



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

作成 2025/07



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 (WSL) を利用
2. 他のマシンへのログイン: ターミナルソフト (TeraTermなど) を利用
3. 専用アプリケーションの使用: "Cygwin"など

本資料では、Windows Subsystem for Linux (WSL) の導入について解説する。



Windows Subsystem for Linux

Windows Subsystem for Linux (WSL) は、Windows 11上でLinux (Ubuntu) コマンドを使用できる環境である。Windows 11に標準で搭載されているが、初期状態では使用できないため、後述の手順で設定を変更する必要がある。

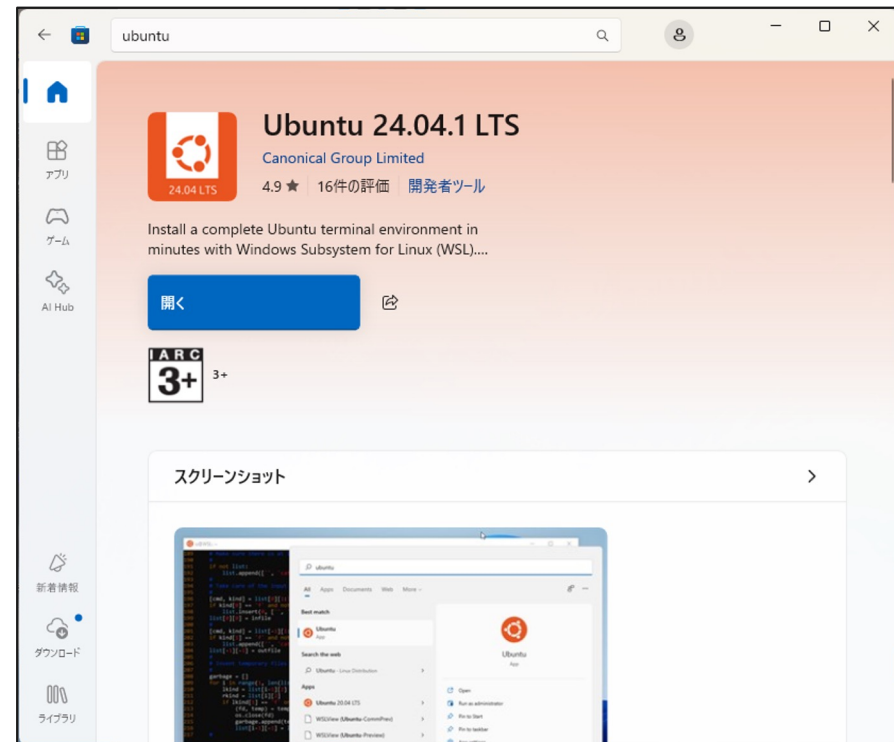
WSLで提供されるのはCUI (Command Line Interface) のみであり、マウス操作ではなくキーボードでコマンドを入力して操作する。
これにより、WindowsではインストールできなかったLinux用ソフトウェアをUbuntuを利用してWindows端末にインストールすることが可能になる。

本資料では、UbuntuというLinuxディストリビューションを使用する。

Windows Subsystem for Linuxの導入

Microsoft Storeから **Ubuntu 24.04 LTS** を検索して入手する。

2025/07/28現在では、
Ubuntu 24.04.01と
Ubuntu 24.04.05が存在するが、
ここでは24.04.01を入手することとする。

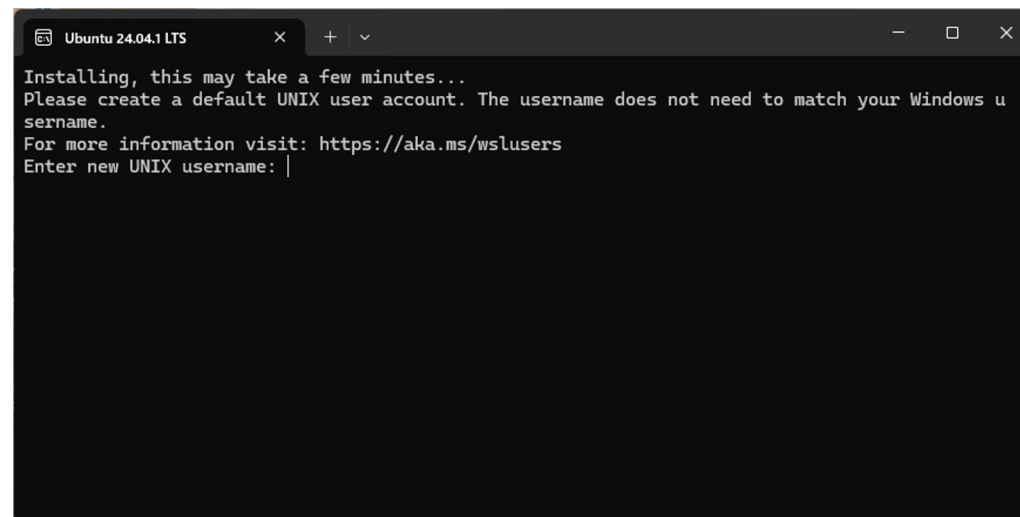


Ubuntuの起動

インストールが完了するとUbuntuが起動する。
起動しない場合は、Windowsの検索バーから
「Ubuntu」を検索し、実行すること。

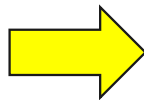
初回起動時には、
「Unix (Ubuntu) で使用するユーザネーム」と
「パスワード」の設定が求められる。

⚠ 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は確認できません。
画面上には何も表示されませんが、正常に入力されているため、タイプミスしないように注意してパスワードを設定してください。



Ubuntuの起動

このように
[設定したアカウント名]@[PC名]:~\$
のように表示されれば導入完了となる



```
nibb@DESKTOP-589RG83: ~  
Installing, this may take a few minutes...  
Please create a default UNIX user account. The username does not need to match your Windows u  
sername.  
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 24.04.1 LTS (GNU/Linux 6.6.87.2-microsoft-standard-WSL2 x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/pro  
  
System information as of Mon Jul 28 11:05:38 JST 2025  
  
System load:  0.0      Processes:            56  
Usage of /:   0.1% of 1006.85GB  Users logged in:     0  
Memory usage: 11%      IPv4 address for eth0: 172.31.127.14  
Swap usage:   0%  
  
This message is shown once a day. To disable it please create the  
/home/nibb/.hushlogin file.  
nibb@DESKTOP-589RG83:~$ |
```

rootについて

前スライドでは、ユーザネーム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"と呼ばれる機能を持つアカウントになる。

rootとは、管理者ユーザー・特権ユーザーのことであり、
Windowsでいう管理者アカウントに近い権限を持つ。

この機能を持つユーザーは「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ー名とパスワードの管理には十分ご注意ください。





Windows Terminalの導入

Windows Subsystem for Linux (WSL) で、UbuntuというLinuxの環境を使えるようになった。
このUbuntuを使って、専用のコマンドやプログラムを動かすことができる。

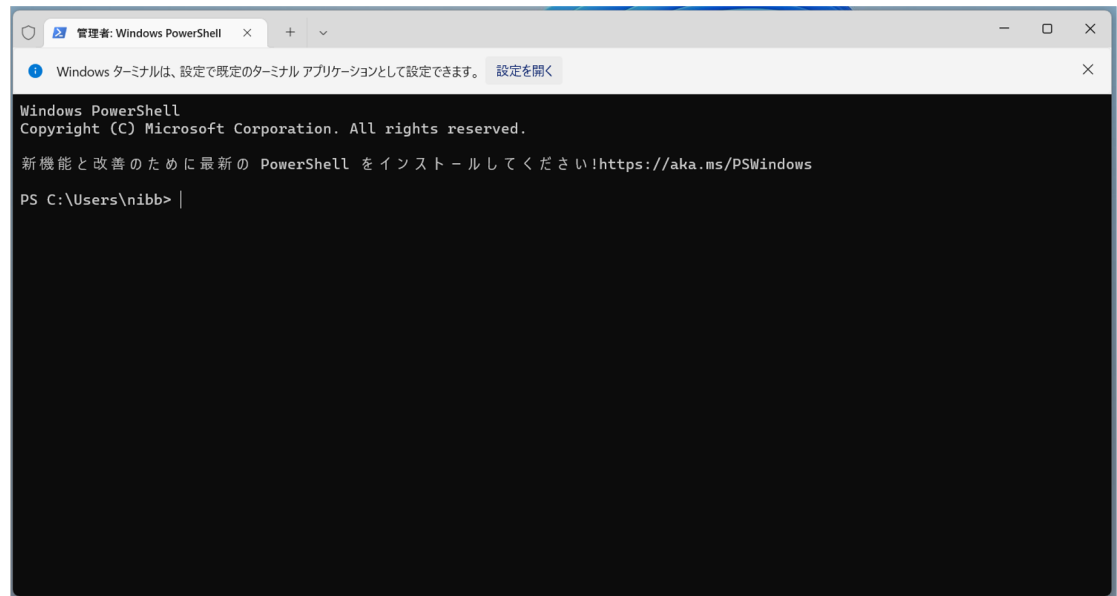
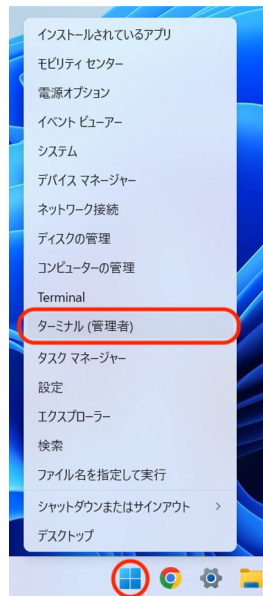
Ubuntuをそのまま使うことも可能だが、普段Windowsで使っているファイル(テキストや音声データなど)がある場所とは別の場所に、Ubuntu用の作業場所が設定されてしまう。
これだと、Windowsのファイルにアクセスするときに少し手間がかかる。

そこで「Windows Terminal」を使うと、Ubuntuを起動したときに、
普段Windowsで使っているファイルがある場所(例: デスクトップやドキュメントフォルダなど)を、
そのままUbuntuの作業場所として使えるようになる。

これにより、Windows上のファイルにとっても簡単にアクセスできるようになり、
Ubuntuをさらに便利に使えるようになる。

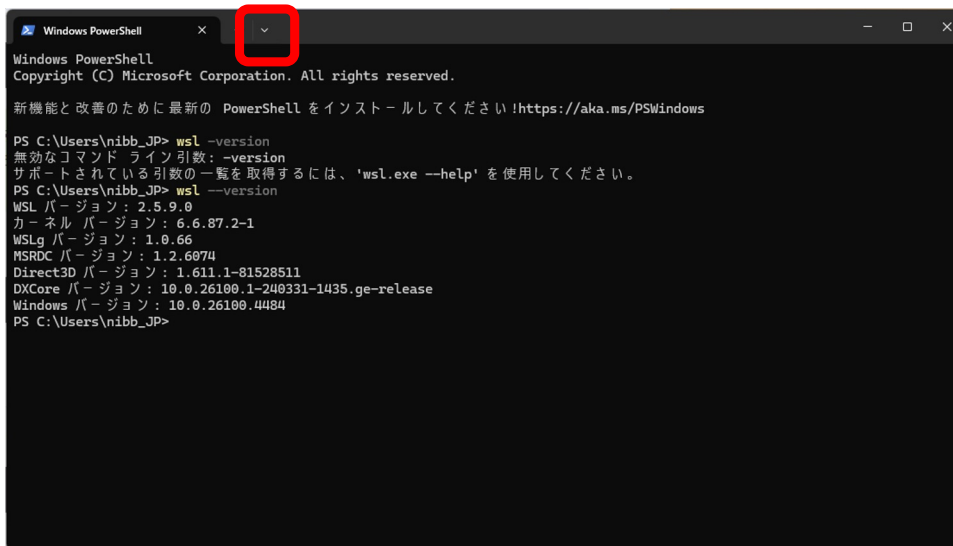
この資料では、Windows Terminalを使って、より使いやすい環境を整える方法を説明する。

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 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:\Users\nibb_JP> wsl --version
無効なコマンド 引数: --version
サポートされている引数の一覧を取得するには、'wsl.exe --help' を使用してください。
PS C:\Users\nibb_JP> wsl --version
WSL バージョン: 2.5.9.0
カーネル バージョン: 6.6.87.2-1
WSLg バージョン: 1.0.66
MSRDC バージョン: 1.2.6074
Direct3D バージョン: 1.611.1-81528511
DXCore バージョン: 10.0.26100.1-240331-1435.ge-release
Windows バージョン: 10.0.26100.4484
PS C:\Users\nibb_JP>
```

Windows Terminalを起動すると、Windows PowerShellが起動する。

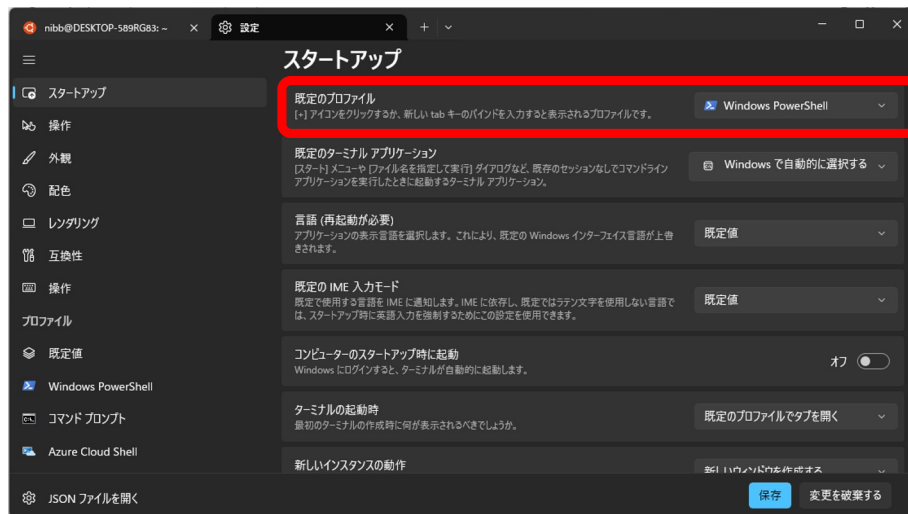
Windows PowerShellの右側の▼をクリックすると、どのシェルを使うかと聞かれるため、「Ubuntu」を選択する。

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。

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

apt コマンド

Ubuntu環境では、ソフトウェアのインストールや削除などを簡単に行うための「パッケージ管理ツール」として、apt (Advanced Package Tool) コマンドを使用する。

- 管理者権限について:
ソフトウェアのインストールや削除を行う際には、管理者としての特別な権限が必要である。
apt コマンドは管理者権限でコマンドを実行するための sudo コマンドと組み合わせて使用する。
 - sudo とは?:
"SuperUser DO" の略で、「管理者として（特別な権限で）コマンドを実行する」という意味。
安全のため、パスワードが求められたらUbuntuのパスワードを入力してください。
 - 例: `sudo apt install libncurses5-dev` (libncurses5-dev というソフトウェアをインストールする例)
- apt コマンド自体はUbuntuに最初からインストールされているため、別途導入する必要はない。

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)が古いままになっている。まずはこれを更新する。
 - `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。
 - `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う。
 - `sudo apt dist-upgrade`



(補足資料)apt コマンドの一例

- apt コマンドの紹介(実行する必要はない)
 - `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。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。



(補足資料)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る。
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る。
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
- 準備ができたら、apt-file を実行する。
 - `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

(補足資料) apt 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 -I. -c -o cram/cram_io.o
cram/cram_io.c:57:10: fatal error: lzma.h: No such file or directory
```

    - 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。
 - 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。