

Overview of On-Line Course

Kengo Nakajima
Information Technology Center
The University of Tokyo

<http://nkl.cc.u-tokyo.ac.jp/21w/>

First of All ...

- **Please make sure that:**
 - the OS of your PC is the most updated version
 - proper anti-virus software with the most updated version is installed to your PC
 - the most recent version of Zoom is installed
- **You need to know Unix/Linux and Editors (e.g. vi, emacs), if you want to use “Supercomputers”**
 - **List of Unix/Linux Commands (Wikipedia)**
 - https://en.wikipedia.org/wiki/List_of_Unix_commands
 - **Online Manual for Emacs (Screen Editor for Linux/Unix)**
 - <https://www.gnu.org/software/emacs/manual/>
- **My e-mail: nakajima@cc.u-tokyo.ac.jp**

Information

- Access UTAS (UTokyo Academic Affairs System) and ITC-LMS (ITC Learning Management System)
 - <https://utas.adm.u-tokyo.ac.jp/>
 - <https://itc-lms.ecc.u-tokyo.ac.jp/>
- Class Materials
 - <http://nkl.cc.u-tokyo.ac.jp/21w/>
- Most of the important information (including Zoom URL for Classes) will be given through ITC-LMS
 - Please check ITC-LMS AT LEAST ONCE a WEEK
 - You can receive e-mail from ITC-LMS, if you have new “notification”
- **Please register your info. at URL in UTAS/ITC-LMS.**
 - **You receive detailed info. of lectures.**
 - **You need to register the info. just once.**

Zoom “Manners”

- **Please install the most recent version of Zoom**
 - Please login via g.ecc.u-tokyo.ac.jp account
- **Basically, please keep muted, and keep camera off.**
- **Please do not video the lectures**
 - I will record the class, and tell you the address of the recorded file (on cloud) via ITC-LMS
- **Please do not forward URL’s of the classes to anybody, even he/she is a U.Tokyo student**
- I often use “Raising Hand” (e.g. How many of you learned OpenMP before ?). Please respond.
- **Q/A**
 - Every 10-15 minutes, I make a break for Q/A.
 - Turn on your microphone for questions
 - You can also use chat/e-mail/Slack

Questions

- Q1: OS of Your PC
 - Windows: Many Win Users
 - MAC
 - Linux
- Q2: Your “Mother Tongue”
 - Fortran
 - C/C++
 - Java
 - Python
 - **Anyway, you must be familiar with Fortran or C/C++**
 - Materials for both of C and Fortran are provided, although I will show you those in C on the screen
- Q3: Do you like to use “Slack” ?
 - I will send invitation to each of you, while you can decline to join

Software to be installed

	C	Fortran
Windows	Cygwin Paraview FEM-in-C	Cygwin Paraview FEM-in-F
Mac	Paraview	Paraview
Linux	FEM-in-C	FEM-in-F

- Cygwin <https://www.cygwin.com/>
- Paraview <http://www.paraview.org>
- Target Application by Finite Element Method (FEM)
 - FEM-in-F (Fortran) <http://nkl.cc.u-tokyo.ac.jp/files/fem-f.tar>
 - FEM-in-C (C) <http://nkl.cc.u-tokyo.ac.jp/files/fem-c.tar>
 - **Default compilers in Makefile's are "cc (FEM-in-C, C)" and "gfortran (FEM-in-F, Fortran)". Please modify Makefile's according to compilers on your PC**

- **In the following several pages, installation procedures of Cygwin are described**
- **Mac and Unix/Linux users do not need Cygwin**

Cygwin: Unix-like Environment on Windows

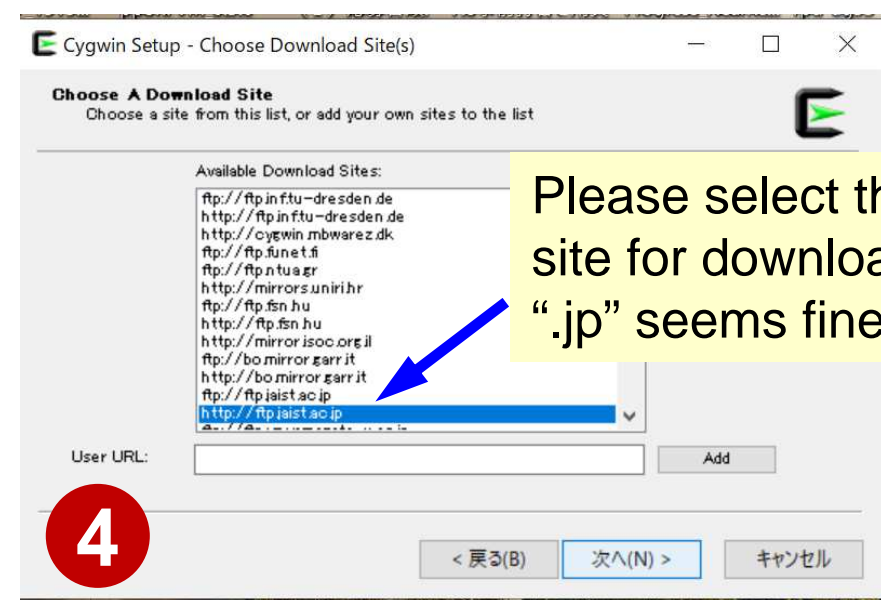
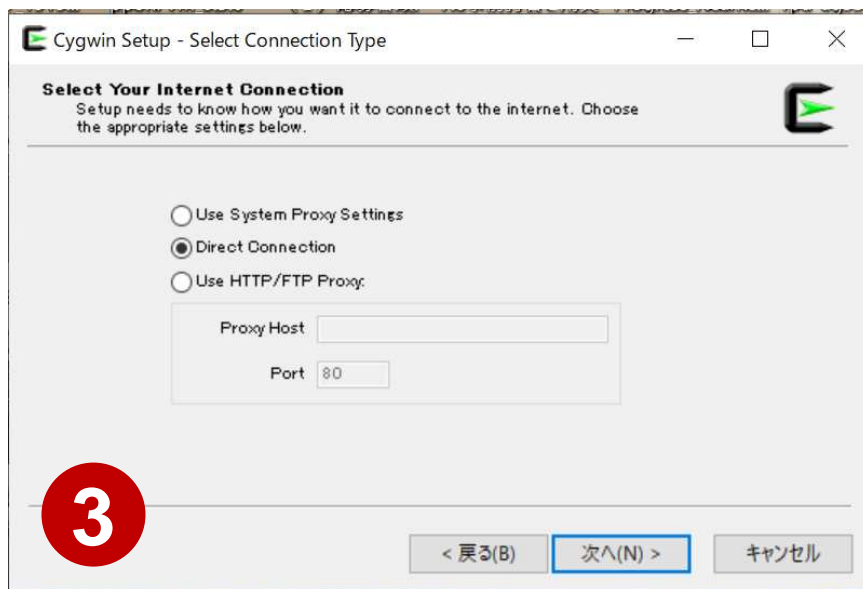
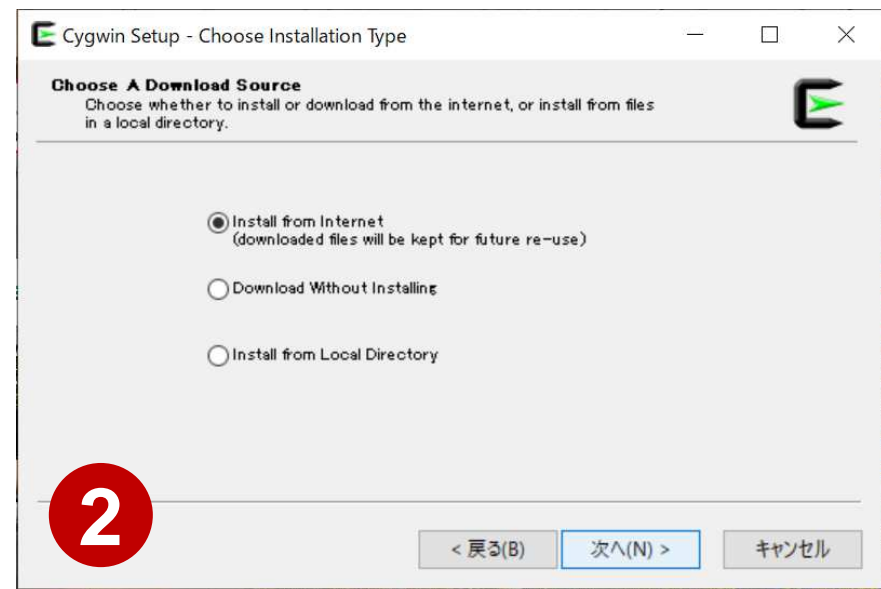
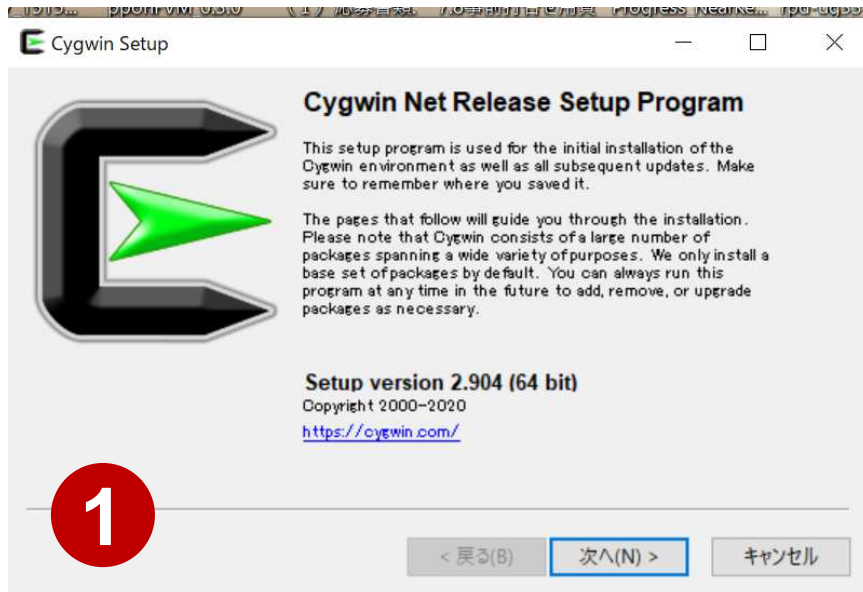
<https://www.cygwin.com/>

- Cygwin User's Guide
 - <https://cygwin.com/cygwin-ug-net.html>
- The first things what you have to do
 - Get “setup-x86_64/32.exe (installer)” from Cygwin site
 - Connect to the Internet and Double-Click the Installer
 - Just follow the instructions



Just Follow as they guide ...

<https://www.cygwin.com/>



Starting from the Default Installation

<https://www.cygwin.com/>

- Most of the fundamental tools/functions are installed, but some essential ones for this class are NOT ...
 - Therefore, you need to add them manually
 - You can install them later if you double-click the installer
- **The following functions must be installed, because they are not necessarily installed as default**
 - **gcc-core (for C/C++ users)**
 - **gcc-fortran (for Fortran users)**
 - **openssh (for all users)**
 - **openssl (for all users)**
 - **make (for all users)**
 - **emacs, vim etc.**
- **You can also check whether these are installed, or not**

Checking installation of “gcc-core”

Type “gcc-core”

The screenshot shows the 'Cygwin Setup - Select Packages' window. The search bar contains 'gcc-core' and is highlighted with a red box. A red arrow points from the text 'Type “gcc-core”' to the search bar. The table below shows the search results.

Package	Current	New	Src?	Categories	Size	Description
cygwin32-gcc-core		Skip	<input type="checkbox"/>	Devel	16,464k	GCC for Cygwin 32bit toolchain (C, OpenMP)
djgpp-gcc-core		Skip	<input type="checkbox"/>	Devel	7,926k	GCC for DJGPP toolchain (C)
gcc-core	9.3.0-1	Keep	<input type="checkbox"/>	Devel	20,500k	GNU Compiler Collection (C, OpenMP)
mingw64-i686-gcc-core		Skip	<input type="checkbox"/>	Devel	16,851k	GCC for Win32 (i686-w64-mingw32) toolchain (C, OpenMP)
mingw64-x86_64-gcc-core		Skip	<input type="checkbox"/>	Devel	17,464k	GCC for Win64 toolchain (C, OpenMP)

At the bottom of the window, there are navigation buttons: '< 戻る(B)', '次へ(N) >', and 'キャンセル'. The system tray at the bottom right shows the time 20:19 and date 2020/04/14.

Checking installation of “gcc-core”

Type “gcc-core”

The screenshot shows the 'Cygwin Setup - Select Packages' window. The search bar contains 'gcc-core'. The search results table is as follows:

Package	Current	New	Src?	Categories	Size	Description
cygwin32-gcc-core		Skip	<input type="checkbox"/>	Devel	16,464k	GCC for Cygwin 32bit toolchain (C, OpenMP)
gcc-core	9.3.0-1	Keep	<input type="checkbox"/>	Devel	20,500k	GNU Compiler Collection (C, OpenMP)
mingw64-x86_64-gcc-core		Skip	<input type="checkbox"/>	Devel	17,464k	GCC for Win64 toolchain (C, OpenMP)

A red box highlights the search bar containing 'gcc-core'. A blue box highlights the row for 'gcc-core', which shows 'Keep' in the 'New' column. A blue arrow points from a text box to this row.

“gcc-core” is already installed if you find “Keep”

Buttons at the bottom: < 戻る(B), 次へ(N) >, キャンセル

Installing “g++” (1/4)

Type “g++”

Cygwin Setup - Select Packages

Select Packages
Select packages to install

View Full Search g++ Clear

Keep Best Sync Test

Package	Current	New	Src?	Categories	Size	Description
cygwin32-gcc-g++		Skip	<input type="checkbox"/>	Devel	10,456k	GCC for Cygwin 32bit toolchain (C++)
dirac-gcc-g++		Skip	<input type="checkbox"/>	Devel	8,279k	GCC for DJGPP toolchain (C++)
gcc-g++		Skip	<input type="checkbox"/>	Devel	16,257k	GNU Compiler Collection (C++)
mingw64-i686-gcc-g++		Skip	<input type="checkbox"/>	Devel	14,358k	GCC for Win32 (i686-w64-mingw32) toolchain (C++)
mingw64-x86_64-gcc-g++		Skip	<input type="checkbox"/>	Devel	14,781k	GCC for Win64 toolchain (C++)

If “Skip” appears, it is not installed yet

In the following 4-pages, a example of manual install of “g++” is described.

Hide obsolete packages

< 戻る(B) 次へ(N) > キャンセル

Installing “g++” (2/4)

Cygwin Setup - Select Packages

Select Packages
Select packages to install

View Full Search g++ Clear

Keep Best Sync Test

Package	Current	New	Src?	Categories	Size	Description
cygwin32-gcc-g++		Skip	<input type="checkbox"/>	Devel	10,456k	GCC for Cygwin 32bit toolchain (C++)
dragon-gcc-g++		Skip	<input type="checkbox"/>	Devel	8,279k	GCC for DJGPP toolchain (C++)
gcc-g++		Skip	<input type="checkbox"/>	Devel	16,257k	GNU Compiler Collection (C++)
mingw64-i686-gcc-g++		Skip	<input type="checkbox"/>	Devel	14,358k	GCC for Win32 (i686-w64-mingw32) toolchain (C++)
mingw64-x86_64-gcc-g++		Skip	<input type="checkbox"/>	Devel	14,781k	GCC for Win64 toolchain (C++)

Hide obsolete packages

< 戻る(B) 次へ(N) > キャンセル

20:18
2020/04/14

Pull down this

Installing “g++” (3/4)

Cygwin Setup - Select Packages

Select Packages
Select packages to install

View Full Search g++ Clear Keep Best Sync Test

Package	Current	New	Src?	Categories	Size	Description
cygwin32-gcc-g++		Skip	<input type="checkbox"/>	Devel	10,456k	GCC for Cygwin 32bit toolchain (C++)
djgpp-gcc-g++		Skip	<input type="checkbox"/>	Devel	8,279k	GCC for DJGPP toolchain (C++)
gcc-g++		Skip	<input type="checkbox"/>	Devel	16,257k	GNU Compiler Collection (C++)
mingw64-i686-gcc-g++		Skip	<input type="checkbox"/>	Devel	14,358k	GCC for Win32 (i686-w64-mingw32) toolchain (C++)
mingw64-x86_64-gcc-g++		Skip	<input type="checkbox"/>	Devel	14,781k	GCC for Win64 toolchain (C++)

Uninstall
 Skip
 7.4.0-1
 8.3.0-1 (Test)
 9.2.0-1 (Test)
 9.2.0-2
 9.2.0-3
 9.3.0-1

You can select the proper version (generally, the most recent one)

Hide obsolete packages

< 戻る(B) 次へ(N) > キャンセル

20:18 2020/04/14

Installing “g++” (4/4)

Cygwin Setup - Select Packages

Select Packages
Select packages to install

View Full Search g++ Clear Keep Best Sync Test

Package	Current	New	Src?	Categories	Size	Description
cygwin32-gcc-g++		Skip	<input type="checkbox"/>	Devel	10,456k	GCC for Cygwin 32bit toolchain (C++)
djgpp-gcc-g++		Skip	<input type="checkbox"/>	Devel	8,279k	GCC for DJGPP toolchain (C++)
gcc-g++		9.3.0-1	<input type="checkbox"/>	Devel	16,257k	GNU Compiler Collection (C++)
mingw64-i686-gcc-g++		Skip	<input type="checkbox"/>	Devel	14,358k	GCC for Win32 (i686-w64-mingw32) toolchain (C++)
mingw64-x86_64-gcc-g++		Skip	<input type="checkbox"/>	Devel	14,781k	GCC for Win64 toolchain (C++)

If the version number appears instead of “Skip”,
“g++” is selected for installation
(Not installed yet)

Click this

Hide obsolete packages

< 戻る(B) 次へ(N) > キャンセル

21:58
2020/04/14

Confirm Successful Installation of “gcc”

```
$ gcc -v
```

```
組み込み spec を使用しています。
```

```
COLLECT_GCC=gcc
```

```
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-pc-cygwin/9.3.0/lto-wrapper.exe
```

```
ターゲット: x86_64-pc-cygwin
```

```
configure 設定: /cygdrive/i/szsz/tmp/gcc/gcc-9.3.0-1.x86_64/src/gcc-9.3.0/configure --
```

```
srcdir=/cygdrive/i/szsz/tmp/gcc/gcc-9.3.0-1.x86_64/src/gcc-9.3.0 --prefix=/usr --exec-prefix=/usr --
```

```
localstatedir=/var --sysconfdir=/etc --docdir=/usr/share/doc/gcc --htmldir=/usr/share/doc/gcc/html -C
```

```
--build=x86_64-pc-cygwin --host=x86_64-pc-cygwin --target=x86_64-pc-cygwin --without-libiconv-prefix -
```

```
--without-libintl-prefix --libexecdir=/usr/lib --enable-shared --enable-shared-libgcc --enable-static -
```

```
--enable-version-specific-runtime-libs --enable-bootstrap --enable-_cxa_atexit --with-dwarf2 --with-
```

```
tune=generic --enable-languages=c,c++,fortran,lto,objc,obj-c++ --enable-graphite --enable-
```

```
threads=posix --enable-libatomic --enable-libgomp --enable-libquadmath --enable-libquadmath-support --
```

```
disable-libssp --enable-libada --disable-symvers --with-gnu-ld --with-gnu-as --with-cloog-
```

```
include=/usr/include/cloog-isl --without-libiconv-prefix --without-libintl-prefix --with-system-zlib -
```

```
--enable-linker-build-id --with-default-libstdcxx-abi=gcc4-compatible --enable-libstdcxx-filesystem-ts
```

```
スレッドモデル: posix
```

```
gcc バージョン 9.3.0 (GCC)
```

Confirm Successful Installation of “gfortran”

```
$ gfortran -v
```

```
組み込み spec を使用しています。
```

```
COLLECT_GCC=gfortran
```

```
ターゲット: x86_64-pc-cygwin
```

```
configure 設定: /cygdrive/i/szsz/tmp/gcc/gcc-9.3.0-1.x86_64/src/gcc-9.3.0/configure --srcdir=/cygdrive/i/szsz/tmp/gcc/gcc-9.3.0-1.x86_64/src/gcc-9.3.0 --prefix=/usr --exec-prefix=/usr --localstatedir=/var --sysconfdir=/etc --docdir=/usr/share/doc/gcc --htmldir=/usr/share/doc/gcc/html -C --build=x86_64-pc-cygwin --host=x86_64-pc-cygwin --target=x86_64-pc-cygwin --without-libiconv-prefix --without-libintl-prefix --libexecdir=/usr/lib --enable-shared --enable-shared-libgcc --enable-static --enable-version-specific-runtime-libs --enable-bootstrap --enable__cxa_atexit --with-dwarf2 --with-tune=generic --enable-languages=c,c++,fortran,lto,objc,obj-c++ --enable-graphite --enable-threads=posix --enable-libatomic --enable-libgomp --enable-libquadmath --enable-libquadmath-support --disable-libssp --enable-libada --disable-symvers --with-gnu-ld --with-gnu-as --with-cloog-include=/usr/include/cloog-isl --without-libiconv-prefix --without-libintl-prefix --with-system-zlib --enable-linker-build-id --with-default-libstdcxx-abi=gcc4-compatible --enable-libstdcxx filesystem-ts
```

```
スレッドモデル: posix
```

```
gcc バージョン 9.3.0 (GCC)
```

Confirm Successful Installation of “ssh-keygen (OpenSSH)”

```
$ ssh-keygen --h
```

```
ssh-keygen: unknown option -- -
```

```
usage: ssh-keygen [-q] [-b bits] [-C comment] [-f output_keyfile] [-m format]
                [-t dsa | ecdsa | ecdsa-sk | ed25519 | ed25519-sk | rsa]
```

```
                [-N new_passphrase] [-O option] [-w provider]
ssh-keygen -p [-f keyfile] [-m format] [-N new_passphrase]
                [-P old_passphrase]
```

```
ssh-keygen -i [-f input_keyfile] [-m key_format]
```

```
ssh-keygen -e [-f input_keyfile] [-m key_format]
```

```
ssh-keygen -y [-f input_keyfile]
```

```
(...)
```

```
ssh-keygen -L [-f input_keyfile]
```

```
ssh-keygen -A [-f prefix_path]
```

```
ssh-keygen -k -f krl_file [-u] [-s ca_public] [-z version_number]
                file ...
```

```
ssh-keygen -Q -f krl_file file ...
```

```
ssh-keygen -Y find-principals -s signature_file -f allowed_signers_file
```

```
ssh-keygen -Y check-novalidate -n namespace -s signature_file
```

```
ssh-keygen -Y sign -f key_file -n namespace file ...
```

```
ssh-keygen -Y verify -f allowed_signers_file -I signer_identity
                -n namespace -s signature_file [-r revocation_file]
```

Confirm Successful Installation of “ssh (OpenSSH)”

```
$ ssh
```

```
usage: ssh [-46AaCfGgKkMnqsTtVvXxYy] [-B bind_interface]
          [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
          [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
          [-i identity_file] [-J [user@]host[:port]] [-L address]
          [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
          [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
          [-w local_tun[:remote_tun]] destination [command]
```

Confirm Successful Installation of “make, emacs, vi etc”

```
$ make -version
```

```
GNU Make 4.3
```

```
このプログラムは x86_64-pc-cygwin 用にビルドされました
```

```
Copyright (C) 1988-2020 Free Software Foundation, Inc.
```

```
ライセンス GPLv3+: GNU GPL バージョン 3 以降 <http://gnu.org/licenses/gpl.html>
```

```
これはフリーソフトウェアです: 自由に変更および配布できます。
```

```
法律の許す限り、 無保証 です。
```

```
$ emacs -version
```

```
GNU Emacs 26.3
```

```
Copyright (C) 2019 Free Software Foundation, Inc.
```

```
GNU Emacs comes with ABSOLUTELY NO WARRANTY.
```

```
You may redistribute copies of GNU Emacs
```

```
under the terms of the GNU General Public License.
```

```
For more information about these matters, see the file named COPYING.
```

```
$ vi -version
```

```
VIM - Vi IMproved 8.2 (2019 Dec 12, compiled Mar 30 2020 21:54:08)
```

```
Garbage after option argument: “-version”
```

```
More info with: “vim -h”
```

Please install ALL environments before the Next Lecture

- Generally, installation of Cygwin takes a time (60-90 mins)
- I recommend you to complete this during coming weekend
- Another choice is WSL (Windows Subsystem for Linux), if you are familiar with both of Linux and Windows