

MyShell 设计文档

设计文档目录：

- 第一部分：需求概述
- 第二部分：使用说明
- 第三部分：实现步骤
- 第四部分：功能展示

第一部分：功能（需求）概述

使用 C 语言编写一个简单的 Shell（MyShell），为用户提供 Linux 的命令行式交互界面。

- 开发与运行环境：Linux 发行版 Ubuntu 22.04.4 LTS
- 可以运行 **内部指令**（`cd` 和 `exit`，暂不支持 `history`）
- 可以运行 **外部指令**（如 `ls`，`ps` 等等，以及其他用户的可执行文件）
- 支持标准输入和标准输出 **重定向**
- 可以通过 **管道** 连接多个命令

第二部分：使用说明

项目结构：

```
.
├── MyShell 设计文档.pdf      # 设计文档
├── main.c                   # MyShell 源文件
├── Makefile
├── case                     # 测试用例
│   ├── hello.c
│   ├── isOdd.c
│   ├── plus.c
│   ├── square.c
│   ├── sub1.c
│   └── sum.c
```

编译 MyShell 源文件：

```
$ make
```

运行 MyShell 可执行文件

```
$ make run
```

生成辅助测试用例

```
$ make test
```

第三部分：实现步骤

命令的解析：层次抽象的文法解析

我们当然可以直接对标准输入的一列字符流进行分析，解析出命令。

但计算机程序的构造和解析强调 **抽象（Abstraction）**，我们可以采用适当的抽象，进行 **层次化** 解析，从而实现对输入字符流的文法解析，如图 1 所示。

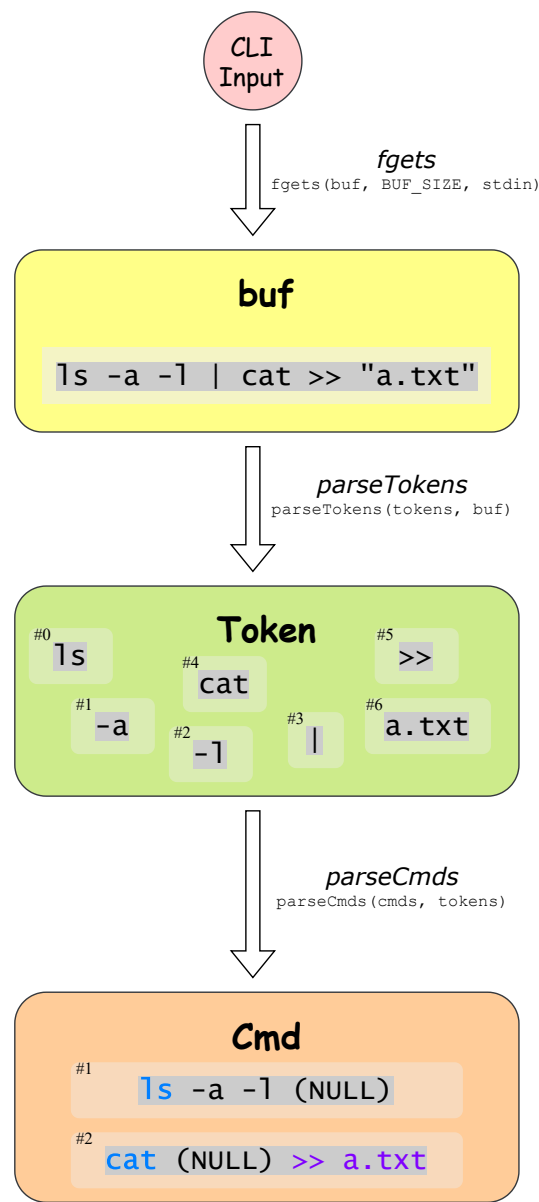


图 1：命令解析流程图

Token

`fgets` 从标准输入后得到 `buf`，然后 `parseTokens` 函数将字符流拆解为一系列原子单元 `Token`（不可再细分的最小单元）。

`Token` 结构体不但记录字符串信息（`char *str` 和 `unsigned int len`），而且划分单元的类型，为下一步进行做好铺垫。

其中包括 7 种类型：`TOKEN_SYM`（如 `cd`，`-l`，`xxx.x`）、`TOKEN_STR`（如 `"a.txt"`，`'hello unix'`）、`TOKEN_PIPE`（即 `|`）、`TOKEN_REDIRECT_IN`（即 `<`）、`TOKEN_REDIRECT_OUT`（即 `>`）、`TOKEN_REDIRECT_APD`（即 `>>`）。

```
typedef enum {
    TOKEN_SYM, TOKEN_STR, TOKEN_PIPE, TOKEN_END, TOKEN_REDIR_IN, TOKEN_REDIR_OUT,
    TOKEN_REDIR_APD
} Token_t;

typedef struct {
    Token_t type;
    char *str;
    unsigned int len;
} Token;
```

Cmd

接着上一步，若干 `Token` 将构成一个 `Cmd`，管道 `|` 连接的多个 `Cmd` 正是用户最开始给出的一条“指令”。

`Cmd` 结构体记录可执行文件的路径 `cmd_target` 和参数列表 `cmd_args`，二者将被 `execvp` 系统调用使用。

除此之外，`Cmd` 结构体还记录重定向路径信息 `cmd_file_in`、`cmd_file_out`、`cmd_file_apd`（若无则为 `NULL`）。以上信息（如有）将被 `dup2` 系统调用用于复制文件描述符表。

```
typedef struct {
    char *cmd_file_in, *cmd_file_out, *cmd_file_apd, *cmd_target;
    char **cmd_args;
} Cmd;
```

交由 `parseCmds(cmds, tokens)` 完成以上步骤之后，文法解析的部分就大功告成了。

命令的执行：系统调用与进程 fork

在命令的执行 `execCmd` 和 `execCmdAll` 部分，我们使用到的系统调用如下

系统调用	作用
<code>getcwd</code>	获取当前目录
<code>chdir</code>	切换当前目录
<code>exit</code>	出现异常或使用 <code>exit</code> 命令时终止进程
<code>open</code>	打开文件
<code>dup2</code>	复制文件描述符表
<code>fork</code>	创建进程（与 <code>exec</code> 类命令搭配使用）
<code>execvp</code>	执行程序（与 <code>fork</code> 命令搭配使用）
<code>wait</code>	等待进程
<code>pipe</code>	创建管道
<code>close</code>	关闭管道读/写端

基于前面解析出的 `Cmd` 信息，便可以执行命令：

- 对于内部指令（在 MyShell 中为 `cd` 和 `exit`），直接使用相应的系统调用；
- 对应外部资料，将通过 `fork` 创建子进程，子进程使用 `execvp` 来执行程序。

命令的执行过程如下面的流程图所示。

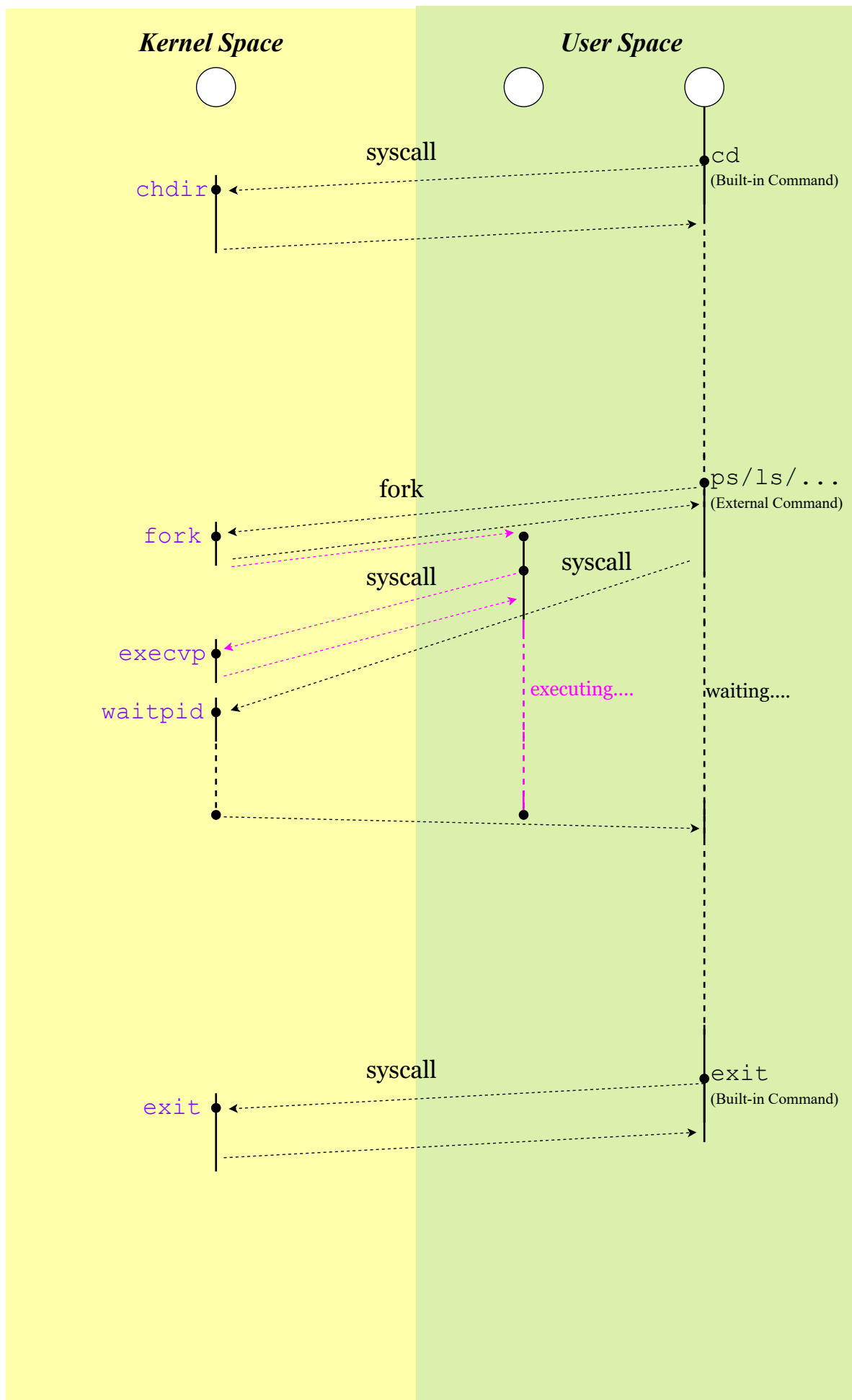


图 2: 命令执行流程图

外部命令的执行

不带参数与带参数

```
xyk@xyk-virtual-machine: /mnt/hgfs/_share/RosySpring
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ps
  PID TTY          TIME CMD
 2833 pts/0        00:00:00 bash
 3931 pts/0        00:00:00 make
 3932 pts/0        00:00:00 main
 3936 pts/0        00:00:00 ps
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ls -a -l
total 190
drwxrwxrwx 1 root root  4096 5月 14 22:11 .
drwxrwxrwx 1 root root    0 5月 14 19:34 ..
-rwxrwxrwx 1 root root 1075 5月 13 16:54 1.txt
-rwxrwxrwx 1 root root   194 5月 14 19:59 a.txt
-rwxrwxrwx 1 root root   131 5月 13 16:43 deal.py
drwxrwxrwx 1 root root  4096 5月 14 17:56 .git
-rwxrwxrwx 1 root root 16000 5月 14 17:49 hello
-rwxrwxrwx 1 root root   338 5月 14 17:49 hello.c
-rwxrwxrwx 1 root root 21416 5月 14 22:07 main
-rwxrwxrwx 1 root root 11147 5月 14 22:06 main.c
-rwxrwxrwx 1 root root 61169 5月 14 16:52 main.exe
-rwxrwxrwx 1 root root   105 5月 14 01:19 Makefile
-rwxrwxrwx 1 root root 16016 5月 13 21:20 test
-rwxrwxrwx 1 root root   362 5月 13 21:19 test.c
-rwxrwxrwx 1 root root 54277 5月 13 21:14 test.exe
drwxrwxrwx 1 root root    0 5月 14 22:11 tmpDir
drwxrwxrwx 1 root root    0 5月 13 16:31 .vscode
Arren's MyShell:/mnt/hgfs/_share/RosySpring $
```

带固定数量参数

```
xyk@xyk-virtual-machine: /mnt/hgfs/_share/RosySpring
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ gcc -o plus plus.c
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ plus
Unreachable Code!
Error Occurred in the EXECUTION of your command!
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./plus
error! usage: ./plus <arg1> <arg2>'
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./plus 1 2
1 + 2 = 3
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./plus -1000 1999
-1000 + 1999 = 999
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./plus 988937 28276
988937 + 28276 = 1017213
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./plus 2024 514
2024 + 514 = 2538
Arren's MyShell:/mnt/hgfs/_share/RosySpring $
```

变长参数

```
xyk@xyk-virtual-machine: /mnt/hgfs/_share/RosySpring
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ gcc -o sum sum.c
sum.c: In function 'main':
sum.c:19:27: warning: format '%d' expects a matching 'int' argument [-Wformat=]
   19 |         printf(i == 1 ? "%d" : " + %d");
      |                        ~^
      |                        |
      |                        int
sum.c:19:37: warning: format '%d' expects a matching 'int' argument [-Wformat=]
   19 |         printf(i == 1 ? "%d" : " + %d");
      |                        ~^
      |                        |
      |                        int
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./sum 1 2 3 4 5 6 7 8 9 10
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./sum 1
1 = 1
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./sum 1 2
1 + 2 = 3
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./sum -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4
10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 + 0 + 1 + 2 + 3 + 4 = -45
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./sum 9999 -99999 99999999 9999999
9999 + 99999 + 99999999 + 9999999 = 109909998
Arren's MyShell:/mnt/hgfs/_share/RosySpring $
```

支持文件重定向

```
xyk@xyk-virtual-machine: /mnt/hgfs/_share/RosySpring
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./square 2 > 1.txt
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ cat 1.txt
4
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./isOdd < 1.txt
4 is even!
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./isOdd < 1.txt >> 1.txt
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ cat 1.txt
4
4 is even!
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ gcc -o hello hello.c
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./hello >> 1.txt
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ cat 1.txt
4
4 is even!
hello
argc:1
"hello" runs ok!
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./square 3 | ./sub >> 1.txt
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ cat 1.txt
4
4 is even!
hello
argc:1
"hello" runs ok!
Unreachable Code!
Error Occurred in the EXECUTION of your command!
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./square 3 | ./sub1 >> 1.txt
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ cat 1.txt
4
4 is even!
hello
argc:1
"hello" runs ok!
Unreachable Code!
Error Occurred in the EXECUTION of your command!
8
Arren's MyShell:/mnt/hgfs/_share/RosySpring $
```


支持管道

```
xyk@xyk-virtual-machine: /mnt/hgfs/_share/RosySpring
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ cat < a.txt | grep 'int' | awk '{print NR,$0}'
1 int main(int argc, char **argv){
2     printf("error! usage: ./sum <arg1> <arg2> <arg3> ...'\n");
3     int sum = 0;
4     int tmp;
5     for (int i = 1; i < argc; ++i) {
6         printf(i == 1 ? "%d" : " + %d");
7         printf(" = %d\n",sum);
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./square 3 | ./isOdd
9 is odd
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./square -2 | ./isOdd
4 is even!
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ gcc -o sub1 sub1.c
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./square -1 | ./sub1 | ./sub1 | ./sub1 | ./sub1
-3
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./square 3 | ./sub1 | ./sub1
7
Arren's MyShell:/mnt/hgfs/_share/RosySpring $
```

错误提示

```
Arren's MyShell:/mnt/hgfs/_share/RosySpring $ echo "2442'
Error Occurred in Quotation Marks near Your String!
make: *** [Makefile:8: run] Error 255

Arren's MyShell:/mnt/hgfs/_share/RosySpring $ cd "3"
Error Occurred in you cd path!
Check your path!
make: *** [Makefile:8: run] Error 255
xyk@xyk-virtual-machine:/mnt/hgfs/_share/RosySpring$

Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./isOdd <
Error Occurred in your redirection path!
make: *** [Makefile:8: run] Error 255

Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ps |
Error Occurred near you pipe!
Empty command after pipe!
make: *** [Makefile:8: run] Error 255

Arren's MyShell:/mnt/hgfs/_share/RosySpring $ ./isOdd < < 23
Error Occurred in your redirection path!
make: *** [Makefile:8: run] Error 255
```


- 此外，我还加深了对 C 语言这门本身就 and Unix 系统紧密相关的语言的掌握，C 语言深入底层，向下直接能和系统打交道，向上又能服务用户。而且，“没学过指针，就不算学过 C 语言”，这次写 MyShell 时一些一级指针和二级指针、指针的深浅拷贝相关的调试经历，也让我对指针的掌握更加牢固。