Universidad Politécnica de Madrid

Grado en Ingeniería Electrónica y Automática

Asignatura

INFORMÁTICA

Curso 2016-2017

Universidad Politécnica de Madrid

Grado en Ingeniería Electrónica y Automática

Datos del Grupo

Eduardo Gibert Almela 54627
David Tertre Boyé 54882

Introducción

Vamos a hacer un programa que sirva para descifrar una contraseña numérica escrita en un fichero. Además informará del tiempo que ha tardado en hacerlo.

Desarrollo

Para llevar a cabo las funciones del programa, se define una dimensión N=9 para el número de dígitos de la contraseña (como máximo), ya que pasados los 10 se alargan demasiado las operaciones.

El programa también cuenta con una estructura de datos llamada **CONT,** utilizada en la función **Cambiocontraseña.** Contiene en su interior dos cadenas de caracteres, una para la contraseña, **cad[N],** y otra para la contraseña verificada, **cadv[N].**

El programa cuenta con cuatro funciones aparte del main.

```
1) void Titulo(void);
```

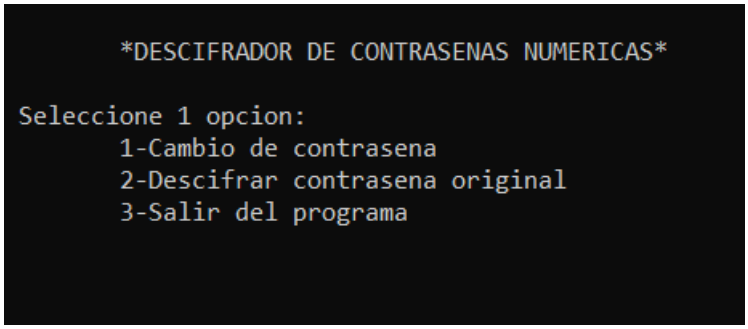
Esta función se ejecuta al inicio del programa, no recibe nada y no devuelve nada, solo se encarga de imprimir el titulo del programa por pantalla.

Al pulsar cualquier tecla la función acaba, la pantalla se limpia con system(“cls”), y pasa a la siguiente función , el menú principal.

```
2) int Panelp(void);
```

Se muestra al usuario el menú principal donde se permiten tres opciones: 1-cambiar la contraseña, 2-descifrarla o 3-salir del programa. La función no recibe nada, pero devuelve un número. Según el número que devuelva la función, el usuario entra en una de las 3 opciones anteriores. Si no

pones ninguno de los 3 números anteriores, el programa te dice que ese número no es posible y te reinicia el panel mediante un do-while.




```
3) void Cambiocontraseña(struct CONT *);
```

Permite al usuario entrar en una interfaz en la que se le permite renovar la contraseña y se le muestran los requisitos de ésta.
Se utilizan tres bucles “do-while”, cada uno dentro de otro, para verificar las condiciones de cada dígito introducido.
La función recibe el puntero a la estructura CONT y no devuelve nada.

Utilizamos **la asignación dinámica de memoria** (malloc) para reservar espacio en la memoria(como mucho N dígitos), y (realloc) para liberar los espacios “no acupados” por la contraseña dependiendo del numero de dígitos que haya introducido el usuario.

Para registrar la clave se usa el puntero a la estructura **CONT**. La segunda(cadv) se introduce para confirmar la primera.

Los dígitos son mostrados en pantalla como asteriscos y en caso de que el usuario presione la tecla de retroceso, se eliminará el último carácter introducido (se sustituye por ‘\0’) y se borrará un asterisco. Para finalizar la escritura de la clave el usuario presiona ENTER, lo que provoca el cierre del primer bucle.
A través de un bucle “for” se verifican las condiciones de la contraseña y en caso de fallar, vuelve a comenzar el bucle principal, en caso de acertar, pasa al siguiente bucle “do-while” que pide la contraseña por segunda vez, para asegurar que el usuario la recuerda o que no se ha equivocado en escribirla. Las 2 cadenas se comparan con la función strcmp, de la librería string.h.
Si coinciden, se cierra al fin el bucle principal y la contraseña es actualizada y se guarda en el fichero Contraseña.txt. En caso contrario, la función se reinicia.

```
4) int Descifrador(int *);
```

Esta función es mucho más simple que la anterior. Tal y como podemos observar, entra un puntero (int *) de la contraseña escaneada previamente del fichero llamado Contraseña.txt, y devuelve el número descifrado (int).
Tan sólo cuenta con un bucle "for", cuya variable comienza en 0 hasta 10 elevado a la dimensión N (**pow(10,N)**). El condicional "if" compara el número de la variable con el de la contraseña. Si son distintos, se le suma uno y continúa, y si son iguales, se cierra el bucle con el valor último de la variable: la contraseña descifrada.

En la función *main*, a parte de las variables, se declaran los ficheros. Uno de ellos sirve para escribir el código a descifrar (**Contraseña.txt**) y el otro para mostrar la clave ya decodificada con el tiempo empleado en hacerlo (**Resultado.txt**).

Se ejecuta la función del título y, a continuación, la del menú, donde se presentan tres opciones: cambiar la contraseña, descifrarla o salir del programa.

En el primer caso, se inicia la función "Cambiocontraseña". Una vez cambiada la clave, se abre el fichero que la contiene (en modo escritura "w"). En caso de estar dañado el archivo, se muestra un mensaje de "apertura incorrecta" y se regresa al menú principal. Sino, se accede con normalidad al fichero, se reemplaza la contraseña y se cierra para volver al menú principal.

En el segundo caso, abre el fichero **Contraseña**.txt (ahora en modo lectura "r") y realiza un bucle de lectura del fichero para guardar la clave y compararla posteriormente. Posteriormente se abre el fichero **Resultado.txt** (modo "w") sobre el que se va a escribir.
Comienza a cronometrar, captando el instante actual con la función comienzo=clock(); (de la biblioteca time.h) y guardándolo en la variable "comienzo".
Inmediatamente después se inicia la función de descifrador y cuando ésta finaliza, se graba nuevamente el instante actual en la variable "final". El tiempo de decodificación es por lo tanto igual a "final" – "comienzo".
La contraseña decodificada y el tiempo ocurrido quedan grabados en el fichero Resultado.txt y se imprimen por pantalla.
Se cierran los 2 ficheros y se regresa al menú principal.

Por último, en el tercer caso, se muestra un mensaje de “Hasta la vista!” y el programa finaliza (saliendo del bucle principal mediante un flag).

Si no se elige ninguno de estos tres casos, se muestra un mensaje de “Seleccione una opción posible” y se repite el bucle principal.

Conclusiones

A diferencia de lo que creíamos, ha sido mucho más laborioso crear una función que permita registrar una contraseña, con unos valores determinados, que una función que se dedique a decodificarla (NUMERICA). Hemos intentado que la contraseña tenga caracteres también, pero descodificarla era bastante complicado y solo lo conseguíamos para cadenas de caracteres muy cortas, así que al final optamos que fuera numérica. Nos ha costado pensar cada bucle y función (sobre todo el de cambio de contraseña), pero el resultado ha valido la pena. Nos ha parecido una forma muy buena de repasar y afianzar los conocimientos que hemos adquirido durante el curso, sobre todo pensando en cursos posteriores.

Bibliografía

Libro “Introducción a la programación en C”.
Apuntes tomados en clase.
Tutorialspoint.com (para la biblioteca time.h).