

**UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO
CURSO DE ENGENHARIA DE COMPUTAÇÃO
DISCIPLINA DE SISTEMAS OPERACIONAIS**



RELATÓRIO ESCALONADOR – SISTEMAS OPERACIONAIS

WAGNER LOCH

**PELOTAS, OUTUBRO DE 2019
Wagner Loch – wloch@inf.ufpel.edu.br**

Introdução

Um escalonador de processos é um programa responsável por executar processos concorrentes, dando prioridades para alguns e pausando outros quando há necessidade.

Este programa é responsável por escolher qual processo será executado pela CPU, se preocupando com eficiência e escolhendo as melhores técnicas de chaveamento, isto é, troca entre processos sempre priorizando a satisfação do usuário, ou seja, sem percepções de trocas.

Objetivos do Escalonador

Um bom programa escalonador deve se preocupar e atentar-se a alguns detalhes, tais como, evitar o adiamento indefinido de processos, maximizar ao máximo a produtividade, ser previsível, não desperdiçar recursos, favorecer processos de alta prioridade etc.

Um dos maiores problemas para um escalonador está na previsibilidade, visto que os processos são de natureza imprevisível.

Existem diversos algoritmos de escalonamento, onde alguns fazem uso da prioridade do processo enquanto outros são justos quanto ao tempo que o processo está aguardando.

Desenvolvimento

O desenvolvimento deste trabalho se deu utilizando o conceito de programação paralela de memória compartilhada OpenMP. Para tal utilização, é necessário fazer a compilação do programa utilizando a seguinte linha de comando:

```
gcc -fopenmp escalonador.c -o escalonador
```

Após o processo de compilação, a execução do programa se dá utilizando o seguinte comando:

```
./escalonador 2 1 1024 processos.txt onde o primeiro argumento (2) é o número de CPUs disponíveis na simulação, o segundo argumento (1) é o slice de tempo, o terceiro argumento (1024) é a quantidade de memória disponível para a simulação e
```

o quarto e último argumento é o nome do arquivo cujo contém os processos que serão executados na simulação.

Sobre o Código

O código foi desenvolvido utilizando a linguagem C e conforme já mencionado, utilizando OpenMP para execuções em paralelo. Todo o código é de fácil entendimento e está bem documentado com diversos comentários sobre as operações realizadas.

Dificuldades Encontradas

Durante a execução do trabalho, algumas dificuldades em sua realização foram encontradas, tais como, o tratamento de prioridades. Foi feito o desenvolvimento porém não demonstrou-se satisfatório, visto que em alguns testes, os processos ficavam em alternância e nunca chegavam a sua conclusão. Por este motivo, resolveu-se abstrair do trabalho este requisito. O código desenvolvido encontra-se comentado no arquivo fonte.

Desta forma, a execução do escalonador acontece utilizando o algoritmo FIFO (First In, First Out), que nada mais é do que uma fila, onde o primeiro processo que chegar ao Sistema Operacional é executado em sua totalidade, se houver mais CPUs disponíveis, outros processos são imediatamente iniciados assim que chegarem. Novos processos devem aguardar a finalização dos primeiros para poderem iniciar.

Resultados

Ao executar algumas simulações, podemos perceber o quão demorado pode ser finalizar a fila de execução quando dispomos de poucas unidades de processamento disponíveis. Ao aumentar o número de recursos o tempo total para a execução decai consideravelmente. É exatamente isto que ocorre em nossos sistemas. Podemos perceber nitidamente ao utilizar computadores e sistemas mais antigos e que possuem poucos recursos computacionais disponíveis a lentidão na execução de tarefas triviais, que são rapidamente executadas em sistemas mais

modernos. Isto comprova o quão importante é aliar um bom hardware com um bom software. Ter um bom hardware porém um software deficitário ou vice-versa somente trará dores de cabeça e diversos problemas durante a utilização. Junto aos resultados, é demonstrado o tempo total de execução, onde podemos perceber a sua diminuição com o aumento de recursos.