OPERATING SYSTEMS

Professor: Hungwen Li

HOMEWORK 3

Name: Jeyanthh Venkatachari Ravikumar SJSU ID: 010720528

4.6

1. Any recursive Program or sequential program like a factorial of a number, in which multi-threading is of least use as each step depends on the output of the previous step. In these programs, multithreading is of least use. Ex: Fibonacci series.

2.An Unix Program or a shell program which maintains own process control block and directory space or an Unix command waiting for user response.

4.7 In case of a program with frequent **page faults**, A multithreading environment involving multiple kernel level threads can be effective as it can be swapped by another kernel thread.

However in a single user threaded environment with frequent page faults, in case a page fault occurs, the system has to wait for other events.

In these scenarios, **multithreaded solution using multiple kernel threads provide better performance than a single-threaded solution on a single-processor system**

4.8 The following components of a program state are shared across threads in a multithreaded

Environment

- Heap Memory
- Global Variables

However each thread will have separate register values and stack memory

4.11

In case of single processor systems, A single processors will implement concurrency by allocating a time slice for each thread to execute.

So multiple threads can be concurrently executed in a single processor, leading to concurrency without parallelism in place.

Parallelism is possible in case of multiprocessor/multicore environments.

4.12

To determine Speedup gain of an application which has 60% parallel component

a. 2 Processing cores

Speedup <= 1/ (S+ ((1-S)/N))

Where S- Serial Portion, N: number of cores

Here S=40%=.40; N=2

Therefore, Speedup<=1/.40+ (.6/2)=1/.7 =1.43

Moving from one 1 core to 2 cores will lead to a speedup of 1.43

b. 4 Processing cores

Speedup <= 1/ (S+ ((1-S)/N))

Where S- Serial Portion, N: number of cores

Here S=40%=.40; N=4

Therefore, Speedup<=1/.40+ (.6/4)=1/.55 =1.88

Moving from one 1 core to 2 cores will lead to a speedup of 1.82