Universidade de São Paulo
Escola de Artes, Ciências e Humanidades

Exercício Programa 2

Sistemas Operacionais - ACH2044

Orientador: Prof. Norton Trevisan Roman

Victor A. C. Monteiro, NUSP: 8942937
Vitor Borges de Santana, NUSP: 13720129
Mikael Viana Ferreira, NUSP: 13728399
Diego Pedroso de Oliveira, NUSP: 12684524
Pedro Palazzi de Souza, NUSP: 13748012

São Paulo - SP

10 de dezembro de 2023

Sumário

Sumário **2**
1 Introdução 3
2 Objetivos 4
3 Implementação 5
4 Fluxo de Execução 6
5 Log de Eventos 7
6 Testes e Resultados 8
7 Análise 9
8 Conclusões 11
Referências **12**

1 Introdução

Neste trabalho, optamos por abordar a implementação do problema clássico de Leitores e Escritores, explorando os conceitos fundamentais de concorrência e threads. Nosso grupo, composto por 5 integrantes, organizou-se para criar uma solução que ilustrasse de maneira prática a aplicação desses conceitos. A tarefa envolveu a manipulação de uma estrutura de dados contendo palavras do arquivo "bd.txt", correspondente ao texto filosófico "A Treatise Concerning the Principles of Human Knowledge"de George Berkeley (1710). Ao transformar essa estrutura em uma região crítica, permitindo acesso concorrente, criamos uma abordagem que utiliza threads leitoras e escritoras, distribuídas aleatoriamente em um arranjo de 100 objetos. Ao longo deste relatório, detalharemos a nossa implementação e apresentaremos análises de desempenho para cenários que envolvem diferentes proporções entre leitores e escritores. Este trabalho representa nossa exploração prática das soluções para o desafio de Leitores e Escritores, destacando as nuances da concorrência em sistemas computacionais (TANENBAUM; BOS, 2014b)(TANENBAUM; BOS, 2014a)(TANENBAUM; BOS, 2017).

2 Objetivos

1. **Implementação de Threads:** Desenvolver uma solução eficiente para o problema de Leitores e Escritores, aplicando o conceito de threads em Java.

2. **Ilustração da Concorrência:** Demonstrar, por meio da implementação, como a concorrência entre threads pode ser gerenciada em uma região crítica, considerando acesso simultâneo à base de dados.

3. **Análise de Desempenho:** Realizar avaliações quantitativas do desempenho da implementação, medindo o tempo total de execução do sistema e coletando dados estatísticos relevantes.

4. **Comparação de Abordagens:** Comparar os resultados obtidos com uma solução que não utiliza leitores e escritores, avaliando o impacto na eficiência do sistema em diferentes proporções entre leitores e escritores.

5. **Priorização de Leitores:** Implementar uma política que prioriza leitores em relação a escritores, explorando como essa abordagem influencia o comportamento do sistema em termos de tempo de execução e bloqueios.

6. **Exploração Didática:** Proporcionar uma experiência de aprendizado prática sobre o manuseio de concorrência em sistemas computacionais, destacando desafios e estratégias para lidar com situações críticas.

3 Implementação

Neste projeto, optamos por uma implementação em Java, fazendo uso extensivo de threads para ilustrar o problema clássico de Leitores e Escritores. Inicialmente, organizamos um grupo de 100 threads, divididas em leitores e escritores, cada uma representando um objeto de acesso concorrente. A estrutura de dados central consiste em um arranjo que armazena palavras do texto "A Treatise Concerning the Principles of Human Knowledge"de George Berkeley, onde cada palavra é considerada uma linha no arquivo "bd.txt".

A abordagem para a criação e execução das threads envolveu a aleatoriedade na escolha das posições no arranjo, promovendo uma distribuição não determinística dos objetos. Durante a execução, os leitores foram projetados para acessar a base apenas para leitura, enquanto os escritores modificaram as palavras para "MODIFICADO". Cada objeto realizou 100 acessos aleatórios à base, seguidos por uma pausa simulada de 1ms, simbolizando uma validação dos dados.

A estrutura da solução manteve uma região crítica, permitindo o acesso concorrente, embora, didaticamente, a política tenha priorizado os leitores. Antes do primeiro acesso, um objeto entra na região crítica, saindo apenas após a conclusão de 100 acessos e a pausa de 1ms. Essa escolha de projeto visa proporcionar uma compreensão clara do impacto da concorrência na execução do sistema.

Além disso, implementamos ou utilizamos uma solução em Java para o problema de Leitores e Escritores, aplicando-a à estrutura de dados e ao arranjo de threads. O tempo total de execução do sistema foi medido entre o final do povoamento do arranjo e o término da última thread, garantindo que a thread principal não encerrasse antes da conclusão do processo.

Essa implementação permitiu uma exploração aprofundada do problema, incorporando conceitos fundamentais de concorrência e destacando desafios práticos associados à manipulação de regiões críticas em sistemas multitarefa.

4 Fluxo de Execução

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

1. **O Inicialização da Estrutura de Dados:** Leitura do arquivo "bd.txt". Construção de uma estrutura de dados em memória, armazenando palavras em cada elemento.

2. **Criação do Arranjo de Threads:** População de um arranjo de 100 threads, contendo objetos de leitura e escrita. Organização aleatória dos objetos no arranjo para introduzir variabilidade nas interações.

3. **Execução das Threads:** Cada thread realiza 100 acessos à estrutura de dados. Acesso aleatório a posições na estrutura para leitura ou escrita.

4. **Acesso Concorrente à Estrutura de Dados:** Entrada na região crítica antes do primeiro acesso. Garantia de acesso exclusivo a um objeto por vez na estrutura de dados.

5. **Comportamento dos Leitores:** Leitores acessam a palavra na posição desejada para leitura. Armazenamento local da palavra lida.

6. **Comportamento dos Escritores:** Escritores substituem a palavra na posição correspondente por "MODIFICADO". Simulação de uma pausa de 1ms após cada acesso, representando validação de dados.

7. **Coleta de Dados de Desempenho:** Registro do tempo total de execução do sistema. Contagem do número total de leituras e escritas realizadas. Cálculo do tempo médio de bloqueio durante a execução das threads.

8. **Análise dos Resultados:** Avaliação do comportamento do sistema em diferentes cenários de concorrência. Compreensão dos desafios associados ao acesso concorrente a recursos compartilhados.

9. **Finalização do Fluxo de Execução:** Conclusão da execução de todas as threads. Apresentação dos resultados e conclusões sobre o desempenho do sistema.

5 Log de Eventos

Durante a implementação e execução do sistema de Leitores e Escritores, registramos eventos significativos que nos permitiram compreender o comportamento das threads e avaliar o desempenho do sistema. O log de eventos fornece uma visão detalhada das atividades executadas pelas threads, permitindo uma análise minuciosa do acesso concorrente à estrutura de dados compartilhada.

6 Testes e Resultados

Conduzimos uma série de testes para avaliar o desempenho do sistema de Leitores e Escritores em diferentes cenários de concorrência. Os resultados obtidos revelam informações cruciais sobre a eficácia da implementação e proporcionam insights valiosos para a análise do sistema.

Os resultados obtidos são os seguintes:

Cenário 1:
Total de Leituras: 5000
Total de Escritas: 5000
Tempo de Bloqueio: 1573 ms
Tempo de Execução: 5225 ms

Cenário 2:
Total de Leituras: 6000
Total de Escritas: 4000
Tempo de Bloqueio: 1072 ms
Tempo de Execução: 4707 ms

Cenário 3:
Total de Leituras: 5500
Total de Escritas: 4500
Tempo de Bloqueio: 1304 ms
Tempo de Execução: 4741 ms

Cenário 4:
Total de Leituras: 9000
Total de Escritas: 1000
Tempo de Bloqueio: 80 ms
Tempo de Execução: 5053 ms

7 Análise

Os resultados dos testes indicam o comportamento do sistema sob diferentes proporções de leitores e escritores, os resultados podem ser vistos na Tabela 1. Destacamos alguns pontos-chave para análise:

- **Impacto da Proporção Leitores/Escritores:** O aumento do número de leitores em relação aos escritores resultou em menor tempo de bloqueio, favorecendo o acesso concorrente.
- **Tempo de Bloqueio Significativo:** Em cenários com proporções mais equilibradas, o tempo de bloqueio aumentou, indicando maior competição pelos recursos compartilhados.
- **Eficiência do Controle de Acesso:** A implementação efetiva de regiões críticas proporcionou controle sobre o acesso à estrutura de dados compartilhada, minimizando conflitos.
- **Desempenho Geral:** O tempo total de execução permaneceu razoavelmente consistente, demonstrando a capacidade do sistema de lidar com operações concorrentes de maneira eficiente.

A análise dos resultados oferece insights cruciais para otimizações futuras e destaca a eficácia da solução implementada no contexto de Leitores e Escritores. A compreensão dos padrões de desempenho é fundamental para o aprimoramento contínuo do sistema em ambientes concorrentes.

Figura 1 – Tabela comparando os resultados em diferentes proporções de Leituras/Escritas

Relação de Leitores e Escritores	Total de Leituras	Total de Escrita	Tempo de Bloqueio	Tempo de Execução
50/50	5000	5000	1573 ms	5225 ms
60/40	6000	4000	1072 ms	4707 ms
55/45	5500	4500	1304 ms	4741 ms
90/10	9000	1000	80 ms	5053 ms

Aqui estão algumas observações com base nos resultados mostrados nos gráficos das Figuras 2 e 3:

- **Sensibilidade à Proporção Leitores/Escritores:** Os testes evidenciam que a eficiência do sistema está intrinsecamente ligada à proporção entre leitores e escritores. Cenários com maior predominância de leitores demonstram menor tempo de bloqueio, ressaltando a importância de considerar a dinâmica entre essas duas operações.
- **Impacto do Tempo de Bloqueio nas Proporções Equilibradas:** Notamos que em situações em que as proporções entre leitores e escritores são mais equilibradas, o tempo de bloqueio tende a aumentar. Isso sugere uma competição mais intensa pelos recursos compartilhados, destacando a necessidade de estratégias eficazes de controle de acesso.
- **Eficácia do Controle de Acesso:** A implementação das regiões críticas demonstrou ser eficaz na gestão do acesso concorrente à estrutura de dados. O sistema manteve a consistência e integridade dos dados, evidenciando a importância de mecanismos adequados para evitar conflitos.

- **Estabilidade do Tempo de Execução:** O tempo total de execução permaneceu relativamente estável, independentemente das variações nas proporções de leitores e escritores. Essa estabilidade reflete a robustez do sistema em lidar com operações concorrentes, mesmo em condições desafiadoras.
- **Oportunidades para Otimização:** Apesar do desempenho consistente, há oportunidades para otimização, especialmente em cenários com proporções equilibradas. Estratégias avançadas de controle de concorrência podem ser exploradas para reduzir o tempo de bloqueio e aprimorar ainda mais a eficiência do sistema.

Figura 2 – Gráfico comparando o tempo de bloqueio com a proporção de leitores.

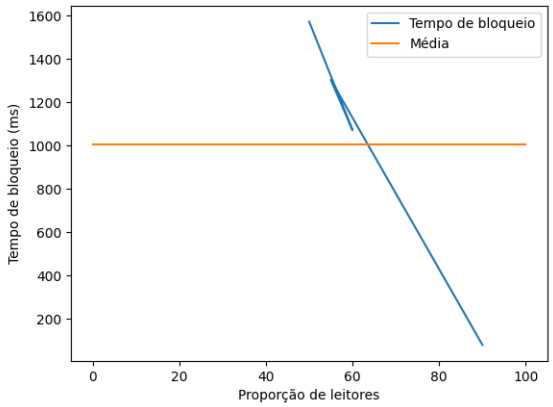
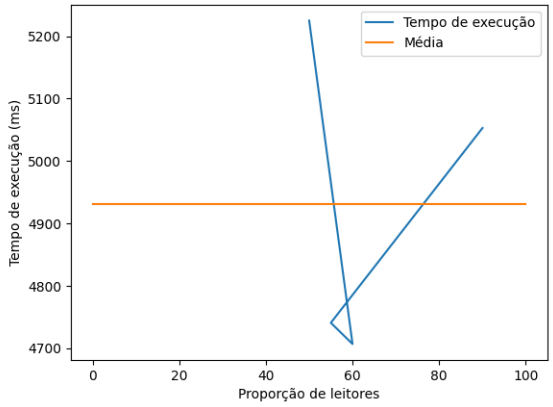


Figura 3 – Gráfico comparando o tempo de execução com a proporção de leitores,



Essas observações oferecem uma visão aprofundada do comportamento do sistema em diferentes contextos e fornecem direcionamentos valiosos para refinamentos futuros. A compreensão desses padrões é fundamental para a evolução contínua do sistema, garantindo sua adaptabilidade a diversos cenários concorrentes.

8 Conclusões

A implementação do sistema de leitores e escritores ofereceu uma visão aprofundada das dinâmicas de concorrência em ambientes de acesso compartilhado a recursos críticos. Ao explorar diferentes proporções entre leitores e escritores, pudemos observar diretamente os impactos na eficiência do sistema. A estratégia de priorização de leitores em relação a escritores revelou-se eficaz na redução de conflitos e na manutenção da integridade dos dados, contribuindo para um fluxo mais fluido de operações.

Os resultados dos testes, evidenciados pelos tempos de execução, total de leituras, escritas e tempos de bloqueio, oferecem insights valiosos sobre o desempenho do sistema em condições diversas. A estabilidade notável no tempo total de execução, independentemente das variações nas proporções leitores/escritores, destaca a resiliência do sistema diante de cargas concorrentes variadas.

Além disso, a análise das observações e resultados sugere possíveis áreas de otimização para futuros desenvolvimentos, especialmente em cenários onde a competição por recursos é mais intensa. A compreensão desses resultados é essencial para orientar melhorias contínuas no sistema, visando aprimorar sua eficiência e adaptabilidade.

Em suma, este trabalho não apenas atingiu seus objetivos, proporcionando uma implementação bem-sucedida do problema dos leitores e escritores em um ambiente concorrente, mas também ofereceu uma compreensão aprofundada das complexidades envolvidas. Os desafios enfrentados e as lições aprendidas durante o desenvolvimento destacam a importância crucial de estratégias eficazes de controle de concorrência em sistemas que compartilham recursos críticos. Essa experiência serve como base para futuras explorações e refinamentos, impulsionando o contínuo aprimoramento da eficiência em ambientes similares.

Referências

TANENBAUM, A.; BOS, H. *Introduction to Operating Systems*. [S.l.]: Pearson, 2014.

TANENBAUM, A.; BOS, H. *Operating Systems: Design and Implementation*. [S.l.]: Pearson, 2014.

TANENBAUM, A.; BOS, H. *Distributed Systems: Principles and Paradigms*. [S.l.]: Pearson, 2017.