

## 图书在版编目(CIP)数据

文秘手册 / 金 炜 主编

—企业管理出版社, 1996.3 ISBN 7-80001-669-2

I . 文… II . 金… III . 行政—文秘手册

## 文 秘 手 册

---

金 炜 主编

出版发行: 企业管理出版社出版

开本: 787mm × 1092mm 1 / 16 印张: 132

字数: 2108千字

印刷: 1996年3月第1版

版次: 1996年3月第1次印刷

书号: ISBN 7-80001-669-2

---

定价: 76.00元 (图书共19册)

# 目 录

## 文秘微机操作指南之二

第一章 .....	猿
第一节 配置 混杂 .....	猿
第二节 重定向、管道和过滤程序 .....	猿
第三节 提高系统的运行速度 .....	猿
第四节 合理使用内存空间 .....	猿
第五节 两台计算机的简单互连 .....	猿
第六节 磁盘倍增技术 .....	猿
第二章 微机 混杂常用命令表 .....	猿



# 文秘微机操作 指南之二



# 第一章 微机 操作系统使用技巧

## 第一节 配置 操作系统

### 一、操作系统引导与 操作系统维护的功能

每次开启计算机,计算机都要借助于 硬盘(只读存储器),在经过加电自检(POST)—— 硬盘的初始化之后,通过一个被称为引导装入程序的指令代码将磁盘上的 操作系统引导入内存,从而开始启动 操作系统。在 操作系统的启动过程中,主要进行下面三个方面的工作。首先通过查找,将在根目录之下的 引导程序、配置程序和 操作系统配置文件一起装入内存;第二,读取 操作系统文件,并根据该文件中的命令对计算机的硬环境进行软配置;第三,查找 操作系统文件,如果找到该文件,则执行该文件中的命令,否则完成 操作的启动,出现 操作提示符。

操作系统和 操作系统是两个一般来讲存在于磁盘根目录之下的文件。在这两个文件中含有一些特殊的 操作命令,用户可以使用这些命令,根据计算机系统的实际情况进行适当的 操作环境设置,为应用软件创造一个良好的运行环境。合理的 操作系统设置有利于优化内存管理、提高软件的运行速度、充分利用系统的硬件资源。操作系统其实是一个特殊的批处理程序,它一方面配合 操作系统的系统设置;一方面由于它具有‘自动批处理’的功能,每次启动 操作后都自动运行该文件中的命令,故此用户可以将某些命令加入 操作系统文件之中,使系统在开机之后,不用键入任何命令,即可自动达到所要工作的环境。例如某个名为 用户的用户,每次都只应和 的工作,则在 操作系统文件中键入以下一段命令后,每次系统启动,都会自动进入 的环境。

```
操作系统  
操作系统:操作系统:操作系统:操作系统  
阅:  
操作系统  
操作系统  
操作系统
```

操作系统	操作系统
操作系统	操作系统
操作系统	操作系统

操作系统和 操作系统文件都可以通过文本编辑软件,如 提供的 或

来建立或修改。有些用户习惯用 `cd` 命令来建立这两个文件,例如在根目录下键入:

```
cd /usr && mkdir /usr/share
```

`cd` 命令将自动将光标移至下一行,用户即可键入 `cd /usr/share` 文件中想包括的命令,当输入完命令之后,同时按下 `Ctrl+D` 或在或仅按下 `Ctrl+Z` 键即可生成 `cd /usr/share` 文件。下面是利用 `cd` 命令来建立 `cd /usr/share` 和 `cd /usr/share` 文件的例子,在例子后面附有简单的命令解释,具体的命令语法在后面的几节中介绍。

例: 建立 `cd /usr/share` 文件

```
cd /usr && mkdir /usr/share
cd /usr/share && cd /usr/share
cd /usr/share && cd /usr/share
cd /usr/share && cd /usr/share
cd /usr/share && cd /usr/share
```

在上面的例子中,第一条命令将设备驱动程序 `cd /usr/share` 加载到常规内存,使用户可以使用扩充内存,第二条命令和第三条命令设置磁盘缓冲区数和同时可打开的文件数分别为 `cd /usr/share`,第四条命令将设备驱动程序 `cd /usr/share` 加载到常规内存, `cd /usr/share` 程序用扩充内存来仿真扩展内存的使用。最后一行 `cd /usr/share` 表示同时按下 `Ctrl+D` 和 `Ctrl+Z` 键,结束 `cd /usr/share` 文件录入。

例: 建立一个 `cd /usr/share` 文件

```
cd /usr && mkdir /usr/share
cd /usr/share && cd /usr/share
cd /usr/share && cd /usr/share
cd /usr/share && cd /usr/share
cd /usr/share && cd /usr/share
```

其中,第一条命令设置路径,第二条命令设置标准的 `cd /usr/share` 提示符 `cd /usr/share` 并且将文本前景设为白色,背景设为蓝色,第三条命令将内存驻留程序 `cd /usr/share` 加载入内存,第四条命令将 `cd /usr/share` 内存驻留程序加载入内存,例子中的 `cd /usr/share` 表示同时按下 `Ctrl+D` 和 `Ctrl+Z` 键,结束文件输入。

虽然用户可以任意修改 `cd /usr/share` 和 `cd /usr/share` 文件中的内容,但在实际运用中,用户应该根据具体情况进行合理设置,才能够提高系统的运行效率。有些软件,例如 `cd /usr/share` 或 `cd /usr/share` 在安装之时,能够根据硬件环境对 `cd /usr/share` 和 `cd /usr/share` 进行自动修改,在此时,用户就不必重新为这些软件设置 `cd /usr/share` 环境。一般而言,如果计算机系统的软硬件配置没有较大变化,则不该过多地去改动 `cd /usr/share` 的设置命令。有些用户的计算机系统内,常常会安有多个操作系统、多个汉字系统、多个网络系统的情况,为了使用这些不同

的系统需要有不同的 配置文件, 配置文件因以上版本提供了一组名为 配置文件, 能够更为灵活地配置系统环境。

最后应当注意, 配置文件只有在启动 系统时才被执行。因此在修改了 配置文件之后, 要想使这些修改的内容起作用, 需要重新启动计算机系统。要利用修改后的 配置文件一般也须重新启动。但在某些情况下, 例如在单纯修改了 配置文件, 而 配置文件中的命令与 配置文件的命令没有任何关系的情况下, 为了激活 配置文件中的命令, 只须在 系统提示符下直接运行 即可。此时可将 看成是一个纯粹的批处理文件。

## 二、系统的主要命令

### 系统的命令划分

与 系统相关的命令可划分为以下几种情况：

(只有用于 系统中才有效的命令)

这些命令只能用于 系统文件之中, 如果在其它情况下(例如直接在 系统提示符下) 执行这些命令则不起作用。这类命令有：

命令	功能
disk	为磁盘存取设置缓冲区数目
date	设置非美国的时间、日期、货币等符号标记
device	在常规内存中安装设备驱动程序
devicehigh	在上位内存( 高端)中安装设备驱动程序
dos	将 系统的内存驻留部分装入 ( 高端内存), 并且指定 系统是否控制上位内存块( 高端)
drive	改变 系统给块设备(如软盘)分配的缺省参数
files	定义同时可打开的文件控制块的数目
filesz	指定可同时打开的文件个数
load	在 系统中读入并执行一个内存驻留程序( 驻留)
lastdrive	指定 系统可存取的驱动器的最后一个字母
loadkey	指定系统启动时 键键设为 还是
path	如果将 系统文件从 盘的根目录中移去, 用本命令指出它当前所在的驱动器和目录

命 令	功 能
控制杂	控制用于进行内部硬件中断的特殊处理的 的数目
控制杂	当在 中,设置 时,则在系统启动时,按下 键,就不会有跳过 和 而不执行配置命令的情况。同样,此时按下 键,也不会出现逐条提示执行 和 文件中命令的情况

( 能在 中运行,同时还可以在 提示符下执行的命令  
这组命令主要有以下三条:

命 令	功 能
月	设置 是普通式中断还是扩充式中断,当 设为 时,将系统的 设为扩充式中断。普通式中断只可以中断键盘的输入,屏幕与打印机的输出,而扩充式中断除了普通式中断功能外,亦可以中断磁盘的读写操作。如果用户没有使用 命令,则缺省条件是 ,即普通式中断
磁	注释,系统将 后的字符串都视为注释
控	设置,可显示、设置或删除 系统的环境变量

( 只用于 中,而且只能用于进行多重配置( )的命令

这组命令在 以上版本中使用,它们可以在 文件内定义若干不同系统的配置,形成不同的用户配置块,在执行 文件时,用户可根据实际情况,选择理想的配置块。它们只能用于 文件之中,并且只能用于进行定义多种系统配置时使用。这些命令有:

命 令	功 能
隔	包括另一配置块中的所有命令
项	设置菜单项的正文和背景颜色,它只能作用于 中的 块
项	默认菜单项。在 块中,可以用此命令指定某配置名称及 秒,当出现启动菜单后,如果用户在 秒内还未作出选择,则系统默认使用 指定的配置块。

命 令	功 能
!@#\$%^&*~	该命令指定在 !@#\$%^&*~ 块中每一个配置块的名称 ,以备用户选择
~!@#\$%^&*~	在 !@#\$%^&*~ 块中定义二级菜单的名称

与 !@#\$%^&\*~ 相关的两个字符和两个功能键

在 !@#\$%^&\*~ 文件中 ,可以使用两个特定的符号。

“ ;”号。分号“ ;”在 !@#\$%^&\*~ 文件中相当于 !@#\$%^&\*~ 命令 ,表示注释 ,加在某个字符串之前 ,表示这个字符串及这行后面的字符串都是注释。在编辑 !@#\$%^&\*~ 文件时用户也可将此字符加在某些想要取消的命令前面 ,使得系统在启动时 ,将带有“ ;”的命令视为注释行而不加执行。“ ;”号与 !@#\$%^&\*~ 的区别在于 !@#\$%^&\*~ 可用于批处理或 !@#\$%^&\*~ 文件之中 ,而“ ;”只能用于 !@#\$%^&\*~ 之中。

“ ?”号。将此字符紧接在 !@#\$%^&\*~ 文件中的某个命令之后 “ 越”符之前 ,则可以使计算机在执行该命令时 ,进行选择提示。例如在有时需要将 !@#\$%^&\*~ 置于高端内存( 内存) ,而有时则需要驻留在常规内存的情况下 ,可将 !@#\$%^&\*~ 命令由

```
!@#$%^&*~ 越!@#$%^&*~
```

改为

```
!@#$%^&*~ ? 越!@#$%^&*~
```

则在系统启动后 ,都会有

```
!@#$%^&*~ 越!@#$%^&*~ 再,晕]? 再
```

的提示。用户可根据实际情况选定 再或 晕

在启动系统之时 ,有两个功能键对于控制 !@#\$%^&\*~ 文件中命令的执行是很有用的。这两个功能键在系统启动时也对 !@#\$%^&\*~ 文件中的命令起控制作用。

云缘键。云缘键可以使系统在启动时 ,不去执行 !@#\$%^&\*~ 和 !@#\$%^&\*~ 文件中的命令。在系统启动时 ,当屏幕上出现“ !@#\$%^&\*~ .”提示的时候 ,如果用户按下 云缘键 ,则系统不去执行 !@#\$%^&\*~ 和 !@#\$%^&\*~ 中的任何命令 ,而直接进入 !@#\$%^&\*~ 提示符号状态。未执行 !@#\$%^&\*~ 和 !@#\$%^&\*~ 文件时 ,系统自动设置基本配置。

云愿键。云愿键可以使系统在启动时 ,逐条提示用户选择是否执行 !@#\$%^&\*~ 和 !@#\$%^&\*~ 文件中的每一条命令。它和 云缘一样 ,只有在系统启动时 ,当屏幕上出现“ !@#\$%^&\*~ .”提示时 ,按下 云愿键才有效。

下面是这两个功能键的例子。

例 1: 在启动系统过程中 ,当屏幕上出现“ !@#\$%^&\*~ .”时 ,按下 云缘键 ,则屏幕上会出现以下情况。



设备驱动程序	功 能
硬盘驱动程序	使用 缓存和扩充内存的管理程序,支持 缓存的存取工作,能将扩充内存模拟为扩展内存使用
内存管理程序	用于管理扩充内存的设备驱动程序,它能让符合 规范、需要使用扩充内存的程序访问 扩充内存
并行磁盘系统	同于两台计算机之间的相互连接
休眠驱动程序	将 装入内存,则可以使计算机在程序和设备处于空闭状态时,节省用电,它只用于便携式计算机
硬盘虚拟盘	在 中设置虚拟盘
兼容驱动程序	由旧版本 开发的程序欲执行于新版本上时,需先安装 驱动程序
内存驻留程序	利用剩余内存来建立磁盘高速缓冲内存,在 版本中,可以在 文件中,由 命令装入内存,而在 以上版本中,是一个内存驻留程序( ),可以在 里作为一个命令直接运行

的命令格式为

设备驱动程序名

“越”号之后的设备驱动程序名包括驱动器、路径及其文件全称,在这之后,还可以包括一些配置参数及开关。例如下面两条命令:

例: 管理扩充内存

例: 将 装入内存,并且使这种 只提供上位内存

第一条命令会将 装入内存,管理扩充内存。

第二条命令将 装入内存,并且使这种 只提供上位内存(缓存)的管理,但不提供仿真扩展内存方式使用扩充内存。在 中,如果同时使用这两条命令,必须 在前, 在后,顺序不能颠倒。因为 负责扩充内存的管理,而在一般情况下, 运行时可能会用到扩充内存,如果没有 对扩充内存管理,则 运行会出错。

下面的例子利用 命令,在扩充内存中安装一个虚拟磁盘。虚拟磁盘又被人称为 磁盘, 会给它一个驱动器字母,用户可以像对待真正的磁盘驱动器一样,对其进行读、写等操作。由于这种虚拟磁盘的读写是在内存中操作,因而利用虚拟的速度比利用硬盘的速度快得多。

例 在有 内存的 计算机中,建立一个 的虚拟磁盘。只要在 那

在 `sysconfig` 文件中,作如下设置。

```

悦: 攒跃栽月云惰葬云颞颞杂
   译决裁云悦: 攒诸劣赞云颞颞杂
   译决裁云悦: 攒诸劣赞酒云颞颞杂云
   译决裁云悦: 攒诸劣赞云颞颞杂云颞颞杂

```

则重启系统后,屏幕上会显示

```

西云颞颞杂云颞颞杂 阅颞颞杂颞颞杂颞颞杂颞颞杂:
   阅颞颞杂
   杂颞颞杂云
   杂颞颞杂颞颞杂颞颞杂颞颞杂
   粤颞颞杂颞颞杂颞颞杂颞颞杂
   阅颞颞杂颞颞杂颞颞杂颞颞杂

```

虚拟磁盘的盘符是最后一个外存逻辑磁盘代号之后的字母。本例中,外存逻辑磁盘有粤、月、悦、阅,故虚拟盘为颞

在这里,还得对 `sysconfig` 文件中的第三条命令作一简单介绍。`颞颞杂` 是建立虚拟磁盘的设备驱动程序,其命令格式如下:

`颞颞杂` 盘符 I 路径 颞颞杂 颞颞杂 颞颞杂 盘大小 I 扇区大小 I 根目录大小 I 颞 [颞]

在例 `颞颞杂` 中,指出 `颞颞杂` 磁盘(虚拟磁盘)大小为 `颞颞杂`,其它参数缺省,而应用了 `颞` 开关,它表示使用扩充内存建立 `颞颞杂` 盘。另外一个可供选择的开关 `颞`,表示使用扩展内存建立 `颞颞杂`。

有些低版本的工具软件,例如 `颞颞杂` 在高版本的 `颞杂` 下运行时,系统会出现“`颞颞杂`”的提示,而这些软件也不能进行工作。下面的操作将使这些工具软件能在 `颞杂` 或 `颞杂` 以上的版本内运行。

(员在 `sysconfig` 中加入下面一条命令:

```
译决裁云悦: 攒诸劣赞栽栽颞颞杂
```

加入该命令,可使下次启动系统时,能够将版本表装入内存。

(圆运行 `颞颞杂` 应用程序。例如为了使用 `颞颞杂` 版本的 `颞颞杂`,可在 `颞杂` 揭示符下键入

```
悦: 攒跃栽栽颞颞杂颞颞杂颞颞杂
```

它将 `颞颞杂` 文件名)和 `颞颞杂` (颞杂版本号)加入到版本表内。

(猿重新启动计算机,此时系统会把第 `颞` 步内修改了的版本表装入内存。

(源即可运行 `颞颞杂`。这主要因为高版本的 `颞杂` 系统能够从版本表中获知 `颞颞杂` 的版本号,因而可以在高版本的 `颞杂` 中执行低版本的程序。

```
颞颞杂颞颞杂颞颞杂
```

loadmem命令与 loadmem类似,这个命令也是用于安装设备驱动程序的命令,而与 loadmem不同的是,它不将设备驱动程序加载到基本内存,而是装入上位内存块(mem)。上位内存(mem)是地址在 1024K 区域内的(ram)内存,系统硬件在使用部分(mem)之后剩余的(mem)空间,被称为上位内存块(mem)。将设备驱动程序加载到(mem)后,会腾出更多的常规内存。

在使用 loadmem和 loadmem命令时,还应当注意,loadmem和 loadmem这三个设备驱动程序只能用 loadmem命令装入常规内存,而其它设备驱动程序则既可用 loadmem装入常规内存,也可使用 loadmem将其加载到上位内存块(mem)。

将 loadmem装入 mem,可以在 config文件中加入:

```
loadmem mem 越说: 拨端杂拨端杂
```

### loadmem命令

loadmem命令可以让用户决定 loadmem的常驻内存部分装入常规内存还是装入高端内存。高端内存(mem)是 mem之外的,第一个扩充的 mem存储区。loadmem命令将 loadmem的常驻内存部分装入 mem(高端内存),则可以让出更多的基本内存供应用程序使用。loadmem的命令格式如下:

```
loadmem 越 选项, 选项 I mem, 选项 mem
```

其中,选项表示将 loadmem的常驻内存部分装入到高端内存(mem);选项表示 loadmem常驻内存部分驻留在基本内存;mem表示 loadmem命令可支持对上位内存(mem)的管理,选项 mem表示 loadmem命令不支持 mem管理。

如果在 config文件中加入了

```
loadmem 越 mem
```

系统启动后,loadmem的常驻内存部分被加载到高端内存(mem)区域,并且 loadmem支持对上位内存(mem)的管理。

### loadmem命令

loadmem命令可以将一些常驻内存程序(也叫内存驻留程序、终止并驻留程序,简称驻留)装入内存。驻留程序一般是用 loadmem或命令行装入的,而用 config文件装入可以节省几个字节的 ram,但并不是所有的驻留程序都能这样安全地装入的。一般来讲 loadmem系统所带的驻留程序都可用 loadmem装入。下面是 loadmem的命令格式:

```
loadmem 越 选项 I 选项 选项 选项 选项
```

下面是 loadmem的例子

```
loadmem 越说: 拨端杂拨端杂 选项 选项
```

### 三、在 config文件中使用 loadmem命令

loadmem和 loadmem版本在 config文件的命令中增加了一组名为 loadmem(多重配置)的命令。这组命令可以为一个 config文件书写多个配置块,使得系统在启动





专用的块名，而 `USER` 是用户为自行定义的配置块所取的块名。

在 `USER` 块中，主要完成主菜单的定义。每个菜单项由“`USER`”来定义。在例 10.1 中，定义了两个菜单项，分别名为 `USER` 和 `USER`，它们其实是后面两个用户配置块的块名，同时在这里也是两个人的人名。在 `USER` 块中，还可以用“`USER`”命令来定义子菜单名。在系统启动后，执行 `USER` 时，菜单项的名称就会显示在屏幕上供用户选择。

`USER` 块定义了系统启动时需要共同加载的公共配置命令。例如 `USER` 和 `USER` 都需要将 `USER` 设备驱动程序装入内存，则无需分别在 `USER` 和 `USER` 的配置块中说明，而将“`USER`”放入 `USER` 块中共同使用。当 `USER` 拥有 `USER` 块时，系统会先去执行 `USER` 块中的命令，然后才去执行用户所选择块中的用户配置命令。例如在例 10.1 的 `USER` 文件中，假如在启动菜单中选择了 `USER`，则系统先执行 `USER` 块中的 `USER` 命令，之后才执行 `USER` 块的两条命令。

在例 10.1 中，使用了两个 `USER` 块，第一个 `USER` 块下有 `USER` 命令；而第二个 `USER` 块在 `USER` 文件末尾，不含任何一条命令，它有什么用途呢？其实第二个 `USER` 块是为某些新的应用程序而装备的。有些应有程序在安装时，会将一些适合于该应用程序的系统配置命令添加在 `USER` 文件之尾。当 `USER` 文件使用了 `USER` 命令之后，在 `USER` 文件的底端，如果有一个 `USER` 块，则可以将这些应用程序安装时添加的配置命令加入到文件 `USER` 底部的 `USER` 块中，使系统开启时，不论选择哪种配置块，都执行添加在文件 `USER` 末端 `USER` 块中的命令。

`USER` 块和 `USER` 块是 `USER` 和 `USER` 各自为自己工作环境而分别定义的用户配置块。用户配置块的块名应当和 `USER` 命令使用的块名一一对应。在例 10.1 中，`USER` 块是为桌面排版程序而设置的，由于桌面排版程序（`USER`）需要使用扩展内存，所以有装入 `USER` 设备驱动程序的命令；`USER` 块中，由于不使用桌面排版程序，而需要网络软件，故而配置了 `USER`。

有些使用了 `USER` 命令的 `USER` 文件，需要 `USER` 对其进行配合。例如，例 10.1 中的 `USER` 即是配合例 10.1 的 `USER` 配置的文件。在执行 `USER` 的启动菜单时，如果用户选择了某个用户配置块，`USER` 就会定义一个环境变量 `USER` 并以选定的配置块名称作为它的值（即环境变量 `USER` 选定的配置块名）。因此在 `USER` 文件中，系统就可以使用 `USER` 命令来选择一组命令。

如果在例 10.1 的 `USER` 启动菜单中，用户选择了 `USER`，则会使环境变量 `USER` 在 `USER` 执行完相应的 `USER` 命令之后，系统执行 `USER` 文件中的命令，首先执行以下三条命令：

```
USER USER
USER USER
USER USER
```

在执行这三条命令之后，由于已选定 `USER`，则在执行 `USER` 时就相当于执行 `USER`，从而使流程跳到标号 `USER` 之下，执行下面三条命令：

