Reference Manual

Generated by Doxygen 1.7.3

Sun Nov 13 2011 20:26:28

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 handlers Namespace Reference

Namespace relacionado as funcoes que manipulam interrupcoes de software.

Functions

- void sigChildHandler (int)

  Handler para o sinal SIGCHLD. Levanta uma flag dizendo que um sinal vindo de um provesso filho foi lancado.

- bool getDeathStatus ()
- void setDeathStatusFalse ()

  Altera a flag de sinais recebidos. Altera o valor da flag de sinais recebidos para false.

- void setDeathStatusTrue ()

  Altera a flag de sinais recebidos. Altera o valor da flag de sinais recebidos para true.

5.1.1 Detailed Description

Namespace relacionado as funcoes que manipulam interrupcoes de software.

5.1.2 Function Documentation

5.1.2.1 bool handlers::getDeathStatus ( )

Returns

Retorna se ha novos sinais para serem processados.

5.1.2.2 void handlers::setDeathStatusFalse ( )

Altera a flag de sinais recebidos. Altera o valor da flag de sinais recebidos para false.

5.1.2.3 void handlers::setDeathStatusTrue ( )

Altera a flag de sinais recebidos. Altera o valor da flag de sinais recebidos para true.

5.1.2.4 void handlers::sigChildHandler (int *signum*)

Handler para o sinal SIGCHLD. Levanta uma flag dizendo que um sinal vindo de um provesso filho foi lancado.

Chapter 6

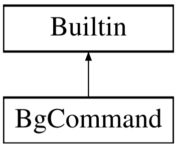
Class Documentation

6.1 BgCommand Class Reference

Classe que implementa o comando bg.

```
#include <Builtin.hpp>
```

Inheritance diagram for BgCommand:



6.1.1 Detailed Description

Classe que implementa o comando bg. Sintaxe: bg %<JOBID> | bg

Se o comando for chamado sem o JOBID, sera utilizado o job mais recentemente aberto, colocado em foreground ou background ou, caso ja tenha sido fechado, o mais antigo aberto.

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

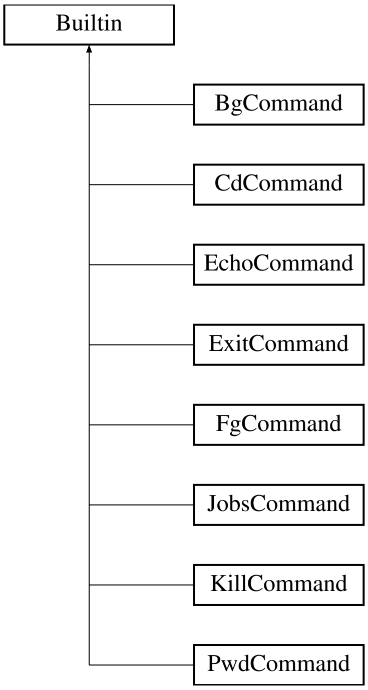
- Builtin.hpp
- Builtin.cpp

6.2 Builtin Class Reference

Classe abstrata para comandos Built-In.

```
#include <Builtin.hpp>
```

Inheritance diagram for Builtin:



Public Member Functions

- void run (const char ∗args[ ], Executor ∗executor)

  Executa um comando built in.

- virtual bool forkable ()

  Se o comando pode ser executado como um proceso independente.

6.2.1 Detailed Description

Classe abstrata para comandos Built-In.

See also

Executor

6.2.2 Member Function Documentation

6.2.2.1 bool Builtin::forkable ( ) `[virtual]`

Se o comando pode ser executado como um proceso independente.

6.2.2.2 void Builtin::run (const char ∗ *args[]*, **Executor** ∗ *executor*)

Executa um comando built in.

Parameters

args	Argumentos do comando, incluindo seu nome.
executor	Pointeiro para uma instancia de Executor.

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

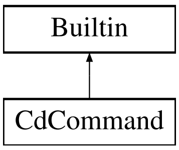
- Builtin.hpp
- Builtin.cpp

6.3 CdCommand Class Reference

Classe que implementa o comando cd.

```
#include <Builtin.hpp>
```

Inheritance diagram for CdCommand:



6.3.1 Detailed Description

Classe que implementa o comando cd. Sintaxe: cd <diretorio>

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

- Builtin.hpp
- Builtin.cpp

6.4 Command Class Reference

Representa um comando entrado pelo usuario. O comando representa tudo o que esta antes de um pipe (|) ou & ou final de linha.

```
#include <Command.hpp>
```

Public Member Functions

- Command (std::vector< std::string > &parameters, std::string in=std::string(), std::string out=std::string(), std::string err=std::string(), bool outAppend=false, bool errAppend=false)
- ~Command ()
- std::string getIn ()
- std::string getOut ()
- std::string getErr ()
- bool getOutAppend ()
- bool getErrAppend ()
- const char ** getExecv ()

6.4.1 Detailed Description

Representa um comando entrado pelo usuario. O comando representa tudo o que esta antes de um pipe (|) ou & ou final de linha.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 Command::Command (std::vector< std::string > & *parameters*, std::string *in* = `std::string()`, std::string *out* = `std::string()`, std::string *err* = `std::string()`, bool *outAppend* = `false`, bool *errAppend* = `false`)

Parameters

parameters	Parametros utilizados na chamada do comando.
in	Nome do aquivo de redirecionamento de entrada
out	Nome do aquivo de redirecionamento de saida.
err	Nome do aquivo de redirecionamento de erro.
outAppend	Se o redirecionamento de saida concatenara com o arquivo ja existente.
errAppend	Se o redirecionamento de erro concatenara com o arquivo ja existente.

6.4.2.2 Command::~Command ( )

6.4.3 Member Function Documentation

6.4.3.1 std::string Command::getErr ( )

Returns

Nome do arquivo de redirecionamento de erro.

6.4.3.2 bool Command::getErrAppend ( )

Returns

Se deve haver anexacao no arquivo de erro.

6.4.3.3 const char ∗∗ Command::getExecv ( )

6.4.3.4 std::string Command::getIn ( )

Returns

Nome do arquivo de redirecionamento de entrada.

6.4.3.5 std::string Command::getOut ( )

Returns

Nome do arquivo de redirecionamento de saida.

6.4.3.6 bool Command::getOutAppend ( )

Returns

Se deve haver anexacao no arquivo de saida.

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

- Command.hpp
- Command.cpp

6.5 CommandLine Class Reference

Represeta uma linha de comando.

```
#include <CommandLine.hpp>
```

Public Member Functions

- CommandLine & operator= (CommandLine &commandLine)
- CommandLine (std::list< Command ∗ > ∗pipeline, bool background)

  Construtor.

- ∼CommandLine ()
- Command ∗ next ()

> *Navega pela pipeline existente A cada utilizacao, o comando extraido e retirado completamente da pipeline existente.*

- bool hasNext ()
- bool isBackground ()

6.5.1 Detailed Description

Represeta uma linha de comando. Uma linha de comando pode conter uma serie de comandos em pipeline e termina quando ha um & ou um final de linha. A linha de comando pode ser gerada por uma instancia de Parser

See also

Command, Parser

6.5.2 Constructor & Destructor Documentation

6.5.2.1 CommandLine::CommandLine (std::list< Command * > * *pipeline*, bool *background*)

Construtor.

Parameters

pipeLine	lista de comandos representando a pipeline
background	verdadeira, caso a pipeline deva ser executada em segundo plano

6.5.2.2 CommandLine::∼CommandLine ( )

6.5.3 Member Function Documentation

6.5.3.1 bool CommandLine::hasNext ( )

Returns

Verdadeiro se a pipeline nao esta vazia

6.5.3.2 bool CommandLine::isBackground ( )

Returns

Verdadeiro se a pipeline deve ser executada em segundo plano

6.5.3.3 Command * CommandLine::next ( )

Navega pela pipeline existente A cada utilizacao, o comando extraido e retirado completamente da pipeline existente.

Returns

> Um ponteiro para o proximo comando da pipeline existente ou NULL o ultimo comando retirado tenha sido o ultimo

6.5.3.4 CommandLine& CommandLine::operator= (CommandLine & *commandLine*)

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

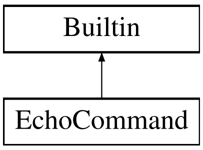
- CommandLine.hpp
- CommandLine.cpp

6.6 EchoCommand Class Reference

Class que implementa o comando echo.

`#include <Builtin.hpp>`

Inheritance diagram for EchoCommand:



6.6.1 Detailed Description

Class que implementa o comando echo. Sintaxe: echo <arg1> <arg2> ... <arg_n>

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

- Builtin.hpp
- Builtin.cpp

6.7 Executor Class Reference

Responsavel pela execucao. Executa uma linha de comando.

`#include <Executor.hpp>`

Classes

- struct Job

  Estrutura que representa um job. Guarda os dados necessarios para controlar processos.

Public Member Functions

- Executor ()
- void run (CommandLine ∗commandLine, std::map< std::string, Builtin ∗ > &bCommands)

  Executa uma linha de comando. Executa os comandos de uma CommandLine.

- void cleanUp ()

  Realiza ajustes na lista de jobs, se necessario. Percorre a lista de jobs, removendo, atualizando status, quando necessario.

- void setForeground (int pid)

  Controle interno. Atualiza o job que esta em foreground.

- unsigned getLastForeground ()
- void setLastForeground (unsigned jobid)

  Altera qual processo deve ir para foreground. Quando ocorrer uma chamada a fg, o processo configurado sera utilizado.

- std::list< Job > ∗ getJobs ()

6.7.1 Detailed Description

Responsavel pela execucao. Executa uma linha de comando. Uma instancia de Executor guarda a lista de processos que sao extraidos das linhas de comando recebidas. Controla a atualizacao dos estados dos processos.

See also

CommandLine, Command

6.7.2 Constructor & Destructor Documentation

6.7.2.1 Executor::Executor ( )

6.7.3 Member Function Documentation

6.7.3.1 void Executor::cleanUp ( )

Realiza ajustes na lista de jobs, se necessario. Percorre a lista de jobs, removendo, atualizando status, quando necessario.

See also

Job, handlers

6.7.3.2 std::list< **Executor::Job** > ∗ Executor::getJobs ( )

Returns

Ponteiro para a lista de jobs.

6.7.3.3 unsigned Executor::getLastForeground ( )

Returns

JobID do processo que deve ir para foreground.

6.7.3.4 void Executor::run (**CommandLine** ∗ *commandLine*, std::map< std::string, **Builtin** ∗ > & *bCommands*)

Executa uma linha de comando. Executa os comandos de uma CommandLine.

See also

Parser, CommandLine, Builtin

Parameters

commandLine	Linha de comando a ser executada.
bCommands	Mapa que associa nomes de comandos internos a comandos internos.

6.7.3.5 void Executor::setForeground (int *pid*)

Controle interno. Atualiza o job que esta em foreground.

6.7.3.6 void Executor::setLastForeground (unsigned *jobid*)

Altera qual processo deve ir para foreground. Quando ocorrer uma chamada a fg, o processo configurado sera utilizado.

See also

FgCommand, BgCommand

Parameters

jobid	JobID do processo a ser configurado.

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

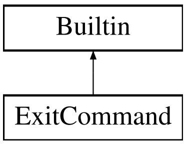
- Executor.hpp
- Executor.cpp

6.8 ExitCommand Class Reference

Classe que implementa o comando exit.

`#include <Builtin.hpp>`

Inheritance diagram for ExitCommand:



6.8.1 Detailed Description

Classe que implementa o comando exit. Sintaxe: exit, quit

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

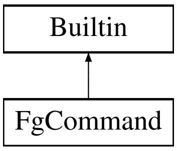
- Builtin.hpp
- Builtin.cpp

6.9 FgCommand Class Reference

Classe que implementa o comando fg.

`#include <Builtin.hpp>`

Inheritance diagram for FgCommand:



6.9.1 Detailed Description

Classe que implementa o comando fg. Sintaxe: fg %<JOBID> | fg

Se o comando for chamado sem o JOBID, sera utilizado o job mais recentemente aberto, colocado em foreground ou background ou, caso ja tenha sido fechado, o mais antigo aberto.

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

- Builtin.hpp
- Builtin.cpp

6.10 Executor::Job Struct Reference

Estrutura que representa um job. Guarda os dados necessarios para controlar processos.

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

Public Member Functions

- Job ()

Public Attributes

- std::string name
- pid_t pid
- unsigned jobid
- bool stopped
- bool dead

6.10.1 Detailed Description

Estrutura que representa um job. Guarda os dados necessarios para controlar processos.

See also

Executor

6.10.2 Constructor & Destructor Documentation

6.10.2.1 Executor::Job::Job ( )

6.10.3 Member Data Documentation

6.10.3.1 bool **Executor::Job::dead**

6.10.3.2 unsigned **Executor::Job::jobid**

6.10.3.3 std::string **Executor::Job::name**

6.10.3.4 pid_t **Executor::Job::pid**

6.10.3.5 bool **Executor::Job::stopped**

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

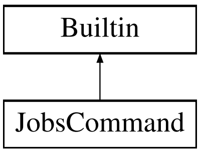
- Executor.hpp
- Executor.cpp

6.11 JobsCommand Class Reference

Classe que implementa o comando jobs.

`#include <Builtin.hpp>`

Inheritance diagram for JobsCommand:



6.11.1 Detailed Description

Classe que implementa o comando jobs. Sintaxe: jobs

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

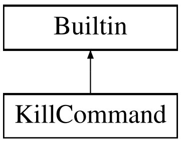
- Builtin.hpp
- Builtin.cpp

6.12 KillCommand Class Reference

Classe que implementa o comando kill.

`#include <Builtin.hpp>`

Inheritance diagram for KillCommand:



6.12.1 Detailed Description

Classe que implementa o comando kill. Sintaxe: kill %<JOBID>

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

- Builtin.hpp
- Builtin.cpp

6.13 MyTypo Class Reference

Firulas.

`#include <MyTypo.hpp>`

Public Types

- enum Style {
  NORMAL = 0, BOLD, UNDER = 4, BLINK,
  INVERT = 7 }
- enum Foreground {
  BLACK = 30, RED, GREEN, BROWN,
  BLUE, PURPLE, CYAN, GRAY }
- enum Background {
  B_TRANSPARENT = -1, B_BLACK = 40, B_RED, B_GREEN,
  B_BROWN, B_BLUE, B_PURPLE, B_CYAN,
  B_GRAY }

Public Member Functions

- void set_foreground (int color)
- void set_background (int color)
- void set_style (int color)
- int get_foreground (void)
- int get_background (void)
- int get_style (void)
- bool get_opened (void)
- MyTypo (int style=NORMAL, int foreground=BLACK, int background=B_TRANSPARENT)
- void toogleOpened (void)
- void setParameters (int style=NORMAL, int foreground=BLACK, int background=B_TRANSPARENT)
- std::ostream & openTag (std::ostream &ost)
- std::ostream & closeTag (std::ostream &ost)

6.13.1 Detailed Description

Firulas. Serve apenas para deixar a saida mais bonita, com diferentes cores e estilos para a fonte, no terminal

6.13.2 Member Enumeration Documentation

6.13.2.1 enum MyTypo::Background

Enumerator:

B_TRANSPARENT

B_BLACK

B_RED

B_GREEN

B_BROWN

B_BLUE

B_PURPLE

B_CYAN

B_GRAY

6.13.2.2 enum MyTypo::Foreground

Enumerator:

BLACK

RED

GREEN

BROWN

BLUE

PURPLE

CYAN

GRAY

6.13.2.3 enum MyTypo::Style

Enumerator:

NORMAL

BOLD

UNDER

BLINK

INVERT

6.13.3 Constructor & Destructor Documentation

6.13.3.1 MyTypo::MyTypo (int *style* = NORMAL, int *foreground* = BLACK, int *background* = B_TRANSPARENT)

6.13.4 Member Function Documentation

6.13.4.1 std::ostream & MyTypo::closeTag (std::ostream & *ost*)

6.13.4.2 int MyTypo::get_background (void)

6.13.4.3 int MyTypo::get_foreground (void)

6.13.4.4 bool MyTypo::get_opened (void)

6.13.4.5 int MyTypo::get_style (void)

6.13.4.6 std::ostream & MyTypo::openTag (std::ostream & *ost*)

6.13.4.7 void MyTypo::set_background (int *color*)

6.13.4.8 void MyTypo::set_foreground (int *color*)

6.13.4.9 void MyTypo::set_style (int *color*)

6.13.4.10 void MyTypo::setParameters (int *style* = NORMAL, int *foreground* = BLACK, int *background* = B_TRANSPARENT)

6.13.4.11 void MyTypo::toogleOpened (void)

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

- MyTypo.hpp
- MyTypo.cpp

6.14 Parser Class Reference

Utilizado para converter a entrada do usuario em uma CommandLine.

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

Public Member Functions

- Parser ()
- CommandLine ∗ readCommandLine ()
- bool newLine ()

6.14.1 Detailed Description

Utilizado para converter a entrada do usuario em uma CommandLine. A linha de usuario deve ter a seguinte forma

<comando><paramentro>...<parametro>|...|<comando>...[< <entrada>][[1>|>|>>][2>]<saida>|&><saida>][&]

6.14.2 Constructor & Destructor Documentation

6.14.2.1 Parser::Parser ( )

6.14.3 Member Function Documentation

6.14.3.1 bool Parser::newLine ( )

Returns

Verdadeiro caso esteja em uma nova linha de comando (nao necessariamente endl, mas fim da pipeline por &

6.14.3.2 CommandLine ∗ Parser::readCommandLine ( )

Returns

Le e interpreta uma linha de comando dada pelo usuario no stdin

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

- Parser.hpp
- Parser.cpp

6.15 Program Class Reference

```
#include <Program.hpp>
```

Public Member Functions

- int run ()

6.15.1 Member Function Documentation

6.15.1.1 int Program::run ( )

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

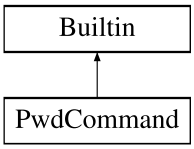
- Program.hpp
- Program.cpp

6.16 PwdCommand Class Reference

Classe que implementa o comando pwd.

```
#include <Builtin.hpp>
```

Inheritance diagram for PwdCommand:



6.16.1 Detailed Description

Classe que implementa o comando pwd. Sintaxe: pwd

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

- Builtin.hpp
- Builtin.cpp

Chapter 7

File Documentation

7.1 Builtin.cpp File Reference

```
#include "Builtin.hpp"
#include <unistd.h>
#include <iostream>
#include <list>
#include "MyTypo.hpp"
#include <cstdio>
#include <cstdlib>
#include <signal.h>
#include <sys/types.h>
#include <sys/wait.h>
#include "Handlers.hpp"
```

7.2 Builtin.hpp File Reference

```
#include "Executor.hpp"
```

Classes

- class Builtin

  Classe abstrata para comandos Built-In.

- class CdCommand

  Classe que implementa o comando cd.

- class PwdCommand

  Classe que implementa o comando pwd.

- class BgCommand

  Classe que implementa o comando bg.

- class FgCommand

  Classe que implementa o comando fg.

- class JobsCommand

  Classe que implementa o comando jobs.

- class ExitCommand

  Classe que implementa o comando exit.

- class KillCommand

  Classe que implementa o comando kill.

- class EchoCommand

  Class que implementa o comando echo.

7.3 Command.cpp File Reference

```
#include "Command.hpp"
```

7.4 Command.hpp File Reference

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

Classes

- class Command

  Representa um comando entrado pelo usuario. O comando representa tudo o que esta antes de um pipe (|) ou & ou final de linha.

7.5 CommandLine.cpp File Reference

```
#include "CommandLine.hpp"
```

7.6 CommandLine.hpp File Reference

```
#include "Command.hpp"
#include <list>
#include <string>
```

Classes

- class CommandLine

  Represeta uma linha de comando.

7.7 Executor.cpp File Reference

```
#include "Executor.hpp"
#include <unistd.h>
#include <fcntl.h>
#include <cstdio>
#include <sys/stat.h>
#include <sys/wait.h>
#include <sys/types.h>
#include <iostream>
#include <cstdlib>
#include <utility>
#include "MyTypo.hpp"
#include "Handlers.hpp"
```

7.8 Executor.hpp File Reference

```
#include <string>
#include <vector>
#include <list>
#include <sys/types.h>
#include "Command.hpp"
#include "CommandLine.hpp"
#include <termios.h>
```

```
#include "Builtin.hpp"
#include <map>
```

Classes

- class Executor

  Responsavel pela execucao. Executa uma linha de comando.

- struct Executor::Job

  Estrutura que representa um job. Guarda os dados necessarios para controlar processos.

7.9 Handlers.cpp File Reference

```
#include "Handlers.hpp"
#include <iostream>
```

Variables

- bool deathStatus = false

7.9.1 Variable Documentation

7.9.1.1 bool **deathStatus** = false

7.10 Handlers.hpp File Reference

Namespaces

- namespace handlers

  Namespace relacionado as funcoes que manipulam interrupcoes de software.

Functions

- void handlers::sigChildHandler (int)

  Handler para o sinal SIGCHLD. Levanta uma flag dizendo que um sinal vindo de um provesso filho foi lancado.

- bool handlers::getDeathStatus ()
- void handlers::setDeathStatusFalse ()

Altera a flag de sinais recebidos. Altera o valor da flag de sinais recebidos para false.

- void handlers::setDeathStatusTrue ()

  Altera a flag de sinais recebidos. Altera o valor da flag de sinais recebidos para true.

7.11 main.cpp File Reference

```
#include "Program.hpp"
```

Functions

- int main (void)

7.11.1 Function Documentation

7.11.1.1 int main (void)

7.12 MyTypo.cpp File Reference

```
#include "MyTypo.hpp"
```

Functions

- std::ostream & operator<< (std::ostream &ost, MyTypo &mt)

7.12.1 Function Documentation

7.12.1.1 std::ostream& operator<< (std::ostream & *ost*, MyTypo & *mt*)

7.13 MyTypo.hpp File Reference

```
#include <iostream>
```

Classes

- class MyTypo

  Firulas.

Functions

- std::ostream & operator<< (std::ostream &ost, MyTypo &mt)

7.13.1 Function Documentation

7.13.1.1 std::ostream& operator<< (std::ostream & *ost,* MyTypo & *mt*)

7.14 Parser.cpp File Reference

```
#include "Parser.hpp"
#include <iostream>
#include <string>
#include "Command.hpp"
#include <signal.h>
```

7.15 Parser.hpp File Reference

```
#include <string>
#include "CommandLine.hpp"
```

Classes

- class Parser

  Utilizado para converter a entrada do usuario em uma CommandLine.

7.16 Program.cpp File Reference

```
#include <iostream>
#include "CommandLine.hpp"
#include "Parser.hpp"
#include "MyTypo.hpp"
#include "Executor.hpp"
#include <signal.h>
#include "Handlers.hpp"
#include <map>
#include "Builtin.hpp"
```

```
#include "Program.hpp"
```

7.17 Program.hpp File Reference

Classes

- class Program

Index