

C string function - strlen()

Descripción

La función **size_t strlen(const char *str)** calcula la longitud de la cadena apuntada por **str** sin incluir el caracter nulo de terminación.

Declaración

Declaración de la función strlen().

```
size_t strlen(const char *str)
```

Parámetros

- **str** – Es la cadena a la cual se le calculará su longitud.

Valor de Retorno

Esta función retorna la longitud de la cadena.

Ejemplo

Ejemplo de uso de la función strcmp().

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

int main () {
    char str[20] = "EjemploDeStrcpm";
    int len;

    len = strlen(str);
    printf("Length of |%s| is |%d|\n", str, len);

    return(0);
}
```

```
}
```

Resultado

```
Length of |EjemploDeStrcpm| is |15|
```

C string function - strcpy()

Descripción

La función **char *strcpy(char *dest, const char *src)** copia la cadena apuntada por **src** a cadena apuntada por **dest**

Declaración

Declaración de la función strcpy().

```
char *strcpy(char *dest, const char *src)
```

Parámetros

- **dest** – Es el puntero de la cadena sobre la cual se copiará el contenido.
- **src** – Es la cadena a ser copiada.

Valor de Retorno

Retorna un puntero sobre la cadena copiada **dest**

Ejemplo

Ejemplo de uso de la función strcpy().

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

```
int main () {  
    char src[40];  
    char dest[100];  
  
    strcpy(src, "Ejemplo de uso de string copy");  
    strcpy(dest, src);  
  
    printf("Final copied string : %s\n", dest);  
  
    return(0);  
}
```

Resultado

Final copied string : Ejemplo de uso de string copy

C string function - strcat()

Descripción

La función **char *strcat(char *dest, const char *src** concatena la cadena apuntada por **src** al final de la cadena apuntada por **dest**

Declaración

Declaración de la función **strcat()**.

```
char *strcat(char *dest, const char *src)
```

Parámetros

- **dest** – Es el puntero de la cadena destino, que debe contener un string de C y debe ser lo suficientemente largo para almacenar la cadena concatenada resultante.
- **src** – Es la cadena a ser concatenada. No debe ser más grande que la cadena destino.

Valor de Retorno

Retorna un puntero sobre la cadena concatenada dest

Ejemplo

Ejemplo de uso de la función strcat().

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

int main () {
    char src[50] = "Esta es la cadena a ser concatenada";
    char dest[90] = "Esta es la cadena destino";

    strcat(dest, src);

    printf("Final destination string : |%s|", dest);

    return(0);
}
```

Resultado

Final destination string : |Esta es la cadena destinoEsta es la cadena a ser c



C string function - strcmp()

Descripción

La función `int strcmp(const char *str1, const char *str2)` compara la cadena apuntada por `str1` con la cadena apuntada por `str2`

Declaración

Declaración de la función strcmp().

```
int strcmp(const char *str1, const char *str2)
```

Parámetros

- **str1** – Es la primer cadena usada en la comparación.
- **str2** – Es la segunda cadena usada en la comparación.

Valor de Retorno

Esta función retorna valores enteros según los siguientes casos:

- si el valor de retorno < 0 entonces str1 es menor a str2.
- si el valor de retorno > 0 entonces str1 es mayor a str2.
- si el valor de retorno $= 0$ entonces str1 es igual a str2.

Ejemplo

Ejemplo de uso de la función strcmp().

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

int main () {
    char str1[15] = "abcdef";
    char str2[15] = "ABCDEF";
    int ret;

    ret = strcmp(str1, str2);

    if(ret < 0) {
        printf("str1 is less than str2");
    } else if(ret > 0) {
        printf("str2 is less than str1");
    } else {
        printf("str1 is equal to str2");
    }

    return(0);
}
```

Resultado

```
str2 is less than str1
```

C string function - strstr()

Descripción

La función **char *strstr(const char *haystack, const char *needle)** encuentra la primer ocurrencia de la subcadena **needle** en la cadena **haystack**. Los caracteres de fin de cadena '\0' no son comparados.

Declaración

Declaración de la función strstr().

```
char *strstr(const char *haystack, const char *needle)
```

Parámetros

- **haystack** – Es la cadena principal sobre la cual se buscará.
- **needle** – Es la pequeña cadena que será buscada sobre la cadena principal.

Valor de Retorno

Esta función retorna un puntero a la primer ocurrencia sobre la cadena principal en donde hubo coincidencia con la pequeña cadena, caso contrario retorna null

Ejemplo

Ejemplo de uso de la función strstr().

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

```
int main () {  
    const char haystack[] = "UsoDeFuncionStrsrtParaBuscarStrings";  
    const char needle[10] = "Funcion";  
    char *ret;  
  
    ret = strstr(haystack, needle);  
  
    printf("The substring is: %s\n", ret);  
  
    return(0);  
}
```

Resultado

The substring is: FuncionStrsrtParaBuscarStrings

C string function - strcspn()

Descripción

La función **size_t strcspn(const char *str1, const char *str2)** calcula el largo sobre el segmento inicial de la cadena apuntada por **str1** que consiste puramente de caracteres no contenidos en la cadena apuntada por **str2**

Declaración

Declaración de la función strcspn().

```
size_t strcspn(const char *str1, const char *str2)
```

Parámetros

- **str1** – Es la cadena principal que será analizada.
- **str2** – Es la cadena que contiene los caracteres a ser buscados en la cadena principal.

Valor de Retorno

Esta función retorna la cantidad de caracteres del segmento inicial de la cadena str1 que no están contenidos en str2.

Ejemplo

Ejemplo de uso de la función strcspn().

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

int main () {
    char str1[20] = "ABCDEF4960910";
    char str2[10] = "013";
    int len;

    len = strcspn(str1, str2);

    printf("Primera coincidencia encontrada en pos = %d\n", len);

    return(0);
}
```

Resultado

```
Primera coincidencia encontrada en pos = 9
```

Ejemplo 2

Puede ser usado para reemplazar un caracter de salto de línea al final de una cadena que querramos eliminar.

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

int main () {
    char str1[20] = "ABCDEF4960910\n";
    char str2[2] = "\n";
    int pos;
```



```
pos = strcspn(str1, str2);  
str1[pos] = '\\0';  
  
printf("Cadena sin salto de linea: %s\\n", str1);  
  
return(0);  
}
```

Resultado

Cadena sin salto de linea: ABCDEF4960910