

Sistemas Operacionais I

Trabalho 1: Ray-Tracing


Grupo: Breno Brandão, Bruno Fontes, Ingryd Moura, Leonídia Barreto

Programa original

- O tempo aproximado de execução do programa original é de 15,33 segundos
- Foi utilizado um processador de 4 núcleos na execução de todas as versões do programas



Processos

- Foram criadas 4 funções:
 - `divide()`: Retorna as divisões da imagem, de acordo com o número de divisões pedido.
 - `comecaraytracerloopcoordenadas()`: Centraliza chamadas de funções que tratam imagem, é onde está a criação de processos (`fork`).
 - `raytracerLoop()` e `raytracerLooppref()`: Tratam parte da imagem passada.
 - Uso de shared memory no `fork()`
 - `shmget()`: Cria endereço na memória.
 - `shmat()`: Retorna ponteiro para endereço na memória.
 - O tempo médio de execução, fazendo uso de 1 processo é de 15,60 segundos
 - O tempo médio de execução, utilizando 20 processos é de 7 segundos
- 

Threads

Foi criada a função `raytracerLoop` contendo os loops que tratam parte da imagem. Além disso, foi criado um vetor da estrutura `varFromLoop`, que contém os parâmetros necessários para a execução do `raytracerLoop`.

- Acontecem os seguintes tempos médios de execução:
 - Com 1 thread: 15,30 segundos
 - Com 4 threads: 8,40 segundos
 - Com 8 threads: 7 segundos
 - Com 12 threads: 6,90 segundos



OpenMP

Como threads compartilham o mesmo espaço de endereçamento, foi preciso declarar as variáveis `s` e `j` dentro do primeiro `for`, para que toda a imagem fosse carregada corretamente.

- Acontecem os seguintes tempos médios de execução:
 - Com 1 thread: 15,80 segundos
 - Com 4 threads: 7,44 segundos
 - Com 8 threads: 7,14 segundos
 - Com 12 threads: 6,90 segundos





Fim