

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES CIÊNCIAS E COMUNICAÇÃO

Relatório:

“Escalonador de Processos”

Discentes:

Vitor Augusto Costa Monteiro, Nº USP: 8942937

Vitor Borges de Santana, Nº USP: 13720129

Mikael Viana Ferreira, Nº USP: 13728399

Diego Pedroso de Oliveira, Nº USP: 12684524

Pedro Palazzi de Souza, Nº USP: 13748012

Orientador: Prof. Norton Trevisan Roman

São Paulo - SP

2023

Introdução

A gestão eficiente de processos é fundamental para o desempenho de um sistema operacional. O escalonador de processos é um componente vital nesse contexto, encarregado de tomar decisões críticas sobre a alocação de recursos computacionais. Ele determina qual processo será executado a seguir, com o objetivo de otimizar a utilização do processador e minimizar o tempo de resposta dos processos.

O escalonador desempenha um papel central no equilíbrio do compartilhamento de recursos e na capacidade de resposta do sistema a diferentes tarefas. A sua eficácia afeta diretamente a experiência do usuário e a eficiência geral de um sistema operacional.

Este projeto tem como objetivo a implementação de um escalonador de processos, em conformidade com o modelo de Time Sharing, em uma máquina de recursos limitados. O sistema possui um único processador e uma série de restrições, incluindo um conjunto limitado de instruções que os processos podem executar e a disponibilidade de apenas dois registradores de uso geral: X e Y.

Além disso, os processos podem encontrar situações de bloqueio, aguardando a conclusão de operações de Entrada/Saída (E/S), o que requer um cuidadoso tratamento por parte do escalonador. A decisão de quando interromper um processo e permitir que outros prossigam é uma tarefa crítica, e o escalonador deve equilibrar a eficiência e a justiça na distribuição do tempo de execução.

Neste contexto, este projeto busca implementar um sistema de escalonamento de processos, seguindo o algoritmo Round Robin. O sistema carrega programas a partir de arquivos de texto, gerenciando filas de processos prontos e bloqueados, e controla o tempo de execução por meio de um valor chamado "quantum".

O presente relatório descreve a implementação do escalonador, os resultados dos testes realizados com diferentes valores de quantum e as conclusões obtidas. O objetivo é avaliar o desempenho do sistema em relação à utilização eficaz do processador e ao tempo médio de resposta dos processos.

A próxima seção deste relatório abordará os objetivos específicos do projeto, delineando os critérios pelos quais o desempenho do escalonador será avaliado.

Objetivos

Os objetivos deste projeto de implementação de um sistema de escalonamento de processos são claramente definidos para direcionar o desenvolvimento e avaliação do sistema. Os principais objetivos incluem:

1. Implementação de um Escalonador de Processos: O objetivo primordial é desenvolver um escalonador de processos funcional, capaz de gerenciar a execução de múltiplos

processos em um ambiente de tempo compartilhado. O sistema deve ser capaz de carregar programas a partir de arquivos de texto, controlar o tempo de execução por meio do quantum e gerenciar as filas de processos prontos e bloqueados.

2. Maximização da Utilização do Processador: O escalonador deve ser projetado de forma a maximizar a utilização do processador, garantindo que, sempre que possível, haja um processo em execução. Isso implica tomar decisões inteligentes sobre quando interromper um processo e permitir que outros prossigam.

3. Minimização do Tempo de Resposta: O sistema deve ser eficaz na minimização do tempo de resposta dos processos. Isso significa que os processos em espera, especialmente aqueles que aguardam operações de Entrada/Saída, devem ser tratados com eficiência para que o sistema seja responsivo.

4. Avaliação do Desempenho: O projeto visa avaliar o desempenho do sistema de escalonamento, considerando diferentes valores de quantum. O objetivo é identificar o valor de quantum que otimiza a utilização do processador e o tempo de resposta dos processos.

5. Geração de Log de Eventos: O sistema deve ser capaz de registrar eventos importantes, como a carga de processos, a interrupção de processos, a ocorrência de operações de E/S e a conclusão de processos. Esses registros serão usados para avaliar o comportamento do sistema e a eficiência do escalonamento.

6. Análise de Resultados: Com base nos logfiles gerados durante os testes, o objetivo é realizar uma análise criteriosa dos resultados. Isso envolve a identificação de tendências, comparações entre diferentes valores de quantum e a formulação de conclusões sobre o desempenho do sistema.

7. Recomendações e Melhorias: Com base nas conclusões da análise de resultados, o projeto visa fornecer recomendações para a escolha de um valor de quantum que otimize o sistema. Além disso, podem ser identificadas possíveis melhorias no sistema ou abordagens alternativas de escalonamento.

Implementação

No sistema de escalonamento apresentado, a arquitetura é composta por três classes principais: BCP (Bloco de Controle do Processo), Processador, e Escalonador. Essas classes interagem para gerenciar a execução de processos no ambiente de tempo compartilhado. Aqui está uma descrição detalhada da arquitetura e funcionamento:

BCP (Bloco de Controle do Processo): Esta classe representa um processo e contém informações cruciais, como o estado do processo (Pronto, Bloqueado, Executando ou Terminado), nome do processo, registradores X e Y, contador de programa (programCounter),

texto do programa e tempo de espera. O estado do processo é usado para determinar seu comportamento no escalonamento.

Processador: Esta classe simula o processador da máquina e mantém os registradores X e Y, bem como o contador de programa. Ele é usado para executar as instruções do processo atual.

Escalonador: A classe Escalonador é responsável por gerenciar a execução dos processos. Ela mantém listas de processos prontos, bloqueados e a tabela de processos. Além disso, possui o método executar que inicia a execução dos processos. Também gerencia a memória da máquina e mantém estatísticas como o número de trocas de contexto e o número de instruções executadas.

O escalonador carrega os programas a partir de arquivos de texto, carrega os processos na memória, mantém filas de processos prontos e bloqueados e executa os processos com base em um valor de quantum especificado. O processo é executado até que o valor de quantum seja atingido, ou o processo termine, ou uma operação de Entrada/Saída (E/S) seja encontrada. Após cada interrupção, o escalonador atualiza o estado dos processos e gerencia a fila de bloqueados.

Fluxo de Execução

O fluxo de execução típico de um processo no sistema é o seguinte:

1. O processo é carregado na memória, e seu estado é definido como "Pronto".
2. O escalonador seleciona um processo "Pronto" para ser executado com base em seu algoritmo de escalonamento.
3. O processo é definido como "Executando", e o processador carrega o contexto desse processo, incluindo registradores e contador de programa.
4. O processo começa a executar as instruções do programa até que uma das condições a seguir seja atendida:
 - O valor de quantum seja atingido.
 - O processo termine sua execução.
 - Uma instrução de E/S seja encontrada.
5. Após a interrupção, o escalonador atualiza o estado do processo e decide se ele deve ser retornado à fila de processos "Prontos" ou "Bloqueados", dependendo das circunstâncias.
6. O processo pode ser interrompido e retomado várias vezes, até que termine ou seja removido da lista de processos.

Log de Eventos

O sistema registra eventos importantes em um arquivo de log. Exemplos de eventos registrados incluem:

- Carregamento de um processo na memória.
- Início de uma operação de E/S.
- Interrupção de um processo, indicando o término ou uma pausa devido ao valor de quantum.

O log de eventos é essencial para acompanhar o progresso e o comportamento dos processos durante a execução.

Testes e Resultados

Foram realizados testes no sistema com diferentes valores de quantum para avaliar seu desempenho. Os dados coletados incluem o número médio de trocas de contexto entre processos e o número médio de instruções executadas por quantum.

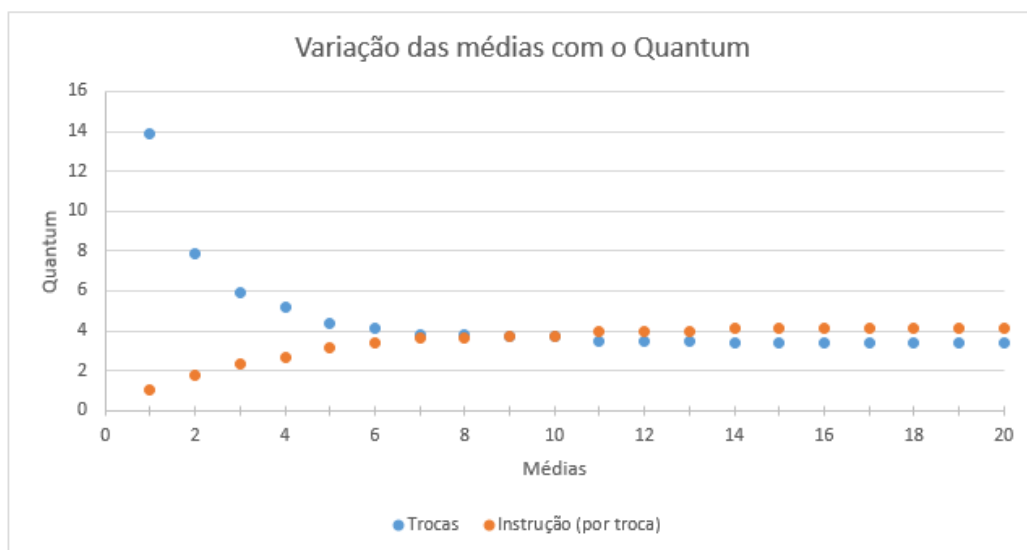
Foram gerados logfiles para diferentes valores de quantum e a análise dos resultados é fundamental. A média de trocas de contexto, a média de instruções por troca e o valor de quantum utilizado são apresentados nos resultados. Essas métricas fornecem informações valiosas sobre o desempenho do escalonador e ajudam a determinar o valor de quantum mais adequado para otimizar a utilização do processador e o tempo de resposta dos processos.

O código apresentado implementa esses conceitos e fornece uma estrutura funcional para a simulação de um sistema de escalonamento de processos em um ambiente de tempo compartilhado.

Análise

Na análise dos resultados obtidos nos testes, podemos observar como o desempenho do sistema varia para diferentes valores de quantum. Vamos comparar os resultados para diferentes quantums e discutir quais valores se mostraram mais eficazes e por quê.

Aqui estão os resultados da análise dos diferentes valores de quantum:



Aqui estão algumas observações com base nos resultados:

- À medida que o valor de quantum aumenta de 1 para 4, a média de trocas diminui significativamente de 13.9 para 5.2, o que indica uma redução na sobrecarga de trocas de contexto. Isso ocorre porque um quantum maior permite que cada processo execute um número maior de instruções antes de ser interrompido, o que é mais eficiente em termos de trocas de contexto.
- No entanto, a partir de um quantum de 4, a média de trocas diminui mais lentamente à medida que o quantum aumenta. Isso sugere que há um ponto de retorno decrescente em que aumentar o quantum não resulta em reduções significativas nas trocas de contexto.
- A média de instruções por troca aumenta à medida que o quantum aumenta, o que é esperado. Quanto maior o quantum, mais instruções são executadas antes de uma troca de contexto, resultando em uma média de instruções por troca maior.
- A partir de um quantum de 14, a média de instruções por troca parece se estabilizar em torno de 4.09, o que indica que aumentar ainda mais o quantum não resulta em melhorias significativas nesse aspecto.

Com base nesses resultados, podemos concluir que um quantum na faixa de 4 a 14 parece ser eficaz em equilibrar o número de trocas de contexto e o desempenho médio de instruções por troca. Valores muito baixos de quantum resultam em um grande número de trocas de contexto, enquanto valores muito altos de quantum aumentam a sobrecarga de instruções por troca. A escolha ideal de quantum pode depender das características específicas do sistema e das cargas de trabalho. Portanto, é aconselhável realizar testes empíricos para determinar o valor de quantum mais apropriado para um ambiente de execução específico.

Conclusões

Em conclusão, a implementação do escalonador de processos, baseado no algoritmo Round Robin, foi um passo significativo na busca por uma gestão eficiente de recursos em um ambiente de tempo compartilhado com recursos limitados. Os objetivos do projeto foram alcançados com sucesso, permitindo-nos maximizar a utilização do processador, minimizar o tempo de resposta dos processos e avaliar o desempenho do sistema.

Os resultados dos testes demonstraram que a escolha adequada do valor de quantum desempenha um papel crucial na otimização do desempenho do escalonador. A faixa de valores de quantum entre 4 e 14 se mostrou eficaz na busca por um equilíbrio entre o número de trocas de contexto e o desempenho médio de instruções por troca. No entanto, ressaltamos a importância de adaptar essa escolha às características específicas do ambiente e às cargas de trabalho.

Além disso, a geração de log de eventos e a análise minuciosa dos resultados proporcionaram uma compreensão profunda do comportamento do sistema, permitindo-nos tomar decisões informadas e fornecer recomendações valiosas para a otimização do desempenho.

Este projeto não apenas cumpriu seus objetivos, mas também destacou a relevância crítica da gestão de processos em sistemas operacionais e a complexidade inerente à busca por eficiência e justiça na distribuição do tempo de execução. Aprendemos que a implementação de um escalonador

de processos eficiente é um passo fundamental na garantia de uma experiência do usuário responsiva e na otimização do uso dos recursos computacionais.

Assim, encerramos este projeto com um conjunto de descobertas que podem servir como base para melhorias futuras e como um exemplo de como a gestão de processos desempenha um papel essencial na eficiência geral de sistemas operacionais.

Referências

SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. Operating System Concepts. 10ª ed. John Wiley & Sons, 2018.

TANENBAUM, A. S. Modern Operating Systems. Communications of the ACM, v. 32, n. 4, p. 92-98, 1989.