



C 语言标准库函数

龙马工作室搜集整理制作





索引

A

abort	10
abs	10
absread	11
abswrite	11
acos	11
asctime	12
asin	12
assert	13
atan	13
atexit	13
atof	14
atoi	14
atol	15

B

bcd	16
biosdisk	16
biosprint	17
bsearch	17

C

calloc	18
ceil	18





cgets	19
chdir	19
_chmod	19
chmod	20
clearer	20
clock	20
close	21
closegraph	21
cos	21
ctime	22
ctrlbrk	22

D

delay	23
disable	23
difftime	23
div	24

E

enable	25
exit	25
exp	25

F

fabs	27
farcalloc	27
farfree	28
farmalloc	28
farrealloc	28
fclose	29
fcloseall	29
feof	29





fgetc	30
fgets	30
fgetpos	31
fflush	31
floodfill	31
floor	32
fnmerge	32
fnsplit	32
fopen	33
fp_off	33
fprintf	34
fputc	34
fputs	34
fp_seg	35
free	35
freopen	36
frexp	36
fscanf	36
fseek	37
fsetpos	37
ftell	37
fwrite	38

G

gcv	39
getc	39
getchar	39
getcurdir	40
getcwd	40
getdate	41
getdfree	41
getdisk	41
getenv	42
getimage	42
getmaxx	42





getmaxy -----	43
getpixel-----	43
getpsp-----	43
gettime -----	44
getvect-----	44
getx -----	45
gety -----	45
gmtime -----	45

I

imagesize -----	47
initgraph-----	47
inport -----	47
inportb-----	48
int86 -----	48
int86x-----	49
intdos -----	49
intdosx-----	49
intr -----	50
isalnum -----	50
isalpha -----	50
isctrl -----	51
isdigit -----	51
isgraph-----	52
islower-----	52
isupper-----	52
itoa -----	53

K

kbhit -----	54
keep-----	54



L

labs-----	55
ldexp-----	55
ldiv-----	56
localtime-----	56
log-----	56
log10-----	57
longjmp-----	57
_lrotl-----	57
_lrotr-----	58
lseek-----	58
ltoa-----	59

M

malloc-----	60
mkdir-----	60
mblen-----	61
memchr-----	61
memcmp-----	61
memcpy-----	62
memmove-----	62
memset-----	63
mktemp-----	63
mk_fp-----	63
modf-----	64
movedata-----	64

O

_open-----	65
open-----	65
outport-----	66



outtextxy -----66

P

peek-----67
perror -----67
pow -----67
printf-----68
putch-----68
putchar -----68
putenv -----69
putimage -----69
putpixel-----70
puts -----70

Q

qsort -----70

R

_read-----72
read -----72
realloc -----73
rectangle -----73
remove -----73
rename-----74
restorecrtnode -----74
rewind -----75
rmdir-----75
_rotl-----75
_rotr-----76





S

scanf	77
searchpath	77
setactivepage	78
setbuf	78
setcolor	78
setdisk	79
setgraphmode	79
setjmp	79
setlinestyle	80
setlocale	80
setvbuf	81
setvisualpage	81
signal	81
sin	82
sinh	82
sprintf	82
sqrt	83
srand	83
sscanf	83
strcmp	84
strcpy	84
strcspn	85
strerror	85
strlen	86
strncat	86
strncmp	86
strncpy	87
strchr	87
strstr	88
strtok	88
strtoul	88
system	89



T

tan	90
time	90
tmpnam	91
tolower	91
toupper	91

U

ungetc	93
--------	----

V

vfprintf	94
vprintf	94
vsprintf	95

W

wctombs	96
wctomb	96
_write	97
write	97





A

abort

功能

异常终止程序

头文件

stdlib.h

语法

```
void abort(void);
```

abs

功能

整数的绝对值

头文件

stdlib.h

语法

```
int abs(int j);
```

返回值

整数 j 的绝对值。如果不能表示 j 的绝对值，那么函数的行为是未定义的。





absread

功能

读磁盘绝对扇区函数

头文件

dos.h

语法

```
int absread(int drive,int num,int sectnum,void *buf)
```

返回值

0:成功; -1:失败。

abswrite

功能

写磁盘绝对扇区函数

头文件

dos.h

语法

```
int abswrite(int drive,int nsects,int lsect,void *buffer)
drive=0(A 驱动器)、1(B 驱动器)、
nsects=要写的扇区数(最多 64K 个);
lsect=起始逻辑扇区号;
buffer=要写入数据的内存起始地址。
```

返回值

0:成功; -1:失败。

acos

功能





反余弦

头文件

math.h

语法

```
math.hdouble acos(double x);
```

返回值

x 的反余弦值。返回值的范围在 0 到 π 之间。如果 x 的值不在 -1 到 +1 之间，那么就会发生定义域错误。

asctime

功能

把日期和时间转换成 ASCII 码

头文件

time.h

语法

```
char *asctime(const struct tm *timeptr);
```

返回值

指向以空字符结尾的字符串的指针。

asin

功能

反正弦

头文件

math.h

语法





```
double asin(double x);
```

返回值

x 的反正弦值。返回值的范围在 $-\pi/2$ 到 $\pi/2$ 之间。如果 x 的值不在-1 到+1 之间，那么就会发

生定义域错误。

assert

功能

诊断表达式的真值

头文件

assert.h

语法

```
void assert(int expression);
```

atan

功能

反正切

头文件

math.h

语法

```
double atan(double x);
```

返回值

x 的反正切值。返回值的范围在 $-\pi/2$ 到 $\pi/2$ 之间。

atexit

功能

在程序退出处注册要调用的函数





头文件

stdlib.h

语法

```
int atexit(void (*func)(void));
```

返回值

如果成功，返回零。如果不成功，则返回非零（达到由实现定义的限制）。

atof

功能

将字符串转换成浮点数的函数

头文件

math.h、stdlib.h

语法

```
double atof(const char *s)
```

返回值

字符串的转换值。

atoi

功能

将字符串转换成整型数的函数

头文件

stdlib.h

语法

```
int atoi(const char *s)
```

返回值

字符串的转换值。若出错则返回 0。





atol

功能

将字符串转换成长整型数的函数

头文件

stdlib.h

语法

```
long atol(const char *s)
```

返回值

字符串的转换值。若出错则返回 0。





B

bcd

功能

把一个数转换成对应的 BCD 码的函数

头文件

bcd.h

语法

```
bcd bcd(int x)
bcd bcd(double x)
bcd bcd(double x,int decimals)
```

biosdisk

功能

调用 BIOS 磁盘驱动程序函数

头文件

bios.h

语法

```
char biosdisk(int cmd,int drive,int head,int track,int sector,int nsects,void *buffer)
```

返回值

0x00-操作成功
0x01-错误命令
0x03-企图写具有写保护的磁盘



0x07-驱动器参数设置错误
0x10-磁盘读/CRC/ECC 错误

biosprint

功能

调用 BIOS 打印机 I/O 接口的函数

头文件

bios.h

语法

```
int biosprint(int cmd,int abyte,int port)
```

返回值

打印机状态

位 0(设备超时); 位 3(I/O 出错);位 4(打印机已选择); 位 5(没纸);位 6(打印机确认);位 7(不忙)

bsearch

功能

二分检索

头文件

stdlib.h

语法

```
void *bsearch(const void *key, const void *base,size_t memb,size_t size, int  
(*compar)(const void *, const void *));
```

返回值

指向数组元素的指针，此数组元素是用来测试是否等于关键字的。



C

calloc

功能

分配内存函数

头文件

stdlib.h、calloc.h

语法

```
void *calloc(size_t nitems, size_t size)
```

返回值

返回指向新分配内存的指针。空间不够则返回 NULL。

ceil

功能

求不小于 num 的最小双精度整数

头文件

math.h

语法

```
double ceil(double num)
```





cgets

功能

读字符串函数

头文件

conio.h

语法

```
char *cgets(char *str)
```

返回值

指向 str[2]的指针。

chdir

功能

改变当前目录的函数

头文件

dir.h

语法

```
int chdir(const char *path)
```

返回值

0(成功); -1(失败)

_chmod

功能

改变文件存取权限的函数

头文件

io.h、dos.h





语法

```
int _chmod(const char *path,int func[,int attrib])
```

返回值

失败时返回-1。成功时返回文件的属性字节,如果 返回值&FA_RDONLY 不等于 0,则文件是只读文件;

chmod

功能

改变文件存取权限的函数

头文件

io.h、sys\stat.h

语法

```
int chmod(const char *path,int amode)
```

返回值

0(成功); -1(失败)

clearer

功能

清除流错误

头文件

stdio.h

语法

```
void clearerr(FILE *stream);
```

clock

功能





测得从程序开始到调用处处理机所用的时间

头文件

time.h

语法

```
long clock(void)
```

close

功能

关闭文件函数

头文件

io.h

语法

```
int close(int handle)
```

返回值

0(成功); -1(失败)

closegraph

功能

关闭图形函数

头文件

graphics.h

语法

```
void far closegraph(void)
```

COS

功能





计算 \arg (弧度)的余弦值

头文件

math.h

语法

```
double cos(double arg)
```

ctime

功能

把日期和时间转换成字符串

头文件

time.h

语法

```
char *ctime(const time_t *timer);
```

返回值

指向字符串的指针，此字符串描述了本地时间，此时间等价于 `timer` 指向的日历时间。
等价于 `asctime(localtime(timer))`。

ctrlbrk

功能

设置 `ctrl-break` 处理程序的函数

头文件

dos.h

语法

```
void ctrlbrk(int(*handle)(void))
```





D

delay

功能

暂停函数

头文件

dos.h

语法

```
void delay(unsigned milliseconds)
```

disable

功能

屏蔽中断的宏

头文件

dos.h

语法

```
void disable(void)
```

difftime

功能

时间差

头文件





time.h

语法

```
double difftime(time_t time1, time_t time0);
```

返回值

time0（较早的时间）和 time1 之间的差值，此值按秒来衡量。

div

功能

整数除法

头文件

stdlib.h

语法

```
div_t div(int numer, int denom);
```

返回值

含有 quot（numer 除以 denom 时的商）和 rem（余数）的结构。如果无法表示结果，那么函数的行为是未定义的。





E

enable

功能

开硬件中断的宏

头文件

dos.h

语法

```
void enable(void)
```

exit

功能

退出程序

头文件

stdlib.h

语法

```
void exit(int status);
```

exp

功能

指数

头文件





math.h

语法

```
double exp(double x);
```

返回值

e 的 x 次幂的值（即 e^x ）。如果 x 的数过大，那么可能会发生取值范围错误。





F

fabs

功能

浮点数的绝对值

头文件

math.h

语法

```
double fabs(double x);
```

返回值

x 的绝对值。

faralloc

功能

从远堆中分配内存的函数

头文件

alloc.h

语法

```
void far *faralloc(unsigned long nunits,unsigned long unitsz)
```

返回值

返回指向新分配块的指针，若内存不够，则返回 NULL。





farfree

功能

从远堆中释放一块已分配内存的函数

头文件

alloc.h

语法

```
void farfree(void far *block)
```

farmalloc

功能

从远堆中分配内存的函数

头文件

alloc.h

语法

```
void far *farmalloc(unsigned long nbytes)
```

返回值

返回指向新分配内存的指针，若内存不够，则返回 NULL。

farrealloc

功能

调整远堆中已分配块的函数

头文件：

alloc.h

语法

```
void far *farrealloc(void far *oldblock,unsigned long nbytes)
```



返回值

返回调整后的新内存地址。若不能重新分配，则返回 NULL。

fclose

功能

关闭一个流函数

头文件

stdio.h

语法

```
int fclose(FILE *stream)
```

返回值

0(成功); EOF(失败)

fcloseall

功能

关闭打开的流的函数

头文件

stdio.h

语法

```
int fcloseall(void)
```

返回值

关闭流的总数。如果发现错误则返回 EOF。

feof

功能

检查文件是否结束的函数

头文件

stdio.h

语法

```
int feof(FILE *fp)
```

返回值

文件结束返回非 0 值，否则返回 0。

fgetc

功能

从文件中读取字符

头文件

stdio.h

语法

```
int fgetc(FILE *stream);
```

返回值

读到的字符。如果 fgetc 函数遇到流的末尾，则设置流的文件尾指示器并且返回 EOF。如果读取发生错误，fgetc 函数设置流的错误指示器并且返回 EOF。

fgets

功能

从流中读取一字符串的函数

头文件

stdio.h

语法

```
char *fgets(char s[],int n,FILE *stream)
```

返回值



成功时返回字符串参数 s,出错或遇到文件结束时, 返回 NULL。

fgetpos

功能

获得文件位置

头文件

stdio.h

语法

```
int fgetpos(FILE *stream, fpos_t *pos);
```

返回值

如果成功就返回零。如果调用失败, 则返回非零值, 并且把由实现定义的错误码存储到 errno 中。

fflush

功能

清洗文件缓冲区

头文件

stdio.h

语法

```
int fflush(FILE *stream);
```

返回值

如果成功就返回零。如果检测到错误, 就返回 EOF。

floodfill

功能

填充区域的函数





头文件

graphics.h

语法

```
void far floodfill(int x,int y,int border)
```

floor

功能

求不大于 num 的最大双精度整数

头文件

math.h

语法

```
double floor(double num)
```

fnmerge

功能

建立文件路径函数

头文件

dir.h

语法

```
void fnmerge(char *path,const char *drive,const char *dir, const char *name,const char *ext)
```

fnsplit

功能

分解完整的路径名函数

头文件





dir.h

语法

```
int fnsplit(char *path,const char *drive,const char *dir,const char *name,const char *ext)
```

返回值:

如果有扩展名, 则返回值&EXTENSION!=0

如果有文件名, 则返回值&FILENAME!=0

如果有目录名, 则返回值&DIRECTORY!=0

如果有驱动器号, 则返回值&DIRVE!=0

fopen

功能

打开一个流函数

头文件

stdio.h

语法

```
FILE *fopen(const char *filename,const char *mode)
```

返回值

指明流的指针(成功时)或 NULL(失败时)

fp_off

功能

获取远地址偏移量的宏

头文件

dos.h

语法

```
unsigned FP_OFF(void far *p)
```

返回值



偏移量

fprintf

功能

传送输出到一个流中的函数

头文件

stdio.h

语法

```
int fprintf(FILE *stream,const char *format[,argument,...])
```

返回值

成功则返回输出的字节数，错误则返回 EOF。

fputc

功能

向文件写字符

头文件

stdio.h

语法

```
int fputc(int c, FILE *stream);
```

返回值

c（写入的字符）。如果写发生错误，fputc 函数会为 stream 设置错误指示器，并且返回 EOF。

fputs

功能

向文件写字符串





头文件

stdio.h

语法

```
int fputs(const char *s, FILE *stream);
```

返回值

如果成功，返回非负值。如果写发生错误，则返回 EOF。

fp_seg

功能

获取远地址段值的宏

头文件

dos.h

语法

```
unsigned FP_SEG(void far *p)
```

返回值

段地址值

free

功能

释放已分配内存的函数

头文件

stdlib.h、alloc.h

语法

```
void free(void *block)
```



freopen

功能

重新打开文件

头文件

stdio.h

语法

```
FILE *freopen(const char *filename, const char *mode, FILE *stream);
```

返回值

如果操作成功，返回 stream 的值。如果无法打开文件则返回空指针。

frexp

功能

分解成小数和指数

头文件

math.h

语法

```
double frexp(double value, int *exp);
```

fscanf

功能

格式化输入函数

头文件

stdio.h

语法

```
int fscanf(FILE *stream, const char *format[, address,...])
```





fseek

功能

移动文件指针函数

头文件

stdio.h

语法

```
int fseek(FILE *stream, long offset, int whence)
```

返回值

0 表示成功，非 0 值表示失败

fsetpos

功能

设置文件位置

头文件

stdio.h

语法

```
int fsetpos(FILE *stream, const fpos_t *pos);
```

返回值

如果成功就返回零。如果调用失败，返回非零值，并且把由实现定义的错误码存储在 errno 中。

ftell

功能

确定文件位置

头文件

stdio.h





语法

```
long int ftell(FILE *stream);
```

返回值

返回 `stream` 指向的流的当前文件位置指示器。如果调用失败，返回-1L，并且把由实现定义的错误码存储在 `errno` 中。

fwrite

功能

把参数写入流中的函数

头文件

`stdio.h`

语法

```
size_t fwrite(const void *ptr,size_t size,size_t n,FILE *stream)
```

返回值

调用成功时返回实际写的数据项数，出错时返回一短整形数值。





G

gcvt

功能

把双精度数转化成字符串的函数

头文件

stdlib.h

语法

```
char*gcvt(value,ndigit,buf)
```

getc

功能

从流中取字符的宏

头文件

stdio.h

语法

```
int getc(FILE *stream)
```

getchar

功能

从 stdin 流中读取一个字符的函数

头文件





stdio.h

语法

字符型变量=getchar

getcurdir

功能

读取指定驱动器的当前目录的函数

头文件

dir.h

语法

```
int getcurdir(int drive,char directory)
drive=0(缺省);1(A 驱动器); ...
directory 用来存放目录名，不包括驱动器名，不以反斜杠开始。
```

返回值

0(调用成功);1(出错)

getcwd

功能

读取当前目录的函数

头文件

dir.h

语法

```
char *getcwd(char *buf,int buflen)
```

返回值

若 buf 非空，调用成功返回 buf,出错返回 NULL;若 buf 为 NULL,返回指向已经分配的内存缓冲区地址。





getdate

功能

读取系统日期函数

头文件

dos.h

语法

```
void getdate(pdate)struct date{int da_year;int da_day;int da_mon;};  
struct date *pdate;
```

getdfree

功能

读取磁盘空闲空间的函数

头文件

dir.h

语法

```
void getdfree(drive,dtable)
```

getdisk

功能

读取当前磁盘驱动器号函数

头文件

dir.h

语法

```
int getdisk(void)
```





getenv

功能

读取环境变量的当前值的函数

头文件

stdlib.h

语法

```
char *getenv(const char *name)
```

getimage

功能

将指定区域的位图象存入内存的函数

头文件

graphics.h

语法

```
void far getimage(int left,int top,int right,int bottom,void far *bitmap)
```

getmaxx

功能

返回屏幕上最大的 X 坐标值的函数

头文件

graphics.h

语法

```
int far getmaxx(void)
```





getmaxy

功能

返回屏幕上最大的 Y 坐标值的函数

头文件

graphics.h

语法

```
int far getmaxy(void)
```

getpixel

功能

读取像素颜色的函数

头文件

graphics.h

语法

```
unsigned far getpixel(int x,int y)
```

getpsp

功能

使用 DOS 系统调用 0x62，返回程序段前缀（PSP）的段地址

头文件

dos.h

语法

```
unsigned getpsp(void)
```





gettime

功能

读取系统时间的函数

头文件

time.h

语法

```
void gettime(ptime)
struct time
{
    unsigned char ti_min;
    unsigned char ti_hour;
    unsigned char ti_hund;//百分之一秒
    unsigned char ti_sec;
};
struct time *ptime;
```

getvect

功能

读取中断向量函数

头文件

dos.h

语法

```
void interrupt(*getvect(int interruptno))
```

返回值

指定中断向量的当前 4 字节值。





getx

功能

返回当前图形方式下位置的 X 坐标值的函数

头文件

graphics.h

语法

```
int far getx(void);
```

gety

功能

返回当前图形方式下位置的 Y 坐标值的函数

头文件

graphics.h

语法

```
int far gety(void);
```

gmtime

功能

转换成格林威治标准时间

头文件

time.h

语法

```
struct tm *gmtime(const time_t *timer);
```

返回值

指向结构的指针，此结构包含的分解的 UTC（协调世界时间—从前的格林威治时间）值等





价于 `timer` 指向的日历时间。如果 UTC 无效，则返回空指针。





I

imagesize

功能

返回保存图像缓冲区大小的函数

头文件

graphics.h

语法

```
unsigned far imagesize(int left,int top,int right,int bottom)
```

返回值

返回存储区的大小(字节)，若所需内存大于等于 64K-1 字节，则返回 0xffff(-1)。

initgraph

功能

显示模式控制函数(初始化图形系统函数)

头文件

graphics.h

语法

```
void far initgraph(int far *driver,int far *mode,char far*path)
```

inport

功能





从端口中读入一个字的函数

头文件

dos.h

语法

```
int inport(int portid)
```

inportb

功能

从端口读入一个字节的函数

头文件

dos.h

语法

```
unsigned char inportb(int portid)
```

int86

功能

执行中断函数(调用 8086 软中断函数)

头文件

dos.h

语法

```
int int86(int intno,union REGS *inregs,union REGS *outregs)
```

返回值

中断处理结束时的 AX 寄存器的值，错误时 outregs-x.cflag!=0





int86x

功能

执行中断函数 (通用 8086 软中断接口函数)

头文件

dos.h

语法

```
int int86x(int intno,union REGS *inregs,union REGS *outregs,struct SREGS *segregs)
```

intdos

功能

通用 dos 中断接口函数

头文件

dos.h

语法

```
int intdos(inregs,outregs)
union REGS *inregs;//调用时的各寄存器的值
union REGS *outregs;//返回寄存器的值
```

intdosx

功能

通用 dos 中断接口函数

```
int intdosx(inregs,outregs,segregs)
```

头文件

dos.h

语法

```
union REGS *inregs;调用时的各寄存器的值
```





union REGS *outregs;返回寄存器的值

struct SREGS *seregs;调用时的段寄存器的值（还设置返回时的值）

intr

功能

执行 8086 软中断函数(改变软中断接口函数)

头文件

dos.h

语法

```
void intr(int intno,struct REGPACK *preg)
```

isalnum

功能

测试是字母或数字

头文件

ctype.h

语法

```
int isalnum(int c);
```

返回值

如果 isalnum 是字母或数字，返回非零值；否则返回零。（如果 isalph(c)或 isdigit(c) 为真，则 c 是字母或数字。）

isalpha

功能

测试字母

头文件





ctype.h

语法

```
int isalpha(int c);
```

返回值

如果 `isalnum` 是字母，返回非零值；否则返回零。（如果 `islower(c)` 或 `isupper(c)` 为真，则 `c` 是字母。）

isctrl

功能

测试控制字符

头文件

ctype.h

语法

```
int isctrl(int c);
```

返回值

如果 `c` 是控制字符，返回非零值；否则返回零。

isdigit

功能

测试数字

头文件

ctype.h

语法

```
int isdigit(int c);
```

返回值

如果 `c` 是数字，返回非零值；否则返回零。





isgraph

功能

测试图形字符

头文件

ctype.h

语法

```
int isgraph(int c);
```

返回值

如果 `c` 是显示字符（除了空格），返回非零值；否则返回零。

islower

功能

测试小写字母

头文件

ctype.h

语法

```
int islower(int c);
```

返回值

如果 `c` 是小写字母，返回非零值；否则返回零。

isupper

功能

测试大写字母

头文件

ctype.h





语法

```
int isupper(int c);
```

返回值

如果 `c` 是大写字母，返回非零值；否则返回零。

itoa

功能

把整形数转换为字符串的函数

头文件

`stdlib.h`

语法

```
char *itoa(int value,char *string,int radix)
```

返回值

指向 `string` 的指针





K

kbhit

功能

检查当前按下的键的函数

头文件

conio.h

语法

```
int kbhit(void)
```

返回值

如果按键有效，返回一非零值，否则返回 0。

keep

功能

驻留并退出函数

头文件

dos.h

语法

```
void keep(unsigned char status,unsigned size)
```



L

labs

功能

长整数的绝对值

头文件

stdlib.h

语法

```
longint labs(long int j);
```

返回值

j 的绝对值。如果不能表示 j 的绝对值，那么函数的行为是未定义的。

ldexp

功能

联合小数和指数

头文件

math.h

语法

```
double ldexp(double x, int exp);
```

返回值

$x \times 2^{\text{exp}}$ 的值。可能会发生取值范围错误。



ldiv

功能

长整数除法

头文件

stdlib.h

语法

```
ldiv_t ldiv(long int numer, long int denom);
```

返回值

含有 quot（numer 除以 denom 的商）和 rem（余数）的结构。如果无法表示结果，那么函数的行为是未定义的。

localtime

功能

转换成区域时间

头文件

time.h

语法

```
struct tm *localtime(const time_t *timer);
```

返回值

指向结构的指针，此结构含有的分解时间等价于 timer 指向的日历时间。

log

功能

求 num 的自然对数值

头文件





math.h

语法

```
double log(double num)
```

log10

功能

求 num 以 10 为底的对数值

头文件

math.h

语法

```
double log10(double num)
```

longjmp

功能

非区域跳转

头文件

setjmp.h

语法

```
void longjmp(jmp_buf env, int val);
```

返回值

如果 val 非零，它将是 setjmp 的返回值；如果 val 为 1，则 setjmp 返回 1。

__lrotl

功能

将一个无符号长整形数左循环移位的函数

头文件





stdlib.h

语法

```
unsigned long _lrotl(unsigned long value,int count)
```

返回值

将 value 向左循环移动 count 位后的值。

__lrotr

功能

将一个无符号长整形数右循环移位的函数

头文件

stdlib.h

语法

```
unsigned long _lrotr(unsigned long value,int count)
```

返回值

将 value 向右循环移动 count 位后的值。

lseek

功能

移动文件指针函数

头文件

io.h

语法

```
long lseek(int handle,long offset,int fromwhere)
```

handle=open 函数的返回值(文件句柄);

fromwhere= 文件位置

(SEEK_SET(0)从文件头位置; SEEK_CUR(1)从当前文件指针位置; SEEK_END(2)从文件结尾位置)





返回值

指针新位置的偏移量（相对于文件开始处）,出错时返回-1L。

ltoa

功能

把长整形数转换为字符串的函数

头文件

stdlib.h

语法

```
char *ltoa(long value,char *string,int radix)
```

返回值

指向 string 的指针





M

malloc

功能

分配内存函数

头文件

alloc.h、stdlib.h

语法

```
void *malloc(size_t size)
```

返回值

返回新分配内存的地址，若无足够内存，返回 NULL。

mkdir

功能

创建目录函数

头文件

dir.h

语法

```
int mkdir(const char *path)
```

返回值

0(成功);-1(失败)





mblen

功能

计算多字节字符的长度

头文件

stdlib.h

语法

```
int mblen(const char *s, size_t n);
```

返回值

如果 *s* 是空指针，返回非零值还是零值依赖于多字节字符是否是依赖状态编码。如果 *s* 指向空字符则返回零；如果接下来 *n* 个或几个字节形成了一个有效的字符，那么返回 *s* 指向的多字节字符中的字节数量；否则返回 1。

memchr

功能

搜索内存块字符

头文件

string.h

语法

```
void *memchr(const void *s, int c, size_t n);
```

返回值

指向字符的指针，此字符是 *s* 所指向对象的前 *n* 个字符中第一个遇到的字符 *c*。如果没有找到 *c*，则返回空指针。

memcmp

功能

比较内存块





头文件

string.h

语法

```
int memcmp(const void *s1, const void *s2, size_t n);
```

返回值

负整数、零还是正整数依赖于 s1 所指向对象的前 n 个字符是小于、等于还是大于 s2 所指向对象的前 n 个字符。

memcpy

功能

复制内存块

头文件

string.h

语法

```
void *memcpy(void *s1, const void *s2, size_t n);
```

返回值

s1（指向目的的指针）。

memmove

功能

复制内存块

头文件

string.h

语法

```
void *memmove(void *s1, const void *s2, size_t n);
```

返回值





s1（指向目的的指针）。

memset

功能

初始化内存块

头文件

string.h

语法

```
void *memset(void *s, int c, size_t n);
```

返回值

（指向内存块的指针）。

mktemp

功能

建立一个唯一的文件名的函数

头文件

dir.h

语法

```
char *mktemp(char *template)
```

mk_fp

功能

设置一个远指针的宏

头文件

dos.h

语法





```
void far *MK_FP(unsigned seg,unsigned ofs)
```

返回值

远指针

modf

功能

分解成整数和小数部分

头文件

math.h

语法

```
double modf(double value, double *iptr);
```

返回值

value 的小数部分。

movedata

功能

拷贝数据函数

头文件

mem.h、string.h

语法

```
void movedata(unsigned srcseg,unsigned srcoff,unsigned dstseg,unsigned  
dstoff,size_t n)
```



O

`_open`

功能

打开一个文件进行读写的函数

头文件

`fcntl.h`、`io.h`

语法

```
int _open(path,oflags)
char *path;//已建立文件的路径名
int oflags;//方式
```

返回值

0(正常结束); -1(错误)

`open`

功能

打开文件进行读写的函数

头文件

`fcntl.h`、`io.h`

语法

```
int open(const char *path,int access[,unsigned mode])
```

返回值

文件句柄（成功时）；-1(失败时)。成功时，文件指针指向文件头。



outport

功能

输出一个字到端口中的函数

头文件

dos.h

语法

```
void outport(int portid,int value)
```

outtextxy

功能

在指定位置显示一字符串的函数

头文件

graphics.h

语法

```
void far outtextxy(int x,int y,char far *textstring)
```





P

peek

功能

返回指定内存中字的函数

头文件

dos.h

语法

```
int peek(int seg,unsigned offset)
```

perror

功能

显示错误信息

头文件

stdio.h

语法

```
void perror(const char *s);
```

pow

功能

求以 b 为底的 exp 次幂的值

头文件





math.h

语法

```
double pow(double b,double exp)
```

printf

功能

写格式化输出到 stdout 的函数

头文件

stdio.h

语法

```
int printf(const char *format, ...);
```

返回值

写入的字符数量。如果发生错误就返回负值。

putch

功能

向屏幕输出字符的函数

头文件

conio.h

语法

```
putch('转义字符');  
putch('单个字符');  
putch(字符变量);
```

putchar

功能





在 stdout 上输出字符的宏

头文件

stdio.h

语法

```
int putchar(int c)
```

返回值

成功返回字符 c,失败返回 EOF。

putenv

功能

将字符串放入当前环境中的函数

头文件

stdlib.h

语法

```
int putenv(const char *name)
```

返回值

0(成功);-1(失败)

putimage

功能

重新写屏函数(输出一个位图象到图形屏幕上的函数)

头文件

graphics.h

语法

```
void far putimage(int left,int top,void far *buf,int op)
```





putpixel

功能

写像素点函数

头文件

graphics.h

语法

```
void far putpixel(int x,int y,int color)
```

puts

功能

输出一字符串到 stdout(标准输出)的函数

头文件

stdio.h

语法

```
puts("字符串");  
puts(字符串变量名即字符串变量的地址)
```

Q

qsort

功能

排序数组

头文件





stdlib.h

语法

```
void qsort(void *base, size_t memb, size_t size, int (*compar)(const void *, const void *));
```





R

`_read`

功能

读文件函数

头文件

`io.h`

语法

```
int _read(handle,buffer,count)
int handle;//文件句柄
void *buffer;//存放读出数据的缓冲区
unsigned count;//一次读出的字节数
```

返回值

0-0xfffe(实际读出的字节数); -1(错误)

`read`

功能

读文件函数

头文件

`io.h`

语法

```
int read(int handle,void *buf,unsigned len)
```



返回值

读入缓冲区的字节数，如果文件以文本方式打开，则不包括回车符和 Ctrl-Z 字符。

realloc

功能

重新分配内存函数

头文件

stdlib.h

语法

```
void *realloc(void *block,size_t size)  
block 指向用 malloc、calloc 或 realloc 已得到的内存。  
size 是重新分配的字节。
```

返回值

重新分配的块地址。若不能重分配，则返回 NULL。

rectangle

功能

画一个矩形的函数

头文件

graphics.h

语法

```
void far rectangle(int left,int top,int right,int bottom)
```

remove

功能

删除一个文件的函数



头文件

stdio.h

语法

```
int remove(const char *filename)
```

返回值

0(成功); -1(失败)

rename

功能

文件改名函数

头文件

stdio.h

语法

```
int rename(const char *oldname,const char *newname)
```

返回值

0(成功);-1(失败)

restorecrtmode

功能

恢复屏幕视频模式为调用 initgraph 前的设置的函数

头文件

graphics.h

语法

```
void far restorecrtmode(void)
```





rewind

功能

返回到文件头

头文件

stdio.h

语法

```
void rewind(FILE *stream);
```

rmdir

功能

删除目录函数

头文件

dir.h

语法

```
int rmdir(const char *path)
```

返回值

0(成功);-1(操作出错)

__rotl

功能

将一个无符号整形数左循环移位的函数

头文件

stdlib.h

语法

```
unsigned __rotl(unsigned value,int count)
```





返回值

将 `value` 向左循环移动 `count` 位后的值。

_rotr

功能

将一个无符号整形数右循环移位的函数

头文件

`stdlib.h`

语法

```
unsigned _rotr(unsigned value,int count)
```

返回值

将 `value` 向右循环移动 `count` 位后的值。





S

scanf

功能

格式化输入函数

头文件

stdio.h

语法

```
scanf("输入格式",输入项系列)
```

```
scanf("%f",&实型变量)
```

```
scanf("%c",&字符变量)
```

```
scanf("%s",字符串变量)
```

```
scanf("%d",&整型变量)
```

注：需先定义以上各种变量如 `char *字符串变量`；

searchpath

功能

按 dos 路径查找一个文件的函数

头文件

dir.h

语法

```
char *searchpath(const char *file)
```

返回值





指向完整路径名字符串的指针。定位失败返回 NULL。

setactivepage

功能

设置图形输出活动页的函数

头文件

graphics.h

语法

```
void far setactivepage(int page)
```

setbuf

功能

设置缓冲区

头文件

stdio.h

语法

```
void setbuf(FILE *stream, char *buf);
```

setcolor

功能

设置当前要画的线颜色的函数

头文件

graphics.h

语法

```
void far setcolor(int color)
```





setdisk

功能

设置当前驱动器的函数

头文件

dir.h

语法

```
int setdisk(int drive)
```

setgraphmode

功能

将系统设置成图形模式并清屏的函数

头文件

graphics.h

语法

```
void far setgraphmode(int mode)
```

返回值

如果给了一个对当前驱动程序来说无效的模式，则返回-10。

setjmp

功能

准备非局部跳转

头文件

setjmp.h

语法

```
int setjmp(jmp_buf env);
```





返回值

当直接调用时，返回为零。当从 `longjmp` 函数调用中返回时，返回非零值。

setlinestyle

功能

设置当前画线宽度和类型的函数

头文件

`graphics.h`

语法

```
void far setlinestyle(int linestyle,unsigned upattern,int thickness)
```

其中：`linestyle=0` 为实线；`1` 为点线；`2` 为中心线；`3` 为破折线；`4` 为用户定义。
`thickness=1` 为一个像素宽；`3` 为三个像素宽。

只有当 `linestyle=4` 时，`upattern` 才起作用。

setlocale

功能

设置地区

头文件

`locale.h`

语法

```
char *setlocale(int category, const char *locale);
```

返回值

如果 `locale` 是空指针，就返回一个指向与当前地区的 `category` 相关的字符串的指针。
否则，返回一个指向与新地区的 `category` 相关的字符串的指针。如果操作失败，则返回空指针。





setvbuf

功能

设置缓冲区

头文件

stdio.h

语法

```
int setvbuf(FILE *stream, char *buf, int mode, size_t size);
```

返回值

如果操作成功，就返回零。如果 mode 无效或者无法满足要求，则返回非零值。

setvisualpage

功能

设置可见的图形页号的函数

头文件

graphics.h

语法

```
void far setvisualpage(int page)
```

signal

功能

安装信号处理函数

头文件

signal.h

语法

```
void (*signal(int sig, void (*func)(int)))(int);
```



返回值

指向此信号前一个处理函数的指针。如果无法安装处理函数，则返回 SIG_ERR。

sin

功能

计算 arg(弧度)的正弦值

头文件

math.h

语法

```
double sin(double arg)
```

sinh

功能

双曲正弦

头文件

math.h

语法

```
double sinh(double x);
```

返回值

x 的双曲正弦值（按照弧度衡量的）。如果 x 的数过大，那么可能会发生取值范围错误。

sprintf

功能

格式化输出到数组的函数

头文件



stdio.h

语法

```
int sprintf(buf,format,arg_list)
```

返回值

实际写入数组的字符个数。

sqrt

功能

平方根

头文件

math.h

语法

```
double sqrt(double x);
```

返回值

x 的平方根。如果 x 是负数，则会发生定义域错误。

srand

功能

启动伪随机数产生器

头文件

stdlib.h

语法

```
void srand(unsigned int seed);
```

sscanf

功能





格式串读

头文件

stdio.h

语法

```
int sscanf(const char *s, const char *format, ...);
```

返回值

成功读入并且存储的数据项数量。如果在可以读入任意数据项之前到达了字符串末尾，就返回 EOF。

strcmp

功能

比较字符串

头文件

string.h

语法

```
int strcmp(const char *s1, const char *s2);
```

返回值

负数、零还是正整数，依赖于 s1 所指向的字符串是小于、等于还是大于 s2 所指的字符串。

strcpy

功能

字符串复制

头文件

string.h

语法





```
char *strcpy(char *s1, const char *s2);
```

返回值

s1（指向目的指针）。

strcspn

功能

搜索集合中不在初始范围内的字符串

头文件

string.h

语法

```
size_t strcspn(const char *s1, const char *s2);
```

返回值

最长的初始字符段的长度，此初始字符段由 s1 指向的，但是不包含 s2 指向的字符串中的任何字符。

strerror

功能

把错误数转换成为字符串

头文件

string.h

语法

```
char *strerror(int errnum);
```

返回值

指向字符串的指针，此字符串含有的出错消息对应 errnum 的值。



strlen

功能

字符串长度

头文件

string.h

语法

```
size_t strlen(const char *s);
```

返回值

s 指向的字符串长度，不包括空字符。 13.5 节、23.5 节

strncat

功能

有限制的字符串的连接

头文件

string.h

语法

```
char *strncat(char *s1, const char *s2, size_t n);
```

返回值

s1（指向连接后字符串的指针）。

strncmp

功能

有限制的字符串比较

头文件

string.h





语法

```
int strncmp(const char *s1, const char *s2, size_t n);
```

返回值

负整数、零还是正整数，依赖于 s1 所指向的数组的前 n 个字符是小于、等于还是大于 s2 所指向的数组的前 n 个字符。如果在其中某个数组中遇到空字符，比较都会停止。

strcpy

功能

有限制的字符串复制

头文件

string.h

语法

```
char *strcpy(char *s1, const char *s2, size_t n);
```

返回值

s1（指向目的的指针）。

strchr

功能

反向搜索字符串中字符

头文件

string.h

语法

```
char *strchr(const char *s, int c);
```

返回值

指向字符的指针，此字符是 s 所指向字符串中最后一个遇到的字符 c。如果没有找到 c，则返回空指针。





strstr

功能

搜索子字符串

头文件

string.h

语法

```
char *strstr(const char *s1, const char *s2);
```

返回值

指针，此指针指向 s1 字符串中的字符第一次出现在 s2 字符串中的位置。如果没有发现匹配，就返回空指针。

strtok

功能

搜索字符串记号

头文件

string.h

语法

```
char *strtok(char *s1, const char *s2);
```

返回值

指向记号的第一个字符的指针。如果没有发现记号，就返回空指针。

strtoul

功能

把字符串转换成无符号长整数

头文件





stdlib.h

语法

```
unsigned long int strtoul(const char *nptr, char **endptr, int base);
```

返回值

转换的数。如果没有转换可以执行，则返回零。如果无法表示数，则返回 ULONG_MAX。

system

功能

执行操作系统命令

头文件

stdlib.h

语法

```
int system(const char *string);
```

返回值

当 `string` 是空指针时，如果命令处理器有效，则返回非零值。如果 `string` 不是空指针，则返回由实现定义的值。





T

tan

功能

计算 \arg (弧度)的正切值

头文件

math.h

语法

```
double tan(double arg)
```

time

功能

当前时间

头文件

time.h

语法

```
time_t time(time_t *timer);
```

返回值

当前的日历时间。如果日历时间无效，则返回(time_t)-1。如果 timer 不是空指针，也把返回值存储到 timer 指向的对象中。





tmpnam

功能

产生临时文件名

头文件

stdio.h

语法

```
char *tmpnam(char *s);
```

返回值

指向文件名的指针。

tolower

功能

转换成小写字母

头文件

ctype.h

语法

```
int tolower(int c);
```

返回值

如果 c 是大写字母，则返回相应的小写字母。如果 c 不是大写字母，则返回无变化的 c。

toupper

功能

转换成大写字母

头文件

ctype.h



语法

```
int toupper(int c);
```

返回值

如果 `c` 是小写字母，则返回相应的大写字母。如果 `c` 不是小写字母，则返回无变化的 `c`。



U

ungetc

功能

未读取的字符

头文件

stdio.h

语法

```
int ungetc(int c, FILE *stream);
```

返回值

c（回退的字符）。如果没有读取操作或者文件定位操作就试图回退过多的字符，那么函数将会返回 EOF。





V

vfprintf

功能

用可变实际参数列表格式化写文件

头文件

stdio.h

语法

```
int vfprintf(FILE *stream, const char *format, va_list arg);
```

返回值

写入的字符数量。如果发生错误就返回负值。

vprintf

功能

用可变实际参数列表格式化写

头文件

stdio.h

语法

```
int vprintf(const char *format, va_list arg);
```

返回值

写入的字符数量。如果发生错误就返回负值。





vsprintf

功能

用可变实际参数列表格式化写字符串

头文件

stdio.h

语法

```
int vsprintf(char *s, const char *format, va_list arg);
```

返回值

存储的字符数量，但不计空字符。





W

wcstombs

功能

把宽字符串转换成多字节字符串

头文件

stdlib.h

语法

```
size_t wcstombs(char *s, const wchar_t *pwcs, size_t n);
```

返回值

存储的字节数，不包括空字符。如果遇到一个代码不对应有效多字节字符时，则返回 (size_t) -1。

wctomb

功能

把宽字符转换成多字节字符

头文件

stdlib.h

语法

```
int wctomb(char *s, wchar_t wchar);
```

返回值

如果 s 是空指针，则返回非零值或零值，这依赖于多字节字符是否是依赖状态编码的。





如果 `wchar` 对应一个有效的多字节字符，则返回字符中字节的数量，如果不是这样，则返回 -1。

`_write`

功能

写文件函数

头文件

`io.h`

语法

```
int _write(handle,buffer,count)
int handle;//文件的句柄
void *buffer;//存放数据的缓冲区
unsigned count;//写入的字节数
```

返回值

0-0xffff(实际写入的字节数);-1(错误)

`write`

功能

写文件函数

头文件

`io.h`

语法

```
int write(int handle,char *buf,unsigned len)
```

返回值

实际写入的字节数(不包括回车符),出错时返回-1。

