

Laboratorio ARM

Organización del Computador

Parte 3

Objetivo

Acceder a valores en memoria con ldr

Almacenar valores en memoria con str

Iterar sobre vectores en memoria

Temario

- Operaciones de acceso a memoria
 - Load (ldr)
 - Store (str)
- Arrays

Operaciones de memoria

Repaso

Ya vimos uno de los usos de `ldr` para manejar strings

Repaso

Ya vimos uno de los usos de `ldr` para manejar strings

```
.data
mensaje:
    .asciz "hola_mundo"

.text
ldr r0, =mensaje
```

Repaso

Ya vimos uno de los usos de `ldr` para manejar strings

```
.data
mensaje:
.asciz "hola_mundo"

.text
ldr r0, =mensaje
```

- Carga la dirección de inicio de "hola_mundo" en r0

Enteros en memoria

La directiva `.word` define un entero de 32 bits en memoria, así como `.asciz` define una cadena en memoria

Enteros en memoria

La directiva `.word` define un entero de 32 bits en memoria, así como `.asciz` define una cadena en memoria

```
.data
mensaje:
    .asciz "hola_mundo"
entero1:
    .word 42
entero2:
    .word 38
```

Leer enteros desde memoria

Paso 1: usar `ldr` para cargar la dirección en un registro

Leer enteros desde memoria

Paso 1: usar `ldr` para cargar la dirección en un registro

```
.data
entero1:
    .word 42
entero2:
    .word 38
.text
ldr r0, =entero1
```

Leer enteros desde memoria

Paso 2: usar `ldr` con `[]` para cargar el valor en la memoria

```
.data
entero1:
    .word 42
entero2:
    .word 38
.text
ldr r0, =entero1
ldr r1, [r0]
```

Almacenar enteros en memoria

Paso 1: usar `ldr` para cargar la dirección en un registro

Almacenar enteros en memoria

Paso 1: usar `ldr` para cargar la dirección en un registro

```
.data
entero1:
    .word 42
entero2:
    .word 38
.text
```

```
ldr r0, =entero1
```

...	
entero1	42
entero2	38
...	

Almacenar enteros en memoria

Paso 2: usar `str` para almacenar una valor en esa dirección

```
.data
entero1:
    .word 42
entero2:
    .word 38
.text
ldr r0, =entero1
mov r1, #57
```

...	
entero1	42
entero2	38
...	

Almacenar enteros en memoria

Paso 2: usar `str` para almacenar una valor en esa dirección

```
.data
entero1:
    .word 42
entero2:
    .word 38
.text
ldr r0, =entero1
mov r1, #57
str r1, [r0]
```

...	
entero1	57
entero2	38
...	

Práctica I

Imprimir y reemplazar enteros
almacenados en memoria

Arrays

Definiendo Arrays

Única diferencia con las variables:

Se especifican múltiples valores con la directiva `.word`

Definiendo Arrays

Única diferencia con las variables:

Se especifican múltiples valores con la directiva `.word`

```
.data  
entero1:  
    .word 42
```

Definiendo Arrays

Única diferencia con las variables:

Se especifican múltiples valores con la directiva `.word`

```
.data
entero1:
    .word 42
array:
    .word 32, 65, 76, 87
```

Accediendo Arrays

Enfoque básico: incrementar la dirección de memoria

Accediendo Arrays

Enfoque básico: incrementar la dirección de memoria

```
.data
arr:
    .word 32, 65, 76
    .text
ldr r0, =arr
ldr r1, [r0]
add r0, r0, #4
ldr r2, [r0]
add r0, r0, #4
ldr r3, [r0]
```


Accediendo Arrays

Enfoque básico: incrementar la dirección de memoria

```
.data
arr:
    .word 32, 65, 76
    .text
ldr r0, =arr
ldr r1, [r0]
add r0, r0, #4
ldr r2, [r0]
add r0, r0, #4
ldr r3, [r0]
```

-Offset de 4 bytes porque una palabra es de 4 bytes

Práctica 2

Registro Indirecto

Más sobre acceso a memoria

```
ldr r3, [r0]
```

La instrucción anterior utiliza el modo de direccionamiento indirecto por registro

Modo de direccionamiento

Cómo el procesador accede a algo

Registro indirecto

Acceso a la memoria realizado a través de una dirección en un registro

Práctica 3

Mínimo valor en un vector

Práctica 4

Sumar constante a un vector