

Docker, WSL, Ubuntu

岡野浩三

2024 年 4 月 30 日

1 はじめに

オペレーティングシステム (OS) の演習授業では Ubuntu を実際のオペレーティングシステムとして使いますが、その演習で OS の設定をいろいろ変更する予定です。このように OS をいろいろ変更してみる場合、仮想機械上で処理するほうがいろいろと安全ですし、気兼ねなくできる利点があります。この授業ではいままでその目的を達成するために過去においては Virtual Box を用いてきました。さらに、2021 年度からは Docker Desktop を用いてきました。2023 年からは Windows 11 環境の普及と Docker Desktop の有償化の流れを受けて、WSL 上での仮想 Ubuntu を使うことにします。

この資料では参考のため Docker の説明を 2 節で少しだけおこない、WSL 上での Ubuntu イメージの導入を説明します。

2 Docker

最近の Intel の CPU は仮想 PC を容易に実行できる命令セットが用意されており、Intel 系の PC を用いる OS では仮想 PC をサポートするソフトウェアを動作させることができる。例えば VirtualBox はそのような仮想 PC をサポートするソースソフトウェアの 1 つであり、オープンソースソフトウェアでもある。

仮想 PC は物理実態のある PC 上で複数の PC を仮想的に動かす機構である。この機構を用いると例えば、あなたのノート PC 上で複数の LinuxOS を簡単に動かすことができる。この機構はネットワーククラウドを構成するサーバ上でも広く用いられおり、安定してかつエコに動作するサーバ群を提供する基盤となっている。

最近では仮想コンテナ技術が発展している。この仮想コンテナ技術を容易に実現できるツールとして Docker が存在する。

Docker は Linux 上でコンテナと呼ばれる小さな独立した実行環境を提供するシステムである。コンテナは他のコンテナとは独立して実行されるのでとりわけ共用ライブラリの使用などにおいて相互の干渉が起こりにくく、システム障害などにつよく、また、VirtualBox のような旧来の仮想環境にくらべて軽量という特徴を持つ。

したがって Windows や MacOS からは直接 Docker の機構を使うことができないが、間に Linux の仮想実行環境を 1 層分置くことにより Windows からも Docker 機能を使えるようにしている。

Docker を Windows 上で容易に扱えるようにしたパッケージが Docker Desktop である。当初は無償で利用可能であったため、広く普及したが残念ながら 2022 年 1 月より有料ライセンスとなった(個人での利用は無料である)。

なお Docker 自体は引き続き無料で使用できる。有償化したのは Docker Desktop である。

一方、WSL (Windows Subsystem for Linux) は Windows 上で Linux を仮想機械を動かす機構であり、この数年で Windows の標準機能として安定して利用できるようになってきてる。WSL を用いることにより Docker Desktop より、より、お手軽に Linux の仮想環境を構築できる。

以下では、WSL を用いた Ubuntu の環境導入について説明する。

3 準備 ホストの BIOS 仮想化

この資料ではホスト(あなたのノート PC)の OS が Windows11 であつ、CPU が比較的最新の Intel 製であることを仮定する。

Macbook の場合は Macbook 上で Linux を仮想機械として動かす方法について自力で調べることに。なおこの授業の大半の演習は MacOS の Terminal を用いて MacOS の標準シェル (Zsh) から実行可能である。

この資料では Windows 11 であることを前提に説明する。

まず bios の設定を確認し、Intel Virtualization Technology の機能を enabled にする。bios の設定方法は各自の PC で異なるので頑張って設定すること。通常リブートをし、Windows が立ち上がるまでに del キーや f2 キーなど特定のキーを押すことで bios 設定にはいることができる。

多くの学生が使用していると思われる (Let's note) の場合、起動直後の Panasonic のロゴが出てるときに F2 連打で BIOS モードになるようです。また、Let's note の BIOS では、詳細の欄にある CPU 情報で Intel Virtualization Technology の有効/無効を切り替えできるようです。なお、Intel Virtualization Technology はデフォルトで有効になっているようです。

4 WSL(2) のインストール

この資料は以下の記事を参考にしてている。 <https://eng-blog.iij.ad.jp/archives/14205>

具体的な手順は以下で説明する。

5 Windows Terminal のインストール

Windows Terminal が入っていないければインストールする。Windows Store から簡単にインストールできるので導入手順は略する。

以降の作業を行う場合、管理者権限で Windows Terminal を起動することに注意する。

6 WSL のアップデート

Windows11 では WSL も version2 が標準で入っているはずである。

管理者権限で起動した Windows Terminal の Power shell 上で

```
wsl --update
```

を実行する。これにより WSL の環境が最新版にアップデートされる。もし、アップデートが実行されない場合は WSL が中途半端にしか入っていない可能性がある。

実行されない（コマンドとして認識されない）場合、管理者権限で Command Promopt を起動し

```
wsl --install
```

を実行する。これにより WSL の環境のインストールと最新の Ubuntu のインストール作業が一気に終わるようである。

Powershell 上で

```
wsl --update
```

が成功した場合は次に進む。

```
wsl --install
```

で Ubuntu のインストールまで進む場合は次節の

```
sudo apt-get update
sudo apt-get install vim
sudo apt-get install gcc make perl
sudo apt-get install gdb
sudo apt-get install bc
sudo apt-get install x11-apps
sudo apt-get install man
```

から作業を進める。なおこのときにインストールされる ubuntu のバージョンが Ubuntu-22.04 でない場合は次節の

```
wsl --install -d Ubuntu-22.04
```

から進めるとよい。

7 Ubuntu のインストール

次に WSL 上で動作する仮想環境 Linux として Ubuntu をインストールする。

```
wsl --install -d Ubuntu
```

でもよいがここでは

```
wsl --install -d Ubuntu-22.04
```

のようにバージョン込で実行することにする。

インストール中に username を聞かれるので自分の名前などを ascii で与える (例 okano)。また ubuntu 用のパスワードも入れる。パスワードは2回入力が必要とめられる。また Linux ではパスワードとして入力した文字が改行以外エコーしないので注意すること (驚かず進めること)。

インストールが終わり Ubuntu のターミナルが表示されたら設定をいったん終了するために

```
exit
```

を実行する。

さらに Windows Terminal も一旦終了し、再度起動する。そうすると

Windows Terminal の上端メニュー右端の下むきの V をクリックすると Ubuntu のターミナルのアイコンが表示されるようになる。それを起動すると Ubuntu がターミナルから使えるようになる。

新規にインストールした Ubuntu はプログラミング用のコマンド類は入っていないので以下のコマンドを実行して追加インストールする。

```
sudo apt-get update
sudo apt-get install vim
sudo apt-get install gcc make perl
sudo apt-get install gdb
sudo apt-get install bc
sudo apt install x11-apps
sudo apt-get install man
```

sudo を最初に実行したときに認証が必要とめられるので最初に設定したパスワードを入れる。

最近はインタラクティブな用途では apt-get の代わりに apt を使うことが推奨されつつあるようであり上記のコマンドにおいて apt-get の代わりに apt を使用してもよい。

なお x11-apps の実行にはそれなりの時間を要する。

8 WSL 上での仮想環境の管理

WSL 上で作成した仮想環境 (上述で作成した Ubuntu など) について、削除、リセット、export, import など様々な管理を簡単に行うことができる。この授業などで OS の設定をあやまって削除したり変更したりして動作がおかしくなったときに、このような操作をすることで容易に元に戻すことができる。当然ながら、事前にバックアップ (スナップショット) を作成しておかないと、スナップショットの復元によるお手軽な復旧はできない。適切なタイミングでスナップショットを随時、作成しておくことを強く勧める。

以下の内容などが参考になるであろう。

<https://qiita.com/PoodleMaster/items/b54db3608c4d343d27c4>

9 ホスト PC とのファイル共有

演習で作成するファイルはホスト PC（ホスト OS）と共有できることが望ましい。デフォルトで Windows の各 SSD は /mnt 以下にマウントされている。マウントの状況は Ubuntu から df コマンドを実行すると確認できる。

```
okano@DESKTOP-E2J5VJH : /mnt/d/Docker/work/03h$ df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
none	8149980	4	8149976	1%	/mnt/wsl
none	975477096	270117476	705359620	28%	/usr/lib/wsl/drivers
none	8149980	0	8149980	0%	/usr/lib/wsl/lib
/dev/sdc	1055762868	1642828	1000416568	1%	/
none	8149980	84	8149896	1%	/mnt/wslg
rootfs	8146736	1936	8144800	1%	/init
none	8149980	20	8149960	1%	/run
none	8149980	0	8149980	0%	/run/lock
none	8149980	0	8149980	0%	/run/shm
none	8149980	0	8149980	0%	/run/user
none	8149980	72	8149908	1%	/mnt/wslg/versions.txt
none	8149980	72	8149908	1%	/mnt/wslg/doc
drvfs	975477096	270117476	705359620	28%	/mnt/c
drvfs	1953497084	904081784	1049415300	47%	/mnt/d
drvfs	488369148	113372	488255776	1%	/mnt/e
drvfs	51196	12520	38676	25%	/mnt/g
drvfs	2930265596	1358479704	1571785892	47%	/mnt/l
drvfs	975477096	305385460	670091636	32%	/mnt/n

```
okano@DESKTOP-E2J5VJH : /mnt/d/Docker/work/03h$
```

10 正常にインストールされたかの確認

以下の 2 つのコマンドを実行し、OS のリリースバージョンと現ログインユーザーを確認する。

```
$ cat /etc/os-release
$ who
```