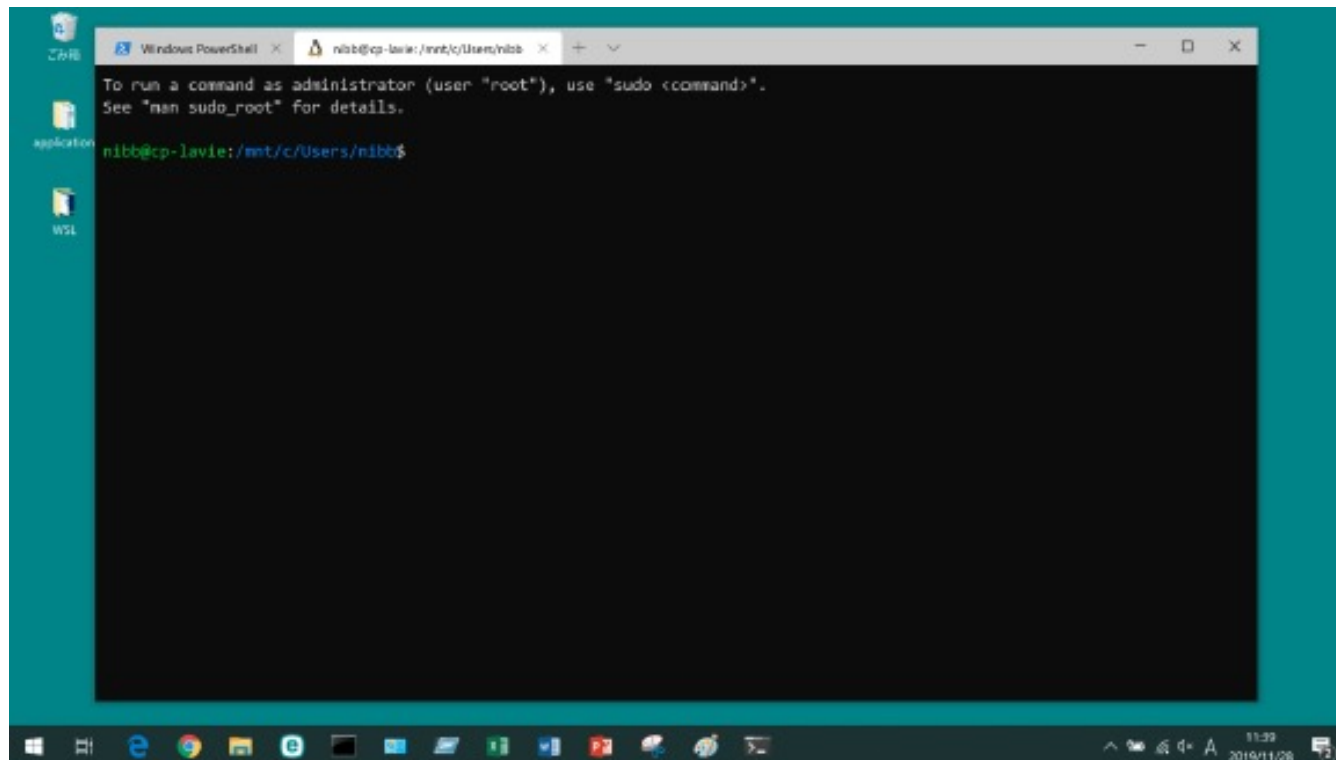


UNIX環境の構築(Windows編)

Windows環境でUNIXコマンドを使用するためには



概要

- ▶ Windowsは非Unix系OSなので、そのままではUnixコマンドを使用できない。
- ▶ 以前では、Windows上でUnixコマンドを使用するために専用アプリケーションのインストールが必須であった。それでも、Unix用に作成されたソフトウェアをすべて使用することはできず、やや不便であった。
- ▶ 近年では、Unix用に作成されたソフトウェアをWindows上でも使用することができる環境が整いつつある。
- ▶ 本資料はWindows上で、Unix用に作成されたソフトウェアを実行できる状態にまで環境を構築する一例を紹介する。

方法の概要

- ▶ Windows上でUnixコマンドを扱うためには、大きく分けて3つの方法がある。
- ▶ 1) Windows標準ツールの使用
...Windows Subsystem for Linuxを使う。
- ▶ 2) 他のマシンへのログイン
...ターミナルソフト(TeraTerm等)を使う。
- ▶ 3) 専用アプリケーションの使用
..."Cygwin"が有名。
- ▶ 今回はWindows Subsystem for Linuxの導入を取り扱う。

Windows Subsystem for Linux

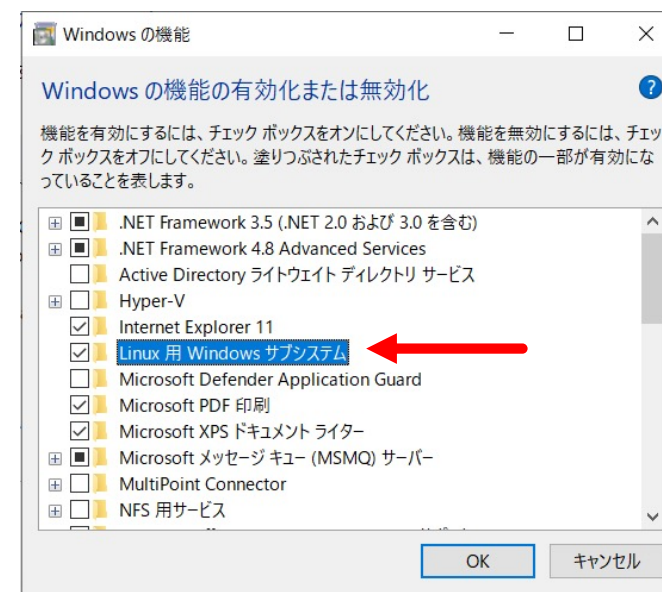
- ▶ Windows10でLinux(Ubuntu)コマンドを使える環境。
- ▶ Windows10に標準で搭載されている。
- ▶ ただし初期状態では使えないため、後述する手順で設定を変更する必要がある。
- ▶ Windows Subsystem for Linuxで提供されるものはCUIのみ。
(マウスで操作せず、キーボードでコマンドを入力することで操作)
- ▶ WindowsでインストールできなかったLinux用ソフトウェアを、Ubuntuを利用することで、Windows端末にインストールできる。

Windows側でのLinux Subsystemの有効化



- ▶ スタートボタンを右クリックし、「アプリと機能」を選択
- ▶ 設定画面の右上にある「プログラムと機能」を選択

Windows側でのLinux Subsystemの有効化



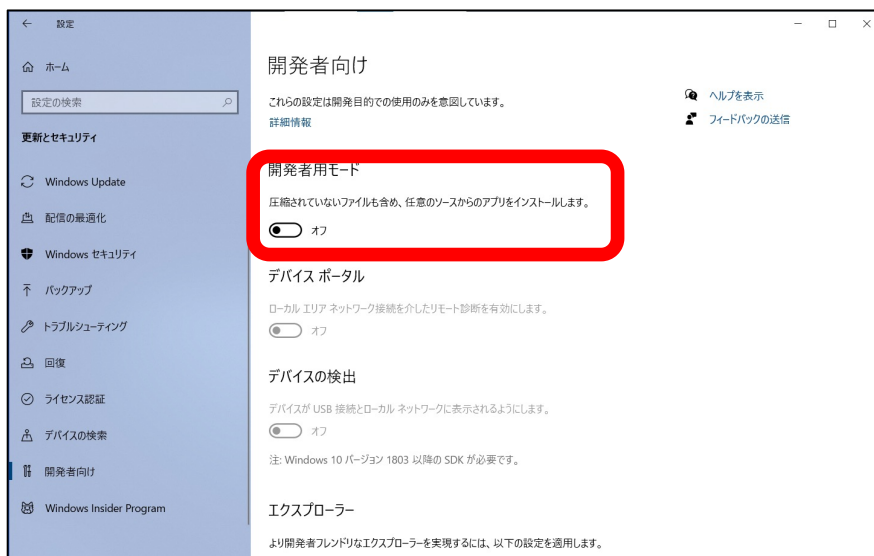
- ▶ 左側の一覧から、「Windowsの機能の有効化または無効化」を選択
- ▶ 「Linux用 Windows サブシステム」を探し、チェックを入れる

Windows側でのLinux Subsystemの有効化



- ▶ インストールが開始されるためしばらく待つ。
- ▶ 終了したら指示に従い、PCを再起動させる。

開発者モードの有効化



開発者向け機能を使う

開発者モードをオンにすると、Microsoft Store 以外からアプリをインストールして実行する場合などに、デバイスと個人データがセキュリティ上のリスクにさらされる可能性や、デバイスに問題が起こる可能性があります。

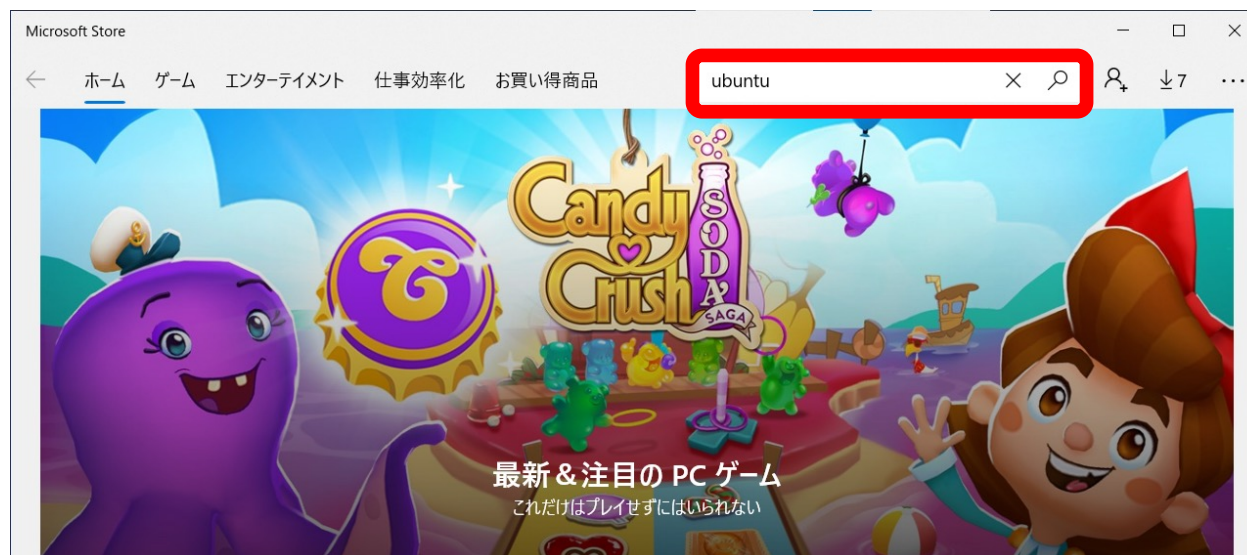
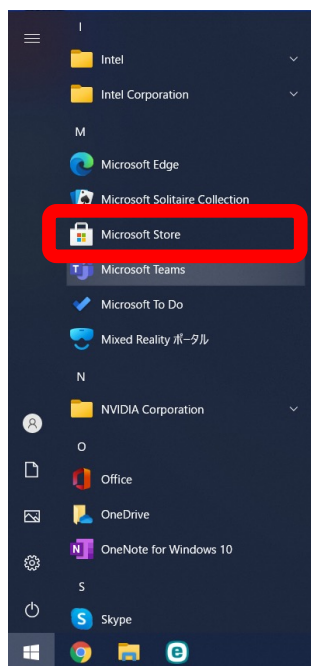
開発者モードをオンにしますか？

はい

いいえ

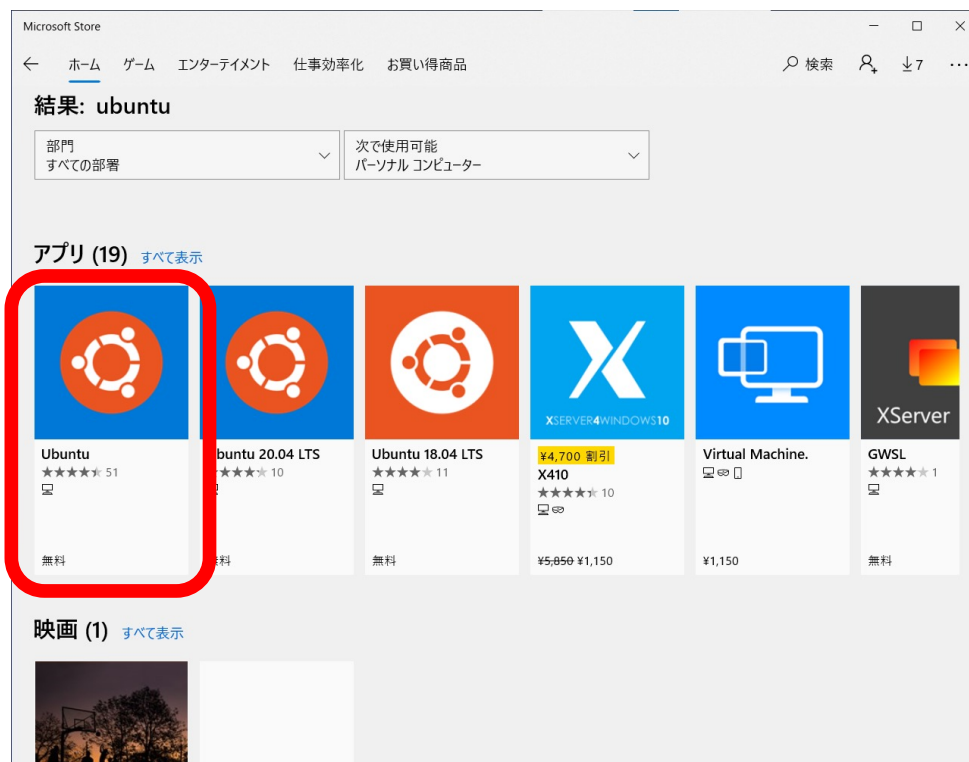
- ▶ 再起動後、スタートメニューから「設定」→「更新とセキュリティ」へ進む。
- ▶ 左側のメニューから「開発者向け」を選択し、「開発者モード」にチェックを入れる。
- ▶ 注意書きが出るが、そのまま「はい」を選択して進める。

Ubuntuのインストール



- ▶ スタートメニューから「Microsoft Store」を選択。
- ▶ 右上の検索欄から「Ubuntu」と検索

Ubuntuのインストール



- ▶ 検索結果から「Ubuntu」を選択
- ▶ 3つのUbuntuがあるが、基本的には「LTS(Long Term Support)と書かれていないもの」を選べばよい。
- ▶ このバージョンで動かしたいという明確な理由がある場合にはLTSを使用する。

Ubuntuのインストール



- ▶ 「入手」を選択するとインストールが開始される。
- ▶ しばらく待つと「起動」が表示される。

Ubuntuの起動

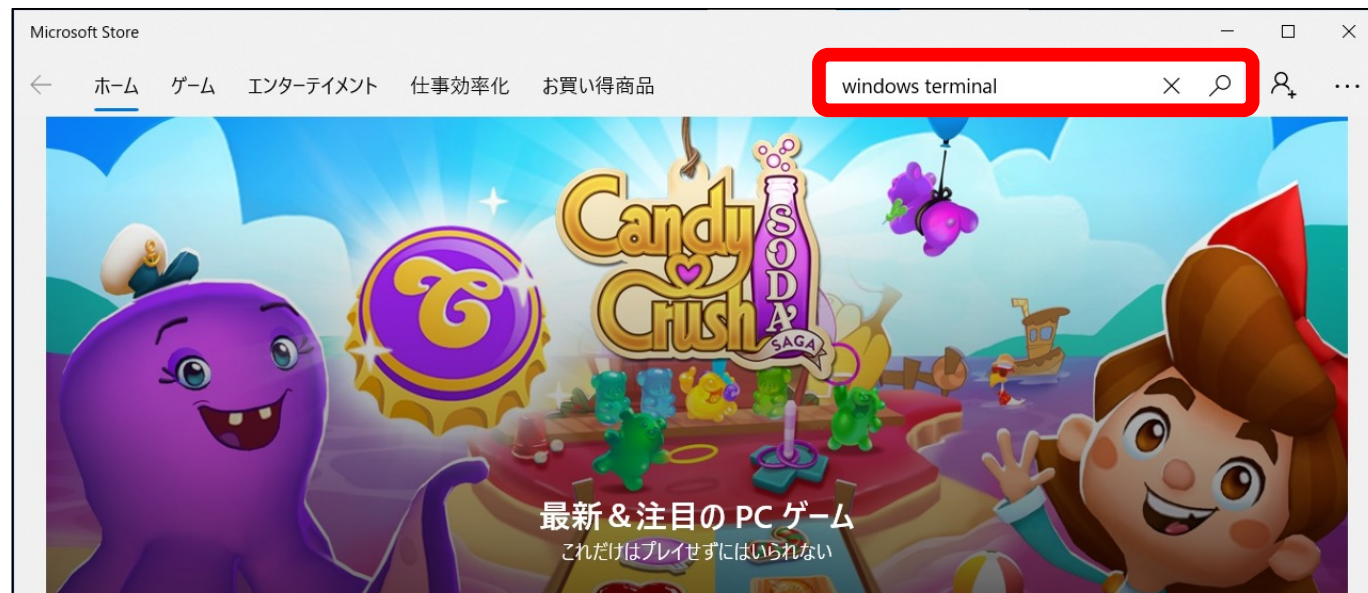
```
nibb@nibb-PC: ~  
Installing, this may take a few minutes...  
Please create a default UNIX user account. The username does not need to match your Windows username.  
For more information visit: https://aka.ms/wslusers  
Enter new UNIX username: nibb  
New password:  
Retype new password:  
passwd: 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 20.04.2 LTS (GNU/Linux 4.4.0-19041-Microsoft x86_64)  
  
 * Documentation:  https://help.ubuntu.com  
 * Management:    https://landscape.canonical.com  
 * Support:       https://ubuntu.com/advantage  
  
System information as of Mon Aug 23 16:49:05 JST 2021  
  
System load:  0.52   Processes:            7  
Usage of /home: unknown   Users logged in:    0  
Memory usage: 14%    IPv4 address for wifi0: 133.48.63.160  
Swap usage:   0%  
  
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.
```

- ▶ Ubuntuを起動し、しばらく待つと「Unixで使用するユーザネーム」(例ではnibb)と、「パスワード」の設定が求められる。
- ▶ パスワードを入力する際、キーボードで入力しても画面上で確認することはできない。
- ▶ 画面上に何も表示されないが、正常に入力されているため、タイプミスしないようにパスワードを設定する。

Windows Terminalの導入

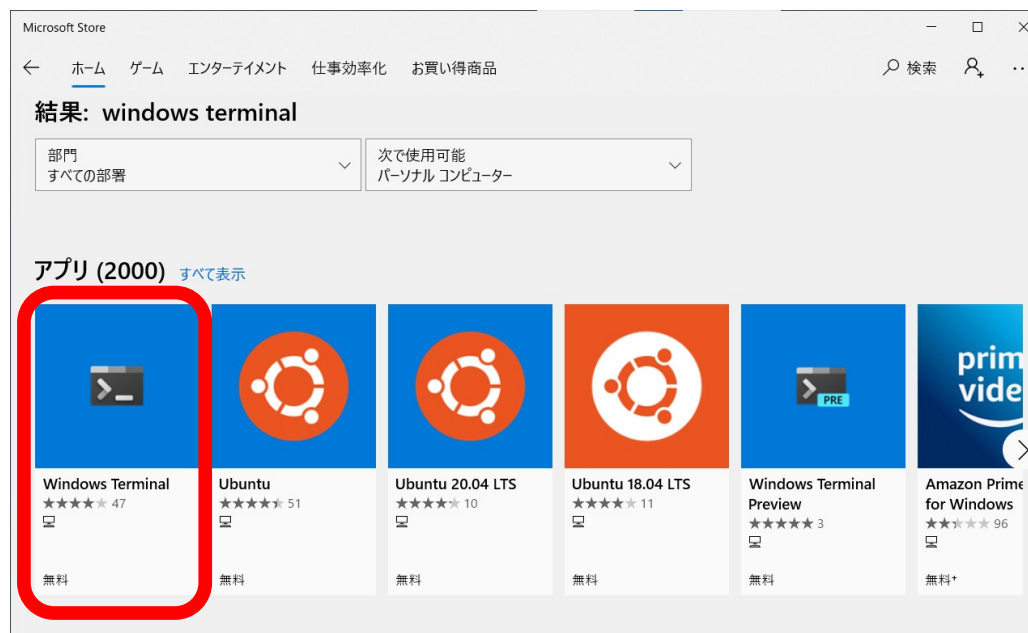
- ▶ UnixコマンドやソフトウェアはUbuntuを用いて実行する。
- ▶ 今回インストールしたUbuntuをそのまま利用してもよいが、シェルの起動時のホームディレクトリが「Ubuntuでの」ホームディレクトリになるので、Windowsのユーザーディレクトリとは異なるパスになる。
- ▶ 「Windows Terminal」を利用することで、起動時のホームディレクトリをWindowsのユーザーディレクトリにすることができる。
これにより、直感的にWindows上のファイルにアクセスすることができる。
- ▶ 今回はこのWindows Terminalを利用して、Ubuntu単体で利用する場合より、さらに利用しやすい環境を整える。

Windows Terminalの導入



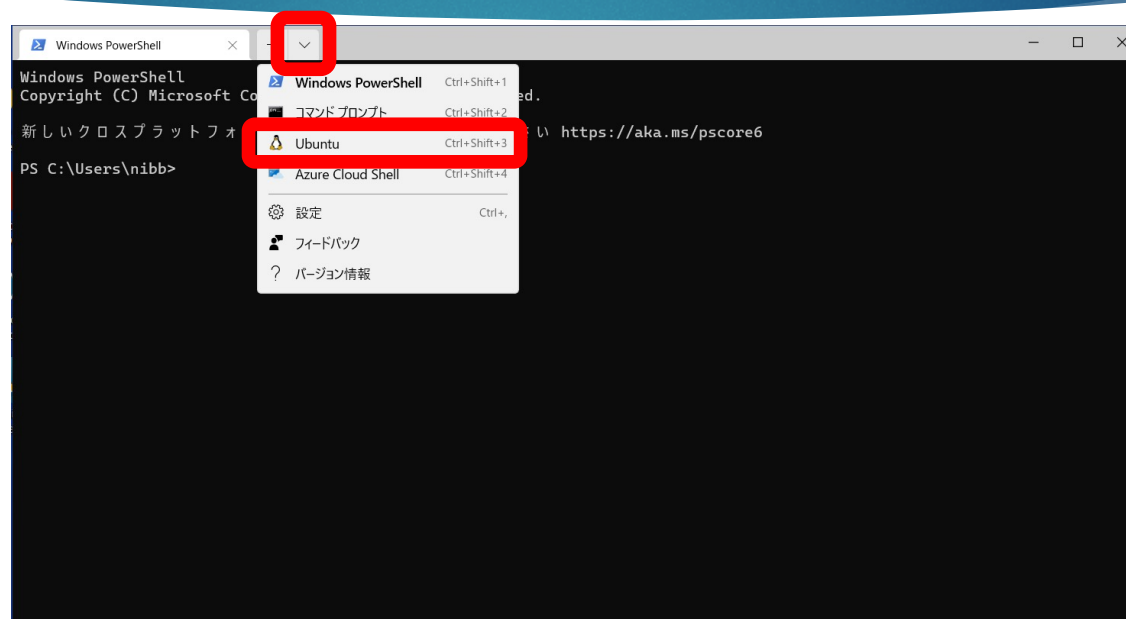
- ▶ スタートメニューから「Microsoft Store」を選択。
- ▶ 右上の検索欄から「Windows Terminal」と検索

Windows Terminalの導入



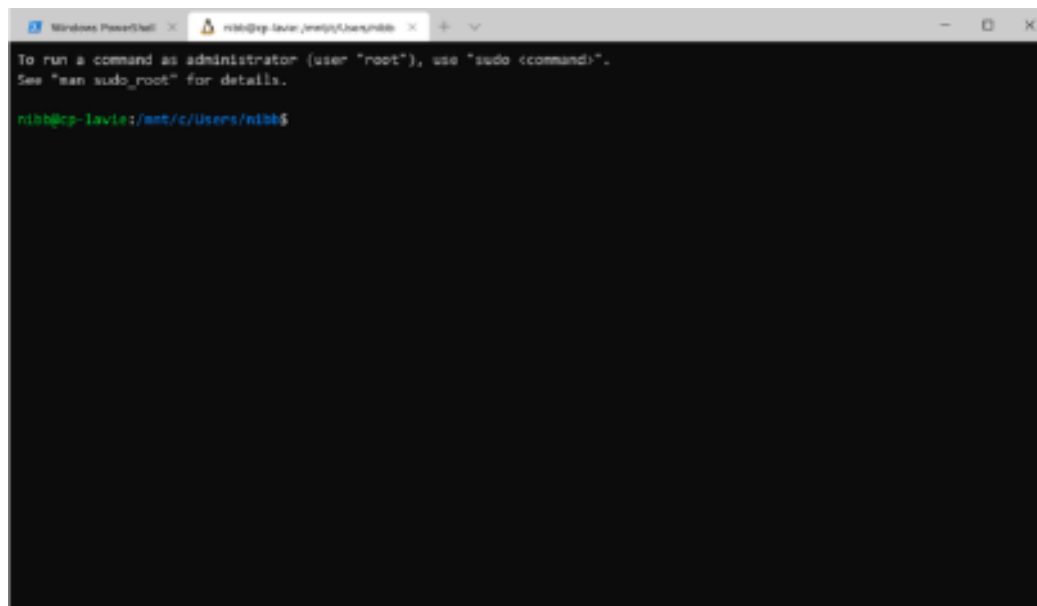
- ▶ 「Windows Terminal」を選択
- ▶ 「Ubuntu」同様にインストールし、起動させる

Windows Terminal上でのUbuntuの利用



- ▶ Windows Terminalを起動すると、Windows PowerShellが起動する。
- ▶ Windows PowerShellの右側をクリックすると、どのシェルを使うかと聞かれるため、「Ubuntu」を選択する。

Windows Terminal上でのUbuntuの利用

A screenshot of a Windows Terminal window. The title bar shows 'Windows PowerShell' and a tab for 'nibb@cp-lavie: /mnt/c/Users/nibb'. The terminal content displays instructions for running commands as administrator, followed by the prompt 'nibb@cp-lavie: /mnt/c/Users/nibb\$'.

- ▶ これでWindows Terminal上でUbuntuを利用することができるようになった。
- ▶ GITCでの講義ではmacOSのターミナルというものを利用することになるが、Windowsで操作する場合はこのUbuntuを用いることで同様の操作が可能になる。

apt コマンド

- ▶ Ubuntu 環境ではソフトウェアの導入を単純化するパッケージ管理をapt (Advanced Package Tool) コマンドで行います。
- ▶ パッケージのインストールや削除には管理者権限が必要です。
sudo コマンドと併用します。
例) `sudo apt install libncurses5-dev // libncurses5-dev のインストール`
- ▶ aptコマンド自体はプリインストールです。
- ▶ トレーニングコース前に次ページ以降の で囲っているコマンドをUbuntuで実行しておいてください。

apt の更新

- ▶ Ubuntuを起動した時点では、apt でインストールできるパッケージ一覧(インデックス)が古いままです。まずはこれを更新します。

```
sudo apt update //インデックスの更新
```

- ▶ インデックスを更新したあと、インストール済みのパッケージを更新するには以下のコマンドを使います。

```
sudo apt upgrade //インストール済みのパッケージを一括更新
```

- ▶ Ubuntu自体の更新は以下のコマンドで行います。

```
sudo apt dist-upgrade
```

apt コマンドの一例

- ▶ よく使用するaptコマンドの例です。

`apt list --installed` // インストール済みのパッケージ一覧を表示します。

`apt search name` // パッケージを名前で検索します。（部分一致）

`apt show package-name` // パッケージの説明を表示します。

`sudo apt install package-name` // パッケージをインストールします。

`sudo apt remove package-name` // パッケージを削除します。設定ファイルは維持されます。

`sudo apt purge package-name` // パッケージを完全削除します。

apt-file

- ▶ ソースからmakeする際に、「xxx.hが無い」といったエラーが表示されることがあります。このような場合、そのライブラリがあるパッケージを検索する必要があります。apt search ではパッケージ名しか検索できないので、apt-file コマンドを使用します。

```
sudo apt install apt-file // apt-fileのインストール  
sudo apt update // インデックス更新
```

- ▶ 準備ができたなら、apt-file を実行します。

```
apt-file search library-name // 名前のファイルが含まれるパッケージを検索
```

apt-file (例)

- ▶ make時に以下のエラーが出た。

```
make[1]: Entering directory '/mnt/c/Users/username/Desktop/samtools-1.11/htslib-1.11' gcc -g -Wall -O2 -fvisibility=hidden -l. -c -o cram/cram_io.o cram/cram_io.c  
cram/cram_io.c:57:10: fatal error: lzma.h: No such file or directory  
#include <lzma.h>
```

... lzma.h が無いので、lzma.hをインストールできるパッケージを探す。

- ▶ apt-file search lzma.h

```
$ apt-file search lzma.h  
libboost1.65-dev: /usr/include/boost/iostreams/filter/lzma.hpp  
libbotan-2-dev: /usr/include/botan-2/botan/lzma.h  
liblzma-dev: /usr/include/lzma.h  
... (以下略)
```

... sudo apt install liblzma-dev を実行。

2) 他のマシンへのログイン

- ▶ Windows環境のPCから、ネットワークを介してUNIX環境が準備されているマシン(サーバ)にログインして、その環境を使用する方法。
- ▶ ターミナルソフトウェアが使われる。
- ▶ TeraTerm(<https://ja.osdn.net/projects/term2/>)
- ▶ ログインする先のサーバがなければ使用することができない。
(ローカル環境では使用できない)

3) 専用アプリケーション

- ▶ Cygwin(<https://www.Cygwin.com>)
- ▶ Cygwinは、Windowsにインストールして使用するアプリケーションで、手軽にUNIX環境を実現できる。
- ▶ 広く使われているため、日本語でのドキュメントも存在する。
- ▶ UNIX系のプログラムを自力でコンパイル可能。
- ▶ 新しくソフトウェアをインストールする場合、パッケージ管理コマンド(yum, rpm等) は使用できない。