

# **Clase 5. Lectura y escritura de archivos, manejo de cadenas de caracteres.**

## Archivos

- ❖ Apertura de Archivos (recordatorio)
- ❖ Ejemplo de apertura de archivo
- ❖ Escritura de archivos
- ❖ Lectura de archivos
- ❖ Ejemplo, lectura
- ❖ Caracter nulo, secuencias de escape

Manejo de strings

# Archivos

# Apertura de Archivos (recordatorio)

## Archivos

### ❖ Apertura de Archivos (recordatorio)

❖ Ejemplo de apertura de archivo

❖ Escritura de archivos

❖ Lectura de archivos

❖ Ejemplo, lectura

❖ Caracter nulo, secuencias de escape

Manejo de strings

```
1 FILE *fopen( const char* filename , const char* mode );
```

## Modo Descripción

“r”

Abre el archivo *existente* para lectura.

“w”

Abre o *crea* el archivo para escritura.

“a”

Abre o crea un archivo para escritura al final de su contenido.

“r+”

Abre el archivo para lectura y escritura. El archivo debe de existir y escribe desde el principio (y sobreescribe el contenido actual) y lee desde donde terminó de escribir.

“w+”

Abre el archivo para escritura y lectura. No es necesario que exista.

“a+”

Abre el archivo para escritura lectura. Puede no existir, y escribe/lee desde el final del archivo actual.

# Ejemplo de apertura de archivo

## Archivos

- ❖ Apertura de Archivos (recordatorio)

- ❖ Ejemplo de apertura de archivo

- ❖ Escritura de archivos

- ❖ Lectura de archivos

- ❖ Ejemplo, lectura

- ❖ Caracter nulo, secuencias de escape

## Manejo de strings

En este ejemplo, note:

- Se valida si pudo abrirse el archivo.
- Solo se cierra si pudo abrirse.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 int main()
6 {
7     FILE *in=fopen("archivo.txt","w");
8     if (in!=NULL)
9     {
10        fprintf(in,"Mundo Hola!\n");
11        fclose(in);
12    } else {
13        printf("archivo.txt no se pudo abrir\n");
14    }
15 }
```

# Escritura de archivos

Funciones para escribir a archivo (en stdio.h):

- `int fputc( int c, FILE *fp );` Copia el contenido en `c` al archivo apuntado por `fp`. Si la copia es exitosa devuelve el valor del caracter escrito, en otro caso devuelve EOF.
- `int fputs(const char *str, FILE *fp);` Copia el contenido de una cadena de caracteres al archivo en `fp`. Devuelve un valor no negativo si es exitosa, y EOF sino.

```
1 int main ()
2 {
3     FILE *out=fopen( "archivo.txt", "w" );
4     char s[]={ "Palabra" };
5     if (out!=NULL)
6     {
7         fprintf( out, "%s", s );
8         fputs( s, out );
9         fputc( s[5], out );
10        fclose( out );
11    }
12 }
```

## Archivos

❖ Apertura de Archivos (recordatorio)

❖ Ejemplo de apertura de archivo

❖ **Escritura de archivos**

❖ Lectura de archivos

❖ Ejemplo, lectura

❖ Caracter nulo, secuencias de escape

## Manejo de strings

# Lectura de archivos

Funciones para leer de archivo (en stdio.h):

- `int fgetc( FILE * fp );` Lee un caracter contenido en el archivo apuntado por `fp`. Si la copia es exitosa devuelve el valor del caracter, en otro caso devuelve EOF.
- `char *fgets( char *buf, int n, FILE *fp );` Lee de `fp` hasta `n-1` caracteres y los copia a `buf`. Devuelve un valor no negativo si es exitosa, y EOF sino.

```
1  int main () {  
2    FILE *in=fopen( "archivo.txt", "r" );  
3    char s[]={ "Palabra" };  
4    if ( in !=NULL )  
5    {  
6      fscanf( in , "%s" ,s );  
7      printf ( "%s %c\n" ,s ,s [4] ) ;  
8      fgets ( s ,2 , in ) ;  
9      s [4]= fgetc ( in ) ;  
10     fclose ( in ) ;  
11     printf ( "%s %c\n" ,s ,s [4] ) ;  
12   }  
}
```

## Archivos

❖ Apertura de Archivos (recordatorio)

❖ Ejemplo de apertura de archivo

❖ Escritura de archivos

❖ Lectura de archivos

❖ Ejemplo, lectura

❖ Caracter nulo, secuencias de escape

Manejo de strings

# Ejemplo, lectura

Palabra 123456

## Archivos

- ❖ Apertura de Archivos (recordatorio)
- ❖ Ejemplo de apertura de archivo
- ❖ Escritura de archivos
- ❖ Lectura de archivos
- ❖ **Ejemplo, lectura**
- ❖ Caracter nulo, secuencias de escape

## Manejo de strings

```
1 int main() {
2     FILE *in=fopen("archivo.txt","r");
3     char s[]={ "Palabra" };
4     if (in!=NULL)
5     {
6         fscanf(in,"%s",s);
7         printf("%s %c\n",s,s[4]);
8         fgets(s,2,in);
9         s[4]=fgetc(in);
10        fclose(in);
11        printf("%s %c\n",s,s[4]);
12    }
13 }
```

Salida:  
Palabra b  
1

# Caracter nulo, secuencias de escape

## Archivos

- ❖ Apertura de Archivos (recordatorio)
- ❖ Ejemplo de apertura de archivo
- ❖ Escritura de archivos
- ❖ Lectura de archivos
- ❖ Ejemplo, lectura

## ❖ Caracter nulo, secuencias de escape

## Manejo de strings

Las secuencias de escape comienzan con una diagonal invertida (backslash). Cualquier compilador las imprime en la salida como su correspondiente valor ASCII (pueden tener un efecto diferente si el sistema no esta basado o no reconoce de la misma forma los caracteres ASCII):

<code>\a</code>	Alarma (beep, o campana).
<code>\b</code>	backspace.
<code>\f</code>	Formfeed (termina la pagina actual).
<code>\n</code>	Nueva línea.
<code>\r</code>	Retorno de carro.
<code>\t</code>	Tabulador.
<code>\v</code>	Tab vertical.
<code>\\</code>	Backslash.
<code>\'</code>	Comilla simple.
<code>\"</code>	Doble comilla.
<code>\?</code>	Signo de interrogación.
<code>\nnn</code>	Caracter asociado con el número nnn (generalmente el de código ASCII).
<code>\xhh</code>	Similar al anterior pero con numero hexadecimal.



- ❖ String
- ❖ string.h
- ❖ string.h
- ❖ Tarea, prog 3.1
- ❖ Referencias
- ❖ Traer maquina

# Manejo de strings

# String

Archivos

Manejo de strings

❖ String

❖ string.h

❖ string.h

❖ Tarea, prog 3.1

❖ Referencias

❖ Traer maquina

- Un string es una secuencia (arreglo) de caracteres que termina con caracter nulo o valor cero: `'\0'` o `(char)0`. Ej.

```
'H' 'o' 'l' 'a' '\0'
```

# String

Archivos

Manejo de strings

❖ String

❖ string.h

❖ string.h

❖ Tarea, prog 3.1

❖ Referencias

❖ Traer maquina

- Un string es una secuencia (arreglo) de caracteres que termina con caracter nulo o valor cero: `' \0'` o `(char)0`. Ej.

```
'H' 'o' 'l' 'a' '\0'
```

- Apuntador a string. Es la dirección de memoria donde comienza el almacenamiento del string. `char *cadena=(char *)malloc(5*sizeof(char)); cadena="Hola";`.

# String

Archivos

Manejo de strings

❖ String

❖ string.h

❖ string.h

❖ Tarea, prog 3.1

❖ Referencias

❖ Traer maquina

- Un string es una secuencia (arreglo) de caracteres que termina con caracter nulo o valor cero: `' \0 '` o `(char)0`. Ej.

```
'H' 'o' 'l' 'a' '\0'
```

- Apuntador a string. Es la dirección de memoria donde comienza el almacenamiento del string. `char *cadena=(char *)malloc(5*sizeof(char)); cadena="Hola";`.
- C no tiene operadores nativos para el manejo de arreglos por lo que todo se realiza por medio de funciones, entre ellas las de la librería estándar `string.h`.

# string.h

- **size\_t strlen ( const char \* str );** Devuelve el largo de una cadena, lo determina por medio del caracter nulo.
- **char \* strcpy ( char \* destination, const char \* source );** Devuelve el largo de una cadena, lo determina por medio del caracter nulo. Se copia hasta el caracter nulo, si las cadenas (origen y destino) se traslapan, el comportamiento no está definido.

Archivos

Manejo de strings

❖ String

❖ string.h

❖ string.h

❖ Tarea, prog 3.1

❖ Referencias

❖ Traer maquina

```
1 #include <string.h>
  int main()
3 {
  char s1[12], s2[24], *s3={"mundo! \n"}, *a1, *a2;
5 printf("Longitud de s3=%u\n", strlen(s3));
  a1=strcpy(s1, s3);
7 printf("Copia de s3=%s a1=%p s1=%p\n", a1, a1, s1);
  a2=strcpy(s2, "Hola ");
9 strcat(s2, s1);
  printf("Copia de a2=%s \n", a2);
11 return 0;
  }
13
```

# string.h

- **char \* strcat ( char \* destination, const char \* source );**  
Concatena dos cadenas de caracteres.
- **int strcmp ( const char \* str1, const char \* str2 );** Compara dos cadenas de caracteres. Devuelve 0 si son iguales, y diferente de 0 sino.

Archivos

Manejo de strings

❖ String

❖ string.h

❖ string.h

❖ Tarea, prog 3.1

❖ Referencias

❖ Traer maquina

```
1 #include <string.h>
2 int main()
3 {
4     char s1[12], s2[24], *s3={"mundo! \n"}, *a1, *a2;
5     printf("Longitud de s3=%u\n", strlen(s3));
6     a1=strcpy(s1, s3);
7     printf("Copia de s3=%s a1=%p s1=%p\n", a1, a1, s1);
8     a2=strcpy(s2, "Hola ");
9     strcat(s2, s1);
10    printf("Copia de a2=%s \n", a2);
11    return 0;
12 }
```

# string.h

## Archivos

## Manejo de strings

### ❖ String

### ❖ string.h

### ❖ string.h

### ❖ Tarea, prog 3.1

### ❖ Referencias

### ❖ Traer maquina

```
1 #include <string.h>
2 int main()
3 {
4     char s1[12], s2[24], *s3={"mundo! \n"}, *a1, *a2;
5     printf("Longitud de s3=%u\n", strlen(s3));
6     a1=strcpy(s1, s3);
7     printf("Copia de s3=%s a1=%p s1=%p\n", a1, a1, s1);
8     a2=strcpy(s2, "Hola ");
9     strcat(s2, s1);
10    printf("Copia de a2=%s \n", a2);
11    if(strcmp("Hola Mundo!", s2)==0)
12        printf("No es igual\n");
13    if(strcmp("Hola mundo! \n", s2)==0)
14        printf("Es igual\n");
15    return 0;
16 }
```

Longitud de s3=8

Copia de s3=mundo!

a1=0x7fff9f19a020 s1=0x7fff9f19a020

Copia de a2=Hola mundo!

Es igual

# string.h

Archivos

Manejo de strings

❖ String

❖ string.h

❖ string.h

❖ Tarea, prog 3.1

❖ Referencias

❖ Traer maquina

- **void \* memchr ( const void \* ptr, int value, size\_t num );**  
Encuentra la primera ocurrencia de “value” (unsigned char).

```
1 #include <string.h>
2 int main()
3 {
4     char *s3={"Hola \n mundo!"},*p;
5     int val='\n';
6     printf("Valor del salto de linea=%i %i\n", '\n', val);
7     p=memchr ( s3, val, strlen(s3));
8     printf("Encontre un salto de linea en=%p %c",p,*p,*p
9         ,*p);
10    return 0;
11 }
```

Valor del salto de linea=10 10

Encontre un salto de linea en=0x40073d



# string.h

Archivos

Manejo de strings

❖ String

❖ string.h

❖ string.h

❖ Tarea, prog 3.1

❖ Referencias

❖ Traer maquina

- **int memcmp ( const void \* ptr1, const void \* ptr2, size\_t num );** Compara los primero “num: bytes apuntador por ptr1, con los de ptr2.

```
1 #include <string.h>
2 int main()
3 {
4     char *s1={"mundo! Hola"}, *s2={"Adios mundo\n"}, *s3;
5     s3=s2+6;
6     if (memcmp(s1, s3, 3)==0)
7         printf("%c%c%c es igual con %c%c%c \n", s1[0], s1[1],
8             s1[2], s3[0], s3[1], s3[2]);
9     return 0;
10 }
```

mun es igual con mun

# string.h

## Archivos

### Manejo de strings

#### ❖ String

#### ❖ string.h

#### ❖ string.h

#### ❖ Tarea, prog 3.1

#### ❖ Referencias

#### ❖ Traer maquina

- **void \* memcpy ( void \* destination, const void \* source, size\_t num );** Copia los primeros num bytes (no verifica caracter nulo ni nada).
- **void \* memmove ( void \* destination, const void \* source, size\_t num );** Copia del fuente al destino permitiendo que sean la misma memoria, se utiliza un buffer intermedio (puede mover bytes dentro del mismo arreglo).
- **void \* memset ( void \* ptr, int value, size\_t num );** Asigna el valor de value a los primeros num caracteres de ptr.
- **char \* strchr ( char \* str, int character );** Encuentra la primera ocurrencia de character en str.
- **int strcoll ( const char \* str1, const char \* str2 );** Compara str1 con str2, pero la comparacion depende de como se han fijado lo "locales" en el sistema (la informacion de internacionalización).

# string.h

- **size\_t strcspn ( const char \* str1, const char \* str2 );**  
Encuentra la primera ocurrencia de cualquier caracter de str2 en str1.

Archivos

Manejo de strings

❖ String

❖ string.h

❖ string.h

❖ Tarea, prog 3.1

❖ Referencias

❖ Traer maquina

```
1 #include <string.h>
2 int main()
3 {
4     char palabra [] = "fcba73";
5     char numero [] = "1234567890";
6     int i;
7     i = strcspn (palabra , numero);
8     printf ("En %s hay un numero en la posicion %d.\n" ,
9           palabra , i);
10    return 0;
11 }
```

En fcba73 hay un numero en la posicion 4.

# string.h

- **char \* strerror ( int errnum );** Interpreta errnum como un código de error y devuelve el string correspondiente. errno es una librería y una variable (también llamada errno) entera que define errores de llamadas a sistema y algunas funciones.

Archivos

Manejo de strings

❖ String

❖ string.h

❖ string.h

❖ Tarea, prog 3.1

❖ Referencias

❖ Traer maquina

```
2 #include <string.h>
2 #include <errno.h>
4 int main ()
  {
6   FILE * pFile ;
   pFile = fopen ("unexist.ent", "r");
8   if (pFile == NULL)
       printf ("Error opening file unexist.ent: %s\n",
           strerror(errno));
10  return 0;
   }
```

Salida:

```
Error opening file unexist.ent: No such file or direc
```

Ejemplo de:

<http://www.cplusplus.com/reference/cstring/strerror/>

# *string.h*

## Archivos

### Manejo de strings

❖ String

❖ string.h

❖ **string.h**

❖ Tarea, prog 3.1

❖ Referencias

❖ Traer maquina

- **char \* strncat ( char \* destination, const char \* source, size\_t num );** Anexa los primeros num caracteres de fuente a destino.
- **int strncmp ( const char \* str1, const char \* str2, size\_t num );** Compara los primeros num caracteres o hasta donde encuentre el carácter nulo.
- **char \* strncpy ( char \* destination, const char \* source, size\_t num );** Copia num caracteres de fuente a destino. Si no se encuentra el caracter nulo después de num caracteres, entonces no se inserta (el caracter nulo) a la cadena destino, si se encuentra antes, entonces las posiciones se rellenan con el carácter nulo hasta completar las num posiciones.

# string.h

**char \* strpbrk (char \* str1, const char \* str2 );** Regresa la primera ocurrencia de cualquier caracter en str2 que se encuentre en str1.

## Archivos

### Manejo de strings

- ❖ String
- ❖ string.h
- ❖ string.h
- ❖ Tarea, prog 3.1
- ❖ Referencias
- ❖ Traer maquina

```
1 #include <stdio.h>
2 #include <string.h>
3 int main ()
4 {
5     char str[] = "This is a sample string";
6     char key[] = "aeiou"; char * pch;
7     printf ("Vowels in '%s ': ", str);
8     pch = strpbrk (str, key);
9     while (pch != NULL){
10         printf ("%c ", *pch);
11         pch = strpbrk (pch+1, key);
12     }
13     printf ("\n");
14     return 0;
15 }
```

Salida:

```
Vowels in 'This is a sample string': i i a a e i
```

Ejemplo de:

<http://www.cplusplus.com/reference/cstring/strpbrk/>

# string.h

**char \* strchr ( const char \* str, int character );** Regresa la última ocurrencia de character en str.

## Archivos

### Manejo de strings

- ❖ String
- ❖ string.h
- ❖ string.h
- ❖ Tarea, prog 3.1
- ❖ Referencias
- ❖ Traer maquina

```
1 #include <stdio.h>
2 #include <string.h>
3 int main ()
4 {
5     char str [] = "This is a sample string";
6     char * pch;
7     pch=strchr ( str , 's' );
8     printf ("Last occurrence of 's' found at %d \n",pch-
9         str+1);
10    return 0;
11 }
```

Salida:

Last occurrence of 's' found at 18

Ejemplo de:

<http://www.cplusplus.com/reference/cstring/strchr/>

# string.h

**size\_t strspn ( const char \* str1, const char \* str2 );** Regresa el largo de la porción inicial de str1 que consiste "solo" en caracteres de str2.

## Archivos

### Manejo de strings

- ❖ String
- ❖ string.h
- ❖ string.h
- ❖ Tarea, prog 3.1
- ❖ Referencias
- ❖ Traer maquina

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main ()
5 {
6     int i;
7     char strttext [] = "129th";
8     char strttext2 [] = "AB129th";
9     char cset [] = "1234567890";
10    i = strspn (strttext ,cset);
11    printf ("The initial number has %d digits.\n",i);
12    i = strspn (strttext2 ,cset);
13    printf ("The initial number has %d digits.\n",i);
14    return 0;
15 }
```

Salida:

```
The initial number has 3 digits.
```

```
The initial number has 0 digits.
```

Ejemplo modificado de:

<http://www.cplusplus.com/reference/cstring/strspn/>



# *string.h*

**char \* strstr (char \* str1, const char \* str2 );** Regresa el apuntado de la primera ocurrencia de str2 en str1 o un apuntador a null si no existe la ocurrencia.

## Archivos

### Manejo de strings

- ❖ String
- ❖ string.h
- ❖ **string.h**
- ❖ Tarea, prog 3.1
- ❖ Referencias
- ❖ Traer maquina

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main ()
5 {
6     char str [] = "This is a simple string";
7     char * pch;
8     pch = strstr (str, "simple");
9     strncpy (pch, "sample", 6);
10    puts (str);
11    return 0;
12 }
```

Salida:

```
This is a sample string
```

Ejemplo modificado de:

<http://www.cplusplus.com/reference/cstring/strstr/>

# string.h

**size\_t strxfrm ( char \* destination, const char \* source, size\_t num );** El comportamiento básico es regresar el tamaño del source y copiar num caracteres a destination. En caso de que se defina un locale, este se utiliza para transformar destination.

Archivos

Manejo de strings

❖ String

❖ string.h

❖ string.h

❖ Tarea, prog 3.1

❖ Referencias

❖ Traer maquina

```
1 #include <stdio.h>
2 #include <string.h>
3 int main(void)
4 {
5     char source[] = "Hola mundo";
6     char destination[50];
7     printf("La longitud de source es = %d\n", strxfrm(
8     destination, source, 4));
9     printf("Contenido de destination = %s\n",
10    destination);
11    printf("Contenido de source = %s\n", source);
12    return 0;
13 }
```

Salida:

La longitud de source es = 10

Contenido de destination = Hola

Contenido de source = Hola mundo

# Tarea, prog 3.1

◆ Determinar si Guanajuato es el centro de México, o que tan cerca está de serlo. Se proporciona un archivo de datos. Contiene una matriz con 3 tipos de valores de caracteres: 'n' que indica que ese pixel no es parte del territorio de México. 'm' si lo es, y 'g' si además es Guanajuato. El centroide de México es simplemente el promedio de las posiciones que son territorio Mexicano, el cual tiene dos coordenadas, lo mismo para el de Guanajuato. Imprimir los centroides y su distancia.

1. Lea de consola nombres de archivos de entrada y de salida.
2. Función1 recibe el nombre del archivo de entrada y devuelve un char\*\* con "toda" la información necesaria para los cálculos y reproducir el archivo original. En el archivo de entrada: dos o mas espacios consecutivos en blanco pueden considerarse como uno solo, al igual que los saltos de línea.
3. Función2 recibe el char \*\* de la anterior y un vector double de 5 posiciones que contendrá: los centroides del Mexico y Guanajuato (en coordenadas relativas al mapa), y su distancia.
4. Función3 escribir el archivo de salida con la misma información que el original (de 'n', 'm' y 'g'). Recibe un char \*\* y el nombre de archivo de salida
5. Requisitos mínimos del programa: cálculo correcto, el archivo de salida y de entrada contienen los mismos datos. Si el archivo de salida se usa como de entrada, debe imprimir un archivo igual a si mismo y calcular los mismos centroides y distancia. Las funciones deben cumplir con el prototipo mencionado, no debe haber accesos indebidos a memoria, o memoria no devuelta.

Bonus Mientras menos memoria se requiera y se utilice en el programa este tendrá una mayor bonificación. Sugerencia: realicen un programa que cumpla con los requerimientos y después (si hay tiempo) traten de optimizarlo.

Archivos

Manejo de strings

❖ String

❖ string.h

❖ string.h

❖ Tarea, prog 3.1

❖ Referencias

❖ Traer maquina

# Graficar datos en R

## Archivos

## Manejo de strings

- ❖ String
- ❖ string.h
- ❖ string.h
- ❖ Tarea, prog 3.1
- ❖ Referencias
- ❖ Traer maquina

```
1 mapa=as.matrix(read.table('mapa.txt'))
  mapa[mapa=='n']=1
3  mapa[mapa=='m']=2
  mapa[mapa=='g']=3
5  image(1:nrow(mapa), 1:ncol(mapa), matrix(as.numeric(t(
      mapa)), byrow=TRUE, nrow= nrow(mapa), ncol=ncol(mapa))
      , col=rainbow(3))
```

# Referencias

## Archivos

---

## Manejo de strings

---

- ❖ String
- ❖ string.h
- ❖ string.h
- ❖ Tarea, prog 3.1
- ❖ Referencias
- ❖ Traer maquina

Libreria String.h

<http://www.cplusplus.com/reference/cstring/>

# *Traer maquina*

Archivos

Manejo de strings

- ❖ String
- ❖ string.h
- ❖ string.h
- ❖ Tarea, prog 3.1
- ❖ Referencias
- ❖ **Traer maquina**

Traer su maquina la siguiente clase, haremos un ejercicio.