

MAC 0219/5742 – Introdução à Computação Concorrente, Paralela e Distribuída

Prof. Dr. Alfredo Goldman

MiniEP3 - Paralelismo com Processos versão 1.0

Monitores: Giuliano Belinassi e Matheus Tavares

1 Introdução

Vimos como implementar concorrência dentro de um processo através de *threads*, algo muito eficiente e com diversos recursos para controlar e sincronizar o trabalho realizado por cada *thread*. Entretanto, é perfeitamente possível realizar programação concorrente/-paralela utilizando apenas processos.

Diferentemente de *threads*; processos têm a característica de haver o seu segmento de memória isolado dos demais fluxos de execução, permitindo um melhor isolamento da memória quando comparado as *threads*. Isso evita, por exemplo, com que falhas de segurança em um navegador vazem dados através de abas distintas. Um navegador que utiliza esse conceito para implementar concorrência entre abas é o Google Chrome¹.

Neste Mini-EP vocês deverão implementar um simples problema de redução (cálculo de π) usando **apenas** processos. Para isso, vocês deverão implementar um mecanismo de comunicação entre os processos utilizando as ferramentas já existentes no Linux.

2 Problema

Vocês deverão calcular usando Integração de Riemman:

$$\pi = 4 \int_0^1 \sqrt{1-x^2} dx = 4 \sum_i^n \sqrt{1-x_i^2} \Delta x \quad (1)$$

Utilizando **apenas** processos, isto é, vocês podem criar novos processos usando `fork()`, e aguardar com que os processos terminem usando o `waitpid()`, mas não podem criar novas *threads*. Vocês podem adaptar o programa de cálculo do π visto em aula.

Isso pode ser feito de várias maneiras. Uma possibilidade é usar os segmentos de memória compartilhada, seja o do extinto **System V** através das funções `shmget` e `shmat`, ou o do **BSD** através do `mmap`; ambos suportados pelo Linux². Também é possível criar um FIFO do UNIX, e trocar informações através dele. Outra maneira é abrir **sockets** para a troca de mensagens.

Só não usem arquivos. Ao fazer isso nada é aprendido nesse Mini-EP, além de ser uma alternativa ineficiente por causa da necessidade de acesso ao disco. ☺

¹<https://blog.chromium.org/2008/09/multi-process-architecture.html>

²https://gerardnico.com/os/linux/shared_memory

3 Especificação do Programa

Seu EP deve ser capaz de atender os seguintes requisitos:

1. Usar a Integração de Riemman para calcular π , conforme descrito acima.
2. Conter um `Makefile` que gera o binário `pi_process`. Este binário deve aceitar como argumento a seguinte linha de comando:

```
./pi_process NUM_PROCESSOS NUM_PONTOS
```

Você não precisa sanitizar os parametros de entrada.

3. Todo o paralelismo deve ser feito usando apenas processos, ou seja, não pode abrir *threads* dentro de um processo.
4. O programa deverá imprimir na saída padrão (`stdout`) uma aproximação de π .
5. Qualquer recurso do Sistema Operacional que precise ser representado por arquivos (ex. um FIFO) deve ser criado no `/tmp` ou `/var/tmp`.
6. Para essa atividade, **é muito importante** que os recursos compartilhados alocados sejam limpos ao final da execução do programa, inclusive em casos onde ele foi abortado com erros. O Linux **não** limpa esses recursos automaticamente, e é perfeitamente possível que ocorra erros ao tentar alocar esses recursos novamente, ou causar graves vazamentos de memória que apenas podem ser resolvidos desalocando a memória na linha de comando, ou reiniciando a máquina.

4 Entrega

Deverá ser entregue um pacote no sistema PACA com uma pasta com o nome e o sobrenome do estudante que o submeteu no seguinte formato: `nome_sobrenome.zip`. Essa pasta deve ser comprimida em formato ZIP e deve conter dois itens:

- Os códigos fonte do programa, em conjunto com um `Makefile` que o compila, gerando o executável especificado.
- Um relatório em `.txt` ou `.pdf` com o nome dos integrantes, e uma breve explicação sobre a sua solução, desafios encontrados. Imagens também podem ser inseridas no pacote. Relatórios em `.doc`, `.docx` ou `odt` **não** serão aceitos.

Em caso de dúvidas, use o fórum de discussão do Paca. A data de entrega deste Exercício Programa é até às **16:00h do dia 30 de Maio**.

Boa Sorte!