

## 参考文献

- [1] Intel Co. INTEL 80386 Programmer's Reference Manual 1986, INTEL CORPORATION, 1987.
- [2] James L. Turley. Advanced 80386 Programming Techniques. Osborne McGraw-Hill, 1988.
- [3] Brian W. Kernighan, Dennis M. Ritchie. The C programming Language. Prentice-Hall 1988.
- [4] Leland L. Beck. System Software: An Introduction to Systems Programming, 3<sup>rd</sup>. Addison-Wesley, 1997.
- [5] Richard M. Stallman, Using and Porting the GNU Compiler Collection, For GCC Version 2.95, the Free Software Foundation, 1998.
- [6] The Open Group Base Specifications Issue 6 IEEE Std 1003.1-2001, The IEEE and The Open Group.
- [7] David A Rusling, The Linux Kernel, 1999. <http://www.tldp.org/>
- [8] Linux Kernel Source Code, <http://www.kernel.org/>
- [9] Digital co.ltd. VT100 User Guide, <http://www.vt100.net/>
- [10] FreeBSD Online Manual, <http://www.freebsd.org/cgi/man.cgi>
- [11] Maurice J. Bach 著, 陈葆珏, 王旭, 柳纯录, 冯雪山译, UNIX 操作系统设计. 机械工业出版社, 2000 年 4 月.
- [12] John Lions 著, 尤晋元译, 莱昂氏 UNIX 源代码分析, 机械工业出版社, 2000 年 7 月.
- [13] Andrew S. Tanenbaum 著 王鹏, 尤晋元等译, 操作系统: 设计与实现 (第 2 版), 电子工业出版社, 1998 年 8 月.
- [14] Alessandro Rubini, Jonathan 著, 魏永明, 骆刚, 姜君译, Linux 设备驱动程序, 中国电力出版社, 2002 年 11 月.
- [15] Daniel P. Bovet, Marco Cesati 著, 陈莉君, 冯锐, 牛欣源 译, 深入理解 LINUX 内核, 中国电力出版社 2001 年.
- [16] 张载鸿. 微型机(PC 系列)接口控制教程, 清华大学出版社, 1992 年.
- [17] 李风华, 周利华, 赵丽松. MS-DOS 5.0 内核剖析. 西安电子科技大学出版社, 1992 年.
- [18] RedHat 7.3 操作系统在线手册. <http://www.plinux.org/cgi-bin/man.cgi>
- [19] Ivan Bowman, Conceptual Architecture of the Linux Kernel. <http://plg.uwaterloo.ca/~itbowman/CS746G/a1/>
- [20] Clark L. Coleman. Using Inline Assembly with gcc. <http://oldlinux.org/Linux.old/>
- [21]

# 附录

## 附录 1 内核主要常数

### 1 系统最大进程数

系统最大进程（任务）数为 64。

### 2 进程的运行状态

```
#define TASK_RUNNING      0    // 进程正在运行或已准备就绪。
#define TASK_INTERRUPTIBLE 1    // 进程处于可中断等待状态。
#define TASK_UNINTERRUPTIBLE 2  // 进程处于不可中断等待状态，主要用于 I/O 操作等待。
#define TASK_ZOMBIE       3    // 进程处于僵死状态，已经停止运行，但父进程还没发信号。
#define TASK_STOPPED      4    // 进程已停止。
```

### 3 内存页长度

PAGE\_SIZE = 1024 字节

### 3 系统主设备编号

与 Minix 系统的设备编号一样，因此可以使用 minix 的文件系统。

```
0 - 没有用到 (nodev)
1 - /dev/mem      内存设备。
2 - /dev/fd       软盘设备。
3 - /dev/hd       硬盘设备。
4 - /dev/ttyx     tty 串行终端设备。
5 - /dev/tty      tty 终端设备。
6 - /dev/lp       打印设备。
7 - unnamed pipes 没有命名的管道。
```

### 4 硬盘逻辑设备编号方法

由于 1 个硬盘中可以存在 1--4 个分区，因此硬盘还依据分区不同用次设备号进行指定分区。因此硬盘的逻辑设备号由以下方式构成：

设备号 = 主设备号 \* 256 + 次设备号  
 也即 dev\_no = (major << 8) + minor  
 两个硬盘的所有逻辑设备号见下表所示。

附表 1.1 硬盘逻辑设备号

逻辑设备号	对应设备文件	说明
0x300	/dev/hd0	代表整个第 1 个硬盘
0x301	/dev/hd1	表示第 1 个硬盘的第 1 个分区

0x302	/dev/hd2	表示第 1 个硬盘的第 2 个分区
0x303	/dev/hd3	表示第 1 个硬盘的第 3 个分区
0x304	/dev/hd4	表示第 1 个硬盘的第 4 个分区
0x305	/dev/hd5	代表整个第 2 个硬盘
0x306	/dev/hd6	表示第 2 个硬盘的第 1 个分区
0x307	/dev/hd7	表示第 2 个硬盘的第 2 个分区
0x308	/dev/hd8	表示第 2 个硬盘的第 3 个分区
0x309	/dev/hd9	表示第 2 个硬盘的第 4 个分区

其中 0x300 和 0x305 并不与哪个分区对应，而是代表整个硬盘。

从 linux 内核 0.95 版后已经不使用这种烦琐的命名方式，而是使用与现在相同的命名方法了。



## 附录 2 内核数据结构

这里集中列出了内核中的主要数据结构，并给予简单说明，注明了每个结构所在的文件和具体位置。作为阅读时参考。

### 1 执行文件结构 `a.out` (`include/a.out.h`, 第 6 行)

`a.out` (Assembly out) 执行文件头格式结构。

```
struct exec {
    unsigned long a_magic      // 执行文件魔数。使用 N_MAGIC 等宏访问。
    unsigned a_text           // 代码长度，字节数。
    unsigned a_data           // 数据长度，字节数。
    unsigned a_bss            // 文件中的未初始化数据区长度，字节数。
    unsigned a_syms           // 文件中的符号表长度，字节数。
    unsigned a_entry          // 执行开始地址。
    unsigned a_trsize         // 代码重定位信息长度，字节数。
    unsigned a_drsize         // 数据重定位信息长度，字节数。
};
```

### 2 文件锁定操作结构 `flock` (`include/fcntl.h`, 43 行)

文件锁定操作数据结构。

```
struct flock {
    short l_type;             // 锁定类型 (F_RDLCK, F_WRLCK, F_UNLCK)。
    short l_whence;           // 开始偏移 (SEEK_SET, SEEK_CUR 或 SEEK_END)。
    off\_t l_start;             // 阻塞锁定的开始处。相对偏移 (字节数)。
    off\_t l_len;               // 阻塞锁定的大小；如果是 0 则为到文件末尾。
    pid\_t l_pid;              // 加锁的进程 id。
};
```

### 3 `sigaction` 的数据结构 (`include/signal.h`, 48 行)

`sigaction` 的数据结构。

```
struct sigaction {
    void (*sa_handler)(int);
    sigset\_t sa_mask;
    int sa_flags;
    void (*sa_restorer)(void);
};
```

`sa_handler` 是对应某信号指定要采取的行动。可以是上面的 `SIG_DFL`，或者是 `SIG_IGN` 来忽略该信号，也可以是指向处理该信号函数的一个指针。

`sa_mask` 给出了对信号的屏蔽码，在信号程序执行时将阻塞对这些信号的处理。另外，引起触发信号处理的信号也将被阻塞，除非使用了 `SA_NOMASK` 标志。

`sa_flags` 指定改变信号处理过程的信号集。

`sa_restorer` 恢复过程指针，是用于保存原返回的过程指针。

## 4 终端窗口大小属性结构 (include/termios.h, 36 行)

窗口大小 (Window size) 属性结构。在窗口环境中可用于基于屏幕的应用程序。ioctl 中的 TIOCGWINSZ 和 TIOCSWINSZ 用来读取或设置这些信息。

```
struct winsize {
    unsigned short ws_row;        // 窗口字符行数。
    unsigned short ws_col;        // 窗口字符列数。
    unsigned short ws_xpixel;     // 窗口宽度，像素值。
    unsigned short ws_ypixel;     // 窗口高度，像素值。
};
```

## 5 termio(s)结构 (include/termios.h, 44 行)

AT&T 系统 V 的 termio 结构。其中控制字符数据长度 NCC = 8。

```
struct termio {
    unsigned short c_iflag;        // 输入模式标志。
    unsigned short c_oflag;        // 输出模式标志。
    unsigned short c_cflag;        // 控制模式标志。
    unsigned short c_lflag;        // 本地模式标志。
    unsigned char c_line;          // 线路规程 (速率)。
    unsigned char c_cc[NCC];       // 控制字符数组。
};
```

POSIX 的 termios 结构 (第 54 行)。其中控制字符数据长度 NCC = 17。

```
struct termios {
    unsigned long c_iflag;         // 输入模式标志。
    unsigned long c_oflag;         // 输出模式标志。
    unsigned long c_cflag;         // 控制模式标志。
    unsigned long c_lflag;         // 本地模式标志。
    unsigned char c_line;          // 线路规程 (速率)。
    unsigned char c_cc[NCCS];      // 控制字符数组。
};
```

以上定义的两个终端数据结构 termio 和 termios 是分别属于两类 UNIX 系列 (或克隆), termio 是在 AT&T 系统 V 中定义的, 而 termios 是 POSIX 标准指定的。两个结构基本一样, 只是 termio 使用短整数类型定义模式标志集, 而 termios 使用长整数定义模式标志集。由于目前这两种结构都在使用, 因此为了兼容性, 大多数系统都同时支持它们。另外, 以前使用的是一类似的 sgtty 结构, 目前已基本不用。

## 6 时间结构 (include/time.h, 第 18 行)

```
struct tm {
    int tm_sec;                   // 秒数 [0, 59]。
    int tm_min;                   // 分钟数 [0, 59]。
    int tm_hour;                  // 小时数 [0, 59]。
    int tm_mday;                  // 1 个月的天数 [0, 31]。
    int tm_mon;                   // 1 年中月份 [0, 11]。
    int tm_year;                  // 从 1900 年开始的年数。
    int tm_wday;                  // 1 星期中的某天 [0, 6] (星期天 = 0)。
```

```

    int tm_yday;                // 1 年中的某天 [0, 365]。
    int tm_isdst;               // 夏令时标志。
};

```

## 7 文件访问/修改结构 (include/utime.h, 第 6 行)

```

struct utimbuf {
    time_t actime;              // 文件访问时间。从 1970.1.1:0:0 开始的秒数。
    time_t modtime;            // 文件修改时间。从 1970.1.1:0:0 开始的秒数。
};

```

## 8 缓冲区头结构 buffer\_head (include/linux/fs.h, 第 68 行)

缓冲区头数据结构。在程序中常用 bh 来表示 buffer\_head 类型变量的缩写。

```

struct buffer_head {
    char * b_data;              // 指向数据块的指针 (数据块为 1024 字节)。
    unsigned long b_blocknr;    // 块号。
    unsigned short b_dev;       // 数据源的设备号 (0 表示未用)。
    unsigned char b_uptodate;   // 更新标志: 表示数据是否已更新。
    unsigned char b_dirt;       // 修改标志: 0-未修改, 1-已修改。
    unsigned char b_count;      // 使用该数据块的用户数。
    unsigned char b_lock;       // 缓冲区是否被锁定, 0-未锁; 1-已锁定。
    struct task_struct * b_wait; // 指向等待该缓冲区解锁的任务。
    struct buffer_head * b_prev; // 前一块 (这四个指针用于缓冲区的管理)。
    struct buffer_head * b_next; // 下一块。
    struct buffer_head * b_prev_free; // 前一空闲块。
    struct buffer_head * b_next_free; // 下一空闲块。
};

```

## 9 内存中磁盘索引节点结构 (include/linux/fs.h, 第 93 行)

这是在内存中的 i 节点结构。磁盘上的索引节点结构 d\_inode 只包括前 7 项。

```

struct m_inode {
    unsigned short i_mode;      // 文件类型和属性 (rwx 位)。
    unsigned short i_uid;       // 用户 id (文件拥有者标识符)。
    unsigned long i_size;       // 文件大小 (字节数)。
    unsigned long i_mtime;      // 修改时间 (自 1970.1.1:0 算起, 秒)。
    unsigned char i_gid;        // 组 id (文件拥有者所在的组)。
    unsigned char i_nlinks;     // 文件目录项链接数。
    unsigned short i_zone[9];   // 直接 (0-6)、间接 (7) 或双重间接 (8) 逻辑块号。
                                // zone 是区的意思, 可译成区段, 或逻辑块。

    /* these are in memory also */
    struct task_struct * i_wait; // 等待该 i 节点的进程。
    unsigned long i_atime;       // 最后访问时间。
    unsigned long i_ctime;       // i 节点自身修改时间。
    unsigned short i_dev;        // i 节点所在的设备号。
    unsigned short i_num;        // i 节点号。
};

```

```

    unsigned short i_count;        // i 节点被使用的次数，0 表示该 i 节点空闲。
    unsigned char i_lock;         // 锁定标志。
    unsigned char i_dirt;         // 已修改(脏)标志。
    unsigned char i_pipe;         // 管道标志。
    unsigned char i_mount;        // 安装标志。
    unsigned char i_seek;         // 搜寻标志(lseek 时)。
    unsigned char i_update;       // 更新标志。
};

```

## 10 文件结构 (include/linux/fs.h, 第 116 行)

文件结构，用于在文件句柄与 i 节点之间建立关系。

```

struct file {
    unsigned short f_mode;        // 文件操作模式 (RW 位)
    unsigned short f_flags;       // 文件打开和控制的标志。
    unsigned short f_count;       // 对应文件句柄 (文件描述符) 数。
    struct m_inode * f_inode;     // 指向对应 i 节点。
    off_t f_pos;                 // 文件位置 (读写偏移值)。
};

```

## 11 磁盘超级块结构 (include/linux/fs.h, 第 124 行)

内存中磁盘超级块结构。磁盘上的超级块结构 d\_super\_block 只包括前 8 项。

```

struct super_block {
    unsigned short s_ninodes;     // 节点数。
    unsigned short s_nzones;     // 逻辑块数。
    unsigned short s_imap_blocks; // i 节点位图所占用的数据块数。
    unsigned short s_zmap_blocks; // 逻辑块位图所占用的数据块数。
    unsigned short s_firstdatazone; // 第一个数据逻辑块号。
    unsigned short s_log_zone_size; // log(数据块数/逻辑块)。(以 2 为底)。
    unsigned long s_max_size;     // 文件最大长度。
    unsigned short s_magic;       // 文件系统魔数。
/* These are only in memory */
    struct buffer_head * s_imap[8]; // i 节点位图缓冲块指针数组 (占用 8 块, 可表示 64M)。
    struct buffer_head * s_zmap[8]; // 逻辑块位图缓冲块指针数组 (占用 8 块)。
    unsigned short s_dev;         // 超级块所在的设备号。
    struct m_inode * s_isup;      // 被安装的文件系统根目录的 i 节点。(isup=super i)
    struct m_inode * s_imount;    // 被安装到的 i 节点。
    unsigned long s_time;        // 修改时间。
    struct task_struct * s_wait;  // 等待该超级块的进程。
    unsigned char s_lock;         // 被锁定标志。
    unsigned char s_rd_only;      // 只读标志。
    unsigned char s_dirt;        // 已修改(脏)标志。
};

```



## 12 目录项结构 (include/linux/fs.h, 第 157 行)

文件目录项结构。

```
struct dir\_entry {
    unsigned short inode;        // i 节点。
    char name[NAME\_LEN];        // 文件名。
};
```

## 13 硬盘分区表结构 (include/linux/hdreg.h, 第 52 行)

硬盘分区表结构。参见下面列表后信息。

```
struct partition {
    unsigned char boot_ind;      // 引导标志。0x80-该分区可引导操作系统。
    unsigned char head;          // 分区起始磁头号。
    unsigned char sector;        // 分区起始扇区号(位 0-5)和起始柱面号高 2 位(位 6-7)。
    unsigned char cyl;          // 分区起始柱面号低 8 位。
    unsigned char sys_ind;       // 分区类型字节。0x0b-DOS; 0x80-Old Minix; 0x83-Linux
    unsigned char end_head;      // 分区的结束磁头号。
    unsigned char end_sector;    // 结束扇区号(位 0-5)和结束柱面号高 2 位(位 6-7)。
    unsigned char end_cyl;       // 结束柱面号低 8 位。
    unsigned int start_sect;     // 分区起始物理扇区号(从 0 开始计)。
    unsigned int nr_sects;       // 分区占用的扇区数。
};
```

为了实现多个操作系统共享硬盘资源, 硬盘可以在逻辑上分为 1--4 个分区。每个分区之间的扇区号是邻接的。分区表由 4 个表项组成, 每个表项由 16 字节组成, 对应一个分区的信息, 存放有分区的大小和起止的柱面号、磁道号、扇区号和引导标志。分区表存放在硬盘的 0 柱面 0 头第 1 个扇区的 0x1BE--0x1FD 处。4 个分区中同时只能有一个分区是可引导的。

## 14 段描述符结构 (include/linux/head.h, 第 4 行)

CPU 中描述符的简单格式。

```
struct desc\_struct {                // 定义了段描述符的数据结构。该结构仅说明每个描述
    unsigned long a,b;              // 符是由 8 个字节构成, 每个描述符表共有 256 项。
} desc\_table[256];
```

## 15 i387 使用的结构 (include/linux/sched.h, 第 40 行)

这是数学协处理器使用的结构, 主要用于保存进程切换时 i387 的执行状态信息。

```
struct i387\_struct {
    long    cwd;        // 控制字(Control word)。
    long    swd;        // 状态字(Status word)。
    long    twd;        // 标记字(Tag word)。
    long    fip;        // 协处理器代码指针。
    long    fcs;        // 协处理器代码段寄存器。
    long    foo;
    long    fos;
    long    st_space[20]; /* 8*10 bytes for each FP-reg = 80 bytes */
};
```

```
};
```

## 16 任务状态段结构 (include/linux/sched.h, 第 51 行)

任务状态段数据结构 (参见附录)。

```
struct tss\_struct {
    long    back_link;        /* 16 high bits zero */
    long    esp0;
    long    ss0;              /* 16 high bits zero */
    long    esp1;
    long    ss1;              /* 16 high bits zero */
    long    esp2;
    long    ss2;              /* 16 high bits zero */
    long    cr3;
    long    eip;
    long    eflags;
    long    eax, ecx, edx, ebx;
    long    esp;
    long    ebp;
    long    esi;
    long    edi;
    long    es;                /* 16 high bits zero */
    long    cs;                /* 16 high bits zero */
    long    ss;                /* 16 high bits zero */
    long    ds;                /* 16 high bits zero */
    long    fs;                /* 16 high bits zero */
    long    gs;                /* 16 high bits zero */
    long    ldt;               /* 16 high bits zero */
    long    trace_bitmap;     /* bits: trace 0, bitmap 16-31 */
    struct i387\_struct i387;
};
```

## 17 进程 (任务) 数据结构 task (include/linux/sched.h, 第 78 行)

这是任务 (进程) 数据结构, 或称为进程描述符。

```
struct task_struct {
    long state                任务的运行状态 (-1 不可运行, 0 可运行(就绪), >0 已停止)。
    long counter              任务运行时间计数(递减)(滴答数), 运行时间片。
    long priority              运行优先数。任务开始运行时 counter = priority, 越大运行越
    长。
    long signal                信号。是位图, 每个比特位代表一种信号, 信号值=位偏移值+1。
    struct sigaction sigaction[32] 信号执行属性结构, 对应信号将要执行的操作和标志信息。
    long blocked              进程信号屏蔽码 (对应信号位图)。
    int exit_code              任务执行停止的退出码, 其父进程会取。
    unsigned long start_code   代码段地址。
    unsigned long end_code     代码长度 (字节数)。
```

```

unsigned long end_data    数据长度（字节数）。
unsigned long brk        总长度（字节数）。
unsigned long start_stack 堆栈段地址。
long pid                进程标识号(进程号)。
long father             父进程号。
long pgrp               父进程组号。
long session            会话号。
long leader              会话首领。
unsigned short uid       用户标识号（用户 id）。
unsigned short euid      有效用户 id。
unsigned short suid      保存的用户 id。
unsigned short gid       组标识号（组 id）。
unsigned short egid      有效组 id。
unsigned short sgid      保存的组 id。
long alarm               报警定时值（滴答数）。
long utime               用户态运行时间（滴答数）。
long stime               系统态运行时间（滴答数）。
long cutime              子进程用户态运行时间。
long cstime              子进程系统态运行时间。
long start_time          进程开始运行时刻。
unsigned short used_math 标志：是否使用了协处理器。
int tty                  进程使用 tty 的子设备号。-1 表示没有使用。
unsigned short umask      文件创建属性屏蔽位。
struct m_inode * pwd      当前工作目录 i 节点结构。
struct m_inode * root     根目录 i 节点结构。
struct m_inode * executable 执行文件 i 节点结构。
unsigned long close_on_exec 执行时关闭文件句柄位图标志。（参见 include/fcntl.h）
struct file * filp[NR_OPEN] 进程使用的文件表结构。
struct desc_struct ldt[3] 任务的局部描述符表。0-空，1-代码段 cs，2-数据和堆栈段 ds&ss。
struct tss_struct tss     进程的任务状态段信息结构。

```

```
};
```

## 18 tty 等待队列结构（include/linux/tty.h，第 16 行）

tty 等待队列数据结构。

```

struct tty\_queue {
    unsigned long data;                // 等待队列缓冲区中当前数据指针（字符数[??]）。
                                        // 对于串口终端，则存放串口端口地址。

    unsigned long head;                // 缓冲区中数据头指针。
    unsigned long tail;               // 缓冲区中数据尾指针。
    struct task\_struct * proc_list;    // 等待进程列表。
    char buf[TTY_BUF_SIZE];           // 队列的缓冲区。
};

```

## 19 tty 结构 (include/linux/tty.h, 第 45 行)

tty 数据结构。

```
struct tty\_struct {
    struct termios termios;           // 终端 io 属性和控制字符数据结构。
    int pgrp;                         // 所属进程组。
    int stopped;                      // 停止标志。
    void (*write)(struct tty\_struct * tty); // tty 写函数指针。
    struct tty\_queue read_q;          // tty 读队列。
    struct tty\_queue write_q;         // tty 写队列。
    struct tty\_queue secondary;       // tty 辅助队列(存放规范模式字符序列),
};                                   // 可称为规范(熟)模式队列。
extern struct tty\_struct tty\_table[]; // tty 结构数组。
```

## 20 文件状态结构 (include/sys/stat.h, 第 6 行)

```
struct stat {
    dev\_t st_dev;    // 含有文件的设备号。
    ino\_t st_ino;    // 文件 i 节点号。
    umode\_t st_mode; // 文件属性 (见下面)。
    nlink\_t st_nlink; // 指定文件的连接数。
    uid\_t st_uid;    // 文件的用户(标识)号。
    gid\_t st_gid;    // 文件的组号。
    dev\_t st_rdev;   // 设备号(如果文件是特殊的字符文件或块文件)。
    off\_t st_size;   // 文件大小 (字节数) (如果文件是常规文件)。
    time\_t st_atime; // 上次(最后)访问时间。
    time\_t st_mtime; // 最后修改时间。
    time\_t st_ctime; // 最后节点修改时间。
};
```

## 21 文件访问与修改时间结构 (include/sys/times.h, 第 6 行)

```
struct tms {
    time\_t tms_utime; // 用户使用的 CPU 时间。
    time\_t tms_stime; // 系统 (内核) CPU 时间。
    time\_t tms_cutime; // 已终止的子进程使用的用户 CPU 时间。
    time\_t tms_cstime; // 已终止的子进程使用的系统 CPU 时间。
};
```

## 22 ustat 结构 (include/sys/types.h, 第 39 行)

```
struct ustat {
    daddr\_t f_tfree;
    ino\_t f_tinode;
    char f_fname[6];
    char f_fpack[6];
};
```

## 23 系统名称头文件（include/sys/utsname.h，第 6 行）

```
struct utsname {  
    char sysname[9];    // 本版本操作系统的名称。  
    char nodename[9];   // 与实现相关的网络中节点名称。  
    char release[9];    // 本实现的当前发行级别。  
    char version[9];    // 本次发行的版本级别。  
    char machine[9];    // 系统运行的硬件类型名称。  
};
```

## 24 块设备请求项结构（kernel/blk\_dev/blk.h，第 23 行）

下面是请求队列中项的结构。其中如果 dev=-1，则表示没有使用该项。

```
struct request {  
    int dev;                // 使用的设备号，未用时为-1。  
    int cmd;                // 命令(READ 或 WRITE)。  
    int errors;             // 作时产生的错误次数。  
    unsigned long sector;   // 起始扇区。(1 块=2 扇区)  
    unsigned long nr_sectors; // 读/写扇区数。  
    char * buffer;          // 数据缓冲区。  
    struct task\_struct * waiting; // 任务等待操作执行完成的地方。  
    struct buffer\_head * bh;   // 缓冲区头指针(include/linux/fs.h, 68)。  
    struct request * next;    // 指向下一请求项。  
};
```





P	保留										E	T	E	M	P	CRO
G	Reserved										T	S	M	P	E	

控制寄存器 CR0 含有系统整体的控制标志。其中：

- PE - 保护模式开启位 (Protection Enable, 比特位 0)。如果设置了该比特位, 就会使处理器开始在保护模式下运行。
- MP - 协处理器存在标志 (Math Present, 比特位 1)。用于控制 WAIT 指令的功能, 以配合协处理的运行。
- EM - 仿真控制 (Emulation, 比特位 2)。指示是否需要仿真协处理器的功能。
- TS - 任务切换 (Task Switch, 比特位 3)。每当任务切换时处理器就会设置该比特位, 并且在解释协处理器指令之前测试该位。
- ET - 扩展类型 (Extention Type, 比特位 4)。该位指出了系统中所含有的协处理器类型 (是 80287 还是 80387)。
- PG - 分页操作 (Paging, 比特位 31)。该位指示出是否使用页表将线性地址变换成物理地址。

## 内存管理

内存管理主要涉及处理器的内存寻址机制。80x86 使用两步将一个分段形式的逻辑地址转换为实际物理内存地址。

- 段变换, 将一个由段选择符和段内偏移构成的逻辑地址转换为一个线性地址;
- 页变换, 将线性地址转换为对应的物理地址。该步是可选的。

在分页机制开启时, 通过将前面所述的段转换和页转换组合在一起, 即实现了从逻辑地址到物理地址的两个转换阶段。

### 段变换

下图示出了处理器是如何将一个逻辑地址转换为线性地址的。在转换过程中 CPU 使用了以下一些数据结构:

- 段描述符 (Segment Descriptors);
- 描述符表 (Descriptor tables);
- 选择符 (Selectors);
- 段寄存器 (Segment Registers)。

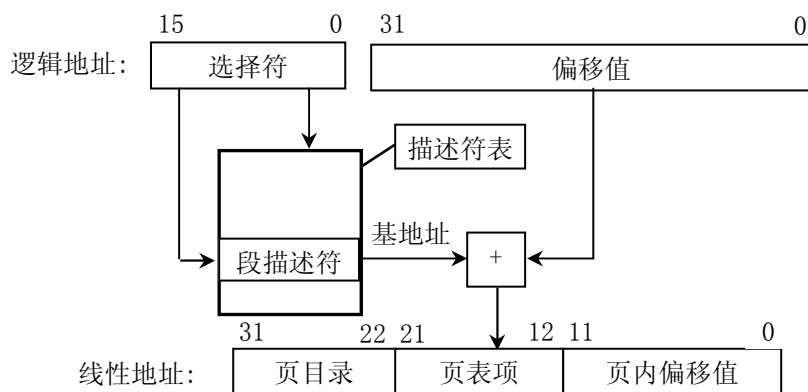
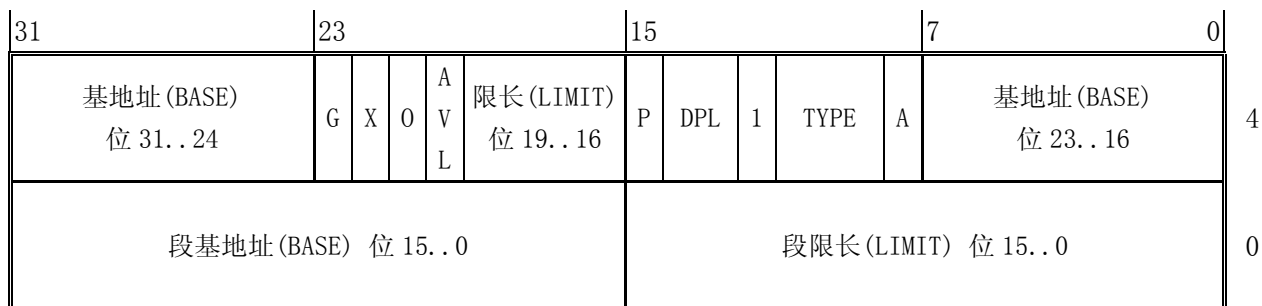


图 段变换示意图

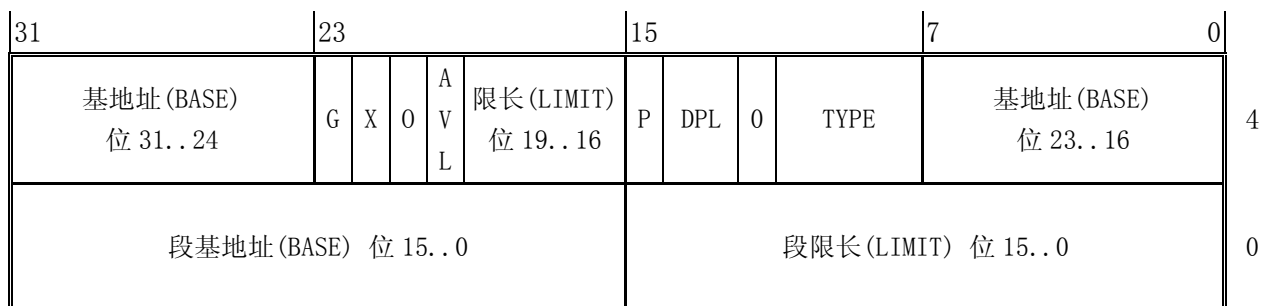


## 段描述符

段描述符向 CPU 提供了将逻辑地址映射为线性地址所必要的信息。描述符是由程序编译器、链接器、加载器或操作系统创建的。下图示出了描述符的两种一般格式。所有种类的描述符都具有这两种格式之一。段描述符的各个字段的含义如下：



a. 用于程序代码段和数据段的描述符



b. 用于特殊系统段的描述符

图 描述符的一般格式

**基地址 (BASE)：**定义段在 4GB 线性空间中的位置。处理器会将基地址的三个部分组合成一个 32 位的值。  
**段限长 (LIMIT)：**定义了段的最大长度。处理器将组合段限长的两个部分形成一个 20 位的值。处理器会依据颗粒度 (Granularity) 位字段的值来解释段限长域的实际含义：

1. 当以 1 字节为单元时，则定义了最高可为 1MB 字节的长度；
2. 当以 4KB 字节为单元时，则定义了最高可为 4GB 字节的长度。在加载时限长值将左移 12 位。

**颗粒度 (Granularity)：**指定了限长字段值代表的单元含义。当为 0 时，限长单元值为 1 字节；当该位为 1 时，限长的单元值为 4KB 字节。

**类型 (TYPE)：**用于区分各种不同类型的描述符。

**描述符特权级 (Descriptor Privilege Level - DPL)：**用于保护机制。共有 4 级：0-3。0 级是最高特权级，3 级是最低特权级。

**段存在位 (Segment-Present bit - P)：**如果该位为零，则该描述符无效，不能用于地址变换过程。当指向该描述符的选择符被加载到段寄存器中时，处理器就会发出一个异常信号。

**访问位 (Accessed bit - A)：**当处理器访问过该段时就会设置该比特位。

## 描述符表

段描述符是保存在描述符表中的，有两类描述符表：

- 全局描述符表 (Global descriptor table - GDT)；
- 局部描述符表 (Local descriptor table - LDT)。

描述符表是由 8 字节构成的描述符项的内存中的一个数组，见下图所示。描述符表的长度是可变的，最多可以含有 8192 ( $2^{13}$ ) 个描述符。但是对于 GDT 表，其第一个描述符 (索引 0) 是不用的。

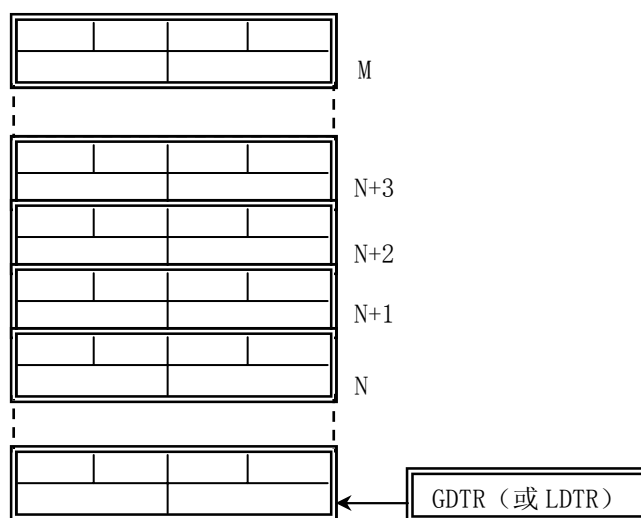


图 描述符表示意图

处理器是通过使用 GDTR 和 LDTR 寄存器来定位 GDT 表和当前的 LDT 表。这两个寄存器以线性地址的方式保存了描述符表的基地址和表的长度。指令 lgdt 和 sgdt 用于访问 GDTR 寄存器；指令 lldt 和 slldt 用于访问 LDTR 寄存器。lgdt 使用的是内存中一个 6 字节操作数来加载 GDTR 寄存器的。头两个字节代表描述符表的长度，后 4 个字节是描述符表的基地址。然而请注意，访问 LDTR 寄存器的指令 lldt 所使用的操作数却是一个 2 字节的操作数，表示全局描述符表 GDT 中一个描述符项的选择符。该选择符所对应的 GDT 表中的描述符项应该对应一个局部描述符表。选择符的含义见下面说明。



## 选择符(Selectors)

逻辑地址的选择符部分是用于指定一描述符的，它是通过指定一描述符表并且索引其中的一个描述符项完成的。下图示出了选择符的格式。各字段的含义为：

索引值(Index)：用于选择指定描述符表中 8192 个描述符中的一个。处理器将该索引值乘上 8(描述符的字节长度)，并加上描述符表的基地址即可访问表中指定的段描述符。

表指示器(Table Indicator - TI)：指定选择符所引用的描述符表。值为 0 表示指定 GDT 表，值为 1 表示指定当前的 LDT 表。

请求者的特权级(Requestor's Privilege Level - RPL)：用于保护机制。



由于 GDT 表的第一项(索引值为 0)没有被使用，因此一个具有索引值 0 和表指示器值也为 0 的选择符（也即指向 GDT 的第一项的选择符）可以用作为一个空(null)选择符。当一个段寄存器（不能是 CS 或 SS）加载了一个空选择符时，处理器并不会产生一个异常。但是若使用这个段寄存器访问内存时就会

产生一个异常。对于初始化还未使用的段寄存器以陷入意外的引用来说，这个特性是很有用的。

## 段寄存器

处理器将描述符中的信息保存在段寄存器中，因而可以避免在每次访问内存时查询描述符表。

每个段寄存器都有一个“可见”部分和一个“不可见”部分，见下图所示。这些段地址寄存器的可见部分是由程序来操作的，就好象它们只是简单的 16 位寄存器。不可见部分则是由处理器来处理的。

对这些寄存器的加载操作使用的是普通程序指令，这些指令可以分为两类：

1. 直接加载指令；例如，MOV，POP，LDS，LSS，LGS，LFS。这些指令显式地引用了指定的段寄存器。
2. 隐式加载指令；例如，远调用 CALL 和远跳转 JMP。这些指令隐式地引用了 CS 段寄存器，并用新值加载到 CS 中。

程序使用这些指令会把 16 位的选择符加载到段寄存器的可见部分，而处理器则会自动地从描述符表中将一个描述符的基地址、段限长、类型以及其它信息加载到段寄存器中的不可见部分中去。

	16 位可见部分	隐藏部分
CS		
SS		
DS		
ES		
FS		
GS		

## 页变换（翻译）

在地址变换的第二阶段，CPU 将线性地址转换为物理地址。地址变换的这个阶段实现了基于分页的虚拟内存系统和分页级保护的基本功能。页变化这一步是可选的，页变换仅在设置了 CR0 的 PG 比特位后才起作用，该比特位是在软件初始化时由操作系统设置的。如果操作系统需要实现多个虚拟 8086 任务、基于分页的保护机制或基于分页的虚拟内存，那么就一定要设置该位。

### 页框（帧）(Page Frame)

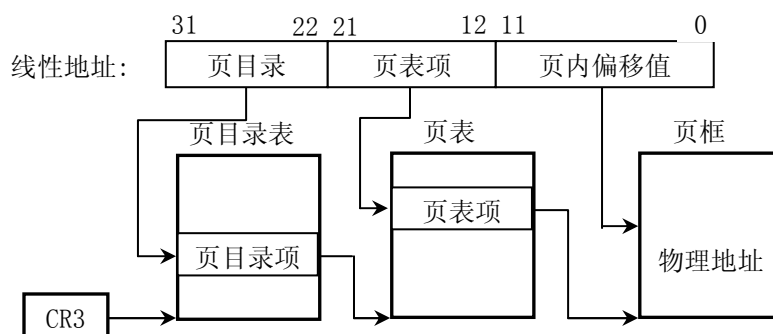
页框是一个物理内存地址连续的 4K 字节单元。它以字节为边界，大小固定。

### 线性地址(Linear Address)

线性地址通过指定一个页表、页表中的某一页以及该页中的偏移值，从而间接地指向对应的物理地址。下图示出了线性地址的格式。

31	22	21	12	11	0
页目录 (DIR)		页 (PAGE)		偏移值 (OFFSET)	

下图示出了处理器将一个线性地址转换成物理地址的方法。通过使用两级页表，处理器将一个线性地址的页目录字段(DIR)、页字段(PAGE)和偏移字段(OFFSET)翻译成对应的物理地址。寻址机制使用线性地址的页目录字段作为页目录中的索引值、使用页表字段作为页目录所指定页表中的索引值、使用偏移字段作为页表所确定的内存页中的字节偏移值。



## 页表(Page Table)

页表只是一个简单的 32 位页指示器的数组。页表本身也是一页内存，因此它含有 4K 字节的内存，可容纳 1K 个 32 位的项。

这里使用了两级页表来定位一页内存页。最高层是页目录，页目录可定位最多 1K 个第二级页表，而每个二级页表可以定位最多 1K 内存页。因此，一个页目录定位的所有页表可以寻址 1M 内存页 ( $2^{20}$ )。由于每一页内存含有 4K 字节 ( $2^{12}$ )，最终一个页目录所指定的页表可以寻址 80386 的整个物理地址空间 ( $2^{20} * 2^{12} = 2^{32}$ )。

当前页目录的物理地址是存储在 CPU 控制寄存器 CR3 中的，因此该寄存器也被称为页目录基地址寄存器 (page directory base register - PDBR)。内存管理软件可以选择对所有的任务只使用一个页目录，或每个任务使用一个页目录，也可以组合两个任务使用一个页目录。

## 页表项(Page-Table Entries)

各级页表所使用的页表项是相同的，其格式见下图所示。

31	12	11	0
页框地址 位 31..12 (PAGE FRAME ADDRESS)		可用 (AVAIL)	<div> <div>0</div> <div>0</div> <div>D</div> <div>A</div> <div>0</div> <div>0</div> <div>/</div> <div>U</div> <div>/</div> <div>R</div> <div>/</div> <div>S</div> <div>/</div> <div>W</div> <div>P</div> </div>

其中，页框地址 (PAGE FRAME ADDRESS) 指定了一页内存的物理起始地址。因为内存页是位于 4K 边界上的，所以其低 12 比特总是 0。在一个页目录中，页表项的页框地址是一个页表的起始地址；在第二级页表中，页表项的页框地址是包含期望内存操作的页框的地址。

存在位 (PRESENT - P) 确定了一个页表项是否可以用于地址转换过程。P=1 表示该项可用。当目录表项或第二级表项的 P=0 时，则该表项时无效的，不能用于地址转换过程。此时该表项的其它所有比特位都可供程序使用；处理器不对这些位进行测试。

当 CPU 试图使用一个页表项进行地址转换时，如果此时任意一级页表项的 P=0，则处理器就会发出页异常信号。对于支持分页虚拟内存的软件系统中，页不存在 (page-not-present) 异常处理程序就可以把所请求的页加入到物理内存中。此时导致异常的指令就可以被重新执行。

已访问 (Accessed - A) 和已修改 (Dirty - D) 比特位提供了有关页使用的信息。除了页目录项中的已修改位，这些比特位将由硬件置位，但不复位。

在对一页内存进行读或写操作之前，处理器将设置相关的目录和二级页表项的已访问位。在向一个二级页表项所涵盖的地址进行写操作之前，处理器将设置该二级页表项的已修改位，而页目录项中的已修改位是不用的。当需求的内存超出实际物理内存量时，支持分页虚拟内存的操作系统可以使用这些位

来确定那些页可以从内存中取走。操作系统必须负责检测和复位这些比特位。

读/写位 (Read/Write - R/W) 和用户/超级用户位 (User/Supervisor - U/S) 并不用于地址转换，但用于分页级的保护机制，是由处理器在地址转换过程中同时操作的。

### 页转换高速缓冲

为了最大地提高地址转换的效率，处理器将最近所使用的页表数据存放在芯片上的高速缓冲中。操作系统设计人员必须在当前页表改变时刷新高速缓冲，可使用以下两种方式之一：

1. 通过使用 MOV 指令重新加载 CR3 页目录基址寄存器；
2. 通过执行一个任务切换。

## 多任务 (Multitasking)

为了提供有效的、受保护的多任务机制，80x86 使用了一些特殊的数据结构。支持多任务运行的寄存器和数据结构主要有任务状态段 (Task State Segment) 和任务寄存器 (Task register)。使用这些数据结构，CPU 可以快速地从任务的执行切换到另一个任务，并保存原有任务的内容。

### 任务状态段 (Task State Segment - TSS)

处理器管理一个任务的所有信息存储在一个特殊类型的段中，即任务状态段 TSS。下图给出了 TSS 的格式。其中的字段可分为两类：

- 处理器只读其中信息的静态字段集 (图中灰色部分)；
- 每次任务切换时处理器将会更新的动态字段集。

31	23	15	7	0	
I/O 映射图基地址 (MAP BASE)					64
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0					局部描述符表 (LDT) 的选择符 60
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0					GS 5C
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0					FS 58
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0					DS 54
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0					SS 50
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0					CS 4C
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0					ES 48
EDI					44
ESI					40
EBP					3C
ESP					38
EBX					34
EDX					30
ECX					2C
EAX					28
EFLAGS					24
指令指针 (EIP)					20
页目录基址寄存器 CR3 (PDBR)					1C
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0					SS2 18
ESP2					14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0					SS1 10

ESP1																0C
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0										SS0						08
ESP0																04
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0										前一执行任务 TSS 的描述符						00

任务状态段 TSS 可以处于线性空间的任何位置。TSS 与其它段一样，也是使用段描述符来定义的。访问 TSS 的描述符会导致任务切换。因此，在大多数系统中都将描述符的 DPL（描述符特权级）字段设置为最高特权级 0，这样就可以只允许可信的软件执行任务的切换。TSS 的描述符只能放在全局描述符表 GDT 中。

## 任务寄存器

任务寄存器（Task Register - TR）的作用与一般段寄存器的类似，它通过指向 TSS 来确定当前执行的任务。它也有 16 位的可见部分和不可见部分。可见部分中的选择符用于在 GDT 表中选择一个 TSS 描述符，处理器使用不可见部分来存放描述符中的基地址和段限长值。指令 LTR 和 STR 用于修改和读取任务寄存器中的可见部分，指令所使用的操作数是一 16 位的选择符。

另外，还有一种提供对 TSS 间接、受保护引用的任务门描述符（Task Gate Descriptor）。这种描述符是在一般段描述符格式的基地址位 15..0 字段（第 3、4 字节）中存放的是一个 TSS 描述符的选择符，并利用其中的特权级字段（DPL）来控制使用描述符执行任务切换的权限。见下面有关中断描述符表 IDT 描述符中的说明。

在以下 4 种情况下，CPU 会切换执行的任务：

1. 当前任务执行了一条引用 TSS 描述符的 JMP 或 CALL 指令；
2. 当前任务执行了一条引用任务门的 JMP 或 CALL 指令；
3. 引用了中断描述符表（IDT）中任务门的中断或异常；
4. 当嵌套任务标志 NT 置位时，当前任务执行了一个 IRET 指令。

## 中断和异常

中断和异常是一种特殊类型的控制转换。它们改变了正常程序流而去处理其它的事件（例如外部事件、出错报告或异常条件）。中断与异常的主要区别在于中断常用于处理 CPU 外部的异步事件，而异常则是处理 CPU 在执行过程中本身检测到的问题。

外部中断源有两种：由 CPU 的 INTR 引脚输入的可屏蔽中断和 NMI 引脚输入的不可屏蔽中断。同样，异常也有两类：由 CPU 检测到的出错、陷阱或放弃事件以及编程设置的“软中断”（如 INT 3 指令等）。

处理器使用标识号（中断号）来识别每种类型的中断或异常。处理器所能识别的不可屏蔽中断 NMI 和异常的标识号是预先确定的，范围是 0 到 31（0x00-0x1f）。目前这些号码并没有全都使用，未确定的号码由 Intel 公司留作今后使用。

可屏蔽中断的标识号由外部中断控制器（如 8259A 可编程中断控制器）确定，并在 CPU 的中断识别阶段通知 CPU。8259A 所分配的中断号可以通过编程指定，可使用的标识号范围是 32 到 255（0x20-0xff）。Linux 系统将 32-47 分配给了可屏蔽中断，余下的 48-255 用来标识其它软中断。当 Linux 只使用了号码 128（0x80）作为系统调用的中断向量号。

## 中断描述符表

中断描述符表（Interrupt Descriptor Table - IDT）将每个中断或异常标识号与处理相应事件程序指令的一个描述符相关联。与 GDT 和 LDT 相似，IDT 是一个 8 字节描述符数组，但其第 1 项可以含有一个描述符。处理器通过将中断号异常号乘上 8 即可索引 IDT 中对应的描述符。IDT 可以位于物理内存的任何地方。处理器是使用 IDT 寄存器（IDTR）来定位 IDT 的。修改和复制 IDT 的指令是 LIDT 和 SIDT。

与 GDT 表的操作一样，IDT 也是使用 6 字节数据的内存地址作为操作数的。前两个字节表示表的限长，后 4 个字节是表的线性基地址。

## IDT 描述符

在中断描述符表 IDT 中可以含有三类描述符中的任意一种：

- 任务门 (Task gates)；
- 中断门 (Interrupt gates)；
- 陷阱门 (Trap gates)；

下图给出了任务门、中断门和陷阱门描述符的格式。

31	23	15	7	0
(未使用)		P	DPL	0 0 1 0 1
TSS 段选择符 (SELECTOR)		(未使用)		

80X86 任务门描述符

31	23	15	7	0
偏移值(OFFSET) 位 31..16		P	DPL	0 1 1 1 0
段选择符 (SELECTOR)		偏移值 (OFFSET) 位 15..0		

80X86 中断门描述符

31	23	15	7	0
偏移值(OFFSET) 位 31..16		P	DPL	0 1 1 1 1
段选择符 (SELECTOR)		偏移值 (OFFSET) 位 15..0		

80X86 陷阱门描述符

## 中断任务和中断过程

正如 CALL 指令能调用一个过程或任务一样，一个中断或异常也能“调用”中断处理程序，该程序是一个过程或一个任务。当响应一个中断或异常时，CPU 使用中断或异常的标识号来索引 IDT 表中的描述符。如果 CPU 索引到一个中断门或陷阱门时，它就调用处理过程；如果是一个任务门，它就引起任务切换。

中断门或陷阱门间接地指向一个过程，该过程将在当前执行任务上下文中执行。门描述符中的段选择符指向 GDT 或当前 LDT 中的一个可执行段的描述符。门描述符中的偏移字段值指向中断或异常处理过程的开始处。

80X86 执行一个中断或异常处理过程的方式与 CALL 指令调用一个过程的方式非常相似，只是两者在使用堆栈上略有不同。中断会在把原指令指针压入堆栈之前，把原标志寄存器 EFLAGS 的内容也推入堆栈中。对于与段有关的异常，CPU 还会将一个错误码压入异常处理程序的堆栈上。

对于中断过程处理结束的返回操作，中断返回指令 IRET 与 RET 相似，但是 IRET 为了去除压入堆栈的 EFLAGS 值，ESP 会多递增 4 个字节。

中断门与陷阱门的区别在于对中断允许标志 IF 的影响。由中断门向量引起的中断会复位 IF，因为可以避免其它中断干扰当前中断的处理。随后的 IRET 指令会从堆栈上恢复 IF 的原值；而通过陷阱门产生的中断不会改变 IF。

IDT 表中的任务门描述符间接地指向一个任务状态段 TSS。任务门描述符中的段选择符指向 GDT 表中的一个 TSS 描述符。当产生的中断或异常指向 IDT 中的一个任务门描述符，就会导致任务切换，从而会在独立的任务中处理中断。Linux 系统中并没有使用任务门描述符。

# 索引

由于内核代码相对比较庞大，很多变量/函数在源代码的很多程序中被使用/调用，因此对其进行索引比较困难。本索引主要根据变量或函数名称给出定义它的程序文件名、行号和所在页码。

\_\_strtok  
 include/string.h, 275, 定义为变量  
 \_\_GNU\_EXEC\_MACROS\_\_  
 include/a.out.h, 4, 定义为预处理宏  
 \_\_LIBRARY\_\_  
 init/main.c, 7, 定义为预处理宏  
 lib/close.c, 7, 定义为预处理宏  
 lib/dup.c, 7, 定义为预处理宏  
 lib/\_exit.c, 7, 定义为预处理宏  
 lib/open.c, 7, 定义为预处理宏  
 lib/execve.c, 7, 定义为预处理宏  
 lib/setuid.c, 7, 定义为预处理宏  
 lib/string.c, 13, 定义为预处理宏  
 lib/wait.c, 7, 定义为预处理宏  
 lib/write.c, 7, 定义为预处理宏  
 \_\_NR\_access  
 include/unistd.h, 93, 定义为预处理宏  
 \_\_NR\_acct  
 include/unistd.h, 111, 定义为预处理宏  
 \_\_NR\_alarm  
 include/unistd.h, 87, 定义为预处理宏  
 \_\_NR\_break  
 include/unistd.h, 77, 定义为预处理宏  
 \_\_NR\_brk  
 include/unistd.h, 105, 定义为预处理宏  
 \_\_NR\_chdir  
 include/unistd.h, 72, 定义为预处理宏  
 \_\_NR\_chmod  
 include/unistd.h, 75, 定义为预处理宏  
 \_\_NR\_chown  
 include/unistd.h, 76, 定义为预处理宏  
 \_\_NR\_chroot  
 include/unistd.h, 121, 定义为预处理宏  
 \_\_NR\_close  
 include/unistd.h, 66, 定义为预处理宏  
 \_\_NR\_creat  
 include/unistd.h, 68, 定义为预处理宏  
 \_\_NR\_dup  
 include/unistd.h, 101, 定义为预处理宏  
 \_\_NR\_dup2  
 include/unistd.h, 123, 定义为预处理宏  
 \_\_NR\_execve  
 include/unistd.h, 71, 定义为预处理宏  
 \_\_NR\_exit  
 include/unistd.h, 61, 定义为预处理宏  
 \_\_NR\_fcntl  
 include/unistd.h, 115, 定义为预处理宏  
 \_\_NR\_fork  
 include/unistd.h, 62, 定义为预处理宏  
 \_\_NR\_fstat  
 include/unistd.h, 88, 定义为预处理宏  
 \_\_NR\_ftime  
 include/unistd.h, 95, 定义为预处理宏  
 \_\_NR\_getegid  
 include/unistd.h, 110, 定义为预处理宏  
 \_\_NR\_geteuid  
 include/unistd.h, 109, 定义为预处理宏  
 \_\_NR\_getgid  
 include/unistd.h, 107, 定义为预处理宏  
 \_\_NR\_getpgrp  
 include/unistd.h, 125, 定义为预处理宏  
 \_\_NR\_getpid  
 include/unistd.h, 80, 定义为预处理宏  
 \_\_NR\_getppid  
 include/unistd.h, 124, 定义为预处理宏  
 \_\_NR\_getuid  
 include/unistd.h, 84, 定义为预处理宏  
 \_\_NR\_gtty  
 include/unistd.h, 92, 定义为预处理宏  
 \_\_NR\_ioctl  
 include/unistd.h, 114, 定义为预处理宏  
 \_\_NR\_kill



- include/unistd.h, 97, 定义为预处理宏  
\_\_NR\_link
- include/unistd.h, 69, 定义为预处理宏  
\_\_NR\_lock
- include/unistd.h, 113, 定义为预处理宏  
\_\_NR\_lseek
- include/unistd.h, 79, 定义为预处理宏  
\_\_NR\_mkdir
- include/unistd.h, 99, 定义为预处理宏  
\_\_NR\_mknod
- include/unistd.h, 74, 定义为预处理宏  
\_\_NR\_mount
- include/unistd.h, 81, 定义为预处理宏  
\_\_NR\_mpx
- include/unistd.h, 116, 定义为预处理宏  
\_\_NR\_nice
- include/unistd.h, 94, 定义为预处理宏  
\_\_NR\_open
- include/unistd.h, 65, 定义为预处理宏  
\_\_NR\_pause
- include/unistd.h, 89, 定义为预处理宏  
\_\_NR\_phys
- include/unistd.h, 112, 定义为预处理宏  
\_\_NR\_pipe
- include/unistd.h, 102, 定义为预处理宏  
\_\_NR\_prof
- include/unistd.h, 104, 定义为预处理宏  
\_\_NR\_ptrace
- include/unistd.h, 86, 定义为预处理宏  
\_\_NR\_read
- include/unistd.h, 63, 定义为预处理宏  
\_\_NR\_rename
- include/unistd.h, 98, 定义为预处理宏  
\_\_NR\_rmdir
- include/unistd.h, 100, 定义为预处理宏  
\_\_NR\_setgid
- include/unistd.h, 106, 定义为预处理宏  
\_\_NR\_setpgid
- include/unistd.h, 117, 定义为预处理宏  
\_\_NR\_setregid
- include/unistd.h, 131, 定义为预处理宏  
\_\_NR\_setreuid
- include/unistd.h, 130, 定义为预处理宏  
\_\_NR\_setsid
- include/unistd.h, 126, 定义为预处理宏  
\_\_NR\_setuid
- include/unistd.h, 83, 定义为预处理宏  
\_\_NR\_setup
- include/unistd.h, 60, 定义为预处理宏  
\_\_NR\_sgetmask
- include/unistd.h, 128, 定义为预处理宏  
\_\_NR\_sigaction
- include/unistd.h, 127, 定义为预处理宏  
\_\_NR\_signal
- include/unistd.h, 108, 定义为预处理宏  
\_\_NR\_ssetmask
- include/unistd.h, 129, 定义为预处理宏  
\_\_NR\_stat
- include/unistd.h, 78, 定义为预处理宏  
\_\_NR\_stime
- include/unistd.h, 85, 定义为预处理宏  
\_\_NR\_stty
- include/unistd.h, 91, 定义为预处理宏  
\_\_NR\_sync
- include/unistd.h, 96, 定义为预处理宏  
\_\_NR\_time
- include/unistd.h, 73, 定义为预处理宏  
\_\_NR\_times
- include/unistd.h, 103, 定义为预处理宏  
\_\_NR\_ulimit
- include/unistd.h, 118, 定义为预处理宏  
\_\_NR\_umask
- include/unistd.h, 120, 定义为预处理宏  
\_\_NR\_umount
- include/unistd.h, 82, 定义为预处理宏  
\_\_NR\_uname
- include/unistd.h, 119, 定义为预处理宏  
\_\_NR\_unlink
- include/unistd.h, 70, 定义为预处理宏  
\_\_NR\_ustat
- include/unistd.h, 122, 定义为预处理宏  
\_\_NR\_utime
- include/unistd.h, 90, 定义为预处理宏  
\_\_NR\_waitpid
- include/unistd.h, 67, 定义为预处理宏  
\_\_NR\_write
- include/unistd.h, 64, 定义为预处理宏  
\_\_va\_rounded\_size

- include/stdarg.h, 9, 定义为预处理宏
- \_A\_OUT\_H
- include/a.out.h, 2, 定义为预处理宏
- \_BLK\_H
- kernel/blk\_drv/blk.h, 2, 定义为预处理宏
- \_BLOCKABLE
- kernel/sched.c, 24, 定义为预处理宏
- \_bmap
- fs/inode.c, 72, 定义为函数
- \_bucket\_dir
- lib/malloc.c, 60, 定义为struct类型
- \_C
- include/ctype.h, 7, 定义为预处理宏
- \_CONFIG\_H
- include/config.h, 2, 定义为预处理宏
- \_CONST\_H
- include/const.h, 2, 定义为预处理宏
- \_ctmp
- include/ctype.h, 14, 定义为变量
- lib/ctype.c, 9, 定义为变量
- \_ctype
- include/ctype.h, 13, 定义为变量
- lib/ctype.c, 10, 定义为变量
- \_CTYPE\_H
- include/ctype.h, 2, 定义为预处理宏
- \_D
- include/ctype.h, 6, 定义为预处理宏
- \_ERRNO\_H
- include/errno.h, 2, 定义为预处理宏
- \_exit
- include/unistd.h, 208, 定义为函数原型
- lib/\_exit.c, 10, 定义为函数
- \_FCNTL\_H
- include/fcntl.h, 2, 定义为预处理宏
- \_FDREG\_H
- include/fdreg.h, 7, 定义为预处理宏
- \_fs
- kernel/traps.c, 34, 定义为预处理宏
- \_FS\_H
- include/fs.h, 7, 定义为预处理宏
- \_get\_base
- include/sched.h, 214, 定义为预处理宏
- \_hashfn
- fs/buffer.c, 128, 定义为预处理宏
- \_HDREG\_H
- include/hdreg.h, 7, 定义为预处理宏
- \_HEAD\_H
- include/head.h, 2, 定义为预处理宏
- \_HIGH
- include/sys/wait.h, 7, 定义为预处理宏
- \_I\_FLAG
- kernel/chr\_drv/tty\_io.c, 29, 定义为预处理宏
- \_L
- include/ctype.h, 5, 定义为预处理宏
- \_L\_FLAG
- kernel/chr\_drv/tty\_io.c, 28, 定义为预处理宏
- \_LDT
- include/sched.h, 156, 定义为预处理宏
- \_LOW
- include/sys/wait.h, 6, 定义为预处理宏
- \_MM\_H
- include/mm.h, 2, 定义为预处理宏
- \_N\_BADMAG
- include/a.out.h, 36, 定义为预处理宏
- \_N\_HDROFF
- include/a.out.h, 40, 定义为预处理宏
- \_N\_SEGMENT\_ROUND
- include/a.out.h, 95, 定义为预处理宏
- \_N\_TXTENDADDR
- include/a.out.h, 97, 定义为预处理宏
- \_NSIG
- include/signal.h, 9, 定义为预处理宏
- \_O\_FLAG
- kernel/chr\_drv/tty\_io.c, 30, 定义为预处理宏
- \_P
- include/ctype.h, 8, 定义为预处理宏
- \_PC\_CHOWN\_RESTRICTED
- include/unistd.h, 51, 定义为预处理宏
- \_PC\_LINK\_MAX
- include/unistd.h, 43, 定义为预处理宏
- \_PC\_MAX\_CANON
- include/unistd.h, 44, 定义为预处理宏
- \_PC\_MAX\_INPUT
- include/unistd.h, 45, 定义为预处理宏
- \_PC\_NAME\_MAX
- include/unistd.h, 46, 定义为预处理宏
- \_PC\_NO\_TRUNC
- include/unistd.h, 49, 定义为预处理宏

- `_PC_PATH_MAX`  
include/unistd.h, 47, 定义为预处理宏
- `_PC_PIPE_BUF`  
include/unistd.h, 48, 定义为预处理宏
- `_PC_VDISABLE`  
include/unistd.h, 50, 定义为预处理宏
- `_POSIX_CHOWN_RESTRICTED`  
include/unistd.h, 7, 定义为预处理宏
- `_POSIX_NO_TRUNC`  
include/unistd.h, 8, 定义为预处理宏
- `_POSIX_VDISABLE`  
include/unistd.h, 9, 定义为预处理宏
- `_POSIX_VERSION`  
include/unistd.h, 5, 定义为预处理宏
- `_PTRDIFF_T`  
include/sys/types.h, 15, 定义为预处理宏
- `_S`  
include/ctype.h, 9, 定义为预处理宏
- `kernel/sched.c`, 23, 定义为预处理宏
- `_SC_ARG_MAX`  
include/unistd.h, 33, 定义为预处理宏
- `_SC_CHILD_MAX`  
include/unistd.h, 34, 定义为预处理宏
- `_SC_CLOCKS_PER_SEC`  
include/unistd.h, 35, 定义为预处理宏
- `_SC_JOB_CONTROL`  
include/unistd.h, 38, 定义为预处理宏
- `_SC_NGROUPS_MAX`  
include/unistd.h, 36, 定义为预处理宏
- `_SC_OPEN_MAX`  
include/unistd.h, 37, 定义为预处理宏
- `_SC_SAVED_IDS`  
include/unistd.h, 39, 定义为预处理宏
- `_SC_VERSION`  
include/unistd.h, 40, 定义为预处理宏
- `_SCHED_H`  
include/sched.h, 2, 定义为预处理宏
- `_set_base`  
include/sched.h, 188, 定义为预处理宏
- `_set_gate`  
include/asm/system.h, 22, 定义为预处理宏
- `_set_limit`  
include/sched.h, 199, 定义为预处理宏
- `_set_seg_desc`  
include/asm/system.h, 42, 定义为预处理宏
- `_set_tssldt_desc`  
include/asm/system.h, 52, 定义为预处理宏
- `_SIGNAL_H`  
include/signal.h, 2, 定义为预处理宏
- `_SIZE_T`  
include/sys/types.h, 5, 定义为预处理宏
- `include/time.h`, 10, 定义为预处理宏
- `include/stddef.h`, 10, 定义为预处理宏
- `include/string.h`, 9, 定义为预处理宏
- `_SP`  
include/ctype.h, 11, 定义为预处理宏
- `_STDARG_H`  
include/stdarg.h, 2, 定义为预处理宏
- `_STDDEF_H`  
include/stddef.h, 2, 定义为预处理宏
- `_STRING_H`  
include/string.h, 2, 定义为预处理宏
- `_SYS_STAT_H`  
include/sys/stat.h, 2, 定义为预处理宏
- `_SYS_TYPES_H`  
include/sys/types.h, 2, 定义为预处理宏
- `_SYS_UTSNAME_H`  
include/sys/utsname.h, 2, 定义为预处理宏
- `_SYS_WAIT_H`  
include/sys/wait.h, 2, 定义为预处理宏
- `_syscall0`  
include/unistd.h, 133, 定义为预处理宏
- `_syscall1`  
include/unistd.h, 146, 定义为预处理宏
- `_syscall2`  
include/unistd.h, 159, 定义为预处理宏
- `_syscall3`  
include/unistd.h, 172, 定义为预处理宏
- `_TERMIOS_H`  
include/termios.h, 2, 定义为预处理宏
- `_TIME_H`  
include/time.h, 2, 定义为预处理宏
- `_TIME_T`  
include/sys/types.h, 10, 定义为预处理宏
- `include/time.h`, 5, 定义为预处理宏
- `_TIMES_H`  
include/sys/times.h, 2, 定义为预处理宏

- `_TSS`  
`include/sched.h`, 155, 定义为预处理宏
- `_TTY_H`  
`include/tty.h`, 10, 定义为预处理宏
- `_U`  
`include/ctype.h`, 4, 定义为预处理宏
- `_UNISTD_H`  
`include/unistd.h`, 2, 定义为预处理宏
- `_UTIME_H`  
`include/utime.h`, 2, 定义为预处理宏
- `_X`  
`include/ctype.h`, 10, 定义为预处理宏
- `ABRT_ERR`  
`include/hdreg.h`, 47, 定义为预处理宏
- `ACC_MODE`  
`fs/namei.c`, 21, 定义为预处理宏
- `access`  
`include/unistd.h`, 189, 定义为函数原型
- `acct`  
`include/unistd.h`, 190, 定义为函数原型
- `add_entry`  
`fs/namei.c`, 165, 定义为函数
- `add_request`  
`kernel/blk_drv/ll_rw_blk.c`, 64, 定义为函数
- `add_timer`  
`include/sched.h`, 144, 定义为函数原型
- `kernel/sched.c`, 272, 定义为函数
- `alarm`  
`include/unistd.h`, 191, 定义为函数原型
- `ALRMASK`  
`kernel/chr_drv/tty_io.c`, 17, 定义为预处理宏
- `argv`  
`init/main.c`, 165, 定义为变量
- `argv_rc`  
`init/main.c`, 162, 定义为变量
- `asctime`  
`include/time.h`, 35, 定义为函数原型
- `attr`  
`kernel/chr_drv/console.c`, 77, 定义为变量
- `B0`  
`include/termios.h`, 133, 定义为预处理宏
- `B110`  
`include/termios.h`, 136, 定义为预处理宏
- `B1200`  
`include/termios.h`, 142, 定义为预处理宏
- `B134`  
`include/termios.h`, 137, 定义为预处理宏
- `B150`  
`include/termios.h`, 138, 定义为预处理宏
- `B1800`  
`include/termios.h`, 143, 定义为预处理宏
- `B19200`  
`include/termios.h`, 147, 定义为预处理宏
- `B200`  
`include/termios.h`, 139, 定义为预处理宏
- `B2400`  
`include/termios.h`, 144, 定义为预处理宏
- `B300`  
`include/termios.h`, 140, 定义为预处理宏
- `B38400`  
`include/termios.h`, 148, 定义为预处理宏
- `B4800`  
`include/termios.h`, 145, 定义为预处理宏
- `B50`  
`include/termios.h`, 134, 定义为预处理宏
- `B600`  
`include/termios.h`, 141, 定义为预处理宏
- `B75`  
`include/termios.h`, 135, 定义为预处理宏
- `B9600`  
`include/termios.h`, 146, 定义为预处理宏
- `bad_flp_intr`  
`kernel/blk_drv/floppy.c`, 233, 定义为函数
- `bad_rw_intr`  
`kernel/blk_drv/hd.c`, 242, 定义为函数
- `BADNESS`  
`fs/buffer.c`, 205, 定义为预处理宏
- `BBD_ERR`  
`include/hdreg.h`, 50, 定义为预处理宏
- `BCD_TO_BIN`  
`init/main.c`, 74, 定义为预处理宏
- `beepcount`  
`kernel/chr_drv/console.c`, 697, 定义为变量
- `blk_dev`  
`kernel/blk_drv/ll_rw_blk.c`, 32, 定义为struct类型
- `kernel/blk_drv/blk.h`, 50, 定义为struct类型
- `blk_dev_init`

- init/main.c, 46, 定义为函数原型  
kernel/blk\_drv/ll\_rw\_blk.c, 157, 定义为函数  
blk\_dev\_struct  
kernel/blk\_drv/blk.h, 45, 定义为struct类型  
block\_read  
fs/read\_write.c, 18, 定义为函数原型  
fs/block\_dev.c, 47, 定义为函数  
BLOCK\_SIZE  
include/fs.h, 49, 定义为预处理宏  
BLOCK\_SIZE\_BITS  
include/fs.h, 50, 定义为预处理宏  
block\_write  
fs/read\_write.c, 19, 定义为函数原型  
fs/block\_dev.c, 14, 定义为函数  
bmap  
fs/inode.c, 140, 定义为函数  
include/fs.h, 176, 定义为函数原型  
bottom  
kernel/chr\_drv/console.c, 73, 定义为变量  
bounds  
kernel/traps.c, 48, 定义为函数原型  
bread  
fs/buffer.c, 267, 定义为函数  
include/fs.h, 189, 定义为函数原型  
bread\_page  
fs/buffer.c, 296, 定义为函数  
include/fs.h, 190, 定义为函数原型  
breada  
fs/buffer.c, 322, 定义为函数  
include/fs.h, 191, 定义为函数原型  
brelse  
fs/buffer.c, 253, 定义为函数  
include/fs.h, 188, 定义为函数原型  
brk  
include/unistd.h, 192, 定义为函数原型  
BRKINT  
include/termios.h, 84, 定义为预处理宏  
BS0  
include/termios.h, 122, 定义为预处理宏  
BS1  
include/termios.h, 123, 定义为预处理宏  
BSDLY  
include/termios.h, 121, 定义为预处理宏  
bucket\_desc  
lib/malloc.c, 52, 定义为struct类型  
bucket\_dir  
lib/malloc.c, 77, 定义为变量  
buffer\_block  
include/fs.h, 66, 定义为类型  
BUFFER\_END  
include/const.h, 4, 定义为预处理宏  
buffer\_head  
include/fs.h, 68, 定义为struct类型  
buffer\_init  
fs/buffer.c, 348, 定义为函数  
include/fs.h, 31, 定义为函数原型  
buffer\_memory\_end  
init/main.c, 99, 定义为变量  
buffer\_wait  
fs/buffer.c, 33, 定义为变量  
BUSY\_STAT  
include/hdreg.h, 31, 定义为预处理宏  
calc\_mem  
mm/memory.c, 413, 定义为函数  
CBAUD  
include/termios.h, 132, 定义为预处理宏  
cfgetispeed  
include/termios.h, 216, 定义为函数原型  
cfgetospeed  
include/termios.h, 217, 定义为函数原型  
cfsetispeed  
include/termios.h, 218, 定义为函数原型  
cfsetospeed  
include/termios.h, 219, 定义为函数原型  
change\_ldt  
fs/exec.c, 154, 定义为函数  
change\_speed  
kernel/chr\_drv/tty\_ioctl.c, 24, 定义为函数  
CHARS  
include/tty.h, 30, 定义为预处理宏  
chdir  
include/unistd.h, 194, 定义为函数原型  
check\_disk\_change  
fs/buffer.c, 113, 定义为函数  
include/fs.h, 168, 定义为函数原型  
chmod  
include/sys/stat.h, 51, 定义为函数原型  
include/unistd.h, 195, 定义为函数原型

- chown  
include/unistd.h, 196, 定义为函数原型  
chr\_dev\_init  
init/main.c, 47, 定义为函数原型  
kernel/chr\_drv/tty\_io.c, 347, 定义为函数  
chroot  
include/unistd.h, 197, 定义为函数原型  
CIBAUD  
include/termios.h, 162, 定义为预处理宏  
clear\_bit  
fs/bitmap.c, 25, 定义为预处理宏  
clear\_block  
fs/bitmap.c, 13, 定义为预处理宏  
cli  
include/asm/system.h, 17, 定义为预处理宏  
CLOCAL  
include/termios.h, 161, 定义为预处理宏  
clock  
include/time.h, 30, 定义为函数原型  
clock\_t  
include/time.h, 16, 定义为类型  
CLOCKS\_PER\_SEC  
include/time.h, 14, 定义为预处理宏  
close  
include/unistd.h, 198, 定义为函数原型  
CMOS\_READ  
init/main.c, 69, 定义为预处理宏  
kernel/blk\_drv/hd.c, 28, 定义为预处理宏  
CODE\_SPACE  
mm/memory.c, 49, 定义为预处理宏  
command  
kernel/blk\_drv/floppy.c, 121, 定义为变量  
con\_init  
include/tty.h, 66, 定义为函数原型  
kernel/chr\_drv/console.c, 617, 定义为函数  
con\_write  
include/tty.h, 73, 定义为函数原型  
kernel/chr\_drv/console.c, 445, 定义为函数  
controller\_ready  
kernel/blk\_drv/hd.c, 161, 定义为函数  
coprocessor\_error  
kernel/traps.c, 58, 定义为函数原型  
coprocessor\_segment\_overrun  
kernel/traps.c, 52, 定义为函数原型  
copy\_buffer  
kernel/blk\_drv/floppy.c, 155, 定义为预处理宏  
copy\_mem  
kernel/fork.c, 39, 定义为函数  
copy\_page  
mm/memory.c, 54, 定义为预处理宏  
copy\_page\_tables  
include/sched.h, 29, 定义为函数原型  
mm/memory.c, 150, 定义为函数  
copy\_process  
kernel/fork.c, 68, 定义为函数  
copy\_strings  
fs/exec.c, 104, 定义为函数  
copy\_to\_cooked  
include/tty.h, 75, 定义为函数原型  
kernel/chr\_drv/tty\_io.c, 145, 定义为函数  
COPYBLK  
fs/buffer.c, 283, 定义为预处理宏  
cp\_stat  
fs/stat.c, 15, 定义为函数  
CPARENB  
include/termios.h, 158, 定义为预处理宏  
CPARODD  
include/termios.h, 159, 定义为预处理宏  
cr  
kernel/chr\_drv/console.c, 224, 定义为函数  
CRO  
include/termios.h, 111, 定义为预处理宏  
CR1  
include/termios.h, 112, 定义为预处理宏  
CR2  
include/termios.h, 113, 定义为预处理宏  
CR3  
include/termios.h, 114, 定义为预处理宏  
CRDLY  
include/termios.h, 110, 定义为预处理宏  
CREAD  
include/termios.h, 157, 定义为预处理宏  
creat  
include/unistd.h, 199, 定义为函数原型  
include/fcntl.h, 51, 定义为函数原型  
create\_block  
fs/inode.c, 145, 定义为函数  
include/fs.h, 177, 定义为函数原型

- create\_tables  
fs/exec.c, 46, 定义为函数
- CRTSCTS  
include/termios.h, 163, 定义为预处理宏
- crw\_ptr  
fs/char\_dev.c, 19, 定义为类型
- crw\_table  
fs/char\_dev.c, 85, 定义为变量
- CS5  
include/termios.h, 152, 定义为预处理宏
- CS6  
include/termios.h, 153, 定义为预处理宏
- CS7  
include/termios.h, 154, 定义为预处理宏
- CS8  
include/termios.h, 155, 定义为预处理宏
- csi\_at  
kernel/chr\_drv/console.c, 391, 定义为函数
- csi\_J  
kernel/chr\_drv/console.c, 239, 定义为函数
- csi\_K  
kernel/chr\_drv/console.c, 268, 定义为函数
- csi\_L  
kernel/chr\_drv/console.c, 401, 定义为函数
- csi\_m  
kernel/chr\_drv/console.c, 299, 定义为函数
- csi\_M  
kernel/chr\_drv/console.c, 421, 定义为函数
- csi\_P  
kernel/chr\_drv/console.c, 411, 定义为函数
- CSIZE  
include/termios.h, 151, 定义为预处理宏
- CSTOPB  
include/termios.h, 156, 定义为预处理宏
- ctime  
include/time.h, 36, 定义为函数原型
- cur\_rate  
kernel/blk\_drv/floppy.c, 113, 定义为变量
- cur\_spec1  
kernel/blk\_drv/floppy.c, 112, 定义为变量
- CURRENT  
kernel/blk\_drv/blk.h, 93, 定义为预处理宏
- CURRENT\_DEV  
kernel/blk\_drv/blk.h, 94, 定义为预处理宏
- current\_DOR  
kernel/sched.c, 204, 定义为变量
- kernel/blk\_drv/floppy.c, 48, 定义为变量
- current\_drive  
kernel/blk\_drv/floppy.c, 115, 定义为变量
- CURRENT\_TIME  
include/sched.h, 142, 定义为预处理宏
- current\_track  
kernel/blk\_drv/floppy.c, 120, 定义为变量
- d\_inode  
include/fs.h, 83, 定义为struct类型
- d\_super\_block  
include/fs.h, 146, 定义为struct类型
- daddr\_t  
include/sys/types.h, 31, 定义为类型
- DAY  
kernel/mktime.c, 22, 定义为预处理宏
- debug  
kernel/traps.c, 44, 定义为函数原型
- DEC  
include/tty.h, 25, 定义为预处理宏
- DEFAULT\_MAJOR\_ROOT  
tools/build.c, 37, 定义为预处理宏
- DEFAULT\_MINOR\_ROOT  
tools/build.c, 38, 定义为预处理宏
- del  
kernel/chr\_drv/console.c, 230, 定义为函数
- delete\_char  
kernel/chr\_drv/console.c, 363, 定义为函数
- delete\_line  
kernel/chr\_drv/console.c, 378, 定义为函数
- desc\_struct  
include/head.h, 4, 定义为struct类型
- desc\_table  
include/head.h, 6, 定义为类型
- dev\_t  
include/sys/types.h, 26, 定义为类型
- DEVICE\_INTR  
kernel/blk\_drv/blk.h, 72, 定义为预处理宏
- kernel/blk\_drv/blk.h, 81, 定义为预处理宏
- kernel/blk\_drv/blk.h, 97, 定义为函数原型
- DEVICE\_NAME  
kernel/blk\_drv/blk.h, 63, 定义为预处理宏
- kernel/blk\_drv/blk.h, 71, 定义为预处理宏

- kernel/blk\_drv/blk.h, 80, 定义为预处理宏  
device\_not\_available  
kernel/traps.c, 50, 定义为函数原型  
DEVICE\_NR  
kernel/blk\_drv/blk.h, 65, 定义为预处理宏  
kernel/blk\_drv/blk.h, 74, 定义为预处理宏  
kernel/blk\_drv/blk.h, 83, 定义为预处理宏  
DEVICE\_OFF  
kernel/blk\_drv/blk.h, 67, 定义为预处理宏  
kernel/blk\_drv/blk.h, 76, 定义为预处理宏  
kernel/blk\_drv/blk.h, 85, 定义为预处理宏  
DEVICE\_ON  
kernel/blk\_drv/blk.h, 66, 定义为预处理宏  
kernel/blk\_drv/blk.h, 75, 定义为预处理宏  
kernel/blk\_drv/blk.h, 84, 定义为预处理宏  
DEVICE\_REQUEST  
kernel/blk\_drv/blk.h, 64, 定义为预处理宏  
kernel/blk\_drv/blk.h, 73, 定义为预处理宏  
kernel/blk\_drv/blk.h, 82, 定义为预处理宏  
kernel/blk\_drv/blk.h, 99, 定义为函数原型  
die  
kernel/traps.c, 63, 定义为函数  
tools/build.c, 46, 定义为函数  
difftime  
include/time.h, 32, 定义为函数原型  
DIR\_ENTRIES\_PER\_BLOCK  
include/fs.h, 56, 定义为预处理宏  
dir\_entry  
include/fs.h, 157, 定义为struct类型  
dir\_namei  
fs/namei.c, 278, 定义为函数  
div\_t  
include/sys/types.h, 36, 定义为类型  
divide\_error  
kernel/traps.c, 43, 定义为函数原型  
DMA\_READ  
include/fdreg.h, 68, 定义为预处理宏  
DMA\_WRITE  
include/fdreg.h, 69, 定义为预处理宏  
do\_bounds  
kernel/traps.c, 134, 定义为函数  
do\_coprocessor\_error  
kernel/traps.c, 169, 定义为函数  
do\_coprocessor\_segment\_overrun  
kernel/traps.c, 149, 定义为函数  
do\_debug  
kernel/traps.c, 124, 定义为函数  
do\_device\_not\_available  
kernel/traps.c, 144, 定义为函数  
do\_div  
kernel/vsprintf.c, 35, 定义为预处理宏  
do\_divide\_error  
kernel/traps.c, 97, 定义为函数  
do\_double\_fault  
kernel/traps.c, 87, 定义为函数  
do\_execve  
fs/exec.c, 182, 定义为函数  
do\_exit  
kernel/exit.c, 102, 定义为函数  
kernel/traps.c, 39, 定义为函数原型  
kernel/signal.c, 13, 定义为函数原型  
mm/memory.c, 31, 定义为函数原型  
do\_fd\_request  
kernel/blk\_drv/floppy.c, 417, 定义为函数  
do\_floppy\_timer  
kernel/sched.c, 245, 定义为函数  
do\_general\_protection  
kernel/traps.c, 92, 定义为函数  
do\_hd\_request  
kernel/blk\_drv/hd.c, 294, 定义为函数  
do\_int3  
kernel/traps.c, 102, 定义为函数  
do\_invalid\_op  
kernel/traps.c, 139, 定义为函数  
do\_invalid\_TSS  
kernel/traps.c, 154, 定义为函数  
do\_nmi  
kernel/traps.c, 119, 定义为函数  
do\_no\_page  
mm/memory.c, 365, 定义为函数  
do\_overflow  
kernel/traps.c, 129, 定义为函数  
do\_rd\_request  
kernel/blk\_drv/ramdisk.c, 23, 定义为函数  
do\_reserved  
kernel/traps.c, 176, 定义为函数  
do\_segment\_not\_present  
kernel/traps.c, 159, 定义为函数



- do\_signal
- kernel/signal.c, 82, 定义为函数
- do\_stack\_segment
- kernel/traps.c, 164, 定义为函数
- do\_timer
- kernel/sched.c, 305, 定义为函数
- do\_tty\_interrupt
- kernel/chr\_drv/tty\_io.c, 342, 定义为函数
- do\_wp\_page
- mm/memory.c, 247, 定义为函数
- double\_fault
- kernel/traps.c, 51, 定义为函数原型
- DRIVE
- kernel/blk\_drv/floppy.c, 54, 定义为预处理宏
- drive\_busy
- kernel/blk\_drv/hd.c, 202, 定义为函数
- DRIVE\_INFO
- init/main.c, 59, 定义为预处理宏
- drive\_info
- init/main.c, 102, 定义为struct类型
- DRQ\_STAT
- include/hdreg.h, 27, 定义为预处理宏
- dup
- include/unistd.h, 200, 定义为函数原型
- dup2
- include/unistd.h, 248, 定义为函数原型
- dupfd
- fs/fcntl.c, 18, 定义为函数
- E2BIG
- include/errno.h, 26, 定义为预处理宏
- EACCES
- include/errno.h, 32, 定义为预处理宏
- EAGAIN
- include/errno.h, 30, 定义为预处理宏
- EBADF
- include/errno.h, 28, 定义为预处理宏
- EBUSY
- include/errno.h, 35, 定义为预处理宏
- ECC\_ERR
- include/hdreg.h, 49, 定义为预处理宏
- ECC\_STAT
- include/hdreg.h, 26, 定义为预处理宏
- ECHILD
- include/errno.h, 29, 定义为预处理宏
- ECHO
- include/termios.h, 172, 定义为预处理宏
- ECHOCTL
- include/termios.h, 178, 定义为预处理宏
- ECHOE
- include/termios.h, 173, 定义为预处理宏
- ECHOK
- include/termios.h, 174, 定义为预处理宏
- ECHOKE
- include/termios.h, 180, 定义为预处理宏
- ECHONL
- include/termios.h, 175, 定义为预处理宏
- ECHOPRT
- include/termios.h, 179, 定义为预处理宏
- EDEADLK
- include/errno.h, 54, 定义为预处理宏
- EDOM
- include/errno.h, 52, 定义为预处理宏
- EEXIST
- include/errno.h, 36, 定义为预处理宏
- EFAULT
- include/errno.h, 33, 定义为预处理宏
- EFBIG
- include/errno.h, 46, 定义为预处理宏
- EINTR
- include/errno.h, 23, 定义为预处理宏
- EINVAL
- include/errno.h, 41, 定义为预处理宏
- EIO
- include/errno.h, 24, 定义为预处理宏
- EISDIR
- include/errno.h, 40, 定义为预处理宏
- EMFILE
- include/errno.h, 43, 定义为预处理宏
- EMLINK
- include/errno.h, 50, 定义为预处理宏
- EMPTY
- include/tty.h, 26, 定义为预处理宏
- empty\_dir
- fs/namei.c, 543, 定义为函数
- ENAMETOOLONG
- include/errno.h, 55, 定义为预处理宏
- end
- fs/buffer.c, 29, 定义为变量

- end\_request  
kernel/blk\_drv/blk.h, 109, 定义为函数
- ENFILE  
include/errno.h, 42, 定义为预处理宏
- ENODEV  
include/errno.h, 38, 定义为预处理宏
- ENOENT  
include/errno.h, 21, 定义为预处理宏
- ENOEXEC  
include/errno.h, 27, 定义为预处理宏
- ENOLCK  
include/errno.h, 56, 定义为预处理宏
- ENOMEM  
include/errno.h, 31, 定义为预处理宏
- ENOSPC  
include/errno.h, 47, 定义为预处理宏
- ENOSYS  
include/errno.h, 57, 定义为预处理宏
- ENOTBLK  
include/errno.h, 34, 定义为预处理宏
- ENOTDIR  
include/errno.h, 39, 定义为预处理宏
- ENOTEMPTY  
include/errno.h, 58, 定义为预处理宏
- ENOTTY  
include/errno.h, 44, 定义为预处理宏
- envp  
init/main.c, 166, 定义为变量
- envp\_rc  
init/main.c, 163, 定义为变量
- ENXIO  
include/errno.h, 25, 定义为预处理宏
- EOF\_CHAR  
include/tty.h, 40, 定义为预处理宏
- EPERM  
include/errno.h, 20, 定义为预处理宏
- EPIPE  
include/errno.h, 51, 定义为预处理宏
- ERANGE  
include/errno.h, 53, 定义为预处理宏
- ERASE\_CHAR  
include/tty.h, 38, 定义为预处理宏
- EROFS  
include/errno.h, 49, 定义为预处理宏
- ERR\_STAT  
include/hdreg.h, 24, 定义为预处理宏
- errno  
include/unistd.h, 187, 定义为变量  
include/errno.h, 17, 定义为变量  
lib/errno.c, 7, 定义为变量
- ERROR  
include/errno.h, 19, 定义为预处理宏
- ESPIPE  
include/errno.h, 48, 定义为预处理宏
- ESRCH  
include/errno.h, 22, 定义为预处理宏
- ETXTBSY  
include/errno.h, 45, 定义为预处理宏
- EXDEV  
include/errno.h, 37, 定义为预处理宏
- exec  
include/a.out.h, 6, 定义为struct类型
- execl  
include/unistd.h, 204, 定义为函数原型
- execle  
include/unistd.h, 206, 定义为函数原型
- execlp  
include/unistd.h, 205, 定义为函数原型
- execv  
include/unistd.h, 202, 定义为函数原型
- execve  
include/unistd.h, 201, 定义为函数原型
- execvp  
include/unistd.h, 203, 定义为函数原型
- exit  
include/unistd.h, 207, 定义为函数原型
- EXT\_MEM\_K  
init/main.c, 58, 定义为预处理宏
- EXTA  
include/termios.h, 149, 定义为预处理宏
- EXTB  
include/termios.h, 150, 定义为预处理宏
- F\_DUPFD  
include/fcntl.h, 23, 定义为预处理宏
- F\_GETFD  
include/fcntl.h, 24, 定义为预处理宏
- F\_GETFL  
include/fcntl.h, 26, 定义为预处理宏

- F\_GETLK  
include/fcntl.h, 28, 定义为预处理宏
- F\_OK  
include/unistd.h, 22, 定义为预处理宏
- F\_RDLCK  
include/fcntl.h, 38, 定义为预处理宏
- F\_SETFD  
include/fcntl.h, 25, 定义为预处理宏
- F\_SETFL  
include/fcntl.h, 27, 定义为预处理宏
- F\_SETLK  
include/fcntl.h, 29, 定义为预处理宏
- F\_SETLKW  
include/fcntl.h, 30, 定义为预处理宏
- F\_UNLCK  
include/fcntl.h, 40, 定义为预处理宏
- F\_WRLCK  
include/fcntl.h, 39, 定义为预处理宏
- fcntl  
include/unistd.h, 209, 定义为函数原型  
include/fcntl.h, 52, 定义为函数原型
- FD\_CLOEXEC  
include/fcntl.h, 33, 定义为预处理宏
- FD\_DATA  
include/fdreg.h, 17, 定义为预处理宏
- FD\_DCR  
include/fdreg.h, 20, 定义为预处理宏
- FD\_DIR  
include/fdreg.h, 19, 定义为预处理宏
- FD\_DOR  
include/fdreg.h, 18, 定义为预处理宏
- FD\_READ  
include/fdreg.h, 62, 定义为预处理宏
- FD\_RECALIBRATE  
include/fdreg.h, 60, 定义为预处理宏
- FD\_SEEK  
include/fdreg.h, 61, 定义为预处理宏
- FD\_SENSEI  
include/fdreg.h, 64, 定义为预处理宏
- FD\_SPECIFY  
include/fdreg.h, 65, 定义为预处理宏
- FD\_STATUS  
include/fdreg.h, 16, 定义为预处理宏
- FD\_WRITE  
include/fdreg.h, 63, 定义为预处理宏
- FFO  
include/termios.h, 128, 定义为预处理宏
- FF1  
include/termios.h, 129, 定义为预处理宏
- FFDLY  
include/termios.h, 127, 定义为预处理宏
- file  
include/fs.h, 116, 定义为struct类型
- file\_read  
fs/read\_write.c, 20, 定义为函数原型
- fs/file\_dev.c, 17, 定义为函数
- file\_table  
fs/file\_table.c, 9, 定义为变量
- include/fs.h, 163, 定义为变量
- file\_write  
fs/read\_write.c, 22, 定义为函数原型
- fs/file\_dev.c, 48, 定义为函数
- find\_buffer  
fs/buffer.c, 166, 定义为函数
- find\_empty\_process  
kernel/fork.c, 135, 定义为函数
- find\_entry  
fs/namei.c, 91, 定义为函数
- find\_first\_zero  
fs/bitmap.c, 31, 定义为预处理宏
- FIRST\_LDT\_ENTRY  
include/sched.h, 154, 定义为预处理宏
- FIRST\_TASK  
include/sched.h, 7, 定义为预处理宏
- FIRST\_TSS\_ENTRY  
include/sched.h, 153, 定义为预处理宏
- flock  
include/fcntl.h, 43, 定义为struct类型
- floppy  
kernel/blk\_drv/floppy.c, 114, 定义为变量
- floppy\_change  
include/fs.h, 169, 定义为函数原型
- kernel/blk\_drv/floppy.c, 139, 定义为函数
- floppy\_deselect  
include/fdreg.h, 13, 定义为函数原型
- kernel/blk\_drv/floppy.c, 125, 定义为函数
- floppy\_init  
init/main.c, 49, 定义为函数原型

kernel/blk\_drv/floppy.c, 457, 定义为函数  
floppy\_interrupt  
kernel/blk\_drv/floppy.c, 104, 定义为函数原型  
floppy\_off  
include/fs.h, 172, 定义为函数原型  
include/fdreg.h, 11, 定义为函数原型  
kernel/sched.c, 240, 定义为函数  
floppy\_on  
include/fs.h, 171, 定义为函数原型  
include/fdreg.h, 10, 定义为函数原型  
kernel/sched.c, 232, 定义为函数  
floppy\_on\_interrupt  
kernel/blk\_drv/floppy.c, 404, 定义为函数  
floppy\_select  
include/fdreg.h, 12, 定义为函数原型  
floppy\_struct  
kernel/blk\_drv/floppy.c, 82, 定义为struct类型  
floppy\_type  
kernel/blk\_drv/floppy.c, 85, 定义为变量  
flush  
kernel/chr\_drv/tty\_ioctl.c, 39, 定义为函数  
FLUSHO  
include/termios.h, 181, 定义为预处理宏  
fn\_ptr  
include/sched.h, 38, 定义为类型  
fork  
include/unistd.h, 210, 定义为函数原型  
free  
include/kernel.h, 12, 定义为预处理宏  
free\_block  
fs/bitmap.c, 47, 定义为函数  
include/fs.h, 193, 定义为函数原型  
free\_bucket\_desc  
lib/malloc.c, 92, 定义为变量  
free\_dind  
fs/truncate.c, 29, 定义为函数  
free\_ind  
fs/truncate.c, 11, 定义为函数  
free\_inode  
fs/bitmap.c, 107, 定义为函数  
include/fs.h, 195, 定义为函数原型  
free\_list  
fs/buffer.c, 32, 定义为变量  
free\_page  
include/mm.h, 8, 定义为函数原型  
mm/memory.c, 89, 定义为函数  
free\_page\_tables  
include/sched.h, 30, 定义为函数原型  
mm/memory.c, 105, 定义为函数  
free\_s  
include/kernel.h, 10, 定义为函数原型  
lib/malloc.c, 182, 定义为函数  
free\_super  
fs/super.c, 40, 定义为函数  
fstat  
include/sys/stat.h, 52, 定义为函数原型  
include/unistd.h, 233, 定义为函数原型  
FULL  
include/tty.h, 29, 定义为预处理宏  
GCC\_HEADER  
tools/build.c, 33, 定义为预处理宏  
gdt  
include/head.h, 9, 定义为变量  
GDT\_CODE  
include/head.h, 12, 定义为预处理宏  
GDT\_DATA  
include/head.h, 13, 定义为预处理宏  
GDT\_NUL  
include/head.h, 11, 定义为预处理宏  
GDT\_TMP  
include/head.h, 14, 定义为预处理宏  
general\_protection  
kernel/traps.c, 56, 定义为函数原型  
get\_base  
include/sched.h, 226, 定义为预处理宏  
get\_dir  
fs/namei.c, 228, 定义为函数  
get\_ds  
include/asm/segment.h, 54, 定义为函数  
get\_empty\_inode  
fs/inode.c, 194, 定义为函数  
include/fs.h, 183, 定义为函数原型  
get\_empty\_page  
mm/memory.c, 274, 定义为函数  
get\_free\_page  
include/mm.h, 6, 定义为函数原型  
mm/memory.c, 63, 定义为函数  
get\_fs

- include/asm/segment.h, 47, 定义为函数  
get\_fs\_byte  
include/asm/segment.h, 1, 定义为函数  
get\_fs\_long  
include/asm/segment.h, 17, 定义为函数  
get\_fs\_word  
include/asm/segment.h, 9, 定义为函数  
get\_hash\_table  
fs/buffer.c, 183, 定义为函数  
include/fs.h, 185, 定义为函数原型  
get\_limit  
include/sched.h, 228, 定义为预处理宏  
get\_new  
kernel/signal.c, 40, 定义为函数  
get\_pipe\_inode  
fs/inode.c, 228, 定义为函数  
include/fs.h, 184, 定义为函数原型  
get\_seg\_byte  
kernel/traps.c, 22, 定义为预处理宏  
get\_seg\_long  
kernel/traps.c, 28, 定义为预处理宏  
get\_super  
fs/super.c, 56, 定义为函数  
include/fs.h, 197, 定义为函数原型  
get\_termio  
kernel/chr\_drv/tty\_ioctl.c, 76, 定义为函数  
get\_termios  
kernel/chr\_drv/tty\_ioctl.c, 56, 定义为函数  
getblk  
fs/buffer.c, 206, 定义为函数  
include/fs.h, 186, 定义为函数原型  
GETCH  
include/tty.h, 31, 定义为预处理宏  
getegid  
include/unistd.h, 215, 定义为函数原型  
geteuid  
include/unistd.h, 213, 定义为函数原型  
getgid  
include/unistd.h, 214, 定义为函数原型  
getpgrp  
include/unistd.h, 250, 定义为函数原型  
getpid  
include/unistd.h, 211, 定义为函数原型  
getppid  
include/unistd.h, 249, 定义为函数原型  
getuid  
include/unistd.h, 212, 定义为函数原型  
gid\_t  
include/sys/types.h, 25, 定义为类型  
gmtime  
include/time.h, 37, 定义为函数原型  
gotoxy  
kernel/chr\_drv/console.c, 88, 定义为函数  
hash  
fs/buffer.c, 129, 定义为预处理宏  
hash\_table  
fs/buffer.c, 31, 定义为变量  
hd  
kernel/blk\_drv/hd.c, 59, 定义为变量  
HD\_CMD  
include/hdreg.h, 21, 定义为预处理宏  
HD\_COMMAND  
include/hdreg.h, 19, 定义为预处理宏  
HD\_CURRENT  
include/hdreg.h, 16, 定义为预处理宏  
HD\_DATA  
include/hdreg.h, 10, 定义为预处理宏  
HD\_ERROR  
include/hdreg.h, 11, 定义为预处理宏  
HD\_HCYL  
include/hdreg.h, 15, 定义为预处理宏  
hd\_i\_struct  
kernel/blk\_drv/hd.c, 45, 定义为struct类型  
hd\_info  
kernel/blk\_drv/hd.c, 49, 定义为struct类型  
kernel/blk\_drv/hd.c, 52, 定义为struct类型  
hd\_init  
init/main.c, 48, 定义为函数原型  
kernel/blk\_drv/hd.c, 343, 定义为函数  
hd\_interrupt  
kernel/blk\_drv/hd.c, 67, 定义为函数原型  
HD\_LCYL  
include/hdreg.h, 14, 定义为预处理宏  
HD\_NSECTOR  
include/hdreg.h, 12, 定义为预处理宏  
hd\_out  
kernel/blk\_drv/hd.c, 180, 定义为函数  
HD\_PRECOMP

- include/hdreg.h, 18, 定义为预处理宏  
HD\_SECTOR  
include/hdreg.h, 13, 定义为预处理宏  
HD\_STATUS  
include/hdreg.h, 17, 定义为预处理宏  
hd\_struct  
kernel/blk\_drv/hd.c, 56, 定义为struct类型  
head  
kernel/blk\_drv/floppy.c, 117, 定义为变量  
HIGH\_MEMORY  
mm/memory.c, 52, 定义为变量  
HOUR  
kernel/mktime.c, 21, 定义为预处理宏  
HUPCL  
include/termios.h, 160, 定义为预处理宏  
HZ  
include/sched.h, 5, 定义为预处理宏  
I\_BLOCK\_SPECIAL  
include/const.h, 9, 定义为预处理宏  
I\_CHAR\_SPECIAL  
include/const.h, 10, 定义为预处理宏  
I\_CRNL  
kernel/chr\_drv/tty\_io.c, 42, 定义为预处理宏  
I\_DIRECTORY  
include/const.h, 7, 定义为预处理宏  
I\_MAP\_SLOTS  
include/fs.h, 39, 定义为预处理宏  
I\_NAMED\_PIPE  
include/const.h, 11, 定义为预处理宏  
I\_NLCR  
kernel/chr\_drv/tty\_io.c, 41, 定义为预处理宏  
I\_NOCR  
kernel/chr\_drv/tty\_io.c, 43, 定义为预处理宏  
I\_REGULAR  
include/const.h, 8, 定义为预处理宏  
I\_SET\_GID\_BIT  
include/const.h, 13, 定义为预处理宏  
I\_SET\_UID\_BIT  
include/const.h, 12, 定义为预处理宏  
I\_TYPE  
include/const.h, 6, 定义为预处理宏  
I\_UCLC  
kernel/chr\_drv/tty\_io.c, 40, 定义为预处理宏  
i387\_struct  
include/sched.h, 40, 定义为struct类型  
ICANON  
include/termios.h, 170, 定义为预处理宏  
ICRNL  
include/termios.h, 91, 定义为预处理宏  
ID\_ERR  
include/hdreg.h, 48, 定义为预处理宏  
idt  
include/head.h, 9, 定义为变量  
IEXTEN  
include/termios.h, 183, 定义为预处理宏  
iget  
fs/inode.c, 244, 定义为函数  
include/fs.h, 182, 定义为函数原型  
IGNBRK  
include/termios.h, 83, 定义为预处理宏  
IGNCR  
include/termios.h, 90, 定义为预处理宏  
IGNPAR  
include/termios.h, 85, 定义为预处理宏  
IMAXBEL  
include/termios.h, 96, 定义为预处理宏  
immoutb\_p  
kernel/blk\_drv/floppy.c, 50, 定义为预处理宏  
IN\_ORDER  
kernel/blk\_drv/blk.h, 40, 定义为预处理宏  
inb  
include/asm/io.h, 5, 定义为预处理宏  
inb\_p  
include/asm/io.h, 17, 定义为预处理宏  
INC  
include/tty.h, 24, 定义为预处理宏  
INC\_PIPE  
include/fs.h, 63, 定义为预处理宏  
INDEX\_STAT  
include/hdreg.h, 25, 定义为预处理宏  
init  
init/main.c, 45, 定义为函数原型  
init/main.c, 168, 定义为函数  
kernel/chr\_drv/serial.c, 26, 定义为函数  
init\_bucket\_desc  
lib/malloc.c, 97, 定义为函数  
INIT\_C\_CC  
include/tty.h, 63, 定义为预处理宏

INIT\_REQUEST  
kernel/blk\_drv/blk.h, 127, 定义为预处理宏  
INIT\_TASK  
include/sched.h, 113, 定义为预处理宏  
init\_task  
kernel/sched.c, 58, 定义为union类型  
INLCR  
include/termios.h, 89, 定义为预处理宏  
ino\_t  
include/sys/types.h, 27, 定义为类型  
inode\_table  
fs/inode.c, 15, 定义为变量  
include/fs.h, 162, 定义为变量  
INODES\_PER\_BLOCK  
include/fs.h, 55, 定义为预处理宏  
INPCK  
include/termios.h, 87, 定义为预处理宏  
insert\_char  
kernel/chr\_drv/console.c, 336, 定义为函数  
insert\_into\_queues  
fs/buffer.c, 149, 定义为函数  
insert\_line  
kernel/chr\_drv/console.c, 350, 定义为函数  
int3  
kernel/traps.c, 46, 定义为函数原型  
interruptible\_sleep\_on  
include/sched.h, 146, 定义为函数原型  
kernel/sched.c, 167, 定义为函数  
INTMASK  
kernel/chr\_drv/tty\_io.c, 19, 定义为预处理宏  
INTR\_CHAR  
include/tty.h, 36, 定义为预处理宏  
invalid\_op  
kernel/traps.c, 49, 定义为函数原型  
invalid\_TSS  
kernel/traps.c, 53, 定义为函数原型  
invalidate  
mm/memory.c, 39, 定义为预处理宏  
invalidate\_buffers  
fs/buffer.c, 84, 定义为函数  
invalidate\_inodes  
fs/inode.c, 43, 定义为函数  
ioctl  
include/unistd.h, 216, 定义为函数原型  
ioctl\_ptr  
fs/ioctl.c, 15, 定义为类型  
ioctl\_table  
fs/ioctl.c, 19, 定义为变量  
iput  
fs/inode.c, 150, 定义为函数  
include/fs.h, 181, 定义为函数原型  
iret  
include/asm/system.h, 20, 定义为预处理宏  
irq13  
kernel/traps.c, 61, 定义为函数原型  
is\_digit  
kernel/vsprintf.c, 16, 定义为预处理宏  
IS\_SEEKABLE  
include/fs.h, 24, 定义为预处理宏  
isalnum  
include/ctype.h, 16, 定义为预处理宏  
isalpha  
include/ctype.h, 17, 定义为预处理宏  
isascii  
include/ctype.h, 28, 定义为预处理宏  
iscntrl  
include/ctype.h, 18, 定义为预处理宏  
isdigit  
include/ctype.h, 19, 定义为预处理宏  
isgraph  
include/ctype.h, 20, 定义为预处理宏  
ISIG  
include/termios.h, 169, 定义为预处理宏  
islower  
include/ctype.h, 21, 定义为预处理宏  
isprint  
include/ctype.h, 22, 定义为预处理宏  
ispunct  
include/ctype.h, 23, 定义为预处理宏  
isspace  
include/ctype.h, 24, 定义为预处理宏  
ISTRIP  
include/termios.h, 88, 定义为预处理宏  
isupper  
include/ctype.h, 25, 定义为预处理宏  
isxdigit  
include/ctype.h, 26, 定义为预处理宏  
IUCLC

- include/termios.h, 92, 定义为预处理宏  
IXANY  
include/termios.h, 94, 定义为预处理宏  
IXOFF  
include/termios.h, 95, 定义为预处理宏  
IXON  
include/termios.h, 93, 定义为预处理宏  
jiffies  
include/sched.h, 139, 定义为变量  
kernel/sched.c, 60, 定义为变量  
KBD\_FINNISH  
include/config.h, 19, 定义为预处理宏  
kernel\_mktime  
init/main.c, 52, 定义为函数原型  
kernel/mktime.c, 41, 定义为函数  
keyboard\_interrupt  
kernel/chr\_drv/console.c, 56, 定义为函数原型  
kill  
include/unistd.h, 217, 定义为函数原型  
include/signal.h, 57, 定义为函数原型  
KILL\_CHAR  
include/tty.h, 39, 定义为预处理宏  
kill\_session  
kernel/exit.c, 46, 定义为函数  
KILLMASK  
kernel/chr\_drv/tty\_io.c, 18, 定义为预处理宏  
L\_CANON  
kernel/chr\_drv/tty\_io.c, 32, 定义为预处理宏  
L\_ECHO  
kernel/chr\_drv/tty\_io.c, 34, 定义为预处理宏  
L\_ECHOCTL  
kernel/chr\_drv/tty\_io.c, 37, 定义为预处理宏  
L\_ECHOE  
kernel/chr\_drv/tty\_io.c, 35, 定义为预处理宏  
L\_ECHOK  
kernel/chr\_drv/tty\_io.c, 36, 定义为预处理宏  
L\_ECHOKE  
kernel/chr\_drv/tty\_io.c, 38, 定义为预处理宏  
L\_ISIG  
kernel/chr\_drv/tty\_io.c, 33, 定义为预处理宏  
LAST  
include/tty.h, 28, 定义为预处理宏  
last\_pid  
kernel/fork.c, 22, 定义为变量  
LAST\_TASK  
include/sched.h, 8, 定义为预处理宏  
last\_task\_used\_math  
include/sched.h, 137, 定义为变量  
kernel/sched.c, 63, 定义为变量  
LATCH  
kernel/sched.c, 46, 定义为预处理宏  
ldiv\_t  
include/sys/types.h, 37, 定义为类型  
LDT\_CODE  
include/head.h, 17, 定义为预处理宏  
LDT\_DATA  
include/head.h, 18, 定义为预处理宏  
LDT\_NUL  
include/head.h, 16, 定义为预处理宏  
LEFT  
include/tty.h, 27, 定义为预处理宏  
kernel/vsprintf.c, 31, 定义为预处理宏  
lf  
kernel/chr\_drv/console.c, 204, 定义为函数  
link  
include/unistd.h, 218, 定义为函数原型  
ll\_rw\_block  
include/fs.h, 187, 定义为函数原型  
kernel/blk\_drv/ll\_rw\_blk.c, 145, 定义为函数  
lldt  
include/sched.h, 158, 定义为预处理宏  
localtime  
include/time.h, 38, 定义为函数原型  
lock\_buffer  
kernel/blk\_drv/ll\_rw\_blk.c, 42, 定义为函数  
lock\_inode  
fs/inode.c, 28, 定义为函数  
lock\_super  
fs/super.c, 31, 定义为函数  
LOW\_MEM  
mm/memory.c, 43, 定义为预处理宏  
lseek  
include/unistd.h, 219, 定义为函数原型  
ltr  
include/sched.h, 157, 定义为预处理宏  
m\_inode  
include/fs.h, 93, 定义为struct类型  
main



- init/main.c, 104, 定义为函数  
tools/build.c, 57, 定义为函数  
main\_memory\_start  
init/main.c, 100, 定义为变量  
MAJOR  
include/fs.h, 33, 定义为预处理宏  
MAJOR\_NR  
kernel/blk\_drv/hd.c, 25, 定义为预处理宏  
kernel/blk\_drv/floppy.c, 41, 定义为预处理宏  
kernel/blk\_drv/ramdisk.c, 17, 定义为预处理宏  
make\_request  
kernel/blk\_drv/l1\_rw\_blk.c, 88, 定义为函数  
malloc  
include/kernel.h, 9, 定义为函数原型  
lib/malloc.c, 117, 定义为函数  
MAP\_NR  
mm/memory.c, 46, 定义为预处理宏  
MARK\_ERR  
include/hdreg.h, 45, 定义为预处理宏  
match  
fs/namei.c, 63, 定义为函数  
math\_emulate  
kernel/math/math\_emulate.c, 18, 定义为函数  
math\_error  
kernel/math/math\_emulate.c, 37, 定义为函数  
math\_state\_restore  
kernel/sched.c, 77, 定义为函数  
MAX  
fs/file\_dev.c, 15, 定义为预处理宏  
MAX\_ARG\_PAGES  
fs/exec.c, 39, 定义为预处理宏  
MAX\_ERRORS  
kernel/blk\_drv/hd.c, 34, 定义为预处理宏  
kernel/blk\_drv/floppy.c, 60, 定义为预处理宏  
MAX\_HD  
kernel/blk\_drv/hd.c, 35, 定义为预处理宏  
MAX\_REPLIES  
kernel/blk\_drv/floppy.c, 65, 定义为预处理宏  
MAY\_EXEC  
fs/namei.c, 29, 定义为预处理宏  
MAY\_READ  
fs/namei.c, 31, 定义为预处理宏  
MAY\_WRITE  
fs/namei.c, 30, 定义为预处理宏  
mem\_init  
init/main.c, 50, 定义为函数原型  
mm/memory.c, 399, 定义为函数  
mem\_map  
mm/memory.c, 57, 定义为变量  
mem\_use  
kernel/sched.c, 48, 定义为函数原型  
memchr  
include/string.h, 379, 定义为函数  
memcmp  
include/string.h, 363, 定义为函数  
memcpy  
include/string.h, 336, 定义为函数  
include/asm/memory.h, 8, 定义为预处理宏  
memmove  
include/string.h, 346, 定义为函数  
memory\_end  
init/main.c, 98, 定义为变量  
memset  
include/string.h, 395, 定义为函数  
MIN  
fs/file\_dev.c, 14, 定义为预处理宏  
MINIX\_HEADER  
tools/build.c, 32, 定义为预处理宏  
MINOR  
include/fs.h, 34, 定义为预处理宏  
MINUTE  
kernel/mktime.c, 20, 定义为预处理宏  
mkdir  
include/sys/stat.h, 53, 定义为函数原型  
mkfifo  
include/sys/stat.h, 54, 定义为函数原型  
mknod  
include/unistd.h, 220, 定义为函数原型  
mktime  
include/time.h, 33, 定义为函数原型  
mode\_t  
include/sys/types.h, 28, 定义为类型  
moff\_timer  
kernel/sched.c, 203, 定义为变量  
mon\_timer  
kernel/sched.c, 202, 定义为变量  
month  
kernel/mktime.c, 26, 定义为变量

- mount  
include/unistd.h, 221, 定义为函数原型  
mount\_root  
fs/super.c, 242, 定义为函数  
include/fs.h, 200, 定义为函数原型  
move\_to\_user\_mode  
include/asm/system.h, 1, 定义为预处理宏  
N\_ABS  
include/a.out.h, 128, 定义为预处理宏  
N\_BADMAG  
include/a.out.h, 31, 定义为预处理宏  
N\_BSS  
include/a.out.h, 137, 定义为预处理宏  
N\_BSSADDR  
include/a.out.h, 107, 定义为预处理宏  
N\_COMM  
include/a.out.h, 140, 定义为预处理宏  
N\_DATA  
include/a.out.h, 134, 定义为预处理宏  
N\_DATADDR  
include/a.out.h, 100, 定义为预处理宏  
N\_DATOFF  
include/a.out.h, 48, 定义为预处理宏  
N\_DRELOFF  
include/a.out.h, 56, 定义为预处理宏  
N\_EXT  
include/a.out.h, 147, 定义为预处理宏  
N\_FN  
include/a.out.h, 143, 定义为预处理宏  
N\_INDR  
include/a.out.h, 164, 定义为预处理宏  
N\_MAGIC  
include/a.out.h, 18, 定义为预处理宏  
N\_SETA  
include/a.out.h, 178, 定义为预处理宏  
N\_SETB  
include/a.out.h, 181, 定义为预处理宏  
N\_SETD  
include/a.out.h, 180, 定义为预处理宏  
N\_SETT  
include/a.out.h, 179, 定义为预处理宏  
N\_SETV  
include/a.out.h, 184, 定义为预处理宏  
N\_STAB  
include/a.out.h, 153, 定义为预处理宏  
N\_STROFF  
include/a.out.h, 64, 定义为预处理宏  
N\_SYMOFF  
include/a.out.h, 60, 定义为预处理宏  
N\_TEXT  
include/a.out.h, 131, 定义为预处理宏  
N\_TRELOFF  
include/a.out.h, 52, 定义为预处理宏  
N\_TXTADDR  
include/a.out.h, 69, 定义为预处理宏  
N\_TXTOFF  
include/a.out.h, 43, 定义为预处理宏  
N\_TYPE  
include/a.out.h, 150, 定义为预处理宏  
N\_UNDF  
include/a.out.h, 125, 定义为预处理宏  
NAME\_LEN  
include/fs.h, 36, 定义为预处理宏  
namei  
fs/namei.c, 303, 定义为函数  
include/fs.h, 178, 定义为函数原型  
NCC  
include/termios.h, 43, 定义为预处理宏  
NCCS  
include/termios.h, 53, 定义为预处理宏  
new\_block  
fs/bitmap.c, 75, 定义为函数  
include/fs.h, 192, 定义为函数原型  
new\_inode  
fs/bitmap.c, 136, 定义为函数  
include/fs.h, 194, 定义为函数原型  
next\_timer  
kernel/sched.c, 270, 定义为变量  
nice  
include/unistd.h, 222, 定义为函数原型  
NLO  
include/termios.h, 108, 定义为预处理宏  
NL1  
include/termios.h, 109, 定义为预处理宏  
NLDLY  
include/termios.h, 107, 定义为预处理宏  
nlink\_t  
include/sys/types.h, 30, 定义为类型

nlist  
include/a.out.h, 111, 定义为struct类型  
NMAGIC  
include/a.out.h, 25, 定义为预处理宏  
nmi  
kernel/traps.c, 45, 定义为函数原型  
NOFLSH  
include/termios.h, 176, 定义为预处理宏  
nop  
include/asm/system.h, 18, 定义为预处理宏  
NPAR  
kernel/chr\_drv/console.c, 54, 定义为预处理宏  
npar  
kernel/chr\_drv/console.c, 75, 定义为变量  
NR\_BLK\_DEV  
kernel/blk\_drv/blk.h, 4, 定义为预处理宏  
NR\_BUFFERS  
fs/buffer.c, 34, 定义为变量  
include/fs.h, 48, 定义为预处理宏  
nr\_buffers  
include/fs.h, 166, 定义为变量  
NR\_FILE  
include/fs.h, 45, 定义为预处理宏  
NR\_HASH  
include/fs.h, 47, 定义为预处理宏  
NR\_HD  
kernel/blk\_drv/hd.c, 50, 定义为预处理宏  
kernel/blk\_drv/hd.c, 53, 定义为变量  
NR\_INODE  
include/fs.h, 44, 定义为预处理宏  
NR\_OPEN  
include/fs.h, 43, 定义为预处理宏  
NR\_REQUEST  
kernel/blk\_drv/blk.h, 15, 定义为预处理宏  
NR\_SUPER  
include/fs.h, 46, 定义为预处理宏  
NR\_TASKS  
include/sched.h, 4, 定义为预处理宏  
NRDEVS  
fs/char\_dev.c, 83, 定义为预处理宏  
fs/ioctl.c, 17, 定义为预处理宏  
NSIG  
include/signal.h, 10, 定义为预处理宏  
NULL  
include/sys/types.h, 20, 定义为预处理宏  
include/unistd.h, 18, 定义为预处理宏  
include/stddef.h, 14, 定义为预处理宏  
include/stddef.h, 15, 定义为预处理宏  
include/string.h, 5, 定义为预处理宏  
include/sched.h, 26, 定义为预处理宏  
include/fs.h, 52, 定义为预处理宏  
number  
kernel/vsprintf.c, 40, 定义为函数  
O\_ACCMODE  
include/fcntl.h, 7, 定义为预处理宏  
O\_APPEND  
include/fcntl.h, 15, 定义为预处理宏  
O\_CREAT  
include/fcntl.h, 11, 定义为预处理宏  
O\_CRNL  
kernel/chr\_drv/tty\_io.c, 47, 定义为预处理宏  
O\_EXCL  
include/fcntl.h, 12, 定义为预处理宏  
O\_LCUC  
kernel/chr\_drv/tty\_io.c, 49, 定义为预处理宏  
O\_NDELAY  
include/fcntl.h, 17, 定义为预处理宏  
O\_NLCR  
kernel/chr\_drv/tty\_io.c, 46, 定义为预处理宏  
O\_NLRET  
kernel/chr\_drv/tty\_io.c, 48, 定义为预处理宏  
O\_NOCTTY  
include/fcntl.h, 13, 定义为预处理宏  
O\_NONBLOCK  
include/fcntl.h, 16, 定义为预处理宏  
O\_POST  
kernel/chr\_drv/tty\_io.c, 45, 定义为预处理宏  
O\_RDONLY  
include/fcntl.h, 8, 定义为预处理宏  
O\_RDWR  
include/fcntl.h, 10, 定义为预处理宏  
O\_TRUNC  
include/fcntl.h, 14, 定义为预处理宏  
O\_WRONLY  
include/fcntl.h, 9, 定义为预处理宏  
OCRNL  
include/termios.h, 102, 定义为预处理宏  
OFDEL

- include/termios.h, 106, 定义为预处理宏
- off\_t
- include/sys/types.h, 32, 定义为类型
- offsetof
- include/stddef.h, 17, 定义为预处理宏
- OFILL
- include/termios.h, 105, 定义为预处理宏
- OLCUC
- include/termios.h, 100, 定义为预处理宏
- OMAGIC
- include/a.out.h, 23, 定义为预处理宏
- ONLCR
- include/termios.h, 101, 定义为预处理宏
- ONLRET
- include/termios.h, 104, 定义为预处理宏
- ONOCR
- include/termios.h, 103, 定义为预处理宏
- oom
- mm/memory.c, 33, 定义为函数
- open
- include/unistd.h, 223, 定义为函数原型
- include/fcntl.h, 53, 定义为函数原型
- lib/open.c, 11, 定义为函数
- open\_namei
- fs/namei.c, 337, 定义为函数
- include/fs.h, 179, 定义为函数原型
- OPOST
- include/termios.h, 99, 定义为预处理宏
- ORIG\_ROOT\_DEV
- init/main.c, 60, 定义为预处理宏
- ORIG\_VIDEO\_COLS
- kernel/chr\_drv/console.c, 43, 定义为预处理宏
- ORIG\_VIDEO\_EGA\_AX
- kernel/chr\_drv/console.c, 45, 定义为预处理宏
- ORIG\_VIDEO\_EGA\_BX
- kernel/chr\_drv/console.c, 46, 定义为预处理宏
- ORIG\_VIDEO\_EGA\_CX
- kernel/chr\_drv/console.c, 47, 定义为预处理宏
- ORIG\_VIDEO\_LINES
- kernel/chr\_drv/console.c, 44, 定义为预处理宏
- ORIG\_VIDEO\_MODE
- kernel/chr\_drv/console.c, 42, 定义为预处理宏
- ORIG\_VIDEO\_PAGE
- kernel/chr\_drv/console.c, 41, 定义为预处理宏
- ORIG\_X
- kernel/chr\_drv/console.c, 39, 定义为预处理宏
- ORIG\_Y
- kernel/chr\_drv/console.c, 40, 定义为预处理宏
- origin
- kernel/chr\_drv/console.c, 69, 定义为变量
- outb
- include/asm/io.h, 1, 定义为预处理宏
- outb\_p
- include/asm/io.h, 11, 定义为预处理宏
- output\_byte
- kernel/blk\_drv/floppy.c, 194, 定义为函数
- overflow
- kernel/traps.c, 47, 定义为函数原型
- PAGE\_ALIGN
- include/sched.h, 186, 定义为预处理宏
- page\_exception
- kernel/traps.c, 41, 定义为函数原型
- page\_fault
- kernel/traps.c, 57, 定义为函数原型
- PAGE\_SIZE
- include/a.out.h, 79, 定义为预处理宏
- include/a.out.h, 88, 定义为预处理宏
- include/a.out.h, 92, 定义为预处理宏
- include/mm.h, 4, 定义为预处理宏
- PAGING\_MEMORY
- mm/memory.c, 44, 定义为预处理宏
- PAGING\_PAGES
- mm/memory.c, 45, 定义为预处理宏
- panic
- include/kernel.h, 5, 定义为函数原型
- include/sched.h, 35, 定义为函数原型
- kernel/panic.c, 16, 定义为函数
- par
- kernel/chr\_drv/console.c, 75, 定义为变量
- parallel\_interrupt
- kernel/traps.c, 60, 定义为函数原型
- PARENB
- include/termios.h, 165, 定义为预处理宏
- PARMRK
- include/termios.h, 86, 定义为预处理宏
- PARODD
- include/termios.h, 166, 定义为预处理宏
- partition

- include/hdreg.h, 52, 定义为struct类型
- pause
- include/unistd.h, 224, 定义为函数原型
- PENDIN
- include/termios.h, 182, 定义为预处理宏
- permission
- fs/namei.c, 40, 定义为函数
- pg\_dir
- include/head.h, 8, 定义为变量
- pid\_t
- include/sys/types.h, 23, 定义为类型
- pipe
- include/unistd.h, 225, 定义为函数原型
- PIPE\_EMPTY
- include/fs.h, 61, 定义为预处理宏
- PIPE\_FULL
- include/fs.h, 62, 定义为预处理宏
- PIPE\_HEAD
- include/fs.h, 58, 定义为预处理宏
- PIPE\_SIZE
- include/fs.h, 60, 定义为预处理宏
- PIPE\_TAIL
- include/fs.h, 59, 定义为预处理宏
- PLUS
- kernel/vsprintf.c, 29, 定义为预处理宏
- port\_read
- kernel/blk\_drv/hd.c, 61, 定义为预处理宏
- port\_write
- kernel/blk\_drv/hd.c, 64, 定义为预处理宏
- pos
- kernel/chr\_drv/console.c, 71, 定义为变量
- printbuf
- init/main.c, 42, 定义为变量
- printf
- include/kernel.h, 6, 定义为函数原型
- init/main.c, 151, 定义为函数
- printk
- include/kernel.h, 7, 定义为函数原型
- kernel/printk.c, 21, 定义为函数
- ptrdiff\_t
- include/sys/types.h, 16, 定义为类型
- include/stddef.h, 6, 定义为类型
- put\_fs\_byte
- include/asm/segment.h, 25, 定义为函数
- put\_fs\_long
- include/asm/segment.h, 35, 定义为函数
- put\_fs\_word
- include/asm/segment.h, 30, 定义为函数
- put\_page
- include/mm.h, 7, 定义为函数原型
- mm/memory.c, 197, 定义为函数
- put\_super
- fs/super.c, 74, 定义为函数
- PUTCH
- include/tty.h, 33, 定义为预处理宏
- ques
- kernel/chr\_drv/console.c, 76, 定义为变量
- QUIT\_CHAR
- include/tty.h, 37, 定义为预处理宏
- QUITMASK
- kernel/chr\_drv/tty\_io.c, 20, 定义为预处理宏
- quotient
- kernel/chr\_drv/tty\_ioctl.c, 18, 定义为变量
- R\_OK
- include/unistd.h, 25, 定义为预处理宏
- raise
- include/signal.h, 56, 定义为函数原型
- rd\_init
- init/main.c, 51, 定义为函数原型
- kernel/blk\_drv/ramdisk.c, 52, 定义为函数
- rd\_length
- kernel/blk\_drv/ramdisk.c, 21, 定义为变量
- rd\_load
- kernel/blk\_drv/hd.c, 68, 定义为函数原型
- kernel/blk\_drv/ramdisk.c, 71, 定义为函数
- rd\_start
- kernel/blk\_drv/ramdisk.c, 20, 定义为变量
- read
- include/unistd.h, 226, 定义为函数原型
- READ
- include/fs.h, 26, 定义为预处理宏
- read\_inode
- fs/inode.c, 17, 定义为函数原型
- fs/inode.c, 294, 定义为函数
- read\_intr
- kernel/blk\_drv/hd.c, 250, 定义为函数
- read\_pipe
- fs/read\_write.c, 16, 定义为函数原型

- fs/pipe.c, 13, 定义为函数
- read\_super
- fs/super.c, 100, 定义为函数
- READA
- include/fs.h, 28, 定义为预处理宏
- READY\_STAT
- include/hdreg.h, 30, 定义为预处理宏
- recal\_interrupt
- kernel/blk\_drv/floppy.c, 343, 定义为函数
- recal\_intr
- kernel/blk\_drv/hd.c, 37, 定义为函数原型
- kernel/blk\_drv/hd.c, 287, 定义为函数
- recalibrate
- kernel/blk\_drv/hd.c, 39, 定义为变量
- kernel/blk\_drv/floppy.c, 44, 定义为变量
- recalibrate\_floppy
- kernel/blk\_drv/floppy.c, 362, 定义为函数
- release
- kernel/exit.c, 19, 定义为函数
- relocation\_info
- include/a.out.h, 193, 定义为struct类型
- remove\_from\_queues
- fs/buffer.c, 131, 定义为函数
- reply\_buffer
- kernel/blk\_drv/floppy.c, 66, 定义为变量
- request
- kernel/blk\_drv/ll\_rw\_blk.c, 21, 定义为变量
- kernel/blk\_drv/blk.h, 23, 定义为struct类型
- kernel/blk\_drv/blk.h, 51, 定义为变量
- reserved
- kernel/traps.c, 59, 定义为函数原型
- reset
- kernel/blk\_drv/hd.c, 40, 定义为变量
- kernel/blk\_drv/floppy.c, 45, 定义为变量
- reset\_controllor
- kernel/blk\_drv/hd.c, 217, 定义为函数
- reset\_floppy
- kernel/blk\_drv/floppy.c, 386, 定义为函数
- reset\_hd
- kernel/blk\_drv/hd.c, 230, 定义为函数
- reset\_interrupt
- kernel/blk\_drv/floppy.c, 373, 定义为函数
- respond
- kernel/chr\_drv/console.c, 323, 定义为函数
- RESPONSE
- kernel/chr\_drv/console.c, 85, 定义为预处理宏
- restore\_cur
- kernel/chr\_drv/console.c, 440, 定义为函数
- result
- kernel/blk\_drv/floppy.c, 212, 定义为函数
- ri
- kernel/chr\_drv/console.c, 214, 定义为函数
- ROOT\_DEV
- fs/super.c, 29, 定义为变量
- include/fs.h, 198, 定义为变量
- ROOT\_INO
- include/fs.h, 37, 定义为预处理宏
- rs\_init
- include/tty.h, 65, 定义为函数原型
- kernel/chr\_drv/serial.c, 37, 定义为函数
- rs\_write
- include/tty.h, 72, 定义为函数原型
- kernel/chr\_drv/serial.c, 53, 定义为函数
- rsl\_interrupt
- kernel/chr\_drv/serial.c, 23, 定义为函数原型
- rs2\_interrupt
- kernel/chr\_drv/serial.c, 24, 定义为函数原型
- rw\_char
- fs/read\_write.c, 15, 定义为函数原型
- fs/char\_dev.c, 95, 定义为函数
- rw\_interrupt
- kernel/blk\_drv/floppy.c, 250, 定义为函数
- rw\_kmem
- fs/char\_dev.c, 44, 定义为函数
- rw\_mem
- fs/char\_dev.c, 39, 定义为函数
- rw\_memory
- fs/char\_dev.c, 65, 定义为函数
- rw\_port
- fs/char\_dev.c, 49, 定义为函数
- rw\_ram
- fs/char\_dev.c, 34, 定义为函数
- rw\_tty
- fs/char\_dev.c, 27, 定义为函数
- rw\_ttyx
- fs/char\_dev.c, 21, 定义为函数
- S\_IFBLK
- include/sys/stat.h, 22, 定义为预处理宏

- S\_IFCHR  
include/sys/stat.h, 24, 定义为预处理宏
- S\_IFDIR  
include/sys/stat.h, 23, 定义为预处理宏
- S\_IFIFO  
include/sys/stat.h, 25, 定义为预处理宏
- S\_IFMT  
include/sys/stat.h, 20, 定义为预处理宏
- S\_IFREG  
include/sys/stat.h, 21, 定义为预处理宏
- S\_IRGRP  
include/sys/stat.h, 42, 定义为预处理宏
- S\_IROTH  
include/sys/stat.h, 47, 定义为预处理宏
- S\_IRUSR  
include/sys/stat.h, 37, 定义为预处理宏
- S\_IRWXG  
include/sys/stat.h, 41, 定义为预处理宏
- S\_IRWXO  
include/sys/stat.h, 46, 定义为预处理宏
- S\_IRWXU  
include/sys/stat.h, 36, 定义为预处理宏
- S\_ISBLK  
include/sys/stat.h, 33, 定义为预处理宏
- S\_ISCHR  
include/sys/stat.h, 32, 定义为预处理宏
- S\_ISDIR  
include/sys/stat.h, 31, 定义为预处理宏
- S\_ISFIFO  
include/sys/stat.h, 34, 定义为预处理宏
- S\_ISGID  
include/sys/stat.h, 27, 定义为预处理宏
- S\_ISREG  
include/sys/stat.h, 30, 定义为预处理宏
- S\_ISUID  
include/sys/stat.h, 26, 定义为预处理宏
- S\_ISVTX  
include/sys/stat.h, 28, 定义为预处理宏
- S\_IWGRP  
include/sys/stat.h, 43, 定义为预处理宏
- S\_IWOTH  
include/sys/stat.h, 48, 定义为预处理宏
- S\_IWUSR  
include/sys/stat.h, 38, 定义为预处理宏
- S\_IXGRP  
include/sys/stat.h, 44, 定义为预处理宏
- S\_IXOTH  
include/sys/stat.h, 49, 定义为预处理宏
- S\_IXUSR  
include/sys/stat.h, 39, 定义为预处理宏
- SA\_NOCLDSTOP  
include/signal.h, 37, 定义为预处理宏
- SA\_NOMASK  
include/signal.h, 38, 定义为预处理宏
- SA\_ONESHOT  
include/signal.h, 39, 定义为预处理宏
- save\_cur  
kernel/chr\_drv/console.c, 434, 定义为函数
- save\_old  
kernel/signal.c, 28, 定义为函数
- saved\_x  
kernel/chr\_drv/console.c, 431, 定义为变量
- saved\_y  
kernel/chr\_drv/console.c, 432, 定义为变量
- sbrk  
include/unistd.h, 193, 定义为函数原型
- sched\_init  
include/sched.h, 32, 定义为函数原型
- kernel/sched.c, 385, 定义为函数
- schedule  
include/sched.h, 33, 定义为函数原型
- kernel/sched.c, 104, 定义为函数
- scr\_end  
kernel/chr\_drv/console.c, 70, 定义为变量
- scrdown  
kernel/chr\_drv/console.c, 170, 定义为函数
- scrup  
kernel/chr\_drv/console.c, 107, 定义为函数
- sector  
kernel/blk\_drv/floppy.c, 116, 定义为变量
- seek  
kernel/blk\_drv/floppy.c, 46, 定义为变量
- SEEK\_CUR  
include/unistd.h, 29, 定义为预处理宏
- SEEK\_END  
include/unistd.h, 30, 定义为预处理宏
- seek\_interrupt  
kernel/blk\_drv/floppy.c, 291, 定义为函数

- SEEK\_SET  
include/unistd.h, 28, 定义为预处理宏
- SEEK\_STAT  
include/hdreg.h, 28, 定义为预处理宏
- seek\_track  
kernel/blk\_drv/floppy.c, 119, 定义为变量
- segment\_not\_present  
kernel/traps.c, 54, 定义为函数原型
- SEGMENT\_SIZE  
include/a.out.h, 76, 定义为预处理宏  
include/a.out.h, 82, 定义为预处理宏  
include/a.out.h, 85, 定义为预处理宏  
include/a.out.h, 89, 定义为预处理宏  
include/a.out.h, 93, 定义为预处理宏
- selected  
kernel/blk\_drv/floppy.c, 122, 定义为变量
- send\_break  
kernel/chr\_drv/tty\_ioctl.c, 51, 定义为函数
- send\_sig  
kernel/exit.c, 35, 定义为函数
- set\_base  
include/sched.h, 211, 定义为预处理宏
- set\_bit  
fs/super.c, 22, 定义为预处理宏  
fs/bitmap.c, 19, 定义为预处理宏
- set\_cursor  
kernel/chr\_drv/console.c, 313, 定义为函数
- set\_fs  
include/asm/segment.h, 61, 定义为函数
- set\_intr\_gate  
include/asm/system.h, 33, 定义为预处理宏
- set\_ldt\_desc  
include/asm/system.h, 66, 定义为预处理宏
- set\_limit  
include/sched.h, 212, 定义为预处理宏
- set\_origin  
kernel/chr\_drv/console.c, 97, 定义为函数
- set\_system\_gate  
include/asm/system.h, 39, 定义为预处理宏
- set\_termio  
kernel/chr\_drv/tty\_ioctl.c, 97, 定义为函数
- set\_termios  
kernel/chr\_drv/tty\_ioctl.c, 66, 定义为函数
- set\_trap\_gate  
include/asm/system.h, 36, 定义为预处理宏
- set\_tss\_desc  
include/asm/system.h, 65, 定义为预处理宏
- setgid  
include/unistd.h, 230, 定义为函数原型
- setpgid  
include/unistd.h, 228, 定义为函数原型
- setpgrp  
include/unistd.h, 227, 定义为函数原型
- setsid  
include/unistd.h, 251, 定义为函数原型
- setuid  
include/unistd.h, 229, 定义为函数原型
- setup\_DMA  
kernel/blk\_drv/floppy.c, 160, 定义为函数
- setup\_rw\_floppy  
kernel/blk\_drv/floppy.c, 269, 定义为函数
- SETUP\_SECTS  
tools/build.c, 42, 定义为预处理宏
- share\_page  
mm/memory.c, 344, 定义为函数
- show\_stat  
kernel/sched.c, 37, 定义为函数
- show\_task  
kernel/sched.c, 26, 定义为函数
- sig\_atomic\_t  
include/signal.h, 6, 定义为类型
- SIG\_BLOCK  
include/signal.h, 41, 定义为预处理宏
- SIG\_DFL  
include/signal.h, 45, 定义为预处理宏
- SIG\_IGN  
include/signal.h, 46, 定义为预处理宏
- SIG\_SETMASK  
include/signal.h, 43, 定义为预处理宏
- SIG\_UNBLOCK  
include/signal.h, 42, 定义为预处理宏
- SIGABRT  
include/signal.h, 17, 定义为预处理宏
- sigaction  
include/signal.h, 48, 定义为struct类型
- include/signal.h, 66, 定义为函数原型
- sigaddset  
include/signal.h, 58, 定义为函数原型



- SIGALRM  
include/signal.h, 26, 定义为预处理宏
- SIGCHLD  
include/signal.h, 29, 定义为预处理宏
- SIGCONT  
include/signal.h, 30, 定义为预处理宏
- sigdelset  
include/signal.h, 59, 定义为函数原型
- sigemptyset  
include/signal.h, 60, 定义为函数原型
- sigfillset  
include/signal.h, 61, 定义为函数原型
- SIGFPE  
include/signal.h, 20, 定义为预处理宏
- SIGHUP  
include/signal.h, 12, 定义为预处理宏
- SIGILL  
include/signal.h, 15, 定义为预处理宏
- SIGINT  
include/signal.h, 13, 定义为预处理宏
- SIGIOT  
include/signal.h, 18, 定义为预处理宏
- sigismember  
include/signal.h, 62, 定义为函数原型
- SIGKILL  
include/signal.h, 21, 定义为预处理宏
- SIGN  
kernel/vsprintf.c, 28, 定义为预处理宏
- sigpending  
include/signal.h, 63, 定义为函数原型
- SIGPIPE  
include/signal.h, 25, 定义为预处理宏
- sigprocmask  
include/signal.h, 64, 定义为函数原型
- SIGQUIT  
include/signal.h, 14, 定义为预处理宏
- SIGSEGV  
include/signal.h, 23, 定义为预处理宏
- sigset\_t  
include/signal.h, 7, 定义为类型
- SIGSTKFLT  
include/signal.h, 28, 定义为预处理宏
- SIGSTOP  
include/signal.h, 31, 定义为预处理宏
- sigsuspend  
include/signal.h, 65, 定义为函数原型
- SIGTERM  
include/signal.h, 27, 定义为预处理宏
- SIGTRAP  
include/signal.h, 16, 定义为预处理宏
- SIGTSTP  
include/signal.h, 32, 定义为预处理宏
- SIGTTIN  
include/signal.h, 33, 定义为预处理宏
- SIGTTOU  
include/signal.h, 34, 定义为预处理宏
- SIGUNUSED  
include/signal.h, 19, 定义为预处理宏
- SIGUSR1  
include/signal.h, 22, 定义为预处理宏
- SIGUSR2  
include/signal.h, 24, 定义为预处理宏
- size\_t  
include/sys/types.h, 6, 定义为类型
- include/time.h, 11, 定义为类型
- include/stddef.h, 11, 定义为类型
- include/string.h, 10, 定义为类型
- skip\_atoi  
kernel/vsprintf.c, 18, 定义为函数
- sleep\_if\_empty  
kernel/chr\_drv/tty\_io.c, 122, 定义为函数
- sleep\_if\_full  
kernel/chr\_drv/tty\_io.c, 130, 定义为函数
- sleep\_on  
include/sched.h, 145, 定义为函数原型
- kernel/sched.c, 151, 定义为函数
- SMALL  
kernel/vsprintf.c, 33, 定义为预处理宏
- SPACE  
kernel/vsprintf.c, 30, 定义为预处理宏
- SPECIAL  
kernel/vsprintf.c, 32, 定义为预处理宏
- speed\_t  
include/termios.h, 214, 定义为类型
- STO  
kernel/blk\_drv/floppy.c, 67, 定义为预处理宏
- STO\_DS  
include/fdreg.h, 30, 定义为预处理宏

- ST0\_ECE  
include/fdreg.h, 33, 定义为预处理宏
- ST0\_HA  
include/fdreg.h, 31, 定义为预处理宏
- ST0\_INTR  
include/fdreg.h, 35, 定义为预处理宏
- ST0\_NR  
include/fdreg.h, 32, 定义为预处理宏
- ST0\_SE  
include/fdreg.h, 34, 定义为预处理宏
- ST1  
kernel/blk\_drv/floppy.c, 68, 定义为预处理宏
- ST1\_CRC  
include/fdreg.h, 42, 定义为预处理宏
- ST1\_EOC  
include/fdreg.h, 43, 定义为预处理宏
- ST1\_MAM  
include/fdreg.h, 38, 定义为预处理宏
- ST1\_ND  
include/fdreg.h, 40, 定义为预处理宏
- ST1\_OR  
include/fdreg.h, 41, 定义为预处理宏
- ST1\_WP  
include/fdreg.h, 39, 定义为预处理宏
- ST2  
kernel/blk\_drv/floppy.c, 69, 定义为预处理宏
- ST2\_BC  
include/fdreg.h, 47, 定义为预处理宏
- ST2\_CM  
include/fdreg.h, 52, 定义为预处理宏
- ST2\_CRC  
include/fdreg.h, 51, 定义为预处理宏
- ST2\_MAM  
include/fdreg.h, 46, 定义为预处理宏
- ST2\_SEH  
include/fdreg.h, 49, 定义为预处理宏
- ST2\_SNS  
include/fdreg.h, 48, 定义为预处理宏
- ST2\_WC  
include/fdreg.h, 50, 定义为预处理宏
- ST3  
kernel/blk\_drv/floppy.c, 70, 定义为预处理宏
- ST3\_HA  
include/fdreg.h, 55, 定义为预处理宏
- ST3\_TZ  
include/fdreg.h, 56, 定义为预处理宏
- ST3\_WP  
include/fdreg.h, 57, 定义为预处理宏
- stack\_segment  
kernel/traps.c, 55, 定义为函数原型
- start\_buffer  
fs/buffer.c, 30, 定义为变量
- include/fs.h, 165, 定义为变量
- START\_CHAR  
include/tty.h, 41, 定义为预处理宏
- startup\_time  
include/sched.h, 140, 定义为变量
- init/main.c, 53, 定义为变量
- kernel/sched.c, 61, 定义为变量
- stat  
include/sys/stat.h, 6, 定义为struct类型
- include/sys/stat.h, 55, 定义为函数原型
- include/unistd.h, 232, 定义为函数原型
- state  
kernel/chr\_drv/console.c, 74, 定义为变量
- STATUS\_BUSY  
include/fdreg.h, 24, 定义为预处理宏
- STATUS\_BUSYMASK  
include/fdreg.h, 23, 定义为预处理宏
- STATUS\_DIR  
include/fdreg.h, 26, 定义为预处理宏
- STATUS\_DMA  
include/fdreg.h, 25, 定义为预处理宏
- STATUS\_READY  
include/fdreg.h, 27, 定义为预处理宏
- STDERR\_FILENO  
include/unistd.h, 15, 定义为预处理宏
- STDIN\_FILENO  
include/unistd.h, 13, 定义为预处理宏
- STDOUT\_FILENO  
include/unistd.h, 14, 定义为预处理宏
- sti  
include/asm/system.h, 16, 定义为预处理宏
- stime  
include/unistd.h, 234, 定义为函数原型
- STOP\_CHAR  
include/tty.h, 42, 定义为预处理宏
- str

include/sched.h, 159, 定义为预处理宏  
strcat  
include/string.h, 54, 定义为函数  
strchr  
include/string.h, 128, 定义为函数  
strcmp  
include/string.h, 88, 定义为函数  
strcpy  
include/string.h, 27, 定义为函数  
strcspn  
include/string.h, 185, 定义为函数  
strerror  
include/string.h, 13, 定义为函数原型  
strftime  
include/time.h, 39, 定义为函数原型  
STRINGIFY  
tools/build.c, 44, 定义为预处理宏  
strlen  
include/string.h, 263, 定义为函数  
strncat  
include/string.h, 68, 定义为函数  
strncmp  
include/string.h, 107, 定义为函数  
strncpy  
include/string.h, 38, 定义为函数  
strpbrk  
include/string.h, 209, 定义为函数  
strrchr  
include/string.h, 145, 定义为函数  
strspn  
include/string.h, 161, 定义为函数  
strstr  
include/string.h, 236, 定义为函数  
strtok  
include/string.h, 277, 定义为函数  
super\_block  
fs/super.c, 27, 定义为变量  
include/fs.h, 124, 定义为struct类型  
include/fs.h, 164, 定义为变量  
SUPER\_MAGIC  
include/fs.h, 41, 定义为预处理宏  
suser  
include/kernel.h, 21, 定义为预处理宏  
SUSPEND\_CHAR  
include/tty.h, 43, 定义为预处理宏  
switch\_to  
include/sched.h, 171, 定义为预处理宏  
sync  
include/unistd.h, 235, 定义为函数原型  
sync\_dev  
fs/buffer.c, 59, 定义为函数  
fs/super.c, 18, 定义为函数原型  
include/fs.h, 196, 定义为函数原型  
sync\_inodes  
fs/inode.c, 59, 定义为函数  
include/fs.h, 174, 定义为函数原型  
sys\_access  
fs/open.c, 47, 定义为函数  
include/sys.h, 34, 定义为函数原型  
sys\_acct  
include/sys.h, 52, 定义为函数原型  
kernel/sys.c, 77, 定义为函数  
sys\_alarm  
include/sys.h, 28, 定义为函数原型  
kernel/sched.c, 338, 定义为函数  
sys\_break  
include/sys.h, 18, 定义为函数原型  
kernel/sys.c, 21, 定义为函数  
sys\_brk  
include/sys.h, 46, 定义为函数原型  
kernel/sys.c, 168, 定义为函数  
sys\_call\_table  
include/sys.h, 74, 定义为变量  
sys\_chdir  
fs/open.c, 75, 定义为函数  
include/sys.h, 13, 定义为函数原型  
sys\_chmod  
fs/open.c, 105, 定义为函数  
include/sys.h, 16, 定义为函数原型  
sys\_chown  
fs/open.c, 121, 定义为函数  
include/sys.h, 17, 定义为函数原型  
sys\_chroot  
fs/open.c, 90, 定义为函数  
include/sys.h, 62, 定义为函数原型  
sys\_close  
fs/open.c, 192, 定义为函数  
fs/exec.c, 32, 定义为函数原型

fs/fcntl.c, 16, 定义为函数原型  
include/sys.h, 7, 定义为函数原型  
kernel/exit.c, 17, 定义为函数原型  
sys\_creat  
fs/open.c, 187, 定义为函数  
include/sys.h, 9, 定义为函数原型  
sys\_dup  
fs/fcntl.c, 42, 定义为函数  
include/sys.h, 42, 定义为函数原型  
sys\_dup2  
fs/fcntl.c, 36, 定义为函数  
include/sys.h, 64, 定义为函数原型  
sys\_execve  
include/sys.h, 12, 定义为函数原型  
sys\_exit  
fs/exec.c, 31, 定义为函数原型  
include/sys.h, 2, 定义为函数原型  
kernel/exit.c, 137, 定义为函数  
sys\_fcntl  
fs/fcntl.c, 47, 定义为函数  
include/sys.h, 56, 定义为函数原型  
sys\_fork  
include/sys.h, 3, 定义为函数原型  
sys\_fstat  
fs/stat.c, 47, 定义为函数  
include/sys.h, 29, 定义为函数原型  
sys\_ftime  
include/sys.h, 36, 定义为函数原型  
kernel/sys.c, 16, 定义为函数  
sys\_getegid  
include/sys.h, 51, 定义为函数原型  
kernel/sched.c, 373, 定义为函数  
sys\_geteuid  
include/sys.h, 50, 定义为函数原型  
kernel/sched.c, 363, 定义为函数  
sys\_getgid  
include/sys.h, 48, 定义为函数原型  
kernel/sched.c, 368, 定义为函数  
sys\_getpgrp  
include/sys.h, 66, 定义为函数原型  
kernel/sys.c, 201, 定义为函数  
sys\_getpid  
include/sys.h, 21, 定义为函数原型  
kernel/sched.c, 348, 定义为函数

sys\_getppid  
include/sys.h, 65, 定义为函数原型  
kernel/sched.c, 353, 定义为函数  
sys\_getuid  
include/sys.h, 25, 定义为函数原型  
kernel/sched.c, 358, 定义为函数  
sys\_gtty  
include/sys.h, 33, 定义为函数原型  
kernel/sys.c, 36, 定义为函数  
sys\_ioctl  
fs/ioctl.c, 30, 定义为函数  
include/sys.h, 55, 定义为函数原型  
sys\_kill  
include/sys.h, 38, 定义为函数原型  
kernel/exit.c, 60, 定义为函数  
sys\_link  
fs/namei.c, 721, 定义为函数  
include/sys.h, 10, 定义为函数原型  
sys\_lock  
include/sys.h, 54, 定义为函数原型  
kernel/sys.c, 87, 定义为函数  
sys\_lseek  
fs/read\_write.c, 25, 定义为函数  
include/sys.h, 20, 定义为函数原型  
sys\_mkdir  
fs/namei.c, 463, 定义为函数  
include/sys.h, 40, 定义为函数原型  
sys\_mknod  
fs/namei.c, 412, 定义为函数  
include/sys.h, 15, 定义为函数原型  
sys\_mount  
fs/super.c, 200, 定义为函数  
include/sys.h, 22, 定义为函数原型  
sys\_mpx  
include/sys.h, 57, 定义为函数原型  
kernel/sys.c, 92, 定义为函数  
sys\_nice  
include/sys.h, 35, 定义为函数原型  
kernel/sched.c, 378, 定义为函数  
sys\_open  
fs/open.c, 138, 定义为函数  
include/sys.h, 6, 定义为函数原型  
sys\_pause  
include/sys.h, 30, 定义为函数原型

kernel/sched.c, 144, 定义为函数  
kernel/exit.c, 16, 定义为函数原型  
sys\_phys  
include/sys.h, 53, 定义为函数原型  
kernel/sys.c, 82, 定义为函数  
sys\_pipe  
fs/pipe.c, 71, 定义为函数  
include/sys.h, 43, 定义为函数原型  
sys\_prof  
include/sys.h, 45, 定义为函数原型  
kernel/sys.c, 46, 定义为函数  
sys\_ptrace  
include/sys.h, 27, 定义为函数原型  
kernel/sys.c, 26, 定义为函数  
sys\_read  
fs/read\_write.c, 55, 定义为函数  
include/sys.h, 4, 定义为函数原型  
sys\_rename  
include/sys.h, 39, 定义为函数原型  
kernel/sys.c, 41, 定义为函数  
sys\_rmdir  
fs/namei.c, 587, 定义为函数  
include/sys.h, 41, 定义为函数原型  
sys\_setgid  
include/sys.h, 47, 定义为函数原型  
kernel/sys.c, 72, 定义为函数  
sys\_setpgid  
include/sys.h, 58, 定义为函数原型  
kernel/sys.c, 181, 定义为函数  
sys\_setregid  
include/sys.h, 72, 定义为函数原型  
kernel/sys.c, 51, 定义为函数  
sys\_setreuid  
include/sys.h, 71, 定义为函数原型  
kernel/sys.c, 118, 定义为函数  
sys\_setsid  
include/sys.h, 67, 定义为函数原型  
kernel/sys.c, 206, 定义为函数  
sys\_setuid  
include/sys.h, 24, 定义为函数原型  
kernel/sys.c, 143, 定义为函数  
sys\_setup  
include/sys.h, 1, 定义为函数原型  
kernel/blk\_drv/hd.c, 71, 定义为函数

sys\_sgetmask  
include/sys.h, 69, 定义为函数原型  
kernel/signal.c, 15, 定义为函数  
sys\_sigaction  
include/sys.h, 68, 定义为函数原型  
kernel/signal.c, 63, 定义为函数  
sys\_signal  
include/sys.h, 49, 定义为函数原型  
kernel/signal.c, 48, 定义为函数  
SYS\_SIZE  
tools/build.c, 35, 定义为预处理宏  
sys\_ssetmask  
include/sys.h, 70, 定义为函数原型  
kernel/signal.c, 20, 定义为函数  
sys\_stat  
fs/stat.c, 36, 定义为函数  
include/sys.h, 19, 定义为函数原型  
sys\_stime  
include/sys.h, 26, 定义为函数原型  
kernel/sys.c, 148, 定义为函数  
sys\_stty  
include/sys.h, 32, 定义为函数原型  
kernel/sys.c, 31, 定义为函数  
sys\_sync  
fs/buffer.c, 44, 定义为函数  
include/sys.h, 37, 定义为函数原型  
kernel/panic.c, 14, 定义为函数原型  
sys\_time  
include/sys.h, 14, 定义为函数原型  
kernel/sys.c, 102, 定义为函数  
sys\_times  
include/sys.h, 44, 定义为函数原型  
kernel/sys.c, 156, 定义为函数  
sys\_ulimit  
include/sys.h, 59, 定义为函数原型  
kernel/sys.c, 97, 定义为函数  
sys\_umask  
include/sys.h, 61, 定义为函数原型  
kernel/sys.c, 230, 定义为函数  
sys\_umount  
fs/super.c, 167, 定义为函数  
include/sys.h, 23, 定义为函数原型  
sys\_uname  
include/sys.h, 60, 定义为函数原型

- kernel/sys.c, 216, 定义为函数
- sys\_unlink
- fs/namei.c, 663, 定义为函数
- include/sys.h, 11, 定义为函数原型
- sys\_ustat
- fs/open.c, 19, 定义为函数
- include/sys.h, 63, 定义为函数原型
- sys\_utime
- fs/open.c, 24, 定义为函数
- include/sys.h, 31, 定义为函数原型
- sys\_waitpid
- include/sys.h, 8, 定义为函数原型
- kernel/exit.c, 142, 定义为函数
- sys\_write
- fs/read\_write.c, 83, 定义为函数
- include/sys.h, 5, 定义为函数原型
- sysbeep
- kernel/chr\_drv/console.c, 79, 定义为函数原型
- kernel/chr\_drv/console.c, 699, 定义为函数
- sysbeepstop
- kernel/chr\_drv/console.c, 691, 定义为函数
- system\_call
- kernel/sched.c, 51, 定义为函数原型
- TAB0
- include/termios.h, 116, 定义为预处理宏
- TAB1
- include/termios.h, 117, 定义为预处理宏
- TAB2
- include/termios.h, 118, 定义为预处理宏
- TAB3
- include/termios.h, 119, 定义为预处理宏
- TABDLY
- include/termios.h, 115, 定义为预处理宏
- table\_list
- kernel/chr\_drv/tty\_io.c, 99, 定义为变量
- task
- include/sched.h, 136, 定义为变量
- kernel/sched.c, 65, 定义为变量
- TASK\_INTERRUPTIBLE
- include/sched.h, 20, 定义为预处理宏
- TASK\_RUNNING
- include/sched.h, 19, 定义为预处理宏
- TASK\_STOPPED
- include/sched.h, 23, 定义为预处理宏
- task\_struct
- include/sched.h, 78, 定义为struct类型
- TASK\_UNINTERRUPTIBLE
- include/sched.h, 21, 定义为预处理宏
- task\_union
- kernel/sched.c, 53, 定义为union类型
- TASK\_ZOMBIE
- include/sched.h, 22, 定义为预处理宏
- tcdrain
- include/termios.h, 220, 定义为函数原型
- tcflow
- include/termios.h, 221, 定义为函数原型
- TCFLSH
- include/termios.h, 18, 定义为预处理宏
- tcflush
- include/termios.h, 222, 定义为函数原型
- TCGETA
- include/termios.h, 12, 定义为预处理宏
- tcgetattr
- include/termios.h, 223, 定义为函数原型
- TCGETS
- include/termios.h, 8, 定义为预处理宏
- TCIFLUSH
- include/termios.h, 205, 定义为预处理宏
- TCIOFF
- include/termios.h, 201, 定义为预处理宏
- TCIOFLUSH
- include/termios.h, 207, 定义为预处理宏
- TCION
- include/termios.h, 202, 定义为预处理宏
- TCOFLUSH
- include/termios.h, 206, 定义为预处理宏
- TCOOFF
- include/termios.h, 199, 定义为预处理宏
- TCOON
- include/termios.h, 200, 定义为预处理宏
- TCSADRAIN
- include/termios.h, 211, 定义为预处理宏
- TCSAFLUSH
- include/termios.h, 212, 定义为预处理宏
- TCSANOW
- include/termios.h, 210, 定义为预处理宏
- TCSBRK
- include/termios.h, 16, 定义为预处理宏

- tcsendbreak  
include/termios.h, 224, 定义为函数原型  
TCSETA  
include/termios.h, 13, 定义为预处理宏  
TCSETAF  
include/termios.h, 15, 定义为预处理宏  
tcsetattr  
include/termios.h, 225, 定义为函数原型  
TCSETAW  
include/termios.h, 14, 定义为预处理宏  
TCSETS  
include/termios.h, 9, 定义为预处理宏  
TCSETSF  
include/termios.h, 11, 定义为预处理宏  
TCSETSW  
include/termios.h, 10, 定义为预处理宏  
TCXONC  
include/termios.h, 17, 定义为预处理宏  
tell\_father  
kernel/exit.c, 83, 定义为函数  
termio  
include/termios.h, 44, 定义为struct类型  
termios  
include/termios.h, 54, 定义为struct类型  
ticks\_to\_floppy\_on  
include/fs.h, 170, 定义为函数原型  
include/fdreg.h, 9, 定义为函数原型  
kernel/sched.c, 206, 定义为函数  
time  
include/unistd.h, 236, 定义为函数原型  
include/time.h, 31, 定义为函数原型  
time\_init  
init/main.c, 76, 定义为函数  
TIME\_REQUESTS  
kernel/sched.c, 264, 定义为预处理宏  
time\_t  
include/sys/types.h, 11, 定义为类型  
include/time.h, 6, 定义为类型  
timer\_interrupt  
kernel/sched.c, 50, 定义为函数原型  
timer\_list  
kernel/sched.c, 266, 定义为struct类型  
kernel/sched.c, 270, 定义为变量  
times  
include/sys/times.h, 13, 定义为函数原型  
include/unistd.h, 237, 定义为函数原型  
TIOCEXCL  
include/termios.h, 19, 定义为预处理宏  
TIOCGPGRP  
include/termios.h, 22, 定义为预处理宏  
TIOCGSOFTCAR  
include/termios.h, 32, 定义为预处理宏  
TIOCGWINSZ  
include/termios.h, 26, 定义为预处理宏  
TIOCINQ  
include/termios.h, 34, 定义为预处理宏  
TIOCM\_CAR  
include/termios.h, 192, 定义为预处理宏  
TIOCM\_CD  
include/termios.h, 195, 定义为预处理宏  
TIOCM\_CTS  
include/termios.h, 191, 定义为预处理宏  
TIOCM\_DSR  
include/termios.h, 194, 定义为预处理宏  
TIOCM\_DTR  
include/termios.h, 187, 定义为预处理宏  
TIOCM\_LE  
include/termios.h, 186, 定义为预处理宏  
TIOCM\_RI  
include/termios.h, 196, 定义为预处理宏  
TIOCM\_RNG  
include/termios.h, 193, 定义为预处理宏  
TIOCM\_RTS  
include/termios.h, 188, 定义为预处理宏  
TIOCM\_SR  
include/termios.h, 190, 定义为预处理宏  
TIOCM\_ST  
include/termios.h, 189, 定义为预处理宏  
TIOCMBIC  
include/termios.h, 30, 定义为预处理宏  
TIOCMBIS  
include/termios.h, 29, 定义为预处理宏  
TIOCMGET  
include/termios.h, 28, 定义为预处理宏  
TIOCMSET  
include/termios.h, 31, 定义为预处理宏  
TIOCNXCL  
include/termios.h, 20, 定义为预处理宏

- TIOCOUTQ  
include/termios.h, 24, 定义为预处理宏
- TIOCSCTTY  
include/termios.h, 21, 定义为预处理宏
- TIOCSGRP  
include/termios.h, 23, 定义为预处理宏
- TIOCSOFTCAR  
include/termios.h, 33, 定义为预处理宏
- TIOCSTI  
include/termios.h, 25, 定义为预处理宏
- TIOCSWINSZ  
include/termios.h, 27, 定义为预处理宏
- tm  
include/time.h, 18, 定义为struct类型
- tmp\_floppy\_area  
kernel/blk\_drv/floppy.c, 105, 定义为变量
- tms  
include/sys/times.h, 6, 定义为struct类型
- toascii  
include/ctype.h, 29, 定义为预处理宏
- tolower  
include/ctype.h, 31, 定义为预处理宏
- top  
kernel/chr\_drv/console.c, 73, 定义为变量
- TOSTOP  
include/termios.h, 177, 定义为预处理宏
- toupper  
include/ctype.h, 32, 定义为预处理宏
- track  
kernel/blk\_drv/floppy.c, 118, 定义为变量
- transfer  
kernel/blk\_drv/floppy.c, 309, 定义为函数
- trap\_init  
include/sched.h, 34, 定义为函数原型
- kernel/traps.c, 181, 定义为函数
- TRKO\_ERR  
include/hdreg.h, 46, 定义为预处理宏
- truncate  
fs/truncate.c, 47, 定义为函数
- include/fs.h, 173, 定义为函数原型
- try\_to\_share  
mm/memory.c, 292, 定义为函数
- tss\_struct  
include/sched.h, 51, 定义为struct类型
- TSTPMASK  
kernel/chr\_drv/tty\_io.c, 21, 定义为预处理宏
- TTY\_BUF\_SIZE  
include/termios.h, 4, 定义为预处理宏
- include/tty.h, 14, 定义为预处理宏
- tty\_init  
include/tty.h, 67, 定义为函数原型
- kernel/chr\_drv/tty\_io.c, 105, 定义为函数
- tty\_intr  
kernel/chr\_drv/tty\_io.c, 111, 定义为函数
- tty\_ioctl  
fs/ioctl.c, 13, 定义为函数原型
- kernel/chr\_drv/tty\_ioctl.c, 115, 定义为函数
- tty\_queue  
include/tty.h, 16, 定义为struct类型
- tty\_read  
fs/char\_dev.c, 16, 定义为函数原型
- include/tty.h, 69, 定义为函数原型
- kernel/chr\_drv/tty\_io.c, 230, 定义为函数
- tty\_struct  
include/tty.h, 45, 定义为struct类型
- tty\_table  
include/tty.h, 55, 定义为struct类型
- kernel/chr\_drv/tty\_io.c, 51, 定义为struct类型
- tty\_write  
fs/char\_dev.c, 17, 定义为函数原型
- include/kernel.h, 8, 定义为函数原型
- include/sched.h, 36, 定义为函数原型
- include/tty.h, 70, 定义为函数原型
- kernel/chr\_drv/tty\_io.c, 290, 定义为函数
- TYPE  
kernel/blk\_drv/floppy.c, 53, 定义为预处理宏
- tzset  
include/time.h, 40, 定义为函数原型
- u\_char  
include/sys/types.h, 33, 定义为类型
- uid\_t  
include/sys/types.h, 24, 定义为类型
- ulimit  
include/unistd.h, 238, 定义为函数原型
- umask  
include/sys/stat.h, 56, 定义为函数原型
- include/unistd.h, 239, 定义为函数原型
- umode\_t



- include/sys/types.h, 29, 定义为类型
- umount
- include/unistd.h, 240, 定义为函数原型
- un\_wp\_page
- mm/memory.c, 221, 定义为函数
- uname
- include/sys/utsname.h, 14, 定义为函数原型
- include/unistd.h, 241, 定义为函数原型
- unexpected\_floppy\_interrupt
- kernel/blk\_drv/floppy.c, 353, 定义为函数
- unexpected\_hd\_interrupt
- kernel/blk\_drv/hd.c, 237, 定义为函数
- unlink
- include/unistd.h, 242, 定义为函数原型
- unlock\_buffer
- kernel/blk\_drv/ll\_rw\_blk.c, 51, 定义为函数
- kernel/blk\_drv/blk.h, 101, 定义为函数
- unlock\_inode
- fs/inode.c, 37, 定义为函数
- usage
- tools/build.c, 52, 定义为函数
- USED
- mm/memory.c, 47, 定义为预处理宏
- user\_stack
- kernel/sched.c, 67, 定义为变量
- ushort
- include/sys/types.h, 34, 定义为类型
- ustat
- include/sys/types.h, 39, 定义为struct类型
- include/unistd.h, 243, 定义为函数原型
- utimbuf
- include/utime.h, 6, 定义为struct类型
- utime
- include/unistd.h, 244, 定义为函数原型
- include/utime.h, 11, 定义为函数原型
- utsname
- include/sys/utsname.h, 6, 定义为struct类型
- va\_arg
- include/stdarg.h, 24, 定义为预处理宏
- va\_end
- include/stdarg.h, 22, 定义为预处理宏
- include/stdarg.h, 21, 定义为函数原型
- va\_list
- include/stdarg.h, 4, 定义为类型
- va\_start
- include/stdarg.h, 13, 定义为预处理宏
- include/stdarg.h, 16, 定义为预处理宏
- VDISCARD
- include/termios.h, 77, 定义为预处理宏
- VEOF
- include/termios.h, 68, 定义为预处理宏
- VEOL
- include/termios.h, 75, 定义为预处理宏
- VEOL2
- include/termios.h, 80, 定义为预处理宏
- VERASE
- include/termios.h, 66, 定义为预处理宏
- verify\_area
- include/kernel.h, 4, 定义为函数原型
- kernel/fork.c, 24, 定义为函数
- video\_erase\_char
- kernel/chr\_drv/console.c, 67, 定义为变量
- video\_mem\_end
- kernel/chr\_drv/console.c, 64, 定义为变量
- video\_mem\_start
- kernel/chr\_drv/console.c, 63, 定义为变量
- video\_num\_columns
- kernel/chr\_drv/console.c, 59, 定义为变量
- video\_num\_lines
- kernel/chr\_drv/console.c, 61, 定义为变量
- video\_page
- kernel/chr\_drv/console.c, 62, 定义为变量
- video\_port\_reg
- kernel/chr\_drv/console.c, 65, 定义为变量
- video\_port\_val
- kernel/chr\_drv/console.c, 66, 定义为变量
- video\_size\_row
- kernel/chr\_drv/console.c, 60, 定义为变量
- video\_type
- kernel/chr\_drv/console.c, 58, 定义为变量
- VIDEO\_TYPE\_CGA
- kernel/chr\_drv/console.c, 50, 定义为预处理宏
- VIDEO\_TYPE\_EGAC
- kernel/chr\_drv/console.c, 52, 定义为预处理宏
- VIDEO\_TYPE\_EGAM
- kernel/chr\_drv/console.c, 51, 定义为预处理宏
- VIDEO\_TYPE\_MDA
- kernel/chr\_drv/console.c, 49, 定义为预处理宏

- VINTR  
include/termios.h, 64, 定义为预处理宏
- VKILL  
include/termios.h, 67, 定义为预处理宏
- VLNEXT  
include/termios.h, 79, 定义为预处理宏
- VMIN  
include/termios.h, 70, 定义为预处理宏
- VQUIT  
include/termios.h, 65, 定义为预处理宏
- VREPRINT  
include/termios.h, 76, 定义为预处理宏
- vsprintf  
init/main.c, 44, 定义为函数原型  
kernel/printk.c, 19, 定义为函数原型  
kernel/vsprintf.c, 92, 定义为函数
- VSTART  
include/termios.h, 72, 定义为预处理宏
- VSTOP  
include/termios.h, 73, 定义为预处理宏
- VSUSP  
include/termios.h, 74, 定义为预处理宏
- VSWTC  
include/termios.h, 71, 定义为预处理宏
- VT0  
include/termios.h, 125, 定义为预处理宏
- VT1  
include/termios.h, 126, 定义为预处理宏
- VTDLY  
include/termios.h, 124, 定义为预处理宏
- VTIME  
include/termios.h, 69, 定义为预处理宏
- VWERASE  
include/termios.h, 78, 定义为预处理宏
- W\_OK  
include/unistd.h, 24, 定义为预处理宏
- wait  
include/sys/wait.h, 20, 定义为函数原型  
include/unistd.h, 246, 定义为函数原型  
lib/wait.c, 13, 定义为函数
- wait\_for\_keypress  
fs/super.c, 19, 定义为函数原型  
kernel/chr\_drv/tty\_io.c, 140, 定义为函数
- wait\_for\_request  
kernel/blk\_drv/ll\_rw\_blk.c, 26, 定义为变量  
kernel/blk\_drv/blk.h, 52, 定义为变量
- wait\_motor  
kernel/sched.c, 201, 定义为变量
- wait\_on  
include/fs.h, 175, 定义为函数原型
- wait\_on\_buffer  
fs/buffer.c, 36, 定义为函数
- wait\_on\_floppy\_select  
kernel/blk\_drv/floppy.c, 123, 定义为变量
- wait\_on\_inode  
fs/inode.c, 20, 定义为函数
- wait\_on\_super  
fs/super.c, 48, 定义为函数
- wait\_until\_sent  
kernel/chr\_drv/tty\_ioctl.c, 46, 定义为函数
- waitpid  
include/sys/wait.h, 21, 定义为函数原型  
include/unistd.h, 245, 定义为函数原型
- wake\_up  
include/sched.h, 147, 定义为函数原型  
kernel/sched.c, 188, 定义为函数
- WAKEUP\_CHARS  
kernel/chr\_drv/serial.c, 21, 定义为预处理宏
- WEXITSTATUS  
include/sys/wait.h, 15, 定义为预处理宏
- WIFEXITED  
include/sys/wait.h, 13, 定义为预处理宏
- WIFSIGNALED  
include/sys/wait.h, 18, 定义为预处理宏
- WIFSTOPPED  
include/sys/wait.h, 14, 定义为预处理宏
- WIN\_DIAGNOSE  
include/hdreg.h, 41, 定义为预处理宏
- WIN\_FORMAT  
include/hdreg.h, 38, 定义为预处理宏
- WIN\_INIT  
include/hdreg.h, 39, 定义为预处理宏
- WIN\_READ  
include/hdreg.h, 35, 定义为预处理宏
- WIN\_RESTORE  
include/hdreg.h, 34, 定义为预处理宏
- win\_result  
kernel/blk\_drv/hd.c, 169, 定义为函数

WIN\_SEEK  
include/hdreg.h, 40, 定义为预处理宏  
WIN\_SPECIFY  
include/hdreg.h, 42, 定义为预处理宏  
WIN\_VERIFY  
include/hdreg.h, 37, 定义为预处理宏  
WIN\_WRITE  
include/hdreg.h, 36, 定义为预处理宏  
winsize  
include/termios.h, 36, 定义为struct类型  
WNOHANG  
include/sys/wait.h, 10, 定义为预处理宏  
WRERR\_STAT  
include/hdreg.h, 29, 定义为预处理宏  
write  
include/unistd.h, 247, 定义为函数原型  
WRITE  
include/fs.h, 27, 定义为预处理宏  
write\_inode  
fs/inode.c, 18, 定义为函数原型  
fs/inode.c, 314, 定义为函数  
write\_intr  
kernel/blk\_drv/hd.c, 269, 定义为函数  
write\_pipe  
fs/read\_write.c, 17, 定义为函数原型  
fs/pipe.c, 41, 定义为函数  
write\_verify  
kernel/fork.c, 20, 定义为函数原型  
mm/memory.c, 261, 定义为函数  
WRITEA  
include/fs.h, 29, 定义为预处理宏  
WSTOPSIG  
include/sys/wait.h, 17, 定义为预处理宏  
WTERMSIG  
include/sys/wait.h, 16, 定义为预处理宏  
WUNTRACED  
include/sys/wait.h, 11, 定义为预处理宏  
X\_OK  
include/unistd.h, 23, 定义为预处理宏  
XCASE  
include/termios.h, 171, 定义为预处理宏  
XTABS  
include/termios.h, 120, 定义为预处理宏  
y  
kernel/chr\_drv/console.c, 72, 定义为变量  
YEAR  
kernel/mktime.c, 23, 定义为预处理宏  
Z\_MAP\_SLOTS  
include/fs.h, 40, 定义为预处理宏  
ZEROPAD  
kernel/vsprintf.c, 27, 定义为预处理宏  
ZMAGIC  
include/a.out.h, 27, 定义为预处理宏

