

25

3

EXPLORER

OPEN EDITORS

HOMEWORK

ATol

main.asm

main.o

readline.asm

readline.o

writeline.asm

writeline.o

PrintLine

.gitignore

compile.sh

FibonacciFunction

FibonacciFunction.cpp

main

main.asm

OUTLINE

TIMELINE

main.asm

atoi.asm

readline.asm

writeline.asm

itoa.asm

build.sh

finalproject > ASM main.asm

1 global _main

2 extern _exit

3 extern readline

4 extern writeline

5 extern _atoi

6 extern _itoa

7

8 section .bss

9 num1 resq 1 ; reserve space for first 64-bit number

10 num2 resq 1 ; reserve space for second 64-bit number

11 buffer1 resb 1024 ; input buffer for first string

12 buffer2 resb 1024 ; input buffer for second string

13 outbuf resb 1024 ; output buffer for result string

14

15 section .data

16 prompt1 db "Enter number 1: ", 0

17 prompt2 db "Enter number 2: ", 0

18 resultmsg db "The multiplication result is: ", 0

19 newline db 10, 0 ; newline character

20

21 section .text

22 _main:

23 ; ---- Prompt for first number ----

24 lea rdi, [rel prompt1] ; load address of prompt1 into rdi register

25 call writeline ; call writeline to print prompt1

26

27 ; ---- Read first number ----

28 lea rdi, [rel buffer1] ; load address of buffer1 into rdi register

29 mov rsi, 1024 ; buffer size

30 call readline ; call readline to read input from console into buffer1

31

32 lea rdi, [rel buffer1] ; load address of buffer1 into rdi register

33 call _atoi ; call _atoi to convert string to integer

34 mov [rel num1], rax ; store full 64-bit number in num1

35

36 ; ---- Prompt for second number ----

37 lea rdi, [rel prompt2] ; load address of prompt2 into rdi register

38 call writeline ; call writeline to print prompt2

39

40 ; ---- Read second number ----

41 lea rdi, [rel buffer2] ; load address of buffer2 into rdi register

42 mov rsi, 1024 ; buffer size

43 call readline ; call readline to read input from console into buffer2

44

45 lea rdi, [rel buffer2] ; load address of buffer2 into rdi register

46 call _atoi ; call _atoi to convert string to integer

Ln 19, Col 37

Spaces: 4

UTF-8

LF

{ } x86 and x86_64 Assembly