

Detalles de implementación

Para lograr una mayor división del trabajo y con el objetivo de testear mejor las distintas funcionalidades que fueron necesario programar para la elaboración del tp, se decidió dividir al código en distintas librerías e incluir los distintos archivos *.h* en caso de necesitarlos.

La primera librería escrita fue *bit_library.h*. En esta, se pueden encontrar distintas funciones que permiten realizar operaciones con bits, desde encontrar si un bit dentro de un byte está encendido o no hasta elaborar máscaras determinadas. Esta librería fue codificada sabiendo que las funciones que provee serían necesarias para realizar las operaciones necesarias para realizar el *encoding* y *decoding*.

En segundo lugar, nos encontramos con la librería *base_64.h*. Esta librería es la que servirá como interfaz para el programa principal. En ella, se encuentran las funciones encargadas de realizar el *encoding* y el *decoding* de los distintos bytes que se pasen por parámetro.

Por último, tenemos el problema principal. En la función *main()* hay un ciclo que recorre el vector de parámetros que se recibe al ejecutar el programa, y los va procesando hasta terminar. En caso de que haya algún error en el pasaje de parámetros, se imprime un error por **stderr**. Los errores están relacionados a un mal orden en los parámetros, a la utilización de nombres incorrectos y también al pasaje de una cantidad incorrecta de los mismos. Luego, en el programa hay dos funciones que se encargan de codificar o decodificar un archivo, que son las que se encargan de leer el archivo de entrada, codificar o decodificar según corresponda, y escribir sobre el archivo de salida.

Comandos de compilación

Corridas de prueba

A continuación se detallan las corridas de prueba del programa. Se intentó que éstas sean lo más abarcativas posibles, mostrando casos en los que funciona el encoding y el decoding, tomando archivos reales físicos y también los archivos estándar de entrada y salida, así como también se intenta mostrar cuáles son todos los posibles errores por los cuáles el programa muestra un error.

Empecemos entonces por algunas corridas triviales por *stdin*, para mostrar que se codifica y decodifica correctamente.

Codificamos el caracter ASCII M

Código fuente

Código MIPS32