C语言字符串处理

函数名：strcpy
功　能：将参数src字符串拷贝至参数dest所指的地址
用　法：char *strcpy(char *dest, const char *src);
返回值：返回参数dest的字符串起始地址
说　明：如果参数dest所指的内存空间不够大，可能会造成缓冲溢出的错误情况，在编写程序时需特别留意，或者用strncpy()来取代；
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);  // 输出：abcdefghi
    return 0;
 }
```

函数名：strncpy
功　能：将字符串src前n个字符拷贝到字符串dest
用　法：char *strncpy(char *dest, const char *src, size_t n);
返回值：返回参数dest的字符串起始地址
说　明：不像strcpy()，strncpy()不会向dest追加结束标记'\0'；
　　src和dest所指的内存区域不能重叠，且dest必须有足够的空间放置n个字符；
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);  // 输出：abc
   return 0;
}
```

函数名：strcat
功　能：字符串拼接函数
用　法：char *strcat(char *dest, const char *src);
返回值：返回dest字符串起始地址
说　明：strcat() 会将参数src字符串复制到参数dest所指的字符串尾部；
　　dest最后的结束字符'\0'会被覆盖掉，并在连接后的字符串的尾部再增加一个'\0'；
　　dest与src所指的内存空间不能重叠，且dest要有足够的空间来容纳要复制的字符串；
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);  // 输出：Borland C++
   return 0;
}
```

函数名：strncat
功　能：将n个字符追加到字符串的结尾
用　法：char *strncat(char *dest, const char *src, size_t n);
返回值：返回dest字符串起始地址
说　明：strncat()将会从字符串src的开头拷贝n个字符到dest字符串尾部，dest要有足够的空间来容纳要拷贝的字符串；
　　如果n大于字符串src的长度，那么仅将src全部追加到dest的尾部；
　　strncat()会将dest字符串最后的'\0'覆盖掉，字符追加完成后，再追加'\0'；
程序例：

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

int main()
{
    char url[100] = "http://blog.csdn.net";
    char path[30] = "/cpp/u/string/";
    strncat(url, path, 1000);  // 1000远远超过path的长度
    printf("%s\n", url);  // 输出：http://blog.csdn.net/cpp/u/string/
    return  0;
}
```

函数名：strchr
功　能：在一个字符串中查找给定字符的第一个匹配之处
用　法：char *strchr(const char *str, int c);
返回值：如果找到指定的字符则返回该字符所在地址，否则返回NULL
说　明：返回的地址是字符串在内存中随机分配的地址再加上你所搜索的字符在字符串的位置；
　　字符串str的结束标志‘\0’也会被纳入检索范围，所以str的最后一个字符也可以被定位；
　　如果希望查找某字符在字符串中最后一次出现的位置，可以使用 strrchr() 函数；
程序例：

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

int main()
{
    char *s = "01234567890123456789012345678890";
    char *p;
    p = strchr(s, '5');
    printf("%ld\n", s);  // 输出：134513940
    printf("%ld\n", p);  // 输出：134513945
    p = strrchr(s, '5');
    printf("%ld\n", p);  // 输出：134513965
    return 0;
}
```

函数名：strcmp
功　能：字符串比较
用　法：int strcmp(const char *s1, const char *s2);
返回值：根据ASCII码比较，若参数s1和s2字符串相同则返回0，s1若大于s2则返回大于0的值，s1若小于s2则返回小于0的值
说　明：它是区分大小写比较的，如果希望不区分大小写进行字符串比较，可以使用stricmp函数
程序例：

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

int main(void)
 {
    char *a = "aBcDeF";
    char *b = "AbCdEf";
    char *c = "aacdef";
    char *d = "aBcDeF";
    printf("strcmp(a, b) : %d\n", strcmp(a, b));  // 输出：1
    printf("strcmp(a, c) : %d\n", strcmp(a, c));  // 输出：-1
    printf("strcmp(a, d) : %d\n", strcmp(a, d));  // 输出：0
    return 0;
 }
```

函数名：strlen
功　能：计算指定的字符串s的长度，不包括结束字符'\0'
用　法：size_t strlen(const char *s);
返回值：返回字符串s的字符数
说　明：strlen() 函数计算的是字符串的实际长度，遇到第一个'\0'结束；
　　如果你只定义没有给它赋初值，这个结果是不定的，它会从首地址一直找下去，直到遇到'\0'停止；
　　sizeof返回的是变量声明后所占的内存数，不是实际长度，此外sizeof不是函数，仅仅是一个操作符，strlen()是函数；
程序例：

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

int main()
{
    char str[5] = "abcd";
    printf("strlen(str)=%d, sizeof(str)=%d\n", strlen(str), sizeof(str));  // 输出：strlen(str)=4, sizeof(str)=5
    return 0;
}
```

函数名：strtok
功　能：根据分界符将字符串分割成一个个片段
用　法：char *strtok(char *s, const char *delim);
返回值：返回下一个分割后的字符串指针，如果已无从分割则返回NULL
说　明：当strtok()在参数s的字符串中发现到参数delim的分割字符时则会将该字符改为'\0'字符；
　　在第一次调用时，strtok()必须赋予参数s字符串，往后的调用则将参数s设置成NULL；
程序例：

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

int main()
{
    char s[] = "ab-cd : ef";
    char *delim = "-: ";
    char *p;
    printf("%s \n", strtok(s, delim));
    // 输出：ab
    //       cd
    //       ef
    while((p = strtok(NULL, delim)))
        printf("%s ", p);
        printf("\n");
}
```

函数名: strstr
功　能: 检索子串在字符串中首次出现的位置
用　法: char *strstr(char *str, char * substr);
返回值: 返回字符串str中第一次出现子串substr的地址；如果没有检索到子串，则返回NULL
程序例:

```
#include<stdio.h>
#include<string.h>
int main(){
    char *str = "HelloWorldHelloWorld";
    char *substr = "World";
    char *s = strstr(str, substr);
    printf("%s\n", s);  // 输出：WorldHelloWorld
    return 0;
}
```

函数名: strpbrk
功　能: 返回两个字符串中首个相同字符的位置
用　法: char *strpbrk(char *s1, char *s2);
返回值: 如果s1、s2含有相同的字符，那么返回指向s1中第一个相同字符的指针，否则返回NULL
说　明: strpbrk()不会对结束符'\0'进行检索
程序例:

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

int main(){
    char* s1 = "see you again";
    char* s2 = "you";
    char* p = strpbrk(s1,s2);
    if(p){
        printf("The result is: %s\n",p);  // 输出：The result is: you again
    }else{
        printf("Sorry!\n");
    }
    return 0;
}
```

函数名：atoi
功　能：将字符串转换成整数(int)
用　法：int atoi (const char * str);
返回值：返回转换后的整型数；如果str不能转换成int或者str为空字符串，那么将返回0
说　明：ANSI C规范定义了 stof()、atoi()、atol()、strtod()、strtol()、strtoul() 共6个可以将字符串转换为数字的函数，可以对比学习；
　　另外在C99/C++11规范中又新增了5个函数，分别是 atoll()、strtof()、strtold()、strtoll()、strtoull()；
程序例：

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

int main ()
{
    int i;
    char buffer[256];
    printf ("Enter a number: ");
    fgets (buffer, 256, stdin);
    i = atoi (buffer);
    printf ("The value entered is %d.", i);
    return 0;
}
```