

Actividad 1.

Rodríguez Tabares Juan

Ingeniería en Computación

Estructuras de datos II

Centro Universitario de Ciencias Exactas e Ingenierías

Universidad de Guadalajara

Abstract

Este documento comprende la realización de la primera actividad de la asignatura Estructura de Datos II llevada a cabo en el calendario 2020B.

1 ¿Cuál es la ventaja fundamental de usar archivos?

La ventaja que mas predomina de todas seria el hecho de que se guardan los datos con los que estamos trabajando en lugar de que se eliminen al termino de cualquier programa.

Esto ayuda bastante ya que si requerimos de algun dato dentro de este solo hacemos la consulta, al contrario si no utilizamos archivos perdemos los datos y tendremos un problema al momento de querer recuperar datos.

2 ¿Qué es un flujo?

Los flujos proporcionan canales de comunicación entre archivos y programas. Así, el flujo de entrada estándar permite que un programa lea datos del teclado, el flujo de salida estándar permite que un programa imprima datos a la pantalla.

3 En términos generales ¿Qué Operaciones se repiten a la hora de trabajar con cualquier fichero y en qué orden?

Apertura.- Antes de acceder a un fichero, tanto para consultar como para actualizar su información, es necesario abrirlo. Esta operación se debe realizar previamente a las operaciones de lectura o escritura.

Escritura (modificación).- Consiste en actualizar el contenido del fichero bien añadiéndole nuevos datos o borrando parte de los que contenía.

Lectura (consulta).- Esta operación consiste en leer la información contenida en fichero sin alterarla.

Cierre.- Cuando se ha terminado de consultar o modificar un fichero, del mismo modo que se tuvo que abrir para realizar alguna operación de lectura/escritura sobre él.

El orden es como fueron colocados anteriormente

4 ¿Que es el almacenamiento primario?

También conocida como almacenamiento principal o la memoria, es el área en un equipo de cómputo en el que se almacenan los datos para un rápido acceso por el procesador de la computadora.

5 ¿Que es el almacenamiento secundario?

La memoria secundaria, memoria auxiliar, memoria periférica o memoria externa, también conocida como almacenamiento secundario, es el conjunto de dispositivos y soportes de almacenamiento de datos que conforman el subsistema de memoria de la computadora, junto con la memoria primaria o principal.

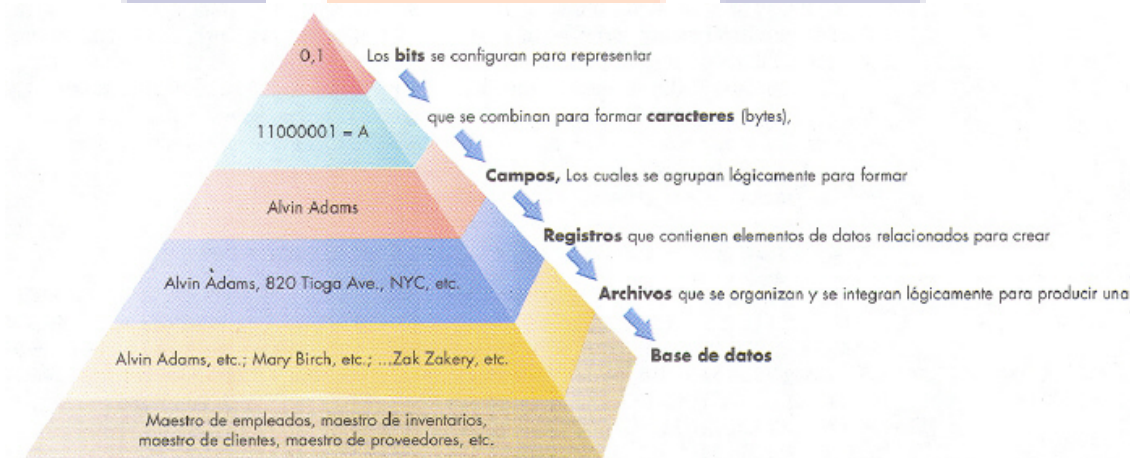
6 ¿Qué es un archivo?

Es una colección de registros los cuales están conformados por datos que consisten en información que se desea guardar para su próximo uso en futuras acciones

7 ¿Qué es un registro?

Es un conjunto de campos también conocido como un tipo de dato estructurado, en el cual se tienen datos que pertenecen a una misma entidad

8 Dibuja la estructura jerárquica de datos



9 ¿Cuáles son los tipos de soporte utilizados en la gestión de archivos y en que consisten?

El soporte es el medio físico donde se almacenan los datos. Los tipos de soportes utilizados en la gestión de archivos son:

- **Soportes secuenciales.**- son aquellos en los que los registros están escritos unos a continuación de otros y para acceder a un determinado registro n se necesita pasar por los registros $n-1$ registros anteriores.

La secuencia puede corresponder al orden físico de los registros en el archivo o bien al orden de claves (ascendente o descendente) de los registros.

- **Soportes Direccionalles.**- se estructuran de modo que las informaciones registradas se pueden localizar directamente por su dirección y no se requiere pasar por los registros precedentes.

10 ¿Cuáles son los tipos de acceso a los registros de un archivo?

Acceso Secuencial. Exige el tratamiento de elemento, para esto es necesario una exploración secuencial comenzando desde el primer momento (Pascal permite este acceso).

Secuenciales: archivo de texto que debe ser leído del principio hasta el final.

Acceso Directo. Permite procesar o acceder a un elemento determinado y referencia directamente por su posición en el soporte de almacenamiento (Turbo Pascal permite este acceso).

Aleatorios: es un archivo con registros de un mismo largo. Un programa puede acceder directamente cualquier registro sin tener que leer los registros previos.

Binarios: es un archivo que lee byte por byte sin asumir ninguna estructura. Los archivos Binarios no son un nuevo tipo de archivo, pero sí una nueva forma de manipular cualquier tipo de archivo. Las técnicas de archivo binarios permiten leer o cambiar cualquier byte de un archivo. Son herramientas extremadamente potentes, pero como toda herramienta potente debe manejarse con cuidado

11 ¿Cuáles son las tres organizaciones fundamentales de los archivos?

1. Organización Secuencial.
2. Organización Directa o Aleatoria (Random).
3. Organización Secuencial Indexada (Indexed).

12 Menciona las operaciones fundamentales que conciernen a los registros de un archivo

- Creación.
- Consulta.
- Actualización (Altas, Bajas, Modificación, Consulta).
- Clasificación.
- Reorganización.
- Destrucción (Borrado).
- Reunión/ Fusión
- Rotulación / Estallido

13 ¿Qué es un puntero a un archivo secuencial? ¿Cuál es la relación entre un puntero a un archivo secuencial y un área de buffer?

El puntero a un archivo es el hilo común que unifica el sistema de E/S con buffer. Un puntero a un archivo es un puntero a una información que define varias cosas sobre él, incluyendo el nombre, el estado y la posición actual del archivo. En esencia identifica un archivo específico y utiliza la secuencia asociada para dirigir el funcionamiento de las funciones de E/S con buffer. Un puntero a un archivo es una variable de tipo puntero al tipo FILE que se define en STDIO.H. Un programa necesita utilizar punteros a archivos para leer o escribir en los mismos. Para obtener una variable de este tipo se utiliza una secuencia como esta: **FILE *F;**

14 ¿Qué se entiende por abrir un archivo? ¿Cómo se realiza?

En programación abrir un archivo podemos entender que se necesita ver el contenido o modificar el contenido que este tiene.

Existen varias formas de abrir un archivo pero la más conocida es la función:

```
FILE *fopen(const char nombrearchivo, const char modo);
```

15 Describe lo que hace la función fopen, sus modos de apertura y la descripción de cada uno de ellos

La función fopen() abre una secuencia para que pueda ser utilizada y la asocia a un archivo. Su prototipo es:

```
FILE *fopen(const char nombrearchivo, const char modo);
```

Donde nombrearchivo es un puntero a una cadena de caracteres que representan un nombre válido del archivo y puede incluir una especificación del directorio. La cadena a la que apunta modo determina cómo se abre el archivo. La siguiente tabla muestra los valores permitidos para modo.

Modo	Significado
r	Abre un archivo de texto para lectura.
w	Crea un archivo de texto para escritura.
a	Abre un archivo de texto para añadir.
rb	Abre un archivo binario para lectura.
wb	Crea un archivo binario para escritura.
ab	Abre un archivo binario para añadir.
r+	Abre un archivo de texto para lectura / escritura.
w+	Crea un archivo de texto para lectura / escritura.
a+	Añade o crea un archivo de texto para lectura / escritura.
r+b	Abre un archivo binario para lectura / escritura.
w+b	Crea un archivo binario para lectura / escritura.
a+b	Añade o crea un archivo binario para lectura / escritura.

16 ¿Cuál es el propósito de la función fclose? ¿Debe ser llamada esta función dentro de un programa que usa archivos?

La función fclose() cierra una secuencia que fue abierta mediante una llamada a fopen(). Escribe toda la información que todavía se encuentre en el buffer en el disco y realiza un cierre formal del archivo a nivel del sistema operativo. Un error en el cierre de una secuencia puede generar todo tipo de problemas, incluyendo la pérdida de datos, destrucción de archivos y posibles errores intermitentes en el programa. El prototipo de esta función es:

```
int fclose(FILE *F);
```

El no utilizar esta función puede llevar a provocar la pérdida de datos.

17 Describe que hace la función ferror

La función ferror() determina si se ha producido un error en una operación sobre un archivo. Su prototipo es:

```
int ferror(FILE *F);
```

Donde F es un puntero a un archivo válido. Devuelve cierto si se ha producido un error durante la última operación sobre el archivo. En caso contrario, devuelve falso. Debido a que cada operación sobre el archivo actualiza la condición de error, se debe llamar a `ferror()` inmediatamente después de la operación de este tipo; si no se ase así, el error puede perderse.

18 Describe que hace la función feof

Cuando se abre un archivo para entrada binaria, se puede leer un valor entero igual de la marca EOF. Esto podría hacer que la rutina de lectura indicase una condición de fin de archivo aún cuando el fin físico del mismo no se haya alcanzado. Para resolver este problema, C incluye la función `feof()`, que determina cuando se ha alcanzado el fin del archivo leyendo datos binarios. La función tiene el siguiente prototipo:

```
int feof(FILE *F);
```

Su prototipo se encuentra en `STDIO.H`. Devuelve cierto si se ha alcanzado el final del archivo, en cualquier otro caso, 0. Por supuesto, se puede aplicar este método a archivos de texto también.

References

- [1] VIRGINIA VALERO CANTÚ. (2012). TEMA 7. 19 DE SEPTIEMBRE DEL AÑO 2020, DE UNIVERSIDAD DE CONCEPCIÓN SITIO WEB: [HTTPS://W3.UAL.ES/ ABECERRA/ID/ARCHIVOS.PDF](https://w3.ual.es/ABECERRA/ID/ARCHIVOS.PDF)
- [2] ROSA ERENDIRA REYES LUNA. (2015). ORGANIZACIÓN DE ARCHIVOS. 19 DE SEPTIEMBRE DEL AÑO 2020, DE CENTRO UNIVERSITARIO UAEM ZUMPANGO SITIO WEB: [HTTP://RI.UAEMEX.MX/OCA/VIEW/20.500.11799/34751/1/SECME-19554.PDF](http://ri.uaemex.mx/OCA/VIEW/20.500.11799/34751/1/SECME-19554.PDF)