, 共可容纳 6 个文件名。

未用 (249—255):

数据块控制字 (256): 在命令行管理方式下, 命令行可以有一定的简化格式, 应用程序对未规定的参数自动赋给缺席值。但是每个命令行中都有若干个基本参数, 如果不给定这些基本参数, 应用程序就无法运行, 管理程序认为这是不确定状态。对于不确定状态, 应用程序将重新用人机交互方式提示用户规定参数。为了区别参数是否充分, 设置了数据控制字。应用程序在读出参数之前, 首先检查数据块控制字。当参数充分时继续进行, 否则以人机交互方式提示用户输入参数及各种有关文件名。

2. 命令行格式

每个命令行包括命令关键字, 参数和文件名, 各部分之间用分离符分开。一般的格式为:

命令行=命令关键字[(参数 1, 参数 2, …/)文件名 1, 文件名 2…]

其中, 参数为整型或实型数值参数。文件名包括图像文件名和其它辅助文件名。对于图像文件名, 其格式为:

图像文件名=FILENAME[(NBAND: B1, B2, B3, B4)]

式中, NBAND: 规定使用的图像波段数,

B1, B2, B3, B4: 相应于 NBAND 的波段号。

在命令行方式下, 各种参数一次给出, 一般无需再用交互方式输入。命令行还具有简化格式。如果命令行中不规定任何参数, 而只给出了命令关键字, 则解释程序认为是不确

定状态,它将重新控制相应的应用程序进入人机交互方式,提示用户从终端输入所需的参数。这时运行应用程序的状态与 MENU 状态完全相同。

下面以统计、打印图像直方图命令行和在彩色显示器上显示磁盘文件命令行为例,简单说明命令行格式和用法。

(1) 统计、打印直方图

语法: TEST[(/IN, OUT/)/FILENAME 1, (FILENAME 2)].

功能: 在 CRT 终端上显示图像的直方图。

参数: ·IN· 类型: 整型; 缺席参数: 0.

当 IN 为 1 时, FILENAME 1 规定为输入直方图文件,

当 IN 为 0 时, FILENAME 1 为输入图像文件。

·OUT· 类型: 整型; 缺席参数: 0.

当 OUT 为 1 时, FILENAME 2 规定为直方图输出文件:

当 OUT 为 0 时,无直方图输出文件产生,

·FILENAME 1· 输入图像文件名或输入直方图文件名。

·FILENAME 2· 输出直方图文件名。

举例: TFSTCL NAME (1:2)

TESTCL/0, 1/NAME(1:2), HISTOUT

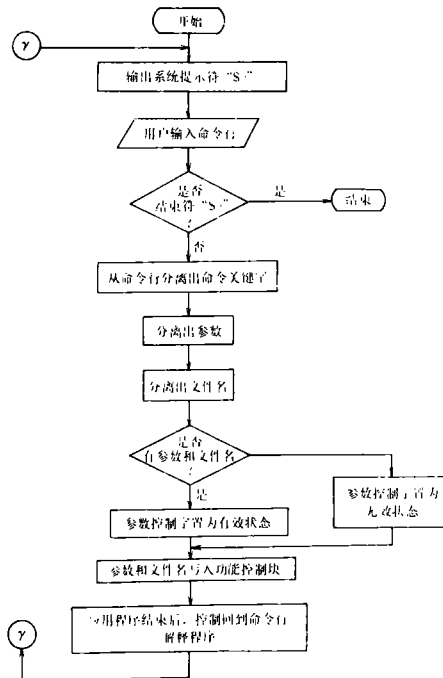


图 4 命令行解释程序图

Fig. 4 The Flow chart of the Command line Interpreter

TESTCL/1/HISTIN

TESTCL

以上均为合法的命令。

(2) 显示磁盘图像

语法: DISCL/XO, YO, XSTEP, YSTEP, IMG/FILENAME.

功能: 将磁盘图像文件送往彩色监视器并显示。

参数: ·XO· 类型: 整型; 缺席参数: 1.

X 方向的偏移量

·YO· 类型: 整型; 缺席参数: 1

Y 方向的偏移量

·XSTEP· 类型: 实型; 缺席参数: 1.0

X 方向的步长。XSTEP 可以是任何非负实数, 大于 1 表示缩小, 小于 1 表示放大。

·YSTEP· 类型: 实型; 缺席参数: 1.0, Y 方向的步长, 与 XSTEP 相对应。

·IMG· 类型: 整型; 缺席参数: 1, 显示器图像平面号。

·FILENAME· 磁盘图像文件名。

举例: DISCL/1, 1, 1, 1, 1/BJ·IMG(1; 1)

DISCL/20, 20, 20, 20, 2/BJ·IMG(1:2)

DISCL BJ·IMG

图 4 为命令行解释程序的流程图。

(三) 并发程序设计

IRSA-2 系统是建立在 AOS 操作系统之上的图像处理软件系统。AOS 是一种能平行处理的多用户操作系统。为了充分发挥 AOS 操作系统的功能, 图像处理软件系统也应具有多用户平行处理的特点。从简单顺序程序过渡到并发程序有许多特殊问题需要解决, 其中最主要的是数据共享问题。

1. 并发程序概念

设有 n 个进程 $P_1, P_2, P_3, \dots, P_n$, 若在某一时刻它们同处于开始点与终止点之间, 则称这些进程是并发的。并发进程的动作序列, 在时间上可以是重叠的, 也可以是交叉的。若有三台处理机, 进程 P_1, P_2, P_3 分别在这三台处理机上同时执行, 则这三个进程的动作在时间上是重叠的, 如图 5(a) 所示。若只有一台处理机, 则在任一时刻只能执行一个进程, 三个进程交替执行, 如图 5(b) 所示。若假定每一个进程的执行速度都是正的 (即任何一个动作都在有限时间内完成), 则在图 5(b) 中每个进程被中断的时刻完全是随机的, 因此, 图 5(b) 与图 5(a) 实质上是相同的, 只不过图 5(a) 是真正的并行, 图 5(b) 是逻辑上的并行而已。

2. 进程之间的直接相互制约关系

若一组并发进程中任何两个进程之间都不存在公共变量, 则称这组进程为不相交的进程。在不相交进程中, 每一个进程都独立于其它进程, 它的运行环境与顺序进程一样, 是封闭的, 即它的运行环境不为别的进程所改变。因此, 每个进程的运行结果是确定的。

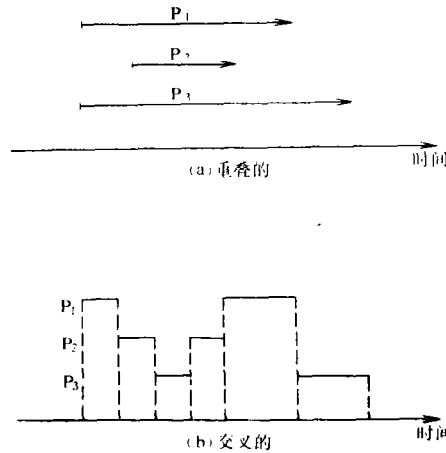


图 5 并发进程

Fig. 5 The Concurrency Process

这就是说,不相交进程并发执行不会引起与时间有关的错误。

但是,如果并发进程之间存在公共变量,就出现了相互制约的关系。即进程中有一些

动作,这些动作的执行,对其它进程的执行产生影响。进程之间的制约关系具体表现为竞争和协作。竞争就是对共享资源的竞争。例如,各命令行解释进程对命令行功能控制块的共享。在共享数据时,各进程对数据的使用可以是各种各样的:读出、写入、修改或删除等。协作是指各进程之间交换数据,例如进程间的通讯技术(Interprocess Communication, IPC)。交换数据的目的是通知或委托其它进程做某件事。

3. 并发性控制的要求

并发程序之间存在相互制约的关系,因此必须对这种关系加以控制,否则将无法达到预期的目的。对进程之间共享数据的控制必须满足下列要求:

(1) 安全性。在一个时刻只允许一个进程使用共享数据。否则,对共享数据来说,操作的结果往往是破坏性的。

(2) 活动性。活动性表现为两个方面,一是当若干个进程竞争共享数据时,则应在有限时间内达到目的,即它们不应因互相阻止而导致彼此

都永远不能使用,即所谓死锁;二是任一进程在使用共享数据时,必须在有限的时间内释放。

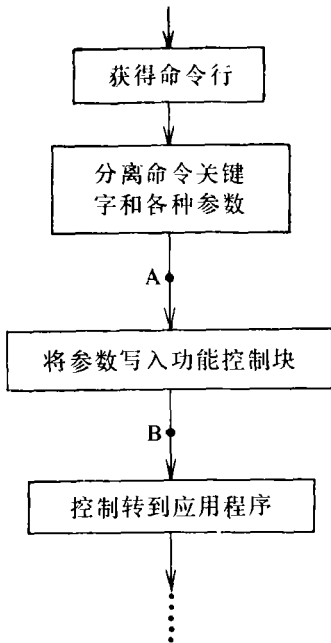


图 6 简单的命令行解释程序

Fig. 6 The Simplified Command Line Interpreter Flow Chart

(3) 合理性。合理性包括两个方面,第一,对各进程使用共享数据的顺序不作任何规定,当一个进程欲使用共享数据时,只要其它进程不在使用,就应允许该进程使用;第二,当同时有多个进程要求使用时,则不规定优先权。

4. 并发管理程序设计

根据上述对并发程序的要求来考虑管理程序的实现方法。对多级分支清单管理程序,整个过程无公共变量,只有控制权在 MENU 与应用程序之间的转换。应用程序自身独立地完成某项处理功能,与外界无任何关系,即为封闭的。因此, MENU 结构管理程序可以直接用于多用户并发运行。对命令行解释程序,由于需要向应用程序传递参数,因此,存在一个共享数据块 CLFCB。各命令行解释进程将对此共享数据块进行竞争。

简单的命令行解释程序的流程图如图 6 所示。当一个进程将参数写入 CLFCB 之后,若在应用程序将参数读出之前,第二个进程又将参数写入,则第一次写入的内容将被破坏,从而会使该应用程序读出错误的参数。

为保证运行的安全性,在参数写入之后至应用程序读出参数之前这段时间内,应排斥其它进程再访问 CLFCB。为此,在命令行功能控制块中加入一个进程控制字,如图 7 所示。利用进程控制字来表示 CLFCB 被占用或空闲。

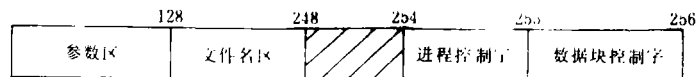


图 7 在功能控制块中加入控制字

Fig. 7 The Functional Control Words added into the Functional Control Block

在向 CLFCB 写入参数之前,命令行解释进程首先检查 CLFCB 的状态,若进程控制字为占用状态,则该进程被延迟。经过一定的延迟之后,再重新检查 CLFCB 的状态,直到空闲为止。当 CLFCB 空闲时,进程首先将参数写入 CLFCB,然后置进程控制字为占用状态,再把控制传递给应用程序。在应用程序中,先读出参数,然后释放 CLFCB,允许其它进程使用。

因此,命令行解释程序的流程图将修改成图 8 所示。此流程可保证共享数据 CLFCB 一次只为一个进程使用。它保证了完全性,但活动性却受到破坏。当进程在写入 CLFCB 之后,而尚未进入应用程序(即在应用程序释放 CLFCB 之前)时被打断,则 CLFCB 将永久被占用,而使系统发生死锁。

在一般情况下,各进程访问 CLFCB,从占用到释放,在有限的时间内一定可以完成。因此,可以设置一个时间阈值,当进程检查 CLFCB 次数超过阈值时,将认为发生了死锁。系统将对 CLFCB 进行初始化,这样就可以解决系统死锁问题。

5. 实现流程图

综合以上考虑,最终使用的流程图如图 9 所示。该方法可较好地保证安全性、活动性及合理性。成功地完成了从简单顺序程序到并发程序的过渡,从而使图像处理子系统具

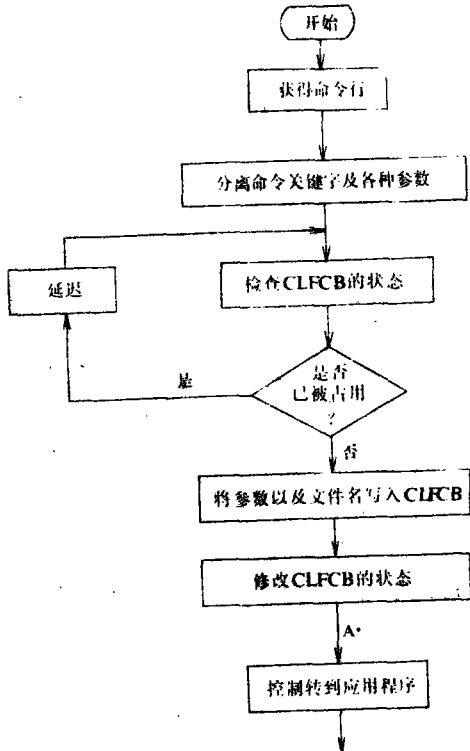


图 8 检查占用状态的流程

Fig. 8 The Block Diagram Shows the detection of Busy State

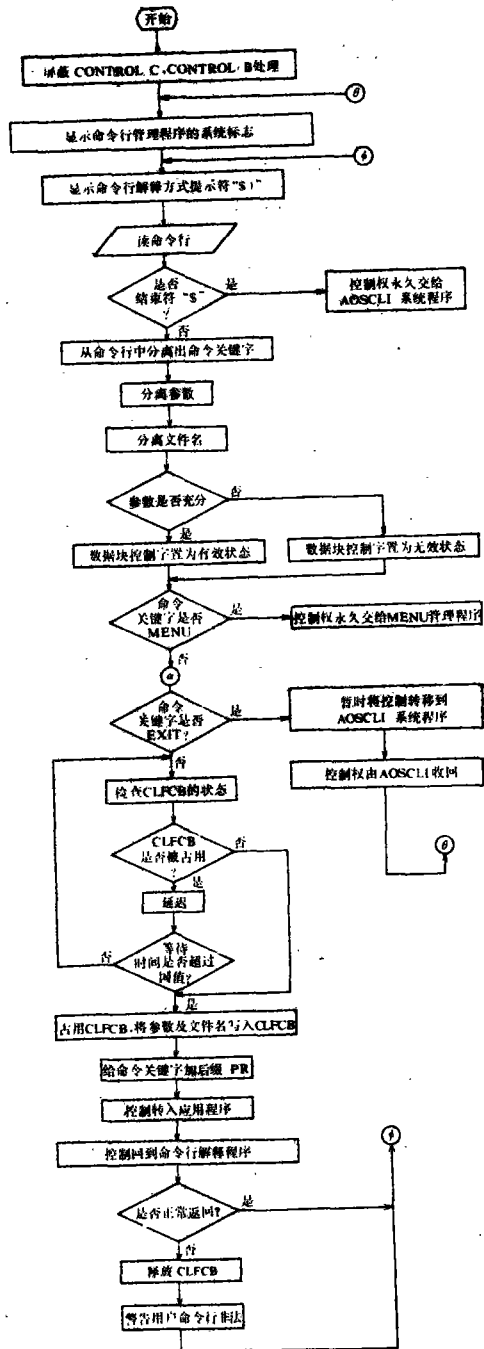


图 9 命令行解释程序图

Fig. 9 The Flow chart of Command line Interpreter

三、结 束 语

使用证明, 本管理软件设计合理, 性能完善。在本系统管理下, 各用户能有效地使用各种图像处理应用程序。它真正起到了用户与处理功能之间的接口作用, 使 IRSA-2 系统成为运行性的遥感图像处理系统。

进行了命令行与多级功能分支清单相结合的结构设计, 因此, 系统能方便、灵活、可靠和高效地运行。系统具有较好的扩充性能, 能满足用户程序纳入及软件扩充的需要。实现了并发程序设计, 使系统具备了并行性, 实现了多用户, 充分发挥了计算机的效率。

本系统管理程序使用标准 FORTRAN 语言编写, 机器针对性语句用专门的子程序调用来代替。软件移植时, 只要改变这些子程序即可, 因此, 本软件易于移植和推广。

参 考 文 献

- [1] 杨世仁, 李丽, IRSA-2 遥感图像分析处理系统, 环境遥感, 1(1), 1986.
- [2] 李丽, 高朋, 虚拟 COMTAL 显示器的设计与实现, 环境遥感, 1(3), 1986.

THE SYSTEM MANAGEMENT SOFTWARE OF THE DIGITAL IMAGE PROCESSING SYSTEM —THE DEVELOPMENT OF IRSA-2 DIGITAL IMAGE PROCESSING SYSTEM (PART 3)

Gan Peng Li Li

(Institute of Remote Sensing Application, Academy Sinica)

Abstract

This paper describes the system management software of the IRSA-2 Digital Image Processing System. Both the tree structured functional menu and the command line interpreter are developed in the IRSA-2 suboperating system which makes the IRSA-2 Digital Image Processing System more convenient and efficient in application. User developed image processing software can be easily brought into the IRSA-2 system. Therefore, IRSA-2 system is expandable and transferable. The digital image processing programs are processed as concurrent processes. The multi-user capability and efficiency of the entire system are thus improved significantly.