

ACH 2026 – Redes de Computadores

Atividade 04 : Servidor Web Multithreaded

Este projeto tem como objetivo implementar um servidor Web multithreaded, que será capaz de processar múltiplas requisições de serviços simultâneas em paralelo. O servidor Web deve ser capaz de enviar sua home page ao browser Web, efetuar registros (logs) de todas as operações realizadas, não mostrar o conteúdo de diretórios, além de permitir a configuração de filtros.

Informações gerais:

O projeto será realizado pelas duplas de trabalho da disciplina.

As atividades 2 e 3 implementam uma versão mais simples do que a aqui proposta e podem ser re-utilizadas na atividade 4. Lembre-se que as suas decisões de projeto devem estar justificadas no relatório.

Entrega do trabalho: Os alunos deverão entregar o relatório (em formato pdf via Tidia) e o código fonte do exercício-programa documentado e manual de compilação e execução.

Data limite para entrega: 24 de setembro 2016. Não serão aceitas entregas posteriores!

Especificação:

O servidor Web deve implementar a versão 1.0 do HTTP, definido na RFC-1945 [1], onde requisições HTTP separadas são enviadas para cada componente da página Web. Este servidor deverá manipular múltiplas requisições simultâneas de serviços em paralelo. Isso significa que o servidor Web é multithreaded. No thread principal, o servidor escuta uma porta fixa. Quando recebe uma requisição de conexão TCP, ele ajusta a conexão TCP através de outra porta e atende essa requisição em um thread separado.

Quando o servidor Web encontrar um erro, ele deve enviar uma mensagem de resposta com a fonte HTML apropriada, de forma que a informação do erro seja exibida na janela do browser.

Além destas funcionalidades básicas, o servidor Web deve:

1. efetuar registros (logs) de todas as operações realizadas;
2. permitir a configuração da listagem ou não de conteúdo de diretórios;
3. permitir autenticação de acesso para o diretório **restrito** (sub-diretório do diretório principal do servidor web).

Logs:

O arquivo de log deve ser escrito em formato texto, e ser acessível através do servidor web. Ele deve conter as seguintes informações (mínimas): endereço origem, porta origem, horário da requisição, conteúdo requisitado, quantidade de bytes transmitidas em resposta a requisição.

Configuração da listagem ou não de conteúdo de diretórios:

Esta característica deve ser configurável (opção de execução do programa ou arquivo de configuração). Quando usuário acessa uma url do tipo <http://host.usp.br/diretorio/>, as seguintes respostas são possíveis (ou seja, opções de configuração):

1. o conteúdo do diretório é listado em uma página html (opção que permite a listagem do diretório).
2. uma página com uma mensagem do tipo “O conteúdo do diretório não pode ser listado” é retornada ao usuário (opção que não permite a listagem do diretório).
3. uma página padrão é exibida: nesse caso, o servidor web exibe a página chamada `index.html`, e se esta não existir, exibe a mensagem de erro da opção 2.

Autenticação de acesso:

O servidor web deve as funções relativas a “Autenticação de Acesso” descritas no item 11 da RFC1945 [1]. Assim, quando o servidor receber uma requisição de acesso ao diretório **restrito** (sub-diretório do diretório principal do servidor web), ele deverá enviar mensagem ao cliente solicitando as informações de autenticação (usuário e senha).

Dicas:

1. A variável `socket` será usada para armazenar uma referência ao socket de conexão. O servidor http faz uso da camada de transporte! Portanto, NÃO utilize um projeto da categoria Web no NetBeans ou equivalente na sua ferramenta de desenvolvimento.
2. Lembre-se que de acordo com a especificação HTTP, é necessário terminar cada linha da mensagem de resposta do servidor com um *carriage return* (CR) e um *line feed* (LF).
3. Existem três partes para a mensagem de resposta: a linha de status, os cabeçalhos da resposta e o corpo da entidade. A linha de status e os cabeçalhos da resposta são terminados pela sequência de caracteres CRLF. No caso de uma requisição de um arquivo não existente, o servidor deve retornar *404 Not Found* na linha de status da mensagem de resposta e incluir uma mensagem de erro no formato de um documento HTML no corpo da entidade.
4. Enquanto você desenvolve o código, pode testar seu servidor a partir de um browser Web. Mas lembre que o servidor não está atendendo pela porta padrão 80; logo, é preciso especificar o número da porta junto à URL que você fornecer ao browser Web. Por exemplo, se o nome da sua máquina é `host.usp.br`, seu servidor está escutando a porta 6789, e você quer recuperar o arquivo `index.html`, então deve especificar ao browser a seguinte URL: <http://host.usp.br:6789/index.html>. Se você omitir “:6789”, o browser irá assumir a porta 80, que, provavelmente, não terá nenhum servidor à escuta. Você pode escolher qualquer porta acima de 1024.

Relatório:

O relatório deverá possuir formato condizente com um relatório científico, contendo uma introdução, desenvolvimento das seções, conclusão e referências bibliográficas.

O relatório deve conter a descrição da arquitetura do servidor, bem como justificativas para as decisões de projeto. Além disso, também deve incluir instruções de uso ou configurações, se necessário.

Referências:

[1] “RFC 1945 - Hypertext Transfer Protocol – HTTP/1.0”. T. Berners-Lee, R. Fielding, H. Frystyk. May 1996. Disponível em <http://www.faqs.org/rfcs/rfc1945.html>