Installation of OpenMX on WSL on Windows 10/11

The document describes the procedure for installing OpenMX Ver. 3.9 on WSL on a Windows 10/11 PC. After installing WSL, Intel one API Base Toolkit, and HPC Toolkit, we install OpenMX. The installation of WLS on Windows 11 is easier than the case of Windows 10, the document will not discuss the installation for Windows 11.

Taisuke Ozaki

ISSP, the Univ. of Tokyo

April 2022

Remarks

- This is based on information as of April 2022, and may not work in the latest and/or different environments.
- This document is provided for information sharing purposes only. You are responsible for any damage caused by the installation process.
- This document is based on information from various web sites, and describes procedures that have been successfully implemented through trial and error. I would like to thank all the people who provided information on the web.

A list of installation procedures

You can install OpenMX on your Windows 10 PC by following these steps.

1. Checking your PC specs and free disk space.

   The Intel One API Base Toolkit and HPC Toolkit (described later) require about 30 GB of free disk space for installation. Make sure that you have enough free disk space. In general, the higher the clock speed of the CPU, the faster the execution speed of OpenMX. 1.8GHz or higher is recommended. The execution speed of OpenMX also depends on the writing speed of the disk.

2. Installation of WSL

   WSL (Windows Subsystem for Linux) enables a Linux environment on a Windows PC. The performance of OpenMX depends on the version of WSL, and the installation of an X server using VcXsrv is also recommended.

3. Installation of Intel one API Base Toolkit and HPC Toolkit

   By installing one API Base Toolkit and HPC Toolkit, one can use Intel compilers, Math Kernel Library (MKL), are Intel MPI.

4. Installation of OpenMX Ver. 3.9

   After the steps 1-3, we install OpenMX Ver. 3.9 with patch3.9.9.

PC environment used for testing

PC1

PC
Device name: LAPTOP-OS9EEKFG
Processor: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz
RAM: 16.0 GB
Disk: SSD, Reading speed 3000MB/s, Writing speed 1900MB/s

Windows 10
Edition: Windows 10 Pro
Version: 21H1
System: 64 bit

PC2

PC
Device name: LAPTOP-OS9EEKFG
Processor: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
RAM: 16.0 GB
Disk: SSD, Reading speed 3000MB/s, Writing speed 1900MB/s

Windows 10
Edition: Windows 10 Home
Version: 21H1
System: 64 bit

Checking your PC specs and free disk space

The Intel One API Base Toolkit and HPC Toolkit (described later) require about 30 GB of free disk space for installation. Make sure that you have enough free disk space. In general, the higher the clock speed of the CPU, the faster the execution speed of OpenMX. 1.8GHz or higher is recommended. The execution speed of OpenMX also depends on the writing speed of the disk.

The system requirements for installing the Intel one API Base Toolkit can be found on the web below.

https://software.intel.com/content/www/us/en/develop/articles/intel-oneapi-base-toolkit-system-requirements.html

The system requirements for installing the Intel one API HPC Toolkit can be found on the web below.

https://software.intel.com/content/www/us/en/develop/articles/intel-oneapi-hpc-toolkit-system-requirements.html

Installation of WSL #1

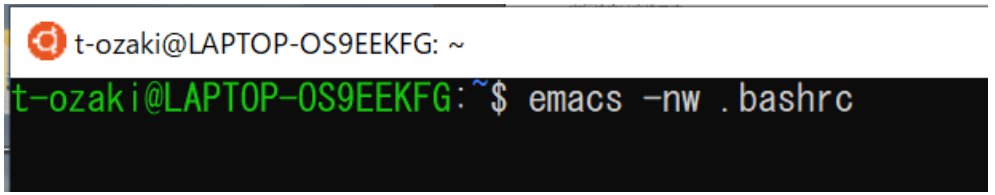
- WSL (Windows Subsystem for Linux) allows you to introduce a Linux environment on a Windows PC.
- Please refer to Microsoft's official website for the installation procedure.

  https://docs.microsoft.com/es-us/windows/wsl/install-win10

- The above web explains two methods, "Simplified Installation for Windows Insiders" and "Manual Installation Steps", and I used the latter method to install.
- Also, if you search for "wsl install" on Google, etc., you will find many sites that explain the installation procedure.
- The execution speed of OpenMX depends on the version of WSL; please install WSL1 and WSL2, perform benchmark calculations, and then select the appropriate version.
- We recommend Ubuntu 18.04 LTS as the Linux distribution, which has been confirmed to be stable for the OpenMX execution. (Make a note of the password you set during installation, as it will be used in the future.)

Installation of WSL #2

- After installing WSL, open the console window and modify the .bashrc using an editor. (In the example on the right, emacs is used, but you can use any editor you are familiar with.)



- Add the following line to the end of the .bashrc and save it.

The WSL and Windows file systems are managed separately, so it is easier to move to the Windows file system even when working on WSL; setting cdh in alias makes it easier to move to the windows file system.

For WSL1

```
export DISPLAY=:0.0
export LIBGL_ALWAYS_INDIRECT=1
alias xterm='xterm -rv -fn 12x24 &'
alias uxterm='uxterm -fa "DejaVu Sans Mono:style=Book" -fd IPAGothic:style=Regular -fs 12 &'
alias cdh='cd /mnt/c/Users/Taisuke¥ Ozaki/work/'
```

This part can be changed to suit your own environment.

For WSL2

```
export DISPLAY=$(awk '/nameserver / {print $2; exit}' /etc/resolv.conf 2>/dev/null):0
export LIBGL_ALWAYS_INDIRECT=1
alias xterm='xterm -rv -fn 12x24 &'
alias uxterm='uxterm -fa "DejaVu Sans Mono:style=Book" -fd IPAGothic:style=Regular -fs 12 &'
alias cdh='cd /mnt/c/Users/Taisuke¥ Ozaki/work/'
```

This part can be changed to suit your own environment.

Installation of WSL #3

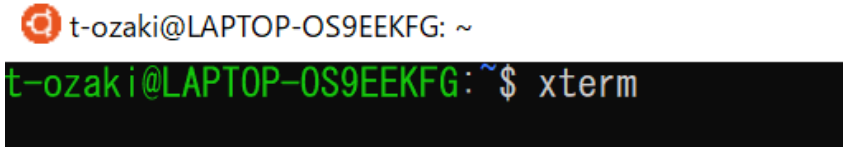
- The X server seems to be used when installing the Intel one API Base Toolkit and HPC Toolkit. So we will install an "X server".

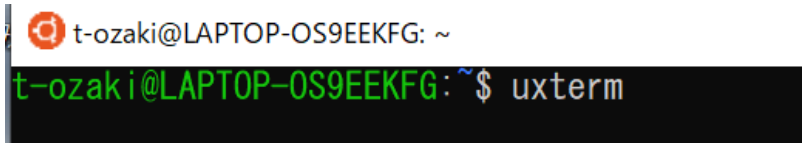
- By installing an X server, various GUI software on Linux can be used. It is also possible to open and work with multiple X-terminals, which is a CUI, improving work efficiency.

- There are several software packages available to install X server, such as VcXsrv, Cygwin/X, and Xming, and I am using VcXsrv. From the following site, you can download the VcXsrv installer.

https://sourceforge.net/projects/vcxsrv/

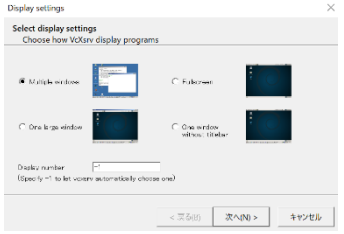
After installing the X server, you can launch xterm or uxterm (Japanese available) from the console window of WSL. Note: You can use the following command to start up xterm or uxterm by setting the alias in the .bashrc file as described in the previous page.





Installation of WSL #4

If you have installed the X server using VcXsrv and you are using WSL2, specify -ac for "Additional parameters for VcXsrv" in the "Display setting" of the X server (right figure).



Extra settings

Extra settings

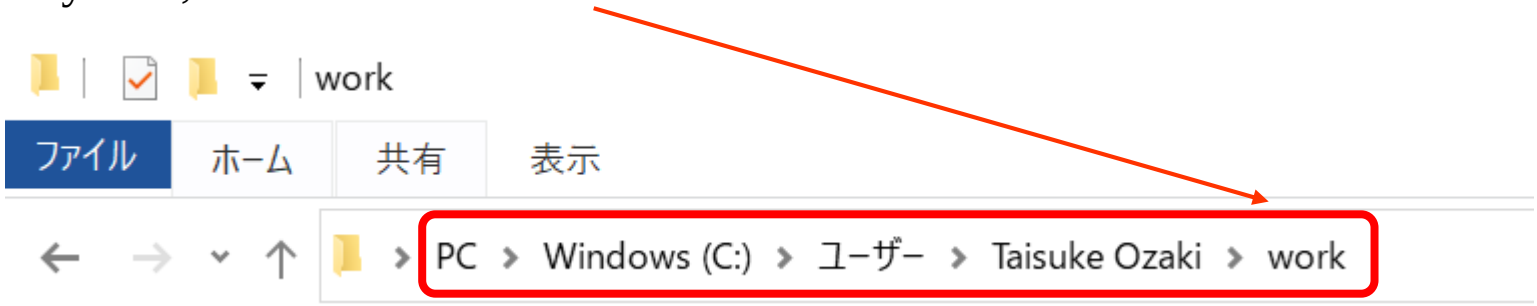
Clipboard
Start the integrated clipboard manager
Primary Selection
Also map the PRIMARY selection to the windows clipboard.
Native opengl
Use the native windows opengl library (wgl). Make sure to export the LIBGL_ALWAYS_INDIRECT environment variable.
Disable access control
Use this when you want vcxsrv to accept connections from all clients.
Additional parameters for VcXsrv
-ac

< 戻る(B) 次へ(N) > キャンセル

Installation of WSL #5

Create a working directory for WSL in Windows Explorer.

In my case, I have created a work as follows.



Correspondingly, the following alias is set in the .bashrc.

alias cdh='cd /mnt/c/Users/Taisuke¥ Ozaki/work/'

For more information about .bashrc, please refer to the slide "Introducing WSL #2".

Note that the .bashrc exists in the home directory managed by WSL, which can be checked from the WSL command prompt with the command shown right.

```
t-ozaki@LAPTOP-OS9EEKFG: ~
t-ozaki@LAPTOP-OS9EEKFG:~$ cd
t-ozaki@LAPTOP-OS9EEKFG:~$ ls -A .bashrc
.bashrc
t-ozaki@LAPTOP-OS9EEKFG:~$ tail .bashrc
  fi
fi

export DISPLAY=$(awk '/nameserver / {print $2; exit}' /etc/resolv.conf 2>/dev/null):0
export LIBGL_ALWAYS_INDIRECT=1
alias xterm='xterm -rv -fn 12x24 &'
alias uxterm='uxterm -fa "DejaVu Sans Mono:style=Book" -fd IPAGothic:style=Regular -fs 12 &'
alias cdh='cd /mnt/c/Users/Taisuke¥ Ozaki/work/'

source /opt/intel/oneapi/setvars.sh
t-ozaki@LAPTOP-OS9EEKFG:~$
```

Installation of WSL #6

Execute the following required command in the console window of WSL.

```
sudo apt update
sudo apt upgrade
sudo apt install make
sudo apt install gcc
sudo apt install g++
sudo apt install libnss3
```

Install the necessary applications if you need.

```
sudo apt install gnuplot
sudo apt install gnuplot-x11
sudo apt install emacs
sudo apt install vim          etc.
```

Installation of Intel one API Base Toolkit #1

The Intel one API Base Toolkit includes the following libraries and others.

Intel® oneAPI Collective Communications Library
Intel® oneAPI Data Analytics Library
Intel® oneAPI Deep Neural Network Library
Intel® oneAPI DPC++/C++ Compiler
Intel® oneAPI DPC++ Library
Intel® oneAPI Math Kernel Library
Intel® oneAPI Threading Building Blocks
Intel® oneAPI Video Processing Library
Intel® Advisor
Intel® Distribution for GDB*
Intel® Distribution for Python*
Intel® DPC++ Compatibility Tool
Intel® Integrated Performance Primitives
Intel® VTune™ Profiler
Optional: Intel® FPGA Add-on for oneAPI Base Toolkit

The Base Toolkit is permitted to be used in accordance with the terms of the End User License Agreement. Please refer to the following URL for details.

https://software.intel.com/content/www/us/en/develop/articles/end-user-license-agreement.html

Installation of Intel one API Base Toolkit #2

(1) Download the Base Kit (l_BaseKit_p_2021.3.0.3219_offline.sh) from the Intel website:

https://registrationcenter-download.intel.com/akdlm/irc_nas/17977/l_BaseKit_p_2021.3.0.3219_offline.sh

It is 3.4GB in size, so it will take a time (about 4 minutes) to download.

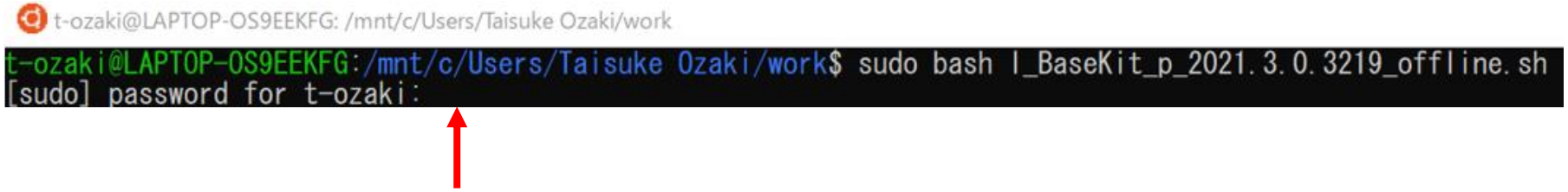
This document assumes the use of I_BaseKit_p_2021.3.0.3219_offline.sh.
The author has not succeeded in installing using the latest version of Base ToolKit.

(2) Copy "I_BaseKit_p_2021.3.0.3219_offline.sh" to your "work" directory.

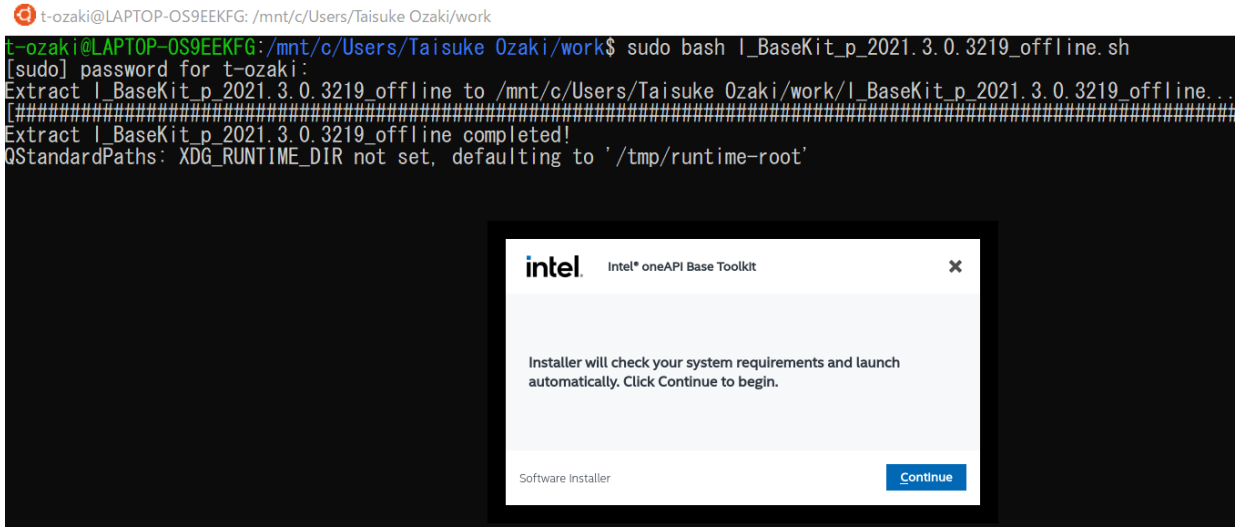
Installation of Intel one API Base Toolkit #3

(3) Execute the following command from the WSL command prompt.

sudo bash l_BaseKit_p_2021.3.0.3219_offline.sh



After entering the password that you set when you installed Ubuntu, you will be asked to check the system requirements and accept the End User License Agreement. After accepting, the installation will start.



When it finishes successfully, DPC++/C++ Compiler, Math Kernel Library, etc. will be installed under /opt/intel/oneapi/.

Installation of Intel one API HPC Toolkit #1

The Intel one API HPC Toolkit includes the following libraries and others.

- Intel® oneAPI DPC++/C++ Compiler
- Intel® oneAPI Fortran Compiler
- Intel® C++ Compiler Classic
- Intel® Cluster Checker
- Intel® Inspector
- Intel® MPI Library
- Intel® Trace Analyzer and Collector

Install the Intel one API HPC Toolkit after installing the Base Toolkit due to the dependency on the Base Toolkit.

The Base Toolkit is permitted to be used in accordance with the terms of the End User License Agreement. Please refer to the following URL for details.

https://software.intel.com/content/www/us/en/develop/articles/end-user-license-agreement.html

Installation of Intel one API HPC Toolkit #2

(1) Download the HPC Kit (l_HPCKit_p_2021.3.0.3230_offline.sh) from the Intel website:

https://registrationcenter-download.intel.com/akdlm/irc_nas/17912/l_HPCKit_p_2021.3.0.3230_offline.sh

It is 1.25GB in size, so it will take a time (about 2 minutes) to download.

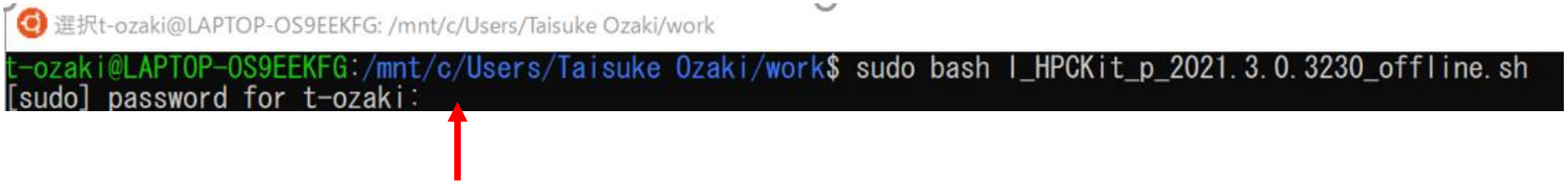
This document assumes the use of l_HPCKit_p_2021.3.0.3230_offline.sh.
The author has not succeeded in installing using the latest version of HPC ToolKit.

(2) Copy "l_HPCKit_p_2021.3.0.3230_offline.sh" to your "work" directory.

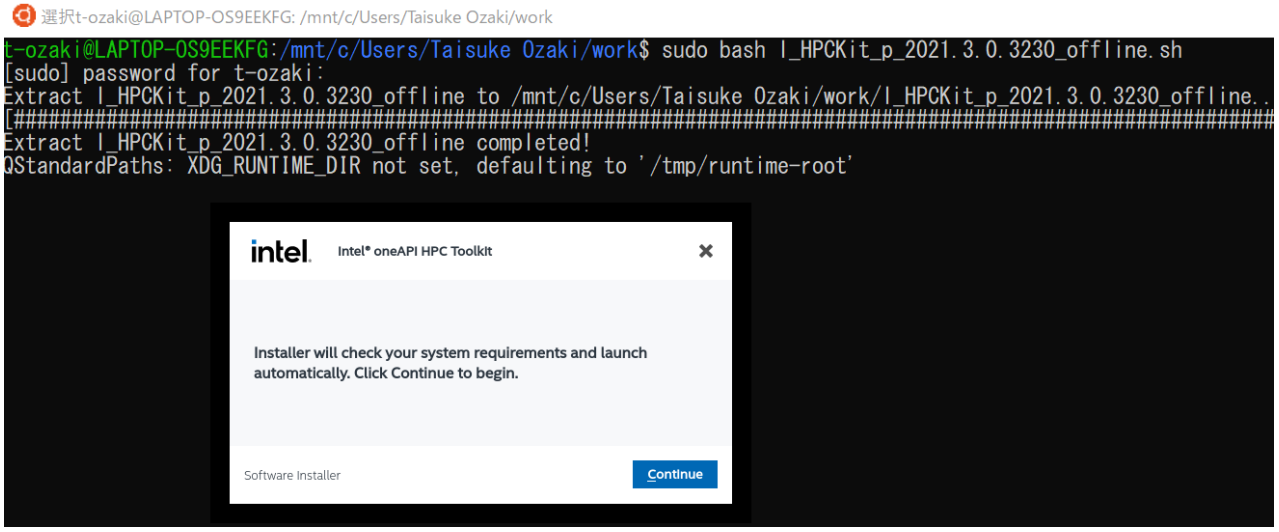
Installation of Intel one API HPC Toolkit #3

(3) Execute the following command from the WSL command prompt.

sudo bash l_HPCKit_p_2021.3.0.3230_offline.sh



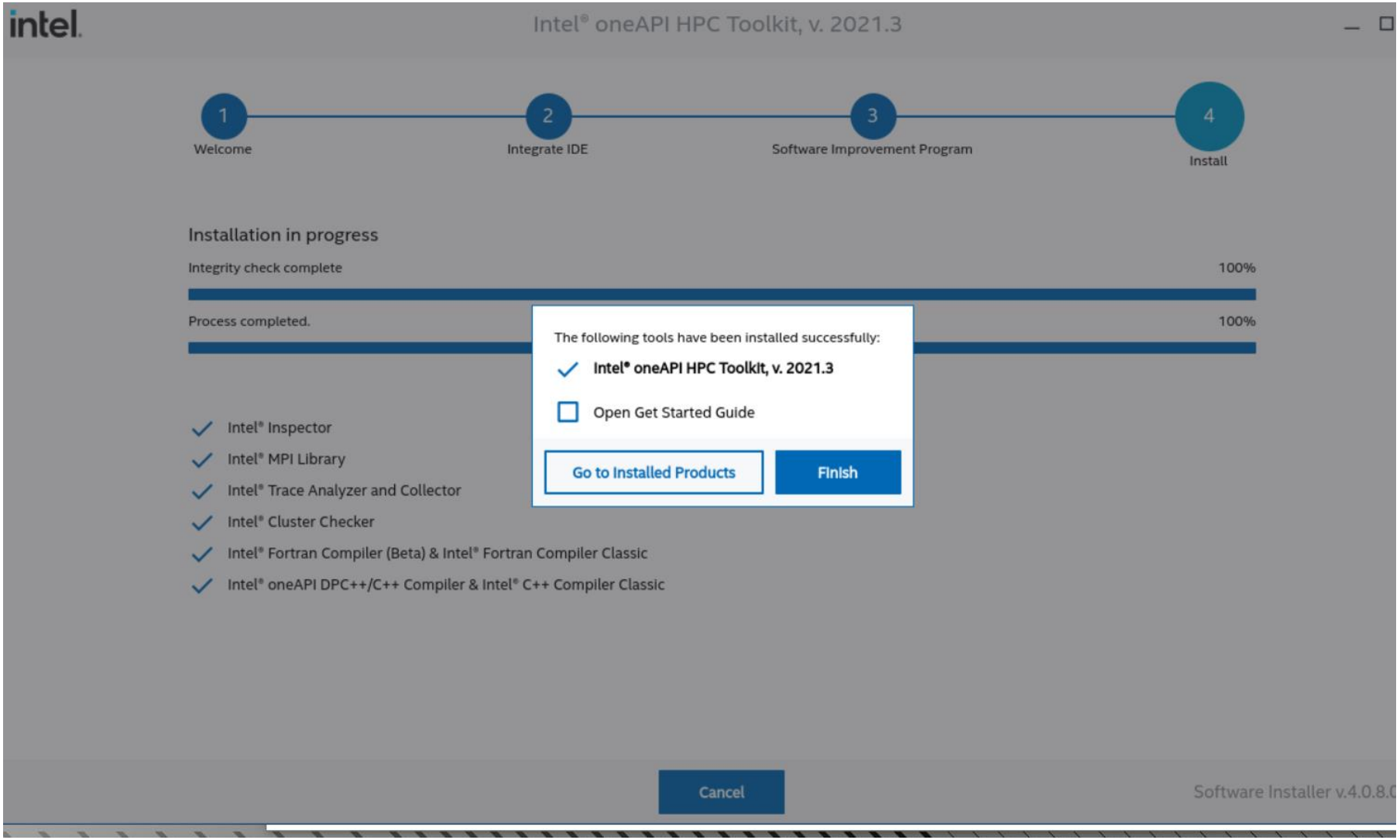
After entering the password that you set when you installed Ubuntu, you will be asked to check the system requirements and accept the End User License Agreement. After accepting, the installation will start.



When it finishes successfully, the Fortran compiler and MPI Library will be installed under /opt/intel/oneapi/.

Installation of Intel one API HPC Toolkit #4

The final screen when the installation is successful.



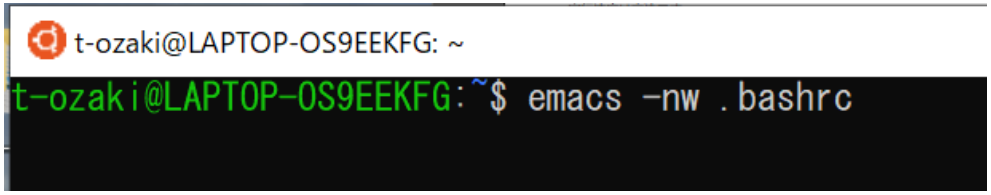
Confirming the installation of Intel one API Base Toolkit and HPC Toolkit #1

(1) From the WSL command prompt, execute the following command.

source /opt/intel/oneapi/setvars.sh

(2) Using an editor, add the following to the last line of the .bashrc

source /opt/intel/oneapi/setvars.sh



Contents of the .bashrc

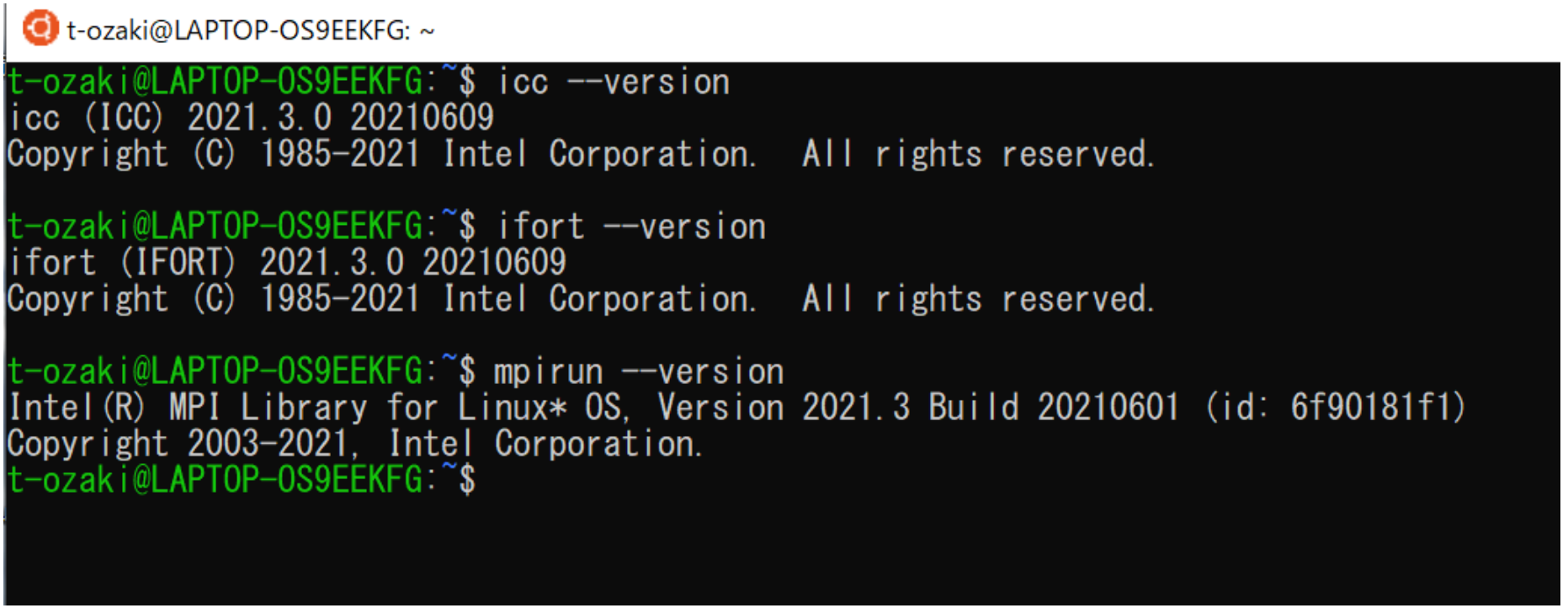
```
export DISPLAY=$(awk '/nameserver / {print $2; exit}' /etc/resolv.conf 2>/dev/null):0
export LIBGL_ALWAYS_INDIRECT=1
alias xterm='xterm -rv -fn 12x24 &'
alias uxterm='uxterm -fa "DejaVu Sans Mono:style=Book" -fd IPAGothic:style=Regular -fs 12 &'
alias cdh='cd /mnt/c/Users/Taisuke¥ Ozaki/work/'

source /opt/intel/oneapi/setvars.sh
```

The added line at the last line of .bashrc.

Confirming the installation of Intel one API Base Toolkit and HPC Toolkit #2

(3) From the WSL command prompt, execute the following commands. If the version information is displayed, the installation has been successful.



Installation of OpenMX Ver. 3.9 #1

http://www.openmx-square.org/

(1) Download the following files

openmx3.9.tar.gz
patch3.9.9.tar.gz

from the OpenMX website.

Click “Download” to go to the following page.

openmx3.9.tar.gz patch3.9.9.tar.gz

Each of these can be downloaded by clicking on the links below.

Welcome to OpenMX

Open source package for Material eXplorer

- What's new
  - Patch3.9.6 to OpenMX Ver 3.9 (Aug. 21, 2021)
  - Patch3.9.2 to OpenMX Ver 3.9 (Feb. 11, 2020)
- What is OpenMX?
- Download
- Manual of Ver. 3.9
- Manual of Ver. 3.8
- Technical Notes
- Video Lectures
- Publications
- OpenMX Forum
- OpenMX Viewer
- Workshop
- Database of Results
- Database of VPS and PAO
  - Ver. 2019
  - Ver. 2019 for core excitations
- ADPACK
- Miscellaneous informations

Up

Download of OpenMX

Available packages in terms of GNU-GPL Version 3 (GPLv3)

- openmx3.9 (release date: 03/Dec./2019, 158 MB) + patch (21/Aug./2021) README.txt
- openmx3.8 (release date: 03/April/2016, 136 MB) + patch (12/June/2018) README.txt
- openmx3.7 (release date: 23/May/2013, 112 MB) + patch (21/Feb./2015) README.txt

Installation of OpenMX Ver. 3.9 #2

(2) Copy openmx3.9.tar.gz and pach3.9.9.tar.gz to the "work" directory. You can place them under the directory specified by "alias cdh" in the .bashrc.

(3) In the WSL console, do the following

```
$ tar zxvf openmx3.9.tar.gz
$ cp ./patch3.9.9.tar.gz openmx3.9/source
$ cd openmx3.9/source
$ tar zxvfmp patch3.9.9.tar.gz
$ mv kpoint.in ../work/
```

(4) In the openmx3.9/souce makefile, specify the following using an editor.

```
MKLROOT = /opt/intel/oneapi/mkl/2021.3.0/
CC = mpiicc -O3 -xHOST -ip -no-prec-div -qopenmp -I${MKLROOT}/include -I${MKLROOT}/include/fftw
FC = mpiifort -O3 -xHOST -ip -no-prec-div -qopenmp
LIB= -L${MKLROOT}/lib/intel64 -lmkl_scalapack_lp64 -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core  -lifcore -lmkl_blacs_intelmpi_lp64
-liomp5 -lpthread -lm -ldl
```

Note that the LIB line is a single line.

(5) In the WSL console, do the following

```
$ make all
$ make install
```

When the installation is completed successfully, the executable file: openmx will be generated in "openmx3.9/work".

Benchmark calculations of OpenMX Ver. 3.9 #1

We performed the runtest in which OpenMX automatically performs calculations with 14 input files and compares the results with stored data. The calculations were performed on two Windows 10 PCs below.

PC1

PC
Device name: LAPTOP-OS9EEKFG
Processor: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz
RAM: 16.0 GB
Disk: SSD, Reading speed 3000MB/s, Writing speed 1900MB/s
Windows 10
Edition: Windows 10 Pro
Version: 21H1
System: 64 bit

PC2

PC
Device name: LAPTOP-OS9EEKFG
Processor: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
RAM: 16.0 GB
Disk: SSD, Reading speed 3000MB/s, Writing speed 1900MB/s
Windows 10
Edition: Windows 10 Home
Version: 21H1
System: 64 bit

As for runtest, please refer the website:

http://www.openmx-square.org/openmx_man3.9/node17.html

Benchmark calculations of OpenMX Ver. 3.9 #2

An example of the results of runtest by PC1

WSL1: mpirun -np 4 ./openmx -runtest -nt 1

1 input_example/Benzene.dat	Elapsed time(s)= 10.35	diff Utot= 0.000000000031	diff Force= 0.000000000003
2 input_example/C60.dat	Elapsed time(s)= 48.24	diff Utot= 0.000000000000	diff Force= 0.000000000001
3 input_example/CO.dat	Elapsed time(s)= 23.61	diff Utot= 0.000000000096	diff Force= 0.000000000262
4 input_example/Cr2.dat	Elapsed time(s)= 21.58	diff Utot= 0.000000000944	diff Force= 0.000000000294
5 input_example/Crys-MnO.dat	Elapsed time(s)= 84.86	diff Utot= 0.000000000003	diff Force= 0.000000000004
6 input_example/GaAs.dat	Elapsed time(s)= 99.17	diff Utot= 0.000000000000	diff Force= 0.000000000000
7 input_example/Glycine.dat	Elapsed time(s)= 11.67	diff Utot= 0.000000000001	diff Force= 0.000000000000
8 input_example/Graphite4.dat	Elapsed time(s)= 7.99	diff Utot= 0.000000000006	diff Force= 0.000000000127
9 input_example/H2O-EF.dat	Elapsed time(s)= 8.70	diff Utot= 0.000000000000	diff Force= 0.000000000002
10 input_example/H2O.dat	Elapsed time(s)= 8.13	diff Utot= 0.000000000000	diff Force= 0.000000000011
11 input_example/HMn.dat	Elapsed time(s)= 29.00	diff Utot= 0.000000000131	diff Force= 0.000000000021
12 input_example/Methane.dat	Elapsed time(s)= 6.54	diff Utot= 0.000000000016	diff Force= 0.000000000001
13 input_example/Mol_MnO.dat	Elapsed time(s)= 17.34	diff Utot= 0.000000000370	diff Force= 0.000000000118
14 input_example/Ndia2.dat	Elapsed time(s)= 8.71	diff Utot= 0.000000000001	diff Force= 0.000000000000

Total elapsed time (s) 385.90

As for runtest, please refer the website:

http://www.openmx-square.org/openmx_man3.9/node17.html

Benchmark calculations of OpenMX Ver. 3.9 #3

An example of the results of runtest by PC2

WSL1: mpirun -np 6 ./openmx -runtest -nt 1

1	input_example/Benzene.dat	Elapsed time(s)=	5.45	diff Utot= 0.000000000040	diff Force= 0.000000000012
2	input_example/C60.dat	Elapsed time(s)=	24.39	diff Utot= 0.000000000000	diff Force= 0.000000000000
3	input_example/CO.dat	Elapsed time(s)=	13.21	diff Utot= 0.000000000096	diff Force= 0.000000000261
4	input_example/Cr2.dat	Elapsed time(s)=	9.92	diff Utot= 0.000000000907	diff Force= 0.000000000172
5	input_example/Crys-MnO.dat	Elapsed time(s)=	24.60	diff Utot= 0.000000000003	diff Force= 0.000000000005
6	input_example/GaAs.dat	Elapsed time(s)=	38.56	diff Utot= 0.000000000000	diff Force= 0.000000000000
7	input_example/Glycine.dat	Elapsed time(s)=	6.48	diff Utot= 0.000000000001	diff Force= 0.000000000000
8	input_example/Graphite4.dat	Elapsed time(s)=	4.32	diff Utot= 0.000000000006	diff Force= 0.000000000124
9	input_example/H2O-EF.dat	Elapsed time(s)=	4.99	diff Utot= 0.000000000000	diff Force= 0.000000000002
10	input_example/H2O.dat	Elapsed time(s)=	3.97	diff Utot= 0.000000000000	diff Force= 0.000000000011
11	input_example/HMn.dat	Elapsed time(s)=	20.27	diff Utot= 0.000000000131	diff Force= 0.000000000021
12	input_example/Methane.dat	Elapsed time(s)=	3.15	diff Utot= 0.000000000011	diff Force= 0.000000000001
13	input_example/Mol_MnO.dat	Elapsed time(s)=	10.51	diff Utot= 0.000000000370	diff Force= 0.000000000118
14	input_example/Ndia2.dat	Elapsed time(s)=	4.98	diff Utot= 0.000000000001	diff Force= 0.000000000000

Total elapsed time (s) 174.79

As for runtest, please refer the website:

http://www.openmx-square.org/openmx_man3.9/node17.html

Benchmark calculations of OpenMX Ver. 3.9 #4

The computation time for runtest is summarized below.

	WSL Version	MPI procs.	OMP threads:	Elapsed time (sec.)
PC1	1	4	1	385.90
	1	2	2	1805.65
	2	4	1	275.30
	2	2	2	275.55
PC2	1	6	1	174.79
	2	6	1	196.43

We see that the proper choice of WSL version depends on the PC. It is also confirmed that the computation time is comparable to those of the cluster machines shown in the following web.

http://www.openmx-square.org/openmx_man3.9/node17.html

If you have a PC with enough memory, you will be able to perform a variety of calculations.