CSE3666 — Lab 4

Mike Medved

February 18th, 2022

1 Prompt

In this lab, we refactor the code we wrote in lab 3. We will implement two functions.

1. **Remove Spaces**
   This is a leaf function. We can copy the code in Step 2 from lab3. Revise it if necessary. Note the following:

   - str and res are strings. Like an array, when we pass a string to a function, we only put its starting address in an argument register. For example, register a0 has the starting address of string str.
   - This function is a leaf function. It does not need to use the stack.
   - The function does not return a value.

2. **Print Non-Spaces**
   The second function is print_ns. The prototype is as follows. The steps to be implemented in the function are listed below. Do NOT use la. All the information the function needs is the argument.

   - Save registers. This step should be done later after Steps 2 to 4 are completed.
   - Create local array res of 128 bytes on the stack. We only need to move the stack pointer to allocate space (128 bytes) for the string.
   - Call remove_spaces function to remove spaces in s and save the result in res.
     Note that we need to put string s's address in a0, and res's address in a1.
   - Print res with a system call.
   - Restore registers and stack, and return s.

   Clearly mark each step in your code. Again, we write code for Steps 2, 3, and 4 first so that we know what registers to save and restore in Steps 1 and 5.

   Step through the function and observe the values in registers ra and sp.
   Also examine the stack in Data Segment window.

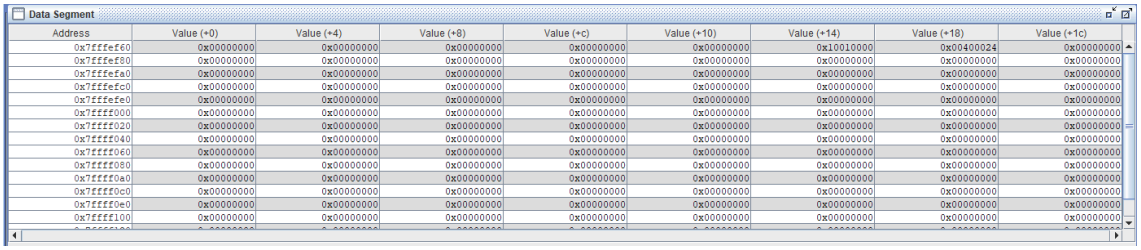
2 Deliverables

```
print_ns:
    # store the result pointer in a1
    add  a1, sp, x0

    # allocate 136 bytes to the stack
    # - 128 for the string
    # - 4 for the return address
    # - 4 for the original string
    addi  sp, sp, -136

    sw    ra, 4(sp)      # store current return address on the stack
    sw    a0, 0(sp)      # preserve original string so we can return it later
    jal   remove_spaces  # invoke remove_spaces

    # use system call 4 to print the result to stdout
    addi  a0, a1, 0
    addi  a7, x0, 4
    ecall

    # restore state so we can return back to the main loop
    lw    ra, 4(sp)
    lw    a0, 0(sp)
    addi  sp, sp, 136
    jr    ra
```

3 Run Examples

```
a b c
abc

abcd e
abcdef

ab c d e f gh i j
abcdefghij

abcdefghijklmnopqrstuvwxyz
abcdefghijklmnopqrstuvwxyz

This string has escape sequences \n and special characters **
Thisstringhasescapesequences\nandspecialcharacters**

RISC-V: The Free and Open RISC Instruction Set Architecture
RISC-V:TheFreeandOpenRISCInstructionSetArchitecture

This string has lots of spaces and other funny characters!!!
Thisstringhaslotsofspacesandotherfunnycharacters!!!
```

4 Data Segment View


Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x7fffef60	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x10010000	0x00400024	0x00000000
0x7fffef80	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffefa0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffefc0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffefe0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff0a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff0c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff0e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000