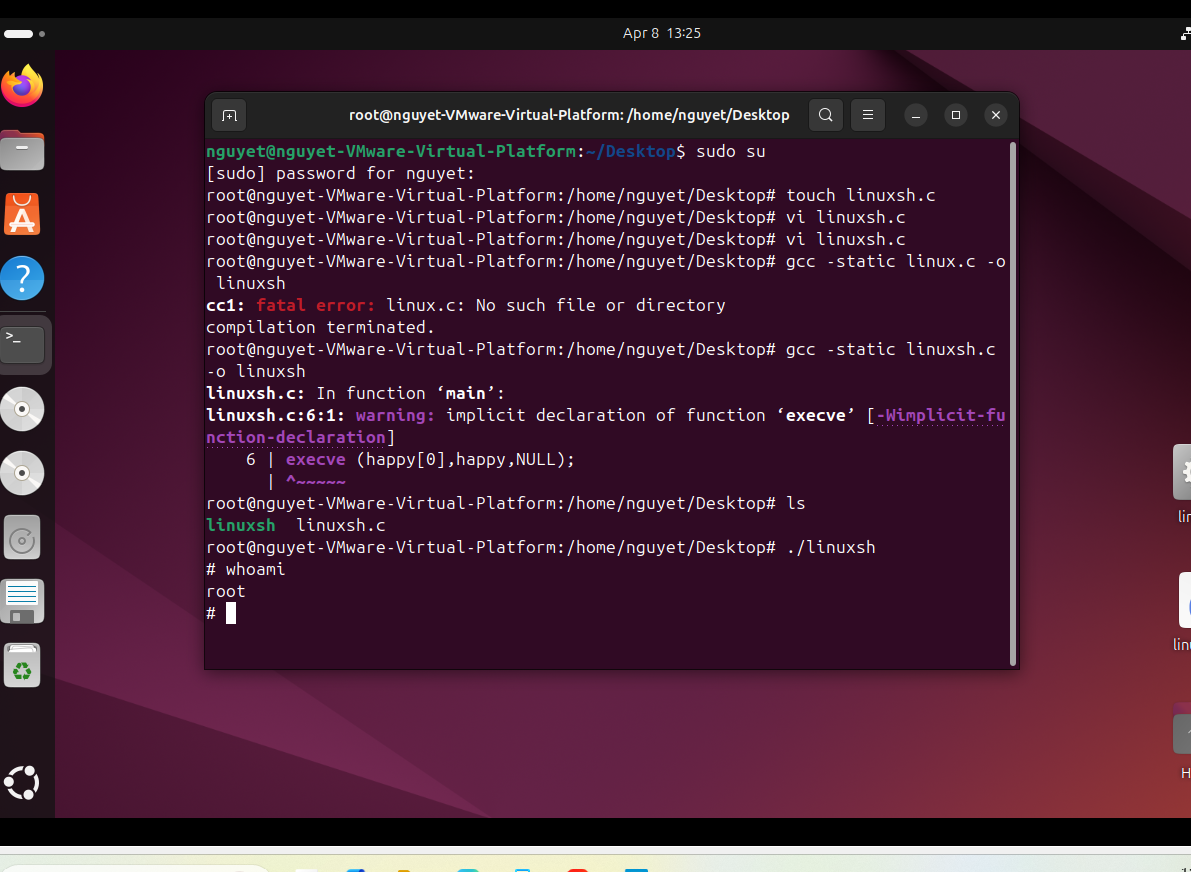
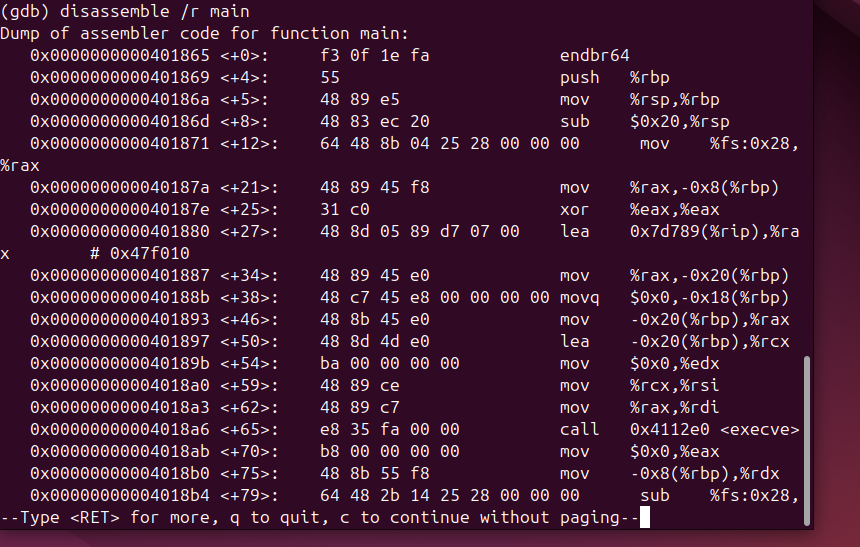
* Bước 2:



* Bước 3: Dùng gdb để lấy mã asemmbly  
  #gdb ./linuxsh  
  gdb>disas /r main



- Cột đầu tiên: Địa chỉ bộ nhớ

- Cột thứ 2: Opcode dạng hex (mã máy)

- Cột thứ 3 trở đi: Mã Assembly tương ứng

* Bước 4: So sánh với mã assembly sau:  
  ;execve("/bin//sh", 0, 0) for linux/x86-64  
  section .text  
    global \_start  
      \_start:  
        push rax                               ; rax on stack  
        xor rdx, rdx                ; rdx= NULL  
        xor rsi, rsi                ; rsi=NULL  
        mov rbx,'/bin//sh'            ;rbx="bin//sh"  
        push rbx                ; first on stack  
        push rsp                ; second on stack  
        pop rdi                ; rdi="/bin//sh",0  
        mov al, 59                      
        syscall

- Đặt rdi là “/bin//sh”:

+ shellcode thủ công: push, pop, mov

+ Mã GDB sinh ra từ C: mov, lea

- Đặt rsi, rdx = NULL:

+ Shellcode thủ công: xor

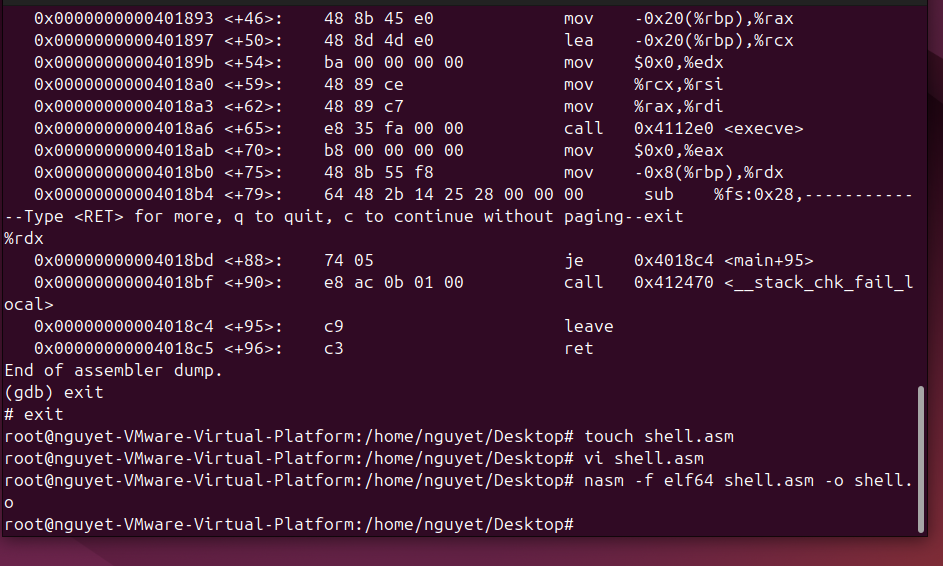
+ Mã GDB sinh ra từ C: mov, xor

- Gọi syscall hoặc execve:

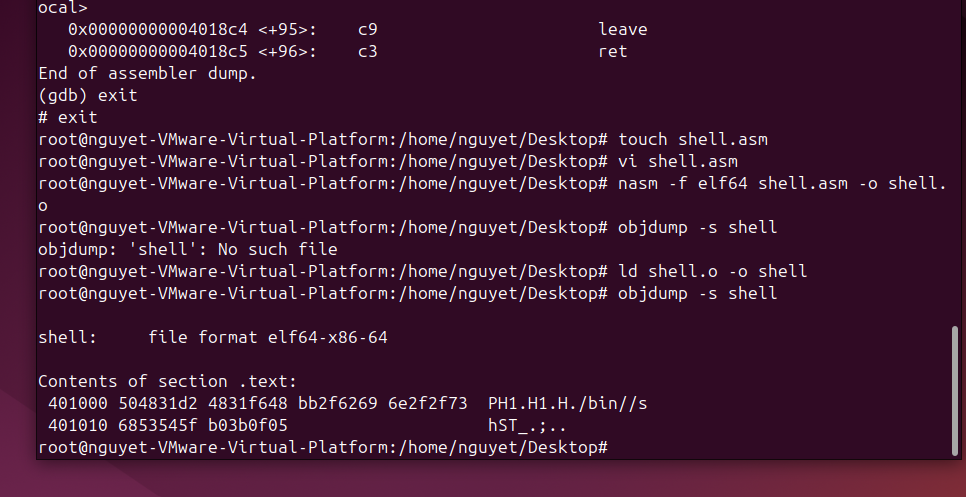
+ Shellcodethủ công: syscall

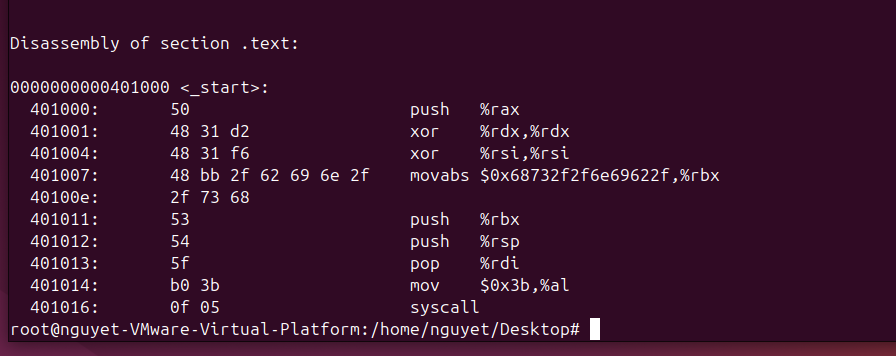
+ Mã GDB sinh ra từ C: call execve

* Bước 5: Biên dịch bằng nasm:



* Bước 6: Dùng objdump để lấy mã máy (shellcode)  
  objdump –d shell

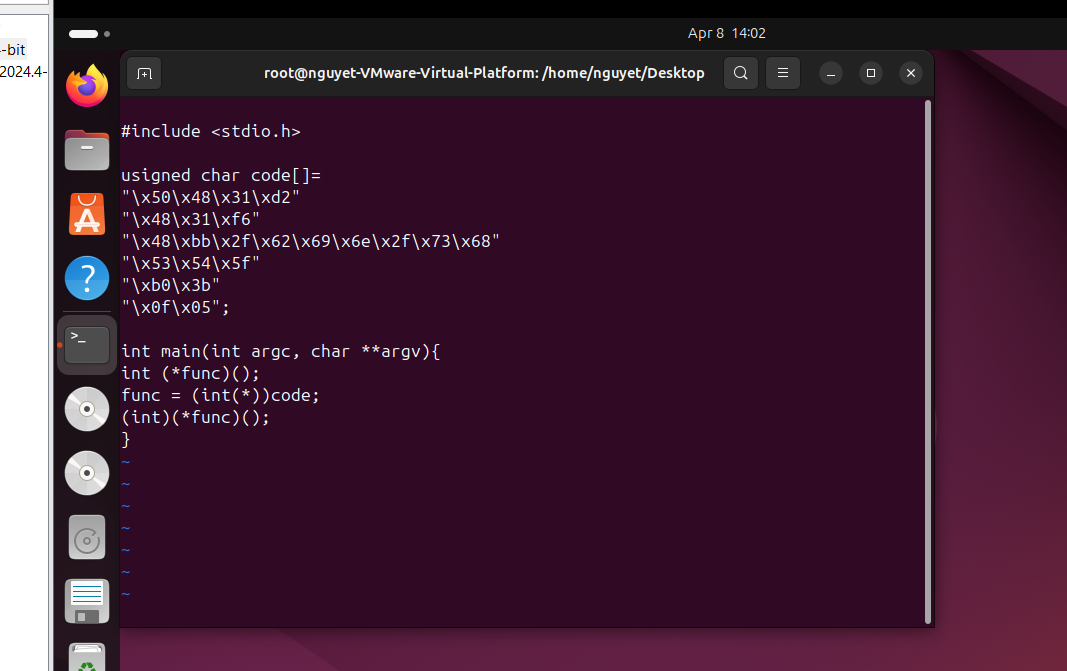




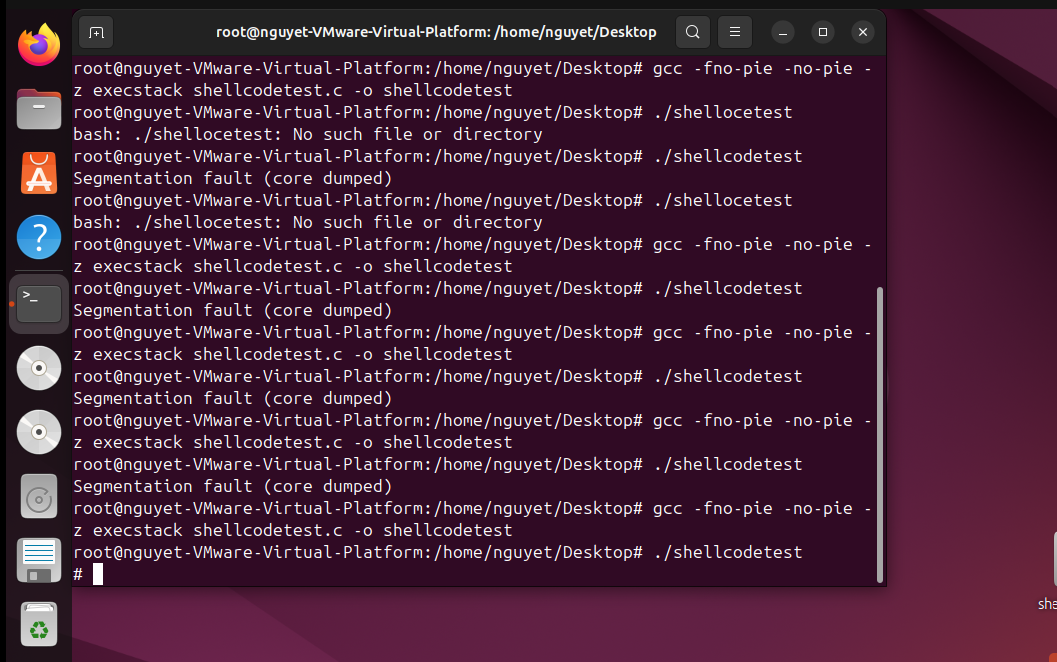
- Mã máy: 50 48 31 d2 48 31 f6 48 bb 2f 62 69 6e 2f 2f 73 68 53 54 5f b0 3b 0f 05

- Mã máy trên đươcj viết dưới dạng hex

* Bước 7: lưu thành shellcodetest.c



* Bước 8: Biên dịch và thực thi:



- Phát sinh shell thành công

- fno-pie -no-pie: tắt tính năng ASLR (địa chỉ base cố định) giúp shellcode hoạt động ổn định

- Dòng # cuối cùng cho thấy shellocode đã phát sinh thành công

- Trong trường hợp cho ra kết quả Segementation fault: nghĩa là shellcode chưa phát sinh thành công. Có thể là do mã máy chưa đúng.