Laboratório 9
Introdução à programação multiprocesso em C

Programação Concorrente (ICP-361)
Profa. Silvana Rossetto

[1]Instituto de Computação/CCMN/UFRJ

Introdução

O objetivo deste Laboratório é **experimentar a programação concorrente em programas multiprocessos**. Para cada atividade, siga o roteiro proposto e responda às questões colocadas.

Atividade 1

Objetivo: Mostrar como criar um programa multiprocesso em C.

Roteiro:

1. Leia os programas `fork.c` e `fork-execve.c` e tente entender o que eles fazem (**acompanhe a explanação da professora**).
2. Compile e execute os programas e observe os resultados impressos na tela.

Atividade 2

Objetivo: Mostrar como implementar comunicação e sincronização entre processos filhos de um mesmo programa, usando **memória compartilhada**.

Roteiro:

1. Leia o programa `fork-soma.c`. *Ele mostra que após criar um processo filho, os espaços de memória não são compartilhados.* **Qual será a saída do programa?** (acompanhe a explanação da professora).
2. Compile e execute o programa e avalie os resultados.
3. Leia o programa `fork-mem.c`. Esse programa mostra uma forma de *compartilhar memória entre processos*. **Qual será a saída do programa?** (acompanhe a explanação da professora).
4. Leia o programa `fork-soma1.c`. Em que ele se diferencia do programa `fork-soma`? **Qual será a saída do programa?** (acompanhe a explanação da professora).
5. Leia o programa `fork-soma2.c`. Em que ele se diferencia do programa `fork-soma1`? **Qual será a saída do programa?** (acompanhe a explanação da professora).

Atividade 3

Objetivo: Medir e comparar o desempenho de criação e finalização de processos e threads.

Roteiro:

1. Leia os programas `fork-time.c` e `thread-time.c` (acompanhe a explanação da professora).
2. Compile e execute os programas e compare os resultados dos tempos medidos.

Atividade 4

Objetivo: Mostrar como implementar comunicação e sincronização entre processos filhos de um mesmo programa, usando **pipe**.

Roteiro:

1. Leia o programa `fork-pipe.c`. Um pipe é um buffer circular que permite comunicação entre dois processos no modelo produtor/consumidor.
2. Compile e execute o programa e observe seus resultados (é necessário digitar a mensagem a ser transmitica na entrada padrão, o outro processo imprimirá a mesma mensagem na saída padrão).