## 操作系统

## 第三次作业

**线程**

1. **Discuss the difference between user-level thread and kernel level thread.**

调度方式上，内核线程的调度和切换由内核完成，与进程的调度和切换十分相似，而用户线程的调度不需内核的支持或特权。

调度单位上，内核线程的调度中时间片分配给线程，进程的线程数越多，获得的CPU时间越多。用户线程的调度以进程为单位进行，时间片分配给进程，多线程进程的每个线程较慢。

另外，如果内核是单线程的，那么一个用户线程发起系统调用而阻塞，则整个进程阻塞。而内核线程发起系统调用而阻塞，不会影响其他线程的运行。

1. **Which of the following components of program state are shared across threads in a multithreaded process?**

**a. Register values**

**b. Heap memory**

**c. Global variables**

**d. Stack memory**

bc

多线程进程中的线程共享堆内存和全局变量，不共享寄存器和栈内存。

1. **The program shown below uses the Pthreads API. What would be output from the program at LINE C and LINE P?**

**#include**

**#include**

**int value=0;**

**void \*runner(void \*param); /\* the thread \*/**

**int main()**

**{**

**int pid;**

**pthread\_t tid;**

**pthread\_attr\_t attr;**

**pid=fork();**

**if(pid==0)**

**{**

**pthread\_attr\_init(&attr);**

**pthread\_create(&tid, &attr, runner, NULL);**

**pthread\_join(tid, NULL);**

**printf("CHILD: value=%d\n", value);  /\* LINE C \*/**

**}else if(pid>0){**

**wait(NULL);**

**printf("PARENT: value=%d\n",value);　/\* LINE P \*/**

**}**

**}**

**void \*runner(void \*param)**

**{**

**value=5;**

**pthread\_exit(0);**

**}**

LINE C : CHILD : value=5 LINE P : PARENT : value=0

子进程修改全局变量不影响父进程中的值，因此父进程中value的值不变为0；子进程中，线程修改了全局变量的值为5，进程中的线程与进程共享全局变量，因此value的值修改为5。

1. **请说明三种多线程模型及其优缺点。**

多线程模型包括多对一模型、一对一模型和多对多模型。

多对一模型（Many-to-One）是指多个用户级线程映像进单个内核级线程，其缺点是并发性低，任一时刻只能有一个线程可以访问内核。一个用户线程发起系统调用而阻塞，则整个进程阻塞。

一对一模型（One-to-One）是指一个用户级线程映射到一个内核级线程，它提供了更好的并发性，一个用户级线程阻塞允许另一个线程运行，而缺点是每创建一个用户级线程就需要创建一个内核线程，带来了额外的开销，需限制应用中的线程数目。

多对多模型（Many-to-Many）是指多个用户级线程映射到多个内核级线程，它解决了多对一模型和一对一模型的缺点，不限制应用的线程数，多个线程可以并发。