**#include** "stdio.h"

**#include** <pthread.h>

**#include** <stdlib.h>

**int** count =0;

**static** **void** \* **thread\_task**(void \*args)

{

    count++;

**printf**("thread %d is running\n", count);

}

**int** **main**()

{

    pthread\_t tid1, tid2;

**int** i1 = **pthread\_create**(&tid1, **NULL**, thread\_task, **NULL**);

**int** i2 = **pthread\_create**(&tid2, **NULL**, thread\_task, **NULL**);

**if** (i1 + i2 **<** 0)

    {

**printf**("pthread\_create error\n");

**return** -1;

    }

**if** (**pthread\_join**(tid1, **NULL**) **&&** **pthread\_join**(tid2, **NULL**))

    {

**printf**("thread is not exit...\n");

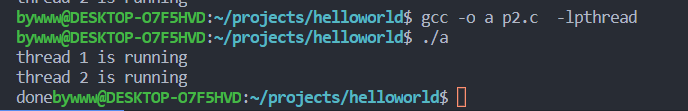
**return** -2;

    }

**printf**("done");

**return** 0;

}



**#include** "stdio.h"

**#include** <pthread.h>

**#include** <stdlib.h>

struct Args

{

**int** num;

**char** ch;

};

**void** \***thread\_task**(void \*args)

{

    struct Args\* b = (struct Args \*)args;

**printf**("thread %d  %c is running\n",b**->**num,b**->**ch);

}

**int** **main**()

{

    pthread\_t tid1, tid2;

    struct Args\* b;

    b = (struct Args \*)**malloc**(sizeof(struct Args));

    b**->**num = 10;

    b**->**ch = 'c';

**int** i1 = **pthread\_create**(&tid1, **NULL**, thread\_task, (**void** \*)b);

**int** i2 = **pthread\_create**(&tid2, **NULL**, thread\_task,(**void** \*)b);

**if** (i1 + i2 **<** 0)

    {

**printf**("pthread\_create error\n");

**return** -1;

    }

**if** (**pthread\_join**(tid1, **NULL**) **&&** **pthread\_join**(tid2, **NULL**))

    {

**printf**("thread is not exit...\n");

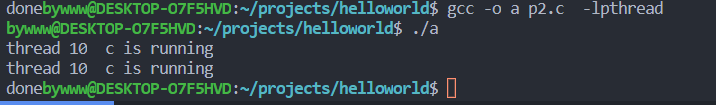
**return** -2;

    }

**printf**("done");

**return** 0;

}



**#include** "stdio.h"

**#include** <pthread.h>

**#include** <stdlib.h>

**#include** <semaphore.h>

**int** **less**(int a, int b, int flag)

{

**return** (**!**flag) **?** a **>** b **:** a **<** b;

}

**void** **bubble\_sort**(int arr**[]**, int len, int flag)

{

**int** i, j, temp;

**for** (i = 0; i **<** len - 1; i++)

**for** (j = 0; j **<** len - 1 - i; j++)

**if** (**less**(arr[j], arr[j + 1], flag))

            {

                temp = arr[j];

                arr[j] = arr[j + 1];

                arr[j + 1] = temp;

            }

}

sem\_t lock;

pthread\_t tid1, tid2;

**int** data[10] = {10, 9, 39, 230, 40, 1, 42, 32, 45, 450};

**int** len = 10;

**void** \***thread1\_task**(void \*args)

{

**sem\_wait**(&lock);

**bubble\_sort**(data, len, 0);

**for** (**int** i = 0; i **<** len; i++)

    {

**printf**("%d\t", data[i]);

    }

**printf**("\n");

**sem\_post**(&lock);

**pthread\_exit**(**NULL**);

}

**void** \***thread2\_task**(void \*args)

{

**sem\_wait**(&lock);

**bubble\_sort**(data, len, 1);

**for** (**int** i = 0; i **<** len; i++)

    {

**printf**("%d\t", data[i]);

    }

**printf**("\n");

**sem\_post**(&lock);

**pthread\_exit**(**NULL**);

}

**int** **main**()

{

**sem\_init**(&lock, 0, 1);

    tid1 = **pthread\_create**(&tid1, **NULL**, thread1\_task, **NULL**);

    tid2 = **pthread\_create**(&tid2, **NULL**, thread2\_task, **NULL**);

**if** (tid1 + tid2 **<** 0)

    {

**printf**("pthread\_create error\n");

**return** -1;

    }

**pthread\_join**(tid1, **NULL**);

**pthread\_join**(tid2, **NULL**);

**printf**("done\n");

**return** 0;

}

