Fundamentos de los Sistemas Operativos
Ficha de entrega de práctica

**: campo obligatorio*
IMPORTANTE: esta ficha no debe superar las DOS PÁGINAS de extensión

Grupo de prácticas*: Grupo 1 - 41
Miembro 1: Alejandro Vialard Santana
Miembro 2:

Número de la práctica*: 2	**Fecha de entrega*: 08/03/2022**

Descripción del trabajo realizado*
El objetivo principal de esta práctica era realizar una versión propia de la librería <string.h> que contiene el lenguaje C. Para lograrlo, se han creado diferentes funciones que la librería original presenta (mi_strlen, mi_strcpy, mi_strcat, mi_strup y mi_strequals) que nos permitirán trabajar con cadenas de caracteres. Otro de los objetivos, es aprender a utilizar punteros, que en ciertas funciones hemos tenido que utilizarlos.

Para la implementación del código hemos seguido lo siguiente:

- *Método **mi_strlen:** recorremos por completo la cadena pasada por parámetro hasta llegar al carácter nulo, que indica el final de la string. A su vez, por cada carácter que se encuentra, se aumenta en uno un contador que indica el número de caracteres que contiene, dando lugar a que al final de el tamaño.*
- *Método **mi_strcpy:** almacenamos en una variable el inicio del puntero de s1, ya que se va a ir avanzando. Luego, mientras la cadena el contenido de la cadena que se quiere copiar no sea nulo, se igualará el contenido de s2 al de s1, e incrementamos ambos punteros para ir avanzando. Por último, se añade el carácter nulo al puntero de s1 y retornamos el inicio de s1.*
- *Método* **mi_strcat:** creamos una variable que sea un puntero que apunta al final de la cadena s1. Después, hacemos un bucle mientras el carácter de s2 no sea nulo (que indica que ha llegado al final de la cadena) e incrementamos el puntero creado, por tanto aumenta el tamaño de la cadena y lo igualamos al contenido de s2, lo que hará que se concatene al final de s1 el primer valor de s2. Para finalizar, se añade el carácter nulo a la variable puntero y se retorna la dirección de s1.
- Método **mi_strup:** creamos una variable puntero y en ella utilizamos la función malloc() para reservar espacio en la memoria dinámica con el tamaño de la string (con el método mi_strlen). Luego, se retorna mi_strcpy del puntero creado y la string, de tal manera que se copian los caracteres en el puntero. Por último, utilizamos free() en el main para liberar la memoria.
- Método **mi_strequals:** hacemos un bucle que mientras s1 o s2 no lleguen al final, se compara el contenido de s1 con el de s2, y si son distintos se retorna 0. Incrementamos s1 y s2 para ir avanzando en las posiciones de las cadenas. Si no se cumple, quiere decir que no hay valores distintos y se devuelve un 1.

Además de editar el código de la librería, hemos añadido diferentes tests para cada uno de los métodos en el archivo proporcionado "test_mistring.c".

Horas de trabajo invertidas* Miembro 1: 3 horas aproximadamente Miembro2:

Cómo probar el trabajo*
*En la carpeta comprimida estará el **test_mistring.c**, que es el código de las pruebas para los métodos implementados; también habrá una carpeta llamada "lib", en ella hemos guardado todo lo relacionado con la biblioteca. Si accedemos a esa carpeta, veremos el* **libmistring.a** *que era necesario crearlo para poder utilizar la librería (mediante el comando "ar -crs libmistring.a mistring/mistring.o"); también hay una carpeta llamada "mistring" en ella está el archivo **mistring.c** donde están implementados los códigos de los métodos, **mistring.h** que es la cabecera y **mistring.o**, que es el código objeto de la biblioteca.*

Ahora, para poder ejecutarlo realizaremos los siguientes pasos:

Grupo de prácticas*: Grupo 1 - 41
Miembro 1: Alejandro Vialard Santana
Miembro 2:

Número de la práctica*: 2	**Fecha de entrega*: 08/03/2022**

1. *Mediante la terminal accederemos a la ruta donde tengamos la carpeta descomprimida, pongamos que es Documentos. Por tanto haremos, "cd Documentos/practica2/".*
2. *Ahora estaremos en la carpeta principal, en la que se encuentra la carpeta "lib" y el test_mistring.c (para comprobarlo podemos usar el comando ls -l).*
3. *Después necesitamos compilar el archivo "test_mistring.c" para poder ver las pruebas. Para ello, estando en la ruta "Documentos/practica2/", haremos el siguiente comando "gcc test_mistring.c -o test_mistring -lmistring -L./lib".*
4. *Por último, tendremos que ejecutarlo con "./test_mistring", y aparecerá en pantalla las pruebas realizadas.*

En caso de querer ver el código tanto de las pruebas como de mistring, debe de situarse en la ruta de cada archivo y ejecutar "gedit nombre_archivo" en caso de querer un editor de texto con interfaz, si no puede usar vim, que es un editor de texto en línea de comandos.

Incidencias
(errores no resueltos, anomalías, cualquier cosa que se salga de lo normal)

Comentarios
(observaciones adicionales que quieran anotar)