

## Easy Tasks:

### 1. Constant loop

```
yajie@yajie-VirtualBox:~/Desktop$ ./ve482sh
ve482sh $ ls
back_up.c  p1      pipe_sh  pipesh.c  redirect  src_orig
out.txt    pipe    pipe_sh.c README.txt redirect.c ve482sh
ve482sh $ mkdir dir
ve482sh $ ls
back_up.c  out.txt  pipe      pipe_sh.c README.txt redirect.c ve482sh
dir        p1      pipe_sh   pipesh.c  redirect  src_orig
ve482sh $
```

### 2. Simple command

```
yajie@yajie-VirtualBox:~/Desktop$ ./ve482sh
ve482sh $ ls
back_up.c  p1      pipe_sh  pipesh.c  redirect  src_orig
out.txt    pipe    pipe_sh.c README.txt redirect.c ve482sh
ve482sh $ mkdir dir
ve482sh $ ls
back_up.c  out.txt  pipe      pipe_sh.c README.txt redirect.c ve482sh
dir        p1      pipe_sh   pipesh.c  redirect  src_orig
ve482sh $
```

### 3. More complex command

```
ve482sh $ sudo apt-get update
[sudo] password for yajie:
Hit:1 http://cn.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://cn.archive.ubuntu.com/ubuntu xenial-updates InRelease [95.7 kB]
Get:3 http://cn.archive.ubuntu.com/ubuntu xenial-backports InRelease [92.2 kB]
Get:4 http://cn.archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [394 kB]
Hit:5 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:6 http://ppa.launchpad.net/webupd8team/sublime-text-2/ubuntu xenial InRelease
Get:7 http://cn.archive.ubuntu.com/ubuntu xenial-updates/main i386 Packages [389 kB]
Hit:8 http://ppa.launchpad.net/webupd8team/sublime-text-3/ubuntu xenial InRelease
Get:9 http://cn.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [332 kB]
Get:10 http://cn.archive.ubuntu.com/ubuntu xenial-updates/universe i386 Packages [328 kB]
Fetched 1,632 kB in 2s (754 kB/s)
Reading package lists... Done
ve482sh $
```

### 4. Pipe

```
ve482sh $ ls | grep p
back_up.c
p1
pipe
pipe_sh
pipe_sh.c
pipesh.c
ve482sh $
```

### 5. File I/O redirection >

```
ve482sh $ echo 123 > out.txt
ve482sh $ cat out.txt
123
ve482sh $
```



## 9. More complex commands with pipes

```
ve482sh $ ls | xargs grep .c
back_up.c:#include <stdio.h>
back_up.c:#include <stdlib.h>
back_up.c:#include <unistd.h>
back_up.c:#include <fcntl.h>
back_up.c:#include <sys/types.h>
back_up.c:#include <sys/wait.h>
back_up.c:#include <sys/stat.h>
back_up.c:#include <string.h>
back_up.c:#include <signal.h>
back_up.c:    if (sig == SIGINT) printf("Please press enter to continue:\n");
back_up.c:int quote_count(char *line, int *position) {
back_up.c:    int count = 0;
back_up.c:        position[count] = i;
back_up.c:        count = count + 1;
back_up.c:    if ((count % 2) == 1) {
back_up.c:        count = quote_count(line, position);
back_up.c:        return count;
back_up.c:void parse(char *line, char **arg) {
back_up.c:    char *temp = line;
back_up.c:    position = (int *)malloc(MAX_LINE_LEN*sizeof(int));
back_up.c:    int space_no = 1;
back_up.c:    quote_no = quote_count(line, position);
back_up.c:    char *space = strchr(line, ' ');
back_up.c:    while (space != NULL) {
back_up.c:        space[0] = '\0';
back_up.c:        arg[i] = (char *)malloc(MAX_LINE_LEN*sizeof(char));
back_up.c:        if (space > (temp + position[j]) && space < (tem
p + position[j+1])) break;
back_up.c:        while (space[space_no] == ' ') {
back_up.c:            if ((space + space_no) >= (temp
+ position[k])) break;
back_up.c:            space_no = space_no + 1;

redirect.c:    close(out);
redirect.c:    // execute grep
redirect.c:    execvp("echo", grep_args);
redirect.c:    cmd = line;
redirect.c:    arg[i] = (char **)malloc(MAX_LINE_LEN*sizeof(char *));
redirect.c:    parse(cmd, arg[i]);
redirect.c:    p = strchr(line, '|');
redirect.c:    /*else if ((p = strchr(line, '>')) != NULL) {
redirect.c:        *option = 'c';
redirect.c:        cmd = line;
redirect.c:        arg[i] = (char **)malloc(MAX_LINE_LEN*sizeof(char *));
redirect.c:        parse(cmd, arg[i]);
redirect.c:        cmd = line;
redirect.c:        arg[i] = (char **)malloc(MAX_LINE_LEN*sizeof(char *));
redirect.c:        parse(cmd, arg[i]);
redirect.c:        cmd = line;
redirect.c:        arg[i] = (char **)malloc(MAX_LINE_LEN*sizeof(char *));
redirect.c:        parse(cmd, arg[i]);
redirect.c:        cmd = line;
redirect.c:        arg[i] = (char **)malloc(MAX_LINE_LEN*sizeof(char *));
redirect.c:        parse(cmd, arg[i]);
redirect.c:    else if ((p = strchr(line, '<')) != NULL) {
redirect.c:        cmd = line;
redirect.c:        arg[i] = (char **)malloc(MAX_LINE_LEN*sizeof(char *));
redirect.c:        parse(cmd, arg[i]);
redirect.c:        cmd = line;
redirect.c:        arg[i] = (char **)malloc(MAX_LINE_LEN*sizeof(char *));
redirect.c:        parse(cmd, arg[i]);
redirect.c:        cmd = line;
redirect.c:        arg[i] = (char **)malloc(MAX_LINE_LEN*sizeof(char *));
redirect.c:        parse(cmd, arg[i]);
grep: src_orig: Is a directory
Binary file ve482sh matches
ve482sh $
```

10. Clean exit

```
ve482sh $ exit  
yajie@yajie-VirtualBox:~/Desktop$
```

Medium Tasks:

1. Display working directory

```
yajie@yajie-VirtualBox:~/Desktop$ ./ve482sh  
ve482sh $ pwd  
/home/yajie/Desktop  
ve482sh $
```

2. Wait for command to be completed

```
ve482sh $ echo abc >  
>  
>  
> out.txt  
ve482sh $ ls |  
>  
> grep 4  
ve482sh  
ve482sh $ echo "Hello  
> kk  
> world"  
Hello  
kk  
xt world  
ve482sh $
```

3. Handle quotes

```
ve482sh $ echo "sj k hi kw " | grep hi  
sj k hi kw
```

4. Catch errors

```
ve482sh $ cat <  
Error near <!  
ve482sh $ cd ppp  
Error occured!
```

Hard Tasks:

1. Change working directory

```
ve482sh $ pwd
/home/yajie/Desktop
ve482sh $ ls
back_up.c  out1.txt  p1      pipe_sh  pipesh.c  redirect  src_orig
dir        out.txt   pipe    pipe_sh.c README.txt redirect.c ve482sh
ve482sh $ cd dir
ve482sh $ ls
ve482sh $ pwd
ve482sh $ cd /home
ve482sh $ pwd
ve482sh $ cd Desktop
Error occured!
ve482sh $ pwd
/home
ve482sh $
```

2. Ctrl+c is handled

```
ve482sh $ ^CPlease press enter to continue:
ve482sh $ ls
ve482sh $ ^CPlease press enter to continue:
ve482sh $ echo 123
```