北京希望电脑公司

HOPE

适用于 IBM PC AT 286、386 及其兼容机

微型机编程工具和编程环境

适用于 IBM PC AT 286、386 及其兼容机

微型机编程工具和编程环境

北京希望电脑公司

一九九一年六月

前　言

本书的推出基于下列事实：许多实用编程问题在子程序大全之类的书中讲述得太过简单，相反在许多教科书中又讲得太过复杂和太过专业化．以我们的经验来看，编程员在面对这样的问题时，最好以专家如何解决这类问题经验为指导，例如，在写一个把文本行存入缓冲区的子程序，程序员最好参考一本讲述文本管理技术的书，也就是描述了如何用开始和结束字节，字节长度，控制码来组织内存，以及文本如果扩展和紧缩以插入和删除文件中的元素．由于该子程序基于一般的文本管理算法，所以只须提供基础算法，至于如何变成代码，已经很简单了．

因为该书是一流专家的经验的结晶，要求读者具备足够的编程经验．

本书以完全程序和代码结构形式提供了大量实用的例子，例如，脱机打印，设备内部通讯，和磁盘文件传输．本书对作者认为极有用的中断给予详细的说明．

编者　韩耀武、华真、德和　1991.6.28

目录

第一部分　编程工具和编程环境 …… (1)

第一章　编程工具 …… (2)

1.0　引言

1.1　程序开发硬件 …… (2)

1.2　参考资料 …… (2)

1.3　软件工具 …… (3)

1.3.1　文本编辑器 …… (3)

1.3.2　汇编器 …… (3)

1.3.3　连接器 …… (4)

1.3.4　库和库设备 …… (4)

1.3.4.1　Microsoft PASCAL 和 FORTRAN 数学库 …… (5)

1.3.4.2　Intel 8087 模拟器和支持库 …… (7)

1.3.4.3　其它库 …… (7)

1.3.5　调试程序 …… (7)

1.3.6　辅助设备 …… (8)

1.4　选择开发系统 …… (8)

1.4.1　Microsoft 宏汇编器（Microsoft Macro Assembler） …… (8)

1.4.2　Intel ASM86 …… (9)

1.5　软件工程 …… (9)

1.5.1　程序开发的阶段 …… (10)

1.5.1.1　定义及资源分配 …… (10)

1.5.1.2　程序设计 …… (10)

1.5.1.3　程序编码 …… (11)

1.5.1.4　程序文档 …… (11)

1.5.1.5　正确性检验 …… (13)

1.5.2　工程技术 …… (15)

1.5.2.1　模块化 …… (15)

1.5.2.2　标准化 …… (15)

1.5.2.3　程序流程图 …… (16)

第二章　程序环境 …… (18)

2.0　引言

2.1 内存结构 …… (18)
2.1.1 内存空间 …… (18)
2.1.2 物理地址计算 …… (19)
2.2 程序驻留 …… (19)
2.2.1 段式内存 …… (19)
2.2.2 逻辑地址 …… (20)
2.2.3 附加段寄存器 …… (21)
2.2.4 例外段分配 …… (22)
2.2.5 重新定位和装入 …… (26)
2.3 程序结构 …… (26)
2.3.1 惯用的段定义 …… (26)
2.3.2 SEGMENT 和 ENDS 指令 …… (28)
2.3.3 ASSUME 指令 …… (29)
2.3.4 GROUP 指令 …… (30)
2.3.5 程序设计考虑因素 …… (31)
2.4 共同驻留的软件 …… (32)
2.4.1 回避系统软件 …… (33)
2.4.2 IBM 的内存映射 …… (33)
2.4.3 IBM BIOS …… (33)
2.4.4 自举加载器 …… (35)
2.4.5 DOS …… (36)
2.4.6 兼容性问题 …… (36)
2.5 生成一个定制的系统 …… (37)
2.5.1 回避 DOS …… (37)
2.5.2 定制自举加载器 …… (38)
2.5.3 配置系统盘 …… (40)

第三章 对硬件编程 …… (44)
3.1 微处理器 …… (44)
3.1.1 8086／8088 处理器家族 …… (44)
3.1.2 8087 数学协处理器家族 …… (45)
3.2 中断 …… (46)
3.2.1 中断机构 …… (46)
3.2.2 中断向量表 …… (47)
3.2.3 获取中断控制 …… (49)
3.2.4 MS DOS 下的中断 …… (51)
3.2.5 保护安装的中断 …… (52)
3.2.6 设置段寄存器 …… (53)
3.2.7 截取执行 …… (55)

3.2.8 中断处理与优先级 …… (57)
3.2.9 外部中断 …… (57)
3.2.10 PS/2 系统上的中断共享 …… (60)
3.3 其它可编程组件 …… (60)
3.3.1 8255 可编程外围界面(PPI) …… (61)
3.3.2 系统定时器 …… (61)
3.3.2.1 访问系统定时器 …… (62)
3.3.2.2 再编程定时器 …… (66)
3.3.3 对扬声器的输出 …… (78)

第二部分 编程技术 …… (85)

第四章 文本管理 …… (86)
4.1 文本处理 …… (86)
4.1.1 字符代码 …… (86)
4.1.2 控制码 …… (87)
4.1.3 文本文件块 …… (88)
4.2 文本编辑器的结构 …… (88)
4.2.1 文本缓冲区 …… (88)
4.2.2 嵌入控制码 …… (89)
4.2.3 终止符字节 …… (89)
4.2.4 位移字节 …… (92)
4.2.5 其它控制字符 …… (95)
4.2.6 用控制码给文本定块 …… (95)
4.2.7 以行为基础和以屏幕为基础的编辑器 …… (98)
4.3 文本文件管理 …… (98)
4.3.1 文本文件结构 …… (98)
4.3.2 文本输入 …… (98)
4.3.3 文本文件显示 …… (98)
4.3.3.1 移动光标实例 …… (99)
4.3.3.2 无光标编辑 …… (110)
4.3.4 视频到内存操作 …… (127)
4.4 编辑文本文件 …… (130)
4.4.1 文件扩展和压缩 …… (130)
4.4.2 文本文件控制 …… (133)
4.4.3 基本的编辑操作 …… (134)
4.4.3.1 插入 …… (134)
4.4.3.2 替换 …… (134)
4.4.3.3 删除 …… (134)

4.5 特殊的文本操作 …… (135)
4.5.1 连续输入 …… (135)
4.5.2 文本排版 …… (147)
4.5.2.1 填加空格 …… (147)
4.5.2.2 调和空间 …… (148)

第五章 数字计算 …… (149)
5.0 引言
5.1 8087 数字协处理器的范围与设计 …… (149)
5.1.1 8087 芯片概述 …… (150)
5.1.2 局限性 …… (150)
5.1.3 8086 与 8087 界面 …… (151)
5.1.4 同步操作 …… (151)
5.2 8087 结构 …… (152)
5.2.1 堆栈寄存器 …… (152)
5.2.2 控制寄存器 …… (153)
5.2.3 状态寄存器 …… (155)
5.2.4 指令与数据指针 …… (156)
5.2.5 8087 标志寄存器 …… (156)
5.3 数字数据类型 …… (157)
5.3.1 数字数据转换 …… (158)
5.3.2 数据类型编码 …… (158)
5.3.3 8087 的非正规数 …… (159)
5.4 对 8087 编程 …… (160)
5.4.1 8087 编码格式 …… (161)
5.4.2 8087 异常处理 …… (161)
5.5 8087 指令组 …… (163)
5.5.1 数据传递指令 …… (163)
5.5.2 算法指令 …… (164)
5.5.3 比较指令 …… (165)
5.5.4 超越函数指令 …… (166)
5.5.5 常数指令 …… (166)
5.5.6 处理器控制指令 …… (167)
5.6 仿真器和支持软件 …… (168)
5.6.1 Intel 支持软件包 …… (168)
5.6.1.1 使用 8087 仿真器 …… (169)
5.6.1.2 使用十进制转换程序库 …… (172)
5.6.1.3 使用初等函数程序库 …… (184)
5.6.2 Microsoft 仿真器 …… (186)

5.7 计算机算法原理 …… (187)
5.7.1 内存中数据的排列 …… (187)
5.7.2 向量和矩阵操作 …… (189)
5.7.3 线性系统的处理 …… (189)
5.7.4 迭代法和 Heuristic 法 …… (191)

第六章 数字字母显示 …… (193)
6.0 引言
6.1 IBM 视频硬件 …… (194)
6.1.1 来自 BIOS 的视频数据 …… (195)
6.1.2 IBM / PC 和 PS / 2 视频系统 …… (204)
6.1.2.1 单色显示适配器(MDA) …… (204)
6.1.2.2 彩色图形适配器(CGA) …… (206)
6.1.2.3 增强图形适配器(EGA) …… (207)
6.1.2.4 PCjr 显示器硬件 …… (208)
6.1.2.5 PS / 2 30 型 MCGA 系统 …… (209)
6.1.2.6 PS / 2 50 / 60 / 70 / 80 型 VGA 系统 …… (209)
6.1.2.7 IBM 高分辨率板 …… (210)
6.1.3 光标操作 …… (210)
6.1.3.1 隐藏光标 …… (210)
6.1.3.2 修改光标 …… (210)
6.1.3.3 控制光标 …… (212)
6.1.4 对 6845 CRT 控制器编程 …… (218)
6.1.4.1 开关 CRT …… (218)
6.1.4.2 字符同步输出 …… (220)
6.2 字母数字视频显示编程 …… (221)
6.2.1 可移植性讨论 …… (221)
6.2.2 DOS 字符输出视频服务 …… (221)
6.2.3 BIOS 字符输出视频服务 …… (225)
6.2.4 获得直接存取视频缓冲区 …… (234)
6.2.5 彩色 / 单色兼容 …… (243)
6.2.6 字符图形 …… (246)
6.2.6.1 画框技巧 …… (246)
6.2.6.2 属性操作 …… (246)
6.2.7 BLOCKDEMO 程序 …… (246)

第七章 通讯 …… (258)
7.1 串行和并行 …… (258)
7.1.1 对并行口编程 …… (259)

7.1.2 通过 DOS 访问并行口 …… (259)
7.1.3 通过 BIOS 访问并行口 …… (259)
7.1.3.1 BIOS 服务号 0, INT 17H …… (260)
7.1.3.2 BIOS 服务号 1, INT 17H …… (260)
7.1.3.3 BIOS 服务号 2, INT 17H …… (261)
7.1.4 并行口的直接控制 …… (262)
7.1.4.1 接口地址 …… (262)
7.1.4.2 输出数据功能 …… (263)
7.1.4.3 输入状态功能 …… (263)
7.1.4.4 输出控制功能 …… (265)
7.1.4.5 打印机中断 …… (266)
7.1.5 打印机应用的编程技术 …… (267)
7.1.5.1 基本的打印机控制功能 …… (267)
7.1.5.2 行索引 …… (268)
7.1.5.3 文本对中 …… (269)
7.1.5.4 格式长度 …… (269)
7.1.6 打印机控制样本例程 …… (269)
7.2 串行接口 …… (272)
7.2.1 通讯协议 …… (273)
7.2.2 对串行接口编程 …… (273)
7.2.2.1 用 DOS 对串行口编程 …… (274)
7.2.3 通过 BIOS 访问串行接口 …… (274)
7.2.3.1 BIOS 服务号 0, INT 14H …… (274)
7.2.3.2 BIOS 服务号 1, INT 14H …… (277)
7.2.3.3 BIOS 服务号 2, INT 14H …… (278)
7.2.3.4 BIOS 服务号 3, INT 14H …… (278)
7.2.4 串行口的直接控制 …… (278)
7.2.4.1 发送器保持寄存器 THR(xF8H) …… (280)
7.2.4.2 接收器数据寄存器 RDR(xF8H) …… (281)
7.2.4.3 波特率分部(LSD)(xF8H)和(MSB)(xF9H) …… (281)
7.2.4.4 中断启动寄存器 IER(xF9H) …… (282)
7.2.4.5 中断识别寄存器 IIR(xFAH) …… (283)
7.2.4.6 线路控制寄存器 LCK(xFBH) …… (283)
7.2.4.7 调制解调器控制寄存器 MCR(xFCH) …… (284)
7.2.4.8 线路状态寄存器 LSR(xFDH) …… (284)
7.2.4.9 调制解调器状态寄存器 MSR(xFEH) …… (286)
7.3 通讯中的硬件中断 …… (286)
7.3.1 中断服务例程 …… (288)
7.3.2 环形缓冲器 …… (289)

7.3.3 开发中断驱动通讯程序 …… (289)
7.3.4 TERMINAL 程序 …… (290)

第八章 颜色和图形 …… (304)
8.0 引言
8.1 EGA/VGA 结构 …… (305)
8.1.1 APA 高分辨率模式 …… (305)
8.1.2 视频内存结构 …… (305)
8.1.3 颜色和调色板 …… (307)
8.1.4 边界颜色和全局寄存器 …… (308)
8.1.5 字母数字模式中的颜色 …… (309)
8.2 对 EGA-VGA 图形编程 …… (319)
8.2.1 扩展 BIOS 视频操作 …… (320)
8.2.1.1 BIOS 扩展服务号 16,INT 10H
(AH=16:设置调色板寄存器) …… (320)
8.2.1.2 BIOS 扩展服务号 17, INT 10H (AH=17:字符生成器) …… (322)
8.2.1.3 BIOS 扩展服务号 18, INT 10H
(AH=18:EGA 信息和多种服务) …… (324)
8.2.1.4 BIOS 扩展服务号 19, INT 10H (AH=19:写字串) …… (325)
8.2.1.5 BIOS 扩展服务号 26, INT 10H
(AH=26:监控器和适配器信息) …… (325)
8.2.2 图形读写模式 …… (326)
8.2.3 视图子系统可编程硬件 …… (326)
8.2.3.1 通用图形寄存器 …… (326)
8.2.3.2 定序寄存器 …… (327)
8.2.3.3 CRT 控制寄存器 …… (327)
8.2.3.4 图形控制寄存器 …… (327)
8.2.3.5 属性控制寄存器 …… (328)
8.2.4 开发图形 …… (328)
8.2.5 字节边界例程 …… (328)
8.2.6 字节和象素边界例程 …… (330)
8.2.7 缓冲区地址计算 …… (332)
8.2.8 画直线 …… (335)
8.2.8.1 水平和竖直线 …… (335)
8.2.8.2 斜线 …… (336)
8.2.9 用 8087 画图 …… (336)
8.2.9.1 圆逼近 …… (337)
8.2.9.2 其它画图功能 …… (337)
8.2.9.3 程序 GRAGDEMO …… (337)

8.2.10 矩形填充例程 …… (360)
8.2.11 图形块显示例程 …… (360)
8.2.12 USFLAG 程序 …… (360)
8.2.13 屏幕绘图算法 …… (374)
8.3 文本与图形 …… (375)
8.3.1 图形模式中的文本字符显示 …… (375)
8.3.2 作为图形工具的文本 …… (376)
8.3.2.1 可选择的字符根 …… (376)
8.3.2.2 同时使用多种字符根 …… (376)
8.3.3 图形模式中的常用字符 …… (376)
8.3.3.1 软件字符生成器 …… (376)
8.3.3.2 使用 BIOS 字符生成器 …… (377)
8.4 动作与图象变换 …… (377)
8.4.1 图象显示 …… (377)
8.4.2 异或(XOR)象素 …… (378)
8.4.3 BILBOARD 程序 …… (378)

第九章 数据输入和存贮 …… (384)
9.1 键盘 …… (384)
9.1.1 键盘硬件及操作 …… (384)
9.1.1.1 LOCK 键 …… (385)
9.1.1.2 SYS REQ 键 …… (385)
9.1.1.3 HOT 键 …… (385)
9.1.2 BIOS 中的键盘数据 …… (386)
9.1.2.1 键盘缓冲区 …… (386)
9.1.2.2 第一个键盘状态字节 …… (387)
9.1.2.3 第二个键盘状态字节 …… (388)
9.1.3 键盘中断处理程序 …… (388)
9.1.4 替换键盘中断处理程序 …… (389)
9.1.5 截取键盘中断 …… (392)
9.1.5.1 扫描码表的问题 …… (392)
9.1.5.2 抑制 HOT 键 …… (392)
9.1.5.3 生成 HOT 键 …… (397)
9.1.5.4 BIOS 键盘截取 …… (398)
9.1.6 键盘不反弹和响应速率 …… (401)
9.1.6.1 改变延迟和响应速率 …… (401)
9.1.6.2 TYPMATIC 程序 …… (401)
9.1.7 对锁模式键编程 (Num、Caps、Scroll) …… (407)
9.1.7.1 从软件中开关 LOCK 键 …… (407)

9.1.7.2 TOGGLE 程序 …… (407)
9.1.8 BREAK 键 …… (409)
9.1.8.1 使用 Break 键处理程序 …… (409)
9.1.8.2 DOS Ctrl—C 处理程序 …… (410)
9.2 对键盘输入编程 …… (411)
9.2.1 BIOS 键盘服务 …… (411)
9.2.1.1 BIOS 服务号 0，INT 16H (AH＝1:读键盘状态) …… (411)
9.2.1.2 BIOS 服务号 2，INT 16H (AH＝2: 取键盘状态字节) …… (412)
9.2.1.3 BIOS 服务号 3，INT 16H
(AH＝3:设置响应速率和延迟时间) …… (412)
9.2.2 DOS 键盘服务 …… (412)
9.2.2.1 DOS 功能号 10，INT 21H (缓冲区的键盘输入) …… (413)
9.3 磁盘存贮导论 …… (413)
9.3.1 DOS 下的盘存贮机置 …… (414)
9.3.2 磁盘存贮格式 …… (414)
9.3.2.1 BIOS 下的磁盘格式 …… (415)
9.3.2.2 DOS 下的磁盘格式 …… (416)
9.3.3 DOS 磁盘操作的逻辑结构 …… (416)
9.3.3.1 磁盘转换区(DAT) …… (417)
9.3.3.2 文件控制块 (FCB) …… (417)
9.3.3.3 程序段前缀 (PSP) …… (418)
9.4 对磁盘操作编程 …… (419)
9.4.1.1 BIOS 服务号 0，INT 13H (AH＝0:重置软盘系统) …… (419)
9.4.1.2 BIOS 服务号 1，INT 13H
(AH＝1:读上次软盘操作的状态) …… (420)
9.4.1.3 BIOS 服务号 2，INT 13H (AH＝2:读软盘扇区到内存) … (420)
9.4.1.4 BOIS 服务号 3，INT 13H
(AH＝3:将内存写入软盘扇区) …… (420)
9.4.1.5 BIOS 服务号 4，INT 13H (AH＝4:检查软盘扇区) …… (421)
9.4.1.6 BIOS 服务号 5，INT 13H (AH＝5:格式化软盘磁道) …… (421)
9.4.2 DOS 服务综述 …… (421)
9.4.2.1 DOS 服务号 14，INT 21H
(AH＝14: 设置缺省磁盘驱动器) …… (422)
9.4.2.2 DOS 服务号 23，INT 21H (AH＝23: 改变文件名) …… (422)
9.4.2.3 DOS 服务号 25，INT 21H (AH＝25: 获得缺省的驱动器) … (423)
9.4.2.4 DOS 服务号 26, INT 21H (AH＝26: 获得缺省的驱动器) … (423)
9.4.2.5 DOS 服务号 41，INT 21H (AH＝41: 从串创建 FCB) …… (423)
9.4.3 使用 FCB 的 DOS 磁盘服务 …… (424)
9.4.3.1 DOS 服务号 15，INT 21H (AH＝15: 使用 FCB 打开文件) … (424)

9.4.3.2 DOS 服务号 16，INT 21H (AH＝16: 使用 FCB 关闭文件) … (424)
9.4.3.3 DOS 服务号 20，INT 21H
(AH＝20: 使用 FCB 顺序读文件) … (424)
9.4.3.4 DOS 服务号 21，INT 21H
(AH＝21: 使用 FCB 顺序写文件) … (424)
9.4.3.5 DOS 服务号 22，INT 21H
(AH＝22: 使用 FCB 创建文件) … (425)
9.4.4 使用文件句柄的 DOS 磁盘服务 … (425)
9.4.4.1 DOS 服务号 60，INT 21H
(AH＝60: 创建文件并获得句柄) … (425)
9.4.4.2 DOS 服务号 61，INT 21H
(AH＝61 :打开文件并获得句柄) … (426)
9.4.4.3 DOS 服务号 62，INT 21H
(AH＝62: 使用句柄关闭文件) … (427)
9.4.4.4 DOS 服务号 63，INT 21H
(AH＝63: 使用句柄读文件) … (427)
9.4.4.5 DOS 服务号 64, INT 21H (使用句柄写文件) … (427)
9.4.4.6 DOS 服务号 78, INT 21H
(AH＝78: 搜寻第一个匹配的 ASCIIZ 串) … (427)
9.4.4.7 DOS 服务号 79, INT 21H
(AH＝21:搜寻下一个匹配的文件名) … (428)
9.4.5 FILECOM 程序 … (428)

第一部分　　编程工具和编程环境

第一章　编程工具

1.0　引言

编程比专门训练的专业更具技巧性，因为程序员设计和制造智能产品．就象其它技能一样，编程需要工具．对于汇编语言程序员来说这些工具可以划分为三类:

1.设备: 计算机和硬件。

2.参考资料: 说明书、手册和期刊杂志。

3.软件: 编辑器，汇编器，连接器，调试程序，库以及其它辅助程序。

1.1　程序开发硬件

设备和硬件的选择通常只是有关个人偏爱、偶然需要或者财政限制的问题，但是有另外一些因素有时应加以考虑．IBM 和其它的设备制造厂商努力在它们的产品上保持软件的向上兼容性，这意味着在最初的 IBM PC 上开发的程序可以毫不困难地在 IBM PC XT、PCjr、PC AT、便携式 Portable PC 和 PS／2 系列的计算机上运行．为了保证向下兼容性，制造厂商就不得不放弃产品的改进和技术的进步，这使一些汇编语言程序员在老的硬件上开发他们的程序时感到比较安全。

独立于计算机且用于程序开发中的专业软件将在各个硬件上测试以承认其兼容性，这对于那些分布公用的程序尤为重要．测试应严格进行，因为软件评价和测试是一个科学的学科．如果测试表明在不同的硬件上程序的运算不同，如果可能的话它就应被修正，或者最起码偏差应在程序说明书中清楚地用文件记录下来。

一个经常被汇编语言程序员使用的小片硬件是电子计算机．许多新的常规计算机现在有二进制和十六进制的换算．其它型号是为程序员特别设计的，象 Texas Instruments Programmer 和 HP Computer Scicutist model 16C．后者执行旋转、平移和逻辑运算以及标准的数字系统转换．最近，Casto 采用了一种廉价的程序员计算机，命名为 Computer Math Calc Model CM−100，适合于大部分实际需要。

1.2　参考资料

很遗憾，并没有一本出版的说明书或手册搜集了为 IBM 微型计算机编程所需的全部资料。对于许多硬设部件的技术数据散布在各种文献中，这使得程序员很大部分的时间用于搜集和组织这些资料。

有用的编程资料可以在教科书和商业书籍、杂志和期刊、制造厂商的说明小册子、软件说明书和技术参考手册中找到。列出所有的已出版的与 IBM 微型计算机编程有关或关于单独的硬设部件的有价值的书籍要花很长时间．这里不作阐述。关于可编程硬设部件资料的一个特别来源将提及:《IBM 技术参考手册》．这本书由 IBM 公司为 PC 和 PS／2 系列的各种配置而发行，包括了程序员无法估价的宝贵资料。

1.3 软件工具

装备有计算机、可能还有打印机、并且手头有说明书、文章和手册，程序员就几乎准备好了编程（除了软件）。选择并且使用目前已有的软件开发工具并非简单之事．因为没有那一个系统包括了所有需要的特性。这些性能包括如下：

1. 一个有效的文本编辑器，与用到的汇编器相兼容。
2. 一个汇编器。
3. 一个与由汇编器生成的目标文件或库中的目标文件相兼容的连接器。
4. 一个专业例程库。
5. 辅助程序，如调试程序和库管理程序。

标记选择合适系统时所遇到的基本问题的关键词是有效（Efficient）和兼容（Compatible）。下面各节讨论一些可用的选项，涉及的主要系统是 Microsoft 和 Intel 公司的。

1.3.1 文本编辑器

许多年内，Microsoft 公司出售的开发系统不包括文本编辑器，Intel 出售的开发系统也不包括文本编辑器；编译 Basic、Pasical、FORTRAN、C、PL／I 和其它的高级语言程序出售时也没有文本编辑器。它们的手册中偶然（好象这个问题并不重要一样）提到某些形式的编辑器为使用系统所需。只是最近 Microsoft 改正了它们的态度，在它们的语言产品中包含了编辑器。

程序员也可以用字处理程序作为语言编辑器，但并非所有的字处理程序都适合于这一目的。准备用作编辑器的程序至少应该包括下面的一些典型特点：

1. 在RAM和磁盘中优化的代码空间，必须为程序文件留下尽可能多的存储器和磁盘存储。
2. 至少 60K 的文本存储器存储量。
3. 逐行地和逐页地以及高速文本文件显示的双向显示屏翻卷。访问存储器文件的任何一页不应超过 0.5 秒。
4. 单一的键命令，应该包括文件搜索、文本编辑、存储器状态显示和直接的打印机访问。
5. 与 Microsoft 公司的 MASM 和 Intel 公司的 ASM86 汇编程序的兼容性。
6. 存储部分驻留文件和把磁盘文件插入到驻留文件的磁盘文件操作，这些操作允许生成和使用以磁盘为基础的过程和例程库。

上面并没有包括许多字处理程序所需的特性，象复杂的边界控制、文本的标题和注脚页号、图形字符显示以及复杂的搜索和替换功能。编程语言编辑器一般可以不要这些选项而执行。

语言编辑器所需的效率实际上取决于用汇编语言编码的程序。另外的需求（不易证明正确）是程序必须高度可靠。缺少编程经验比起丢失几个小时的编码工作来更易受挫（由于编辑程序的误操作（glitch））。

1.3.2 汇编器

任何汇编语言开发系统的基本程序都是汇编器，它的功能是从存储器或从编辑程序生成的磁盘文件中读入源程序行并且生成适当的机器码．大部分的汇编器也生成特殊的重定位符号（被连接程序使用）．如果汇编器具有操作带有特殊控制和命令的源语句组的能力，它就被称为有宏能力．在本书中所讨论的汇编器都有这种选项，所以被称为宏汇编器．

Microsoft MASM 汇编器和 Intel ASM86 是相似的程序，两者都处理磁盘基础上的源程序并且生成磁盘基础上的重定位目标文件．为了生成一个在 DOS 下运行的程序文件，由汇编器生成的文件必须进一步由一个叫连接器的程序来处理．虽然 Microsoft 和 Intel 汇编器都识别几乎一致的命令集合和源程序行格式，但是在程序控制和汇编运算器中仍有很大差别．因此，最初为 MASM 编写的源文本要被 ASM86 读入时，一般必须被编辑，相反也一样．

1. 3. 3　连接器

连接器程序处理由汇编器或编译器生成的一个或几个目标模块并且将这些模块综合到一个文件中．在连接器生成的文件中，源程序中的某些重定位和外部地址被判定，而其它的地址则保存直到加载时间．

在 Microsoft 系统中，由连接器生成的文件是一个在 DOS 下可运行的程序；在 Intel 系统中，连接器生成的文件必须在它作为 DOS 程序运行之前由名为 UDI2DOS 的一个设备处理．

1. 3. 4　库和库设备

目标码库，无论是商业购买的或个人开发的，对于汇编语言程序员都是非常重要的软件工具．Microsoft 和 Intel 系统都能处理库模块并且获得处理设备．

库是一个有用的例程和过程的集合，这些例程和过程可以有选择地并入源文件或文件中．通常库中的所有例程希望找到它们的在栈中按一定顺序放置的或者存储在数据段预定义的外部标号下的操作数据．通过使用商业化的例程库或使用编辑他或她个人的例程而成的库，汇编语言程序员避免了重新创造软件的车轮（指一些基本功能（常用）），不幸的是，并没有很多商业化库可用，并且那些库也并不总是正确地执行．有眼力的买主在购买例程库前应该确认例程库已经适当的测试并且已形成文档．

在开发个人例程库中，最好采用约定俗成的名字作为外部或公共符号的系统，这种方法可使用户在程序码中容易地识别出任一指向库模块的标记．用于这些标记和数据地址的名字系统由库的设计者负责．下面的观点对于生成库模块可能是有益的：

1. 库模块码应定义为 FAR，而对模块的访问应通过 CALL 命名．

```
;.....library module.....
;
Public   TEST__LM
;
TEST__LM      RPOC      FAR
```