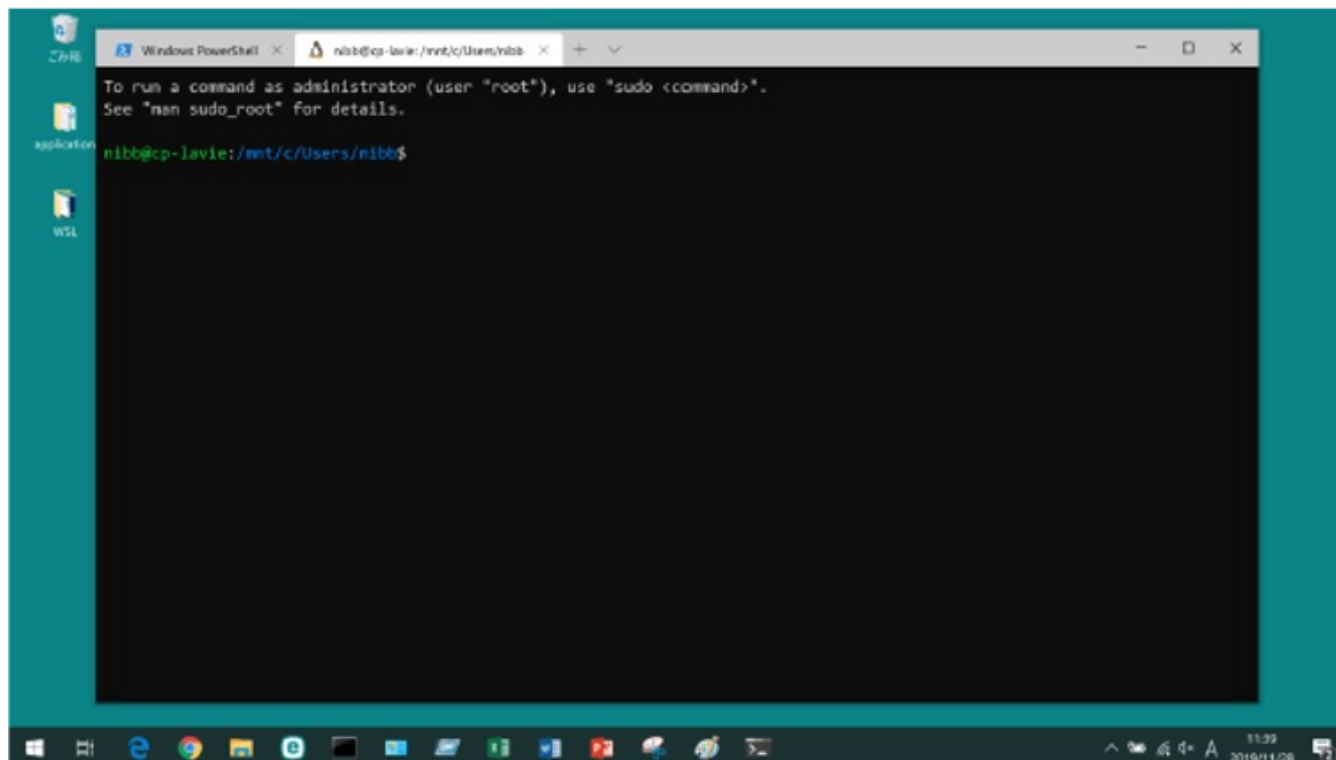


# UNIX環境の構築(Windows11編)

Windows環境でUNIXコマンドを使用するためには



作成 2022/12

# 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Windows11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。

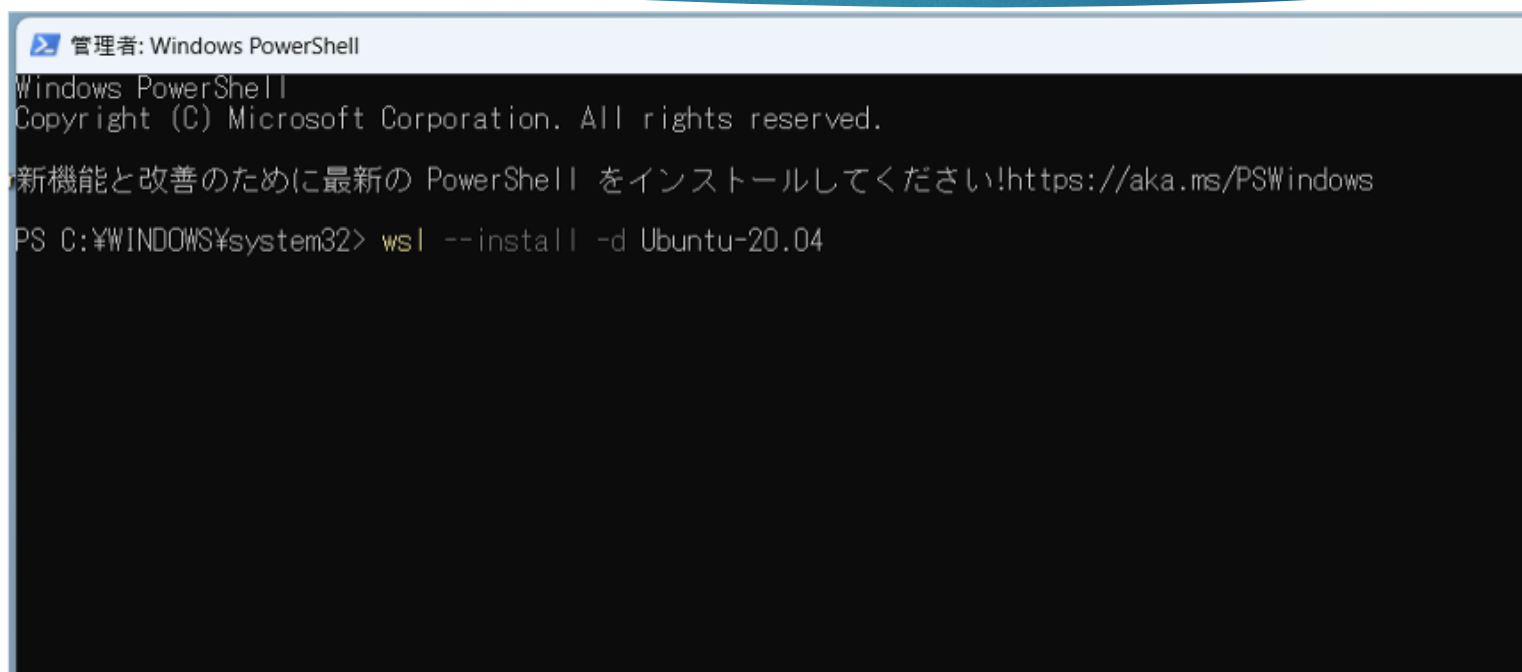
# 方法の概要

- ▶ 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

- ▶ Windows11でLinux(Ubuntu)コマンドを使える環境。
- ▶ Windows11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。
- ▶ 今回はUbuntu というLinux ディストリビューションを使用します

# Windows Subsystem for Linuxの導入

A screenshot of a Windows PowerShell terminal window. The title bar reads '管理者: Windows PowerShell'. The terminal text includes the Windows PowerShell copyright notice, a message about installing the latest PowerShell, and the command 'wsl --install -d Ubuntu-20.04' entered at the prompt.

```
管理者: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新機能と改善のために最新の PowerShell をインストールしてください!https://aka.ms/PSWindows
PS C:\WINDOWS\system32> wsl --install -d Ubuntu-20.04
```

起動したPowershell上に

**wsl --install -d Ubuntu-20.04** と入力し、エンターキーを押す

# Windows Subsystem for Linuxの導入

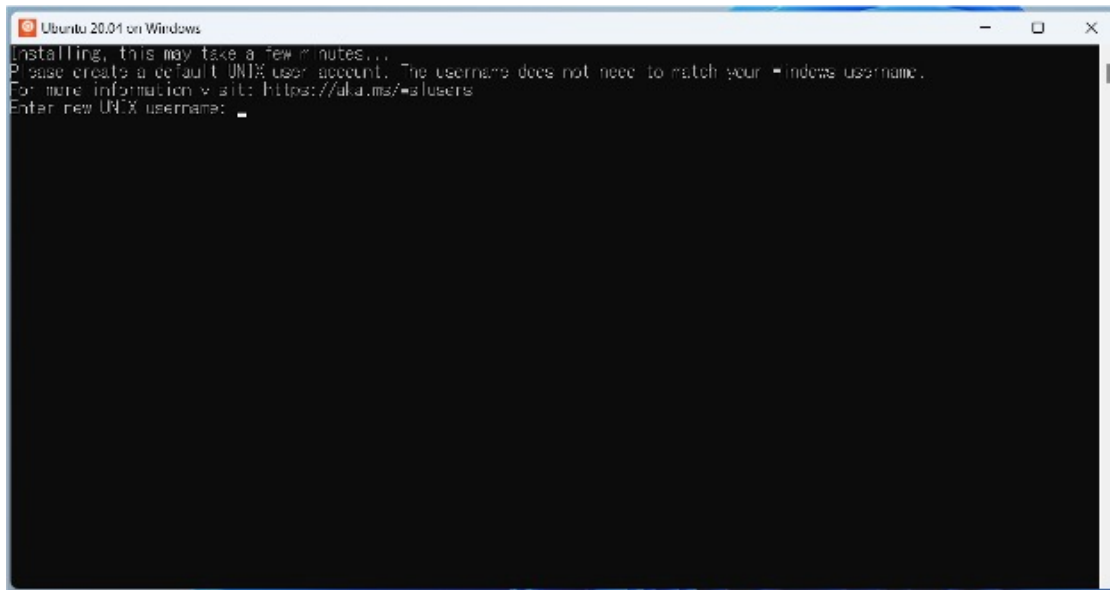
```
管理者: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新機能と改善のために最新の PowerShell をインストールしてください!https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl --install -d Ubuntu-20.04
インストール中: 仮想マシン プラットフォーム
仮想マシン プラットフォーム はインストールされました。
インストール中: Linux 用 Windows サブシステム
Linux 用 Windows サブシステム はインストールされました。
インストール中: Ubuntu 20.04 LTS
Ubuntu 20.04 LTS はインストールされました。
要求された操作は正常に終了しました。変更を有効にするには、システムを再起動する必要があります。
PS C:\WINDOWS\system32>
```

wslコマンドの実行が終わるとPC再起動を促されるため  
PCを再起動する

# Ubuntuの起動



- ▶ PCを再起動すると、自動的にUbuntuが起動する
- ▶ 「Unix(Ubuntu)で使用するユーザネーム」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。

# rootについて

- ▶ 前スライドではユーザ名とパスワードをこちらで指定した。
- ▶ Ubuntuを初めて起動した際に指定するユーザは"root"と呼ばれる機能をもつ。
- ▶ root...管理者ユーザ・特権ユーザのこと
- ▶ windowsでいう管理者アカウントに近い
- ▶ この機能を持つユーザは"万能でありなんでもできる"状態になる。  
そのためファイルの保存や削除はもちろん、端末のアプリケーションや機能の追加や削除もできるアカウントになる。
- ▶ こちらで設定したユーザ・パスワードの管理には十分ご注意ください。

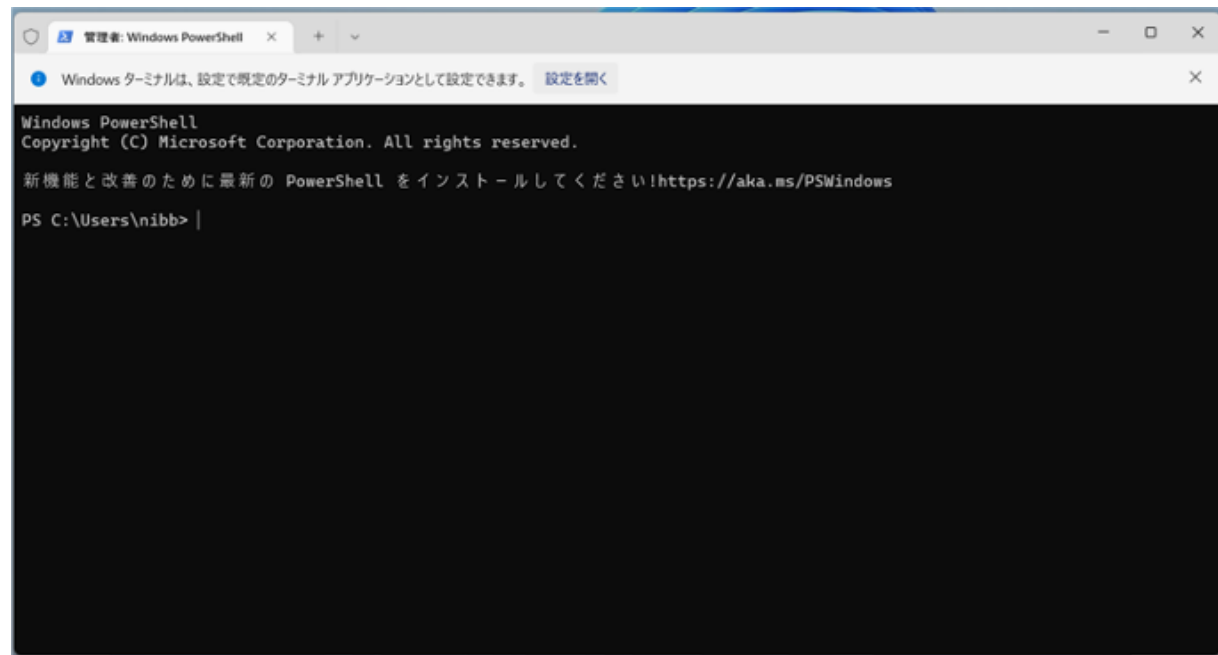
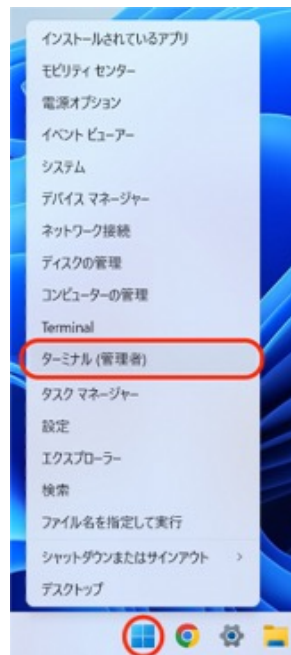




# Windows Terminalの導入

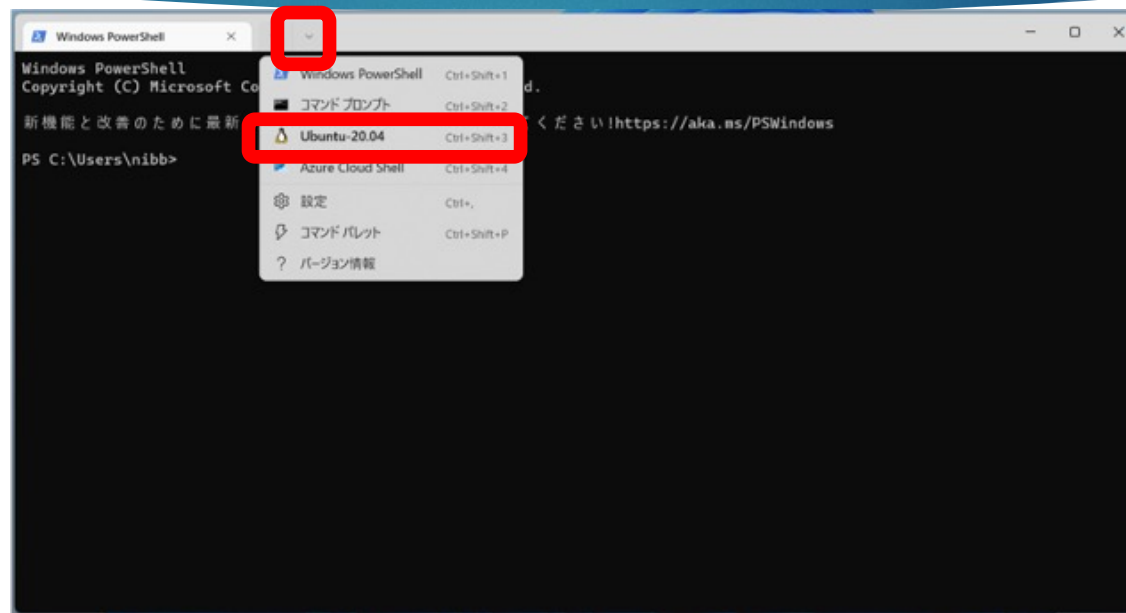
- ▶ UnixコマンドやソフトウェアはUbuntuを用いて実行する。
- ▶ 今回インストールしたUbuntuをそのまま利用してもよいが、シェルの起動時のホームディレクトリが「Ubuntuでの」ホームディレクトリになるので、Windowsのユーザーディレクトリとは異なるパスになる。
- ▶ 「Windows Terminal」を利用することで、起動時のホームディレクトリをWindowsのユーザーディレクトリにすることができる。  
これにより、直感的にWindows上のファイルにアクセスすることができる。
- ▶ 今回はこのWindows Terminalを利用して、Ubuntu単体で利用す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

# Windows Terminalの起動



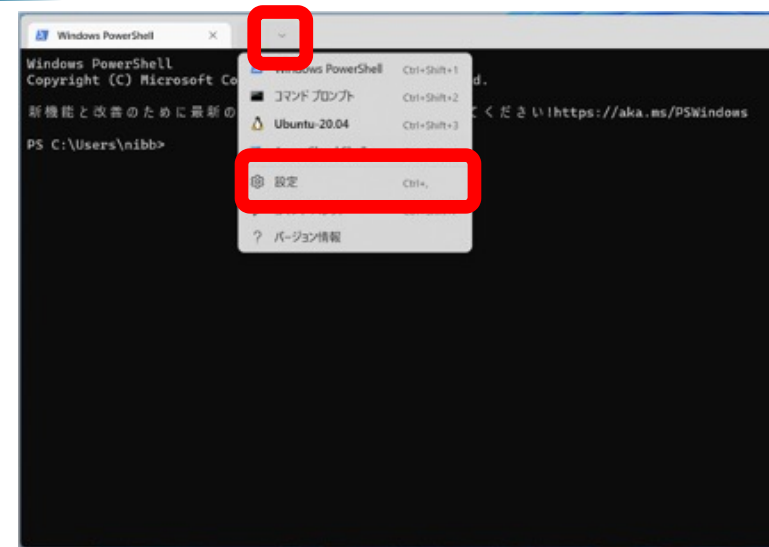
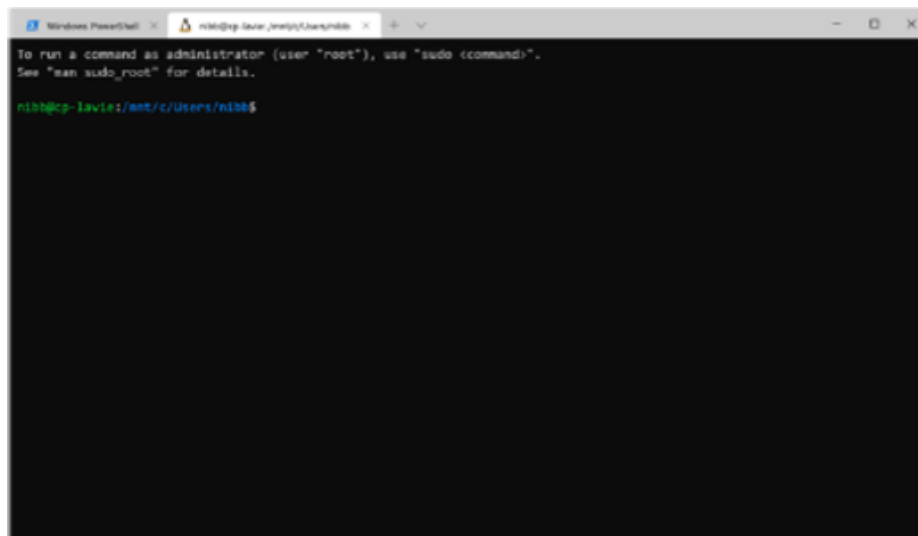
スタートアイコンを右クリックし、  
ターミナル(管理者) を選択するとWindows Terminalが起動する

# 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



- ▶ 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。
- ▶ GISC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。
- ▶ 次スライドでより使用しやすいように設定を変更する。Windows Terminalの"設定"を開く

# Windows Terminal上でのUbuntuの利用



- ▶ "スタートアップ" から "既定のプロファイル" をUbuntuに変更
- ▶ "Ubuntu" から "開始ディレクトリ" をDesktopに変更 最後に右下の変更を選択
- ▶ この2点を変更することで次回Windows Terminalを開いた時にDesktopから操作が可能になります

# 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/ttssh2/>)
- ▶ 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。
- ▶ 今ではほとんど使われない
- ▶ 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。