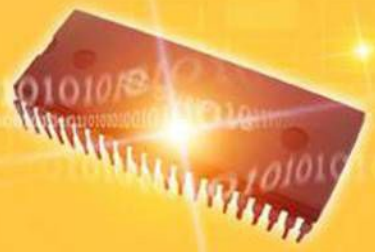
凌阳教育
www.sunplusedu.com
值得信赖的教育品牌

嵌入式系统工程师



Tel: 400-705-9680 , Email: edu@sunplusapp.com , BBS: bbs.sunplusedu.com

指针在字符串处理中的应用

Tel: 400-705-9680 , Email: edu@sunplusapp.com , BBS: bbs.sunplusedu.com

字符串处理函数

- 对字符串的处理，在嵌入式编程、应用编程、网络编程中会大量的遇到

  字符串拷贝、连接、比较、切割、变换...

- 要求熟练使用常见字符串处理函数，并会编写典型的字符串操作函数

字符串处理函数

1）字符串整体操作函数

- strlen　　//长度测量
- strcpy/strncpy　　//拷贝
- strcat/strncat　　//连接
- strcmp/strncmp　　//比较

- #include <string.h>

字符串处理函数

- strlen

原型：int strlen（const char *str）

功能：返回字符串的实际长度，不含’\0’

01.strlen.c

```
1  #include <stdio.h>
2  #include <string.h>
3  int main()
4  {
5      char str1[20]="hello";
6      char *str2 ="hello";
7      printf("%d\n",sizeof(str1));
8      printf("%d\n",sizeof(str2));
9      printf("%d\n",strlen(str1));
10     printf("%d\n",strlen(str2));
11     return 0;
12 }
```

第二个参数用const修饰，表示“只读”

字符串处理函数

- **strcpy**

  原型:char *strcpy(char *dest, const char *src)

  功能: 把src所指向的字符串复制到dest所指向的空间中

  返回值: 返回dest字符串的首地址

  注意: '\0'也会拷贝过去

- **strncpy**

  原型:char *strncpy(char *dest, const char *src,int num)

  功能: 把src指向字符串的前num个复制到dest所指向的空间中

  返回值: 返回dest字符串的首地址

  注意: '\0'不拷贝

字符串处理函数

02.strcpy_strncpy.c

```
1  #include <stdio.h>
2  #include <string.h>
3  int main()
4  {
5      char dest1[20]="123456789";
6      char dest2[20]="123456789";
7      char *src ="hello world";
8
9      strcpy(dest1,src);
10     printf("%s\n",dest1);
11
12     strncpy(dest2,src,5);
13     printf("%s\n",dest2);
14
15     dest2[5]='\0';
16     printf("%s\n",dest2);
17     return 0;
18 }
```

字符串处理函数

- **strcat**

  原型：char *strcat(char *str1,char *str2)

  功能：将str2连接到str1后面

  返回值：返回str1字符串的首地址

  注意：'\0'会一起拷贝过去

- **strncat**

  原型：char *strncat(char *str1,char *str2,int num)

  功能：将str2前num个字母连接到str1后面

  返回值：返回str1字符串的首地址

  注意：'\0'会一起拷贝过去

字符串处理函数

➢ 03.strcat_strncat.c

```
1  #include <stdio.h>
2  #include <string.h>
3  int main()
4  {
5      char str1[20]="123";
6      char str2[20]="123";
7
8      char *src ="hello world";
9      printf("%s\n",strcat(str1,src));
10     printf("%s\n",strncat(str2,src,5));
11     return 0;
12 }
```

字符串处理函数

- **strcmp**

  原型：int strcmp(char *str1, char *str2)

  功能：比较str1和str2的大小;

  返回值：相等返回0;

  str1大于str2返回>0

  str1小于str2返回<0

- **strncmp**

  原型：int strncmp(char *str1, char *str2, int num)

  功能：比较str1和str2的前num个字符串的大小;

  返回值：相等返回0;

  str1前num个大于str2返回>0

  str1前num个小于str2返回<0

字符串处理函数

04.strcmp_strncmp.c

```c
#include <stdio.h>
#include <string.h>
int main()
{
    char *str1 = "hello world";
    char *str2 = "hello kitty";

    if( strcmp(str1,str2) == 0)
        printf("str1==str2\n");
    else if(strcmp(str1,str2) > 0)
        printf("str1>str2\n");
    else
        printf("str1<str2\n");

    if( strncmp(str1,str2,5) == 0)
        printf("str1==str2\n");
    else if(strcmp(str1,"hello world") > 0)
        printf("str1>str2\n");
    else
        printf("str1<str2\n");

    return 0;
}
```

字符串处理函数

2）字符串变换函数

```
strchr              //字符匹配函数
strstr              //字符串匹配
memset              //空间设定函数
atoi/atol/atof      //字符串转换功能
strtok              //字符串分割函数
```

字符串处理函数

- **strchr**

  原型: char* strchr(const char *str1, char ch)

  功能: 在字符串str1中查找字母ch出现的位置

  返回值: 返回第一次出现的ch地址

  如果找不到，返回NULL

- **strstr**

  原型: char* strchr(const char *str1, char* str2)

  功能: 在字符串str1中查找字符串str2出现的位置

  返回值: 返回第一次出现的str2地址

  如果找不到，返回NULL

字符串处理函数

05.strchr_strstr.c

```
1  #include <string.h>
2  #include <stdio.h>
3  int main()
4  {
5      char str1[20]="hello $#$ world";
6      char ch='$';
7      char str2[20]="$#$";
8      char *result;
9
10     result=strchr(str1,ch);
11     printf("%s\n",result);
12     printf("%d\n",result-str1);
13
14     result=strstr(str1,str2);
15     printf("%s\n",result);
16     printf("%d\n",result-str1);
17     return 0;
18 }
```

字符串处理函数

- **memset**

原型：void* memset(void *str,char c,int n)

功能：将str所指向的内存区的前n个全部用c填充

常用于清除指定空间，比如数组或malloc的空间

返回值：返回str的地址

字符串处理函数

06.memset.c

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  int main()
5  {
6      char str1[20]="hello $#$ world";
7      char *str2=NULL;
8      memset(str1,'$',strlen(str1));
9      printf("str1=%s\n",str1);
10     str2=(char *)malloc(10*sizeof(char));
11     memset(str2,0,10*sizeof(char));
12     free(str2);
13     return 0;
14 }
```

字符串处理函数

- **atoi/atol/atof** //字符串转换功能

```
int atoi(const char*str);
long atol(const char*str);
double atof(const char*str);
```

07.atoi.c

功能：将str所指向的**数字字符串**转化为int\long\double

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      char str1[]="12345",str2[]="56789",str3[]="123.456";
6      int num1;
7      long num2;
8      double num3;
9      num1=atoi(str1);
10     num2=atol(str2);
11     num3=atof(str3);
12     printf("num1=%d,num2=%d,num3=%lf\n",num1,num2,num3);
13     return 0;
14 }
```

字符串处理函数

- strtok 字符串切割函数

  char *strtok(char s[], const char *delim);

  功能：strtok()用来将字符串分割成一个个片段。

  参数1：s指向欲分割的字符串

  参数2：delim则为分割字符串中包含的所有字符。

  当strtok()在参数s的字符串中发现参数delim中包含的分割字符时,则会将该字符改为\0 字符，当连续出现多个时只替换第一个为\0。

  在第一次调用时：strtok()必需给予参数s字符串

  往后的调用则将参数s设置成NULL，每次调用成功则返回指向被分割出片段的指针

字符串处理函数

➢ 例：strtok

08.strtok.c

```
1  #include <stdio.h>
2  #include <string.h>
3  int main()
4  {
5      char str1[]="adc*fvcv*!ebcy!hghbdfg*casdert";
6      char *str2="*!";
7      char *result[10];
8      int num=0,i;
9      result[num]=strtok(str1,str2);
10     while(result[num]!=NULL)
11     {
12         num++;
13         result[num]=strtok(NULL,str2);
14     }
15     for(i=0;i<num;i++)
16         printf("result[%d]=%s\n",i,result[i]);
17     return 0;
18 }
```

字符串处理函数

3）格式化字符串操作函数

- int sprintf(char *buf, const char *format, ...);

  \\输出到buf指定的内存区域。

例:

```
char buf[20];
sprintf(buf,"%d:%d:%d",2013,10,1);
```

- int sscanf(const char *buf,const char *format, ...);

  \\从buf指定的内存区域中读入信息

例: int a, b, c;

```
sscanf("2013:10:1", "%d:%d:%d", &a, &b, &c);
```

字符串处理函数

09.sscanf、sprintf.c

```c
#include <stdio.h>
#include <string.h>
int main()
{
    char buf[20];
    int a, b, c;

    sprintf(buf,"%d:%d:%d",2013,10,1);
    printf("buf=%s\n",buf);

    sscanf("2013:10:1", "%d:%d:%d", &a, &b, &c);
    printf("a=%d, b=%d, c=%d\n",a,b,c);
    return 0;
}
```

标准I/O库函数

- sscanf高级用法

① 跳过数据：%*s或%*d

例：sscanf("1234 5678", "%*d %s", buf);

② 读指定宽度的数据：%[width]s

例：sscanf("12345678", "%4s", buf);

③ 支持集合操作：

%[a-z] 表示匹配a到z中任意字符(尽可能多的匹配)

%[aBc] 匹配a、B、c中一员，贪婪性

%[^a] 匹配非a的任意字符，贪婪性

%[^a-z]表示读取除a-z以外的所有字符

标准I/O库函数

1、取仅包含指定字符集的字符串

```
sscanf("123abcABC", "%[1-9a-z]", buf);
printf("%s\n",buf);
```

结果为：123abc

2、取到指定字符集为止的字符串

```
sscanf("123abcABC", "%[^A-Z]", buf);
printf("%s\n", buf);
```

结果为：123abc

3、取到指定字符为止的字符串

```
sscanf("123456 abcdedf", "%[^ ]", buf);
printf("%s\n", buf);
```

结果为：123456

标准I/O库函数

10.sscanf.c

```
#include <stdio.h>
#include <string.h>
int main()
{
    char buf[20];
    sscanf("1234 5678", "%*d %s", buf);
    printf("buf1=%s\n",buf);
    sscanf("12345678", "%4s", buf);
    printf("buf2=%s\n",buf);
    sscanf("123abcABC", "%[1-9a-z]", buf);
    printf("buf3=%s\n",buf);
    sscanf("123abcABC", "%[^A-Z]", buf);
    printf("buf4=%s\n",buf);
    sscanf("123456 abcdedf", "%[^ ]", buf);
    printf("buf5=%s\n",buf);
    return 0;
}
```

凌阳教育
www.sunplusedu.com

值得信赖的教育品牌

Tel: 400-705-9680 , Email: edu@sunplusapp.com , BBS: bbs.sunplusedu.com