

Philip Yun 65035200

Nsh

Features implemented:

- Background execution
- Comments
- Meta characters without spaces
- Exit command
- Single quotes
- Infinite piping
- Infinite length command lines with no memory leaks
- Ctrl-c translated as blank line to parent process

Manual Page:

```
Usage: nsh [nsh_scripts...]
```

```
nsh      ---- interactive shell mode
nsh files ---- run files as nsh scripts
```

### Interactive mode

Interactive is prompted with a “? “ prompt. To run a command and its arguments, simply type them at the command prompt and hit enter.

The interactive mode shell fully supports the ability to use file redirection and piping. Additionally, you may also run tasks in the background or ignore comments in the shell.

Example usage:

Pipes:

```
? ls -l | grep rwx
-rwxrwxr-x 1 philip philip 238 Jun 2 03:06 background.nsh*
-rwxrwxr-x 1 philip philip 191 Jun 2 03:06 changeDir.nsh*
-rwxrwxr-x 1 philip philip 82 Jun 2 03:07 exit.nsh*
-rwxrwxr-x 1 philip philip 35912 Jun 2 03:07 fd.nsh*
-rwxrwxr-x 1 philip philip 424 Jun 2 03:08 file_input.nsh*
-rwxrwxr-x 1 philip philip 311 Jun 2 03:08 file_output.nsh*
-rwxrwxr-x 1 philip philip 871 Jun 2 03:08 noSpaceParse.nsh*
-rwxrwxr-x 1 philip philip 258 Jun 2 03:08 pipes.nsh*
drwxrwxr-x 2 philip philip 4096 Jun 2 02:38 test_scripts/
? cat file.c | grep int | wc -w
5
```

#### File I/O

```
? # output
? ls -l > out.txt
? ls -l out.txt
-rwxrwxr-x 1 philip philip 238 Jun  2 03:06 out.txt

? #concatenation
? ls -l >> out.txt
? ls -l out.txt
-rwxrwxr-x 1 philip philip 576 Jun  2 03:06 out.txt

? #input
? echo hello > sample.txt
? sed 's/hello/goodbye/g' < sample.txt
```

#### Comments:

Comments can be established with a #. Anything following a # will be ignored

```
? echo hello world #ignored
hello world
```

#### Background execution:

Tasks can be sent to the background to do other work while a task is executing

```
? ping -c 2 8.8.8.8&
? PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=5.51 ms
ls
background.nsh      file_input.nsh      nsh      parse.h
changeDir.nsh file_output.nsh  nsh.c    parse.o
exit.nsh      Makefile      nsh.o    pipes.nsh
fd.nsh      noSpaceParse.nsh  parse.c  test_scripts
? 64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=5.64 ms

--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 5.517/5.580/5.643/0.063 ms
```

To initiate a process to the background, end the command line with an &

*Including an & to any character that is not the last character has undefined behavior.*

## Single Quotes

Single quotes can be used to include spaces in arguments

```
? touch 'test file'
? ls
test file
```

## Keyboard signals

Ctrl-C when at the main shell acts as a blank line.

```
? ^C
?
```

Ctrl-D will send the EOF signal to the shell and end the shell

```
? ^D
exit
```

Issuing a Ctrl-D signal at a non empty line is undefined behavior.

You may also terminate your shell with the “exit” command

```
? exit
exit
```

## Cd

You may change the working directory of the nsh shell by issuing the cd command

```
? pwd
/home/user/Docs
? cd
? pwd
/home/user
? cd /bin
? pwd
/bin
```

Usage for the cd command is

```
cd [directory]
```

## Test Scripts:

### Some caveats:

Many test scripts use comments and single quoting, proving that the comments and single quotes are being handled correctly by the parser.

#### fd.nsh

fd.nsh runs  
ls -l | sed 's/philip/bob/g' > out.txt  
916 times.

The reason for this is to test the file descriptor capacity and see if the FD's open sys call ever fails. No error ever gets printed to stderr, so FD's are handled correctly

#### file\_input.nsh

This script creates a file with many lines saying either  
Hello ... OR  
Goodbye ...

That file is then redirected into

grep hello

Full cmd: grep hello < input.txt

The result is lines of text that say hello. None of the lines say goodbye, so the file input works as expected.

#### File\_output.nsh

This sends the output of ls -l to test\_output.txt. It is then concatenated with ls -l again.

We ls -l the output file each iteration to see that the file doubles in size. Output and concatenation works as expected

#### Pipes.nsh

pipes.nsh attempts to run three command lines. Each command line increases the number of pipes by 1. Each one works correctly thus showing that the amount of pipes can be extended to more than 1

noSpaceParse.nsh

This script tries to use nsh symbols (File i/o and pipes) with varying space formats

Cmd<file

Cmd <file

Cmd< file

These patterns all show the same results indicating the parser works with an arbitrary amount of spaces

exit.nsh

Exit should do nothing, as it just exits the nsh shell immediately

changeDir.nsh

This script will cd into other directories and then immediately print the directory it resides in

Background.nsh

This script will run a ping request 4 times in the background, then send an echo to the screen. If the echo comes before the ping terminates, the background functionality is working