

Um Inspetor HTTP baseado em Proxy Server

Universidade de Brasília

Teleinformática e Redes 2

Turma A: 2018/2
Professor: João Gondim

Grupo:
Gabriel Augusto Correia [15/0126000]
Luís Eduardo Luz Silva [15/0137885]

1. Apresentação teórica

1.1 Proxy Server Web

No contexto deste trabalho, o Proxy Server Web é um host intermediário entre o cliente e o servidor, que obtém as requisições do cliente, envia para o servidor, obtém a resposta e redireciona para o cliente, tendo a capacidade de alteração ou barramento em qualquer etapa descrita da comunicação.

1.2 Transfer Control Protocol (TCP)

O TCP é um protocolo da camada de transporte sobre o protocolo de rede IP que possui um serviço que garante entrega ordenada de pacotes, sendo portanto confiável e a escolha para dados que necessitem de integridade e sincronia, em particular páginas web e seus objetos.

1.3 HyperText Transfer Protocol (HTTP)

O HTTP é um protocolo simples da camada de aplicação do tipo request-response que normalmente é executado sobre TCP. Especifica que mensagens clientes enviam a servidores e que resposta os servidores podem enviar aos clientes. Ambas as mensagens são codificadas em texto ASCII.

2. Arquitetura do sistema

O sistema foi arquitetado por uma abordagem funcional: comunicação, parser e spider. Os pares de arquivos que implementam essas funcionalidades são, respectivamente, `comm.h`, `comm.cpp`; `parser.h`, `parser.cpp` e `spider.h`, `spider.cpp`. Para detalhes da implementação de cada função, ver a Documentação.

2.1 Parser (`parser.h`, `parser.cpp`)

O módulo de parser executa funções de tratamento de texto focadas nas mensagens HTTP. Em específico, há funções para obter a URL, host e arquivo do pedido, formatação dos dados para um formato conveniente ao spider e obtenção do caminho do arquivo. É utilizado por todos os demais módulos.

2.2 Comm (`comm.g`, `comm.cpp`)

O módulo `comm` (de comunicação) possui funções relacionadas ao estabelecimento de conexão com o servidor e gerenciamento dos pedidos, ou seja, envio, redirecionamento, transferência, obtenção do endereço IP, etc. É utilizado pelo `spider\dump`.

2.3 Spider (`spider.h`, `spider.cpp`)

O módulo `spider`, intuitivamente, executa as funcionalidades de spider e dump. Nela são definidas a estrutura de árvore do spider, as funções de manipulação de grafos (adição de nó, busca, remoção) e a função `spider`, que faz a árvore hierárquica dos arquivos do site escolhido. O dump se implementa utilizando-se o spider para gerar uma lista de arquivos para download.

2.4 Main (`main.cpp`)

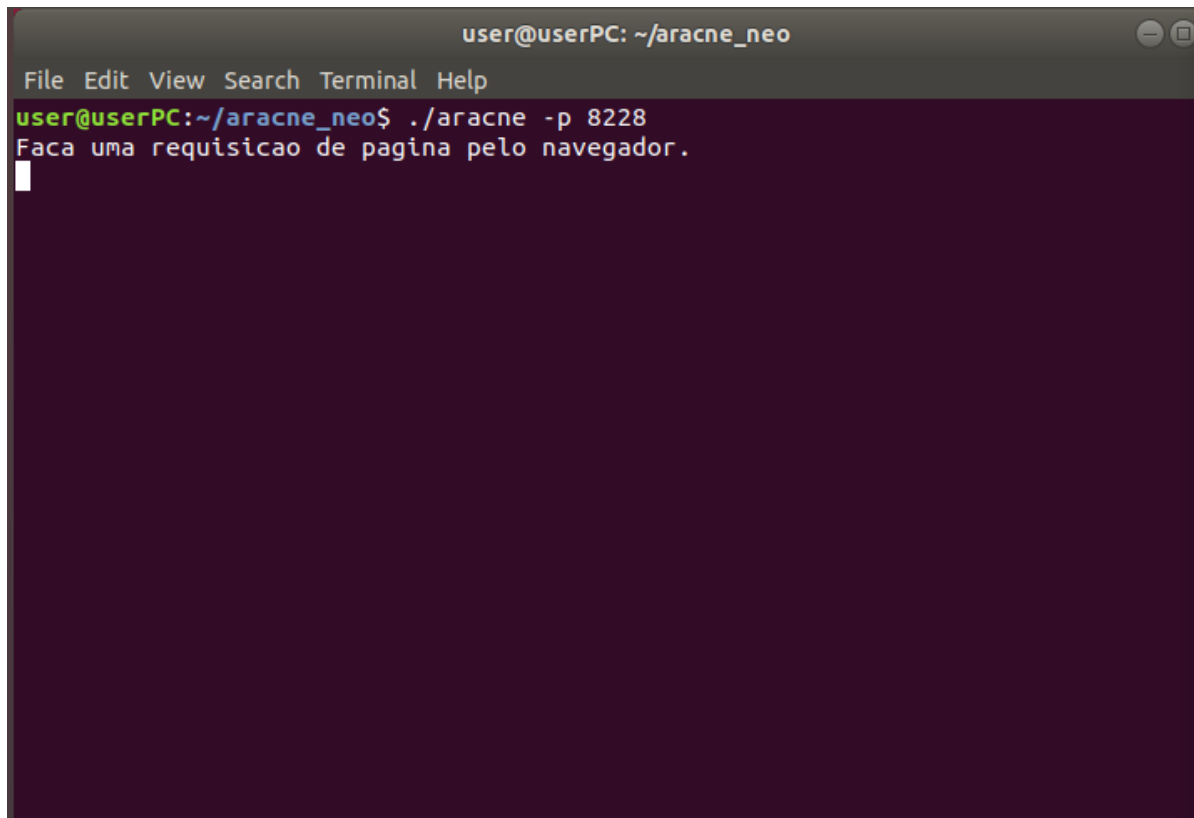
Ponto de entrada do programa. Efetua o loop principal, abrindo uma porta (escolhida pelo usuário ou 8228 por default) que escuta as requisições do browser, executa as funções de proxy, executa o `spider\dump` e aguarda por outra requisição.

3. Documentação

A documentação está na pasta Doxygen e foi gerado a partir do código.

4. Funcionamento aracne

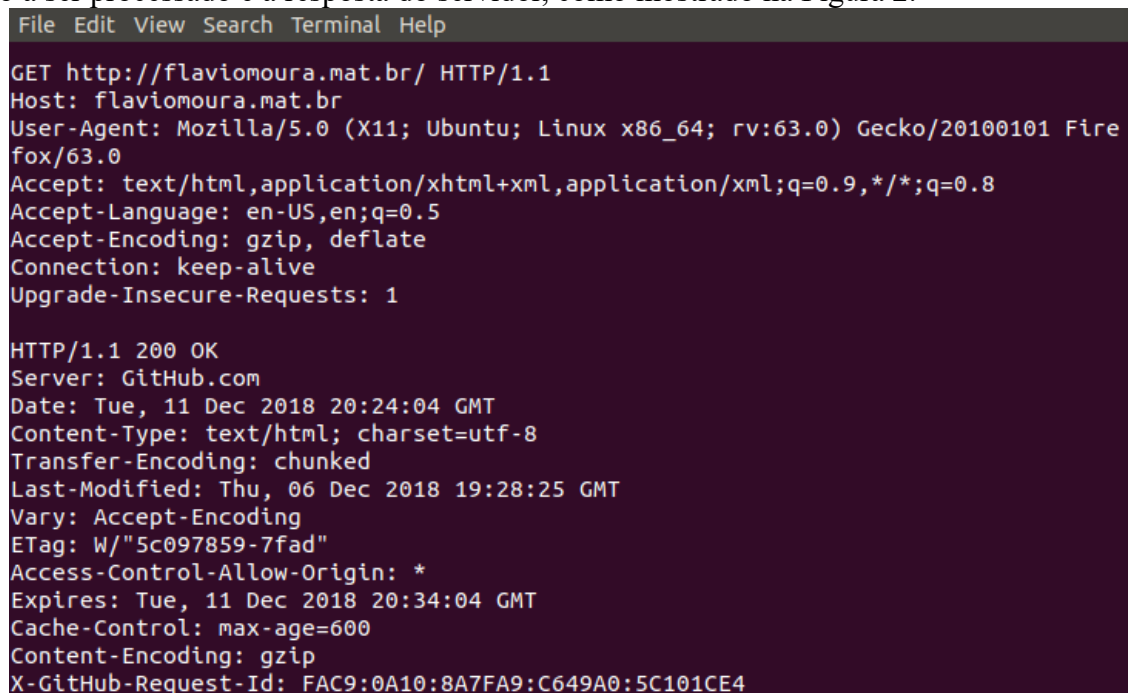
Para invocar o programa, basta chama-lo sem comandos adicionais (porta padrão 8228) ou adicionar o comando -p (ou -P) para selecionar a porta, como mostrado na Figura 1. Ao chamar o programa, ele aguarda uma requisição do browser, que deve estar com proxy configurado na porta desejada (no caso, 8228).

A terminal window titled 'user@userPC: ~/aracne_neo' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'user@userPC:~/aracne_neo\$'. The command './aracne -p 8228' has been entered. Below the command, the text 'Faca uma requisicao de pagina pelo navegador.' is displayed. A white cursor is positioned on the line following the command.

```
user@userPC: ~/aracne_neo
File Edit View Search Terminal Help
user@userPC:~/aracne_neo$ ./aracne -p 8228
Faca uma requisicao de pagina pelo navegador.
█
```

Figura 1: chamada ao programa e espera por requisição do browser.

Utilizando o browser para fazer um pedido HTTP (e não HTTPS!), o aracne irá exibir o pedido a ser processado e a resposta do servidor, como mostrado na Figura 2.

A terminal window showing an HTTP request and response. The request is a GET to http://flaviomoura.mat.br/. The response is an HTTP/1.1 200 OK from GitHub.com, with various headers including Date, Content-Type, Transfer-Encoding, Last-Modified, Vary, ETag, Access-Control-Allow-Origin, Expires, Cache-Control, Content-Encoding, and X-GitHub-Request-Id.

```
File Edit View Search Terminal Help
GET http://flaviomoura.mat.br/ HTTP/1.1
Host: flaviomoura.mat.br
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:63.0) Gecko/20100101 Fire
fox/63.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Server: GitHub.com
Date: Tue, 11 Dec 2018 20:24:04 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Last-Modified: Thu, 06 Dec 2018 19:28:25 GMT
Vary: Accept-Encoding
ETag: W/"5c097859-7fad"
Access-Control-Allow-Origin: *
Expires: Tue, 11 Dec 2018 20:34:04 GMT
Cache-Control: max-age=600
Content-Encoding: gzip
X-GitHub-Request-Id: FAC9:0A10:8A7FA9:C649A0:5C101CE4
```

Figura 2: exibição da requisição e resposta.

Então, o programa executa o spider e, após processamento, mostra a árvore hierárquica gerada, como na Figura 3.

```
File Edit View Search Terminal Help
http://flaviomoura.mat.br/lc1.html
    http://flaviomoura.mat.br/files/org.css
    http://flaviomoura.mat.br/files/lc1/planodeEnsino2018-2.pdf
    http://flaviomoura.mat.br/files/lc1/inducaao.pdf
    http://flaviomoura.mat.br/files/lc1/lista1-2018-2.pdf
    http://flaviomoura.mat.br/files/lc1/lista1-2018-2gabarito.pdf
    http://flaviomoura.mat.br/files/lc1/prova1-2018-2gabarito.pdf
    http://flaviomoura.mat.br/files/lc1/planodeEnsino2018-2.pdf
    http://flaviomoura.mat.br/files/lc1/lista2-2018-2.pdf
    http://flaviomoura.mat.br/files/lc1/lista2-2018-2gabarito.pdf
    http://flaviomoura.mat.br/files/lc1/prova2-2018-2gabarito.pdf
    http://flaviomoura.mat.br/files/lc1/inducaao.pdf
    http://flaviomoura.mat.br/files/lc1/exercicio1.pvs
    http://flaviomoura.mat.br/files/lc1/lista1-2018-2.pdf
    http://flaviomoura.mat.br/files/lc1/lista1-2018-2gabarito.pdf
    http://flaviomoura.mat.br/files/lc1/prova1-2018-2gabarito.pdf
http://flaviomoura.mat.br/paa.html
    http://flaviomoura.mat.br/files/org.css
    http://flaviomoura.mat.br/files/PAA/planoEnsino2018-2.pdf
    http://flaviomoura.mat.br/files/PAA/notasdeaulas.pdf
    http://flaviomoura.mat.br/files/PAA/lista1-2018-2.pdf
    http://flaviomoura.mat.br/files/PAA/lista1-2018-2gabarito.pdf
    http://flaviomoura.mat.br/files/PAA/prova1-2018-2gabarito.pdf
```

Figura 3: execução do spider, hierarquia representada por indentação.

Por fim, o programa executa o dump dos arquivos listados no spider, como mostrado na Figura 4. Após isso, o programa volta para o estado de escuta passiva da Figura 1, aguardando a próxima requisição do cliente.

```
File Edit View Search Terminal Help
Downloading: http://flaviomoura.mat.br/files/fmoura.jpg
.....
Downloading: http://flaviomoura.mat.br/files/vfp/vfp-2018-2-divulgacao.pdf
.....
Downloading: http://flaviomoura.mat.br/files/XIVSeminarioInformal.pdf
.....
Downloading: http://flaviomoura.mat.br/files/FormalProofs4U.pdf
...
Downloading: http://flaviomoura.mat.br/files/alcs.html
.....
Downloading: http://flaviomoura.mat.br/files/livroIaC.html
....
Downloading: http://flaviomoura.mat.br/files/igpl14.html
.....
Downloading: http://flaviomoura.mat.br/files/jal08.pdf
.....
Downloading: http://flaviomoura.mat.br/files/jal08.html
....
Downloading: http://flaviomoura.mat.br/files/jancl06.pdf
```

Figura 4: dump.

Ressalvas: o programa permite a inspeção mas não a edição das mensagens\pacotes HTTP. A implementação de uma interface gráfica foi abandonada no decorrer do projeto em vista da priorização da execução correta das funcionalidades de rede.

5. Bibliografia

- [1] KUROSE, J. F.; ROSS, K. W., Computer networking – A Top-Down Approach, 7th Ed., Pearson, 2017.
- [2] DONAHOO, M. J.; CALVERT, K. L., TCP/IP Sockets in C, 2nd Ed., Morgan Kaufmann, 2009.