Clayton Mentzer

cmentzer@ccis.neu.edu

Homework Assignment 2

Computer Organization

10/4/12

.globl main

.data

initial\_string: .asciiz "The initial array is: "

space: .asciiz ", "

.align 5

strings:.asciiz "Joe"

.align 5

.asciiz "Jenny"

.align 5

.asciiz "Jill"

.align 5

.asciiz "John"

.align 5

.asciiz "Jeff"

.align 5

.asciiz "Joyce"

.align 5

.asciiz "Jerry"

.align 5

.asciiz "Janice"

.align 5

.asciiz "Jake"

.align 5

.asciiz "Jocelyn"

.align 5

.asciiz "Jessie"

.align 5

.asciiz "Jess"

.align 5

.asciiz "Jane"

.align 5

mynames: .space 56 #This array will be initialized in .text

size: .word 13

main:

.text

#for i = 0 to i = 13, mynames[i] points to strings[i]

#t0 is i, t1 is the array of pointers mynames, and t2 is the array of strings

#strings.

li $t0, 0

la $t5, size

lw $t5, ($t5)

la $t1, mynames

la $t2, strings

initialize:

sw $t2, 0($t1)

add $t1, $t1, 4

add $t2, $t2, 32

add $t0, $t0, 1

ble $t0, $t5, initialize

addi $t1, $t1, -56

la $t5, size

lw $t5, ($t5)

li $t0, 0

la $a0, initial\_string

li $v0, 4

syscall # Print the text stating "The inital array is: "

print:

lw $a0, 0($t1)

li $v0, 4

syscall

la $a0, space

syscall

add $t1, $t1, 4

add $t0, $t0, 1

ble $t0, $t5, print

sort:

la $t5, size

lw $t5 ($t5)

beq $t0, $t5, print\_final

la $a0, 0($t1)

addi $t1, $t1, 4

la $a1, 0($t1)

str\_lt: # $t0 is x, $t1 is y

addi $sp, $sp, -16 # create call frame

sw $ra, 0($sp) # save return address

move $t0, $a0

move $t1, $a1

loop: # $t2 is \*x, $t3 is \*y

lb $t2, 0($t0)

lb $t3, 0($t1)

beq $t2, $zero, end\_loop # if \*x -- '\0', exit loop

beq $t3, $zero, end\_loop # if \*y -- '\0', exit loop

blt $t2, $t3, returnOne # if ( \*x < \*y ) return 1

blt $t2, $t2, returnZero # if ( \*x < \*y ) return 0

addi $t0, $t0, 1 # x++ (x is of type 'char \*'; So, add 1 for next char)

addi $t1, $t1, 1 # y++ (y is of type 'char \*'; So, add 1 for next char)

b loop # go to next iteration of for loop

end\_loop:

beq $t3, $zero, returnZero # if ( \*y == '\0' ) return 0

b returnOne # return 1

returnZero:

li $v0, 0 # return 0

b fncDone

returnOne:

li $v0, 1 # return 1

fncDone:

lw $ra, 0($sp) # restore return address

addi $sp, $sp, 16 # free call frame

print\_final:

li $v0 10

syscall