**Proyecto 1**

Curso Sistemas Operativos 2021-01

Juan Camilo Naranjo Piedrahita

Este proyecto trata de crear un código que nos permita crear una base de datos en memoria que almacene registros de estudiantes

La lógica que implemente en mi algoritmo es crear un espacio reservado en memoria para una n cantidad de estudiantes, cada registro de un estudiante tiene almacenado la cedula, nombre y el semestre que cursa.

La base de datos en realidad es un int\* que almacena la dirección en memoria del primer registro del espacio reservado donde se guardaran los registros de los estudiantes

Se tienen 4 variables globales las cuales son:

Pestudiantes = dirección del espacio reservado

dbSize = tamaño de la base de datos

dbName = donde se almacena el nombre de la base de datos

dbCont = contador de los registros ingresados en la base de datos

Se implementan 8 comandos para simular como funciona una terminal los cuales son:

**mkdb nombre tamaño:** crea una base de datos especificando el nombre y la cantidad de registros.

**loaddb nombre**: carga la base de datos en memoria desde el archivo especificado. El comando debe indicar si la base de datos se cargó correctamente o no existe. La base de datos debe cargarse en memoria dinámica antes de poder aplicar los siguientes comandos.

**savedb nombre**: este comando salva la base de datos en el archivo especificado.

**readall**: lee todos los registros de la base de datos.

**readsize**: lee la cantidad de registros de la base datos.

**mkreg cedula nombre semestre**: crea un nuevo registro en la base de datos.

**readreg cédula**: busca en la base de datos por número de cédula. En caso de encontrar la cédula imprime el registro completo.

**exit**: salir del programa. Antes de terminar debe preguntar si se desea salvar la base de datos en el archivo especificado con el comando loaddb.

Estos son los resultados del proyecto el cual buscaba entender y manejar correctamente los espacios en memoria reservados mediante un código en lenguaje C

Explicación y resultados

**mkdb**: el comando mkdb crea una base de datos nueva, el formato que tiene es mkdb nombre tamaño, el usuario lo ingresa tal cual es el formato, al leer con scanf el espacio se lee como un \n entonces solo basta con hacer 3 scanf seguidos para almacenar el comando, el nombre y el tamaño en variables distintas (esta explicación es para todas las entradas desde el buffer que tiene el programa).

Ahora si, como funciona mkdb… al dar inicio del programa, se crea una variable la cual se llama dbName, pestudiantes, dbSize y dbCont que inicia en 0, al ejecutar el proceso de mkdb se almacena el nombre de la base de datos en dbName, el tamaño en dbSize y se crea en un nuevo espacio reservado en memoria donde se van a almacenar los registros de los estudiantes, esa dirección del primer registro se almacena en el puntero pestudiantes

**mkreg:** este comando crea un nuevo registro en la base de datos, lo que en realidad hace es almacenar el cc, nombre y semestre, datos que ingresan desde consola, los almacena en el siguiente registro vacio en la memoria reservada, la forma en que lo hace es con el dbCont… por ejemplo, el siguiente registro vacio se ubica en la dirección (pestudiantes + dbCont) primero valida que dbCont sea menor que dbSize si no lo es, muestra un mensaje de error, luego al almacenar el registro correctamente aumenta el contador en 1 y si dbCont es igual a dbSize nos avisa que la base de datos está llena



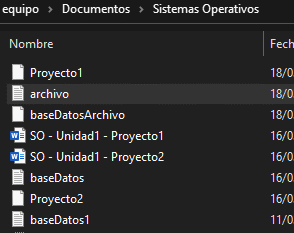
**readall: e**ste comando lee todos los registros que hay en la base de datos, la manera en que lo hace es con un for donde lee todos los registros en el espacio reservado en memoria desde (pestudiantes + 0) hasta (pestudiantes + dbCont) así nos aseguramos de leer todos los registros que han sido ingresados y no alguno que este en vacio

**readsize:** este comando lee la cantidad de registros que hay en la base de datos la forma en que lo hace es simplemente imprimiendo en consola el valor de la variable dbCont la cual es el contador de los registros ingresados satisfactoriamente



**readreg:** este comando busca en los registros un numero de cedula, la manera en que lo hace es recorriendo la memoria reservada desde (pestudiantes + 0) hasta (pestudiantes + dbCont) buscando que la cedula del registro sea igual a la cedula buscada, si la encuentra, imprime el registro y se sale del método, si no lo hace, al terminar el ciclo se ejecuta una línea donde se imprime en consola que el registro no se encontró

**loaddb:** este comando carga un archivo que contiene la base de datos, el archivo en su primera línea tiene el nombre de la base de datos y el tamaño, en las siguientes líneas se encuentran los registros con la siguiente forma: cc nombre semestre



**savedb:** este comando guarda la base de datos con el nombre indicado desde la consola, la manera en que lo hace es creando un archivo y en la primera línea del archivo almacena el nombre y el tamaño de la base de datos, en las siguientes líneas almacena los datos de los registros que están en el espacio en memoria reservado

**exit:** este comando cierra el programa, antes de hacerlo pregunta si desea salvar la base de datos, si la respuesta es sí, ejecuta el proceso del comando savedb



En el main la manera de implementar los comandos es dentro de un while wl cual siempre espera una entrada por consola y compara si la entrada es igual a un comando, cuando se cumple esa condición se ejecuta el proceso del comando que cumple, este ciclo se ejecuta hasta que ingresa el comando exit y cambia la condición del ciclo