

Organización del Procesador

2024

Trabajo Práctico Obligatorio

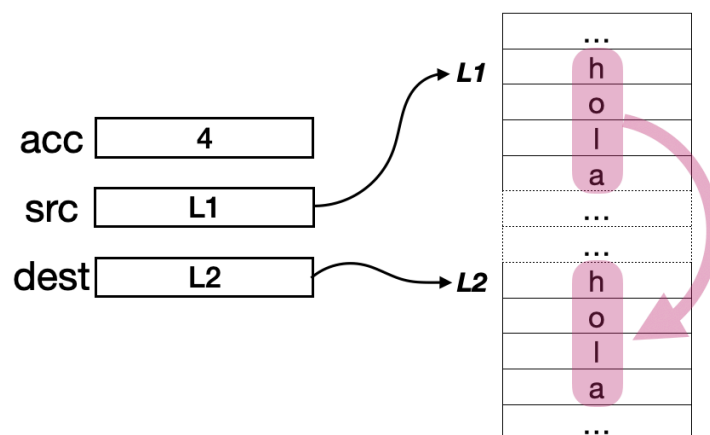
Este trabajo deberá ser resuelto en grupos de hasta 3 participantes. Los integrantes del grupo se podrán distribuir el trabajo pero TODOS deben conocer en profundidad parte del mismo y de manera general su totalidad. El mismo será evaluado de manera individual y la evaluación formará parte de la nota general del alumno (cada integrante) en la asignatura. Fecha límite de entrega: jueves 21 de noviembre. Las defensas podrán llevarse a cabo durante el transcurso hábil del presente cuatrimestre en horarios que se coordinarán previamente..

El objetivo del proyecto es mejorar la Arquitectura Wombat, incorporando una instrucción que permita copiar una cadena (string) de una longitud n , de una posición a otra en memoria. Para ello se propone agregar 3 (tres) instrucciones a la arquitectura por defecto.

- Una instrucción (**loadsrc**) que permita establecer la dirección origen (primera dirección donde se encuentra la cadena a copiar)
- Una instrucción (**loaddest**) que permita establecer la dirección destino (primera dirección de donde se pretende copiar la cadena)
- Una instrucción (**strcopy**) que copia la cadena dirección por dirección desde el origen al destino con la longitud indicada en el **acc**.

Como ejemplo de uso, supongamos que tenemos reservado una etiqueta **L1** con la cadena "hola" (**L1: .ascii "hola"**) y una **L2** inicialmente sin ningún texto o simplemente 0. Luego de cargar 4 (operando inmediato) en el **acc** y **L1** y **L2** en los registros **src** y **dest** respectivamente, al ejecutar **strcopy**, el resultado debería ser como el descrito en la figura siguiente:

```
...  
load 4  
loadsrc L1  
loaddest L2  
strcopy  
...
```



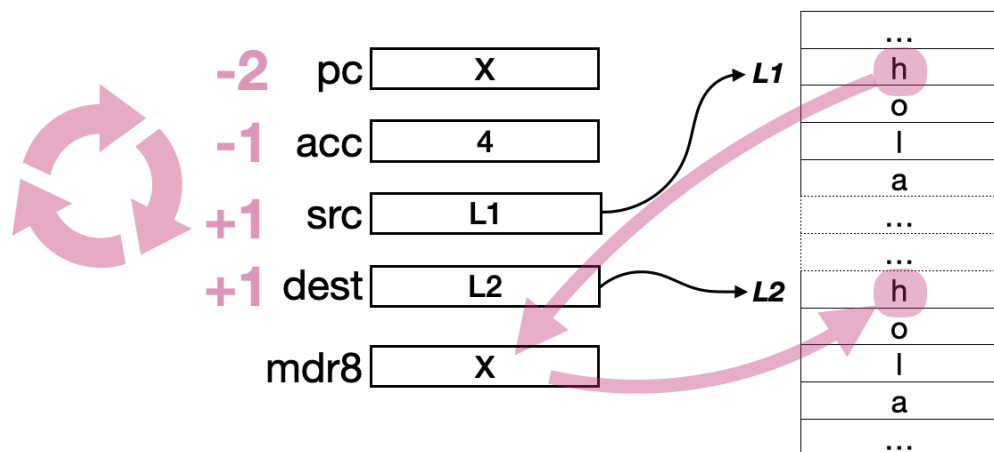
Pasos y ayuda a considerar:

- Inicialmente utilizar la **wombat** para declarar y ejecutar un programa que utilice cadenas (**L1: .ascii "<string>"**), y observar cómo se carga en memoria (ocupa un byte por letra como en **nasm**).
- Deberán crear 2 registros **src** y **dest** para almacenar las correspondientes direcciones. Con ello también agregar las 2 primeras instrucciones solicitadas **loadsrc** y **loaddest**.
- Luego enfocarse en la creación del **strcopy**, para ello, lo primero que hay que tener en cuenta es el tamaño de un carácter (1Byte) y la actual interfaz de datos con la memoria **mdr** (2Bytes). Para poder copiar de un carácter, se debe crear un registro

de interfaz que sea de 1Byte, con el nombre a elección, por ej: **mdr8**. Luego crear para éste las microinstrucciones que permitan su conexión y uso con la memoria (similares a las 2 del mdr): *Main[mar]->mdr8* y *mdr8->Main[mar]*.

Luego, una idea de implementación de la **strcopy** sería la siguiente:

- si el acc es 0 entonces **ignorar**(omisión) las siguientes microinstrucciones hasta el final *End*.
- copiar en el **mar** la dirección origen, leer el caracter al **mdr8**, copiar al **mar** el destino, escribir el caracter que está en **mdr8**. Con esto se copia un caracter de la cadena origen en la de destino.
- actualizar para que se **repita** la operación con el próximo caracter. Para esto y creando microinstrucciones de incremento y decremento de los registros necesarios, debería incrementar **src** y **dest**, para operar con el próximo caracter, decrementar el **acc** (un caracter de la cadena ya se copió) y deshacer el avance a la próxima instrucción que produjo el **fetch** decrementando el **pc**.



Cómo entregar el proyecto:

SOLO un integrante del grupo deberá entregar un archivo comprimido (.zip) que contendrá.

- Un archivos de texto indicando
 - Nombre, Apellido y DNI de cada integrante del grupo
 - Notas o explicaciones a tener en cuenta durante la revisión del trabajo.
- El archivo .cpu (Máquina) que contiene la implementación de las instrucciones del proyecto
- Al menos un archivo de prueba, es decir, un programa que utilice la funcionalidad incorporada.