PAIL

ASSIGNMENTS

Name: Vaibhav Rawat

Class: SY-06

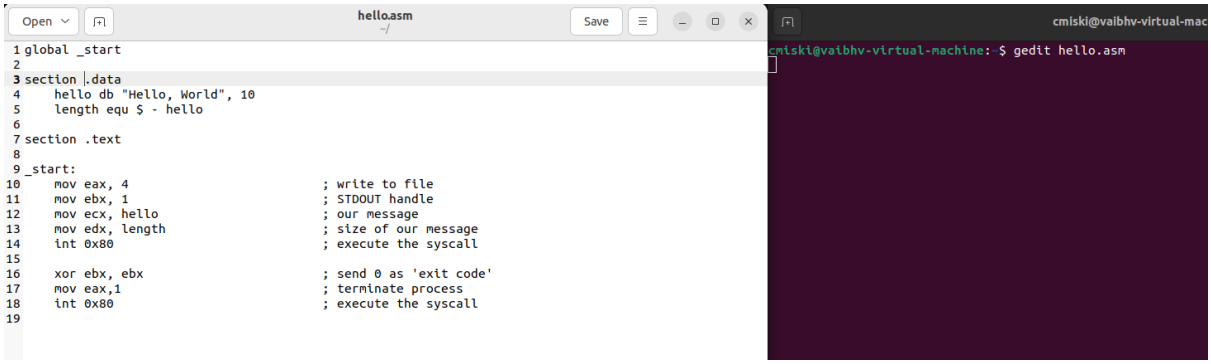
RollNo: 59

EnrollmentNo: ADT24SOCB1344

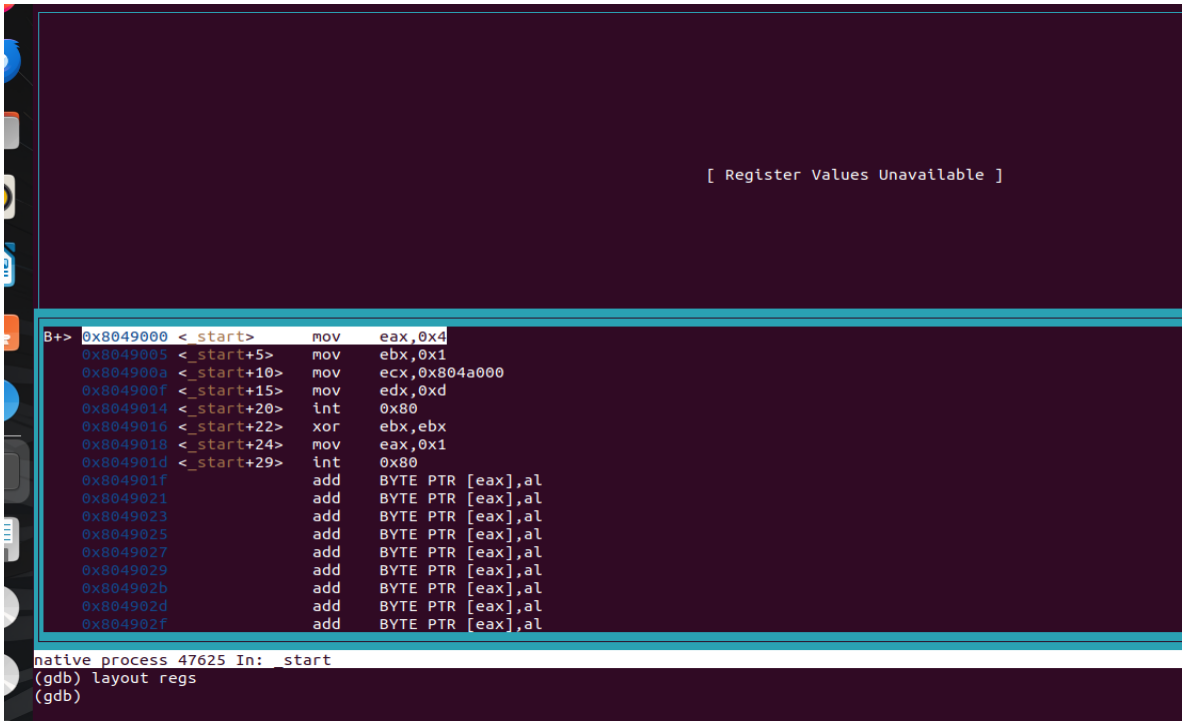
GitHub Repo: https://github.com/cmiski/PAIL.git

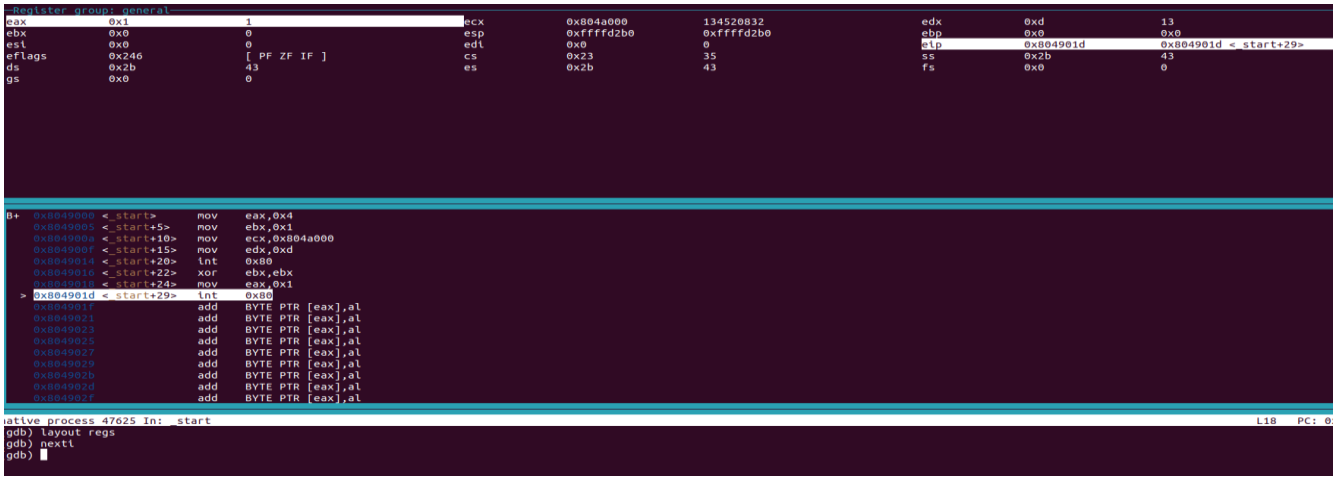
Lab-1

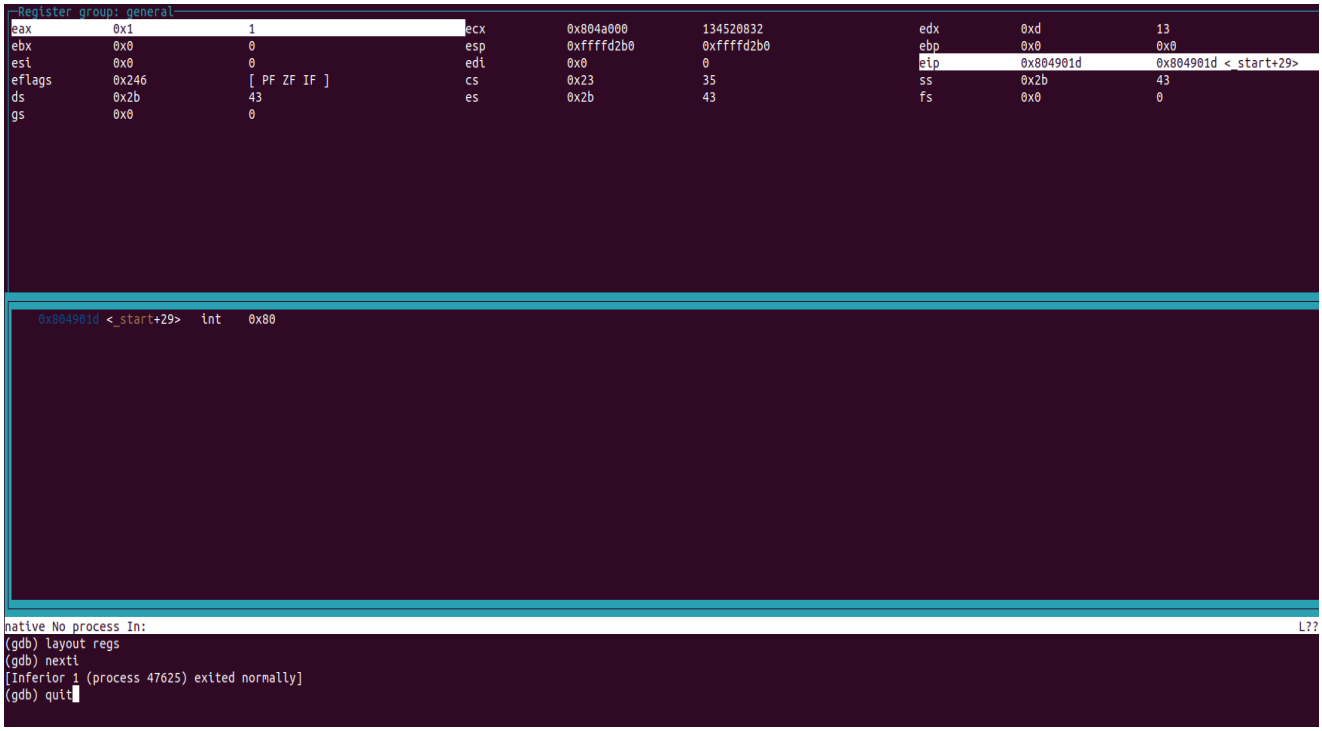
Hello World : A simple assembly program that prints “Hello, World!”.



```
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
ebx 0x0 0
esi 0x0 0
eflags 0x246 [PF ZF IF]
ds 0x2b 43
gs 0x0 0
ecx 0x804a000 134520832
esp 0xffffd2b0 0xffffd2b0
edi 0x0 0
cs 0x23 35
es 0x2b 43
edx 0xd 13
ebp 0x0 0x0
eip 0x804901d 0x804901d <_start+29>
ss 0x2b 43
fs 0x0 0
B+ 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 0x80
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 L18 PC: 0
gdb) layout regs
gdb) nexti
gdb)


Register group: general
eax 0x1 1
ebx 0x0 0
esi 0x0 0
eflags 0x246 [PF ZF IF]
ds 0x2b 43
gs 0x0 0
ecx 0x804a000 134520832
esp 0xffffd2b0 0xffffd2b0
edi 0x0 0
cs 0x23 35
es 0x2b 43
edx 0xd 13
ebp 0x0 0x0
eip 0x804901d 0x804901d <_start+29>
ss 0x2b 43
fs 0x0 0
0x804901d <_start+29> int 0x80
native No process In: L??
(gdb) layout regs
(gdb) nexti
[Inferior 1 (process 47625) exited normally]
(gdb) quit

Lab-2

Print Name & Surname – Prints my name on the first line and my surname on the second line.

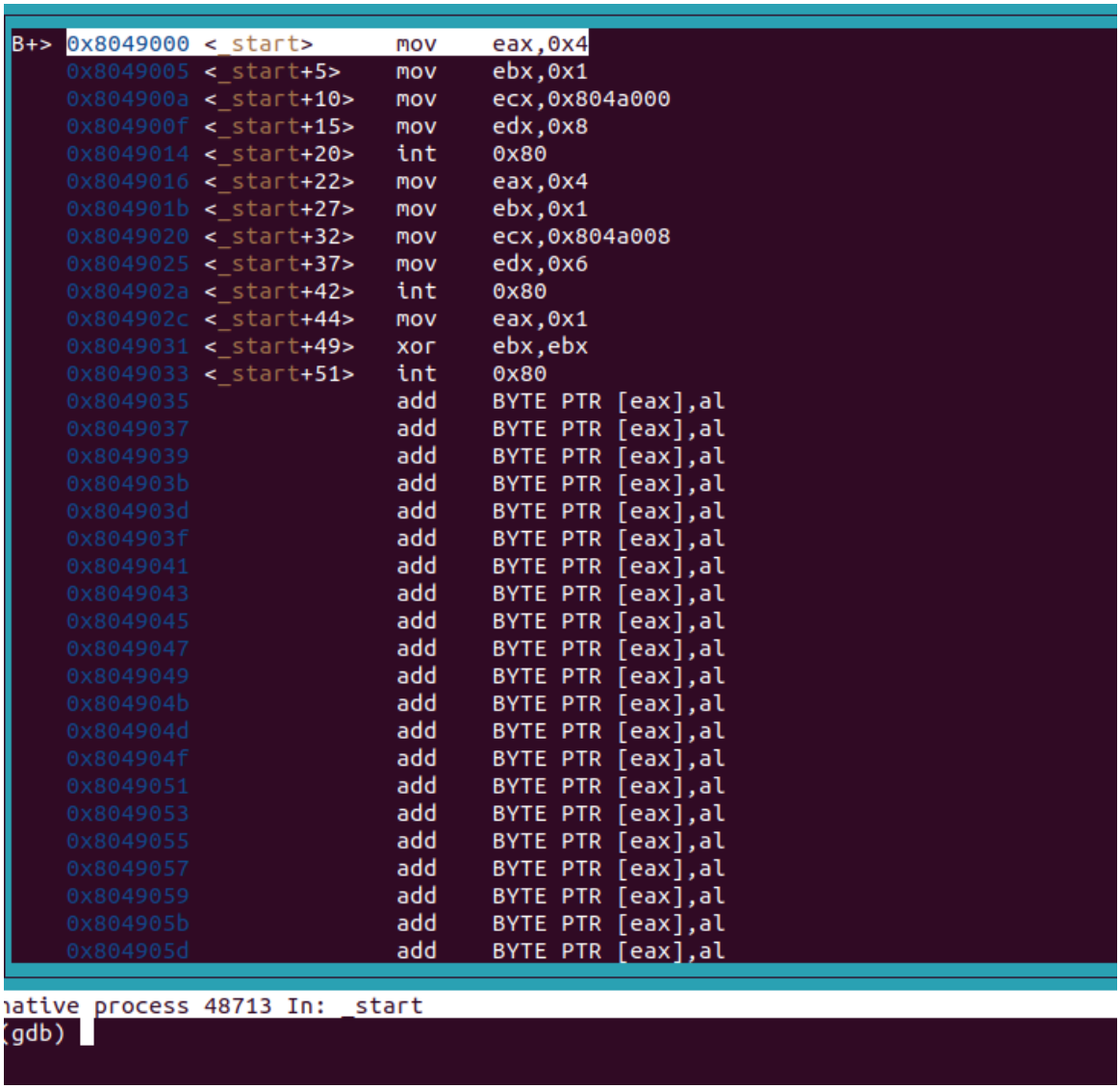
Open

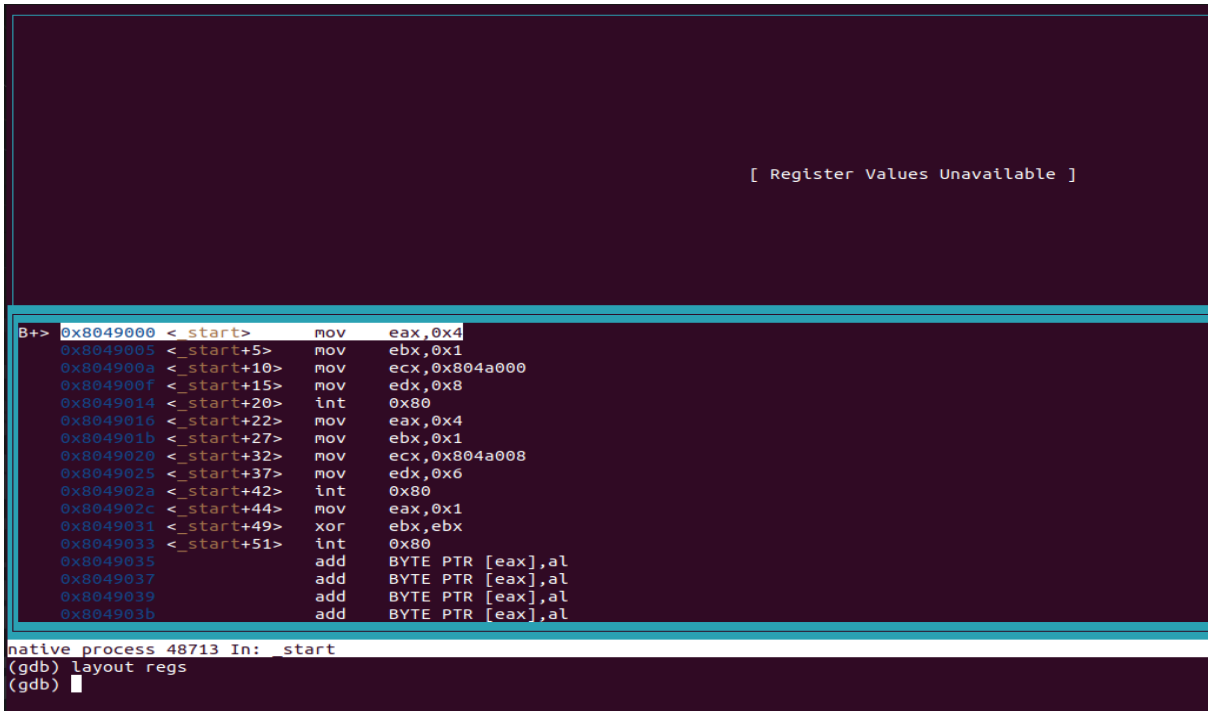
myname.asm
~/assembly_lab1

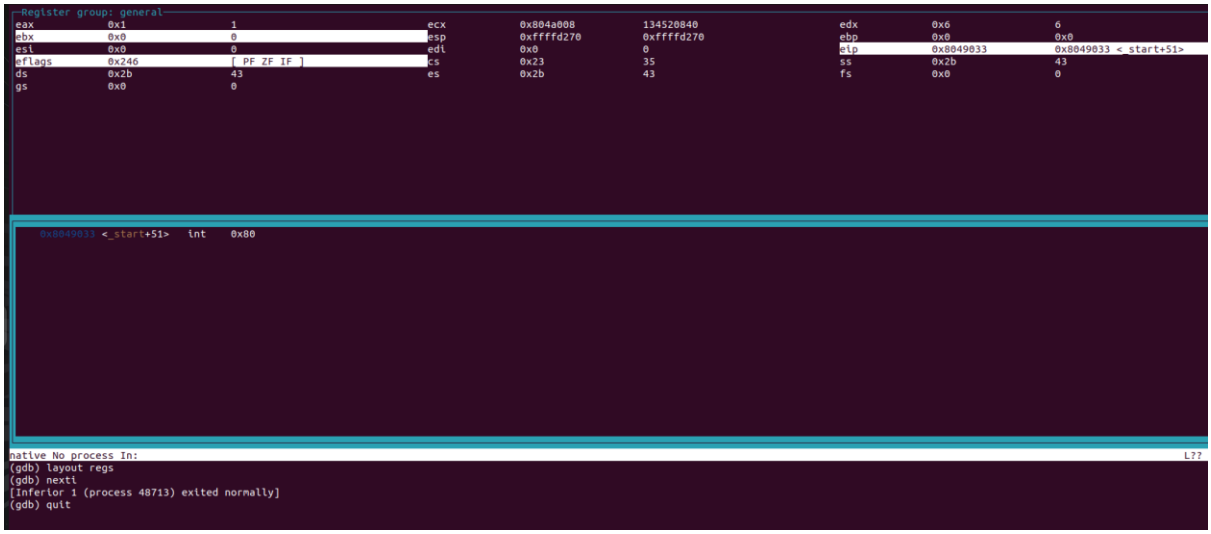
```
1  global _start
2
3  section .data
4          name db "Vaibhav",10
5          len_name equ $ - name
6          surname db "Rawat",10
7          len_surname equ $ - surname
8
9  section .text
10
11 _start:
12         mov eax, 4
13         mov ebx, 1
14         mov ecx, name
15         mov edx, len_name
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 0x8049000: 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)



Lab-3

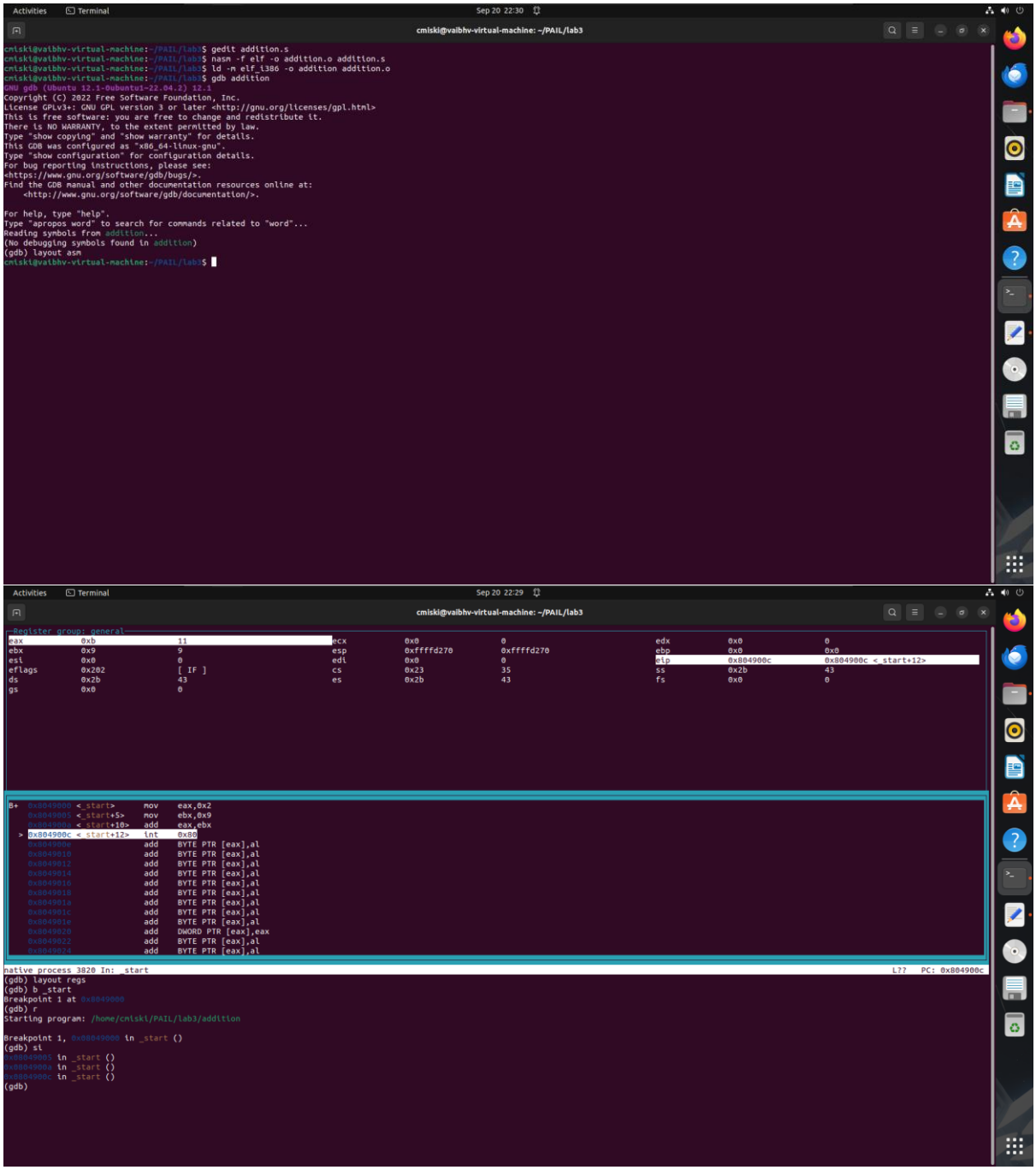
Addition

1) In registers:

Code:

```
section .text

global _start

_start:
	MOV eax,2
	MOV ebx,9
	ADD eax,ebx
	int 80h
```

Output:



2) Using Variables

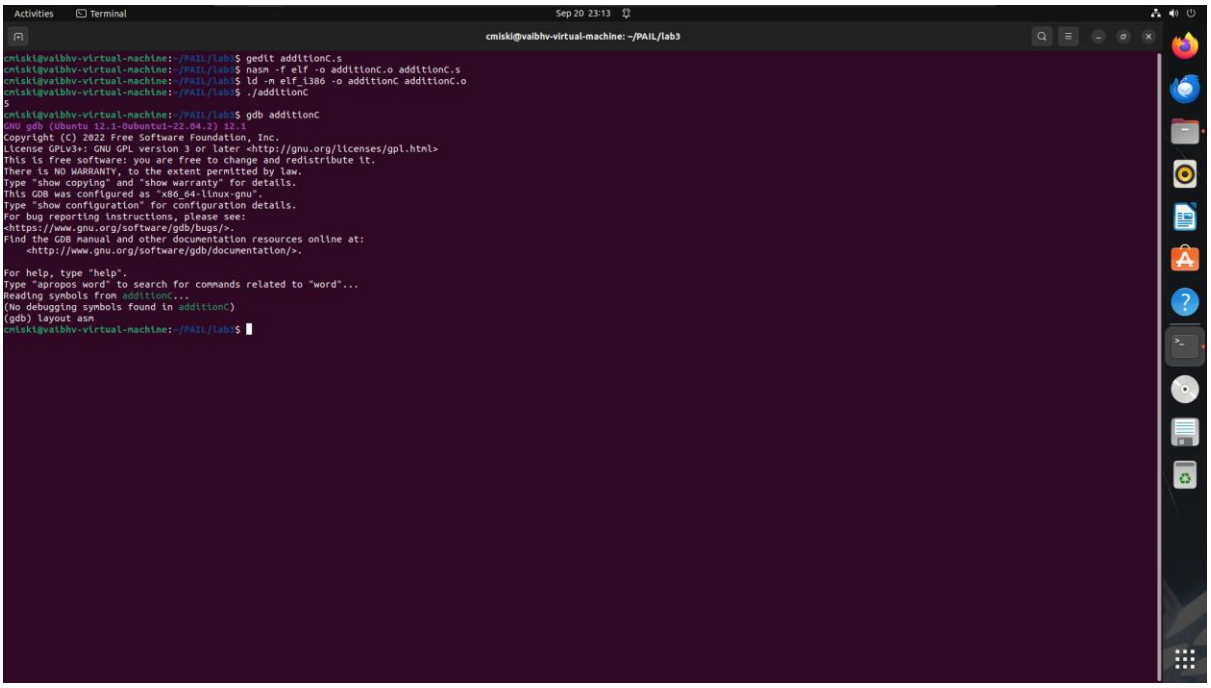
Code:

```
section .data
        num1 db 2
        num2 db 3
        result db 0
```

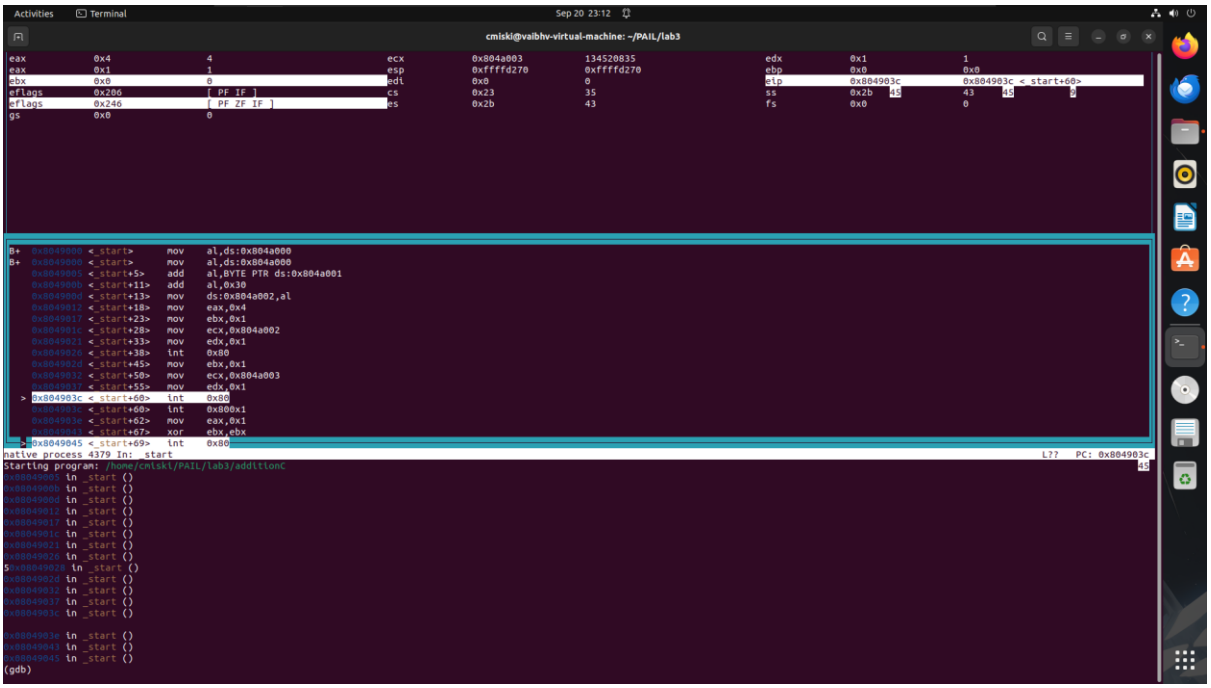
```
        newline db 10

section .text
global _start

_start:
        MOV al,[num1]
        ADD al,[num2]

        ADD al,'0'
        MOV [result],al

        MOV eax,5
        MOV ebx,1
        MOV ecx,result
        MOV edx,1
        INT 80h

        MOV eax,4
        MOV ebx,1
        MOV ecx,newline
        MOV edx,1
        INT 80h

        MOV eax,1
        XOR ebx,ebx
        INT 80h
```

Output:

1) Terminal:



2) GDB:



3) Taking User Input

Code:

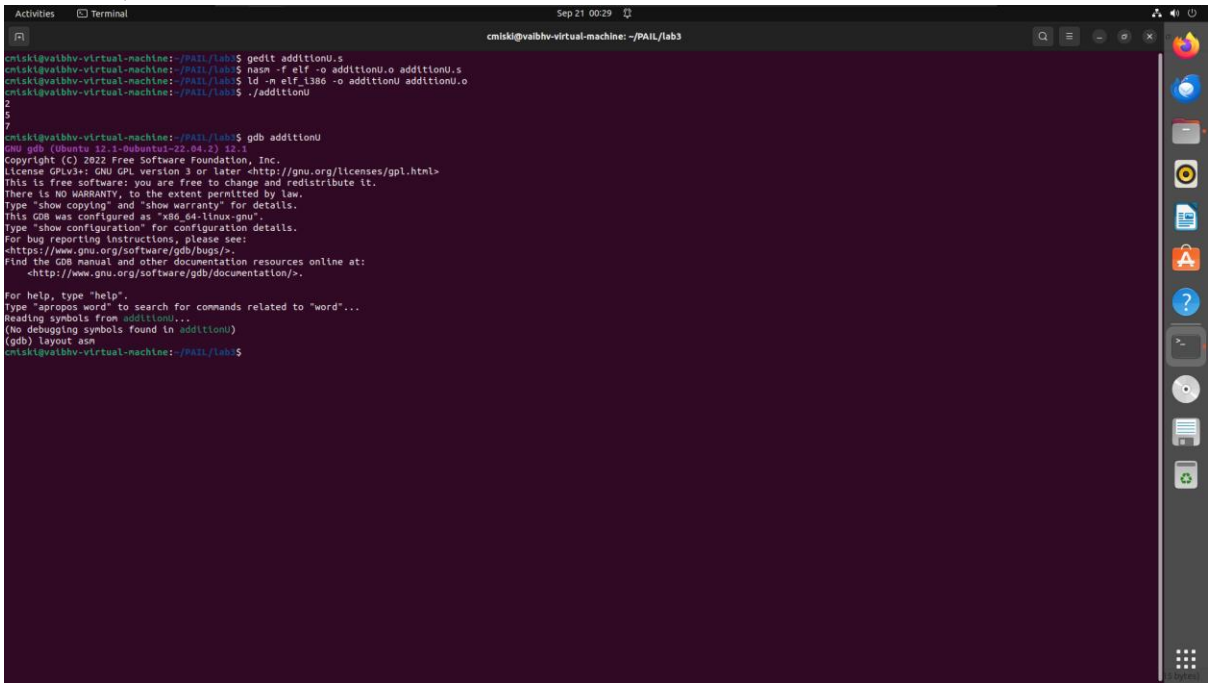
```
section .data
```

```
	result db 0
	newline db 10

section .bss
	input1 RESB 4
	input2 RESB 4

section .text

global _start

_start:
	MOV eax,3
	MOV ebx,0
	MOV ecx,input1
	MOV edx,4
	INT 80h

	MOV eax,3
	MOV ebx,0
	MOV ecx,input2
	MOV edx,4
	INT 80h

	MOV al,[input1]
	SUB al,'0'
	MOV bl,al

	MOV al,[input2]
	SUB al,'0'
	ADD al,bl
```

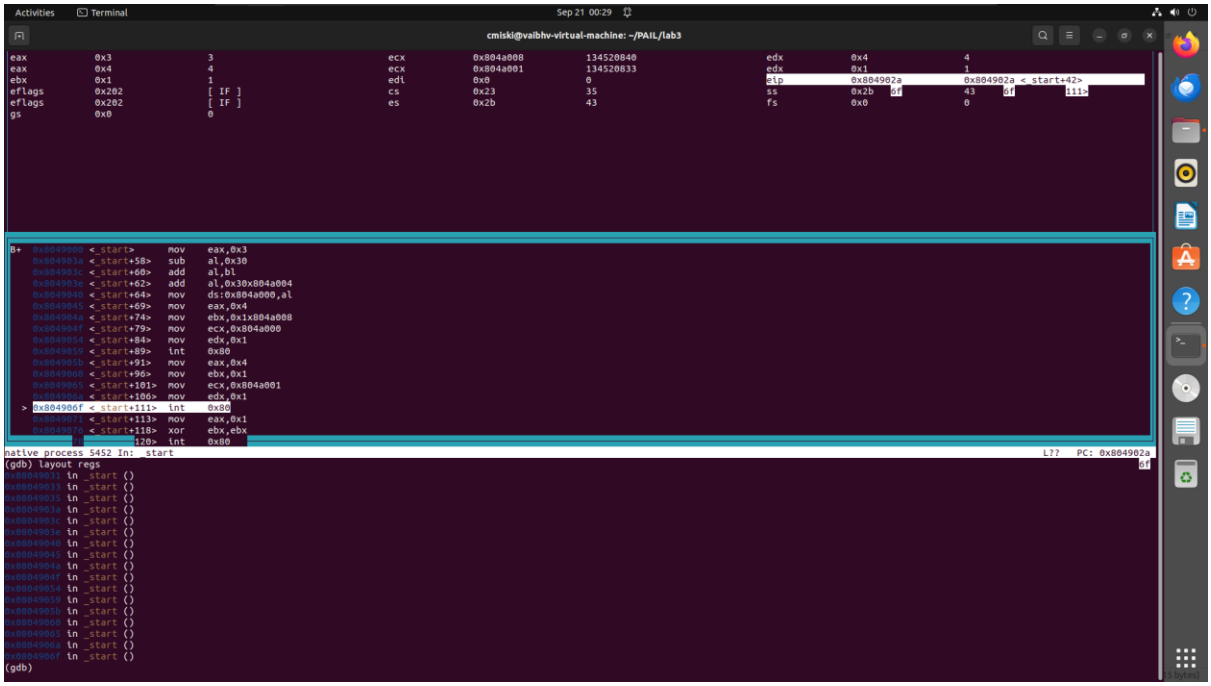
```
ADD al,'0'
MOV [result],al

MOV eax,4
MOV ebx,1
MOV ecx,result
MOV edx,1
INT 80h

MOV eax,4
MOV ebx,1
MOV ecx,newline
MOV edx,1
INT 80h

MOV eax,1
XOR ebx,ebx
INT 80h
```

Output:

1)Terminal:



2) GDB:



Subtraction

A] on registers

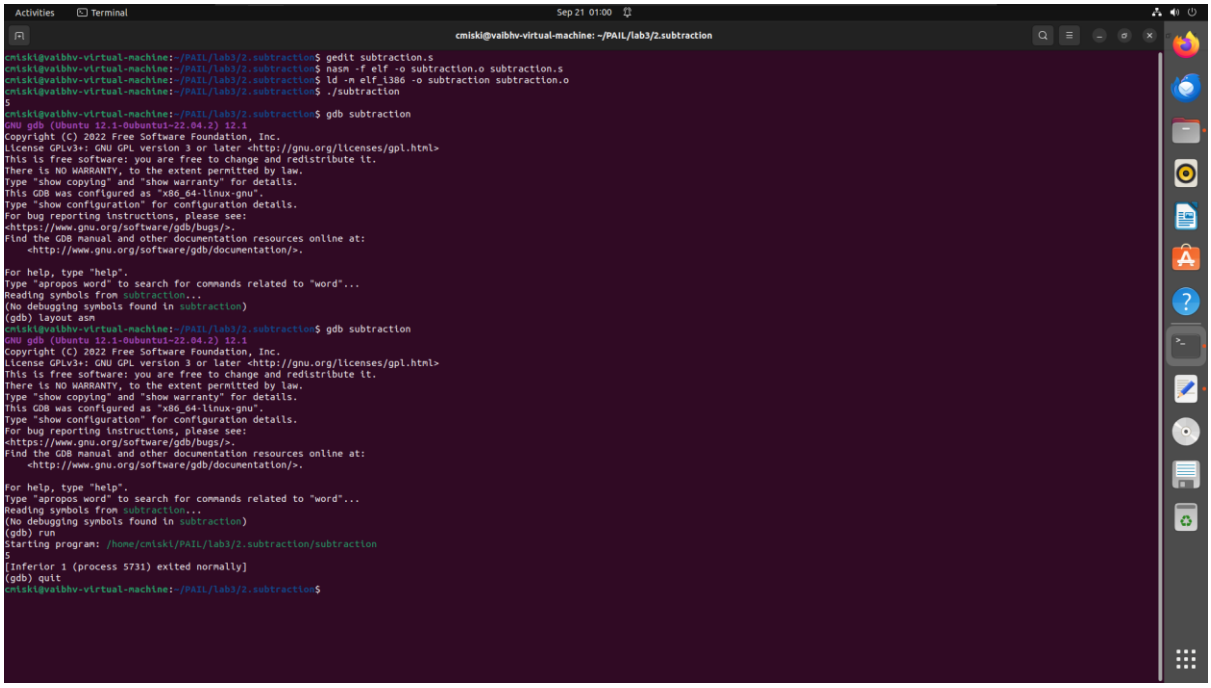
CODE:

```
section .data
	result db 0
	newline db 10

section .text

global _start

_start:
	MOV al,7
	MOV bl,2
	SUB al,bl
	ADD al,'0'
	MOV [result],al
```

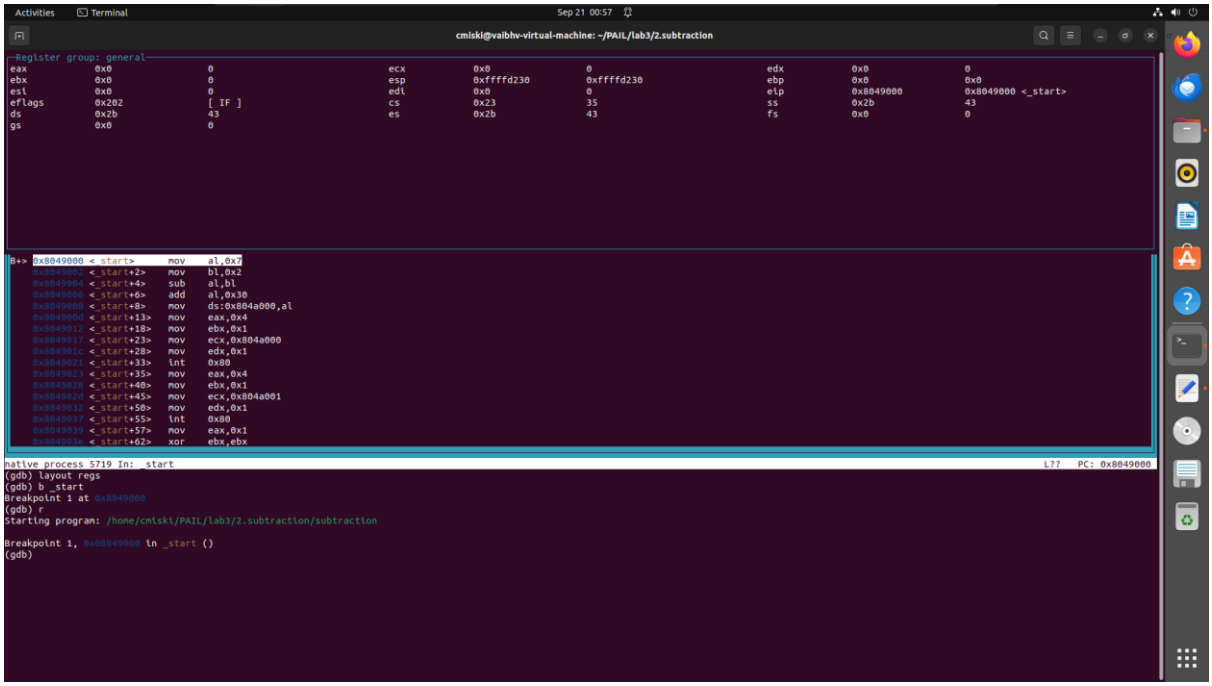
```
MOV eax,4
MOV ebx,1
MOV ecx,result
MOV edx,1
INT 80h

MOV eax,4
MOV ebx,1
MOV ecx,newline
MOV edx,1
INT 80h

MOV eax,1
XOR ebx,ebx
INT 80h
```

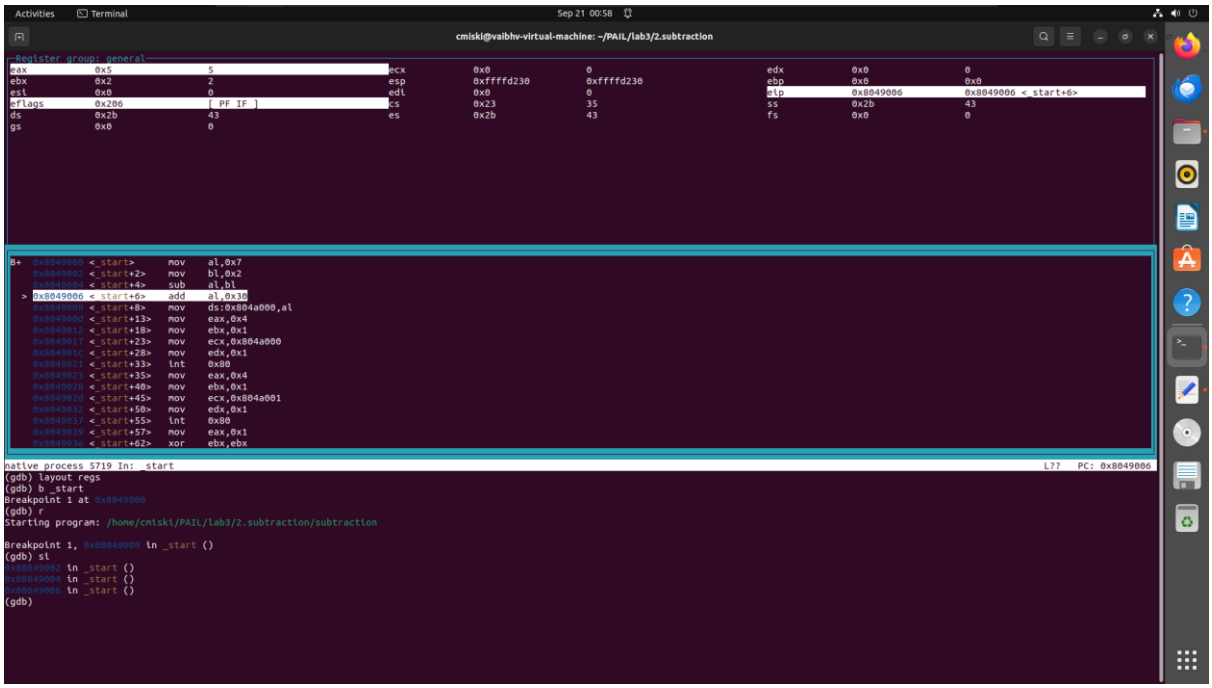
OUTPUT:

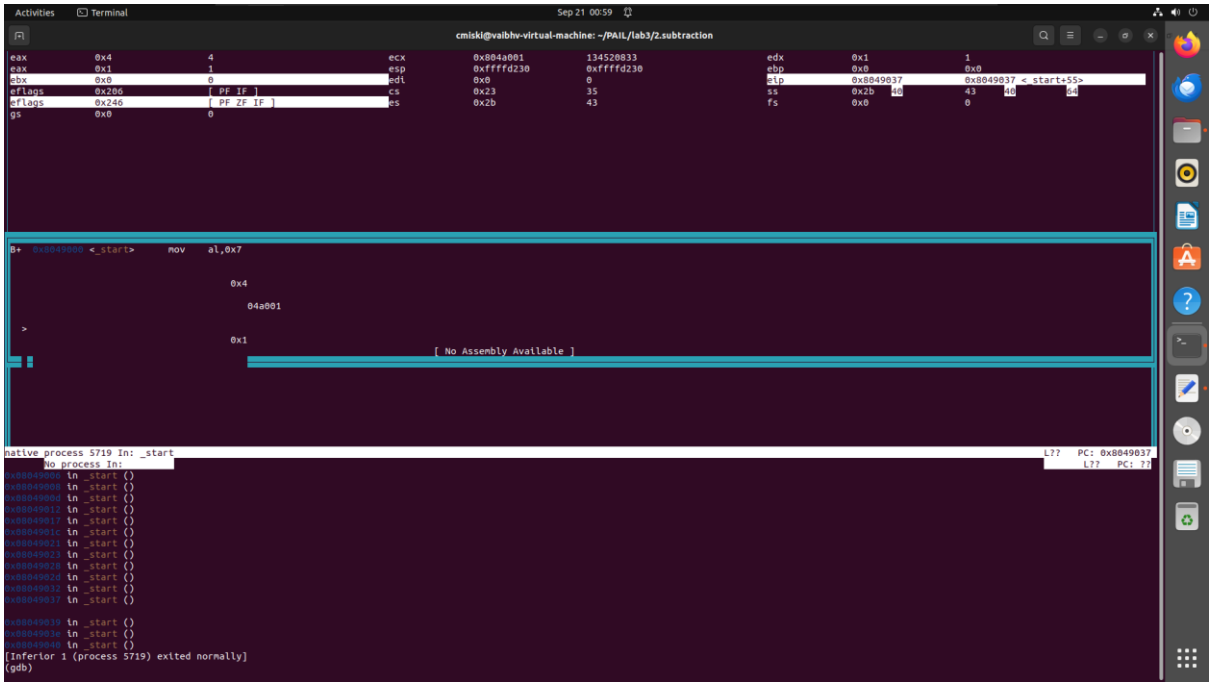
1. CONSOLE:



2. GDB:


Activities
Terminal
Sep 21 00:57
cmiski@vaibhv-virtual-machine: ~/PAIL/lab3/2.subtraction
Register group: general
eax 0x0 0 ecx 0x0 0 edx 0x0 0
ebx 0x0 0 esp 0xffffd230 0xffffd230 ebp 0x0 0x0
esi 0x0 0 edi 0x0 0 eip 0x8049000 0x8049000 <_start>
eflags 0x202 [IF] cs 0x23 35 ss 0x2b 43
ds 0x2b 43 es 0x2b 43 fs 0x0 0
gs 0x0 0
B+> 0x8049000 <_start> mov al,0x7
0x8049002 <_start+2> mov bl,0x2
0x8049004 <_start+4> sub al,bl
0x8049006 <_start+6> add al,0x30
0x8049008 <_start+8> mov ds:0x804a000,al
0x804900d <_start+13> mov eax,0x4
0x8049012 <_start+18> mov ebx,0x1
0x8049017 <_start+23> mov ecx,0x804a000
0x804901c <_start+28> mov edx,0x1
0x8049021 <_start+33> int 0x80
0x8049023 <_start+35> mov eax,0x4
0x8049028 <_start+40> mov ebx,0x1
0x804902d <_start+45> mov ecx,0x804a001
0x8049032 <_start+50> mov edx,0x1
0x8049037 <_start+55> int 0x80
0x8049039 <_start+57> mov eax,0x1
0x804903e <_start+62> xor ebx,ebx
native process 5719 In: _start L?? PC: 0x8049000
(gdb) layout regs
(gdb) b _start
Breakpoint 1 at 0x8049000
(gdb) r
Starting program: /home/cmiski/PAIL/lab3/2.subtraction/subtraction
Breakpoint 1, 0x08049000 in _start ()
(gdb)


Activities
Terminal
Sep 21 00:58
cmiski@vaibhv-virtual-machine: ~/PAIL/lab3/2.subtraction
Register group: general
eax 0x5 5 ecx 0x0 0 edx 0x0 0
ebx 0x2 2 esp 0xffffd230 0xffffd230 ebp 0x0 0x0
esi 0x0 0 edi 0x0 0 eip 0x8049006 0x8049006 <_start+6>
eflags 0x206 [PF IF] cs 0x23 35 ss 0x2b 43
ds 0x2b 43 es 0x2b 43 fs 0x0 0
gs 0x0 0
B+ 0x8049000 <_start> mov al,0x7
0x8049002 <_start+2> mov bl,0x2
0x8049004 <_start+4> sub al,bl
> 0x8049006 <_start+6> add al,0x30
0x8049008 <_start+8> mov ds:0x804a000,al
0x804900d <_start+13> mov eax,0x4
0x8049012 <_start+18> mov ebx,0x1
0x8049017 <_start+23> mov ecx,0x804a000
0x804901c <_start+28> mov edx,0x1
0x8049021 <_start+33> int 0x80
0x8049023 <_start+35> mov eax,0x4
0x8049028 <_start+40> mov ebx,0x1
0x804902d <_start+45> mov ecx,0x804a001
0x8049032 <_start+50> mov edx,0x1
0x8049037 <_start+55> int 0x80
0x8049039 <_start+57> mov eax,0x1
0x804903e <_start+62> xor ebx,ebx
native process 5719 In: _start L?? PC: 0x8049006
(gdb) layout regs
(gdb) b _start
Breakpoint 1 at 0x8049000
(gdb) r
Starting program: /home/cmiski/PAIL/lab3/2.subtraction/subtraction
Breakpoint 1, 0x08049000 in _start ()
(gdb) si
0x08049002 in _start ()
0x08049004 in _start ()
0x08049006 in _start ()
(gdb)



B] Using User Inputs

CODE:

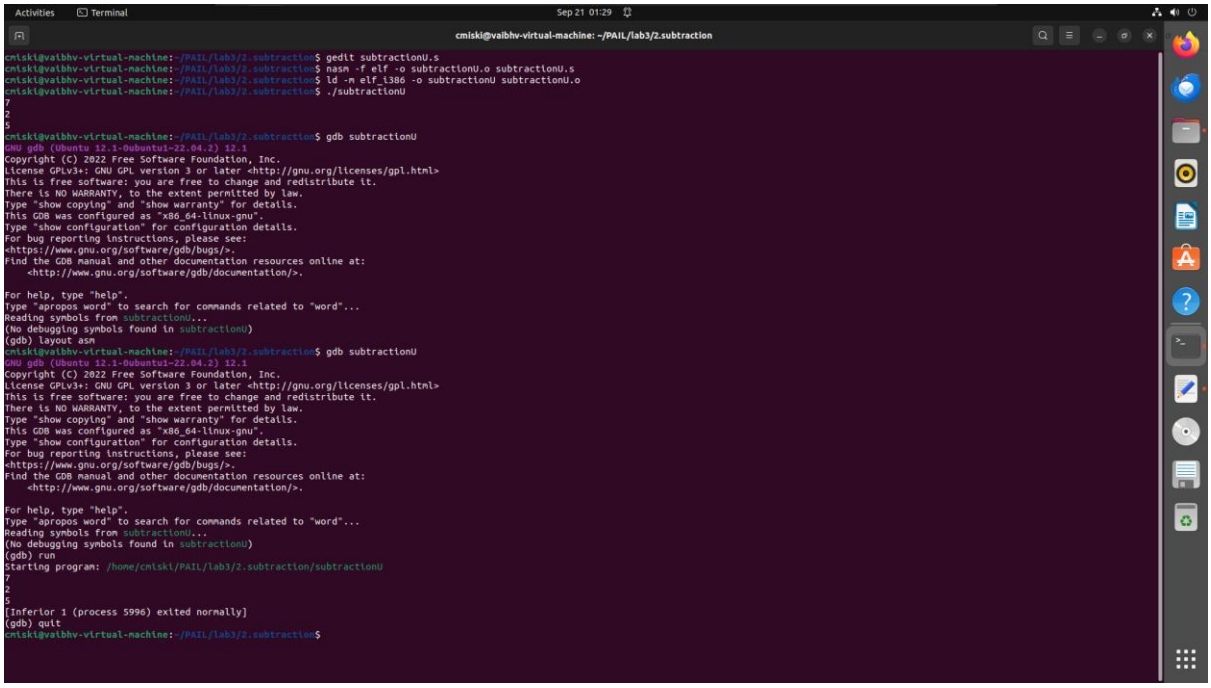
```
section .data
	result db 0
	newline db 10

section .bss
	input1 RESB 4
	input2 RESB 4

section .text
global _start

_start:
	MOV eax,3
	MOV ebx,0
	MOV ecx,input1
	MOV edx,4
	INT 80h
```

```
MOV eax,3
MOV ebx,0
MOV ecx,input2
MOV edx,4
INT 80h

MOV al,[input1]
SUB al,'0'
MOV bl,al

MOV al,[input2]
SUB al,'0'
SUB bl,al

ADD bl,'0'
MOV [result],bl

MOV eax,4
MOV ebx,1
MOV ecx,result
MOV edx,1
INT 80h

MOV eax,4
MOV ebx,1
MOV ecx,newline
MOV edx,1
INT 80h

MOV eax,1
XOR ebx,ebx
```

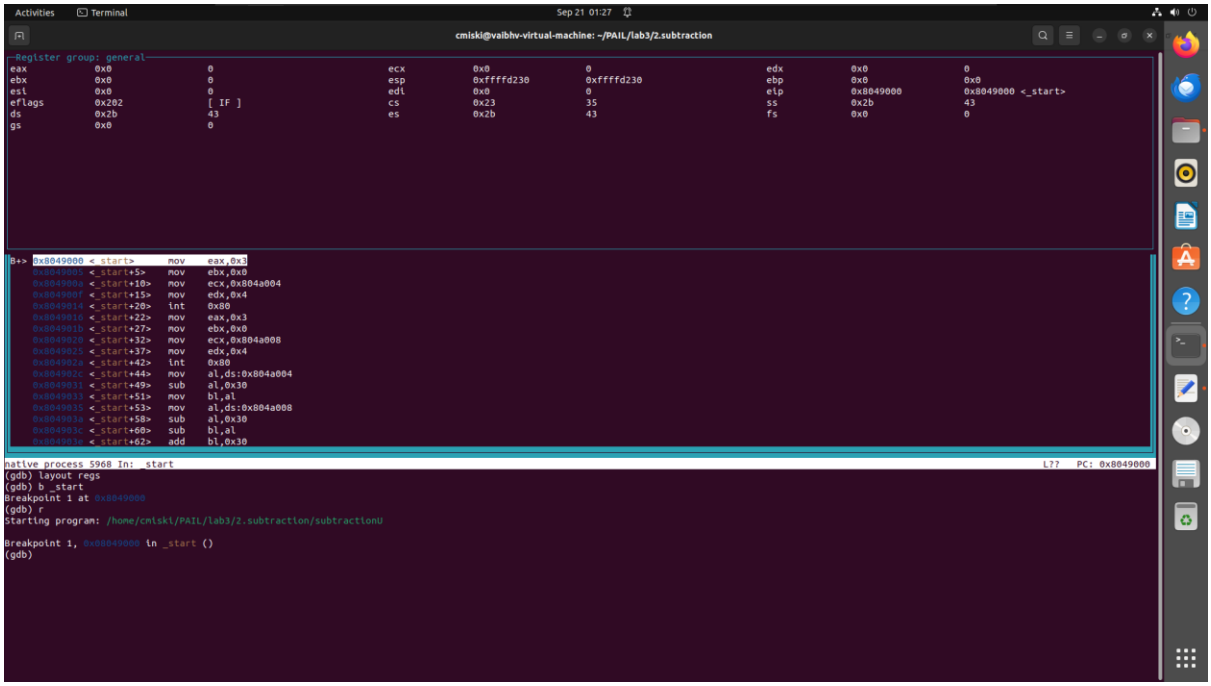
INT 80h

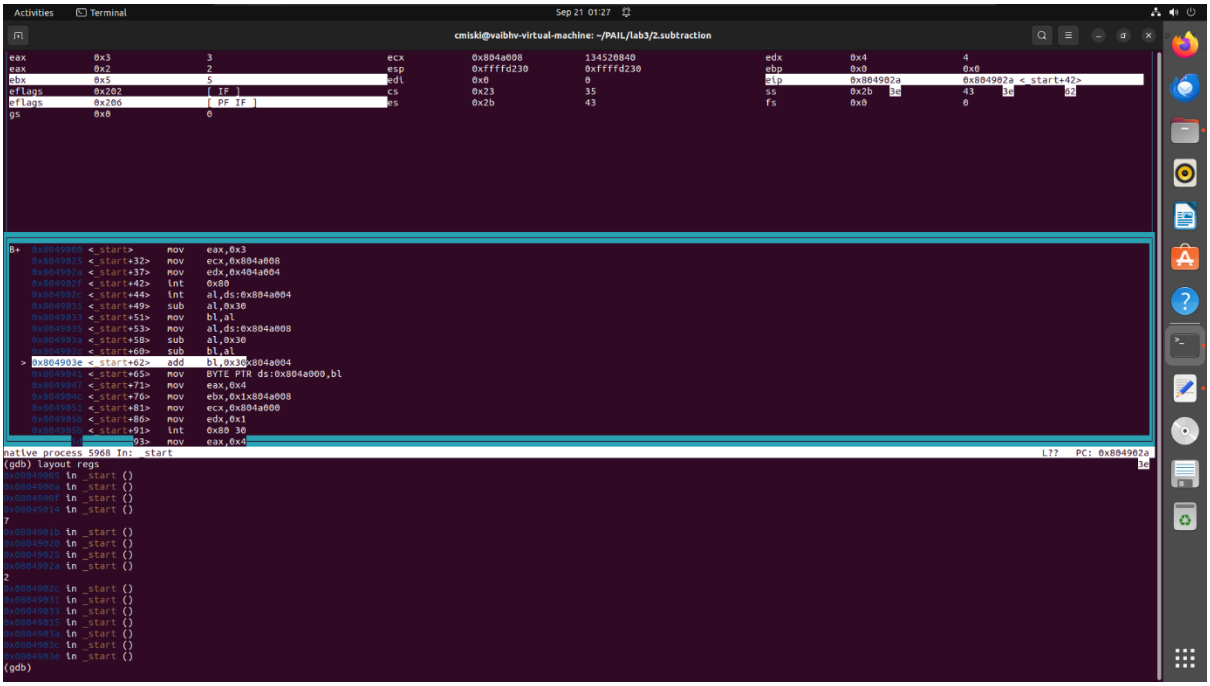
OUTPUT

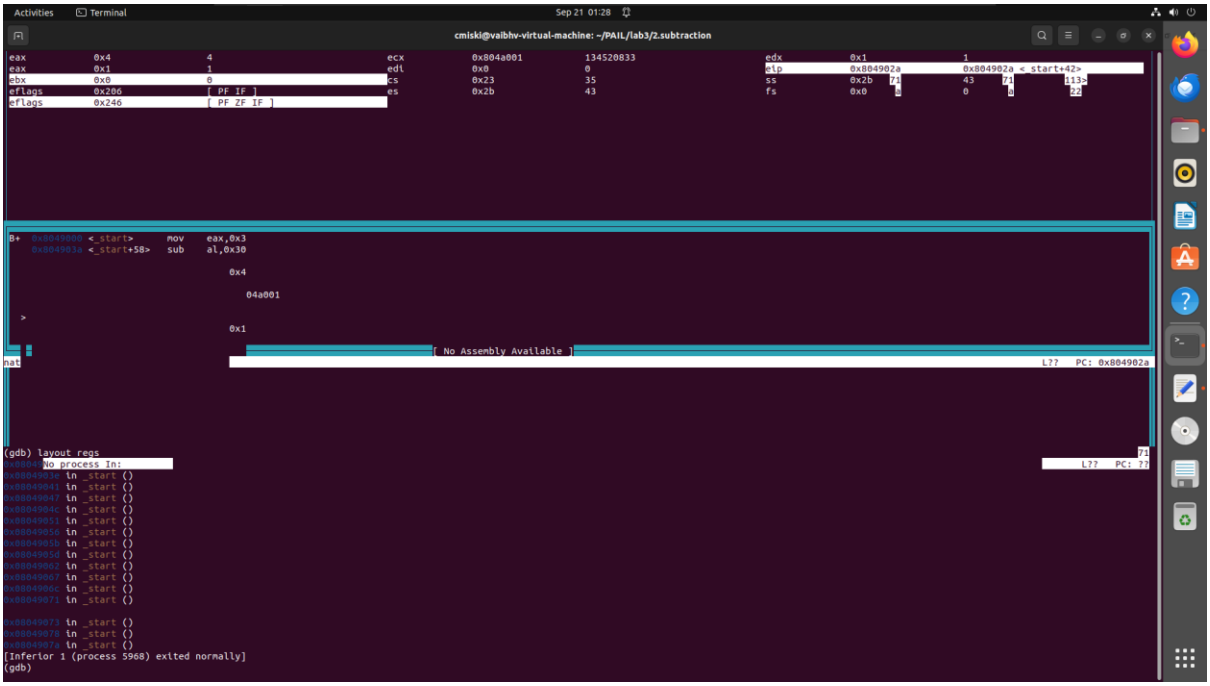
1. Terminal:



2. GDB:







MULTIPLICATION

A] on registers

Code:

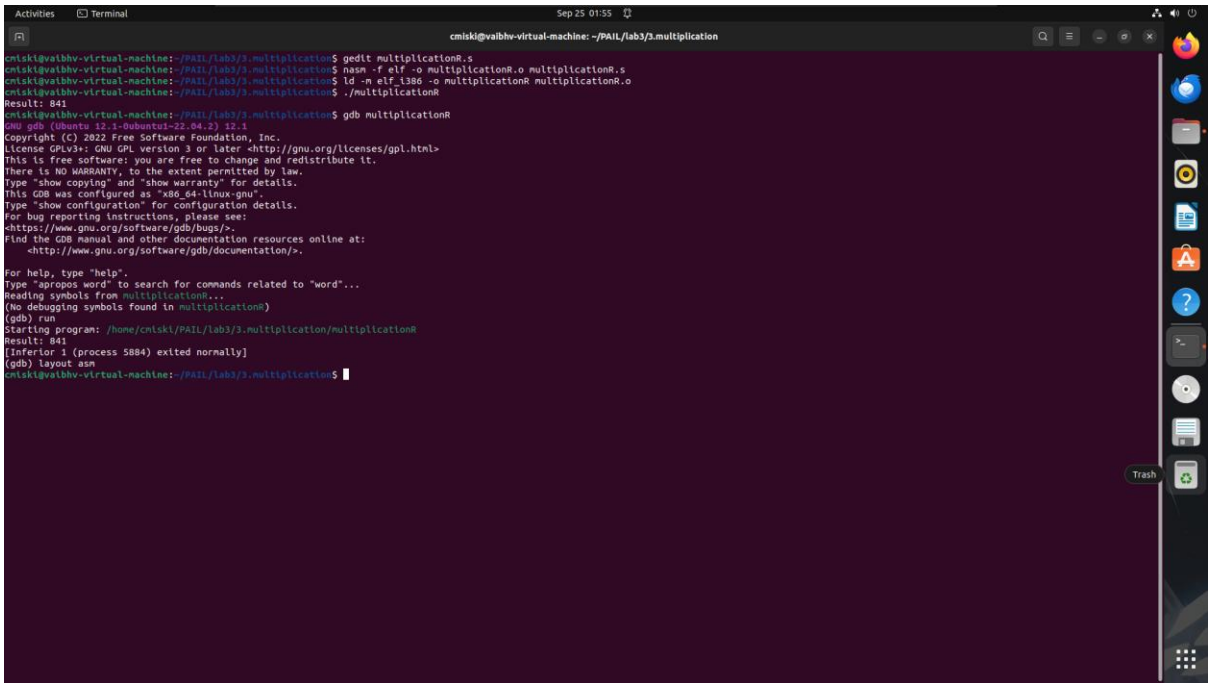
```
section .data

    resmsg db "Result: ", 0
```

```
    lenr equ $-resmsg
    newline db 0xA, 0

section .bss
    result resb 12

section .text
    global _start

_start:
    mov eax, 29
    mov ebx, 29
    imul eax, ebx

    mov edi, result + 11
    mov byte [edi], 0

.convert_loop:
    xor edx, edx
    mov ecx, 10
    div ecx
    add dl, '0'
    dec edi
    mov [edi], dl
    test eax, eax
    jnz .convert_loop

    mov eax, 4
    mov ebx, 1
    mov ecx, resmsg
    mov edx, lenr
    int 0x80
```

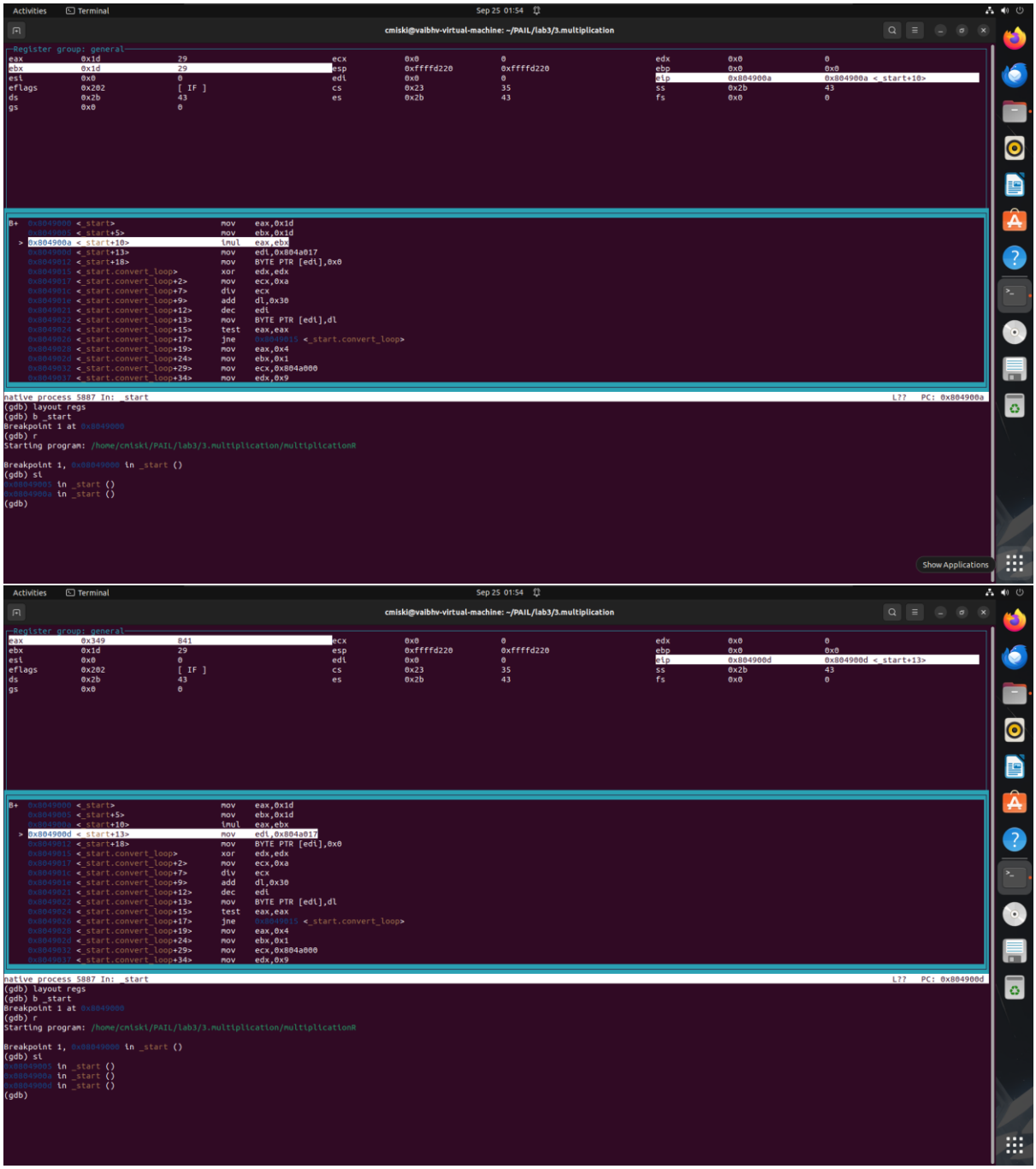
```
mov eax, 4
mov ebx, 1
mov ecx, edi
mov edx, result + 11
sub edx, edi
int 0x80

mov eax, 4
mov ebx, 1
mov ecx, newline
mov edx, 1
int 0x80

mov eax, 1
xor ebx, ebx
int 0x80
```

Output:

1]Console



2]GDB



B] taking user inputs

Code:

```
section .data
    msg1 db "Enter first number: ", 0
    len1 equ $-msg1
    msg2 db "Enter second number: ", 0
    len2 equ $-msg2
```

```
    resmsg db "Result: ", 0
    lenr equ $-resmsg
    newline db 0xA, 0

section .bss
    num1 resb 12
    num2 resb 12
    result resb 24

section .text
    global _start

_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, msg1
    mov edx, len1
    int 0x80

    mov eax, 3
    mov ebx, 0
    mov ecx, num1
    mov edx, 12
    int 0x80

    mov esi, num1
    xor eax, eax
.convert1:
    mov bl, [esi]
    cmp bl, 0xA
    je .done1
    cmp bl, 0
```

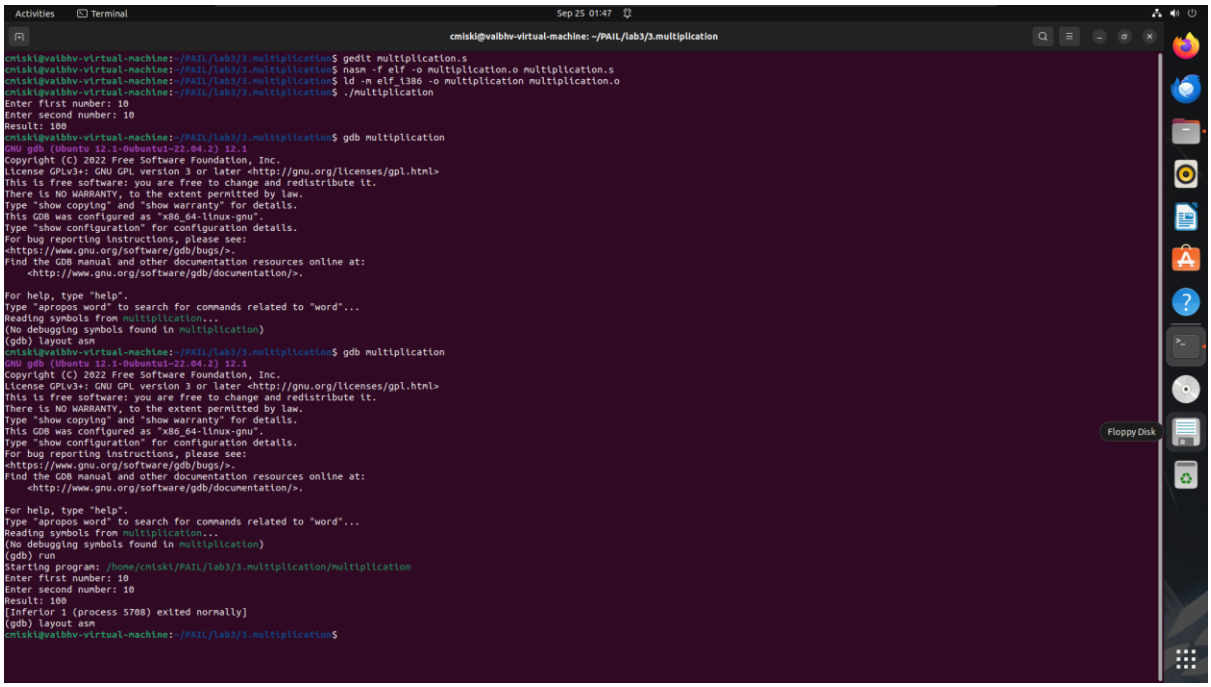
```
    je .done1
    sub bl, '0'
    imul eax, eax, 10
    add eax, ebx
    inc esi
    jmp .convert1
.done1:
    mov ebx, eax

    mov eax, 4
    mov ebx, 1
    mov ecx, msg2
    mov edx, len2
    int 0x80

    mov eax, 3
    mov ebx, 0
    mov ecx, num2
    mov edx, 12
    int 0x80

    mov esi, num2
    xor eax, eax
.convert2:
    mov bl, [esi]
    cmp bl, 0xA
    je .done2
    cmp bl, 0
    je .done2
    sub bl, '0'
    imul eax, eax, 10
    add eax, ebx
```

```
    inc esi
    jmp .convert2
.done2:
    mov ecx, eax

    mov eax, ebx
    imul eax, ecx

    mov edi, result + 23
    mov byte [edi], 0

.convert_loop:
    xor edx, edx
    mov ebx, 10
    div ebx
    add dl, '0'
    dec edi
    mov [edi], dl
    test eax, eax
    jnz .convert_loop

    mov eax, 4
    mov ebx, 1
    mov ecx, resmsg
    mov edx, lenr
    int 0x80

    mov eax, 4
    mov ebx, 1
    mov ecx, edi
    mov edx, result + 23
    sub edx, edi
```

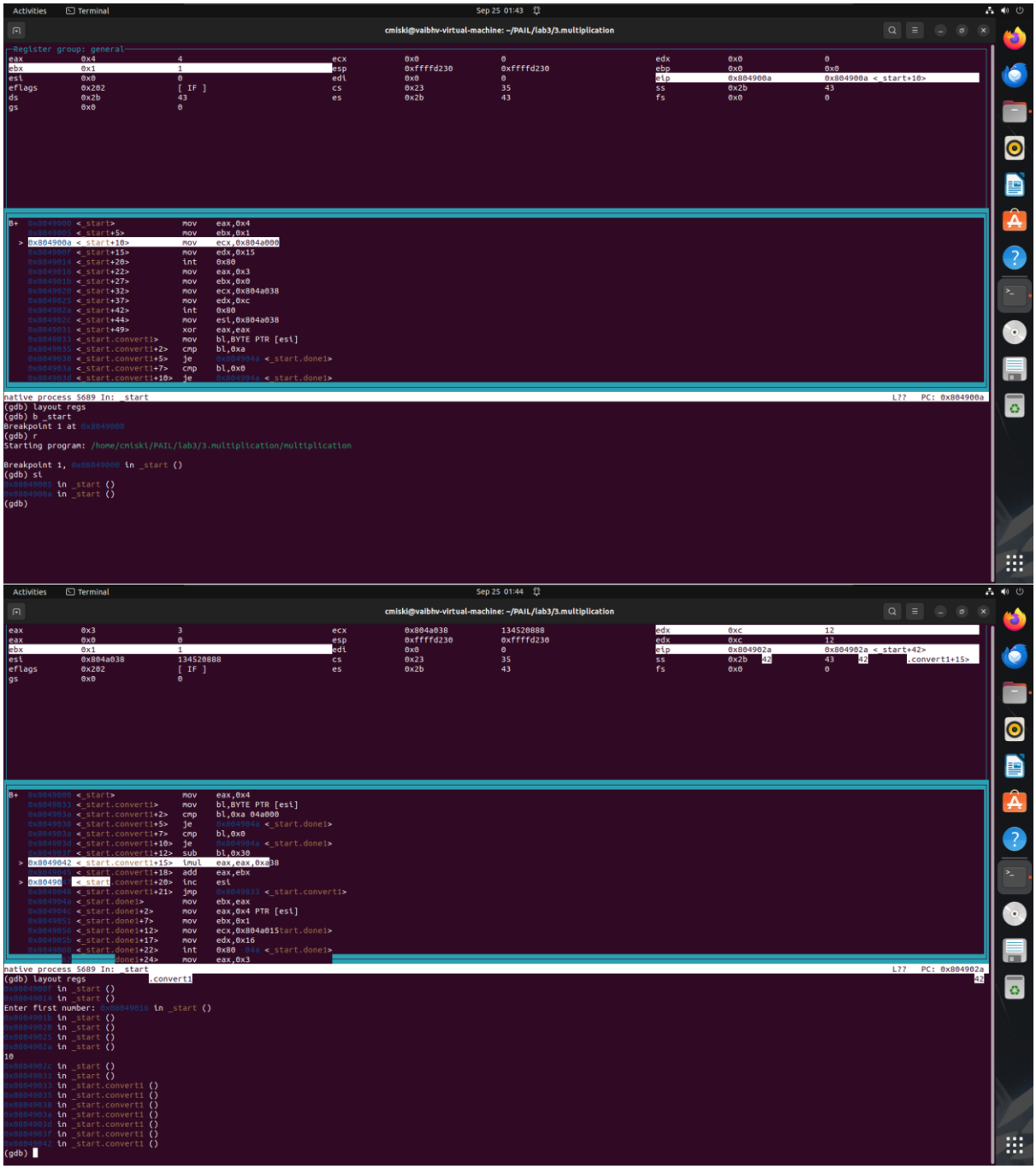
```
    int 0x80

    mov eax, 4
    mov ebx, 1
    mov ecx, newline
    mov edx, 1
    int 0x80

    mov eax, 1
    xor ebx, ebx
    int 0x80
```

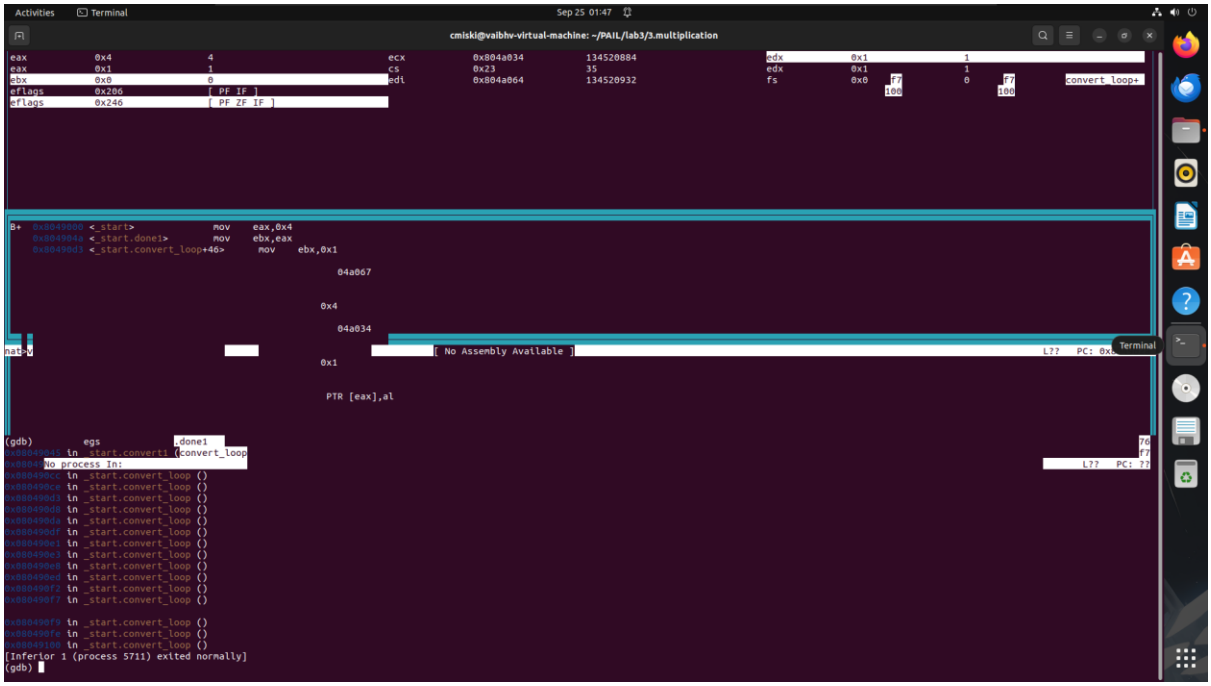
Output:

1]Console:



2]GDB:


Activities
Terminal
Sep 25 01:43
cmiski@vaibhv-virtual-machine: ~/PAIL/lab3/3.multiplication
Register group: general
eax 0x4 4
ebx 0x1 1
esi 0x0 0
eflags 0x202 [IF]
ds 0x2b 43
gs 0x0 0
ecx 0x0 0
esp 0xffffd230 0xffffd230
edi 0x0 0
cs 0x23 35
es 0x2b 43
edx 0x0 0
ebp 0x0 0x0
eip 0x804900a 0x804900a <_start+10>
ss 0x2b 43
fs 0x0 0
B+ 0x8049000 <_start> mov eax,0x4
0x8049005 <_start+5> mov ebx,0x1
> 0x804900a <_start+10> mov ecx,0x804a000
0x804900f <_start+15> mov edx,0x15
0x8049014 <_start+20> int 0x80
0x8049016 <_start+22> mov eax,0x3
0x804901b <_start+27> mov ebx,0x0
0x8049020 <_start+32> mov ecx,0x804a038
0x8049025 <_start+37> mov edx,0xc
0x804902a <_start+42> int 0x80
0x804902c <_start+44> mov esi,0x804a038
0x8049031 <_start+49> xor eax,eax
0x8049033 <_start.convert1> mov bl,BYTE PTR [esi]
0x8049035 <_start.convert1+2> cmp bl,0xa
0x8049038 <_start.convert1+5> je 0x804904a <_start.done1>
0x804903a <_start.convert1+7> cmp bl,0x0
0x804903d <_start.convert1+10> je 0x804904a <_start.done1>
native process 5689 In: _start L?? PC: 0x804900a
(gdb) layout regs
(gdb) b _start
Breakpoint 1 at 0x8049000
(gdb) r
Starting program: /home/cmiski/PAIL/lab3/3.multiplication/multiplication
Breakpoint 1, 0x08049000 in _start ()
(gdb) si
0x8049005 in _start ()
0x804900a in _start ()
(gdb)
Activities
Terminal
Sep 25 01:44
cmiski@vaibhv-virtual-machine: ~/PAIL/lab3/3.multiplication
eax 0x3 3
eax 0x0 0
ebx 0x1 1
esi 0x804a038 134520888
eflags 0x202 [IF]
gs 0x0 0
ecx 0x804a038 134520888
esp 0xffffd230 0xffffd230
edi 0x0 0
cs 0x23 35
es 0x2b 43
edx 0xc 12
edx 0xc 12
eip 0x804902a 0x804902a <_start+42>
ss 0x2b 43
fs 0x0 0
B+ 0x8049000 <_start> mov eax,0x4
0x8049033 <_start.convert1> mov bl,BYTE PTR [esi]
0x8049035 <_start.convert1+2> cmp bl,0xa 0x4a000
0x8049038 <_start.convert1+5> je 0x804904a <_start.done1>
0x804903a <_start.convert1+7> cmp bl,0x0
0x804903d <_start.convert1+10> je 0x804904a <_start.done1>
0x804903f <_start.convert1+12> sub bl,0x30
> 0x8049042 <_start.convert1+15> imul eax,eax,0xa38
0x8049045 <_start.convert1+18> add eax,ebx
> 0x8049047 <_start.convert1+20> inc esi
0x8049048 <_start.convert1+21> jmp 0x8049033 <_start.convert1>
0x804904a <_start.done1> mov ebx,eax
0x804904c <_start.done1+2> mov eax,0x4 PTR [esi]
0x8049051 <_start.done1+7> mov ebx,0x1
0x8049056 <_start.done1+12> mov ecx,0x804a015start.done1>
0x804905b <_start.done1+17> mov edx,0x16
0x8049060 <_start.done1+22> int 0x80 04a <_start.done1>
done1+24> mov eax,0x3
native process 5689 In: _start L?? PC: 0x804902a
(gdb) layout regs .convert1 42
0x8049007 in _start ()
0x8049014 in _start ()
Enter first number: 0x8049016 in _start ()
0x804901b in _start ()
0x8049020 in _start ()
0x8049025 in _start ()
0x804902a in _start ()
10
0x804902c in _start ()
0x8049031 in _start ()
0x8049033 in _start.convert1 ()
0x8049035 in _start.convert1 ()
0x8049038 in _start.convert1 ()
0x804903a in _start.convert1 ()
0x804903d in _start.convert1 ()
0x804903f in _start.convert1 ()
0x8049042 in _start.convert1 ()
(gdb)



Division

A] on registers

Code:

```
section .data
    qmsg db "Quotient: ",0
    lenq equ $-qmsg
    rmsg db " Remainder: ",0
    lenr equ $-rmsg
    newline db 0xA,0

section .bss
    quotient resb 12
    remainder resb 12

section .text
    global _start
```

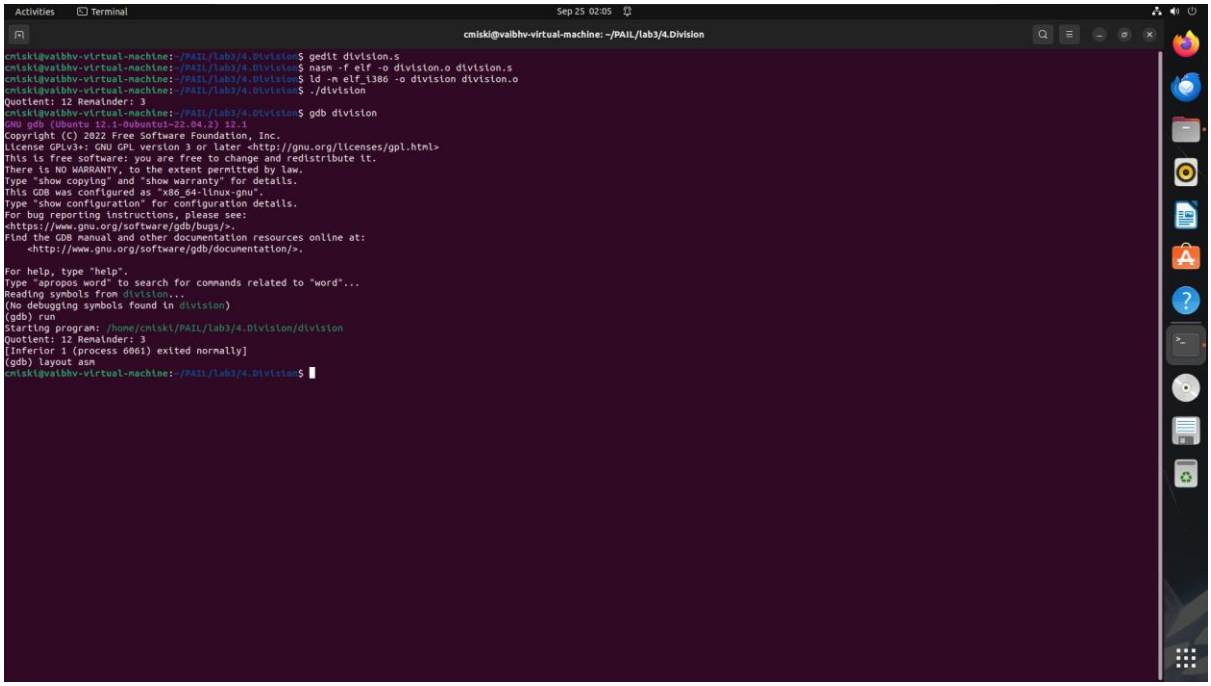
```
_start:
    mov eax,123       ; dividend
    mov ebx,10        ; divisor
    xor edx,edx       ; clear EDX before division
    div ebx           ; EAX = quotient, EDX = remainder

    mov esi,eax       ; store quotient
    mov edi,edx       ; store remainder

    ; Convert quotient to string
    mov eax,esi
    mov ecx,quotient+11
    mov byte [ecx],0
.q_loop:
    xor edx,edx
    mov ebp,10
    div ebp
    add dl,'0'
    dec ecx
    mov [ecx],dl
    test eax,eax
    jnz .q_loop

    ; Convert remainder to string
    mov eax,edi
    mov ecx,remainder+11
    mov byte [ecx],0
.r_loop:
    xor edx,edx
    mov ebp,10
    div ebp
    add dl,'0'
```

```
dec ecx
mov [ecx],dl
test eax,eax
jnz .r_loop

; Print quotient
mov eax,4
mov ebx,1
mov ecx,qmsg
mov edx,lenq
int 0x80
mov eax,4
mov ebx,1
mov ecx,ecx
mov ecx,ecx
mov ecx,ecx
mov ecx,quotient
mov edx,quotient+11
sub edx,ecx
int 0x80

; Print remainder
mov eax,4
mov ebx,1
mov ecx,rmsg
mov edx,lenr
int 0x80
mov eax,4
mov ebx,1
mov ecx,remainder
mov edx,remainder+11
sub edx,ecx
```

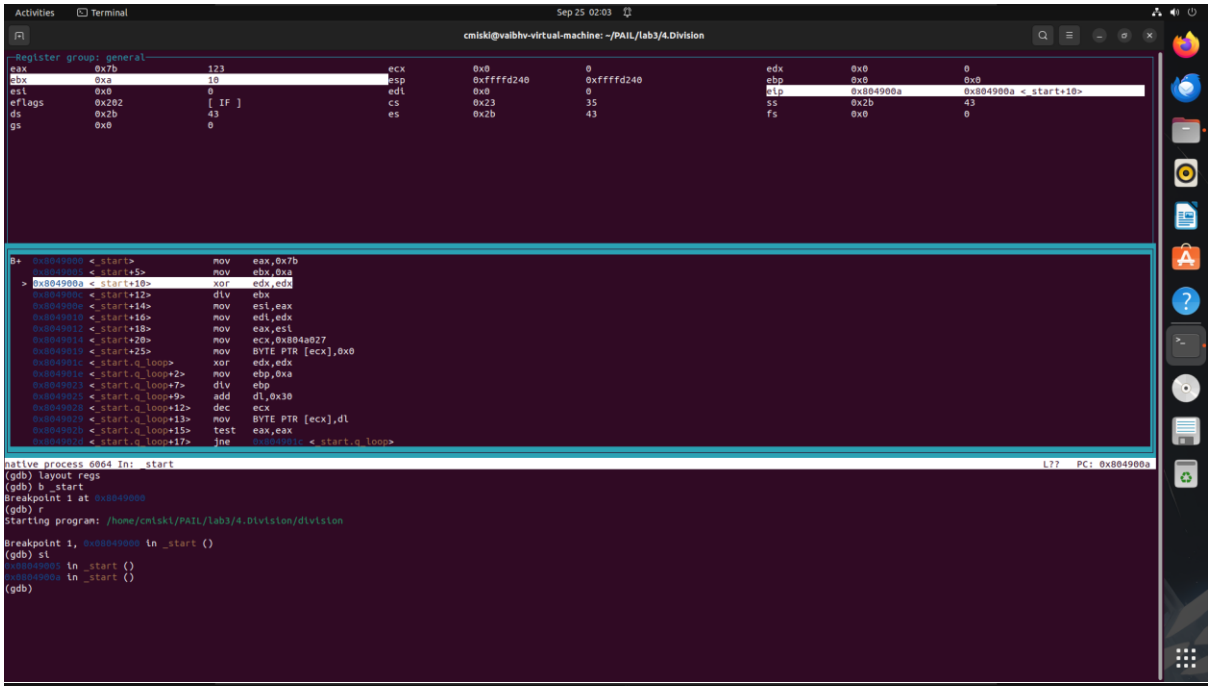
```
    int 0x80

    ; Print newline
    mov eax,4
    mov ebx,1
    mov ecx,newline
    mov edx,1
    int 0x80

    ; Exit
    mov eax,1
    xor ebx,ebx
    int 0x80
```

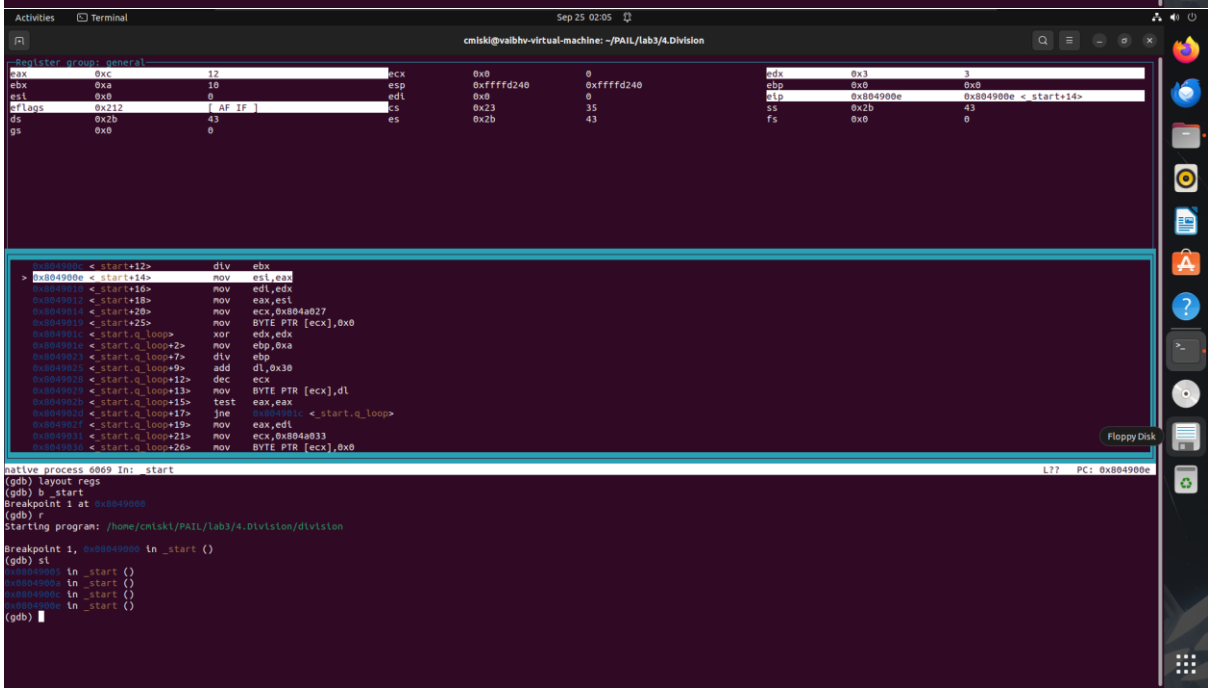
Output:

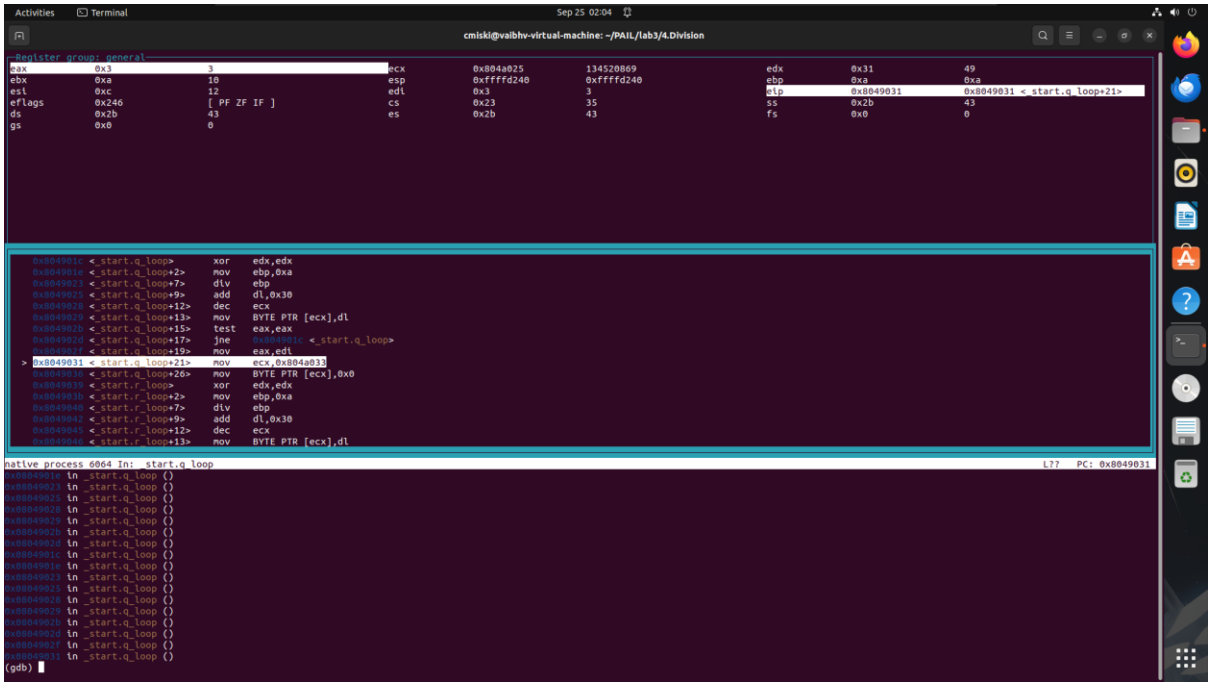
1] Console



2] GDB


Activities
Terminal
Sep 25 02:03
cmiski@vaibhv-virtual-machine: ~/PAIL/lab3/4.Division
Register group: general
eax 0x7b 123
ebx 0xa 10
esi 0x0 0
eflags 0x202 [IF]
ds 0x2b 43
gs 0x0 0
ecx 0x0 0
esp 0xffffd240 0xffffd240
edl 0x0 0
cs 0x23 35
es 0x2b 43
edx 0x0 0
ebp 0x0 0x0
eip 0x804900a 0x804900a <_start+10>
ss 0x2b 43
fs 0x0 0
B+ 0x8049000 <_start> mov eax,0x7b
0x8049005 <_start+5> mov ebx,0xa
> 0x804900a <_start+10> xor edx,edx
0x804900c <_start+12> div ebx
0x804900e <_start+14> mov esi,eax
0x8049010 <_start+16> mov edi,edx
0x8049012 <_start+18> mov eax,esi
0x8049014 <_start+20> mov ecx,0x804a027
0x8049019 <_start+25> mov BYTE PTR [ecx],0x0
0x804901c <_start.q_loop> xor edx,edx
0x804901e <_start.q_loop+2> mov ebp,0xa
0x8049023 <_start.q_loop+7> div ebp
0x8049025 <_start.q_loop+9> add dl,0x30
0x8049028 <_start.q_loop+12> dec ecx
0x8049029 <_start.q_loop+13> mov BYTE PTR [ecx],dl
0x804902b <_start.q_loop+15> test eax,eax
0x804902d <_start.q_loop+17> jne 0x804901c <_start.q_loop>
native process 6064 In: _start
L?? PC: 0x804900a
(gdb) layout regs
(gdb) b _start
Breakpoint 1 at 0x8049000
(gdb) r
Starting program: /home/cmiski/PAIL/lab3/4.Division/division
Breakpoint 1, 0x08049000 in _start ()
(gdb) si
0x08049005 in _start ()
0x0804900a in _start ()
(gdb)


Activities
Terminal
Sep 25 02:05
cmiski@vaibhv-virtual-machine: ~/PAIL/lab3/4.Division
Register group: general
eax 0xc 12
ebx 0xa 10
esi 0x0 0
eflags 0x212 [AF IF]
ds 0x2b 43
gs 0x0 0
ecx 0x0 0
esp 0xffffd240 0xffffd240
edl 0x0 0
cs 0x23 35
es 0x2b 43
edx 0x3 3
ebp 0x0 0x0
eip 0x804900e 0x804900e <_start+14>
ss 0x2b 43
fs 0x0 0
0x804900c <_start+12> div ebx
> 0x804900e <_start+14> mov esi,eax
0x8049010 <_start+16> mov edi,edx
0x8049012 <_start+18> mov eax,esi
0x8049014 <_start+20> mov ecx,0x804a027
0x8049019 <_start+25> mov BYTE PTR [ecx],0x0
0x804901c <_start.q_loop> xor edx,edx
0x804901e <_start.q_loop+2> mov ebp,0xa
0x8049023 <_start.q_loop+7> div ebp
0x8049025 <_start.q_loop+9> add dl,0x30
0x8049028 <_start.q_loop+12> dec ecx
0x8049029 <_start.q_loop+13> mov BYTE PTR [ecx],dl
0x804902b <_start.q_loop+15> test eax,eax
0x804902d <_start.q_loop+17> jne 0x804901c <_start.q_loop>
0x804902f <_start.q_loop+19> mov eax,edi
0x8049031 <_start.q_loop+21> mov ecx,0x804a033
0x8049036 <_start.q_loop+26> mov BYTE PTR [ecx],0x0
Floppy Disk
native process 6069 In: _start
L?? PC: 0x804900e
(gdb) layout regs
(gdb) b _start
Breakpoint 1 at 0x8049000
(gdb) r
Starting program: /home/cmiski/PAIL/lab3/4.Division/division
Breakpoint 1, 0x08049000 in _start ()
(gdb) si
0x08049005 in _start ()
0x0804900a in _start ()
0x0804900c in _start ()
0x0804900e in _start ()
(gdb)



B] Taking user inputs

Code:

```
section .data
    msg1 db "Enter dividend: ",0
    len1 equ $-msg1
    msg2 db "Enter divisor: ",0
    len2 equ $-msg2
    qmsg db "Quotient: ",0
    lenq equ $-qmsg
    rmsg db " Remainder: ",0
    lenr equ $-rmsg
    newline db 0xA,0

section .bss
    num1 resb 12
    num2 resb 12
    quotient resb 12
    remainder resb 12

section .text
```

```
    global _start

_start:
    ; ---- Read dividend ----
    mov eax,4
    mov ebx,1
    mov ecx,msg1
    mov edx,len1
    int 0x80

    mov eax,3
    mov ebx,0
    mov ecx,num1
    mov edx,12
    int 0x80

    ; ---- Convert dividend to integer ----
    mov esi,num1
    xor eax,eax
.convert1:
    mov bl,[esi]
    cmp bl,0xA          ; stop at newline
    je .done1
    sub bl,'0'
    imul eax,eax,10
    add eax,ebx
    inc esi
    jmp .convert1
.done1:
    mov esi,eax          ; store dividend in ESI

    ; ---- Read divisor ----
```

```
    mov eax,4
    mov ebx,1
    mov ecx,msg2
    mov edx,len2
    int 0x80

    mov eax,3
    mov ebx,0
    mov ecx,num2
    mov edx,12
    int 0x80

    ; ---- Convert divisor to integer ----
    mov edi,num2
    xor eax,eax
.convert2:
    mov bl,[edi]
    cmp bl,0xA
    je .done2
    sub bl,'0'
    imul eax,eax,10
    add eax,ebx
    inc edi
    jmp .convert2
.done2:
    mov edi,eax

    ; ---- Perform division ----
    mov eax,esi
    xor edx,edx
    div edi
    mov esi,eax
```

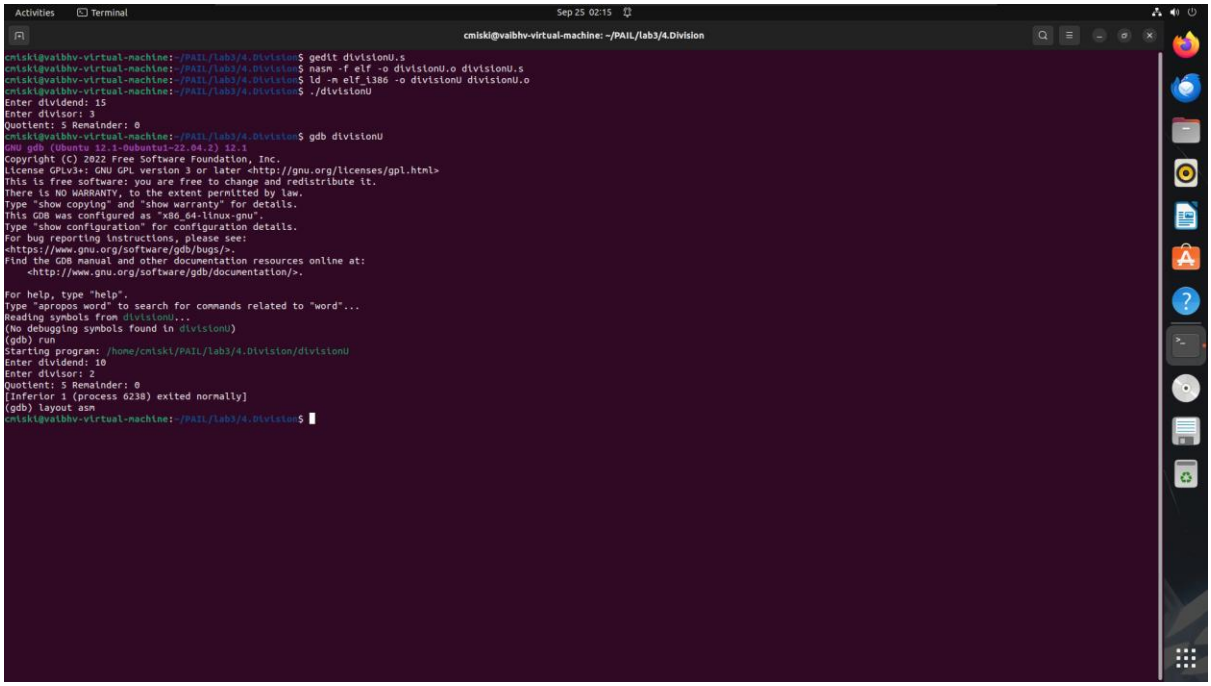
```
    mov edi,edx

    ; ---- Convert quotient to string ----
    mov eax,esi
    mov ecx,quotient+11
    mov byte [ecx],0
.q_loop:
    xor edx,edx
    mov ebp,10
    div ebp
    add dl,'0'
    dec ecx
    mov [ecx],dl
    test eax,eax
    jnz .q_loop

    ; ---- Convert remainder to string ----
    mov eax,edi
    mov ecx,remainder+11
    mov byte [ecx],0
.r_loop:
    xor edx,edx
    mov ebp,10
    div ebp
    add dl,'0'
    dec ecx
    mov [ecx],dl
    test eax,eax
    jnz .r_loop

    ; ---- Print quotient ----
    mov eax,4
```

```
mov ebx,1
mov ecx,qmsg
mov edx,lenq
int 0x80
mov eax,4
mov ebx,1
mov ecx,quotient
mov edx,quotient+11
sub edx,ecx
int 0x80

; ---- Print remainder ----
mov eax,4
mov ebx,1
mov ecx,rmsg
mov edx,lenr
int 0x80
mov eax,4
mov ebx,1
mov ecx,remainder
mov edx,remainder+11
sub edx,ecx
int 0x80

; ---- Print newline ----
mov eax,4
mov ebx,1
mov ecx,newline
mov edx,1
int 0x80

; ---- Exit ----
```

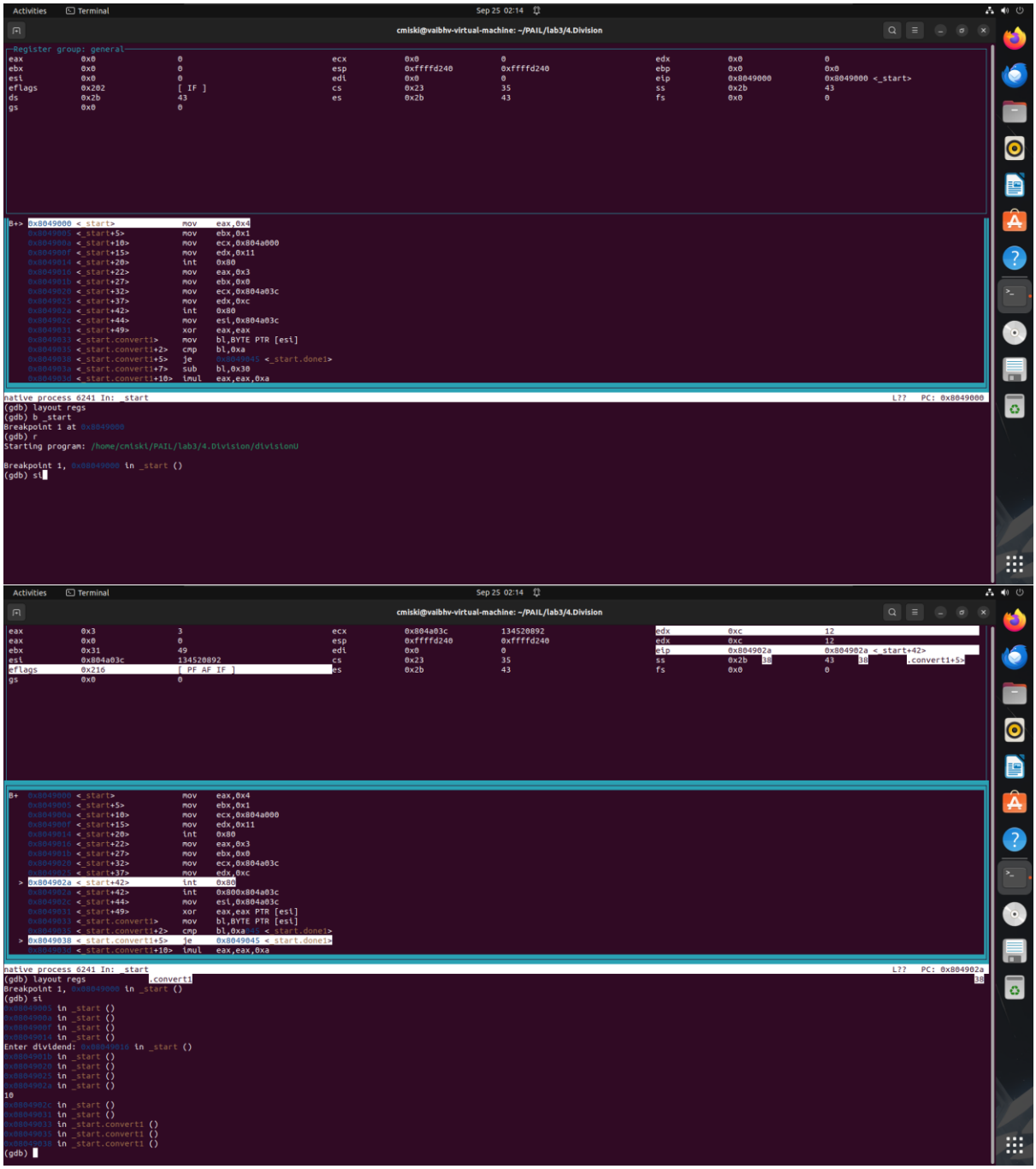
```
    mov eax,1
    xor ebx,ebx
    int 0x80
```

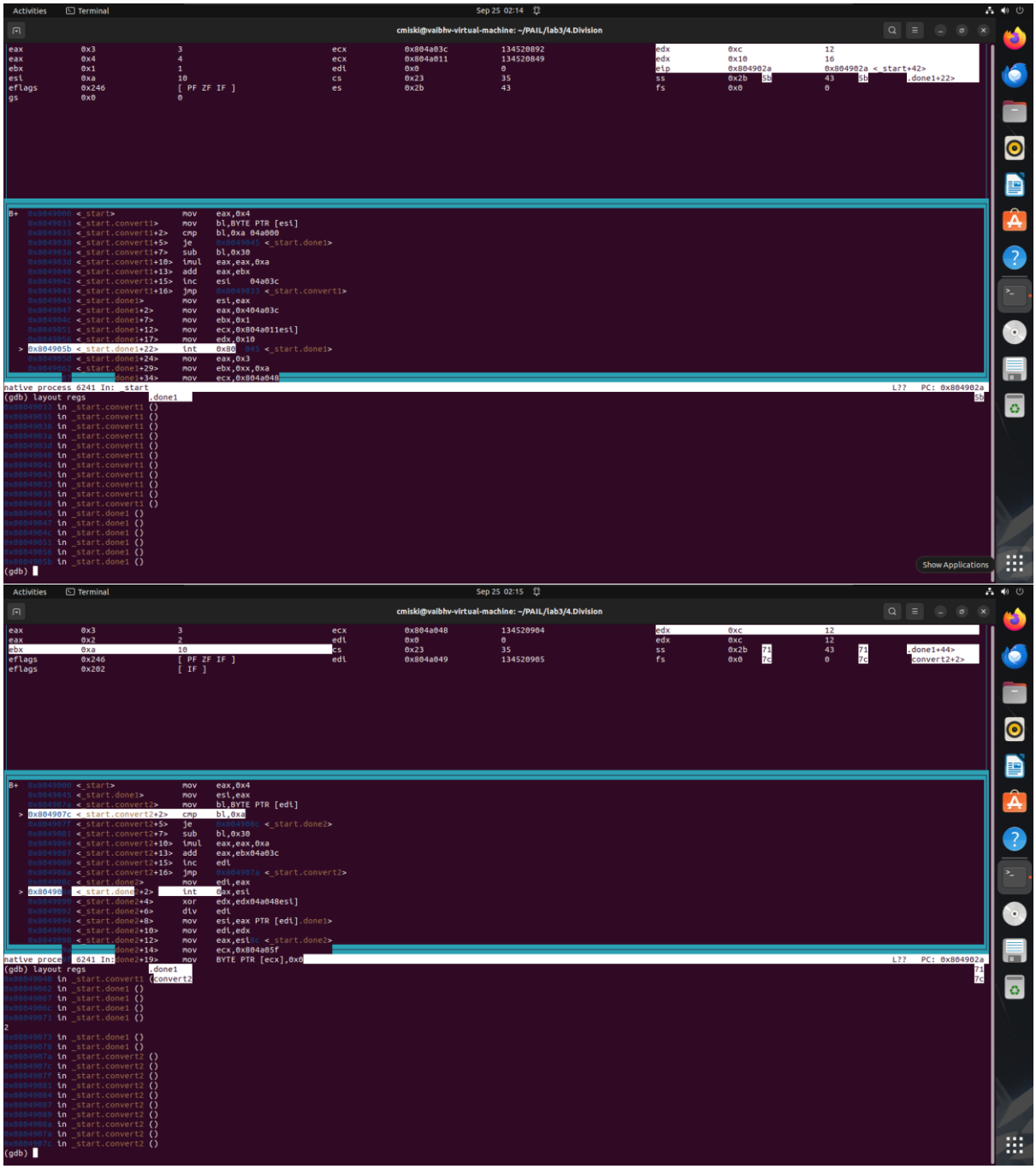
Output:

1] Console



2] GDB


Activities
Terminal
Sep 25 02:14
cmiski@vaibhv-virtual-machine: ~/PAIL/lab3/4.Division
Register group: general
eax 0x0 0 ecx 0x0 0 edx 0x0 0
ebx 0x0 0 esp 0xffffd240 0xffffd240 ebp 0x0 0x0
esi 0x0 0 edi 0x0 0 eip 0x8049000 0x8049000 <_start>
eflags 0x202 [IF] cs 0x23 35 ss 0x2b 43
ds 0x2b 43 es 0x2b 43 fs 0x0 0
gs 0x0 0
B+> 0x8049000 <_start> mov eax,0x4
0x8049005 <_start+5> mov ebx,0x1
0x804900a <_start+10> mov ecx,0x804a000
0x804900f <_start+15> mov edx,0x11
0x8049014 <_start+20> int 0x80
0x8049016 <_start+22> mov eax,0x3
0x804901b <_start+27> mov ebx,0x0
0x8049020 <_start+32> mov ecx,0x804a03c
0x8049025 <_start+37> mov edx,0xc
0x804902a <_start+42> int 0x80
0x804902c <_start+44> mov esi,0x804a03c
0x8049031 <_start+49> xor eax,eax
0x8049033 <_start.convert1> mov bl,BYTE PTR [esi]
0x8049035 <_start.convert1+2> cmp bl,0xa
0x8049038 <_start.convert1+5> je 0x8049045 <_start.done1>
0x804903a <_start.convert1+7> sub bl,0x30
0x804903d <_start.convert1+10> imul eax,eax,0xa
native process 6241 In: _start L?? PC: 0x8049000
(gdb) layout regs
(gdb) b _start
Breakpoint 1 at 0x8049000
(gdb) r
Starting program: /home/cmiski/PAIL/lab3/4.Division/divisionU
Breakpoint 1, 0x08049000 in _start ()
(gdb) st
Activities
Terminal
Sep 25 02:14
cmiski@vaibhv-virtual-machine: ~/PAIL/lab3/4.Division
eax 0x3 3 ecx 0x804a03c 134520892 edx 0xc 12
eax 0x0 0 esp 0xffffd240 0xffffd240 edx 0xc 12
ebx 0x31 49 edi 0x0 0 eip 0x804902a 0x804902a <_start+42>
esi 0x804a03c 134520892 cs 0x23 35 ss 0x2b 43 0x8049038 <_start.convert1+5>
eflags 0x216 [PF AF IF] es 0x2b 43 fs 0x0 0
gs 0x0 0
B+ 0x8049000 <_start> mov eax,0x4
0x8049005 <_start+5> mov ebx,0x1
0x804900a <_start+10> mov ecx,0x804a000
0x804900f <_start+15> mov edx,0x11
0x8049014 <_start+20> int 0x80
0x8049016 <_start+22> mov eax,0x3
0x804901b <_start+27> mov ebx,0x0
0x8049020 <_start+32> mov ecx,0x804a03c
0x8049025 <_start+37> mov edx,0xc
> 0x804902a <_start+42> int 0x80
0x804902c <_start+44> mov esi,0x804a03c
0x8049031 <_start+49> xor eax,eax
0x8049033 <_start.convert1> mov bl,BYTE PTR [esi]
0x8049035 <_start.convert1+2> cmp bl,0xa
> 0x8049038 <_start.convert1+5> je 0x8049045 <_start.done1>
0x804903a <_start.convert1+7> sub bl,0x30
0x804903d <_start.convert1+10> imul eax,eax,0xa
native process 6241 In: _start L?? PC: 0x804902a
(gdb) layout regs
Breakpoint 1, 0x08049000 in _start ()
(gdb) st
Enter dividend:
10



Lab-4

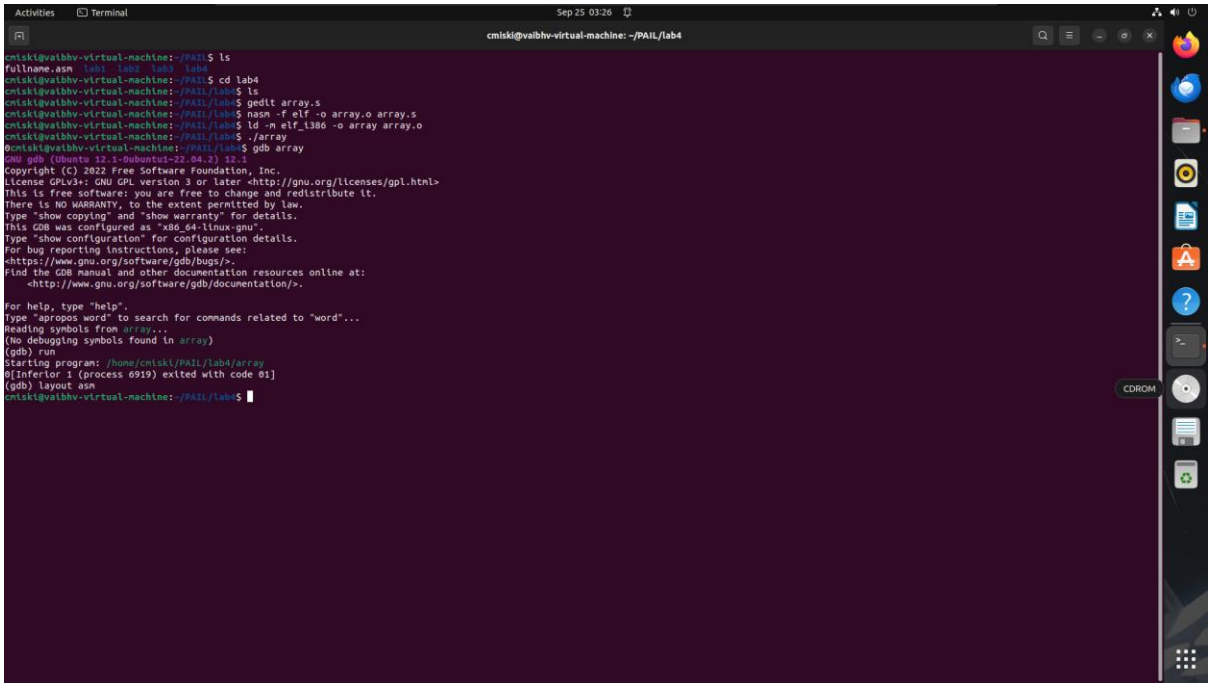
A] Array Addition (result less than 10 and 2nd by using AAM)

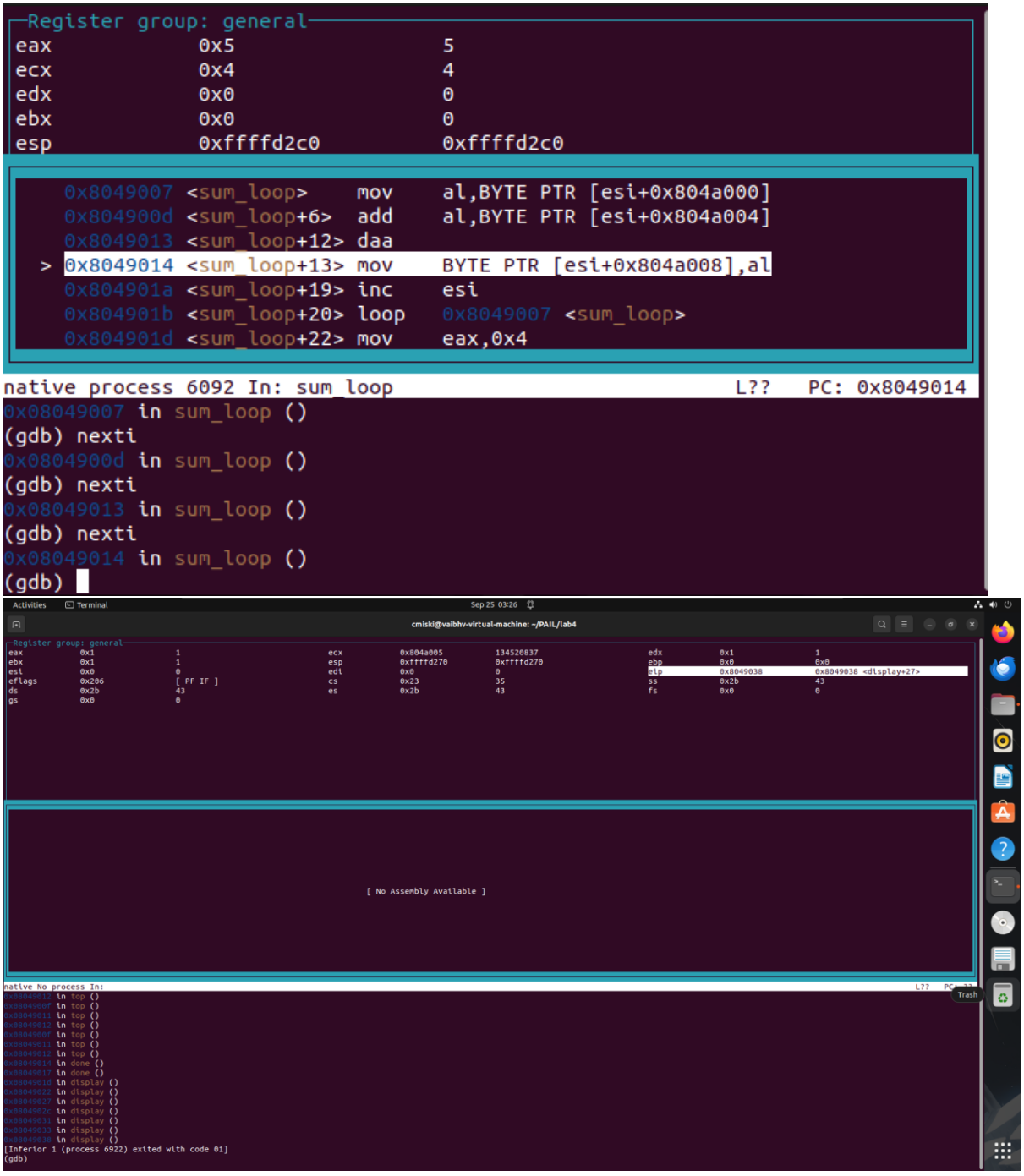
Code:

```
section .text

global _start
```

```
_start:

    mov eax, x
    mov ebx, 0
    mov ecx, 5

top:
    add bl, [eax]
    inc eax
    loop top

done:
    add bl, '0'
    mov [sum], bl

display:
    mov edx, 1
    mov ecx, sum
    mov ebx, 1
    mov eax, 4
    int 0x80

    mov eax, 1
    int 0x80

section .data
x:
    times 5 db 0

sum:
    db 0, 0xa
```

Output:


Activities
Terminal
Sep 25 03:26
cmiski@vaibhv-virtual-machine: ~/PAIL/lab4
cmiski@vaibhv-virtual-machine:~/PAIL$ ls
fullname.asm lab1 lab2 lab3 lab4
cmiski@vaibhv-virtual-machine:~/PAIL$ cd lab4
cmiski@vaibhv-virtual-machine:~/PAIL/lab4$ ls
cmiski@vaibhv-virtual-machine:~/PAIL/lab4$ gedit array.s
cmiski@vaibhv-virtual-machine:~/PAIL/lab4$ nasm -f elf -o array.o array.s
cmiski@vaibhv-virtual-machine:~/PAIL/lab4$ ld -m elf_i386 -o array array.o
cmiski@vaibhv-virtual-machine:~/PAIL/lab4$ ./array
0cmiski@vaibhv-virtual-machine:~/PAIL/lab4$ gdb array
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 array...
(No debugging symbols found in array)
(gdb) run
Starting program: /home/cmiski/PAIL/lab4/array
0[Inferior 1 (process 6919) exited with code 01]
(gdb) layout asm
cmiski@vaibhv-virtual-machine:~/PAIL/lab4$
CDROM



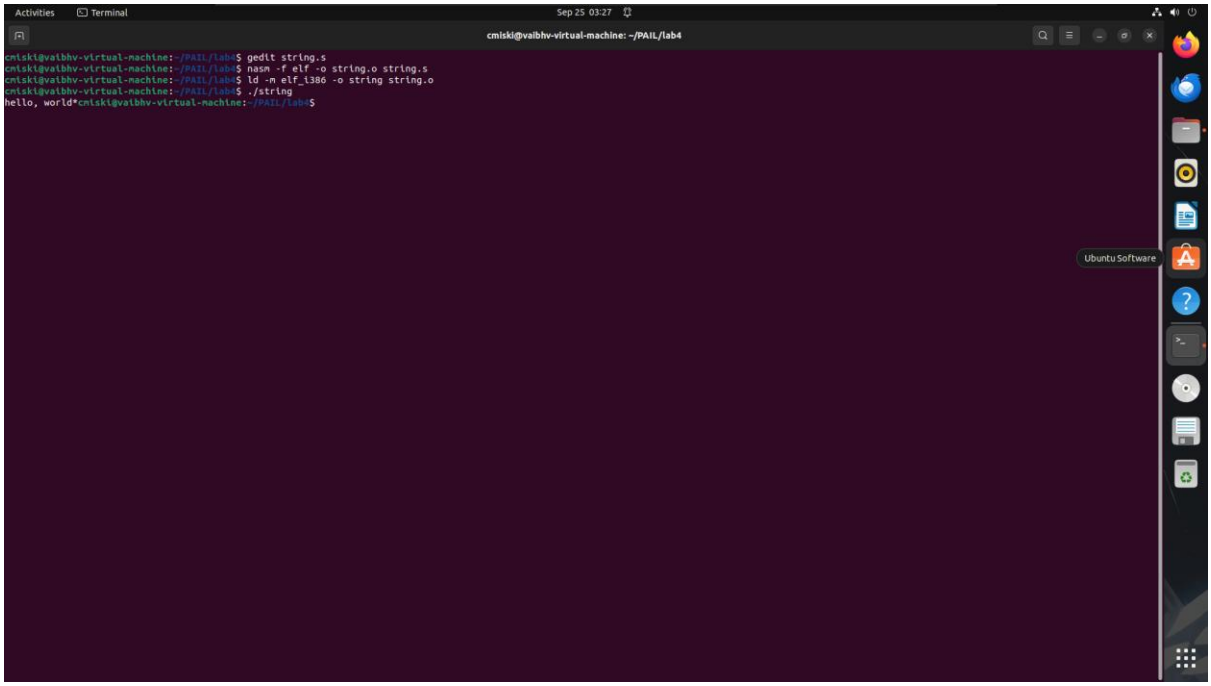
B] String Operation

Code:

```
global _start


section .text
```

```
_start:
    cld
    mov ecx, len
    mov esi, s1
    mov edi, s2

loop_here:
    lodsb
    or al, 20h
    stosb
    loop loop_here

    ; print result
    mov edx, len
    mov ecx, s2
    mov ebx, 1
    mov eax, 4
    int 0x80

    ; exit
    mov eax, 1
    xor ebx, ebx
    int 0x80

section .data
s1 db 'HELLO, WORLD', 0xa
len equ $-s1

section .bss
s2 resb len
```

Output:


Activities
Terminal
Sep 25 03:27
cmlski@vaibhv-virtual-machine: ~/PAIL/lab4
cmlski@vaibhv-virtual-machine:~/PAIL/lab4$ gedit string.s
cmlski@vaibhv-virtual-machine:~/PAIL/lab4$ nasm -f elf -o string.o string.s
cmlski@vaibhv-virtual-machine:~/PAIL/lab4$ ld -m elf_i386 -o string string.o
cmlski@vaibhv-virtual-machine:~/PAIL/lab4$ ./string
hello, world*cmlski@vaibhv-virtual-machine:~/PAIL/lab4$
Ubuntu Software