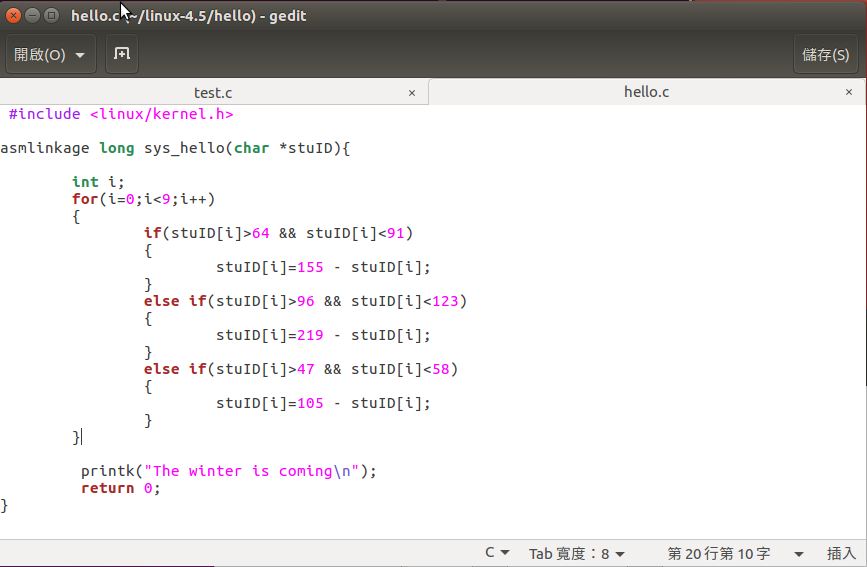
「CSC061作業系統」程式作業一

|  |  |
| --- | --- |
| 學號：A1035507 | 姓名：吳希炫 |

|  |  |
| --- | --- |
| 聲明與自評 | 項目 |
| 1..[獨立進行] | 榮譽聲明：  1.[獨立進行]本作業獨立進行，沒有任何抄襲行為，本作業有提供他人參考如列。  2.[共同討論]本作業與他人討論後進行，沒有任何抄襲行為，共同討論人如列。  3.[部份參考]本作業有參考他人版本進行，被參考者如列。  4.[部份抄襲]本作業有抄襲他人版本或資料進行，被抄襲者或資料如列。 |
| 3. Linux | 使用平台：  1. Android x86  2. Raspberry Pi  3. Linux |
| 1.完全符合 | 完成聲明：作業是否達到要求(請在執行結果處說明完成的程度)  1.完全符合  2.部份符合  3.未完成 |

**\* 新增一個System Call**

1. 作業進行過程與執行結果 (含畫面與說明)

1. 下載kernel source(4.5.0版本)並解壓縮
2. 進入剛解壓縮的資料夾，創建資料夾並撰寫system call的程式
3. 在同資料夾創立Makefile文件並在裡面加入code

obj-y := 資料夾名.o

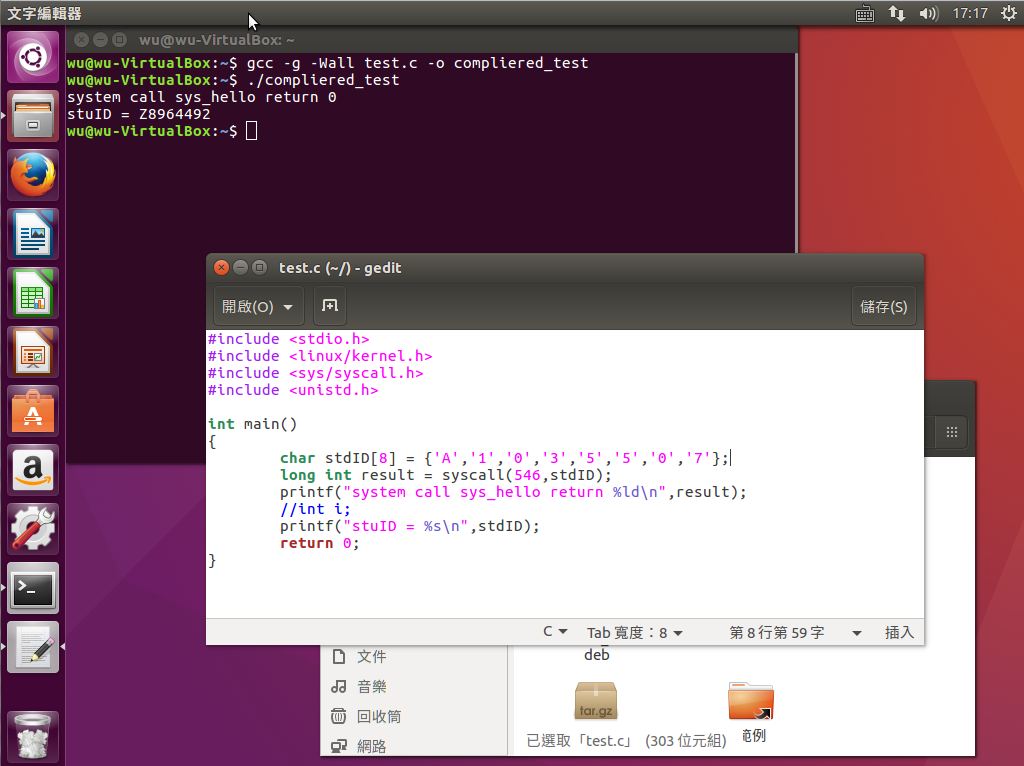
1. 回到kernel的根目錄，在Makefile中找到

core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/

改成

core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/ 資料夾名/

1. 新增systemcall到system table
2. 新增systemcall到system call的header
3. 編譯
4. 非常非常久之後，安裝編譯好的deb檔
5. Reboot
6. 撰寫用到新systemcall的C程式
7. 執行



2. 心得(300字)

雖然網路上很多教學了，但是實際上做會遇到蠻多困難的，像是一開始需要安裝很多套件，才能夠順利的編譯，而這些套件又因為目前kernel的不同而安裝不同的套件，因此我看了很多網路教學，才得以使編譯順利進行。而我要安裝kernel 4.5.0版時，我發現如果目前電腦的kernel版本比要安裝的kernel版本還高時，系統也會默認使用目前的kernel，需要將目前的kernel移除，網路上也很少人遇到這個問題，只能自己摸索出來。另外，linux的作業系統之不同位元也影響操作的步驟，原本在這部分卡住很久，後來才發現我使用64位元的系統，但卻把systemcall加入到32位元的檔案。

經過這次的作業，我也更加熟悉linux的作業系統，包括了解linux的操作方式、了解systemcall如何運作、使用虛擬機等等，雖然老師指定的android x86沒有時間再去挑戰了，但是這次的作業收穫也還蠻多的，也把課本的知識與實作連結在一起，讓記憶變加深刻。

3. 程式原始碼

**Systemcall**

#include <linux/kernel.h>

asmlinkage long sys\_hello(char \*stuID){

int i;

for(i=0;i<9;i++)

{

if(stuID[i]>64 && stuID[i]<91)

{

stuID[i]=155 - stuID[i];

}

else if(stuID[i]>96 && stuID[i]<123)

{

stuID[i]=219 - stuID[i];

}

else if(stuID[i]>47 && stuID[i]<58)

{

stuID[i]=105 - stuID[i];

}

}

printk("The winter is coming\n");

return 0;

}

**Test**

#include <stdio.h>

#include <linux/kernel.h>

#include <sys/syscall.h>

#include <unistd.h>

int main()

{

char stdID[8] = {'A','1','0','3','5','5','0','7'};

long int result = syscall(546,stdID);

printf("system call sys\_hello return %ld\n",result);

printf("stuID = %s\n",stdID);

return 0;

}