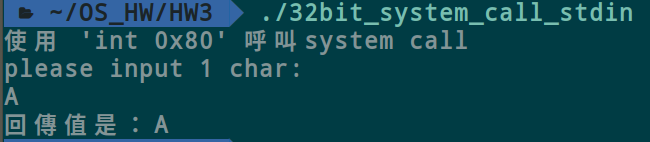
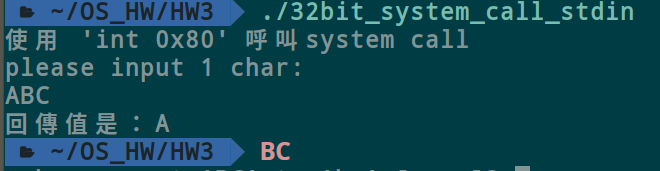
hw03.system call 406410114 資工三 郭晏誠

1. 使用組合語言呼叫system call，從stdin讀進一個字元

(1)32bit

(i)輸出結果：只吃一個字元

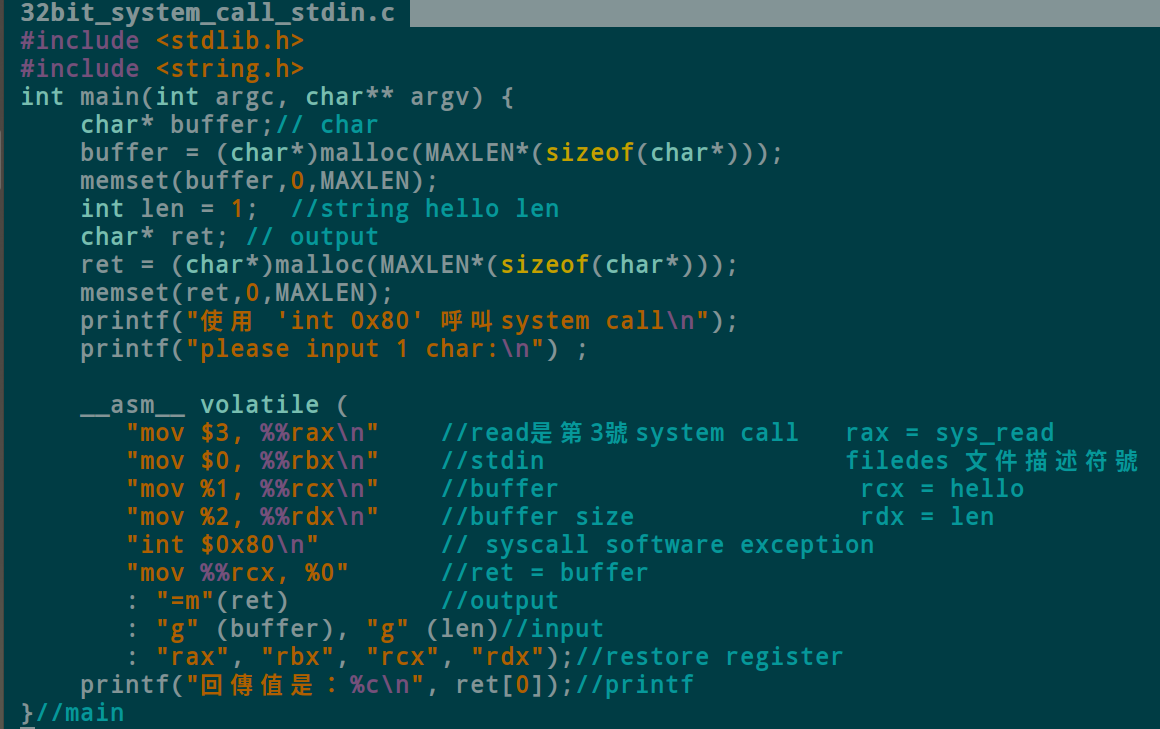




(ii)修改使得buffer和ret有指定的記憶體

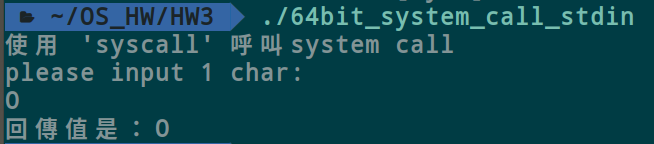
(iii)使用3號中斷 sys\_read、並把參數改成0代表stdin

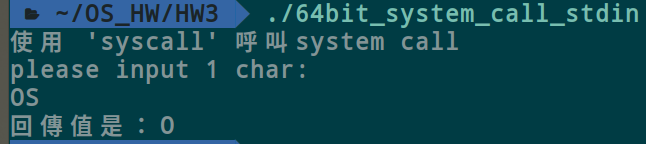
(iv)並且把buffer read進來的字串放到ret



(2)64bit

(i)輸出結果：只吃一個字元

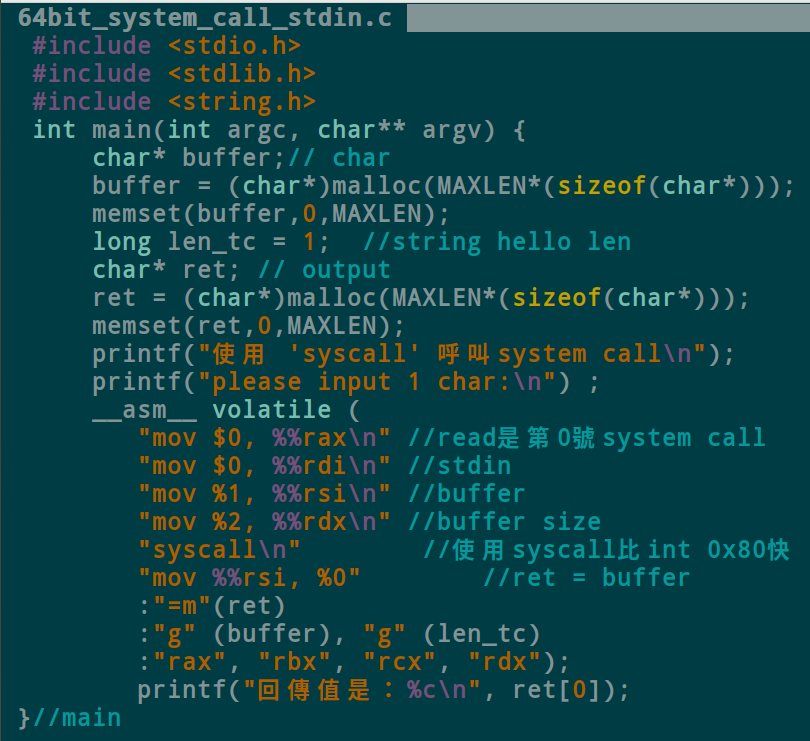




(ii)修改使得buffer和ret有指定的記憶體

(iii)使用3號中斷 sys\_read、並把參數改成0代表stdin

(iv)並且把buffer read進來的字串放到ret



//youtube筆記

32bit\_system\_call.c

#include <stdio.h>

#include <string.h>

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

char\* hello = "hello world\n";// char

int len = strlen(hello)+1; //string hello len

long ret; // output

printf("使用 'int 0x80' 呼叫system call\n");

\_\_asm\_\_ volatile (

"mov $4, %%rax\n" //write是第4號system call

"mov $2, %%rbx\n" //stderr filedes 文件描述符號

"mov %1, %%rcx\n" //buffer rcx = hello

"mov %2, %%rdx\n" //buffer size rdx = len

"int $0x80\n" //發出system call exception

//int $0x80將系統調用號傳入eax(this is 4 sys\_write)，

//各個參數按照ebx、ecx、edx的順序傳遞到寄存器中，系統調用返回值儲存到eax寄存器。

//ssize\_t write(int fd, const void \*buf, size\_t count);

"mov %%rax, %0"

//system call的回傳值放在rax ret = sys\_write\_return

: "=m"(ret)

: "g" (hello), "g" (len)

: "rax", "rbx", "rcx", "rdx");

printf("回傳值是：%ld\n", ret);

}//main

64bit\_system\_call.c

#include <stdio.h>

#include <string.h>

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

char\* hello\_tc = "全世界，你好\n";

long len\_tc = strlen(hello\_tc)+1; //注意我宣告為long，因為long是64位元

long ret;

printf("使用 'syscall' 呼叫system call\n");

\_\_asm\_\_ volatile (

"mov $1, %%rax\n" //write是第1號system call

"mov $2, %%rdi\n" //stderr register用法不一樣

"mov %1, %%rsi\n" //buffer register用法不一樣

"mov %2, %%rdx\n" //buffer size

"syscall\n" //使用syscall比int 0x80快 AMD提出

"mov %%rax, %0" //system call的回傳值依然放在AX

:"=m"(ret)

:"g" (hello\_tc), "g" (len\_tc)

:"rax", "rbx", "rcx", "rdx");

printf("回傳值是：%ld\n", ret);

}//main