

Projeto de TR2 - Aracne

1.0

Generated by Doxygen 1.8.13

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	dump.cpp File Reference	3
2.1.1	Function Documentation	3
2.1.1.1	cutHead()	3
2.1.1.2	dump()	4
2.1.1.3	fixRefs()	4
2.1.1.4	generateMap()	4
2.1.1.5	makeDump()	5
2.2	inspector.cpp File Reference	5
2.2.1	Macro Definition Documentation	6
2.2.1.1	MAXRCVLEN	6
2.2.2	Function Documentation	6
2.2.2.1	inspector()	6
2.2.2.2	readBinaryFile()	6
2.2.2.3	readTextFile()	7
2.2.2.4	writeFile()	7
2.2.3	Variable Documentation	7
2.2.3.1	fml	8
2.3	main.cpp File Reference	8
2.3.1	Function Documentation	8
2.3.1.1	main()	8

2.4	proxy.cpp File Reference	8
2.4.1	Macro Definition Documentation	9
2.4.1.1	MAXRCVLEN	9
2.4.2	Function Documentation	9
2.4.2.1	proxy()	9
2.5	receive.cpp File Reference	9
2.5.1	Function Documentation	10
2.5.1.1	createNewSocket()	10
2.5.1.2	freeMemory()	10
2.6	request.cpp File Reference	10
2.6.1	Function Documentation	11
2.6.1.1	getHostValue()	11
2.6.1.2	makeRequest()	11
2.7	spyder.cpp File Reference	11
2.7.1	Function Documentation	12
2.7.1.1	buildReference()	12
2.7.1.2	generateTree()	12
2.7.1.3	isHTML()	13
2.7.1.4	isReallyHTML()	13
2.7.1.5	searchChildren()	14
2.7.1.6	spyder()	14
	Index	15

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

dump.cpp	3
inspector.cpp	5
main.cpp	8
proxy.cpp	8
receive.cpp	9
request.cpp	10
spyder.cpp	11

Chapter 2

File Documentation

2.1 dump.cpp File Reference

```
#include "connection.hpp"
```

Functions

- void [makeDump](#) (string baseURL)
- int [dump](#) (set< string > requests, string baseURL)
- void [generateMap](#) (map< string, string > &mapRefs, set< string > &requests, string baseURL)
- string [cutHead](#) (string serverRequest)
- void [fixRefs](#) (string &serverResponse, map< string, string > &mapRefs)

2.1.1 Function Documentation

2.1.1.1 cutHead()

```
string cutHead (  
    string serverRequest )
```

Função que corta o cabeçalho e retorna a stream de dados da resposta do servidor.

Parameters

<i>serverRequest</i>	String que contém a resposta vinda do servidor.
----------------------	---

Returns

string: Uma string com os dados.

Definition at line 127 of file dump.cpp.

2.1.1.2 dump()

```
int dump (
    set< string > requests,
    string baseUrl )
```

Função reliza de fato o resgate das requisições passadas como parâmetro

Parameters

<i>requests</i>	Set com requisições que deve ser feitas em baseUrl.
<i>baseUrl</i>	Endereço base onde as requisições serão feitas. Ex: www.ba.gov.br.

Returns

int: Um inteiro indicando sucesso das requisições feitas. Ele pode ser ignorado.

Definition at line 15 of file dump.cpp.

2.1.1.3 fixRefs()

```
void fixRefs (
    string & serverResponse,
    map< string, string > & mapRefs )
```

Função que corrige as referências no HTML apontando para o local correto.

Parameters

<i>&serverResponse</i>	Referência à uma string que contém a resposta do servidor (deve ser um HTML).
<i>&mapRefs</i>	Referência à um Map que contém as referências corretas.

Definition at line 143 of file dump.cpp.

2.1.1.4 generateMap()

```
void generateMap (
    map< string, string > & mapRefs,
    set< string > & requests,
    string baseUrl )
```

Função que realiza o mapeamento das referências para as referências corretas que serão colocadas no html depois de baixado.

Parameters

<i>&mapRefs</i>	Referência para um Map com requisições que devem ser mapeadas.
<i>&requests</i>	Referência para um Set que contém as requisições que devem ser feitas em baseUrl.
<i>baseUrl</i>	Endereço base onde as requisições serão feitas. Ex: www.unb.br.

Definition at line 90 of file dump.cpp.

2.1.1.5 makeDump()

```
void makeDump (
    string baseUrl )
```

Função base de Dump. Ela realiza um spider seguido de um dump. O resultado é colocado na pasta raiz dentro de uma pasta com o nome passado como parâmetro

Parameters

<i>baseUrl</i>	URL base do domínio desejado.
----------------	-------------------------------

Definition at line 5 of file dump.cpp.

2.2 inspector.cpp File Reference

```
#include "connection.hpp"
```

Macros

- `#define MAXRCVLEN 2000`

Functions

- int `inspector` (int PORTNUM)
- std::vector< unsigned char > `readBinaryFile` (string filename)
- std::string `readTextFile` (string path)
- bool `writeFile` (string path, vector< unsigned char > dados)

Variables

- struct freeMemoryList `fml`

2.2.1 Macro Definition Documentation

2.2.1.1 MAXRCVLEN

```
#define MAXRCVLEN 2000
```

Definition at line 3 of file inspector.cpp.

2.2.2 Function Documentation

2.2.2.1 inspector()

```
int inspector (  
    int PORTNUM )
```

Função que realiza inspeção da requisição vinda do browser e da resposta do servidor

Parameters

<i>PORTNUM</i>	Número da porta que será aberta para receber requisições vindas do browser.
----------------	---

Returns

int: Um inteiro indicando sucesso das requisições feitas. Ele pode ser ignorado.

Definition at line 8 of file inspector.cpp.

2.2.2.2 readBinaryFile()

```
std::vector<unsigned char> readBinaryFile (  
    string filename )
```

Função que faz a leitura do binária do arquivo passada como parâmetro

Parameters

<i>filename</i>	Uma string que apontando o arquivo à ser lido
-----------------	---

Returns

std::vector<unsigned char>: Um vetor de unsigned char com o bytes lidos do arquivo.

Definition at line 80 of file inspector.cpp.

2.2.2.3 readTextFile()

```
std::string readTextFile (
    string path )
```

Função que faz a leitura em modo de texto do arquivo passada como parâmetro.

Parameters

<i>path</i>	Uma string que apontando o arquivo à ser lido.
-------------	--

Returns

std::string: Um string contendo o conteúdo do arquivo.

Definition at line 113 of file inspector.cpp.

2.2.2.4 writeFile()

```
bool writeFile (
    string path,
    vector< unsigned char > dados )
```

Função que faz a escrita dos dados passados como parâmetro em path.

Parameters

<i>path</i>	Uma string que apontando o arquivo à ser escrito.
<i>dados</i>	Um vector de unsigned char com os dados à serem escritos no arquivo.

Returns

bool: Um booleano indicando sucesso na escrito arquivo.

Definition at line 127 of file inspector.cpp.

2.2.3 Variable Documentation

2.2.3.1 fml

```
struct freeMemoryList fml
```

Definition at line 5 of file inspector.cpp.

2.3 main.cpp File Reference

```
#include "connection.hpp"
```

Functions

- int [main](#) (int argc, char *argv[])

2.3.1 Function Documentation

2.3.1.1 main()

```
int main (  
    int argc,  
    char * argv[ ] )
```

Função principal do projeto e que liga todos os módulos de acordo com a opção escolhida pelo usuário

Parameters

<i>argc</i>	
<i>argv</i>	

Returns

int

Definition at line 5 of file main.cpp.

2.4 proxy.cpp File Reference

```
#include "connection.hpp"
```

Macros

- `#define MAXRCVLEN 2000`

Functions

- `int proxy (int PORTNUM)`

2.4.1 Macro Definition Documentation

2.4.1.1 MAXRCVLEN

```
#define MAXRCVLEN 2000
```

Definition at line 3 of file proxy.cpp.

2.4.2 Function Documentation

2.4.2.1 proxy()

```
int proxy (  
    int PORTNUM )
```

Função do proxy que recebe as requisições vindas do browser e passa imediatamente para o socket ligado ao servidor sem a possibilidade de editar, assim como sua resposta.

Parameters

<i>PORTNUM</i>	Um inteiro dizendo qual porta será aberta para receber requisições do browser.
----------------	--

Returns

int: Um inteiro indicando sucesso na execução do proxy.

Definition at line 7 of file proxy.cpp.

2.5 receive.cpp File Reference

```
#include "connection.hpp"
```

Functions

- int [createNewSocket](#) (uint16_t portNum, uint16_t parallelConnections)
- void [freeMemory](#) ()

2.5.1 Function Documentation

2.5.1.1 createNewSocket()

```
int createNewSocket (
    uint16_t portNum,
    uint16_t parallelConnections )
```

Função que cria um socket para escutar as requisições vindas do browser.

Parameters

<i>portNum</i>	Porta que será aberta para receber requisições.
<i>parallelConnections</i>	Número de conexões em paralelo que podem ser recebidas do browser

Returns

int: Um inteiro com o descriptor do socket.

Definition at line 3 of file receive.cpp.

2.5.1.2 freeMemory()

```
void freeMemory ( )
```

Função que libera a memória após a execução do programa evitando memory leak.

Definition at line 41 of file receive.cpp.

2.6 request.cpp File Reference

```
#include "connection.hpp"
```

Functions

- vector< unsigned char > [makeRequest](#) (std::string msg_string)
- std::string [getHostValue](#) (std::string msg_string)

2.6.1 Function Documentation

2.6.1.1 getHostValue()

```
std::string getHostValue (
    std::string msg_string )
```

Função que resgata o valor do campo Host para fazer um query DNS em busca do IP de origem. O campo Host precisa ser passado na requisição HTTP.

Parameters

<i>msg_string</i>	Uma string que será inspecionada para extrair o valor de Host.
-------------------	--

Returns

std::string: Uma string que contém o valor do campo.

Definition at line 51 of file request.cpp.

2.6.1.2 makeRequest()

```
vector<unsigned char> makeRequest (
    std::string msg_string )
```

Função que realiza a requisição no servidor de origem.

Parameters

<i>msg_string</i>	Uma string que será passada ao socket para enviar ao servidor.
-------------------	--

Returns

vector <unsigned char>="": Um vector de unsigned char com a resposta recebida.

Definition at line 5 of file request.cpp.

2.7 spyder.cpp File Reference

```
#include "connection.hpp"
#include "spyder.hpp"
```

Functions

- set< string > [spyder](#) (string baseURL)
- void [buildReference](#) (set< string > &result, string response, string baseURL)
- bool [isHTML](#) (string url, string baseURL)
- bool [isReallyHTML](#) (string url, string baseURL)
- set< string > [searchChildren](#) (string url, string baseURL)
- Tree [generateTree](#) (string baseURL, int levels)

2.7.1 Function Documentation

2.7.1.1 buildReference()

```
void buildReference (
    set< string > & result,
    string response,
    string baseURL )
```

<Insere em result os arquivos/diretórios encontrados em response>

Parameters

<i>result</i>	Endereço do set onde serão inseridas as referências.
<i>response</i>	resposta obtida do request.
<i>baseURL</i>	URL base do domínio desejado.

Definition at line 26 of file spyder.cpp.

2.7.1.2 generateTree()

```
Tree generateTree (
    string baseURL,
    int levels )
```

Gera a árvore hipertextual.

Parameters

<i>baseURL</i>	URL base do domínio desejado.
<i>levels</i>	Número de níveis máximos desejados para a árvore..

Returns

Tree: Árvore gerada.

Definition at line 204 of file spyder.cpp.

2.7.1.3 isHTML()

```
bool isHTML (
    string url,
    string baseURL )
```

Inspeciona um cabeçalho para saber se um caminho é HTML para saber se deve ser inspecionado.

Parameters

<i>url</i>	Url da referência.
<i>baseURL</i>	URL base do domínio desejado.

Returns

bool: Indica se é ou não HTML.

Definition at line 129 of file spyder.cpp.

2.7.1.4 isReallyHTML()

```
bool isReallyHTML (
    string url,
    string baseURL )
```

<Verifica se o cabeçalho da url já foi inspecionado para retornar a informação se a url é ou não um HTML.>

Parameters

<i>url</i>	Url da referência.
<i>baseURL</i>	URL base do domínio desejado.

Returns

bool: Indica se é ou não HTML.

Definition at line 157 of file spyder.cpp.

2.7.1.5 searchChildren()

```
set<string> searchChildren (
    string url,
    string baseUrl )
```

<Busca pelas referências um nível exatamente abaixo da url.>

Parameters

<i>url</i>	Url que será buscada.
<i>baseUrl</i>	URL base do domínio desejado.

Returns

set<string>: Conjunto de referências encontradas na url.

Definition at line 177 of file spyder.cpp.

2.7.1.6 spyder()

```
set<string> spyder (
    string baseUrl )
```

Função principal do spyder.

Parameters

<i>baseUrl</i>	URL base do domínio desejado.
----------------	-------------------------------

Returns

set<string>: Set com os nomes das referências encontradas.

Definition at line 6 of file spyder.cpp.

Index

- buildReference
 - spyder.cpp, 12
- createNewSocket
 - receive.cpp, 10
- cutHead
 - dump.cpp, 3
- dump
 - dump.cpp, 4
- dump.cpp, 3
 - cutHead, 3
 - dump, 4
 - fixRefs, 4
 - generateMap, 4
 - makeDump, 5
- fixRefs
 - dump.cpp, 4
- fml
 - inspector.cpp, 7
- freeMemory
 - receive.cpp, 10
- generateMap
 - dump.cpp, 4
- generateTree
 - spyder.cpp, 12
- getHostValue
 - request.cpp, 11
- inspector
 - inspector.cpp, 6
- inspector.cpp, 5
 - fml, 7
 - inspector, 6
 - MAXRCVLEN, 6
 - readBinaryFile, 6
 - readTextFile, 7
 - writeFile, 7
- isHTML
 - spyder.cpp, 13
- isReallyHTML
 - spyder.cpp, 13
- MAXRCVLEN
 - inspector.cpp, 6
 - proxy.cpp, 9
- main
 - main.cpp, 8
- main.cpp, 8
 - main, 8
- makeDump
 - dump.cpp, 5
- makeRequest
 - request.cpp, 11
- proxy
 - proxy.cpp, 9
- proxy.cpp, 8
 - MAXRCVLEN, 9
 - proxy, 9
- readBinaryFile
 - inspector.cpp, 6
- readTextFile
 - inspector.cpp, 7
- receive.cpp, 9
 - createNewSocket, 10
 - freeMemory, 10
- request.cpp, 10
 - getHostValue, 11
 - makeRequest, 11
- searchChildren
 - spyder.cpp, 13
- spyder
 - spyder.cpp, 14
- spyder.cpp, 11
 - buildReference, 12
 - generateTree, 12
 - isHTML, 13
 - isReallyHTML, 13
 - searchChildren, 13
 - spyder, 14
- writeFile
 - inspector.cpp, 7