

Analisis 1er parcial

Informatica II

Juan Pablo Mesa Marin

Departamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
Marzo de 2023

Índice

1. Analisis del problema	2
2. Esquema de tareas	3
3. Algoritmo implementado	4
4. Problemas de desarrollo que presentó	4
5. Evolucion del algoritmo	4

1. Analisis del problema

El parcial se divide en tres partes, para comenzar nos centraremos en las dos primeras y conforme se vaya avanzando se pensará una posible solución para la tercera.

Analisis 1:

Comenzando con la primera parte se nos pide registrar la información de una materia, todo estará ubicado en un archivo txt ya creado, por lo cual necesitaremos una función que abra un archivo txt y escriba sobre el, los datos serían: Código, nombre, HTD, HTP, nc.

La idea sería crear una función que reciba cómo parametro de entrada la dirección de un archivo.txt y no retorne nada ya que solo necesitamos que pida información y la escriba en el archivo. La función creará un arreglo de `char[5][30]` (`char[30], char[30], char[30], char[30], char[30]`) Utilizar caracteres especiales en este caso elegiré `”` al final de cada arreglo de char para saber el fin de la palabra, es decir una vez en usuario ingresé la información, en la última posición pondré `”` para indicar que la palabra finalizó, esto nos servirá para cuando toque escribir el `char[30]` en el archivo txt, para que solo recorra hasta que encuentre el carácter `”`. Una vez se tenga el arreglo creado se procederá a pedirle la información al usuario en orden para luego de llenarlo, recorrer el arreglo mientras se va escribiendo en el archivo .txt carácter a caracter, luego de terminar pondríamos un `para` dejar el cursor en el renglón de abajo y eso sería todo lo que hacía la 1ra función que se encarga de la 1ra parte del programa.

Analisis 2:

Ahora enfocándonos en la segunda parte necesitaremos registrar horarios de una materia con conocer su código. Entonces haremos una función sin parametros de entrada y no retorne nada, ya que el horario será ingresado en un archivo .txt ya creado. En el cuerpo de la función se le pedirá al usuario el codigo de la materia y el de horarios a ingresar, ya teniendo ese dato crearemos un arreglo en memoria dinámica de tamaño `n+2`, donde `n` es el de horarios de la materia, tipo `char[30], char[30], ... , n+1` enesimo `char[30]` en este arreglo la posición 0 almacenará el código de la materia, la posición 1 el de horarios de la materia y las otras posiciones los diferentes horarios. El formato a seguir al momento de ingresar el horario será el siguiente: en caso de que la materia tenga una sola clase se ha de ingresar (L12-14) si son dos (WV10-12) y siendo tres (MJ8-10,M6-9). Ya después de ingresar los horarios y que quede listo el arreglo dinámico, procederemos a escribir en el txt, en la 1ra línea irá el código de la materia, la 2da el de horarios, y las siguientes los horarios, una línea por cada horario

Analisis 3:

Para la parte tres se nos pide un algoritmo que recomiende espacios de estudio pero para esto 1ro tenemos que poder registrar materias, así que empecemos por ahí.

1ro declararemos una variable `n` igual a 1, declararemos un puntero triple en memoria dinámica de tamaño `n][4][30]`, una vez el usuario matricule un curso se aumentara `n` y se creará un puntero triple temporal en memoria dinámica de tamaño `n][4][30]`, se hará un `for` para copiar las datos del puntero original

al temporal, se liberará la memoria del puntero original para luego volver a ser declarado de tamaño[n][4][30] recordemos que n aumento 1, ahora copiar los datos del puntero temporal al original y liberar la memoria del temporal y así sucesivamente cada que matricule un nuevo curso, también declararemos un puntero triple en memoria dinámica de tamaño[7][24][30] 7 elementos (días) que contienen 24 elementos (horas) cada uno y con 30 caracteres en cada posición, inicialmente la llenaremos de "Libre", para así representar las 24 horas de cada día de la semana, el "libre" significa que esa hora tal como la palabra lo dice está libre y en caso de que esté ocupada, llevará el nombre de la materia en dicha hora.

2do se creará una función con el puntero triple de los cursos ya matriculados como parametro de entrada y sin retorno, que se encargará de abrir el archivo de los cursos e imprimir todo lo que hay dentro de el, evaluando mientras imprime línea por línea si esa materia no está ya matriculada, para hacer esta evaluación sacaremos el código de las materias matriculadas, este dato se encuentra en el puntero triple, al finalizar cerramos el archivo. Esta función se invocará cada que el usuario desee registrar una materia nueva.

3ro crearemos una función para matricular, los parametros de entrada son el puntero triple de los cursos ya matriculados, el puntero triple que representa las horas de cada día, no tendrá retorno ya que modificará por referencia. En el cuerpo de la función le pediremos al usuario que ingrese el código de la materia a matricular, luego abriremos el archivo de los cursos para apartir del código ingresar los datos que necesitamos en el puntero de los cursos, luego de cerrar el archivo de los cursos abriremos el archivo de los horarios y se imprimiran los horarios de la materia elegida, una vez impresos los horarios de la materia elegida se cerrará el archivo, y se procederá a pedirle al usuario que ingrese el horario deseado tal cual como se le muestre en la impresión, una vez ingresado el horario, se procederá a desglosar el char[30] para así hallar las posiciones de la matriz en las cuales se pondrá el nombre de la materia (recordar que la matriz representa las horas de toda la semana), después de tener las posiciones se procederá a abrir el archivo de los cursos para así apartir del código de la materia, hallar el nombre y ese nombre será el que se ingrese en las posiciones de la matriz que representa las horas.

4to crearemos una función para ver horario actual (solo clases con profesor) con el puntero triple que representa las horas de cada día como parametro de entrada, no tiene retorno ya que solo será de imprimir. Se recorrerá el puntero con for para imprimir lo que contiene cada hora dando así el horario.

2. Esquema de tareas

- 1 - Crear repositorio
- 2 - Hacer el analisis punto a punto del parcial
- 3 - Implementar algoritmo en base al analisis

4 - Probar algoritmo

5 - Grabar video

3. Algoritmo implementado

4. Problemas de desarrollo que presentó

1er inconveniente: Para empezar se tenia pensado poner caracteres especiales al final de cada dato ingresado por el usuario para llevar un control de hasta donde leer pero se tuvo un inconveniente y fue que los arreglos de char solo almacenan palabras sin espacios, por ende ya no es necesario el caracter especial ya que puede ser remplazado por

2do inconveniente: el 2do inconveniente es el formato a seguir para registrar los horarios y como leerlos para trabajar con ellos, aun no tengo una idea muy clara de como hacerlo

5. Evolucion del algoritmo

14/04/2023 Se empieza con la 1ra parte que es una funcion que recibe como parametro de entrada el nombre de un archivo de texto para luego en el cuerpo de la funcion crear un puntero doble con memoria dinamica para almacenar los datos ingresados por el usuario, los mismos datos que luego seran leidos para ser escritos en un archivo .txt con un ciclo for que recorra el doble puntero, para finalizar se eliminan los arreglos creados con memoria dinamica para liberar esos espacios de memoria.

17/04/2023 Se continua con una funcion que no tiene parametros de entrada, ni retornos pero se encarga de pedirle al usuario el codigo de la materia y el de horarios, luego se crea un puntero doble con memoria dinamica para almacenar los datos, para luego con un for recorrer el doble puntero e ir escribiendo en el txt donde se encuentran los horarios de los cursos