

Reference Manual

Generated by Doxygen 1.7.4

Tue Nov 15 2011 21:06:40

Contents

1	Module Index	1
1.1	Modules	1
2	Class Index	3
2.1	Class List	3
3	Module Documentation	5
3.1	shooSH	5
3.1.1	Detailed Description	5
3.2	Builtin	5
3.2.1	Detailed Description	6
3.2.2	Function Documentation	6
3.2.2.1	background	6
3.2.2.2	changeDirectory	7
3.2.2.3	echo	7
3.2.2.4	executeBuiltin	7
3.2.2.5	foreground	7
3.2.2.6	sendsigterm	7
3.3	Initializer	8
3.3.1	Detailed Description	8
3.4	Parser	9
3.4.1	Detailed Description	9
3.5	Redirector	9
3.5.1	Detailed Description	9
3.6	shooSHlib	9
3.6.1	Detailed Description	10

4	Class Documentation	11
4.1	Executor Class Reference	11
4.1.1	Detailed Description	11
4.1.2	Member Function Documentation	11
4.1.2.1	execute	11
4.2	Job Class Reference	12
4.2.1	Detailed Description	12
4.2.2	Member Function Documentation	13
4.2.2.1	getProcess	13
4.2.2.2	setProcessCommand	13
4.2.2.3	setProcessFile	13
4.3	Parser Class Reference	13
4.3.1	Detailed Description	14
4.3.2	Member Function Documentation	14
4.3.2.1	parseLine	14
4.4	Piper Class Reference	14
4.4.1	Detailed Description	14
4.4.2	Member Function Documentation	14
4.4.2.1	execPipe	14
4.5	Process Class Reference	15
4.5.1	Detailed Description	15
4.6	Redirector Class Reference	15
4.6.1	Detailed Description	16
4.6.2	Member Function Documentation	16
4.6.2.1	init	16

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

shooSH	5
Builtin	5
Initializer	8
Parser	9
Redirector	9
shooSHlib	9

Chapter 2

Class Index

2.1 Class List

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

Executor (Classe que executa os comandos)	11
Job (Classe de Job)	12
Parser (Parser de linha)	13
Piper	14
Process	15
Redirector	15

Chapter 3

Module Documentation

3.1 shooSH

Functions

- void `shooSH_init` (void)
Funcao de inicializacao da Shell.
- void `shooSH_clean` (void)
Funcao para liberar os recursos alocados pela Shell.
- void `shooSH_run` (void)
Funcao que executa a Shell.

Variables

- int `currID`
ID do ultimo `Job` criado.

3.1.1 Detailed Description

Funcoes globais utilizadas pela shell. }@

3.2 Builtin

Functions

- void `jobs` (void)
Mostra todos os jobs da lista.
- void `showHistory` (void)

Mostra comandos digitados ate o momento.

- void [sendsigterm](#) (int id)

Envia um sinal SIGTERM para o job dado por id.

- void [echo](#) (int argc, char *args[])

Escreve os argumentos com 1 espaco entre eles.

- void [changeDirectory](#) (const char *dir)

Muda o diretorio atual.

- void [showDirectoryName](#) (void)

Mostra o diretorio atual.

Exceptions

-1	Diretorio maior que 256 caracteres.
----	-------------------------------------

- void [foreground](#) (int id)

Coloca o job em execucao em foreground.

- void [background](#) (int id)

Coloca o job em execucao em background.

- int [executeBuiltin](#) ([Process](#) *p)

Executa comandos builtin.

Variables

- std::list< std::string > [history](#)

Historico de comandos.

3.2.1 Detailed Description

Funcoes integradas }@

3.2.2 Function Documentation

3.2.2.1 void background (int id)

Coloca o job em execucao em background.

Parameters

<i>id</i>	ID do job.
-----------	------------

Exceptions

-ID	Job nao encontrado.
-----	-------------------------------------

3.2.2.2 void changeDirectory (const char * *dir*)

Muda o diretorio atual.

Parameters

<i>dir</i>	Diretorio para o qual ocorrera a mudanca.
------------	---

3.2.2.3 void echo (int *argc*, char * *args*[])

Escreve os argumentos com 1 espaco entre eles.

Parameters

<i>argc</i>	Numero de palavras a serem escritas + 1.
<i>args</i>	Palavras a serem escritas, comeca com "echo".

3.2.2.4 int executeBuiltin (Process * *p*)

Executa comandos builtin.

Parameters

<i>p</i>	Processo a ser executado.
----------	---------------------------

Returns

0 Comando nao eh builtin 1 Faltam parametros 2 Erro nos parametros 0 Sucesso

3.2.2.5 void foreground (int *id*)

Coloca o job em execucao em foreground.

Parameters

<i>id</i>	ID do job.
-----------	------------

Exceptions

<i>-ID</i>	Job nao encontrado.
------------	-------------------------------------

3.2.2.6 void sendsigterm (int *id*)

Envia um sinal SIGTERM para o job dado por id.

Parameters

<code>id</code>	ID do job.
-----------------	------------

Exceptions

<code>-id</code>	Job inexistente.
------------------	----------------------------------

3.3 Initializer

Functions

- void [hanchld](#) (int signum, siginfo_t *siginfo, void *context)
Handler para sinais de processos filhos.
- void [hanttin](#) (int signum, siginfo_t *siginfo, void *context)
Handler para o sinal SIGTTIN.
- void [hanttou](#) (int signum)
Handler para o sinal SIGTTOU.
- void [init_tty](#) (void)
Inicializa sinais e entrega o terminal para shell.

Variables

- std::list< [Job](#) * > [jobList](#)
Lista de Jobs da shell.
- struct termios [shooSHTermios](#)
Atributos da shell.
- pid_t [shooSHPGID](#)
Group ID da shell.
- struct sigaction [sa_chld](#)
Estrutura para o sinal SIGCHLD.
- struct sigaction [sa_ttin](#)
Estrutura para o sinal SIGTTIN.
- struct sigaction [sa_ttou](#)
Estrutura para o sinal SIGTTOU.

3.3.1 Detailed Description

Atributos da shell e funcoes de inicializacao. }@

3.4 Parser

Defines

- `#define NUM_WORDS 100`
- `#define MAX_LENGTH 1024`
- `#define MAX_FILENAME 128`
- `#define PARSCMD 0`
- `#define PARSNEXT 1`
- `#define PARSPARAM 2`
- `#define PARSRIN 3`
- `#define PARROUT 4`
- `#define PARSTWO 5`
- `#define PARSRERR 6`
- `#define PARSFILNAME 7`
- `#define PARSSUCCESS 8`
- `#define PARFAIL 9`

3.4.1 Detailed Description

Definicoes de limites e estados da maquina de estados.

3.5 Redirector

Defines

- `#define REDIN 1`
- `#define REDOUTT 2`
- `#define REDOUTA 4`
- `#define REDERRT 8`
- `#define REDERRA 16`

3.5.1 Detailed Description

Definicoes de flags de redirecionamento

3.6 shooSHlib

Defines

- `#define shooSH_EXIT 1`
- `#define shooSH_PIPE 2`

- #define **shooSH_REDIR** 4
- #define **shooSH_NOP** 8
- #define **shooSH_FAIL** 16
- #define **shooSH_BG** 32

3.6.1 Detailed Description

Definicoes de flags da shell

Chapter 4

Class Documentation

4.1 Executor Class Reference

Classe que executa os comandos.

```
#include <Executor.hpp>
```

Public Member Functions

- void [execute](#) ([Job](#) *job)
Executa um job.

4.1.1 Detailed Description

Classe que executa os comandos.

[Executor](#)

4.1.2 Member Function Documentation

4.1.2.1 void [Executor::execute](#) ([Job](#) * *job*) [[inline](#)]

Executa um job.

Parameters

<i>job</i>	Ponteiro para o job a ser executado.
------------	--------------------------------------

Exceptions

-1	Ocorreu um erro no fork.
----	--------------------------

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

- Executor.hpp

4.2 Job Class Reference

Classe de [Job](#).

```
#include <Job.hpp>
```

Public Member Functions

- void [createProcess](#) (void)
Cria um processo e o insere no vetor de processos.
- void [destroy](#) (void)
Desaloca todos os recursos.
- std::string [getCommand](#) (void)
- int [getID](#) (void)
- pid_t [getPGID](#) (void)
- [Process](#) * [getProcess](#) (int i)
Retorna um processo do vetor de processos.
- struct termios * [getTermios](#) (void)
- bool [hasPipe](#) (void)
- bool [hasExited](#) (void)
- bool [hasFailed](#) (void)
- bool [inBG](#) (void)
- bool [isNop](#) (void)
- bool [isStopped](#) (void)
- int [size](#) (void)
- void [addFlag](#) (int)
- void [cont](#) (void)
- void [setBG](#) (bool)
- void [setCommand](#) (std::string command)
- void [setID](#) (int)
- void [setPGID](#) (pid_t)
- void [setProcessCommand](#) (char **command, int size)
Seta o comando do ultimo processo inserido.
- void [setProcessFile](#) (std::string filename, int flag)
Seta o nome do arquivo relacionado a um tipo de redirecionamento.
- void [stop](#) (void)
- void [print](#) (void)

4.2.1 Detailed Description

Classe de [Job](#).

4.2.2 Member Function Documentation

4.2.2.1 `Process * Job::getProcess (int i)`

Retorna um processo do vetor de processos.

Parameters

<i>i</i>	Indice do processo no vetor.
----------	------------------------------

4.2.2.2 `void Job::setProcessCommand (char ** command, int size)`

Seta o comando do ultimo processo inserido.

Parameters

<i>command</i>	Matriz de caracteres cuja ultima linha deve ser (char*)NULL.
<i>size</i>	Numero de linhas da matriz sem contar a nula.

4.2.2.3 `void Job::setProcessFile (std::string filename, int flag)`

Seta o nome do arquivo relacionado a um tipo de redirecionamento.

Parameters

<i>filename</i>	Nome do arquivo.
<i>flag</i>	Flag indicando o tipo de redirecionamento

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

- lib/Job.hpp
- lib/Job.cpp

4.3 Parser Class Reference

[Parser](#) de linha.

```
#include <Parser.hpp>
```

Public Member Functions

- [Job * parseLine](#) (void)

Pega uma linha de comando. Implementada como uma maquina de estados para reconhecer a linguagem gerada pela gramatica indicada na especificacao.

4.3.1 Detailed Description

[Parser](#) de linha.

4.3.2 Member Function Documentation

4.3.2.1 `Job * Parser::parseLine (void)`

Pega uma linha de comando. Implementada como uma maquina de estados para reconhecer a linguagem gerada pela gramatica indicada na especificacao.

Returns

Ponteiro para o job gerado.

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

- lib/Parser.hpp
- lib/Parser.cpp

4.4 Piper Class Reference

```
#include <Piper.hpp>
```

Public Member Functions

- void [execPipe](#) ([Job](#) *job)
Executa um job com pipe. Implementada como uma fila invertida, ou seja, o primeiro comando a ser executado sera o processo com maior PID.

4.4.1 Detailed Description

Classe para execucao de comandos com pipe.

4.4.2 Member Function Documentation

4.4.2.1 `void Piper::execPipe (Job * job) [inline]`

Executa um job com pipe. Implementada como uma fila invertida, ou seja, o primeiro comando a ser executado sera o processo com maior PID.

Parameters

<i>job</i>	Ponteiro para o job a ser executado
------------	-------------------------------------

Exceptions

-1	Erro no fork
----	--------------

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

- Piper.hpp

4.5 Process Class Reference

```
#include <Process.hpp>
```

Public Member Functions

- void **destroy** (void)
Desaloca a matriz de comandos.
- bool **hasRedirection** (void)
- void **setCommand** (char **, int)
- void **setFilename** (std::string, int)
- void **setPID** (pid_t)
- char ** **getCommand** (void)
- std::string * **getFilenames** (void)
- pid_t **getPID** (void)
- int * **getRedirFlags** (void)
- int **size** (void)
- void **print** (void)

4.5.1 Detailed Description

Classe de processo

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

- lib/Process.hpp
- lib/Process.cpp

4.6 Redirector Class Reference

```
#include <Redirector.hpp>
```

Public Member Functions

- void `init` (std::string filename[3], int flag[3])
Abre os arquivos necessarios e duplica descritores.
- void `end` (void)
Fecha os descritores de arquivos.

4.6.1 Detailed Description

Classe com funcoes de redirecionamento

4.6.2 Member Function Documentation

4.6.2.1 void Redirector::init (std::string filename[3], int flag[3])

Abre os arquivos necessarios e duplica descritores.

Parameters

<i>filename</i>	Nomes dos arquivos.
<i>flag</i>	Flags indicando os tipos de redirecionamento

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

- lib/Redirector.hpp
- lib/Redirector.cpp

Index

- background
 - Builtin, [6](#)
- Builtin, [5](#)
 - background, [6](#)
 - changeDirectory, [6](#)
 - echo, [7](#)
 - executeBuiltin, [7](#)
 - foreground, [7](#)
 - sendsigterm, [7](#)
- changeDirectory
 - Builtin, [6](#)
- echo
 - Builtin, [7](#)
- execPipe
 - Piper, [14](#)
- execute
 - Executor, [11](#)
- executeBuiltin
 - Builtin, [7](#)
- Executor, [11](#)
 - execute, [11](#)
- foreground
 - Builtin, [7](#)
- getProcess
 - Job, [13](#)
- init
 - Redirector, [16](#)
- Initializer, [8](#)
- Job, [12](#)
 - getProcess, [13](#)
 - setProcessCommand, [13](#)
 - setProcessFile, [13](#)
- parseLine
 - Parser, [14](#)
- Parser, [9](#), [13](#)
 - parseLine, [14](#)
 - Piper, [14](#)
 - execPipe, [14](#)
 - Process, [15](#)
- Redirector, [9](#), [15](#)
 - init, [16](#)
- sendsigterm
 - Builtin, [7](#)
- setProcessCommand
 - Job, [13](#)
- setProcessFile
 - Job, [13](#)
- shooSH, [5](#)
- shooSHlib, [9](#)