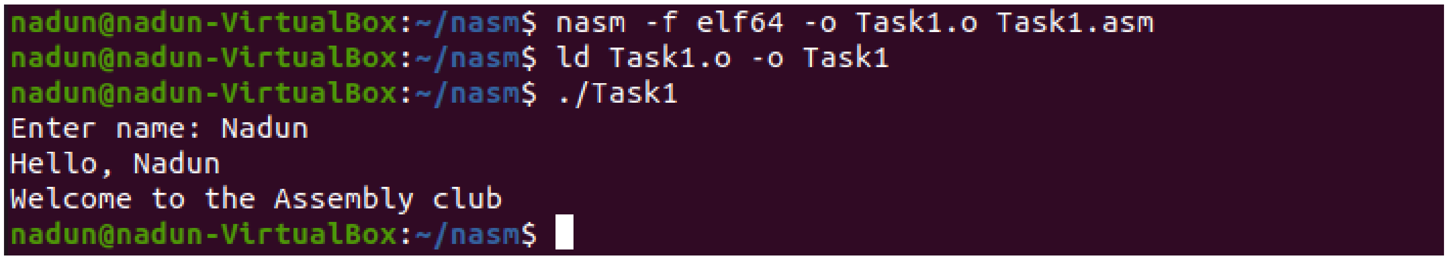
Mini Project **CS/2018/016**

1)

```
;use macro to write a string

%macro write_string 2       ;macro has 2 arguments
        mov rax,1           ;Id of system call for sys_write
        mov rdi,1           ;filedescriptor-standard output
        mov rsi,%1          ;first argument- location of string to write
        mov rdx,%2          ;second argument- length of string
        syscall             ;request a service from the kernel to write
%endmacro


;use macro to read string

%macro read_string 2        ;macro has 2 arguments
        mov rax,0           ;Id of system call for sys_read
        mov rdi,0           ;filedescriptor-standard input
        mov rsi,%1          ;first argument- memory location of input string
        mov rdx,%2          ;second argument- length of string
        syscall             ;request a service from the kernel to read
%endmacro


section .data                                       ;data section
        string1 db "Enter name: "                   ;first string to be printed
        length1 equ $ - string1                     ;length of the string1
        string2 db "Hello, "                        ;second string to be printed
        length2 equ $ - string2                     ;length of the string2
        string3 db "Welcome to the Assembly club",10        ;third string to be printed
                                                    ;and newline character
        length3 equ $ - string3                     ;length of the string3
```

```
section .bss                ;bss section-data is allocated for future use
        name resb 20        ;reserve a space for name

section .text               ;text section- actual code
        global _start       ;link to the label "_start"

_start:                                         ;it is executed first
        write_string string1, length1           ;request the macro to display string1
        read_string name, 20                    ;request the macro to read name
        write_string string2, length2           ;request the macro to display string2
        write_string name, 20                   ;request the macro to display name
        write_string string3, length3           ;request the macro to display string3
        call _exit                              ;call subroutine "_exit"

_exit:                      ;subroutine
        mov rax, 60         ;Id of system call for sys_exit
        mov rdi, 0          ;no errorcode
        syscall             ;request a service from the kernel to exit
```



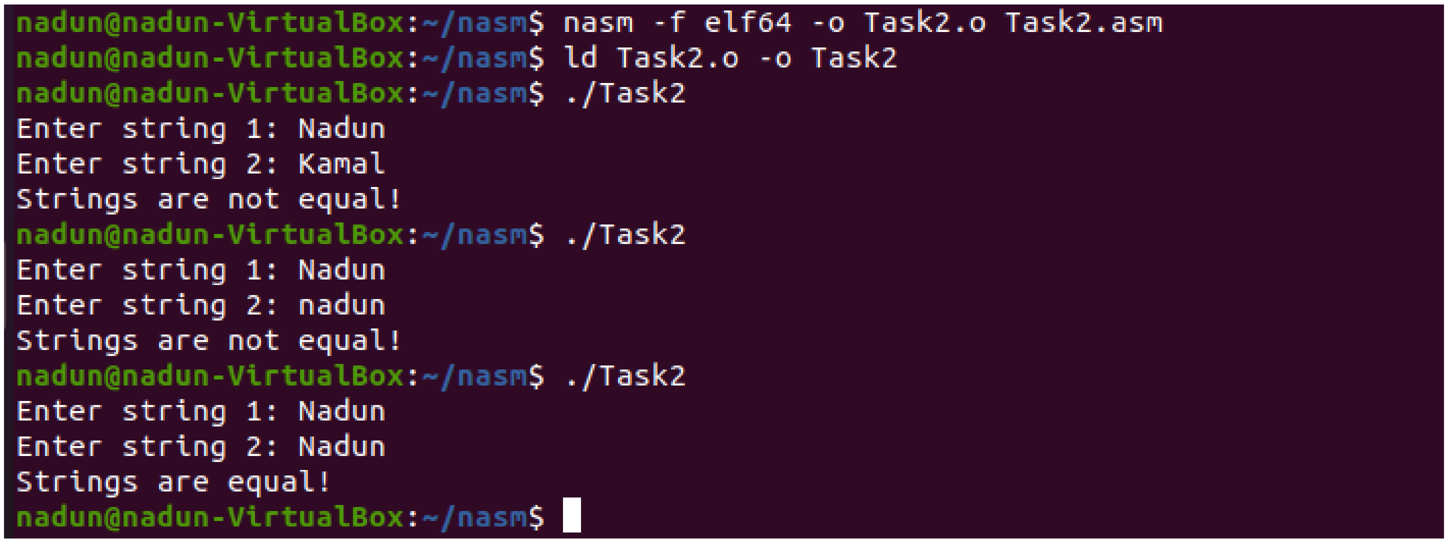
```
nadun@nadun-VirtualBox:~/nasm$ nasm -f elf64 -o Task1.o Task1.asm
nadun@nadun-VirtualBox:~/nasm$ ld Task1.o -o Task1
nadun@nadun-VirtualBox:~/nasm$ ./Task1
Enter name: Nadun
Hello, Nadun
Welcome to the Assembly club
nadun@nadun-VirtualBox:~/nasm$ 
```

2)

```
;use macro to write a string
%macro write_string 2       ;macro has 2 arguments
        mov rax,1           ;Id of system call for sys_write
        mov rdi,1           ;filedescriptor-standard output
        mov rsi,%1          ;first argument- location of string to write
        mov rdx,%2          ;second argument- length of string
        syscall             ;request a service from the kernel to write
%endmacro

;use macro to read string
%macro read_string 2        ;macro has 2 arguments
        mov rax,0           ;Id of system call for sys_read
        mov rdi,0           ;filedescriptor-standard input
        mov rsi,%1          ;first argument- memory location of input string
        mov rdx,%2          ;second argument- length of string
        syscall             ;request a service from the kernel to read
%endmacro

section .data                           ;data section
        string1 db "Enter string 1: "   ;first string to be printed
        length1 equ $ - string1         ;length of the string1
        string2 db "Enter string 2: "   ;second string to be printed
        length2 equ $ - string2         ;length of the string2
        string3 db "Strings are equal!",10      ;if flag is 1, then print string3 and move to newline
        length3 equ $ - string3                 ;length of the string3
        string4 db "Strings are not equal!",10  ;if flag is 0, then print string3 and move to newline
        length4 equ $ - string4                 ;length of the string4
```

```
section .bss                    ;bss section-data is allocated for future use
        str1 resb 20            ;reserve a space for str1
        str2 resb 20            ;reserve a space for str2

section .text                   ;text section- actual code
        global _start           ;link to the label "_start"

_start:                                         ;it is executed first
        write_string string1, length1           ;request the macro to display string1
        read_string str1, 20                    ;request the macro to read str1
        write_string string2, length2           ;request the macro to display string2
        read_string str2, 20                    ;request the macro to read str2

        mov rax, [str1]             ;load the string from str1
        mov rbx, [str2]             ;load the string from str2
        cmp rax, rbx                ;comparisons-check equality
        je _stringEqual             ;if equal,then jump ot label "_stringEqual"
        jno _stringNotEqual         ;if not equal,then jump ot label "_stringNotEqual"



_stringEqual:                       ;label _stringEqual -print when the strings are equal
        write_string string3, length3           ;request the macro to display string3
        call _exit                              ;call subroutine "_exit"
```

```
_stringNotEqual:            ;a label _stringNotEqual -print when the strings are not equal
        write_string string4, length4           ;request the macro to display string4
        call _exit                              ;call subroutine "_exit"


_exit:                      ;subroutine
        mov rax, 60         ;Id of system call for sys_exit
        mov rdi, 0          ;no errorcode
        syscall             ;request a service from the kernel to exit
```



```
nadun@nadun-VirtualBox:~/nasm$ nasm -f elf64 -o Task2.o Task2.asm
nadun@nadun-VirtualBox:~/nasm$ ld Task2.o -o Task2
nadun@nadun-VirtualBox:~/nasm$ ./Task2
Enter string 1: Nadun
Enter string 2: Kamal
Strings are not equal!
nadun@nadun-VirtualBox:~/nasm$ ./Task2
Enter string 1: Nadun
Enter string 2: nadun
Strings are not equal!
nadun@nadun-VirtualBox:~/nasm$ ./Task2
Enter string 1: Nadun
Enter string 2: Nadun
Strings are equal!
nadun@nadun-VirtualBox:~/nasm$
```