项目结构

```
|- err⋅h 错误处理宏定义
|- def.h 参数宏定义
|- main.c 主程序
|- ps.c 输出路径和命令提示符
|- parse.c 解析输入
|- cmd.c 运行命令
```

使用方式

\$./myshell

```
功能概述
```

\$ cd myshell && make

路径提示

对当前所在的路径进行提示,用绿色表示路径,用 \$ 分隔命令和路径。

~/source/myshell \$

如图所示,还支持路径折叠(即将 HOME 目录折叠成 ~)。

myshell - ./myshell.o - ./myshell.o - myshell.o - 80×24

?? img/ ?? report.md ~/source/myshell \$ ls -a

~/source/myshell \$ git status --short

cmd.c

README.md

~/source/myshell \$ grep -r "#include \".*\.h\"" .

多参数支持

```
.DS_Store
                    cmd.h
                                                               test
                                  main.c
                                                parse.o
      •git
                    cmd.o
                                  main.o
                                                ps.c
     Makefile
                    def.h
                                  myshell.o
                                                ps.h
      ~/source/myshell $
字符转义
找到所有 "#include "*.h"
```

myshell - ./myshell.o - ./myshell.o - myshell.o - 84×23

myshell - ./myshell.o - ./myshell.o - myshell.o - 80×24

parse.c

parse.h

ps.o

report.md

err.h

img

./parse.h:#include "cmd.h" ./cmd.c:#include "cmd.h" ./cmd.c:#include "stdlib.h" ./main.c:#include "parse.h"

./main.c:#include "ps.h" ./main.c:#include "def.h" ./parse.c:#include "parse.h"
./parse.c:#include "cmd.h"

./parse.c:#include "def.h"

```
./parse.c:#include "err.h"
      ~/source/myshell $
管道支持
统计 c 文件数量:
      myshell - ./myshell.o - ./myshell.o - myshell.o - 80×24
```

~/source/myshell \$ ls -la | sed -n '/.*\.c/p' | wc -l

~/source/myshell \$

```
文件重定向
      myshell - ./myshell.o - ./myshell.o - myshell.o - 86×25
      ~/source/myshell $ ls > 1.txt
      ~/source/myshell $ cat 1.txt
      1.txt
      Makefile
      README.md
      cmd.c
```

parse.o ps.c ps.h ps.o

test

cmd.h cmd.o def.h err.h img

main.c main.o

myshell.o parse.c parse.h

report.md

~/source/myshell \$ wc < 1.txt

```
20 137
                20
        ~/source/myshell $
错误提示
        myshell - ./myshell.o - ./myshell.o - myshell.o - 86×25
        [~/source/myshell(master x) ./myshell.o
       ~/source/myshell $ ls|
Error: Syntax error!
~/source/myshell $ ls <|
        Error: Syntax error!
       ~/source/myshell $ wrong_command
wrong_command: No such file or directory
        Error: process exited with status 127 ~/source/myshell $
```

myshell - ./myshell.o - ./myshell.o - myshell.o - 84×23

内部命令 cd

~/source/myshell \$ cd ..
~/source \$ cd myshell

~/source/myshell \$

内部命令 exit: 退出

```
myshell — bash — bash — bash — 84×23
      ~/source/myshell $ ls
     Makefile
                     cmd.o
                                   main.c
                                                  parse.h
                                                                 ps.o
      README.md
                     def.h
                                   main.o
                                                                 report.md
                                                  parse.o
      cmd.c
                     err.h
                                   myshell.o
                                                                 test
                                                  ps.c
      cmd.h
                     img
                                   parse.c
                                                  ps.h
      ~/source/myshell $ exit
      bash-3.2$
系统调用
```

• execvp():调用命令 • waitpid: 主进程等待子进程结束并获取返回值 流程图

解析输入 (parse)

str

token

token_list

parse

commands

执行单个命令

开始

cd

chdir 或报错

管道重定向

dup2

外部命令

文件重定向

open + dup2

waitpid

fork

子进程

执行指令

execvp

判断错误 + 错误处理

errno

结束子进程并返回状态

主进程错误处理反馈

• exit(): 终止子进程

• getcwd(): 获取所在路径

• chdir(): 进入路径, 用于 cd 命令

• dup2(): 复制文件描述符,用于文件重定向

• getenv(): 获取 \$HOME 环境变量,用于路径折叠

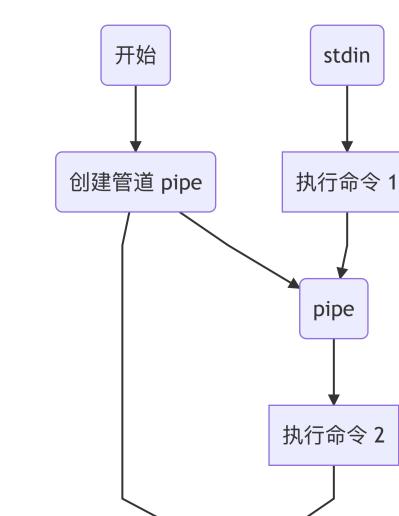
• open() / close(): 打开/关闭文件描述符, 用于文件重定向

ps.c

cmd.c

exit 0

exit



pipe

pipe

执行命令 N

stdout

组合多个命令(管道)

源代码 项目开源在 roife/Myshell。