

PAIL

ASSIGNMENT 1 – LAB 1

Name: Vaibhav Rawat

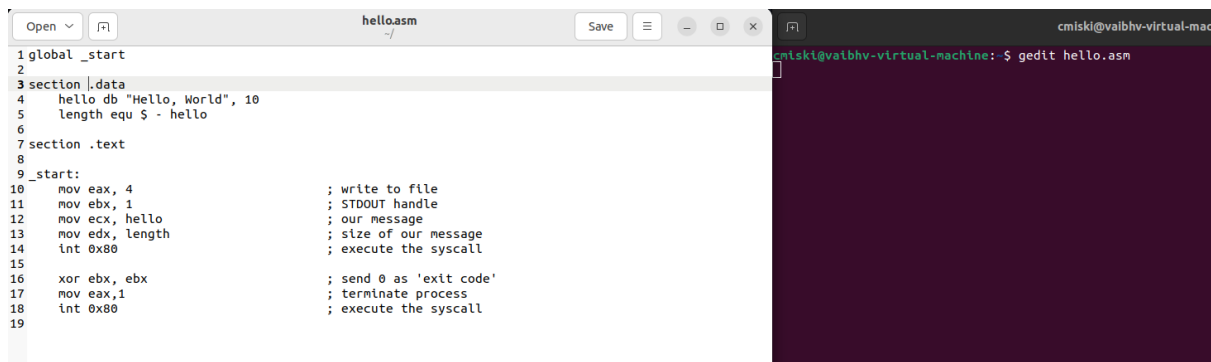
Class: SY-06

RollNo: 59

EnrollmentNo: ADT24S0CB1344

GitHub Repo: https://github.com/cmiski/PAIL_ASSIGNMENT-1

1. Hello World : A simple assembly program that prints “Hello, World!”.



The screenshot shows a code editor window titled 'hello.asm' with the following assembly code:

```
1 global _start
2
3 section .data
4     hello db "Hello, World", 10
5     length equ $ - hello
6
7 section .text
8
9 _start:
10    mov eax, 4           ; write to file
11    mov ebx, 1           ; STDOUT handle
12    mov ecx, hello       ; our message
13    mov edx, length      ; size of our message
14    int 0x80             ; execute the syscall
15
16    xor ebx, ebx          ; send 0 as 'exit code'
17    mov eax, 1           ; terminate process
18    int 0x80             ; execute the syscall
19
```



The screenshot shows a terminal window with the following commands and output:

```
cmiski@vaibhv-virtual-machine: ~
cmiski@vaibhv-virtual-machine:~$ gedit hello.asm
cmiski@vaibhv-virtual-machine:~$ nasm -f elf32 hello.asm
cmiski@vaibhv-virtual-machine:~$ ld -m elf_i386 hello.o -o hello
cmiski@vaibhv-virtual-machine:~$ ./hello
Hello, World
cmiski@vaibhv-virtual-machine:~$
```

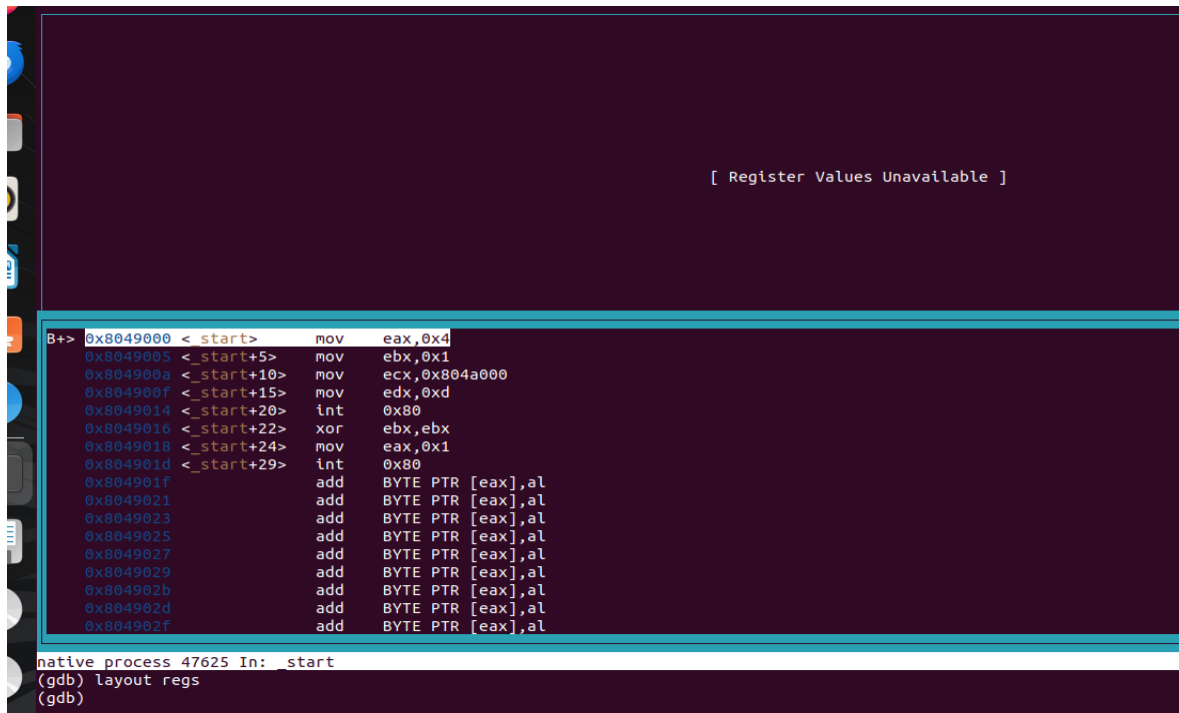
```

cmiski@vaibhv-virtual-machine:~$ gedit hello.asm
cmiski@vaibhv-virtual-machine:~$ nasm -f elf32 -g hello.asm -o hello.o
cmiski@vaibhv-virtual-machine:~$ ld -m elf_i386 hello.o -o hello
cmiski@vaibhv-virtual-machine:~$ ./hello
Hello, World
cmiski@vaibhv-virtual-machine:~$ 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)
(gdb) break _start
Breakpoint 1 at 0x8049000: file hello.asm, line 10.
(gdb) run
Starting program: /home/cmiski/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,0xd
0x08049014 <+20>:     int     0x80
0x08049016 <+22>:     xor     ebx,ebx
0x08049018 <+24>:     mov     eax,0x1
0x0804901d <+29>:     int     0x80
End of assembler dump.
(gdb) layout asm

```



```

Register group: general
eax      0x1      1      ecx      0x804a000      134520832      edx      0xd      13
ebx      0x0      0      esp      0xffffd2b0      0xffffd2b0      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

0+ 0x8049000 <_start> mov     eax,0x4
0x8049005 <_start+5> mov     ebx,0x1
0x804900a <_start+10> mov     ecx,0x804a000
0x804900f <_start+15> mov     edx,0xd
0x8049014 <_start+20> int     0x80
0x8049016 <_start+22> xor     ebx,ebx
0x8049018 <_start+24> mov     eax,0x1
> 0x804901d <_start+29> int     0x0
0x804901f add     BYTE PTR [eax],al
0x8049021 add     BYTE PTR [eax],al
0x8049023 add     BYTE PTR [eax],al
0x8049025 add     BYTE PTR [eax],al
0x8049027 add     BYTE PTR [eax],al
0x8049029 add     BYTE PTR [eax],al
0x804902b add     BYTE PTR [eax],al
0x804902d add     BYTE PTR [eax],al
0x804902f add     BYTE PTR [eax],al

native process 47625 In: start
(gdb) layout regs
(gdb) nexti
(gdb)

```

```

Register group: general
eax      0x1      1      ecx      0x804a000      134520832      edx      0xd      13
ebx      0x0      0      esp      0xffffd2b0      0xffffd2b0      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     0x0

native No process In:
(gdb) layout regs
(gdb) nexti
[Inferior 1 (process 47625) exited normally]
(gdb) quit

```

2: Print Name & Surname - Prints my name on the first line and my surname on the second line.

```
Open ▾  myname.asm
~/assembly_lab1

1 global _start
2
3 section .data
4     name db "Vaibhav",10
5     len_name equ $ - name
6     surname db "Rawat",10
7     len_surname equ $ - surname
8
9 section .text
10
11 _start:
12     mov eax, 4
13     mov ebx, 1
14     mov ecx, name
15     mov edx, len_name
16     int 0x80
17
18     mov eax, 4
19     mov ebx, 1
20     mov ecx, surname
21     mov edx, len_surname
22     int 0x80
23
24     mov eax,1
25     xor ebx, ebx
26     int 0x80
```

```
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ gedit myname.asm
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ nasm -f elf32 myname.asm
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ ld -m elf_i386 myname.o -o myname
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ ./myname
Vaibhav
Rawat
cmiski@vaibhv-virtual-machine:~/assembly_lab1$
```

```
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ nasm -f elf32 -g myname.asm -o myname.o
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ ld -m elf_i386 myname.o -o myname
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ ./myname
```

Vaibhav

Rawat

```
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ gdb ./myname
```

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 ./myname...

(gdb)

(gdb) break _start

Breakpoint 1 at 0x08049000: file myname.asm, line 12.

(gdb) run

Starting program: /home/cmiski/assembly_lab1/myname

Breakpoint 1, _start () at myname.asm:12

12 mov eax, 4

(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,0x8
    0x08049014 <+20>: int    0x80
    0x08049016 <+22>: mov    eax,0x4
    0x0804901b <+27>: mov    ebx,0x1
    0x08049020 <+32>: mov    ecx,0x804a008
    0x08049025 <+37>: mov    edx,0x6
    0x0804902a <+42>: int    0x80
    0x0804902c <+44>: mov    eax,0x1
    0x08049031 <+49>: xor    ebx,ebx
    0x08049033 <+51>: int    0x80
```

End of assembler dump.

(gdb) layout asm

```

B+> 0x8049000 < start>    mov     eax,0x4
0x8049005 <_start+5>      mov     ebx,0x1
0x804900a <_start+10>     mov     ecx,0x804a000
0x804900f <_start+15>     mov     edx,0x8
0x8049014 <_start+20>     int      0x80
0x8049016 <_start+22>     mov     eax,0x4
0x804901b <_start+27>     mov     ebx,0x1
0x8049020 <_start+32>     mov     ecx,0x804a008
0x8049025 <_start+37>     mov     edx,0x6
0x804902a <_start+42>     int      0x80
0x804902c <_start+44>     mov     eax,0x1
0x8049031 <_start+49>     xor      ebx,ebx
0x8049033 <_start+51>     int      0x80
0x8049035                add     BYTE PTR [eax],al
0x8049037                add     BYTE PTR [eax],al
0x8049039                add     BYTE PTR [eax],al
0x804903b                add     BYTE PTR [eax],al
0x804903d                add     BYTE PTR [eax],al
0x804903f                add     BYTE PTR [eax],al
0x8049041                add     BYTE PTR [eax],al
0x8049043                add     BYTE PTR [eax],al
0x8049045                add     BYTE PTR [eax],al
0x8049047                add     BYTE PTR [eax],al
0x8049049                add     BYTE PTR [eax],al
0x804904b                add     BYTE PTR [eax],al
0x804904d                add     BYTE PTR [eax],al
0x804904f                add     BYTE PTR [eax],al
0x8049051                add     BYTE PTR [eax],al
0x8049053                add     BYTE PTR [eax],al
0x8049055                add     BYTE PTR [eax],al
0x8049057                add     BYTE PTR [eax],al
0x8049059                add     BYTE PTR [eax],al
0x804905b                add     BYTE PTR [eax],al
0x804905d                add     BYTE PTR [eax],al

```

```

native process 48713 In: _start
(gdb)

```

[Register Values Unavailable]

```

B+> 0x8049000 < start>    mov     eax,0x4
0x8049005 <_start+5>      mov     ebx,0x1
0x804900a <_start+10>     mov     ecx,0x804a000
0x804900f <_start+15>     mov     edx,0x8
0x8049014 <_start+20>     int      0x80
0x8049016 <_start+22>     mov     eax,0x4
0x804901b <_start+27>     mov     ebx,0x1
0x8049020 <_start+32>     mov     ecx,0x804a008
0x8049025 <_start+37>     mov     edx,0x6
0x804902a <_start+42>     int      0x80
0x804902c <_start+44>     mov     eax,0x1
0x8049031 <_start+49>     xor      ebx,ebx
0x8049033 <_start+51>     int      0x80
0x8049035                add     BYTE PTR [eax],al
0x8049037                add     BYTE PTR [eax],al
0x8049039                add     BYTE PTR [eax],al
0x804903b                add     BYTE PTR [eax],al

```

```

native process 48713 In: _start
(gdb) layout regs
(gdb)

```

```

--Register group: general
eax    0x1      1      ecx    0x804a008    134520840    edx    0x6      6
ebx    0x0      0      esp    0xffffd270    0xffffd270    ebp    0x0      0x0
esi    0x0      0      edi    0x0      0      eip    0x8049033    0x8049033 < start+51>
eflags 0x246    [ PF ZF IF ]    cs    0x23      35    ss    0x2b      43
ds      0x2b      43    es    0x2b      43    fs    0x0      0
gs      0x0      0

0x8049033 < start+51> int 0x80

Native No process in:
(gdb) layout regs
(gdb) nextl
[Inferior 1 (process 48713) exited normally]
(gdb) quit

```

3: Pushing both of these codes to GitHub via terminal.

```

cmiski@vaibhv-virtual-machine:~$ cd ~/assembly_lab1/
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/cmiski/assembly_lab1/.git/
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ git add hello.asm myname.asm
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ git commit -m "Lab 1 codes"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'cmiski@vaibhv-virtual-machine.(none)')
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ git config --global user.email "oshorwtt1@gmail.com"
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ git config --global user.name "cmiski"
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ git commit -m "Lab 1 codes"
[master (root-commit) 8cb7bae] Lab 1 codes
 2 files changed, 45 insertions(+)
 create mode 100644 hello.asm
 create mode 100644 myname.asm
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ git branch -M main
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ git remote add origin https://github.com/cmiski/PAIL_ASSIGNMENT-1.git
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ git push -u origin main

```

```
cmiski@vaibhv-virtual-machine:~/assembly_lab1$ git push -u origin main
```

```
Enumerating objects: 4, done.
```

```
Counting objects: 100% (4/4), done.
```

```
Delta compression using up to 2 threads
```

```
Compressing objects: 100% (4/4), done.
```

```
Writing objects: 100% (4/4), 635 bytes | 635.00 KiB/s, done.
```

```
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
```

```
To github.com:cmiski/PAIL_ASSIGNMENT-1.git
```

```
* [new branch]      main -> main
```

```
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

```
cmiski@vaibhv-virtual-machine:~/assembly_lab1$
```