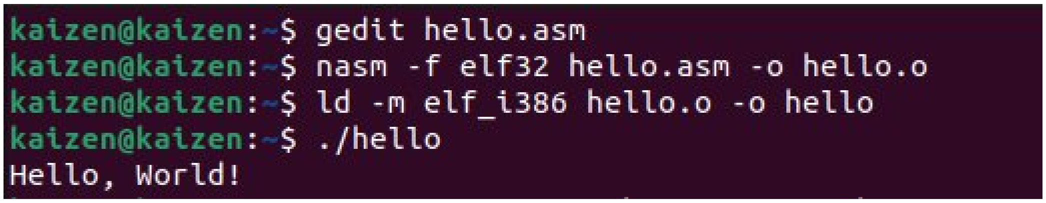
PAIL Assignment 1 - Lab 1

Name : Samarth D Valsange
Roll no: 46
Enrollment No. ADT24SOCB0996
SY-06 Batch (B)

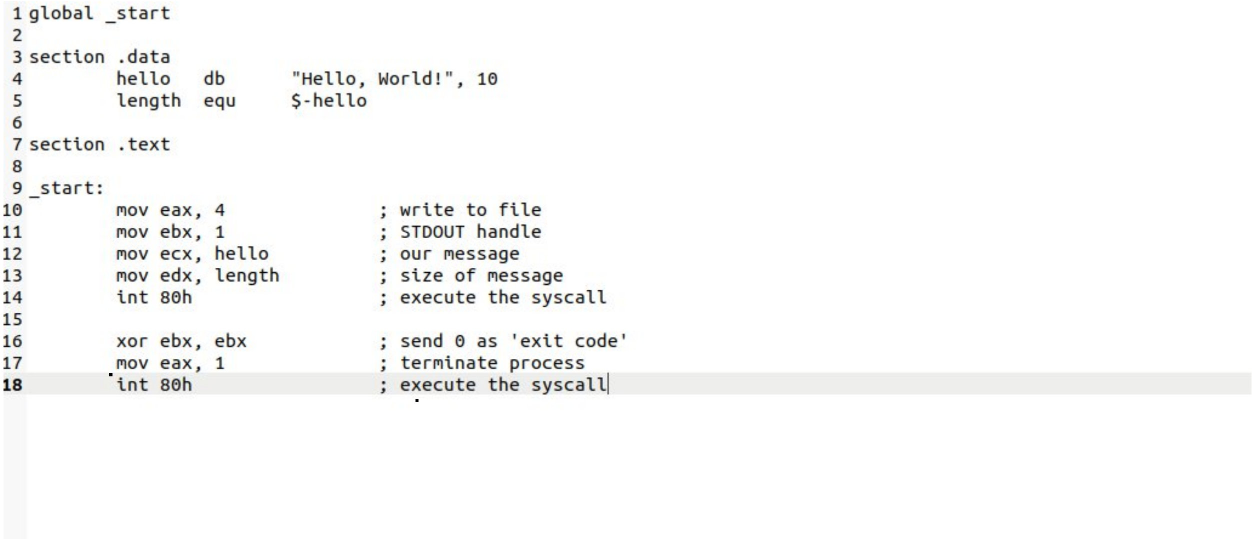
1. Hello World (hello.asm)

This program prints Hello, World! using NASM and Linux system calls.



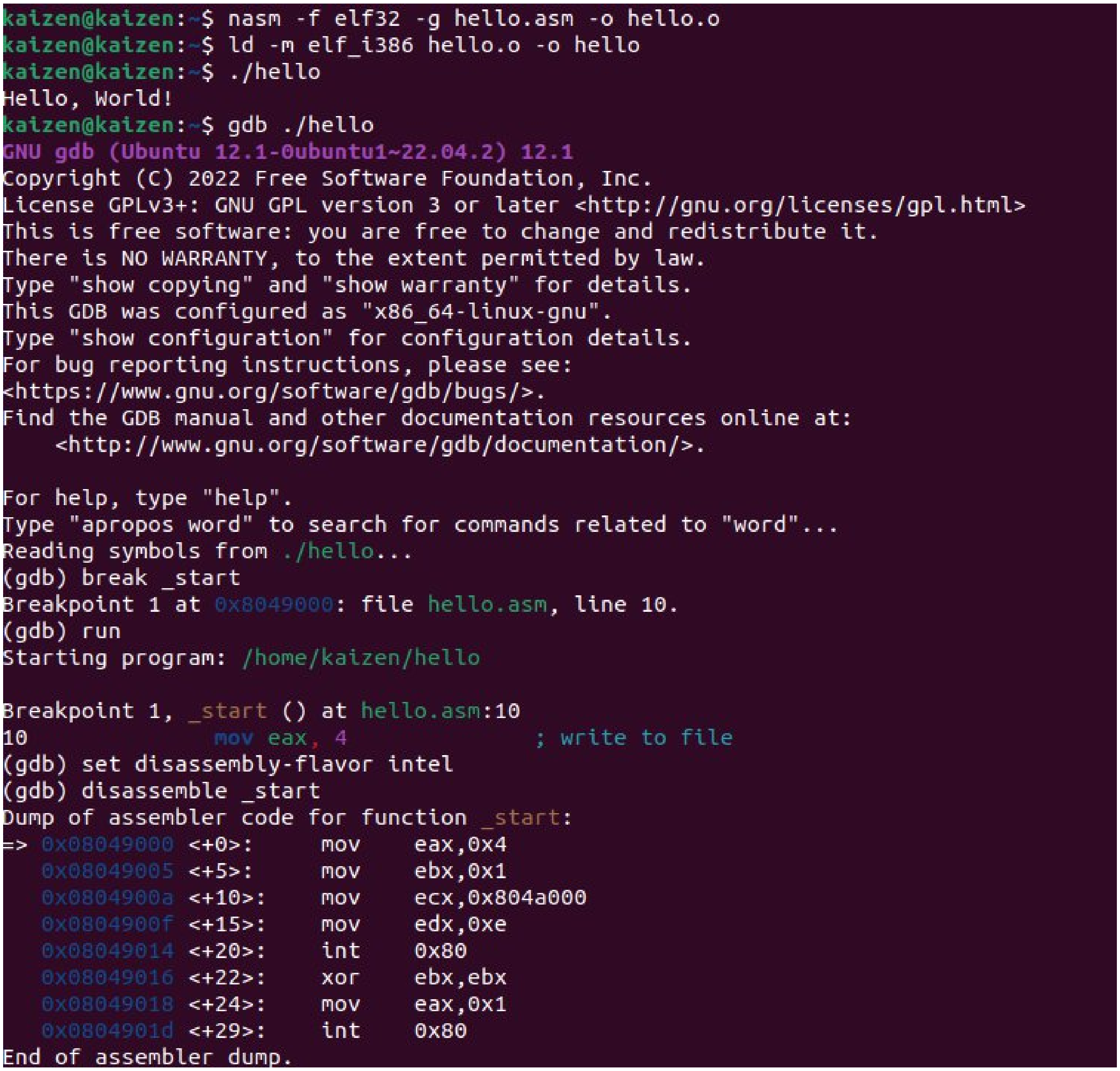
```
kaizen@kaizen:~$ gedit hello.asm
kaizen@kaizen:~$ nasm -f elf32 hello.asm -o hello.o
kaizen@kaizen:~$ ld -m elf_i386 hello.o -o hello
kaizen@kaizen:~$ ./hello
Hello, World!
```

hello.asm file

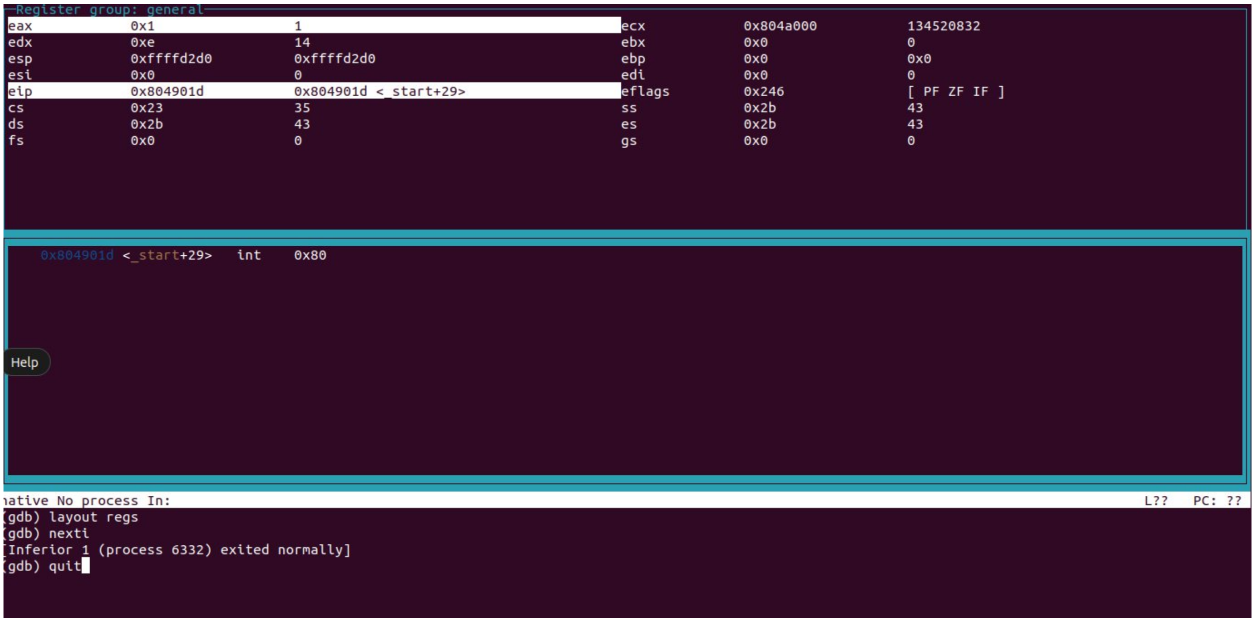


```
global _start

section .data
        hello   db      "Hello, World!", 10
        length  equ     $-hello

section .text

_start:
        mov eax, 4              ; write to file
        mov ebx, 1              ; STDOUT handle
        mov ecx, hello          ; our message
        mov edx, length         ; size of message
        int 80h                 ; execute the syscall

        xor ebx, ebx            ; send 0 as 'exit code'
        mov eax, 1              ; terminate process
        int 80h                 ; execute the syscall
```

2. GDB Use on hello.asm

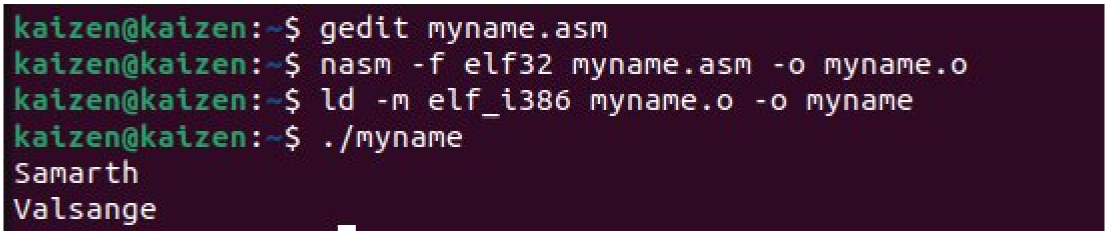


```
kaizen@kaizen:~$ nasm -f elf32 -g hello.asm -o hello.o
kaizen@kaizen:~$ ld -m elf_i386 hello.o -o hello
kaizen@kaizen:~$ ./hello
Hello, World!
kaizen@kaizen:~$ gdb ./hello
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04.2) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./hello...
(gdb) break _start
Breakpoint 1 at 0x8049000: file hello.asm, line 10.
(gdb) run
Starting program: /home/kaizen/hello

Breakpoint 1, _start () at hello.asm:10
10              mov eax, 4              ; write to file
(gdb) set disassembly-flavor intel
(gdb) disassemble _start
Dump of assembler code for function _start:
=> 0x08049000 <+0>:     mov    eax,0x4
   0x08049005 <+5>:     mov    ebx,0x1
   0x0804900a <+10>:    mov    ecx,0x804a000
   0x0804900f <+15>:    mov    edx,0xe
   0x08049014 <+20>:    int    0x80
   0x08049016 <+22>:    xor    ebx,ebx
   0x08049018 <+24>:    mov    eax,0x1
   0x0804901d <+29>:    int    0x80
End of assembler dump.
```



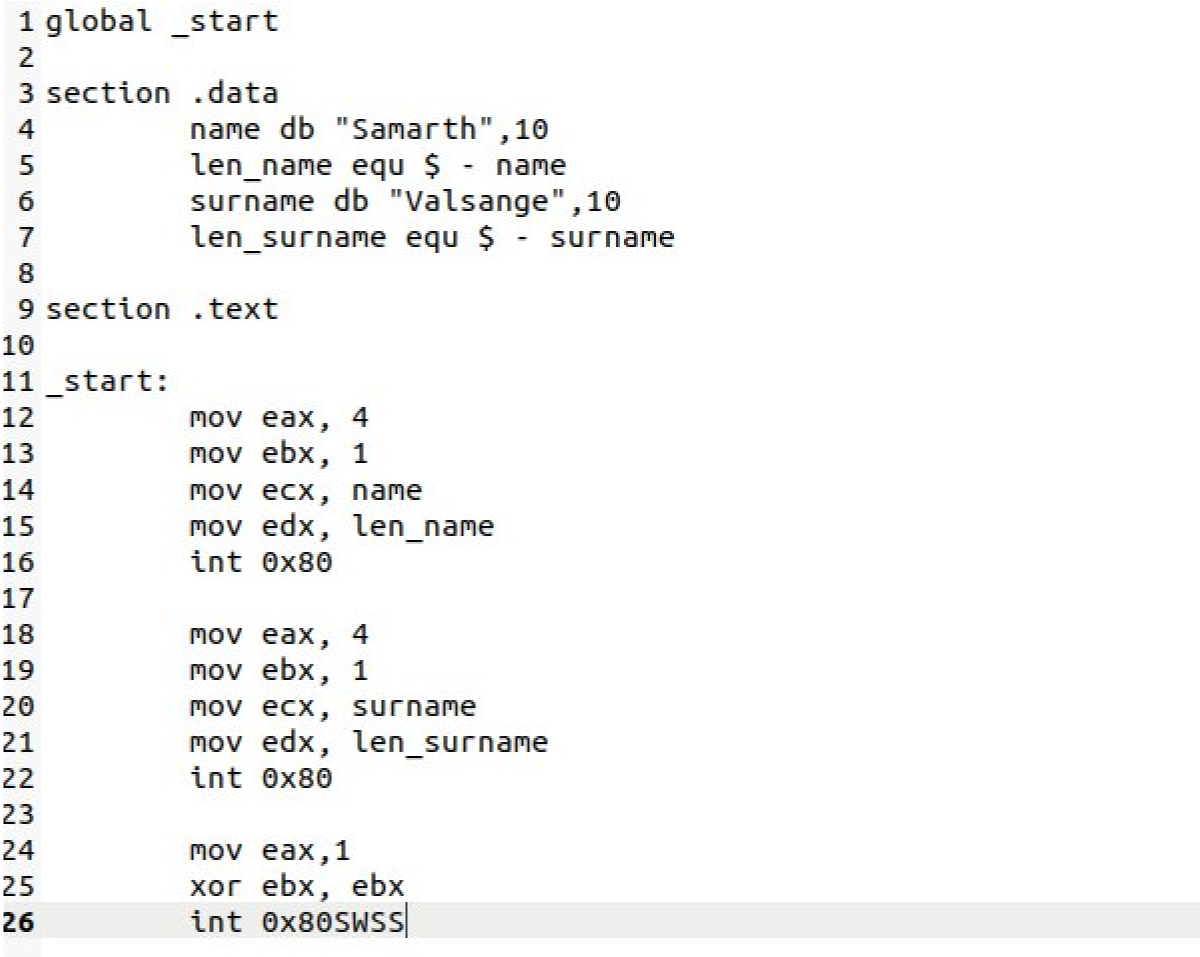
```
Register group: general
eax            0x1                 1                         ecx            0x804a000           134520832
edx            0xe                 14                        ebx            0x0                 0
esp            0xffffd2d0          0xffffd2d0                ebp            0x0                 0x0
esi            0x0                 0                         edi            0x0                 0
eip            0x804901d           0x804901d <_start+29>     eflags         0x246               [ PF ZF IF ]
cs             0x23                35                        ss             0x2b                43
ds             0x2b                43                        es             0x2b                43
fs             0x0                 0                         gs             0x0                 0

    0x804901d <_start+29>   int    0x80

native No process In:                                                                                  L??   PC: ??
(gdb) layout regs
(gdb) nexti
[Inferior 1 (process 6332) exited normally]
(gdb) quit
```

3. Print First Name and Surname (myname.asm)



```
kaizen@kaizen:~$ gedit myname.asm
kaizen@kaizen:~$ nasm -f elf32 myname.asm -o myname.o
kaizen@kaizen:~$ ld -m elf_i386 myname.o -o myname
kaizen@kaizen:~$ ./myname
Samarth
Valsange
```

myname.asm



```
global _start

section .data
        name db "Samarth",10
        len_name equ $ - name
        surname db "Valsange",10
        len_surname equ $ - surname

section .text

_start:
        mov eax, 4
        mov ebx, 1
        mov ecx, name
        mov edx, len_name
        int 0x80

        mov eax, 4
        mov ebx, 1
        mov ecx, surname
        mov edx, len_surname
        int 0x80

        mov eax,1
        xor ebx, ebx
        int 0x80SWSS
```

GITHUB LINK: https://github.com/KaizenPixel/PAI-LAB-ASSIGNMENT.git