

VG101 Lab Manual

Lab 5

Instructor: Dr. Yifei ZHU

TA: Hangrui CAO

TA: Haoxuan SHAN

TA: Qinhang WU

TA: Muchen XU

Table of Contents

- MISC: WSL
- MISC: Shell
- Getting Start with C
- Tricks in C

Workflow

Content	Approx. Time
MISC: WSL (<i>for Windows user only</i>)	20 mins
MISC: Shell	25 mins
Compile C Program	20 mins
Break	10 mins
Warm-up: Basic Syntax Exercise	25 mins
Seminar: Tricks in C	20 mins
Break	10 mins
Exercise	50 mins

Command Line Interface (Shell)

Computers these days have a variety of interfaces for giving them commands; fanciful graphical user interfaces, voice interfaces, and even AR/VR are everywhere. These are great for 80% of use-cases, but they are often fundamentally restricted in what they allow you to do — you cannot press a button that isn't there or give a voice command that hasn't been programmed. To take full advantage of the tools your computer provides, we have to go old-school and drop down to a textual interface: The Shell.

Nearly all platforms you can get your hand on has a shell in one form or another, and many of them have several shells for you to choose from. While they may vary in the details, at their core they are all roughly the same: they allow you to run programs, give them input, and inspect their output in a semi-structured way.

Find a Shell



- Mac User
 - Find the terminal icon in your applications dock at the bottom of your screen and click it.
 - Type `CMD + SPACE` or click the magnifying glass in your top right corner to open up the "Spotlight" application and then just type "Terminal". It should be the first result.



- Windows User
 - Open the "Command Prompt" or "Git Bash" by searching in the Windows Search Bar.
 - Input the shell you'd like to use (including `cmd`, `ws1`, or `powershell`) in the address bar and press `Enter` to open a terminal in the current folder.
- Linux User
 - I believe you can find your way :D.

When you launch your terminal, you will see a prompt that often looks like this:

```
1 | $
```

or

```
1 | >
```

First Command

Type in `whoami` and press `Enter` to get your username.

Now we will only focus on Unix Shell. For Windows Command Prompt users, please refer to [Intro to command line](#) or the appendix to look up the equivalent commands.

Useful Shell Command

- `pwd`: print working directory.
- `cd`: change directory.
 - For example, `cd ../` brings you to your parent directory.
- `ls`: list files and folders under a directory.
- `mkdir`: make directory.
- `rm`: remove.
 - This is an extremely dangerous command. See the famous [bumblebee accident](#).
- `cp`: copy.
 - Takes 2 arguments: `source` and `dest`.
 - Be very careful if both source and destination are existing folders.
 - `-r` Copy files/folders recursively. Folders requires this option.
- `mv`: move.

- Takes 2 arguments: `source` and `dest`.
 - Be very careful if both source and destination are existing folders.
 - Can be used for rename by making `source = dest`.
- `cat`: concatenate.
 - Takes multiple arguments and print their content one by one to `stdout`.
- `echo`: prints its arguments to `stdout`.
- `man [command]`: get help for a command.

Compile C Program

It is well-known that C is a compiled language, which must be compiled is before running. So in this lab section, we will guide you how to compile a C program smartly in command line.

Compile C with gcc

- `gcc`: GNU project C and C++ compiler
 - `infile`: pass in the file to be compiled.
 - `gcc test.cpp` will generate an executable called "a.out".
 - `-std=standard`: specify the standard you are using.
 - `-std=c11` means the program is compiled under standard `c11`.
 - `-o outfile`: specify the name of the output file.
 - `-o test` will generate an executable called "test.out".
 - `-Dmacro`: specify the macro that you apply before compiling.
 - `-DScarlet` is equivalent to `#define Scarlet` for all input files.
 - `-g`: turn on debugging options.
 - `-Wwarn`: turn on compile warning options.
 - `-Wall`: turn on **all** compile warning. Always adding `-Wall` will help you a lot.
 - `-Werror`: stop compiling when any warning occurs.
 - `-Olevel`: turn on the optimization.
 - `-O3` will optimize more compared with `-O2`.
 - `-O0` performs no optimization (default option).

Grading Rubric

Criteria	Weight	Available Time	Due Time	Entry
Attendance	30%	4:00pm, June.12	11:59am, June.13	Canvas Assignment
In-lab quiz	70%	9:00pm, June.12	11:59pm, June.14	Canvas Quiz
Shell bonus	20%	9:00pm, June.12	2:00pm, June.14	Canvas Quiz

- For the attendance score, you need to submit your code for **exercises** in the **worksheet** on Canvas. We won't judge the correctness of your code. You'll earn full credits as long as you've tried these exercises.
- Unless extreme cases, late lab submissions will not be accepted.

- The attendance, in-lab quiz, and the Shell bonus will be ceiled together if greater than 100%.

Appendix

Equivalent Command in Command Prompt and Unix Shell

Here are several equivalent command in Command Prompt and Unix Shell.

Command (Windows)	Command (Mac OS / Linux / WSL)	Description	Example
exit	exit	close the window	exit
cd	cd	change directory	cd test
cd	pwd	show the current directory	cd (Windows) or pwd (Mac OS / Linux)
dir	ls	list directories/files	dir
copy	cp	copy file	copy c:\test\test.txt c:\windows\test.txt
move	mv	move file	move c:\test\test.txt c:\windows\test.txt
mkdir	mkdir	create a new directory	mkdir testdirectory
rmdir (or del)	rm	delete a file	del c:\test\test.txt
rmdir /S	rm -r	delete a directory	rm -r testdirectory
type	cat	get content in the file	type c:\windows\test.txt
[CMD] /?	man [CMD]	get help for a command	cd /? (Windows) or man cd (Mac OS / Linux)

Windows Subsystem for Linux (WSL)

Windows Subsystem for Linux is a desktop environment inside Windows, which allows you to directly run your program virtually on Linux, without building a development environment by virtual machines.

[Installation tutorial from VG100-Manuel-SU2020](#)

[Installation Tutorial with illustration](#). You only need to follow instructions before the `install` `vcxsrv` sections.

Credit

- [Shell](#), Missing-Semester
- [A Command Line Crash Course](#), Viking Code School
- [Intro to command line](#), Django Girls
- [Shell](#), Ziqiao Ma