

**UNIVERSIDAD INTERAMERICANA DE PANAMÁ**

**FACULTAD DE INGENIERIA, ARQUITECTURA Y DISEÑO**

**CARRERA:** Licenciatura en Ingeniería en Sistemas Computacionales

MATERIA: **Estructura de Datos**

Estudiante - Cédula

Hernán Domínguez- 8-970-599

Profesor Facilitador:

**María De Los A. Morales De Dimares**

**Investigación #:1**

**Tema: Registros, Conjuntos y Archivos Secuenciales.**

**Fecha de Entrega:**

**FECHA: 25 DE NOVIEMBRE DEL 2020**

Índice

*Introducción* ………………………………………………………………3

*Registros* ……………………………………………………………………4

*Conjuntos …………………………………………………………………………… 5*

*Archivos Secuenciales…………………………………………………………………………5*

*Conclusión …………………………………………………………………………… 6*

*Bibliografía ……………………………………………………………………………7*

*Anexos …………………………………………………………………………… 8*

Introducción

Una estructura de datos es una forma particular de organizar datos en una computadora para que puedan ser utilizados de manera eficiente. Diferentes tipos de estructuras de datos son adecuados para diferentes tipos de aplicaciones, y algunos son altamente especializados para tareas específicas.

Las estructuras de datos son un medio para manejar grandes cantidades de datos de manera eficiente para usos tales como grandes bases de datos y servicios de indización de Internet. Por lo general, las estructuras de datos eficientes son clave para diseñar algoritmos eficientes. Algunos métodos formales de diseño y lenguajes de programación destacan las estructuras de datos, en lugar de los algoritmos, como el factor clave de organización en el diseño de software.

Registros

Un registro es una estructura de datos que permite almacenar un conjunto de elementos no necesariamente del mismo tipo.

A diferencia de los vectores y matrices, que permiten almacenar un conjunto de datos del mismo tipo, un registro normalmente almacena un conjunto de datos que están relacionados entre sí.

Ejemplos de registros podrían ser los datos de un alumno (nro de teléfono, apellido y nombre, carrera que cursa), una historia clínica de un paciente (nro de documento, obra social que tiene, enfermedades), etc.

Ejemplos

Aquí vemos un ejemplo de registros en el lenguaje C:

**struct** usuario {

**char** nombre [50];

**char** apellido [60];

**int** teléfono**;**

**int** edad;

};

Como se puede observar, la estructura, en este caso llamada “usuario” puede almacenar tanto cadenas como números enteros.

Conjuntos

Un conjunto es una colección (contenedor) de ciertos valores, sin ningún orden concreto ni valores repetidos. Su correspondencia en las matemáticas sería el conjunto finito. Sin tener en cuenta la secuencia, ni el hecho de que no haya valores repetidos, se asemeja a una lista. Un conjunto puede verse como una cadena asociativa (array) (mapeado parcial) donde no se atiende al valor de cada par clave-valor.

Los conjuntos pueden implementarse usando diversas estructuras de datos. Con una estructura de datos ideal se comprueba si un objeto se encuentra en el conjunto, además de activarse otras operaciones útiles tales como la iteración sobre todos los objetos del conjunto, la realización de uniones o intersecciones entre dos conjuntos, o la toma del complemento de un conjunto en algún dominio limitado. Cualquier estructura de datos en cadena asociativa puede usarse para implementar un conjunto, dejando que los juegos de claves sean los elementos del conjunto, e ignorando los valores.

Ejemplo de Conjuntos en Python

Para crear un conjunto por asignación debemos indicar sus elementos encerrados entre llaves y separados por coma.

conjunto1= {1, 5, 10, 20}

print(conjunto1)

Los elementos de un conjunto pueden ser de distinto tipo:

conjunto2= {"juan", 20, 6.4, True}

print(conjunto2)

Podemos definir elementos de un conjunto de tipo tupla, pero no de tipo lista, diccionario o conjunto:

conjunto3= {("juan", 33), ("ana", 44)}

print(conjunto3)

Archivos Secuenciales

Es la forma básica de organizar un conjunto de registros, que forman un archivo, utilizando una organización secuencial. En un archivo organizado secuencialmente, lo registros quedan grabados consecutivamente cuando el archivo se utiliza como entrada. En la mayoría de los casos, los registros de un archivo secuencial quedan ordenados de acuerdo con el valor de algún campo de cada registro. Semejante archivo se dice que es un archivo ordenado; el campo, o los campos, cuyo valor se utiliza para determinar el ordenamiento es conocido como la llave del ordenamiento. Un archivo puede ordenarse ascendente o descendentemente con base en su llave de ordenamiento.

Los archivos pueden ser abiertos en uno de varios modos. Para crear ón archivo o para descartar el contenido de un archivo antes de escribir datos, abra el archivo para escritura ("w").

Para leer un archivo existente, ábralo para lectura ("r"). Para añadir registros al final de un archivo existente, abra el archivo para agregar ("a"). Para abrir un archivo de tal forma que pueda ser escrito y leído, abra el archivo para actualizar en uno de los tres modos de actualización, "r+", "w+" o "a+". El modo "r+" abre un archivo para lectura y escritura.

El modo "w+" genera un archivo para lectura y escritura. Si el archivo ya existe, el archivo es abierto y el contenido actual de dicho archivo se descarta.

El modo "a+" abre un archivo para lectura y escritura, toda escritura se efectuará al final del archivo. Si el archivo no existe será creado. Si al abrir un archivo, ocurre un error, fopen regresará NULL.

Ejemplo en C

fscanf (apuntador, "%d%s%f", &campo1, campo2, &campo3)

Cada vez que ejecute el enunciado fscanf, se leerá otro registro del archivo, y los campos de registro, campo1, campo2 y campo3 tomarán nuevos valores.

Para recuperar secuencialmente datos de un archivo, el programa normalmente empieza a leer a partir del principio del archivo, y lee todos los datos de forma consecutiva, hasta que encuentra los datos deseados

Conclusión

En este proyecto se ha logrado aprender a cabalidad las estructuras de datos mencionadas: Registros, Conjuntos y Archivos Secuenciales. Además, se pudieron ver ejemplos en los lenguajes de programación C (para los registros y archivos secuenciales) y Python (Para los Conjuntos). Aunque vale la pena decir que, ambos lenguajes poseen la capacidad de manejar estas estructuras, (excepto los registros, donde no encontré alguna forma de implementación en Python); por lo que el hecho de mostrar más de un lenguaje en los ejemplos fue decisión totalmente mía, para comprender como funcionaban en ambos lenguajes.

Bibliografía

Tutoriales Programación Ya. “Estructura de datos tipo conjunto (set)” Tutoriales Programación Ya. <https://www.tutorialesprogramacionya.com/pythonya/detalleconcepto.php?punto=95&codigo=96&inicio=90#:~:text=La%20cuarta%20estructura%20de%20datos,y%20no%20se%20encuentran%20ordenados>

Wikipedia. “Estructura de Datos” Fundación Wikimedia, Inc., última vez el 16 nov 2020, 13:42. <https://es.wikipedia.org/wiki/Estructura_de_datos>

Wikipedia. “Registro (Estructura de Datos)” Fundación Wikimedia, Inc., última vez el 19 nov 2020, 23:25. <https://es.wikipedia.org/wiki/Registro_(base_de_datos)>

Tutor de C++ “Tutor de C++” Tutor de C++. <http://decsai.ugr.es/~jfv/ed1/c++/cdrom3/TIC-CD/web/tema25/teoria_4.htm>

Tutoriales Programación Ya. “Estructura de datos tipo registro (struct)” Tutoriales Programación Ya. <https://www.tutorialesprogramacionya.com/cya/detalleconcepto.php?punto=27&codigo=27&inicio=15>

Anexos

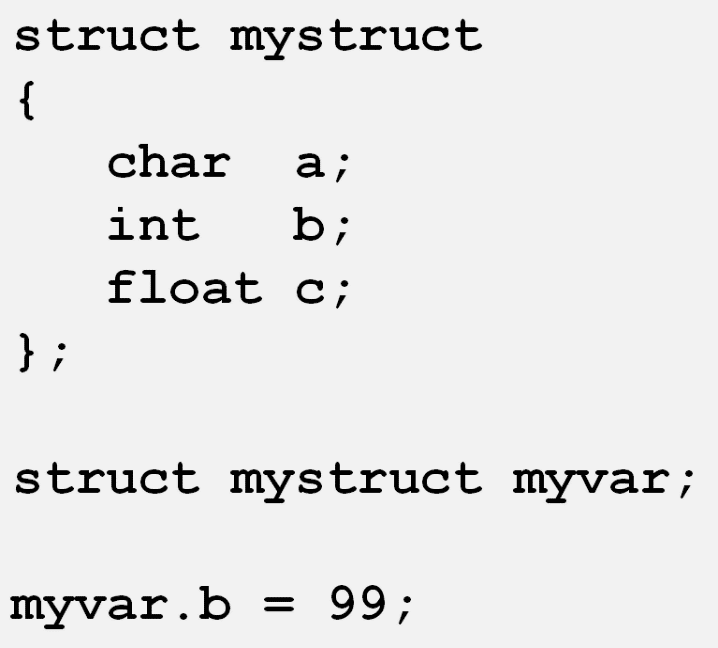


Ilustración 1. Ejemplo de Registro(struct) en C

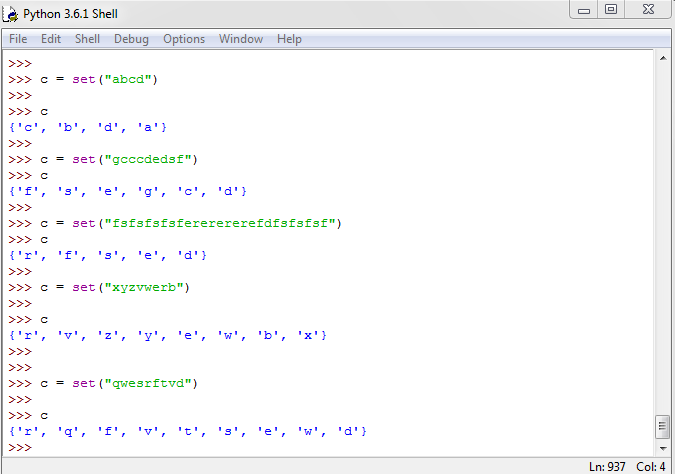


Ilustración 2. Ejemplo de Estructura de Datos Conjunto (SET) en una terminal de Python

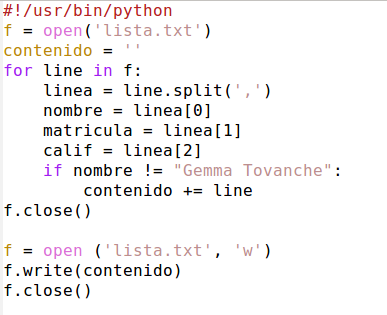


Ilustración 3. Ejemplo de Archivos Secuenciales en Python