

MP Practical- 3

Name: Siddhesh Dilip Khairnar
PRN No: 22110398 Roll No: 272028

PAGE NO.:
DATE: / /

Experiment no: 3

Aim: Write ALP to print "Hello! world" Program using 16, 32, & 64 bit model and segmentation.

Theory: An ALP can be divided into 3 section.

(i) The data section (ii) Text section (iii) The bss section.

(i) The data section: The data section is used for declaring initialized data or constant. This data does not change at runtime.

(ii) The bss section: The bss section is used for declaring variable. syntax for declaring is section.bss

(iii) The text section: The text section is used for keeping the actual code. This section must begin with the declaration global main, which tell the kernel where the Program execution begin. The syntax for declaring text is section.text.

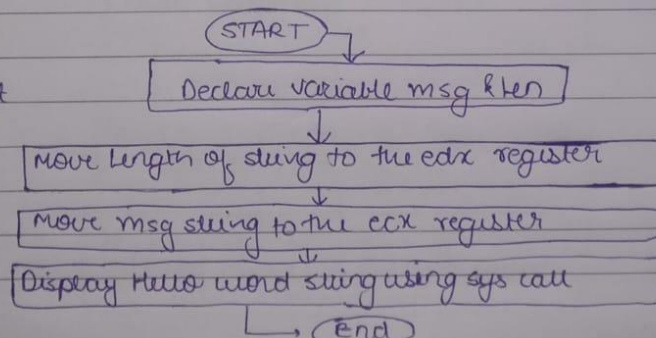
global main

main:

Algorithm:

- (i) Declare variable msg and len in data section
- (ii) Move length of string to the edx registration register
- (iii) Move msg string to the ecx register.
- (iv) By using sys call number 4 & 1 Hello world string is displayed on the terminal.

Flowchart



Name- Siddhesh Dilip Khairnar

Roll No.- 272028

Batch- B2

Code:

section.text

global _start ;must be declared for using gcc

_start: ;tell linker entry point

mov edx, len ;message length

mov ecx, msg ;message to write

mov ebx, 1 ;file descriptor (stdout)

mov eax, 4 ;system call number (sys_write)

int 0x80 ;call kernel

mov edx, len1 ;message length

mov ecx, msg1 ;message to write

mov ebx, 1 ;file descriptor (stdout)

mov eax, 4 ;system call number (sys_write)

int 0x80 ;call kernel

mov edx, len2 ;message length

mov ecx, msg2 ;message to write

mov ebx, 1 ;file descriptor (stdout)

Name- Siddhesh Dilip Khairnar

Roll No.- 272028

Batch- B2

```
mov    eax, 4    ;system call number (sys_write)
```

```
int     0x80     ;call kernel
```

```
mov    eax, 1    ;system call number (sys_exit)
```

```
int     0x80     ;call kernel
```

section.data

```
msg     db      'Ritesh Jawale',0xa    ;our dear string
```

```
len     equ     $ - msg                ;length of our dear string
```

```
msg1    db      'AI&DS',0xa
```

```
len1    equ     $ - msg1
```

```
msg2    db      'VIIT',0xa
```

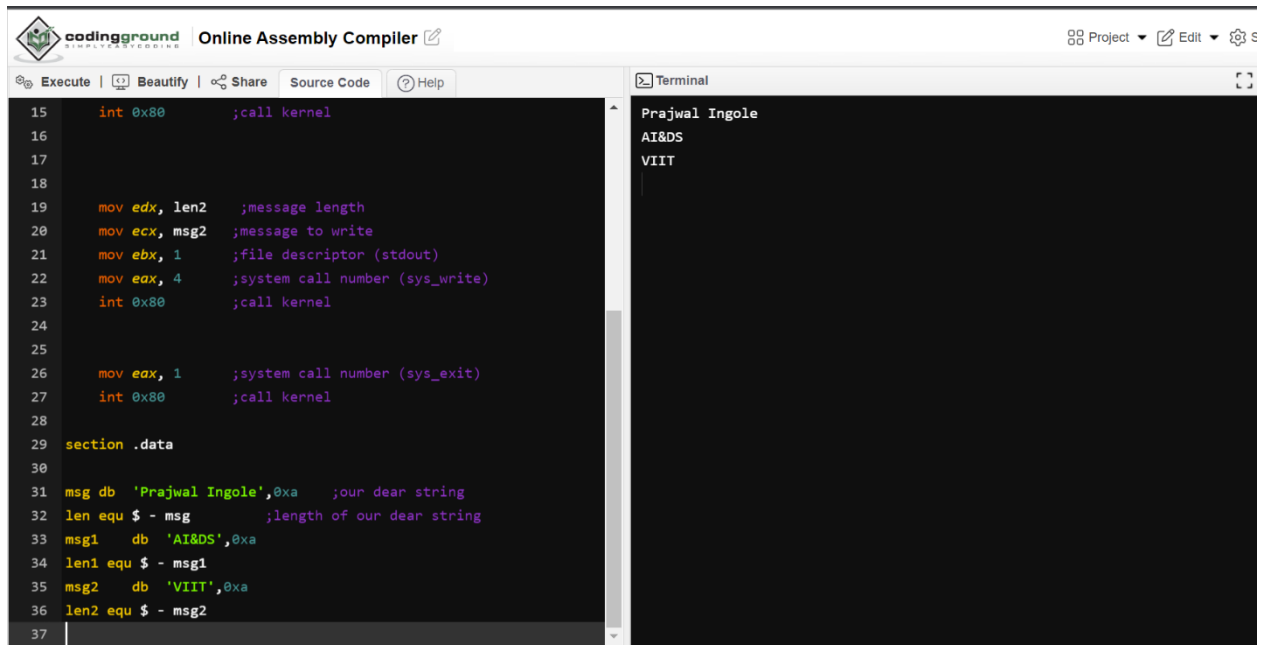
```
len2    equ     $ - msg2
```

Name- Siddhesh Dilip Khairnar

Roll No.- 272028

Batch- B2

Output:



The screenshot displays the 'Online Assembly Compiler' interface. The left pane contains assembly code for a program that prints two strings. The right pane, titled 'Terminal', shows the output of the program.

```
15      int 0x80      ;call kernel
16
17
18
19      mov edx, len2  ;message length
20      mov ecx, msg2  ;message to write
21      mov ebx, 1     ;file descriptor (stdout)
22      mov eax, 4     ;system call number (sys_write)
23      int 0x80      ;call kernel
24
25
26      mov eax, 1     ;system call number (sys_exit)
27      int 0x80      ;call kernel
28
29      section .data
30
31      msg db 'Prajwal Ingole',0xa  ;our dear string
32      len equ $ - msg      ;length of our dear string
33      msg1 db 'AI&DS',0xa
34      len1 equ $ - msg1
35      msg2 db 'VIIT',0xa
36      len2 equ $ - msg2
37
```

The terminal output shows the following text:

```
Prajwal Ingole
AI&DS
VIIT
```