

Resolución de Problemas y Algoritmos

Clase 11: Archivos de texto para entrada y salida.

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

Archivos de texto en Pascal (TEXT)

En Pascal, existe un tipo predefinido "TEXT" que permite trabajar con **archivos de texto**.

Program ejemplo;
VAR documento: TEXT;

- A primera vista parece que fuera lo mismo que **FILE OF char**. Pero en esta clase veremos que no es así.
- El tipo **TEXT** tiene características propias que pueden ser vistas como facilidades para determinados problemas.
- Recordemos que un **tipo de dato** define los valores y las operaciones que pueden usarse sobre ellos.
 - Valores:** **TEXT** es un **tipo estructurado**, que permite almacenar una secuencia de caracteres ASCII.
 - Operaciones:** todas las operaciones de **FILE** y además agrega: **readln**, **writeln**, y **EOLN** (end of line).

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

Características del tipo TEXT

- Una característica del tipo **TEXT** es que es permite el manejo de archivos creados con otros editores de texto (como el block de notas o notepad, el editor de lázarus, etc.). Observe que tiene que ser cualquier editor que genere un *texto plano (plain text)*.
- Usando un archivo **TEXT** podemos *procesar* desde nuestro programa en Pascal cualquier archivo de texto en memoria secundaria (creado con otro programa en Pascal o no).
- También, desde un programa en Pascal podemos crear y escribir en un archivo **TEXT**, y luego ver el contenido del archivo con cualquier editor de texto que tengamos en nuestro dispositivo.
- Observación interesante: el "buffer" es un archivo **TEXT**. Todo lo que puede hacer con el buffer lo puede hacer en un **TEXT**.

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

Problema propuesto

- Considere un archivo llamado "texto.txt" creado con un editor como este:

Problema: escriba un programa para abrir un archivo de texto ya existente llamado "texto.txt", y mostrar por consola su contenido.

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

Mostrar el archivo "texto.txt"

Program leer; {Este programa permite leer todo el contenido de un archivo de texto, (carácter por carácter) y mostrarlo en pantalla }

VAR T: TEXT; elemento: char;

begin

assign(T, 'texto.txt'); ← Aquí indico el nombre del archivo a mostrar por consola

reset(T); //abre archivo para leer

while not eof(T) do

begin

read(T,elemento);

write(elemento);

end;

close(T);

end.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 5

¿Qué es ENTER?

En las máquinas de escribir mecánicas al finalizar un renglón había que hacer dos movimientos: (1) retorno de carro (2) nueva línea

En algunos sistemas operativos
ENTER tiene asociados 2 caracteres:

(1) **ASCII 13:** retorno de carro (CR: carriage return)

(2) **ASCII 10:** nueva línea (LF: line feed)

Los símbolos ASCII 13 y 10 son **caracteres de control** y al imprimirlos en pantalla producen un efecto en lugar de mostrar algo visible. Vea por ejemplo:

Program uno; Begin WRITE(CHR(65)); WRITE(CHR(66)); End .	Program dos; Begin WRITE(CHR(65)); WRITE(CHR(10)); WRITE(CHR(66)); End .	Program tres; Begin WRITE(CHR(65)); WRITE(CHR(13)); WRITE(CHR(66)); End .	Program cuatro; Begin WRITE(CHR(65)); WRITE(CHR(13)); WRITE(CHR(10)); WRITE(CHR(66)); End .
---	---	--	--

¿Cómo estará implementado writeln?

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) 22/04/2016

Otro archivo de texto

Observe que el código fuente de un programa en Pascal es un archivo de texto. Por lo tanto, al igual que "texto.txt" es posible mostrarlo por consola, carácter a carácter.

Esta es la salida del programa leer asignando el manejador T a el archivo potencia.pas. Es el primer paso para hacer un compilador.

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

Mostrar el archivo de texto "potencia.pas"

Program leer; {Este programa permite leer todo el contenido de un archivo de texto, (carácter por carácter) y mostrarlo en pantalla }

```

VAR T: TEXT; elemento: char;
begin
assign(T, 'potencia.pas');
reset(T); {abre archivo para leer}
while not eof(T) do
begin
read(T,elemento);
write(elemento);
end;
close(T);
end.
    
```

Ahora el manejador T está asociado al código fuente de potencia.pas

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

Operaciones sobre archivos de texto en Pascal

Además de todas las operaciones vistas sobre archivos secuenciales (FILE) se agregan:

Función predefinida:

- eoln(F)** (end of line): retorna **TRUE** si se llegó al final de una línea y **FALSE** en caso contrario.

Procedimientos predefinidos:

- writeln(T)**: escribe un fin de línea (enter) en T.
- readln(T)**: avanza en el texto hasta llegar a un fin de línea (enter) y se prepara para leer el carácter siguiente.

Observaciones:

- readln(T,e)** es equivalente a **read(T,e); readln(T)**
- writeln(T,e)** es equivalente a **write(T,e); writeln(T)**
- El "buffer" es un archivo TEXT. Todo lo que puede hacer con read y write en el buffer lo puede hacer en un TEXT.

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

End Of Line (fin de línea) EOL

- End-of-line (EOL)** fin de línea es un carácter especial o secuencia de caracteres, que indica el final de una línea de texto y el paso a la siguiente. Se le llama así porque el carácter a la derecha de EOL aparecerá en la línea de abajo.
- Los sistemas operativos representan EOL con los caracteres ASCII: LF (Salto de línea) o/y CR (Retorno de carro) pero puede ser de diferente manera. Por ejemplo:
 - LF**: Multics, Unix, GNU/Linux, AIX, Xenix, MacOSX, FreeBSD, BeOS, Amiga, RISC OS,
 - CR+LF**: MS-DOS, OS/2, Microsoft Windows, Symbian
 - CR**: Apple II family, Mac OS
- En Pascal, las primitivas predefinidas **eoln**, **readln**, **writeln**, nos permiten "abstraernos" de esto, ya que conocen como está implementado EOL en cada sistema operativo para el cual se compilará nuestro programa.

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

Características de archivos de texto (TEXT)

- Una característica del tipo **TEXT** es que utilizando los procedimientos predefinidos **write** o **writeln** puedo escribir elementos de cualquier tipo simple y son transformados a texto automáticamente.
- La transformación a texto es automática y funciona igual que la transformación a texto al usar **write** o **writeln** para mostrar en la consola.
- Incluyendo al formato de números reales o enteros utilizando el símbolo "...".
- Tenga en cuenta que el "buffer" es un archivo **TEXT**. Todo lo que puede hacer con el buffer lo puede hacer en un **TEXT**.
- Esto los diferencia a **TEXT** de FILE OF **char**;

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

Program crearTXT; {Ejemplo: diferencias entre FILE OF char y TEXT}

```

VAR T: TEXT; // archivo de texto
A: FILE OF char; // archivo de datos
begin
assign(A, 'texto.txt'); //cuidado: la extensión txt no lo hace archivo TEXT
rewrite(A); // crea un archivo de caracteres (no de texto)
write(A, 'H'); // escribe correctamente la letra H en el archivo
write(A, 126); // ERROR de compilación (tipos incompatibles, esperaba char)
assign(T, 'otro-texto.txt');
rewrite(T); // crea un archivo de texto
write(T, 'H'); // escribe correctamente la letra H en el archivo
write(T, 126); // transforma el número 126 al texto "126" y lo escribe en T
write(T, 3.518); // transforma el real a texto y lo escribe en T
write(T, 3.518:0:2); // transforma el real al texto "3.52" y lo escribe en T
writeln(T); // deja renglón en blanco
writeln(T, 'Hasta luego amigos'); // Escribe en T todo el texto entre comillas
close(T); close(A);
end.
    
```

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) 22/04/2016

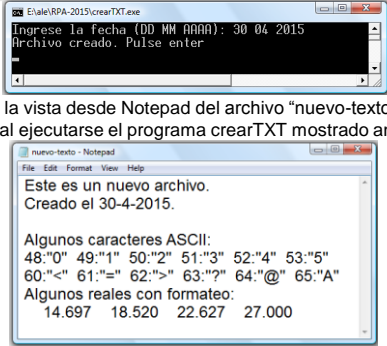
```

Program crearTXT; {Ejemplo: crear y escribir en archivo de texto}
VAR T: TEXT; dia, mes, anio, i: integer;
begin
write('Ingrese la fecha (DD MM AAAA): '); readln(dia, mes, anio);
assign(T, 'nuevo-texto.txt');
rewrite(T); // crea el archivo para escribir en él
writeln(T, 'Este es un nuevo archivo.');// escribe texto
writeln(T, 'Creado el ', dia, '-', mes, '-', anio, '.');// escribe texto y valores
writeln(T); // deja renglón en blanco
writeln(T, 'Algunos caractes ASCII');
FOR i:=48 to 53 do write(T, i, ' ', CHR(i), ' ');
writeln(T); // baja de línea
FOR i:=60 to 65 do write(T, i, ' ', CHR(i), ' ');
writeln(T); writeln(T, 'Algunos reales con formato: ');
FOR I:= 6 TO 9 DO write(T, SQRT(I*I):10:3);
close(T); writeln('Archivo creado. Pulse enter'); readln;
end.
    
```

VER EFECTO

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

Archivo generado con crearTXT



Esta es la vista desde Notepad del archivo "nuevo-texto.txt" creado al ejecutarse el programa crearTXT mostrado antes.

VOLVER

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

Problema: escriba un programa para abrir un archivo de texto ya existente llamado "texto.txt", y contar cuantas líneas tiene.

Program líneas; {Este programa cuenta las líneas de un archivo de texto aprovechando el prodemiento predefinido readln que avanza hasta el final de una línea del archivo (ie, un enter) }

```

VAR T: TEXT; cant:integer;
begin
assign(T, 'texto.txt'); reset(T); cant:=0;
while not eof(T) do
begin readln(T); cant:=cant+1; end;
writeln('Cantidad de líneas: ', cant);
close(T);
end.
    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 15

Problema: escriba un programa para abrir un archivo de texto ya existente llamado "texto.txt", y generar otro que cuando encuentre un punto baje de línea.

Program líneas;

```

VAR T1,T2: TEXT; cant:integer; ch: char;
begin
assign(T1, 'texto.txt'); reset(T1);
assign(T2, 'otro.txt'); rewrite(T2);
while not eof(T1) do
begin
read(T1,ch);
write(T2,ch);
if ch='.' then writeln(T2);
end;
close(T1); close(T2);
end.
    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 16

Características de archivos de texto (TEXT)

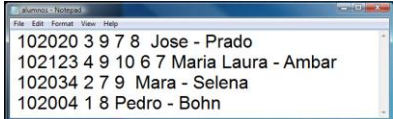
- Otra característica de **TEXT**, es que si el archivo a procesar tiene un formato uniforme y este formato se conoce de antemano, entonces utilizando los procedimientos **read** y **readln** puedo leer con variables de diferentes tipos simples y se realizará automáticamente la conversión adecuada.
- La conversión de texto a otro tipo de dato simple es la misma que ocurre al usar **read** o **readln** para lectura de lo ingresado por teclado (y que queda en el buffer).
- Recuerde:** el "buffer" es un archivo de tipo **text**. Todo lo que puede hacer con el buffer lo puede hacer con un **text**.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 17

Aplicación de procesamiento de texto

Considere un archivo de tipo TEXT donde cada línea tiene un formato fijo con los siguientes elementos:
 un número de LU, la cantidad (N) de notas de un alumno, seguido de una secuencia de N enteros que representan sus notas, y finalmente un texto con nombre y apellido (separados por un guión).

• Ejemplo:

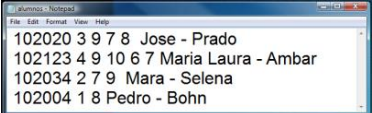


Se asume que el archivo no tiene errores de carga de datos.

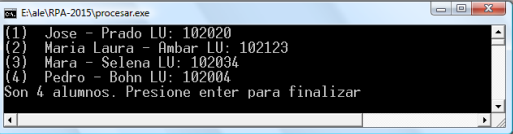
Resolución de Problemas y Algoritmos Dr. Alejandro J. García 18

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) 22/04/2016

Problema propuesto




Problema propuesto: escriba un programa que a partir de los datos en alumnos.txt, muestre un listado numerado como se muestra en el ejemplo a continuación. Para cada alumno se debe mostrar: nombre, apellido, "LU:" y número de libreta. Indicar al final la cantidad total.



Resolución de Problemas y Algoritmos Dr. Alejandro J. García 19

Problema propuesto



Algoritmo procesar alumnos.txt:
Inicializar contador en 0
Mientras no llegue al final del archivo:
 leer un número de libreta
 leer cantidad de notas y saltar todas las notas siguientes
 incrementar contador en 1 y mostrar contador en pantalla
 mostrar todo el texto hasta el final de la línea (nom. y apellido)
 mostrar "LU:" y el número de libreta leído

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 20

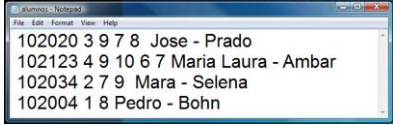
```

Program procesar; {procesa el texto de alumnos.txt}
VAR T: TEXT; lu, cant, nota,i,cont: integer; elemento: char;
begin
assign(T,'alumnos.txt'); reset(T); cont:=0;
while not eof(T) do begin
    cont:=cont+1;
    read(T, lu); // guardo lu para mostrar luego
    read(T, cant); // leo cantidad de notas a saltar
    for i:=1 to cant do read(T,nota); // salteo todas las notas
    write('(',cont,') ');
    while not eoln(T) do // muestro nombre y apellido
        begin read(T,elemento); write(elemento); end;
    writeln(' LU: ', lu); // muestro lu leida aantes
end; // fin del while not eof(T)
write('Son ',cont,' alumnos. Presione enter para finalizar');
close(T); readln;
end.
    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 21

Aplicación propuesta

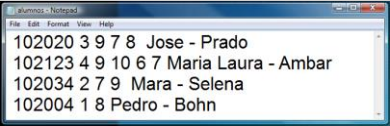
Considere un archivo de tipo TEXT donde cada línea tiene un formato fijo con los siguientes elementos: un número de LU, la cantidad (N) de notas de un alumno, seguido de una secuencia de N enteros que representan sus notas, y finalmente un texto con nombre y apellido (separados por un guión). Por ejemplo:



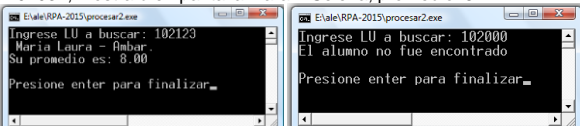
Problema propuesto: Escriba un programa que solicite un número de libreta (LU), y si lo encuentra en "alumnos.txt" muestre nombre, apellido y el promedio de las notas del alumno. Por ejemplo, si LU es 102034, mostrará en pantalla Mara – Selena, promedio: 8.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 22

Aplicación propuesta



Problema propuesto: Escriba un programa que solicite un número de libreta (LU), y si lo encuentra en "alumnos.txt" muestre nombre, apellido y el promedio de las notas del alumno. Por ejemplo, si LU es 102034, mostrará en pantalla Mara – Selena, promedio: 8.



Resolución de Problemas y Algoritmos Dr. Alejandro J. García 23

Algoritmo

Problema propuesto: Escriba un programa que solicite un número de libreta (LU), y si lo encuentra en "alumnos.txt" muestre nombre, apellido y el promedio de las notas del alumno. Por ejemplo, si LU es 102034, mostrará en pantalla Mara – Selena, promedio: 8.

Algoritmo promedio alumno:
solicitar LU
Recorre el archivo de texto desde el comienzo.
Mientras no llegue al final del archivo y no encontró la LU
hacer: leer un número de libreta
Si el número leído es la LU buscada
entonces: calcular suma de notas
 mostrar nombre y apellido
Mostrar promedio
de lo contrario: saltar al final de la línea del archivo

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 24

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) 22/04/2016

```
Program procesar2; //busca y muestra promedio alumno
VAR T: text; buscado,lu,cant,nota,i,suma:integer; c:char; encuentre:boolean;
begin
assign(T, 'alumnos.txt'); reset(T);
write('Ingrese LU a buscar: '); readln(buscado); encuentre:=false;
while not eof(T) and not encuentre do begin
read(T, lu);
IF lu = buscado then
begin // si no es el alumno buscado, calculo y muestro los datos en pantalla
read(T, cant); suma:=0; encuentre:=true;
for i:=1 to cant do begin read(T,nota); suma:=suma+nota; end; // calcula suma
while not eoln(T) do begin read(T,c); write(c); end; // muestra nombre-apellido
writeln('.'); writeln('Su promedio es:', suma/cant:5:2);
end // del if
else readln(T); // si no es el alumno buscado saltea la línea completa
end; \\ while not eof
close(T); if not encuentre then writeln('El alumno no fue encontrado');
writeln; write('Presione enter para finalizar'); readln;
end.
```



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) 22/04/2016