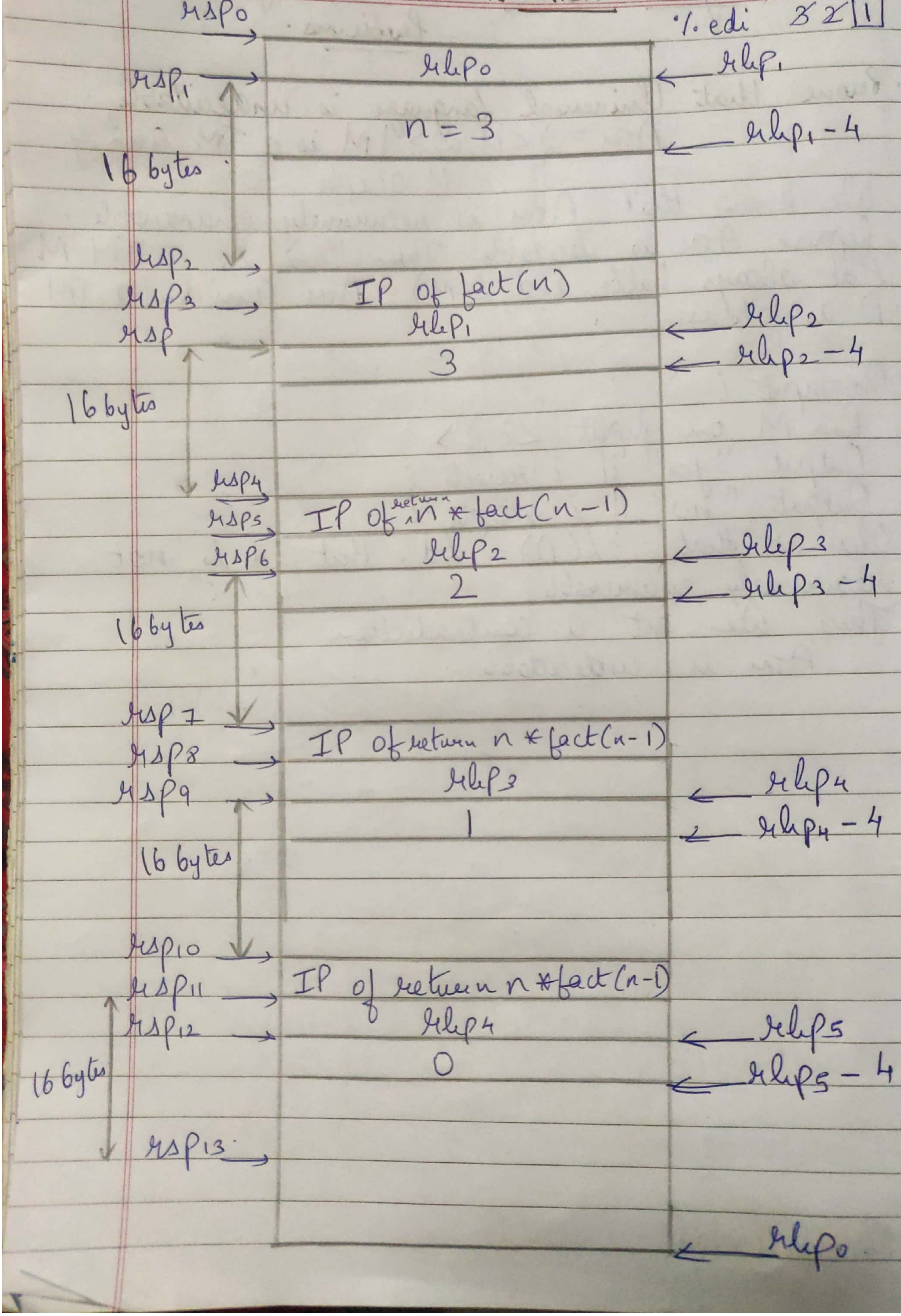
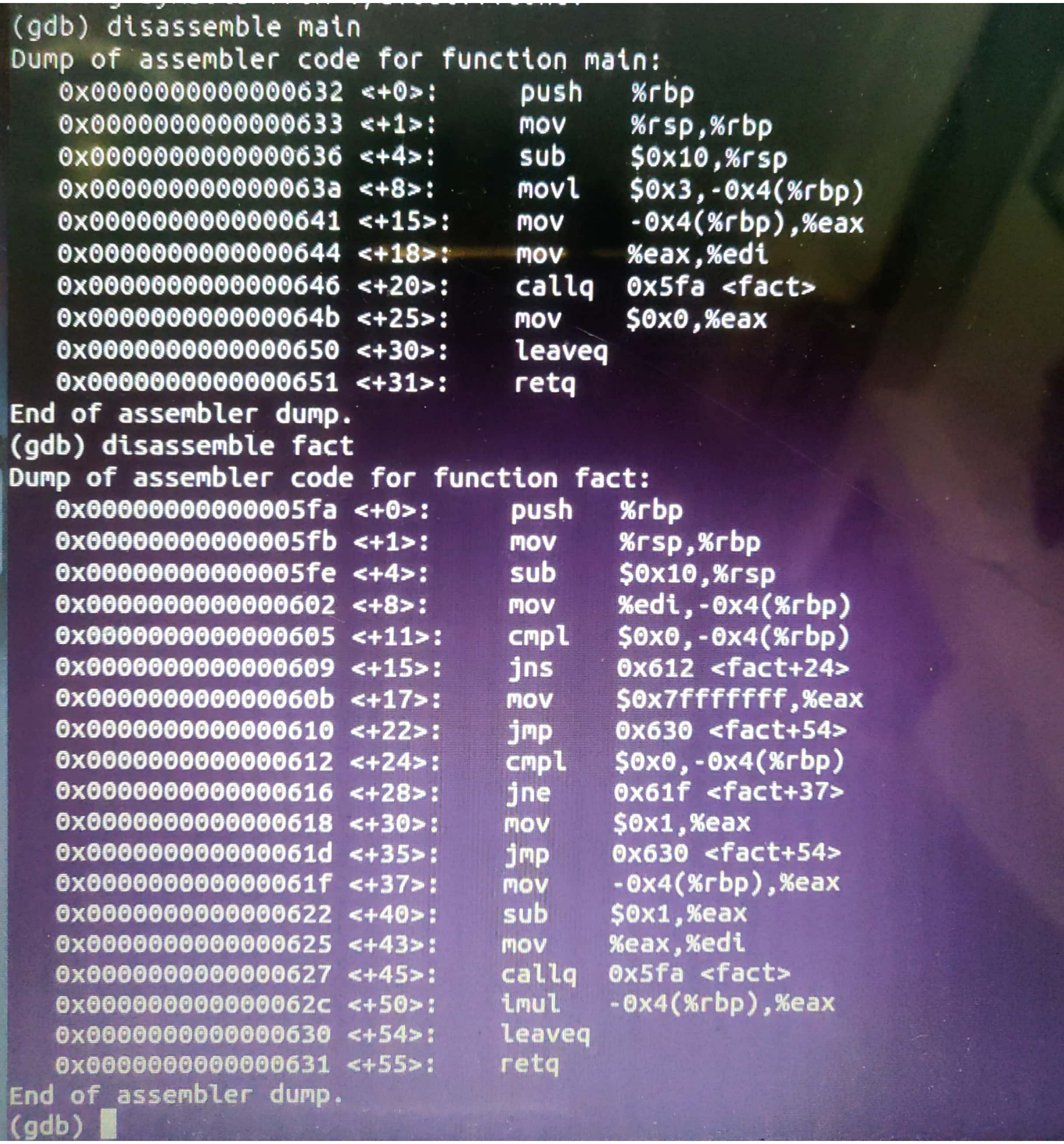
STACK DIAGRAM OF A RECURSIVE PROGRAM

%eax ~~1~~ ~~2~~ $\boxed{6}$

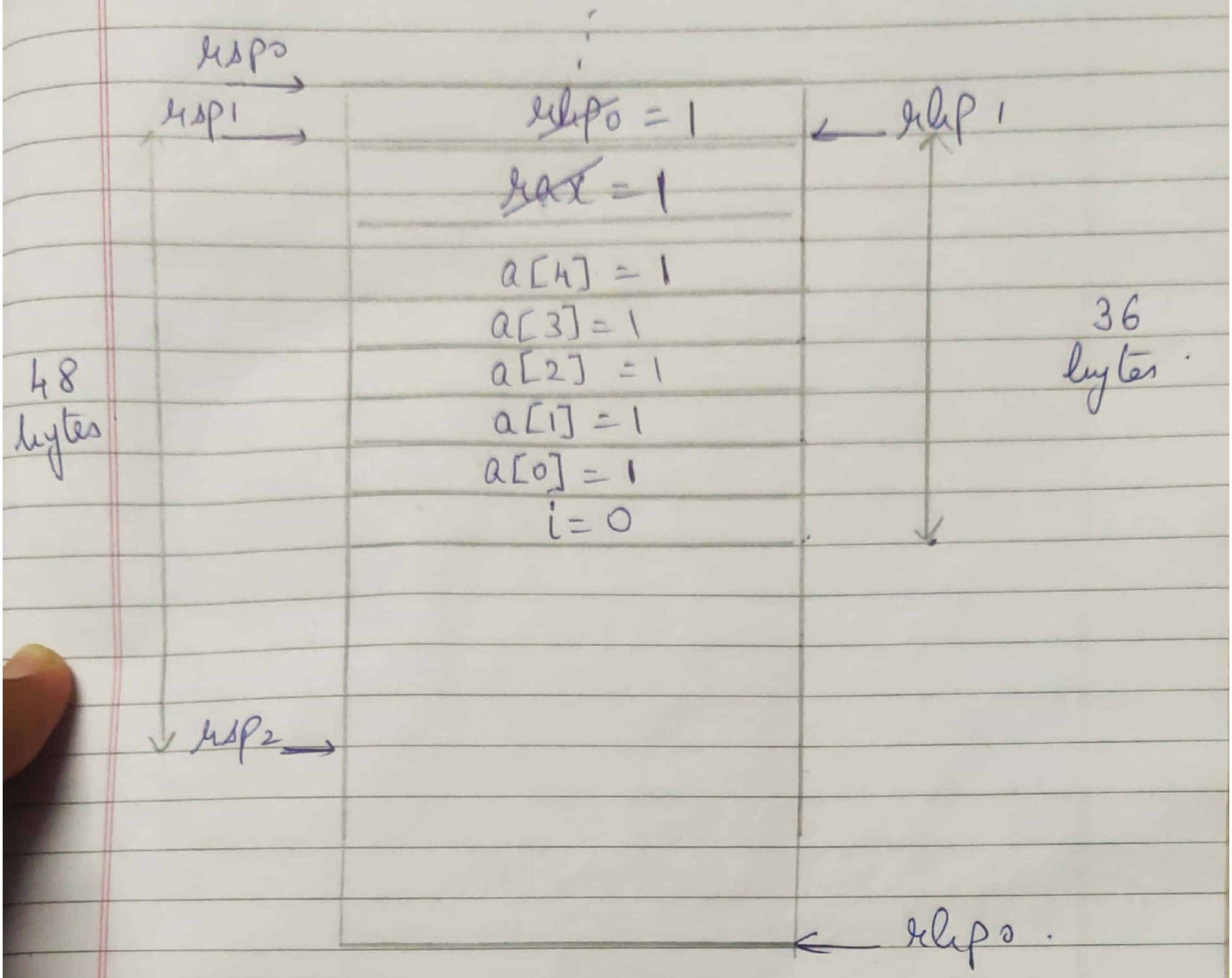
%edi ~~3~~ ~~2~~ $\boxed{1}$





```
(gdb) disassemble main
Dump of assembler code for function main:
   0x0000000000000632 <+0>:     push   %rbp
   0x0000000000000633 <+1>:     mov    %rsp,%rbp
   0x0000000000000636 <+4>:     sub    $0x10,%rsp
   0x000000000000063a <+8>:     movl   $0x3,-0x4(%rbp)
   0x0000000000000641 <+15>:    mov    -0x4(%rbp),%eax
   0x0000000000000644 <+18>:    mov    %eax,%edi
   0x0000000000000646 <+20>:    callq  0x5fa <fact>
   0x000000000000064b <+25>:    mov    $0x0,%eax
   0x0000000000000650 <+30>:    leaveq
   0x0000000000000651 <+31>:    retq
End of assembler dump.
(gdb) disassemble fact
Dump of assembler code for function fact:
   0x00000000000005fa <+0>:     push   %rbp
   0x00000000000005fb <+1>:     mov    %rsp,%rbp
   0x00000000000005fe <+4>:     sub    $0x10,%rsp
   0x0000000000000602 <+8>:     mov    %edi,-0x4(%rbp)
   0x0000000000000605 <+11>:    cmpl   $0x0,-0x4(%rbp)
   0x0000000000000609 <+15>:    jns    0x612 <fact+24>
   0x000000000000060b <+17>:    mov    $0x7fffffff,%eax
   0x0000000000000610 <+22>:    jmp    0x630 <fact+54>
   0x0000000000000612 <+24>:    cmpl   $0x0,-0x4(%rbp)
   0x0000000000000616 <+28>:    jne    0x61f <fact+37>
   0x0000000000000618 <+30>:    mov    $0x1,%eax
   0x000000000000061d <+35>:    jmp    0x630 <fact+54>
   0x000000000000061f <+37>:    mov    -0x4(%rbp),%eax
   0x0000000000000622 <+40>:    sub    $0x1,%eax
   0x0000000000000625 <+43>:    mov    %eax,%edi
   0x0000000000000627 <+45>:    callq  0x5fa <fact>
   0x000000000000062c <+50>:    imul   -0x4(%rbp),%eax
   0x0000000000000630 <+54>:    leaveq
   0x0000000000000631 <+55>:    retq
End of assembler dump.
(gdb) 
```

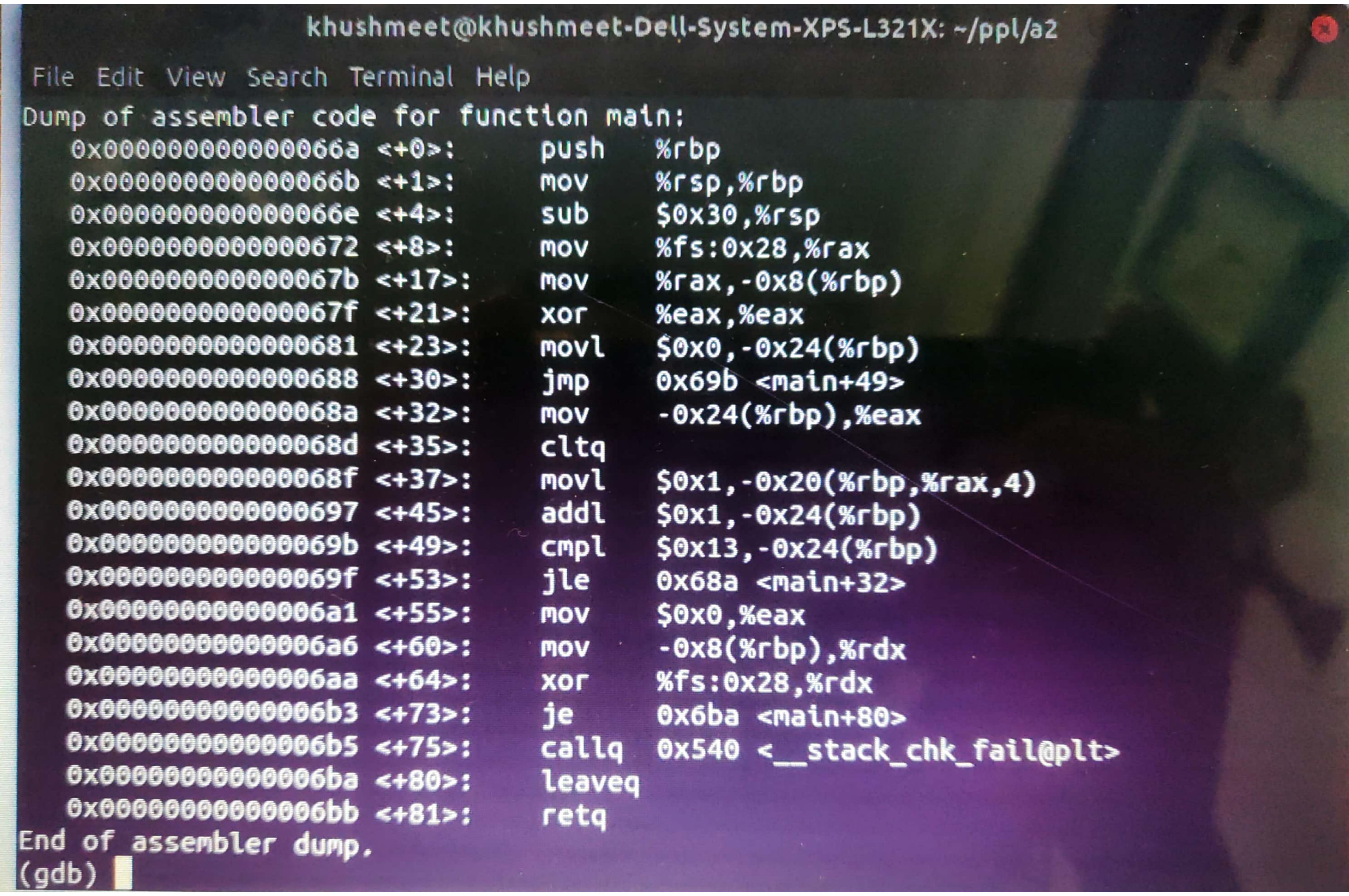
STACK DIAGRAM OF STACK SMASHING.



Address Values are Overwritten

After the While loop ends, stack smashing is detected.

rsp and rbp jump to unknown locations



khushmeet@khushmeet-Dell-System-XPS-L321X: ~/ppl/a2
File Edit View Search Terminal Help

```
Dump of assembler code for function main:
   0x000000000000066a <+0>:     push   %rbp
   0x000000000000066b <+1>:     mov    %rsp,%rbp
   0x000000000000066e <+4>:     sub    $0x30,%rsp
   0x0000000000000672 <+8>:     mov    %fs:0x28,%rax
   0x000000000000067b <+17>:    mov    %rax,-0x8(%rbp)
   0x000000000000067f <+21>:    xor    %eax,%eax
   0x0000000000000681 <+23>:    movl   $0x0,-0x24(%rbp)
   0x0000000000000688 <+30>:    jmp    0x69b <main+49>
   0x000000000000068a <+32>:    mov    -0x24(%rbp),%eax
   0x000000000000068d <+35>:    cltq
   0x000000000000068f <+37>:    movl   $0x1,-0x20(%rbp,%rax,4)
   0x0000000000000697 <+45>:    addl   $0x1,-0x24(%rbp)
   0x000000000000069b <+49>:    cmpl   $0x13,-0x24(%rbp)
   0x000000000000069f <+53>:    jle    0x68a <main+32>
   0x00000000000006a1 <+55>:    mov    $0x0,%eax
   0x00000000000006a6 <+60>:    mov    -0x8(%rbp),%rdx
   0x00000000000006aa <+64>:    xor    %fs:0x28,%rdx
   0x00000000000006b3 <+73>:    je     0x6ba <main+80>
   0x00000000000006b5 <+75>:    callq  0x540 <__stack_chk_fail@plt>
   0x00000000000006ba <+80>:    leaveq
   0x00000000000006bb <+81>:    retq
End of assembler dump.
(gdb) 
```