

DIGITAL 命令语言

介绍和使用

毛时言 编

国防科技大学计算机研究所

一九八五年六月

前 言

DIGITAL 命令语言 (简称 *DCL*) 对建立和控制一个进程的运行环境是一个很有用的工具, 一个命令是对操作系统一个特定动作的直接请求; 命令串可以组成命令过程。*DCL* 为 *VAX/VMS* 用户提供了一批丰富的命令, 最基本的有一百七十多条, 它们包括: 程序开发和执行、资源分配、环境控制、作业控制、文件维护、实用程序和操作控制等命令, 主要用于:

1. 交互程序的开发和执行
2. 设备和数据文件的处理
3. 交互和批处理程序的执行和控制

各类命令的作用和用法, 除了实用程序另作专门描述, 不包括在此外, 其余均作了介绍, 对命令的格式、功能和使用时进行了描述。本文中介绍的命令仅是 *DCL* 命令中的一部分, 选择常用的, 主要的命令加以阐述。

由于时间仓促, 文中难免有错, 不对的地方, 请批评指正。

编 者

一九八五年四月

目 录

第一章 用户注册	1
第二章 命令格式	4
第三章 命令过程	7
第四章 命令概貌	10
第五章 命令分类	21
一. 程序开发和执行命令	21
1. 程序开发步骤	21
2. EDIT 命令	23
(一) EDT 编辑	24
(二) SOS 编辑	40
(三) SLP 编辑	56
3. FORTRAN 命令	61
4. MACRO 命令	66
5. LINK 命令	72
6. RUN 命令	79
二. 资源分配和控制命令	84
1. ALLOCATE 命令	84
2. DEALLOCATE 命令	84
3. MOUNT 命令	85

4. DISMOUNT 命令	89
5. SET WORK-SET 命令	91
6. SET PROTECTION/DEVICE 命令	92
7. SHOW PROCESS/QUOTA 命令	92
8. SET PROCESS/(NO) RESOURE-WAIT 命令	93
9. SHOW MEMORY 命令	94
10. SHOW QUOTA 命令	95

三. 文件处理命令 96

1. DIRECTORY 命令	96
2. COPY 命令	100
3. TYPE 命令	103
4. INITIALIZE 命令	104
5. CREATE/DIRECTORY 命令	108
6. DELETE 命令	111
7. PURGE 命令	112
8. PRINT 命令	113
9. DIFFERENCE 命令	115
10. SEACH 命令	119
11. DUMP 命令	121
12. APPEND 命令	123
13. RENAME 命令	125
14. UNLOCK 命令	126
15. CLOSE 命令	127
16. WRITE 命令	127
17. OPEN 命令	127
18. SET PROTECTION 命令	129

四. 环境控制命令	130
1. ASSIGN 命令	130
2. DEFINE 命令	131
3. DEASSIGN 命令	132
4. SET UIC 命令	133
5. SET DEVICE 命令	134
6. SHOW TIME 命令	135
7. SET TIME 命令	135
8. SHOW DEVICE 命令	135
9. SHOW DEFAULT 命令	138
10. SET DEFAULT 命令	138
11. SHOW LOGICAL 命令	139
12. SET PRINTER 命令	140
13. SET PROCESS 命令	141
14. SHOW PROCESS 命令	143
15. SHOW ERROR 命令	145
16. SET PASSWORD 命令	146
17. SET DIRECTORY 命令	146
18. SHOW USER 命令	147
19. SHOW SYSTEM 命令	147

五. 作业控制命令	148
1. STOP 命令	148
2. CONTINUE 命令	149
3. IF 命令	150
4. GOTO 命令	150
5. ON 命令	151
6. INITIALIZE/QUEUE 命令	152

7.	START/QUEUE 命令	153
8.	STOP/QUEUE 命令	154
9.	STOP/REQUEUE 命令	154
10.	DELETE/QUEUE 命令	155
11.	SHOW QUEUE 命令	155
12.	ASSIGN/MERGE 命令	155
13.	SET QUEUE/ENTRY 命令	156

六. 操作控制命令 159

1.	LOGIN 过程	159
2.	LOGOUT 命令	160
3.	SUBMIT 命令	160
4.	REQUEST 命令	164
5.	REPLY 命令	164

第一章 用户注册

当系统被引导起来后，并且终端已联机，则可在终端或控制台上按 <ESCAPE> 键、<CTRL/Y>、<CTRL/C> 或 <RETURN> 键，出现：

Username : 用户名 <RETURN>

Password : 口令 <RETURN>

Welcome to VAX/VMS Version 3.0

这时，系统已为用户建立进程，并为进程建立下列各种缺省特性：

1. 帐号或帐户名 —— 为用户记录使用计算机资源所用的登记名。

例：ACCOUNT : VMS

表示帐号为 VMS，具有该帐号的用户，使用计算机资源的记录登记在 VMS 帐号上。

2. 用户识别码 (UIC) —— 该识别码提供同组号及同组中的成员号，同组中的成员之间可以相互通讯，可以读，执行对方的文件，可以共享系统资源。

例：UIC = [377, 001]

由组号和成员号组成。

组号 —— 377

成员号 —— 001

每一组可以有 () - 377) 8 个成员。

3. 缺省磁盘设备和目录名 —— 提供用户建立和使用文件的缺省设备和目录。

例: `$ SHOW DEFAULT`

`DRAO: [ABCDIR]`

表示用户所用的缺省设备为 DRAO 磁盘, 缺省目录是 [ABCDIR]。

4. 进程的输入、输出和错误流的缺省设备 —— 给系统和进程提供, 与命令及输入、输出用。

可用进程名 `SYS$INPUT`, `SYS$OUTPUT`, `SYS$ERROR` 分别等价于进程的输入、输出、错误流的缺省设备名。

例: `SYS$INPUT = TTA1`

↑ ↑
等价进程名 进程的输入缺省设备

`SYS$OUTPUT = TTA1`

↑ ↑
等价进程名 进程的输出缺省设备

`SYS$ERROR = TTA1`

↑ ↑
等价进程名 进程的错误流的缺省设备

进程的输入、输出、错误流的缺省设备, 根据进程执行的方式不同而不同, 附表如下:

进程逻辑名	进程的缺省设备
<code>SYS\$COMMAND</code> (原型的 <code>SYS\$INPUT</code> 流)	交互方式：终端 批方式：磁盘 命令过程：终端
<code>SYS\$INPUT</code> (进程的缺省输入流)	交互方式：终端 批方式：磁盘 命令过程：磁盘
<code>SYS\$OUTPUT</code> (进程的缺省输出流)	交互方式：终端 批方式：磁盘 命令过程：终端
<code>SYS\$ERROR</code> (系统写消息的缺省设备)	交互方式：终端 批方式：磁盘 命令过程：终端

5. 特权和资源定额——用户能使用的系统资源和系统功能的多少。

例：

特权：`BYPASS`——能旁路 UIC 保护，可对全部文件执行读、写、执行及删除。

`EXQUOTA`——允许超出资源定额，能使用户使用的磁盘空间超过系统给该用户的定额。

资源定额：`CPUTIME`——规定进程使用的 CPU 时间极限。

6. 命令解释程序——阅读和翻译输入请求，表明用户所使用的命令隶属于哪一个操作系统之下，确定了命令使用的范围。

例：MCR → 在 RSX-11M 下，提示符 >

DCL → 在 VAX/VMS 下，提示符 \$

当命令解释程序是 MCR 时，即当 CCL=MCR 时，则用户在 RSX-11M 操作系统下使用命令。

当命令解释程序是 DCL 时，即当 CCL=DCL 时，则用户在 VAX/VMS 操作系统下使用命令。

第二章 命令格式

一条命令的一般格式是：

\$ [标号:] 命令名 [选项] [参数-1] ... [参数-n]

其中：

① \$——必须出现在被执行命令的第一个位置上。

② []——方括号用于括起来快速输入的值。

例如：\$ COPY [选项] 意为用户可不必提供任何选项就能正确地发出 COPY 命令。

③ 标号：——可以标识任何命令，可以作命令的同义语。借助于 GOTO 命令，标号可以被用于转移控制流，也可用于文件。标号放在命令名前边，并用冒号 [:] 隔开。

例：① \$ ABC: SHOW DAY

1-JAN-1985 12:00:14

② 7 BB:

:

\$ AA:GOTO BB:

- ④ 命令名 —— 指出命令完成的动作。系统的所有命令均采用英语动词，它描述主要执行的功能。

例：\$ TYPE ABC.FOR

表示在当前的首缺输出流上显示类型为 FOR 的文件 ABC 的内容。

- ⑤ 限定词 —— 用于修改命令的首缺操作。限定词总是以一个斜线 (/) 开头，命令名和参数都可被限定。

例：i) \$ PRINT /JOB_COUNT=3 MYFILE.DAT

其中：/JOB_COUNT=4 是命令限定词，表示请求每个作业打印 4 次。若不写它，则执行首缺操作——整个作业打印 1 次。

ii) \$ PRINT /DELETE MYFILE.DAT

其中：/DELETE 是文件限定词，表示在打印数据文件 MYFILE.DAT 后，将该文件删去。若不指定，则在打完该文件后不删去，首缺操作为 /NODELETE。

限定词往往带有限定值，在限定词与限定值之间必须用符号 (=) 或冒号 (:) 隔开。若限定词需要一个值的列表，该表必须用圆括号括起来。

例：① /COUNT_JOB=3

 ↑ ↑
 限定词 限定值

② /BLOCKS=(5:5, E:6)

 ↑ ↑
 限定词 限定值的列表

此限定词与限定值列表的含义为：从第 5 块开始到第 6 块结束。

另外，假脱词与假脱值里均不允许有空格。一些假脱词前冠以 NO 取反意。

例： /OBJECT 产生目标文件
/NOOBJECT 不产生目标文件

② 参数——规定在执行命令时用的值，或是在进一步定义命令所起的作用。

命令名和第一个参数之间至少要用一个空格或制表符隔开，参数间用一个或多个空格和 / 或制表符隔开，交互用户可以在回答提示时提供参数，若参数是个列表值，则列表上的项要用逗号隔开。

例： \$DELETE A, B, C

删除 A, B 和 C 文件。

\$COPY A, B, C

将文件 A 和 B 拷贝，移到 C 文件，若 C 原来有文件，则 C 原来文件被替换了。

在参数中，往往需要用 (+) 号将多个参数联接起来进行处理。即：

串联字符 (+) —— 表示把左边的文件和右边的文件联接起来，然后作为一个文件进行处理。

例： \$FORTRAN A+B

FORTRAN 命令连接 A.FOR 和 B.FOR 文件，然后作为一个文件被 FORTRAN 编译程序编译，产生一个目标文件 A.OBJ。

\$LINK A+B/EXE=C

LINK 命令将 A.OBJ 目标模块和 B.OBJ 目标模块联接起来，然后作为一个目标模块被连接程序连接。

⑦ 续写字符(—)——表示该命令在下一行继续

例: \$ COPY A.DAT—
\$—;2 B.DAT

表示把版本为2的数据文件A.DAT;2复制到B.DAT文件上去。

⑧ 注解符(!)——表示注解的开始, 注解只说明用户信息而对命令的处理不起作用。

例: \$ COPY A.DAT B.DAT
! FILE A.DAT TO FILE B.DAT
! COMMAND PROCEDURE FOLLOWS

此三行中, 后二行均为注解, 起作用的是第一行。在DCL命令中, 小写字母同大写字母一样被处理, 除非它们是写在引用串中。

所有命令名, 限定词和参数关键字, 都可以缩写为头4个字母。

第三章 命令过程

除了执行单条命令, 用户还可以用DCL命令构成命令语言程序, 即建立一个命令过程。命令过程是一个文件, 该文件由一些包含DCL命令和可供选择的数据组成, 命令过程中的每个命令前必须以小写字母开始, 没有的解称为数据行。一旦命令过程建立以后, 只用一个单独命令(即@)就可以执行命令过程中的所有命令。

可以请求系统把命令过程作为一个成批作业来处理, 该方法可使用户终端保持自由以进行交互工作。

调用格式为:

@ 文件说明 [p_1 [p_2 [... p_8]]]

命令限定词: /OUTPUT=文件说明

其中:

文件说明——即被执行的命令过程。此过程的命令已从文件或从设备中装入。若不指定文件类型, 则默认类型为 COM。

[p_1 [p_2 [... p_8]]]——允许命令过程使用的 1 到 8 个任选参数, 可以是字符串值或数字串值, 这些符号对命令过程来说是面部的。未规定的参数解释为空串。参数之间须用空格隔开。

参数规定: 如果第一个参数的数位用符号 (/) 开头, 则必须用引号对此参数括起。

如果一个参数包含有嵌入空白或文字小写字母时, 则须将这一参数用引号括起。

如果一个参数包含有文字引号, 则必须将每个字符都用引号括起, 并在字符串内使用双引号。

例:

① \$ @ MY.COM

执行命令过程 MY.COM。

MY.COM 的内容为: \$ FORTRAN A.FOR

\$ LINK A.OBJ

\$ RUN A.EXE

因此, 执行命令过程 MY.COM 也就是顺序执行命令

\$ FOR A.FOR

\$ LINK A.OBJ

\$ RUN A.EXE

的过程。由于没有指定参数, 前 $p_1, \dots, p_8$ 为空串。

② \$ LINK @ S.COM

其中: S.COM 是一个命令过程。

S.COM 内容为:

LINK /EXE A

B, C/EXE=D

在连接程序连接过程中, 使用了命令过程 S.COM。因此, LINK 命令的具体执行步骤为:

\$ LINK /EXE A

\$ LINK B, C/EXE=D

连接程序连接目标模块 A.OBJ, 并建立可执行映像 A.EXE, 接着, 连接程序又连接目标模块 B.OBJ 和 C.OBJ, 最终建立可执行映像文件 D.EXE。由于没有指定参数, 故 P1, ...PB 为空串。

③ \$ @ TEST "Never say" "quit"

此例中, 命令过程为 TEST.COM。Never say "quit" 为参数 P1, 由于参数中含有文字引号, 所以参数必须用引号括起。

④ \$ @ MASTER/OUTPUT=MASTER.LOG

此命令执行命令过程 MASTER.COM, 输出信息写在 MASTER.LOG 文件上。

第四章 命令概貌

VAX/VMS 有许多许多条命令，最基本的有一百七十多条，它们内容非常丰富，功能强而广泛，使用灵活方便，随着指定项的不同和参数的不同，一条命令可以有多种功能和特性。VAX/VMS 系统的大量命令为用户提供极大方便，为了使读者有一个概貌，现将 DCL 的基本命令及功能按字母顺序列表如下：

DCL 命令	功 能
ACCOUNTWG	调用系统实用程序选择、分类、列表和说明记账数据。
ALLOCATE	指定一个设备为一个单用户使用，而且给设备任意分配一个逻辑名，一旦某个设备已被独占，其它用户就不能访问。
ANALYZE/CRASH_DUMP	调用系统转储分析程序检查指定的转储文件。
ANALYZE/DISK_STRUCTURE	调用 VAX-11 检验实用程序检查 FILE-11 一般结构和二重结构磁盘卷宗。
ANALYZE/IMAGE	提供映像文件内容的说明。
ANALYZE/OBJECT	提供目标文件内容的说明。
ANALYZE/RMS_FILE	调用 ANALYZE/RMS_FILE 实用程序检查和分析 VAX-11 RMS 文件的内部结构。
ANALYZE/SYSTEM	调用系统转储分析程序检查正在运行的系统。
APPEND	把一个或多个文件加到另一个文件的

ASSIGN

末尾。

定义一个与逻辑名相关的文件说明或一个设备名。

ASSIGN/MERGE

把作业从一个队列移到另一个队列中。

ASSIGN/QUEUE

给一个设备分配一个逻辑队列。

ATTACH

指定被连接的进程名，在终端上控制作业从一个进程转到另一个进程。

BACKUP

调用 BACKUP 实用程序以磁盘或磁带保存或恢复文件。

BASIC

调用 VAX-11 BASIC 编译程序输入和编译 BASIC 语言源语句。

BASIC/RSX-11

调用 PDP-11 BASIC-PLUS-2 编译程序开始一个 BASIC 会话。

BLISS

调用 VAX-11 BLISS-32 编译程序以编译一个或多个 BLISS 源。

CANCEL

打止一个预定在进程中执行的映象的周期执行。

CC

调用 VAX-11 C 编译程序编译 C 源程序。

CLOSE

取消一个顺序文件或设备为输入或输出通道。

COBOL

调用 VAX-11 COBOL 编译程序编译 COBOL 语言源语句。

COBOL/74

调用 VAX-11 COBOL-74 编译程序以编译 COBOL 语言源语句。

COBOL/RSX-11

调用 PDP-11 COBOL-74/VAX 编译程序以编译 COBOL 语言源语句。

CONTINUE

继续执行一个被中断的命令。

• 11 •