GSI018 – SISTEMAS OPERACIONAIS

Operating Systems – William Stallings – 7th Edition

Chapter 04 – Processes and Threads

Murielly Oliveira Nascimento – 11921BSI222 – murielly.nascimento@ufu.br

PROBLEMS

4.1 It was pointed out that two advantages of using multiple threads within a process are that (1) less work is involved in creating a new thread within an existing process than in creating a new process, and (2) communication among threads within the same process is simplified. Is it also the case that a mode switch between two threads within the same process involves less work than a mode switch between two threads in different processes?

Sim, porque duas threads num mesmo processo vão precisar guardar menos informações ao alterarem seus estados do que threads em diferentes processos.

4.2 In the discussion of ULTs versus KLTs, it was pointed out that a disadvantage of ULTs is that when a ULT executes a system call, not only is that thread blocked, but also all of the threads within the process are blocked. Why is that so?

Na maioria dos sistemas operacionais, muitas das chamadas são bloqueantes o que leva ao bloqueio de todo o processo, não só da "thread" que executou a chamada.

4.4 Consider an environment in which there is a one-to-one mapping between user-level threads and kernel-level threads that allows one or more threads within a process to issue blocking system calls while other threads continue to run. Explain why this model can make multithreaded programs run faster than their single-threaded counterparts on a uniprocessor computer.

Em abordagens single-thread há um único encadeamento de execução por processo, enquanto que em multithreading o SO consegue oferecer suporte a vários caminhos de execução. Dessa forma, no primeiro cenário a máquina fica esperando por operações de E/S terminarem, e já no segundo outros processos são executados nesse meio tempo.

4.5 If a process exits and there are still threads of that process running, will they continue to run?

Não, quando um processo termina os seus threads também são encerrados.

4.7 Many current language specifications, such as for C and C++, are inadequate for multithreaded programs. This can have an impact on compilers and the correctness of code, as this problem illustrates. Consider the following declarations and function definition:

```
int global_positives = 0;
```

```
typedef struct list {
struct list *next;
double val;
} * list;
void count_positives(list l) {
list p;
for(p = l; p; p = p -> next )
if( p -> val > 0.0 )
++global_positives;
}
```

Now consider the case in which thread A performs "count_positives(<list containing only negative values>)" while thread B performs "++global_positives".

a. What does the function do?

Linguagens como C e C++ seguem o paradigma de programação procedimental, no qual o programa executa linhas de código de cima para baixo, especificamente. Se a thread B incrementa o número de inteiros positivos a cada iteração a thread A não irá avaliar corretamente quantos números naturais há na lista.

b. The C language only addresses single-threaded execution. Does the use of two parallel threads create any problems or potential problems?

Sim, como visto no caso acima a lógica do programa é comprometida pelo uso de threads paralelas. Também, há a questão da declaração de variáveis, um thread pode usar variáveis alocadas, localmente, em outros escopos comprometendo o funcionamento do programa.