EI338 OS Project Report2

陈子轩

2018 年 11 月 4 日

1 Project 1-UNIX Shell

1.1 Design Idea

for the demands in this project, like using & to determine concurrency, using !! to run history command, using <, > to redirect I/O, etc. I both put this detection in one function, called **shell_parser**.

```
int shell_parser(char * command, char ** args, char ** option);
```

this function take the **command** as input, which is the command string user just input. **args** is the argument list, just as the project demand. We put all our command observation in **option**, it is interpreted as follows.

```
option[0]: if & exist in the end, assign to 1, else 0
option[1]: if !! exist, assign to 1, else 0
option[2]: record the redirect > file name, 0 if not exist
option[3]: record the redirect < file name, 0 if not exist
option[4]: record pipe command name, 0 if not exist
```

for example , we input a line of command

```
cat testfile > out.txt &
```

then the option will be like

```
option[0]: 1
option[1]: 0
option[2]: out.txt
option[3]: 0
option[4]: 0
```

The advantage of this design idea is that it has strong expansibility. Suppose in the future we have to detect another kind of charactor or string, then we just add a new option and design detection techique in our **shell_parser** function.

2 Project 2—Linux Kernel Module for Task Information

In this project we learned how to write to a proc file– Setting the field .write in struct file operations to

```
.write = proc write
```

We also learned using **pid_task** function to get a pcb of a certain process.
In this project, we use a global variable **pid** to record the pid of input process while writing to proc/task_info file. and we get the pcb of the process in **proc_read** function. So whenever the system reads the file, we output the pcb of it.