Sean Mulhall

Computer Org

Project 3

.data

buffer: .byte 0:200

upper: .byte 0:26

lower: .byte 0:26

str1: .asciiz "Enter a string:\n"

str2: .asciiz "The transformed string:\n"

str3: .asciiz "Enter in the symbols you want to use for uppercase characters:\n"

str4: .asciiz "Enter in the symbols you want to use for lowercase:\n"

nl: .asciiz "\n"

.text

main: addi $v0, $zero, 4

la $a0, str3

syscall # prints prompt for upper case

addi $v0, $zero, 8

la $a0, upper

addi $a1, $zero, 26

syscall # reads a string

addi $v0, $zero, 4 # newline

la $a0, nl

syscall

addi $v0, $zero, 4

la $a0, str4

syscall # prints prompt for lower case

addi $v0, $zero, 8

la $a0, lower

addi $a1, $zero, 26

syscall # reads a string

addi $v0, $zero, 4 # newline

la $a0, nl

syscall

strings:

addi $v0, $zero, 4

la $a0, str1

syscall # prints prompt for string to be transformed

addi $v0, $zero, 8

la $a0, buffer

addi $a1, $zero, 200

syscall # reads a string

la $s0, buffer #$s0 = address of buffer

addiu $t2, $zero, 10 # loads newline character into $t2

for: lbu $t0, 0($s0) #$t0 = byte we are examinging

beq $t0, $zero, exit #terminate on empty string

beq $t0, $t2, endfor #stops if encounters newline

andi $t1, $t0, 0xDF #see if it is a letter by seeing if it can convert to uppercase

sltiu $t3, $t1, 65 # A = 65

bne $t3, $zero, next

sltiu $t3, $t1, 91 # Z = 90

beq $t3, $zero, next

andi $t1, $t0, 0x20

beq $t1, $zero, uppercase #see if its upper or lowercase

subi $t4, $t0, 97 #get array index of what character we want to #swap using $t4 as the offset

la $s1, lower #grab the address of the lower string

add $s1, $s1, $t4

lbu $s2, 0($s1)

sb $s2, 0($s0)

j next

uppercase:

subi $t4, $t0, 65 #get array index of what character we want to #swap using $t4 as the offset

la $s1, upper #grab the address of the upper string

add $s1, $s1, $t4

lbu $s2, 0($s1)

sb $s2, 0($s0)

j next

next: addi $s0, $s0, 1

j for

endfor:

addi $v0, $zero, 4

la $a0, str2

syscall # prints the message str2

addi $v0, $zero, 4

la $a0, buffer

syscall # prints the transformed string

j strings

exit: addi $v0, $zero, 10

syscall # exit

Enter in the symbols you want to use for uppercase characters:

!@#$%^&\*ABCDEFGHIJKLMNOPQ

Enter in the symbols you want to use for lowercase:

RSTUVWXYZabcdefghijklmnop

Enter a string:

INTRODUCITON

The transformed string:

AFLJG$M#ALGF

Enter a string:

Welcome to this book! We're delighted

The transformed string:

OVcTfdV kf kYZj Sffb! OV'iV UVcZXYkVU

Enter a string:

to have this opportunity to convey the

The transformed string:

kf YRmV kYZj fggfikleZkp kf TfemVp kYV

Enter a string:

to have this opportunity to convey the

The transformed string:

kf YRmV kYZj fggfikleZkp kf TfemVp kYV

Enter a string:

excitement of the world of computer systems.

The transformed string:

VoTZkVdVek fW kYV nficU fW TfdglkVi jpjkVdj.

Enter a string:

This is not a dry and dreary field, ...

The transformed string:

LYZj Zj efk R Uip ReU UiVRip WZVcU, ...

Enter a string:

The instruction add “$s0, $zero, $s1” copies the number

The transformed string:

LYV ZejkilTkZfe RUU $j0, $Enter a string:

in register $s1 into register $s0.

The transformed string:

Ze iVXZjkVi $j1 Zekf iVXZjkVi $j0.

Enter a string: