Clase 1- Bases de datos

jueves, 9 de marzo de 2023 14:47

- Listas, vectores, árboles → No persistentes. (Apago la compu y los pierdo)

Base de datos → Conjunto de archivos → Almacenamiento de datos
→ Colección de datos relacionados → Datos persistentes (se guardan cuando apago en memoria secundaria)

Propiedades BD → colección coherente de datos → Lógica detrás del diseño de la BD.
Soportes físicos en archivos en dispositivos persistentes de datos

* Archivos, ¿Qué es? → colección de registros (datos) guardados en memoria secundaria.
→ Aspecto común y propósito particular.

Hardware → Memoria principal - secundaria
→ ≠ tiempo de acceso (ram nanosegundos, segundo. milisegundos)

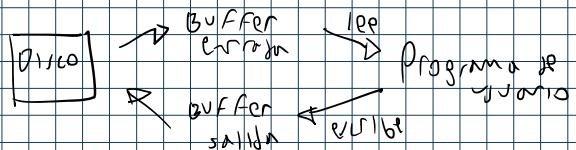
optimizar acceso a memoria secundaria programando

Organización de archivos → Secuencia de bytes.
- Difícil identificar contenido y fin de cada dato.
- Archivos de texto
→ secuencia de registros o campos
- Campos: unidad + pequeña
- Registro: conjunto de campos agrupados que definen un elemento

Accesos a archivos → secuencial física: uno tras otro y en el orden en que están puestos
→ secuencial lógica (lógico): Acceso según orden de otra estructura.
(Ej: a cada alumno por dni, los accedo x orden este)
→ Directo: voy de una sin tener q' pasar por los anteriores.

Tipos de accesos → secuencial: registros son accesibles según alguna clave
→ serie: Acceso solo luego de haber procesado el anterior
→ Directo: Acceso de una

Buffers
- Mem. intermedia e/ un archivo y un programa.
- Sirve de buffer cuando pasa mucho a secundaria.
- Datos residen en buffer hasta ser almacenados o tomados
- SO se encarga de manipularlos



operaciones básicas con archivos:

Dato a dos niveles: Físico: Cuando se almacenan en mem secundaria.

Lógico: Cuando lo veo yo → Crear/abrir/read, write/ eof/seek (acceder a pos)

Sin archivos, n-1 archivos

- Variable
  - Var archivo: file of Tipo_de_dato;
- Tipo
  - Type archivo: file of Tipo_de_dato;
  - Var arch: archivo

} mejor

Ejemplos

```
Type emple = record
    nombre: string [20];
    direccion: string [40];
    edad: integer;
    end;
numero = file of integer;
empleado = file of emple;
Var arch_num: numero;
Var arch_emp: empleado,
```

→ Archivo de campo
→ Archivo de registros.

```
Program Generar_Archivo;
  type archivo = file of integer; {definición del tipo de dato para el archivo }
  var arc_logico: archivo;        {variable que define el nombre lógico del archivo}
      nro: integer;               {nro será utilizada para obtener la información de teclado}
      arc_fisico: string[12];     {utilizada para obtener el nombre físico del archivo desde teclado}
begin
    write( 'Ingrese el nombre del archivo:' );
    read( arc_fisico ); { se obtiene el nombre del archivo}
    assign( arc_logico, arc_fisico );
    rewrite( arc_logico ); { se crea el archivo }
    read( nro ); { se obtiene de teclado el primer valor }
    while nro <> 0 do begin
        write( arc_logico, nro ); { se escribe en el archivo cada número }
        read( nro );
    end;
    close( arc_logico );  { se cierra el archivo abierto oportunamente con la instrucción rewrite }
end.
```

Rewrite (nombre_logico);
- De solo escritura (creación)

→ crear y abrir

Reset (nombre_logico);
- Lectura Escritura (apertura)

→ recorrerlo, editarlo, etc.

Nombre lógico representa una variable de tipo archivo sobre la que se realizó la asignación.

Close(nombre_logico);
- Cierre de archivo
- Esta instrucción indica que no se va a trabajar más con el archivo. Significa poner una marca de EOF (end of file) al final del mismo.

Si hago rewrite sobre algo ya escrito lo pierdo.

Read(nombre_logico, variable);

Write(nombre_logico, variable);

Estas operaciones leen y/o escriben sobre los buffers relaciona-dos a los archivos

No se realizan directamente sobre la memoria secundaria

En ambos casos la variable debe ser del mismo tipo que los elementos que se declararon como parte del archivo

30 Archivos → Ej 3 Modificación de Datos de un archivo (continuación)

```
Procedure actualizar (Var Emp:empleados); {se recibe como parám. por referencia}
  var E: registro;
  begin
    Reset( Emp );
    while not eof( Emp ) do begin
        Read( Emp, E);          → lee dato
        E.salario:=E.salario * 1.1;   → modifica reg.
        Seek( Emp,  filepos(Emp) -1 );   → busca la pos → pq read avanza 1 implícitamente.
        Write( Emp, E );
    end;
    close( Emp );
end;
```

FILE POS ARRANCA EN 1, RESTARLE 1 PARA MODIFICAR COSAS

- Assign: Asigna nom lógico a físico.
  - Puede ir la ruta si una vez asign (arch, ruta)

Hay q' cerrar siempre pq sino las últimas modificaciones al arch. no se guardan.

Read y Write mueven implícitamente el puntero

Binario a txt

```
2: begin
       Reset(arch); {abre archivo binario}
     Rewrite(carga);  {crea archivo de texto, se
   utiliza el mismo de opcion 1 a modo ejemplo}
 while (not eof(arch)) do begin
       Read(arch, votos); {lee votos del arch binario}
       With votos do
       WriteLn(codProv:5,     codLoc:5,     nroMesa:5,
 cantVotos:5,   desc:5);   {escribe en pantalla el
 registro}
       With votos do
       WriteLn(carga,'   ',codProv,'   ',codLoc,'   ',
 nroMesa,'   ',cantVotos,'   ',desc);  {escribe en el
 archivo texto los campos separados por el carácter
 blanco}
```

Txt a binario

```
Case opc of      Que sucede cuando tengo más de
     1: begin    String en un archivo de texto?
       Write('Nombre del archivo de carga: ');
       ReadLn(nomArch2);
       Assign(carga, nomArch2);
       Reset(carga); {abre archivo de texto con datos}
       Rewrite(arch); {crea nuevo archivo binario}
       while (not eof(carga)) do begin
         With votos do ReadLn(carga, codProv, codLoc,
nroMesa, cantVotos, desc); {lectura de archivo de texto}
             Write(arch, votos); {escribe binario}
       end;
       Write('Archivo cargado.');
       ReadLn;
       Close(arch);   Close(carga)   {cierra los dos
archivos} end;
```