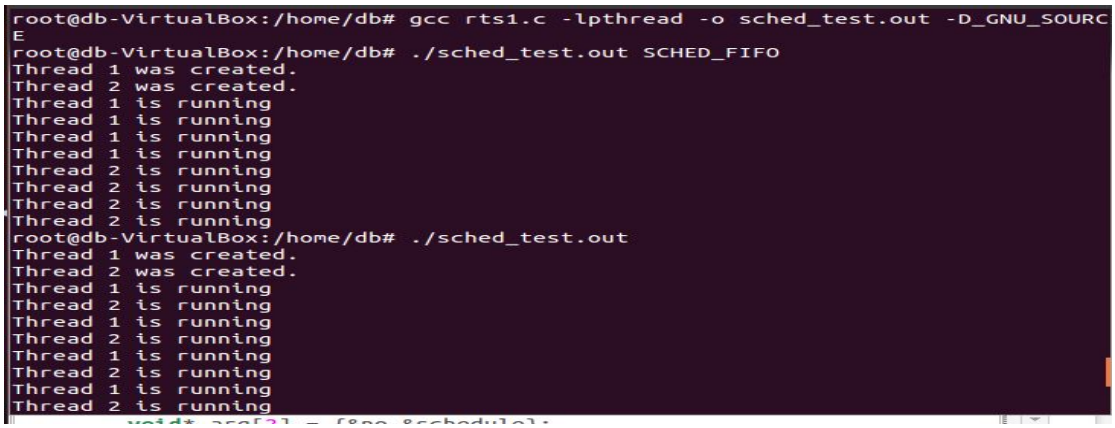
R06922134 葉沛陽 autumn11017@gmail.com



Hint

1.Int sched_setscheduler(pid_t pid, int policy, const struct sched_param *param);
將 PID 所指定的進程的調度策略和調度參數分別設置為 param 指向的 sched_param 結構中指定的 policy 和參數。

2.The policy corresponding value define in /include/linux/sched.h

```
s = sched_setaffinity(0, sizeof(mask), &mask);

struct sched_param param;
param.sched_priority = sched_get_priority_max(SCHED_FIFO)
s = sched_setscheduler(0, SCHED_FIFO, &param);
```

這些皆包含在 sched.h 裡面，必須 include 才能使用

3. Set the priority of real-time process (sched_param*param)

```
param.sched_priority = sched_get_priority_max(SCHED_FIFO);
s = sched_setscheduler(0, SCHED_FIFO, &param);
```

將策略設為 SCHED_FIFO，priority 設為 max(99)

4. The permission to run real-time process
需要 root 權限，否則會失敗。

5. CPU affinity

```
cpu_set_t mask;
CPU_ZERO(&mask);
CPU_SET(0, &mask);
s = sched_setaffinity(0, sizeof(mask), &mask);
```

讓 pthreads 都在固定的 CPU 跑，上圖為固定在 CPU 0 上。

Other problems

1.在 build kernel 時，遇到 Your display is too small to run Menuconfig!這個問題

```
Your display is too small to run Menuconfig!
It must be at least 19 lines by 80 columns.
make[1]: *** [menuconfig] Error 1
make: *** [menuconfig] Error 2
root@db-VirtualBox:/usr/src/linux-2.6.32.68#
```

試了網路上很多變大螢幕尺寸的方法，還是不行。

解決辦法: 安裝 guest additions cd (謝謝助教解答)

2. Pthread_create 如何帶兩個參數?因為要分辨是 thread1 or 2 以及是否有 SCHED_FIFO 需要有兩個參數進入 function。

解決辦法:一開始查到可以使用 struct 方式，後來又找到可以使用陣列的方式。

```
void* arg[2] = {&no,&schedule};

 error t1 = pthread create(&pid t1,NULL,func,arg);
void* func(void* data){
    int s = 0;
    int *noptr = ((void**)data)[0];//t1 t2
    int *schptr = ((void**)data)[1];//schedule
```

3.Thread 2 先跑的情況:投影片範例中 thread 1 是比 thread 2 先跑，結果我的結果卻是 thread 2 先跑。找了網路上很久，才發現就算 code 中先寫 thread1 的 pthread_create() 再寫 thread2，在 function 中的 sched_setscheduler()的順序也不一定是 thread1 會先跑。

```
root@db-VirtualBox:/home/db# ./sched_test.out SCHED_FIFO
Thread 2 is running
Thread 2 is running
Thread 2 is running
Thread 2 is running
Thread 1 is running
Thread 1 is running
Thread 1 is running
Thread 1 is running
```

解決辦法: 使用 barrier，它可以讓 thread 執行到 pthread_barrier_wait 時，等待所有 thread 到齊，再依照正確的順序 run。

```
pthread barrier t barrier;
pthread_barrier_init(&barrier, NULL, 2);
pthread_barrier_wait(&barrier);
```

4.compile 出現下圖錯誤

```
/tmp/ccWbE7rt.o: In function `func':
rts1.c:(.text+0x39): undefined reference to `CPU_ZERO'
rts1.c:(.text+0x4f): undefined reference to `CPU_SET'
collect2: ld returned 1 exit status
```

解決辦法: compile 時加上 -D_GNU_SOURCE

```
root@db-VirtualBox:/home/db# gcc rts1.c -lpthread -o sched_test.out -D_GNU_SOURC
E
```