

## • búsqueda binaria

### • Data

message: .asciz "ingrese el numero: "

esp: .asciz " "

.align 2

array: .space 20

val: .asciz "Ingrese el valor que busca"

res: .asciz "el valor se encuentra en  
el indice: "

N-res: .asciz "el valor no se encuentra"

### • text

#### main:

la \$t0, array # vector

li \$t1, 0 # indice

li \$t2, 5 # n

#### Hacer:

beq \$t1, \$t2, reset #

li \$v0, 4

la \$a0, message

syscall

li \$v0, 5

syscall

sw \$v0, 0(\$t0) # Guardar en vector

addi \$t0, \$t0, 4 #

addi \$t1, \$t1, 1

S llenar

reset:

li \$t1, 0 # low

li \$t2, 4 # high

la

Valor:

li \$v0, 4

la \$a0, val

syscall

li \$v0, 5

syscall

move \$s0, \$v0 # s0 es la el valor ingresado

while:

blt \$t2, \$t1, N-encontrado

add \$t3, \$t1, \$t2 # t3 = low + high

srl \$t4, \$t3, 1 # mid = t3 / 2

mul \$t6, \$t4, 9

la \$t0, array # rencio base

add \$t0, \$t0, \$t6 # direccion arraymid ]

lw \$t5, 0(\$t0) # T5 = arraymid ]

beq \$s0, \$t5, encontrado

blt \$s0, \$t5, menor

addi \$t1, \$t4, 1 # low = mid + 1

as while

menos:

addi \$t2, \$t4, -1 # high = mid - 1

as while

N-encontrado:

Li \$VO, 4

La, \$AO, \$res

syscall

\$exit

encontrado:

Li \$VO, 4

La \$AO, \$res

syscall

Li \$VO, 1

move '\$AO, \$T4' #imprimir indice mrd

syscall

exit:

Li \$VO, 10

syscall