

ddt - RCI

Generated by Doxygen 1.8.9.1

Tue Mar 10 2015 15:54:28



# Contents

<b>1</b>	<b>Data Structure Index</b>	<b>1</b>
1.1	Data Structures . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Data Structure Documentation</b>	<b>5</b>
3.1	args_parse_options Struct Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Field Documentation . . . . .	5
3.1.2.1	argc . . . . .	5
3.1.2.2	argv . . . . .	5
3.1.2.3	startup_data . . . . .	5
3.2	startup_data Struct Reference . . . . .	5
3.2.1	Field Documentation . . . . .	6
3.2.1.1	dest_size . . . . .	6
3.2.1.2	destination . . . . .	6
3.2.1.3	family . . . . .	6
3.2.1.4	ringport . . . . .	6
3.3	transversal_data Struct Reference . . . . .	6
3.3.1	Field Documentation . . . . .	6
3.3.1.1	ext_addr . . . . .	6
3.3.1.2	ring . . . . .	6
3.3.1.3	startup_data . . . . .	6
3.3.1.4	t . . . . .	7
3.3.1.5	u . . . . .	7
<b>4</b>	<b>File Documentation</b>	<b>9</b>
4.1	include/common.h File Reference . . . . .	9
4.2	include/net_common.h File Reference . . . . .	9
4.2.1	Function Documentation . . . . .	9
4.2.1.1	getsockaddr . . . . .	9

4.3	include/net_udp.h File Reference	10
4.3.1	Function Documentation	10
4.3.1.1	createsocket_udp	10
4.3.1.2	getudpdest	10
4.4	include/options.h File Reference	10
4.4.1	Function Documentation	10
4.4.1.1	parse_options	10
4.5	src/main.c File Reference	11
4.5.1	Detailed Description	11
4.5.2	Function Documentation	11
4.5.2.1	main	11
4.6	src/net_common.c File Reference	11
4.6.1	Macro Definition Documentation	12
4.6.1.1	_XOPEN_SOURCE	12
4.6.2	Function Documentation	12
4.6.2.1	getsockaddr	12
4.7	src/net_udp.c File Reference	12
4.7.1	Function Documentation	12
4.7.1.1	createsocket_udp	12
4.7.1.2	getudpdest	12
4.8	src/options.c File Reference	13
4.8.1	Detailed Description	13
4.8.2	Macro Definition Documentation	13
4.8.2.1	_XOPEN_SOURCE	13
4.8.3	Function Documentation	13
4.8.3.1	parse_options	13
4.8.3.2	usage	13
	<b>Index</b>	<b>15</b>

# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">args_parse_options</a>	Estrutura com os argumentos para a função <a href="#">parse_options</a> . . . . .	5
<a href="#">startup_data</a>	Estrutura com os dados a ser recolhidos pela linha de comandos . . . . .	5
<a href="#">transversal_data</a>	Estrutura com as referências necessárias aos vários módulos do projecto . . . . .	6



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

include/ <a href="#">common.h</a>	Cabeçalho com declarações das estruturas de dados transversais aos vários módulos do projecto . . . . .	9
include/ <a href="#">net_common.h</a>	Cabeçalhos e protótipos associados às funções relacionadas com a rede mas não associadas a nenhum protocolo em específico . . . . .	9
include/ <a href="#">net_udp.h</a>	Cabeçalhos e protótipos associados às funções relacionadas com o protocolo UDP . . . . .	10
include/ <a href="#">options.h</a>	Cabeçalhos e protótipos associados ao módulo de processamento das opções de linha de comandos . . . . .	10
src/ <a href="#">main.c</a>	Ficheiro principal do projecto . . . . .	11
src/ <a href="#">net_common.c</a>	Ficheiro responsável pela implementação das funções relacionadas com rede mas não associadas a nenhum dos protocolos utilizados . . . . .	11
src/ <a href="#">net_udp.c</a>	Ficheiro responsável pela implementação das funções de rede associadas ao protocolo UDP . . . . .	12
src/ <a href="#">options.c</a>	Ficheiro responsável pela interpretação dos argumentos passados via linha de comandos . . . . .	13





## Chapter 3

# Data Structure Documentation

### 3.1 args\_parse\_options Struct Reference

Estrutura com os argumentos para a função [parse\\_options](#).

```
#include <options.h>
```

#### Data Fields

- `int * argc`  
*Ponteiro para o local em memória do contador de argumentos.*
- `char *** argv`  
*Ponteiro para o local em memória dos argumentos.*
- `struct startup\_data * startup\_data`  
*Ponteiro para uma das estruturas transversais ao projecto.*

#### 3.1.1 Detailed Description

Esta estrutura,

#### 3.1.2 Field Documentation

3.1.2.1 `int* args_parse_options::argc`

3.1.2.2 `char*** args_parse_options::argv`

3.1.2.3 `struct startup_data* args_parse_options::startup_data`

The documentation for this struct was generated from the following file:

- `include/options.h`

### 3.2 startup\_data Struct Reference

Estrutura com os dados a ser recolhidos pela linha de comandos.

```
#include <common.h>
```

## Data Fields

- char [ringport](#) [16]  
*Porto do servidor TCP.*
- int [family](#)  
*Família da socket.*
- struct sockaddr \* [destination](#)  
*Estrutura de destino para as mensagens UDP.*
- socklen\_t [dest\\_size](#)  
*Tamanho da estrutura de destino para as mensagens UDP.*

### 3.2.1 Field Documentation

3.2.1.1 socklen\_t startup\_data::dest\_size

3.2.1.2 struct sockaddr\* startup\_data::destination

3.2.1.3 int startup\_data::family

3.2.1.4 char startup\_data::ringport[16]

The documentation for this struct was generated from the following file:

- include/[common.h](#)

## 3.3 transversal\_data Struct Reference

Estrutura com as referências necessárias aos vários módulos do projecto.

```
#include <common.h>
```

## Data Fields

- int [u](#)  
*Socket UDP.*
- int [t](#)  
*Socket TCP.*
- int [ring](#)  
*Anel a ser utilizado.*
- struct [startup\\_data](#) [startup\\_data](#)  
*Parâmetros da linha de comandos.*
- char [ext\\_addr](#) [40]  
*IP externo do servidor.*

### 3.3.1 Field Documentation

3.3.1.1 char transversal\_data::ext\_addr[40]

3.3.1.2 int transversal\_data::ring

3.3.1.3 struct startup\_data transversal\_data::startup\_data

3.3.1.4 int transversal\_data::t

3.3.1.5 int transversal\_data::u

The documentation for this struct was generated from the following file:

- include/[common.h](#)



## Chapter 4

# File Documentation

### 4.1 include/common.h File Reference

Cabeçalho com declarações das estruturas de dados transversais aos vários módulos do projecto.

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/ip.h>
```

#### Data Structures

- struct [startup\\_data](#)  
*Estrutura com os dados a ser recolhidos pela linha de comandos.*
- struct [transversal\\_data](#)  
*Estrutura com as referências necessárias aos vários módulos do projecto.*

### 4.2 include/net\_common.h File Reference

Cabeçalhos e protótipos associados às funções relacionadas com a rede mas não associadas a nenhum protocolo em específico.

#### Functions

- void [getsockaddr](#) (char \*node, char \*service, int \*family, socklen\_t \*size, struct sockaddr \*\*sockaddress, int protocol)  
*Função de leitura de endereços e portos e criação de estruturas de destino.*

#### 4.2.1 Function Documentation

**4.2.1.1 void getsockaddr ( char \* node, char \* service, int \* family, socklen\_t \* size, struct sockaddr \*\* sockaddress, int protocol )**

Esta função recebe um endereço e

A função getaddrinfo() é utilizada em vez da função gethostbyname(), por uma questão de preferência pessoal e também pelo facto da função getaddrinfo() estar especificada como thread-safe (a função gethostbyname() só é thread-safe em algumas implementações). Os detalhes da função getaddrinfo() podem ser consultados em [RFC 2553](#).

## 4.3 include/net\_udp.h File Reference

Cabeçalhos e protótipos associados às funções relacionadas com o protocolo UDP.

### Functions

- void [getudpdest](#) (char \*node, char \*service, struct [startup\\_data](#) \*startup\_data)  
*Função de determinação do destino do protocolo UDP.*
- void [createsocket\\_udp](#) (struct [transversal\\_data](#) \*transversal\_data)  
*Função de abertura da socket UDP com o servidor de arranque.*

### 4.3.1 Function Documentation

#### 4.3.1.1 void createsocket\_udp ( struct transversal\_data \* transversal\_data )

Esta função recebe como parâmetros a estrutura de dados transversal ao programa e abre uma socket UDP tendo como destino o servidor de arranque.

Adicionalmente, ao abrir a socket é utilizada a função getsockname() de maneira a determinar o endereço IP externo da máquina.

#### 4.3.1.2 void getudpdest ( char \* node, char \* service, struct startup\_data \* startup\_data )

Esta função recebe o endereço e porto do servidor de arranque e coloca na estrutura comum a estrutura correspondente a esse endereço.

Na estrutura comum é alocada memória para a struct sockaddr de destino.

## 4.4 include/options.h File Reference

Cabeçalhos e protótipos associados ao módulo de processamento das opções de linha de comandos.

```
#include "common.h"
```

### Data Structures

- struct [args\\_parse\\_options](#)  
*Estrutura com os argumentos para a função [parse\\_options](#).*

### Functions

- void \* [parse\\_options](#) (struct [args\\_parse\\_options](#) \*params)  
*Função responsável pela interpretação dos argumentos passados via linha de comandos.*

### 4.4.1 Function Documentation

#### 4.4.1.1 void\* parse\_options ( struct args\_parse\_options \* params )

Esta função recorre à função getopt() de maneira a obter os diversos parâmetros passados via linha de comandos. Os argumentos são passados via ponteiro para estrutura de maneira a que a função seja compatível com threads.

## 4.5 src/main.c File Reference

Ficheiro principal do projecto.

```
#include <stdlib.h>
#include <stdio.h>
#include "common.h"
#include "options.h"
#include "net_udp.h"
```

### Functions

- `int main (int argc, char **argv)`  
*Função principal do programa ddt.*

#### 4.5.1 Detailed Description

Este ficheiro serve como chave de abóbada para os vários módulos desenvolvidos e responsáveis pelas diversas funcionalidades do projecto.

#### 4.5.2 Function Documentation

##### 4.5.2.1 `int main ( int argc, char ** argv )`

Esta função serve-se dos módulos criados nos restantes ficheiros-fonte para implementar as funcionalidades necessárias ao projecto, como especificadas no enunciado.

## 4.6 src/net\_common.c File Reference

Ficheiro responsável pela implementação das funções relacionadas com rede mas não associadas a nenhum dos protocolos utilizados.

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <string.h>
#include "common.h"
#include "net_common.h"
```

### Macros

- `#define _XOPEN_SOURCE`

### Functions

- `void getsockaddr (char *node, char *service, int *family, socklen_t *size, struct sockaddr **sockaddress, int protocol)`  
*Função de leitura de endereços e portos e criação de estruturas de destino.*

## 4.6.1 Macro Definition Documentation

### 4.6.1.1 #define \_XOPEN\_SOURCE

## 4.6.2 Function Documentation

### 4.6.2.1 void getsockaddr ( char \* node, char \* service, int \* family, socklen\_t \* size, struct sockaddr \*\* sockaddress, int protocol )

Esta função recebe um endereço e

A função getaddrinfo() é utilizada em vez da função gethostbyname(), por uma questão de preferência pessoal e também pelo facto da função getaddrinfo() estar especificada como thread-safe (a função gethostbyname() só é thread-safe em algumas implementações). Os detalhes da função getaddrinfo() podem ser consultados em [RFC 2553](#).

## 4.7 src/net\_udp.c File Reference

Ficheiro responsável pela implementação das funções de rede associadas ao protocolo UDP.

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <string.h>
#include <stdint.h>
#include <arpa/inet.h>
#include "common.h"
#include "net_common.h"
#include "net_udp.h"
```

## Functions

- void [getudpdest](#) (char \*node, char \*service, struct [startup\\_data](#) \*startup\_data)  
*Função de determinação do destino do protocolo UDP.*
- void [createsocket\\_udp](#) (struct [transversal\\_data](#) \*transversal\_data)  
*Função de abertura da socket UDP com o servidor de arranque.*

### 4.7.1 Function Documentation

#### 4.7.1.1 void createsocket\_udp ( struct transversal\_data \* transversal\_data )

Esta função recebe como parâmetros a estrutura de dados transversal ao programa e abre uma socket UDP tendo como destino o servidor de arranque.

Adicionalmente, ao abrir a socket é utilizada a função getsockname() de maneira a determinar o endereço IP externo da máquina.

#### 4.7.1.2 void getudpdest ( char \* node, char \* service, struct startup\_data \* startup\_data )

Esta função recebe o endereço e porto do servidor de arranque e coloca na estrutura comum a estrutura correspondente a esse endereço.



Na estrutura comum é alocada memória para a struct sockaddr de destino.

## 4.8 src/options.c File Reference

Ficheiro responsável pela interpretação dos argumentos passados via linha de comandos.

```
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "common.h"
#include "options.h"
#include "net_udp.h"
```

### Macros

- `#define _XOPEN_SOURCE`

### Functions

- void `usage` (char \*appname)  
*Função que mostra a mensagem Usage.*
- void \* `parse_options` (struct `args_parse_options` \*params)  
*Função responsável pela interpretação dos argumentos passados via linha de comandos.*

#### 4.8.1 Detailed Description

As funções implementadas neste módulo correspondem à interpretação dos argumentos passados via linha de comandos, e correspondente transferência para a estrutura de dados transversal aos vários módulos do projecto.

#### 4.8.2 Macro Definition Documentation

##### 4.8.2.1 `#define _XOPEN_SOURCE`

#### 4.8.3 Function Documentation

##### 4.8.3.1 void\* `parse_options` ( struct `args_parse_options` \* *params* )

Esta função recorre à função `getopt()` de maneira a obter os diversos parâmetros passados via linha de comandos. Os argumentos são passados via ponteiro para estrutura de maneira a que a função seja compatível com threads.

##### 4.8.3.2 void `usage` ( char \* *appname* )



# Index

- `_XOPEN_SOURCE`
    - `net_common.c`, 12
    - `options.c`, 13
- `argc`
  - `args_parse_options`, 5
- `args_parse_options`, 5
  - `argc`, 5
  - `argv`, 5
  - `startup_data`, 5
- `argv`
  - `args_parse_options`, 5
- `createsocket_udp`
  - `net_udp.c`, 12
  - `net_udp.h`, 10
- `dest_size`
  - `startup_data`, 6
- `destination`
  - `startup_data`, 6
- `ext_addr`
  - `transversal_data`, 6
- `family`
  - `startup_data`, 6
- `getsockaddr`
  - `net_common.c`, 12
  - `net_common.h`, 9
- `getudpdest`
  - `net_udp.c`, 12
  - `net_udp.h`, 10
- `include/common.h`, 9
- `include/net_common.h`, 9
- `include/net_udp.h`, 10
- `include/options.h`, 10
- `main`
  - `main.c`, 11
- `main.c`
  - `main`, 11
- `net_common.c`
  - `_XOPEN_SOURCE`, 12
  - `getsockaddr`, 12
- `net_common.h`
  - `getsockaddr`, 9
- `net_udp.c`
  - `createsocket_udp`, 12
  - `getudpdest`, 12
- `net_udp.h`
  - `createsocket_udp`, 10
  - `getudpdest`, 10
- `options.c`
  - `_XOPEN_SOURCE`, 13
  - `parse_options`, 13
  - `usage`, 13
- `options.h`
  - `parse_options`, 10
- `parse_options`
  - `options.c`, 13
  - `options.h`, 10
- `ring`
  - `transversal_data`, 6
- `ringport`
  - `startup_data`, 6
- `src/main.c`, 11
- `src/net_common.c`, 11
- `src/net_udp.c`, 12
- `src/options.c`, 13
- `startup_data`, 5
  - `args_parse_options`, 5
  - `dest_size`, 6
  - `destination`, 6
  - `family`, 6
  - `ringport`, 6
  - `transversal_data`, 6
- `t`
  - `transversal_data`, 6
- `transversal_data`, 6
  - `ext_addr`, 6
  - `ring`, 6
  - `startup_data`, 6
  - `t`, 6
  - `u`, 7
- `u`
  - `transversal_data`, 7
- `usage`
  - `options.c`, 13