1．DEBUG程序的调用

在DOS提示符下，可键入命令：

C>DEBUG [d:][path][文件名][参数1][参数2]

其中文件名是被调试文件的名字，它必须是可执行文件(EXE)，两个参数是运行被调试文件时所需要的命令参数，在**DEBUG**程序调入后，出现提示符”\_”，此时，可键入所需的**DEBUG**命令。

2．DEBUG的主要命令

（1）显示内存单元内容的命令D,格式为：

－D [地址] 或

－D [范围]

（2）修改内存单元内容的命令E，它有两种格式

1）用给定的内容代替指定范围的单元内容：

－E 地址 内容表

例如：－E DS: 100 F3 “XYZ” 8D

其中F3，“X”，“Y”，“Z”和8D各占一个字节，用这5个字节代替原内存单元DS:100到104的内容，“X”，“Y”，“Z”将分别按它们的ASCII码值代入。

2）逐个单元相继修改

－E 地址

例如：

－E 100：

18E4: 0100 89 78

此命令是将原100号单元的内容89修改为78。78是程序员键入的。

（3）检查和修改寄存器内容的命令R，它有三种方式：

1）显示CPU内部所有寄存器的内容和标志位状态，格式为：

－R

R命令显示中标志位状态的含义如下表所示：

|  |  |  |
| --- | --- | --- |
| 标 志 名 | 置位 | 复位 |
| 溢出overflow（是/否） | OV | NV |
| 方向direction（减量/增量） | DN | UP |
| 中断interrupt（允许/屏蔽） | EI | DI |
| 符号sign（负/正） | NG | PL |
| 零zero（是/否） | ZR | NZ |
| 辅助进位auxiliary carry（是/否） | AC | NA |
| 奇偶parity（偶/奇） | PE | PO |
| 进位carry（是/否） | CY | NC |

2）显示和修改某个指定寄存器内容，格式为：

－R 寄存器名

例如打入：－R AX

系统将响应如下：AX F1F4:

表示AX当前内容为F1F4，此时若不对其做修改，可按enter键，否则，键入修改后内容，如：

－R BX

BX 0369

059F

则BX内容由0369改为059F

1）显示和修改标志位状态，命令格式为：

－RF

系统将给出响应，如

OV DN EI NG ZR AC PE CY－

这时若不做修改可按enter键，否则在“－”之后键入修改值，键入顺序任意。如

OV DN EI NG ZR AC PE CY－PONZDINV

（4）运行命令G，格式为：

－G [=地址1][地址2[地址3…]]

其中地址1规定了运行的起始地址，后面的若干地址均为断点地址。

（5）追踪命令T，有两种格式：

2）逐条指令追踪：

－T [=地址]

该命令从指定地址起执行一条指令后停下来，显示寄存器的内容和状态值。

3）多条指令追踪：

－T [=地址] [值]

该命令从指定地址起执行n条命令后停下来，n由值确定。

（6）汇编命令A,格式为：

－A [地址]

该命令从指定地址开始允许输入汇编语句，把它们汇编成机器代码相继存放在从指定地址开始的存储器中。

（7）反汇编命令U，有两种格式：

1）－U [地址]

该命令从指定地址开始，反汇编32个字节，若地址省略，则从上一个U命令的最后一条指令的下一单元开始显示32个字节。

2）－U 范围

该命令对指定范围的内存单元进行反汇编，例如：

－U 04BA: 0100 0108或

－U 04BA: 0100 L9

此二命令是等效的。

（8）退出DEBUG命令Q，该命令格式为：

Q 退出DEBUG程序，返回DOS。