

C++プログラミング開発環境の整備について

情報科学科 世木寛之

C++のプログラミング開発環境を WINDOWS パソコンに整備する方法はいくつか存在します。ご自分のパソコンに C++のプログラミング環境がないようでしたら、以下のいずれか 1 つで良いので、実施してプログラミング開発環境を構築してみてください。(これまで学科サーバーを利用していた方についてもプログラミング開発環境を構築してください。また、VisualStudio など下記の方法以外のプログラミング開発環境がある場合には、新たに開発環境を構築する必要はありません。) なお、不具合の原因になるので、WINDOWS のユーザー名は漢字・ひらがな・カタカナは使わないほうが望ましいと思います。例えば、「世木」ではなく、「segi」でユーザー名を作るということです。

- ① MobaXterm というアプリケーションをインストールする。
⇒別ファイルの MobaXterm インストール手順書を参照
- ② WINDOWS 10 であれば、linux をインストールする機能が存在するので、この機能を使う。
⇒1)windows update をかけて OS を最新の状態にする。
⇒2)タスクバーの検索ボックスで「機能の有効化」と入力・検索し、
「Windows の機能の有効化または無効化」を選択し、
「Windows Subsystem for Linux」をチェックする。
⇒3)Microsoft Store のホームページで ubuntu を検索し、
「Ubuntu18.04 LTS」を選択し、「入手」をクリックする。
(ちなみに私の 1 台のパソコンはこの入手から先に進むことができませんでした・・・)
⇒4)無事にインストールが完了したら、
WINDWOS タスクバー「スタート」の「U」の欄の「Ubuntu 18.04 LTS」をクリック
ターミナルが立ち上がる
⇒5)「sudo apt update」「sudo apt upgrade」「sudo apt install emacs25-lucid」
「sudo apt install emacs-mozc」「sudo apt install vim」「sudo apt install lv」
「sudo apt install dbus-x11」「sudo apt install gconf2」「sudo apt install p7zip-full」
「sudo apt install fonts-ipafont」「sudo apt install gcc」「sudo apt install g++」
「sudo apt install make」などをターミナルから実行
⇒6)Sakura エディタなどプログラムを作成するエディタを WINDWOS 側にインストール
⇒7)Windows のエクスプローラ上で「C:¥jouprou_work¥test.cpp」のファイルは、
Ubuntu 上では「/mnt/c/jouprou_work/test.cpp」に存在することを理解して、
プログラミングを行う。この際、作業ディレクトリ名やファイル名は漢字・ひらがな・カタカナは使わないほうが望ましい。
- ③ WINDOWS 11 でも、linux をインストールする機能が存在するので、この機能を使う。
⇒1)windows update をかけて OS を最新の状態にする。
⇒2)「スタート」をクリックして「コマンドプロンプト」と入力・検索し、
「管理者として実行」をクリックする。
⇒3)コマンドプロンプト上で、「wsl -l -o」を実行する。
デフォルトの「Ubuntu」で良ければ、「wsl --install」を実行する。
⇒4)無事にインストールが完了したら、WINDOWS を再起動する。
再起動後、WINDWOS タスクバー「スタート」の「U」の欄の「Ubuntu」をクリック

するとターミナルが立ち上がる。

- ⇒5) 「sudo apt update」「sudo apt upgrade」「sudo apt install emacs」
「sudo apt install lv」「sudo apt install gconf2」「sudo apt install p7zip-full」
「sudo apt install fonts-ipafont」「sudo apt install gcc」「sudo apt install g++」
「sudo apt install make」などをターミナルから実行
- ⇒6) Sakura エディタなどプログラムを作成するエディタを WINDWOS 側にインストール
- ⇒7) Windows のエクスプローラ上で「C:\joupro_work\test.cpp」のファイルは、
Ubuntu 上では「/mnt/c/joupro_work/test.cpp」に存在することを理解して、
プログラミングを行う。この際、作業ディレクトリ名やファイル名は漢字・ひらがな・カタカナは使わないほうが望ましい。

- ④ VirtualBox というアプリケーションをインストールし、ゲスト OS として Ubuntu 18.04 LTS を動かす。

⇒別ファイルの VirtualBox&Ubuntu インストール手順書を参照

- ⑤ VMware Workstation Player というアプリケーションをインストールし、ゲスト OS として Ubuntu 18.04 LTS を動かす。

- ⇒1) 「VMware Workstation 16 Player for Windows」をダウンロード・インストール
- ⇒2) 「<https://www.ubuntulinux.jp/News/ubuntu1804-ja-remix>」などから、
Ubuntu 18.04 LTS の日本語 Remix の ISO イメージを、
デスクトップなど自分が分かる場所にダウンロードする。
- ⇒3) インストールした「VMware Workstation 15 Player」を起動。
- ⇒4) 「新規仮想マシンの作成」をクリックし、
インストーラーディスクイメージファイルとして、
先ほどダウンロードした ISO イメージを指定し、「次へ」をクリック
- ⇒5) すべて記入して「次へ」をクリック
(ユーザー名は漢字・ひらがな・カタカナは使わないほうが望ましい)
- ⇒6) 特に変える必要がなければ「次へ」をクリック
- ⇒7) 特に変える必要がなければ「次へ」をクリック
- ⇒8) 特に変える必要がなければ「完了」をクリック
- ⇒9) インストールの途中で「VMware Tools」をインストールするか聞かれたら、
インストールするを選択。
また、管理者権限が必要かどうか聞かれたら、必要を選択しインストールを進める
- ⇒10) 右クリックして「端末を開く」を選択し、
「sudo apt update」「sudo apt install open-vm-tools-desktop」を実行する。
- ⇒11) 「sudo apt upgrade」「sudo apt install emacs25-lucid」
「sudo apt install emacs-mozc」「sudo apt install vim」「sudo apt install lv」
「sudo apt install dbus-x11」「sudo apt install gconf2」「sudo apt install p7zip-full」
「sudo apt install fonts-ipafont」「sudo apt install gcc」「sudo apt install g++」
「sudo apt install make」などを端末から実行。
- ⇒12) 画面左下「アプリケーションを表示する」から「設定」を選択し、
「プライバシー」の「画面ロック」を「OFF」に設定、
「電源管理」の「ブランクスクリーン」を「しない」に設定。
- ⇒13) Windows と Ubuntu の共有フォルダを設定する。
Windows 側にフォルダを作成する。(例えば C ドライブ直下に joupro_work を作成)
「VMware Workstation 15 Player」を起動し、「Ubuntu 64 ビット」を選択し、
「Player」をクリック、「管理」を選択し、「仮想マシンを設定」を選択し、
「オプション」タブを選択し、「共有フォルダ」を選択し、「常に有効」をクリック、

- 「追加」をクリックし、「参照」をクリックし、先ほど作成したフォルダを選択。
- ⇒14)仮想マシンを再生し、「`sudo mkdir /mnt/hgfs`」を端末から実行。
- さらに、「`sudo cp /etc/fstab /etc/fstab_org`」を端末から実行。
- 最後に、「`sudo emacs /etc/fstab`」を端末から実行し、最後の行に、
「`.host:/mnt/hgfs fuse.vmhgfs-fuse allow_other 0 0`」を追加する。
- ⇒15)Windows のエクスプローラ上で「`C:¥jopro_work¥test.cpp`」のファイルは、
Ubuntu 上では「`/mnt/hgfs/jopro_work/test.cpp`」に存在することを理解して、
プログラミングを行う。

以上