

Mini-Shell Test Guide

Use this guide to verify your implementation. Test each section in order.

How to Use

1. Compile your shell: `make`
 2. Run your shell: `./mysh`
 3. Execute each test command below
 4. Check if the output matches the expected result
 5. Mark each test as PASS or FAIL
-

Test 1: Basic Command Execution

```
1 | mysh> ls
```

Expected: List of files in current directory

```
1 | mysh> ls -l
```

Expected: Detailed file listing

```
1 | mysh> ls -l -a
```

Expected: Detailed listing including hidden files

```
1 | mysh> echo hello world
```

Expected: `hello world`

```
1 | mysh> /bin/echo test
```

Expected: `test`

Test 2: Built-in Commands

Test 2.1: pwd

```
1 | mysh> pwd
```

Expected: Current working directory path

Test 2.2: cd with argument

```
1 | mysh> cd /tmp  
2 | mysh> pwd
```

Expected: /tmp

Test 2.3: cd without argument (go to HOME)

```
1 | mysh> cd  
2 | mysh> pwd
```

Expected: Your home directory (e.g., /Users/yourname on macOS, /home/yourname on Linux)

Test 2.4: cd to invalid directory

```
1 | mysh> cd /nonexistent_directory_12345
```

Expected: Error message (e.g., cd failed: No such file or directory), shell continues running

Test 2.5: exit

```
1 | mysh> exit
```

Expected: Shell terminates

Restart your shell with ./mysh before continuing.

Test 3: Output Redirection

Test 3.1: Basic output redirection (>)

```
1 | mysh> echo hello > test1.txt  
2 | mysh> cat test1.txt
```

Expected: hello

Test 3.2: Overwrite existing file (>)

```
1 | mysh> echo first > test1.txt
2 | mysh> echo second > test1.txt
3 | mysh> cat test1.txt
```

Expected: `second` (only one line, "first" was overwritten)

Test 3.3: Redirect command output (>)

```
1 | mysh> ls > filelist.txt
2 | mysh> cat filelist.txt
```

Expected: List of files in the directory

Test 4: Append Redirection

Test 4.1: Append to file (>>)

```
1 | mysh> echo line1 > append_test.txt
2 | mysh> echo line2 >> append_test.txt
3 | mysh> echo line3 >> append_test.txt
4 | mysh> cat append_test.txt
```

Expected:

```
1 | line1
2 | line2
3 | line3
```

Test 5: Input Redirection

Test 5.1: Basic input redirection (<)

```
1 | mysh> echo one > count_test.txt
2 | mysh> echo two >> count_test.txt
3 | mysh> echo three >> count_test.txt
4 | mysh> wc -l < count_test.txt
```

Expected: `3` (or `3` with padding)

Test 5.2: Input redirection with different command

```
1 | mysh> cat < count_test.txt
```

Expected:

```
1 | one
2 | two
3 | three
```

Test 6: Pipe

Test 6.1: Basic pipe

```
1 | mysh> ls | wc -l
```

Expected: Number of files in directory

Test 6.2: Pipe with grep

```
1 | mysh> ls | grep mysh
```

Expected: Lines containing "mysh" (should show `mysh` and `mysh.c`)

Test 6.3: Pipe with arguments on both sides

```
1 | mysh> ls -l | grep "\.c"
```

Expected: Only lines containing ".c"

Test 6.4: Pipe with echo

```
1 | mysh> echo hello world | wc -w
```

Expected: 2

Test 7: Error Handling

Test 7.1: Command not found

```
1 | mysh> nonexistent_command_xyz
```

Expected: Error message, shell continues running

Test 7.2: Cannot open file for redirection

```
1 | mysh> cat < nonexistent_file.txt
```

Expected: Error message (e.g., `open failed: No such file or directory`), shell continues

Test 7.3: Empty input

```
1 | mysh>
```

Expected: Just prints another prompt, no error

Test 8: EOF Handling

Press `Ctrl+D` on an empty prompt.

Expected: Shell exits gracefully (may print a newline)

Cleanup

After testing, remove test files:

```
1 | rm -f test1.txt filelist.txt append_test.txt count_test.txt
```

Summary Checklist

| Test | Feature | PASS/FAIL |
|------|----------------------------|-----------|
| 1 | Basic command execution | |
| 2.1 | pwd | |
| 2.2 | cd with argument | |
| 2.3 | cd without argument | |
| 2.4 | cd error handling | |
| 2.5 | exit | |
| 3.1 | Output redirection (>) | |
| 3.2 | Overwrite with (>) | |
| 3.3 | Redirect command output | |
| 4.1 | Append redirection (>>) | |
| 5.1 | Input redirection (<) | |
| 5.2 | Input redirection with cat | |
| 6.1 | Basic pipe | |
| 6.2 | Pipe with grep | |
| 6.3 | Pipe with arguments | |
| 6.4 | Pipe with echo | |
| 7.1 | Command not found error | |
| 7.2 | File not found error | |
| 7.3 | Empty input | |
| 8 | EOF handling (Ctrl+D) | |