Práctica 2: Libería de manipulación de cadenas de caracteres.

Diseño de Sistemas Operativos

Pablo Castro Valiño (pablo.castro1@udc.es)

Curso 2012-2013

Índice

1. Procedimiento para la realización de la práctica.

1.1. Creación de la librería.

1.1.1. Introducción.

Para la implementación de esta librería de manipulación de strings mediante técnicas de mapping, la primera decisión que nos hemos visto obligados a tomar, ha sido motivada por la necesidad de devolver al usuario la misma posición de memoria que él nos ha proporcionado.

Para solventar esta dificultad hemos creado una estructura de datos (*cadena*) compuesta por un puntero a una estructura de iov's y el tamaño de dicha estructura. De esta forma, cuando hagamos cambios en el string, crearemos uno nuevo y guardaremos su posición de memoria en nuestra estructura *cadena* al igual que su nuevo tamaño, devolviendo al usuario el puntero a la *cadena* correspodiente que nos había pasado.

1.1.2. STR_INSERT_V.

Función que realiza la insercción del string 'origen' en 'destino' en la posicion indicada y sin sobreescritura.

1.1.3. STR_DELETE_V.

Función que realiza el borrado de un rango de caracteres de un string.

1.1.4. STR_DUP_V.

Función que realiza una copia de una cadena en otra.

1.1.5. STR_CAT_V.

Función para la concatenación de dos strings dados.

1.1.6. STR_CMP_V.

Función de comparación de dos strings.

1.1.7. STR_PRINTF_V.

Función que imprime una cadena.

1.1.8. [OPT] INSERT_OVER.

Función que realiza una copia de un string existente en otro en la posición indicada, básicamente es igual que *STR_INSERT_V* pero con sobreescritura.

1.2. Diseño y ejecución de pruebas experimentales.

1.2.1. Introducción y estructura de los tests.

La implementación de las pruebas se ha dividido en dos ficheros, uno para las pruebas de funcionamiento (**tests_funcionamiento.c**(6)) y otro para las de tiempos (**tests_tiempos.c**(4)). También se ha separado en otro fichero **tiempos_lib.c**(5), las funciones que realizan las mediciones de los tiempos.

1.2.2. Tests de funcionamiento.

Para implementación de estos tests, se han tenido en cuenta diversos casos generales y que probablemente se podrían ampliar en caso de disponer de más tiempo pero que deben ser suficientes para comprobar el correcto funcionamiento de la práctica.

Comparación: Las pruebas de comparación que se han realizado han sido con tamaños de nodo iguales variando si los strings eran iguales o diferentes y con tamaños de nodos distintos, también variando la igualdad de los strings.

Insertado: Se han realizado insercciones en la primera y última posición de una cadena y al principio, en el medio y al final de un nodo intermedio.

Borrado: Se han eliminado caracteres dentro de un nodo y también entre nodos, borrando el final de uno y el principio del siguiente. Del mismo modo se ha contemplado el borrado del primer y del último nodo completo, de dos nodos completos y por último el borrado del único nodo existente en el string.

Concatenado: Únicamente se ha procedido a realizar una simple concatenación entre dos cadenas de dos nodos cada una.

Copia: Para la copia únicamente se ha creado una nueva cadena con el valor de la anterior.

[OPT] Insertado con sobreescritura: Finalmente, para la función de insertado con sobreescritura se han considerado los casos que se indican a continuación.

Se ha copiado un string de un nodo con el mismo tamaño que el segundo nodo del destino, de forma que se sobreescribiese un nodo completamente. Y por último se ha comprobado la copia entre nodos, al igual que en la primera posición y la última.

1.2.3. Tests de duración.

Como se ha observado mediante una comprobación empírica, los tiempos de inserción no son los mismos para cada inserción independiente ya que la posible aparición de interrupciones en el sistema no está bajo nuestro control, por lo tanto se ha decidido aplicar las funciones un número determinado de iteraciones y calcular el tiempo medio de realización de las mismas.

No se ha realizado comprobación alguna, en referencia al posible valor de error que puedan devolver las funciones implementadas en estos tests ya que su comprobación implicaría aumentar el tiempo de ejecución de las pruebas. Este es el motivo por el que si introducimos índices incorrectos, la aplicación después de mostrar el error, producirá un *segmentation fault*.

1.3. Conclusiones.

Como conclusiones cabe destacar la correcta implementación de la mayor parte de funciones. Todas a excepción de la comparación y la parte optativa obtienen mejores resultados que las funciones típicas de manejo de strings.

Se han realizado tests con strings de tamaños comprendidos entre 2K y 1M en los cuales podemos ver, como a medida que los strings crecen, nuestras funciones mantienen el tiempo constante mientras que las de tratamiento habitual de strings, aumentan fuertemente.

2. Índice de archivos adjuntos.

1. **Makefile** Se ha introducido un fichero *Makefile* para agilizar la compilación.

   - **make all** Compila todos los ficheros.
   - **make tests** Compila los tests a realizar.
   - **make clean** Elimina todos los ficheros ejecutables.
   - **make tar** Crea un tar.gz con todos los ficheros fuente.

2. **strings_lib.c**
   Contiene la parte obligatoria de la librería de manipulación de strings a traves de las estructuras 'cadena' y también la función optativa de insercción con sobreescritura.

3. **cadena_lib.c**
   Fichero cuyo contenido se corresponde a la implementación de las estructuras 'cadena' utilizadas en la libería de strings, asi como las funciones necesarias para el uso de estas estructuras. Estas funciones serán las que se le proporcionarán al usuario para que pueda hacer uso de nuestra *string_lib*.

4. **tests_tiempos.c**
   Fichero en el que se encuentran implementados los tests que se han realizado para comprobar la eficiencia temporal en la ejecución de nuestra liberería.

5. **tiempos_lib.c**
   Fichero que contiene las funciones de medición de tiempos para cada una de las funciones anteriormente implementadas en **strings_lib.c** (2).

6. **tests_funcionamiento.c**
   Fichero similiar a **tests_tiempos.c**(4) en el que se encuentran los tests creados para comprobar el correcto funcionamiento de las funciones de **strings_lib.c** (2).

7. **old_strings_lib.c**
   Fichero que contiene las implementaciones de insertado, borrado y insertado con sobreescritura para los strings habituales.