操作系统(D)

第 4 次作业

李子龙 518070910095

2021 年 3 月 13 日

4.1 Provide three programming examples in which multithreading provides better performance than a single-threaded solution.

答:

1. Web 服务器可能有多个客户并发访问它，如果一个 Web 服务器作为单个线程的传统进程来进行，那么一次只能处理一个请求，将会导致客户端的等待时间过长。而如果作为多进程的程序，则可以同时处理多个客户端的请求，这样就可以大大缩短等待时间。
2. 计算矩阵乘法时，如果按照传统的方式进行逐行逐列扫描计算，将会耗费大量的时间。而多线程可以采用矩阵分块的方式进行计算，从而大大缩短计算时间。
3. 合并排序又是一个很好的多线程程序例子，它可以使得多个子序列的排序同时进行而不是单线程的逐步等待，这样可以大大缩短排序时间。

4.4 What are two differences between user-level threads and kernel-level threads? Under what circumstances is one type better than the other?

答: **用户线程**位于内核之上，它的管理无需内核支持；而**内核线程**由操作系统来直接支持与管理。

当只需要进行内核指令时，内核进程更好一些。需要与用户进行交互的时候，用户进程更好一些。

4.10 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

答: 全局变量(c.)。每个线程都有自己的寄存器与堆栈。

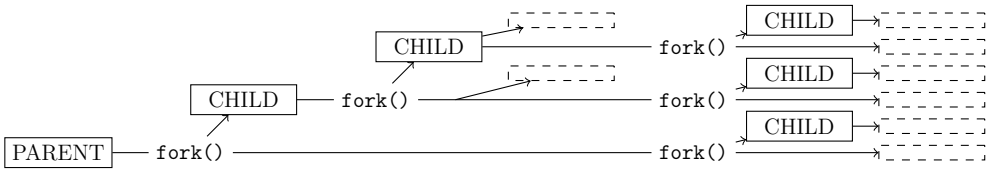
4.17 Consider the following code segment:

```c
pid_t pid;
pid = fork();
if (pid == 0) { /* child process */
    fork();
    thread_create( . . .);
}
fork();
```

a. How many unique processes are created?

b. How many unique threads are created?

答: 共有 6 个不同进程产生，8个不同的线程产生。



4.19 The program shown in Figure 4.23 uses the Pthreads API. What would be the output from the program at `LINE C` and `LINE P`?

```c
#include <pthread.h>
#include <stdio.h>
int value = 0;
void *runner(void *param); /* the thread */
int main(int argc, char *argv[])
{
    pid_t pid;
    pthread_t tid;
    pthread_attr_t attr;
    pid = fork();
    if (pid == 0) { /* child process */
        pthread_attr_init(&attr);
        pthread_create(&tid,&attr,runner,NULL);
        pthread_join(tid,NULL);
        printf("CHILD: value = %d",value); /* LINE C */
    }
    else if (pid > 0) { /* parent process */
        wait(NULL);
        printf("PARENT: value = %d",value); /* LINE P */
    }
}
void *runner(void *param) {
    value = 5;
    pthread_exit(0);
}
```

答: `CHILD: value = 5`，`PARENT: value = 0`。因为线程会改变全局变量的值，而进程之间的变量是真复制，不是引用的，所以父进程的值仍然没有改变。