

ARCHIVOS

ALGORITMOS Y PROGRAMACIÓN I (2008-2009)

Introducción

Un archivo puede definirse como una secuencia de elementos del mismo tipo, que residen generalmente en memoria auxiliar. Los archivos son utilizados cuando el volumen de datos a procesar es grande y no cabe en la memoria central del computador.

En muchas ocasiones es preciso disponer de algún mecanismo que permita almacenar de forma permanente ciertos datos de los programas. Por ejemplo, un programa que gestione la contabilidad de una empresa necesita información inicial (balance hasta ese momento, lista de compras, lista de ventas, entre otros). De igual forma, genera una serie de informaciones que deben ser almacenadas cuando el programa finaliza. Desde el punto de vista del hardware, hay diferentes dispositivos para almacenar información de forma permanente: discos duros, unidades de cinta, CDs, disquetes, etc. Para un programador, el dispositivo físico que se use carece de importancia. Los programas deben funcionar sin importar el dispositivo de almacenamiento.

Para ocultarle al programador los detalles específicos de cada dispositivo hardware, se usa el concepto de archivo. Un archivo es un objeto abstracto sobre el cual se puede leer y escribir información.

Existen dos tipos fundamentales de archivos: archivos de texto y archivos binarios. En los archivos de texto almacena la información como una secuencia de códigos ASCII. Los archivos binarios, almacenan datos que no son interpretables como texto (números, imágenes, etc.).

CARACTERÍSTICAS BÁSICAS DE LOS ARCHIVOS

- Los archivos son utilizados cuando se desea que los datos puedan recuperarse aún después de haber apagado la máquina y también cuando se manejan grandes volúmenes de información.
- Residen en memoria auxiliar, la cual generalmente, es más económica, grande, pero lenta que la memoria principal del computador.
- Los elementos que almacena un archivo pueden ser elementales (texto, números) o estructurados (arreglos, registros). Para propósitos del curso, sólo se puede leer un archivo elemento por elemento.
- La cantidad de componentes que constituyen el archivo, no se conoce previamente. Si tiene *cero elementos* se dice que es un *archivo vacío*.
- Los archivos más comunes son los archivos *secuenciales*. Los elementos de un archivo secuencial están dispuestos uno tras otro, y para acceder un elemento en particular, hay que pasar por todos los anteriores. Por esos se les llama de acceso secuencial, a diferencia de los arreglos que son de acceso aleatorio. Sin embargo, los elementos del archivo también pueden ser

extraídos de manera directa, pero este tipo de acceso no es objetivo del curso.

DECLARACIÓN DE ARCHIVOS

Consiste en indicar el nombre de la variable que será utilizada en el algoritmo para hacer Referencia al archivo.

Declaración por Variable: se declara la variable tipo archivo como si se tratara de una variable de tipo de dato simple.

archivo <identificador_archivo>

Ejemplos: archivo A archivo B, X

Declaración por Tipo: al igual que los arreglos y los registros, se antecede la palabra reservada **Tipo** a la declaración y luego se declaran las variables del tipo.

Tipo archivo <Identificador_Tipo>

Var <Identificador_Tipo> <Identificador_archivo>

Ejemplo: Tipo archivo estudiante

Var estudiante est

OPERACIONES BÁSICAS EN ARCHIVOS SECUENCIALES

- Abrir el archivo
- Cerrar el archivo
- Verificar fin de archivo (FDA)
- Leer del archivo
- Escribir en el archivo

Estas operaciones se consideran operaciones primitivas y reservadas del pseudo-código.

a. Abrir el archivo: Ubica y prepara el archivo para poder trabajar con él.

AbrirArchivo (identificador_archivo, ruta_del_archivo, <Argumentos>)

Donde:

<Argumentos> es uno o más de las siguientes palabras reservadas:

- Escritura: indica que el archivo se abre de solo escritura
- Lectura: indica que el archivo se abre de solo lectura
- Añadir: indica que el archivo se abre de escritura pero todo lo que se escriba se añade al final del archivo

- Texto: indica que el archivo a abrir es un archivo de texto
- Binario: indica que el archivo a abrir es un archivo binario (un archivo diferente a un archivo de texto).

Los argumentos pueden ser combinados con el operador lógico y. Por ejemplo:
AbrirArchivo(A, 'prueba.txt', Lectura y Texto)

Nota: El archivo se abre una sola vez en el algoritmo, al principio del mismo y debe ser cerrado al finalizar el mismo

b. Cerrar el archivo: Cuando se desea dejar de trabajar con un archivo, debe cerrarse para que esté disponible para otras aplicaciones. Sólo los archivos que están abiertos pueden cerrarse. Si un archivo no se cierra se puede perder información en el mismo.

CerrarArchivo(identificador_archivo) Ejemplo: CerrarArchivo(A)

c. Fin de archivo (FDA, EOF): indica si no hay más elementos para leer en el archivo. Sólo se usa para archivos de lectura. Retorna verdadero si se alcanzó el fin del archivo y falso en caso contrario.

FDA(identificador_archivo)

d. Leer del archivo: Lee un elemento del archivo.

LeerArchivo(identificador_archivo, identificador_elemento)

e. Escribir en el archivo: escribe un elemento en el archivo.

EscribirArchivo(identificador_archivo, identificador_elemento)

PASOS PARA RECORRER UN ARCHIVO

Los pasos para recorrer un archivo, se pueden resumir en forma general en los siguientes

1. Declarar los archivos a utilizar y las variables que almacenarán elementos del archivo
2. Abrir el archivo de entrada (que solo se podrá leer) como tipo lectura.
3. Validar si FDA devuelve verdadero. Si esto ocurre mostrar un mensaje indicativo e ir al paso 7 para finalizar el ciclo. Si es falso ir al paso 5 para continuar el recorrido.
4. Abrir el o los archivos de salida a utilizar e inicializar las variables necesarias en el procesamiento dentro del ciclo.
5. Recorrer con un ciclo Repetir o Mientras:

┌ Repetir
 └ Hasta FDA

┌ Mientras no (FDA) hacer
 └ fmientras

6. Dentro de este ciclo la primera operación es la de lectura de cada registro. Si se realizan escrituras en el archivo de salida, estas van en este mismo ciclo.
7. Al finalizar el ciclo, Cerrar el archivo de salida y mostrar resultados.
8. Cerrar el archivo de entrada