

## CC3301 Programación de Software de Sistemas

### Tarea 2 – Semestre Primavera 2021 – Prof.: Luis Mateu

En esta tarea Ud. deberá programar 2 funciones para mover un segmento de un string a otra parte del mismo string. Las 2 funciones pedidas son:

```
void desplazar(char *str, int from, int nbytes, int by);  
char *desplazamiento(char *str, int from, int nbytes, int by);
```

La función *desplazar* mueve *nbytes* partiendo de la posición *from* de *str* en *by* caracteres hacia la derecha si *by* es positivo o en *|by|* caracteres hacia la izquierda si *by* es negativo. Esta función altera el mismo string *str* que recibe como parámetro. La función *desplazamiento* hace lo mismo que *desplazar*, solo que no cambia el parámetro *str*, si no que retorna un nuevo string resultante de mover los caracteres pedidos. Los siguientes son ejemplos de uso de *desplazar*, considerando que *str* es “*abcdefghijkl*”.

Llamada	<i>str</i> inicial	Resultado en <i>str</i>
<code>desplazar(str, 2, 3, 2);</code>	“ <i>abc<u>de</u>fg hij</i> ”	“ <i>abcd<u>cde</u>hij</i> ”
<code>desplazar(str, 5, 4, -3);</code>	“ <i>abcde<u>fgh</u>ij</i> ”	“ <i>ab<u>fgh</u>ighij</i> ”
<code>desplazar(str, 5, 4, 3);</code>	“ <i>abcde<u>fgh</u>ij</i> ”	“ <i>abcdefg<u>hfg</u></i> ”

El primer ejemplo es un desplazamiento a la derecha. El segmento del string que se mueve aparece subrayado tanto en el string inicial como en el resultado. Observe que los 2 primeros caracteres que son desplazados se preservan en su ubicación original. El segundo ejemplo es un desplazamiento a la izquierda. En este caso 3 de los caracteres son preservados en su posición original. El último ejemplo es un caso de desborde. Se pide mover los caracteres más allá de los límites del string. Ud. no debe escribir más allá de los límites del string.

*Restricción:*

- No use el operador de subindicación de arreglos `[ ]` ni su equivalente `*(p+i)`, en su lugar use aritmética de punteros, privilegiando los operadores `++` `--` `+=` `-=`.
- En *desplazar* no use *malloc* para pedir memoria auxiliar. Si se desplaza a la derecha Ud. debe copiar los caracteres de derecha a izquierda. Si se desplaza a la izquierda, copie de izquierda a derecha.
- En *desplazamiento* sí debe usar *malloc* para pedir el espacio ocupado por el resultado.

### Instrucciones

Baje *t2.zip* de U-cursos y descomprímalo. El directorio *T2* contiene los archivos (a) *test-desplazar.c* que prueba si su tarea funciona y compara su eficiencia con la solución del profesor, (b) *prof.ref* con el binario ejecutable de la solución eficiente del profesor, (c) *desplazar.h* que incluye el encabezado de las funciones pedidas, y (d) *Makefile* que le servirá para compilar y ejecutar su tarea. Ud. debe programar las funciones *desplazar* y *desplazamiento* en el archivo *desplazar.c*. Su solución debe ser eficiente, lo que se traduce en que no debe tomar más de un 80% de tiempo de CPU que la solución eficiente del profesor.

Ejecute el comando *make* sin parámetros bajo Debian 11. Le explicará qué requisitos debe cumplir para aprobar su tarea, cuáles son las opciones de compilación y ejecución, cómo entregar su tarea, cómo borrar los archivos intermedios y cuál es el trabajo del comando *make*.

### Entrega

Ud. debe entregar por U-cursos el archivo *desplazar.zip* que genera el comando *make zip*. Recuerde descargar el archivo que subió, descomprimirlo e inspeccionar el contenido para verificar que son los archivos correctos. Se descuenta medio punto por día de atraso, sin considerar recesos, sábados, domingos y festivos. Además se descontará otro medio punto si la compilación arroja *warnings*, aunque pase todos los tests.