# 

# **Escuela Superior Politécnica del Litoral**

# 

# **OAC-Organización y Arquitectura de**

# **Computadores**

# **Desarrollo de Procedimiento “mirror” en MIPS Variable 2: UPPER**

**Integrantes:**

**Aguayo Vinza Neycker Aldrich**

**Ferrín Alcívar Alex Roberto**

**Larrea Sánchez Julio Alfredo**

**Fecha:**

**2016/06/20**

# 

# Resumen:

El proyecto a resolver trata sobre hacer una función que permita, tomando una palabra, invertirla y agregarla a la palabra original. Además, en esta variación del problema, se procede a transformar todas las letras en mayúsculas. Los resultados fueron satisfactorios y cumplieron con las condiciones iniciales del problema.

# Proceso de investigación

## **Identificación del problema:**

Usando MIPS, escribir en memoria a lado del mensaje original el mensaje invertido, además de llevar todos los caracteres a mayúsculas.

## **Objetivos:**

El objetivo principal del grupo es resolver el problema planteado en la identificación del problema, para ello se plantearon algunos objetivos secundarios:

* Entender como poder recorrer caracter por caracter la palabra.
* Averiguar cómo poder agregar la primera palabra al final de la cadena inversa.
* Buscar los valores de las letras en formato ASCII (tabla ASCII de caracteres)

## **Planificación**:

El proyecto se realizó completamente la primera semana de plazo, junto con el proceso de investigación, y la semana 2 se presentó informalmente el proyecto al docente a cargo de este. La presentación oficial será el día de entrega del proyecto establecida en el cronograma.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Actividad** | **Semana01** | **Semana02** | **Semana03** | **Semana04** |
| Proceso de Investigación | x |  |  |  |
| Solución del problema | x |  |  |  |
| Presentación del proyecto |  | x |  | x |

## **Revisión de soluciones existentes**:

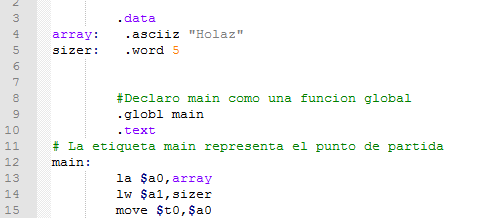
Actualmente no se buscaron soluciones existentes con respecto al proyecto, pero para la resolución del mismo, se tomó como base un ejemplo del libro guía, donde describe la función “clear”, que dado el caso recorría un arreglo de enteros y los ponía en cero.

Observando esa implementación, se entendió cómo llevar rastro de la dirección de memoria de la palabra, caracter por caracter, hasta haberla recorrido toda.

# Solución del problema

## **Descripción de la solución propuesta:**

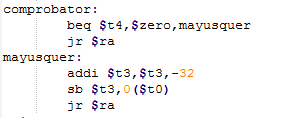
La solución propuesta fue cargando la dirección de memoria de la palabra en un registro, también su máximo tamaño en otro registro, para poder saber el máximo de iteraciones requeridas.



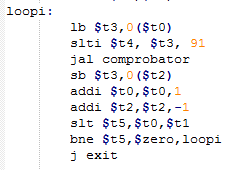
También tendremos en un registro la dirección de la palabra ya agregada el inverso ((tamaño máximo de la palabra 2x-1).



Obtenemos uno a uno, los caracteres de la palabra original, verificamos si es menor a 91 (número decimal del caracter después de la Z mayúscula), de ser ese el caso, lo convertimos a mayúscula, restándole 32 y guardándolo en memoria, donde estaba ubicado el puntero.



Luego se ubica también ese carácter en el puntero que apunta en la dirección de la palabra con el inverso agregado, sumamos 1 al puntero de la palabra y restamos 1 al puntero de la palabra inversa, verificamos si el puntero ya apunta al máximo de la palabra y de ser el caso salimos del programa.



# Conclusiones

Podemos concluir que el programa brinda solución al problema especificado, siempre y cuando se indique en la sección de DATA (datos) además de la cadena, el tamaño de esta.

Se concluye que se puede escribir la palabra en reverso usando un puntero ubicado en el último espacio y recorriendo hacia atrás.

# Bibliografía

* Computer Organization and Design – The Hardware/Software Interface. David A. Patterson & John L. Hennessy