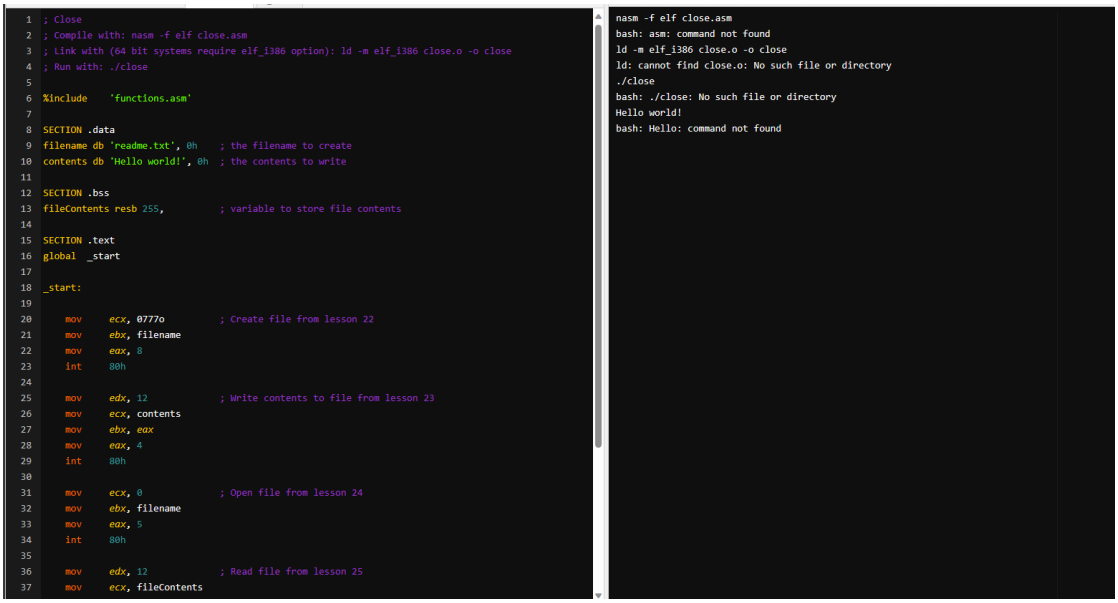
Keene Keannu Kurt C. De Jesus

BSCS3-1

LESSON 26-30 FINAL EXAM

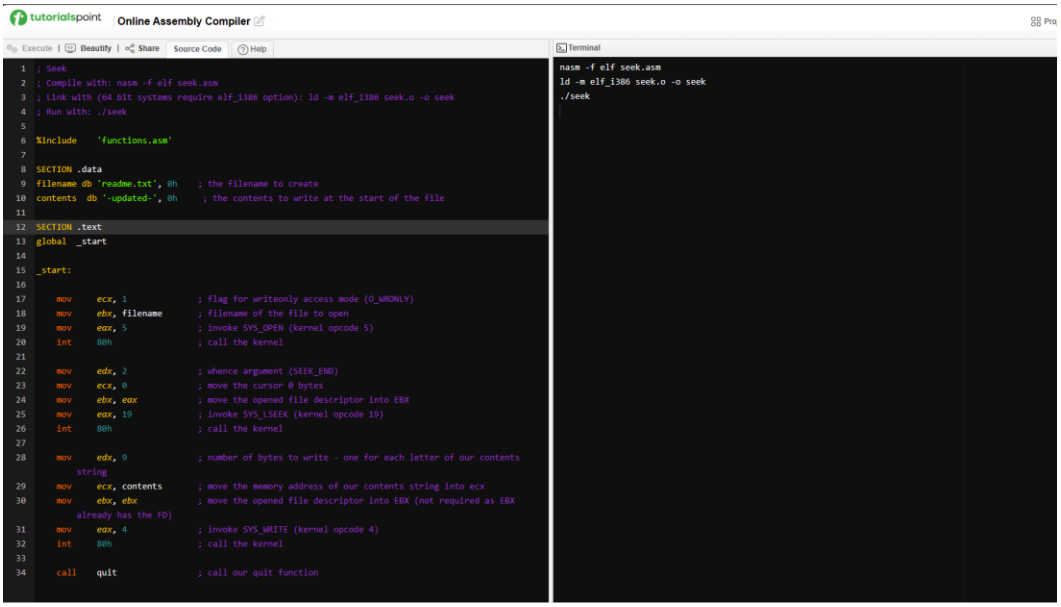
LESSON 26



```
; Close
; Compile with: nasm -f elf close.asm
; Link with (64 bit systems require elf_i386 option): ld -m elf_i386 close.o -o close
; Run with: ./close

%include    'functions.asm'

SECTION .data
filename db 'readme.txt', 0h    ; the filename to create
contents db 'Hello world!', 0h  ; the contents to write

SECTION .bss
fileContents resb 255,          ; variable to store file contents

SECTION .text
global  _start

_start:

    mov     ecx, 0777o          ; Create file from lesson 22
    mov     ebx, filename
    mov     eax, 8
    int     80h

    mov     edx, 12             ; Write contents to file from lesson 23
    mov     ecx, contents
    mov     ebx, eax
    mov     eax, 4
    int     80h

    mov     ecx, 0              ; Open file from lesson 24
    mov     ebx, filename
    mov     eax, 5
    int     80h

    mov     edx, 12             ; Read file from lesson 25
    mov     ecx, fileContents
```

```
nasm -f elf close.asm
bash: asm: command not found
ld -m elf_i386 close.o -o close
ld: cannot find close.o: No such file or directory
./close
bash: ./close: No such file or directory
Hello world!
bash: Hello: command not found
```

LESSON 27



tutorialspoint Online Assembly Compiler

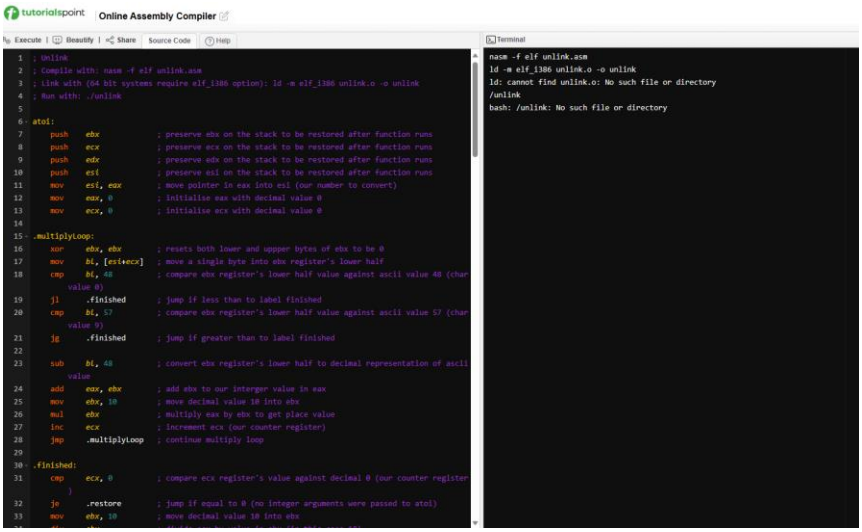
Execute | Beautify | Share | Source Code | Help

```
; Seek
; Compile with: nasm -f elf seek.asm
; Link with (64 bit systems require elf_i386 option): ld -m elf_i386 seek.o -o seek
; Run with: ./seek

%include    'functions.asm'

SECTION .data
filename db 'readme.txt', 0h    ; the filename to create
contents  db '-updated-', 0h    ; the contents to write at the start of the file

SECTION .text
global  _start

_start:

    mov     ecx, 1              ; flag for writeonly access mode (O_WRONLY)
    mov     ebx, filename       ; filename of the file to open
    mov     eax, 5              ; invoke SYS_OPEN (kernel opcode 5)
    int     80h                 ; call the kernel

    mov     edx, 2              ; whence argument (SEEK_END)
    mov     ecx, 0              ; move the cursor 0 bytes
    mov     ebx, eax            ; move the opened file descriptor into EBX
    mov     eax, 19             ; invoke SYS_LSEEK (kernel opcode 19)
    int     80h                 ; call the kernel

    mov     edx, 9              ; number of bytes to write - one for each letter of our contents
        string
    mov     ecx, contents       ; move the memory address of our contents string into ecx
    mov     ebx, ebx            ; move the opened file descriptor into EBX (not required as EBX
        already has the FD)
    mov     eax, 4              ; invoke SYS_WRITE (kernel opcode 4)
    int     80h                 ; call the kernel

    call    quit                ; call our quit function
```

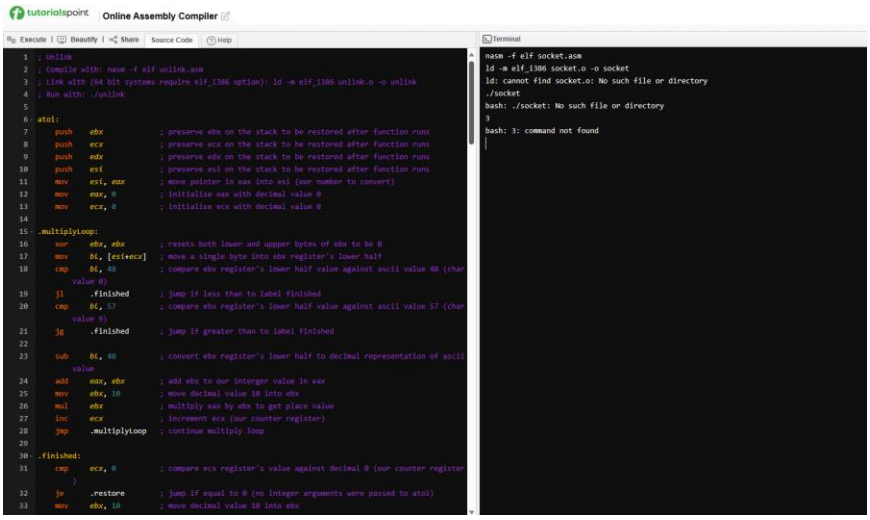
Terminal

```
nasm -f elf seek.asm
ld -m elf_i386 seek.o -o seek
./seek
```

LESSON 28



LESSON 29



LESSON 30