HW4

姓名:黃偉城　　學號:409510095

問題 1:stdin_read 程式碼

```
#include <stdio.h>
#include <unistd.h>

int main(int argc, char **argv){

        char ch ;

        __asm__ volatile(

                "mov $0, %%rax\n" //syscall no.0
                "mov $0, %%rdi\n" // fd
                "mov %0, %%rsi\n" // buf
                "mov $1, %%rdx\n"
                "syscall\n"
                "mov %%rax, %0"
                :
                : "g"(&ch)
                : "rax", "rdi", "rsi", "rdx");

        printf("讀入的字元為'%c'\n", ch);
        return 0 ;

}
```

執行結果:

```
eric0917579@LAPTOP-LASAUQ2T:~/aaa/OS/hw4$ ./stdin_read
a
讀入的字元為'a'
eric0917579@LAPTOP-LASAUQ2T:~/aaa/OS/hw4$
eric0917579@LAPTOP-LASAUQ2T:~/aaa/OS/hw4$
```

問題二:反組譯

```
4       int main(int argc, char **argv){
   0x0000000000401cb5 <+0>:     endbr64
   0x0000000000401cb9 <+4>:     push   %rbp
   0x0000000000401cba <+5>:     mov    %rsp,%rbp
   0x0000000000401cbd <+8>:     sub    $0x20,%rsp
   0x0000000000401cc1 <+12>:    mov    %edi,-0x14(%rbp)
   0x0000000000401cc4 <+15>:    mov    %rsi,-0x20(%rbp)
   0x0000000000401cc8 <+19>:    mov    %fs:0x28,%rax
   0x0000000000401cd1 <+28>:    mov    %rax,-0x8(%rbp)
   0x0000000000401cd5 <+32>:    xor    %eax,%eax
```

這部分是 main 函數的開始。將 edi 和 esi 的值保存到-0x14(%rbp)和-0x20(%rbp)，並將%fs:0x28 段暫存器的值存到-0x8(%rbp)。

```
5
6               char ch ;
7
8               __asm__ volatile(
   0x0000000000401cd7 <+34>:    lea    -0x9(%rbp),%rcx
   0x0000000000401cdb <+38>:    mov    $0x0,%rax
   0x0000000000401ce2 <+45>:    mov    $0x0,%rdi
   0x0000000000401ce9 <+52>:    mov    %rcx,%rsi
   0x0000000000401cec <+55>:    mov    $0x1,%rdx
   0x0000000000401cf3 <+62>:    syscall
   0x0000000000401cf5 <+64>:    mov    %rax,%rcx
```

這一部分包含使用__asm__ volatile 指令。它使用 syscall 指令執行從標準輸入中讀取一個字符並將其存在變量 ch 中。系统調用的结果存在%rcx 中。

```
20              printf("讀入的字元為'%c'\n", ch);
   0x0000000000401cf8 <+67>:    movzbl -0x9(%rbp),%eax
   0x0000000000401cfc <+71>:    movsbl %al,%eax
   0x0000000000401cff <+74>:    mov    %eax,%esi
   0x0000000000401d01 <+76>:    lea    0x932fc(%rip),%rdi        # 0x4950
   0x0000000000401d08 <+83>:    mov    $0x0,%eax
   0x0000000000401d0d <+88>:    callq  0x410990 <printf>
```

這一部分使用 printf 函数。它從-0x9(%rbp)加載字符到%eax