Archivos Binarios

El lenguaje C ofrece elementos de programación (Tipos e instrucciones) que permiten implementar el manejo de archivos binarios en los programas.

Las sentencias base para el manejo de archivos binarios en lenguaje C son:

Abrir archivo **fopen** idVarArch = fopen(nombreArch, modoApertura);

Modos: **wb** (Crear archivo, escritura) **rb** (leer archivo)

ab (Anexar información archivo) **rb+** (Lectura/escritura)

Escritura ***fwrite*** ***<noBlq> fwrite(&varDatos, tamBlq, noBlq, FILE *arch);***

varDatos Variable que contiene los datos a transferir al archivo.

tamBlq Cantidad de bytes a transferir al archivo.

int → 4 bytes
float → 4 bytes
char → 1 byte
cad20→ 21 bytes
estructura→ suma de los tipos individuales.

Estos tamaños pueden variar de sistema a sistema, afortunadamente se cuenta con una instrucción del lenguaje que retorna el tamaño de cada tipo, esta instrucción es:

<size_t> sizeof(tipoDato) => size_t ≈ unsigned int

```
tamInt = sizeof(int);
tamFloat = sizeof(float);
tamChar = sizeof(char);
tamPersona = sizeof(PER);
```

noBlq No. de bloques a transferir, con esta instrucción se pueden almacenar varios bloques sin necesidad de un ciclo.

arch id de la variable asociada al archivo destino.

<noBlq> Esta instrucción retorna el No. de bloques transferidos.

Ejemplo: fwrite(&dato, 4, 1, arch);

fwrite(&dato, sizeof(int), 1, arch);

fwrite(&per, sizeof(PER), 1, archPer);

fwrite(arr, sizeof(int), n, arch);

fwrite(arrPer, sizeof(PER), n, archPer);

Lectura ***fread*** ***<nBlq> fread(&varDatos, tamBlq, noBlq, FILE *arch);***

varDatos Variable que recibirá los datos provenientes del archivo.

tamBlq Cantidad de bytes a recuperar del archivo.

noBlq No. de bloques a recuperar del archivo. Con esta instrucción se pueden recuperar varios bloques sin necesidad de un ciclo.

arch id de la variable asociada al archivo fuente.

<noBlq> Esta instrucción retorna el No. de bloques recuperados, este valor nos sirve como detección de fin de archivo.

Ejemplo: fread(&dato, 4, 1, arch);

fread (&dato, sizeof(int), 1, arch);

fread (&per, sizeof(PER), 1, archPer);

fread (arr, sizeof(int), n, arch);

fread (arrPer, sizeof(PER), n, archPer);

Ubicar puntero en una pos. *(en bytes)* Específica. ***fseek*** ***fseek (FILE *arch, offset, origin); => SEEK_SET SEEK_CUR SEEK_END***

Retorna la posición actual (*en bytes*) del puntero de archivo. ***ftell*** ***long int ftell(FILE *arch);***

Cerrar archivo **fclose** ***fclose(idArch);***