OS HW2

108820018 蔡翔宇

4.4: Can a multithreaded solution using multiple user-level threads achieve better performance on a multiprocessor system than on a single-processor system? Explain.

A: No. Because only a single process runs in the operating system and the system won’t schedule the different threads of the process on separate processors.

4.13: Consider a multicore system and a multithreaded program written using the many-to-many threading model. Let the number of user-level threads in the program be greater than the number of processing cores in the system. Discuss the performance implications of the following scenarios.

(a) The number of kernel threads allocated to the program is less than the number of processing cores.

(b) The number of kernel threads allocated to the program is equal to the number of processing cores.

(c) The number of kernel threads allocated to the program is greater than the number of processing cores.

1. Some of the processors would remain idle since the scheduler maps only kernel threads to processors and not user-level threads to processors.
2. It is possible that all of the processors might be utilized simultaneously. However, when a kernel thread blocks inside the kernel, the corresponding processor would remain idle.
3. A blocked kernel thread could be swapped out in favor of another kernel thread that is ready to execute, thereby increasing the utilization of the multiprocessor system.

Q5.8:

Q5.15: Explain the differences in how much the following scheduling algorithms discriminate in favor of short processes:  
(a) FCFS  
(b) RR  
(c) Multilevel feedback queues

(a) discriminating against short jobs since any short jobs arriving after long jobs will have a longer waiting time.

(b) treating all jobs equally so short jobs will be able to leave the system faster since they will finish first.

(c) discriminating favorably toward short jobs.

Q5.19:

(a) Vruntime for A will be smaller than it is for B, resulting in a greater priority for A than B

(b) A will still have a smaller vruntime than B since it is I/O-bound and will require less CPU time.

(c) B will have a smaller vruntime than A since it is I/O-bound and will require less CPU time.

Q6.5:

Interrupts are not sufficient in multiprocessor systems since disabling interrupts only prevents other processes from executing on the processor in which interrupts were disabled; there are no limitations on what processes could be executing on other processors and therefore the process disabling interrupts cannot guarantee mutually exclusive access to program state.

Q6.23:

This operation resumes exactly one other process in monitor if any process is currently suspended due to a wait( ) operation on the condition variable. Of course, if no thread is waiting, then the signal is not saved (and will have no effect)