



Resolución de Problemas y Algoritmos

Clase 8: Almacenamiento en Memoria. Archivos secuenciales en Pascal.



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



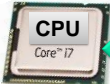
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Bahía Blanca - Argentina

Computadora con arquitectura von Neumann

Una computadora es un sistema digital con tecnología microelectrónica compuesta por:

- 1- CPU (Unidad Central de Proceso)
- 2- Memoria
- 3- Dispositivos de Entrada y Salida

Todo interconectado (por "buses")



← bus

Memoria principal (RAM):
programas en ejecución y datos

← bus

Memoria secundaria
(ej: disco rígido, pen-drive, dvd, unidad de estado sólido)

Archivos de programas, textos, datos, fotos, música, etc.

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

Memoria RAM

RAM concreta y real: módulo 4Gb DDR3

abstracción de bajo nivel (secuencia de bytes)

0	0	1	0	1	0	1	1
1	1	1	0	1	0	0	1
0	1	1	1	0	1	0	1
1	1	1	0	1	0	0	1
1	1	0	1	1	1	1	0
1	0	1	1	1	1	1	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	0	0

abstracción de alto nivel (una traza de RPA)

NUM:	123
ES_PAR:	FALSE
LETRA:	"A"
PRECIO:	12.02

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

Memoria secundaria (disco duro)



Sistema de grabación magnética con uno o más platos o discos rígidos, unidos por un mismo eje que gira a gran velocidad

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

Memoria secundaria (solid state drive)



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

Memoria secundaria (SSD - solid state drive)

Una unidad de estado sólido o SSD (acrónimo de solid-state drive) es un dispositivo de almacenamiento de datos que usa una memoria no volátil, como la memoria flash, o una memoria volátil como SDRAM, para almacenar datos, en lugar de los platos giratorios magnéticos encontrados en los discos duros convencionales. En comparación con los discos duros tradicionales, las unidades de estado sólido son menos sensibles a los golpes, son prácticamente inaudibles y tienen un menor tiempo de acceso y de latencia.

A veces se traduce erróneamente en español la "D" de SSD como "disk" cuando, en realidad, representa la palabra "drive", que podría traducirse como unidad o dispositivo.

A partir de 2010, la mayoría de los SSDs utilizan memoria flash basada en compuertas NAND, que retiene los datos sin alimentación.

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)1998-2013.

CONCEPTOS: Valores de variables en Pascal

- Las variables de los tipos de datos **SIMPLES** vistos hasta el momento en esta materia tiene las siguientes características:
 - (1) Residen en **memoria principal** (RAM, *random-access memory* o *memoria de acceso aleatorio*).
 - (2) Los **valores** que contienen **no perduran** cuando termina la ejecución del programa. (El espacio de memoria utilizado por esas variables es liberado y usado por otros programas)
 - (3) La **cantidad** de **memoria** que usan es **fija** (no cambia en ejecución) y el compilador usa este dato para reservar lugar en memoria.

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

Almacenamiento en memoria

Las variables de los siguientes tipos usan un espacio en memoria que es fijo que no cambia en ejecución (pero puede variar de un compilador a otro).
En Free Pascal (Lazarus) generalmente ocupan:

Tipo:	Tamaño:
Char	1 byte
Boolean	1 byte
Integer	2 o 4 bytes
Real	6 bytes

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

CONCEPTOS: Valores de variables en Pascal

- Sin embargo, muchas veces es útil que determinados **valores puedan perdurar** aunque el programa termine, y que estos valores **puedan ser utilizados** en el futuro por **otro programa**.
- En Pascal existe un **tipo de dato estructurado FILE** (archivo/fichero) que permite almacenar valores que pueden perdurar aún cuando la ejecución del programa termine.
- De esta manera, un programa podrá “leer” elementos generados por otro programa; y además “escribir” datos que podrá leer otro programa (o él mismo pero en otra ejecución)

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

CONCEPTOS: Archivos (Files)

- Para que el contenido de un archivo (file) perdure más allá de la ejecución de un programa y aún cuando la computadora estuviera apagada por un tiempo, **los archivos (files) residen en memoria secundaria** (como un disco rígido por ejemplo).
- Es importante notar que el acceso a memoria secundaria **depende del Sistema Operativo** usado.
- El manejo archivos en Pascal también puede tener **diferencias de un compilador a otro**.
- En esta materia se verán algunos conceptos de archivos secuenciales** y algunos detalles estarán ligados al sistema operativo o al compilador.

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

CONCEPTOS: Archivos secuenciales

- Un Sistema Operativo (SO) puede manejar distintas clases de archivos (de texto, fotos, películas, música, ejecutables).
- Cada Lenguaje de Programación (LP) pueden manejar algunas de estas clases de archivos.
- En Pascal hay una clase de archivo que se denomina archivo secuencial**.
- Un archivo secuencial es una **estructura compuesta por una secuencia de elementos** (componentes) donde hay un orden lineal entre ellos.

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

Conceptos:

Tipo de Dato: define el **conjunto de valores posibles que puede tomar una variable, y también define las operaciones que pueden usarse.**

Los tipos de datos en Pascal pueden ser:

- Simples:** INTEGER, REAL, CHAR, BOOLEAN, y otros que no veremos en RPA
- Estructurados:** FILE (archivo), TEXT (archivo de texto) y otros que no veremos en RPA

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)1998-2013.

Archivos secuenciales en Pascal (FILE OF ...)

- En Pascal, un **archivo secuencial** (FILE) es una **sucesión finita** de **componentes** que pueden accederse una a una, comenzando de la primera.
- Todos los **componentes** deben ser del **mismo tipo** de datos (ej. todos integer, o todos char).
- Por ser un **archivo secuencial**, una vez accedida la primera componente, el acceso a la componente de la posición P se logra luego de haber accedido previamente a la posición P-1.
- La **cantidad** de componentes es **potencialmente infinita** (su límite estará dado por la cantidad de espacio en la computadora donde está el archivo).
- Son almacenados en **Memoria Secundaria**.

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

Declaración de archivos secuenciales en Pascal

- La palabra reservada FILE permite declarar un tipo o una variable de tipo archivo.

Program ejemplo;

```
VAR F1,F2: FILE OF integer;
    apunte: FILE OF char;
    temperatura FILE OF real;
```

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


Representación gráfica

Program ejemplo;

```
VAR F1,F2: FILE OF integer;
    carta, apunte: FILE OF char;
```

- Dado que una variable de tipo FILE es una sucesión de componentes del mismo tipo, se usará la siguiente representación gráfica:

Cada componente es de tipo integer.

F1: 

- 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 15

Nombres de archivos secuenciales en Pascal

- El **identificador** de una variable de **tipo archivo** es un **nombre "lógico"** interno usado por Pascal para referirse a un archivo secuencial.
- Como los valores almacenados en variables de tipo FILE van a estar en memoria secundaria (por ejemplo: disco rígido), el **sistema operativo necesita asignarle un nombre**.
- Este nombre puede ser usado en el futuro por otro programa para usar el mismo archivo.
- La **primitiva ASSIGN** permite vincular el nombre lógico (identificador de variable) y el físico (archivo en el disco rígido).

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

Primitiva ASSIGN

Program ejemplo;

```
VAR numeros: FILE OF integer;
Begin
ASSIGN(numeros, 'mis-numeros.arch');
REWRITE(numeros); {crea el archivo}
...
```

- ASSIGN(F, N)** recibe: **F** que es un identificador de variable de tipo FILE, y **N** que es una secuencia de caracteres que representa un nombre válido de archivo en el sistema operativo usado.
- Una vez ejecutado assign vincula **N a F**.
- F** es llamado manejador (file handler) de **N**, y dentro del código programa toda referencia al archivo se hace usando el identificador **F**.

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

Primitivas de Pascal para archivos secuenciales

- assign(F,N):** vincula F (manejador interno) con N (nombre real en el disco).
- rewrite(F):** crea un archivo nuevo (sobre-escribe si ya existe otro con ese nombre, el cual se pierde).
- write(F,e):** en un archivo creado con rewrite, escribe el valor de "e" a continuación del último elem. de F.
- close(F):** cierra el archivo vinculado al handler F.
- reset(F):** abre un archivo existente 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).
- eof(F)** (end of file): retorna TRUE si se llegó al final de un archivo y FALSE en caso contrario.

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)1998-2013.

Problema: escriba un programa para crear un archivo llamado "mis-numeros.dat" y escribir en él enteros de 1 a un tope ingresado por el usuario.

Algoritmo
 Crear el archivo "mis-numeros.dat" para escribir en él.
 Solicitar un entero tope
 Escribir en el archivo "tope" enteros desde el 1 hasta tope
 Cerrar el archivo
fin.

Observación: En este algoritmo conozco de antemano cuanto elementos quiero escribir en el archivo, entonces en Pascal puedo usar FOR.

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

Problema: escriba un programa para crear un archivo llamado "mis-numeros.dat" y escribir en él enteros de 1 a un tope ingresado por el usuario.

Program crear;
VAR nuevo: FILE OF integer;
 valor,tope: integer;
begin
 assign(nuevo, 'mis-numeros.dat');
 rewrite(nuevo); *{crea archivo vacío y permite escribir en él}*
 writeln('Cantidad de enteros a escribir en el archivo ');
 readln(tope);
for valor:= 1 to tope **do** write(nuevo,valor);
 close(nuevo);
end.

Escribe un entero al final del archivo

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

Problema: escriba un programa para abrir un archivo ya existente llamado "mis-numeros.dat", leer todos sus componentes, y mostrarlos por pantalla.

Algoritmo
 Abrir el archivo "mis-numeros.dat" para leer sus elementos
 Leer uno a uno e ir mostrando en pantalla
 Cerrar el archivo.
fin.

Observación: En este algoritmo NO conozco de antemano cuanto elementos hay en el archivo.
 En Pascal ¿puedo usar FOR? ¿Qué uso? ¿Cuántas veces se repetirá?

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

Problema: escriba un programa para abrir un archivo ya existente llamado "mis-numeros.dat", leer todos sus componentes, y mostrarlos por pantalla.

Program leer;
VAR arch_num: FILE OF integer; elemento: integer;
begin
 assign(arch_num, 'mis-numeros.dat');
 reset(arch_num); *{abre el archivo para leer de él}*
 while not eof(arch_num) **do** *{mientras no llegue al fin}*
begin
 read(arch_num,elemento); writeln(' fue leído:', elemento);
end;
 close(arch_num);
end.

Lee un elemento del archivo y pasa al siguiente

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

Problemas propuestos

- **Problema:** escriba un programa que cuente **cuantos elementos** tiene el archivo "mis-numeros.dat" (ya creado y con números en él).
- **Problema:** escriba un programa que busque **cuantas veces está** el elemento E (ingresado por el usuario) en el archivo "mis-numeros.dat" (ya creado y con números en él).
- **Problema:** escriba un programa que vea si **primer elemento es igual** al **último** en el archivo "mis-numeros.dat" (ya creado y con números en él).
- **Problema:** escriba un programa que vea si los elementos del archivo "mis-numeros.dat" (ya creado y con números en él) **están ordenados de menor a mayor.**

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)1998-2013.