OSlab1实验报告

一、完成目标

本次lab的目标是实现一个简易的命令行 shell (`esh`)，支持以下功能：

- 执行内置命令 `cd` 和 `paths`。
- 执行外部命令并支持管道和重定向。
- 实现命令的顺序执行。
- 支持后台任务管理。(未成功实现)
- 提供错误处理机制，确保用户能够看到相应的错误信息。

二、代码思路

由于代码中存在注释，因此只写出大概思路

1. **命令解析函数**：void parse_command(char *cmd, char **args);

   1、使用 `strtok` 函数将用户输入的命令字符串分割为命令和参数，以便后续处理。

   2、提供一个单独的函数 `parse_command` 来实现命令的分割，提高代码的可读性和可维护性。

2. **命令执行**：void execute_command(char *cmd);

   在 `execute_command` 函数中，根据命令的特性（管道、重定向、分号等）将命令路由到相应的处理函数。

   （1）**内置命令**内置命令直接调用相应的处理函数 `handle_cd` ， `handle_paths`，handle_exit,实现都比较容易

   handle_bg实现存在bug。

   （2）**外部命令**void execute_external_command(char **args);通过 `fork` 和 `exec` 系列函数执行。

3. **管道**：void execute_pipeline(char *cmd);

   实现了 `execute_pipeline` 函数，处理管道的命令。确保管道的最后一个命令的输出能够正确显示，同时在出现错误时输出错误信息。

   遍历所有命令，使用 `fork` 创建子进程。

   在每个子进程中：

   1、如果不是最后一个命令，将输出重定向到下一个管道的写端 (`pipefd[i * 2 + 1]`)。

   2、如果不是第一个命令，将输入重定向到前一个管道的读端 (`pipefd[(i - 1) * 2]`)。

   3、关闭所有管道的文件描述符，以避免资源泄漏。

   4、解析当前命令的参数，并调用 `execvp` 执行命令。如果 `execvp` 失败，打印错误信息并退出该子进程。

4. **重定向**void execute_redirection(char *cmd);

   实现了 `execute_redirection` 函数，处理重定向的命令。

5. **后台任务管理**：void handle_bg_processes();

具体思路是通过 `bg_job_t` 结构体存储后台任务信息，维护一个任务列表。提供 `handle_bg_processes` 函数定期检查后台任务状态，并输出已完成的任务信息。但最后并未成功实现

6. **错误处理**：void print_error_info()；

   框架代码给出

三、遇到困难

1. **管道实现**：

   管道的实现涉及到多个进程之间的通信，如何正确地使用 `pipe()` 和 `fork()` 函数比较耗费时间进行调试。

   经过多次测试和调试，我逐渐理解了进程间通信的基本原理，并实现了管道的功能。

2. **错误处理机制**：
   - 在管道命令中，如果某个命令执行失败，如何在保留最后一个命令输出的同时提供错误反馈。
   - 我设计了一个机制，先执行所有命令，并在最后输出错误信息，这样可以确保用户看到所有可能的输出。

四、可能存在的 bug

1. **未处理的外部命令错误**：

   有些外部命令在执行时可能产生未知bug

   需要在 `execvp` 调用后立即检查返回值，并通过 `print_error_info` 函数输出相关错误信息。

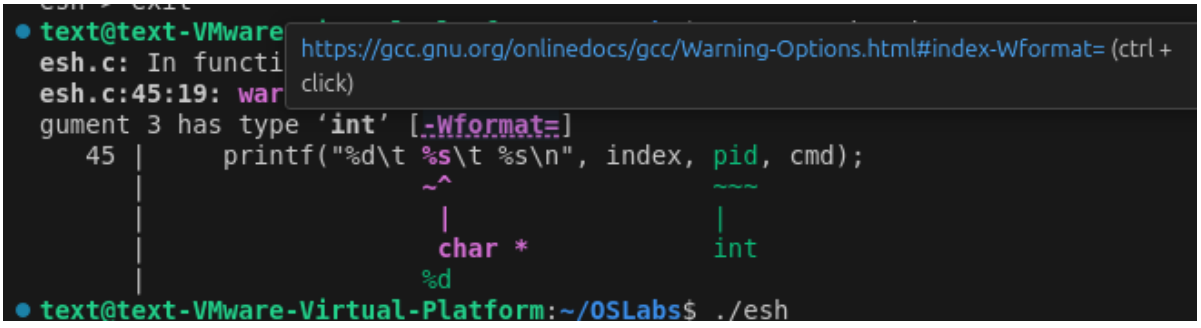
2. **命令格式问题**：

   对于复杂命令（如连续的管道或错误格式的输入），可能未能正确处理导致执行失败或输出错误。

   需要进一步优化命令解析逻辑，确保可以识别和处理不合法的输入格式。

3.

   由于没有实现bg后台功能因此修改了框架代码print_bg_info函数以通过编译




```c
void print_bg_info(int index, int pid, char *cmd) {
    printf("%d\t %d\t %s\n", index, pid, cmd);
    fflush(stdout);
}
```

五、本地测试结果截图

1、内置命令

正常cd和l功能实现，错误cd 报错

```
text@text-VMware-Virtual-Platform:~/OSLabs$ gcc -o esh esh.c
text@text-VMware-Virtual-Platform:~/OSLabs$ ./esh
esh > cd a
esh > ls
a1  a2  a3
esh > cd
An error has occurred
```

paths功能实现

```
An error has occurred
esh > paths
1        /bin
esh > paths a
esh > paths
1        a
esh >
```

2、执行外部命令

```
text@text-VMware-Virtual-Platform:~/OSLabs$ ./esh
esh > cd a
esh > rm -r b
rm: cannot remove 'b': No such file or directory
esh >
```

3、管道

成功实现

```
esh > exit
text@text-VMware-Virtual-Platform:~/OSLabs$ ./esh
esh > ls | echo 1
1
esh > cd a
esh > ech 1 | ls
a2  a3
An error has occurred
esh >
```

4、重定向

成功实现

```
text@text-VMware-Virtual-Platform:~/OSLabs$ ./esh
esh > cd a
esh > ls > 1
esh > cat 1
1
a2
a3
esh >
```

5、顺序执行

成功实现

```
esh > echo 1; ls
1
1  a2  a3
esh > ls
1  a2  a3
esh > ls
1  a2  a3
esh > ech 1;eoh 1;echo 1;
An error has occurred
An error has occurred
1
esh >
```

6、后台执行

未实现

五、bugs

```
text@text-VMware
esh.c: In functi
esh.c:45:19: war
gument 3 has type ‘int’ [-Wformat=]
   45 |     printf("%d\t %s\t %s\n", index, pid, cmd);
      |                  ~^                  ~~~
      |                   |                  |
      |                   char *             int
      |                  %d
text@text-VMware-Virtual-Platform:~/OSLabs$ ./esh
```

https://gcc.gnu.org/onlinedocs/gcc/Warning-Options.html#index-Wformat= (ctrl + click)

```
void print_bg_info(int index, int pid, char *cmd) {
    printf("%d\t %d\t %s\n", index, pid, cmd);
    fflush(stdout);
}
```