

3 años Taringa! > Info > [Programación C/C++] Funcio...

Me gusta 548 024

Seguir

[Programación C/C++] Funciones para cadenas o cstring

Cadenas char, también conocidas como "arreglos de char", "cadenas", o "cstring". Detesto trabajar con ellas y supongo que muchos de ustedes también 😊.

Para hacernos la vida un poco más fácil les traigo este post para que les sirva como referencia, donde voy a listar y explicar las funciones que vienen incluidas en la biblioteca string.h (que viene de C) que nos van a simplificar bastante el trabajo a la hora de trabajar con cadenas de texto.

La primera parte del post es una lista simple con las funciones, y en la segunda parte se explican a fondo.

La parte de las referencias es algo extensa, así que les recomiendo ubicar la función que necesitan en la lista simple, apretar Ctrl+F para abrir el panel de búsqueda de su navegador y buscar la referencia fácilmente a través del nombre de la función.

TARINGA!

LISTA DE FUNCIONES:

DE COPIA:

strcpy()

Copiar una cadena y pegarla sobre otra.

strncpy()

Copia los primeros "n" caracteres de una cadena a otra.

DE CONCATENACIÓN:

strcat()

Concatena dos cadenas.

strncat()

Concatena los primeros "n" caracteres de una cadena al final de otra.

DE COMPARACIÓN:

strcmp()

Compara si dos cadenas son iguales.

strncmp()

Compara si los primeros "n" caracteres de una cadena son iguales a los primeros "n" caracteres de otra.

DE BÚSQUEDA:**strchr()**

Busca en la cadena la primera ocurrencia de un caracter (char) indicado.

strrchr()

Busca en la cadena la última ocurrencia de un caracter (char) indicado.

strcspn()

Busca en una cadena la primera ocurrencia de cualquiera de los caracteres de otra cadena y retorna la posición de la misma.

strpbrk()

Busca en una cadena la primera ocurrencia de cualquiera de los caracteres de otra cadena y retorna un puntero a char con la subcadena formada desde esa posición.

strspn()

Retorna el largo de la porción inicial de una cadena que está formada solamente por los caracteres presentes en otra.

strstr()

Busca una cadena dentro de otra cadena, retorna un puntero a la subcadena.

strtok()

Divide una cadena en segmentos (o tokens) delimitados por los caracteres char que indiquemos.

OTROS:**strerror()**

Retorna una cadena que describe un error producido en el programa (pura magia).

strlen()

Retorna el largo de una cadena.

**DETALLES DE CADA FUNCIÓN:**

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

Devuelve el largo de "cadena" (el tipo *size_t* es como un *unsigned int*). El largo de la cadena es determinado por el caracter nulo o de fin de cadena (barra invertida cero): una cadena o cstring es tan largo como caracteres haya antes del caracter de fin de cadena (sin incluirlo).



Esto no debe confundirse con el tamaño de la cadena como arreglo. Por ejemplo:

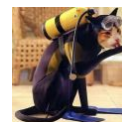
Ir a comentarios

```
char micadena[100] = "probando cadena";
```



Compartir
via mail

define un arreglo de char con un tamaño de 100 chars, pero la cadena con la cual micadena ha sido inicializada tiene un tamaño de 15 caracteres. Por lo



elwear
REGULAR

38
SEGUIDORES

5
POSTS

643
PUNTOS

Tanto, mientras que sizeof(micadena) devuelve 100, strlen(micadena) retorna 15.

Valor de retorno: el largo de la cadena.

Ejemplos de uso:

TI

2

```
/* ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    char entrada[256];
    printf("Introduzca una frase: ");
    gets( entrada );
    printf( "La frase ingresada tiene %d caracteres de largo\n", strlen( entrada ) );
    return 0;
}
```

```
// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
using namespace std;

int main ()
{
    char entrada[256];
    cout<<"Introduzca una frase: ";

    // El método getline() de cin sirve para leer cadenas
    // sus argumentos son: el arreglo de char donde almacenar la cadena
    // el tamaño de dicho arreglo (obtenido con sizeof).
    cin.getline( entrada, sizeof( entrada ) );

    cout<<"La frase ingresada tiene "
    <<strlen( entrada )<<" caracteres de largo.\n";
    return 0;
}
```

Salida:

Introduzca una frase: Hola mundo!
La frase ingresada tiene 11 caracteres de largo.

TARINGA!



A favoritos

`char * strcpy (char * destino, const char * fuente);`

Copia la cadena apuntada por "fuente" al arreglo apuntado por "destino",

insertando por su cuenta el caracter de fin de cadena (barra invertida cero).

Hay que destacar que para que el programa no explote al realizar esta

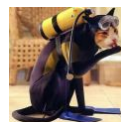
operación, el arreglo "destino" tiene que tener un tamaño de por lo menos la

cantidad de caracteres que haya en "fuente" + 1 (espacio para el caracter de

fin de cadena) o dicho de otra forma: tamaño mínimo de destino debe ser >=

strlen(fuente) + 1.

Compartir
via mail



elwear

REGULAR

38

SEGUIDORES

5

POSTS

643

PUNTOS

T Valor de retorno: puntero a "destino".

11

Ejemplos de uso:



T!

2

```
/* strcpy ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    char cad1[] = "cstring de ejemplo";
    char cad2[40];
    char cad3[40];
    strcpy( cad2, cad1 );
    strcpy( cad3, "copia exitosa" );
    printf( "cad1: %s\ncad2: %s\ncad3: %s\n", cad1, cad2,
    return 0;
}
```

```
// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
using namespace std;

int main ()
{
    char cad1[] = "cstring de ejemplo";
    char cad2[40];
    char cad3[40];
    strcpy( cad2, cad1 );
    strcpy( cad3, "copia exitosa" );
    cout<<"cad1: "<<cad1<<"\ncad2: "
<<cad2<<"\ncad3: "<<cad3;
    return 0;
}
```

Salida:

```
cad1: cstring de ejemplo
cad2: cstring de ejemplo
cad3: copia exitosa
```

TARINGA!

char * strncpy (char * destino, const char * fuente, siz



A favoritos

Copia los primeros "cuantos" caracteres de "fuente" a "destino". Si el final de la cadena "fuente" (indicado por el caracter de fin de cadena 'barra invertida cero') es alcanzado antes de que se copien la cantidad de caracteres indicados por "cuantos", el arreglo "destino" se llena con ceros hasta que una cantidad de "cuantos" caracteres hayan sido escritos en él.

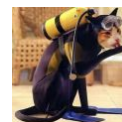


Ir a comentarios

Valor de retorno: puntero a "destino".



Compartir via mail



elwear

REGULAR

38
SEGUIDORES5
POSTS643
PUNTOS

T

11



T!

2

```

/* ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    char cad1[] = "Ser o no ser";
    char cad2[40];
    char cad3[40];

    /* Copiar la cantidad máxima permitida por cad2 */
    strncpy( cad2, cad1, 40 );

    /* copia parcial (solamente 5 chars): */
    strncpy( cad3, cad2, 5 );
    cad3[5] = '\0'; /* caracter de fin de cadena agregar

    puts( cad1 );
    puts( cad2 );
    puts( cad3 );

    return 0;
}

```

```

// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
using namespace std;

int main ()
{
    const int tamanyo = 40;
    char cad1[] = "Ser o no ser";
    char cad2[tamanyo];
    char cad3[tamanyo];

    /* Copiar la cantidad máxima permitida por cad2 */
    strncpy( cad2, cad1, tamanyo );

    /* copia parcial (solamente 5 chars): */
    strncpy( cad3, cad2, 5 );
    cad3[5] = '\0'; /* caracter de fin de cadena agregar

    cout<<cad1<<endl;
    cout<<cad2<<endl;
    cout<<cad3<<endl;

    return 0;
}

```



A favoritos

Salida:

Ir a
comentariosCompartir
via mail

Ser o no ser

Ser o no ser

Ser o

elwear
REGULAR38
SEGUIDORES5
POSTS643
PUNTOS

T

TARINGA!

11



```
char * strcat ( char * destino, const char * fuente );
```

T!

2 Pega una copia de la cadena "fuente" al final de la cadena "destino". El 'pegado' se produce desde la posición del caracter nulo o de fin de cadena (barra invertida cero) en "destino", sobrescribiéndolo con el primer caracter de "fuente" y continuando. Un nuevo caracter de fin de cadena es insertado al final del proceso. Como con strcpy(), hay que resaltar que **el arreglo "destino" tiene que tener un tamaño suficiente para anexar su contenido a la cadena "fuente"**, es decir:

tamaño de destino >= strlen(destino) + strlen(fuente) + 1.

Valor de retorno: un puntero a "destino".

Ejemplos de uso:

```
/* ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    char cad[80];
    char cad2[] = "estan ";

    strcpy( cad, "estas " );

    strcat( cad, "cadenas " );
    strcat( cad, cad2 );
    strcat( cad, "concatenadas." );

    puts( cad );

    return 0;
}
```

```
// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
using namespace std;

int main ()
{
    char cad[80];
    char cad2[] = "estan ";

    strcpy( cad, "estas " );

    strcat( cad, "cadenas " );
    strcat( cad, cad2 );
    strcat( cad, "concatenadas." );

    cout<<cad;

    return 0;
}
```



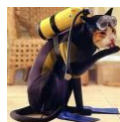
A favoritos



Ir a comentarios



Compartir via mail



elwear

REGULAR

38
SEGUIDORES5
POSTS643
PUNTOS

T

11 Salida:

*estas cadenas estan concatenadas.*

T!

TARINGA!

2

```
char * strncat ( char * destino, const char * fuente, siz
```

Pega los primeros "cuantos" caracteres de fuente al final de "destino", agregando solo el caracter de fin de cadena. Si el tamaño de la cadena "fuente" es menor a la cantidad de caracteres "cuantos", se copiará y pegará solamente los caracteres que haya hasta llegar al caracter de fin de cadena de "fuente". Se deben tener las mismas precauciones de tamaños de arreglos que las indicadas para strcat().

Valor de retorno: puntero a "destino".

Ejemplos de uso:

```
/* ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    char cad1[20];
    char cad2[20];
    strcpy( cad1, "Ser " );
    strcpy( cad2, "o no ser" );
    strncat( cad1, cad2, 4 );
    puts( cad1 );

    return 0;
}
```

```
// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
using namespace std;

int main ()
{
    char cad1[20];
    char cad2[20];
    strcpy( cad1, "Ser " );
    strcpy( cad2, "o no ser" );
    strncat( cad1, cad2, 4 );
    cout<<cad1;

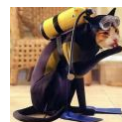
    return 0;
}
```



A favoritos

Ir a
comentariosCompartir
via mail

Salida:

Ser o noelwear
REGULAR38
SEGUIDORES5
POSTS643
PUNTOS

T

11



TARINGA!

```
int strcmp ( const char * cad1, const char * cad2 );
```

T!

² Compara la cadena "cad1" con la cadena "cad2".

Esta función comienza comparando el primer caracter de cada cadena. Si estos caracteres son iguales, continua con el siguiente par hasta que los caracteres comparados son distintos o se alcanza un caracter nulo o de fin de cadena.

Valor de retorno: retorna un valor int que indica la relación entre las dos cadenas:

- Un valor 0 (cero) indica que las dos cadenas son iguales.
- Un valor mayor a cero indica que el primer caracter encontrado que es distinto tiene un valor mayor en cad1 que en cad2.
- Un valor menor a cero indica lo contrario a lo anterior.

Ejemplos de uso:

```
/* ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    char clave[] = "manzana\n";
    char entradaClave[80];
    do {
        printf( "Cual es mi fruta preferida? " );
        fgets( entradaClave, 80, stdin );
    } while ( strcmp( clave, entradaClave ) != 0 );
    puts( "Respuesta correcta!" );
    return 0;
}
```

```
// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
using namespace std;

int main ()
{
    char clave[] = "manzana";
    char entradaClave[80];
    do {
        cout<<"Cual es mi fruta preferida? ";
        cin.getline( entradaClave, 80 );
    } while ( strcmp( clave, entradaClave ) != 0 );
    cout<<"Respuesta correcta!";
    return 0;
}
```



A favoritos



Ir a comentarios

Compartir
via mail

elwear
REGULAR

38
SEGUIDORES

5
POSTS

643
PUNTOS

T

11

*Cual es mi fruta preferida? pera**Cual es mi fruta preferida? manzana**Respuesta correcta!*

TARINGA!

T!

2

```
int strcmp ( const char * cad1, const char * cad2, size_
```

Compara los primeros "cuantos" caracteres de la cadena "cad1" con los primeros "cuantos" caracteres de la cadena "cad2".

Esta función comienza comparando el primer caracter de cada cadena. Si estos caracteres son iguales, continua con el siguiente par hasta que los caracteres comparados son distintos, o se alcanza un caracter nulo o de fin de cadena, o hasta que "cuantos" caracteres ah sido comparados y encontrados iguales, lo que pase primero.

Valor de retorno: retorna un valor int que indica la relación entre las dos cadenas:

- Un valor 0 (cero) indica que las dos cadenas son iguales.
- Un valor mayor a cero indica que el primer caracter encontrado que es distinto tiene un valor mayor en cad1 que en cad2.
- Un valor menor a cero indica lo contrario a lo anterior.

Ejemplos de uso:

```
/* ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    char str[][5] = { "R2D2" , "C3P0" , "R2A6" };
    int n;
    puts ("Buscando astromech droids R2...");
    for( n=0 ; n<3; n++ ) {
        if ( strcmp ( str[n], "R2xx", 2) == 0 )
        {
            printf ("Encontrado: %s\n", str[n] );
        }
    }
    return 0;
}
```

```
// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
using namespace std;

int main ()
{
    char str[][5] = { "R2D2" , "C3P0" , "R2A6" };
    cout<<"Buscando astromech droids R2..."<<endl;
    for( int n=0 ; n<3; n++ ) {
        if ( strcmp ( str[n], "R2xx", 2) == 0 )
        {
            cout<<"Encontrado: "<<str[n]<<endl;
        }
    }
}
```



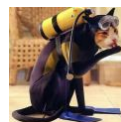
A favoritos



Ir a comentarios



Compartir via mail

**elwear**

REGULAR

38

SEGUIDORES

5

POSTS

643

PUNTOS

```

T
11
    }
    }
    return 0;
}

```

T! Salida:

```

2
    Buscando astromech droids R2...
    Encontrado: R2D2
    Encontrado: R2A6

```

TARINGA!

```

const char * strchr ( const char * cad, int letra ); /* s
char * strchr ( char * cad, int letra ); /* en C y

```

Nota: por más que el prototipo diga "int", "letra" es un char; lo que pasa es que todos los char en realidad son valores numéricos int que corresponden al número del código ascii cuyo valor representan.

La función empieza a buscar el caracter "letra" en la cadena "cad". Si lo encuentra, devuelve un puntero a la posición de la cadena donde lo encontró, es decir, devuelve un puntero a una subcadena dentro de la cadena que comienza con el caracter "letra". El caracter de fin de cadena también se considera parte de la cadena, por lo tanto, también puede ser buscado de esta forma para obtener un puntero al final de la cadena.

Valor de retorno: Un puntero a la primera ocurrencia de "letra" en "cad". Si el caracter no es encontrado, devuelve NULL.

Ejemplos de uso:

```

/* ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    char str[] = "Esta es una cadena de ejemplo";
    char * pch;
    printf( "Buscando el caracter 's' en \"%s\"...\n", str );
    pch = strchr( str, 's' );
    while( pch != NULL )
    {
        printf( "Encontrado en la posicion %d\n", pch-
str+1 );
        printf( "El puntero retornado apunta a: \"%s\"\n", pch );
        /* busca desde después del último encontrado */
        pch = strchr( pch+1, 's' );
    }
    return 0;
}

```

★
A favoritos

💬
Ir a
comentarios

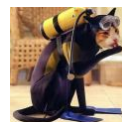
✉
Compartir
via mail

🔗

```

// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
using namespace std;

```



elwear
REGULAR

38
SEGUIDORES

5
POSTS

643
PUNTOS

T

11



T!

2

```

int main ()
{
    char str[] = "Esta es una cadena de ejemplo";
    char * pch;
    cout<<"Buscando el caracter 's' en \""
    <<str<<"...\n\n";
    pch = strchr( str, 's' );
    while( pch != NULL )
    {
        cout<<"Encontrado en la posicion "<<pch-
        str+1<<endl;
        cout<<"El puntero retornado apunta a: "
        <<pch<<endl;
        /* busca desde después del último encontrado */
        pch = strchr( pch+1, 's' );
    }
    return 0;
}

```

Salida:

*Buscando el caracter 's' en "Esta es una cadena de ejemplo"...**Encontrado en la posicion 2**El puntero retornado apunta a: sta es una cadena de ejemplo**Encontrado en la posicion 7**El puntero retornado apunta a: s una cadena de ejemplo*

TARINGA!

```

const char * strrchr ( const char * cad, int letra ); /*
char * strrchr ( char * str, int character ); /* e

```

Retorna un puntero a la última ocurrencia de "letra" en "cad". El caracter de fin de cadena es considerado parte de la cadena, por lo que puede ser buscado para obtener un puntero al final de la misma.

Valor de retorno: Un puntero a la última ocurrencia de "letra" en "cad". Si el caracter no es encontrado, devuelve NULL.

Ejemplos de uso:

```

/* ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    char str[] = "Esta es una cadena de prueba";
    char * pch;
    pch = strrchr( str, 's' );
    printf( "La ultima ocurrencia de 's' fue encontrada en
    str+1 );
    printf( "El puntero retornado apunta a: %s \n", pch );
    return 0;
}

```



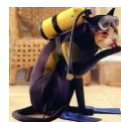
A favoritos



Ir a comentarios



Compartir via mail



elwear

REGULAR

38

SEGUIDORES

5

POSTS

643

PUNTOS

T

11



T!

2

```
// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
using namespace std;

int main ()
{
    char str[] = "Esta es una cadena de prueba";
    char * pch;
    pch = strrchr( str, 's' );
    cout<<"La ultima ocurrencia de 's' fue encontrada en "
    <<pch-str+1<<endl;
    cout<<"El puntero retornado apunta a: "<<pch;
    return 0;
}
```

Salida:

*La ultima ocurrencia de 's' fue encontrada en la posicion 7
El puntero retornado apunta a: s una cadena de prueba*

TARINGA!

```
size_t strcspn ( const char * cad1, const char * cad2 );
```

La función busca dentro de "cad1" la existencia de cualquiera de los caracteres contenidos en "cad2", retornando la cantidad de caracteres que hay en "cad1" antes de la primera ocurrencia de estos valores.

La búsqueda incluye el valor de fin de cadena. Por lo tanto, la función retornará el largo de "str1" si ninguno de los caracteres contenidos en "str2" fueron encontrados en "str1".

Valor de retorno: El largo de la parte inicial de "str1" que no contiene ninguno de los caracteres que componen a "str2".

Ejemplos de uso:

```
/* ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    char cad[] = "fcb73";
    char claves[] = "1234567890";
    int i;
    i = strcspn( cad, claves );
    printf( "El primer numero en cad esta en la posicion %d", i );
    return 0;
}
```

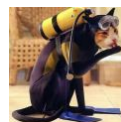
★
A favoritos

💬
Ir a
comentarios

✉
Compartir
via mail



```
// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
```



elwear
REGULAR

38
SEGUIDORES

5
POSTS

643
PUNTOS

```

T
11 using namespace std;

int main ()
{
    char cad[] = "fcba73";
    char claves[] = "1234567890";
    int i = strchr( cad, claves );
    cout<<"El primer numero en cad esta en la posicion "
    <<i+1;
    return 0;
}

```

Salida:

El primer numero en cad esta en la posicion 5

TARINGA!

```

const char * strpbrk ( const char * cad1, const char * ca
char * strpbrk ( char * cad1, const char * cad2 );

```

Retorna un puntero a la primera ocurrencia en "cad1" de cualquiera de los caracteres contenidos en "cad2", o un puntero a NULL si no hubo coincidencias.

La búsqueda no incluye al caracter de fin de cadena de ninguno de los dos cstrings, pero termina ahí.

Ejemplos de uso:

```

/* ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    char cadena[] = "Esto es un cstring de prueba";
    char claves[] = "aeiou";
    char * pch;
    printf( "Vocales en '%s': ", cadena );
    pch = strpbrk( cadena, claves );
    while( pch != NULL )
    {
        printf( "%c ", *pch );
        pch = strpbrk( pch+1, claves );
    }
    printf( "\n" );
    return 0;
}

```



A favoritos



Ir a
comentarios



Compartir
via mail



```

// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
using namespace std;

int main ()

```



elwear
REGULAR

38
SEGUIDORES

5
POSTS

643
PUNTOS

```

T
11 {
    char cadena[] = "Esto es un cstring de prueba";
    char claves[] = "aeiou";
    char * pch;
    cout<<"Vocales en '"<<cadena<<"' : ";
    pch = strpbrk( cadena, claves );
    while( pch != NULL )
    2 {
        cout<<*pch<<" ";
        pch = strpbrk( pch+1, claves );
    }
    return 0;
}

```

Salida:

Vocales en 'Esto es un cstring de prueba': o e u i e u e a

TARINGA!

```
size_t strspn ( const char * cad1, const char * cad2 );
```

Retorna la longitud de la porción inicial de "cad1" que consiste solamente de caracteres que son parte de "cad2".

La búsqueda no incluye el caracter de fin de cadena de ninguna de las dos cadenas, pero termina en ese punto.

Valor de retorno: la longitud de la porción inicial de "cad1" que consiste solamente de caracteres que son parte de "cad2".

Por lo tanto, si todos los caracteres de "cad1" están en "cad2" la función retorna el largo de "cad1", y si el primer caracter de "cad1" no está en "cad2" la función retorna cero.

Ejemplos de uso:

```

/* ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    int i;
    char cadena[] = "129asd";
    char numeros[] = "1234567890";

    i = strspn( cadena, numeros );
    printf( "El numero inicial de la cadena " ^ digi
    return 0;
}

```



A favoritos



Ir a
comentarios



Compartir
via mail



```

// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
using namespace std;

int main ()

```



elwear
REGULAR

38
SEGUIDORES

5
POSTS

643
PUNTOS

```

T
11 {
    char cadena[] = "129asd";
    char numeros[] = "1234567890";
    int i = strstrn( cadena, numeros );
    cout<<"El numero inicial de la cadena tiene "
    << i <<" digitos.";
    return 0;
2 }

```

Salida:

El numero inicial de la cadena tiene 3 digitos.

TARINGA!

```

const char * strstr ( const char * cad1, const char * cad2 );
char * strstr ( char * cad, const char * cad2 ); /*

```

Retorna un puntero a la primera ocurrencia de "cad2" (completa) en "cad1", o un puntero NULL si "cad2" no es parte de "cad1".

El proceso de búsqueda y comparación no incluye al caracter de fin de cadena, pero se detiene al llegar al mismo.

El siguiente ejemplo busca "simple" dentro de la cadena y lo reemplaza por "propia":

```

/* ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    char str[] ="Esta es una cadena simple";
    char * pch;
    pch = strstr( str, "simple" );
    strncpy( pch, "propia", 6 );
    puts( str );
    return 0;
}

```

```

// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
using namespace std;

```

```

int main ()
{
    char str[] ="Esta es una cadena simple";
    char * pch;
    pch = strstr( str, "simple" );
    strncpy( pch, "propia", 6 );
    cout<<str;
    return 0;
}

```

★
A favoritos

💬
Ir a comentarios

✉
Compartir via mail



elwear
REGULAR

38
SEGUIDORES

5
POSTS

643
PUNTOS

T

11 Salida:

*Esta es una cadena propia*

T!

2

```
char * strtok ( char * cad, const char * delimitantes );
```

Divide una cadena en segmentos (o tokens).

Una secuencia de llamadas a esta función divide "cad" en tokens, que son secuencias de caracteres contiguos separados por cualquiera de los caracteres contenidos en la cadena "delimitantes".

En la primera llamada, la función espera un cstring como argumento para "cad", cuyo primer caracter es utilizado como el punto de partida para buscar tokens. En las llamadas siguientes, la función espera un puntero NULL y utiliza la posición inmediatamente siguiente al final del último token como nuevo punto de partida para realizar la búsqueda.

Para determinar el principio y el fin de un token, la función primero busca desde la posición inicial la ubicación del primer caracter en "cad" que **no** está contenido en la cadena "delimitantes" (que se convierte en el principio del token). Y entonces busca, partiendo desde ese principio del token, al primer caracter en "cad" contenido en "delimitantes", que se convierte en el final del token. La búsqueda también se detiene si se llega al caracter de fin de cadena.

Este final del token es automáticamente reemplazado por el caracter de fin de cadena, y el principio del token es retornado como puntero por la función.

Una vez que el caracter de fin de cadena en "cad" es encontrado por strtok(), todas las subsecuentes llamadas a esta función (con un puntero NULL como primer argumento) retornarán un puntero NULL.

El punto donde el último token fue encontrado se mantiene guardado internamente por la función para ser usado en la siguiente llamada.

Valor de retorno: Un puntero al último token encontrado en "cad".

Un puntero NULL si no hay más tokens por retirar.

Ejemplos de uso:

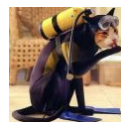
```
/* ejemplo en C */
#include <stdio.h>
#include <string.h>

int main ()
{
    char cad[] = "- Esta, una cadena de prueba";
    char * pch;
    printf( "Partiendo la cadena \"%s\" en tokens:\n\n",
        cad );
    pch = strtok( cad, " ,.-" );
    while( pch != NULL )
    {
        printf( "%s\n", pch );
        pch = strtok( NULL, " ,.-" );
    }
    return 0;
}
```

★
A favoritos

💬
Ir a
comentarios

✉
Compartir
via mail



elwear
REGULAR

38
SEGUIDORES

5
POSTS

643
PUNTOS

T

11



T!

2

```
// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
using namespace std;

int main ()
{
    char cad[] = "- Esta, una cadena de prueba.";
    char * pch;
    cout<<"Partiendo la cadena \""
    <<cad<<"\" en tokens:\n\n";
    pch = strtok( cad, " ,.-" );
    while( pch != NULL )
    {
        cout<< pch <<endl;
        pch = strtok( NULL, " ,.-" );
    }
    return 0;
}
```

Salida:

Partiendo la cadena "- Esta, una cadena de prueba." en tokens:

*Esta
una
cadena
de
prueba*

TARINGA!

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

Devuelve un puntero a una cadena que contiene un mensaje sobre un error producido.

La función interpreta el valor de *errnum*, generando una cadena con un mensaje que describe la condición del error como fue descrita en *errno* por una función de la librería.

El puntero retornado apunta a una cadena estática, que no deberá ser modificada por el programa. Subsecuentes llamadas a esta función pueden sobrescribir su contenido.

La cadena producida por *strerror()* puede ser específica para cada sistema e implementaciones de la librería.



Valor de retorno: Un puntero a una cadena que describe el error cuyo código representa *errnum*.

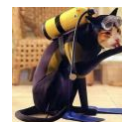
A favoritos



Ejemplos de uso:

Ir a
comentariosCompartir
via mail

```
/* ejemplo en C */
#include <stdio.h>
#include <string.h>
#include <errno.h>
```



elwear

REGULAR

38

SEGUIDORES

5

POSTS

643

PUNTOS

```

T
11 int main ()
    {
        FILE * pFile;
        pFile = fopen( "inexistente.ent", "r" );
        if ( pFile == NULL ) {
            printf("Error al abrir el archivo unexist.ent: %s
T!
2         return 0;
        }
    }

```

```

// Ejemplo de uso en C++
#include <iostream>
#include <cstring>
#include <fstream>
#include <cerrno>
using namespace std;

int main ()
{
    ifstream flujo( "inexistente.ent" );
    if ( !flujo.is_open() ) {
        cout<<"Error al abrir el archivo unexist.ent: "
<<strerror( errno );
    }
    return 0;
}

```

Salida:

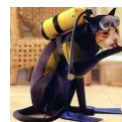
Error al abrir el archivo unexist.ent: No such file or directory

En el ejemplo el programa trata de abrir un archivo que no existe, falla e informa del error. El error es identificado por el macro **errno**, al que se accede haciendo el correspondiente include al principio, y este es pasado como argumento a `strerror()` para obtener su cadena informativa correspondiente.

errno representa el "número del último error". Este macro se expande a un lvalue modificable de tipo `int`. Por lo tanto, puede ser tanto leído como modificado por un programa.

errno es seteado en cero cuando el programa es iniciado, y cualquier función de la librería estándar de C puede modificar su valor a otro diferente de cero, generalmente para señalar una categoría específica de error (ninguna librería vuelve a poner su valor en cero una vez que fue cambiado).

Un programa también puede modificar su valor. De hecho, si esta variable fue pensada para utilizarse para comprobar errores después de la llamada a una función de la librería, debería ser reiniciada a cero por el programa antes de la llamada (ya que cualquier llamada anterior a una función de la librería podría haber alterado su valor).



elwear
REGULAR

38
SEGUIDORES

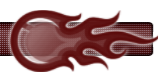
5
POSTS

643
PUNTOS



A favoritos

TARINGA!



Ir a
comentarios

¡ bueno eso ha sido todo. Espero que les sea útil y les sirva como una pequeña referencia para su día a día en el aprendizaje y resolución de problemas. En un futuro post les mostraré algunas maneras de combinar las funciones y sacarles el jugo al máximo empleando sobre todo operaciones con punteros.

Compartir
via mail



T

11 ¡Hasta la próxima!



f Compartir

Twitter Twittear

--

i Fuentes de Información - [Programación C/C++] Funciones para cadenas o cstring

Reference - C++ Reference

Tags: programación | cadenas | programadores | cpp | arreglos | cplusplus | string.h | cstring | cadenas char | arreglos char

© Dar puntos

1	2	3	4	5	6	7	8	9	+10
---	---	---	---	---	---	---	---	---	-----

T!	★	DENUNCIAR	SEGUIR
----	---	-----------	--------

330PUNTOS

36 Votos
50.652 Visitas
24 Favoritos
1 Seguidores

From The Web

Sponsored Links by Taboola

Este juego de estrategia te mantendrá despierto toda la noche!

Sparta: Juego en Linea Gratis

¡Los que juegan a este juego se olvidan de la realidad! ¿Quieres poner a prueba tu valentía?

Soldiers INC.: Juego en Linea Gratis

5 razones por las que 1,000,000 están usando esta app para aprender idiomas

Babbel

Nuevo truco le permite rastrear cualquier cosa o persona usando sólo un teléfono inteligente.

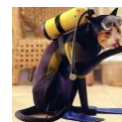
TrackR

Cualquier idioma extranjero en 2 semanas - es posible!

Fast Phrases

¿Cómo Localizar Tu coche Gratis Con Tu Móvil En América Latina?

TechieFans



elwear
REGULAR

38
SEGUIDORES

5
POSTS

643
PUNTOS

Señales que delatan a las mujeres infieles

★ digo.do

A favoritos

Novelas populares que no olvidarás

Ir a digo.do

comentarios



Compartir
via mail

También te puede interesar...

T

11



T!
 2

Ruta 40 Yo te banco! No me

La fotografia perfecta?

Esta chica asiatica tiene el cuerpo

¿Cómo menstrúan las astronautas en

razones por las que no ganas

Decadencia Humana en

14 comentarios - [Programación C/C++] Funciones para cadenas o cstring

cxs00 Hace 3 años **+3**
se agradece, yo le cambiaria el nombre a <pain.h> XD

elwear Hace 3 años **+1**
Si, es un dolor... nunca me voy a olvidar esas ganas de prender fuego cosas cuando terminamos con esto y pasamos a std::string pero qué se le va a hacer es necesario xd

Balium Hace 3 años **+1**
Excelente aporte, muchas gracias!!

problem01 Hace 3 años **+2**
Excelente aporte! A la noche vuelvo con puntos 😊

chero3loco Hace 3 años **+2**
Este si es un aporte 🙌
e insisto que un post sobre estructuras dinámicas [listas simples, dobles, circulares simples y dobles , etc] Ayudaría a comprender estos Temas 😊

Cargar 1 más

chero3loco Hace 3 años
@elwear jaja si bueno son los temas que vamos viendo en clases y pues si le entiendo pero a la logica que se aplica es donde me confundo.

elwear Hace 3 años **+1**
@chero3loco Mmm pues si es de la lógica sola asumiendo que el lector maneja bien los 4 ter mencioné arriba, entonces sería corto el post. En estos días lo hago.

chero3loco Hace 3 años
@elwear pues en realidad no lo hagas tan corto (no asumas que los lectores como yo sabremos TODO) si no una pequeña introducción previa con la info necesaria. Lo espero 🙌

basti21 Hace 2 años



elwear
REGULAR

38 **5** **643**
SEGUIDORES POSTS PUNTOS



A favoritos



Ir a comentarios



Compartir via mail



T

11



T!

2

Como busco la misma palabra 2 veces? ejemplo mi cadena es "heavymetalisheavymetal" el resultado debe ser 3 ya que en el primer heavy recorre y hay 2 metal osea que "heavymetal" se puede formar 2 veces , y en el 2do heavy solo 1 , osea 2+1=3 , como puedo hacer? help pleas

GonzaWonka Hace 2 años

Excelente aporte, justo lo que necesitaba.

BUNBOW Hace 1 año

Excelente

super_postero Hace 9 meses

Muchas gracias, estoy en la universidad y por fin entendi todo esto, super bien explicado, tienes +10 y a fav 😊

MineNeko Hace 9 meses

muchas gracias por toda la info, ahora lo que me queda es practicar, buen aporte, te dejo puntos! 😊

juliosiguil Hace 3 meses

amigo tu post es viejo pero me ayudo la parte de saber que en el strcpy la cadena destino tenia que tener +1 de tamaño eso me estaba volviendo loco y me fue de mucha ayuda arreglo mi problema

medra94 Hace 3 meses

Capo!!!!!!!!!!!! q groso!!!! nunca haia visto ejemplos mas claros que el agua... y justo tengo parcial en 3 horas jajajaj!!!!+10!!!

Nahuel375 Hace 2 meses

Tu post es una luz en medio de la oscuridad de Internet...

Ultimo-Intento Hace 2 meses

demasiado claro, gracias

santy_1993 Hace 1 mes

AYUDAA : Diseñar un programa que permita generar direcciones de correo electrónico. El programa recibe el apellido y nombre de un usuario de la facultad (apellido y nombre se asignan a una sola variable) y debe retornar la dirección de correo electrónico (email) generada. El dominio asignado a la Facultad para el email es: frsf.edu.ar, y el nombre de usuario se forma con la inicial del nombre y a continuación el apellido.

ciencia-educacion

A programar! Mira de lo que es capaz

A favoritos

imagenes

Caricaturas que muestran lo irónico

info

iPhone 7: el smartphone de



imagenes

Ilustraciones oscuras y



elwear

REGULAR

38

SEGUIDORES

5

POSTS

643

PUNTOS

ciencia-educacion

La maquina del tiempo de internet

offtopic

Estas 15 mujeres nunca deberían

Anuncie en Taringa! | Ayuda | Reglas de la Comunidad | Desarrolladores | Trabaja con nosotros! | Reportar bug | Enciclopedia

Argentina | Chile | Colombia | México | Perú | Uruguay | Venezuela | Resto del mundo

Denuncias | Reportar Abuso - DMCA | Términos y Condiciones | Privacidad de datos

11



T!

2



elwear
REGULAR

38
SEGUIDORES

5
POSTS

643
PUNTOS



A favoritos



Ir a
comentarios



Compartir
via mail

