



**Faculdade de Design,
Tecnologia e Comunicação**
Universidade Europeia

Proposta para o trabalho prático de Sistemas Operativos

Tema: Multi-threading Application



Repositório: <https://github.com/Kev1n18/Sistemas-Operativos>

Docente: Pedro Rosa

Grupo:

Wesley Augusto-20200344

Kevin Dos Santos-20210448

Eularinani Ecombe-20210329

Proposta de Projeto: Desenvolvimento de Servidor Web Multi-Threading em Java

1. Descrição do Problema e Motivação:

Com o advento da internet e o crescimento exponencial de usuários online, a demanda por aplicações web eficientes e escaláveis tornou-se uma necessidade premente. Os servidores web desempenham um papel fundamental nesse contexto, sendo responsáveis por receber, processar e responder às solicitações dos clientes de forma rápida e confiável. No entanto, conforme o número de usuários e solicitações aumenta, os servidores enfrentam desafios de desempenho e escalabilidade.

Um dos principais desafios é a capacidade de lidar com múltiplas solicitações de forma simultânea, garantindo uma experiência de usuário fluida e sem interrupções. Tradicionalmente, os servidores web utilizam uma abordagem baseada em threads para lidar com múltiplas solicitações concorrentes. No entanto, o gerenciamento eficiente dessas threads e a prevenção de condições de corrida e outros problemas de concorrência podem ser complexos e desafiadores.

Diante desse cenário, surge a necessidade de explorar técnicas avançadas de programação, como multi-threading, para desenvolver servidores web capazes de lidar com alta concorrência de forma eficiente e escalável. Este projeto visa abordar esse desafio, propondo o desenvolvimento de um **servidor web** simples, porém robusto, utilizando a linguagem de programação Java e explorando os conceitos de multi-threading para lidar com múltiplos clientes simultaneamente.

Identificação de Casos de Uso:

1. **Hospedagem de Sites:** O servidor desenvolvido pode ser utilizado para hospedar uma variedade de sites, desde páginas estáticas até aplicações web dinâmicas, garantindo uma resposta rápida e confiável às solicitações dos usuários.
2. **Aplicações Web com Alta Concorrência:** Serviços online que requerem alta concorrência, como salas de bate-papo, fóruns ou jogos multiplayer, podem se beneficiar significativamente de um servidor capaz de lidar eficientemente com múltiplas conexões simultâneas.
3. **Implementação de APIs e Serviços RESTful:** O servidor desenvolvido pode ser utilizado para implementar serviços RESTful e APIs, permitindo a comunicação eficiente entre clientes e servidores em ambientes distribuídos.

2. Descrição da Solução a Implementar:

Solução Genérica:

A proposta visa desenvolver um robusto servidor web multithreading em Java para atender às demandas crescentes por serviços web de alto desempenho. O servidor será projetado para lidar com múltiplas conexões HTTP de forma simultânea, utilizando um modelo multithreaded para processar cada solicitação de maneira independente e eficiente. Ao adotar essa abordagem, garantiremos uma distribuição eficaz dos recursos do servidor, proporcionando uma resposta rápida e confiável mesmo em situações de pico de tráfego.

Nossa solução se destacará pela sua capacidade de gerenciar eficientemente solicitações concorrentes, minimizando o tempo de espera dos clientes e oferecendo uma experiência de navegação fluída. Além disso, o servidor será configurado para servir não apenas conteúdo HTML, mas também imagens e outros recursos estáticos, ampliando sua versatilidade e utilidade para uma variedade de aplicações web.

Ao implementar esse servidor web multithreading, estaremos capacitando nossos clientes a oferecer serviços web de alta qualidade e confiabilidade, garantindo uma performance excepcional e uma experiência positiva para os usuários finais.

Enquadramento nas Áreas da Unidade Curricular:

Este projeto está alinhado com os objetivos da disciplina ao explorar conceitos avançados de programação em Java e sua aplicação na construção de sistemas distribuídos e servidores web. Ao desenvolver um servidor web multi-threaded em Java, os alunos terão a oportunidade de aprofundar seus conhecimentos em programação orientada a objetos, manipulação de threads e comunicação de rede.

Além disso, o projeto permite explorar conceitos teóricos, como protocolos de rede e protocolo HTTP, na prática, contribuindo para uma compreensão mais abrangente dos fundamentos da computação distribuída e desenvolvimento de sistemas web.

Ao final do projeto, espera-se que os alunos estejam aptos a aplicar os conceitos e técnicas aprendidos em cenários do mundo real, enfrentando desafios comuns na implementação de servidores web e sistemas distribuídos.

Requisitos Técnicos:

1. Proficiência avançada em programação Java e sólido entendimento dos conceitos de programação orientada a objetos, incluindo herança, polimorfismo e encapsulamento.
2. Conhecimento sólido dos princípios fundamentais de redes, incluindo protocolos TCP/IP e HTTP, bem como familiaridade com conceitos como requisição, resposta e cabeçalhos HTTP.
3. Capacidade de utilizar eficazmente ambientes de desenvolvimento integrado (IDEs) para Java, como Eclipse, IntelliJ IDEA, para desenvolver, depurar e testar o servidor web.
4. Experiência prévia e compreensão profunda de programação multithreaded em Java, incluindo sincronização, monitoramento de condições e práticas recomendadas para evitar condições de corrida e deadlocks.
5. Capacidade de projetar e implementar estruturas de dados e algoritmos eficientes para manipular solicitações HTTP concorrentes, garantindo o desempenho, a escalabilidade e a segurança do servidor web.
6. Familiaridade com técnicas de otimização de código e depuração de problemas de desempenho em ambientes multithreading, para garantir uma execução suave e eficiente do servidor sob diferentes condições de carga.

7. Habilidade para documentar adequadamente o código-fonte, incluindo comentários claros e concisos, e seguir boas práticas de desenvolvimento de software, como modularidade, coesão e baixo acoplamento.
8. Conhecimento e experiência em utilizar ferramentas de teste de carga, como o Apache JMeter, para avaliar o desempenho, a escalabilidade e a confiabilidade do servidor web em condições simuladas de tráfego intenso.

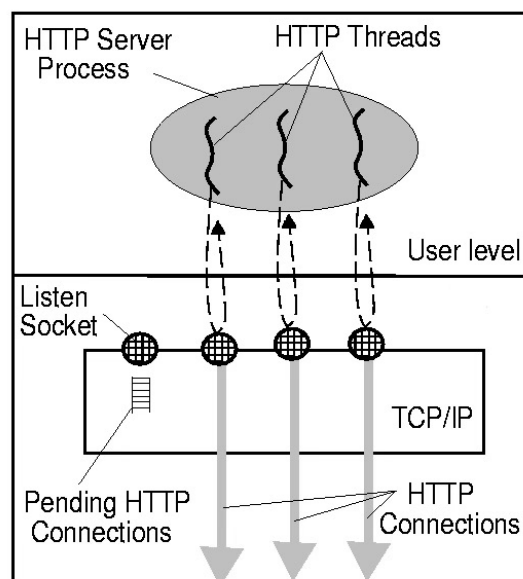
Arquitetura da Solução:

A arquitetura da solução será cuidadosamente projetada para garantir a eficiência e a confiabilidade do servidor web multithreading. No núcleo da arquitetura estará um servidor principal altamente escalável, capaz de gerenciar e coordenar conexões de clientes de forma eficiente.

Cada vez que uma conexão é estabelecida, o servidor principal dinamicamente cria uma nova thread dedicada para lidar com as solicitações desse cliente específico. Essa abordagem permite que o servidor atenda a múltiplas solicitações simultaneamente, sem sobrecarregar o sistema ou comprometer sua estabilidade.

Cada thread será responsável por gerenciar integralmente uma conexão individual, manipulando a entrada e saída de dados associados à solicitação HTTP correspondente. Isso significa que o servidor será capaz de processar solicitações de forma independente, distribuindo eficientemente a carga de trabalho entre as threads disponíveis.

Ao adotar essa arquitetura multithreading, estamos garantindo que o servidor seja altamente responsivo e capaz de lidar com uma ampla gama de cenários de uso, desde solicitações simples até cargas de tráfego intenso. Além disso, essa abordagem maximiza a utilização dos recursos do sistema, proporcionando uma resposta rápida e consistente às solicitações dos clientes.



Tecnologias a Utilizar:

- - Linguagem de programação Java para desenvolver o servidor web, utilizando bibliotecas como java.net para comunicação de rede.
- - IDE para desenvolvimento Java, como IntelliJ IDEA, escolhida por sua eficiência e recursos avançados de desenvolvimento.
- - Ferramentas de teste, incluindo JUnit para testes unitários e Apache JMeter para testes de carga, permitindo uma avaliação abrangente do desempenho e escalabilidade do servidor web em diferentes cenários de uso.

Planeamento e Calendarização:

ATIVIDADE	DURAÇÃO ESTIMADA
Pesquisa e Entendimento do Protocolo HTTP	3 dias
Estudo de Conceitos de Multi-Threading em Java	2 semanas
Projeto da arquitetura da aplicação	1 semana
Implementação do servidor web em Java	4 semanas
Realização de testes extensivos	6 dias
Documentação do processo de desenvolvimento	8 dias
Revisão e entrega do Projeto	20 de maio

Bibliografia:

- <https://leon-wtf.github.io/doc/java-concurrency-in-practice.pdf>
- [https://theswissbay.ch/pdf/Gentoomen Library/Programming/O%27Reilly Designing Series/O%27Reilly Head First Servlets and JSP.pdf](https://theswissbay.ch/pdf/Gentoomen%20Library/Programming/O%27Reilly%20Designing%20Series/O%27Reilly%20Head%20First%20Servlets%20and%20JSP.pdf)
- [Documentação oficial do Java SE.](#)
- [Documentação oficial do protocolo HTTP](#)