============================

函数名: abort

============================

功　能: 异常终止一个进程
用　法: void abort(void);
程序例:
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
  printf("Calling abort()\n");
  abort();
  return 0; /* This is never reached */
}

============================

函数名: abs

============================

功　能: 求整数的绝对值
用　法: int abs(int i);
程序例:
#include <stdio.h>
#include <math.h>

int main(void)
{
  int number = -1234;

  printf("number: %d absolute value: %d\n", number, abs(number));
  return 0;
}

============================

函数名: absread, abswirte

============================

功　能: 绝对磁盘扇区读、写数据
用　法: int absread(int drive, int nsects, int sectno, void *buffer);
 int abswrite(int drive, int nsects, in tsectno, void *buffer);
程序例:

```
/* absread example */

#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <dos.h>

int main(void)
{
  int i, strt, ch_out, sector;
  char buf[512];

  printf("Insert a diskette into drive A and press any key\n");
  getch();
  sector = 0;
  if (absread(0, 1, sector, &buf) != 0)
  {
    perror("Disk problem");
    exit(1);
  }
  printf("Read OK\n");
  strt = 3;
  for (i=0; i<80; i++)
  {
    ch_out = buf[strt+i];
    putchar(ch_out);
  }
  printf("\n");
  return(0);
}
```

============================

函数名: access

============================
功　能: 确定文件的访问权限
用　法: int access(const char *filename, int amode);
程序例:
```
#include <stdio.h>
#include <io.h>

int file_exists(char *filename);

int main(void)
```

```
{
  printf("Does NOTEXIST.FIL exist: %s\n",
  file_exists("NOTEXISTS.FIL") ? "YES" : "NO");
  return 0;
}

int file_exists(char *filename)
{
  return (access(filename, 0) == 0);
}
```

==========================

函数名: acos

==========================

功　能: 反余弦函数
用　法: double acos(double x);
程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
  double result;
  double x = 0.5;

  result = acos(x);
  printf("The arc cosine of %lf is %lf\n", x, result);
  return 0;
}
```

==========================

函数名: allocmem

==========================

功　能: 分配 DOS 存储段
用　法: int allocmem(unsigned size, unsigned *seg);
程序例:

```
#include <dos.h>
#include <alloc.h>
#include <stdio.h>

int main(void)
{
```

```
  unsigned int size, segp;
  int stat;

  size = 64; /* (64 x 16) = 1024 bytes */
  stat = allocmem(size, &segp);
  if (stat == -1)
     printf("Allocated memory at segment: %x\n", segp);
  else
     printf("Failed: maximum number of paragraphs available is %u\n",
             stat);

  return 0;
}
```

===========================

函数名: arc

===========================

功 能: 画一弧线
用 法: void far arc(int x, int y, int stangle, int endangle, int radius);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int midx, midy;
   int stangle = 45, endangle = 135;
   int radius = 100;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();    /* an error occurred */
   if (errorcode != grOk)
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
```

```
        getch();

        exit(1);     /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;
    setcolor(getmaxcolor());

    /* draw arc */
    arc(midx, midy, stangle, endangle, radius);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: asctime

============================

功　能: 转换日期和时间为 ASCII 码
用　法: char *asctime(const struct tm *tblock);
程序例:

```
#include <stdio.h>
#include <string.h>
#include <time.h>

int main(void)
{
    struct tm t;
    char str[80];

    /* sample loading of tm structure  */

    t.tm_sec    = 1;   /* Seconds */
    t.tm_min    = 30;  /* Minutes */
    t.tm_hour   = 9;   /* Hour */
    t.tm_mday   = 22;  /* Day of the Month  */
    t.tm_mon    = 11;  /* Month */
    t.tm_year   = 56;  /* Year - does not include century */
    t.tm_wday   = 4;   /* Day of the week  */
```

```
    t.tm_yday   = 0;   /* Does not show in asctime  */
    t.tm_isdst  = 0;   /* Is Daylight SavTime; does not show in asctime */

    /* converts structure to null terminated
    string */

    strcpy(str, asctime(&t));
    printf("%s\n", str);

    return 0;
}
```

============================

函数名: asin

============================

功　能: 反正弦函数
用　法: double asin(double x);
程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double result;
    double x = 0.5;

    result = asin(x);
    printf("The arc sin of %lf is %lf\n", x, result);
    return(0);
}
```

============================

函数名: assert

============================

功　能: 测试一个条件并可能使程序终止
用　法: void assert(int test);
程序例:

```
#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
```

```
struct ITEM {
    int key;
    int value;
};

/* add item to list, make sure list is not null */
void additem(struct ITEM *itemptr) {
    assert(itemptr != NULL);
    /* add item to list */
}

int main(void)
{
    additem(NULL);
    return 0;
}
```

============================

函数名: atan

============================

功　能: 反正切函数
用　法: double atan(double x);
程序例:
```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double result;
    double x = 0.5;

    result = atan(x);
    printf("The arc tangent of %lf is %lf\n", x, result);
    return(0);
}
```

============================

函数名: atan2

============================

功　能：计算 Y/X 的反正切值
用　法：double atan2(double y, double x);
程序例：
#include <stdio.h>
#include <math.h>

```c
int main(void)
{
    double result;
    double x = 90.0, y = 45.0;

    result = atan2(y, x);
    printf("The arc tangent ratio of %lf is %lf\n", (y / x), result);
    return 0;
}
```

============================

函数名: atexit

============================

功　能：注册终止函数
用　法：int atexit(atexit_t func);
程序例：
#include <stdio.h>
#include <stdlib.h>

```c
void exit_fn1(void)
{
    printf("Exit function #1 called\n");
}

void exit_fn2(void)
{
    printf("Exit function #2 called\n");
}

int main(void)
{
    /* post exit function #1 */
    atexit(exit_fn1);
    /* post exit function #2 */
    atexit(exit_fn2);
    return 0;
```

```
}


============================
```

函数名: atof

```
============================
功　能: 把字符串转换成浮点数
用　法: double atof(const char *nptr);
程序例:
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    float f;
    char *str = "12345.67";

    f = atof(str);
    printf("string = %s float = %f\n", str, f);
    return 0;
}


============================
```

函数名: atoi

```
============================
功　能: 把字符串转换成长整型数
用　法: int atoi(const char *nptr);
程序例:
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    int n;
    char *str = "12345.67";
    n = atoi(str);
    printf("string = %s integer = %d\n", str, n);
    return 0;
}


============================
```

函数名: atol

============================

功　能: 把字符串转换成长整型数
用　法: long atol(const char *nptr);
程序例:

```c
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    long l;
    char *str = "98765432";
    l = atol(lstr);
    printf("string = %s integer = %ld\n", str, l);
    return(0);
}
```

============================

函数名: bar

============================

功　能: 画一个二维条形图
用　法: void far bar(int left, int top, int right, int bottom);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy, i;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
```

```c
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* loop through the fill patterns */
    for (i=SOLID_FILL; i<USER_FILL; i++)
    {
        /* set the fill style */
        setfillstyle(i, getmaxcolor());

        /* draw the bar */
        bar(midx-50, midy-50, midx+50,
            midy+50);

        getch();
    }

    /* clean up */
    closegraph();
    return 0;
}
```

==============================

函数名: bar3d

==============================

功　能: 画一个三维条形图
用　法: void far bar3d(int left, int top, int right, int bottom,
                      int depth, int topflag);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
```

```c
int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy, i;

    /* initialize graphics, local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* loop through the fill patterns */
    for (i=EMPTY_FILL; i<USER_FILL; i++)
    {
        /* set the fill style */
        setfillstyle(i, getmaxcolor());

        /* draw the 3-d bar */
        bar3d(midx-50, midy-50, midx+50, midy+50, 10, 1);

        getch();
    }

    /* clean up */
    closegraph();
    return 0;
}
```

============================

函数名: bdos

============================

功　能: DOS 系统调用

用　法: int bdos(int dosfun, unsigned dosdx, unsigned dosal);

程序例:

```c
#include <stdio.h>
#include <dos.h>

/* Get current drive as 'A', 'B', ... */
char current_drive(void)
{
    char curdrive;

    /* Get current disk as 0, 1, ... */
    curdrive = bdos(0x19, 0, 0);
    return('A' + curdrive);
}

int main(void)
{
    printf("The current drive is %c:\n", current_drive());
    return 0;
}
```

============================

函数名: bdosptr

============================

功　能: DOS 系统调用

用　法: int bdosptr(int dosfun, void *argument, unsigned dosal);

程序例:

```c
#include <string.h>
#include <stdio.h>
#include <dir.h>
#include <dos.h>
#include <errno.h>
#include <stdlib.h>

#define  BUFLEN  80

int main(void)
{
    char  buffer[BUFLEN];
```

```
    int    test;

    printf("Enter full pathname of a directory\n");
    gets(buffer);

    test = bdosptr(0x3B,buffer,0);
        if(test)
        {
    printf("DOS error message: %d\n", errno);
    /* See errno.h for error listings */
    exit (1);
        }

    getcwd(buffer, BUFLEN);
    printf("The current directory is: %s\n", buffer);

    return 0;
}
```

===========================

函数名: bioscom

===========================

功　能: 串行 I/O 通信
用　法: int bioscom(int cmd, char abyte, int port);
程序例:

```
#include <bios.h>
#include <conio.h>

#define COM1        0
#define DATA_READY 0x100
#define TRUE        1
#define FALSE       0

#define SETTINGS ( 0x80 | 0x02 | 0x00 | 0x00)

int main(void)
{
    int in, out, status, DONE = FALSE;

    bioscom(0, SETTINGS, COM1);
    cprintf("... BIOSCOM [ESC] to exit ...\n");
```

```
   while (!DONE)
    {
       status = bioscom(3, 0, COM1);
       if (status & DATA_READY)
   if ((out = bioscom(2, 0, COM1) & 0x7F) != 0)
      putch(out);
   if (kbhit())
   {
      if ((in = getch()) == '\x1B')
         DONE = TRUE;
      bioscom(1, in, COM1);
   }
    }
    return 0;
}
```

============================

函数名: biosdisk

============================

功 能: 软硬盘 I/O
用 法: int biosdisk(int cmd, int drive, int head, int track, int sector
        int nsects, void *buffer);
程序例:

```
#include <bios.h>
#include <stdio.h>

int main(void)
{
   int result;
   char buffer[512];

   printf("Testing to see if drive a: is ready\n");
   result = biosdisk(4,0,0,0,0,1,buffer);
   result &= 0x02;
   (result) ? (printf("Drive A: Ready\n")) :
       (printf("Drive A: Not Ready\n"));

   return 0;
}
```

============================

函数名: biosequip

============================

功　能: 检查设备
用　法: int biosequip(void);
程序例:

```c
#include <bios.h>
#include <stdio.h>

int main(void)
{
    int result;
    char buffer[512];

    printf("Testing to see if drive a: is ready\n");
    result = biosdisk(4,0,0,0,0,1,buffer);
    result &= 0x02;
    (result) ? (printf("Drive A: Ready\n")) :
        (printf("Drive A: Not Ready\n"));

    return 0;
}
```


============================

函数名: bioskey

============================

功　能: 直接使用 BIOS 服务的键盘接口
用　法: int bioskey(int cmd);
程序例:

```c
#include <stdio.h>
#include <bios.h>
#include <ctype.h>

#define RIGHT   0x01
#define LEFT    0x02
#define CTRL    0x04
#define ALT     0x08

int main(void)
```

```
{
   int key, modifiers;

   /* function 1 returns 0 until a key is pressed */
   while (bioskey(1) == 0);

   /* function 0 returns the key that is waiting */
   key = bioskey(0);

   /* use function 2 to determine if shift keys were used */
   modifiers = bioskey(2);
   if (modifiers)
   {
      printf("[");
      if (modifiers & RIGHT) printf("RIGHT");
      if (modifiers & LEFT)  printf("LEFT");
      if (modifiers & CTRL)  printf("CTRL");
      if (modifiers & ALT)   printf("ALT");
      printf("]");
   }
   /* print out the character read */
   if (isalnum(key & 0xFF))
      printf("'%c'\n", key);
   else
      printf("%#02x\n", key);
   return 0;
}
```

============================

函数名: biosmemory

============================

功　能: 返回存储块大小
用　法: int biosmemory(void);
程序例:

```
#include <stdio.h>
#include <bios.h>

int main(void)
{
   int memory_size;

   memory_size = biosmemory();  /* returns value up to 640K */
```

```
    printf("RAM size = %dK\n",memory_size);
    return 0;
}
```

==============================

函数名: biosprint

==============================

功　能: 直接使用 BIOS 服务的打印机 I/O
用　法: int biosprint(int cmd, int byte, int port);
程序例:

```
#include <stdio.h>
#include <conio.h>
#include <bios.h>

int main(void)
{
    #define STATUS  2    /* printer status command */
    #define PORTNUM 0    /* port number for LPT1 */

    int status, abyte=0;

    printf("Please turn off your printer.  Press any key to continue\n");
    getch();
    status = biosprint(STATUS, abyte, PORTNUM);
    if (status & 0x01)
       printf("Device time out.\n");
    if (status & 0x08)
       printf("I/O error.\n");

    if (status & 0x10)
       printf("Selected.\n");
    if (status & 0x20)
       printf("Out of paper.\n");

    if (status & 0x40)
       printf("Acknowledge.\n");
    if (status & 0x80)
       printf("Not busy.\n");

    return 0;
}
```

==========================

函数名: biostime

==========================
功　能: 读取或设置 BIOS 时间
用　法: long biostime(int cmd, long newtime);
程序例:

```c
#include <stdio.h>
#include <bios.h>
#include <time.h>
#include <conio.h>

int main(void)
{
   long bios_time;

   clrscr();
   cprintf("The number of clock ticks since midnight is:\r\n");
   cprintf("The number of seconds since midnight is:\r\n");
   cprintf("The number of minutes since midnight is:\r\n");
   cprintf("The number of hours since midnight is:\r\n");
   cprintf("\r\nPress any key to quit:");
   while(!kbhit())
   {
      bios_time = biostime(0, 0L);

      gotoxy(50, 1);
      cprintf("%lu", bios_time);

      gotoxy(50, 2);
      cprintf("%.4f", bios_time / CLK_TCK);

      gotoxy(50, 3);
      cprintf("%.4f", bios_time / CLK_TCK / 60);

      gotoxy(50, 4);
      cprintf("%.4f", bios_time / CLK_TCK / 3600);
   }
   return 0;
}
```

===========================

函数名: brk

===========================

功　能: 改变数据段空间分配
用　法: int brk(void *endds);
程序例:

```
#include <stdio.h>
#include <alloc.h>

int main(void)
{
   char *ptr;

   printf("Changing allocation with brk()\n");
   ptr = malloc(1);
   printf("Before brk() call: %lu bytes free\n", coreleft());
   brk(ptr+1000);
   printf(" After brk() call: %lu bytes free\n", coreleft());
   return 0;
}
```

===========================

函数名: bsearch

===========================

功　能: 二分法搜索
用　法: void *bsearch(const void *key, const void *base, size_t *nelem,
        size_t width, int(*fcmp)(const void *, const *));
程序例:

```
#include <stdlib.h>
#include <stdio.h>

#define NELEMS(arr) (sizeof(arr) / sizeof(arr[0]))

int numarray[] = {123, 145, 512, 627, 800, 933};

int numeric (const int *p1, const int *p2)
{
   return(*p1 - *p2);
}
```

```c
int lookup(int key)
{
    int *itemptr;

    /* The cast of (int(*)(const void *,const void*))
       is needed to avoid a type mismatch error at
       compile time */
    itemptr = bsearch (&key, numarray, NELEMS(numarray),
        sizeof(int), (int(*)(const void *,const void *))numeric);
    return (itemptr != NULL);
}

int main(void)
{
    if (lookup(512))
        printf("512 is in the table.\n");
    else
        printf("512 isn't in the table.\n");

    return 0;
}
```

============================

函数名: cabs

============================

功　能: 计算复数的绝对值
用　法: double cabs(struct complex z);
程序例:

```c
#include <stdio.h>
#include <math.h>

int main(void)
{
    struct complex z;
    double val;

    z.x = 2.0;
    z.y = 1.0;
    val = cabs(z);
```

```
    printf("The absolute value of %.2lfi %.2lfj is %.2lf", z.x, z.y, val);
    return 0;
}
```

==========================

函数名: calloc

==========================

功　能: 分配主存储器
用　法: void *calloc(size_t nelem, size_t elsize);
程序例:

```
#include <stdio.h>
#include <alloc.h>

int main(void)
{
    char *str = NULL;

    /* allocate memory for string */
    str = calloc(10, sizeof(char));

    /* copy "Hello" into string */
    strcpy(str, "Hello");

    /* display string */
    printf("String is %s\n", str);

    /* free memory */
    free(str);

    return 0;
}
```

==========================

函数名: ceil

==========================

功　能: 向上舍入
用　法: double ceil(double x);
程序例:

```c
#include <math.h>
#include <stdio.h>

int main(void)
{
    double number = 123.54;
    double down, up;

    down = floor(number);
    up = ceil(number);

    printf("original number     %5.2lf\n", number);
    printf("number rounded down %5.2lf\n", down);
    printf("number rounded up   %5.2lf\n", up);

    return 0;
}
```

============================

函数名: cgets

============================

功　能: 从控制台读字符串
用　法: char *cgets(char *str);
程序例:

```c
#include <stdio.h>
#include <conio.h>

int main(void)
{
    char buffer[83];
    char *p;

    /* There's space for 80 characters plus the NULL terminator */
    buffer[0] = 81;

    printf("Input some chars:");
    p = cgets(buffer);
    printf("\ncgets read %d characters: \"%s\"\n", buffer[1], p);
    printf("The returned pointer is %p, buffer[0] is at %p\n", p, &buffer);
```

```c
    /* Leave room for 5 characters plus the NULL terminator */
    buffer[0] = 6;

    printf("Input some chars:");
    p = cgets(buffer);
    printf("\ncgets read %d characters: \"%s\"\n", buffer[1], p);
    printf("The returned pointer is %p, buffer[0] is at %p\n", p, &buffer);
    return 0;
}
```

============================

函数名: chdir

============================

功　能: 改变工作目录
用　法: int chdir(const char *path);
程序例:

```c
#include <stdio.h>
#include <stdlib.h>
#include <dir.h>

char old_dir[MAXDIR];
char new_dir[MAXDIR];

int main(void)
{
    if (getcurdir(0, old_dir))
    {
        perror("getcurdir()");
        exit(1);
    }
    printf("Current directory is: \\%s\n", old_dir);

    if (chdir("\\"))
    {
        perror("chdir()");
        exit(1);
    }

    if (getcurdir(0, new_dir))
    {
        perror("getcurdir()");
```

```
      exit(1);
   }
   printf("Current directory is now: \\%s\n", new_dir);

   printf("\nChanging back to orignal directory: \\%s\n", old_dir);
   if (chdir(old_dir))
   {
      perror("chdir()");
      exit(1);
   }

   return 0;
}
```

==========================

函数名: _chmod, chmod

============================

功　能: 改变文件的访问方式
用　法: int chmod(const char *filename, int permiss);
程序例:

```
#include <sys\stat.h>
#include <stdio.h>
#include <io.h>

void make_read_only(char *filename);

int main(void)
{
   make_read_only("NOTEXIST.FIL");
   make_read_only("MYFILE.FIL");
   return 0;
}

void make_read_only(char *filename)
{
   int stat;

   stat = chmod(filename, S_IREAD);
   if (stat)
      printf("Couldn't make %s read-only\n", filename);
   else
```

```
        printf("Made %s read-only\n", filename);
}
```

==============================

函数名: chsize

==============================

功　能: 改变文件大小
用　法: int chsize(int handle, long size);
程序例:

```
#include <string.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
    int handle;
    char buf[11] = "0123456789";

    /* create text file containing 10 bytes */
    handle = open("DUMMY.FIL", O_CREAT);
    write(handle, buf, strlen(buf));

    /* truncate the file to 5 bytes in size */
    chsize(handle, 5);

    /* close the file */
    close(handle);
    return 0;
}
```

==============================

函数名: circle

==============================

功　能: 在给定半径以(x, y)为圆心画圆
用　法: void far circle(int x, int y, int radius);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
```

```
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int midx, midy;
   int radius = 100;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   midx = getmaxx() / 2;
   midy = getmaxy() / 2;
   setcolor(getmaxcolor());

   /* draw the circle */
   circle(midx, midy, radius);

   /* clean up */
   getch();
   closegraph();
   return 0;
}
```

============================

函数名: cleardevice

============================

功　能: 清除图形屏幕
用　法: void far cleardevice(void);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int midx, midy;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   midx = getmaxx() / 2;
   midy = getmaxy() / 2;
   setcolor(getmaxcolor());

   /* for centering screen messages */
   settextjustify(CENTER_TEXT, CENTER_TEXT);

   /* output a message to the screen */
   outtextxy(midx, midy, "press any key to clear the screen:");

   /* wait for a key */
   getch();

   /* clear the screen */
   cleardevice();

   /* output another message */
   outtextxy(midx, midy, "press any key to quit:");

   /* clean up */
```

```
   getch();
   closegraph();
   return 0;
}
```

===========================

函数名: clearerr

===========================

功　能: 复位错误标志
用　法:void clearerr(FILE *stream);
程序例:

```c
#include <stdio.h>

int main(void)
{
   FILE *fp;
   char ch;

   /* open a file for writing */
   fp = fopen("DUMMY.FIL", "w");

   /* force an error condition by attempting to read */
   ch = fgetc(fp);
   printf("%c\n",ch);

   if (ferror(fp))
   {
      /* display an error message */
      printf("Error reading from DUMMY.FIL\n");

      /* reset the error and EOF indicators */
      clearerr(fp);
   }

   fclose(fp);
   return 0;
}
```

===========================

函数名: **clearviewport**

============================

功　能: 清除图形视区
用　法: void far clearviewport(void);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

#define CLIP_ON 1   /* activates clipping in viewport */

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int ht;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   setcolor(getmaxcolor());
   ht = textheight("W");

   /* message in default full-screen viewport */
   outtextxy(0, 0, "* <-- (0, 0) in default viewport");

   /* create a smaller viewport */
   setviewport(50, 50, getmaxx()-50, getmaxy()-50, CLIP_ON);

   /* display some messages */
   outtextxy(0, 0, "* <-- (0, 0) in smaller viewport");
   outtextxy(0, 2*ht, "Press any key to clear viewport:");
```

```
    /* wait for a key */
    getch();

    /* clear the viewport */
    clearviewport();

    /* output another message */
    outtextxy(0, 0, "Press any key to quit:");

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: _close, close

============================

功　能: 关闭文件句柄
用　法: int close(int handle);
程序例:

```
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>

main()
{
    int handle;
    char buf[11] = "0123456789";

    /* create a file containing 10 bytes */
    handle = open("NEW.FIL", O_CREAT);
    if (handle > -1)
    {
        write(handle, buf, strlen(buf));

        /* close the file */
        close(handle);
    }
```

```
    else
    {
        printf("Error opening file\n");
    }
    return 0;
}
```

============================

函数名: clock

============================

功　能: 确定处理器时间
用　法: clock_t clock(void);
程序例:

```
#include <time.h>
#include <stdio.h>
#include <dos.h>

int main(void)
{
    clock_t start, end;
    start = clock();

    delay(2000);

    end = clock();
    printf("The time was: %f\n", (end - start) / CLK_TCK);

    return 0;
}
```

============================

函数名: closegraph

============================

功　能: 关闭图形系统
用　法: void far closegraph(void);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
```

```c
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int x, y;

    /* initialize graphics mode */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();

    if (errorcode != grOk)  /* an error
       occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    x = getmaxx() / 2;
    y = getmaxy() / 2;

    /* output a message */
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(x, y, "Press a key to close the graphics system:");

    /* wait for a key */
    getch();

    /* closes down the graphics system */
    closegraph();

    printf("We're now back in text mode.\n");
    printf("Press any key to halt:");
    getch();
    return 0;
}
```

============================

函数名: clreol

============================

功　能: 在文本窗口中清除字符到行末
用　法: void clreol(void);
程序例:

```
#include <conio.h>

int main(void)

{
    clrscr();
    cprintf("The function CLREOL clears all characters from the\r\n");
    cprintf("cursor position to the end of the line within the\r\n");
    cprintf("current text window, without moving the cursor.\r\n");
    cprintf("Press any key to continue . . .");
    gotoxy(14, 4);
    getch();

    clreol();
    getch();

    return 0;
}
```

============================

函数名: clrscr

============================

功　能: 清除文本模式窗口
用　法: void clrscr(void);
程序例:

```
#include <conio.h>

int main(void)
{
    int i;

    clrscr();
    for (i = 0; i < 20; i++)
```

```
      cprintf("%d\r\n", i);
   cprintf("\r\nPress any key to clear screen");
   getch();

   clrscr();
   cprintf("The screen has been cleared!");
   getch();

   return 0;
}
```

============================

函数名: coreleft

============================

功　能: 返回未使用内存的大小
用　法: unsigned coreleft(void);
程序例:

```
#include <stdio.h>
#include <alloc.h>

int main(void)
{
   printf("The difference between the highest allocated block and\n");
   printf("the top of the heap is: %lu bytes\n", (unsigned long) coreleft());

   return 0;
}
```

============================

函数名: cos

============================

功　能: 余弦函数
用　法: double cos(double x);
程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
```

```
   double result;
   double x = 0.5;

   result = cos(x);
   printf("The cosine of %lf is %lf\n", x, result);
   return 0;
}
```

============================

函数名: cosh

============================

功　能: 双曲余弦函数
用　法: dluble cosh(double x);
程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
   double result;
   double x = 0.5;

   result = cosh(x);
   printf("The hyperboic cosine of %lf is %lf\n", x, result);
   return 0;
}
```

============================

函数名: country

============================

功　能: 返回与国家有关的信息
用　法: struct COUNTRY *country(int countrycode, struct country *country);
程序例:

```
#include <dos.h>
#include <stdio.h>

#define USA 0
```

```
int main(void)
{
    struct COUNTRY country_info;

    country(USA, &country_info);
    printf("The currency symbol for the USA is: %s\n",
            country_info.co_curr);
    return 0;
}
```

============================

函数名: cprintf

============================

功　能: 送格式化输出至屏幕
用　法: int cprintf(const char *format[, argument, ...]);
程序例:

```
#include <conio.h>

int main(void)
{
    /* clear the screen */
    clrscr();

    /* create a text window */
    window(10, 10, 80, 25);

    /* output some text in the window */
    cprintf("Hello world\r\n");

    /* wait for a key */
    getch();
    return 0;
}
```

============================

函数名: cputs

============================

功　能: 写字符到屏幕
用　法: void cputs(const char *string);

程序例:

```c
#include <conio.h>

int main(void)
{
    /* clear the screen */
    clrscr();

    /* create a text window */
    window(10, 10, 80, 25);

    /* output some text in the window */
    cputs("This is within the window\r\n");

    /* wait for a key */
    getch();
    return 0;
}
```

==============================

函数名: _creat   creat

==============================

功　能: 创建一个新文件或重写一个已存在的文件
用　法: int creat (const char *filename, int permiss);
程序例:

```c
#include <sys\stat.h>
#include <string.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
    int handle;
    char buf[11] = "0123456789";

    /* change the default file mode from text to binary */
    _fmode = O_BINARY;

    /* create a binary file for reading and writing */
    handle = creat("DUMMY.FIL", S_IREAD | S_IWRITE);
```

```c
    /* write 10 bytes to the file */
    write(handle, buf, strlen(buf));

    /* close the file */
    close(handle);
    return 0;
}
```

============================

函数名: creatnew

============================

功　能: 创建一个新文件
用　法: int creatnew(const char *filename, int attrib);
程序例:

```c
#include <string.h>
#include <stdio.h>
#include <errno.h>
#include <dos.h>
#include <io.h>

int main(void)
{
    int handle;
    char buf[11] = "0123456789";

    /* attempt to create a file that doesn't already exist */
    handle = creatnew("DUMMY.FIL", 0);

    if (handle == -1)
        printf("DUMMY.FIL already exists.\n");
    else
    {
        printf("DUMMY.FIL successfully created.\n");
        write(handle, buf, strlen(buf));
        close(handle);
    }
    return 0;
}
```

=============================

函数名: creattemp

=============================

功　能: 创建一个新文件或重写一个已存在的文件
用　法: int creattemp(const char *filename, int attrib);
程序例:

```c
#include <string.h>
#include <stdio.h>
#include <io.h>

int main(void)
{
    int handle;
    char pathname[128];

    strcpy(pathname, "\\");

    /* create a unique file in the root directory */
    handle = creattemp(pathname, 0);

    printf("%s was the unique file created.\n", pathname);
    close(handle);
    return 0;
}
```


=============================

函数名: cscanf

=============================

功　能: 从控制台执行格式化输入
用　法: int cscanf(char *format[,argument, ...]);
程序例:

```c
#include <conio.h>

int main(void)
{
    char string[80];

    /* clear the screen */
    clrscr();
```

```
    /* Prompt the user for input */
    cprintf("Enter a string with no spaces:");

    /* read the input */
    cscanf("%s", string);

    /* display what was read */
    cprintf("\r\nThe string entered is: %s", string);
    return 0;
}
```

============================

函数名: ctime

============================

功　能: 把日期和时间转换为字符串
用　法: char *ctime(const time_t *time);
程序例:

```
#include <stdio.h>
#include <time.h>

int main(void)
{
    time_t t;

    time(&t);
    printf("Today's date and time: %s\n", ctime(&t));
    return 0;
}
```

============================

函数名: ctrlbrk

============================

功　能: 设置 Ctrl-Break 处理程序
用　法: void ctrlbrk(*fptr)(void);
程序例:

```
#include <stdio.h>
#include <dos.h>
```

```
#define ABORT 0

int c_break(void)
{
    printf("Control-Break pressed.  Program aborting ...\n");
    return (ABORT);
}

int main(void)
{
    ctrlbrk(c_break);
    for(;;)
    {
        printf("Looping... Press <Ctrl-Break> to quit:\n");
    }
    return 0;
}
```

==============================

函数名: delay

==============================

功　能: 将程序的执行暂停一段时间(毫秒)
用　法: void delay(unsigned milliseconds);
程序例:

```
/* Emits a 440-Hz tone for 500 milliseconds */
#include <dos.h>

int main(void)
{
    sound(440);
    delay(500);
    nosound();

    return 0;
}
```

==============================

函数名: delline

============================

功　能: 在文本窗口中删去一行
用　法: void delline(void);
程序例:

```
#include <conio.h>

int main(void)
{
   clrscr();
   cprintf("The function DELLINE deletes \
    the line containing the\r\n");
   cprintf("cursor and moves all lines \
    below it one line up.\r\n");
   cprintf("DELLINE operates within the \
    currently active text\r\n");
   cprintf("window.  Press any key to \
    continue . . .");
   gotoxy(1,2);  /* Move the cursor to the
      second line and first column */
   getch();

   delline();
   getch();

   return 0;
}
```

============================

函数名: detectgraph

============================

功　能: 通过检测硬件确定图形驱动程序和模式
用　法: void far detectgraph(int far *graphdriver, int far *graphmode);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/* names of the various cards supported */
```

```c
char *dname[] = { "requests detection",
    "a CGA",
    "an MCGA",
    "an EGA",
    "a 64K EGA",
    "a monochrome EGA",
    "an IBM 8514",
    "a Hercules monochrome",
    "an AT&T 6300 PC",
    "a VGA",
    "an IBM 3270 PC"
  };

int main(void)
{
   /* returns detected hardware info. */
   int gdriver, gmode, errorcode;

  /* detect graphics hardware available */
   detectgraph(&gdriver, &gmode);

   /* read result of detectgraph call */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error
         occurred */
   {
      printf("Graphics error: %s\n", \
      grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error
    code */
   }

   /* display the information detected */
   clrscr();
   printf("You have %s video display \
   card.\n", dname[gdriver]);
   printf("Press any key to halt:");
   getch();
   return 0;
}
```

============================

函数名: difftime

============================

功　能: 计算两个时刻之间的时间差
用　法: double difftime(time_t time2, time_t time1);
程序例:

```c
#include <time.h>
#include <stdio.h>
#include <dos.h>
#include <conio.h>

int main(void)
{
    time_t first, second;

    clrscr();
    first = time(NULL);  /* Gets system
        time */
    delay(2000);         /* Waits 2 secs */
    second = time(NULL); /* Gets system time
        again */

    printf("The difference is: %f \
    seconds\n",difftime(second,first));
    getch();

    return 0;
}
```

============================

函数名: disable

============================

功　能: 屏蔽中断
用　法: void disable(void);
程序例:

```c
/***NOTE: This is an interrupt service
 routine. You cannot compile this program
 with Test Stack Overflow turned on and
```

get an executable file that operates
 correctly. */

```c
#include <stdio.h>
#include <dos.h>
#include <conio.h>

#define INTR 0X1C    /* The clock tick
   interrupt */

void interrupt ( *oldhandler)(void);

int count=0;

void interrupt handler(void)
{
/* disable interrupts during the handling of
   the interrupt */
   disable();
/* increase the global counter */
   count++;
/* reenable interrupts at the end of the
   handler */
   enable();
/* call the old routine */
   oldhandler();
}

int main(void)
{
/* save the old interrupt vector */
   oldhandler = getvect(INTR);

/* install the new interrupt handler */
   setvect(INTR, handler);

/* loop until the counter exceeds 20 */
   while (count < 20)
      printf("count is %d\n",count);

/* reset the old interrupt handler */
   setvect(INTR, oldhandler);

   return 0;
```

```
}
```
============================

函数名: div

============================

功　能: 将两个整数相除，返回商和余数
用　法: div_t (int number, int denom);
程序例:

```c
#include <stdlib.h>
#include <stdio.h>

div_t x;

int main(void)
{
    x = div(10,3);
    printf("10 div 3 = %d remainder %d\n", x.quot, x.rem);

    return 0;
}
```


============================

函数名: dosexterr

============================

功　能: 获取扩展 DOS 错误信息
用　法: int dosexterr(struct DOSERR *dblkp);
程序例:

```c
#include <stdio.h>
#include <dos.h>

int main(void)
{
    FILE *fp;
    struct DOSERROR info;

    fp = fopen("perror.dat","r");
    if (!fp) perror("Unable to open file for
      reading");
    dosexterr(&info);
```

```c
    printf("Extended DOS error \
    information:\n");
    printf("   Extended error: \
    %d\n",info.exterror);
    printf("            Class: \
    %x\n",info.class);
    printf("           Action: \
    %x\n",info.action);
    printf("      Error Locus: \
    %x\n",info.locus);

    return 0;
}
```

============================

函数名: dostounix

============================

功　能: 转换日期和时间为 UNIX 时间格式
用　法: long dostounix(struct date *dateptr, struct time *timeptr);
程序例:

```c
#include <time.h>
#include <stddef.h>
#include <dos.h>
#include <stdio.h>

int main(void)
{
    time_t t;
    struct time d_time;
    struct date d_date;
    struct tm *local;

    getdate(&d_date);
    gettime(&d_time);

    t = dostounix(&d_date, &d_time);
    local = localtime(&t);
    printf("Time and Date: %s\n", \
    asctime(local));

    return 0;
```

}

============================

函数名: drawpoly

============================

功　能: 画多边形
用　法: void far drawpoly(int numpoints, int far *polypoints);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int maxx, maxy;

    /* our polygon array */
    int poly[10];

    /* initialize graphics and local
       variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)
    /* an error occurred */
    {
        printf("Graphics error: %s\n", \
        grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
    /* terminate with an error code */
        exit(1);
    }

    maxx = getmaxx();
    maxy = getmaxy();
```

```c
   poly[0] = 20;          /* 1st vertext */
   poly[1] = maxy / 2;

   poly[2] = maxx - 20; /* 2nd */
   poly[3] = 20;

   poly[4] = maxx - 50; /* 3rd */
   poly[5] = maxy - 20;

   poly[6] = maxx / 2;  /* 4th */
   poly[7] = maxy / 2;
/*
   drawpoly doesn't automatically close
   the polygon, so we close it.
*/
   poly[8] = poly[0];
   poly[9] = poly[1];

   /* draw the polygon */
   drawpoly(5, poly);

   /* clean up */
   getch();
   closegraph();
   return 0;
}
```

============================

函数名: dup

============================

功 能: 复制一个文件句柄
用 法: int dup(int handle);
程序例:

```c
#include <string.h>
#include <stdio.h>
#include <conio.h>
#include <io.h>

void flush(FILE *stream);
```

```c
int main(void)
{
    FILE *fp;
    char msg[] = "This is a test";

    /* create a file */
    fp = fopen("DUMMY.FIL", "w");

    /* write some data to the file */
    fwrite(msg, strlen(msg), 1, fp);

    clrscr();
    printf("Press any key to flush \
    DUMMY.FIL:");
    getch();

    /* flush the data to DUMMY.FIL without
       closing it */
    flush(fp);

    printf("\nFile was flushed, Press any \
    key to quit:");
    getch();
    return 0;
}

void flush(FILE *stream)
{
    int duphandle;

    /* flush TC's internal buffer */
    fflush(stream);

    /* make a duplicate file handle */
    duphandle = dup(fileno(stream));

    /* close the duplicate handle to flush the
       DOS buffer */
    close(duphandle);
}
```

============================

函数名: dup2

```
=============================
```

功　能: 复制文件句柄
用　法: int dup2(int oldhandle, int newhandle);
程序例:

```c
#include <sys\stat.h>
#include <string.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
   #define STDOUT 1

   int nul, oldstdout;
   char msg[] = "This is a test";

   /* create a file */
   nul = open("DUMMY.FIL", O_CREAT | O_RDWR,
      S_IREAD | S_IWRITE);

   /* create a duplicate handle for standard
      output */
   oldstdout = dup(STDOUT);
   /*
      redirect standard output to DUMMY.FIL
      by duplicating the file handle onto the
      file handle for standard output.
   */
   dup2(nul, STDOUT);

   /* close the handle for DUMMY.FIL */
   close(nul);

   /* will be redirected into DUMMY.FIL */
   write(STDOUT, msg, strlen(msg));

   /* restore original standard output
      handle */
   dup2(oldstdout, STDOUT);

   /* close duplicate handle for STDOUT */
   close(oldstdout);
```

```
   return 0;
}
```

==========================

函数名: ecvt

==========================

功　能: 把一个浮点数转换为字符串
用　法: char ecvt(double value, int ndigit, int *decpt, int *sign);
程序例:

```c
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   char *string;
   double value;
   int dec, sign;
   int ndig = 10;

   clrscr();
   value = 9.876;
   string = ecvt(value, ndig, &dec, &sign);
   printf("string = %s      dec = %d \
sign = %d\n", string, dec, sign);

   value = -123.45;
   ndig= 15;
   string = ecvt(value,ndig,&dec,&sign);
   printf("string = %s dec = %d sign = %d\n",
string, dec, sign);


   value = 0.6789e5; /* scientific
notation */
   ndig = 5;
   string = ecvt(value,ndig,&dec,&sign);
   printf("string = %s            dec = %d\
sign = %d\n", string, dec, sign);
```

```
   return 0;
}
```

==========================

函数名: ellipse

==========================

功　能: 画一椭圆
用　法: void far ellipse(int x, int y, int stangle, int endangle,
    int xradius, int yradius);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int midx, midy;
   int stangle = 0, endangle = 360;
   int xradius = 100, yradius = 50;

   /* initialize graphics, local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)
   /* an error occurred */
   {
      printf("Graphics error: %s\n",
      grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1);
   /* terminate with an error code */
   }

   midx = getmaxx() / 2;
```

```
   midy = getmaxy() / 2;
   setcolor(getmaxcolor());

   /* draw ellipse */
   ellipse(midx, midy, stangle, endangle,
    xradius, yradius);

   /* clean up */
   getch();
   closegraph();
   return 0;
}
```

============================

函数名: enable

============================

功　能: 开放硬件中断
用　法: void enable(void);
程序例:

```
/* ** NOTE:
This is an interrupt service routine. You can NOT compile this program
with Test Stack Overflow turned on and get an executable file which will
operate correctly.
*/

#include <stdio.h>
#include <dos.h>
#include <conio.h>

/* The clock tick interrupt */
#define INTR 0X1C

void interrupt ( *oldhandler)(void);

int count=0;

void interrupt handler(void)
{
/*
   disable interrupts during the handling of the interrupt
*/
```

```
   disable();
/* increase the global counter */
   count++;
/*
   re enable interrupts at the end of the handler
*/
   enable();
/* call the old routine */
   oldhandler();
}

int main(void)
{
/* save the old interrupt vector */
   oldhandler = getvect(INTR);

/* install the new interrupt handler */
   setvect(INTR, handler);

/* loop until the counter exceeds 20 */
   while (count < 20)
      printf("count is %d\n",count);

/* reset the old interrupt handler */
   setvect(INTR, oldhandler);

   return 0;
}
```

============================

函数名: eof

============================

功　能: 检测文件结束
用　法: int eof(int *handle);
程序例:

```
#include <sys\stat.h>
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>
```

```c
int main(void)
{
    int handle;
    char msg[] = "This is a test";
    char ch;

    /* create a file */
    handle = open("DUMMY.FIL",
    O_CREAT | O_RDWR,
    S_IREAD | S_IWRITE);

    /* write some data to the file */
    write(handle, msg, strlen(msg));

    /* seek to the beginning of the file */
    lseek(handle, 0L, SEEK_SET);

    /*
        reads chars from the file until hit EOF
    */
    do
    {
        read(handle, &ch, 1);
        printf("%c", ch);
    } while (!eof(handle));

    close(handle);
    return 0;
}
```

============================

函数名: exec...

============================

功　能: 装入并运行其它程序的函数
用　法: int execl(char *pathname, char *arg0, arg1, ..., argn, NULL);
 int execle(char *pathname, char *arg0, arg1, ..., argn, NULL,
     char *envp[]);
 int execlp(char *pathname, char *arg0, arg1, .., NULL);
 int execple(char *pathname, char *arg0, arg1, ..., NULL,
      char *envp[]);
 int execv(char *pathname, char *argv[]);
 int execve(char *pathname, char *argv[], char *envp[]);

```c
 int execvp(char *pathname, char *argv[]);
 int execvpe(char *pathname, char *argv[], char *envp[]);
```
程序例:

```c
/* execv example */
#include <process.h>
#include <stdio.h>
#include <errno.h>

void main(int argc, char *argv[])
{
   int i;

   printf("Command line arguments:\n");
   for (i=0; i<argc; i++)
      printf("[%2d] : %s\n", i, argv[i]);

   printf("About to exec child with arg1 arg2 ...\n");
   execv("CHILD.EXE", argv);

   perror("exec error");

   exit(1);
}
```

===========================

函数名: exit

===========================
功 能: 终止程序
用 法: void exit(int status);
程序例:

```c
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>

int main(void)
{
   int status;

   printf("Enter either 1 or 2\n");
   status = getch();
```

```
   /* Sets DOS errorlevel  */
   exit(status - '0');

/* Note: this line is never reached */
   return 0;
}
```

============================

函数名: exp

============================
功　能: 指数函数
用　法: double exp(double x);
程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
   double result;
   double x = 4.0;

   result = exp(x);
   printf("'e' raised to the power \
   of %lf (e ^ %lf) = %lf\n",
   x, x, result);

   return 0;
}
```

============================

函数名: fabs

============================
功　能: 返回浮点数的绝对值
用　法: double fabs(double x);
程序例:

```
#include <stdio.h>
#include <math.h>
```

```c
int main(void)
{
    float  number = -1234.0;

    printf("number: %f  absolute value: %f\n",
    number, fabs(number));
    return 0;
}
```

===========================

函数名: farcalloc

===========================

功　能: 从远堆栈中申请空间
用　法: void far *farcalloc(unsigned long units, unsigned ling unitsz);
程序例:

```c
#include <stdio.h>
#include <alloc.h>
#include <string.h>
#include <dos.h>

int main(void)
{
    char far *fptr;
    char *str = "Hello";

    /* allocate memory for the far pointer */
    fptr = farcalloc(10, sizeof(char));

    /* copy "Hello" into allocated memory */
    /*
       Note: movedata is used because you
       might be in a small data model, in
       which case a normal string copy routine
       can not be used since it assumes the
       pointer size is near.
    */
    movedata(FP_SEG(str), FP_OFF(str),
      FP_SEG(fptr), FP_OFF(fptr),
            strlen(str));
```

```
    /* display string (note the F modifier) */
    printf("Far string is: %Fs\n", fptr);

    /* free the memory */
    farfree(fptr);

    return 0;
}
```

============================

函数名: **farcoreleft**

============================

功　能: 返回远堆中未作用存储区大小
用　法: long farcoreleft(void);
程序例:

```
#include <stdio.h>
#include <alloc.h>

int main(void)
{
    printf("The difference between the\
     highest allocated block in the\
          far\n");
    printf("heap and the top of the far heap\
          is: %lu bytes\n", farcoreleft());

    return 0;
}
```

============================

函数名: **farfree**

============================

功　能: 从远堆中释放一块
用　法: void farfree(void);
程序例:

```
#include <stdio.h>
#include <alloc.h>
```

```
#include <string.h>
#include <dos.h>

int main(void)
{
    char far *fptr;
    char *str = "Hello";

    /* allocate memory for the far pointer */
    fptr = farcalloc(10, sizeof(char));

    /* copy "Hello" into allocated memory */
    /*
       Note: movedata is used because you might be in a small data model,
       in which case a normal string copy routine can't be used since it
       assumes the pointer size is near.
    */
    movedata(FP_SEG(str), FP_OFF(str),
             FP_SEG(fptr), FP_OFF(fptr),
             strlen(str));

    /* display string (note the F modifier) */
    printf("Far string is: %Fs\n", fptr);

    /* free the memory */
    farfree(fptr);

    return 0;
}
```

==============================

函数名: farmalloc

==============================

功　能: 从远堆中分配存储块
用　法: void far *farmalloc(unsigned long size);
程序例:

```
#include <stdio.h>
#include <alloc.h>
#include <string.h>
#include <dos.h>
```

```c
int main(void)
{
    char far *fptr;
    char *str = "Hello";

    /* allocate memory for the far pointer */
    fptr = farmalloc(10);

    /* copy "Hello" into allocated memory */
    /*
       Note: movedata is used because we might
       be in a small data model, in which case
       a normal string copy routine can not be
       used since it assumes the pointer size
       is near.
    */
    movedata(FP_SEG(str), FP_OFF(str),
      FP_SEG(fptr), FP_OFF(fptr),
      strlen(str));

    /* display string (note the F modifier) */
    printf("Far string is: %Fs\n", fptr);

    /* free the memory */
    farfree(fptr);

    return 0;
}
```

============================

函数名: farrealloc

============================

功　能: 调整远堆中的分配块
用　法: void far *farrealloc(void far *block, unsigned long newsize);
程序例:

```c
#include <stdio.h>
#include <alloc.h>

int main(void)
```

```
{
    char far *fptr;

    fptr = farmalloc(10);
    printf("First address: %Fp\n", fptr);
    fptr = farrealloc(fptr,20);
    printf("New address   : %Fp\n", fptr);
    farfree(fptr);
    return 0;
}
```

==========================

函数名: fclose

==========================

功　能: 关闭一个流
用　法: int fclose(FILE *stream);
程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    FILE *fp;
    char buf[11] = "0123456789";

    /* create a file containing 10 bytes */
    fp = fopen("DUMMY.FIL", "w");
    fwrite(&buf, strlen(buf), 1, fp);

    /* close the file */
    fclose(fp);
    return 0;
}
```

==========================

函数名: fcloseall

==========================

功　能: 关闭打开流

用　法: int fcloseall(void);
程序例:

```c
#include <stdio.h>

int main(void)
{
   int streams_closed;

   /* open two streams */
   fopen("DUMMY.ONE", "w");
   fopen("DUMMY.TWO", "w");

   /* close the open streams */
   streams_closed = fcloseall();

   if (streams_closed == EOF)
      /* issue an error message */
      perror("Error");
   else
      /* print result of fcloseall() function */
      printf("%d streams were closed.\n", streams_closed);

   return 0;
}
```

==========================

函数名: fcvt

==========================
功　能: 把一个浮点数转换为字符串
用　法: char *fcvt(double value, int ndigit, int *decpt, int *sign);
程序例:

```c
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   char *string;
   double value;
   int dec, sign;
```

```c
    int ndig = 10;

    clrscr();
    value = 9.876;
    string = ecvt(value, ndig, &dec, &sign);
    printf("string = %s        dec = %d \
            sign = %d\n", string, dec, sign);

    value = -123.45;
    ndig= 15;
    string = ecvt(value,ndig,&dec,&sign);
    printf("string = %s dec = %d sign = %d\n",
            string, dec, sign);

    value = 0.6789e5; /* scientific
                         notation */
    ndig = 5;
    string = ecvt(value,ndig,&dec,&sign);
    printf("string = %s             dec = %d\
            sign = %d\n", string, dec, sign);

    return 0;
}
```

===========================

函数名: fdopen

============================
功　能: 把流与一个文件句柄相接
用　法: FILE *fdopen(int handle, char *type);
程序例:

```c
#include <sys\stat.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
    int handle;
    FILE *stream;
```

```c
   /* open a file */
   handle = open("DUMMY.FIL", O_CREAT,
    S_IREAD | S_IWRITE);

   /* now turn the handle into a stream */
   stream = fdopen(handle, "w");

   if (stream == NULL)
      printf("fdopen failed\n");
   else
   {
      fprintf(stream, "Hello world\n");
      fclose(stream);
   }
   return 0;
}
```

============================

函数名: feof

============================

功 能: 检测流上的文件结束符
用 法: int feof(FILE *stream);
程序例:

```c
#include <stdio.h>

int main(void)
{
   FILE *stream;

   /* open a file for reading */
   stream = fopen("DUMMY.FIL", "r");

   /* read a character from the file */
   fgetc(stream);

   /* check for EOF */
   if (feof(stream))
      printf("We have reached end-of-file\n");

   /* close the file */
```

```
   fclose(stream);
   return 0;
}
```


```
===========================
```

函数名: ferror

```
===========================
```

功　能: 检测流上的错误
用　法: int ferror(FILE *stream);
程序例:

```
#include <stdio.h>

int main(void)
{
   FILE *stream;

   /* open a file for writing */
   stream = fopen("DUMMY.FIL", "w");

   /* force an error condition by attempting to read */
   (void) getc(stream);

   if (ferror(stream))  /* test for an error on the stream */
   {
      /* display an error message */
      printf("Error reading from DUMMY.FIL\n");

      /* reset the error and EOF indicators */
      clearerr(stream);
   }

   fclose(stream);
   return 0;
}
```


```
===========================
```

函数名: fflush

```
===========================
```

功　能: 清除一个流
用　法: int fflush(FILE *stream);
程序例:

```c
#include <string.h>
#include <stdio.h>
#include <conio.h>
#include <io.h>

void flush(FILE *stream);

int main(void)
{
   FILE *stream;
   char msg[] = "This is a test";

   /* create a file */
   stream = fopen("DUMMY.FIL", "w");

   /* write some data to the file */
   fwrite(msg, strlen(msg), 1, stream);

   clrscr();
   printf("Press any key to flush\
   DUMMY.FIL:");
   getch();

   /* flush the data to DUMMY.FIL without\
      closing it */
   flush(stream);

   printf("\nFile was flushed, Press any key\
   to quit:");
   getch();
   return 0;
}

void flush(FILE *stream)
{
    int duphandle;

    /* flush the stream's internal buffer */
    fflush(stream);
```

```
        /* make a duplicate file handle */
        duphandle = dup(fileno(stream));

        /* close the duplicate handle to flush\
           the DOS buffer */
        close(duphandle);
}
```

==========================

函数名: fgetc

==========================
功　能: 从流中读取字符
用　法: int fgetc(FILE *stream);
程序例:

```
#include <string.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    FILE *stream;
    char string[] = "This is a test";
    char ch;

    /* open a file for update */
    stream = fopen("DUMMY.FIL", "w+");

    /* write a string into the file */
    fwrite(string, strlen(string), 1, stream);

    /* seek to the beginning of the file */
    fseek(stream, 0, SEEK_SET);

    do
    {
        /* read a char from the file */
        ch = fgetc(stream);

        /* display the character */
        putch(ch);
```

```c
    } while (ch != EOF);

    fclose(stream);
    return 0;
}
```

==========================

函数名: fgetchar

==========================
功　能: 从流中读取字符
用　法: int fgetchar(void);
程序例:

```c
#include <stdio.h>

int main(void)
{
    char ch;

    /* prompt the user for input */
    printf("Enter a character followed by \
    <Enter>: ");

    /* read the character from stdin */
    ch = fgetchar();

    /* display what was read */
    printf("The character read is: '%c'\n",
            ch);
    return 0;
}
```

==========================

函数名: fgetpos

==========================
功　能: 取得当前文件的句柄
用　法: int fgetpos(FILE *stream);
程序例:

```c
#include <string.h>
#include <stdio.h>

int main(void)
{
   FILE *stream;
   char string[] = "This is a test";
   fpos_t filepos;

   /* open a file for update */
   stream = fopen("DUMMY.FIL", "w+");

   /* write a string into the file */
   fwrite(string, strlen(string), 1, stream);

   /* report the file pointer position */
   fgetpos(stream, &filepos);
   printf("The file pointer is at byte\
           %ld\n", filepos);

   fclose(stream);
   return 0;
}
```

============================

函数名: fgets

============================
功　能: 从流中读取一字符串
用　法: char *fgets(char *string, int n, FILE *stream);
程序例:

```c
#include <string.h>
#include <stdio.h>

int main(void)
{
   FILE *stream;
   char string[] = "This is a test";
   char msg[20];
```

```c
    /* open a file for update */
    stream = fopen("DUMMY.FIL", "w+");

    /* write a string into the file */
    fwrite(string, strlen(string), 1, stream);

    /* seek to the start of the file */
    fseek(stream, 0, SEEK_SET);

    /* read a string from the file */
    fgets(msg, strlen(string)+1, stream);

    /* display the string */
    printf("%s", msg);

    fclose(stream);
    return 0;
}
```

============================

函数名: filelength

============================
功　能: 取文件长度字节数
用　法: long filelength(int handle);
程序例:

```c
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
    int handle;
    char buf[11] = "0123456789";

    /* create a file containing 10 bytes */
    handle = open("DUMMY.FIL", O_CREAT);
    write(handle, buf, strlen(buf));

    /* display the size of the file */
```

```
    printf("file length in bytes: %ld\n",
    filelength(handle));

    /* close the file */
    close(handle);
    return 0;
}
```

============================

函数名: fillellipse

============================

功　能: 画出并填充一椭圆
用　法: void far fillellipse(int x, int y, int xradius, int yradius);
程序例:

```
#include <graphics.h>
#include <conio.h>

int main(void)
{
    int gdriver = DETECT, gmode;
    int xcenter, ycenter, i;

    initgraph(&gdriver,&gmode,"");
    xcenter = getmaxx() / 2;
    ycenter = getmaxy() / 2;

    for (i=0; i<13; i++)
    {
        setfillstyle(i,WHITE);
        fillellipse(xcenter,ycenter,100,50);
        getch();
    }

    closegraph();
    return 0;
}
```

============================

函数名: fillpoly

============================

功　能: 画并填充一个多边形

用　法: void far fillpoly(int numpoints, int far *polypoints);

程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int i, maxx, maxy;

   /* our polygon array */
   int poly[8];

   /* initialize graphics, local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)
   /* an error occurred */
   {
      printf("Graphics error: %s\n",
             grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1);
      /* terminate with an error code */
   }

   maxx = getmaxx();
   maxy = getmaxy();

   poly[0] = 20;         /* 1st vertext */
   poly[1] = maxy / 2;

   poly[2] = maxx - 20; /* 2nd */
   poly[3] = 20;
```

```c
   poly[4] = maxx - 50; /* 3rd */
   poly[5] = maxy - 20;

   /*
      4th vertex. fillpoly automatically
      closes the polygon.
   */
   poly[6] = maxx / 2;
   poly[7] = maxy / 2;

   /* loop through the fill patterns */
   for (i=EMPTY_FILL; i<USER_FILL; i++)
   {
      /* set fill pattern */
      setfillstyle(i, getmaxcolor());

      /* draw a filled polygon */
      fillpoly(4, poly);

      getch();
   }

   /* clean up */
   closegraph();
   return 0;
}
```

==============================

函数名: findfirst, findnext

==============================

功　能: 搜索磁盘目录; 取得下一个匹配的 findfirst 模式的文件
用　法: int findfirst(char *pathname, struct ffblk *ffblk, int attrib);
 int findnext(struct ffblk *ffblk);
程序例:

```c
/* findnext example */

#include <stdio.h>
#include <dir.h>
```

```
int main(void)
{
    struct ffblk ffblk;
    int done;
    printf("Directory listing of *.*\n");
    done = findfirst("*.*",&ffblk,0);
    while (!done)
    {
        printf("  %s\n", ffblk.ff_name);
        done = findnext(&ffblk);
    }

    return 0;
}
```

===========================

函数名: floodfill

===========================

功　能: 填充一个有界区域
用　法: void far floodfill(int x, int y, int border);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int maxx, maxy;

    /* initialize graphics, local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)
    /* an error occurred */
    {
```

```
        printf("Graphics error: %s\n",
                grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1);
        /* terminate with an error code */
    }

    maxx = getmaxx();
    maxy = getmaxy();

    /* select drawing color */
    setcolor(getmaxcolor());

    /* select fill color */
    setfillstyle(SOLID_FILL, getmaxcolor());

    /* draw a border around the screen */
    rectangle(0, 0, maxx, maxy);

    /* draw some circles */
    circle(maxx / 3, maxy /2, 50);
    circle(maxx / 2, 20, 100);
    circle(maxx-20, maxy-50, 75);
    circle(20, maxy-20, 25);

    /* wait for a key */
    getch();

    /* fill in bounded region */
    floodfill(2, 2, getmaxcolor());

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: floor

============================

功　能: 向下舍入

用　法: double floor(double x);
程序例:

```c
#include <stdio.h>
#include <math.h>

int main(void)
{
    double number = 123.54;
    double down, up;

    down = floor(number);
    up = ceil(number);

    printf("original number     %10.2lf\n",
            number);
    printf("number rounded down %10.2lf\n",
            down);
    printf("number rounded up   %10.2lf\n",
            up);

    return 0;
}
```

============================

函数名: flushall

============================
功　能: 清除所有缓冲区
用　法: int flushall(void);
程序例:

```c
#include <stdio.h>

int main(void)
{
    FILE *stream;

    /* create a file */
    stream = fopen("DUMMY.FIL", "w");

    /* flush all open streams */
    printf("%d streams were flushed.\n",
```

```
    flushall());

    /* close the file */
    fclose(stream);
    return 0;
}
```

============================

函数名: fmod

============================

功 能: 计算 x 对 y 的模, 即 x/y 的余数
用 法: double fmod(double x, double y);
程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double x = 5.0, y = 2.0;
    double result;

    result = fmod(x,y);
    printf("The remainder of (%lf / %lf) is \
            %lf\n", x, y, result);
    return 0;
}
```

============================

函数名: fnmerge

============================

功 能: 建立新文件名
用 法: void fnerge(char *path, char *drive, char *dir);
程序例:

```
#include <string.h>
#include <stdio.h>
#include <dir.h>
```

```
int main(void)
{
    char s[MAXPATH];
    char drive[MAXDRIVE];
    char dir[MAXDIR];
    char file[MAXFILE];
    char ext[MAXEXT];

    getcwd(s,MAXPATH);              /* get the current working directory */
    strcat(s,"\\");                 /* append on a trailing \ character */
    fnsplit(s,drive,dir,file,ext);  /* split the string to separate elems */
    strcpy(file,"DATA");
    strcpy(ext,".TXT");
    fnmerge(s,drive,dir,file,ext);  /* merge everything into one string */
    puts(s);                        /* display resulting string */

    return 0;
}
```

============================

函数名: fopen

============================

功　能: 打开一个流
用　法: FILE *fopen(char *filename, char *type);
程序例:

```
#include <stdlib.h>
#include <stdio.h>
#include <dir.h>

int main(void)
{
    char *s;
    char drive[MAXDRIVE];
    char dir[MAXDIR];
    char file[MAXFILE];
    char ext[MAXEXT];
    int flags;

    s=getenv("COMSPEC"); /* get the comspec environment parameter */
    flags=fnsplit(s,drive,dir,file,ext);
```

```
   printf("Command processor info:\n");
   if(flags & DRIVE)
      printf("\tdrive: %s\n",drive);
   if(flags & DIRECTORY)
      printf("\tdirectory: %s\n",dir);
   if(flags & FILENAME)
      printf("\tfile: %s\n",file);
   if(flags & EXTENSION)
      printf("\textension: %s\n",ext);

   return 0;
}
```

============================

函数名: fprintf

============================

功　能: 传送格式化输出到一个流中
用　法: int fprintf(FILE *stream, char *format[, argument,...]);
程序例:

```
/* Program to create backup of the
   AUTOEXEC.BAT file */

#include <stdio.h>

int main(void)
{
   FILE *in, *out;

   if ((in = fopen("\\AUTOEXEC.BAT", "rt"))
       == NULL)
   {
      fprintf(stderr, "Cannot open input \
       file.\n");
      return 1;
   }

   if ((out = fopen("\\AUTOEXEC.BAK", "wt"))
       == NULL)
   {
      fprintf(stderr, "Cannot open output \
       file.\n");
      return 1;
```

```
    }

    while (!feof(in))
        fputc(fgetc(in), out);

    fclose(in);
    fclose(out);
    return 0;
}
```

============================

函数名: FP_OFF

============================

功　能: 获取远地址偏移量
用　法: unsigned FP_OFF(void far *farptr);
程序例:

```
/* FP_OFF */

#include <dos.h>
#include <stdio.h>

int main(void)
{
    char *str = "fpoff.c";

    printf("The offset of this file in memory\
            is: %Fp\n", FP_OFF(str));

    return 0;
}
```

============================

函数名: FP_SEG

============================

功　能: 获取远地址段值
用　法: unsigned FP_SEG(void far *farptr);
程序例:

```
/* FP_SEG */

#include <dos.h>
#include <stdio.h>

int main(void)
{
    char *filename = "fpseg.c";

    printf("The offset of this file in memory\
    is: %Fp\n", FP_SEG(filename));

    return(0);
}
```

============================

函数名: fputc

============================

功　能: 送一个字符到一个流中
用　法: int fputc(int ch, FILE *stream);
程序例:

```
#include <stdio.h>

int main(void)
{
    char msg[] = "Hello world";
    int i = 0;

    while (msg[i])
    {
        fputc(msg[i], stdout);
        i++;
    }
    return 0;
}
```

============================

函数名: fputchar

============================

功　能: 送一个字符到标准输出流(stdout)中
用　法: int fputchar(char ch);
程序例:

```c
#include <stdio.h>

int main(void)
{
   char msg[] = "This is a test";
   int i = 0;

   while (msg[i])
   {
      fputchar(msg[i]);
      i++;
   }
   return 0;
}
```

============================

函数名: fputs

============================

功　能: 送一个字符到一个流中
用　法: int fputs(char *string, FILE *stream);
程序例:

```c
#include <stdio.h>

int main(void)
{
   /* write a string to standard output */
   fputs("Hello world\n", stdout);

   return 0;
}
```

============================

函数名: fread

============================

功　能: 从一个流中读数据
用　法: int fread(void *ptr, int size, int nitems, FILE *stream);
程序例:

```c
#include <string.h>
#include <stdio.h>

int main(void)
{
    FILE *stream;
    char msg[] = "this is a test";
    char buf[20];

    if ((stream = fopen("DUMMY.FIL", "w+"))
        == NULL)
    {
        fprintf(stderr,
                "Cannot open output file.\n");
        return 1;
    }

    /* write some data to the file */
    fwrite(msg, strlen(msg)+1, 1, stream);

    /* seek to the beginning of the file */
    fseek(stream, SEEK_SET, 0);

    /* read the data and display it */
    fread(buf, strlen(msg)+1, 1, stream);
    printf("%s\n", buf);

    fclose(stream);
    return 0;
}
```

============================

函数名: free

============================

功　能: 释放已分配的块
用　法: void free(void *ptr);
程序例:

```
#include <string.h>
#include <stdio.h>
#include <alloc.h>

int main(void)
{
    char *str;

    /* allocate memory for string */
    str = malloc(10);

    /* copy "Hello" to string */
    strcpy(str, "Hello");

    /* display string */
    printf("String is %s\n", str);

    /* free memory */
    free(str);

    return 0;
}
```

============================

函数名: freemem

============================

功　能: 释放先前分配的 DOS 内存块
用　法: int freemem(unsigned seg);
程序例:

```
#include <dos.h>
#include <alloc.h>
#include <stdio.h>

int main(void)
{
    unsigned int size, segp;
    int stat;

    size = 64; /* (64 x 16) = 1024 bytes */
    stat = allocmem(size, &segp);
    if (stat < 0)
        printf("Allocated memory at segment:\
```

```
        %x\n", segp);
    else
        printf("Failed: maximum number of\
        paragraphs available is %u\n",
        stat);
    freemem(segp);

    return 0;
}
```

==========================

函数名: freopen

==========================

功　能: 替换一个流
用　法: FILE *freopen(char *filename, char *type, FILE *stream);
程序例:

```
#include <stdio.h>

int main(void)
{
    /* redirect standard output to a file */
    if (freopen("OUTPUT.FIL", "w", stdout)
        == NULL)
        fprintf(stderr, "error redirecting\
                stdout\n");

    /* this output will go to a file */
    printf("This will go into a file.");

    /* close the standard output stream */
    fclose(stdout);

    return 0;
}
```

==========================

函数名: frexp

==========================

功　能：把一个双精度数分解为尾数的指数
用　法：double frexp(double value, int *eptr);
程序例：

```c
#include <math.h>
#include <stdio.h>

int main(void)
{
    double mantissa, number;
    int exponent;

    number = 8.0;
    mantissa = frexp(number, &exponent);

    printf("The number %lf is ", number);
    printf("%lf times two to the ", mantissa);
    printf("power of %d\n", exponent);

    return 0;
}
```

==============================

函数名: fscanf

==============================
功　能：从一个流中执行格式化输入
用　法：int fscanf(FILE *stream, char *format[,argument...]);
程序例：

```c
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    int i;

    printf("Input an integer: ");

    /* read an integer from the
       standard input stream */
    if (fscanf(stdin, "%d", &i))
        printf("The integer read was: %i\n",
```

```
              i);
    else
    {
        fprintf(stderr, "Error reading an \
                integer from stdin.\n");
        exit(1);
    }
    return 0;
}
```

===========================

函数名: fseek

===========================
功　能: 重定位流上的文件指针
用　法: int fseek(FILE *stream, long offset, int fromwhere);
程序例:

```
#include <stdio.h>

long filesize(FILE *stream);

int main(void)
{
    FILE *stream;

    stream = fopen("MYFILE.TXT", "w+");
    fprintf(stream, "This is a test");
    printf("Filesize of MYFILE.TXT is %ld bytes\n", filesize(stream));
    fclose(stream);
    return 0;
}

long filesize(FILE *stream)
{
    long curpos, length;

    curpos = ftell(stream);
    fseek(stream, 0L, SEEK_END);
    length = ftell(stream);
    fseek(stream, curpos, SEEK_SET);
    return length;
```

```
}
```

==========================

函数名: fsetpos

==========================

功　能: 定位流上的文件指针
用　法: int fsetpos(FILE *stream, const fpos_t *pos);
程序例:

```
#include <stdlib.h>
#include <stdio.h>

void showpos(FILE *stream);

int main(void)
{
   FILE *stream;
   fpos_t filepos;

   /* open a file for update */
   stream = fopen("DUMMY.FIL", "w+");

   /* save the file pointer position */
   fgetpos(stream, &filepos);

   /* write some data to the file */
   fprintf(stream, "This is a test");

   /* show the current file position */
   showpos(stream);

   /* set a new file position, display it */
   if (fsetpos(stream, &filepos) == 0)
     showpos(stream);
   else
   {
     fprintf(stderr, "Error setting file \
      pointer.\n");
     exit(1);
   }
```

```c
   /* close the file */
   fclose(stream);
   return 0;
}


void showpos(FILE *stream)
{
   fpos_t pos;

   /* display the current file pointer
       position of a stream */
   fgetpos(stream, &pos);
   printf("File position: %ld\n", pos);
}
```

============================

函数名: fstat

============================
功　能: 获取打开文件信息
用　法: int fstat(char *handle, struct stat *buff);
程序例:

```c
#include <sys\stat.h>
#include <stdio.h>
#include <time.h>

int main(void)
{
   struct stat statbuf;
   FILE *stream;

   /* open a file for update */
   if ((stream = fopen("DUMMY.FIL", "w+"))
       == NULL)
   {
      fprintf(stderr, "Cannot open output \
             file.\n");
      return(1);
   }
   fprintf(stream, "This is a test");
   fflush(stream);
```

```c
    /* get information about the file */
    fstat(fileno(stream), &statbuf);
    fclose(stream);

    /* display the information returned */
    if (statbuf.st_mode & S_IFCHR)
        printf("Handle refers to a device.\n");
    if (statbuf.st_mode & S_IFREG)
        printf("Handle refers to an ordinary \
               file.\n");
    if (statbuf.st_mode & S_IREAD)
        printf("User has read permission on \
               file.\n");
    if (statbuf.st_mode & S_IWRITE)
        printf("User has write permission on \
                file.\n");

    printf("Drive letter of file: %c\n",
    'A'+statbuf.st_dev);
    printf("Size of file in bytes: %ld\n",
    statbuf.st_size);
    printf("Time file last opened: %s\n",
    ctime(&statbuf.st_ctime));
    return 0;
}
```

============================

函数名: ftell

============================

功　能: 返回当前文件指针
用　法: long ftell(FILE *stream);
程序例:

```c
#include <stdio.h>

int main(void)
{
    FILE *stream;

    stream = fopen("MYFILE.TXT", "w+");
    fprintf(stream, "This is a test");
    printf("The file pointer is at byte \
```

```
        %ld\n", ftell(stream));
    fclose(stream);
    return 0;
}
```

============================

函数名: fwrite

============================

功　能: 写内容到流中
用　法: int fwrite(void *ptr, int size, int nitems, FILE *stream);
程序例:

```c
#include <stdio.h>

struct mystruct
{
  int i;
  char ch;
};

int main(void)
{
    FILE *stream;
    struct mystruct s;

    if ((stream = fopen("TEST.$$$", "wb")) == NULL) /* open file TEST.$$$ */
    {
        fprintf(stderr, "Cannot open output file.\n");
        return 1;
    }
    s.i = 0;
    s.ch = 'A';
    fwrite(&s, sizeof(s), 1, stream); /* write struct s to file */
    fclose(stream); /* close file */
    return 0;
}
```

============================

函数名: gcvt

```
===========================
```

功　能: 把浮点数转换成字符串
用　法: char *gcvt(double value, int ndigit, char *buf);
程序例:

```c
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
   char str[25];
   double num;
   int sig = 5; /* significant digits */

   /* a regular number */
   num = 9.876;
   gcvt(num, sig, str);
   printf("string = %s\n", str);

   /* a negative number */
   num = -123.4567;
   gcvt(num, sig, str);
   printf("string = %s\n", str);

   /* scientific notation */
   num = 0.678e5;
   gcvt(num, sig, str);
   printf("string = %s\n", str);

   return(0);
}
```

```
===========================
```

函数名: geninterrupt

```
===========================
```

功　能: 产生一个软中断
用　法: void geninterrupt(int intr_num);
程序例:

```c
#include <conio.h>
```

```c
#include <dos.h>

/* function prototype */
void writechar(char ch);

int main(void)
{
   clrscr();
   gotoxy(80,25);
   writechar('*');
   getch();
   return 0;
}

/*
   outputs a character at the current cursor
   position using the video BIOS to avoid the
   scrolling of the screen when writing to
   location (80,25).
*/

void writechar(char ch)
{
   struct text_info ti;
   /* grab current text settings */
   gettextinfo(&ti);
   /* interrupt 0x10 sub-function 9 */
   _AH = 9;
   /* character to be output */
   _AL = ch;
   _BH = 0;                    /* video page */
   _BL = ti.attribute;  /* video attribute */
   _CX = 1;             /* repetition factor */
   geninterrupt(0x10);  /* output the char */
}
```

===========================

函数名: getarccoords

===========================

功　能: 取得最后一次调用 arc 的坐标
用　法: void far getarccoords(struct arccoordstype far *arccoords);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
/* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    struct arccoordstype arcinfo;
    int midx, midy;
    int stangle = 45, endangle = 270;
    char sstr[80], estr[80];

/* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

/* read result of initialization */
    errorcode = graphresult();
/* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
                grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
/* terminate with an error code */
        exit(1);
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

/* draw arc and get coordinates */
    setcolor(getmaxcolor());
    arc(midx, midy, stangle, endangle, 100);
    getarccoords(&arcinfo);

/* convert arc information into strings */
    sprintf(sstr, "*- (%d, %d)",
            arcinfo.xstart, arcinfo.ystart);
    sprintf(estr, "*- (%d, %d)",
            arcinfo.xend, arcinfo.yend);
```

```c
    /* output the arc information */
    outtextxy(arcinfo.xstart,
            arcinfo.ystart, sstr);
    outtextxy(arcinfo.xend,
            arcinfo.yend, estr);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: getaspectratio

============================

功　能: 返回当前图形模式的纵横比
用　法: void far getaspectratio(int far *xasp, int far *yasp);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
/* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int xasp, yasp, midx, midy;

/* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

/* read result of initialization */
    errorcode = graphresult();
/* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
                grapherrormsg(errorcode));
        printf("Press any key to halt:");
```

```
        getch();
/* terminate with an error code */
        exit(1);
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;
    setcolor(getmaxcolor());

/* get current aspect ratio settings */
    getaspectratio(&xasp, &yasp);

/* draw normal circle */
    circle(midx, midy, 100);
    getch();

/* draw wide circle */
    cleardevice();
    setaspectratio(xasp/2, yasp);
    circle(midx, midy, 100);
    getch();

/* draw narrow circle */
    cleardevice();
    setaspectratio(xasp, yasp/2);
    circle(midx, midy, 100);

/* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: getbkcolor

============================

功　能: 返回当前背景颜色
用　法: int far getbkcolor(void);
程序例:

```
#include <graphics.h>
```

```c
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int bkcolor, midx, midy;
   char bkname[35];

/* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

/* read result of initialization */
   errorcode = graphresult();
/* an error occurred */
   if (errorcode != grOk)
   {
      printf("Graphics error: %s\n",
             grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
/* terminate with an error code */
      exit(1);
   }

   midx = getmaxx() / 2;
   midy = getmaxy() / 2;
   setcolor(getmaxcolor());

/* for centering text on the display */
   settextjustify(CENTER_TEXT, CENTER_TEXT);

/* get the current background color */
   bkcolor = getbkcolor();

/* convert color value into a string */
   itoa(bkcolor, bkname, 10);
   strcat(bkname,
 " is the current background color.");

/* display a message */
```

```
   outtextxy(midx, midy, bkname);

/* clean up */
   getch();
   closegraph();
   return 0;
}
```

==========================

函数名: getc

==========================

功　能: 从流中取字符
用　法: int getc(FILE *stream);
程序例:

```
#include <stdio.h>

int main(void)
{
   char ch;

   printf("Input a character:");
/* read a character from the
   standard input stream */
   ch = getc(stdin);
   printf("The character input was: '%c'\n",
          ch);
   return 0;
}
```

==========================

函数名: getcbrk

==========================

功　能: 获取 Control_break 设置
用　法: int getcbrk(void);
程序例:

```
#include <stdio.h>
#include <dos.h>
```

```c
int main(void)
{
   if (getcbrk())
      printf("Cntrl-brk flag is on\n");
   else
      printf("Cntrl-brk flag is off\n");

   return 0;
}
```

============================

函数名: getch

============================

功　能: 从控制台无回显地取一个字符
用　法: int getch(void);
程序例:

```c
#include <stdio.h>
#include <conio.h>

int main(void)
{
   char ch;

   printf("Input a character:");
   ch = getche();
   printf("\nYou input a '%c'\n", ch);
   return 0;
}
```

============================

函数名: getchar

============================

功　能: 从 stdin 流中读字符
用　法: int getchar(void);
程序例:

```c
#include <stdio.h>
```

```
int main(void)
{
    int c;

    /* Note that getchar reads from stdin and
       is line buffered; this means it will
       not return until you press ENTER. */

    while ((c = getchar()) != '\n')
        printf("%c", c);

    return 0;
}
```

============================

函数名: getche

============================

功　能: 从控制台取字符(带回显)
用　法: int getche(void);
程序例:

```
#include <stdio.h>
#include <conio.h>

int main(void)
{
    char ch;

    printf("Input a character:");
    ch = getche();
    printf("\nYou input a '%c'\n", ch);
    return 0;
}
```

============================

函数名: getcolor

============================

功　能: 返回当前画线颜色
用　法: int far getcolor(void);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
/* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int color, midx, midy;
    char colname[35];

/* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

/* read result of initialization */
    errorcode = graphresult();
/* an error occurred */
    if (errorcode != grOk)
    {
       printf("Graphics error: %s\n",
              grapherrormsg(errorcode));
       printf("Press any key to halt:");
       getch();
/* terminate with an error code */
       exit(1);
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;
    setcolor(getmaxcolor());

/* for centering text on the display */
    settextjustify(CENTER_TEXT, CENTER_TEXT);

/* get the current drawing color */
    color = getcolor();

/* convert color value into a string */
    itoa(color, colname, 10);
    strcat(colname,
    " is the current drawing color.");
```

```c
/* display a message */
   outtextxy(midx, midy, colname);

/* clean up */
   getch();
   closegraph();
   return 0;
}
```

==============================

函数名: getcurdir

==============================

功　能: 取指定驱动器的当前目录
用　法: int getcurdir(int drive, char *direc);
程序例:

```c
#include <dir.h>
#include <stdio.h>
#include <string.h>

char *current_directory(char *path)
{
   strcpy(path, "X:\\");       /* fill string with form of response: X:\ */
   path[0] = 'A' + getdisk();   /* replace X with current drive letter */
   getcurdir(0, path+3);  /* fill rest of string with current directory */
   return(path);
}

int main(void)
{
   char curdir[MAXPATH];

   current_directory(curdir);
   printf("The current directory is %s\n", curdir);

   return 0;
}
```

==============================

函数名: getcwd

============================

功　能: 取当前工作目录
用　法: char *getcwd(char *buf, int n);
程序例:

```c
#include <stdio.h>
#include <dir.h>

int main(void)
{
   char buffer[MAXPATH];

   getcwd(buffer, MAXPATH);
   printf("The current directory is: %s\n", buffer);
   return 0;
}
```


============================

函数名: getdate

============================

功　能: 取 DOS 日期
用　法: void getdate(struct *dateblk);
程序例:

```c
#include <dos.h>
#include <stdio.h>

int main(void)
{
   struct date d;

   getdate(&d);
   printf("The current year is: %d\n",
   d.da_year);
   printf("The current day is: %d\n",
   d.da_day);
   printf("The current month is: %d\n",
   d.da_mon);
   return 0;
}
```

============================

函数名: **getdefaultpalette**

============================

功　能: 返回调色板定义结构
用　法: struct palettetype *far getdefaultpalette(void);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
/* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int i;

/* structure for returning palette copy */
   struct palettetype far *pal=(void *) 0;

/* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

/* read result of initialization */
   errorcode = graphresult();
/* an error occurred */
   if (errorcode != grOk)
   {
      printf("Graphics error: %s\n",
             grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
/* terminate with an error code */
      exit(1);
   }

   setcolor(getmaxcolor());

/* return a pointer to the default palette */
   pal = getdefaultpalette();
```

```
   for (i=0; i<16; i++)
   {
      printf("colors[%d] = %d\n", i,
            pal->colors[i]);
      getch();
   }

/* clean up */
   getch();
   closegraph();
   return 0;
}
```

============================

函数名: getdisk

============================

功　能: 取当前磁盘驱动器号
用　法: int getdisk(void);
程序例:

```
#include <stdio.h>
#include <dir.h>

int main(void)
{
   int disk;

   disk = getdisk() + 'A';
   printf("The current drive is: %c\n",
    disk);
   return 0;
}
```

============================

函数名: getdrivername

============================

功　能: 返回指向包含当前图形驱动程序名字的字符串指针
用　法: char *getdrivename(void);

程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
/* request auto detection */
   int gdriver = DETECT, gmode, errorcode;

/* stores the device driver name */
   char *drivername;

/* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

/* read result of initialization */
   errorcode = graphresult();
/* an error occurred */
   if (errorcode != grOk)
   {
      printf("Graphics error: %s\n",
              grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
/* terminate with an error code */
      exit(1);
   }

   setcolor(getmaxcolor());

/* get name of the device driver in use */
   drivername = getdrivername();

/* for centering text on the screen */
   settextjustify(CENTER_TEXT, CENTER_TEXT);

/* output the name of the driver */
   outtextxy(getmaxx() / 2, getmaxy() / 2,
      drivername);

/* clean up */
```

```
   getch();
   closegraph();
   return 0;
}
```

============================

函数名: getdta

============================

功　能: 取磁盘传输地址
用　法: char far *getdta(void);
程序例:

```
#include <dos.h>
#include <stdio.h>

int main(void)
{
   char far *dta;

   dta = getdta();
   printf("The current disk transfer \
   address is: %Fp\n", dta);
   return 0;
}
```

============================

函数名: getenv

============================

功　能: 从环境中取字符串
用　法: char *getenv(char *envvar);
程序例:

```
#include <stdlib.h>
#include <stdio.h>


int main(void)
{
    char *s;
```

```
    s=getenv("COMSPEC");        /* get the comspec environment parameter */
    printf("Command processor: %s\n",s);   /* display comspec parameter */

    return 0;
}
```

============================

函数名: getfat, getfatd

============================

功　能: 取文件分配表信息
用　法: void getfat(int drive, struct fatinfo *fatblkp);
程序例:

```
#include <stdio.h>
#include <dos.h>

int main(void)
{
    struct fatinfo diskinfo;
    int flag = 0;

    printf("Please insert disk in drive A\n");
    getchar();

    getfat(1, &diskinfo);
/* get drive information */

    printf("\nDrive A: is ");
    switch((unsigned char) diskinfo.fi_fatid)
    {
       case 0xFD:
 printf("360K low density\n");
 break;

       case 0xF9:
 printf("1.2 Meg high density\n");
 break;

       default:
 printf("unformatted\n");
 flag = 1;
```

```
    }

    if (!flag)
    {
        printf("  sectors per cluster %5d\n",
          diskinfo.fi_sclus);
        printf("    number of clusters %5d\n",
          diskinfo.fi_nclus);
        printf("     bytes per sector %5d\n",
          diskinfo.fi_bysec);
    }

    return 0;
}
```

==========================

函数名: **getfillpattern**

==========================

功　能: 将用户定义的填充模式拷贝到内存中
用　法: void far getfillpattern(char far *upattern);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int maxx, maxy;
    char pattern[8] = {0x00, 0x70, 0x20, 0x27, 0x25, 0x27, 0x04, 0x04};

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
```

```
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

maxx = getmaxx();
maxy = getmaxy();
setcolor(getmaxcolor());

/* select a user defined fill pattern */
setfillpattern(pattern, getmaxcolor());

/* fill the screen with the pattern */
bar(0, 0, maxx, maxy);

getch();

/* get the current user defined fill pattern */
getfillpattern(pattern);

/* alter the pattern we grabbed */
pattern[4] -= 1;
pattern[5] -= 3;
pattern[6] += 3;
pattern[7] -= 4;

/* select our new pattern */
setfillpattern(pattern, getmaxcolor());

/* fill the screen with the new pattern */
bar(0, 0, maxx, maxy);

/* clean up */
getch();
closegraph();
return 0;
}
```

============================

函数名: getfillsettings

============================

功　能: 取得有关当前填充模式和填充颜色的信息

用　法: void far getfillsettings(struct fillsettingstype far *fillinfo);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/  the names of the fill styles supported */
char *fname[] = { "EMPTY_FILL",
                  "SOLID_FILL",
                  "LINE_FILL",
                  "LTSLASH_FILL",
                  "SLASH_FILL",
                  "BKSLASH_FILL",
                  "LTBKSLASH_FILL",
                  "HATCH_FILL",
                  "XHATCH_FILL",
                  "INTERLEAVE_FILL",
                  "WIDE_DOT_FILL",
                  "CLOSE_DOT_FILL",
                  "USER_FILL"
        };

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   struct fillsettingstype fillinfo;
   int midx, midy;
   char patstr[40], colstr[40];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }
```

```
   midx = getmaxx() / 2;
   midy = getmaxy() / 2;

   /* get information about current fill pattern and color */
   getfillsettings(&fillinfo);

   /* convert fill information into strings */
   sprintf(patstr, "%s is the fill style.", fname[fillinfo.pattern]);
   sprintf(colstr, "%d is the fill color.", fillinfo.color);

   /* display the information */
   settextjustify(CENTER_TEXT, CENTER_TEXT);
   outtextxy(midx, midy, patstr);
   outtextxy(midx, midy+2*textheight("W"), colstr);

   /* clean up */
   getch();
   closegraph();
   return 0;
}
```

============================

函数名: getftime

============================

功　能: 取文件日期和时间
用　法: int getftime(int handle, struct ftime *ftimep);
程序例:

```
#include <stdio.h>
#include <io.h>

int main(void)
{
   FILE *stream;
   struct ftime ft;

   if ((stream = fopen("TEST.$$$",
       "wt")) == NULL)
   {
      fprintf(stderr,
```

```
            "Cannot open output file.\n");
        return 1;
    }
    getftime(fileno(stream), &ft);
    printf("File time: %u:%u:%u\n",
            ft.ft_hour, ft.ft_min,
            ft.ft_tsec * 2);
    printf("File date: %u/%u/%u\n",
    ft.ft_month, ft.ft_day,
    ft.ft_year+1980);
    fclose(stream);
    return 0;
}
```

===========================

函数名: getgraphmode

===========================

功　能: 返回当前图形模式
用　法: int far getgraphmode(void);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
/* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy, mode;
    char numname[80], modename[80];

/* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

/* read result of initialization */
    errorcode = graphresult();
/* an error occurred */
    if (errorcode != grOk)
    {
```

```
        printf("Graphics error: %s\n",
                grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
/* terminate with an error code */
        exit(1);
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

/* get mode number and name strings */
    mode = getgraphmode();
    sprintf(numname,
            "%d is the current mode number.",
            mode);
    sprintf(modename,
            "%s is the current graphics mode",
            getmodename(mode));

/* display the information */
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(midx, midy, numname);
    outtextxy(midx, midy+2*textheight("W"),
                modename);

/* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: getftime

============================

功　能: 取文件日期和时间
用　法: int getftime(int handle, struct ftime *ftimep);
程序例:

```
#include <stdio.h>
#include <io.h>
```

```c
int main(void)
{
   FILE *stream;
   struct ftime ft;

   if ((stream = fopen("TEST.$$$",
        "wt")) == NULL)
   {
      fprintf(stderr,
             "Cannot open output file.\n");
      return 1;
   }
   getftime(fileno(stream), &ft);
   printf("File time: %u:%u:%u\n",
          ft.ft_hour, ft.ft_min,
          ft.ft_tsec * 2);
   printf("File date: %u/%u/%u\n",
   ft.ft_month, ft.ft_day,
   ft.ft_year+1980);
   fclose(stream);
   return 0;
}
```

============================

函数名: getgraphmode

============================

功　能: 返回当前图形模式
用　法: int far getgraphmode(void);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
/* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int midx, midy, mode;
   char numname[80], modename[80];
```

```c
/* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

/* read result of initialization */
   errorcode = graphresult();
/* an error occurred */
   if (errorcode != grOk)
   {
      printf("Graphics error: %s\n",
             grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
/* terminate with an error code */
      exit(1);
   }

   midx = getmaxx() / 2;
   midy = getmaxy() / 2;

/* get mode number and name strings */
   mode = getgraphmode();
   sprintf(numname,
           "%d is the current mode number.",
           mode);
   sprintf(modename,
           "%s is the current graphics mode",
           getmodename(mode));

/* display the information */
   settextjustify(CENTER_TEXT, CENTER_TEXT);
   outtextxy(midx, midy, numname);
   outtextxy(midx, midy+2*textheight("W"),
             modename);

/* clean up */
   getch();
   closegraph();
   return 0;
}
```

==============================

函数名: getimage

============================

功　能: 将指定区域的一个位图存到主存中

用　法: void far getimage(int left, int top, int right, int bottom,
　　　void far *bitmap);

程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <alloc.h>

void save_screen(void far *buf[4]);
void restore_screen(void far *buf[4]);

int maxx, maxy;

int main(void)
{
   int gdriver=DETECT, gmode, errorcode;
   void far *ptr[4];

   /* auto-detect the graphics driver and mode */
   initgraph(&gdriver, &gmode, "");
   errorcode = graphresult(); /* check for any errors */
   if (errorcode != grOk)
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1);
   }
   maxx = getmaxx();
   maxy = getmaxy();

   /* draw an image on the screen */
   rectangle(0, 0, maxx, maxy);
   line(0, 0, maxx, maxy);
   line(0, maxy, maxx, 0);

   save_screen(ptr);     /* save the current screen */
   getch();              /* pause screen */
   cleardevice();        /* clear screen */
```

```c
    restore_screen(ptr); /* restore the screen */
    getch();             /* pause screen */

    closegraph();
    return 0;
}

void save_screen(void far *buf[4])
{
    unsigned size;
    int ystart=0, yend, yincr, block;

    yincr = (maxy+1) / 4;
    yend = yincr;
    size = imagesize(0, ystart, maxx, yend); /* get byte size of image */

    for (block=0; block<=3; block++)
    {
        if ((buf[block] = farmalloc(size)) == NULL)
        {
            closegraph();
            printf("Error: not enough heap space in save_screen().\n");
  exit(1);
        }

        getimage(0, ystart, maxx, yend, buf[block]);
        ystart = yend + 1;
        yend += yincr + 1;
    }
}

void save_screen(void far *buf[4])
{
    unsigned size;
    int ystart=0, yend, yincr, block;

    yincr = (maxy+1) / 4;
    yend = yincr;
    size = imagesize(0, ystart, maxx, yend); /* get byte size of image */

    for (block=0; block<=3; block++)
    {
        if ((buf[block] = farmalloc(size)) == NULL)
        {
```

```
            closegraph();
            printf("Error: not enough heap space in save_screen().\n");
    exit(1);
        }

        getimage(0, ystart, maxx, yend, buf[block]);
        ystart = yend + 1;
        yend += yincr + 1;
    }
}


void restore_screen(void far *buf[4])
{
    int ystart=0, yend, yincr, block;

    yincr = (maxy+1) / 4;
    yend = yincr;

    for (block=0; block<=3; block++)
    {
        putimage(0, ystart, buf[block], COPY_PUT);
        farfree(buf[block]);
        ystart = yend + 1;
        yend += yincr + 1;
    }
}
```

=============================

函数名: getlinesettings

=============================

功　能: 取当前线型、模式和宽度
用　法: void far getlinesettings(struct linesettingstype far *lininfo):
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/* the names of the line styles supported */
char *lname[] = { "SOLID_LINE",
                  "DOTTED_LINE",
```

```c
                    "CENTER_LINE",
                    "DASHED_LINE",
                    "USERBIT_LINE"
                };

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   struct linesettingstype lineinfo;
   int midx, midy;
   char lstyle[80], lpattern[80], lwidth[80];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   midx = getmaxx() / 2;
   midy = getmaxy() / 2;

   /* get information about current line settings */
   getlinesettings(&lineinfo);

   /* convert line information into strings */
   sprintf(lstyle, "%s is the line style.",
           lname[lineinfo.linestyle]);
   sprintf(lpattern, "0x%X is the user-defined line pattern.",
           lineinfo.upattern);
   sprintf(lwidth, "%d is the line thickness.",
    lineinfo.thickness);

   /* display the information */
   settextjustify(CENTER_TEXT, CENTER_TEXT);
   outtextxy(midx, midy, lstyle);
   outtextxy(midx, midy+2*textheight("W"), lpattern);
```

```
   outtextxy(midx, midy+4*textheight("W"), lwidth);

   /* clean up */
   getch();
   closegraph();
   return 0;
}
```

============================

函数名: getmaxcolor

============================

功 能: 返回可以传给函数 setcolor 的最大颜色值
用 法: int far getmaxcolor(void);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int midx, midy;
   char colstr[80];

   /* initialize graphics and local variables
 */ initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   midx = getmaxx() / 2;
   midy = getmaxy() / 2;
```

```c
   /* grab the color info. and convert it to a string */
   sprintf(colstr, "This mode supports colors 0..%d", getmaxcolor());

   /* display the information */
   settextjustify(CENTER_TEXT, CENTER_TEXT);
   outtextxy(midx, midy, colstr);

   /* clean up */
   getch();
   closegraph();
   return 0;
}
```

==========================

函数名: getmaxx

==========================

功　能: 返回屏幕的最大 x 坐标
用　法: int far getmaxx(void);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int midx, midy;
   char xrange[80], yrange[80];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
```

```
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;

/* convert max resolution values into strings */
sprintf(xrange, "X values range from 0..%d", getmaxx());
sprintf(yrange, "Y values range from 0..%d", getmaxy());

/* display the information */
settextjustify(CENTER_TEXT, CENTER_TEXT);
outtextxy(midx, midy, xrange);
outtextxy(midx, midy+textheight("W"), yrange);

/* clean up */
getch();
closegraph();
return 0;
}
```

============================

函数名: getmaxy

============================

功　能: 返回屏幕的最大 y 坐标
用　法: int far getmaxy(void);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;
    char xrange[80], yrange[80];
```

```
    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* convert max resolution values into strings */
    sprintf(xrange, "X values range from 0..%d", getmaxx());
    sprintf(yrange, "Y values range from 0..%d", getmaxy());

    /* display the information */
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(midx, midy, xrange);
    outtextxy(midx, midy+textheight("W"), yrange);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: getmodename

============================
功　能: 返回含有指定图形模式名的字符串指针
用　法: char *far getmodename(int mode_name);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
```

```
int main(void)
{
    /* request autodetection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy, mode;
    char numname[80], modename[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* get mode number and name strings */
    mode = getgraphmode();
    sprintf(numname, "%d is the current mode number.", mode);
    sprintf(modename, "%s is the current graphics mode.", getmodename(mode));

    /* display the information */
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(midx, midy, numname);
    outtextxy(midx, midy+2*textheight("W"), modename);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: getmoderange

============================

功　能：取给定图形驱动程序的模式范围

用 法: void far getmoderange(int graphdriver, int far *lomode,
    int far *himode);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int midx, midy;
   int low, high;
   char mrange[80];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   midx = getmaxx() / 2;
   midy = getmaxy() / 2;

   /* get the mode range for this driver */
   getmoderange(gdriver, &low, &high);

   /* convert mode range info. into strings */
   sprintf(mrange, "This driver supports modes %d..%d", low, high);

   /* display the information */
   settextjustify(CENTER_TEXT, CENTER_TEXT);
   outtextxy(midx, midy, mrange);

   /* clean up */
```

```
    getch();
    closegraph();
    return 0;
}
```

==========================

函数名: getpalette

==========================

功　能: 返回有关当前调色板的信息
用　法: void far getpalette(struct palettetype far *palette);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
/* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    struct palettetype pal;
    char psize[80], pval[20];
    int i, ht;
    int y = 10;

/* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

/* read result of initialization */
    errorcode = graphresult();
/* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
                grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
/* terminate with an error code */
        exit(1);
    }
```

```
/* grab a copy of the palette */
   getpalette(&pal);

/* convert palette info. into strings */
   sprintf(psize, "The palette has %d \
           modifiable entries.", pal.size);

/* display the information */
   outtextxy(0, y, psize);
   if (pal.size != 0)
   {
      ht = textheight("W");
      y += 2*ht;
      outtextxy(0, y, "Here are the current \
  values:");
      y += 2*ht;
      for (i=0; i<pal.size; i++, y+=ht)
      {
  sprintf(pval,
   "palette[%02d]: 0x%02X", i,
   pal.colors[i]);
  outtextxy(0, y, pval);
      }
   }

/* clean up */
   getch();
   closegraph();
   return 0;
}
```

============================

函数名: getpass

============================

功　能: 读一个口令

用　法: char *getpass(char *prompt);

程序例:

```
#include <conio.h>

int main(void)
{
   char *password;
```

```
    password = getpass("Input a password:");
    cprintf("The password is: %s\r\n",
     password);
    return 0;
}
```

============================

函数名: getpixel

============================

功　能: 取得指定像素的颜色
用　法: int far getpixel(int x, int y);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <dos.h>

#define PIXEL_COUNT 1000
#define DELAY_TIME  100  /* in milliseconds */

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int i, x, y, color, maxx, maxy,
        maxcolor, seed;

/* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

/* read result of initialization */
    errorcode = graphresult();
/* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
                grapherrormsg(errorcode));
        printf("Press any key to halt:");
```

```
      getch();
/* terminate with an error code */
      exit(1);
   }

   maxx = getmaxx() + 1;
   maxy = getmaxy() + 1;
   maxcolor = getmaxcolor() + 1;

   while (!kbhit())
   {
/* seed the random number generator */
      seed = random(32767);
      srand(seed);
      for (i=0; i<PIXEL_COUNT; i++)
      {
         x = random(maxx);
         y = random(maxy);
         color = random(maxcolor);
         putpixel(x, y, color);
      }

      delay(DELAY_TIME);
      srand(seed);
      for (i=0; i<PIXEL_COUNT; i++)
      {
         x = random(maxx);
         y = random(maxy);
         color = random(maxcolor);
         if                 (color               ==               getpixel
```

匆? ;
; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;h;(;);;; ;e;t;p;s;p;(;););;; ; ; ; ; ; ; ; ; ; ;
; ;
; ;
; ;
; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;e;s;e;t; ;t;o; ;s;e;g;m;e;n;t; ;o;f; ;t;h;e; ;P
;S;P; ;i;n;e; ;i;s; ;l;o;c
;a;t;e;d; ;a;t; ;o;f;f;s;e;t; ;0;x;8;1; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;
; ; ; ; ;t; ;o;f; ;P;S;P; ; ; ; ; ; ; ;

============================

函数名: gets

============================

功　能: 从流中取一字符串
用　法: char *gets(char *string);
程序例:

```c
#include <stdio.h>

int main(void)
{
    char string[80];

    printf("Input a string:");
    gets(string);
    printf("The string input was: %s\n",
    string);
    return 0;
}
```

==============================

函数名: gettext

==============================

功　能: 将文本方式屏幕上的文本拷贝到存储区
用　法: int gettext(int left, int top, int right, int bottom, void *destin);
程序例:

```c
#include <conio.h>

char buffer[4096];

int main(void)
{
    int i;
    clrscr();
    for (i = 0; i <= 20; i++)
        cprintf("Line #%d\r\n", i);
    gettext(1, 1, 80, 25, buffer);
    gotoxy(1, 25);
    cprintf("Press any key to clear screen...");
    getch();
    clrscr();
    gotoxy(1, 25);
    cprintf("Press any key to restore screen...");
    getch();
```

```
    puttext(1, 1, 80, 25, buffer);
    gotoxy(1, 25);
    cprintf("Press any key to quit...");
    getch();
    return 0;
}
```

============================

函数名: gettextinfo

============================

功 能: 取得文本模式的显示信息
用 法: void gettextinfo(struct text_info *inforec);
程序例:

```
#include <conio.h>

int main(void)
{
    struct text_info ti;
    gettextinfo(&ti);
    cprintf("window left      %2d\r\n",ti.winleft);
    cprintf("window top       %2d\r\n",ti.wintop);
    cprintf("window right     %2d\r\n",ti.winright);
    cprintf("window bottom    %2d\r\n",ti.winbottom);
    cprintf("attribute        %2d\r\n",ti.attribute);
    cprintf("normal attribute %2d\r\n",ti.normattr);
    cprintf("current mode     %2d\r\n",ti.currmode);
    cprintf("screen height    %2d\r\n",ti.screenheight);
    cprintf("screen width     %2d\r\n",ti.screenwidth);
    cprintf("current x        %2d\r\n",ti.curx);
    cprintf("current y        %2d\r\n",ti.cury);
    return 0;
}
```

============================

函数名: gettextsettings

============================

功 能: 返回有关当前图形文本字体的信息
用 法: void far gettextsettings(struct textsettingstype far *textinfo);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/* the names of the fonts supported */
char *font[] = { "DEFAULT_FONT",
                 "TRIPLEX_FONT",
                 "SMALL_FONT",
                 "SANS_SERIF_FONT",
                 "GOTHIC_FONT"
               };

/* the names of the text directions supported */
char *dir[] = { "HORIZ_DIR", "VERT_DIR" };

/* horizontal text justifications supported */
char *hjust[] = { "LEFT_TEXT", "CENTER_TEXT", "RIGHT_TEXT" };

/* vertical text justifications supported */
char *vjust[] = { "BOTTOM_TEXT", "CENTER_TEXT", "TOP_TEXT" };

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    struct textsettingstype textinfo;
    int midx, midy, ht;
    char fontstr[80], dirstr[80], sizestr[80];
    char hjuststr[80], vjuststr[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }
```

```
    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* get information about current text settings */
    gettextsettings(&textinfo);

    /* convert text information into strings */
    sprintf(fontstr, "%s is the text style.", font[textinfo.font]);
    sprintf(dirstr, "%s is the text direction.", dir[textinfo.direction]);
    sprintf(sizestr, "%d is the text size.", textinfo.charsize);
    sprintf(hjuststr, "%s is the horizontal justification.",
            hjust[textinfo.horiz]);
    sprintf(vjuststr, "%s is the vertical justification.",
            vjust[textinfo.vert]);

    /* display the information */
    ht = textheight("W");
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(midx, midy, fontstr);
    outtextxy(midx, midy+2*ht, dirstr);
    outtextxy(midx, midy+4*ht, sizestr);
    outtextxy(midx, midy+6*ht, hjuststr);
    outtextxy(midx, midy+8*ht, vjuststr);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: gettime

============================

功　能: 取得系统时间
用　法: void gettime(struct time *timep);
程序例:

```
#include  <stdio.h>
#include  <dos.h>

int main(void)
{
```

```
    struct  time t;

    gettime(&t);
    printf("The current time is: %2d:%02d:%02d.%02d\n",
            t.ti_hour, t.ti_min, t.ti_sec, t.ti_hund);
    return 0;
}
```

============================

函数名: getvect

============================

功　能: 取得中断向量入口
用　法: void interrupt(*getvect(int intr_num));
程序例:

```c
#include <stdio.h>
#include <dos.h>

void interrupt get_out(); /* interrupt prototype */

void interrupt (*oldfunc)(); /* interrupt function pointer */
int looping = 1;

int main(void)
{
  puts("Press <Shift><Prt Sc> to terminate");

  /* save the old interrupt */
  oldfunc  = getvect(5);

   /* install interrupt handler */
  setvect(5,get_out);

   /* do nothing */
  while (looping);

   /* restore to original interrupt routine */
   setvect(5,oldfunc);

  puts("Success");
  return 0;
```

```
}
void interrupt get_out()
{
   looping = 0; /* change global variable to get out of loop */
}
```

============================

函数名: **getverify**

============================

功　能: 返回 DOS 校验标志状态
用　法: int getverify(void);
程序例:

```
#include <stdio.h>
#include <dos.h>

int main(void)
{
   if (getverify())
      printf("DOS verify flag is on\n");
   else
      printf("DOS verify flag is off\n");
   return 0;
}
```

函数名: **getviewsetting**

功　能: 返回有关当前视区的信息
用　法: void far getviewsettings(struct viewporttype far *viewport);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

char *clip[] = { "OFF", "ON" };

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
```

```c
    struct viewporttype viewinfo;
    int midx, midy, ht;
    char topstr[80], botstr[80], clipstr[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* get information about current viewport */
    getviewsettings(&viewinfo);

    /* convert text information into strings */
    sprintf(topstr, "(%d, %d) is the upper left viewport corner.",
            viewinfo.left, viewinfo.top);
    sprintf(botstr, "(%d, %d) is the lower right viewport corner.",
            viewinfo.right, viewinfo.bottom);
    sprintf(clipstr, "Clipping is turned %s.", clip[viewinfo.clip]);

    /* display the information */
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    ht = textheight("W");
    outtextxy(midx, midy, topstr);
    outtextxy(midx, midy+2*ht, botstr);
    outtextxy(midx, midy+4*ht, clipstr);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

==========================

函数名: getw

==========================
功 能: 从流中取一整数
用 法: int getw(FILE *strem);
程序例:

```c
#include <stdio.h>
#include <stdlib.h>

#define FNAME "test.$$$"

int main(void)
{
   FILE *fp;
   int word;

   /* place the word in a file */
   fp = fopen(FNAME, "wb");
   if (fp == NULL)
   {
      printf("Error opening file %s\n", FNAME);
      exit(1);
   }

   word = 94;
   putw(word,fp);
   if (ferror(fp))
       printf("Error writing to file\n");
   else
       printf("Successful write\n");
   fclose(fp);

   /* reopen the file */
   fp = fopen(FNAME, "rb");
   if (fp == NULL)
   {
      printf("Error opening file %s\n", FNAME);
      exit(1);
   }

   /* extract the word */
   word = getw(fp);
```

```
    if (ferror(fp))
        printf("Error reading file\n");
    else
        printf("Successful read: word = %d\n", word);

    /* clean up */
    fclose(fp);
    unlink(FNAME);

    return 0;
}
```

============================

函数名: getx

============================

功　能: 返回当前图形位置的 x 坐标
用　法: int far getx(void);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    char msg[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
```

```
   }

   /* move to the screen center point */
   moveto(getmaxx() / 2, getmaxy() / 2);

   /* create a message string */
   sprintf(msg, "<-(%d, %d) is the here.", getx(), gety());

   /* display the message */
   outtext(msg);

   /* clean up */
   getch();
   closegraph();
   return 0;
}
```

============================

函数名: gety

============================

功 能: 返回当前图形位置的 y 坐标
用 法: int far gety(void);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   char msg[80];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
```

```
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

    /* move to the screen center point */
    moveto(getmaxx() / 2, getmaxy() / 2);

    /* create a message string */
    sprintf(msg, "<-(%d, %d) is the here.", getx(), gety());

    /* display the message */
    outtext(msg);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: gmtime

============================

功　能: 把日期和时间转换为格林尼治标准时间(GMT)
用　法: struct tm *gmtime(long *clock);
程序例:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <dos.h>

/* Pacific Standard Time & Daylight Savings */
char *tzstr = "TZ=PST8PDT";

int main(void)
{
    time_t t;
    struct tm *gmt, *area;

    putenv(tzstr);
```

```c
    tzset();

    t = time(NULL);
    area = localtime(&t);
    printf("Local time is: %s", asctime(area));
    gmt = gmtime(&t);
    printf("GMT is:        %s", asctime(gmt));
    return 0;
}
```

============================

函数名: gotoxy

============================

功　能: 在文本窗口中设置光标
用　法: void gotoxy(int x, int y);
程序例:

```c
#include <conio.h>

int main(void)
{
    clrscr();
    gotoxy(35, 12);
    cprintf("Hello world");
    getch();
    return 0;
}
```

============================

函数名: gotoxy

============================

功　能: 在文本窗口中设置光标
用　法: void gotoxy(int x, int y);
程序例:

```c
#include <conio.h>

int main(void)
{
    clrscr();
```

```c
   gotoxy(35, 12);
   cprintf("Hello world");
   getch();
   return 0;
}
```

==========================

函数名: graphdefaults

==========================

功　能: 将所有图形设置复位为它们的缺省值
用　法: void far graphdefaults(void);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int maxx, maxy;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "c:\\bor\\Borland\\bgi");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   maxx = getmaxx();
   maxy = getmaxy();

   /* output line with non-default settings */
   setlinestyle(DOTTED_LINE, 0, 3);
```

```
    line(0, 0, maxx, maxy);
    outtextxy(maxx/2, maxy/3, "Before default values are restored.");
    getch();

    /* restore default values for everything */
    graphdefaults();

    /* clear the screen */
    cleardevice();

    /* output line with default settings */
    line(0, 0, maxx, maxy);
    outtextxy(maxx/2, maxy/3, "After restoring default values.");

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

===========================

函数名: grapherrormsg

===========================

功　能: 返回一个错误信息串的指针

用　法: char *far grapherrormsg(int errorcode);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

#define NONSENSE -50

int main(void)
{
    /* FORCE AN ERROR TO OCCUR */
    int gdriver = NONSENSE, gmode, errorcode;

    /* initialize graphics mode */
    initgraph(&gdriver, &gmode, "");
```

```
   /* read result of initialization */
   errorcode = graphresult();

   /* if an error occurred, then output a */
   /* descriptive error message.          */
   if (errorcode != grOk)
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   /* draw a line */
   line(0, 0, getmaxx(), getmaxy());

   /* clean up */
   getch();
   closegraph();
   return 0;
}
```

==========================

函数名: graphresult

==============================

功　能: 返回最后一次不成功的图形操作的错误代码
用　法: int far graphresult(void);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");
```

```
   /* read result of initialization */
   errorcode = graphresult();

   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   /* draw a line */
   line(0, 0, getmaxx(), getmaxy());

   /* clean up */
   getch();
   closegraph();
   return 0;
}
```

============================

函数名: _graphfreemem

============================

功　能: 用户可修改的图形存储区释放函数
用　法: void far _graphfreemem(void far *ptr, unsigned size);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <alloc.h>

int main(void)
{
     /* request auto detection */
     int gdriver = DETECT, gmode, errorcode;
     int midx, midy;

     /* clear the text screen */
     clrscr();
     printf("Press any key to initialize graphics mode:");
     getch();
```

```c
        clrscr();

        /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

        /* read result of initialization */
        errorcode = graphresult();
        if (errorcode != grOk)  /* an error occurred */
        {
            printf("Graphics error: %s\n", grapherrormsg(errorcode));
            printf("Press any key to halt:");
            getch();
            exit(1); /* terminate with an error code */
        }

        midx = getmaxx() / 2;
        midy = getmaxy() / 2;

        /* display a message */
        settextjustify(CENTER_TEXT, CENTER_TEXT);
        outtextxy(midx, midy, "Press any key to exit graphics mode:");

        /* clean up */
        getch();
        closegraph();
        return 0;
}

/* called by the graphics kernel to allocate memory */
void far * far _graphgetmem(unsigned size)
{
        printf("_graphgetmem called to allocate %d bytes.\n", size);
        printf("hit any key:");
        getch();
        printf("\n");

        /* allocate memory from far heap */
        return farmalloc(size);
}

/* called by the graphics kernel to free memory */
void far _graphfreemem(void far *ptr, unsigned size)
{
        printf("_graphfreemem called to free %d bytes.\n", size);
```

```c
    printf("hit any key:");
    getch();
    printf("\n");

    /* free ptr from far heap */
    farfree(ptr);
}
```

============================

函数名: _graphgetmem

============================

功　能: 用户可修改的图形存储区分配函数
用　法: void far *far _graphgetmem(unsigned size);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <alloc.h>

int main(void)
{
    /* request autodetection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;

    /* clear the text screen */
    clrscr();
    printf("Press any key to initialize graphics mode:");
    getch();
    clrscr();

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)      /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
```

```c
        exit(1);                      /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* display a message */
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(midx, midy, "Press any key to exit graphics mode:");

    /* clean up */
    getch();
    closegraph();
    return 0;
}


/* called by the graphics kernel to allocate memory */
void far * far _graphgetmem(unsigned size)
{
    printf("_graphgetmem called to allocate %d bytes.\n", size);
    printf("hit any key:");
    getch();
    printf("\n");

    /* allocate memory from far heap */
    return farmalloc(size);
}


/* called by the graphics kernel to free memory */
void far _graphfreemem(void far *ptr, unsigned size)
{
    printf("_graphfreemem called to free %d bytes.\n", size);
    printf("hit any key:");
    getch();
    printf("\n");

    /* free ptr from far heap */
    farfree(ptr);
}
```

============================

函数名: harderr

============================

功　能: 建立一个硬件错误处理程序

用　法: void harderr(int (*fptr)());

程序例:

```
/*This program will trap disk errors and prompt
the user for action. Try running it with no
disk in drive A: to invoke its functions.*/

#include <stdio.h>
#include <conio.h>
#include <dos.h>
#define IGNORE   0
#define RETRY    1
#define ABORT    2
int buf[500];
/*define the error messages for trapping disk problems*/
static char *err_msg[] = {
    "write protect",
    "unknown unit",
    "drive not ready",
    "unknown command",
    "data error (CRC)",
    "bad request",
    "seek error",
    "unknown media type",
    "sector not found",
    "printer out of paper",
    "write fault",
    "read fault",
    "general failure",
    "reserved",
    "reserved",
    "invalid disk change"
};

error_win(char *msg)
{
    int retval;

    cputs(msg);

/*prompt for user to press a key to abort, retry, ignore*/
    while(1)
```

```c
    {
        retval= getch();
        if (retval == 'a' || retval == 'A')
        {
     retval = ABORT;
     break;
        }
        if (retval == 'r' || retval == 'R')
        {
     retval = RETRY;
     break;
        }
        if (retval == 'i' || retval == 'I')
        {
            retval = IGNORE;
            break;
        }
    }

    return(retval);
}


/*pragma warn -par reduces warnings which occur
due to the non use of the parameters errval,
bp and si to the handler.*/
#pragma warn -par

int handler(int errval,int ax,int bp,int si)
{
    static char msg[80];
    unsigned di;
    int drive;
    int errorno;
    di= _DI;
/*if this is not a disk error then it was
another device having trouble*/

    if (ax < 0)
    {
        /* report the error */
        error_win("Device error");
        /* and return to the program directly requesting abort */
        hardretn(ABORT);
    }
```

```
/* otherwise it was a disk error */
   drive = ax & 0x00FF;
   errorno = di & 0x00FF;
/* report which error it was */
   sprintf(msg, "Error: %s on drive %c\r\nA)bort, R)etry, I)gnore: ",
    err_msg[errorno], 'A' + drive);
/*
return to the program via dos interrupt 0x23 with abort, retry,
or ignore as input by the user.
*/
   hardresume(error_win(msg));
   return ABORT;
}
#pragma warn +par

int main(void)
{
/*
install our handler on the hardware problem interrupt
*/
   harderr(handler);
   clrscr();
   printf("Make sure there is no disk in drive A:\n");
   printf("Press any key ....\n");
   getch();
   printf("Trying to access drive A:\n");
   printf("fopen returned %p\n",fopen("A:temp.dat", "w"));
   return 0;
}
```

============================

函数名: hardresume

============================

功　能: 硬件错误处理函数
用　法: void hardresume(int rescode);
程序例:


```
/* This program will trap disk errors and prompt the user for action. */
/* Try running it with no disk in drive A: to invoke its functions    */

#include <stdio.h>
```

```c
#include <conio.h>
#include <dos.h>

#define IGNORE  0
#define RETRY   1
#define ABORT   2

int buf[500];

/* define the error messages for trapping disk problems */
static char *err_msg[] = {
    "write protect",
    "unknown unit",
    "drive not ready",
    "unknown command",
    "data error (CRC)",
    "bad request",
    "seek error",
    "unknown media type",
    "sector not found",
    "printer out of paper",
    "write fault",
    "read fault",
    "general failure",
    "reserved",
    "reserved",
    "invalid disk change"
};

error_win(char *msg)
{
    int retval;

    cputs(msg);

/* prompt for user to press a key to abort, retry, ignore */
    while(1)
    {
        retval= getch();
        if (retval == 'a' || retval == 'A')
        {
            retval = ABORT;
            break;
        }
```

```c
        if (retval == 'r' || retval == 'R')
        {
            retval = RETRY;
            break;
        }
        if (retval == 'i' || retval == 'I')
        {
            retval = IGNORE;
            break;
        }
    }

    return(retval);
}

/* pragma warn -par reduces warnings which occur due to the non use */
/* of the parameters errval, bp and si to the handler.             */
#pragma warn -par

int handler(int errval,int ax,int bp,int si)
{
    static char msg[80];
    unsigned di;
    int drive;
    int errorno;

    di= _DI;
/* if this is not a disk error then it was another device having trouble */

    if (ax < 0)
    {
        /* report the error */
        error_win("Device error");
        /* and return to the program directly
        requesting abort */
        hardretn(ABORT);
    }
/* otherwise it was a disk error */
    drive = ax & 0x00FF;
    errorno = di & 0x00FF;
/* report which error it was */
    sprintf(msg, "Error: %s on drive %c\r\nA)bort, R)etry, I)gnore: ",
            err_msg[errorno], 'A' + drive);
/* return to the program via dos interrupt 0x23 with abort, retry */
```

```
/* or ignore as input by the user.  */
    hardresume(error_win(msg));
    return ABORT;
}
#pragma warn +par

int main(void)
{
/* install our handler on the hardware problem interrupt */
    harderr(handler);
    clrscr();
    printf("Make sure there is no disk in drive A:\n");
    printf("Press any key ....\n");
    getch();
    printf("Trying to access drive A:\n");
    printf("fopen returned %p\n",fopen("A:temp.dat", "w"));
    return 0;
}
```

==========================

函数名: highvideo

==========================

功　能: 选择高亮度文本字符
用　法: void highvideo(void);
程序例:

```
#include <conio.h>

int main(void)
{
    clrscr();

    lowvideo();
    cprintf("Low Intensity text\r\n");
    highvideo();
    gotoxy(1,2);
    cprintf("High Intensity Text\r\n");

    return 0;
}
```

===========================

函数名: hypot

===========================
功　能: 计算直角三角形的斜边长
用　法: double hypot(double x, double y);
程序例:

```c
#include <stdio.h>
#include <math.h>

int main(void)
{
   double result;
   double x = 3.0;
   double y = 4.0;

   result = hypot(x, y);
   printf("The hypotenuse is: %lf\n", result);

   return 0;
}
```


===========================

函数名: imagesize

===========================
功　能: 返回保存位图像所需的字节数
用　法: unsigned far imagesize(int left, int top, int right, int bottom);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

#define ARROW_SIZE 10

void draw_arrow(int x, int y);

int main(void)
{
```

```c
/* request autodetection */
int gdriver = DETECT, gmode, errorcode;
void *arrow;
int x, y, maxx;
unsigned int size;

/* initialize graphics and local variables */
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk)  /* an error occurred */
{
   printf("Graphics error: %s\n", grapherrormsg(errorcode));
   printf("Press any key to halt:");
   getch();
   exit(1); /* terminate with an error code */
}

maxx = getmaxx();
x = 0;
y = getmaxy() / 2;

/* draw the image to be grabbed */
draw_arrow(x, y);

/* calculate the size of the image */
size = imagesize(x, y-ARROW_SIZE, x+(4*ARROW_SIZE), y+ARROW_SIZE);

/* allocate memory to hold the image */
arrow = malloc(size);

/* grab the image */
getimage(x, y-ARROW_SIZE, x+(4*ARROW_SIZE), y+ARROW_SIZE, arrow);

/* repeat until a key is pressed */
while (!kbhit())
{
   /* erase old image */
   putimage(x, y-ARROW_SIZE, arrow, XOR_PUT);

   x += ARROW_SIZE;
   if (x >= maxx)
       x = 0;
```

```
        /* plot new image */
        putimage(x, y-ARROW_SIZE, arrow, XOR_PUT);
    }

    /* clean up */
    free(arrow);
    closegraph();
    return 0;
}

void draw_arrow(int x, int y)
{
    /* draw an arrow on the screen */
    moveto(x, y);
    linerel(4*ARROW_SIZE, 0);
    linerel(-2*ARROW_SIZE, -1*ARROW_SIZE);
    linerel(0, 2*ARROW_SIZE);
    linerel(2*ARROW_SIZE, -1*ARROW_SIZE);
}
```

==========================

函数名: initgraph

==========================

功　能: 初始化图形系统
用　法: void far initgraph(int far *graphdriver, int far *graphmode,
    char far *pathtodriver);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;

    /* initialize graphics mode */
    initgraph(&gdriver, &gmode, "");
```

```c
    /* read result of initialization */
    errorcode = graphresult();

    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1);                /* return with error code */
    }

    /* draw a line */
    line(0, 0, getmaxx(), getmaxy());

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: inport

============================

功　能: 从硬件端口中输入
用　法: int inp(int protid);
程序例:

```c
#include <stdio.h>
#include <dos.h>

int main(void)
{
    int result;
    int port = 0;  /* serial port 0 */

    result = inport(port);
    printf("Word read from port %d = 0x%X\n", port, result);
    return 0;
}
```

============================

函数名: insline

============================

功　能: 在文本窗口中插入一个空行
用　法: void insline(void);
程序例:

```
#include <conio.h>

int main(void)
{
   clrscr();
   cprintf("INSLINE inserts an empty line in the text window\r\n");
   cprintf("at the cursor position using the current text\r\n");
   cprintf("background color.  All lines below the empty one\r\n");
   cprintf("move down one line and the bottom line scrolls\r\n");
   cprintf("off the bottom of the window.\r\n");
   cprintf("\r\nPress any key to continue:");
   gotoxy(1, 3);
   getch();
   insline();
   getch();
   return 0;
}
```

============================

函数名: installuserdriver

============================

功　能: 安装设备驱动程序到 BGI 设备驱动程序表中
用　法: int far installuserdriver(char far *name, int (*detect)(void));
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/* function prototypes */
int huge detectEGA(void);
void checkerrors(void);
```

```c
int main(void)
{
   int gdriver, gmode;

   /* install a user written device driver */
   gdriver = installuserdriver("EGA", detectEGA);

   /* must force use of detection routine */
   gdriver = DETECT;

   /* check for any installation errors */
   checkerrors();

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* check for any initialization errors */
   checkerrors();

   /* draw a line */
   line(0, 0, getmaxx(), getmaxy());

   /* clean up */
   getch();
   closegraph();
   return 0;
}

/* detects EGA or VGA cards */
int huge detectEGA(void)
{
   int driver, mode, sugmode = 0;

   detectgraph(&driver, &mode);
   if ((driver == EGA) || (driver == VGA))
      /* return suggested video mode number */
      return sugmode;
   else
      /* return an error code */
      return grError;
}

/* check for and report any graphics errors */
```

```
void checkerrors(void)
{
    int errorcode;

    /* read result of last graphics operation */
    errorcode = graphresult();
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1);
    }
}
```
============================

函数名: installuserfont

============================
功　能: 安装未嵌入 BGI 系统的字体文件(CHR)
用　法: int far installuserfont(char far *name);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/* function prototype */
void checkerrors(void);

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode;
    int userfont;
    int midx, midy;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* check for any initialization errors */
```

```c
    checkerrors();

    /* install a user defined font file */
    userfont = installuserfont("USER.CHR");

    /* check for any installation errors */
    checkerrors();

    /* select the user font */
    settextstyle(userfont, HORIZ_DIR, 4);

    /* output some text */
    outtextxy(midx, midy, "Testing!");

    /* clean up */
    getch();
    closegraph();
    return 0;
}

/* check for and report any graphics errors */
void checkerrors(void)
{
    int errorcode;

    /* read result of last graphics operation */
    errorcode = graphresult();
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1);
    }
}
```

==============================

函数名: int86

==============================

功　能: 通用 8086 软中断接口
用　法: int int86(int intr_num, union REGS *inregs, union REGS *outregs);

程序例:

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>

#define VIDEO 0x10

void movetoxy(int x, int y)
{
    union REGS regs;

    regs.h.ah = 2;  /* set cursor postion */
    regs.h.dh = y;
    regs.h.dl = x;
    regs.h.bh = 0;  /* video page 0 */
    int86(VIDEO, &regs, &regs);
}

int main(void)
{
    clrscr();
    movetoxy(35, 10);
    printf("Hello\n");
    return 0;
}
```

============================

函数名: int86x

============================

功　能: 通用 8086 软中断接口
用　法: int int86x(int intr_num, union REGS *insegs, union REGS *outregs,
　　struct SREGS *segregs);
程序例:

```
#include <dos.h>
#include <process.h>
#include <stdio.h>

int main(void)
{
    char filename[80];
```

```c
    union REGS inregs, outregs;
    struct SREGS segregs;

    printf("Enter filename: ");
    gets(filename);
    inregs.h.ah = 0x43;
    inregs.h.al = 0x21;
    inregs.x.dx = FP_OFF(filename);
    segregs.ds = FP_SEG(filename);
    int86x(0x21, &inregs, &outregs, &segregs);
    printf("File attribute: %X\n", outregs.x.cx);
    return 0;
}
```

===========================

函数名: intdos

===========================

功　能: 通用 DOS 接口
用　法: int intdos(union REGS *inregs, union REGS *outregs);
程序例:

```c
#include <stdio.h>
#include <dos.h>

/* deletes file name; returns 0 on success, nonzero on failure */
int delete_file(char near *filename)
{
    union REGS regs;
    int ret;
    regs.h.ah = 0x41;                              /* delete file */
    regs.x.dx = (unsigned) filename;
    ret = intdos(&regs, &regs);

    /* if carry flag is set, there was an error */
    return(regs.x.cflag ? ret : 0);
}

int main(void)
{
    int err;
    err = delete_file("NOTEXIST.$$$");
```

```c
    if (!err)
       printf("Able to delete NOTEXIST.$$$\n");
    else
       printf("Not Able to delete NOTEXIST.$$$\n");
    return 0;
}
```

============================

函数名: intdosx

============================

功　能: 通用 DOS 中断接口
用　法: int intdosx(union REGS *inregs, union REGS *outregs,
        struct SREGS *segregs);
程序例:

```c
#include <stdio.h>
#include <dos.h>

/* deletes file name; returns 0 on success, nonzero on failure */
int delete_file(char far *filename)
{
    union REGS regs; struct SREGS sregs;
    int ret;
    regs.h.ah = 0x41;                       /* delete file */
    regs.x.dx = FP_OFF(filename);
    sregs.ds = FP_SEG(filename);
    ret = intdosx(&regs, &regs, &sregs);

    /* if carry flag is set, there was an error */
    return(regs.x.cflag ? ret : 0);
}

int main(void)
{
    int err;
    err = delete_file("NOTEXIST.$$$");
    if (!err)
       printf("Able to delete NOTEXIST.$$$\n");
    else
       printf("Not Able to delete NOTEXIST.$$$\n");
    return 0;
```

```
}
```

==========================

函数名: intr

==========================

功　能: 改变软中断接口
用　法: void intr(int intr_num, struct REGPACK *preg);
程序例:

```c
#include <stdio.h>
#include <string.h>
#include <dir.h>
#include <dos.h>

#define CF 1   /* Carry flag */

int main(void)
{
   char directory[80];
   struct REGPACK reg;

   printf("Enter directory to change to: ");
   gets(directory);
   reg.r_ax = 0x3B << 8;           /* shift 3Bh into  AH */
   reg.r_dx = FP_OFF(directory);
   reg.r_ds = FP_SEG(directory);
   intr(0x21, &reg);
   if (reg.r_flags & CF)
      printf("Directory change failed\n");
   getcwd(directory, 80);
   printf("The current directory is: %s\n", directory);
   return 0;
}
```

==========================

函数名: ioctl

==========================

功　能: 控制 I/O 设备
用　法: int ioctl(int handle, int cmd[,int *argdx, int argcx]);
程序例:

```c
#include <stdio.h>
#include <dir.h>
#include <io.h>

int main(void)
{
   int stat;

   /* use func 8 to determine if the default drive is removable */
   stat = ioctl(0, 8, 0, 0);
   if (!stat)
      printf("Drive %c is removable.\n", getdisk() + 'A');
   else
      printf("Drive %c is not removable.\n", getdisk() + 'A');
   return 0;
}
```

============================

函数名: isatty

============================

功　能: 检查设备类型
用　法: int isatty(int handle);
程序例:

```c
#include <stdio.h>
#include <io.h>

int main(void)
{
   int handle;

   handle = fileno(stdprn);
   if (isatty(handle))
      printf("Handle %d is a device type\n", handle);
   else
      printf("Handle %d isn't a device type\n", handle);
   return 0;
}
```

============================

函数名: itoa

============================

功　能: 把一整数转换为字符串
用　法: char *itoa(int value, char *string, int radix);
程序例:

```c
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    int number = 12345;
    char string[25];

    itoa(number, string, 10);
    printf("integer = %d string = %s\n", number, string);
    return 0;
}
```

============================

函数名: kbhit

============================

功　能: 检查当前按下的键
用　法: int kbhit(void);
程序例:

```c
#include <conio.h>

int main(void)
{
    cprintf("Press any key to continue:");
    while (!kbhit()) /* do nothing */ ;
    cprintf("\r\nA key was pressed...\r\n");
    return 0;
}
```

功　能: 把一整数转换为字符串

============================

函数名: keep

============================

功　能: 退出并继续驻留
用　法: void keep(int status, int size);
程序例:

```
/***NOTE:
   This is an interrupt service routine.  You
   can NOT compile this program with Test
   Stack Overflow turned on and get an
   executable file which will operate
   correctly.  Due to the nature of this
   function the formula used to compute
   the number of paragraphs may not
   necessarily work in all cases.  Use with
   care!  Terminate Stay Resident (TSR)
   programs are complex and no other support
   for them is provided.  Refer to the
   MS-DOS technical documentation
   for more information.  */
#include <dos.h>
/* The clock tick interrupt */
#define INTR 0x1C
/* Screen attribute (blue on grey) */
#define ATTR 0x7900

/* reduce heaplength and stacklength
to make a smaller program in memory */
extern unsigned _heaplen = 1024;
extern unsigned _stklen  = 512;

void interrupt ( *oldhandler)(void);

void interrupt handler(void)
{
   unsigned int (far *screen)[80];
   static int count;

/* For a color screen the video memory
   is at B800:0000.  For a monochrome
   system use B000:000 */
   screen = MK_FP(0xB800,0);
```

```
/* increase the counter and keep it
   within 0 to 9 */
   count++;
   count %= 10;

/* put the number on the screen */
   screen[0][79] = count + '0' + ATTR;

/* call the old interrupt handler */
   oldhandler();
}

int main(void)
{

/* get the address of the current clock
   tick interrupt */
oldhandler = getvect(INTR);

/* install the new interrupt handler */
setvect(INTR, handler);

/* _psp is the starting address of the
   program in memory.  The top of the stack
   is the end of the program.  Using _SS and
   _SP together we can get the end of the
   stack.  You may want to allow a bit of
   saftey space to insure that enough room
   is being allocated ie:
   (_SS + ((_SP + safety space)/16) - _psp)
*/
keep(0, (_SS + (_SP/16) - _psp));
return 0;
}
```

============================

函数名: labs

============================

功　能: 取长整型绝对值
用　法: long labs(long n);

程序例:

```c
#include <stdio.h>
#include <math.h>

int main(void)
{
    long result;
    long x = -12345678L;

    result= labs(x);
    printf("number: %ld abs value: %ld\n",
        x, result);

    return 0;
}
```

============================

函数名: ldexp

============================

功　能: 计算 value*2 的幂
用　法: double ldexp(double value, int exp);
程序例:

```c
#include <stdio.h>
#include <math.h>

int main(void)
{
    double value;
    double x = 2;

    /* ldexp raises 2 by a power of 3
        then multiplies the result by 2   */
    value = ldexp(x,3);
    printf("The ldexp value is: %lf\n",
        value);

    return 0;
}
```

============================

函数名: ldiv

============================

功　能: 两个长整型数相除，返回商和余数
用　法: ldiv_t ldiv(long lnumer, long ldenom);
程序例:

```c
/* ldiv example */

#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    ldiv_t lx;

    lx = ldiv(100000L, 30000L);
    printf("100000 div 30000 = %ld remainder %ld\n", lx.quot, lx.rem);
    return 0;
}
```

============================

函数名: lfind

============================

功　能: 执行线性搜索
用　法: void *lfind(void *key, void *base, int *nelem, int width,
       int (*fcmp)());
程序例:

```c
#include <stdio.h>
#include <stdlib.h>

int compare(int *x, int *y)
{
    return( *x - *y );
}

int main(void)
{
```

```c
    int array[5] = {35, 87, 46, 99, 12};
    size_t nelem = 5;
    int key;
    int *result;

    key = 99;
    result = lfind(&key, array, &nelem,
          sizeof(int), (int(*)(const void *,const void *))compare);
    if (result)
       printf("Number %d found\n",key);
    else
       printf("Number %d not found\n",key);

    return 0;
}
```

==========================

函数名: line

==========================

功　能: 在指定两点间画一直线
用　法: void far line(int x0, int y0, int x1, int y1);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int xmax, ymax;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   /* an error occurred */
   if (errorcode != grOk)
   {
```

```c
        printf("Graphics error: %s\n",
                graperrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1);
    }

    setcolor(getmaxcolor());
    xmax = getmaxx();
    ymax = getmaxy();

    /* draw a diagonal line */
    line(0, 0, xmax, ymax);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: linerel

============================

功　能: 从当前位置点(CP)到与 CP 有一给定相对距离的点画一直线
用　法: void far linerel(int dx, int dy);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    char msg[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
```

```
    errorcode = graphresult();
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
    grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1);
    }

    /* move the C.P. to location (20, 30) */
    moveto(20, 30);

    /* create and output a
        message at (20, 30) */
    sprintf(msg, " (%d, %d)", getx(), gety());
    outtextxy(20, 30, msg);

    /* draw a line to a point a relative
        distance away from the current
        value of C.P.    */
    linerel(100, 100);

    /* create and output a message at C.P. */
    sprintf(msg, " (%d, %d)", getx(), gety());
    outtext(msg);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

==========================

函数名: localtime

==========================

功　能: 把日期和时间转变为结构
用　法: struct tm *localtime(long *clock);
程序例:

```
#include <time.h>
#include <stdio.h>
#include <dos.h>
```

```c
int main(void)
{
   time_t timer;
   struct tm *tblock;

   /* gets time of day */
   timer = time(NULL);

   /* converts date/time to a structure */
   tblock = localtime(&timer);

   printf("Local time is: %s", asctime(tblock));

   return 0;
}
```

============================

函数名: lock

============================

功　能: 设置文件共享锁
用　法: int lock(int handle, long offset, long length);
程序例:

```c
#include <io.h>
#include <fcntl.h>
#include <sys\stat.h>
#include <process.h>
#include <share.h>
#include <stdio.h>

int main(void)
{
   int handle, status;
   long length;

   /* Must have DOS Share.exe loaded for */
   /* file locking to function properly */

   handle = sopen("c:\\autoexec.bat",
      O_RDONLY,SH_DENYNO,S_IREAD);
```

```c
    if (handle < 0)
    {
        printf("sopen failed\n");
        exit(1);
    }

    length = filelength(handle);
    status = lock(handle,0L,length/2);

    if (status == 0)
        printf("lock succeeded\n");
    else
        printf("lock failed\n");

    status = unlock(handle,0L,length/2);

    if (status == 0)
        printf("unlock succeeded\n");
    else
        printf("unlock failed\n");

    close(handle);
    return 0;
}
```

==========================

函数名: log

==========================

功　能: 对数函数 ln(x)
用　法: double log(double x);
程序例:

```c
#include <math.h>
#include <stdio.h>

int main(void)
{
    double result;
    double x = 8.6872;

    result = log(x);
```

```
    printf("The natural log of %lf is %lf\n", x, result);

    return 0;
}
```

==============================

函数名: log10

==============================

功　能: 对数函数 log
用　法: double log10(double x);
程序例:

```
#include <math.h>
#include <stdio.h>

int main(void)
{
    double result;
    double x = 800.6872;

    result = log10(x);
    printf("The common log of %lf is %lf\n", x, result);

    return 0;
}
```

==============================

函数名: longjump

==============================

功　能: 执行非局部转移
用　法: void longjump(jmp_buf env, int val);
程序例:

```
#include <stdio.h>
#include <setjmp.h>
#include <stdlib.h>

void subroutine(jmp_buf);
```

```
int main(void)
{

   int value;
   jmp_buf jumper;

   value = setjmp(jumper);
   if (value != 0)
   {
      printf("Longjmp with value %d\n", value);
      exit(value);
   }
   printf("About to call subroutine ... \n");
   subroutine(jumper);

   return 0;
}

void subroutine(jmp_buf jumper)
{
   longjmp(jumper,1);
}
```

============================

函数名: lowvideo

============================

功　能: 选择低亮度字符
用　法: void lowvideo(void);
程序例:

```
#include <conio.h>

int main(void)
{
   clrscr();

   highvideo();
   cprintf("High Intesity Text\r\n");
   lowvideo();
   gotoxy(1,2);
   cprintf("Low Intensity Text\r\n");
```

```
   return 0;
}
```

==========================

函数名: lrotl, _lrotl

==========================

功　能: 将无符号长整型数向左循环移位
用　法: unsigned long lrotl(unsigned long lvalue, int count);
 unsigned long _lrotl(unsigned long lvalue, int count);
程序例:

```
/* lrotl example */
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
   unsigned long result;
   unsigned long value = 100;

   result = _lrotl(value,1);
   printf("The value %lu rotated left one bit is: %lu\n", value, result);

   return 0;
}
```

==============================

函数名: lsearch

==============================

功　能: 线性搜索
用　法: void *lsearch(const void *key, void *base, size_t *nelem,
     size_t width, int (*fcmp)(const void *, const void *));
程序例:

```
#include <stdio.h>
#include <stdlib.h>

int compare(int *x, int *y)
```

```
{
   return( *x - *y );
}

int main(void)
{
   int array[5] = {35, 87, 46, 99, 12};
   size_t nelem = 5;
   int key;
   int *result;

   key = 99;
   result = lfind(&key, array, &nelem,
               sizeof(int), (int(*)(const void *,const void *))compare);
   if (result)
      printf("Number %d found\n",key);
   else
      printf("Number %d not found\n",key);

   return 0;
}
```

============================

函数名: lseek

============================
功　能: 移动文件读/写指针
用　法: long lseek(int handle, long offset, int fromwhere);
程序例:

```
#include <sys\stat.h>
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
   int handle;
   char msg[] = "This is a test";
   char ch;
```

```
    /* create a file */
    handle = open("TEST.$$$", O_CREAT | O_RDWR, S_IREAD | S_IWRITE);

    /* write some data to the file */
    write(handle, msg, strlen(msg));

    /* seek to the begining of the file */
    lseek(handle, 0L, SEEK_SET);

    /* reads chars from the file until we hit EOF */
    do
    {
        read(handle, &ch, 1);
        printf("%c", ch);
    }  while (!eof(handle));

    close(handle);
    return 0;
}
```

===========================
main()主函数
===========================

　　每一 C 程序都必须有一 main()函数，可以根据自己的爱好把它放在程序的某个地方。有些程序员把它放在最前面，而另一些程序员把它放在最后面，无论放在哪个地方，以下几点说明都是适合的。

　　1. main() 参数

　　在 Turbo C2.0 启动过程中，传递 main()函数三个参数：argc，argv 和 env。

　　* argc：　整数，为传给 main()的命令行参数个数。

　　* argv：　字符串数组。

　　　　　　　在 DOS 3.X 版本中，argv[0] 为程序运行的全路径名；对 DOS 3.0 以下的版本，argv[0]为空串("") 。

　　　　　　　argv[1] 为在 DOS 命令行中执行程序名后的第一个字符串；

　　　　　　　argv[2] 为执行程序名后的第二个字符串；

　　　　　　　...

　　　　　　　argv[argc]为 NULL。

　　*env:　　安符串数组。env[] 的每一个元素都包含 ENVVAR=value 形式的字符串。其中 ENVVAR 为环境变量如 PATH 或 87。value 为 ENVVAR 的对应值如 C:\DOS，C:\TURBOC(对于 PATH) 或 YES(对于 87)。

　　Turbo C2.0 启动时总是把这三个参数传递给 main()函数，可以在用户程序中说明(或不说明)它们，如果说明了部分(或全部)参数，它们就成为 main()子程序

的局部变量。

请注意：一旦想说明这些参数，则必须按 argc，argv，env 的顺序，如以下的例子：

```
main()
main(int argc)
main(int argc, char *argv[])
main(int argc, char *argv[], char *env[])
```

其中第二种情况是合法的，但不常见，因为在程序中很少有只用 argc，而不用 argv[]的情况。

以下提供一样例程序 EXAMPLE.EXE，演示如何在 main()函数中使用三个参数：

```c
/*program name EXAMPLE.EXE*/
#include <stdio.h>
#include <stdlib.h>
main(int argc, char *argv[], char *env[])
{
    int i;
    printf("These are the %d  command- line  arguments passed  to
            main:\n\n", argc);
    for(i=0; i<=argc; i++)
      printf("argv[%d]:%s\n", i, argv[i]);
    printf("\nThe environment string(s)on this system are:\n\n");
    for(i=0; env[i]!=NULL; i++)
        printf(" env[%d]:%s\n", i, env[i]);
}
```

如果在 DOS 提示符下，按以下方式运行 EXAMPLE.EXE：

C:\example first_argument "argument with blanks" 3 4 "last but one" stop!

注意：可以用双引号括起内含空格的参数，如本例中的：" argument with blanks"和"Last but one")。

结果是这样的：

```
The value of argc is 7
These are the 7 command-linearguments passed to main:
argv[0]:C:\TURBO\EXAMPLE.EXE
argv[1]:first_argument
argv[2]:argument with blanks
argv[3]:3
argv[4]:4
argv[5]:last but one
argv[6]:stop!
argv[7]:(NULL)
The environment string(s) on this system are:
env[0]: COMSPEC=C:\COMMAND.COM
env[1]: PROMPT=$P$G                /*视具体设置而定*/
env[2]: PATH=C:\DOS;C:\TC          /*视具体设置而定*/
```

应该提醒的是：传送 main() 函数的命令行参数的最大长度为 128 个字符（包括参数间的空格）， 这是由 DOS 限制的。

============================

函数名: matherr

============================
功　能：用户可修改的数学错误处理程序
用　法：int matherr(struct exception *e);
程序例:

```
/* This is a user-defined matherr function that prevents
   any error messages from being printed. */

#include<math.h>

int matherr(struct exception *a)
{
    return 1;
}
```

============================

函数名: memccpy

============================
功　能：从源 source 中拷贝 n 个字节到目标 destin 中
用　法：void *memccpy(void *destin, void *source, unsigned char ch,
        unsigned n);
程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char *src = "This is the source string";
    char dest[50];
    char *ptr;

    ptr = memccpy(dest, src, 'c', strlen(src));
```

```
    if (ptr)
    {
       *ptr = '\0';
       printf("The character was found:  %s\n", dest);
    }
    else
       printf("The character wasn't found\n");
    return 0;
}
```

===========================

函数名: malloc

===========================

功　能: 内存分配函数
用　法: void *malloc(unsigned size);
程序例:

```
#include <stdio.h>
#include <string.h>
#include <alloc.h>
#include <process.h>

int main(void)
{
    char *str;

    /* allocate memory for string */
    /* This will generate an error when compiling */
    /* with C++, use the new operator instead. */
    if ((str = malloc(10)) == NULL)
    {
       printf("Not enough memory to allocate buffer\n");
       exit(1);  /* terminate program if out of memory */
    }

    /* copy "Hello" into string */
    strcpy(str, "Hello");

    /* display string */
    printf("String is %s\n", str);

    /* free memory */
```

```
   free(str);

   return 0;
}
```

============================

函数名: memchr

============================

功　能: 在数组的前 n 个字节中搜索字符
用　法: void *memchr(void *s, char ch, unsigned n);
程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
   char str[17];
   char *ptr;

   strcpy(str, "This is a string");
   ptr = memchr(str, 'r', strlen(str));
   if (ptr)
      printf("The character 'r' is at position: %d\n", ptr - str);
   else
      printf("The character was not found\n");
   return 0;
}
```

============================

函数名: memcpy

============================

功　能: 从源 source 中拷贝 n 个字节到目标 destin 中
用　法: void *memcpy(void *destin, void *source, unsigned n);
程序例:

```
#include <stdio.h>
#include <string.h>
int main(void)
{
```

```
   char src[] = "**************************";
   char dest[] = "abcdefghijlkmnopqrstuvwxyz0123456709";
   char *ptr;
   printf("destination before memcpy: %s\n", dest);
   ptr = memcpy(dest, src, strlen(src));
   if (ptr)
      printf("destination after memcpy:  %s\n", dest);
   else
      printf("memcpy failed\n");
   return 0;
}
```

============================

函数名: memicmp

============================

功　能: 比较两个串 s1 和 s2 的前 n 个字节，忽略大小写
用　法: int memicmp(void *s1, void *s2, unsigned n);
程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
   char *buf1 = "ABCDE123";
   char *buf2 = "abcde456";
   int stat;
   stat = memicmp(buf1, buf2, 5);
   printf("The strings to position 5 are ");
   if (stat)
      printf("not ");
   printf("the same\n");
   return 0;
}
```

============================

函数名: memmove

============================

功　能: 移动一块字节
用　法: void *memmove(void *destin, void *source, unsigned n);

程序例：

```c
#include <string.h>
#include <stdio.h>

int main(void)
{
  char *dest = "abcdefghijklmnopqrstuvwxyz0123456789";
  char *src = "**************************";
  printf("destination prior to memmove: %s\n", dest);
  memmove(dest, src, 26);
  printf("destination after memmove:    %s\n", dest);
  return 0;
}
```

============================

函数名: memset

============================

功　能：设置 s 中的所有字节为 ch，s 数组的大小由 n 给定
用　法：void *memset(void *s, char ch, unsigned n);
程序例：

```c
#include <string.h>
#include <stdio.h>
#include <mem.h>

int main(void)
{
   char buffer[] = "Hello world\n";

   printf("Buffer before memset: %s\n", buffer);
   memset(buffer, '*', strlen(buffer) - 1);
   printf("Buffer after memset:  %s\n", buffer);
   return 0;
}
```

============================

函数名: mkdir

============================

功　能: 建立一个目录
用　法: int mkdir(char *pathname);
程序例:

```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <dir.h>

int main(void)
{
  int status;

  clrscr();
  status = mkdir("asdfjklm");
  (!status) ? (printf("Directory created\n")) :
              (printf("Unable to create directory\n"));

  getch();
  system("dir");
  getch();

  status = rmdir("asdfjklm");
  (!status) ? (printf("Directory deleted\n")) :
  (perror("Unable to delete directory"));

  return 0;
}
```


============================

函数名: mktemp

============================
功　能: 建立唯一的文件名
用　法: char *mktemp(char *template);
程序例:

```
#include <dir.h>
#include <stdio.h>

int main(void)
{
```

```
  /* fname defines the template for the
     temporary file.  */

  char *fname = "TXXXXXX", *ptr;

  ptr = mktemp(fname);
  printf("%s\n",ptr);
  return 0;
}
```

============================

函数名: MK_FP

============================

功　能: 设置一个远指针
用　法: void far *MK_FP(unsigned seg, unsigned off);
程序例:

```
#include <dos.h>
#include <graphics.h>

int main(void)
{
   int gd, gm, i;
   unsigned int far *screen;

   detectgraph(&gd, &gm);
   if (gd == HERCMONO)
       screen = MK_FP(0xB000, 0);
   else
       screen = MK_FP(0xB800, 0);
   for (i=0; i<26; i++)
      screen[i] = 0x0700 + ('a' + i);
   return 0;
}
```

============================

函数名: modf

============================

功　能: 把数分为指数和尾数
用　法: double modf(double value, double *iptr);

程序例:

```
#include <math.h>
#include <stdio.h>

int main(void)
{
    double fraction, integer;
    double number = 100000.567;

    fraction = modf(number, &integer);
    printf("The whole and fractional parts of %lf are %lf and %lf\n",
           number, integer, fraction);
    return 0;
}
```

============================

函数名: movedata

============================

功　能: 拷贝字节
用　法: void movedata(int segsrc, int offsrc, int segdest,
　int offdest, unsigned numbytes);
程序例:

```
#include <mem.h>

#define MONO_BASE 0xB000

/* saves the contents of the monochrome screen in buffer */
void save_mono_screen(char near *buffer)
{
    movedata(MONO_BASE, 0, _DS, (unsigned)buffer, 80*25*2);
}

int main(void)
{
    char buf[80*25*2];
    save_mono_screen(buf);
}
```

============================

函数名: moverel

=============================

功　能: 将当前位置(CP)移动一相对距离
用　法: void far moverel(int dx, int dy);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   char msg[80];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   /* move the C.P. to location (20, 30) */
   moveto(20, 30);

   /* plot a pixel at the C.P. */
   putpixel(getx(), gety(), getmaxcolor());

   /* create and output a message at (20, 30) */
   sprintf(msg, " (%d, %d)", getx(), gety());
   outtextxy(20, 30, msg);

   /* move to a point a relative distance */
   /* away from the current value of C.P. */
   moverel(100, 100);
```

```c
    /* plot a pixel at the C.P. */
    putpixel(getx(), gety(), getmaxcolor());

    /* create and output a message at C.P. */
    sprintf(msg, " (%d, %d)", getx(), gety());
    outtext(msg);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

==============================

函数名: movetext

==============================

功　能: 将屏幕文本从一个矩形区域拷贝到另一个矩形区域
用　法: int movetext(int left, int top, int right, int bottom,
　　int newleft, int newtop);
程序例:

```c
#include <conio.h>
#include <string.h>

int main(void)
{
    char *str = "This is a test string";

    clrscr();
    cputs(str);
    getch();

    movetext(1, 1, strlen(str), 2, 10, 10);
    getch();

    return 0;
}
```

==============================

函数名: moveto

==============================

功　能: 将 CP 移到(x, y)
用　法: void far moveto(int x, int y);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   char msg[80];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   /* move the C.P. to location (20, 30) */
   moveto(20, 30);

   /* plot a pixel at the C.P. */
   putpixel(getx(), gety(), getmaxcolor());

   /* create and output a message at (20, 30) */
   sprintf(msg, " (%d, %d)", getx(), gety());
   outtextxy(20, 30, msg);

   /* move to (100, 100) */
   moveto(100, 100);

   /* plot a pixel at the C.P. */
   putpixel(getx(), gety(), getmaxcolor());
```

```
    /* create and output a message at C.P. */
    sprintf(msg, " (%d, %d)", getx(), gety());
    outtext(msg);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: movemem

============================

功　能: 移动一块字节
用　法: void movemem(void *source, void *destin, unsigned len);
程序例:

```
#include <mem.h>
#include <alloc.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
    char *source = "Borland International";
    char *destination;
    int length;

    length = strlen(source);
    destination = malloc(length + 1);
    movmem(source, destination, length);
    printf("%s\n", destination);

    return 0;
}
```

============================

函数名: normvideo

============================

功　能: 选择正常亮度字符

用　法: void normvideo(void);
程序例:

```
#include <conio.h>

int main(void)
{
    normvideo();
    cprintf("NORMAL Intensity Text\r\n");
    return 0;
}
```

==============================

函数名: nosound

==============================

功　能: 关闭 PC 扬声器
用　法: void nosound(void);
程序例:

```
/* Emits a 7-Hz tone for 10 seconds.

    True story: 7 Hz is the resonant frequency of a chicken's skull cavity.
    This was determined empirically in Australia, where a new factory
    generating 7-Hz tones was located too close to a chicken ranch:
    When the factory started up, all the chickens died.

    Your PC may not be able to emit a 7-Hz tone.
*/

int main(void)
{
    sound(7);
    delay(10000);
    nosound();
}
```

==============================

函数名: open

==============================

功　能: 打开一个文件用于读或写
用　法: int open(char *pathname, int access[, int permiss]);
程序例:

```c
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
    int handle;
    char msg[] = "Hello world";

    if ((handle = open("TEST.$$$", O_CREAT | O_TEXT)) == -1)
    {
        perror("Error:");
        return 1;
    }
    write(handle, msg, strlen(msg));
    close(handle);
    return 0;
}
```

===========================

函数名: outport

===========================

功　能: 输出整数到硬件端口中
用　法: void outport(int port, int value);
程序例:

```c
#include <stdio.h>
#include <dos.h>

int main(void)
{
    int value = 64;
    int port = 0;

    outportb(port, value);
    printf("Value %d sent to port number %d\n", value, port);
    return 0;
```

```
}
```

==========================

函数名: **outportb**

==========================

功　能: 输出字节到硬件端口中
用　法: void outportb(int port, char byte);
程序例:

```c
#include <stdio.h>
#include <dos.h>

int main(void)
{
    int value = 64;
    int port = 0;

    outportb(port, value);
    printf("Value %d sent to port number %d\n", value, port);
    return 0;
}
```

==========================

函数名: **outtext**

==========================

功　能: 在视区显示一个字符串
用　法: void far outtext(char far *textstring);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;
```

```c
    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* move the C.P. to the center of the screen */
    moveto(midx, midy);

    /* output text starting at the C.P. */
    outtext("This ");
    outtext("is ");
    outtext("a ");
    outtext("test.");

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

==============================

函数名: outtextxy

==============================

功　能: 在指定位置显示一字符串
用　法: void far outtextxy(int x, int y, char *textstring);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
```

```
int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int midx, midy;

   /* initialize graphics and local variables */
   initgraph( &gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   midx = getmaxx() / 2;
   midy = getmaxy() / 2;

   /* output text at the center of the screen*/
   /* Note: the C.P. doesn't get changed.*/
   outtextxy(midx, midy, "This is a test.");

   /* clean up */
   getch();
   closegraph();
   return 0;
}
```

============================

函数名: parsfnm

============================

功　能: 分析文件名
用　法: char *parsfnm (char *cmdline, struct fcb *fcbptr, int option);
程序例:

```
#include <process.h>
```

```c
#include <string.h>
#include <stdio.h>
#include <dos.h>

int main(void)
{
   char line[80];
   struct fcb blk;

   /* get file name */
   printf("Enter drive and file name (no path - ie. a:file.dat)\n");
   gets(line);

   /* put file name in fcb */
   if (parsfnm(line, &blk, 1) == NULL)
      printf("Error in parsfm call\n");
   else
      printf("Drive #%d  Name: %11s\n", blk.fcb_drive, blk.fcb_name);

   return 0;
}
```

==========================

函数名: peek

==========================

功　能: 检查存储单元
用　法: int peek(int segment, unsigned offset);
程序例:

```c
#include <stdio.h>
#include <conio.h>
#include <dos.h>

int main(void)
{
   int value = 0;

   printf("The current status of your keyboard is:\n");
   value = peek(0x0040, 0x0017);
   if (value & 1)
      printf("Right shift on\n");
   else
```

```
   printf("Right shift off\n");

   if (value & 2)
      printf("Left shift on\n");
   else
      printf("Left shift off\n");

   if (value & 4)
      printf("Control key on\n");
   else
      printf("Control key off\n");

   if (value & 8)
      printf("Alt key on\n");
   else
      printf("Alt key off\n");

   if (value & 16)
      printf("Scroll lock on\n");
   else
      printf("Scroll lock off\n");

   if (value & 32)
      printf("Num lock on\n");
   else
      printf("Num lock off\n");

   if (value & 64)
      printf("Caps lock on\n");
   else
      printf("Caps lock off\n");

   return 0;
}
```

===========================

函数名: peekb

===========================

功　能: 检查存储单元
用　法: char peekb (int segment, unsigned offset);
程序例:

```c
#include <stdio.h>
#include <conio.h>
#include <dos.h>

int main(void)
{
    int value = 0;

    printf("The current status of your keyboard is:\n");
    value = peekb(0x0040, 0x0017);
    if (value & 1)
        printf("Right shift on\n");
    else
        printf("Right shift off\n");

    if (value & 2)
        printf("Left shift on\n");
    else
        printf("Left shift off\n");

    if (value & 4)
        printf("Control key on\n");
    else
        printf("Control key off\n");

    if (value & 8)
        printf("Alt key on\n");
    else
        printf("Alt key off\n");

    if (value & 16)
        printf("Scroll lock on\n");
    else
        printf("Scroll lock off\n");

    if (value & 32)
        printf("Num lock on\n");
    else
        printf("Num lock off\n");

    if (value & 64)
        printf("Caps lock on\n");
    else
        printf("Caps lock off\n");
```

```
   return 0;
}
```

============================

函数名: **perror**

============================

功　能: 系统错误信息
用　法: void perror(char *string);
程序例:

```
#include <stdio.h>

int main(void)
{
   FILE *fp;

   fp = fopen("perror.dat", "r");
   if (!fp)
      perror("Unable to open file for reading");
   return 0;
}
```

============================

函数名: **pieslice**

============================

功　能: 绘制并填充一个扇形
用　法: void far pieslice(int x, int stanle, int endangle, int radius);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int midx, midy;
```

```
    int stangle = 45, endangle = 135, radius = 100;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* set fill style and draw a pie slice */
    setfillstyle(EMPTY_FILL, getmaxcolor());
    pieslice(midx, midy, stangle, endangle, radius);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

==========================

函数名: poke

==========================
功　能: 存值到一个给定存储单元
用　法: void poke(int segment, int offset, int value);
程序例:

```
#include <dos.h>
#include <conio.h>

int main(void)
{
    clrscr();
    cprintf("Make sure the scroll lock key is off and press any key\r\n");
```

```
    getch();
    poke(0x0000, 0x0417, 16);
    cprintf("The scroll lock is now on\r\n");
    return 0;
}
```

============================

函数名: pokeb

============================

功 能: 存值到一个给定存储单元
用 法: void pokeb(int segment, int offset, char value);
程序例:

```
#include <dos.h>
#include <conio.h>

int main(void)
{
    clrscr();
    cprintf("Make sure the scroll lock key is off and press any key\r\n");
    getch();
    pokeb(0x0000, 0x0417, 16);
    cprintf("The scroll lock is now on\r\n");
    return 0;
}
```

============================

函数名: poly

============================

功 能: 根据参数产生一个多项式
用 法: double poly(double x, int n, double c[]);
程序例:

```
#include <stdio.h>
#include <math.h>

/* polynomial:  x**3 - 2x**2 + 5x - 1 */

int main(void)
{
```

```c
    double array[] = { -1.0, 5.0, -2.0, 1.0 };
    double result;

    result = poly(2.0, 3, array);
    printf("The polynomial: x**3 - 2.0x**2 + 5x - 1 at 2.0 is %lf\n",
            result);
    return 0;
}
```

============================

函数名: pow

============================

功　能: 指数函数(x 的 y 次方)
用　法: double pow(double x, double y);
程序例:

```c
#include <math.h>
#include <stdio.h>

int main(void)
{
    double x = 2.0, y = 3.0;

    printf("%lf raised to %lf is %lf\n", x, y, pow(x, y));
    return 0;
}
```

============================

函数名: pow10

============================

功　能: 指数函数(10 的 p 次方)
用　法: double pow10(int p);
程序例:

```c
#include <math.h>
#include <stdio.h>

int main(void)
{
    double p = 3.0;
```

```
    printf("Ten raised to %lf is %lf\n", p, pow10(p));
    return 0;
}
```

============================

函数名: printf

============================

功　能: 产生格式化输出的函数
用　法: int printf(char *format...);
程序例:

```
#include <stdio.h>
#include <string.h>

#define I 555
#define R 5.5

int main(void)
{
    int i,j,k,l;
    char buf[7];
    char *prefix = buf;
    char tp[20];
    printf("prefix  6d    6o    8x       10.2e        "
         "10.2f\n");
    strcpy(prefix,"%");
    for (i = 0; i < 2; i++)
    {
      for (j = 0; j < 2; j++)
          for (k = 0; k < 2; k++)
      for (l = 0; l < 2; l++)
              {
                  if (i==0)  strcat(prefix,"-");
                  if (j==0)  strcat(prefix,"+");
                  if (k==0)  strcat(prefix,"#");
                  if (l==0)  strcat(prefix,"0");
                  printf("%5s |",prefix);
                  strcpy(tp,prefix);
                  strcat(tp,"6d |");
                  printf(tp,I);
                  strcpy(tp,"");
                  strcpy(tp,prefix);
```

```
                strcat(tp,"6o |");
                printf(tp, I);
                strcpy(tp,"");
                strcpy(tp, prefix);
                strcat(tp,"8x |");
                printf(tp, I);
                strcpy(tp,"");
                strcpy(tp, prefix);
          strcat(tp,"10.2e |");
          printf(tp, R);
          strcpy(tp, prefix);
          strcat(tp,"10.2f |");
          printf(tp, R);
          printf("  \n");
          strcpy(prefix,"%");
       }
        }
    return 0;
}
```

============================

函数名: putc

============================

功　能: 输出一字符到指定流中
用　法: int putc(int ch, FILE *stream);
程序例:

```
#include <stdio.h>

int main(void)
{
    char msg[] = "Hello world\n";
    int i = 0;

    while (msg[i])
       putc(msg[i++], stdout);
    return 0;
}
```

============================

函数名: putch

============================

功　能: 输出字符到控制台
用　法: int putch(int ch);
程序例:

```c
#include <stdio.h>
#include <conio.h>

int main(void)
{
    char ch = 0;

    printf("Input a string:");
    while ((ch != '\r'))
    {
        ch = getch();
        putch(ch);
    }
    return 0;
}
```


============================

函数名: putchar

============================

功　能: 在 stdout 上输出字符
用　法: int putchar(int ch);
程序例:

```c
#include <stdio.h>

/* define some box-drawing characters */
#define LEFT_TOP   0xDA
#define RIGHT_TOP  0xBF
#define HORIZ      0xC4
#define VERT       0xB3
#define LEFT_BOT   0xC0
#define RIGHT_BOT  0xD9

int main(void)
{
```

```c
    char i, j;

    /* draw the top of the box */
    putchar(LEFT_TOP);
    for (i=0; i<10; i++)
        putchar(HORIZ);
    putchar(RIGHT_TOP);
    putchar('\n');

    /* draw the middle */
    for (i=0; i<4; i++)
    {
        putchar(VERT);
        for (j=0; j<10; j++)
            putchar(' ');
        putchar(VERT);
        putchar('\n');
    }

    /* draw the bottom */
    putchar(LEFT_BOT);
    for (i=0; i<10; i++)
        putchar(HORIZ);
    putchar(RIGHT_BOT);
    putchar('\n');

    return 0;
}
```

============================

函数名: putenv

============================

功　能: 把字符串加到当前环境中
用　法: int putenv(char *envvar);
程序例:

```c
#include <stdio.h>
#include <stdlib.h>
#include <alloc.h>
#include <string.h>
#include <dos.h>
```

```c
int main(void)
{
   char *path, *ptr;
   int i = 0;

   /* get the current path environment */
   ptr = getenv("PATH");

   /* set up new path */
   path = malloc(strlen(ptr)+15);
   strcpy(path,"PATH=");
   strcat(path,ptr);
   strcat(path,";c:\\temp");

   /* replace the current path and display current environment */
   putenv(path);
   while (environ[i])
       printf("%s\n",environ[i++]);

   return 0;
}
```

==========================

函数名: putimage

==========================

功　能: 在屏幕上输出一个位图
用　法: void far putimage(int x, int y, void far *bitmap, int op);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

#define ARROW_SIZE 10

void draw_arrow(int x, int y);

int main(void)
{
   /* request autodetection */
   int gdriver = DETECT, gmode, errorcode;
```

```c
void *arrow;
int x, y, maxx;
unsigned int size;

/* initialize graphics and local variables */
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk)  /* an error occurred */
{
   printf("Graphics error: %s\n", grapherrormsg(errorcode));
   printf("Press any key to halt:");
   getch();
   exit(1); /* terminate with an error code */
}

maxx = getmaxx();
x = 0;
y = getmaxy() / 2;

/* draw the image to be grabbed */
draw_arrow(x, y);

/* calculate the size of the image */
size = imagesize(x, y-ARROW_SIZE, x+(4*ARROW_SIZE), y+ARROW_SIZE);

/* allocate memory to hold the image */
arrow = malloc(size);

/* grab the image */
getimage(x, y-ARROW_SIZE, x+(4*ARROW_SIZE), y+ARROW_SIZE, arrow);

/* repeat until a key is pressed */
while (!kbhit())
{
   /* erase old image */
   putimage(x, y-ARROW_SIZE, arrow, XOR_PUT);

   x += ARROW_SIZE;
   if (x >= maxx)
       x = 0;

   /* plot new image */
```

```
      putimage(x, y-ARROW_SIZE, arrow, XOR_PUT);
   }

   /* clean up */
   free(arrow);
   closegraph();
   return 0;
}

void draw_arrow(int x, int y)
{
   /* draw an arrow on the screen */
   moveto(x, y);
   linerel(4*ARROW_SIZE, 0);
   linerel(-2*ARROW_SIZE, -1*ARROW_SIZE);
   linerel(0, 2*ARROW_SIZE);
   linerel(2*ARROW_SIZE, -1*ARROW_SIZE);
}
```

===========================

函数名: putpixel

===========================

功　能: 在指定位置画一像素
用　法: void far putpixel (int x, int y, int pixelcolor);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <dos.h>

#define PIXEL_COUNT 1000
#define DELAY_TIME  100  /* in milliseconds */

int main(void)
{
   /* request autodetection */
   int gdriver = DETECT, gmode, errorcode;
   int i, x, y, color, maxx, maxy, maxcolor, seed;

   /* initialize graphics and local variables */
```

```c
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    maxx = getmaxx() + 1;
    maxy = getmaxy() + 1;
    maxcolor = getmaxcolor() + 1;

    while (!kbhit())
    {
        /* seed the random number generator */
        seed = random(32767);
        srand(seed);
        for (i=0; i<PIXEL_COUNT; i++)
        {
x = random(maxx);
            y = random(maxy);
            color = random(maxcolor);
            putpixel(x, y, color);
        }

        delay(DELAY_TIME);
        srand(seed);
        for (i=0; i<PIXEL_COUNT; i++)
        {
x = random(maxx);
y = random(maxy);
color = random(maxcolor);
if (color == getpixel(x, y))
    putpixel(x, y, 0);
        }
    }

    /* clean up */
    getch();
    closegraph();
```

```
   return 0;
}
```

============================

函数名: puts

============================

功　能: 送一字符串到流中
用　法: int puts(char *string);
程序例:

```
#include <stdio.h>
int main(void)
{
    char string[] = "This is an example output string\n";

    puts(string);
    return 0;
}
```

============================

函数名: puttext

============================

功　能: 将文本从存储区拷贝到屏幕
用　法: int puttext(int left, int top, int right, int bottom, void *source);
程序例:

```
#include <conio.h>
int main(void)
{
    char buffer[512];

    /* put some text to the console */
    clrscr();
    gotoxy(20, 12);
    cprintf("This is a test.  Press any key to continue ...");
    getch();

    /* grab screen contents */
    gettext(20, 12, 36, 21,buffer);
    clrscr();
```

```
   /* put selected characters back to the screen */
   gotoxy(20, 12);
   puttext(20, 12, 36, 21, buffer);
   getch();

   return 0;
}
```

===========================

函数名: putw

===========================
功　能: 把一字符或字送到流中
用　法: int putw(int w, FILE *stream);
程序例:

```
#include <stdio.h>
#include <stdlib.h>

#define FNAME "test.$$$"

int main(void)
{
   FILE *fp;
   int word;

   /* place the word in a file */
   fp = fopen(FNAME, "wb");
   if (fp == NULL)
   {
      printf("Error opening file %s\n", FNAME);
      exit(1);
   }

   word = 94;
   putw(word, fp);
   if (ferror(fp))
       printf("Error writing to file\n");
   else
       printf("Successful write\n");
   fclose(fp);
```

```c
    /* reopen the file */
    fp = fopen(FNAME, "rb");
    if (fp == NULL)
    {
        printf("Error opening file %s\n", FNAME);
        exit(1);
    }

    /* extract the word */
    word = getw(fp);
    if (ferror(fp))
        printf("Error reading file\n");
    else
        printf("Successful read: word = %d\n", word);

    /* clean up */
    fclose(fp);
    unlink(FNAME);

    return 0;
}
```

==============================

函数名: qsort

==============================

功　能: 使用快速排序例程进行排序
用　法: void qsort(void *base, int nelem, int width, int (*fcmp)());
程序例:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int sort_function( const void *a, const void *b);

char list[5][4] = { "cat", "car", "cab", "cap", "can" };


int main(void)
{
    int   x;
```

```
   qsort((void *)list, 5, sizeof(list[0]), sort_function);
   for (x = 0; x < 5; x++)
      printf("%s\n", list[x]);
   return 0;
}


int sort_function( const void *a, const void *b)
{
   return( strcmp(a,b) );
}
```

==============================

函数名: raise

==============================

功　能: 向正在执行的程序发送一个信号
用　法: int raise(int sig);
程序例:

```
#include <signal.h>

int main(void)
{
   int a, b;

   a = 10;
   b = 0;
   if (b == 0)
   /* preempt divide by zero error */
      raise(SIGFPE);
   a = a / b;
   return 0;
}
```

==============================

函数名: rand

==============================

功　能: 随机数发生器
用　法: void rand(void);

程序例:

```
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    int i;

    printf("Ten random numbers from 0 to 99\n\n");
    for(i=0; i<10; i++)
        printf("%d\n", rand() % 100);
    return 0;
}
```

============================

函数名: randbrd

============================

功　能: 随机块读
用　法: int randbrd(struct fcb *fcbptr, int reccnt);
程序例:

```
#include <process.h>
#include <string.h>
#include <stdio.h>
#include <dos.h>

int main(void)
{
    char far *save_dta;
    char line[80], buffer[256];
    struct fcb blk;
    int i, result;

    /* get user input file name for dta */
    printf("Enter drive and file name (no path - i.e. a:file.dat)\n");
    gets(line);

    /* put file name in fcb */
    if (!parsfnm(line, &blk, 1))
    {
        printf("Error in call to parsfnm\n");
```

```
      exit(1);
   }
   printf("Drive #%d  File: %s\n\n", blk.fcb_drive, blk.fcb_name);

   /* open file with DOS FCB open file */
   bdosptr(0x0F, &blk, 0);

   /* save old dta, and set new one */
   save_dta = getdta();
   setdta(buffer);

   /* set up info for the new dta */
   blk.fcb_recsize = 128;
   blk.fcb_random = 0L;
   result = randbrd(&blk, 1);

   /* check results from randbrd */
   if (!result)
      printf("Read OK\n\n");
   else
   {
      perror("Error during read");
      exit(1);
   }

   /* read in data from the new dta */
   printf("The first 128 characters are:\n");
   for (i=0; i<128; i++)
      putchar(buffer[i]);

   /* restore previous dta */
   setdta(save_dta);

   return 0;
}
```

============================

函数名: randbwr

============================

功 能: 随机块写
用 法: int randbwr(struct fcp *fcbptr, int reccnt);
程序例:

```c
#include <process.h>
#include <string.h>
#include <stdio.h>
#include <dos.h>

int main(void)
{
   char far *save_dta;
   char line[80];
   char buffer[256] = "RANDBWR test!";
   struct fcb blk;
   int result;

   /* get new file name from user */
   printf("Enter a file name to create (no path - ie. a:file.dat\n");
   gets(line);

   /* parse the new file name to the dta */
   parsfnm(line,&blk,1);
   printf("Drive #%d  File: %s\n", blk.fcb_drive, blk.fcb_name);

   /* request DOS services to create file */
   if (bdosptr(0x16, &blk, 0) == -1)
   {
      perror("Error creating file");
      exit(1);
   }

   /* save old dta and set new dta */
   save_dta = getdta();
   setdta(buffer);

   /* write new records */
   blk.fcb_recsize = 256;
   blk.fcb_random = 0L;
   result = randbwr(&blk, 1);

   if (!result)
      printf("Write OK\n");
   else
   {
      perror("Disk error");
      exit(1);
   }
```

```
   /* request DOS services to close the file */
   if (bdosptr(0x10, &blk, 0) == -1)
   {
      perror("Error closing file");
      exit(1);
   }

   /* reset the old dta */
   setdta(save_dta);

   return 0;
}
```

============================

函数名: random

============================

功　能: 随机数发生器
用　法: int random(int num);
程序例:

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

/* prints a random number in the range 0 to 99 */
int main(void)
{
   randomize();
   printf("Random number in the 0-99 range: %d\n", random (100));
   return 0;
}
```

============================

函数名: randomize

============================

功　能: 初始化随机数发生器
用　法: void randomize(void);
程序例:

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

int main(void)
{
    int i;

    randomize();
    printf("Ten random numbers from 0 to 99\n\n");
    for(i=0; i<10; i++)
        printf("%d\n", rand() % 100);
    return 0;
}
```

============================

函数名: read

============================

功　能: 从文件中读
用　法: int read(int handle, void *buf, int nbyte);
程序例:

```
#include <stdio.h>
#include <io.h>
#include <alloc.h>
#include <fcntl.h>
#include <process.h>
#include <sys\stat.h>

int main(void)
{
    void *buf;
    int handle, bytes;

    buf = malloc(10);

/*
    Looks for a file in the current directory named TEST.$$$ and attempts
    to read 10 bytes from it.  To use this example you should create the
    file TEST.$$$
*/
    if ((handle =
```

```
        open("TEST.$$$", O_RDONLY | O_BINARY, S_IWRITE | S_IREAD)) == -1)
    {
        printf("Error Opening File\n");
        exit(1);
    }

    if ((bytes = read(handle, buf, 10)) == -1) {
        printf("Read Failed.\n");
        exit(1);
    }
    else {
        printf("Read: %d bytes read.\n", bytes);
    }
    return 0;
}
```

============================

函数名: realloc

============================

功　能: 重新分配主存
用　法: void *realloc(void *ptr, unsigned newsize);
程序例:

```
#include <stdio.h>
#include <alloc.h>
#include <string.h>

int main(void)
{
    char *str;

    /* allocate memory for string */
    str = malloc(10);

    /* copy "Hello" into string */
    strcpy(str, "Hello");

    printf("String is %s\n  Address is %p\n", str, str);
    str = realloc(str, 20);
    printf("String is %s\n  New address is %p\n", str, str);

    /* free memory */
```

```
    free(str);

    return 0;
}
```

============================

函数名: rectangle

============================

功　能: 画一个矩形
用　法: void far rectangle(int left, int top, int right, int bottom);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int left, top, right, bottom;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    left = getmaxx() / 2 - 50;
    top = getmaxy() / 2 - 50;
    right = getmaxx() / 2 + 50;
    bottom = getmaxy() / 2 + 50;

    /* draw a rectangle */
```

```
        rectangle(left, top, right, bottom);

        /* clean up */
        getch();
        closegraph();
        return 0;
}
```

==========================

函数名: registerbgidriver

==========================

功　能: 登录已连接进来的图形驱动程序代码
用　法: int registerbgidriver(void(*driver)(void));
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;

    /* register a driver that was added into graphics.lib */
    errorcode = registerbgidriver(EGAVGA_driver);

    /* report any registration errors */
    if (errorcode < 0)
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
```

```
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    /* draw a line */
    line(0, 0, getmaxx(), getmaxy());

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: remove

============================

功　能: 删除一个文件
用　法: int remove(char *filename);
程序例:

```
#include <stdio.h>

int main(void)
{
    char file[80];

    /* prompt for file name to delete */
    printf("File to delete: ");
    gets(file);

    /* delete the file */
    if (remove(file) == 0)
        printf("Removed %s.\n",file);
    else
        perror("remove");

    return 0;
}
```

============================

函数名: rename

============================

功　能: 重命名文件
用　法: int rename(char *oldname, char *newname);
程序例:

```c
#include <stdio.h>

int main(void)
{
   char oldname[80], newname[80];

   /* prompt for file to rename and new name */
   printf("File to rename: ");
   gets(oldname);
   printf("New name: ");
   gets(newname);

   /* Rename the file */
   if (rename(oldname, newname) == 0)
      printf("Renamed %s to %s.\n", oldname, newname);
   else
      perror("rename");

   return 0;
}
```

============================

函数名: restorecrtmode

============================

功　能: 将屏幕模式恢复为先前的 imitgraph 设置
用　法: void far restorecrtmode(void);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
```

```c
int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int x, y;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   x = getmaxx() / 2;
   y = getmaxy() / 2;

   /* output a message */
   settextjustify(CENTER_TEXT, CENTER_TEXT);
   outtextxy(x, y, "Press any key to exit graphics:");
   getch();

   /* restore system to text mode */
   restorecrtmode();
   printf("We're now in text mode.\n");
   printf("Press any key to return to graphics mode:");
   getch();

   /* return to graphics mode */
   setgraphmode(getgraphmode());

   /* output a message */
   settextjustify(CENTER_TEXT, CENTER_TEXT);
   outtextxy(x, y, "We're back in graphics mode.");
   outtextxy(x, y+textheight("W"), "Press any key to halt:");

   /* clean up */
   getch();
```

```
    closegraph();
    return 0;
}
```

==============================

函数名: rewind

==============================

功　能: 将文件指针重新指向一个流的开头
用　法: int rewind(FILE *stream);
程序例:

```
#include <stdio.h>
#include <dir.h>

 int main(void)
 {
    FILE *fp;
    char *fname = "TXXXXXX", *newname, first;

    newname = mktemp(fname);
    fp = fopen(newname,"w+");
    fprintf(fp,"abcdefghijklmnopqrstuvwxyz");
    rewind(fp);
    fscanf(fp,"%c",&first);
    printf("The first character is: %c\n",first);
    fclose(fp);
    remove(newname);

    return 0;
}
```

==============================

函数名: rmdir

==============================

功　能: 删除 DOS 文件目录
用　法: int rmdir(char *stream);
程序例:

```
#include <stdio.h>
#include <conio.h>
```

```
#include <process.h>
#include <dir.h>

#define DIRNAME "testdir.$$$"

int main(void)
{
    int stat;

    stat = mkdir(DIRNAME);
    if (!stat)
            printf("Directory created\n");
    else
    {
        printf("Unable to create directory\n");
        exit(1);
    }

    getch();
    system("dir/p");
    getch();

    stat = rmdir(DIRNAME);
    if (!stat)
            printf("\nDirectory deleted\n");
    else
    {
    perror("\nUnable to delete directory\n");
        exit(1);
    }

    return 0;
}
```

============================

函数名: sbrk

============================

功　能: 改变数据段空间位置
用　法: char *sbrk(int incr);
程序例:

```
#include <stdio.h>
#include <alloc.h>

int main(void)
{
   printf("Changing allocation with sbrk()\n");
   printf("Before sbrk() call: %lu bytes free\n",
   (unsigned long) coreleft());
   sbrk(1000);
   printf(" After sbrk() call: %lu bytes free\n",
   (unsigned long) coreleft());
   return 0;
}
```

============================

函数名: scanf

============================

功　能: 执行格式化输入
用　法: int scanf(char *format[,argument,...]);
程序例:

```
#include <stdio.h>
#include <conio.h>

int main(void)
{
   char label[20];
   char name[20];
   int entries = 0;
   int loop, age;
   double salary;

   struct Entry_struct
   {
      char  name[20];
      int   age;
      float salary;
   } entry[20];

/* Input a label as a string of characters restricting to 20 characters */
   printf("\n\nPlease enter a label for the chart: ");
   scanf("%20s", label);
```

```c
    fflush(stdin);  /* flush the input stream in case of bad input */

/* Input number of entries as an integer */
    printf("How many entries will there be? (less than 20) ");
    scanf("%d", &entries);
    fflush(stdin);   /* flush the input stream in case of bad input */

/* input a name restricting input to only letters upper or lower case */
    for (loop=0;loop<entries;++loop)
    {
        printf("Entry %d\n", loop);
        printf("  Name   : ");
        scanf("%[A-Za-z]", entry[loop].name);
        fflush(stdin);  /* flush the input stream in case of bad input */

/* input an age as an integer */
        printf("  Age    : ");
        scanf("%d", &entry[loop].age);
        fflush(stdin);   /* flush the input stream in case of bad input */

/* input a salary as a float */
        printf("  Salary : ");
        scanf("%f", &entry[loop].salary);
        fflush(stdin); /* flush the input stream in case of bad input */
    }

/* Input a name, age and salary as a string, integer, and double */
    printf("\nPlease enter your name, age and salary\n");
    scanf("%20s %d %lf", name, &age, &salary);


/* Print out the data that was input */
    printf("\n\nTable %s\n",label);
    printf("Compiled by %s  age %d  $%15.2lf\n", name, age, salary);
    printf("------------------------------------------------------\n");
    for (loop=0;loop<entries;++loop)
        printf("%4d | %-20s | %5d | %15.2lf\n",
            loop + 1,
   entry[loop].name,
   entry[loop].age,
            entry[loop].salary);
    printf("------------------------------------------------------\n");
    return 0;
}
```

============================

函数名: searchpath

============================

功　能: 搜索 DOS 路径
用　法: char *searchpath(char *filename);
程序例:

```c
#include <stdio.h>
#include <dir.h>

int main(void)
{
    char *p;

    /* Looks for TLINK and returns a pointer
       to the path  */
    p = searchpath("TLINK.EXE");
    printf("Search for TLINK.EXE : %s\n", p);

    /* Looks for non-existent file  */
    p = searchpath("NOTEXIST.FIL");
    printf("Search for NOTEXIST.FIL : %s\n", p);

    return 0;
}
```

============================

函数名: sector

============================

功　能: 画并填充椭圆扇区
用　法: void far sector(int x, int y, int stangle, int endangle);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
```

```c
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int midx, midy, i;
   int stangle = 45, endangle = 135;
   int xrad = 100, yrad = 50;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   midx = getmaxx() / 2;
   midy = getmaxy() / 2;

   /* loop through the fill patterns */
   for (i=EMPTY_FILL; i<USER_FILL; i++)
   {
      /* set the fill style */
      setfillstyle(i, getmaxcolor());

      /* draw the sector slice */
      sector(midx, midy, stangle, endangle, xrad, yrad);

      getch();
   }

   /* clean up */
   closegraph();
   return 0;
}
```

============================

函数名: **segread**

============================

功　能: 读段寄存器值
用　法: void segread(struct SREGS *segtbl);
程序例:

```c
#include <stdio.h>
#include <dos.h>

int main(void)
{
   struct SREGS segs;

   segread(&segs);
   printf("Current segment register settings\n\n");
   printf("CS: %X   DS: %X\n", segs.cs, segs.ds);
   printf("ES: %X   SS: %X\n", segs.es, segs.ss);

   return 0;
}
```

===========================

函数名: setactivepage

===========================

功　能: 设置图形输出活动页
用　法: void far setactivepage(int pagenum);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* select a driver and mode that supports */
   /* multiple pages.                         */
   int gdriver = EGA, gmode = EGAHI, errorcode;
   int x, y, ht;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
```

```c
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    x = getmaxx() / 2;
    y = getmaxy() / 2;
    ht = textheight("W");

    /*  select the off screen page for drawing */
    setactivepage(1);

    /* draw a line on page #1 */
    line(0, 0, getmaxx(), getmaxy());

    /* output a message on page #1 */
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(x, y, "This is page #1:");
    outtextxy(x, y+ht, "Press any key to halt:");

    /* select drawing to page #0 */
    setactivepage(0);

    /* output a message  on page #0 */
    outtextxy(x, y, "This is page #0.");
    outtextxy(x, y+ht, "Press any key to view page #1:");
    getch();

    /* select page #1 as the visible page */
    setvisualpage(1);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: setallpallette

=============================

功　能: 按指定方式改变所有的调色板颜色
用　法: void far setallpallette(struct palette, far *pallette);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   struct palettetype pal;
   int color, maxcolor, ht;
   int y = 10;
   char msg[80];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   maxcolor = getmaxcolor();
   ht = 2 * textheight("W");

   /* grab a copy of the palette */
   getpalette(&pal);

   /* display the default palette colors */
   for (color=1; color<=maxcolor; color++)
   {
      setcolor(color);
      sprintf(msg, "Color: %d", color);
```

```c
            outtextxy(1, y, msg);
            y += ht;
    }

    /* wait for a key */
    getch();

    /* black out the colors one by one */
    for (color=1; color<=maxcolor; color++)
    {
        setpalette(color, BLACK);
        getch();
    }

    /* restore the palette colors */
    setallpalette(&pal);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: setaspectratio

============================

功　能: 设置图形纵横比
用　法: void far setaspectratio(int xasp, int yasp);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int xasp, yasp, midx, midy;

    /* initialize graphics and local variables */
```

```c
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;
    setcolor(getmaxcolor());

    /* get current aspect ratio settings */
    getaspectratio(&xasp, &yasp);

    /* draw normal circle */
    circle(midx, midy, 100);
    getch();

    /* claer the screen */
    cleardevice();

    /* adjust the aspect for a wide circle */
    setaspectratio(xasp/2, yasp);
    circle(midx, midy, 100);
    getch();

    /* adjust the aspect for a narrow circle */
    cleardevice();
    setaspectratio(xasp, yasp/2);
    circle(midx, midy, 100);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

===========================

函数名: setbkcolor

============================

功　能: 用调色板设置当前背景颜色
用　法: void far setbkcolor(int color);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* select a driver and mode that supports */
   /* multiple background colors.            */
   int gdriver = EGA, gmode = EGAHI, errorcode;
   int bkcol, maxcolor, x, y;
   char msg[80];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   /* maximum color index supported */
   maxcolor = getmaxcolor();

   /* for centering text messages */
   settextjustify(CENTER_TEXT, CENTER_TEXT);
   x = getmaxx() / 2;
   y = getmaxy() / 2;

   /* loop through the available colors */
   for (bkcol=0; bkcol<=maxcolor; bkcol++)
   {
      /* clear the screen */
```

```
        cleardevice();

        /* select a new background color */
        setbkcolor(bkcol);

        /* output a messsage */
        if (bkcol == WHITE)
    setcolor(EGA_BLUE);
        sprintf(msg, "Background color: %d", bkcol);
        outtextxy(x, y, msg);
        getch();
    }

    /* clean up */
    closegraph();
    return 0;
}
```

============================

函数名: setblock

============================

功　能: 修改先前已分配的 DOS 存储段大小
用　法: int setblock(int seg, int newsize);
程序例:

```
#include <dos.h>
#include <alloc.h>
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    unsigned int size, segp;
    int stat;

    size = 64; /* (64 x 16) = 1024 bytes */
    stat = allocmem(size, &segp);
    if (stat == -1)
        printf("Allocated memory at segment: %X\n", segp);
    else
    {
        printf("Failed: maximum number of paragraphs available is %d\n",
```

```
        stat);
        exit(1);
    }

    stat = setblock(segp, size * 2);
    if (stat == -1)
        printf("Expanded memory block at segment: %X\n", segp);
    else
        printf("Failed: maximum number of paragraphs available is %d\n",
                stat);

    freemem(segp);

    return 0;
}
```

==============================

函数名: setbuf

==============================

功 能: 把缓冲区与流相联
用 法: void setbuf(FILE *steam, char *buf);
程序例:

```
#include <stdio.h>

/* BUFSIZ is defined in stdio.h */
char outbuf[BUFSIZ];

int main(void)
{
    /* attach a buffer to the standard output stream */
    setbuf(stdout, outbuf);

    /* put some characters into the buffer */
    puts("This is a test of buffered output.\n\n");
    puts("This output will go into outbuf\n");
    puts("and won't appear until the buffer\n");
    puts("fills up or we flush the stream.\n");

    /* flush the output buffer */
    fflush(stdout);
```

```
    return 0;
}
```

============================

函数名: setcbrk

============================

功　能: 设置 Control-break
用　法: int setcbrk(int value);
程序例:

```
#include <dos.h>
#include <conio.h>
#include <stdio.h>

int main(void)
{
    int break_flag;

    printf("Enter 0 to turn control break off\n");
    printf("Enter 1 to turn control break on\n");

    break_flag = getch() - 0;

    setcbrk(break_flag);

    if (getcbrk())
        printf("Cntrl-brk flag is on\n");
    else
        printf("Cntrl-brk flag is off\n");
    return 0;
}
```

============================

函数名: setcolor

============================

功　能: 设置当前画线颜色
用　法: void far setcolor(int color);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* select a driver and mode that supports */
   /* multiple drawing colors.                */
   int gdriver = EGA, gmode = EGAHI, errorcode;
   int color, maxcolor, x, y;
   char msg[80];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   /* maximum color index supported */
   maxcolor = getmaxcolor();

   /* for centering text messages */
   settextjustify(CENTER_TEXT, CENTER_TEXT);
   x = getmaxx() / 2;
   y = getmaxy() / 2;

   /* loop through the available colors */
   for (color=1; color<=maxcolor; color++)
   {
      /* clear the screen */
      cleardevice();

      /* select a new background color */
      setcolor(color);

      /* output a messsage */
```

```
        sprintf(msg, "Color: %d", color);
        outtextxy(x, y, msg);
        getch();
    }

    /* clean up */
    closegraph();
    return 0;
}
```

============================

函数名: **setdate**

============================

功　能: 设置 DOS 日期
用　法: void setdate(struct date *dateblk);
程序例:

```
#include <stdio.h>
#include <process.h>
#include <dos.h>

int main(void)
{
    struct date reset;
    struct date save_date;

    getdate(&save_date);
    printf("Original date:\n");
    system("date");

    reset.da_year = 2001;
    reset.da_day = 1;
    reset.da_mon = 1;
    setdate(&reset);

    printf("Date after setting:\n");
    system("date");

    setdate(&save_date);
    printf("Back to original date:\n");
    system("date");
```

```
   return 0;
}
```

============================

函数名: setdisk

============================

功　能: 设置当前磁盘驱动器
用　法: int setdisk(int drive);
程序例:

```c
#include <stdio.h>
#include <dir.h>

int main(void)
{
   int save, disk, disks;

   /* save original drive */
   save = getdisk();

   /* print number of logic drives */
   disks = setdisk(save);
   printf("%d logical drives on the system\n\n", disks);

   /* print the drive letters available */
   printf("Available drives:\n");
   for (disk = 0;disk < 26;++disk)
   {
      setdisk(disk);
      if (disk == getdisk())
         printf("%c: drive is available\n", disk + 'a');
   }
   setdisk(save);

   return 0;
}
```

============================

函数名: setdta

============================

功　能：设置磁盘传输区地址
用　法：void setdta(char far *dta);
程序例：

```c
#include <process.h>
#include <string.h>
#include <stdio.h>
#include <dos.h>

int main(void)
{
    char line[80], far *save_dta;
    char buffer[256] = "SETDTA test!";
    struct fcb blk;
    int result;

    /* get new file name from user */
    printf("Enter a file name to create:");
    gets(line);

    /* parse the new file name to the dta */
    parsfnm(line, &blk, 1);
    printf("%d %s\n", blk.fcb_drive, blk.fcb_name);

    /* request DOS services to create file */
    if (bdosptr(0x16, &blk, 0) == -1)
    {
        perror("Error creating file");
        exit(1);
    }

    /* save old dta and set new dta */
    save_dta = getdta();
    setdta(buffer);

    /* write new records */
    blk.fcb_recsize = 256;
    blk.fcb_random = 0L;
    result = randbwr(&blk, 1);
    printf("result = %d\n", result);

    if (!result)
        printf("Write OK\n");
    else
```

```
    {
        perror("Disk error");
        exit(1);
    }

    /* request DOS services to close the file */
    if (bdosptr(0x10, &blk, 0) == -1)
    {
        perror("Error closing file");
        exit(1);
    }

    /* reset the old dta */
    setdta(save_dta);
    return 0;
}
```

============================

函数名: setfillpattern

============================

功　能: 选择用户定义的填充模式
用　法: void far setfillpattern(char far *upattern, int color);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int maxx, maxy;

    /* a user defined fill pattern */
    char pattern[8] = {0x00, 0x70, 0x20, 0x27, 0x24, 0x24, 0x07, 0x00};

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
```

```
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   maxx = getmaxx();
   maxy = getmaxy();
   setcolor(getmaxcolor());

   /* select a user defined fill pattern */
   setfillpattern(pattern, getmaxcolor());

   /* fill the screen with the pattern */
   bar(0, 0, maxx, maxy);

   /* clean up */
   getch();
   closegraph();
   return 0;
}
```

==========================

函数名: setfillstyle

==========================

功　能: 设置填充模式和颜色
用　法: void far setfillstyle(int pattern, int color);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <conio.h>

/* the names of the fill styles supported */
char *fname[] = { "EMPTY_FILL",
                  "SOLID_FILL",
                  "LINE_FILL",
```

```c
                "LTSLASH_FILL",
                "SLASH_FILL",
                "BKSLASH_FILL",
                "LTBKSLASH_FILL",
  "HATCH_FILL",
                "XHATCH_FILL",
                "INTERLEAVE_FILL",
                "WIDE_DOT_FILL",
                "CLOSE_DOT_FILL",
   "USER_FILL"
                };

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int style, midx, midy;
   char stylestr[40];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   midx = getmaxx() / 2;
   midy = getmaxy() / 2;

   for (style = EMPTY_FILL; style < USER_FILL; style++)
   {
      /* select the fill style */
      setfillstyle(style, getmaxcolor());

      /* convert style into a string */
      strcpy(stylestr, fname[style]);

      /* fill a bar */
```

```
        bar3d(0, 0, midx-10, midy, 0, 0);

        /* output a message */
        outtextxy(midx, midy, stylestr);

        /* wait for a key */
        getch();
        cleardevice();
    }

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

===========================

函数名: setftime

===========================

功　能: 设置文件日期和时间
用　法: int setftime(int handle, struct ftime *ftimep);
程序例:

```
#include <stdio.h>
#include <process.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
    struct ftime filet;
    FILE *fp;

    if ((fp = fopen("TEST.$$$", "w")) == NULL)
    {
        perror("Error:");
        exit(1);
    }

    fprintf(fp, "testing...\n");

    /* load ftime structure with new time and date */
```

```
    filet.ft_tsec = 1;
    filet.ft_min = 1;
    filet.ft_hour = 1;
    filet.ft_day = 1;
    filet.ft_month = 1;
    filet.ft_year = 21;

    /* show current directory for time and date */
    system("dir TEST.$$$");

    /* change the time and date stamp*/
    setftime(fileno(fp), &filet);

    /* close and remove the temporary file */
    fclose(fp);

    system("dir TEST.$$$");

    unlink("TEST.$$$");
    return 0;
}
```

==============================

函数名: setgraphbufsize

==============================

功　能: 改变内部图形缓冲区的大小
用　法: unsigned far setgraphbufsize(unsigned bufsize);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

#define BUFSIZE 1000 /* internal graphics buffer size */

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int x, y, oldsize;
    char msg[80];
```

```
   /* set the size of the internal graphics buffer */
   /* before making a call to initgraph.          */
   oldsize = setgraphbufsize(BUFSIZE);

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   x = getmaxx() / 2;
   y = getmaxy() / 2;

   /* output some messages */
   sprintf(msg, "Graphics buffer size: %d", BUFSIZE);
   settextjustify(CENTER_TEXT, CENTER_TEXT);
   outtextxy(x, y, msg);
   sprintf(msg, "Old graphics buffer size: %d", oldsize);
   outtextxy(x, y+textheight("W"), msg);

   /* clean up */
   getch();
   closegraph();
   return 0;
}
```

===========================

函数名: setgraphmode

===========================

功　能: 将系统设置成图形模式且清屏
用　法: void far setgraphmode(int mode);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int x, y;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   x = getmaxx() / 2;
   y = getmaxy() / 2;

   /* output a message */
   settextjustify(CENTER_TEXT, CENTER_TEXT);
   outtextxy(x, y, "Press any key to exit graphics:");
   getch();

   /* restore system to text mode */
   restorecrtmode();
   printf("We're now in text mode.\n");
   printf("Press any key to return to graphics mode:");
   getch();

   /* return to graphics mode */
   setgraphmode(getgraphmode());

   /* output a message */
   settextjustify(CENTER_TEXT, CENTER_TEXT);
   outtextxy(x, y, "We're back in graphics mode.");
```

```
    outtextxy(x, y+textheight("W"), "Press any key to halt:");

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: setjmp

============================

功　能: 非局部转移
用　法: int setjmp(jmp_buf env);
程序例:

```
#include <stdio.h>
#include <process.h>
#include <setjmp.h>

void subroutine(void);

jmp_buf jumper;

int main(void)
{
    int value;

    value = setjmp(jumper);
    if (value != 0)
    {
        printf("Longjmp with value %d\n", value);
        exit(value);
    }
    printf("About to call subroutine ... \n");
    subroutine();
    return 0;
}

void subroutine(void)
{
    longjmp(jumper,1);
```

```
    }
```

==========================

函数名: setlinestyle

==========================

功　能: 设置当前画线宽度和类型
用　法: void far setlinestyle(int linestype, unsigned upattern);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <conio.h>

/* the names of the line styles supported */
char *lname[] = {
    "SOLID_LINE",
    "DOTTED_LINE",
    "CENTER_LINE",
    "DASHED_LINE",
    "USERBIT_LINE"
    };

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;

    int style, midx, midy, userpat;
    char stylestr[40];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
```

```c
      exit(1); /* terminate with an error code */
   }

   midx = getmaxx() / 2;
   midy = getmaxy() / 2;

   /* a user defined line pattern */
   /* binary: "0000000000000001" */
   userpat = 1;

   for (style=SOLID_LINE; style<=USERBIT_LINE; style++)
   {
      /* select the line style */
      setlinestyle(style, userpat, 1);

      /* convert style into a string */
      strcpy(stylestr, lname[style]);

      /* draw a line */
      line(0, 0, midx-10, midy);

      /* draw a rectangle */
      rectangle(0, 0, getmaxx(), getmaxy());

      /* output a message */
      outtextxy(midx, midy, stylestr);

      /* wait for a key */
      getch();
      cleardevice();
   }

   /* clean up */
   closegraph();
   return 0;
}
```

============================

函数名: setmem

============================

功　能: 存值到存储区

用　法: void setmem(void *addr, int len, char value);
程序例:

```c
#include <stdio.h>
#include <alloc.h>
#include <mem.h>

int main(void)
{
    char *dest;

    dest = calloc(21, sizeof(char));
    setmem(dest, 20, 'c');
    printf("%s\n", dest);

    return 0;
}
```

============================

函数名: setmode

============================

功　能: 设置打开文件方式
用　法: int setmode(int handle, unsigned mode);
程序例:

```c
#include <stdio.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
    int result;

    result = setmode(fileno(stdprn), O_TEXT);
    if (result == -1)
        perror("Mode not available\n");
    else
        printf("Mode successfully switched\n");
    return 0;
}
```

==========================

函数名: setpalette

==========================

功　能: 改变调色板的颜色
用　法: void far setpalette(int index, int actural_color);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int color, maxcolor, ht;
   int y = 10;
   char msg[80];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   maxcolor = getmaxcolor();
   ht = 2 * textheight("W");

   /* display the default colors */
   for (color=1; color<=maxcolor; color++)
   {
      setcolor(color);
      sprintf(msg, "Color: %d", color);
```

```
        outtextxy(1, y, msg);
        y += ht;
    }

    /* wait for a key */
    getch();

    /* black out the colors one by one */
    for (color=1; color<=maxcolor; color++)
    {
        setpalette(color, BLACK);
        getch();
    }

    /* clean up */
    closegraph();
    return 0;
}
```

============================

函数名: setrgbpalette

============================

功　能: 定义 IBM8514 图形卡的颜色
用　法: void far setrgbpalette(int colornum, int red, int green, int blue);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* select a driver and mode that supports the use */
    /* of the setrgbpalette function.                  */
    int gdriver = VGA, gmode = VGAHI, errorcode;
    struct palettetype pal;
    int i, ht, y, xmax;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");
```

```
    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    /* grab a copy of the palette */
    getpalette(&pal);

    /* create gray scale */
    for (i=0; i<pal.size; i++)
        setrgbpalette(pal.colors[i], i*4, i*4, i*4);

    /* display the gray scale */
    ht = getmaxy() / 16;
    xmax = getmaxx();
    y = 0;
    for (i=0; i<pal.size; i++)
    {
        setfillstyle(SOLID_FILL, i);
        bar(0, y, xmax, y+ht);
        y += ht;
    }

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

==============================

函数名: settextjustify

==============================

功　能: 为图形函数设置文本的对齐方式
用　法: void far settextjustify(int horiz, int vert);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/* function prototype */
void xat(int x, int y);

/* horizontal text justification settings */
char *hjust[] = { "LEFT_TEXT",
                  "CENTER_TEXT",
                  "RIGHT_TEXT"
                };

/* vertical text justification settings */
char *vjust[] = { "LEFT_TEXT",
    "CENTER_TEXT",
    "RIGHT_TEXT"
                };

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int midx, midy, hj, vj;
   char msg[80];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   midx = getmaxx() / 2;
   midy = getmaxy() / 2;

   /* loop through text justifications */
```

```
    for (hj=LEFT_TEXT; hj<=RIGHT_TEXT; hj++)
       for (vj=LEFT_TEXT; vj<=RIGHT_TEXT; vj++)
      {
          cleardevice();
          /* set the text justification */
          settextjustify(hj, vj);

          /* create a message string */
          sprintf(msg, "%s  %s", hjust[hj], vjust[vj]);

   /* create cross hairs on the screen */
   xat(midx, midy);

          /* output the message */
          outtextxy(midx, midy, msg);
          getch();
      }

   /* clean up */
   closegraph();
   return 0;
}

/* draw an "x" at (x, y) */
void xat(int x, int y)
{
  line(x-4, y, x+4, y);
  line(x, y-4, x, y+4);
}
```

============================

函数名: settextstyle

============================

功　能: 为图形输出设置当前的文本属性
用　法: void far settextstyle (int font, int direction, char size);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
```

```c
/* the names of the text styles supported */
char *fname[] = { "DEFAULT font",
                  "TRIPLEX font",
                  "SMALL font",
                  "SANS SERIF font",
                  "GOTHIC font"
                };

int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int style, midx, midy;
   int size = 1;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   midx = getmaxx() / 2;
   midy = getmaxy() / 2;

   settextjustify(CENTER_TEXT, CENTER_TEXT);

   /* loop through the available text styles */
   for (style=DEFAULT_FONT; style<=GOTHIC_FONT; style++)
   {
      cleardevice();
      if (style == TRIPLEX_FONT)
         size = 4;

      /* select the text style */
      settextstyle(style, HORIZ_DIR, size);

      /* output a message */
```

```
        outtextxy(midx, midy, fname[style]);
        getch();
    }


    /* clean up */
    closegraph();
    return 0;
}
```

============================

函数名: settextstyle

============================
功 能: 为图形输出设置当前的文本属性
用 法: void far settextstyle (int font, int direction, char size);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/* the names of the text styles supported */
char *fname[] = { "DEFAULT font",
                  "TRIPLEX font",
                  "SMALL font",
                  "SANS SERIF font",
                  "GOTHIC font"
                };

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int style, midx, midy;
    int size = 1;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
```

```
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    settextjustify(CENTER_TEXT, CENTER_TEXT);

    /* loop through the available text styles */
    for (style=DEFAULT_FONT; style<=GOTHIC_FONT; style++)
    {
        cleardevice();
        if (style == TRIPLEX_FONT)
            size = 4;

        /* select the text style */
        settextstyle(style, HORIZ_DIR, size);

        /* output a message */
        outtextxy(midx, midy, fname[style]);
        getch();
    }

    /* clean up */
    closegraph();
    return 0;
}
```

============================

函数名: settime

============================
功　能: 设置系统时间
用　法: void settime(struct time *timep);
程序例:

```
#include <stdio.h>
#include <dos.h>
```

```c
int main(void)
{
   struct  time t;

   gettime(&t);
   printf("The current minute is: %d\n", t.ti_min);
   printf("The current hour is: %d\n", t.ti_hour);
   printf("The current hundredth of a second is: %d\n", t.ti_hund);
   printf("The current second is: %d\n", t.ti_sec);

   /* Add one to the minutes struct element and then call settime  */
   t.ti_min++;
   settime(&t);

   return 0;
}
```

============================

函数名: setusercharsize

============================

功　能: 为矢量字体改变字符宽度和高度
用　法: void far setusercharsize(int multx, int dirx, int multy, int diry);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
   /* request autodetection */
   int gdriver = DETECT, gmode, errorcode;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)        /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
```

```c
        printf("Press any key to halt:");
        getch();
        exit(1);                    /* terminate with an error code */
    }

    /* select a text style */
    settextstyle(TRIPLEX_FONT, HORIZ_DIR, 4);

    /* move to the text starting position */
    moveto(0, getmaxy() / 2);

    /* output some normal text */
    outtext("Norm ");

    /* make the text 1/3 the normal width */
    setusercharsize(1, 3, 1, 1);
    outtext("Short ");

    /* make the text 3 times normal width */
    setusercharsize(3, 1, 1, 1);
    outtext("Wide");

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: setvbuf

============================

功　能: 把缓冲区与流相关
用　法: int setvbuf(FILE *stream, char *buf, int type, unsigned size);
程序例:

```c
#include <stdio.h>

int main(void)
{
    FILE *input, *output;
    char bufr[512];

    input = fopen("file.in", "r+b");
```

```
   output = fopen("file.out", "w");

   /* set up input stream for minimal disk access,
      using our own character buffer */
   if (setvbuf(input, bufr, _IOFBF, 512) != 0)
     printf("failed to set up buffer for input file\n");
   else
     printf("buffer set up for input file\n");

   /* set up output stream for line buffering using space that
      will be obtained through an indirect call to malloc */
   if (setvbuf(output, NULL, _IOLBF, 132) != 0)
     printf("failed to set up buffer for output file\n");
   else
     printf("buffer set up for output file\n");

   /* perform file I/O here */

   /* close files */
   fclose(input);
   fclose(output);
   return 0;
}
```

============================

函数名: setvect

============================

功　能: 设置中断矢量入口
用　法: void setvect(int intr_num, void interrupt(*isr)());
程序例:

```
/***NOTE:
    This is an interrupt service routine.  You can NOT compile this
    program with Test Stack Overflow turned on and get an executable
    file which will operate correctly. */

#include <stdio.h>
#include <dos.h>
#include <conio.h>

#define INTR 0X1C    /* The clock tick interrupt */
```

```c
void interrupt ( *oldhandler)(void);

int count=0;

void interrupt handler(void)
{
/* increase the global counter */
   count++;

/* call the old routine */
   oldhandler();
}

int main(void)
{
/* save the old interrupt vector */
   oldhandler = getvect(INTR);

/* install the new interrupt handler */
   setvect(INTR, handler);

/* loop until the counter exceeds 20 */
   while (count < 20)
      printf("count is %d\n",count);

/* reset the old interrupt handler */
   setvect(INTR, oldhandler);

   return 0;
}
```

============================

函数名: setverify

============================

功　能: 设置验证状态
用　法: void setverify(int value);
程序例:

```c
#include <stdio.h>
#include <conio.h>
#include <dos.h>
```

```c
int main(void)
{
    int verify_flag;

    printf("Enter 0 to set verify flag off\n");
    printf("Enter 1 to set verify flag on\n");

    verify_flag = getch() - 0;

    setverify(verify_flag);

    if (getverify())
        printf("DOS verify flag is on\n");
    else
        printf("DOS verify flag is off\n");

    return 0;
}
```

============================

函数名: setviewport

============================

功　能: 为图形输出设置当前视口
用　法: void far setviewport(int left, int top, int right,
　　　　　int bottom, int clipflag);
程序例:

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

#define CLIP_ON 1   /* activates clipping in viewport */

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");
```

```
    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    setcolor(getmaxcolor());

    /* message in default full-screen viewport */
    outtextxy(0, 0, "* <-- (0, 0) in default viewport");

    /* create a smaller viewport */
    setviewport(50, 50, getmaxx()-50, getmaxy()-50, CLIP_ON);

    /* display some text */
    outtextxy(0, 0, "* <-- (0, 0) in smaller viewport");

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: setvisualpage

============================

功　能: 设置可见图形页号
用　法: void far setvisualpage(int pagenum);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
```

```c
/* select a driver and mode that supports */
/* multiple pages.                          */
int gdriver = EGA, gmode = EGAHI, errorcode;
int x, y, ht;

/* initialize graphics and local variables */
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk)   /* an error occurred */
{
   printf("Graphics error: %s\n", grapherrormsg(errorcode));
   printf("Press any key to halt:");
   getch();
   exit(1); /* terminate with an error code */
}

x = getmaxx() / 2;
y = getmaxy() / 2;
ht = textheight("W");

/*  select the off screen page for drawing */
setactivepage(1);

/* draw a line on page #1 */
line(0, 0, getmaxx(), getmaxy());

/* output a message on page #1 */
settextjustify(CENTER_TEXT, CENTER_TEXT);
outtextxy(x, y, "This is page #1:");
outtextxy(x, y+ht, "Press any key to halt:");

/* select drawing to page #0 */
setactivepage(0);

/* output a message  on page #0 */
outtextxy(x, y, "This is page #0.");
outtextxy(x, y+ht, "Press any key to view page #1:");
getch();

/* select page #1 as the visible page */
setvisualpage(1);
```

```
    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

==========================

函数名: setwritemode

==========================

功　能: 设置图形方式下画线的输出模式
用　法: void far setwritemode(int mode);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main()
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int xmax, ymax;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    xmax = getmaxx();
    ymax = getmaxy();

    /* select XOR drawing mode */
    setwritemode(XOR_PUT);
```

```
    /* draw a line */
    line(0, 0, xmax, ymax);
    getch();

    /* erase the line by drawing over it */
    line(0, 0, xmax, ymax);
    getch();

    /* select overwrite drawing mode */
    setwritemode(COPY_PUT);

    /* draw a line */
    line(0, 0, xmax, ymax);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: signal

============================

功　能: 设置某一信号的对应动作
用　法: int signal(int sig, sigfun fname);
程序例:

```
/* This example installs a signal handler routine for SIGFPE,
   catches an integer overflow condition, makes an adjustment
   to AX register, and returns. This example program MAY cause
   your computer to crash, and will produce runtime errors
   depending on which memory model is used.
*/

#pragma inline
#include <stdio.h>
#include <signal.h>

void Catcher(int sig, int type, int *reglist)
{
    printf("Caught it!\n");
```

```c
    *(reglist + 8) = 3;                /* make return AX = 3 */
}

int main(void)
{
    signal(SIGFPE, Catcher);
    asm     mov     ax,07FFFH       /* AX = 32767 */
    asm     inc     ax              /* cause overflow */
    asm     into                    /* activate handler */

    /* The handler set AX to 3 on return. If that hadn't happened,
       there would have been another exception when the next 'into'
       was executed after the 'dec' instruction. */
    asm     dec     ax              /* no overflow now */
    asm     into                    /* doesn't activate */
    return 0;
}
```

==========================

函数名: sin

==========================

功　能: 正弦函数
用　法: double sin(double x);
程序例:

```c
#include <stdio.h>
#include <math.h>

int main(void)
{
    double result, x = 0.5;

    result = sin(x);
    printf("The sin() of %lf is %lf\n", x, result);
    return 0;
}
```

==========================

函数名: sinh

============================

功　能: 双曲正弦函数
用　法: double sinh(double x);
程序例:

```c
#include <stdio.h>
#include <math.h>

int main(void)
{
    double result, x = 0.5;

    result = sinh(x);
    printf("The hyperbolic sin() of %lf is %lf\n", x, result);
    return 0;
}
```

============================

函数名: sleep

============================

功　能: 执行挂起一段时间
用　法: unsigned sleep(unsigned seconds);
程序例:

```c
#include <dos.h>
#include <stdio.h>

int main(void)
{
    int i;

    for (i=1; i<5; i++)
    {
        printf("Sleeping for %d seconds\n", i);
        sleep(i);
    }
    return 0;
}
```

===========================

函数名: sopen

===========================

功　能: 打开一共享文件
用　法: int sopen(char *pathname, int access, int shflag, int permiss);
程序例:

```c
#include <io.h>
#include <fcntl.h>
#include <sys\stat.h>
#include <process.h>
#include <share.h>
#include <stdio.h>

int main(void)
{
   int handle;
   int status;

   handle = sopen("c:\\autoexec.bat", O_RDONLY, SH_DENYNO, S_IREAD);

   if (!handle)
   {
      printf("sopen failed\n");
      exit(1);
   }

   status = access("c:\\autoexec.bat", 6);
   if (status == 0)
      printf("read/write access allowed\n");
   else
      printf("read/write access not allowed\n");

   close(handle);
   return 0;
}
```

===========================

函数名: sound

================================

功　能: 以指定频率打开 PC 扬声器

用　法: void sound(unsigned frequency);

程序例:

```c
/* Emits a 7-Hz tone for 10 seconds.
   Your PC may not be able to emit a 7-Hz tone. */
#include <dos.h>

int main(void)
{
    sound(7);
    delay(10000);
    nosound();
    return 0;
}
```

================================

函数名: spawnl

================================

功　能: 创建并运行子程序

用　法: int spawnl(int mode, char *pathname, char *arg0,
        arg1, ... argn, NULL);

程序例:

```c
#include <process.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    int result;

    clrscr();
    result = spawnl(P_WAIT, "tcc.exe", NULL);
    if (result == -1)
    {
        perror("Error from spawnl");
        exit(1);
```

```
    }
    return 0;
}
```

============================

函数名: spawnle

============================

功　能: 创建并运行子程序
用　法: int spawnle(int mode, char *pathname, char *arg0,
        arg1,..., argn, NULL);
程序例:

```
/* spawnle() example */

#include <process.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    int result;

    clrscr();
    result = spawnle(P_WAIT, "tcc.exe", NULL, NULL);
    if (result == -1)
    {
        perror("Error from spawnle");
        exit(1);
    }
    return 0;
}
```

============================

函数名: sprintf

============================

功　能: 送格式化输出到字符串中
用　法: int sprintf(char *string, char *farmat [,argument,...]);
程序例:

```c
#include <stdio.h>
#include <math.h>

int main(void)
{
    char buffer[80];

    sprintf(buffer, "An approximation of Pi is %f\n", M_PI);
    puts(buffer);
    return 0;
}
```

============================

函数名: sqrt

============================

功　能: 计算平方根
用　法: double sqrt(double x);
程序例:

```c
#include <math.h>
 #include <stdio.h>

 int main(void)
 {
    double x = 4.0, result;

    result = sqrt(x);
    printf("The square root of %lf is %lf\n", x, result);
    return 0;
}
```

============================

函数名: srand

============================

功　能: 初始化随机数发生器
用　法: void srand(unsigned seed);
程序例:

```c
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
```

```
int main(void)
{
    int i;
    time_t t;

    srand((unsigned) time(&t));
    printf("Ten random numbers from 0 to 99\n\n");
    for(i=0; i<10; i++)
        printf("%d\n", rand() % 100);
    return 0;
}
```

===========================

函数名: sscanf

===========================

功　能: 执行从字符串中的格式化输入
用　法: int sscanf(char *string, char *format[,argument,...]);
程序例:

```
#include <stdio.h>
#include <conio.h>

int main(void)
{
    char label[20];
    char name[20];
    int entries = 0;
    int loop, age;
    double salary;

    struct Entry_struct
    {
        char   name[20];
        int    age;
        float salary;
    } entry[20];

/* Input a label as a string of characters restricting to 20 characters */
    printf("\n\nPlease enter a label for the chart: ");
    scanf("%20s", label);
    fflush(stdin);  /* flush the input stream in case of bad input */
```

```c
/* Input number of entries as an integer */
   printf("How many entries will there be? (less than 20) ");
   scanf("%d", &entries);
   fflush(stdin);   /* flush the input stream in case of bad input */


/* input a name restricting input to only letters upper or lower case */
   for (loop=0;loop<entries;++loop)
   {
      printf("Entry %d\n", loop);
      printf(" Name   : ");
      scanf("%[A-Za-z]", entry[loop].name);
      fflush(stdin);  /* flush the input stream in case of bad input */


/* input an age as an integer */
      printf(" Age    : ");
      scanf("%d", &entry[loop].age);
      fflush(stdin);  /* flush the input stream in case of bad input */


/* input a salary as a float */
      printf(" Salary : ");
      scanf("%f", &entry[loop].salary);
      fflush(stdin); /* flush the input stream in case of bad input */
   }

/* Input a name, age and salary as a string, integer, and double */
   printf("\nPlease enter your name, age and salary\n");
   scanf("%20s %d %lf", name, &age, &salary);



/* Print out the data that was input */
   printf("\n\nTable %s\n",label);
   printf("Compiled by %s  age %d  $%15.2lf\n", name, age, salary);
   printf("--------------------------------------------------\n");
   for (loop=0;loop<entries;++loop)
      printf("%4d | %-20s | %5d | %15.2lf\n",
         loop + 1,
   entry[loop].name,
   entry[loop].age,
   entry[loop].salary);
   printf("--------------------------------------------------\n");
   return 0;
}
```

============================

函数名: stat

============================

功　能: 读取打开文件信息
用　法: int stat(char *pathname, struct stat *buff);
程序例:

```
#include <sys\stat.h>
#include <stdio.h>
#include <time.h>

#define FILENAME "TEST.$$$"

int main(void)
{
   struct stat statbuf;
   FILE *stream;

   /* open a file for update */
   if ((stream = fopen(FILENAME, "w+")) == NULL)
   {
      fprintf(stderr, "Cannot open output file.\n");
      return(1);
   }

   /* get information about the file */
   stat(FILENAME, &statbuf);

   fclose(stream);

   /* display the information returned */
   if (statbuf.st_mode & S_IFCHR)
      printf("Handle refers to a device.\n");
   if (statbuf.st_mode & S_IFREG)
      printf("Handle refers to an ordinary file.\n");
   if (statbuf.st_mode & S_IREAD)
      printf("User has read permission on file.\n");
   if (statbuf.st_mode & S_IWRITE)
      printf("User has write permission on file.\n");

   printf("Drive letter of file: %c\n", 'A'+statbuf.st_dev);
   printf("Size of file in bytes: %ld\n", statbuf.st_size);
```

```
    printf("Time file last opened: %s\n", ctime(&statbuf.st_ctime));
    return 0;
}
```

================================

函数名: _status87

================================

功　能: 取浮点状态
用　法: unsigned int _status87(void);
程序例:

```
#include <stdio.h>
#include <float.h>

int main(void)
{
    float x;
    double y = 1.5e-100;

    printf("Status 87 before error: %x\n", _status87());

    x = y;  /* <-- force an error to occur */
    y = x;

    printf("Status 87 after error : %x\n", _status87());
    return 0;
}
```

================================

函数名: stime

================================

功　能: 设置时间
用　法: int stime(long *tp);
程序例:

```
#include <stdio.h>
#include <time.h>
#include <dos.h>
```

```c
int main(void)
{
    time_t t;
    struct tm *area;

    t = time(NULL);
    area = localtime(&t);
    printf("Number of seconds since 1/1/1970 is: %ld\n", t);
    printf("Local time is: %s", asctime(area));

    t++;
    area = localtime(&t);
    printf("Add a second:  %s", asctime(area));

    t += 60;
    area = localtime(&t);
    printf("Add a minute:  %s", asctime(area));

    t += 3600;
    area = localtime(&t);
    printf("Add an hour:   %s", asctime(area));

    t += 86400L;
    area = localtime(&t);
    printf("Add a day:     %s", asctime(area));

    t += 2592000L;
    area = localtime(&t);
    printf("Add a month:   %s", asctime(area));

    t += 31536000L;
    area = localtime(&t);
    printf("Add a year:    %s", asctime(area));
    return 0;
}
```

==============================

函数名: stpcpy

==============================

功　能: 拷贝一个字符串到另一个
用　法: char *stpcpy(char *destin, char *source);

程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char string[10];
    char *str1 = "abcdefghi";

    stpcpy(string, str1);
    printf("%s\n", string);
    return 0;
}
```

=============================

函数名: strcat

=============================

功　能: 字符串拼接函数
用　法: char *strcat(char *destin, char *source);
程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char destination[25];
    char *blank = " ", *c = "C++", *Borland = "Borland";

    strcpy(destination, Borland);
    strcat(destination, blank);
    strcat(destination, c);

    printf("%s\n", destination);
    return 0;
}
```

=============================

函数名: strchr

==============================

功　能: 在一个串中查找给定字符的第一个匹配之处\
用　法: char *strchr(char *str, char c);
程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
 {
    char string[15];
    char *ptr, c = 'r';

    strcpy(string, "This is a string");
    ptr = strchr(string, c);
    if (ptr)
       printf("The character %c is at position: %d\n", c, ptr-string);
    else
       printf("The character was not found\n");
    return 0;
 }
```


==============================

函数名: strcmp

==============================

功　能: 串比较
用　法: int strcmp(char *str1, char *str2);
程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
 {
    char *buf1 = "aaa", *buf2 = "bbb", *buf3 = "ccc";
    int ptr;

    ptr = strcmp(buf2, buf1);
    if (ptr > 0)
```

```c
      printf("buffer 2 is greater than buffer 1\n");
   else
      printf("buffer 2 is less than buffer 1\n");

   ptr = strcmp(buf2, buf3);
   if (ptr > 0)
      printf("buffer 2 is greater than buffer 3\n");
   else
      printf("buffer 2 is less than buffer 3\n");

   return 0;
}
```

============================

函数名: strncmpi

============================

功　能: 将一个串中的一部分与另一个串比较, 不管大小写
用　法: int strncmpi(char *str1, char *str2, unsigned maxlen);
程序例:

```c
#include <string.h>
#include <stdio.h>

int main(void)
{
   char *buf1 = "BBB", *buf2 = "bbb";
   int ptr;

   ptr = strcmpi(buf2, buf1);

   if (ptr > 0)
      printf("buffer 2 is greater than buffer 1\n");

   if (ptr < 0)
      printf("buffer 2 is less than buffer 1\n");

   if (ptr == 0)
      printf("buffer 2 equals buffer 1\n");

   return 0;
}
```

==========================

函数名: strcpy

==========================

功　能: 串拷贝
用　法: char *strcpy(char *str1, char *str2);
程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
 {
    char string[10];
    char *str1 = "abcdefghi";

    strcpy(string, str1);
    printf("%s\n", string);
    return 0;
 }
```

==========================

函数名: strcspn

==========================

功　能: 在串中查找第一个给定字符集内容的段
用　法: int strcspn(char *str1, char *str2);
程序例:

```
#include <stdio.h>
#include <string.h>
#include <alloc.h>

int main(void)
 {
    char *string1 = "1234567890";
    char *string2 = "747DC8";
    int length;
```

```
    length = strcspn(string1, string2);
    printf("Character where strings intersect is at position %d\n", length);

    return 0;
}
```

============================

函数名: strdup

============================

功　能: 将串拷贝到新建的位置处
用　法: char *strdup(char *str);
程序例:

```
#include <stdio.h>
#include <string.h>
#include <alloc.h>

int main(void)
{
    char *dup_str, *string = "abcde";

    dup_str = strdup(string);
    printf("%s\n", dup_str);
    free(dup_str);

    return 0;
}
```

============================

函数名: stricmp

============================

功　能: 以大小写不敏感方式比较两个串
用　法: int stricmp(char *str1, char *str2);
程序例:

```
#include <string.h>
#include <stdio.h>
```

```
int main(void)
{
    char *buf1 = "BBB", *buf2 = "bbb";
    int ptr;

    ptr = stricmp(buf2, buf1);

    if (ptr > 0)
      printf("buffer 2 is greater than buffer 1\n");

    if (ptr < 0)
      printf("buffer 2 is less than buffer 1\n");

    if (ptr == 0)
      printf("buffer 2 equals buffer 1\n");

    return 0;
}
```

============================

函数名: strerror

============================

功　能: 返回指向错误信息字符串的指针
用　法: char *strerror(int errnum);
程序例:

```
#include <stdio.h>
#include <errno.h>

int main(void)
{
    char *buffer;
    buffer = strerror(errno);
    printf("Error: %s\n", buffer);
    return 0;
}
```

============================

函数名: strcmpi

==============================

功　能: 将一个串与另一个比较，不管大小写
用　法: int strcmpi(char *str1, char *str2);
程序例:

```c
#include <string.h>
#include <stdio.h>

int main(void)
{
   char *buf1 = "BBB", *buf2 = "bbb";
   int ptr;

   ptr = strcmpi(buf2, buf1);

   if (ptr > 0)
      printf("buffer 2 is greater than buffer 1\n");

   if (ptr < 0)
      printf("buffer 2 is less than buffer 1\n");

   if (ptr == 0)
      printf("buffer 2 equals buffer 1\n");

   return 0;
}
```


==============================

函数名: strncmp

==============================

功　能: 串比较
用　法: int strncmp(char *str1, char *str2, int maxlen);
程序例:

```c
#include <string.h>
#include <stdio.h>

int  main(void)
```

```
{
   char *buf1 = "aaabbb", *buf2 = "bbbccc", *buf3 = "ccc";
   int ptr;

   ptr = strncmp(buf2, buf1, 3);
   if (ptr > 0)
      printf("buffer 2 is greater than buffer 1\n");
   else
      printf("buffer 2 is less than buffer 1\n");

   ptr = strncmp(buf2, buf3, 3);
   if (ptr > 0)
      printf("buffer 2 is greater than buffer 3\n");
   else
      printf("buffer 2 is less than buffer 3\n");

   return(0);
}
```

============================

函数名: strncmpi

============================

功　能: 把串中的一部分与另一串中的一部分比较, 不管大小写
用　法: int strncmpi(char *str1, char *str2);
程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
   char *buf1 = "BBBccc", *buf2 = "bbbccc";
   int ptr;

   ptr = strncmpi(buf2, buf1, 3);

   if (ptr > 0)
      printf("buffer 2 is greater than buffer 1\n");

   if (ptr < 0)
      printf("buffer 2 is less than buffer 1\n");
```

```
    if (ptr == 0)
        printf("buffer 2 equals buffer 1\n");

    return 0;
}
```

============================

函数名: **strncpy**

============================

功　能: 串拷贝
用　法: char *strncpy(char *destin, char *source, int maxlen);
程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char string[10];
    char *str1 = "abcdefghi";

    strncpy(string, str1, 3);
    string[3] = '\0';
    printf("%s\n", string);
    return 0;
}
```

============================

函数名: **strnicmp**

============================

功　能: 不注重大小写地比较两个串
用　法: int strnicmp(char *str1, char *str2, unsigned maxlen);
程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char *buf1 = "BBBccc", *buf2 = "bbbccc";
```

```c
    int ptr;

    ptr = strnicmp(buf2, buf1, 3);

    if (ptr > 0)
       printf("buffer 2 is greater than buffer 1\n");

    if (ptr < 0)
       printf("buffer 2 is less than buffer 1\n");

    if (ptr == 0)
       printf("buffer 2 equals buffer 1\n");

    return 0;
}
```

===========================

函数名: strnset

===========================

功　能: 将一个串中的所有字符都设为指定字符
用　法: char *strnset(char *str, char ch, unsigned n);
程序例:

```c
#include <stdio.h>
#include <string.h>

int main(void)
{
    char *string = "abcdefghijklmnopqrstuvwxyz";
    char letter = 'x';

    printf("string before strnset: %s\n", string);
    strnset(string, letter, 13);
    printf("string after  strnset: %s\n", string);

    return 0;
}
```

===========================

函数名: strpbrk

============================

功　能: 在串中查找给定字符集中的字符
用　法: char *strpbrk(char *str1, char *str2);
程序例:

```c
#include <stdio.h>
#include <string.h>

int main(void)
{
   char *string1 = "abcdefghijklmnopqrstuvwxyz";
   char *string2 = "onm";
   char *ptr;

   ptr = strpbrk(string1, string2);

   if (ptr)
      printf("strpbrk found first character: %c\n", *ptr);
   else
      printf("strpbrk didn't find character in set\n");

   return 0;
}
```

============================

函数名: strrchr

============================

功　能: 在串中查找指定字符的最后一个出现
用　法: char *strrchr(char *str, char c);
程序例:

```c
#include <string.h>
#include <stdio.h>

int main(void)
{
   char string[15];
   char *ptr, c = 'r';
```

```
    strcpy(string, "This is a string");
    ptr = strrchr(string, c);
    if (ptr)
        printf("The character %c is at position: %d\n", c, ptr-string);
    else
        printf("The character was not found\n");
    return 0;
}
```

==============================

函数名: strrev

==============================

功　能: 串倒转
用　法: char *strrev(char *str);
程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char *forward = "string";

    printf("Before strrev(): %s\n", forward);
    strrev(forward);
    printf("After strrev():  %s\n", forward);
    return 0;
}
```

==============================

函数名: strset

==============================

功　能: 将一个串中的所有字符都设为指定字符
用　法: char *strset(char *str, char c);
程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
```

```
{
    char string[10] = "123456789";
    char symbol = 'c';

    printf("Before strset(): %s\n", string);
    strset(string, symbol);
    printf("After strset():  %s\n", string);
    return 0;
}
```

===========================

函数名: strspn

===========================

功　能: 在串中查找指定字符集的子集的第一次出现
用　法: int strspn(char *str1, char *str2);
程序例:

```
#include <stdio.h>
#include <string.h>
#include <alloc.h>

int main(void)
{
    char *string1 = "1234567890";
    char *string2 = "123DC8";
    int length;

    length = strspn(string1, string2);
    printf("Character where strings differ is at position %d\n", length);
    return 0;
}
```

===========================

函数名: strstr

===========================

功　能: 在串中查找指定字符串的第一次出现
用　法: char *strstr(char *str1, char *str2);
程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char *str1 = "Borland International", *str2 = "nation", *ptr;

    ptr = strstr(str1, str2);
    printf("The substring is: %s\n", ptr);
    return 0;
}
```

============================

函数名: strtod

============================

功　能: 将字符串转换为 double 型值
用　法: double strtod(char *str, char **endptr);
程序例:

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    char input[80], *endptr;
    double value;

    printf("Enter a floating point number:");
    gets(input);
    value = strtod(input, &endptr);
    printf("The string is %s the number is %lf\n", input, value);
    return 0;
}
```

============================

函数名: strtok

============================

功　能: 查找由在第二个串中指定的分界符分隔开的单词
用　法: char *strtok(char *str1, char *str2);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char input[16] = "abc,d";
    char *p;

    /* strtok places a NULL terminator
    in front of the token, if found */
    p = strtok(input, ",");
    if (p)   printf("%s\n", p);

    /* A second call to strtok using a NULL
    as the first parameter returns a pointer
    to the character following the token  */
    p = strtok(NULL, ",");
    if (p)   printf("%s\n", p);
    return 0;
}
```

============================

函数名: strtol

============================

功　能: 将串转换为长整数
用　法: long strtol(char *str, char **endptr, int base);
程序例:

```
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    char *string = "87654321", *endptr;
    long lnumber;

    /* strtol converts string to long integer  */
    lnumber = strtol(string, &endptr, 10);
    printf("string = %s   long = %ld\n", string, lnumber);
```

```
    return 0;
}
```

============================

函数名: strupr

============================

功　能: 将串中的小写字母转换为大写字母
用　法: char *strupr(char *str);
程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char *string = "abcdefghijklmnopqrstuvwxyz", *ptr;

    /* converts string to upper case characters */
    ptr = strupr(string);
    printf("%s\n", ptr);
    return 0;
}
```

============================

函数名: swab

============================

功　能: 交换字节
用　法: void swab (char *from, char *to, int nbytes);
程序例:

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

char source[15] = "rFna koBlrna d";
char target[15];

int main(void)
{
```

```
    swab(source, target, strlen(source));
    printf("This is target: %s\n", target);
    return 0;
}
```

=============================

函数名: system

=============================

功　能: 发出一个 DOS 命令
用　法: int system(char *command);
程序例:

```
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    printf("About to spawn command.com and run a DOS command\n");
    system("dir");
    return 0;
}
```

=============================

函数名: tan

=============================

功　能: 正切函数
用　法: double tan(double x);
程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double result, x;

    x = 0.5;
    result = tan(x);
    printf("The tan of %lf is %lf\n", x, result);
    return 0;
```

}

============================

函数名: tanh

============================

功　能: 双曲正切函数
用　法: double tanh(double x);
程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double result, x;

    x = 0.5;
    result = tanh(x);
    printf("The hyperbolic tangent of %lf is %lf\n", x, result);
    return 0;
}
```

============================

函数名: tell

============================

功　能: 取文件指针的当前位置
用　法: long tell(int handle);
程序例:

```
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
    int handle;
```

```
    char msg[] = "Hello world";

    if ((handle = open("TEST.$$$", O_CREAT | O_TEXT | O_APPEND)) == -1)
    {
        perror("Error:");
        return 1;
    }
    write(handle, msg, strlen(msg));
    printf("The file pointer is at byte %ld\n", tell(handle));
    close(handle);
    return 0;
}
```

============================

函数名: textattr

============================

功　能: 设置文本属性
用　法: void textattr(int attribute);
程序例:

```
#include <conio.h>

int main(void)
{
    int i;

    clrscr();
    for (i=0; i<9; i++)
    {
        textattr(i + ((i+1) << 4));
        cprintf("This is a test\r\n");
    }

    return 0;
}
```

============================

函数名: textbackground

==============================

功　能: 选择新的文本背景颜色
用　法: void textbackground(int color);
程序例:

```c
#include <conio.h>

int main(void)
{
    int i, j;

    clrscr();
    for (i=0; i<9; i++)
    {
        for (j=0; j<80; j++)
          cprintf("C");
        cprintf("\r\n");
        textcolor(i+1);
        textbackground(i);
    }

    return 0;
}
```

==============================

函数名: textcolor

==============================

功　能: 在文本模式中选择新的字符颜色
用　法: void textcolor(int color);
程序例:

```c
#include <conio.h>

int main(void)
{
    int i;

    for (i=0; i<15; i++)
    {
        textcolor(i);
```

```
        cprintf("Foreground Color\r\n");
    }


    return 0;
}
```


============================

函数名: textheight

============================

功　能: 返回以像素为单位的字符串高度
用　法: int far textheight(char far *textstring);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int y = 0;
    int i;
    char msg[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)  /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    /* draw some text on the screen */
    for (i=1; i<11; i++)
```

```
    {
        /* select the text style, direction, and size */
        settextstyle(TRIPLEX_FONT, HORIZ_DIR, i);

        /* create a message string */
        sprintf(msg, "Size: %d", i);

        /* output the message */
        outtextxy(1, y, msg);

        /* advance to the next text line */
        y += textheight(msg);
    }

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: textmode

============================

功　能: 将屏幕设置成文本模式
用　法: void textmode(int mode);
程序例:

```
#include <conio.h>

int main(void)
{
    textmode(BW40);
    cprintf("ABC");
    getch();

    textmode(C40);
    cprintf("ABC");
    getch();

    textmode(BW80);
    cprintf("ABC");
```

```
        getch();

        textmode(C80);
        cprintf("ABC");
        getch();

        textmode(MONO);
        cprintf("ABC");
        getch();

        return 0;
}
```

============================

函数名: textwidth

============================

功　能: 返回以像素为单位的字符串宽度
用　法: int far textwidth(char far *textstring);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
        /* request auto detection */
        int gdriver = DETECT, gmode, errorcode;
        int x = 0, y = 0;
        int i;
        char msg[80];

        /* initialize graphics and local variables */
        initgraph(&gdriver, &gmode, "");

        /* read result of initialization */
        errorcode = graphresult();
        if (errorcode != grOk)  /* an error occurred */
        {
                printf("Graphics error: %s\n", grapherrormsg(errorcode));
                printf("Press any key to halt:");
```

```
        getch();
        exit(1); /* terminate with an error code */
    }

    y = getmaxy() / 2;

    settextjustify(LEFT_TEXT, CENTER_TEXT);
    for (i=1; i<11; i++)
    {
        /* select the text style, direction, and size */
        settextstyle(TRIPLEX_FONT, HORIZ_DIR, i);

        /* create a message string */
        sprintf(msg, "Size: %d", i);

        /* output the message */
        outtextxy(x, y, msg);

        /* advance to the end of the text */
        x += textwidth(msg);
    }

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

============================

函数名: time

============================

功　能: 取一天的时间
用　法: logn time(long *tloc);
程序例:

```
#include <time.h>
#include <stdio.h>
#include <dos.h>

int main(void)
{
    time_t t;
```

```
    t = time(NULL);
    printf("The number of seconds since January 1, 1970 is %ld",t);
    return 0;
}
```

============================

函数名: tmpfile

============================

功　能: 以二进制方式打开暂存文件
用　法: FILE *tmpfile(void);
程序例:

```
#include <stdio.h>
#include <process.h>

int main(void)
{
    FILE *tempfp;

    tempfp = tmpfile();
    if (tempfp)
        printf("Temporary file created\n");
    else
    {
        printf("Unable to create temporary file\n");
        exit(1);
    }

    return 0;
}
```

============================

函数名: tmpnam

============================

功　能: 创建一个唯一的文件名
用　法: char *tmpnam(char *sptr);
程序例:

```
#include <stdio.h>

int main(void)
{
    char name[13];

    tmpnam(name);
    printf("Temporary name: %s\n", name);
    return 0;
}
```

==============================

函数名: tolower

==============================

功　能: 把字符转换成小写字母
用　法: int tolower(int c);
程序例:

```
#include <string.h>
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int length, i;
    char *string = "THIS IS A STRING";

    length = strlen(string);
    for (i=0; i<length; i++)
    {
        string[i] = tolower(string[i]);
    }
    printf("%s\n",string);

    return 0;
}
```

==============================

函数名: toupper

===========================

功　能: 把字符转换成大写字母
用　法: int toupper(int c);
程序例:

```c
#include <string.h>
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int length, i;
    char *string = "this is a string";

    length = strlen(string);
    for (i=0; i<length; i++)
    {
        string[i] = toupper(string[i]);
    }

    printf("%s\n",string);

    return 0;
}
```

===========================

函数名: tzset

===========================

功　能: UNIX 时间兼容函数
用　法: void tzset(void);
程序例:

```c
#include <time.h>
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    time_t td;
```

```
    putenv("TZ=PST8PDT");
    tzset();
    time(&td);
    printf("Current time = %s\n", asctime(localtime(&td)));
    return 0;
}
```

==============================

函数名: ultoa

==============================

功　能: 转换一个无符号长整型数为字符串
用　法: char *ultoa(unsigned long value, char *string, int radix);
程序例:

```
#include <stdlib.h>
#include <stdio.h>

int main( void )
{
    unsigned long lnumber = 3123456789L;
    char string[25];

    ultoa(lnumber,string,10);
    printf("string = %s  unsigned long = %lu\n",string,lnumber);

    return 0;
}
```

==============================

函数名: ungetc

==============================

功　能: 把一个字符退回到输入流中
用　法: int ungetc(char c, FILE *stream);
程序例:

```
#include <stdio.h>
#include <ctype.h>
```

```c
int main( void )
{
    int i=0;
    char ch;

    puts("Input an integer followed by a char:");

    /* read chars until non digit or EOF */
    while((ch = getchar()) != EOF && isdigit(ch))
        i = 10 * i + ch - 48; /* convert ASCII into int value */

    /* if non digit char was read, push it back into input buffer */
    if (ch != EOF)
        ungetc(ch, stdin);

    printf("i = %d, next char in buffer = %c\n", i, getchar());
    return 0;
}
```

============================

函数名: ungetch

============================

功　能: 把一个字符退回到键盘缓冲区中
用　法: int ungetch(int c);
程序例:

```c
#include <stdio.h>
#include <ctype.h>
#include <conio.h>

int main( void )
{
    int i=0;
    char ch;

    puts("Input an integer followed by a char:");

    /* read chars until non digit or EOF */
    while((ch = getche()) != EOF && isdigit(ch))
        i = 10 * i + ch - 48; /* convert ASCII into int value */
```

```c
   /* if non digit char was read, push it back into input buffer */
   if (ch != EOF)
      ungetch(ch);

   printf("\n\ni = %d, next char in buffer = %c\n", i, getch());
   return 0;
}
```

===========================

函数名: unixtodos

===========================

功　能: 把日期和时间转换成 DOS 格式
用　法: void unixtodos(long utime, struct date *dateptr,
　　struct time *timeptr);
程序例:

```c
#include <stdio.h>
#include <dos.h>

char *month[] = {"---", "Jan", "Feb", "Mar", "Apr", "May", "Jun",
                 "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};

#define SECONDS_PER_DAY 86400L   /* the number of seconds in one day */

struct date dt;
struct time tm;

int main(void)
{
   unsigned long val;

/* get today's date and time */
   getdate(&dt);
   gettime(&tm);
   printf("today is %d %s %d\n", dt.da_day, month[dt.da_mon], dt.da_year);

/* convert date and time to unix format (number of seconds since Jan 1, 1970 */
   val = dostounix(&dt, &tm);
/* subtract 42 days worth of seconds */
   val -= (SECONDS_PER_DAY * 42);
```

```c
/* convert back to dos time and date */
    unixtodos(val, &dt, &tm);
    printf("42 days ago it was %d %s %d\n",
            dt.da_day, month[dt.da_mon], dt.da_year);
    return 0;
}
```

============================

函数名: unlink

============================

功　能: 删掉一个文件
用　法: int unlink(char *filename);
程序例:

```c
#include <stdio.h>
#include <io.h>

int main(void)
{
    FILE *fp = fopen("junk.jnk","w");
    int status;

    fprintf(fp,"junk");

    status = access("junk.jnk",0);
    if (status == 0)
        printf("File exists\n");
    else
        printf("File doesn't exist\n");

    fclose(fp);
    unlink("junk.jnk");
    status = access("junk.jnk",0);
    if (status == 0)
        printf("File exists\n");
    else
        printf("File doesn't exist\n");


    return 0;
}
```

============================

函数名: unlock

============================

功　能: 解除文件共享锁
用　法: int unlock(int handle, long offset, long length);
程序例:

```c
#include <io.h>
#include <fcntl.h>
#include <sys\stat.h>
#include <process.h>
#include <share.h>
#include <stdio.h>

int main(void)
{
    int handle, status;
    long length;

    handle = sopen("c:\\autoexec.bat", O_RDONLY, SH_DENYNO, S_IREAD);

    if (handle < 0)
    {
        printf("sopen failed\n");
        exit(1);
    }

    length = filelength(handle);
    status = lock(handle, 0L, length/2);

    if (status == 0)
        printf("lock succeeded\n");
    else
        printf("lock failed\n");

    status = unlock(handle, 0L, length/2);

    if (status == 0)
        printf("unlock succeeded\n");
    else
```

```
      printf("unlock failed\n");

   close(handle);
   return 0;
}
```

============================

函数名: **vfprintf**

============================

功 能: 送格式化输出到一流中
用 法: int vfprintf(FILE *stream, char *format, va_list param);
程序例:

```
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>

FILE *fp;

int vfpf(char *fmt, ...)
{
   va_list argptr;
   int cnt;

   va_start(argptr, fmt);
   cnt = vfprintf(fp, fmt, argptr);
   va_end(argptr);

   return(cnt);
}

int main(void)
{
   int inumber = 30;
   float fnumber = 90.0;
   char string[4] = "abc";

   fp = tmpfile();
   if (fp == NULL)
   {
```

```
        perror("tmpfile() call");
        exit(1);
    }

    vfpf("%d %f %s", inumber, fnumber, string);
    rewind(fp);
    fscanf(fp,"%d %f %s", &inumber, &fnumber, string);
    printf("%d %f %s\n", inumber, fnumber, string);
    fclose(fp);

    return 0;
}
```

==========================

函数名: **vfscanf**

==========================

功　能: 从流中执行格式化输入
用　法: int vfscanf(FILE *stream, char *format, va_list param);
程序例:

```
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>

FILE *fp;

int vfsf(char *fmt, ...)
{
    va_list  argptr;
    int cnt;

    va_start(argptr, fmt);
    cnt = vfscanf(fp, fmt, argptr);
    va_end(argptr);

    return(cnt);
}

int main(void)
{
    int inumber = 30;
```

```c
    float fnumber = 90.0;
        char string[4] = "abc";

    fp = tmpfile();
    if (fp == NULL)
    {
        perror("tmpfile() call");
        exit(1);
    }
    fprintf(fp,"%d %f %s\n", inumber, fnumber, string);
    rewind(fp);

    vfsf("%d %f %s",&inumber,&fnumber, string);
    printf("%d %f %s\n", inumber, fnumber, string);
    fclose(fp);

    return 0;
}
```

===========================

函数名: vprintf

===========================

功　能: 送格式化输出到 stdout 中
用　法: int vprintf(char *format, va_list param);
程序例:

```c
#include <stdio.h>
#include <stdarg.h>

int vpf(char *fmt, ...)
{
    va_list argptr;
    int cnt;

    va_start(argptr, format);
    cnt = vprintf(fmt, argptr);
    va_end(argptr);

    return(cnt);
}

int main(void)
```

```
{
    int inumber = 30;
    float fnumber = 90.0;
    char *string = "abc";

    vpf("%d %f %s\n", inumber, fnumber, string);

    return 0;
}
```

============================

函数名: vscanf

============================

功　能: 从 stdin 中执行格式化输入
用　法: int vscanf(char *format, va_list param);
程序例:

```
#include <stdio.h>
#include <conio.h>
#include <stdarg.h>

int vscnf(char *fmt, ...)
{
    va_list argptr;
    int cnt;

    printf("Enter an integer, a float,  and a string (e.g. i,f,s,)\n");
    va_start(argptr, fmt);
    cnt = vscanf(fmt, argptr);
    va_end(argptr);

    return(cnt);
}

int main(void)
{
    int inumber;
    float fnumber;
    char string[80];

    vscnf("%d, %f, %s", &inumber, &fnumber, string);
    printf("%d %f %s\n", inumber, fnumber, string);
```

```
   return 0;
}
```

==========================

函数名: vsprintf

==========================

功　能: 送格式化输出到串中
用　法: int vsprintf(char *string, char *format, va_list param);
程序例:

```c
#include <stdio.h>
#include <conio.h>
#include <stdarg.h>

char buffer[80];

int vspf(char *fmt, ...)
{
    va_list argptr;
    int cnt;

    va_start(argptr, fmt);
    cnt = vsprintf(buffer, fmt, argptr);
    va_end(argptr);

    return(cnt);
}

int main(void)
{
    int inumber = 30;
    float fnumber = 90.0;
    char string[4] = "abc";

    vspf("%d %f %s", inumber, fnumber, string);
    printf("%s\n", buffer);
    return 0;
}
```

==========================

函数名: vsscanf

==========================

功　能: 从流中执行格式化输入
用　法: int vsscanf(char *s, char *format, va_list param);
程序例:

```
#include <stdio.h>
#include <conio.h>
#include <stdarg.h>

char buffer[80] = "30 90.0 abc";

int vssf(char *fmt, ...)
{
    va_list  argptr;
    int cnt;

    fflush(stdin);

    va_start(argptr, fmt);
    cnt = vsscanf(buffer, fmt, argptr);
    va_end(argptr);

    return(cnt);
}

int main(void)
{
    int inumber;
    float fnumber;
    char string[80];

    vssf("%d %f %s", &inumber, &fnumber, string);
    printf("%d %f %s\n", inumber, fnumber, string);
    return 0;
}
```

==========================

函数名: wherex

==============================

功　能: 返回窗口内水平光标位置

用　法: int wherex(void);

程序例:

```
#include <conio.h>

int main(void)
{
    clrscr();
    gotoxy(10, 10);
    cprintf("Current location is X: %d  Y: %d\r\n", wherex(), wherey());
    getch();

    return 0;
}
```

==============================

函数名: wherey

==============================

功　能: 返回窗口内垂直光标位置

用　法: int wherey(void);

程序例:

```
#include <conio.h>

int main(void)
{
    clrscr();
    gotoxy(10, 10);
    cprintf("Current location is X: %d  Y: %d\r\n", wherex(), wherey());
    getch();

    return 0;
}
```

==============================

函数名: window

============================

功　能: 定义活动文本模式窗口

用　法: void window(int left, int top, int right, int bottom);

程序例:

```
#include <conio.h>

int main(void)
{

   window(10, 10, 40, 11);
   textcolor(BLACK);
   textbackground(WHITE);
   cprintf("This is a test\r\n");

   return 0;
}
```

============================

函数名: write

============================

功　能: 写到一文件中

用　法: int write(int handel, void *buf, int nbyte);

程序例:

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys\stat.h>
#include <io.h>
#include <string.h>

int main(void)
{
   int handle;
   char string[40];
   int length, res;

   /*
```

```c
   Create a file named "TEST.$$$" in the current directory and write
   a string to it.  If "TEST.$$$" already exists, it will be overwritten.
*/

   if ((handle = open("TEST.$$$", O_WRONLY | O_CREAT | O_TRUNC,
                       S_IREAD | S_IWRITE)) == -1)
   {
      printf("Error opening file.\n");
      exit(1);
   }

   strcpy(string, "Hello, world!\n");
   length = strlen(string);

   if ((res = write(handle, string, length)) != length)
   {
      printf("Error writing to the file.\n");
      exit(1);
   }
   printf("Wrote %d bytes to the file.\n", res);

   close(handle);
   return 0;
}
```