

C/C++ Program Design

LAB 1

Qimei Liao (廖琪梅)

Email: liaoqm@mail.sustech.edu.cn

Office: Room 509 , Building 10, Innovation Park

CONTENTS

- Download and install GCC on Windows 10 (Based on Windows Subsystem for Linux)
- Download and install LLVM on macOS
- Download and install GCC on Windows x (Based on Cygwin)
- Download and install the editor (VSCode)
- Compile, link and run C/C++ programs
- Exercises

1 Download and install GCC on Windows 10 (Based on WSL)

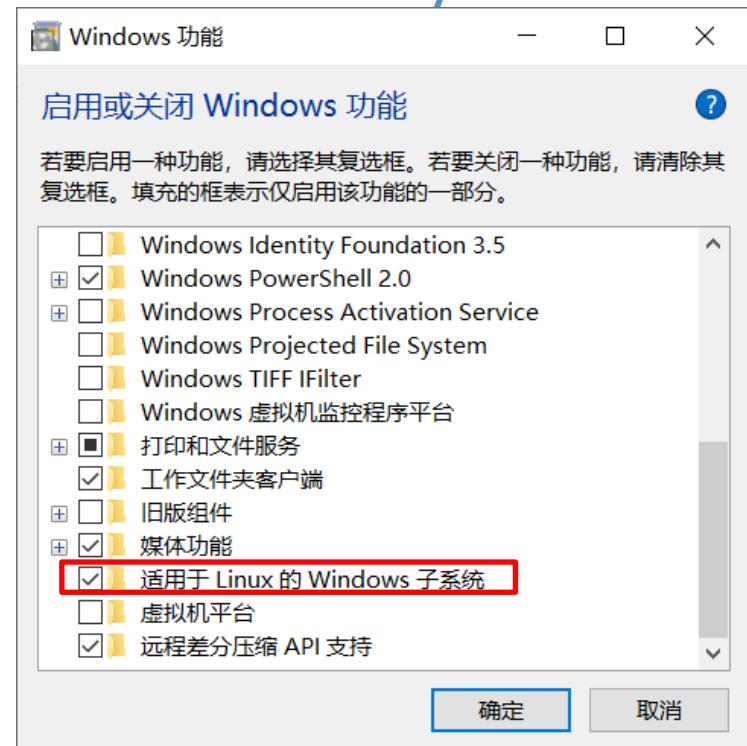
1.1 Install WSL on Windows 10

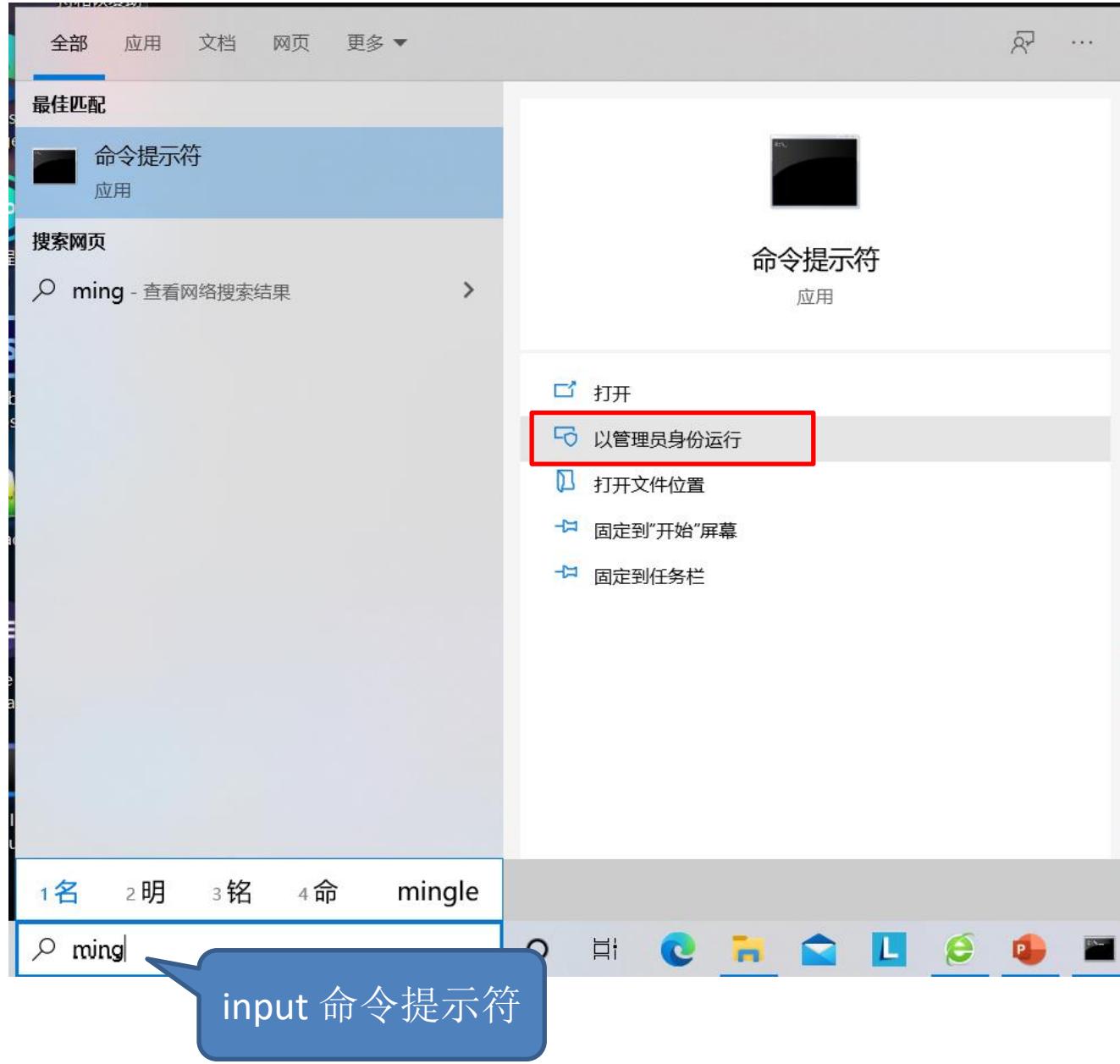
**1.2 Install GCC (the GNU Compiler Collection) on
WSL**

1.3 Verify GCC on WSL

1.1 Install WSL on Windows 10

- **Step one:** enable the Windows Subsystem for Linux
 - Open powershell as Administrator and run:
`dism.exe /online /enable-feature
/featurename:Microsoft-Windows-Subsystem-Linux
/all /norestart`





input 命令提示符

管理员: 命令提示符

C:\Windows\system32>dim.exe/online/enable-feature/featurename:Microsoft-Windows-Subsystem-Linux/all/norestart
'dim.exe' 不是内部或外部命令，也不是可运行的程序
或批处理文件。

C:\Windows\system32>dism.exe/online/enable-feature/featurename:Microsoft-Windows-Subsystem-Linux/all/norestart
部署映像服务和管理工具
版本: 10.0.18362.1139

错误: 87

online/enable-feature/featurename 选项未知。
有关详细信息，请通过运行 DISM.exe /? 参阅帮助。

可以在 C:\Windows\Logs\DISM\dism.log 上找到 DISM 日志文件

C:\Windows\system32>dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
部署映像服务和管理工具
版本: 10.0.18362.1139

映像版本: 10.0.18363.1198

启用一个或多个功能

[=====100.0%=====]

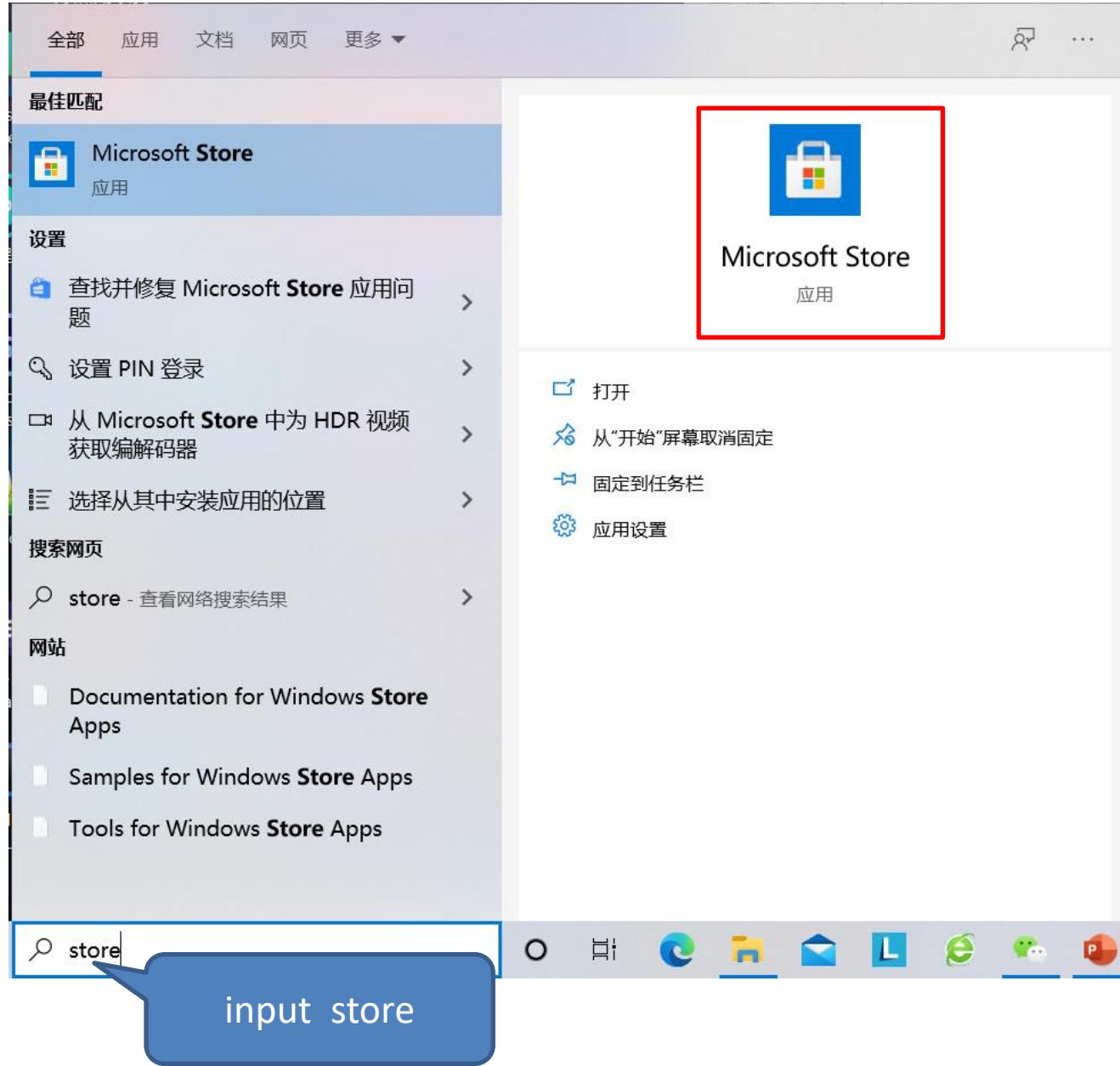
操作成功完成。

C:\Windows\system32>

Notice the space
in the commands

1.1 Install WSL on Windows 10(cont.)

- **Step two:** restart your machine
- **Step three:** install your Linux distribution of choice:
 - Open the **Microsoft Store** and select your preferred Linux distribution
 - Search Linux in Microsoft Store



主页 游戏 娱乐 高效工作

search Linux

Linux



↓ 19



select Ubuntu 应用

Ubuntu
应用Kali Linux
应用Linux Cheatsheet
应用SUSE Linux Enterprise Server 15
SP1
应用SUSE Linux Enterprise Server 12
SP5
应用

Minecraft for Windows 10 Starter Collection



Microsoft 365



精选娱乐应用

显示全部



← 主页 游戏 娱乐 高效工作 促销品搜索 19

Ubuntu

Canonical Group Limited • 开发人员工具 > 实用程序共享

Ubuntu on Windows allows you to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.

更多

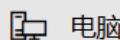
3+

免费获取

你可以在 Xbox One 主机上购买。(你所在的地区不支持通过 microsoft.com 购买。)

概述系统要求评论相关

可用于



电脑

介绍

跨设备使用

通过 Microsoft 登录，并在任何兼容设备上使用。

不，谢谢

登录

X



登录

maydlee@163.com

input your
Microsoft account

X

没有帐户?[创建一个!](#)

If you have no account, you
can create one

[忘记了用户名](#)

下一步

X



← maydleee@163.com

输入密码

input your
password

.....



[忘记了密码?](#)

登录



maydlee@163.com

在设备上的任何位置使用此帐户

Windows 会记住帐户，并简化登录应用和网站的过程。单击“下一步”即表示能够查找丢失设备、将设置同步到其他设备和向 Cortana 寻求帮助。

[仅限 Microsoft 应用](#)

下一步

×

对你的帐户使用 Windows Hello

若要使用 maydle@163.com 轻松安全地访问应用，需要为它设置 Windows Hello 人脸、指纹或 PIN。

如果你已设置，我们将为此帐户自动添加它。系统可能会要求你使用 Windows Hello 重新验证。

确定

← 主页 游戏 娱乐 高效工作 促销品搜索 20 ⋮

正在下载 Ubuntu .. 136.7 MB, 共 444.5 MB



获取关于更快速下载的详细信息 8.6 Mb/秒



X

启动



Ubuntu

Canonical Group Limited • 开发人员工具 > 实用程序共享

Ubuntu on Windows allows you to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.

更多

免费

获取

你可以在 Xbox One 主机上购买。(你所在的地区不支持通过 microsoft.com 购买。)

概述系统要求评论相关

[←](#)[主页](#)[游戏](#)[娱乐](#)[高效工作](#)[促销品](#) [搜索](#)

↓19

...



此产品已安装。

[启动](#)

...



Ubuntu

Canonical Group Limited · [开发人员工具 > 实用程序](#) [共享](#)

Ubuntu on Windows allows you to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.

[更多](#)

3+

免费

你可以在 Xbox One 主机上购买。(你所在的地区不支持通过 microsoft.com 购买。)

[概述](#)[系统要求](#)[评论](#)[相关](#)

maydee@LAPTOP-U1MOON2F: ~

Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: <https://aka.ms/>

Enter new UNIX username: maydee

New password:

Retype new password:

passwd: password updated successfully

Installation successful!

To run a command as administrator (user "root"), use "sudo <command>".

See "man sudo_root" for details.

Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 4.4.0-18362-Microsoft x86_64)

- * Documentation: <https://help.ubuntu.com>
- * Management: <https://landscape.canonical.com>
- * Support: <https://ubuntu.com/advantage>

System information as of Wed Jan 13 19:02:56 CST 2021

System load:	0.52
Usage of /home:	unknown
Memory usage:	33%
Swap usage:	0%
Processes:	7
Users logged in:	0
IPv4 address for wifi0:	10.20.87.166
IPv6 address for wifi0:	2001:da8:201d:1101:915c:c8fe:13c8:54ae

1 update can be installed immediately.
0 of these updates are security updates.

input new UNIX username
and new password

```
选择maydlee@LAPTOP-U1MOON2F: ~

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Wed Jan 13 19:02:56 CST 2021

System load:          0.52
Usage of /home:       unknown
Memory usage:         33%
Swap usage:           0%
Processes:            7
Users logged in:     0
IPv4 address for wifi0: 10.20.87.166
IPv6 address for wifi0: 2001:da8:201d:1101:915c:c8fe:13c8:54ae

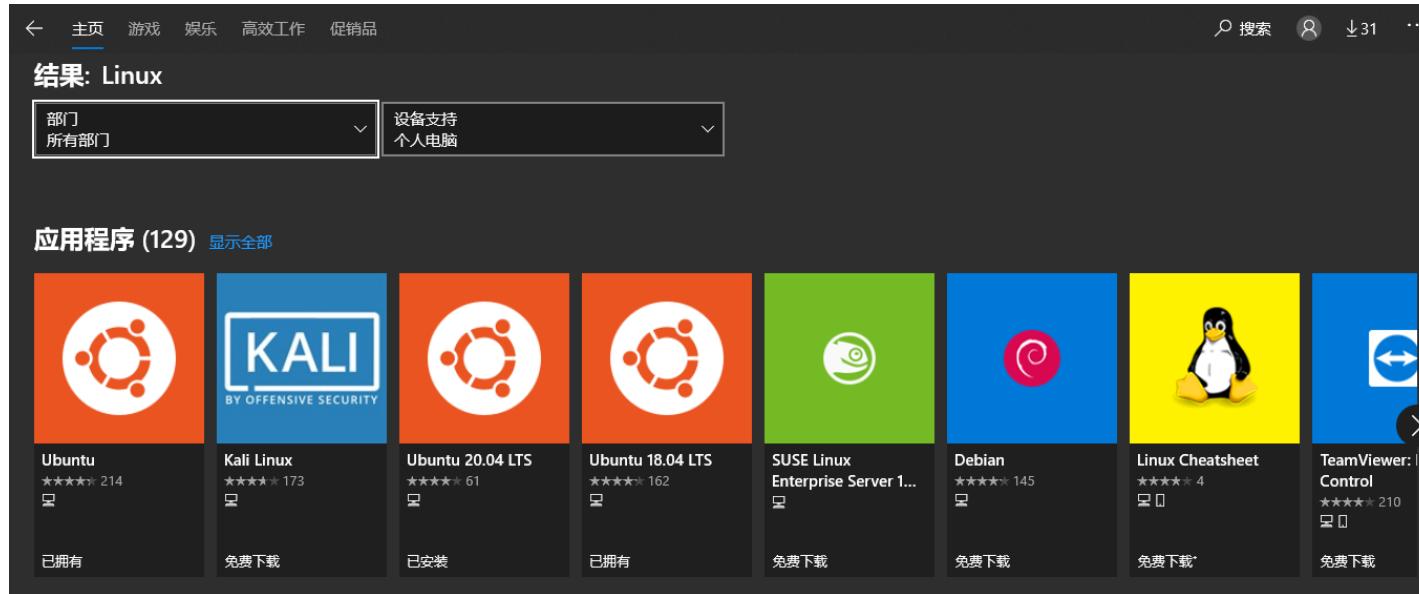
1 update can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

This message is shown once once a day. To disable it please create the
/home/maydlee/.hushlogin file.
maydlee@LAPTOP-U1MOON2F:~$
```

Linux command prompt

1.1 Install WSL on Windows 10 (Cont.)

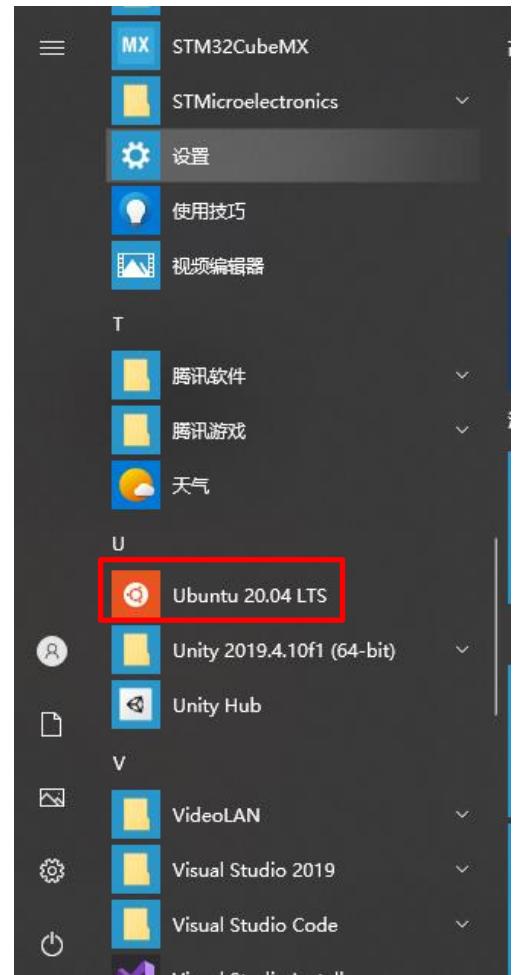


We prefer Ubuntu 20.04 LTS and use this as an example in our class as this is the latest LTS version.

If you are interested in WSL2 which supports more features of Linux (like Docker) you can visit: <https://docs.microsoft.com/zh-cn/windows/wsl/install-win10#manual-installation-steps> for more information.

1.2 Install GCC on WSL

- Once you finished the installation of Ubuntu 20.04 LTS, you can find it on your start menu.
- Open it and you will see a Terminal for Linux
- You can set username and password for your system (Please remember this password as you need it to switch to root user later)



1.2 Install GCC on WSL (Cont.)

- You can change the source list of apt if you want.
(This step may improve your download speed later)
 - For more information you can visit:
<https://mirrors.sustech.edu.cn/help/ubuntu.html#introduction>
- Use the command below to install GNU: (If you are using any Linux distribution based on debian you can use below to install, too)
 - `sudo apt update;sudo apt install g++ -y`
 - The first command will update your apt library
(apt:Advanced Packaging Tools)
 - The second will install g++ and its independence

maydleee@LAPTOP-U1MOON2F: ~

```
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
```

System information as of Wed Jan 13 19:02:56 CST 2021

```
System load:          0.52
Usage of /home:       unknown
Memory usage:         33%
Swap usage:           0%
Processes:            7
Users logged in:     0
IPv4 address for wifi0: 10.20.87.166
IPv6 address for wifi0: 2001:da8:201d:1101:915c:c8fe:13c8:54ae
```

```
1 update can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable
```

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

This message is shown once once a day. To disable it
/home/maydleee/.hushlogin file.

```
maydleee@LAPTOP-U1MOON2F: ~$ sudo apt update
[sudo] password for maydleee: _
```

input the command and
your password

maydlee@LAPTOP-U1MOON2F: ~

To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.

To check for new updates run: sudo apt update

This message is shown once once a day. To disable it please create the
/home/maydlee/.hushlogin file.

maydlee@LAPTOP-U1MOON2F: \$ sudo apt update

[sudo] password for maydlee:

```
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Hit:2 http://archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [438 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [97.2 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [5616 B]
Get:9 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [107 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [15.9 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 c-n-f Metadata [376 B]
Get:12 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [523 kB]
Get:13 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [70.2 kB]
Get:14 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [9464 B]
Get:15 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [832 B]
Get:16 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [288 B]
Get:17 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [116 B]
Get:18 http://archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
```

74% [18 Translation-en 3091 kB/5124 kB 60%]

239 kB/s 19s

```
maydee@LAPTOP-U1MOON2F: ~
Reading package lists... Done
Building dependency tree
Reading state information... Done
155 packages can be upgraded. Run 'apt list --upgradable'
maydee@LAPTOP-U1MOON2F ~$ sudo apt install g++ -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
binutils binutils-common binutils-x86_64-linux-gnu cpp cpp-9 g++-9 gcc gcc-10-base gcc-9 gcc-9-base libasan5
libatomic1 libbinutils libc-dev-bin libc6 libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libgcc-9-dev
libgcc-s1 libgomp1 libis122 libitm1 liblsan0 libmpc3 libquadmath0 libstdc++-9-dev libstdc++6 libtsan0 libubsan1
linux-libc-dev manpages-dev
Suggested packages:
binutils-doc cpp-doc gcc-9-locales g++-multilib g++-9-multilib gcc-9-doc gcc-multilib make autoconf automake libtool
flex bison gdb gcc-doc gcc-9-multilib glibc-doc libstdc++-9-doc
The following NEW packages will be installed:
binutils binutils-common binutils-x86_64-linux-gnu cpp cpp-9 g++ g++-9 gcc gcc-9 gcc-9-base libasan5 libatomic1
libbinutils libc-dev-bin libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libgcc-9-dev libgomp1 libis122
libitm1 liblsan0 libmpc3 libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev manpages-dev
The following packages will be upgraded:
gcc-10-base libc6 libgcc-s1 libstdc++6
4 upgraded, 31 newly installed, 0 to remove and 151 not upgraded.
Need to get 42.0 MB of archives.
After this operation, 169 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 gcc-10-base amd64 10.2.0-5ubuntu1~20.04 [19.7 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libstdc++6 amd64 10.2.0-5ubuntu1~20.04 [503 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libgcc-s1 amd64 10.2.0-5ubuntu1~20.04 [41.6 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libc6 amd64 2.31-0ubuntu9.1 [2712 kB]
5% [4 libc6 1068 kB/2712 kB 39%] 193 kB/s 3min 29s
```

input the command to install g++

 maydlee@LAPTOP-U1MOON2F: ~

```
Setting up libquadmath0:amd64 (10.2.0-5ubuntu1~20.04) ...
Setting up libmpc3:amd64 (1.1.0-1) ...
Setting up libatomic1:amd64 (10.2.0-5ubuntu1~20.04) ...
Setting up libubsan1:amd64 (10.2.0-5ubuntu1~20.04) ...
Setting up libcrypt-dev:amd64 (1:4.4.10-10ubuntu4) ...
Setting up libisl22:amd64 (0.22.1-1) ...
Setting up libbinutils:amd64 (2.34-6ubuntu1) ...
Setting up libc-dev-bin (2.31-0ubuntu9.1) ...
Setting up libcc1-0:amd64 (10.2.0-5ubuntu1~20.04) ...
Setting up liblsan0:amd64 (10.2.0-5ubuntu1~20.04) ...
Setting up libitm1:amd64 (10.2.0-5ubuntu1~20.04) ...
Setting up gcc-9-base:amd64 (9.3.0-17ubuntu1~20.04) ...
Setting up libtsan0:amd64 (10.2.0-5ubuntu1~20.04) ...
Setting up libctf0:amd64 (2.34-6ubuntu1) ...
Setting up libasan5:amd64 (9.3.0-17ubuntu1~20.04) ...
Setting up cpp-9 (9.3.0-17ubuntu1~20.04) ...
Setting up libc6-dev:amd64 (2.31-0ubuntu9.1) ...
Setting up binutils-x86-64-linux-gnu (2.34-6ubuntu1) ...
Setting up binutils (2.34-6ubuntu1) ...
Setting up libgcc-9-dev:amd64 (9.3.0-17ubuntu1~20.04) ...
Setting up cpp (4:9.3.0-1ubuntu2) ...
Setting up gcc-9 (9.3.0-17ubuntu1~20.04) ...
Setting up libstdc++-9-dev:amd64 (9.3.0-17ubuntu1~20.04) ...
Setting up gcc (4:9.3.0-1ubuntu2) ...
Setting up g++-9 (9.3.0-17ubuntu1~20.04) ...
Setting up g++ (4:9.3.0-1ubuntu2) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Processing triggers for man-db (2.9.1-1) ...
```

Progress: [99%] [#####]

1.3 Verify GCC on WSL

- You can input command: `gcc --version` or `g++ --version` to check whether the GCC is installed well

```
maydlee@LAPTOP-U1MOON2F: ~
Setting up libbinutils:amd64 (2.34-6ubuntu1) ...
Setting up libc-dev-bin (2.31-0ubuntu9.1) ...
Setting up libcc1-0:amd64 (10.2.0-5ubuntu1~20.04) ...
Setting up liblsan0:amd64 (10.2.0-5ubuntu1~20.04) ...
Setting up libitm1:amd64 (10.2.0-5ubuntu1~20.04) ...
Setting up gcc-9-base:amd64 (9.3.0-17ubuntu1~20.04) ...
Setting up libtsan0:amd64 (10.2.0-5ubuntu1~20.04) ...
Setting up libctf0:amd64 (2.34-6ubuntu1) ...
Setting up libasan5:amd64 (9.3.0-17ubuntu1~20.04) ...
Setting up cpp-9 (9.3.0-17ubuntu1~20.04) ...
Setting up libc6-dev:amd64 (2.31-0ubuntu9.1) ...
Setting up binutils-x86-64-linux-gnu (2.34-6ubuntu1) ...
Setting up binutils (2.34-6ubuntu1) ...
Setting up libgcc-9-dev:amd64 (9.3.0-17ubuntu1~20.04) ...
Setting up cpp (4:9.3.0-1ubuntu2) ...
Setting up gcc-9 (9.3.0-17ubuntu1~20.04) ...
Setting up libstdc++-9-dev:amd64 (9.3.0-17ubuntu1~20.04) ...
Setting up gcc (4:9.3.0-1ubuntu2) ...
Setting up g++-9 (9.3.0-17ubuntu1~20.04) ...
Setting up g++ (4:9.3.0-1ubuntu2) ...
update-alternatives: using /usr/bin/g++ to provide
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9)
maydlee@LAPTOP-U1MOON2F: ~$ gcc --version
gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Input `gcc --version` or `g++ --version` to check if the compiler is installed successfully

2 Download and install LLVM on macOS

- 2.1 Install Command Line Tools on macOS**
- 2.2 Verify the LLVM**

2.1 Install CLT on macOS

- Open the Terminal on your Mac
- Input `g++` to check whether the CLT or GCC is installed
- If not, the system will guide you to install CLT
- You can also install CLT by package provided by Apple:

<https://developer.apple.com/download/more/>

L

2.2 Verify LLVM on macOS

- The same as verifying GNU, using: `g++ --version`

```
[gdjs2@xiaoZhaoqideMacBook-Pro ~]
$ g++ --version
Configured with: --prefix=/Library/Developer/CommandLineTools/usr --with-gxx-include-dir=/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/c++/4.2.1
Apple clang version 12.0.0 (clang-1200.0.32.28)
Target: x86_64-apple-darwin20.2.0
Thread model: posix
InstalledDir: /Library/Developer/CommandLineTools/usr/bin
```

3.1 Install Cygwin(Optional)

Download url: <http://cygwin.com/install.html>

The screenshot shows the Cygwin installation page. On the left is a sidebar with links for Cygwin, Cygwin/X, Community, Documentation, Contributing, and Related Sites. The main content area has a large 'Cygwin' logo and the tagline 'Get that *Linux* feeling - on Windows'. Below this is a section titled 'Installing and Updating Cygwin Packages'. It contains two main download links: 'setup-x86_64.exe' for 64-bit Windows and 'setup-x86.exe' for 32-bit Windows. A red arrow points from the text 'Select an appropriate version to download based on your OS' to the 'setup-x86.exe' link.

Cygwin
Install Cygwin
Update Cygwin
Search Packages
Licensing Terms

Cygwin/X

Community
Reporting Problems
Mailing Lists
Newsgroups
IRC channels
Gold Stars
Mirror Sites
Donations

Documentation
FAQ
User's Guide
API Reference
Acronyms

Contributing
Snapshots
Source in Git
Cygwin Packages

Related Sites

Cygwin

Get that *Linux* feeling - on Windows

Installing and Updating Cygwin Packages

Installing and Updating Cygwin for 64-bit versions of Windows

Run [setup-x86_64.exe](#) any time you want to update or install a Cygwin package for 64-bit windows. The [signature](#) for [setup-x86_64.exe](#) can be used to verify the validity of this binary.

Installing and Updating Cygwin for 32-bit versions of Windows

Run [setup-x86.exe](#) any time you want to Select an appropriate version to download based on your OS of this binary.

Signing key transition

The key used to sign setup binaries has been updated. During the transition period, signatures are made using both old (676041BA) and new (1A698DE9E2E56300) public keys [here](#). See [this mail](#) for more details.

General installation notes

When installing packages for the first time, the setup program *does not install every package*. Only the **minimal base packages** from the Cygwin distribution are installed by default, which takes up about 100 MB.

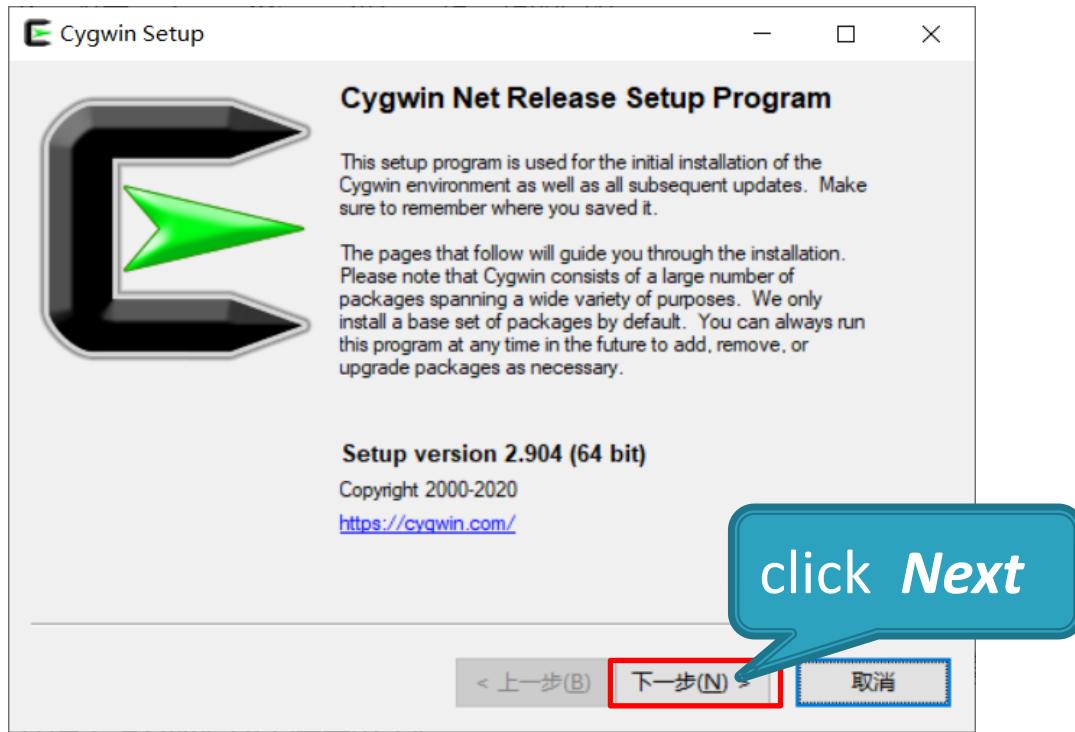
Clicking on categories and packages in the setup program package installation screen allows you to select what is installed or updated.

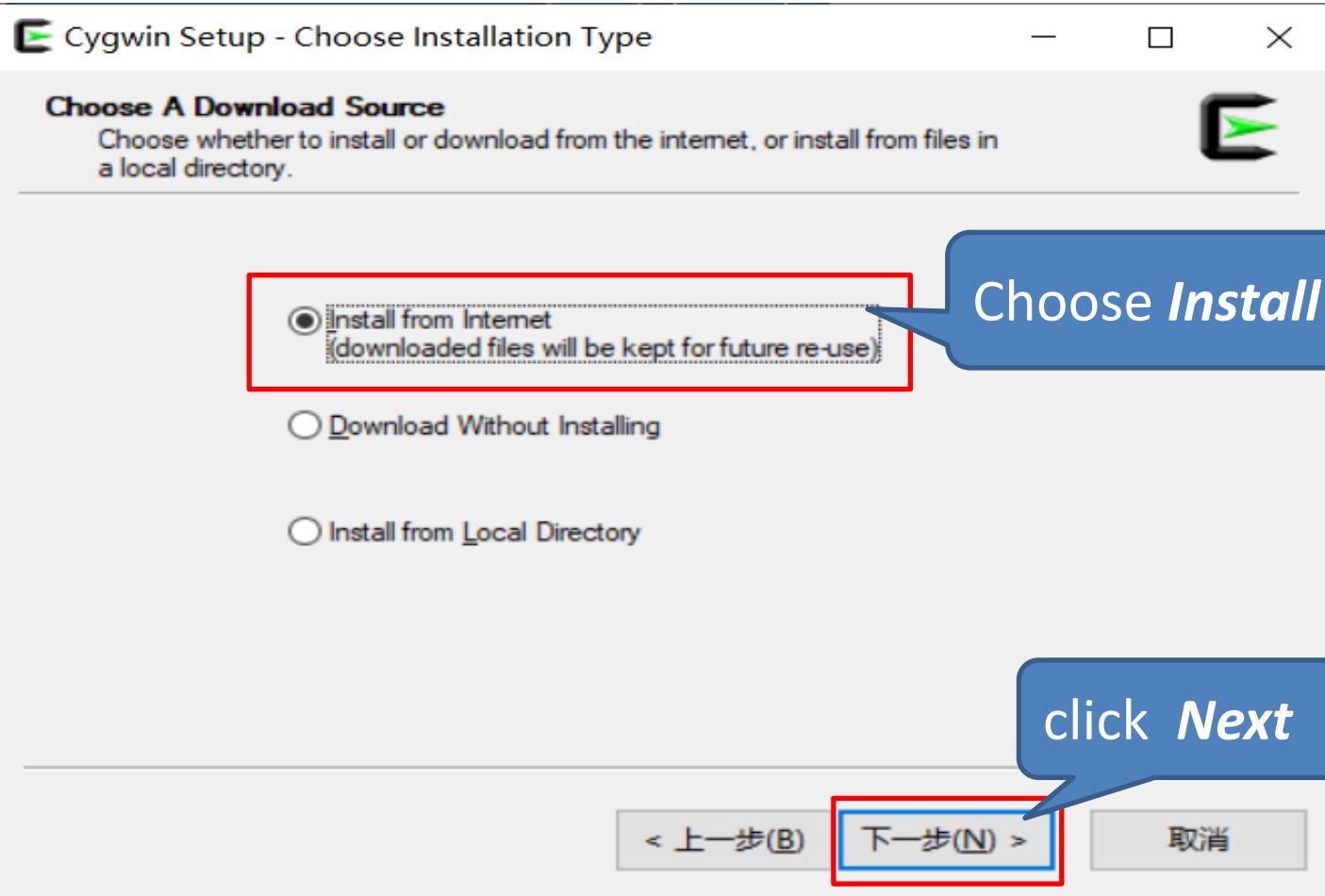
Individual packages like *bash*, *gcc*, *less*, etc. are released independently of the Cygwin DLL, so the Cygwin DLL version is not useful as a general Cygwin release number. The setup program tracks the versions of all installed components and provides the mechanism for **installing** or **updating** everything available from this site for Cygwin.

Once you've installed your desired subset of the Cygwin distribution, the setup program will remember what you selected, so re-running it will update your system with any new package releases.

On Windows Vista and later, the setup program will check by default if it runs with administrative privileges and, if not, will try to elevate the process. If you want to avoid this behaviour and install under an unprivileged account just for your own usage, run setup with the `--no-admin` option.

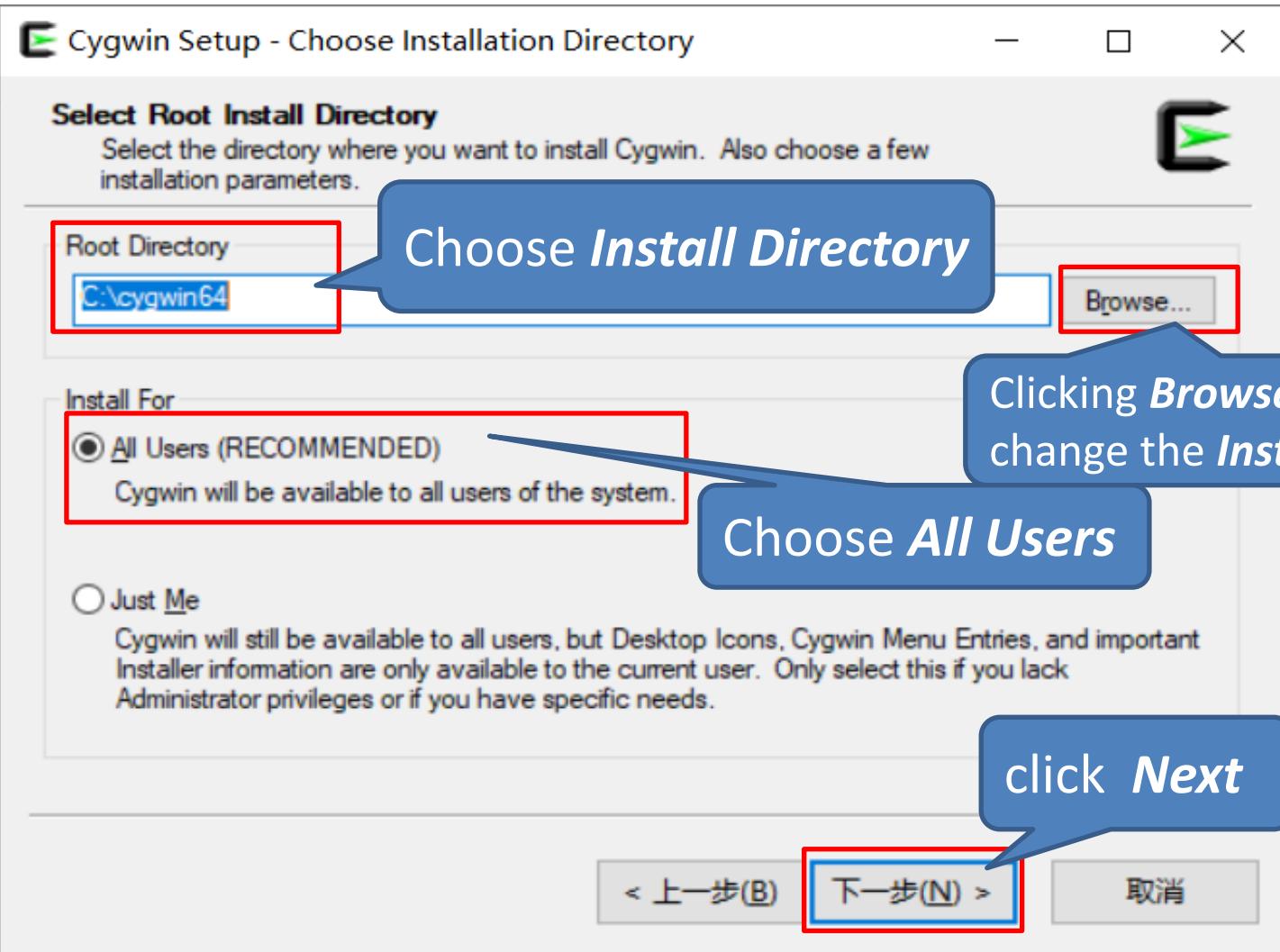
1. When the download is completed, you can **double-click** the *setup-x86_64.exe* to **run the installer**.





Choose *Install from Internet*

click *Next*



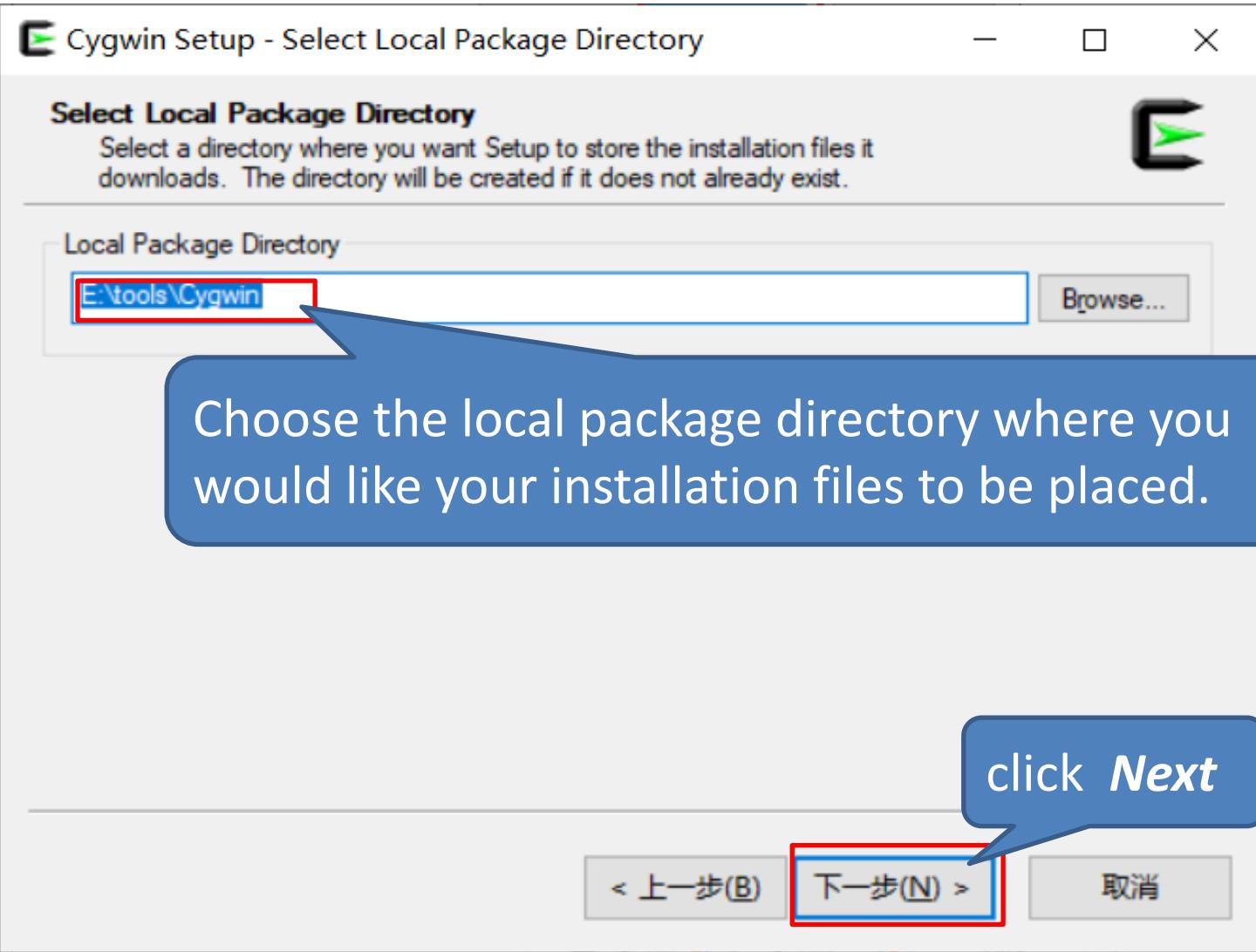
Choose *Install Directory*

Browse...

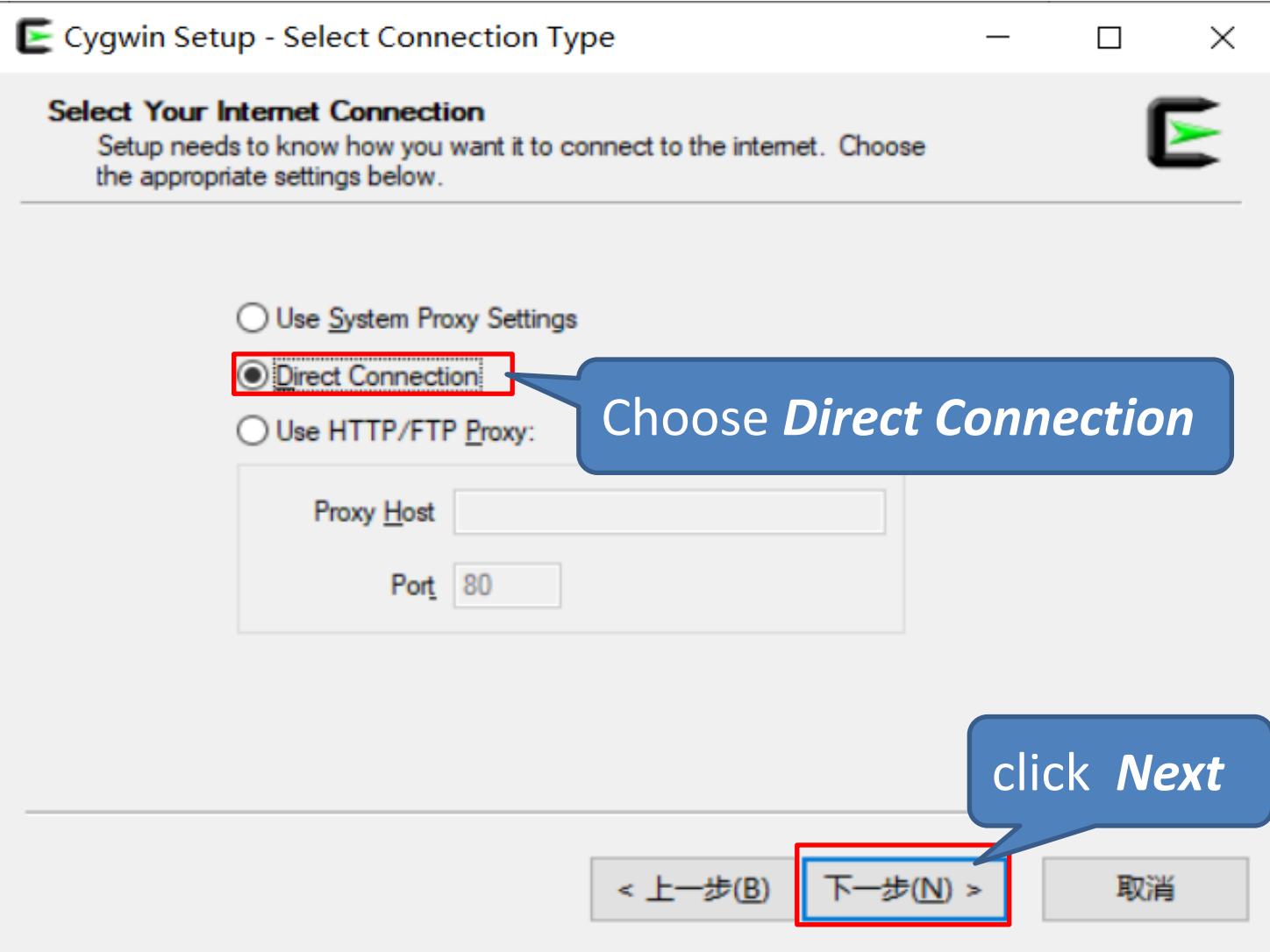
Clicking *Browse* button can
change the *Install Directory*

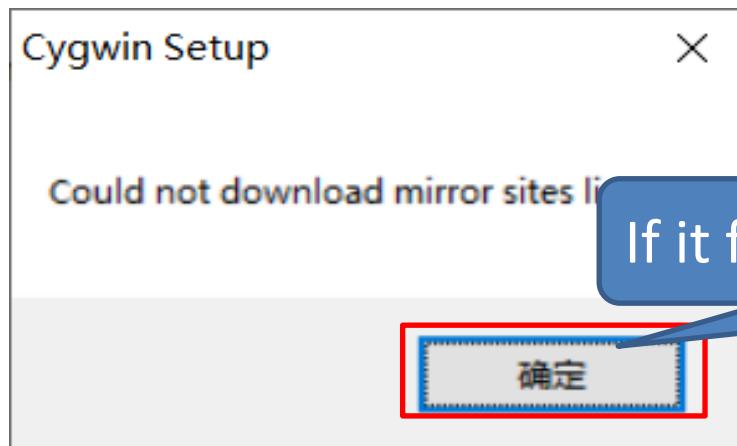
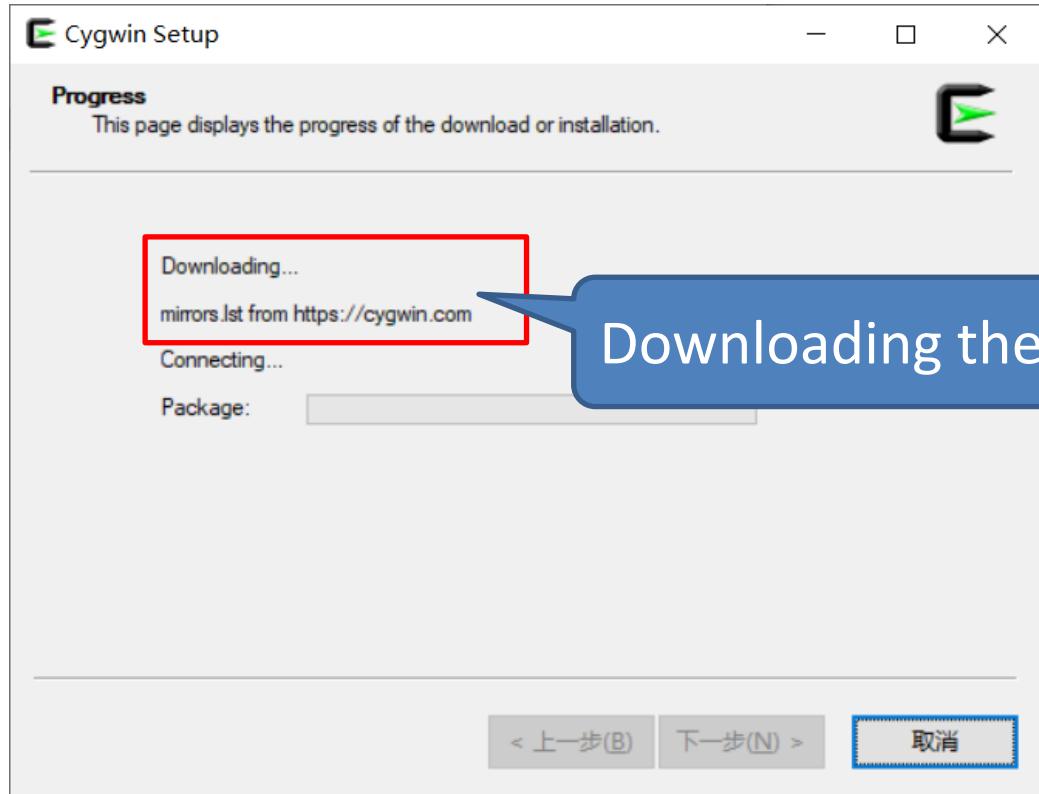
Choose *All Users*

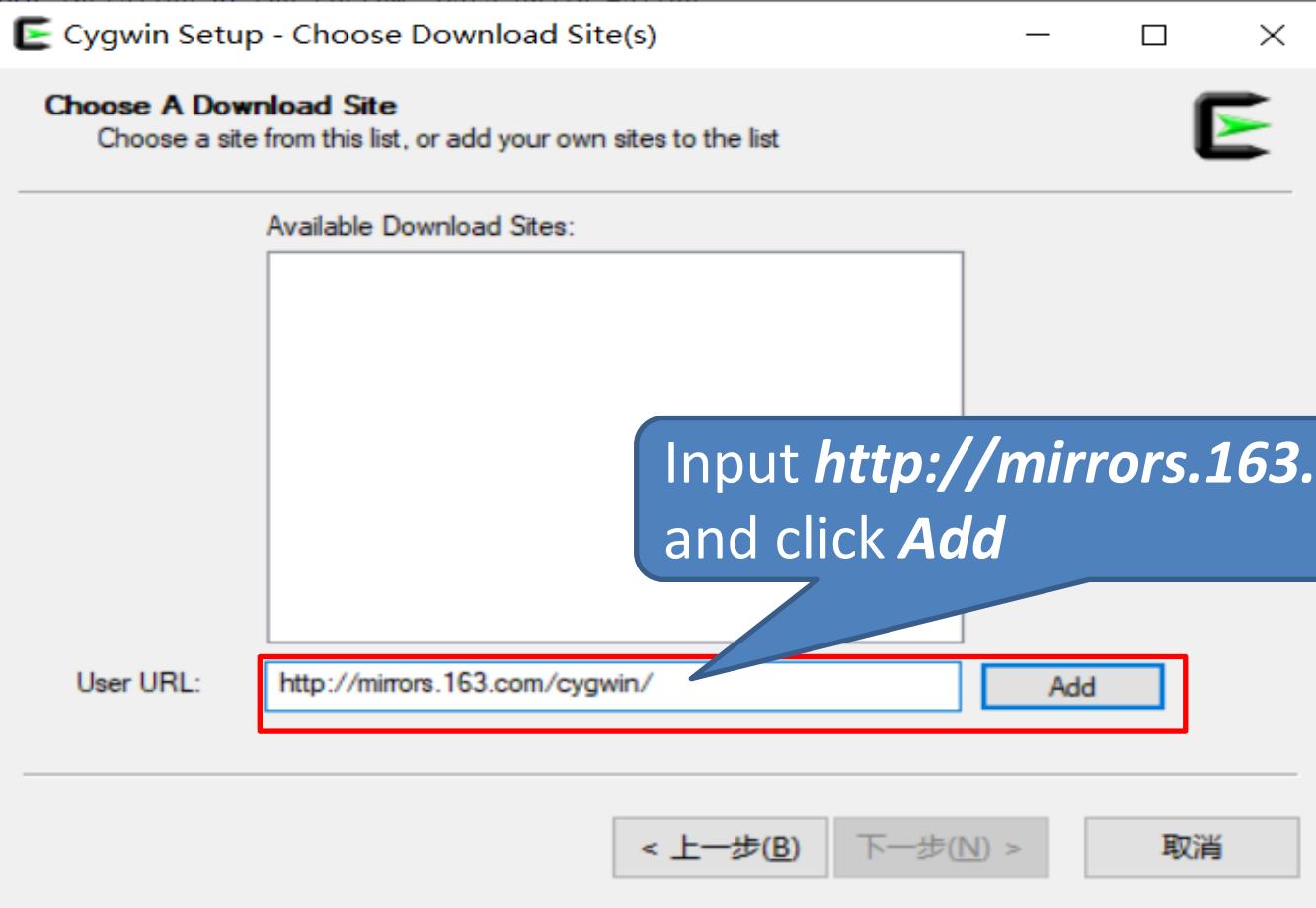
click *Next*



click ***Next***







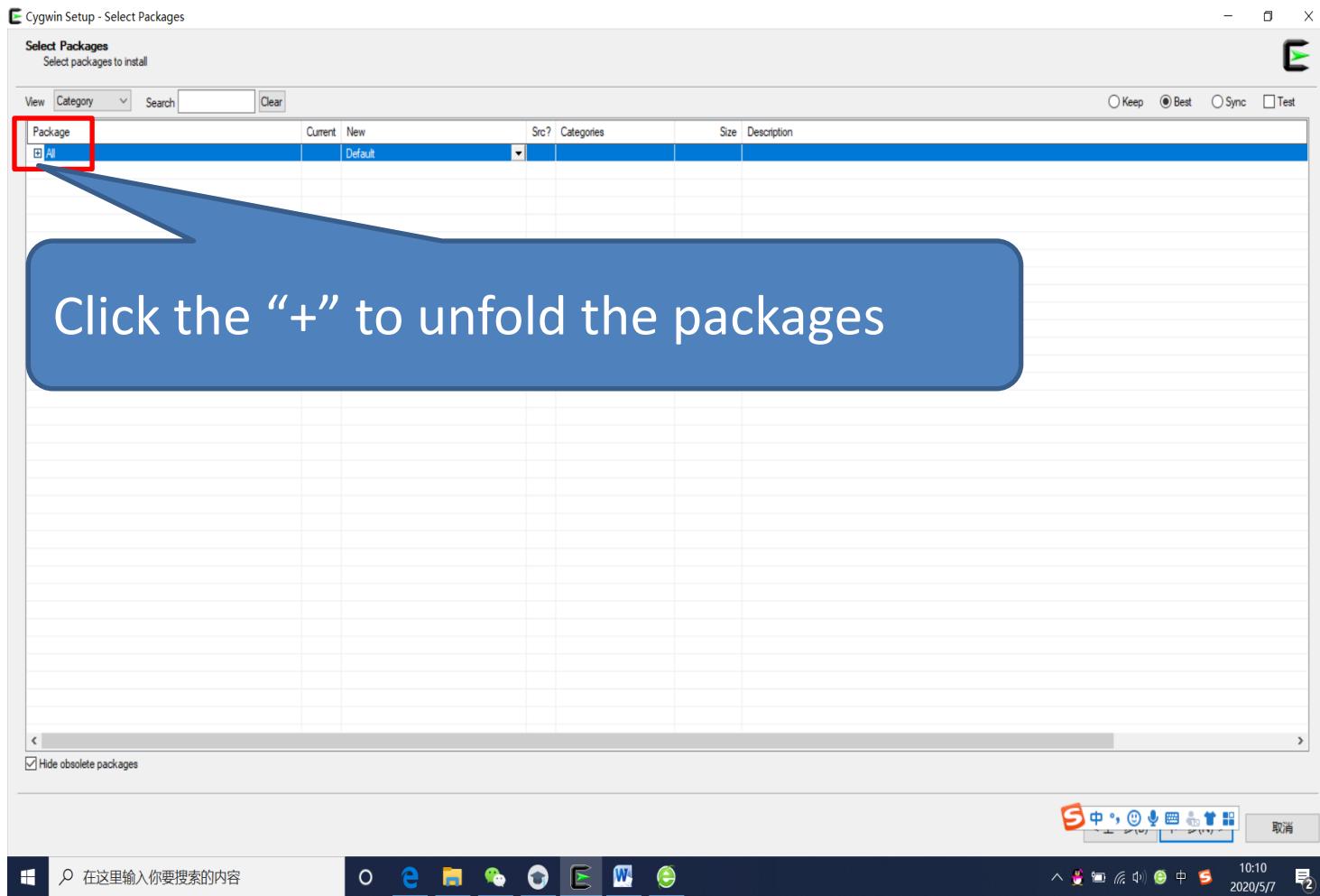
Input *http://mirrors.163.com/cygwin/*
and click **Add**



click *Next*

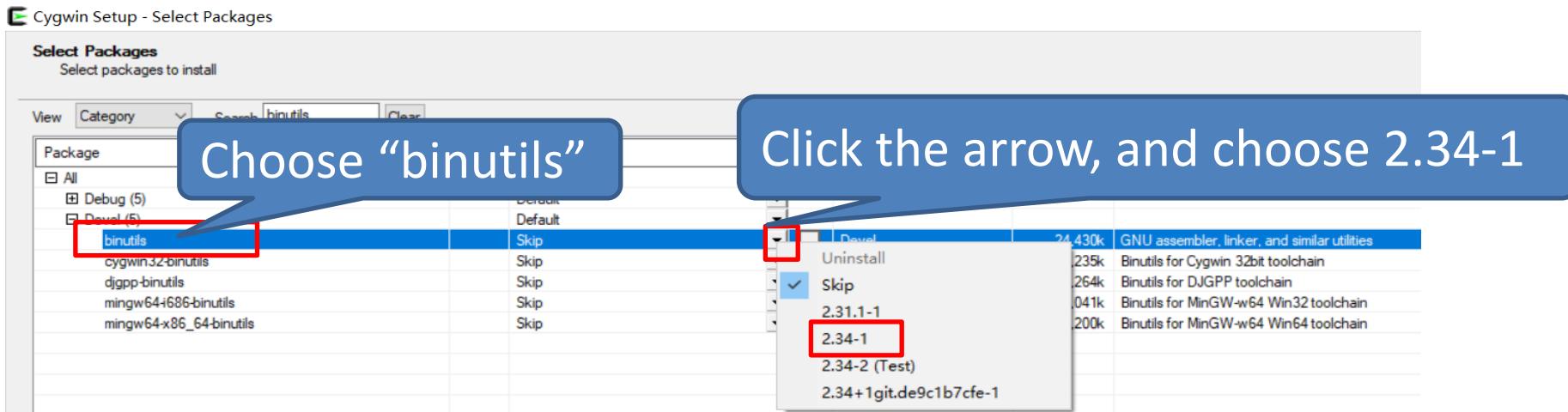
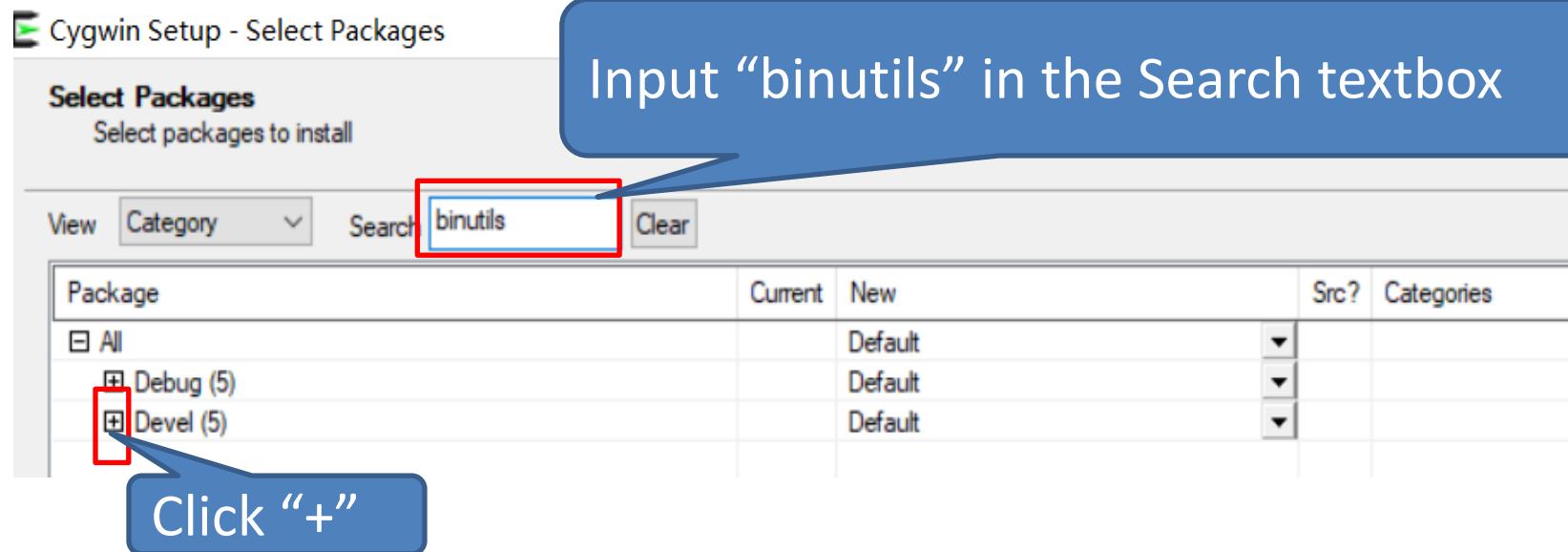
Select the packages you want to install

Packages you may want to install include: **binutils, gcc, mingw, gdb, make and cmake.**



Select the packages you want to install

Packages you may want to install include: binutils, gcc, mingw, gdb, make and cmake.



Cygwin Setup - Select Packages

Package	Current	New	Src?	Categories	Size	Description
All	Default					
Debug (5)	Default					
Devel (5)	Default					
binutils	2.34-1		<input checked="" type="checkbox"/>	Devel	64.10k	GNU assembler, linker, and similar utilities
cygwin32-binutils	Skip		<input type="checkbox"/>	Devel	3.235k	Binutils for Cygwin 32bit toolchain
djgpp-binutils	Skip		<input type="checkbox"/>	Devel	1.264k	Binutils for DJGPP toolchain
mingw64-686-binutils	Skip		<input type="checkbox"/>	Devel	2.041k	Binutils for MinGW-w64 Win32 toolchain
mingw64-x86_64-binutils	Skip		<input type="checkbox"/>	Devel	2.200k	Binutils for MinGW-w64 Win64 toolchain

binutils

Cygwin Setup - Select Packages

Package	Current	New	Src?	Categories	Size	Description
All	Default					
Debug (5)	Default					
Devel (28)	Default					
colorgcc	Skip		<input type="checkbox"/>	Devel, Perf	14k	Colorizer for GCC warning/error messages
cygwin32-gcc-core	6.4.0-1		<input checked="" type="checkbox"/>	Devel	90.85k	GCC for Cygwin 32bit toolchain (C, OpenMP)
cygwin32-gcc-fortran	Skip		<input type="checkbox"/>	Devel	7.014k	GCC for Cygwin 32bit toolchain (Fortran)
cygwin32-gcc-g++	6.4.0-1		<input checked="" type="checkbox"/>	Devel	84.845k	GCC for Cygwin 32bit toolchain (C++)
djgpp-gcc-ada	Skip		<input type="checkbox"/>	Devel	13.501k	GCC for DJGPP toolchain (Ada)
djgpp-gcc-core	Skip		<input type="checkbox"/>	Devel	7.926k	GCC for DJGPP toolchain (C)
djgpp-gcc-fortran	Skip		<input type="checkbox"/>	Devel	6.296k	GCC for DJGPP toolchain (Fortran)
djgpp-gcc-g++	Skip		<input type="checkbox"/>	Devel	8.279k	GCC for DJGPP toolchain (C++)
djgpp-gcc-objc	Skip		<input type="checkbox"/>	Devel	5.087k	GCC for DJGPP toolchain (Objective-C)
djgpp-gcc-objc++	Skip		<input type="checkbox"/>	Devel	5.349k	GCC for DJGPP toolchain (Objective-C++)
gcc-core	7.4.0-1		<input checked="" type="checkbox"/>	Devel	79.856k	GNU Compiler Collection (C, OpenMP)
gcc-fortran	Skip		<input type="checkbox"/>	Devel	9.333k	GNU Compiler Collection (Fortran)
gcc-a++	7.4.0-1		<input checked="" type="checkbox"/>	Devel	73.275k	GNU Compiler Collection (C++)
gcc-objc	Skip		<input type="checkbox"/>	Devel	6.755k	GNU Compiler Collection (Objective-C)
gcc-objc++	Skip		<input type="checkbox"/>	Devel	7.037k	GNU Compiler Collection (Objective-C++)
gcc-tools-epoch1-autounf	Skip		<input type="checkbox"/>	Devel	425k	(gcc-special) automatic configure script builder
gcc-tools-epoch1-automake	Skip		<input type="checkbox"/>	Devel	419k	(gcc-special) a tool for generating GNU-compliant Makefiles
gcc-tools-epoch2-autounf	Skip		<input type="checkbox"/>	Devel	712k	(gcc-special) automatic configure script builder
gcc-tools-epoch2-automake	Skip		<input type="checkbox"/>	Devel	589k	(gcc-special) a tool for generating GNU-compliant Makefiles
gcmakedep	Skip		<input type="checkbox"/>	Devel	9k	X Makefile dependency tool for GCC
mingw64-686-gcc-core	7.4.0-1		<input checked="" type="checkbox"/>	Devel	76.174k	GCC for Win32 (686-w64-mingw32) toolchain (C, OpenMP)
mingw64-686-gcc-fortran	Skip		<input type="checkbox"/>	Devel	9.083k	GCC for Win32 (686-w64-mingw32) toolchain (Fortran)
mingw64-686-gcc-g++	7.4.0-1		<input checked="" type="checkbox"/>	Devel	72.106k	GCC for Win32 (686-w64-mingw32) toolchain (C++)
mingw64-686-gcc-objc	Skip		<input type="checkbox"/>	Devel	13.619k	GCC for Win32 (686-w64-mingw32) toolchain (Objective-C,C++)
mingw64-x86_64-gcc-core	7.4.0-1		<input checked="" type="checkbox"/>	Devel	76.666k	GCC for Win64 toolchain (C, OpenMP)
mingw64-x86_64-gcc-fortran	Skip		<input type="checkbox"/>	Devel	9.436k	GCC for Win64 toolchain (Fortran)
mingw64-x86_64-gcc-g++	7.4.0-1		<input checked="" type="checkbox"/>	Devel	72.489k	GCC for Win64 toolchain (C++)
mingw64-x86_64-gcc-objc	Skip		<input type="checkbox"/>	Devel	13.806k	GCC for Win64 toolchain (Objective-C,C++)
LIB (2)	Default					
Perl (1)	Default					

GCC

mingw

Hide obsolete packages



10:45
2020/5/7

Cygwin Setup - Select Packages

Select Packages
Select packages to install

View Category Clear

Keep Best Sync Test

Package	Current	New	Src?	Categories	Size	Description
All	Default					
Database (1)	Default					
Debug (3)	Default					
Devel (4)	Default					
cgdb	Skip			Devel	71k	A curses-based interface to the GNU Debugger (GDB)
gdb	8.1.1-1			Devel	22,502k	The GNU Debugger
mingw64-gdb	Skip			Devel	56k	GNU dbm implementation for Win32 toolchain
mingw64-x86_64-gdb	Skip			Devel	57k	GNU dbm implementation for Win64 toolchain
Libs (4)	Default					

gdb

Cygwin Setup - Select Packages

Select Packages
Select packages to install

View Full Clear

Keep Best Sync Test

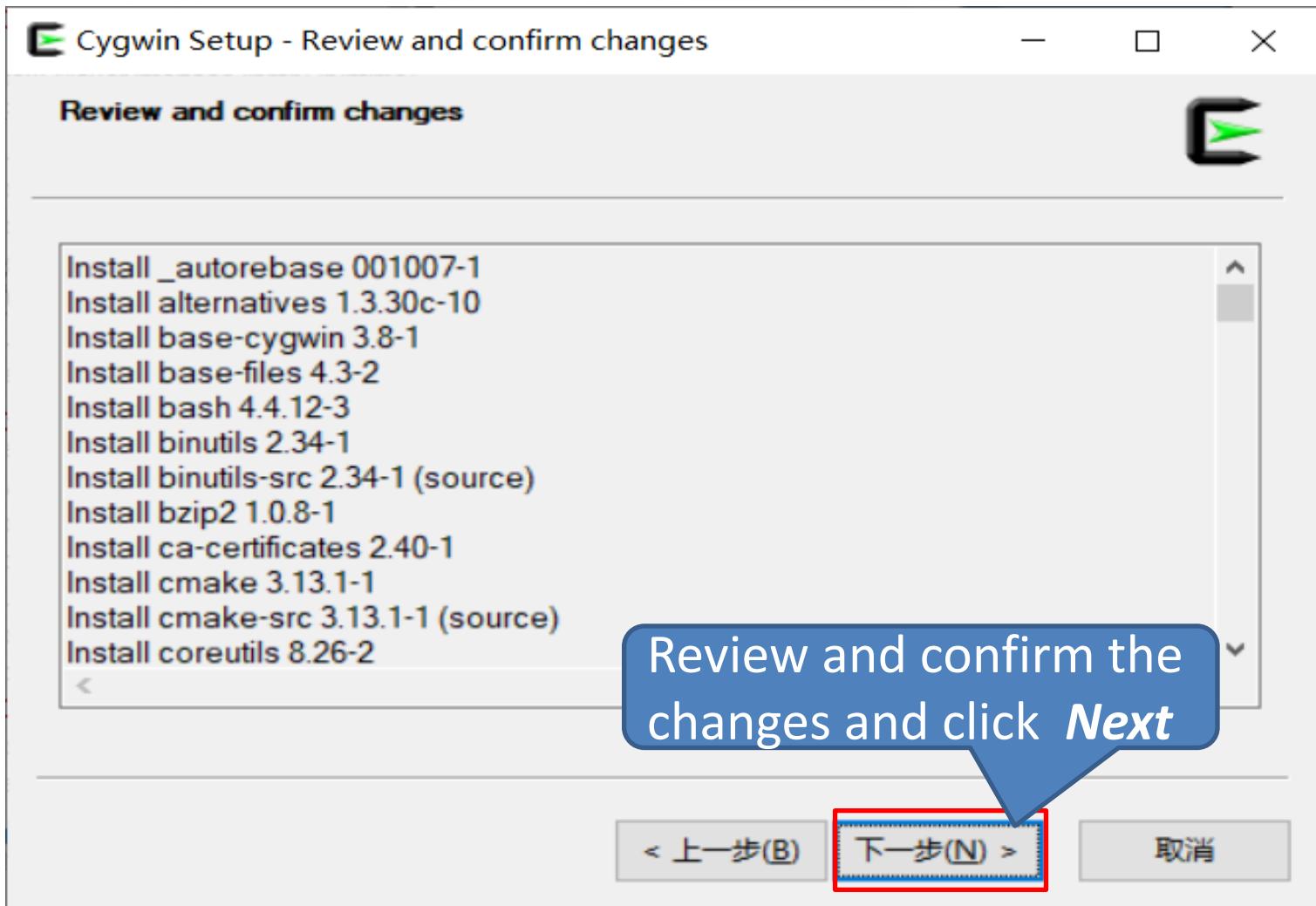
Package	Current	New	Src?	Categories	Size	Description
automake-1.4	Skip			Devel	248k	(1.4) a tool for generating GNU-compliant Makefiles
automake-1.5	Skip			Devel	332k	(1.5) a tool for generating GNU-compliant Makefiles
automake-1.6	Skip			Devel	365k	(1.6) a tool for generating GNU-compliant Makefiles
automake-1.7	Skip			Devel	426k	(1.7) a tool for generating GNU-compliant Makefiles
automake-1.8	Skip			Devel	499k	(1.8) a tool for generating GNU-compliant Makefiles
automake-1.9	Skip			Devel	557k	(1.9) a tool for generating GNU-compliant Makefiles
cmake	3.13.1-1			Devel	12,953k	Cross-platform makefile generation system
cmake-debuginfo	Skip			Debug	223,140k	Debug info for cmake
cmake-doc	Skip			Devel	1,351k	Cross-platform makefile generation system (documentation)
cmake-gui	Skip			Devel	1,502k	Cross-platform makefile generation system (GUI)
emacs-cmake	Skip			Editors	4k	Cross-platform makefile generation system (Emacs mode)
extra-cmake-modules	Skip			Devel	281k	Extra CMake Modules for KDE
gcc-tools-epoch1-automake	Skip			Devel	419k	(gcc-special) a tool for generating GNU-compliant Makefiles
gcc-tools-epoch2-automake	Skip			Devel	589k	(gcc-special) a tool for generating GNU-compliant Makefiles
gcmakedep	Skip			Devel	6k	X Makelife dependency tool for GCC
imake	Skip			Devel	35k	X Imake legacy build system
imake-debuginfo	Skip			Debug	64k	Debug info for imake
libWMaker-devel	Skip			Libs	3k	Window Maker interface library (development)
libWMaker1	Skip			Libs	4k	Window Maker interface library (runtime)
libpagemakertools	Skip			Graphics	5k	MS Publisher file converters
libpagemaker0.0-debuginfo	Skip			Debug	591k	Debug info for libpagemaker0.0
libpagemaker0.0-devel	Skip			Libs	2k	Adobe PageMaker import filter library (development)
libpagemaker0.0-docs	Skip			Libs	105k	Adobe PageMaker import filter library (API documentation)
libpagemaker0.0_0	Skip			Libs	65k	Adobe PageMaker import filter library (runtime)
make	4.2.1-2			Devel	1,822k	The GNU version of the 'make' utility
make-debuginfo	Skip			Debug	441k	Debug info for make
makedepend	Skip			Devel	29k	X Makefile dependency tool
makedepend-debuginfo	Skip			Debug	75k	Debug info for makedepend
makepasswd	Skip			Perl_Utils	14k	Generate and encrypt passwords
makeself	Skip			Archive	26k	Utility to generate self-extractable archives
mingw64-gdb	Skip			Libs	76k	Adobe PageMaker import filter library for Win32 toolchain
mingw64-gdb-q4-make	Skip			Devel	7,326k	Qt4 development tools for Win32 toolchain
mingw64-gdb-q4-gmake-debuginfo	Skip			Debug	79,179k	Debug info for mingw64-gdb-q4-gmake

click Next

下一步 > 取消

在这里输入你要搜索的内容

11:07 2020/5/7



Review and confirm the
changes and click **Next**

Cygwin Setup - Installation Status and Create Icons

Create Icons

Tell setup if you want it to create a few icons for convenient access to the Cygwin environment.



Create icon on Desktop

Add icon to Start Menu

Installation Status

Installation Complete

Keep the default chooses
and click **Finish**

< 上一步(B)

完成

取消

3.2 Verify Cygwin

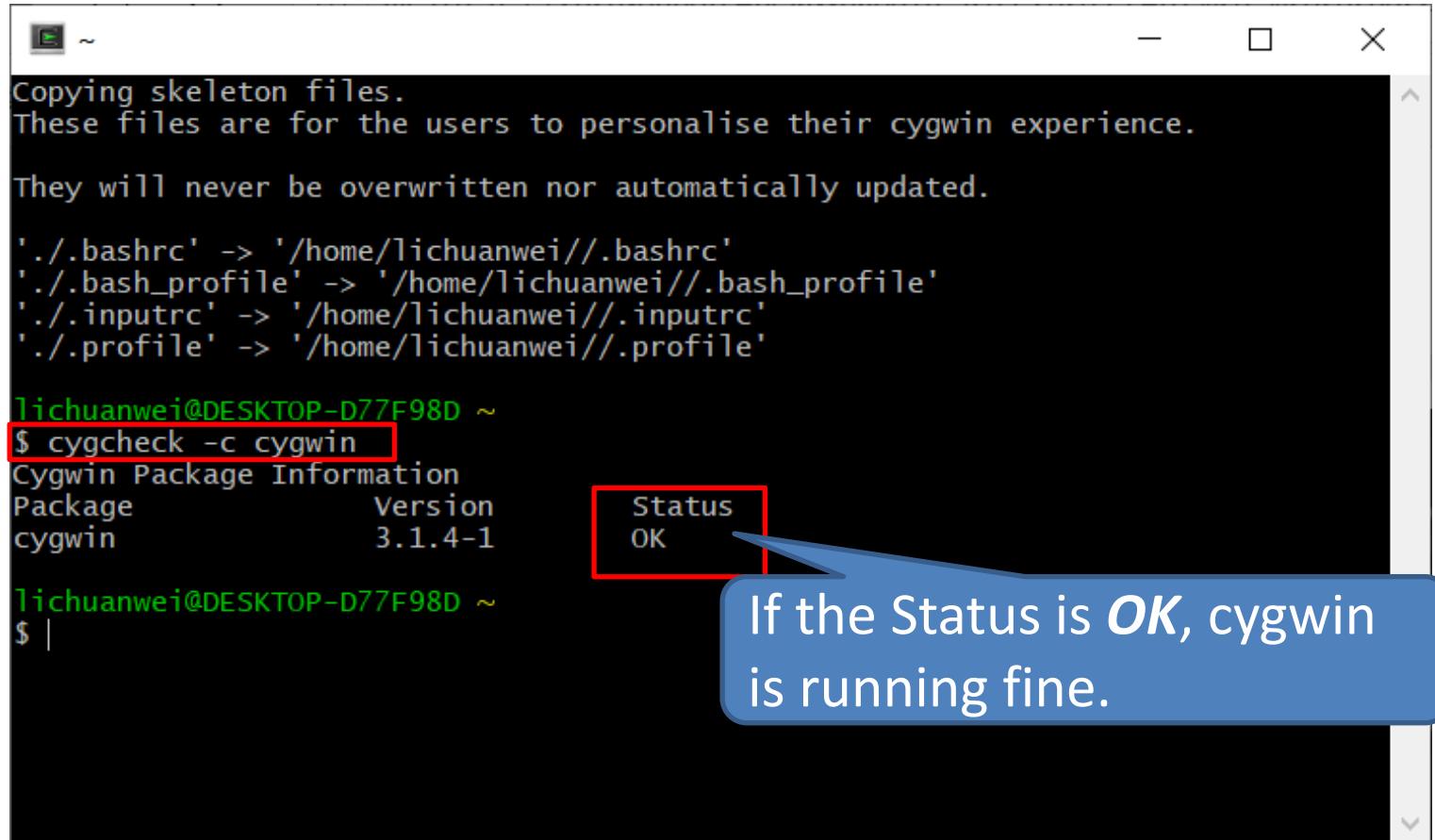
1. Double click the Cygwin icon on the desktop to start Cygwin.



The screenshot shows a terminal window titled 'Cygwin64 Terminal'. The window contains the following text output:

```
Copying skeleton files.  
These files are for the users to personalise their cygwin experience.  
They will never be overwritten nor automatically updated.  
./.bashrc' -> '/home/lichuanwei//.bashrc'  
./.bash_profile' -> '/home/lichuanwei//.bash_profile'  
./.inputrc' -> '/home/lichuanwei//.inputrc'  
./.profile' -> '/home/lichuanwei//.profile'  
  
lichuanwei@DESKTOP-D77F98D ~  
$
```

2. Input “**cygcheck –c cygwin**” after \$ prompt.



```
Copying skeleton files.  
These files are for the users to personalise their cygwin experience.  
They will never be overwritten nor automatically updated.  
./.bashrc' -> '/home/lichuanwei//.bashrc'  
./.bash_profile' -> '/home/lichuanwei//.bash_profile'  
./.inputrc' -> '/home/lichuanwei//.inputrc'  
./.profile' -> '/home/lichuanwei//.profile'  
  
lichuanwei@DESKTOP-D77F98D ~  
$ cygcheck -c cygwin  
Cygwin Package Information  
Package          Version  
cygwin          3.1.4-1  
  
lichuanwei@DESKTOP-D77F98D ~  
$ |
```

The terminal window shows the output of the `cygcheck -c cygwin` command. The status is highlighted with a red box and a callout bubble pointing to it, stating "If the Status is *OK*, cygwin is running fine."

Status
OK

If the Status is *OK*, cygwin is running fine.

3. Verify the GCC installation by listing the version of gcc, g++, gdb, make and cmake.

```
lizhuanwei@DESKTOP-D77F98D ~
$ gcc --version
gcc (GCC) 7.4.0
Copyright © 2017 Free Software Foundation, Inc.
本程序是自由软件：请参看源代码的版权声明。本软件没有任何担保：
包括没有适销性和某一专用目的下的适用性担保。

lizhuanwei@DESKTOP-D77F98D ~
$ g++ --version
g++ (GCC) 7.4.0
Copyright © 2017 Free Software Foundation, Inc.
本程序是自由软件：请参看源代码的版权声明。本软件没有任何担保：
包括没有适销性和某一专用目的下的适用性担保。

lizhuanwei@DESKTOP-D77F98D ~
$ gdb --version
GNU gdb (GDB) (Cygwin 8.1.1-1) 8.1.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-pc-cygwin".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".

lizhuanwei@DESKTOP-D77F98D ~
$ make --version
GNU Make 4.2.1
为 x86_64-unknown-cygwin 编译
Copyright (C) 1988-2016 Free Software Foundation, Inc.
许可证：GPLv3+，GNU 通用公共许可证第 3 版或更新版本<http://gnu.org/licenses/gpl.html>。
本软件是自由软件：您可以自由修改和重新发布它。
在法律允许的范围内没有其他保证。

lizhuanwei@DESKTOP-D77F98D ~
$ cmake --version
cmake version 3.13.1

CMake suite maintained and supported by Kitware (kitware.com/cmake).

lizhuanwei@DESKTOP-D77F98D ~
$
```

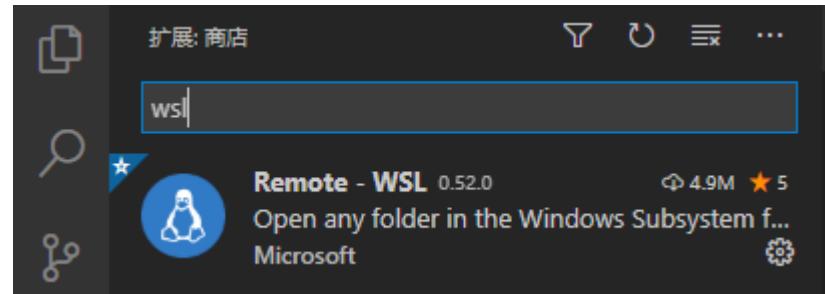
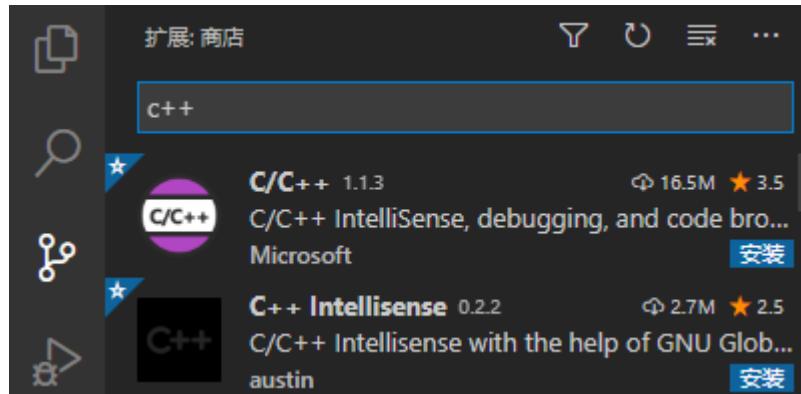
4 Download and Install Editor

To install **VSCode**, you can visit: <https://code.visualstudio.com/> to download the package for your platform (Linux, Windows or macOS).

After you install VSCode, we need to install two plugins at least to support our development:

1. C/C++ plugin
2. Remote - WSL plugin

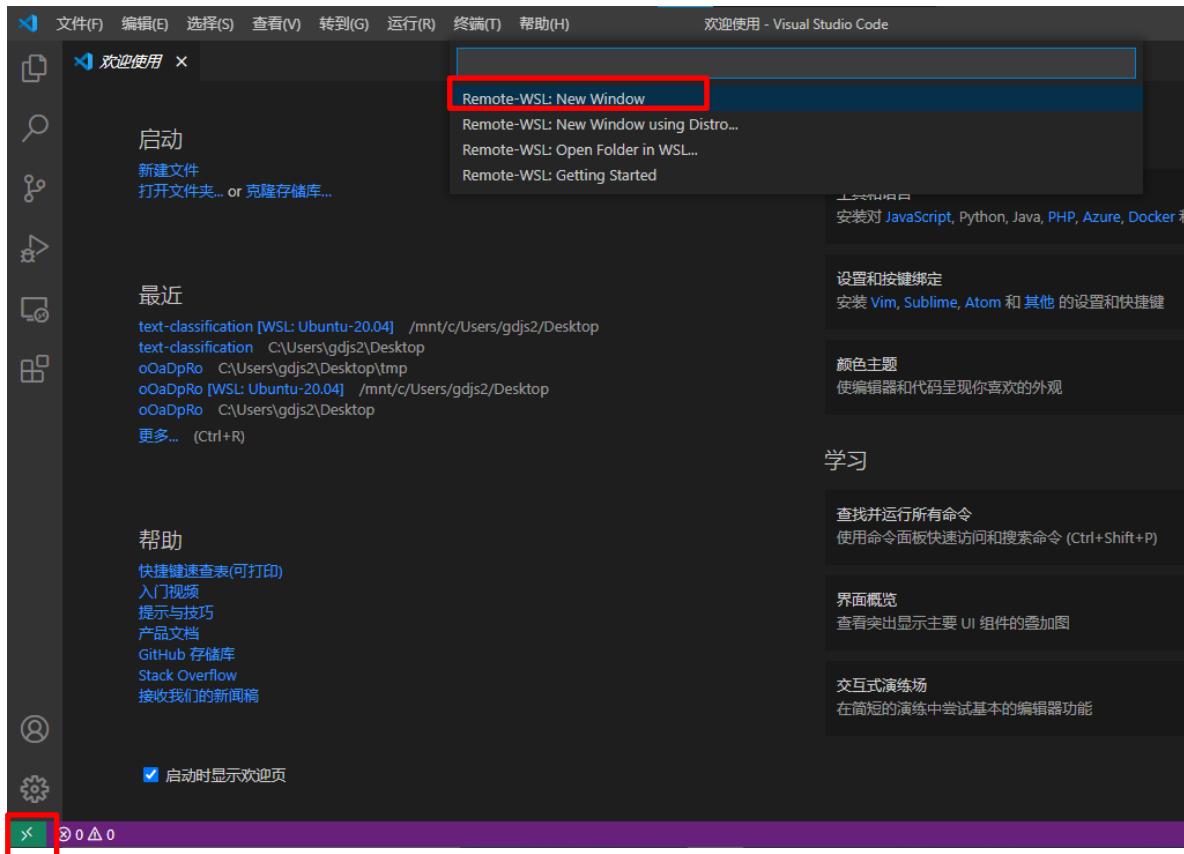
To install them, you can search **c++** and **wsl** respectively.



4. Download and install editor (Cont.)

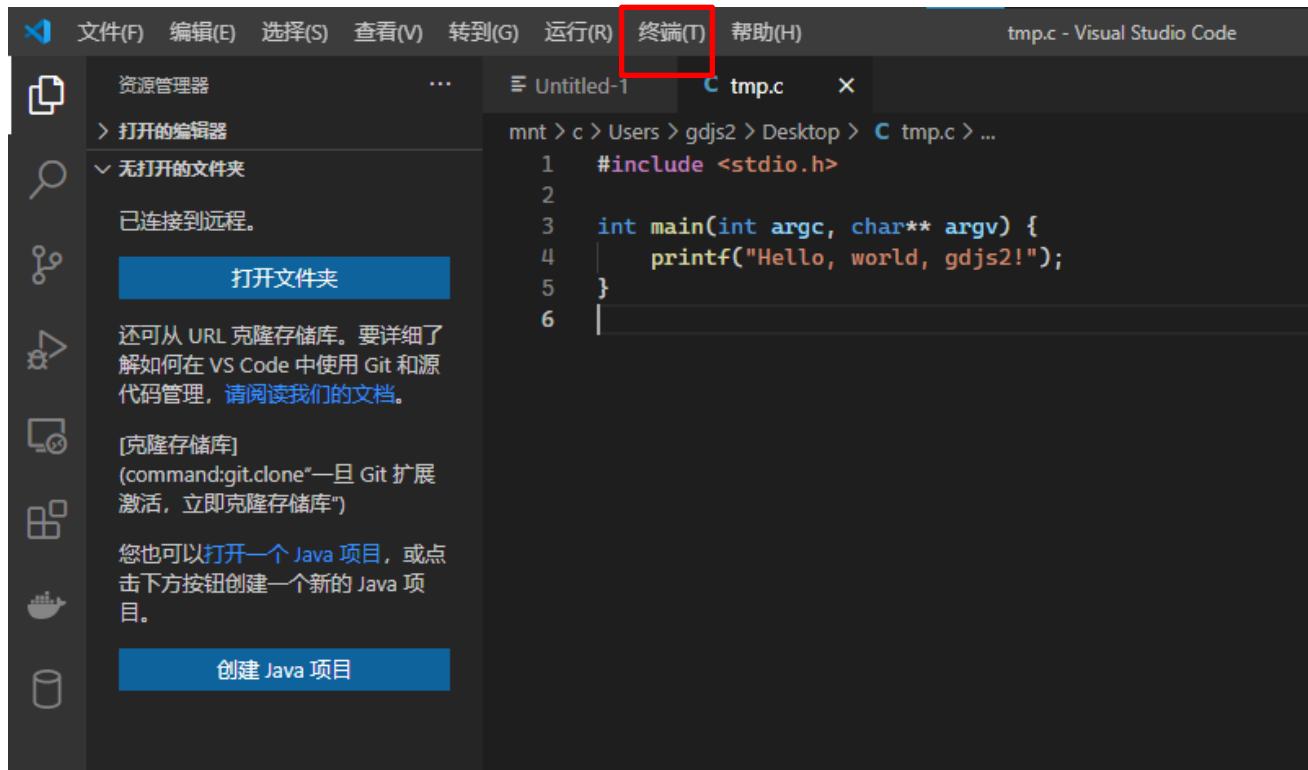
Now, you need to switch your VSCode to WSL system.

Click the green button on the bottom. And choose Remote WSL: New window (or others you need).



4. Download and install editor (Cont.)

Now your VSCode is working on the WSL system. We can create our first C program. We strongly recommend you to find more functions of VSCode, OPEN TERMINAL etc.,



Note: The suffix of the program must be **.c** or **.cpp** for c program and c++ program respectively.

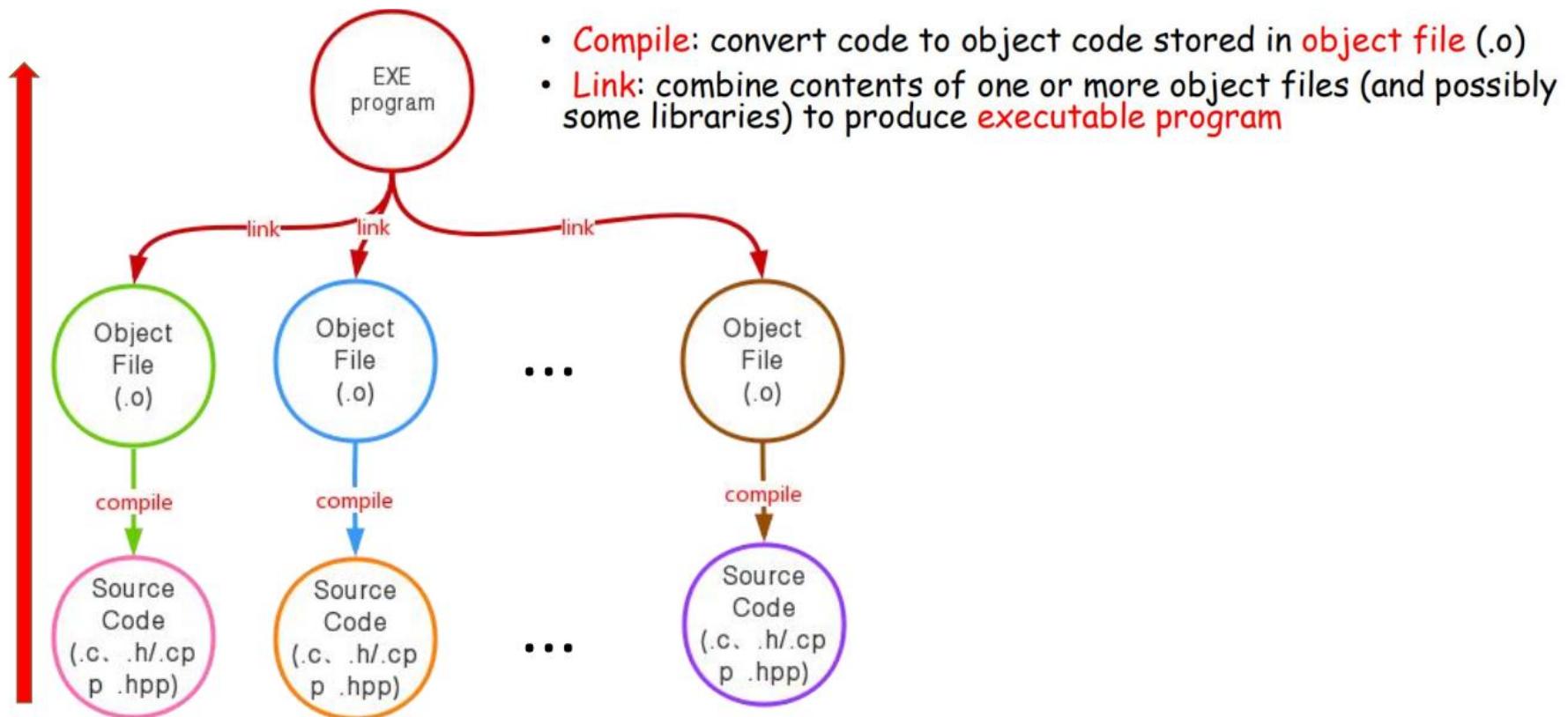
5 Compile, Link and Run C/C++ Programs

5.1 Learn the program compilation process

5.2 Compile ,Link and Run C Programs

5.3 Compile ,Link and Run C++ Programs

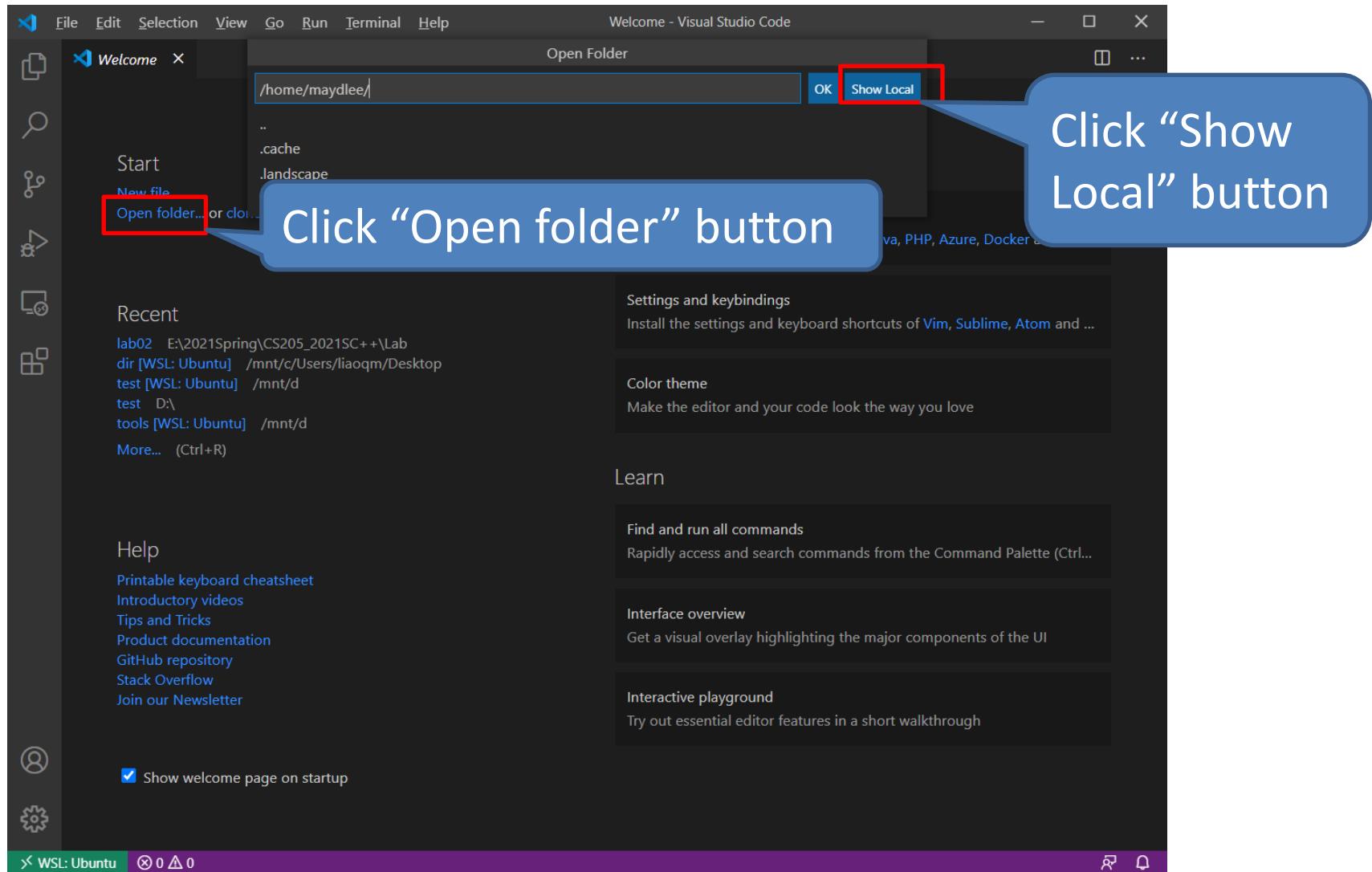
5.1 The program compilation process



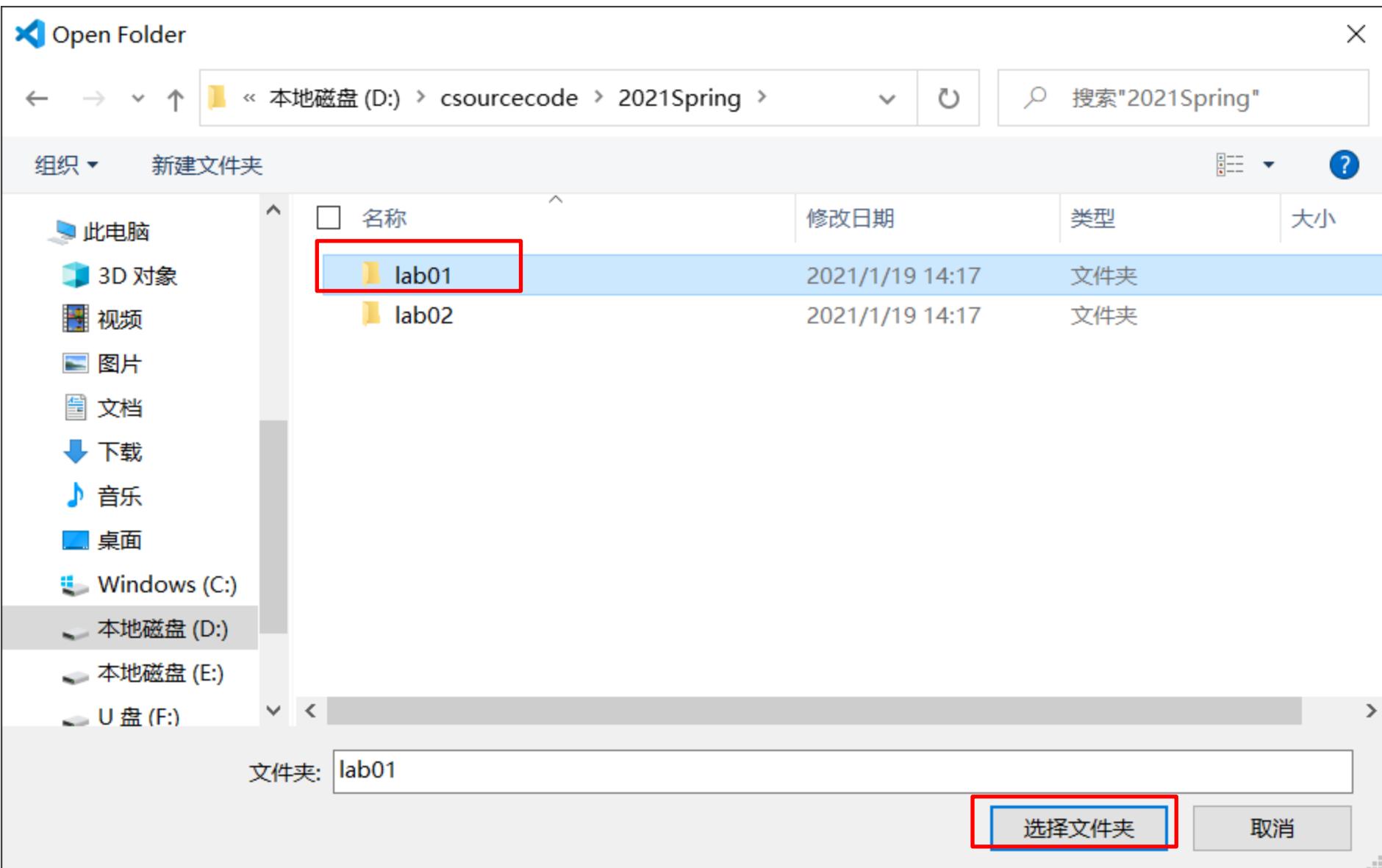
5.2 Compile, Link and Run C programs

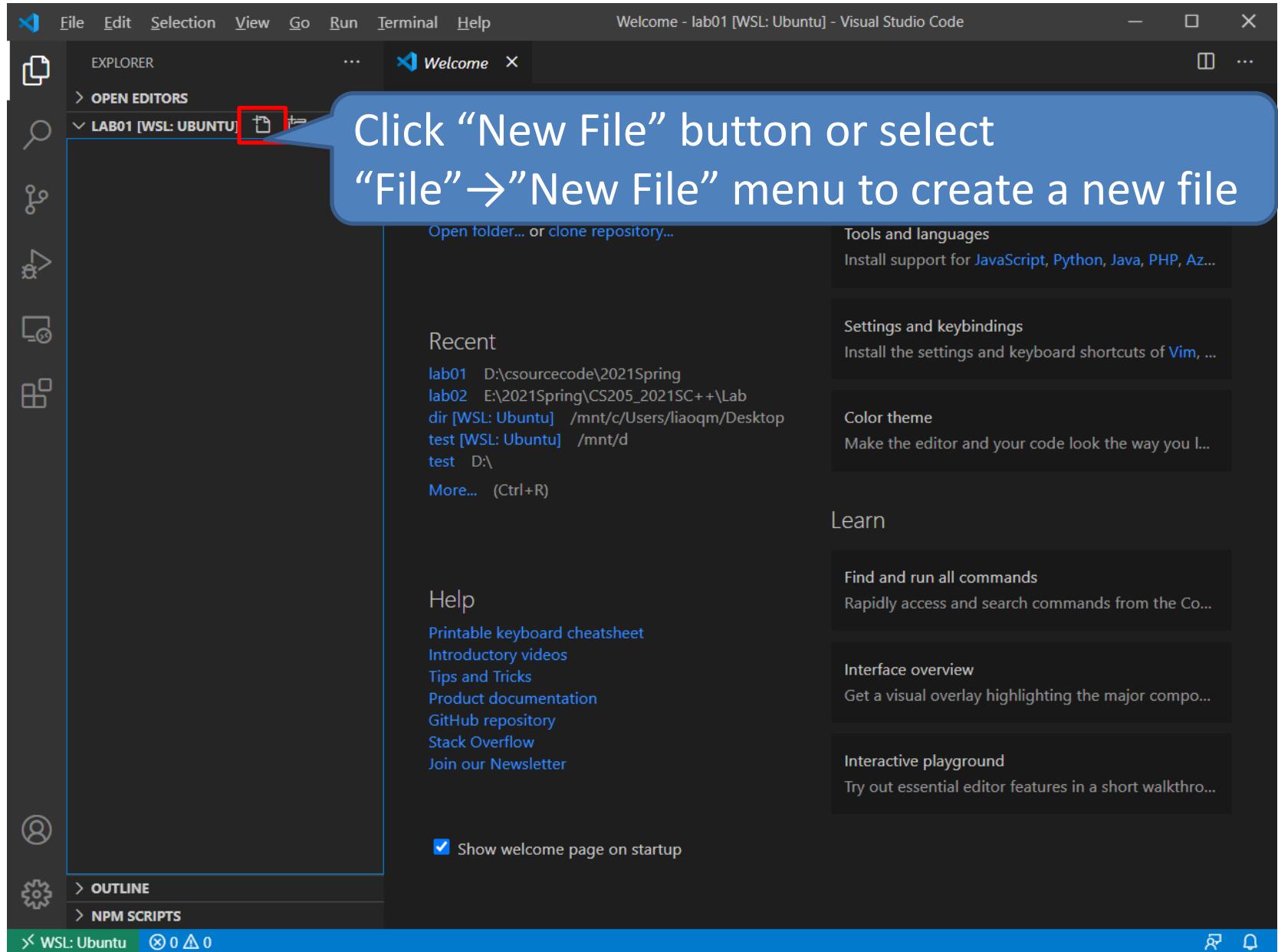
Compile/Link/Run a simple C program – hello.c

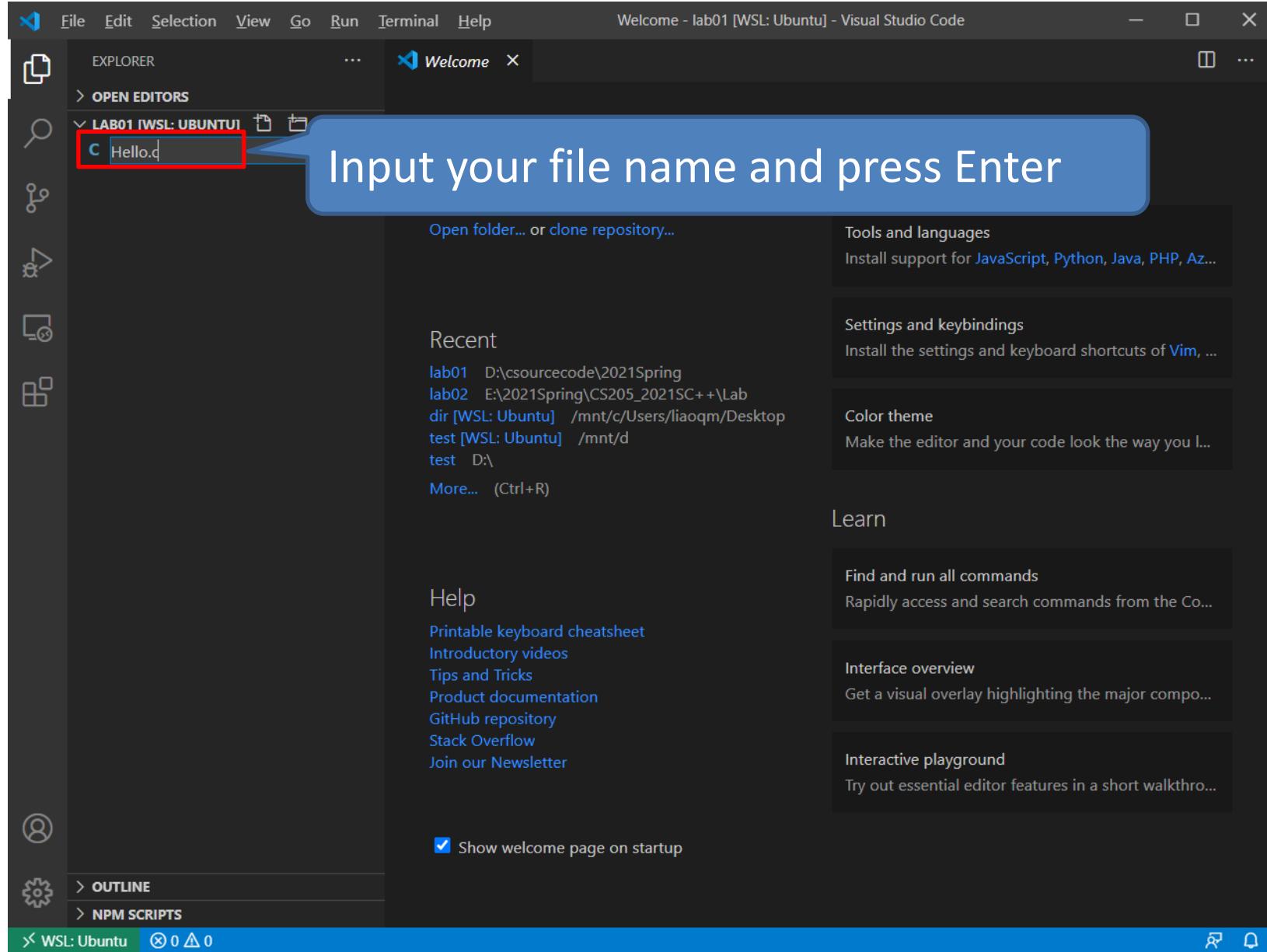
Write your codes in VScode



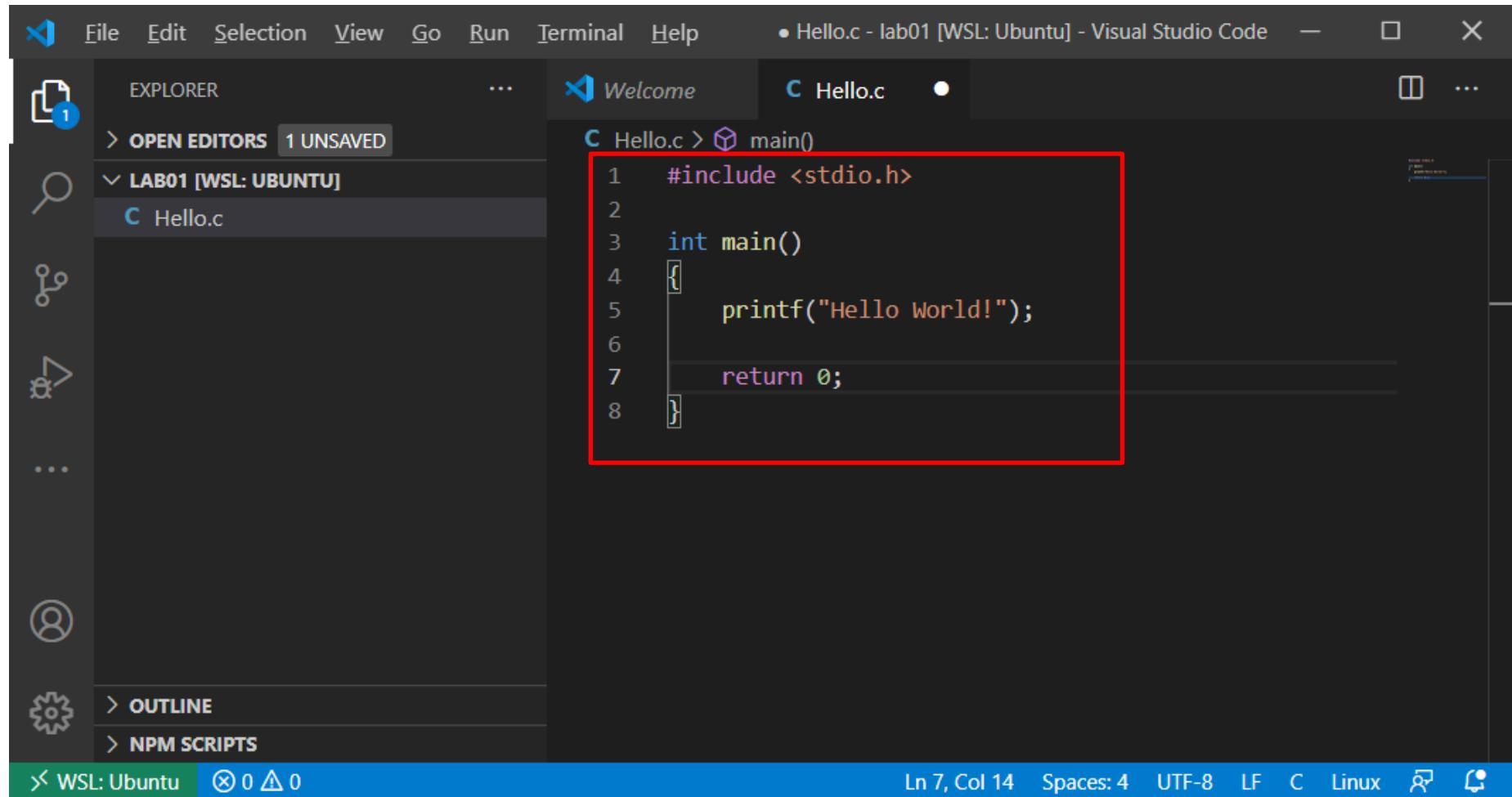
Find your folder where the code will be stored and select the folder.







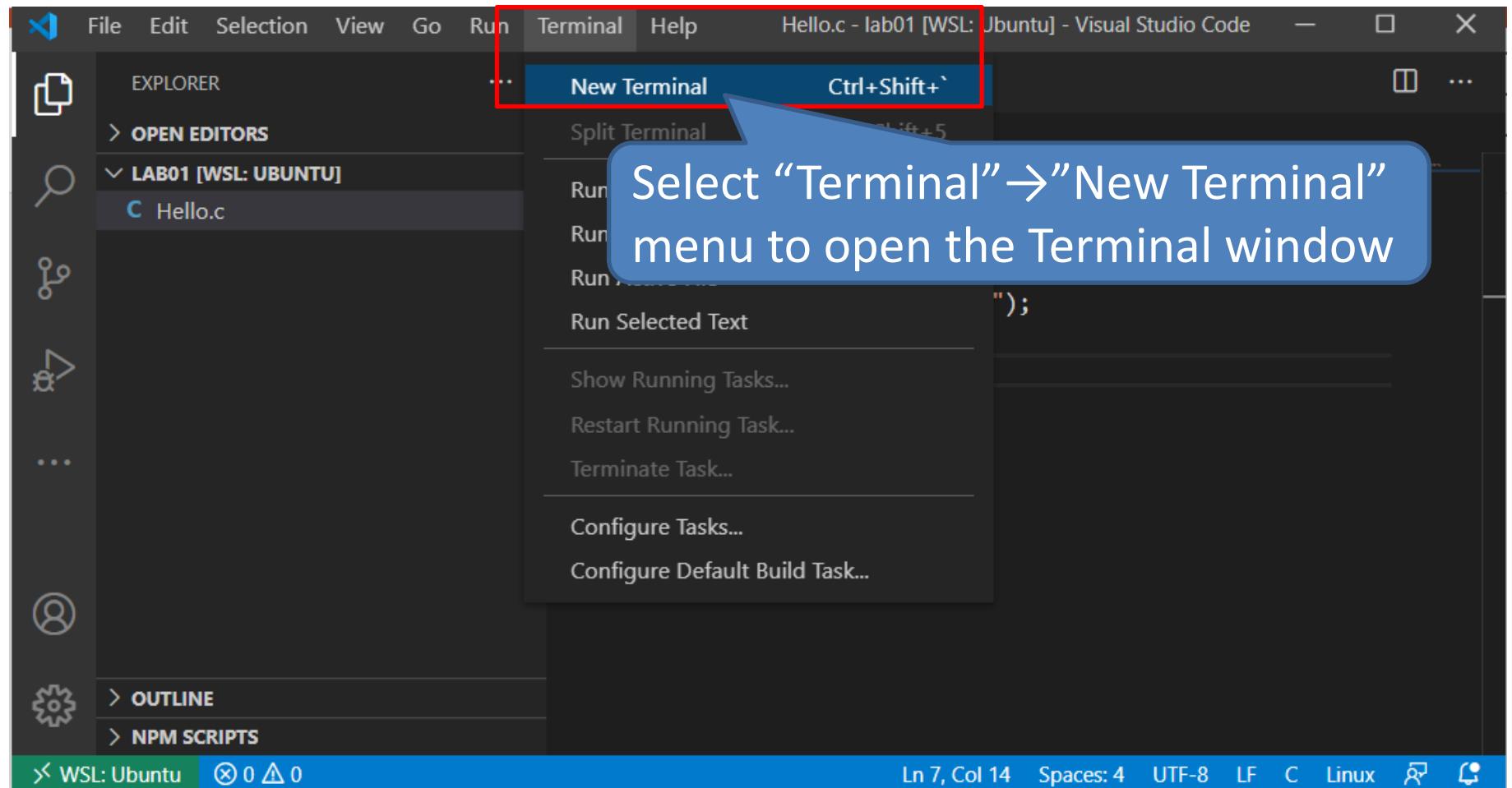
Input the code below and save the file.



The screenshot shows the Visual Studio Code interface with a dark theme. The left sidebar contains icons for File Explorer, Search, Problems, and others. The main area shows an open editor for a file named 'Hello.c' located in a folder named 'LAB01 [WSL: UBUNTU]'. The code in the editor is:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Hello World!");
6
7     return 0;
8 }
```

A red rectangle highlights the entire code block. The status bar at the bottom indicates the file is saved in 'WSL: Ubuntu', has 0 changes, and shows file statistics: Ln 7, Col 14, Spaces: 4, UTF-8, LF, C, Linux.



Select “Terminal”→“New Terminal”
menu to open the Terminal window

Use gcc to compile the .c file.

The screenshot shows the Visual Studio Code interface. The top tab bar has three tabs: "Welcome", "Hello.c", and "helloworld.cpp". The "Hello.c" tab is active, displaying the following C code:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Hello World!\n");
6
7     return 0;
8 }
```

Below the editor is a terminal window titled "1: bash". It shows the user's command-line session:

```
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ gcc -c hello.c
```

The terminal session continues with the following commands and output:

```
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ gcc -c hello.c
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ ls
Hello.c hello.o
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ gcc hello.o -o hello
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ ls
Hello.c hello hello.o
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ ./hello
Hello World!
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ 
```

The output

The default output executable file is called “a.exe”(Windows) or “a.out”(Unix and Mac OS) if you don’t specify the name in compiling and linking step.

The screenshot shows a Visual Studio Code interface running in WSL: Ubuntu. The Explorer sidebar shows files in the 'LAB01 [WSL: UBUNTU]' folder: 'a.out', 'hello', 'Hello.c' (selected), and 'hello.o'. The 'Hello.c' editor tab displays the following code:

```
#include <stdio.h>
int main()
{
    printf("Hello World!\n");
    return 0;
}
```

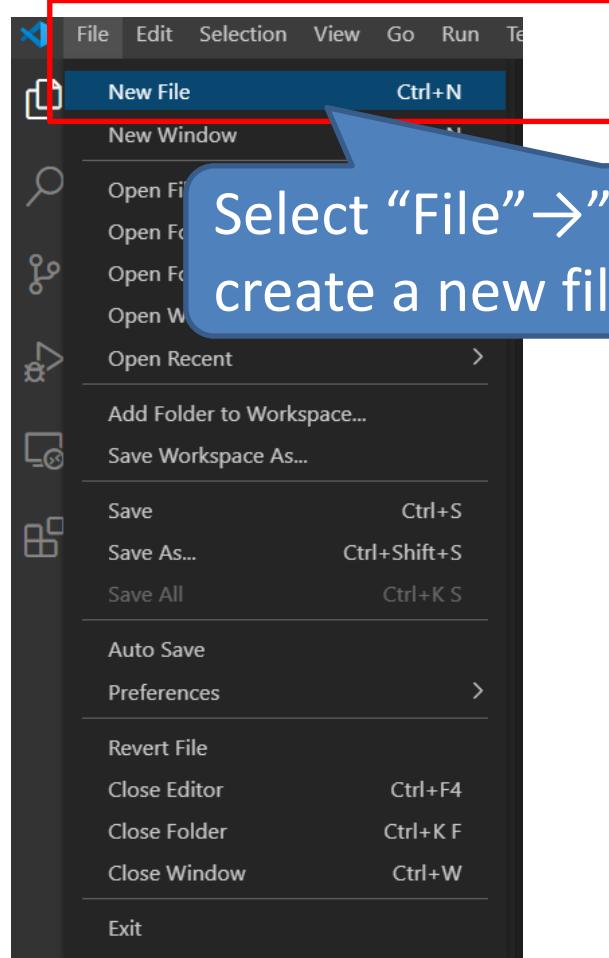
The terminal below shows the command-line steps to compile and run the program:

```
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ gcc -c hello.c
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ ls
Hello.c hello.o
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ gcc hello.o -o hello
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ ls
Hello.c hello hello.o
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ ./hello
Hello World!
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ gcc hello.c
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ ls
Hello.c a.out hello hello.o
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab01$ ./a.out
Hello World!
```

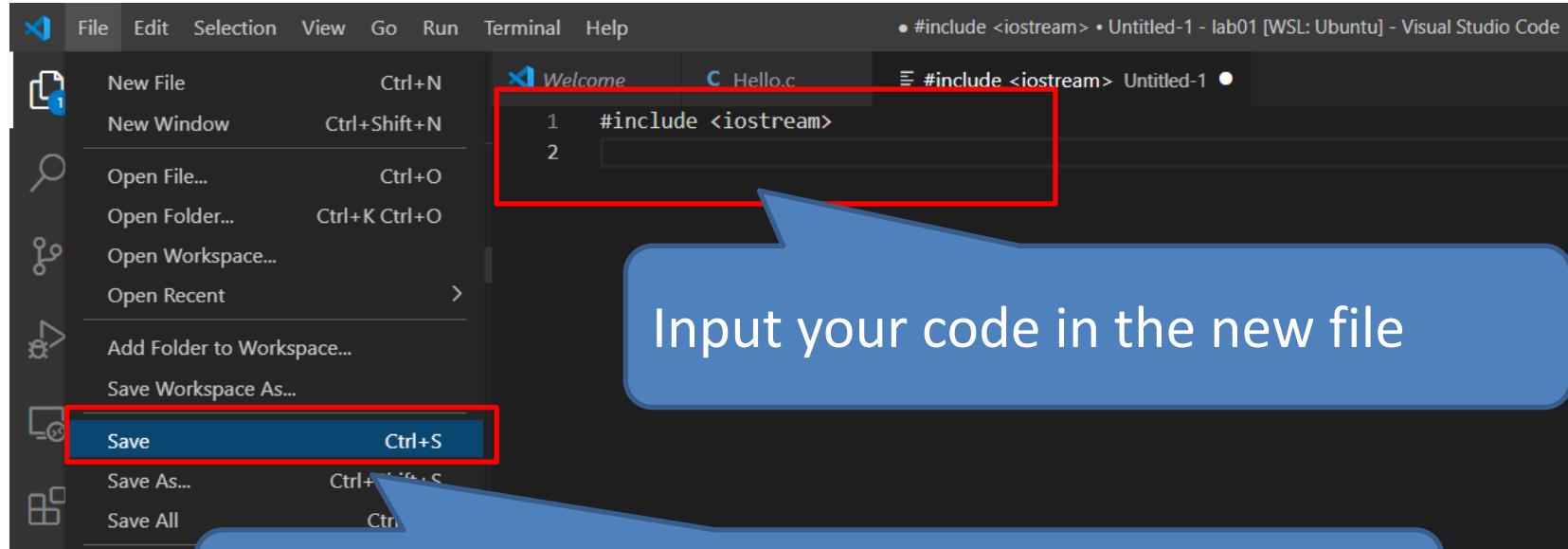
A blue speech bubble points to the terminal output with the text "The output". Red boxes highlight the command "gcc hello.c" in the terminal, the file "a.out" in the file list, and the command "./a.out". Red text annotations "compile and link" and "run a.out" are placed near these highlighted areas.

5.3 Compile, Link and Run C++ programs

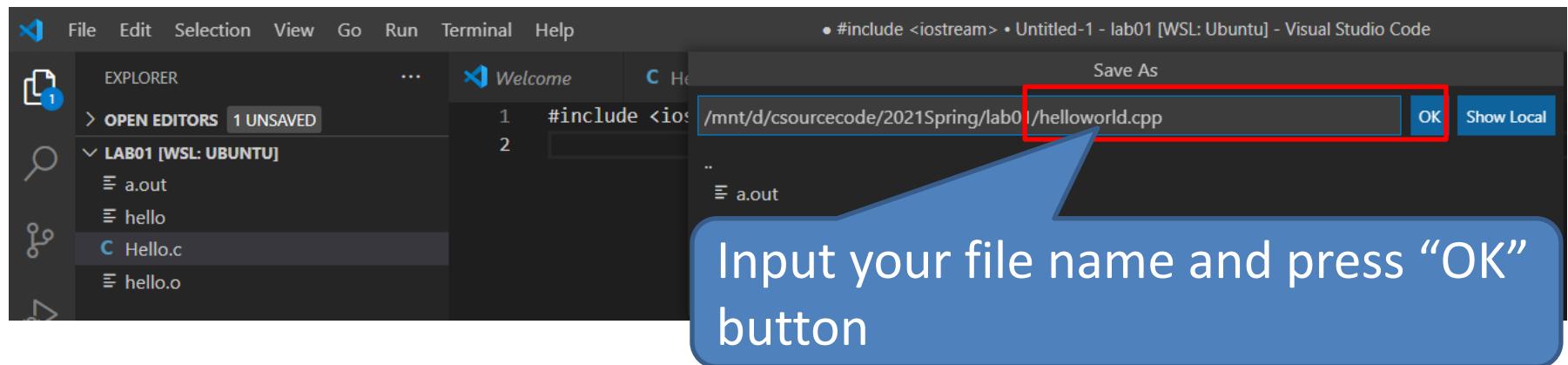
Compile/Link/Run a simple C++ program – helloworld.cpp



Select “File”→“New File” menu to
create a new file



Select “File”→“Save” menu to save the new file



A screenshot of the Visual Studio Code interface showing the saved file 'helloworld.cpp'. The editor tab bar indicates the file is saved. The main editor area displays the same C++ code as the previous screenshot, with syntax highlighting for keywords like 'int' and 'cout'.

You need to use **g++** to compile C++ program, as follows. We use the **-o** option to specify the output file name.

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab01$ g++ -c helloworld.cpp  
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab01$ ls  
Hello.c a.out hello hello.o helloworld.cpp helloworld.o  
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab01$ g++ helloworld.o -o helloworld  
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab01$ ls  
Hello.c a.out hello hello.o helloworld.helloworld.cpp helloworld.o  
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab01$ ./helloworld  
Hello World!!!  
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab01$
```

The output

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab01$ g++ -o helloworld2 helloworld.cpp  
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab01$ ls  
Hello.c a.out hello hello.o helloworld.helloworld.cpp helloworld.o helloworld2  
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab01$ ./helloworld2  
Hello World!!!  
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab01$
```

The output

Compile/Link/Run a simple C program – hello.c using Cygwin

Edit the codes below in Notepad++, and save it as hello.c to the D:\Software\Cygwin\code.

```
1 //hello.c
2 #include <stdio.h>
3 int main()
4 {
5     printf("Hello World!\n");
6     return 0;
7 }
```

Start Cygwin, switch the current directory to the “D:\Software\Cygwin\code” in which hello.c is stored.

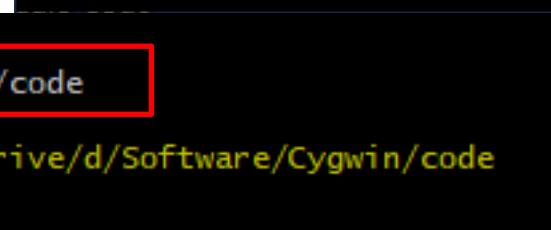
or



The screenshot shows a terminal window with a black background and white text. It displays the following command sequence:

```
lichuanwei@DESKTOP-D77F98D ~
$ cd /cygdrive/d
lichuanwei@DESKTOP-D77F98D /cygdrive/d
$ cd Software
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software
$ cd Cygwin
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin
$ cd code
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$
```

The lines \$ cd /cygdrive/d, \$ cd Software, and \$ cd code are highlighted with red boxes.



The screenshot shows a terminal window with a black background and white text. It displays the following command:

```
lichuanwei@DESKTOP-D77F98D ~
$ cd /cygdrive/d/Software/Cygwin/code
```

This line is highlighted with a red box.

```
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ gcc -c hello.c
```

compile

```
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ ls
hello.c hello.o
```

```
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ gcc hello.o -o hello
```

link

```
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ ls
hello.c hello.exe hello.o
```

```
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ ./hello
Hello World!
```

run

```
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$
```

The default output executable file is called “a.exe”(Windows) or “a.out”(Unix and Mac OS) if you don’t specify the name in compiling and linking step.

```
Lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ gcc hello.c      compile and link
```



```
Lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ ls
a.exe  hello.c  hello.cpp  hello.exe  hello.o  morning.c  morning.exe
```

```
Lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ ./a          run a.exe
Hello World!
```

```
Lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$
```

```
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ gcc morning.c -o morning
```

compile and link

```
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ ls
hello.c  hello.exe  hello.o  morning.c  morning.exe
```

output file name

```
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ ./morning
```

run morning.exe

```
Good morning!
```

```
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
```

Compile/Link/Run a simple C++ program – hello.cpp **using Cygwin**

Edit the codes below in Notepad++, and save it as hello.cpp to the D:\Software\Cygwin\code.

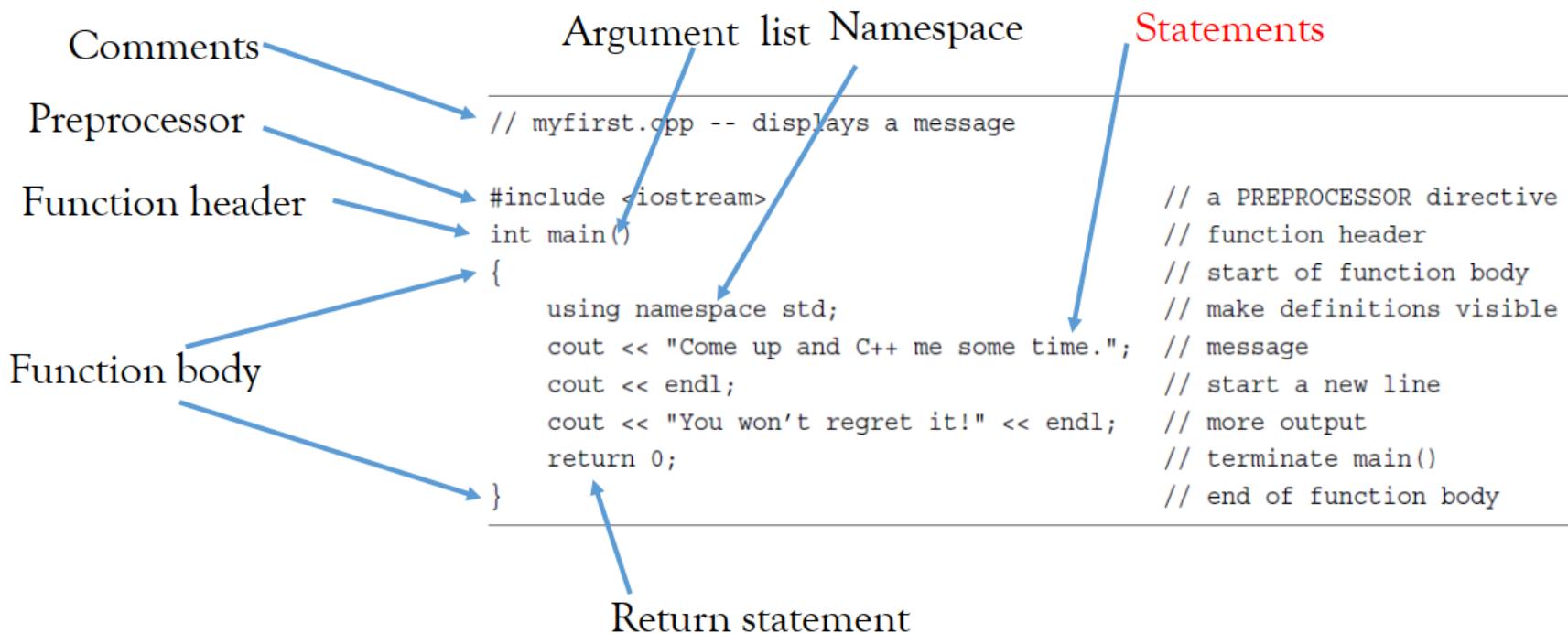
```
1 //hello.cpp
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     cout << "Hello world!" << endl;
7     return 0;
8 }
```

You need to use **g++** to compile C++ program, as follows. We use the **-o** option to specify the output file name.

```
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ g++ -c hello.cpp      compile
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ ls
hello.c  hello.cpp  hello.exe  hello.o  morning.c  morning.exe
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ g++ hello.o -o hello    link
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ ls
hello.c  hello.cpp  hello.exe  hello.o  morning.c  morning.exe
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ ./hello      run
Hello World!
```

```
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ g++ -o hello hello.cpp      compile and link
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ ls
hello.c  hello.cpp  hello.exe  hello.o  morning.c  morning.exe
lichuanwei@DESKTOP-D77F98D /cygdrive/d/Software/Cygwin/code
$ ./hello      run
Hello world!
```

C/C++ program's structure



```
#include <iostream>

void simple(); // function prototype

int main()
{
    using namespace std;
    cout << "main() will call the simple() function:\n";
    simple(); // function call

    cout << "main() is finished with the simple() function.\n";

    return 0;
}

// function definition
void simple()
{
    using namespace std;
    cout << "I'm but a simple function.\n";
}
```

C++ Source Code Style

- One statement per line
- An opening brace and a closing brace for a function, each of which is on its own line
- Statements in a function indented from the braces
- No whitespace around the parentheses associated with a function name

Basic commands of Linux

ls (小写字母l 小写字母s) #以默认方式显示当前目录文件列表

ls -a #显示所有文件包括隐藏文件

ls -l #显示文件属性，包括大小，日期，符号连接，是否可读写及是否可执行

cd dir #切换到当前目录下的dir目录

cd / #切换到根目录

cd .. #切换到到上一级目录

cd ../../ #切换到上二级目录

cd ~ #切换到用户目录，比如是root用户，则切换到/root下

rm file #删除某一个文件

rm -fr dir #删除当前目录下叫dir的整个目录

cp source target #将文件source 复制为 target

cp /root/source . #将/root下的文件source复制到当前目录

NOTE: Each lab attendance will be recorded by the submission of the lab exercise.

6 Exercises

1. Write a program to print your name and address.
2. Many studies suggest that smiling has benefits. Write a program that produces the following output:

Smile! Smile! Smile!

Smile! Smile!

Smile!

Have the program define a function that displays the string Smile! once, and have the program use the function as often as needed.