**Advanced Operating System**

Ch6 Mechanism : Limited Direct Execution homework

**Cheng Lin Hsu**

**Questions1:** measure the costs of a system call and context switch .

**Code**: **q1.c**

一張含有 文字 的圖片

自動產生的描述

**Execution** **Result for q1.c :**

一張含有 文字 的圖片

自動產生的描述

**Solution**: we shows the code of measure system call In Lines 10-19, Than ,we shows the code of measure context switch In Lines 21-60. We declared fd with system call open(), And then we also declared nloops with the value of 1000000 In Lines 10, Than,We declared a struct of timeval start and end In Lines 12, And then , we used gettimeofday() to know the start time in microseconds since 1970 ,In Lines 14-16 we used for loop to performing a 0-byte read 1000000 times In Lines 13 , After that in Lines 17 we used gettimeofday() to know the end time of system call in microseconds since 1970 ,In Lines 18 we print the cost of system call.(as shown as execution result in Lines 2). In Lines 21-23 is used to set the cpu 0 ,In Lines 25 we declared two arrays first\_pipefd[2], second\_pipefd[2],after that we create two pipe with first\_pipefd and second\_pipefd in Lines 26-33,than,we call a system call fork in Lines 35, As we can see that in Lines 40-48 and 49-60 show the code for the child process and the parent process respectively. In Lines 41-43 and 50-52, we used sched\_setaffinity() to make sure we used the same cpu , Than , In Lines 45-48 and 55-58, The first process then issues a write to the first pipe, and waits for a read on the second; upon seeing the first process waiting for something to read from the second pipe, the OS puts the first process in the blocked state, and switches to the other process, which reads from the first pipe and then writes to the second. When the second process tries to read from the first pipe again, it blocks, and thus the back-and-forth cycle of communication continues. And then we used gettimeofday() to measure context switch times in Lines 54 and 59,In Lines 60 we print out the cost of context switch .(as execution result in Lines 3.