User Id : weizhe.goh@digipen.edu | Started : 2019.11.28 11:21:25 | Score :

DigiPen CS Vadim Surov CS100A Assignment #10

Assembler - Functions

Practice to use assember to work with functions.

Rules

Read carefully and check all rules you agree with:

- ☐ Each card has description which must be strictly followed.
- ☐ Keep the code properly formatted (correct indentation, line width is 40 characters max, no empty lines).
- ☐ Your code must represent your own individual work. If something is not clear, ask your instructor for help.
- ☐ Cheating of any kind (copying someone else's work, allowing others to copy your work, collaborating, etc.) will not be tolerated and will be dealt with SEVERELY.

★ Problem

- In this assignment you must develop assembly code that sort elements of a given array of integers in ascending order within following specs and restrictions:
  - Sorting algorithm is Bubble Sort
    - Wikipedia – The algorithm description.
  - The array and its length are given as a and n.
  - Must develop and use function swap_all that starting from the beginning of the array, compare every adjacent pair, swap their position if they are not in the right order.
    - The function has one parameter - number of elements to swap.
  - You main assembly code must use above function. Each time this function called the parameter is getting smaller by one element.
- When implementing functions you must use stack to:
  - store and restore all locally used registers,
  - pass parameters,
  - implement prologue and epilogue,
  - for local memory (if needed).

★ Test

Run

```
.macro PRINT
    push  %rax
    push  %rcx
    mov   $fmt, %edi
    xor   %esi, %esi
    movb  (%rbx,%rax), %sil
    xor   %eax, %eax # Clear AL
    call  printf
    pop   %rcx
    pop   %rax

.endm

.macro swap_all size
    mov \size, %rdx

zero:
    mov %rdx, %rcx
    mov %rcx, %rdx
    mov $a, %rdi
```

Survey

- What is approximate number of hours you spent implementing this assignment?

1 day

- Indicate the specific portions of the assignment that gave you the most trouble

reset pointer

```
repeat:
    xor  %rax, %rax
    mov 0(%rdi), %al
    mov 1(%rdi), %bl
    cmpb %bl , %al
    jle  noSwap

    movb %bl, 0(%rdi)
    movb %al, 1(%rdi)

noSwap:
    inc  %rdi
    loop repeat

    dec %rdx
    cmp $0, %rdx
    jne zero

.endm

    .data
fmt: .asciz "%d "
a:  .byte 5,3,8,4,6,2,7,1,9,5
n:  .quad 10
    .text
    .global main
main:
    push  %rbx # For alignment

    # Place your code here
    #  to sort a in ascending order

    swap_all n

    # Output the array
    mov   n, %rcx
    mov   $a, %rbx
    xor   %rax, %rax
print_next:
    PRINT
    inc   %rax
    loop  print_next

    xor   %eax, %eax # return 0;
    pop   %rbx
    ret
```

1 2 3 4 5 5 6 7 8 9

By signing this document you fully agree that all information provided therein is complete and true in all respects.	Responder sign: