CS-35101

Thomas Moore

5/1/2022

Summary:

The first section of the code I modified and created was a loop for detecting and reversing the case of a character. After that I jump and link to the findMin section of code. There I use addition and several comparisons to attempt to find the minimum ASCIII character.

Conclusion:

This lab was rather challenging not just coding wise but comprehension wise too. This is the first time I had to complete a coding project in MIPS where the code was mostly completed but I had to create my own code to connect it all together. I was very confused about the minimum ASCII character section, but I think my code will work for the requested outputs. The only other problem I could not fix was an asterisk printing at the end of my reversed string.

Lab 5 Code:

```
1.  # Starter code for reversing the case of a 30 character input.
2.  # Put in comments your name and date please.  You will be
3.  # revising this code.
4.  #
5.  # Created by Dianne Foreback
6.  # Students should modify this code
7.  #
8.  # This code displays the authors name (you must change
9.  # outpAuth to display YourFirstName and YourLastName".
10. #
11. # The code then prompts the user for input
12. # stores the user input into memory "varStr"
13. # then displays the users input that is stored in"varStr"
14. #
15. # You will need to write code per the specs for
16. # procedures main, revCase and function findMin.
17. #
18. # revCase will to reverse the case of the characters
19. # in varStr.  You must use a loop to do this.  Another buffer
20. # varStrRev is created to hold the reversed case string.
21. #
```

```
# Please refer to the specs for this project, this is just starter code.
#
# In MARS, make certain in "Settings" to check
# "popup dialog for input syscalls 5,6,7,8,12"
#
        .data     # data segment
            .align 2   # align the next string on a word boundary
outpAuth:   .asciiz  "This is Thomas Moore presenting revCaseMin.\n"
outpPrompt: .asciiz  "Please enter 30 characters (upper/lower case mixed):\n"
            .align 2   #align next prompt on a word boundary
outpStr:    .asciiz  "You entered the string: "
        .align 2   # align users input on a word boundary
outpStrRev: .asciiz   "\nYour string in reverse case is: "
        .align 2   # align the output on word boundary
varStrRev:  .space 32  # reserve 32 characters for the reverse case string
            .align 2   # align  on a word boundary
outpStrMin: .asciiz    "\nThe min ASCII character after reversal is: "
varStr:     .space 32  # will hold the user's input string thestring of 20 bytes
                # last two chars are \n\0  (a new line and null char)
                # If user enters 31 characters then clicks "enter" or hits the
                # enter key, the \n will not be inserted into the 21st element
                # (the actual users character is placed in 31st element).  the
                # 32nd element will hold the \0 character.
                # .byte 32 will also work instead of .space 32
        .align 2   # align next prompt on word boundary
myChar:             .byte 'a'
#
        .text     # code section begins
        .globl      main
main:
#
# system call to display the author of this code
#
            la $a0,outpAuth          # system call 4 for print string needs address of string in $a0
            li $v0,4            # system call 4 for print string needs 4 in $v0
            syscall

#
# system call to prompt user for input
#
            la $a0,outpPrompt       # system call 4 for print string needs address of string in $a0
            li $v0,4            # system call 4 for print string needs 4 in $v0
            syscall
#
```

```
66. # system call to store user input into string thestring
67. #
68.             li $v0,8            # system call 8 for read string needs its call number 8 in $v0
69.                                 # get return values
70.             la $a0,varStr       # put the address of thestring buffer in $t0
71.             li $a1,32                   # maximum length of string to load, null char always at end
72.                                         # but note, the \n is also included providing total len < 22
73.         syscall
74.         #move $t0,$v0                   # save string to address in $t0; i.e. into "thestring"
75. #
76. # system call to display "You entered the string: "
77. #
78.             la $a0,outpStr  # system call 4 for print string needs address of string in $a0
79.             li $v0,4            # system call 4 for print string needs 4 in $v0
80.             syscall
81. #
82. # system call to display user input that is saved in "varStr" buffer
83. #
84.             la $a0,varStr       # system call 4 for print string needs address of string in $a0
85.             li $v0,4            # system call 4 for print string needs 4 in $v0
86.             syscall
87. #
88. # Your code to invoke revCase goes next
89. #
90.             jal revCase #invoke revCase
91.
92. # Exit gracefully from main()
93.         li   $v0, 10      # system call for exit
94.         syscall          # close file
95.
96.
97. ###################################################################
98. # revCase() procedure can go next
99. ###################################################################
100.            #  Write code to reverse the case of the string.  The base address of the
101.            # string should be in $a0 and placed there by main().  main() should also place into
102.            # $a1 the number of characters in the string.
103.            #  You will want to have a label that main() will use in its jal
104.            # instruction to invoke revCase, perhaps revCase:
105.            #
106.            revCase:
107.                    addi $t0, $zero,0 #loop var
108.                    la $t7, varStr
109.            while:
```

```
110.                bgt $t0, $a1, exit #loop while var is less than 31
111.                lb $t2, 0($t7)
112.                blt $t2, 96, upper #checks to see if lowercase branch if not
113.                blt $t2, 65, skip
114.                addi $t2, $t2, -32 #subtracts 32 from lowercase char
115.                sb $t2, 0($t7)
116.                addi $t0, $t0, 1 #increment loop
117.                addi $t7, $t7, 1 # add 4 to index
118.                j while
119.        upper:
120.                addi $t2, $t2, 32 #adds 32 to uppercase char
121.                sb $t2, 0($t7)
122.                addi $t0, $t0, 1 #increment loop var
123.                addi $t7, $t7, 1 # add 4 to index
124.                j while
125.
126.        skip:
127.                addi $t0, $t0, 1 #increment loop var
128.                addi $t7, $t7, 1 # add 4 to index
129.                j while
130.
131.        exit:
132.                move $t5, $ra
133.
134.
135.        #
136.        # After reversing the string, you may print it with the following code.
137.        # This is the system call to display "Your string in reverse case is: "
138.                la $a0,outpStrRev       # system call 4 for print string needs address of string in
    $a0
139.                li $v0,4        # system call 4 for print string needs 4 in $v0
140.                syscall
141.        # system call to display the user input that is in reverse case saved in the varRevStr
    buffer
142.                la $a0,varStr   # system call 4 for print string needs address of string in $a0
143.                li $v0,4        # system call 4 for print string needs 4 in $v0
144.                syscall
145.
146.        #
147.        # Your code to invoke findMin() can go next
148.                jal findMin
149.        # Your code to return to the caller main() can go next
150.                move $ra, $t5
151.                jr $ra
```

```
152.
153.
154.        ######################################################################
155.        # findMin() function can go next
156.        ######################################################################
157.        #  Write code to find the minimum character in the string.  The base address of the
158.        # string should be in $a0 and placed there by revCase.  revCase() should also place into
159.        # $a1 the number of characters in the string.
160.        #  You will want to have a label that revCase() will use in its jal
161.        # instruction to invoke revCase, perhaps findMin:
162.        #
163.        #
164.        findMin:
165.                addi $t0, $zero,0 #loop var declared
166.                la $t6, varStr #load the desired index
167.                lb  $t3, 0($t6)
168.
169.        while2:
170.                bgt $t0, $a1, exit2 #loop while var is less than 31
171.                lb $t2 0($t6) #load the desired index
172.                blt $t2, 65 ,skip2
173.                bgt $t3, $t2, minChar
174.                addi $t0, $t0, 1 #increment loop var
175.                addi $t6, $t6, 1 # add 4 to index
176.                j while2
177.        minChar:
178.                move $t3, $t2
179.                addi $t0, $t0, 1 #increment loop var
180.                addi $t6, $t6, 1 # add 1 to index
181.                j while2
182.                        #jr $ra
183.        skip2:
184.                addi $t0, $t0, 1 #increment loop var
185.                addi $t7, $t7, 1 # add 4 to index
186.                j while2
187.        exit2:
188.                sb $t3, myChar
189.
190.        # write use a loop and find the minimum character
191.
192.        #
193.        # system call to display "The min ASCII character after reversal is:  "
194.                 la $a0,outpStrMin         # system call 4 for print string needs address of string in
    $a0
```

```
195.                li $v0,4          # system call 4 for print string needs 4 in $v0
196.                syscall
197.
198.        # write code for the system call to print the the minimum character
199.
200.                li $v0, 4
201.                la $a0, myChar
202.                syscall
203.
204.        # write code to return to the caller revCase() can go next
205.
206.                jr $ra
```

Example Output:

```
This is Thomas Moore presenting revCaseMin.
Please enter 30 characters (upper/lower case mixed):
**** user input : aBcDEFghiJKlMnOpQRstuvwxYZ
You entered the string: aBcDEFghiJKlMnOpQRstuvwxYZ

Your string in reverse case is: AbCdefGHIjkLmNoPqrSTUVWXyz*
The min ASCII character after reversal is: A
-- program is finished running --


Reset: reset completed.

This is Thomas Moore presenting revCaseMin.
Please enter 30 characters (upper/lower case mixed):
**** user input : HGTVsnsnweGTDE
You entered the string: HGTVsnsnweGTDE

Your string in reverse case is: hgtvSNSNWEgtde*
The min ASCII character after reversal is: E
-- program is finished running --
```