

Diagnostic Shell

Generated by Doxygen 1.8.1.1

Wed Feb 20 2013 09:10:44

Contents

1	Program #2	1
1.1	Course Information	1
1.2	Program Information	1
1.3	Compiling and Usage	2
1.4	Todo, Bugs, and Modifications	2
2	Todo List	3
3	Bug List	5
4	Class Index	7
4.1	Class List	7
5	File Index	9
5.1	File List	9
6	Class Documentation	11
6.1	Process Struct Reference	11
6.2	Processes Class Reference	11
6.2.1	Constructor & Destructor Documentation	11
6.2.1.1	Processes	11
6.2.2	Member Function Documentation	12
6.2.2.1	listProcesses	12
6.2.2.2	matchName	12
6.2.2.3	matchPID	12
6.2.2.4	pidMatches	13
6.3	Reroute Class Reference	13
6.3.1	Constructor & Destructor Documentation	13
6.3.1.1	Reroute	13
6.3.2	Member Function Documentation	14
6.3.2.1	redirect	14
6.3.2.2	redirect	14
6.3.2.3	redirect	15

6.3.2.4	redirect	15
6.3.2.5	redirect_remote	15
6.3.2.6	restore	16
7	File Documentation	17
7.1	dsh.cpp File Reference	17
7.1.1	Detailed Description	17
7.1.2	Function Documentation	17
7.1.2.1	call_function	18
7.1.2.2	changeDir	18
7.1.2.3	cmdnm	18
7.1.2.4	exec_command	19
7.1.2.5	main	19
7.1.2.6	pid	19
7.1.2.7	send_signal	20
7.1.2.8	signalCatcher	20
7.1.2.9	systat	20
7.2	helper.cpp File Reference	21
7.2.1	Detailed Description	21
7.2.2	Function Documentation	21
7.2.2.1	filterBlanks	21
7.2.2.2	isinteger	21
7.2.2.3	parse_redirect	22
7.2.2.4	parse_remote_redirect	22
7.2.2.5	split	23
7.2.2.6	stringToInt	23
7.2.2.7	trimLeadingCharacter	23
7.3	helper.h File Reference	24
7.3.1	Detailed Description	24
7.3.2	Function Documentation	24
7.3.2.1	filterBlanks	24
7.3.2.2	isinteger	24
7.3.2.3	parse_redirect	25
7.3.2.4	parse_remote_redirect	25
7.3.2.5	split	26
7.3.2.6	stringToInt	26
7.3.2.7	trimLeadingCharacter	26
7.4	Processes.cpp File Reference	27
7.4.1	Detailed Description	27
7.5	Processes.h File Reference	27

7.5.1 Detailed Description	27
7.6 Reroute.cpp File Reference	27
7.6.1 Detailed Description	27
7.7 Reroute.h File Reference	27
7.7.1 Detailed Description	28
7.7.2 Enumeration Type Documentation	28
7.7.2.1 anonymous enum	28

Chapter 1

Program #2

1.1 Course Information

Authors

Hayden Waisanen

Date

January 30, 2013

Instructor:

Hayden Waisanen

Course:

Operating Systems - MWTF - 2:00 pm

Location:

McLaury - Room 313

1.2 Program Information

This program implements a simple shell with support for 5 commands including:

- Cmdbnm - returns command name for PID
- Signal - sends signal to another process
- Systat - prints various information about the system
- Exit - terminates execution of this program.

The program operates very similar to a normal shell, and can gracefully handle all types of input.

Program 2 additions:

- add support for all system commands
- Additional shell intrinsic functions
- Implement redirection
- Remote shell pipes

1.3 Compiling and Usage

Compiling Instructions:

```
make
```

Usage:

```
$ dsh
```

1.4 Todo, Bugs, and Modifications

Bug Sending a signal 19 causes the program to enter an infinite loop.

Currently, pipes require spaces to function `ls|cat =>` does not work `ls | cat =>` works

Remote piping commands on server side only allow for using port once The pipe becomes open again after 10 minutes or so.

Todo None

Modifications and Development Timeline:

Date	Modification
-----	-----
Jan 16, 2013	Completed processing of command parameters
Jan 18, 2013	Completed processing of proc for cmdnm and pid commands
Jan 24, 2013	Completed processing of proc for systat
Jan 27, 2013	Finalized signal functionality, documentation.
Feb 15, 2013	Modify documentation
Feb 17, 2013	Implement exec for external commands
Feb 18, 2013	Implement all redirection except external pipes
Feb 19, 2013	Finalize external pipes
Feb 20, 2013	Finalize commenting doxygen and submission

Chapter 2

Todo List

page **Program #2**

None

Chapter 3

Bug List

page [Program #2](#)

Sending a signal 19 causes the program to enter an infinite loop.

Currently, pipes require spaces to function `ls|cat =>` does not work `ls | cat =>` works

Remote piping commands on server side only allow for using port once The pipe becomes open again after 10 minutes or so.

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Process	11
Processes	11
Reroute	13

Chapter 5

File Index

5.1 File List

Here is a list of all documented files with brief descriptions:

dsh.cpp	17
helper.cpp	21
helper.h	24
Processes.cpp	27
Processes.h	27
Reroute.cpp	27
Reroute.h	27

Chapter 6

Class Documentation

6.1 Process Struct Reference

Public Attributes

- string **pid**
- string **name**

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

- [Processes.h](#)

6.2 Processes Class Reference

Public Member Functions

- [Processes](#) ()
- void [listProcesses](#) ()
- string [matchPID](#) (string [pid](#), bool &err)
- string [matchName](#) (string name, bool &err)
- vector< [Process](#) > [pidMatches](#) (string name)

6.2.1 Constructor & Destructor Documentation

6.2.1.1 [Processes::Processes](#) ()

Author

Hayden Waisanen

Description:

Empty constructor

Class:

[Processes](#)

Returns

n/a

6.2.2 Member Function Documentation

6.2.2.1 void Processes::listProcesses ()

Author

Hayden Waisanen

Description:

Helper function used to print out all processes

Class:

[Processes](#)

Returns

n/a

6.2.2.2 string Processes::matchName (string *name*, bool & *err*)

Author

Hayden Waisanen

Description:

This function returns the name PID of the process with name

Class:

[Processes](#)

Parameters

<i>in</i>	<i>string</i>	name - name to match
<i>in</i>	<i>bool</i>	err - flag if the process was not found

Returns

n/a

6.2.2.3 string Processes::matchPID (string *pid*, bool & *err*)

Author

Hayden Waisanen

Description:

Return name of process with pid

Class:

[Processes](#)

Parameters

in	<i>string</i>	pid - pid to be matched
in	<i>bool</i>	err - flag for if the process doesn't exist

Returns

n/a

6.2.2.4 `vector< Process > Processes::pidMatches (string name)`

Author

Hayden Waisanen

Description:

This function returns all processes with the substring name.

Class:

[Processes](#)

Parameters

in	<i>string</i>	name - substring to search
----	---------------	----------------------------

Returns

n/a

The documentation for this class was generated from the following files:

- [Processes.h](#)
- [Processes.cpp](#)

6.3 Reroute Class Reference

Public Member Functions

- [Reroute](#) ()
- bool [redirect](#) (int file_no, int new_file_no)
- bool [redirect](#) (int file_no, int input_pipe[])
- bool [redirect](#) (int file_no, string file_name)
- bool [redirect](#) (int file_no, string ip, string port)
- bool [redirect_remote](#) (int file_no, string port)
- bool [restore](#) (int file_no)

6.3.1 Constructor & Destructor Documentation

6.3.1.1 `Reroute::Reroute ()`

Author

Hayden Waisanen

Description:

Empty constructor

Class:

[Reroute](#)

Returns

n/a

6.3.2 Member Function Documentation**6.3.2.1 bool Reroute::redirect (int *file_no*, int *new_file_no*)****Author**

Hayden Waisanen

Description:

This temporarily redirects the entered *file_no* to *new_file_no*. Must eventually be followed by a restore call to restore previous input or output file descriptor.

Class:

[Reroute](#)

Parameters

<i>in</i>	<i>int</i>	<i>file_no</i> - file descriptor to redirect
<i>in</i>	<i>int</i>	<i>new_file_no</i> - file descriptor to direct to

Returns

bool - success or failure

6.3.2.2 bool Reroute::redirect (int *file_no*, int *input_pipe*[])**Author**

Hayden Waisanen

Description:

This function temporarily redirect STD_IN or STD_OUT to a pipe. The function intelligently chooses which end of the pipe to use

Class:

[Reroute](#)

Parameters

<i>in</i>	<i>int</i>	<i>file_no</i> - Redirect STDIN, 0, or STDOUT, 1
<i>in</i>	<i>int</i>	<i>input_pipe</i> [] - pipe to redirect

Returns

bool - success or failure

6.3.2.3 bool Reroute::redirect (int *file_no*, string *file_name*)**Author**

Hayden Waisanen

Description:

This function temporarily redirect STD_IN or STD_OUT to a file, specified in the *file_name* parameter. Must be followed by a restore method call to restore output or input.

Class:

[Reroute](#)

Parameters

<i>in</i>	<i>int</i>	<i>file_no</i> - Redirect STDIN, 0, or STDOUT, 1
<i>in</i>	<i>int</i>	<i>input_pipe[]</i> - pipe to redirect

Returns

bool - success or failure

6.3.2.4 bool Reroute::redirect (int *file_no*, string *ip*, string *port*)**Author**

Hayden Waisanen

Description:

This function temporarily redirect STD_IN or STD_OUT to a file, specified in the *file_name* parameter. Must be followed by a restore method call to restore output or input.

Class:

[Reroute](#)

Parameters

<i>in</i>	<i>int</i>	<i>file_no</i> - Redirect STDIN, 0, or STDOUT, 1
<i>in</i>	<i>int</i>	<i>input_pipe[]</i> - pipe to redirect

Returns

bool - success or failure

6.3.2.5 bool Reroute::redirect_remote (int *file_no*, string *port*)

Author

Hayden Waisanen

Description:

This function creates a remote output socket for the specified port. Function call must be eventually followed with a restore call to return output to stdout

Class:

[Reroute](#)

Parameters

<code>in</code>	<code>int</code>	file_no - Redirect STDIN, 0, or STDOUT, 1
<code>in</code>	<code>string</code>	port - Port to output to

Returns

bool - success or failure

6.3.2.6 bool Reroute::restore (int file_no)**Author**

Hayden Waisanen

Description:

Pops previous file descriptor off stack and restores it to the specified input or output.

Class:

[Reroute](#)

Parameters

<code>in</code>	<code>int</code>	file_no - Redirect STDIN, 0, or STDOUT, 1
-----------------	------------------	---

Returns

bool - success or failure

The documentation for this class was generated from the following files:

- [Reroute.h](#)
- [Reroute.cpp](#)

Chapter 7

File Documentation

7.1 dsh.cpp File Reference

```
#include <errno.h>
#include <fstream>
#include <iostream>
#include <fcntl.h>
#include <iomanip>
#include <sys/types.h>
#include <stdio.h>
#include <signal.h>
#include <string>
#include <unistd.h>
#include <vector>
#include "Processes.h"
#include "helper.h"
#include <sys/wait.h>
#include <cstdlib>
#include <string.h>
#include <sys/time.h>
#include <sys/resource.h>
#include "Reroute.h"
```

Functions

- void [call_function](#) (string command)
- void [pid](#) (vector< string > params)
- void [cmdnm](#) (vector< string > params)
- void [systat](#) (vector< string > params)
- void [send_signal](#) (vector< string > params)
- void [signalCatcher](#) (int n)
- void [exec_command](#) (vector< string > params)
- void [changeDir](#) (vector< string > params)
- int [main](#) ()

7.1.1 Detailed Description

7.1.2 Function Documentation

7.1.2.1 void call_function (string command)

Author

Hayden Waisanen

Description:

This function routes all commands to their specific functions. It also handles directs including.

- Remote redirects: ((<ip> <port> OR)) <port>
- File redirects: > file.out OR < file.in
- Pipes: cmd1 | cmd2

Parameters

in	params	- vector of all params used to initiate this command
----	--------	--

Returns

n/a

7.1.2.2 void changeDir (vector< string > params)

Author

Hayden Waisanen

Description:

This function allows the user to change the current working directory

Parameters

in	params	- vector of all params used to initiate this command
----	--------	--

Returns

n/a

7.1.2.3 void cmdnm (vector< string > params)

Author

Hayden Waisanen

Description:

This function finds the process name using the pid

Parameters

in	params	- vector of all params used to initiate this command
----	--------	--

Returns

n/a

7.1.2.4 void exec_command (vector< string > *params*)**Author**

Hayden Waisanen

Description:

This function allows for dsh to take any command supported by the system then return the output to stdout. It does this by fork/execing the new command (printing out the child pid while doing so)

Parameters

in	<i>params</i>	- vector of all params used to initiate this command
----	---------------	--

Returns

n/a

7.1.2.5 int main ()**Author**

Hayden Waisanen

Description:

The main function handles registering the signal call back functions, as well as enters the program into the main event loop. When the user enters 'exit' the program will terminate.

none

Returns

int

7.1.2.6 void pid (vector< string > *params*)**Author**

Hayden Waisanen

Description:

This function finds all processes with the given substring in their name.

Parameters

in	<i>params</i>	- vector of all params used to initiate this command
----	---------------	--

Returns

n/a

7.1.2.7 void send_signal (vector< string > *params*)**Author**

Hayden Waisanen

Description:

This function is responsible for taking in the parameters for the "signal" command and sending out a signal to the specified pid. Error checking is done prior to calling the kill command to verify that the process does, in fact, exist.

Parameters

<i>in</i>	<i>params</i>	- vector of all params used to initiate this command
-----------	---------------	--

Returns

n/a

7.1.2.8 void signalCatcher (int *n*)**Author**

Hayden Waisanen

Description:

This is the callback function for when the running process receives a signal. It simply prints out the id of the signal received.

Parameters

<i>in</i>	<i>n</i>	- id of the signal received
-----------	----------	-----------------------------

Returns

n/a

7.1.2.9 void systat (vector< string > *params*)**Author**

Hayden Waisanen

Description:

This function simply imports system information from the /proc directory and displays it to the user.

Parameters

<i>in</i>	<i>params</i>	- vector of all params used to initiate this command
-----------	---------------	--

Returns

n/a

7.2 helper.cpp File Reference

```
#include "helper.h"
```

Functions

- bool [isinteger](#) (string number)
- int [stringToInt](#) (string number)
- void [trimLeadingCharacter](#) (string &str, char chr)
- void [filterBlanks](#) (vector< string > &strs)
- vector< string > [split](#) (string str, char divide)
- void [parse_redirect](#) (vector< string > &command_array, bool &rert_stdin, bool &rert_stdout, string &stdin_file, string &stdout_file)
- void [parse_remote_redirect](#) (vector< string > &command_array, bool &rert_stdin, bool &rert_stdout, string &stdin_ip, string &stdin_port, string &stdout_port)

7.2.1 Detailed Description

7.2.2 Function Documentation

7.2.2.1 void filterBlanks (vector< string > & *strs*)

Author

Hayden Waisanen

Description:

Remove all empty strings from a vector<string>

Parameters

in	<i>vector<string></i>	<i>strs</i> - edited in place
----	-----------------------------	-------------------------------

Returns

n/a

7.2.2.2 bool isinteger (string *number*)

Author

Hayden Waisanen

Description:

Checks to see if the string passed in is a positive integer

Parameters

in	<i>string</i>	number
----	---------------	--------

Returns

true - is a positive integer

7.2.2.3 void parse_redirect (vector< string > & *command_array*, bool & *rert_stdin*, bool & *rert_stdout*, string & *stdin_file*, string & *stdout_file*)

Author

Hayden Waisanen

Description:

This function parses the parameters for a command and returns:

- whether stdout should be redirected
- whether stdin should be redirected
- Removes these arguments from the parameter list

Parameters

in, out	<i>command_array</i>	- the list of parameters
in, out	<i>rert_stdin</i>	- reroute stdin
in, out	<i>rert_stdout</i>	- reroute stdout
in, out	<i>stdin_file</i>	- set to redirect input file if rert_stdin is true
in, out	<i>stdout_file</i>	- set to redirect out file if rert_stdout is true

Returns

n/a

7.2.2.4 void parse_remote_redirect (vector< string > & *command_array*, bool & *rert_stdin*, bool & *rert_stdout*, string & *stdin_ip*, string & *stdin_port*, string & *stdout_port*)

Author

Hayden Waisanen

Description:

This function parses the parameters for a command and returns:

- whether stdout should be redirected (port)
- whether stdin should be redirected (port, ip)
- Removes these arguments from the parameter list

Parameters

in, out	<i>command_array</i>	- the list of parameters
in, out	<i>rert_stdin</i>	- reroute stdin
in, out	<i>rert_stdout</i>	- reroute stdout
in, out	<i>stdin_ip</i>	- reroute stdin ip address
in, out	<i>stdin_port</i>	- reroute stdin port
in, out	<i>stdout_port</i>	- reroute stdout port

Returns

n/a

7.2.2.5 `vector<string> split (string str, char divide)`

Author

Hayden Waisanen

Description:

Splits the inputted *str* on the char *divide* `split(" My mommy loves me lots", ' '); => ("My", "mommy", "loves", "me", "lots")`

Parameters

in	<i>str</i>	- to be split
in	<i>divide</i>	- what to divide on

Returns

n/a

7.2.2.6 `int stringToInt (string number)`

Author

Hayden Waisanen

Description:

This function converts an string to an integer

Parameters

in	<i>string</i>	number
----	---------------	--------

Returns

n/a

7.2.2.7 `void trimLeadingCharacter (string & str, char chr)`

Author

Hayden Waisanen

Description:

Trims every leading instance of *chr* in *str*

Parameters

in, out	<i>str</i>	edited in place
in	<i>chr</i>	

Returns

n/a

7.3 helper.h File Reference

```
#include <string>
#include <sstream>
#include <vector>
```

Functions

- bool [isinteger](#) (string number)
- int [stringToInt](#) (string number)
- void [trimLeadingCharacter](#) (string &str, char chr)
- vector< string > [split](#) (string str, char divide)
- void [filterBlanks](#) (vector< string > &strs)
- void [parse_redirect](#) (vector< string > &command_array, bool &rert_stdin, bool &rert_stdout, string &stdout_file, string &stdin_file)
- void [parse_remote_redirect](#) (vector< string > &command_array, bool &rert_stdin, bool &rert_stdout, string &stdin_ip, string &stdin_port, string &stdout_port)

7.3.1 Detailed Description

7.3.2 Function Documentation

7.3.2.1 void filterBlanks (vector< string > & *strs*)

Author

Hayden Waisanen

Description:

Remove all empty strings from a vector<string>

Parameters

<i>in</i>	<i>vector<string></i>	<i>strs</i> - edited in place
-----------	-----------------------------	-------------------------------

Returns

n/a

7.3.2.2 bool isinteger (string *number*)

Author

Hayden Waisanen

Description:

Checks to see if the string passed in is a positive integer

Parameters

in	string	number
----	--------	--------

Returns

true - is a positive integer

7.3.2.3 void parse_redirect (vector< string > & command_array, bool & rert_stdin, bool & rert_stdout, string & stdin_file, string & stdout_file)

Author

Hayden Waisanen

Description:

This function parses the parameters for a command and returns:

- whether stdout should be redirected
- whether stdin should be redirected
- Removes these arguments from the parameter list

Parameters

in, out	command_array	- the list of parameters
in, out	rert_stdin	- reroute stdin
in, out	rert_stdout	- reroute stdout
in, out	stdin_file	- set to redirect input file if rert_stdin is true
in, out	stdout_file	- set to redirect out file if rert_stdout is true

Returns

n/a

7.3.2.4 void parse_remote_redirect (vector< string > & command_array, bool & rert_stdin, bool & rert_stdout, string & stdin_ip, string & stdin_port, string & stdout_port)

Author

Hayden Waisanen

Description:

This function parses the parameters for a command and returns:

- whether stdout should be redirected (port)
- whether stdin should be redirected (port, ip)
- Removes these arguments from the parameter list

Parameters

in, out	command_array	- the list of parameters
in, out	rert_stdin	- reroute stdin
in, out	rert_stdout	- reroute stdout
in, out	stdin_ip	- reroute stdin ip address
in, out	stdin_port	- reroute stdin port
in, out	stdout_port	- reroute stdout port

Returns

n/a

7.3.2.5 vector<string> split (string *str*, char *divide*)**Author**

Hayden Waisanen

Description:

Splits the inputted str on the char divide split(" My mommy loves me lots", ' '); => ("My", "mommy", "loves", "me", "lots")

Parameters

in	<i>str</i>	- to be split
in	<i>divide</i>	- what to divide on

Returns

n/a

7.3.2.6 int stringToInt (string *number*)**Author**

Hayden Waisanen

Description:

This function converts an string to an integer

Parameters

in	<i>string</i>	number
----	---------------	--------

Returns

n/a

7.3.2.7 void trimLeadingCharacter (string & *str*, char *chr*)**Author**

Hayden Waisanen

Description:

Trims every leading instance of chr in str

Parameters

in, out	<i>str</i>	edited in place
in	<i>chr</i>	

Returns

n/a

7.4 Processes.cpp File Reference

```
#include "Processes.h"
```

7.4.1 Detailed Description

7.5 Processes.h File Reference

```
#include <dirent.h>
#include <vector>
#include <string>
#include <iostream>
#include <fstream>
#include <errno.h>
#include "helper.h"
```

Classes

- struct [Process](#)
- class [Processes](#)

7.5.1 Detailed Description

7.6 Reroute.cpp File Reference

```
#include "Reroute.h"
```

7.6.1 Detailed Description

7.7 Reroute.h File Reference

```
#include <dirent.h>
#include <vector>
#include <string>
#include <iostream>
#include <fstream>
#include <errno.h>
#include "helper.h"
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <cstdlib>
#include <arpa/inet.h>
#include <string.h>
```

Classes

- class [Reroute](#)

Enumerations

- enum { [R_IN](#), [R_OUT](#) }

7.7.1 Detailed Description

Author

Hayden Waisanen

Description:

This class takes care of all opening and closing required for redirecting output and input. Previous file descriptors are saved and placed on a stack. This allows for restoration of output or input.

Common Usage: `redirect(...)` Do some stuff... `restore(...)`

Class:

[Reroute](#)

7.7.2 Enumeration Type Documentation

7.7.2.1 anonymous enum

[Reroute](#) Enum This enum allows for easier reading of calls to [Reroute](#)'s methods.

Enumerator:

R_IN maps to `STD_IN`

R_OUT maps to `STD_OUT`

Index

- call_function
 - dsh.cpp, 17
- changeDir
 - dsh.cpp, 18
- cmdnm
 - dsh.cpp, 18
- dsh.cpp, 17
 - call_function, 17
 - changeDir, 18
 - cmdnm, 18
 - exec_command, 19
 - main, 19
 - pid, 19
 - send_signal, 20
 - signalCatcher, 20
 - systat, 20
- exec_command
 - dsh.cpp, 19
- filterBlanks
 - helper.cpp, 21
 - helper.h, 24
- helper.cpp, 21
 - filterBlanks, 21
 - isinteger, 21
 - parse_redirect, 22
 - parse_remote_redirect, 22
 - split, 23
 - stringToInt, 23
 - trimLeadingCharacter, 23
- helper.h, 24
 - filterBlanks, 24
 - isinteger, 24
 - parse_redirect, 25
 - parse_remote_redirect, 25
 - split, 26
 - stringToInt, 26
 - trimLeadingCharacter, 26
- isinteger
 - helper.cpp, 21
 - helper.h, 24
- listProcesses
 - Processes, 12
- main
 - dsh.cpp, 19
- matchName
 - Processes, 12
- matchPID
 - Processes, 12
- parse_redirect
 - helper.cpp, 22
 - helper.h, 25
- parse_remote_redirect
 - helper.cpp, 22
 - helper.h, 25
- pid
 - dsh.cpp, 19
- pidMatches
 - Processes, 13
- Process, 11
- Processes, 11
 - listProcesses, 12
 - matchName, 12
 - matchPID, 12
 - pidMatches, 13
 - Processes, 11
- Processes.cpp, 27
- Processes.h, 27
- R_IN
 - Reroute.h, 28
- R_OUT
 - Reroute.h, 28
- redirect
 - Reroute, 14, 15
- redirect_remote
 - Reroute, 15
- Reroute, 13
 - redirect, 14, 15
 - redirect_remote, 15
- Reroute, 13
 - restore, 16
- Reroute.h
 - R_IN, 28
 - R_OUT, 28
- Reroute.cpp, 27
- Reroute.h, 27
- restore
 - Reroute, 16
- send_signal
 - dsh.cpp, 20
- signalCatcher
 - dsh.cpp, 20

split
 [helper.cpp, 23](#)
 [helper.h, 26](#)
stringToInt
 [helper.cpp, 23](#)
 [helper.h, 26](#)
systat
 [dsh.cpp, 20](#)

trimLeadingCharacter
 [helper.cpp, 23](#)
 [helper.h, 26](#)