微算機系統實習

實驗6

日期:2021/06/04

1. 組員姓名：

第一組

資工二　108590044　何柏憲

資工二　108590452　林峻霆

1. 實驗步驟截圖與說明：

實驗目標:

• 學習semaphore and mutex

• 撰寫程式以pthread 控制led閃爍  
• 利用Mulitithread 開發LED同步控制功能  
• 在TX2執行

實驗過程:

1.首先我們先參考範例了解此次實驗要做什麼，發現是延伸lab2 的實驗再加上實作semaphore and mutex

2.因為這次是遠距教學，第一週我們組員有見面一起討論此實驗，第二週則是線上meet

3.第二週在meeting 前我們都有彼此先查查月資料和了結題目所以第二週在做的時候非常的順利。

4. 這次我們寫了兩個main，一個給semaphore 一個給mutex然後分別執行他們

5.錄影並執行課程要求的指令

以下是我們的程式碼：

Makefile:

semaphore:

gcc main.c -lpthread -o main

sudo ./main 0111 5

mutex:

gcc main2.c -lpthread -o main2

sudo ./main2 1001 5

main.h

#include <stdio.h>

#include <stdlib.h>

int gpio\_export(unsigned int gpio){

int len;

char buff[64];

FILE \*fp ;

fp = fopen("/sys/class/gpio/export","w");

if (fp == NULL){

perror("gpio/export");

return fp;

}

len = snprintf(buff,sizeof(buff),"%d",gpio);

fwrite(buff,1,sizeof(buff),fp);

fclose(fp);

return 0;

}

int gpio\_unexport(unsigned int gpio){

int len;

char buff[64];

FILE \*fp = fopen("/sys/class/gpio/unexport","w");

if (fp==NULL){

perror("gpio/unexport");

return fp;

}

len = snprintf(buff,sizeof(buff),"%d",gpio);

fwrite(buff,sizeof(buff),1,fp);

fclose(fp);

return 0;

}

int gpio\_set\_dir(unsigned int gpio,char \*dirStatus){

char buff[64];

snprintf(buff,sizeof(buff),"/sys/class/gpio/gpio%d/direction",gpio);

FILE \*fp = fopen(buff,"w");

if(fp==NULL){

perror("gpio/direction");

return fp;

}

if( strcmp(dirStatus,"out") == 0){

fwrite(dirStatus,sizeof(dirStatus),1,fp);

}else{

char s[]="in";

fwrite(s,sizeof(s),1,fp);

}

fclose(fp);

return 0;

}

int gpio\_set\_value(unsigned int gpio,int value){

char buff[64];

snprintf(buff,sizeof(buff),"/sys/class/gpio/gpio%d/value",gpio);

FILE \*fp=fopen(buff,"w");

if(fp==NULL){

//perror("gpio/set-value");

return fp;

}

if(value == 0 ){

char s[] = "0";

fwrite(s,sizeof(s),1,fp);

}else{

char s[] = "1";

fwrite(s,sizeof(s),1,fp);

}

fclose(fp);

return 0;

}

main.c

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <unistd.h>

#include <semaphore.h>

#include "main.h"

sem\_t semaphere;

int pin[] = {396,397,255,481};

int LEDstatus[4]={0};

void turnOn(unsigned int target){

char s[] = "out";

gpio\_export(target);

gpio\_set\_dir(target,"out");

gpio\_set\_value(target,1);

}

void turnOff(unsigned int target){

gpio\_set\_value(target,0);

gpio\_unexport(target);

}

void \*child(void \*arg){

sem\_wait(&semaphere);

for(int j = 0 ; j < 4 ; j++){

if(LEDstatus[j] == 1){

turnOn(pin[j]);

}else{

turnOff(pin[j]);

}

}

sleep(1);

for(int j = 0 ; j < 4 ; j++){

if(LEDstatus[j] == 0){

turnOn(pin[j]);

}else{

turnOff(pin[j]);

}

}

sleep(1);

sem\_post(&semaphere);

pthread\_exit(NULL);

}

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

int times = atoi(argv[2]);

printf("times: %d\n",times);

for(int i = 0 ; i < 4;i++){

LEDstatus[i]= argv[1][i]-'0';

turnOn(pin[i]);

turnOff(pin[i]);

}

sem\_init(&semaphere,0,1);

pthread\_t t[times];

for(int i = 0 ; i < times ;i++){

printf("thread[%d] create\n",i);

pthread\_create(&t[i],NULL,child,(void\*)times);

}

for(int i = 0 ; i < times;i++){

pthread\_join(t[i],NULL);

}

return 0;

}

Main2.c

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <unistd.h>

#include <semaphore.h>

#include "main.h"

pthread\_mutex\_t mutex;

int counter=0;

int pin[] = {396,397,255,481};

int LEDStatus[4]={0};

void turnOn(unsigned int target){

char s[] = "out";

gpio\_export(target);

gpio\_set\_dir(target,"out");

gpio\_set\_value(target,1);

}

void turnOff(unsigned int target){

gpio\_set\_value(target,0);

gpio\_unexport(target);

}

void \*child(void \*arg){

pthread\_mutex\_lock(&mutex);

for(int j = 0 ; j < 4 ; j++){

if(LEDStatus[j] == 1){

turnOn(pin[j]);

}else{

turnOff(pin[j]);

}

}

sleep(1);

for(int j = 0 ; j < 4 ; j++){

if(LEDStatus[j] == 0){

turnOn(pin[j]);

}else{

turnOff(pin[j]);

}

}

sleep(1);

pthread\_mutex\_unlock(&mutex);;

pthread\_exit(NULL);

}

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

int times = atoi(argv[2]);

printf("times: %d\n",times);

for(int i = 0 ; i < 4;i++){

LEDStatus[i]= argv[1][i]-'0';

turnOn(pin[i]);

turnOff(pin[i]);

}

pthread\_mutex\_init(&mutex,0);

pthread\_t t[times];

for(int i = 0 ; i < times ;i++){

printf("thread[%d] create\n",i);

pthread\_create(&t[i],NULL,child,(void\*)times);

}

for(int i = 0 ; i < times;i++){

pthread\_join(t[i],NULL);

}

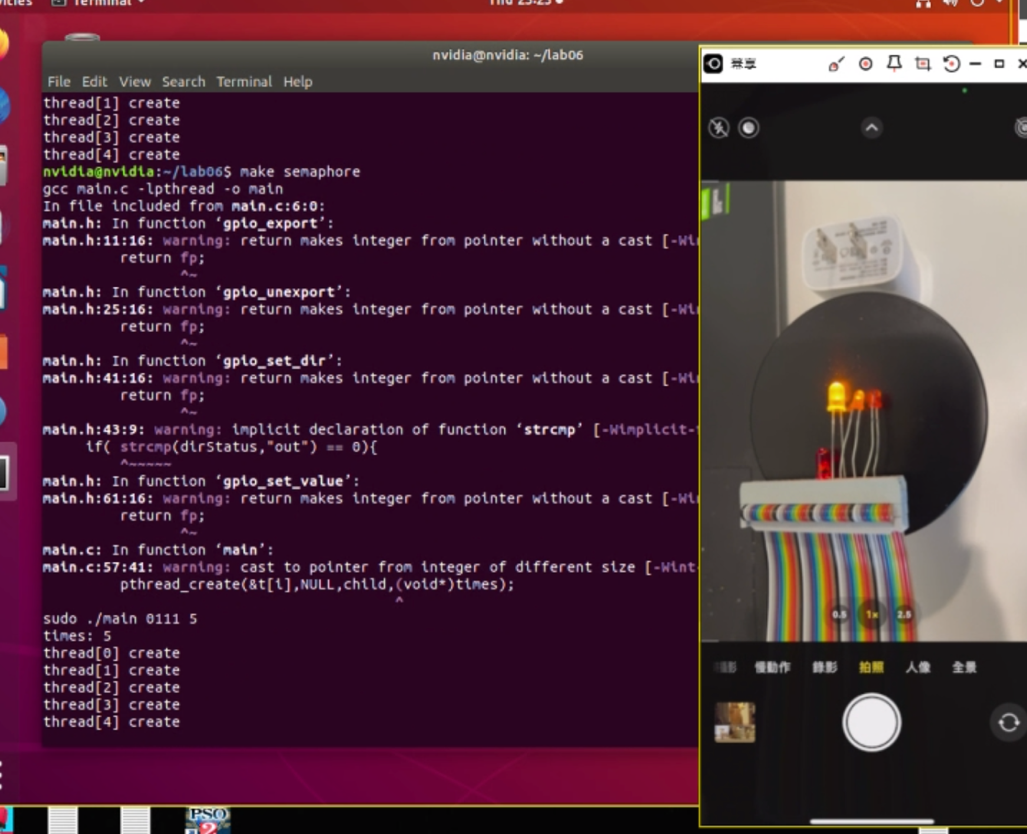
return 0;

}

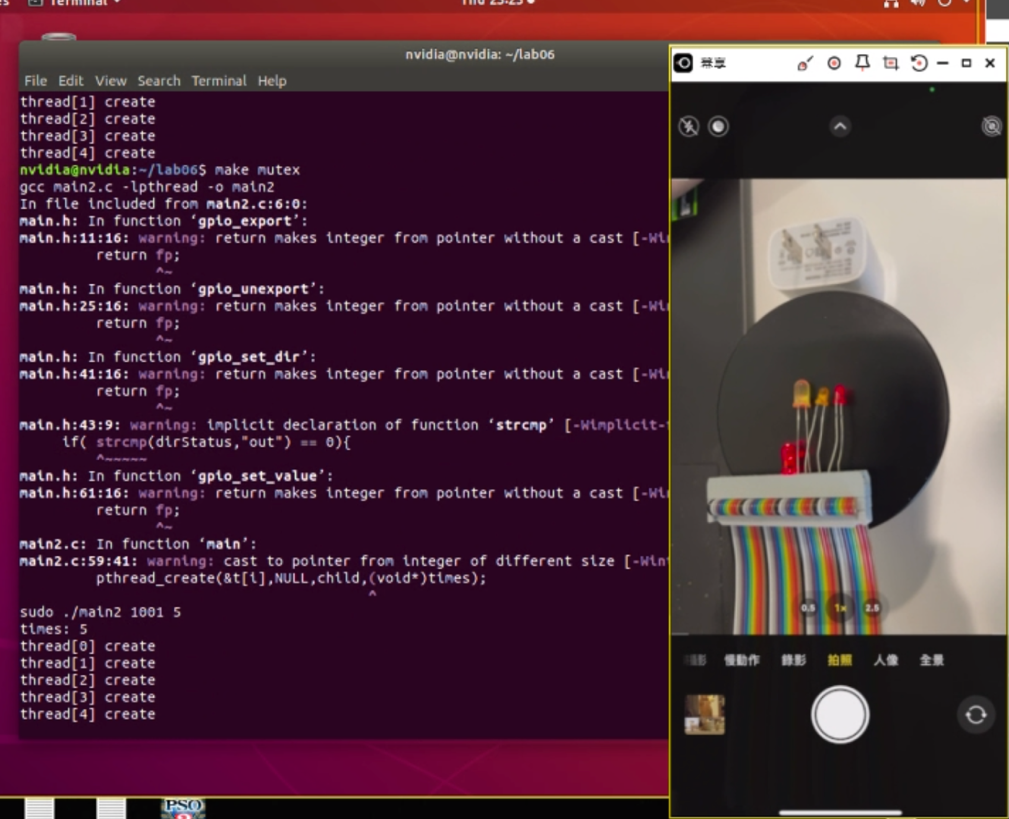
以下是實驗結果

螢幕和Led的結果：

Make semaphore



Make mutex



影片連結：

<https://www.youtube.com/watch?v=U5Agdk-dwTk>

1. 組員貢獻比例：

何柏憲：50%

林峻霆：50%

1. 心得：

何柏憲：

　　 這次實驗跟之前的內容差不多，延續了前次( lab2一開始 )的實驗內容，不過這次都是要在家裡自己做，少了助教的當場協助，做起來也花了更多的時間，但我們經過一番討論後，進度也做了起來，第二週我們一樣採線上meeting，助教和同學的回應給了我們提示，和團隊合作下最後順利的完成了。希望下次能夠在第一週完成進度第二週就能休息了，然後我就負責拍影片。

林峻霆：

這次實驗延續了前次( lab2一開始 )的實驗內容，我覺得遠距的微算機實習真的頗難的，第一週去組員家的時候我們先討論此次作業，回程的時候還下大雨，第二週我們一樣採線上meeting，助教和同學的反應給了我們不少的提示，最後也很順利的完成了，也搞懂了原理。希望下次能夠早點完成看看能不能在第一週完成進度第二週就能溫書了，然後我就負責做報告。