鲁沐浴 主编

C语言最新编程技巧200例

NEW

電子

C语言最新编程技巧200例

鲁沐浴　主编

電子工業出版社

(京)新登字055号

内 容 提 要

本书汇集了C语言最新编程技巧及实用程序200例。内容包括输入技术及实用程序；图形、图像处理技术及实用程序；汉字处理技术及实用程序；打印实用程序；文件查询、删除、拷贝实用程序；病毒检测、消除实用程序；加解密方法及实用程序；其它众多实用程序。

本书适用于广大C语言编程者、爱好者。

C语言最新编程技巧200例

鲁沐浴 主编

责任编辑：晁琴(特约) 张荣琴

电子工业出版社出版

北京市海淀区万寿路173信箱(100036)

电子工业出版社发行 各地新华书店经销

北京市顺义县天竺颖华印刷厂印刷

开本：787×1092毫米 1/16 印张：39.5 字数：1005千字

1995年1月第一版 1995年1月北京第一次印刷

印数：[illegible]册 定价：45.00元

ISBN 7-5053-2625-2/TP·801

前　言

C 语言以简洁的表达方式，高级语言的编程形式，低级语言的操作方式，具有灵活性强、移植性能好、目标代码质量高和丰富的函数调用，为设计高质量的、人机界面友好的程序提供了理想的工具。所以愈来愈受到程序开发者的青睐。其应用范围日趋广泛，应用水平不断提高，应用成果、开发经验、编程技巧和实用程序层出不穷。

为此，我们从近期众多的 C 语言应用开发经验中精选出 200 例，经加工、分类，汇编成册，供广大计算机用户，特别是 C 语言编程者、爱好者学习选用。本书虽无全面的技术理论论述，但一事一议、短小精悍，实用价值大。初学者可即取即用，或稍加修改即可用，节约开发时间，提高工作效率。有经验的设计者也能从中获取有益的启示，给自己的思路打开联想的天窗，从而编制出更完美的程序。

全书分为八部分，内容包括输入技术和实用程序；图形、图像处理技术及实用程序；汉字处理及实用程序；打印实用程序；文件查询、删除、拷贝实用程序；病毒检测、消除实用程序；加解密方法及实用程序；还有其它众多方面的实用程序。

由于有些文章的内容本身具有多重属性，只能按其主要属性予以归类。另外作者众多，水平经验有差别，文章风格各异，则便于交流经验、开拓思路、采众家之长，达到推陈出新之目的。

参加编审工作的还有金秋、海燕、乔晖、林海、靖涛。同时每篇文章都署有原作者。

编　者

1994 年 6 月

目　　录

第三部分　汉字处理技术及实用程序

第四部分　打印实用程序

第一部分

输入技术及实用程序

1.1 Turbo C 多功能控制输入函数

戴林清

笔者为 Turbo C 编写了一个多功能控制输入函数 get()。该函数具有以下功能：

1. 良好的编辑界面，显示数据初值，使用←和→键灵活改变光标位置，退格键和 Del 键删除左边或当前字符，Ins 键随时作插入和修改切换。

2. 较强的控制能力，可使用字符检测、集合检测以及格式修饰功能，在字符检测下滤去所有非法字符，在集合检测下识别输入字符的类型，在格式修饰下自动生成格式符，编辑时不予干扰，同时对输入进行逐位集合检测。

3. 配备了全屏幕编辑基础条件，使用光标键及有关控制键可实时结束编辑，函数返回中断键值，在此基础上稍作加工就能实现全屏幕编辑。

4. 杜绝了半个汉字。

以下是函数的调用方式：

```
int get (int r,int c,char *s,int l,char * ss);
```

各参数意义见源程序中注释。需要说明的是控制串 SS，根据 SS 首字符的取值不同，编辑将采取不同的控制方式。

1. 首字母为“&”时，为字符检测方式。“&”后的所有字符组成合法字符集，编辑时只有合法字符集中的字符才被接收，其余皆视为非法。

2. 首字母为“@”时，为集合检测方式。“@”后的那个字符为集合符，其取值范围及意义如下：

A—只允许字母 N—允许字母和数值

9—只允许数字 X—允许所有字符

编辑时将作类型判别，与集合符意义不符的键值将不予理睬。

3. 首字母非“&”或“@”时，为格式修饰方式。此时 SS 为格式串，其长度应与编辑数据长度相等，不等时程序会自动补齐或截尾。格式串中字符可取任意字符，并约定：如果格式串中某位不是 2 中所述集合符，则该位字符称为格式符，表示被编辑数据的对应位字符只能取该格式符，并且无须编辑，程序在编辑前会自动生成该位字符，而在以后的编辑中总是跳过它；如果格式串中某位是上述集合符，表示被编辑数据对应位字符需要编辑，且取值只能取相应类型的字符。

格式修饰是程序设计的关键，程序中采用连续结构链来记录被格式符分割的各个编辑区间的首尾位置，在编辑过程中随时记录光标所在区间，在此基础上完成光标在各编辑区间的跳动，并使删、插等操作的作用范围限定在本区间内。

读者只需将所列函数文件编译后，用库管理程序加入相应库，就能象调用其它 C 函数一

样直接调用函数 get(),以下是调用实例。

```
#include<stdio.h>
#include<conio.h>
int get (int r, int c,char * s,int l,char * ss);
main()
{ char sl[9],s3[9],s3[9];
  sl[8]=0;memset(sl,' ',8);
  strcpy(s2,s1);strcpy(s3,s1);
  textattr(15); clrscr();
  directvideo=0;/* 取消直接写视频 */
  get(4,9,s1,8,"&abcde");   /* 字符检测 */
  get (5,9,s2,8,"@N");    /* 集合检测 */
  get(6,9,s3,8,"99/99/99"); /* 格式修饰 */
}
```

本文所述 get()函数只能直接编辑字符型数据,对数值型数据则需在编辑前用有关函数转换成字符串,构造相应合法字符集,使用字符检测方式进行编辑,编辑完毕再转换成原来类型的数值。此外,表面上用 get() 编辑时,编辑区域不能超出一行,但只要对参数 S 和 1 进行合理组合,即可达成多行编辑效果。下例是对长 150 的字符串 t 进行三行编辑:

```
get(4,9,t,50,"");get (,5,9,&t[50],50,"");
get(6,9,&t[100],50,"");
```

源程序在 Turbo C2.0 下编译通过,少量改动即可移植到 C 的其它产品下。

源程序清单:

```
#include<stdio.h>
#include<stdib.h>
#include<conio.h>
#include<bios.h>
#include<ctype.h>
#define HOME 327 /* Home 键 */
#define END 335 /* End 键 */
#define LEFT 331   /* ←键 */
#define RIGHT 333   /* →键 */
#define INS 338   /* Ins 键 */
#define DEL 339   /* Del 键 */
#define UP 328   /* ↑键 */
#define DOWN 336   /* ↓键 */
#define ESC 27   /* Esc 键 */
#define ENTER 13   /* 回车键 */
#define BACK 8   /* 退格键 */
typedef struct }   /* 区间结构 */
int f;   /* 本区间起始位置 */
int e;   /* 本区间终止位置 */
}inve;
int inkey(void)   /* 接收键值函数 */
{int i, lo, hi;
  i=bioskey(0);   /* 接收键值 */
  lo=i&0x00ff;   /* 低字节 */
  hi=(i&0xff00)>>8;/* 高字节 */
  return((lo=0)? (hi+256):lo);
}
int get (int r,int c,char *s,int l,char *ss)
/* r:行; c:列; s:数据串; l:长;ss:控制串 */
{inve *b;/* 连续结构键 */
    int n,i,j,k,bno,bmax;
    int p=0;  /* 相对编辑位置 */
    int ins=0;   /* 插入状态标志 */
    int chan=0;   /* 消除半个汉字标记 */
    int loop=1;   /* 继续编辑标志 */
    unsigned char *t;   /* 编辑暂存区 */
    unsigned char cc;   /* 接收后半个汉字 */
    char *set;   /* 指向控制区 */
    char f=1;   /* 为 1 表示 set 为格式串 */
    unsigned char u;
int first,end;   /* 可编辑字符头尾序号 */
if (ss[0]=='&')i=strlen(ss);
else i=1+1;
set=(char *) malloc(i); /* 申请控制串空间 */
if (! set)return 0;
set [i-1]=0;   /* 控制区终止符 */
if(ss[0]=='&'){   /* 为字符检测时 */
   memcpy(set,&ss[1],i-1); /* 合法字符集 */
```

```
    f=0;}    /*置合法字符集标志*/
else if (ss[0]=='@'){    /*为集合检测时*/
    u=toupper(ss[1]);    /*集合符转大写*/
    memset(set,u,1);}    /*格式串各位赋值*/
else if (ss[0]! =0    {/*为格式修饰时*/
    memcpy(set,ss,1);    /*直接拷贝至 set*/
    i=strlen(ss) /*长度不够时反面以 X 补齐*/
    if(i<1)memset (&set[1],'X',1-i);}
else memset (set,'X',1);    /*ss 为空串时*/
if (f){/*有格式串时求各区间首尾序号*/
    if(strchr("ANX9",set[0])){j=1;k=1};
else{j=0;k=0}
i=1;
do {
if (j)
while (i<1&&strchr("ANX9",set[i++1]));
else
while (i<1&&! strchr("ANX9",set[i++1]));
if (i==1)break;
if(! j)k++;
j=! j;
} while(1);
if(! k) return 0;    /*不存在可编辑区间时*/
b=(inve *)malloc(k * sizeof(inve));
if(! b)return 0;
bmax=k-1;    /*最大区间序号(从 0 起算)*/
i=0;k=0;
while(i(1&&! strchr("ANX9",set[i]))i++;
j=1;
do{
      if(j){
      (b+k)->f=i;
      while(i<1&& strchr("ANX9",
set [i]))i++;
      (b+k)->e=i-1;k++;}
      else while(i<1&&! srtchr("ANX9",
set[i]))i++;
      if (i==1)break;
      j=! j;
  }while(1);
}
else{
   b=(inve *)malloc (sizeof(inve));
    if (! b)return 0;
    bmax=0;b->f=0;b->e=1-1;}
first=b->f;/*第一个可编辑字符*/
end=(b+bmax)->e;    /*最后一个可编辑字符
*/
p=first;bno=0;    /*当前编辑位置所在区间号
*/
t=(unsigned char *)malloc(1+1);
if (! t)return 0;
t[1]='\0'; memcpy(t,s);
if (f)for (i=0;i<1;i++)
    if (! strchr("ANX9",set[i]))
t[i]=set[i];
gotoxy(c,r);
for(i=0;i<1;i++)putch(t[i]);
do {
      chan=0;
  gotoxy(c+p,r) /*光标定位到当前编辑位置*/
  switch(n=inkey()){    /*接收键值并处理*/
    case HOME:    /*Home 到首*/
       if(p==first)loop=0 /*已为首则退出
    */
       p=first;bno=0;break;
    case END:    /*End 到尾位置*/
       if (p==end)loop=0 /*已为尾则退出
    */
       p=end;bno=bmax;break;
      case INS:ins=! ins;break;    /*Ins 键切换
    */
      case LEFT:    /*←键左移*/
       if(p==first){loop=0;break;}
       if(p>(b+bno)->f){p--;
         if (t[p]>0xa0)p--;    /*
         汉字要左移 2 格*/
      else {___ bno--; p=(b+bno)->e;
       if (t[p]>-0xa0)p--;    /*
        移到前半个汉字处*/
      }
      break;
       case RIGHT:    /*→键右移*/
       if(p==end){loop=0;break;}
       if(p<(b+bno)-->e){p++;
         if(t[(p-1)>0xa0)p++;}
        /*汉字要右移 2 格*/
      else { bno++;p=(b+bno)->f;}
```

```
    break;
  case BACK:   /*退格键*/
  if (p==first) {    /*已在首位置时退出编辑
*/
    loop=0;n=LEFT;break;}
  if (p>(b+bno)->f){ i=1;
    if (t[p-1]0xa0)i++; /*汉字要退2格
*/
    memmove (&t[p-1],&t[p],(b+bno)->
e-p+1);
    memset (&t[(b+bno)->e-i+1],'',i);
    p-=i; /*整体前移并调整编辑位置*/
    gotoxy(c+p,r);   /*重显更新部分*/
    for (i=p;i<=(b+bno)->e;
i++)putch(t[i]);
  }
  else {bno--; p=(b+bno)->e;
    if(t[p]>0xa0)p--;}/*移到前半汉字处
*/
    break;
  case DEL:   /*删除光标处字符*/
      if (p==(b+bno)->e){t[p]='';putch
('');}
      else { i=1;
    if {t[p]>0xa0)i++   /*汉字要删2格*/
    memmove (&t[p],&t[p+i],(b+bno)->e
-p-i+1);
    memset(&t[(b+bno)->e-i+1],'',);
    for (i=p;i<=(b+bno)->e;     i++)
putch(t[i]);
  }
  break;
  case UP:   /*上移键↑*/
  case DOWN:   /*下移键↓*/
  case ESC:   /*Esc放弃编辑内容*/
  case ENTER: loop=0;break;   /*回车结束
*/
  default:
    if(n>255)cont inue;   /*不睬无效键*/
    u=(unsigned char )n;
    if (! f&&! strchr (set,u)) continue;
    else if (f) switch (set[p]) {
      case'A': if (! isalpha (u)) continue;
        break;
      case 'N': if (! isalnum(u)) continue;
        break;
      case'9': if (! isdigit(u)) continue;
        break; }
  i=1;
  if(u>0xa0){   /*是汉字时*/
    if (p==(b+bno)->e||
(f&&set[p+1]! ='X'))
    continue; /*最尾1位无法输*/
  cc=(unsigned char) inkey();
  i++;   /*继续接收后半个*/
  if ((! ins)&&t[p]<0xa0&&t[p+1]>0xa0){
      memmove (&t[p+2],&t[p+1],(b+
bno)->e-p-1);
      chan=1;}
  }  /*在汉字处输入字符要删除后半汉字*/
    else if ((! ins)&&t[p]>0xa0){
      memcpy(&t[p+1],&t[p+2],(b+bno)
    ->e-p-1);
  t[(b+bno)->e]='';chan=1;}
  if (ins)  /*插入时后部后移*/
  memmove (&t[p+1],&t[p],(b+bno)->e
-P-i+1);
  t[p]=u;   /*存放字符*/
  if (i>1)|t[p+1]=cc;/*保存后半个汉字/
  j=0;  /*统计汉字占位数以消除半个汉字
*/
  for (k=p;k<=(b+bno)->e;k++)
    if (t[k]>0xa0)j++;
   if (j%2) t[(b+bno)->e]='',
/*清尾半个*/
  if (ins||chan) k=(b+bno)->e;
  else k=p+i-1;
  gotoxy(c+p,r);   /*重显*/
  for (j=p;j<=k;j++)putch(t[j]);
  p+=i;   /*改变当前编辑位置*/
  if (p>end) {loop=0;c=ENTER;}
    /*编辑完*/
  else if (p>(b+bno)->e){
      bno++; p=(b+bno)->f;}
  break;
  }
  } while (loop);
  gotoxy(c,r);
```

```
    if (n! =ESC) memcpy(s,t,l);    /* 认可编辑
  */
  else for (i=0;i<1;i++) putch(s[i]);
    free(t); free(set);    /* 释放暂存区 */
    return n;    /* 返回中断编辑的键值 */
  }
```

1.2 可编辑的输入函数

付 辉

笔者在用 TURBO C 2.0 编程时，深感C 所提供的标准输入函数的不完善。表现在无编辑能力、控制能力弱、界面差。为此笔者编制了输入函数：int get (int x, int y, int attr, int len, char * old str)，较好地解决了这一问题。使用函数 get()，能按指定的属性和字段长度进行输入。具有仿全屏幕编辑功能，越界报警。能够在汉字系统下随意录入、删除、插入汉字。具有消除“半个汉字”的功能，键入回车键确认所得、键入 ESC 键保留默认字串。函数的返回值为中断编辑的键值。

例如：

```
...
char * str="可编辑的输入函数实现";
...
get (10, 10, BLUE *16+RED, 30,str);
...
```

运行时，以指定属性(蓝底红色)在(10,10)处，显示字串“可编辑的输入函数实现”，长度30，不足处以空格补足。可运用全屏幕编辑功能对其操作。ENTER 确认，ESC 取消操作。

此函数在浪潮 286－12 上运行通过，有兴趣的同行不妨一试。

程序清单：

```
#include<conio.h>
#include<mem.h>
#include<alloc.h>
#define HOME 327
#define END 335
#define LEFT 331
#define RIGHT 333
#define INS 338
#define DEL 339
#define ESC 27
#define ENTER 13
#define BACK 8
int get (int x, int y, int attr, int len, char *
oldstr)/*oldstr 缺省串*/
{unsigned char * newstr;    /* 编辑串 */
    unsigned char cc, uu;
    int oldlen, loop=1, n, p=0, i, ins=1,j,k;/
*p 当前编辑位置 ins 插入标志 */
newstr=(unsigned char *) malloc (len+1);
oldlen=strlen(oldstr);
memcpy(newstr, oldstr, oldlen);
memset (newstr+oldlen, '' , len-oldlen);
*(newstr+len)=0;    /* 复制缺省串入编辑串
*/
textattr (attr);    /* 指定字串显示属性 */
gotoxy (x,y);    /* 指定显示位置 */
cprintf ("%s", newstr);
if (oldlen! =0 _ P=oldlen-1    /* 如果缺省为空
串，校验编辑位置 */
else p=0
```