

# UNIDAD 5

E/S

# ENTRADA Y SALIDA ESTÁNDAR

- Un archivo contiene caracteres y números estos son procesados por la computadora como series de bytes a los cuales se les llama flujo.
- El lenguaje C trata de la misma manera todos los flujos de archivos aunque algunos provengan del disco duro/USB. pantalla o impresora.

# TIPOS DE FLUJOS

- ◉ Stdin - la entrada estándar para lectura
- ◉ Stdout - salida estándar de escritura
- ◉ Srerr -error estándar para excribir mensajes de error
- ◉ Stdout se a ha utilizado con printf

# OBTENCIÓN DE ENTRADA DE USUARIO

- ◉ `getc()`
- ◉ Lee el siguiente carácter de flujo de archivo y devuelve el carácter como entero.
- ◉ `#include <stdio.h>`
- ◉ `int getc(FILE *stream);`

- ◉ #include <stdio.h>
- ◉ #include <conio.h>

- ◉ int main()
- ◉ {
- ◉     int ch;

- ◉     printf("Escriba ");
- ◉     ch=getc(stdin);
- ◉     printf("caracter %c \n", ch);

- ◉     getch();
- ◉     return 0;
- ◉ }

## GETCHAR()

```
#INCLUDE <STDIO.H>  
INT GETCHAR(VOID);
```

- int main()
- {
- int ch, ch2;
  
- printf("Escriba dos ");
- ch=getc(stdin);
- ch2=getchar();
- printf("caracter %c \n", ch);
- printf("caracter %c \n", ch2);
  
- getch();
- return 0;
- }

# IMPRESIÓN DE SALIDA EN PANTALLA

- ◉ Putc
- ◉ #include <stdio.h>
- ◉ Int putc(int c, FILE \*stream);
  
- ◉ int ch
- ◉ ch =65
- ◉ putc(ch, stdout)

# PUTCHAR()

- ◉ #include <stdio.h>
- ◉ int putchar(int c);
  
- ◉ putchar(65)



- ⦿ `printf("punto flotante %f; entero %d \n", 147.14/10.0, 140/10);`
- ⦿ Cadena de formato
- ⦿ Expresiones

# CONVERSIÓN A NÚMEROS HEXADECIMALES

- ◉ Sabemos que la computadora trabaja con datos binarios (bits). Y a veces necesitamos ver los datos como programadores.
- ◉ Se puede trabajar con datos en notación hexadecimal.
- ◉ El cual es un sistema intermedio entre binario y base 10 .
- ◉ Hexadecimales en base 16
- ◉ El cual puede estar representado con 4 bits (24 significa que 4 bits pueden producir 16 números únicos)

- ◉ 0-9 hexadecimales = decimales
- ◉ A-F representan los digitos 10-15 decimal
  
- ◉ Escribe un programa que muestre la conversión de decimales a hexadecimales utilizando %x, %X

# ESPECIFICACIÓN DE ANCHO

- ◉ %3d
- ◉ %8f
- ◉ Escribe un programa que muestre diferentes anchos mínimos de salida

# ALINEACIÓN DE SALIDA

- ◉ Por default toda salida se coloca a la derecha  
`%8d`
- ◉ `%-8d` especifica el mínimo de campo y lo alinea a la izquierda
- ◉ Escribe un programa que escriba 5 números alineados a la izquierda y derecha.

# USO DEL ESPECIFICADOR DE PRECISIÓN

- Entero
- int =14789
- Punto flotante
- flt=147.369887
- % d, int
  - 14789
- %d,flt
- %2.8, int
- %10.3,flt

# PROGRAMA ESPECIFICADOR DE PRECISIÓN

- int main()
- {
- int c1;
- float c2;
- c1= 123654;
- c2= 1236.258963;
  
- printf("formato predefinido entero c1 = %d \n", c1);
- printf("con precision = %10.10d \n", c1);
- printf("formato predefinido punto flotante c2 = %f \n", c2);
- printf("con precision = %10.3f \n", c2);
- getch();
- return 0;
- }