

Fundamentos de Organización de Datos

Archivos
Merge

Algorítmica clásica sobre archivos

Merge

Proceso mediante el cual se genera un nuevo archivo a partir de otros archivos existentes.

Ejemplo – Merge

```
program ejemplo;  
  const valor_alto = 999999;  
  type  
    producto = record  
      codigo: integer;  
      descripcion: string[30];  
      pu: real;  
      cant: integer;  
    end;  
  arc_productos = file of producto;
```

var

```
det1, det2, det3, mae: arc_productos;  
min, regd1, regd2, regd3: producto;
```

procedure leer (**var** archivo: arc_productos;
 var dato: producto);

begin

if (**not**(**EOF**(archivo))) **then**

read (archivo, dato)

else

 dato.codigo := valor_alto;

end;

```
procedure minimo(var det1, det2, det3: arc_productos;  
                 var r1, r2, r3, min: producto);  
begin  
    if (r1.codigo<=r2.codigo) and (r1.codigo<=r3.codigo) then begin  
        min := r1;  
        leer(det1, r1);  
    end  
    else  
        if (r2.cod <= r3.cod) then begin  
            min := r2;  
            leer(det2, r2);  
        end  
        else begin  
            min := r3;  
            leer(det3, r3)  
        end;  
end;
```

{programa principal}

begin

assign (mae, 'maestro');

assign (det1, 'detalle1');

assign (det2, 'detalle2');

assign (det3, 'detalle3');

rewrite (mae);

reset (det1);

reset (det2);

reset (det3);

leer (det1, regd1);

leer (det2, regd2);

leer (det3, regd3);

minimo (det1, det2, det3,
 regd1, regd2, regd3, min);

*{se procesan todos los registros de los
archivos detalle}*

```
while (min.codigo <> valoralto) do begin  
    write (mae, min);  
    minimo (det1, det2, det3,  
            regd1, regd2, regd3, min);  
  
end;  
close (det1);  
close (det2);  
close (det3);  
close (mae);  
end.
```

Otra variante – Productos repetidos en los archivos detalles

```
while (min.codigo <> valoralto) do begin  
  aux:= min;  
  total := 0;  
  while (min.codigo = aux.codigo) do begin  
    total := total + min.cant;  
    minimo (det1, det2, det3,  
            regd1, regd2, regd3, min);  
  end;  
  aux.cant := total;  
  write (mae, aux);  
end;
```