

### Problem Set 3: The Shell

Screenshot #1: shell ignores comment and runs command `ls` with argument `"-l"`. The output is redirected to file `out`. The process stats are reported. Then, to verify that the output of `ls` really did go to `out`, run command `cat`. Again, process stats for `cat` are reported.

```
Terminal
rm shell
gcc shell.c -o shell
./shell
$ simple shell prompt > ls -l >out

Finished executing command: ls w/ arguments: "-l"
PID: 3549      Exit Status: 0
Real: 0.068865 seconds  User: 0.000000 seconds  System: 0.004000 seconds

$ simple shell prompt > cat out
total 108
-rw-r--r-- 1 george george  75 Oct  1 09:21 demo.c
-rw-r--r-- 1 george george 387 Oct  1 15:57 ls.out
-rw-rw-rw- 1 george george  75 Oct  1 15:26 Makefile
-rw-r--r-- 1 george george   0 Oct  1 16:06 out
-rw-r--r-- 1 george george 170 Oct  1 11:19 script
-rwxr-xr-x 1 george george 12439 Oct  1 16:06 shell
-rw-rw-rw- 1 george george 8741 Oct  1 15:53 shell.c
-rw-r--r-- 1 george george 63198 Oct  1 16:05 Writeup.odt

Finished executing command: cat w/ arguments: "out"
PID: 3551      Exit Status: 0
Real: 0.024537 seconds  User: 0.000000 seconds  System: 0.000000 seconds

$ simple shell prompt > █
```

Screenshot #2: shell runs `cat` on three files: `demo.c`, `blah`, and `blahblah` and redirects output to `out` (created from above). Error messages are redirected to `error`. Then run `cat` to show contents of `out` and `error`

```
Terminal

$ simple shell prompt > cat demo.c blah blahblah >>out 2>error
shell: cat exited with nonzero value 1
$ simple shell prompt > cat out
total 132
-rwxr-xr-x 1 george george 12454 Oct  1 16:21 a.out
-rw-r--r-- 1 george george  75 Oct  1 09:21 demo.c
-rw-rw-rw- 1 george george  75 Oct  1 15:26 Makefile
-rw-rw-rw- 1 george george 6909 Sep 27 18:19 os3.c
-rw-r--r-- 1 george george   0 Oct  1 16:35 out
-rw-r--r-- 1 george george 170 Oct  1 11:19 script
-rwxr-xr-x 1 george george 12439 Oct  1 16:34 shell
-rw-rw-rw- 1 george george 8711 Oct  1 16:29 shell.c
-rw-r--r-- 1 george george 67483 Oct  1 16:08 Writeup.odt
#include <stdio.h>

int main()
{
    printf("Hello, World!\n");
    return 0;
}

Finished executing command: cat w/ arguments: "out"
PID: 3997      Exit Status: 0
Real: 0.002385 seconds  User: 0.000000 seconds  System: 0.000000 seconds

$ simple shell prompt > cat error
cat: blah: No such file or directory
cat: blahblah: No such file or directory

Finished executing command: cat w/ arguments: "error"
PID: 3999      Exit Status: 0
Real: 0.001576 seconds  User: 0.000000 seconds  System: 0.000000 seconds

$ simple shell prompt >
```

As shown in the screenshot, demo.c was a valid file and its contents were appended to the end of out. Since blah and blahblah are not existing files, error messages were generated by cat and sent to error.

Screenshot #3: Running shell as an interpreter for script file script from within shell:

```
./shell
$ simple shell prompt > ./script
This is a script file called script...

Finished executing command: echo w/ arguments: "This" "is" "a" "script" "file" "called" "script..."
PID: 4806      Exit Status: 0
Real: 0.002012 seconds  User: 0.000000 seconds  System: 0.000000 seconds

#!/home/george/Desktop/ECE460/PS3/shell
echo This is a script file called script...
cat <script
gcc demo.c -o demo
./demo
rm demo

Finished executing command: cat w/ arguments: NONE
PID: 4807      Exit Status: 0
Real: 0.002060 seconds  User: 0.000000 seconds  System: 0.000000 seconds

Finished executing command: gcc w/ arguments: "demo.c" "-o" "demo"
PID: 4808      Exit Status: 0
Real: 0.074227 seconds  User: 0.056000 seconds  System: 0.008000 seconds

Hello, World!

Finished executing command: ./demo w/ arguments: NONE
PID: 4813      Exit Status: 0
Real: 0.000706 seconds  User: 0.000000 seconds  System: 0.000000 seconds

Finished executing command: rm w/ arguments: "demo"
PID: 4814      Exit Status: 0
Real: 0.000834 seconds  User: 0.000000 seconds  System: 0.000000 seconds

Finished executing command: ./script w/ arguments: NONE
PID: 4805      Exit Status: 0
Real: 0.082156 seconds  User: 0.056000 seconds  System: 0.008000 seconds

$ simple shell prompt > █
```

Screenshot #4: Running shell as an interpreter for script file script from bash terminal

```
Terminal
george@george-Vostro-1400 ~/Desktop/ECE460/PS3 $ ./shell script
This is a script file called script...

Finished executing command: echo w/ arguments: "This" "is" "a" "script" "file" "called" "script..."
PID: 4862      Exit Status: 0
Real: 0.002000 seconds  User: 0.000000 seconds  System: 0.000000 seconds

#!/home/george/Desktop/ECE460/PS3/shell
echo This is a script file called script...
cat <script
gcc demo.c -o demo
./demo
rm demo

Finished executing command: cat w/ arguments: NONE
PID: 4863      Exit Status: 0
Real: 0.002118 seconds  User: 0.000000 seconds  System: 0.000000 seconds

Finished executing command: gcc w/ arguments: "demo.c" "-o" "demo"
PID: 4864      Exit Status: 0
Real: 0.071000 seconds  User: 0.048000 seconds  System: 0.012000 seconds

Hello, World!

Finished executing command: ./demo w/ arguments: NONE
PID: 4869      Exit Status: 0
Real: 0.000794 seconds  User: 0.000000 seconds  System: 0.000000 seconds

Finished executing command: rm w/ arguments: "demo"
PID: 4870      Exit Status: 0
Real: 0.000977 seconds  User: 0.000000 seconds  System: 0.000000 seconds

george@george-Vostro-1400 ~/Desktop/ECE460/PS3 $ █
```

Screenshot #5: Running a.out with input file: in output file: out and error log: error

```
Terminal
$ simple shell prompt > ./a.out <in >>out 2>error

Finished executing command: ./a.out w/ arguments: NONE
PID: 3085      Exit Status: 0
Real: 0.001969 seconds  User: 0.000000 seconds  System: 0.000000 seconds

$ simple shell prompt > cat in
This is input file: in

Finished executing command: cat w/ arguments: "in"
PID: 3087      Exit Status: 0
Real: 0.002255 seconds  User: 0.000000 seconds  System: 0.000000 seconds

$ simple shell prompt > cat out
total 424
-rwxr-xr-x 1 george george 7451 Oct 2 03:18 a.out
-rwxr-xr-x 1 george george 7291 Oct 2 10:20 demo
-rw-r--r-- 1 george george 285 Oct 2 09:29 demo2.c
-rw-r--r-- 1 george george 75 Oct 1 09:21 demo.c
-rw-rw-rw- 1 george george 75 Oct 1 15:26 Makefile
-rw-r--r-- 1 george george 0 Oct 2 10:25 out
-rwxr--r-- 1 george george 139 Oct 2 00:41 script
-rwxr-xr-x 1 george george 12412 Oct 2 10:24 shell
-rw-rw-rw- 1 george george 7106 Oct 2 02:23 shell.c
-rw-r--r-- 1 george george 372894 Oct 2 10:20 Writeup.odt
Printing this message to stdout
Now printing input to stdout
This is input file: in

Finished executing command: cat w/ arguments: "out"
PID: 3088      Exit Status: 0
Real: 0.002307 seconds  User: 0.000000 seconds  System: 0.000000 seconds

$ simple shell prompt > cat error
Printing this message to stderr

Finished executing command: cat w/ arguments: "error"
PID: 3090      Exit Status: 0
Real: 0.002235 seconds  User: 0.000000 seconds  System: 0.000000 seconds

$ simple shell prompt >
```

The following is demo2.c, the source file used to compile a.out which was invoked in the above screen shot:

```
#include <stdio.h>

int main() {
    int c, i = 0;
    char buf[1024];
    while((c = getchar()) != '\n') {
        *(buf + i) = c;
        i++;
    }
    *(buf + i) = '\0';
    fprintf(stderr, "Printing this message to stderr\n");
    printf("Now printing our input to stdout:\n");
    printf("%s\n", buf);
    return 0;
}
```

This was the script used for the 2 previous screenshots:

```
#!/home/george/Desktop/ECE460/PS3/shell
echo This is a script file called script...
cat <script
gcc demo.c -o demo
./demo
rm demo
```

This is demo.c that was compiled in the above script:

```
#include <stdio.h>

int main()
{
    printf("Hello, World!\n");
    return 0;
}
```

**\*\*IMPORTANT:** I did have a bug in my program where if the last line in my script file was not '\n', then the last command would not be executed.

Screenshot #6: Again, shell running another really simple script called script2 to show that redirection works

Terminal

```
george@george-Vostro-1400 ~/Desktop/ECE460/PS3 $ ./shell script2
Finished executing command: ./a.out w/ arguments: NONE
PID: 2433      Exit Status: 0
Real: 0.001039 seconds  User: 0.000000 seconds  System: 0.000000 seconds

george@george-Vostro-1400 ~/Desktop/ECE460/PS3 $ cat in
This is input file: in
george@george-Vostro-1400 ~/Desktop/ECE460/PS3 $ cat script_out
Now printing our input to stdout:
This is input file: in
george@george-Vostro-1400 ~/Desktop/ECE460/PS3 $ cat script_error
Printing this message to stderr
george@george-Vostro-1400 ~/Desktop/ECE460/PS3 $
```

The following is script2:

```
#!/home/george/Desktop/ECE460/PS3/shell
./a.out <in >script_out 2>script_error
```

a.out is the same as before: demo2.c

The following is the Makefile used to build shell:

```
shell: clean shell.c
    gcc shell.c -o shell
    ./shell
clean:
    clear
    rm shell
```