Test Cases

1. IGCreate:
2. Normal case:
   1. IGCreate(0)

(Lookup)

Expect:

Result:

1. Failure case:
   1. IGCreate(0) //again with the same group id

Expect: error message.

Result: “Failed to create a group.”

* 1. Use a loop to create more than 21 groups //we only support at most 20 interest groups

Expect: error message.

Result: "Failed to create a group at creating group 21.

1. IGPublisher:
2. Normal case:
   1. $ cd /\*change directory to /root \*/

$ ls > hi

$ cat hi

Expect: .ashrc .exrc .lesshst .profile hi

Result: .ashrc .exrc .lesshst .profile hi

* 1. $ ls –l > hi

$ cat hi

Expect: the detailed information about .ashrc .exrc .lesshst .profile hi

Result:

* 1. $ ls –l hi > hi

$ cat hi

Expect: detailed information about hi

Result:

* 1. $ echo hello > hi

$ cat hi

Expect: hello

Result: hello

1. IGSubscriber:
2. Normal case:
   1. $ cd /\*change directory to /root \*/

$ ls | wc

Expect: three numbers showing the # of words, lines, and characters of cmd1 output

Result: 5 5 34

* 1. $ ls –l | wc /\* cmd1 has a parameter \*/

Expect: three numbers showing the # of words, lines, and characters of cmd1 output

Result: 6 47 278

* 1. $ ls | wc -c /\* cmd2 has a parameter \*/

Expect: a number showing the # of characters of cmd1 output

Result: 34

* 1. $ ls –l | wc -l /\* both commands has a parameter \*/

Expect: a number showing the # of lines of cmd1 output

Result: 6

* 1. $ ls | wc | wc -w

Expect: a number showing the # of words of cmd2 output

Result: 3

1. Special case:
   1. $ | abc

Expect: an error message

Result: cannot find abc

* 1. $ |

Expect: an error message

Result: nothing happens, the shell simply ignored it

* 1. $ | cmd /\* Here cmd is a valid command \*/

Expect: execute cmd

Result: cmd works fine

1. IGPublish:
2. Normal case:
   * + - 1. $Alias

Expect: The format should be: Alias othername=”cmd”

Result: The format should be: Alias othername=”cmd”

b) $Alias llss=”ls”

$ls

Expect: Prints out the all the files’ filename in the current path

Result: Prints out the all the files’ filename in the current path

1. 2) Special case:

a) $Alias ls=”llss”

$ls

Expect: Ignore the first command then prints out the all the files’ filename in the current path

Result: Ignore the first command then prints out the all the files’ filename in the current path

b) $Alias ls=”cd”  
 $ls

Expect: Ignore the first command then prints out the all the files’ filename in the current path

Result: Ignore the first command then prints out the all the files’ filename in the current path

c) $Alias ls=””

$ls

Expect: Ignore the first command then prints out the all the files’ filename in the current path

Result: Ignore the first command then prints out the all the files’ filename in the current path

d) $Alias =”ls”

$ls

Expect: Ignore the first command then prints out the all the files’ filename in the current path

Result: Ignore the first command then prints out the all the files’ filename in the current path

e) $Alias ls=””

$ls

Expect: Ignore the first command then prints out the all the files’ filename in the current path

Result: Ignore the first command then prints out the all the files’ filename in the current path

f) $Alias “llss”=ls

$ls

Expect: Ignore the first command then prints out the all the files’ filename in the current path

Result: Ignore the first command then prints out the all the files’ filename in the current path

1. IGRetrive:
2. Normal case:
   * + - 1. When first command is executable.

$ if echo “cmd1”; then echo “cmd2”; else echo “cmd3”; fi

Expect: “cmd1”

“cmd2”

Result: “cmd1”

“cmd2”

* + - * 1. When first command is not executable.

$ if example; then echo “cmd2”; else echo “cmd3”; fi

Expect: Cannot find example

“cmd3”

Result: Cannot find example

“cmd3”

2) Special case:

a) Without entering “then”

$ if “cmd1”; echo “cmd2”; else echo “cmd3”;fi

Expect: Ignore the missing “then” and run the commands

Result: “cmd1”

“cmd2”

b) Without entering “else”

$ if “cmd1”; then echo “cmd2”; echo “cmd3”;fi

Expect: Ignore the missing “else” and run the commands

“cmd1”

“cmd2”

Result: “cmd1”

“cmd2”