

shell

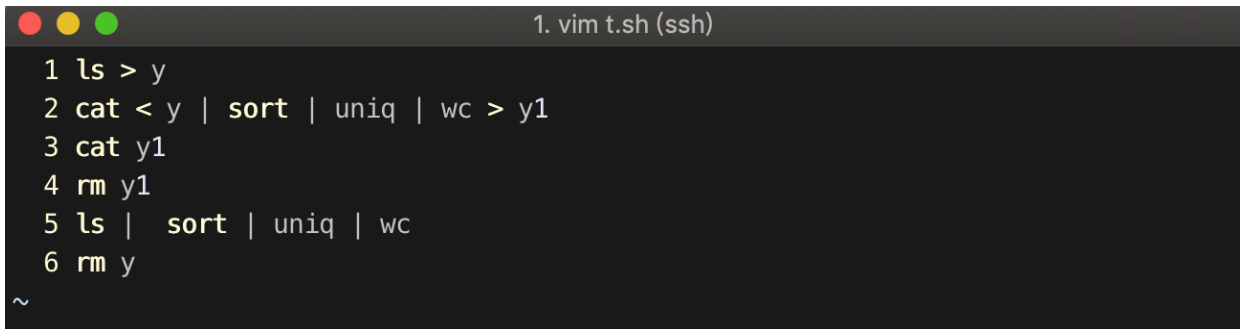
- 姓名：陈姝宇
- 学号：3017218119
- 班级：软工3班

实验内容

实现shell中的可执行命令，重定向命令，以及管道命令

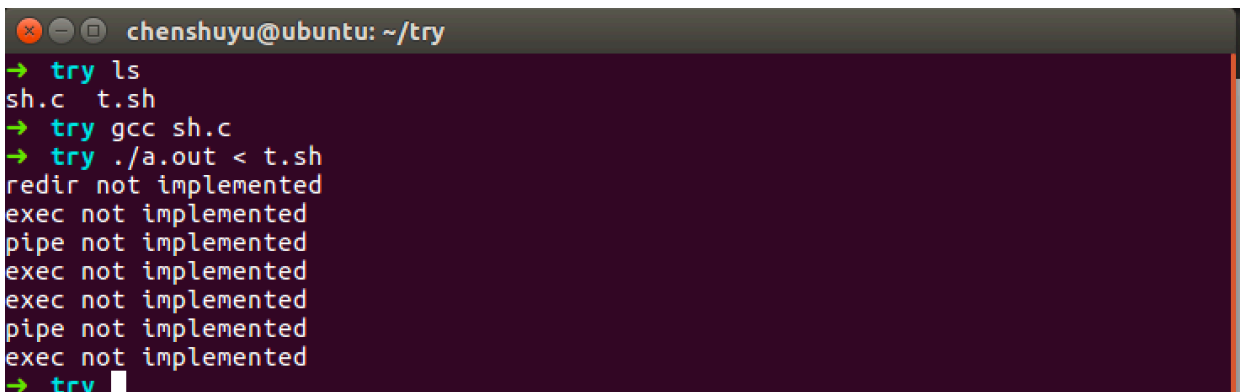
实验步骤

1. 将如下命令加入文件 t.sh 中



```
1. vim t.sh (ssh)
1 ls > y
2 cat < y | sort | uniq | wc > y1
3 cat y1
4 rm y1
5 ls | sort | uniq | wc
6 rm y
~
```

2. 单纯的将文件拷贝进去，并没有完成相应的函数



```
chenshuyu@ubuntu: ~/try
→ try ls
sh.c t.sh
→ try gcc sh.c
→ try ./a.out < t.sh
redir not implemented
exec not implemented
pipe not implemented
exec not implemented
exec not implemented
pipe not implemented
exec not implemented
→ try
```

3. 实验各种命令

- 实现可执行

```

66     if (ecmd->argv[0] == 0)
67     | _exit(0);
68     // fprintf(stderr, "exec not implemented\n");
69     // Your code here ...
70     if (access(ecmd->argv[0], F_OK) == 0)
71     {
72     |     execv(ecmd->argv[0], ecmd->argv);
73     }
74     else
75     {
76     |     const char *bin_path[] = {
77     |         "/bin/",
78     |         "/usr/bin/"};
79     |     char *abs_path;
80     |     int bin_count = sizeof(bin_path) / sizeof(bin_path[0]);
81     |     int found = 0;
82     |     for (int i = 0; i < bin_count && found == 0; i++)
83     |     {
84     |         int pathLen = strlen(bin_path[i]) + strlen(ecmd->argv[0]);
85     |         abs_path = (char *)malloc((pathLen + 1) * sizeof(char));
86     |         strcpy(abs_path, bin_path[i]);
87     |         strcat(abs_path, ecmd->argv[0]);
88     |         if (access(abs_path, F_OK) == 0)
89     |         {
90     |             execv(abs_path, ecmd->argv);
91     |             found = 1;
92     |         }
93     |         free(abs_path);
94     |     }
95     |     if (found == 0)
96     |     {
97     |         fprintf(stderr, "%s: Command not found\n", ecmd->argv[0]);
98     |     }
99     }
100    break;

```

- 实现重定向

```
103  ✓ case '<':  
104      rcmd = (struct redircmd *)cmd;  
105      // fprintf(stderr, "redir not implemented\n");  
106      // Your code here ...close(rcmd->fd);  
107      if (open(rcmd->file, rcmd->flags, 0644) < 0)  
108  ✓  {  
109          fprintf(stderr, "Unable to open file: %s\n", rcmd->file);  
110          exit(0);  
111      }  
112      runcmd(rcmd->cmd);  
113      break;
```

- 实现管道

```

115     case '|':
116         pcmd = (struct pipecmd *)cmd;
117         // fprintf(stderr, "pipe not implemented\n");
118         // Your code here ...
119         if (pipe(p) < 0)
120             fprintf(stderr, "pipe failed\n");
121         if (fork1() == 0)
122         {
123             close(1);
124             dup(p[1]);
125             close(p[0]);
126             close(p[1]);
127             runcmd(pcmd->left);
128         }
129         if (fork1() == 0)
130         {
131             close(0);
132             dup(p[0]);
133             close(p[0]);
134             close(p[1]);
135             runcmd(pcmd->right);
136         }
137         close(p[0]);
138         close(p[1]);
139         wait(&r);
140         wait(&r);
141
142         break;
143     }

```

实验结果

```

→ try gcc sh.c
→ try ./a.out < t.sh
a.out sh.c t.sh y
      0      0      0
      4      4     18

```