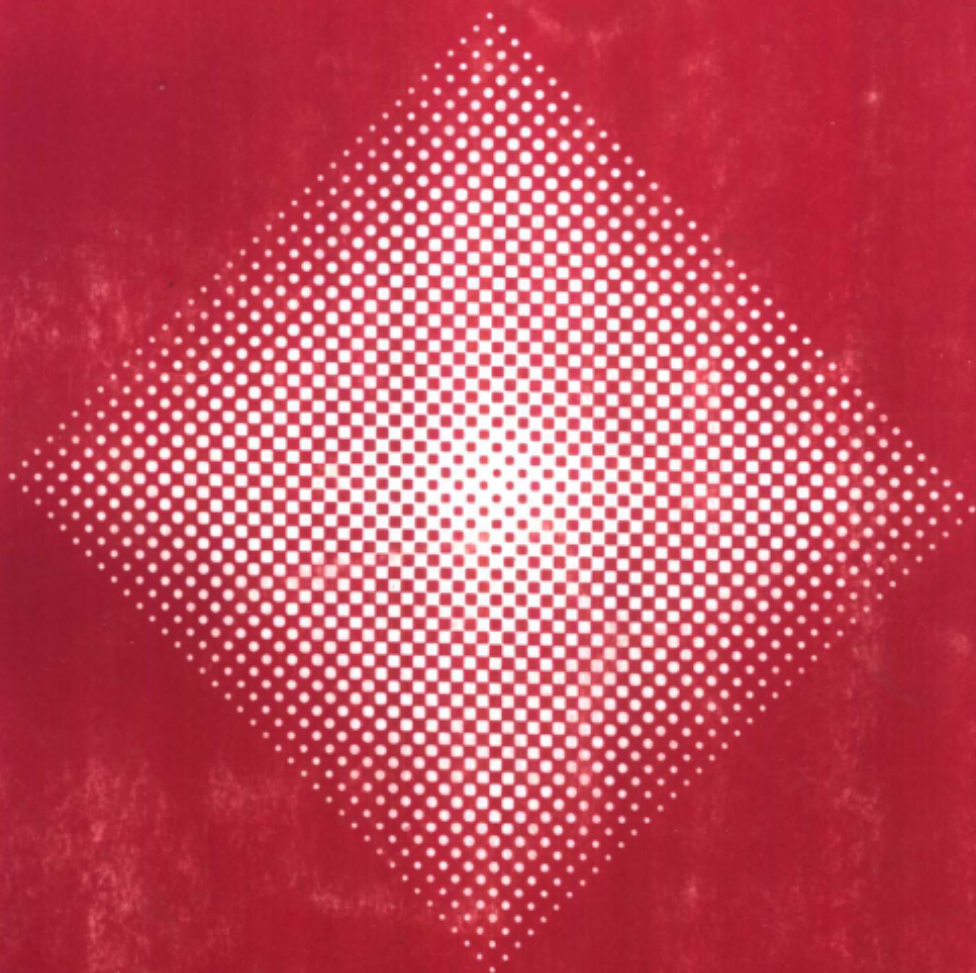


计算机操作系统实验教程

——MINIX操作系统、分析和实现

黄祥喜 编著



中山大学出版社

73.8746
C207

热诚为读者服务

MINIX 是一个教学用分时多用户多任务操作系统,由美国学者 A. S. Tanenbaum 1986 年研制初版。MINIX+1.0 是中山大学计算机科学系黄祥喜副教授 1992 年研制的 MINIX 扩充版。专家鉴定委员会认为,MINIX+1.0 是一个良好的用于操作系统教学的实验环境,具有国内领先水平,并在功能和结构上优于 MINIX 初版;建议在全国高校中推广使用。中山大学出版社电脑发展部作为 MINIX+1.0 实验软件的销售代理,乐意向读者提供软件邮购服务。凭学生证购买,可获八折优惠。

中山大学出版社电脑发展部

地址:广州市新港西路 135 号中山大学校内

邮编:510275 联系人:张亚拉 李文

电话:4425565, 446300-7467

责任编辑:张亚拉 封面设计:梁睿思 责任技编:黄少伟

ISBN 7-306-00880-3



ISBN7-306-00880-3/TP·29

定价:28.00 元

9 787306 008800 >

计算机操作系统实验教程

——MINIX 的使用、分析和实现

黄祥喜 编著

中山大学出版社

· 广 州 ·

(粤)新登字 11 号

版权所有 不得翻印

图书在版编目(CIP)数据

计算机操作系统实验教程/黄祥喜编著. —广州:中山大学出版社, 1994-05

ISBN 7—306—00880—3/TP·29

- I 书名
- II 黄祥喜
- III ①计算机②操作系统
- IV TP3

责任编辑:张亚拉 封面设计:梁睿思
责任技编:黄炎伟 电脑排版:张亚拉

中山大学出版社出版发行
(广州市新港西路135号)
广东省新华书店经销
广东科普印刷厂印刷

787×1092毫米 16开本 35.25印张 113万字
1994年5月第1版 1994年5月第1次印刷
印数:1—5000册 定价:28.00元

前 言

操作系统原理是计算机各专业的核心基础课,实验则是操作系统原理教学的一个重要组成部分。可以说,操作系统原理教学效果的好坏很大程度上取决于实验的好坏。这是因为:

1) 操作系统原理中有些概念很抽象、难理解,比如并发、进程等,通过实验,学生可对这些概念获得感性的认识。

2) 操作系统既涉及硬件,又涉及软件。操作系统中的许多问题都是依靠硬件和软件的相互配合而得到解决的,比如进程切换、设备驱动、中断驱动、虚拟存储管理等。通过实验,学生能真正理解计算机系统中硬件和软件的联系和区别、操作系统与硬件的关系、操作系统与应用软件的关系。

3) 操作系统内容繁杂。学生仅从书本上学习这些内容,很难对操作系统有整体的认识。通过实验,学生能将书本上相对独立的内容贯穿起来,对操作系统由静态的了解深化到动态的了解,从而加强对操作系统的整体性认识。

4) 操作系统是计算机系统中的一个核心系统软件,实验可以培养学生分析和设计这一重要系统软件的能力。

基于以上认识,几年来我们对操作系统实验做了深入的研究。特别仔细分析了美国知名操作系统学者 A. S. Tanenbaum 1986 年研制并被美国许多大学采用的教学用操作系统 MINIX (小的 UNIX 型操作系统,详见 A. S. Tanenbaum 著《操作系统教程》,世界图书出版公司,1990 年出版),并组织研究小组对 MINIX 的重编译及功能扩充等问题做了系统的探索,取得一定的成果。近几年,我们在操作系统原理课程的教学贯穿了 MINIX 实例,受到学生的欢迎。1992 年 4 月,我们研制的 MINIX 扩充版 MINIX+1.0 通过了国家教委组织的专家鉴定。来自复旦大学等国内八所重点高校的专家组成的鉴定委员会认为,本项成果具有国内领先水平,建议在全国高校推广使用。1992 年 5 月,《计算机世界》对该成果做了详细报导。本项成果的公布引起了国内同行的关注,到目前为止已有近 20 所高校来人来函索取 MINIX+1.0 软件 and 资料。为了让更多的读者分享我们的成果,特编写了本书。我们期望,本书的出版能为改善我国目前的操作系统实验教学起到一定的推动作用。

本书共分四部分,十九章,对 UNIX 的使用、MINIX 系统程序设计、MINIX 结构分析、MINIX 的重编译、MINIX 的功能扩充及改造等问题做了系统而细致的论述,并同时给出了整个 MINIX 系统的源代码。本书的编写力求做到内容上具有先进性和连贯性,叙述上由浅入深。学生在学习操作系统课程时,结合本书可以完成三个层次的实验,第一层是操作系统的使用,第二层是操作系统结构分析,第三层是操作系统的实现。这三个层次的实验由低到高、由表及里、具有连续性。

本书可作为操作系统原理课程的实验教材,也可作为操作系统分析、操作系统设计等课程的基本教材。另外,我们可向读者提供与本书配套的 MINIX+1.0 实验软件。读者使用此软件可在机器上亲手修改和扩充 MINIX,然后重编译,生成新的 MINIX 系统。对于广大的 UNIX 用户来讲,本书第一部分可作为 UNIX 的入门教材。对于 UNIX 系统开发者来讲,本书提供的 MINIX 源代码及其实验软件对理解 UNIX 的实现思想是十分有用的。MINIX 的消息机制

和“顾客—服务员结构”对网络和分布式操作系统的设计也具有一定的参考价值。

本书由黄祥喜主编。直接参与本书初稿写作的有：梅坚(第一、二、三章)，陈德坚(第四、五章)，黄东斌(第六、八章)，赖毅强、黎伟强(第七、九、十一章)，李小林、郑则仲(第十、十二、十三、十四章)。李小林、郑则仲、赖毅强、梅坚等同志为 MINIX+1.0 的实现做了大量艰苦细致的工作。本书的编写和出版得到了中山大学岭南学院、计算机系、教务处和出版社许多领导、老师的热心支持和帮助，在此一并表示深切谢意。编者特别要感谢中山大学出版社计算机图书编辑室的张亚拉副教授，为了配合“全国第三届操作系统教学研讨会(1994年5月于中山大学)”的召开，他牺牲了许多休息时间，夜以继日地编辑和排版，使这本百万字书稿能在较短的周期内出版。

本书不妥之处恳请读者斧正。

编者

1994年4月15日

目 录

第一部分 MINIX 操作系统的用户界面

1	MINIX 简介	(3)
2	MINIX 的命令及编程环境	(5)
2.1	MINIX 的命令	(5)
2.2	SHELL	(25)
2.3	MINIX 环境中的 C 语言和汇编语言	(34)
2.4	MINIX 环境中程序的编辑	(38)
3	MINIX 的系统调用及其使用	(41)
3.1	关于进程管理的系统调用	(41)
3.2	关于信号机制的系统调用	(43)
3.3	关于文件管理的系统调用	(44)
3.4	关于目录与文件系统管理的系统调用	(48)
3.5	关于文件保护的系统调用	(50)
3.6	关于时间管理的系统调用	(52)
3.7	系统调用的应用	(52)

第二部分 MINIX 操作系统结构分析

4	MINIX 进程调度和通信程序分析	(63)
4.1	MINIX 结构综论	(63)
4.2	MINIX 的主程序和进程表	(67)
4.3	进程调度程序	(71)
4.4	进程通信程序	(75)
5	MINIX 中断处理和设备驱动程序分析	(80)
5.1	I/O 软件的原理	(80)

5.2	中断处理程序	(80)
5.3	设备驱动程序	(86)
6	MINIX 存储管理程序分析	(127)
6.1	存储管理服务(MM)概述	(127)
6.2	内存管理程序	(131)
6.3	系统调用的实现	(132)
7	MINIX 文件系统分析	(135)
7.1	文件系统服务员(FS)概述	(135)
7.2	MINIX 盘结构和目录	(139)
7.3	缓冲池管理	(155)
7.4	专用块和文件系统的装卸	(160)
7.5	保护机制	(168)
7.6	文件的操作和管理	(173)
7.7	管道和特别文件	(191)
7.8	与设备驱动程序的接口	(198)
7.9	文件系统的其它程序	(200)
8	MINIX 进程控制程序分析	(204)
8.1	进程的创建与撤消	(204)
8.2	进程信号处理	(211)
8.3	系统任务(SYSTEM TASK)	(215)
9	MINIX 库函数	(225)

第三部分 MINIX+1.0 操作系统的使用与实现

10	MINIX+1.0 操作系统的使用	(230)
10.1	MINIX+1.0 的特点	(230)
10.2	MINIX+1.0 的配置	(231)
10.3	MINIX+1.0 的启动和退出	(231)
10.4	MINIX+1.0 的安装	(232)
11	MINIX+1.0 的重编译和引导盘的制作	(233)
11.1	MINIX+1.0 系统的构成	(233)
11.2	MINIX+1.0 系统的引导过程	(233)

11.3	硬盘的使用.....	(235)
11.4	MINIX+1.0 的重编译.....	(236)
11.5	MINIX+1.0 引导盘的制作.....	(239)
11.6	新系统的测试.....	(240)
12	对 MINIX 的基本改造和扩充.....	(242)
12.1	MINIX+1.0 对 3.5 寸高密盘的支持.....	(242)
12.2	AT 机扩展内存的利用及 RAM 盘的改造.....	(243)
13	MINIX+1.0 假脱机系统的实现.....	(247)
13.1	引言.....	(247)
13.2	问题的提出与分析.....	(248)
13.3	假脱机系统的实现.....	(249)
13.4	假脱机系统的安装与使用.....	(251)
14	MINIX+1.0 进程对换的实现.....	(253)
14.1	引言.....	(253)
14.2	问题的分析.....	(253)
14.3	进程对换的实现.....	(256)
14.4	系统的协调.....	(260)
实验题	(264)

第四部分 MINIX 操作系统源代码

15	MINIX 头文件.....	(269)
16	MINIX 内核(kernel)源代码.....	(279)
17	MINIX 存储管理(MM)源代码.....	(399)
18	MINIX 文件系统(FS)源代码.....	(439)
19	MINIX 源代码交叉引用表.....	(528)

第一部分

MINIX 操作系统的用户界面

1 MINIX 简介

操作系统是计算机系统的一个重要系统软件,它的作用主要有两个:一是提高计算机资源的使用效率,一是方便用户使用计算机的各种资源。用户使用计算机是通过操作系统的用户界面进行的。因此要学会使用计算机,首先就要掌握操作系统的用户界面。研究操作系统更是如此。一般将操作系统的用户界面分为三个层次:

1) 系统调用层

该层向程序员用户提供一套标准的操作系统功能调用,使程序员可以编制位于操作系统之上的各种系统程序和应用程序,如编译程序、命令解释器 SHELL、命令实现程序、数据库管理系统等。它是下面两层的基础。

2) 命令层

该层向终端用户提供一套完整的操作命令,以帮助用户和计算机进行有效的通信。

3) 作业控制语言层

该层向批处理用户提供一套作业控制语言,使用户可以方便地将自己对作业(程序)的运行要求以“作业说明书”的形式表达出来。

MINIX 的用户界面只有前面两个层次,第三个层次可以部分地由 SHELL 过程(类似于 DOS 的“批命令文件”)实现。MINIX 的用户界面与 UNIX V7 十分相似,UNIX 的 CAT、LS、MAKE、GREP 等常用命令及 SHELL 均配备于 MINIX 中,并发挥着相同的作用。因此,熟悉 UNIX 的人很快就可以学会使用 MINIX。反过来,学会了 MINIX 的使用,也可以很快地掌握 UNIX。

第一部分重点叙述了 MINIX 的命令、编程环境和系统调用。学生在学习该部分内容时,不仅要掌握各个具体命令和系统调用的功能和用法细节,还应特别注意命令与系统调用、SHELL 之间,系统调用与库函数、操作系统核心之间的联系。

下面对 MINIX 作一个简要的介绍。

在目前的 IBM-PC 机上,运行着多种操作系统。其中最流行的是 MS-DOS、XENIX 等。它们都是商业系统,具有复杂的内部结构,要对它们进行细致的分析是很困难的,更难以进行扩充、修改。因为它们的技术资料并不是全部对外公开的!很多东西需要自己慢慢摸索,甚至不得不研究繁琐的反汇编代码。

因此在操作系统原理课程中,以往的教学方法往往重理论而轻实践。学生没有机会自己动手编制或者修改一个操作系统,哪怕是一个操作系统模型。更由于版权的问题,学生可能完全不知道一个实际的操作系统是怎样实现的。

在 UNIX 出现的初期(第六版),由于 AT&T 公司的许可,其源程序曾被广泛使用及经常的研究。当第七版出现时,UNIX 已成为一个商业系统。AT&T 禁止在课程中学习它的源程序,以免泄漏其商业秘密。于是很多的操作系统课程只教理论,甚至略去 UNIX 的学习。在 UNIX 广泛使用的今天,这是很不合适的,因为大家公认 UNIX 包含着许多操作系统的光辉思想。

鉴于这一情况,美国知名学者 ANDREW S. TANENBAUM 1986 年编制了一个称之为

MINIX 的操作系统。因为 IBM PC 正在普遍使用,故本系统最早在 IBM PC 上实现。顾名思义,MINIX 是一个小的 UNIX 系统:与 UNIX 第七版(UNIX V7)兼容。但出于作者的考虑,它没有包含一行 AT&T 的代码,以回避专利的限制。它可方便地用于课堂和个人学习。这样,学生便能对它进行真正的剖析,而无需担心版权的问题。更由于它提供了整个系统的源程序,学生除对它进行研究外,还可以动手进行扩充、修改。

MINIX 与 UNIX V7 有着相似的用户界面。对于使用 PC 机的普通用户来说,MINIX 无异于任何一个 UNIX 系统。但是,MINIX 的内部结构与 UNIX 又是全然不同,它完全避免了 UNIX 中复杂的内部结构。虽然这样做损失了一部分功能,但由于 MINIX 的目的在于教学,因此易读性是它特别强调的。MINIX 出现较晚,结构比较标准。各种应用程序及系统均出自同一批人的手,其中贯彻着结构化程序设计思想。所以对 MINIX 源程序的阅读理解是很容易的。

作为一个完备的操作系统,MINIX 也和一般的操作系统一样由四大部分组成,即进程管理、输入/输出(设备驱动)、存储管理和文件系统。而在其中的各个方面,MINIX 又有着自身的特点。这些特点无非本着短小精悍、易于修改扩充的精神。

2 MINIX 的命令及编程环境

2.1 MINIX 的命令

MINIX 拥有与 UNIX 类似的用户界面。它的 SHELL 与 UNIX V7 功能相同。除 SHELL 外,MINIX 还提供一整套的命令,它们大部分与 UNIX 中的命令相对应,具有相同的调用方式和标志,并执行相同的功能。下面介绍各命令的使用方法。命令语法描述中,方括号内表示任选项,省略号('...')表示前项重复一次或一次以上,'#' 表示注释,其他符号可按字面意义理解。

命令: ar --- 档案文件维护

语法: ar [adprtvx] 档案文件名 [文件名]

标志: a 把“文件名”指示的文件加进“档案文件名”中。

d 从档案中删除文件名。

p 在标准输出打印档案中文件内容。

r 替换旧文件(如果不存在,同 a)。

t 列出档案表内容。

v 给出档案表更详细内容。

x 从档案中取出文件。

例子: ar r clib *.c # 把所有 c 程序放入 clib 中。

命令: basename --- 分解文件的前缀与后缀

语法: basename file [suffix]

标志: 无

例子: basename /user/ast/file # 分解路径产生 file

basename /user/file.c.c # 分解路径和 .c 产生 file

说明: 移去初始目录名而产生自己的文件名。如果第二个参数出现,则它被解释为后缀被分解。这个程序最初用于 SHELL 的命令文件。

命令: cal --- 打印日历。

语法: cal [month] year

标志: 无

例子: cal 3 1987 # 打印 1987 年 3 月的日历。

说明: 年份可从公元一年到公元 9999 年。注意,87 并不代表 1987 而代表 0087 年。

命令: cat --- 连接文件并写到标准输出上

语法: cat [-u] file ...

标志: -u 无缓冲区输出

例子: cat file # 在终端显示文件
cat file 1 file 2 | lpr # 连接两个文件并打印结果

说明: cat 连接它的所有输入文件,并把结果写到标准输出(一般指终端显示器)上。如果没有输入文件名,或遇到“-”作为文件名,Cat 则从标准输入(一般指终端键盘)上读文件。除非给定-u 标志,否则输出以 512 字节为块加以缓冲。“|”表示管道。“命令 1 | 命令 2”的含义是,执行命令 1,其输出结果作为命令 2 的输入。

命令: chmem --- 改变存储分配

语法: chmem [+][-][=] amount file ...

标志: 无

例子: chmem =50000 a.out # 把 50K 的栈空间给 a.out
chmem -4000 a.out # a.out 的栈减少 4000 字节
chmem +1000 file 1 file 2 # 每个栈增加 1000 字节

说明: 当装入一个程序时,chmem 为 text 和 data+bss 段分配足够的内存,再加上栈区。使用 malloc.brk 或 sbrk 可以增大数据段,但从末端减少栈空间。分配栈空间的总量可以由执行程序的文件首部某一字段给出。

设置栈空间的总量必须很小心。设置得太小程序不能运行,设置得太大又浪费存储。MINIX 不支持存储交换,所以,当存储满时,以后的 FORK 调用将失败。编译器为栈设置尽可能大的空间(64K-text-data),有许多程序不需要那么大的栈。对于无递归的程序,它们不调用 brk,sbrk 或 malloc,且没有局部数组,1K 的空间已足够了。

chmem 命令改变文件首部分配栈长度的字段,所以,运行该程序间接地要求内存总量。任选项“=”设置所说明的栈长度;任选项“+”和“-”使当前栈的长度增加或减少所指定的值。命令执行后打出新旧栈的大小。

命令: chmod --- 修改文件保护模式

语法: chmod mode file ...

标志: 无

例子: chmod 754 file # 文件主为 rwx;同组成员 r-x;其他成员 r--
chmod 4755 file 1 file 2 # 置 SETUID 位

说明: 每一个文件的保护位被置成八进制数,04000 位(即第 12 位)为 SETUID(置有效用户号权),02000 位为 SETGID(置有效用户组号权),最低 9 位是文件主,同组成员和其他成员的 rwx。这里,rwx 分别表示用户对文件的“读权”、“写权”和“执行权”。(参见 3.5 节)

命令: chown --- 改变文件主

语法: chown user file ...

标志: 无

例子: chown ast file 1 file 2 # 使文件主为 ast

说明: 该命令将文件主改变为 user(即,注册说明的用户名)。仅超级用户可以使用这条命令。

命令: clr --- 清屏幕

语法: clr

标志: 无

例子: clr # 清屏幕

说明: 屏幕被清为空格。

命令: cmp --- 比较两文件

语法: cmp -ls file 1 file 2

标志: -l 闹模式,打印出不同的字节(八进数)
-s 静模式,返回状态

例子: cmp file 1 file 2 # 指出文件是否相同
cmp -l file 1 file 2 # 打印出所有不同的字节

说明: 该命令比较两个文件。如果它们相同,则返回状态 0; 否则,返回状态 1; 如果文件不能被打开,则返回状态 2; 如果 file 1 是"-",则对标准输入和 file 2 进行比较。

命令: comm --- 打印两个已排序文件的公共行和不同行

语法: comm [-[123]] file1 file2

标志: -1 不打印第 1 列(仅出现在 file1 的行)
-2 不打印第 2 列(仅出现在 file2 的行)
-3 不打印第 3 列(出现在两个文件的公共行)

例子: comm file1 file2 # 打印所有 3 列
comm -12 file1 file2 # 打印同时出现在 file1 和 file2 中的行

说明: 该命令将两个已排序的文件读入并比较,生成 3 列输出。仅在 file1 中的行列于第 1 列,仅在 file2 中的行列于第 2 列,两个文件都出现的行列于第 3 列。文件名为 "-" 表示标准输入。

命令: cp --- 拷贝文件

语法: cp file1 file2 或: cp file ... directory

标志: 无

例子: cp oldfile newfile # 拷贝 oldfile 到 newfile
cp file1 file2 /usr/ast # 拷贝两个文件到 /usr/ast 目录

说明: cp 拷贝一个文件到另一个文件,或拷贝多个文件到一目录中。文件不能拷贝到自己。

命令: date --- 打印或设置日期和时间

语法: date [-q] [[MMDDYY]hhmm[ss]]

标志: -q 从标准输入读日期

例子: date # 打印日期和时间

date 0221881610 # 设置时间为 1988 年 2 月 21 日下午 4 时 10 分

说明: 无参数的 date 命令打印当前日期和时间; 带有一个参数的设置日期和时间。MMDDYY 对应于月、日和年; hhmmss 对应于时、分和秒。其中,每个域都是两位数字。

命令: dd --- 转换并复制文件

语法: dd [option=value] ...

标志: 无

例子: dd if=/dev/fd0 of=/dev/fd1 # 拷贝盘 0 到 盘 1

dd if=x of=y bs=1w skip=4 # 拷贝 x 到 y, 跳过 4 个字

dd if=x of=y count=3 # 拷贝 3 个 512 字节的块

说明: 这个命令的目的是拷贝部分文件。通过任选项可以说明被拷贝文件块的大小、跳过多少块及拷贝的块数。任选项为:

if = file	输入文件(缺省为标准输入)
of = file	输出文件(缺省为标准输出)
ibs = n	输入块大小(缺省为 512 字节)
obs = n	输出块大小(缺省为 512 字节)
bs = n	块大小; 重置 ibs 和 obs (缺省为 512 字节)
skip = n	读前跳过 n 个输入块
seek = n	写前跳过 n 个输出块
count = n	仅拷贝 n 个输入块
conv = lcase	变换大写字母为小写字母
conv = ucuse	变换小写字母为大写字母
conv = swab	交换每一对字节
conv = noerror	忽略错误继续执行

其中,长度单位都为字节,字母 w、b 或 k 分别表示字(两个字节)、块(512)字节或 K(1024 字节)。当 dd 完成时,它输出读、写满块和部分块的数量。

命令: df --- 报告盘的自由空间和 i 节点

语法: df special...