

Fundamentos de Organización de Datos

1

Curso 2015

Algorítmica clásica sobre archivos

Archivo maestro: Resume información sobre el dominio de un problema específico.

Ejemplo: El archivo de productos de una empresa.

Archivo detalle: Contiene movimientos realizados sobre la información almacenada en el maestro.

Ejemplo: Las ventas de los productos de la empresa realizadas en un día particular.

Algorítmica clásica sobre archivos

Importante: Analizar las precondiciones que en cada caso particular. Los algoritmos a desarrollar deben tener en cuenta estas precondiciones, caso contrario determina la falla de su ejecución.

Actualización de un archivo maestro con un archivo detalle - Precondiciones

- Existe un archivo maestro.
- Existe un único archivo detalle que modifica al maestro.
- Cada registro del detalle modifica a un solo registro del maestro.
- No todos los registros del maestro son necesariamente modificados.
- Cada elemento del maestro que se modifica, es alterado por uno y solo un elemento del archivo detalle.
- Ambos archivos están ordenados por igual criterio.

Ejemplo: Definición de tipos

```
type
  producto = record
    cod: string[4];
    descripcion: string[30];
    pu: real;
    stock: integer;
  end;
  venta_prod = record
    cod: string[4];
    cant_vendida: integer;
  end;

detalle = file of venta_prod;
maestro = file of producto;
```

Ejemplo: variables y operaciones

var

```
regm: producto;  
regd: venta_prod;  
mael: maestro;  
det1: detalle;
```

begin *// Inicio del programa*

```
assign (mael, 'maestro');  
assign (det1, 'detalle');  
reset  (mael);  
reset  (det1);
```

Ejemplo: algoritmo

{Bucle Archivo Detalle}

```
while not (eof(det1)) do begin  
    read(mael, regm); // Lectura Archivo Maestro  
    read(det1, regd); // Lectura Archivo Detalle
```

*{Se busca en el maestro el producto del
detalle}*

```
while (regm.cod <> regd.cod) do  
    read (mael, regm);
```

Ejemplo: algoritmo

{se modifica el stock del producto con la cantidad vendida de ese producto}

```
regm.stock := regm.stock - regd.cant_vendida;
```

{se reubica el puntero en el maestro}

```
seek(mael, filepos(mael)-1);
```

{se actualiza el maestro}

```
write(mael, regm);
```

```
end; // Fin del Bucle Archivo Detalle
```

```
close(det1);
```

```
close(mael);
```

```
end.
```


Actualización de un archivo maestro con un archivo detalle

- Existe un archivo maestro.
- Existe un único archivo detalle que modifica al maestro.
- Cada registro del detalle modifica a un registro del maestro.
- No todos los registros del maestro son necesariamente modificados.
- Cada elemento del archivo maestro puede no ser modificado, o ser **modificado por uno o más elementos del detalle**.
- Ambos archivos están ordenados por igual criterio.

Actualización de un archivo maestro con un archivo detalle

```
type
  producto = record
    cod: string[4];
    descripcion: string[30];
    pu: real;
    stock: integer;
  end;
  venta_prod = record
    cod: string[4];
    cant_vendida: integer;
  end;

detalle = file of venta_prod;
maestro = file of producto;
```

Ejemplo: variables y operaciones

Var

```
regm: producto;  
regd: venta_prod;  
mael: maestro;  
det1: detalle;  
cod_actual: string[4];  
tot_vendido: integer;
```

begin *// Inicio del programa*

```
    assign (mael, 'maestro');  
    assign (det1, 'detalle');  
    reset (mael);  
    reset (det1);
```

Ejemplo: algoritmo

{Bucle Archivo Detalle}

```
while not (eof(det1)) do begin  
    read(mael, regm); // Lectura Archivo Maestro  
    read(det1, regd); // Lectura Archivo Detalle
```

*{Se busca en el maestro el producto del
detalle}*

```
while (regm.cod <> regd.cod) do  
    read (mael, regm);
```

Ejemplo: algoritmo

{se totaliza la cantidad vendida del detalle}

```
cod_actual := regd.cod;
```

```
tot_vendido := 0;
```

```
while (regd.cod = cod_actual) do begin
```

```
    tot_vendido:=tot_vendido+regd.cant_vendida;
```

```
    read(det1,regd);
```

```
end;
```

{se actualiza el stock del producto con la cantidad vendida del mismo}

```
regm.stock := regm.stock - tot_vendido;
```

Ejemplo: algoritmo

{se reubica el puntero en el maestro}

```
seek (mae1, filepos(mae1)-1);
```

{se actualiza el maestro}

```
write(mae1, regm);
```

```
end;
```

```
close(det1);
```

```
close(mae1);
```

```
end.
```

Ejemplo: Diferencia entre los ejemplos

¿Diferencia entre este ejemplo y el anterior?

Se agrega una iteración que permite agrupar todos los registros del detalle que modificarán a un elemento del maestro.

¿Inconvenientes de esta solución?

La segunda operación *read* sobre el archivo detalle se hace sin controlar el fin de datos del mismo. Podría solucionarse agregando una selección (if) que permita controlar dicha operación. Pero cuando finaliza la iteración interna, al retornar a la iteración principal se lee otro registro del archivo detalle, perdiendo de esa forma un dato ya leído.

Actualización de un archivo maestro con un archivo detalle

```
const valoralto = '9999';  
type str4 = string[4];  
  producto = record  
    cod: str4;  
    descripcion: string[30];  
    pu: real;  
    stock: integer;  
  end;  
  venta_prod = record  
    cod: str4  
    cant_vendida: integer;  
  end;  
  detalle = file of venta_prod;  
  maestro = file of producto;
```


Ejemplo

```
var
```

```
    regm: producto;  
    regd: venta_prod;  
    mael: maestro;  
    det1: detalle;  
    total: integer;  
    aux: str4;
```

```
procedure leer (var archivo:detalle; var dato:venta_prod)  
begin  
    if (not eof(archivo)) then  
        read (archivo, dato)  
    else  
        dato.cod:= valoralto;  
end;
```

Ejemplo

```
{programa principal}  
begin  
    assign (mae1, 'maestro');  
    assign (det1, 'detalle');  
    reset (mae1);  
    reset (det1);  
    read (mae1, regm);  
    leer (det1, regd);
```

Ejemplo

*{se procesan todos los registros del archivo
detalle}*

```
while (regd.cod <> valoralto) do begin  
    aux := regd.cod;  
    total := 0;
```

*{se totaliza la cantidad vendida de
productos iguales en el archivo de detalle}*

```
while (aux = regd.cod) do begin  
    total := total + regd.cant_vendida;  
    leer(det1, regd);  
end;
```

Ejemplo

```
{se busca en el maestro el producto del detalle}
while (regm.cod <> aux) do
    read (mael, regm);
{se modifica el stock del producto con la cantidad
total vendida de ese producto}
    regm.cant := regm.cant - total;
{se reubica el puntero en el maestro}
    seek (mael, filepos(mael)-1);
{se actualiza el maestro}
    write(mael, regm);
    {se avanza en el maestro}
    if (not eof (mael)) then
        read(mael, regm);
    end;
close (det1);
close (mael);
end.
```

¿PREGUNTAS?