



Resolución de Problemas y Algoritmos

Clase 9:
Problemas que utilizan entrada y salida con archivos secuenciales.



Dr. Alejandro J. García
<http://cs.uns.edu.ar/~ajg>



Departamento de Ciencias e Ingeniería de la Computación
 Universidad Nacional del Sur
 Bahía Blanca - Argentina

información adicional

- El examen parcial del 29 de abril se tomará en el aula 8 del edificio "rosa" en el complejo de Palihue.
- El examen será de 14 a 18hs.
- A las 18 hs el aula será ocupada por otra cátedra.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 2

Almacenamiento en memoria

- **Memoria RAM:** (Random Access Memory) tiene los programas en ejecución. El procesador debe accederla lo más rápido posible, y por ello es de alto costo (\$100 1Gb en 2014, ej: 4Gb a \$400). Por su tecnología es volátil (los datos no perduran al cortar la energía del sistema)
- Los valores de las variables de tipos integer, char, real y boolean se almacenan en RAM.
- **Memoria secundaria:** de menor costo (\$1 x 1Gb en 2014, ej: 1Tb \$1000) y por ello de menor velocidad. Por su tecnología (magnética, óptica, flash, etc.) permite que los valores perduren aún cuando no tenga energía eléctrica.
- Los valores de los archivos están en memoria secundaria.

Si tiene más curiosidad vea las referencias al final ©

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 3

Archivos (repaso)

Program ejemplo;
VAR F: FILE OF integer;

- Dado que una variable de tipo FILE es una sucesión de componentes del mismo tipo, es usual usar la siguiente **representación gráfica:**

Primer elemento.

→

En F1 cada componente es de tipo integer.

F:

11	-2	5	-12
----	----	---	-----

←

Último elemento.

- Aunque la capacidad de un archivo es **potencialmente** infinita, en cualquier momento dado, el archivo tendrá un número finito de componentes.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 4

Primitivas de Pascal para archivos secuenciales ▶

- **assign(F,N):** vincula el manejador de archivo F con el nombre N que usará el archivo en el Sistema Operativo.
- **rewrite(F):** crea un archivo nuevo con el nombre N vinculado a F (si ya existe otro N lo sobre-escribe).
- **write(F,e):** en un archivo F creado con **rewrite**, escribe el valor de "e" a continuación del último elemento de F.
- **close(F):** cierra el archivo vinculado al *manejador* F.
- **reset(F):** abre un archivo existente de nombre N para leer, y queda preparado para leer el primer elemento.
- **read(F,e):** lee un elemento del archivo F, copia el valor leído en "e" y queda preparado para leer el siguiente elemento (si existe) o queda en el fin del archivo.
- **eof(F)** (end of file – fin de archivo): retorna TRUE si se llegó al final de un archivo o FALSE en caso contrario.

5

Lectura de datos de un archivo

Considere las siguientes variables:
VAR F: FILE OF integer; e: integer;
reset(F): abre el archivo asociado al manejador F para leer, y queda preparado para leer el primer elemento.

F:

11	-2	5	-12	(EOF)
----	----	---	-----	-------

e: ?

↑

read(F,e): lee el elemento del archivo F que está listo para ser leído, copia el valor leído en la variable "e" y queda preparado para leer el siguiente elemento (si existe) o queda en el fin del archivo.

F:

11	-2	5	-12	(EOF)
----	----	---	-----	-------

e: 11

↑

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 6

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 "Resolución de Problemas y Algoritmos. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c) 2014

Lectura de datos de un archivo

Considere las siguientes variables:
VAR F: FILE OF integer; e: integer;

F:

11	-2	5	-12
----	----	---	-----

 (EOF) e:

5

Si ya fueron leídos los tres primeros elementos, como se muestra en la figura de arriba, al hacer `read(F,e)` se lee el último elemento y se llega al final del archivo (end of file) como muestra la figura siguiente:

F:

11	-2	5	-12
----	----	---	-----

 (EOF) e:

-12

¿Qué ocurre si ejecuto `read(F,e)` ahora?

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 7

Lectura de datos de un archivo

Considere las siguientes variables:
VAR F: FILE OF integer; e: integer;

F:

11	-2	5	-12
----	----	---	-----

 (EOF) e:

-12

MUY IMPORTANTE: si la primitiva "`read(F,E)`" es usada sobre el fin de un archivo (o un archivo vacío) es considerado un error de programación ya que dará un error y el programa dejará de ejecutarse.

Por lo tanto antes de usar "`read(F,e)`" debe asegurarse no estar al final del archivo con la función `eof(F)`.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 8

Lectura de datos de un archivo

Considere las siguientes variables:
VAR F: FILE OF integer; e: integer;

Al hacer "`reset(F)`" puede ocurrir que el archivo esté vacío y se tiene una situación como se muestra a continuación:

F: (EOF)

MUY IMPORTANTE: si la primitiva "`read(F,E)`" es usada sobre el fin de un archivo (o un archivo vacío) es considerado un error de programación ya que dará un error y el programa dejará de ejecutarse.

Por lo tanto antes de usar "`read(F,e)`" debe asegurarse no estar al final del archivo con la función `eof(F)`.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 9

Lectura de datos de un archivo

```

...
assign(F, 'enteros');
reset(F);
read(F,e);
writeln(' primero:', e);
...
    
```

MAL

Error de programación: no controla si el archivo está o no vacío

MUY IMPORTANTE: si la primitiva "`read(F,E)`" es usada sobre el fin de un archivo (o un archivo vacío) es considerado un error de programación ya que dará un error y el programa dejará de ejecutarse.

Por lo tanto antes de usar "`read(F,e)`" debe asegurarse no estar al final del archivo con la función `eof(F)`.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 10

Lectura de datos de un archivo

```

...
assign(F, 'enteros');
reset(F);
if not eof(F) then
begin
read(F,e);
writeln(' primero:', e);
end
...
    
```

```

...
assign(F, 'enteros');
reset(F);
while not eof(F) do
begin
read(F,e);
writeln(' encontré:', e);
end
...
    
```

IMPORTANTE: antes de usar "`read(F,e)`" debe asegurarse no estar al final del archivo con la función `eof(F)`.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 11

Lectura de datos de un archivo

```

...
assign(F, 'enteros');
reset(F);
repeat
read(F,e);
writeln(' encontré:', e);
until eof(F)
...
    
```

MAL

Error de programación: no controla si el archivo está o no vacío

IMPORTANTE: antes de usar "`read(F,e)`" debe asegurarse no estar al final del archivo con la función `eof(F)`.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 12

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 "Resolución de Problemas y Algoritmos. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c) 2014

Problema: mostrar todos los elementos de un archivo de números reales llamado "numeros.reales" y además indicar cuantos elementos fueron mostrados. Por ejemplo, si el archivo tiene los elementos: 11, -2 y 5, deberá mostrar e indicar que fueron mostrados 3 números. Si el archivo está vacío deberá indicarlo en pantalla.

Algoritmo

Abrir el archivo para leer.
 Leer uno a uno los elementos, mostrarlos en pantalla, y contar los que se van mostrando.
 Cerrar el archivo.
 Mostrar cuantos fueron leído (o que el archivo estaba vacío).
 fin.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 13

Realizar la traza con estos dos ejemplos

Program mostrar; {muestra y cuenta los elem. de un archivo}

VAR numeros: FILE OF real; cantidad: integer; e:real;

begin

assign(numeros, 'numeros.reales');

reset(numeros); Cantidad:=0;

while not eof(numeros) **do begin**

read(numeros,e); Cantidad:=cantidad+1;

writeln(' Elemento leído:', e);

end;

close(numeros);

if cantidad > 0 **then** writeln('mostrados=', cantidad)

else writeln('el archivo estaba vacío');

end.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 14

Problema : considere un archivo de números reales llamado "temperaturas.marzo" (ya creado). Escriba un programa que **calcule la temperatura promedio.**

Solución: (en el pizarrón)

Observe como la **experiencia** de la solución al problema anterior puede aprovecharse para resolver nuevos problemas donde, aún cuando su enunciado es diferente, presentan semejanzas (también llamada **analogía**) en su solución. Esto puede utilizarse como una estrategia en la resolución de problemas.

Algoritmo: (en el pizarrón)

Programa: (en el pizarrón)

Resolución de Problemas y Algoritmos Dr. Alejandro J. García

Problema : considere un archivo de números enteros llamado "accesos.web.marzo" (ya creado) que tiene la cantidad de accesos a una página web por día en el mes de marzo. Escriba un programa que **calcule la cantidad de accesos promedio en el mes.**

Solución: (en el pizarrón)

Observe como la **experiencia** de la solución al problema anterior puede aprovecharse para resolver nuevos problemas donde, aún cuando su enunciado es diferente, presentan semejanzas (también llamada **analogía**) en su solución.

Algoritmo: (en el pizarrón)

Programa: (en el pizarrón)

Resolución de Problemas y Algoritmos Dr. Alejandro J. García

Problema : considere un archivo de nros. reales llamado "temperaturas.marzo" (ya creado). Escriba un programa que **calcule la temperatura máxima.**

Solución: (en el pizarrón)

Observe nuevamente como la experiencia de la solución al problema anterior puede aprovecharse para resolver problemas donde, aún cuando su enunciado es diferente, presentan semejanzas (también llamada **analogía**) en su solución.

Algoritmo: (en el pizarrón)

Programa: (en el pizarrón)

Resolución de Problemas y Algoritmos Dr. Alejandro J. García

Problema: escriba un programa para copiar el contenido de "mis-numeros.datos", a otro archivo "mis-numeros.copia".

Program copiar; {VERSION INCORRECTA}

VAR F1,F2: FILE OF integer;

begin

assign(F1, 'mis-numeros.datos');

assign(F2, 'mis-numeros.copia');

F1:=F2;

close(F2); close(F1);

end.

MAL

Esto no copia los archivos

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 18

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 "Resolución de Problemas y Algoritmos. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c) 2014

Problema: escriba un programa para copiar el contenido de "mis-numeros.datos", a otro archivo "mis-numeros.copia".

```

Program copiar; {version correcta}
VAR F1,F2:FILE OF integer;
    elemento: integer;
begin
    assign(F1, 'mis-numeros.datos'); reset(F1);
    assign(F2, 'mis-numeros.copia'); rewrite(F2);
    while not eof(F1) do begin
        read(F1,elemento);
        write(F2, elemento);
    end;
    close(F1); close(F2);
end.

```

Lee un elemento del archivo F1
y luego escribe elemento en F2

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

19

Problema: escriba un programa que agregue un elemento a un archivo "mis-numeros.dat" (ya creado y con números en él).

Algoritmo: (primer refinamiento)

- Abre "mis-numeros" para lectura
- Abre un archivo llamado "copia" para escritura
- Pasa todos los elementos de "mis-numeros" a "copia"
- Agrega en "copia" el elemento nuevo
- Cierra los archivos
- Abre "copia" para lectura
- Abre "mis-numeros" para escritura
- Pasa todos los elementos de "copia" a "mis-numeros"

Programa: (tarea para el hogar)

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

Problema: escriba un programa que agregue tantos elementos como el usuario quiera a un archivo "mis-numeros.dat" (ya creado y con números en él).

Algoritmo: (primer refinamiento)

- Abre "mis-numeros" para lectura
- Abre un archivo llamado "copia" para escritura
- Pasa todos los elementos de "mis-numeros" a "copia"
- Mientras el usuario lo indique:
agrega en "copia" todos los elementos ingresados
- Cierra los archivos
- Abre "copia" para lectura
- Abre "mis-numeros" para escritura
- Pasa todos los elementos de "copia" a "mis-numeros"

Programa: (tarea para el hogar)

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

Problemas propuestos

- **Problema:** escriba un programa que vea si **primer elemento es igual al último**.
- **Problema:** escriba un programa que vea si los **elementos del archivo** (ya creado y con números en él) **están ordenados de menor a mayor**. Si no lo está deberá mostrar el primer par de elementos que no lo cumple. Por ejemplo, la secuencia [2,4,5,7,9] está ordenada, pero la secuencia [2, 4, 8, 5, 1] no lo está y el primero par que no lo cumple es (8,5).

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

22

Referencias para los curiosos

http://es.wikipedia.org/wiki/Memoria_de_acceso_aleatorio

http://es.wikipedia.org/wiki/Memoria_secundaria

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

23

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:

"Resolución de Problemas y Algoritmos. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c) 2014