
Table of Contents

- Cover Page
- Makefile documentation
- Features of my shell and Screenshot of my shell in action
- Test Scripts and how to use them
- Screenshots of the output produced by my test scripts
- Discovered bug

CS 146 Project Documentation (nsh shell)

Name: Sam Thapa

Student Id: 10677794

My nsh shell can execute commands that are required for a minimal shell and extra feature I implemented for extra credit is **Supports multiple pipes**

I am using the parser provided by professor Hayes.

```
1  nsh:  main-parse.o parse.o
2      gcc -o nsh main-parse.o parse.o
3
4  main-parse.o:  main-parse.c parse.h
5      gcc -c -o main-parse.o main-parse.c
6
7  parse.o:  parse.c parse.h
8      gcc -c -o parse.o parse.c
9
10 clean:
11     rm *.o nsh
12
```

The source files for this project are

- `main-parse.c parse.c parse.h`

INSTRUCTION TO USE:

In order to successfully compile and produce an executable file. Issue the following command

>> `make`

It will produce the executable file called `nsh`. To run my shell,

>> `./nsh`

After you are done testing my shell, please issue the command

>> `make clean`

`make clean` removes all the object files and a executable file(nsh)

Here, I am going to explain all features of my shell and give a screenshot of what my shell outputs

1. Basic commands

```
shambhut@odin 10:28:28 ~/CompSci146/nsh
[$ make
make: `nsh' is up to date.
shambhut@odin 10:28:30 ~/CompSci146/nsh
[$ ./nsh
[? echo hello world
hello world
[? ls
foo          Makefile  output.txt  randomfile  test_grep.txt
lsout        never    parse.c    tempFolder  test_multiplePipe
main-parse.c never1   parse.h    test_append  test_outfileredirect
main-parse.o nsh     parse.o    test_cd     test.sh
[? date
Fri Jun  8 10:29:01 PDT 2018
[? cat test.sh
ls
cd ..
pwd
```

In this screenshot, it shows my shell executing basic commands like :

- `echo hello world`
- `ls`
- `date`
- `cat filename`

2. Basic input/output Redirection including append

I) `cat < inputfile`

Opens the input file , and outputs its content on terminal screen.

```
[?] ls
foo          main-parse.o  never1      parse.c     randomfile  test_cd      test_outfileredirect
lsout       Makefile     nsh         parse.h     tempFolder  test_grep.txt test.sh
main-parse.c never        output.txt  parse.o     test_append test_multiplePipe
[?] cat < Makefile
nsh:      main-parse.o parse.o
gcc -o nsh main-parse.o parse.o

main-parse.o: main-parse.c parse.h
gcc -c -o main-parse.o main-parse.c

parse.o:  parse.c parse.h
gcc -c -o parse.o parse.c

clean:
rm *.o nsh
? █
```

II) `date > outfile` , `echo "hello world" >> outfile`

It saves the output of date command to outfile and then again "hello world" is appended to the outfile.

```
shambhut@odin 10:57:12 ~/CompSci146/nsh
[$ make
gcc -c -o parse.o parse.c
gcc -o nsh main-parse.o parse.o
shambhut@odin 10:57:14 ~/CompSci146/nsh
[$ ./nsh
[?] date > datefile
[?] cat datefile
Fri Jun  8 10:57:31 PDT 2018
[?] echo "hello world" >> datefile
[?] cat datefile
Fri Jun  8 10:57:31 PDT 2018
"hello world"
? █
```

III) who | fgrep -i .edu > foo

```
$ ./nsh
? who | fgrep -i .edu > foo
? cat foo
vipada pts/1 2018-06-04 08:45 (goddess.ics.uci.edu)
pashak pts/3 2018-06-07 12:52 (dhcp-v014-176.mobile.uci.edu)
vuqt1 pts/8 2018-05-30 09:37 (dhcp-007035.ics.uci.edu)
bsmorton pts/22 2018-05-26 17:33 (vcv072082.vpn.uci.edu)
bsleon pts/24 2018-06-07 15:42 (vcv076223.vpn.uci.edu)
kuanl9 pts/25 2018-06-07 14:02 (vcv071070.vpn.uci.edu)
vuqt1 pts/28 2018-05-29 10:59 (dhcp-007035.ics.uci.edu)
wayne pts/30 2018-06-03 08:53 (bayonet-01.ics.uci.edu)
rajj pts/35 2018-06-07 17:27 (vcv075210.vpn.uci.edu)
jiangag pts/48 2018-06-07 16:00 (vcv071021.vpn.uci.edu)
nier pts/51 2018-06-07 16:50 (vcv064101.vpn.uci.edu)
neilps pts/54 2018-06-07 16:45 (vcv074226.vpn.uci.edu)
abagasao pts/55 2018-06-07 15:15 (vcv064126.vpn.uci.edu)
wenboz10 pts/57 2018-06-07 15:25 (vcv076196.vpn.uci.edu)
achio pts/58 2018-06-07 17:46 (dhcp-v062-254.mobile.uci.edu)
lamk10 pts/59 2018-06-07 17:34 (vcv075208.vpn.uci.edu)
shambhut pts/60 2018-06-07 16:46 (dhcp-v112-055.mobile.uci.edu)
derektv pts/64 2018-06-07 10:55 (red-wolf.ics.uci.edu)
hgpham pts/66 2018-06-07 16:47 (dhcp-v107-253.mobile.uci.edu)
wayne pts/67 2018-06-02 16:20 (bayonet-06.ics.uci.edu)
jiangag pts/68 2018-06-07 16:00 (vcv071021.vpn.uci.edu)
lamk10 pts/70 2018-06-07 17:34 (vcv075208.vpn.uci.edu)
wayne pts/79 2018-05-30 13:42 (bayonet-06.ics.uci.edu)
```

iv) sort < foo | uniq -c >> bar

```
? echo "original content " > bar
? cat bar
"original content "
? sort < foo | uniq -c >> bar
? cat bar
"original content "
  1 abagasao pts/35 2018-06-08 10:41 (vcv076150.vpn.uci.edu)
  1 abagasao pts/55 2018-06-07 15:15 (vcv064126.vpn.uci.edu)
  1 achio pts/111 2018-06-08 11:18 (dhcp-v035-251.mobile.uci.edu)
  1 achio pts/113 2018-06-08 11:18 (dhcp-v035-251.mobile.uci.edu)
  1 achio pts/123 2018-06-07 17:46 (dhcp-v062-254.mobile.uci.edu)
  1 achio pts/145 2018-06-07 17:46 (dhcp-v062-254.mobile.uci.edu)
  1 achio pts/58 2018-06-07 17:46 (dhcp-v062-254.mobile.uci.edu)
  1 ascheere pts/108 2018-06-07 16:25 (vcv064186.vpn.uci.edu)
  1 asifm pts/110 2018-06-07 12:31 (vcv076133.vpn.uci.edu)
  1 asyeo pts/87 2018-06-07 15:51 (vcv076136.vpn.uci.edu)
  1 bavilla pts/54 2018-06-08 10:57 (vcv076223.vpn.uci.edu)
  1 bblanton pts/98 2018-06-07 16:22 (vcv064200.vpn.uci.edu)
  1 bmbakogu pts/57 2018-06-08 08:42 (vcv076022.vpn.uci.edu)
  1 brocha1 pts/98 2018-06-08 10:21 (vcv075006.vpn.uci.edu)
  1 bsleon pts/24 2018-06-07 15:42 (vcv076223.vpn.uci.edu)
  2 bsmorton pts/22 2018-05-26 17:33 (vcv072082.vpn.uci.edu)
  1 derektv pts/147 2018-06-07 17:08 (finback-whale.ics.uci.edu)
  1 derektv pts/64 2018-06-07 10:55 (red-wolf.ics.uci.edu)
  1 derektv pts/95 2018-06-07 15:38 (finback-whale.ics.uci.edu)
  1 eweik pts/70 2018-06-08 09:56 (dhcp-v001-115.mobile.uci.edu)
  1 eweik pts/7 2018-06-08 09:26 (dhcp-v001-115.mobile.uci.edu)
  2 hans pts/112 2018-06-04 15:56 (richard-castle-v2.ics.uci.edu)
  2 hans pts/125 2018-06-04 15:49 (medusa.ics.uci.edu)
  2 hans pts/96 2018-06-01 15:37 (medusa.ics.uci.edu)
  1 hgpham pts/66 2018-06-07 16:47 (dhcp-v107-253.mobile.uci.edu)
  1 jiangag pts/0 2018-06-08 10:42 (vcv075153.vpn.uci.edu)
  1 jiangag pts/113 2018-06-07 16:29 (vcv071021.vpn.uci.edu)
  1 jiangag pts/24 2018-06-08 08:08 (vcv075153.vpn.uci.edu)
  1 jiangag pts/48 2018-06-07 16:00 (vcv071021.vpn.uci.edu)
  1 jiangag pts/68 2018-06-07 16:00 (vcv071021.vpn.uci.edu)
```

“These two commands above show that my shell handles the combination of pipe, input redirection, output redirection along with the append mode”

3) Multiple pipe

The following screenshot shows that my shell can handle multiple pipes.

```
shambhut@odin 12:47:51 ~/CompSci146/nsh
[$ ./nsh
[? ls -l | grep parse | sort -k 5nr
-rw----- 1 shambhut ugrad 6661 Jun  8 10:57 parse.c
-rw----- 1 shambhut ugrad 5176 Jun  8 10:57 parse.o
-rw----- 1 shambhut ugrad 3208 Jun  8 10:27 main-parse.o
-rw----- 1 shambhut ugrad 2629 Jun  7 16:45 parse.h
-rw----- 1 shambhut ugrad 1361 Jun  8 10:26 main-parse.c
? █
```

TEST SCRIPTS AND HOW TO USE THEM

Following are the test scripts for my shell that I have submitted

- test_infile
- test_outfileredirect
- testing_outfile_append
- testing_multiplePipe_cmd
- testing_cd_cmd

To run my test scripts

```
>> make
```

```
>> ./nsh testFilename
```

SCREENSHOT OF THE OUTPUT OF TEST SCRIPTS

i) test_infile

```
shambhut@odin 14:29:39 ~/CompSci146/nsh
$ ./nsh test_infile
"testing input redirection"
"-----"
" taking input as main-parse.c and counting the lines "
86
shambhut@odin 14:32:57 ~/CompSci146/nsh
$ █
```

ii) test_outfileredirect

```
$ ./nsh test_outfileredirect
"testing output redirection"
"making a new file called foo1991"
"redirectig the output of date command"
Fri Jun  8 14:34:53 PDT 2018
"now deleting the file"
```

iii) testing_outfile_append

```
$ ./nsh testing_outfile_append
Testing Append to File
-----
hello world
content after appending
-----
hello world
24
shambhut@odin 14:46:59 ~/CompSci146/nsh
```

iv) testing_multiplePipe_cmd

```
$ ./nsh testing_multiplePipe_cmd
"testing multiple pipe"
"-----"
"first command ls-l , second one grep parse , last one sort by file size reverse"
-rw----- 1 shambhut ugrad 6661 Jun  8 10:57 parse.c
-rw----- 1 shambhut ugrad 5176 Jun  8 10:57 parse.o
-rw----- 1 shambhut ugrad 3208 Jun  8 10:27 main-parse.o
-rw----- 1 shambhut ugrad 2629 Jun  7 16:45 parse.h
-rw----- 1 shambhut ugrad 1361 Jun  8 10:26 main-parse.c
shambhut@odin 14:52:55 ~/CompSci146/nsh
$ █
```


v) testing_cd_cmd

```
$ ./nsh testing_cd_cmd
"Testing cd command"
"_____
"making a unique folder temp1991"
/home/shambhut/CompSci146/nsh
/home/shambhut/CompSci146/nsh/temp1991
/home/shambhut/CompSci146/nsh
"now deleting that temporary dir"
shambhut@odin 14:54:29 ~/CompSci146/nsh
$
```

Discovered Bug

I have a small bug in nsh shell, that is > operator works when there is more than one pipe

Who | grep user | > foo overwrites the foo content works perfectly

Who > foo does not overwrite the foo content.

+++++ END OF DOCUMENTATION +++++