

PROYECTO FINAL “ZIP”

PRIMERA ENTREGA

ORGANIZACION DEL PROCESADOR 1949

INTEGRANTES:

- LUCAS ARMAS
- EZEQUIEL DEPETRIS
- GASTON MASSIMINO

PROYECTO FINAL ZIP – ORGANIZACION DEL PROCESADOR

Codificador

El proyecto consiste en implementar un programa que sea capaz de comprimir una cadena de caracteres, no el alfabeto completo sino solo los caracteres 'A', 'B', 'C' y 'D'. Para realizar dicha implementación se utilizara el lenguaje assembler (80x86). Además, se cuenta con un programa C que cumple la función de ser el programa main, desde donde se llamaran a las subrutinas necesarias para resolver el programa. El problema se lo resolvió con el uso de subrutinas y a continuación se nombrara y mencionara brevemente que hace cada una de ellas:

- **Encode:** es la subrutina “principal” desde el cual se llaman a todas las subrutinas necesarias para la resolución del problema de codificación y es esta misma, quien sera llamada desde el programa C para responder a lo solicitado para esta primera entrega.
- **Counter:** Recorre una vez el arreglo que contiene las letras a codificar contando la ocurrencia de cada una de las letras y almacenando en otro arreglo.
- **Linker:** Da valores a la tabla dependiendo de las ocurrencias de las letras. Las codifica dándole cadenas binarias de 0's y 1's. El orden de la tabla coincide con el orden alfabético por lo que la primera posición corresponde a la 'A' mientras que la ultima a la 'D'
- **MaxElemPos:** subrutina sencilla que lo que hace es devolver la posición en la que se encuentra el máximo elemento de un arreglo y luego en esa posición asigna -1 para que la próxima vez que se llame, no se vea a ese numero como máximo.
- **Codificador:** Deja almacenado en un arreglo de binarios la secuencia final de la codificación y compresión, completando el ultimo componente del arreglo con todos 0s para indicar el fin de secuencia con “00”.
- **Cant:** Subrutina corta que simplemente se le pasa un registro y lo que hace es devolver la cantidad de veces que se debe shiftear.
- **Div4:** Como se necesita trabajar en 8bits, se usa esta subrutina para dividir un registro de 32bit en 4 para obtener la parte baja.

Decodificador

La segunda parte de este proyecto consiste en realizar el procedimiento inverso al que realiza la primera parte. Es decir, desde el mismo programa C se incluirea una funcion que decodifique la cadena de caracteres, usando los datos resultantes de la codificacion, para asi obtener la cadena de caracteres originales.

Para realizar dicha tarea se necesito implementar las siguientes subrutinas, las cuales seran explicadas a continuacion:

- **AssignLetter:** Subrutina muy breve que hace comparaciones utilizando la tabla de codificación para ver a que letra corresponde la codificación que se le pasa.
- **Decoded:** Subrutina que recibe una cadena codificada, una tabla con las codificaciones para cada letra y lo que hace es relacionarlas para volver a obtener la cadena original.