

myshell

Generated by Doxygen 1.8.13

Contents

| | | |
|----------|-------------------------------------|----------|
| 1 | SysNet1Team19Projects | 1 |
| 2 | Bug List | 3 |
| 3 | Data Structure Index | 5 |
| 3.1 | Data Structures | 5 |
| 4 | File Index | 7 |
| 4.1 | File List | 7 |
| 5 | Data Structure Documentation | 9 |
| 5.1 | PARAM Struct Reference | 9 |
| 5.1.1 | Detailed Description | 9 |
| 5.1.2 | Field Documentation | 9 |
| 5.1.2.1 | argumentCount | 9 |
| 5.1.2.2 | argumentVector | 10 |
| 5.1.2.3 | background | 10 |
| 5.1.2.4 | inputRedirect | 10 |
| 5.1.2.5 | outputRedirect | 10 |
| 5.1.2.6 | prompt | 10 |

| | |
|--------------------------------------|-----------|
| 6 File Documentation | 11 |
| 6.1 myshell.c File Reference | 11 |
| 6.1.1 Detailed Description | 11 |
| 6.2 parse.c File Reference | 11 |
| 6.2.1 Detailed Description | 12 |
| 6.2.2 Function Documentation | 12 |
| 6.2.2.1 debugMode() | 12 |
| 6.2.2.2 exitMyShell() | 13 |
| 6.2.2.3 getUserInput() | 13 |
| 6.2.2.4 initMyShell() | 13 |
| 6.2.2.5 parseUserInput() | 14 |
| 6.2.2.6 printParams() | 14 |
| 6.2.2.7 runUserInput() | 15 |
| 6.3 parse.h File Reference | 15 |
| 6.3.1 Detailed Description | 16 |
| 6.3.2 Macro Definition Documentation | 16 |
| 6.3.2.1 MAXARGS | 16 |
| 6.3.3 Function Documentation | 17 |
| 6.3.3.1 debugMode() | 17 |
| 6.3.3.2 exitMyShell() | 17 |
| 6.3.3.3 getUserInput() | 17 |
| 6.3.3.4 initMyShell() | 18 |
| 6.3.3.5 parseUserInput() | 18 |
| 6.3.3.6 printParams() | 19 |
| 6.3.3.7 runUserInput() | 19 |
| 6.3.4 Variable Documentation | 19 |
| 6.3.4.1 arguments | 19 |
| 6.3.4.2 debug | 20 |
| Index | 21 |

Chapter 1

SysNet1Team19Projects

Chapter 2

Bug List

File [myshell.c](#)

No Known Bugs

File [parse.c](#)

No Known Bugs

File [parse.h](#)

No Known Bugs

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

| | |
|------------------------------------|---|
| PARAM | |
| Structure for user input | 9 |

Chapter 4

File Index

4.1 File List

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

| | | |
|---------------------------|---|----|
| myshell.c | Simple shell | 11 |
| parse.c | Myshell controls | 11 |
| parse.h | Function Prototypes for parse.c | 15 |

Chapter 5

Data Structure Documentation

5.1 PARAM Struct Reference

a structure for user input

```
#include <parse.h>
```

Data Fields

- char * [inputRedirect](#)
- char * [outputRedirect](#)
- int [background](#)
- int [argumentCount](#)
- char * [argumentVector](#) [[MAXARGS](#)]
- char [prompt](#) [[MAXCOMMANDLENGTH](#)]

5.1.1 Detailed Description

a structure for user input

[PARAM](#) stores user command line input to execute with myshell

5.1.2 Field Documentation

5.1.2.1 argumentCount

```
int PARAM::argumentCount
```

number of tokens in argument vector

5.1.2.2 argumentVector

`char* PARAM::argumentVector[MAXARGS]`

array of strings

5.1.2.3 background

`int PARAM::background`

either 0 (false) or 1 (true)

5.1.2.4 inputRedirect

`char* PARAM::inputRedirect`

file name or NULL

5.1.2.5 outputRedirect

`char* PARAM::outputRedirect`

file name or NULL

5.1.2.6 prompt

`char PARAM::prompt[MAXCOMMANDLENGTH]`

array for user input

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

- [parse.h](#)

Chapter 6

File Documentation

6.1 myshell.c File Reference

a simple shell

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "parse.h"
```

Functions

- int **main** (int argc, char **argv)

6.1.1 Detailed Description

a simple shell

this file contains the entry point for myshell, main() Copyright (c) 2018 All rights reserved.

Author

Timothy L.J. Stewart
Jeff Murphy

Bug No Known Bugs

6.2 parse.c File Reference

myshell controls

```
#include "parse.h"
```

Functions

- void `initMyShell ()`
initializes vars
- void `getUserInput ()`
prompts and stores users input
- void `parseUserInput ()`
parse user input
- void `debugMode (int argc, char **argv)`
check for -Debug
- void `printParams (Param_t *param)`
prints user parameters
- void `runUserInput ()`
run the user commands
- bool `exitMyShell ()`
checks for exit command

6.2.1 Detailed Description

myshell controls

this file contains the fucntions that control myshell Copyright (c) 2018 All rights reserved.

Author

Timothy L.J. Stewart
Jeff Murphy

Bug No Known Bugs

6.2.2 Function Documentation

6.2.2.1 debugMode()

```
void debugMode (
    int argc,
    char ** argv )
```

check for -Debug

when users calls ./myshell -Debug, debugMode will set global debug to true

Parameters

| | |
|-------------|----------------------------------|
| <i>argc</i> | argument count for myshell call |
| <i>argv</i> | argument vector for myshell call |

Returns

void

6.2.2.2 exitMyShell()

```
bool exitMyShell ( )
```

checks for exit command

if exit command is given, checks for child process before termination

Parameters

| | |
|-------------|--|
| <i>void</i> | |
|-------------|--|

Returns

bool

6.2.2.3 getUserInput()

```
void getUserInput ( )
```

prompts and stores users input

prompts and stores users input

Parameters

| | |
|-------------|--|
| <i>void</i> | |
|-------------|--|

Returns

void

6.2.2.4 initMyShell()

```
void initMyShell ( )
```

initializes vars

set all memory locations to initial values, NULL, O

Parameters

| | |
|-------------|--|
| <i>void</i> | |
|-------------|--|

Returns

void

6.2.2.5 parseUserInput()

```
void parseUserInput ( )
```

parse user input

during myself execution parseUserInput will parse the input into [PARAM](#) struct

Parameters

| | |
|-------------|--|
| <i>void</i> | |
|-------------|--|

Returns

void

6.2.2.6 printParams()

```
void printParams (
    Param\_t * param )
```

prints user parameters

if in debug mode printParams will print out the [PARAM](#) struct

Parameters

| | |
|--------------|---|
| <i>param</i> | a pointer to the PARAM struct |
|--------------|---|

Returns

void

6.2.2.7 runUserInput()

```
void runUserInput ( )
```

run the user commands

runs the parsed user input handles redirects and backgrounding, implements forks, exec and wait system calls

Parameters

| | |
|-------------|--|
| <i>void</i> | |
|-------------|--|

Returns

void

6.3 parse.h File Reference

Function Prototypes for [parse.c](#).

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <sys/wait.h>
#include <sys/types.h>
#include <unistd.h>
```

Data Structures

- struct [PARAM](#)
a structure for user input

Macros

- #define [MAXARGS](#) 32
- #define **MAXCOMMANLENGTH** 256
- #define **EXIT** "exit"

Typedefs

- typedef struct [PARAM](#) Param_t
type definition of [PARAM](#)

Functions

- void `initMyShell ()`
initializes vars
- void `getUserInput ()`
prompts and stores users input
- void `parseUserInput ()`
parse user input
- void `debugMode (int argc, char **argv)`
check for -Debug
- void `printParams (Param_t *param)`
prints user parameters
- void `runUserInput ()`
run the user commands
- bool `exitMyShell ()`
checks for exit command

Variables

- `Param_t` arguments
global var of struct `PARAM`
- bool `debug`
global var for debug mode

6.3.1 Detailed Description

Function Prototypes for `parse.c`.

this file contains function prototypes, data structures and global vars for `parse.c` Copyright (c) 2018 All rights reserved.

Author

Timothy L.J. Stewart
Jeff Murphy

Bug No Known Bugs

6.3.2 Macro Definition Documentation

6.3.2.1 MAXARGS

```
#define MAXARGS 32
```

dont test program with more than this many tokens for input

6.3.3 Function Documentation

6.3.3.1 debugMode()

```
void debugMode (
    int argc,
    char ** argv )
```

check for -Debug

when users calls ./myshell -Debug, debugMode will set global debug to true

Parameters

| | |
|-------------|----------------------------------|
| <i>argc</i> | argument count for myshell call |
| <i>argv</i> | argument vector for myshell call |

Returns

void

6.3.3.2 exitMyShell()

```
bool exitMyShell ( )
```

checks for exit command

if exit command is given, checks for child process before termination

Parameters

| | |
|-------------|--|
| <i>void</i> | |
|-------------|--|

Returns

bool

6.3.3.3 getUserInput()

```
void getUserInput ( )
```

prompts and stores users input

prompts and stores users input

Parameters

| | |
|-------------|--|
| <i>void</i> | |
|-------------|--|

Returns

void

6.3.3.4 initMyShell()

```
void initMyShell ( )
```

initializes vars

set all memory locations to initial values, NULL, 0

Parameters

| | |
|-------------|--|
| <i>void</i> | |
|-------------|--|

Returns

void

6.3.3.5 parseUserInput()

```
void parseUserInput ( )
```

parse user input

during myshell execution parseUserInput will parse the input into [PARAM](#) struct

Parameters

| | |
|-------------|--|
| <i>void</i> | |
|-------------|--|

Returns

void

6.3.3.6 printParams()

```
void printParams (
    Param_t * param )
```

prints user parameters

if in debug mode printParams will print out the [PARAM](#) struct

Parameters

| | |
|--------------|---|
| <i>param</i> | a pointer to the PARAM struct |
|--------------|---|

Returns

void

6.3.3.7 runUserInput()

```
void runUserInput ( )
```

run the user commands

runs the parsed user input handles redirects and backgrounding, implements forks, exec and wait system calls

Parameters

| | |
|-------------|--|
| <i>void</i> | |
|-------------|--|

Returns

void

6.3.4 Variable Documentation

6.3.4.1 arguments

`arguments`

global var of struct [PARAM](#)

an instance of [PARAM](#)

`arguments` is an instance of struct [PARAM](#)

6.3.4.2 debug

debug

global var for debug mode

either 0 (false) or 1 (true)

debug is set in debugMode function with switch by starting myshell -Debug, defeault is false

Index

- argumentCount
 - PARAM, 9
- argumentVector
 - PARAM, 9
- arguments
 - parse.h, 19
- background
 - PARAM, 10
- debug
 - parse.h, 19
- debugMode
 - parse.c, 12
 - parse.h, 17
- exitMyShell
 - parse.c, 13
 - parse.h, 17
- getUserInput
 - parse.c, 13
 - parse.h, 17
- initMyShell
 - parse.c, 13
 - parse.h, 18
- inputRedirect
 - PARAM, 10
- MAXARGS
 - parse.h, 16
- myshell.c, 11
- outputRedirect
 - PARAM, 10
- PARAM, 9
 - argumentCount, 9
 - argumentVector, 9
 - background, 10
 - inputRedirect, 10
 - outputRedirect, 10
 - prompt, 10
- parse.c, 11
 - debugMode, 12
 - exitMyShell, 13
 - getUserInput, 13
 - initMyShell, 13
 - parseUserInput, 14
 - printParams, 14
- runUserInput, 14
- parse.h, 15
 - arguments, 19
 - debug, 19
 - debugMode, 17
 - exitMyShell, 17
 - getUserInput, 17
 - initMyShell, 18
 - MAXARGS, 16
 - parseUserInput, 18
 - printParams, 18
 - runUserInput, 19
- parseUserInput
 - parse.c, 14
 - parse.h, 18
- printParams
 - parse.c, 14
 - parse.h, 18
- prompt
 - PARAM, 10
- runUserInput
 - parse.c, 14
 - parse.h, 19