Documentation for Project1
H9

Teacher: Guanghui Chang, Jackson Marques de Carvalho
Course Title: Operating system
Data: 2023.04.16

directory

1. System documentation

1.1 Data flow diagram

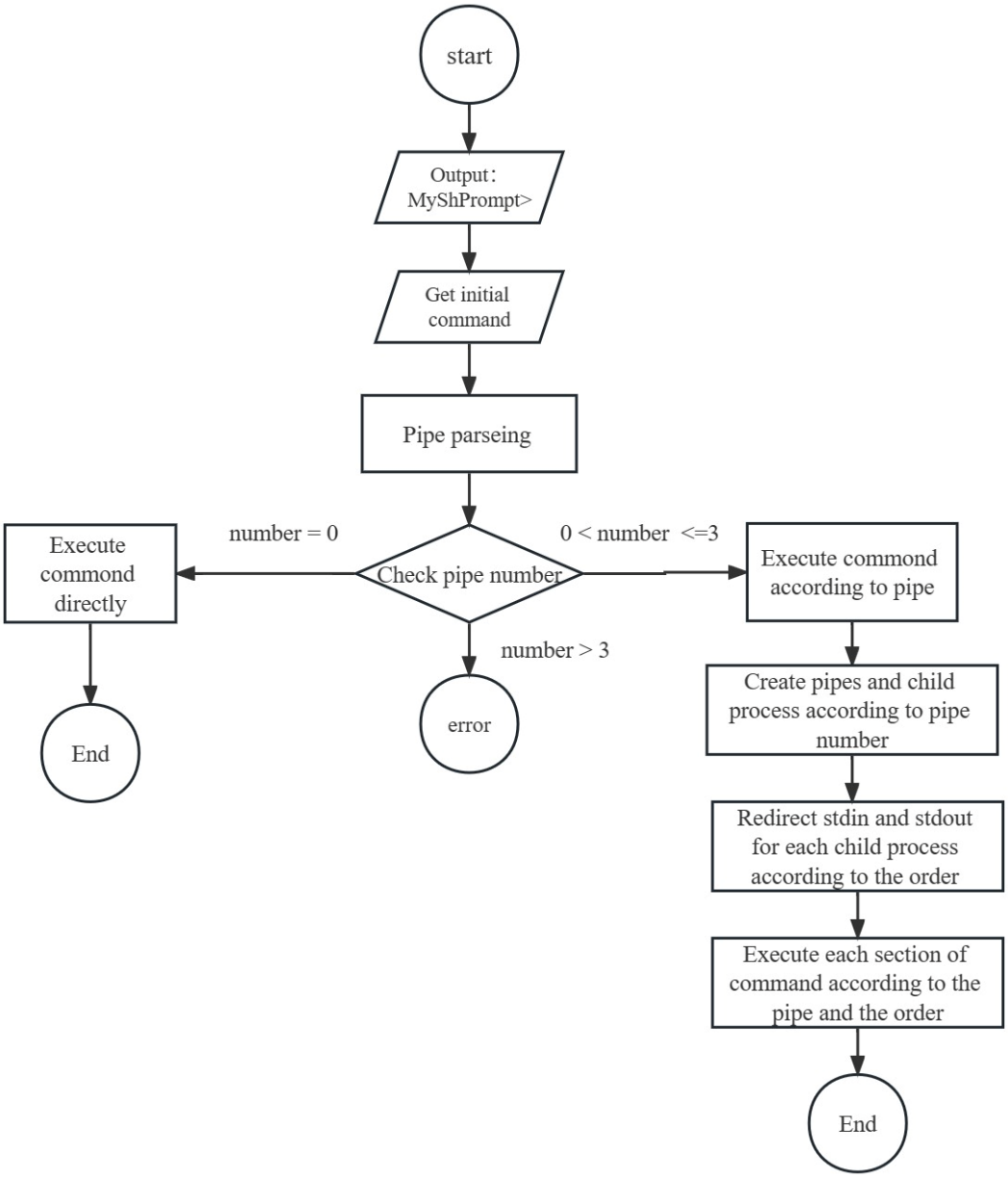

Figure 1 Data flow diagram

1.2 routines

1. Output the shell prompt
2. Read command from stdin
3. Pipe parsing to the initial command. It means to check if the command contains '|'. If it contains pipe descriptor, then splits commands by identifier and get the number of pipes. If it not contains pipe descriptor, then the number of pipes is 0.
4. Execute the command according to the number of pipes. The max number is 3. If the number is 0, just execute the command directly. If the number > 0 and <= 3, then we need to parse each section of command in pipe sides and then executes them according to the pipe and order.

5. If the command contains pipe, we need to create corresponding number of pipes, and one more child process. Each child process presents each section of command in pipe sides. In order to implement the pipe effect in the shell, we need to redirect stdin and stdout of each child process according the executing order and their relationship.

1.3 Implementation details.

1. Use fflush to output prompt to avoid change line. Add '\0' to the initial command.

```
printf("MyShPrompt>");
fflush(stdout);

fgets(input, 1024, stdin);// Gets the initial command
input[strlen(input)-1] = '\0';
```

Figure 2 Detail1

2. Remove ' or " of paraments in the command.

```
// removes \' or \" from the command
for(int i = 0; i < size; i++)
{
        if((argv[i][0] == '\'') || (argv[i][0] == '\"'))
        {
                int len = strlen(argv[i]) - 1;
                char temp[1024] = {0};
                for(int j = 0; j < len - 1; j++)
                {
                        temp[j] = argv[i][j+1];
                }
                temp[len] = '\0';
                argv[i] = temp;
        }
}
```

Figure 3 Detail2

3. Check if the command need to output to specific file

```
// checks if this section commands need to output to special file
if(size >= 3)
{
        if(strcmp(argv[size - 2], ">") == 0)
        {
                int fd = open(argv[size-1], O_WRONLY|O_CREAT, 0777);
                if(fd < 0)
                {
                        perror("Open file error:");
                        exit(-1);
                }

                dup2(fd, 1); // redircts the standerd output to the file

                argv[size-1] = NULL;
                argv[size-2] = NULL;
                size -= 2;
        }
}
```

Figure 4 Detail3

4. Close useless pipe descriptors to save source.

```
                    // closes all of the descirptiors of pipes to save the resource
                    for (int i = 0; i < pipeNum; i++)
                    {
                            close(pipes[i][0]);
                            close(pipes[i][1]);
                    }

                    exec(initPipes, i); // execute the command
                    perror("pipe execution wrong");
                    exit(-1);
            }

    }

    // Closes all of the pipes
    for (int i = 0; i < pipeNum; i++)
    {
            close(pipes[i][0]);
            close(pipes[i][1]);
    }
```

Figure 5 Detail 4

2. Test documentation

2.1 Test without pipe

Input: ls, ls -l, cat city.txt

Output:

```
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ ./mysh
MyShPrompt>ls
aa.txt  a.txt  city.txt  country.txt  mysh  myShell.c  test2  test2.c  test3  test3.c  test4  test4.c  test.c
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ ./mysh
MyShPrompt>ls -l
总用量 116
-rwxrwxr-x 1 h9 h9    24 4月  30 21:14 aa.txt
-rwxrwxr-x 1 h9 h9     0 4月  30 20:50 a.txt
-rw-rw-r-- 1 h9 h9    80 4月  26 18:38 city.txt
-rw-rw-r-- 1 h9 h9   112 4月  26 18:37 country.txt
-rwxrwxr-x 1 h9 h9 17584 4月  30 21:48 mysh
-rw-rw-r-- 1 h9 h9  4549 4月  30 21:48 myShell.c
-rwxrwxr-x 1 h9 h9 17136 4月  30 15:36 test2
-rw-rw-r-- 1 h9 h9   756 4月  30 15:36 test2.c
-rwxrwxr-x 1 h9 h9 17080 4月  30 18:41 test3
-rw-rw-r-- 1 h9 h9   713 4月  30 18:41 test3.c
-rwxrwxr-x 1 h9 h9 16752 4月  30 19:28 test4
-rw-rw-r-- 1 h9 h9   186 4月  30 19:28 test4.c
-rw-rw-r-- 1 h9 h9   440 4月  29 23:54 test.c
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ ./mysh
MyShPrompt>cat city.txt
miami
shanghai
albany
chongqing
tokyo
beijing
detroit
new york
hong kong
macau
```

Figure 6 Test without pipe

2.2 Test with pipe

Input: ls | wc, ls -l | wc, cat city.txt country.txt | wc, cat city.txt country.txt | sort

Output:

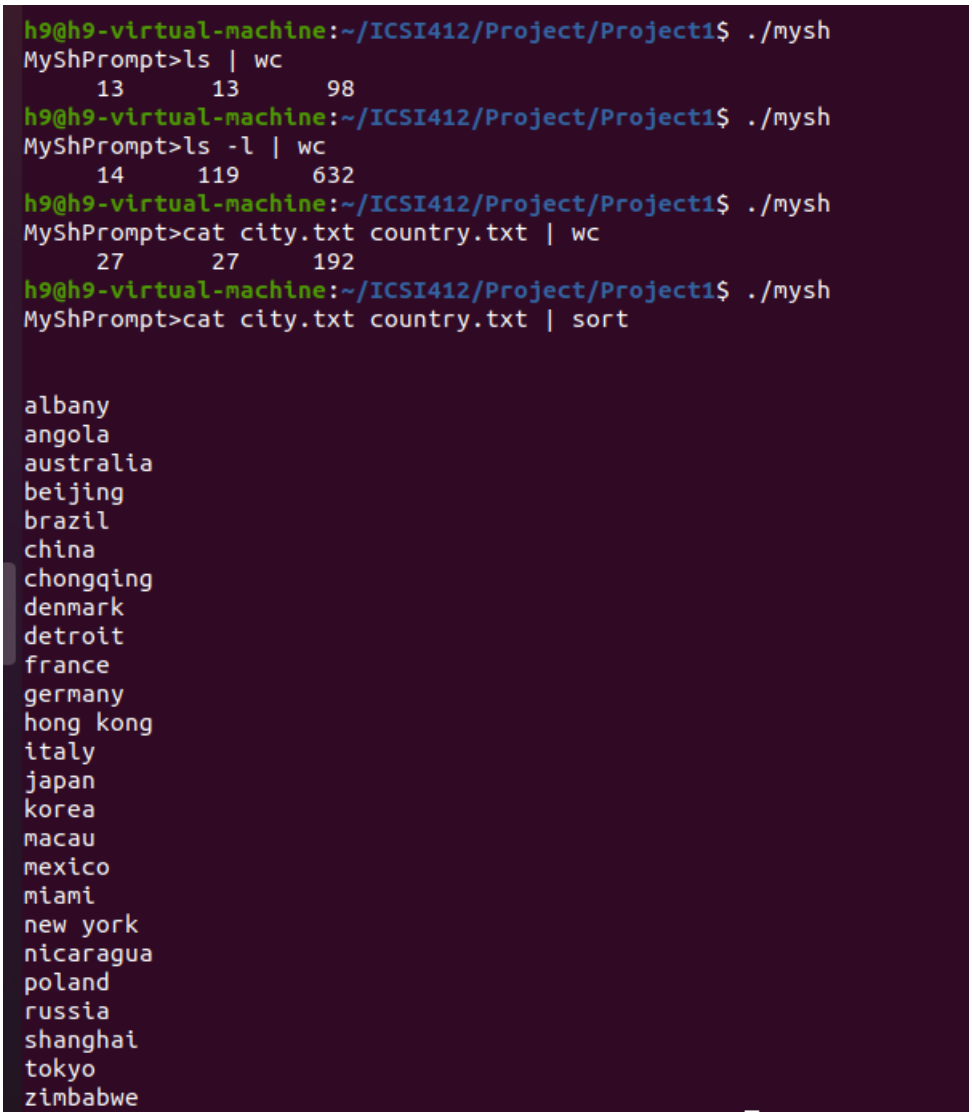

Figure 7 Test with pipe

2.3 Test the function output to specific file

Input: cat country.txt city.txt | sort | more > a.txt

Output:

```
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ ./mysh
MyShPrompt>cat country.txt city.txt | sort | more > a.txt
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ cat a.txt


albany
angola
australia
beijing
brazil
china
chongqing
denmark
detroit
france
germany
hong kong
italy
japan
korea
macau
mexico
miami
new york
nicaragua
poland
russia
shanghai
tokyo
zimbabwe
```

Figure 8 Test the function output to specific file

2.4 Test using to many pipes

Input: cat country.txt city.txt | egerp ‘g’ | sort | more | wc -l > b.txt

Output:

```
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ ./mysh
MyShPrompt>cat country.txt city.txt  | egrep ‘g’ | sort | more | wc -l > b.txt
To many pipes!
```

Figure 9 Test using to many pipes

2.5 Test the given data and example

1. Check data

```
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ cat country.txt city.txt
zimbabwe
russia
australia
brazil
china
denmark
germany
france
angola
italy
japan
korea
poland
mexico
nicaragua

miami
shanghai
albany
chongqing
tokyo
beijing
detroit
new york
hong kong
macau
```

Figure 10 Given data

2. Execute examples by shell

```
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ cat country.txt city.txt  | egrep 'g' | sort | more > countryCitygSorted.txt
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ cat country.txt city.txt  | egrep 'g' | sort | wc -l > countryCitygCount.txt
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ vim countryCitygSorted.txt
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ cat countryCitygSorted.txt
angola
beijing
chongqing
germany
hong kong
nicaragua
shanghai
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ cat countryCitygCount.txt
7
```

Figure 11 Execute examples by shell

3. Execute examples by mysh

```
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ ls
city.txt  country.txt  mysh  myShell.c  test2  test2.c  test3  test3.c  test4  test4.c  test.c
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ ./mysh
MyShPrompt>cat country.txt city.txt  | egrep 'g' | sort | more > countryCitySorted.txt
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ ./mysh
MyShPrompt>cat country.txt city.txt  | egrep 'g' | sort | wc -l > countryCityCount.txt
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ ls
city.txt  countryCityCount.txt  countryCitySorted.txt  country.txt  mysh  myShell.c  test2  test2.c  test3  test3.c  test4  test4.c  test.c
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ vim countryCitySorted.txt
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ cat countryCitySorted.txt
angola
beijing
chongqing
germany
hong kong
nicaragua
shanghai
h9@h9-virtual-machine:~/ICSI412/Project/Project1$ cat countryCityCount.txt
7
```

Figure 12 Execute examples by mysh

3. User documentation

3.1 How to run

1. Execute ./mysh in shell.
2. Input the command (the number of pipes should less than 3).

3.2 Paraments and methods

input: Initial command
pipeNum: The number of pipes
initPipes: Commands after pipe parsing
void parsePipe: Parses initial command according to pipe
void exec: Parses and executes command
void execPipes: Executes commands that parsed by pipe

Note:

This program is made to reproduce shell command line functions. The program focuses on the execution of shell commands, including commands with multiple pipes. The prompt is simply represented and can be changed to the path form if necessary.