

Laboratorio de Computación II

Archivos

Archivos

Conjunto de bytes que son almacenados en un dispositivo con un nombre identificador.



Archivos de sistema



Archivos ejecutables



Archivos de datos

EXE ←
excel.exe notas.xls

Archivos de datos

Datos con un formato y/o estructura específica que son interpretados por programas para poder crearlos, visualizarlos, editarlos y/o borrarlos.



Archivos de datos



Planilla de cálculo



Código fuente



Archivo PDF



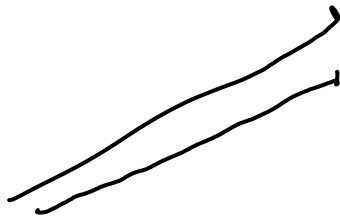
Archivo de clientes

Archivos de datos

Un registro de información de una determinada entidad debe cumplir una serie de reglas para poder ser almacenado en un archivo



Archivo de clientes



Registros deben tener longitud fija.



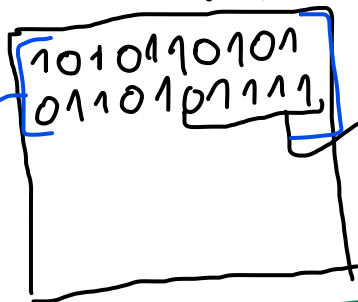
Registros deben ser identificados por un valor único e irrepetible.



El archivo, en consecuencia al ítem anterior, no debiera admitir registros duplicados.

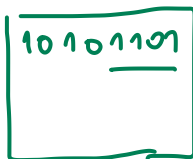
bytes

Stack objPersona



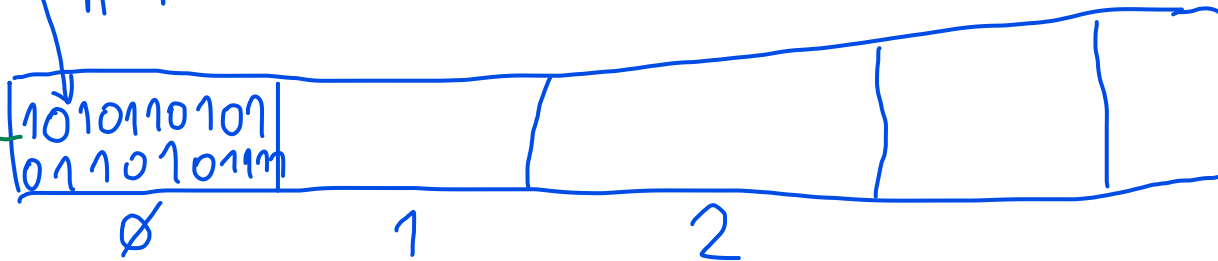
HEAP

1 11



write

read Archivo Personas.dat



Archivos de datos - Escritura

```
class Alumno{  
    private:  
    int legajo;  
    char apellido[50];  
    char nombres[50];  
    int edad;  
};
```



Memoria



Alumno reg = cargar_alumno();



bool guardar_alumno(**Alumno** reg);



Archivo de alumnos

#1	#2	#3
#4	#5	#6
#7	#8	#9

Archivos de datos - Lectura



Archivo de alumnos

ID	Apellido	Nombre	Edad
100	Santos	Mario	40
200	Medina	Gabriel	48
300	Ravenna	Emilio	44
400	Lamponne	Pablo	47


Alumno leer_alumno(int nroRegistro);

Memoria




200	Medina	Gabriel	48
-----	--------	---------	----


Peeeeerooo....



Las funciones de archivos no admiten trabajar con "números de registros".



Las funciones de archivos trabajan con cantidad de bytes a escribir o leer.



Lectura desde archivo en C

```
int fread( &reg, sizeof (Alumno), 1, p)
```

Lee del inicio del archivo apuntado por **p**, una cantidad de 120 bytes (el tamaño de un registro de Alumno) y los almacena en la variable **reg** de tipo Alumno.

Puntero FILE. Tipo de dato que "apuntará" a un archivo en disco

Cantidad registros a leer a la vez. Generalmente será de a uno por vez.

Cantidad de bytes de cada registro a leer. Hipotéticamente 120 bytes por registro.

Registro de tipo Alumno donde se va a guardar la información proveniente del archivo.

Escritura desde archivo en C

```
int fwrite( &reg, sizeof (Alumno), 1, p)
```

Escribe al final del archivo apuntado por **p**, un registro de tamaño 120 bytes (el tamaño de un registro de Alumno) proveniente desde la dirección de **reg**.

Puntero FILE. Tipo de dato que "apuntará" a un archivo en disco

Cantidad registros a escribir a la vez. Generalmente será de a uno por vez.

Cantidad de bytes de cada registro a escribir. Hipotéticamente 120 bytes por registro.

Registro de tipo Alumno de cual proviene la información a guardar en el archivo.

Puntero FILE

Un tipo de dato que nos permitirá "conectar" nuestro programa a un archivo en el dispositivo que indiquemos.

```
FILE *p;
p = fopen("archivo.dat", "rb");
/* Todas las acciones a realizar con el archivo */
fclose(p);
```

Handwritten annotations:

- ↓ (points to `FILE`)
- Met. apertura (points to `fopen`)
- read (points to `r` in `"rb"`)
- binary (points to `b` in `"rb"`)
- a - append (points to `a` in `"a"`)
- w - Write (points to `w` in `"w"`)

fopen

Una función que nos permite conectar un archivo de datos a un puntero FILE. Además, le indicamos el modo de apertura.

```
p = fopen("archivo.dat", "rb");
```

Modo	Nombre	Descripción
wb	write-binary	Destruye el archivo a 0 bytes y lo abre como escritura. Si no existe el archivo, lo crea.
rb	read-binary	Abre el archivo como lectura desde el byte 0. No admite escritura.
ab	append-binary	Abre el archivo desde el final y lo abre como escritura. No admite modificar lo existente, sólo agregar. Si no existe el archivo, lo crea.
rb+	read-binary plus overwrite	Abre el archivo desde el byte 0 y lo abre como escritura. Admite modificar registros existentes.

fclose

Una función que nos permite liberar la conexión de un archivo de datos a un puntero FILE.

```
fclose(p);
```

- Es clave porque no se puede abrir el mismo archivo más de una vez simultáneamente.

- Confirma los datos en aperturas de escritura.

Ejemplo