05

ESTRUCTURAS Y ARCHIVOS

Estructuras

Las estructuras son una colección de variables relacionadas en naturaleza (o semánticamente) pero no necesariamente en sus tipos de datos. Ejemplo: edad, nombre, promedio pueden formar una estructura de datos de un alumno.

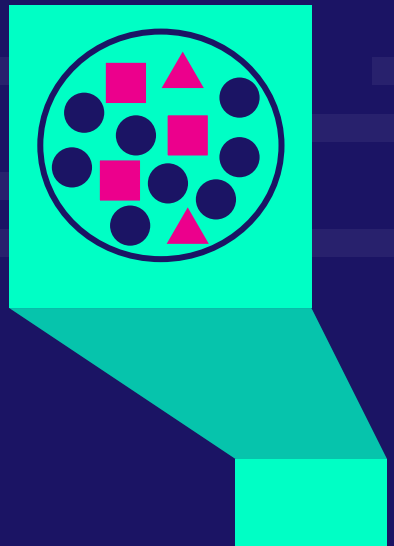
Se utilizan comúnmente para definir un características de una persona, una cosa, un lugar. Se les llama también **registros** puesto que posteriormente se almacenarán en archivos.

Junto con los apuntadores, las estructuras permiten la implementación de estructuras de datos (listas ligadas, colas, pilas y árboles).

Características de una estructura en lenguaje C

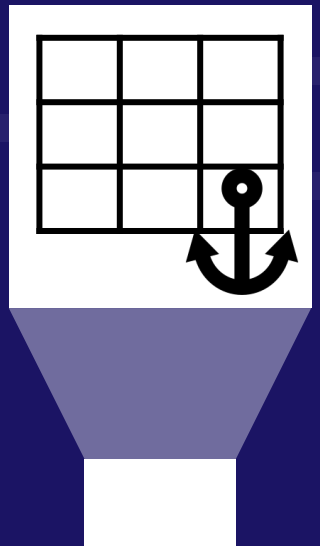
Heterogénea

Es una colección de variables de diferentes tipos



Estático

El número de localidades reservadas por estructura no cambia



Volátil

Se pierde el acceso a los datos cuando el programa termina

Estructuras en lenguaje C

```
struct alumno
{
    char nombre[10];
    char apePat[10];
    char apeMat[10];
    float promedio;
    int estudiaETE;
};
```

Definición de la estructura

```
struct alumno carlos;
```

Declaración de una estructura

```
struct alumno alumnos[5];
```

Declaración de un arreglo de estructuras

Las estructuras son tipos de datos derivados, puesto que se componen de variables de tipos básicos.

Acceso y modificación de valores

```
int main (void)
{
   strcpy(carlos.nombre, “Carlos”);
   strcpy(carlos.apePat, “Campos”);
   strcpy(carlos.apeMat, “De la Garza”);
   carlos.promedio = 8.95;
   carlos.estudiaETE = 1;

   strcpy(alumnos[0].nombre, “Joaquín”);
   strcpy(alumnos[0].apePat, “Pérez”);
   strcpy(alumnos[0].apeMat, “López”);
   alumnos[0].promedio = 7.98;
   alumnos[0].estudiaETE = 0;

   printf(“%s %s: %.2f”,alumnos[0].nombre,
                                    alumnos[0].apePat,alumnos[0].promedio);

   return 0;
}
```

Archivos

Los datos que almacenamos en variables, arreglos o estructuras se guardan en memoria principal (RAM) y al término de la ejecución del programa se "pierden".

Con la finalidad de preservar datos que son relevantes para un programa **se utilizan archivos**, los cuales son almacenados **en memoria secundaria** y son persistentes.

Tipos de archivos

De texto

Consisten en un flujo de caracteres.
Su **lectura** se realiza de forma **secuencial**.

Binarios

También llamados archivos tipeados.
Su **lectura** se realiza en forma aleatoria.

Lectura y escritura de un archivo de texto

```
int main (void)
{
   FILE *arch;
   char res, nombre[10];
   int edad;

   arch = fopen("datos.txt", "w");

   do
   {
     printf("Nombre: ");
     scanf("%10s",nombre);
     printf("Edad: ");
     scanf("%i", &edad);
     getchar();
     fprintf(arch, "%s %i\n", nombre, edad);
     printf("¿Guardar otro?(S/N)");
     scanf("%c", &res);
   }
   while(res != 'N');
   fclose(arch);

   arch = fopen("datos.txt", "r");

   printf("Datos del archivo");

   while(!feof(arch))
   {
      fscanf(arch, "%s%i", nombre, &edad);
      printf("%s, %i\n", nombre, edad);
   }

   fclose(arch);

   return 0;
}
```

Apuntador al archivo lógico

Apertura del archivo

Escritura de datos

Cierre del archivo

Apertura del archivo

Lectura de datos

Cierre del archivo

Función fopen

Esta función realiza la apertura de un archivo. Recibe como argumentos una cadena con la ruta al archivo y un identificador de modo de apertura. La función devuelve un apuntador de tipo FILE o NULL en caso de error.

fopen(“ruta al archivo”, “modo”)

```
arch = fopen(“datos.txt”, “w”);
```

Los modos de apertura válidos se encuentran en la siguiente diapositiva.

Modo	Descripción
r	Abre un archivo existente en modo lectura
w	Abre un archivo para escritura. Si el archivo no existe, lo crea, de lo contrario borra su contenido.
a	Abre o crea un archivo para escribir al final de éste (agrega contenido al archivo).
r+	Abre un archivo para actualizarlo (modo lectura y escritura).
w+	Crea un archivo para leer y escribir.
a+	Abre o crea un archivo para leer y agregar contenido.
rb	Abre un archivo existente para lectura en modo binario.
wb	Crea un archivo para escritura en modo binario.
ab	Abre o crea un archivo para escribir al final del archivo en modo binario.
rb+	Abre un archivo existente para lectura y escritura en modo binario.
wb+	Crea un archivo para escribir y leer en modo binario. Si el archivo ya existe, borra el contenido.
ab+	Abre o crea un archivo para leer y escribir contenido al final del archivo.

Función fclose

Esta función cierra el archivo, es decir cierra la conexión entre el archivo lógico (apuntador de tipo FILE) y el archivo alojado en memoria secundaria.

fclose(apu_archivo)

```
fclose(arch);
```

Función fscanf

Esta función realiza la lectura de los caracteres en el archivo de texto. Convierte el contenido del archivo a los tipos de datos indicados por la máscara de entrada.

fscanf(apu_archivo, máscara, &variable, ...)

```
fscanf(arch, "%s%i", nombre, &edad);
```

Función fprintf

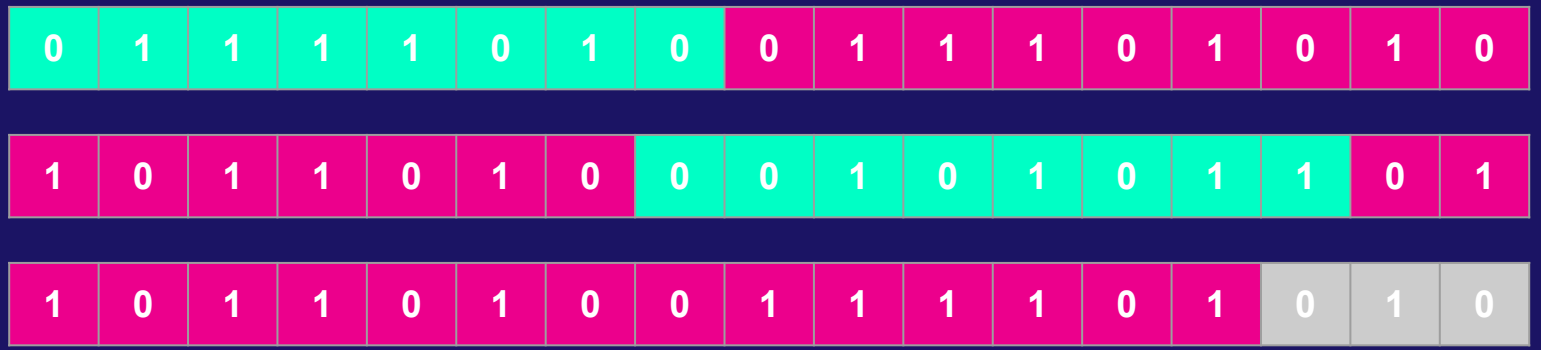
Esta función escribe en el archivo las secuencias de caracteres de acuerdo con las máscaras de salida y los datos proporcionados.

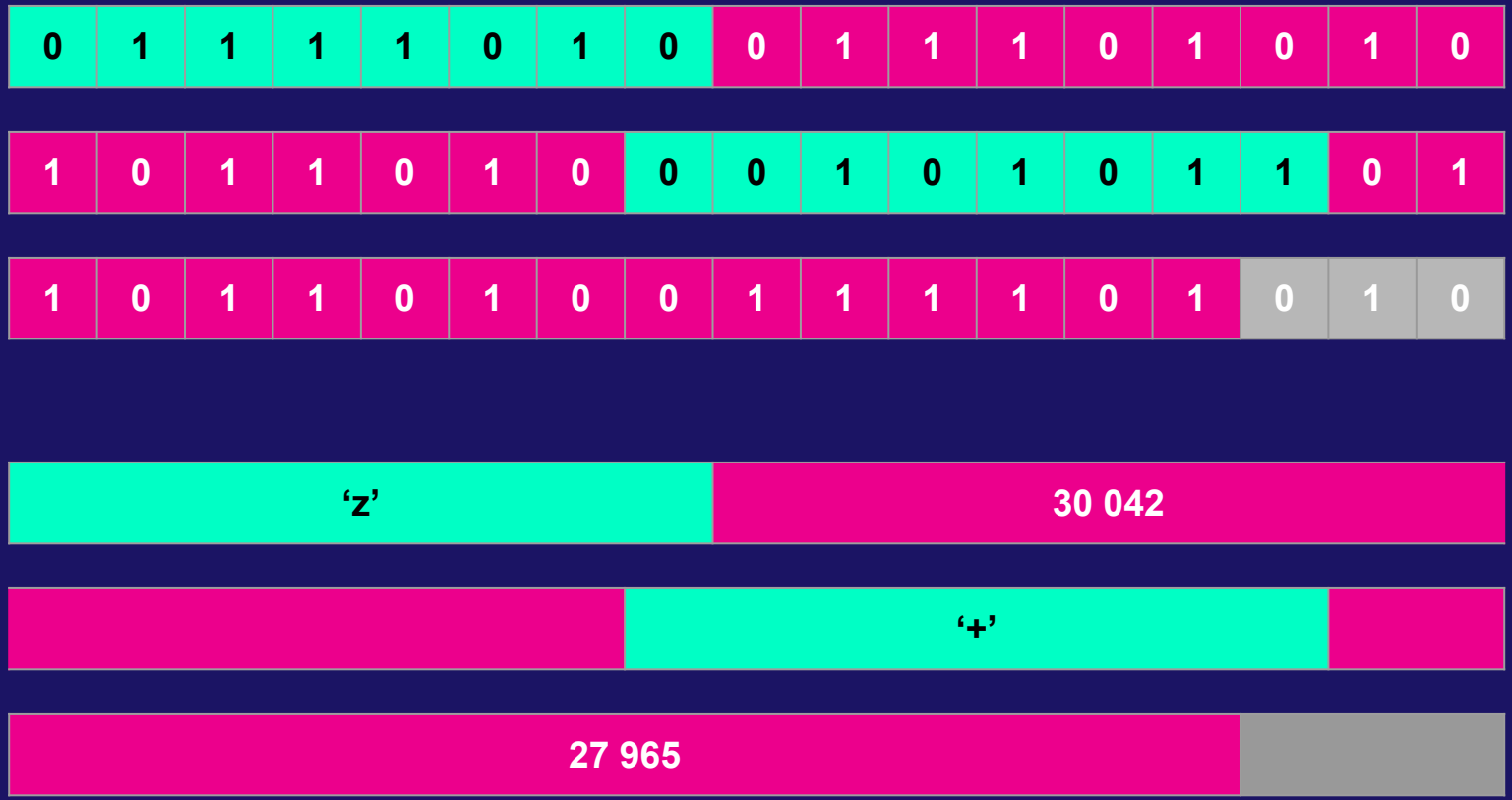
fprintf(apu_archivo, “cadena con formatos”, variable, …)

```
fprintf(arch, "%s %i\n", nombre, edad);
```

Archivos binarios

Los archivos binarios son aquellos archivos cuyo contenido no se puede leer como una secuencia de caracteres. El contenido de estos se conforma por secuencias de bits que al agruparse en estructuras específicas tienen un significado para el programa.




0 1 1 1 1 0 1 0 0 1 1 1 0 1 0 1 0
1 0 1 1 0 1 0 0 0 1 0 1 0 1 1 0 1
1 0 1 1 0 1 0 0 1 1 1 1 0 1 0 1 0
‘z’
30 042
‘+’
27 965

Escritura de un archivo binario

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

struct cuenta
{
  char RFC[10];
  float monto;
};
```

```c
int main(void)
{
  FILE *arch;
  struct cuenta mi_cuenta;

  strcpy(mi_cuenta.RFC, "CAGC951222");
  mi_cuenta.monto = 250368.78;

  arch = fopen("datos.bin", "wb");

  fwrite(&mi_cuenta, sizeof(mi_cuenta), 1, arch);

  fclose(arch);

  return 0;
}
```

Lectura de un archivo binario

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

struct cuenta
{
  char RFC[10];
  float monto;
};
```

```c
int main(void)
{
  FILE *arch;
  struct cuenta mi_cuenta, edo_cuenta;

  arch = fopen("datos.bin", "rb");

  fread(&edo_cuenta, sizeof(edo_cuenta), 1, arch);

  fclose(arch);

  printf("Propietario: %s ", edo_cuenta.RFC);
  printf("Total: %.2f\n", edo_cuenta.monto);

  return 0;
}
```

Acceso aleatorio

```
#include <stdio.h>
int main (void)
{
    FILE *arch;
    struct cuenta edo_cuenta;

    arch = fopen("cuentas.bin", "rb");

    fseek(arch, 3*sizeof(edo_cuenta), SEEK_SET);

    fread(&edo_cuenta, sizeof(edo_cuenta), 1, arch);

    printf("Propietario: %s ", edo_cuenta.RFC);
    printf("Total: %.2f\n", edo_cuenta.monto);

    fclose(arch);
    return 0;
}
```

Renombrado y eliminación de archivos

```
int rename (char* viejo_nombre,  char* nuevo_nombre)
```

La función devuelve cero si no hubo un error.

```
int remove (char* nombre)
```

La función devuelve cero si se pudo eliminar el archivo.