

HƯỚNG DẪN SỬ DỤNG

I. Giới thiệu chung:

1. Chương trình được viết trên ngôn ngữ C99.
2. Thực hiện biên dịch mã nguồn thông qua trình biên dịch gcc, ta được file execute của chương trình Shell.
3. Mở file execute của chương trình shell. Màn hình hiển thị "osh>" chỉ rằng người dùng cần nhập câu lệnh

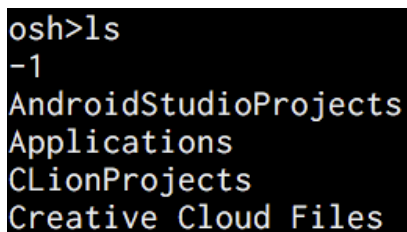


```
shiro ~  
osh> /Users/shiro/CLionProjects/projectcmd/cmake-build-release/projectcmd; exit
```

II. Các loại câu lệnh trong chương trình

1. **Câu lệnh cơ bản** : là câu lệnh chỉ chứa lệnh hoặc chứa lệnh và các tham số cơ bản, Shell phải đợi câu lệnh hoàn thành mới có thể tiếp tục hoạt động

Ví dụ : ls, touch a.txt, cat prog.c,...



```
osh>ls  
-1  
AndroidStudioProjects  
Applications  
CLionProjects  
Creative Cloud Files
```

2. **Câu lệnh có chứa &** : là loại câu lệnh mà việc thực hiện công việc được thực hiện bất đồng bộ với Shell. Shell có thể tiếp tục nhận câu lệnh mới khi lệnh vẫn đang được thực hiện.

```
osh>ls &  
osh>-1  
AndroidStudioProjects  
Applications  
CLionProjects  
Creative Cloud Files
```

(Shell đã yêu cầu người dùng nhập lệnh mới qua osh> thứ 2 trước khi kết quả của lệnh ls được in ra màn hình)

Lệnh : !!

Chức năng là chạy lệnh gần nhất vừa được chạy bởi người sử dụng.

3. Câu lệnh được chuyển hướng input và output :

Lệnh : "<args> [> hoặc <] <tên_file>". Trong đó > là output được đưa ra file còn < là input được nhập từ file.

Chức năng là xuất kết quả của lệnh vào file hoặc nhập dữ liệu từ file vào để thực hiện lệnh.

Ví dụ : "ls > output.txt", "sort -nr < input.txt"

```
osh>cat io.txt  
5  
2  
8  
4osh>sort -n < io.txt  
2  
4  
5  
8
```

(Ta thấy file io.txt chứa 4 số gồm 5 2 4 8 , sau khi sử dụng lệnh sort -n ta được kết quả 2 4 5 8 là sắp xếp tăng dần theo kiểu number)

Lưu ý : lệnh sort chỉ hoạt động khi các số cách nhau bởi dấu xuống dòng.

4. Câu lệnh được kết nối với nhau thông qua đường ống.

Lệnh : "<args1> | <args2>"

Ví dụ : "ls -l | less"

Chức năng : lấy kết quả của lệnh args1 để đưa vào làm dữ liệu cho lệnh args2 hoạt động.

Demo lệnh : "ls -l | head"

```
osh>ls -l | head
total 144
-rw-r--r--  1 shiro  staff   55 Oct  3 20:10 -1
drwxr-xr-x 10 shiro  staff  320 Oct  6 22:10 AndroidStudioProjects
drwx-----@ 3 shiro  staff   96 Aug 26 14:47 Applications
drwxr-xr-x  4 shiro  staff  128 Sep 25 14:26 CLionProjects
drwxrwxr-x@  6 shiro  staff  192 Sep 24 00:13 Creative Cloud Files
drwxr-xr-x  9 shiro  staff  288 Sep 28 21:00 Dealers_Bootstrap4
drwx-----+ 14 shiro  staff  448 Oct  7 21:03 Desktop
drwx-----+ 21 shiro  staff  672 Oct  8 12:15 Documents
drwx-----+  6 shiro  staff  192 Oct  5 01:01 Downloads
osh>
```

Đầu tiên lệnh ls -l sẽ chạy, sau đó lệnh head sẽ lấy kết quả của lệnh ls -l để chạy và show 10 dòng đầu kết quả của lệnh ls -l ra màn hình.

Lưu ý : Lệnh `head` dùng để xem những dòng đầu của tệp tin (theo mặc định là 10 dòng đầu tiên). Chúng ta có thể thay đổi số dòng bằng cách thêm -n vào sau lệnh `head` .



III. Một số lưu ý đặc biệt.

1) Các lệnh chuyển hướng chỉ được chứa duy nhất một "<" hoặc một ">" theo yêu cầu của đề án.

You can assume that commands will contain either one input or one output redirection and will not contain both. In other words, you do not have to be concerned with command sequences such as `sort < in.txt > out.txt`.

2) Các lệnh đường ống chỉ chứa duy nhất một "|" và không chứa các lệnh chuyển hướng.

using the `dup2()` function as described in the previous section. Finally, although several commands can be chained together using multiple pipes, you can assume that commands will contain only one pipe character and will not be combined with any redirection operators.