

連載

Windowsユーザーに贈るLinux超入門

第26回

Visual Studio CodeとWSLを使って開発してみよう

後藤大地

2019/07/10 09:20

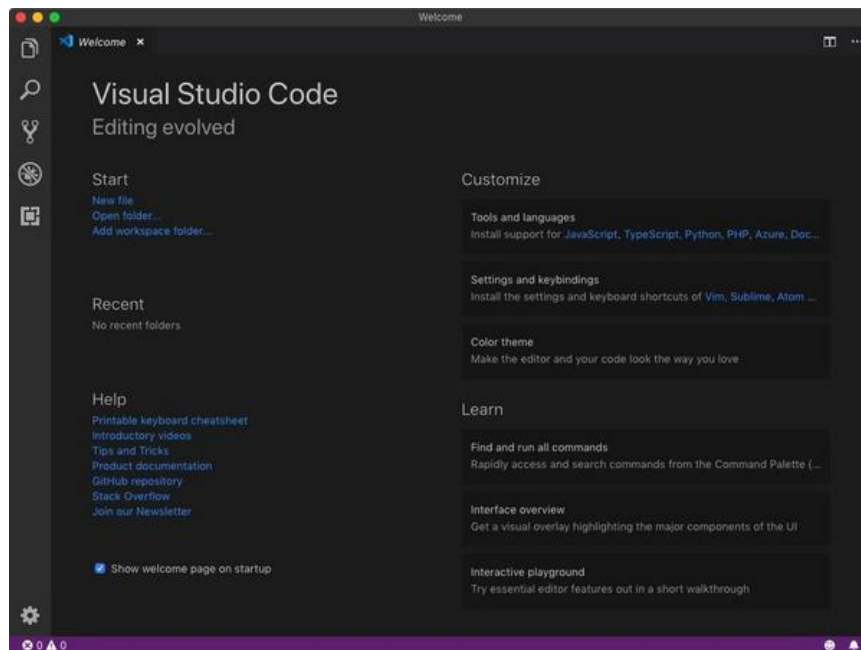
これまでエンドユーザーの観点から、WindowsとLinuxの違いについて説明してきた。しかし今回は、前回のWindows Terminalからの流れを踏まえ、開発者の視点でWindowsとLinuxの違いについてまとめてみたい。

今後、WindowsプラットフォームでLinuxアプリケーションを開発するというスタイルが主流になっていく可能性があるので、両OSの関係性を押さえておいて損はない。

人気の高い開発環境「Visual Studio Code」

現在、開発環境として人気の高いアプリケーションに「[Visual Studio Code](#)」がある。Microsoftが開発しているアプリケーションで、多機能エディタという位置づけにある。WindowsプラットフォームのみならずmacOSやLinuxでも動作可能だ。オープンソース・ソフトウェアとして公開されており、急速に人気を伸ばしている。

Visual Studio Codeはエディタであるにもかかわらず、機能は多岐にわたり、エクステンションをインストールすることでそれぞれの開発環境に敵した機能を提供してくれる。Visual Studio Codeを利用する開発者は増える一方だ。



macOS Mojaveで動作するVisual Studio Code

Visual Studio Codeを使うことでWindows、macOS、Linuxのどの環境においてもソフトウェアの開発を行うことができる。しかし、ここ数カ月で進んだVisual Studio Code with WSLという開発スタイルにも目を向けていただきたい。

WindowsプラットフォームでLinuxアプリケーションを開発する新しいスタイルが生まれつつあるからだ。もしかすると、MicrosoftがWSLの開発に取り組んだ本当の理由はここにあったのではないかとすら思えてくる。

Visual Studio CodeとWSL

WSLを使ってWindowsで動作するVisual Studio CodeでLinux向けのアプリケーションを開発するといった内容のコンテンツはさまざまな人が発信している。例えば、マイクロソフトは最近、ブログで以下のようなコンテンツを公開した。

- [Take your Linux development experience in Windows to the next level with WSL and Visual Studio Code Remote](#)

このコンテンツでは、Visual Studio Codeの開発版にエクステンションを導入して、Windowsプラットフォームで動作するWSL上のLinuxを使ってPythonスクリプトの開発を行うといった内容を紹介している。まず、次のようにWSLで動作するUbuntu 18.04 LTSから開発版のVisual Studio Codeを起動している。



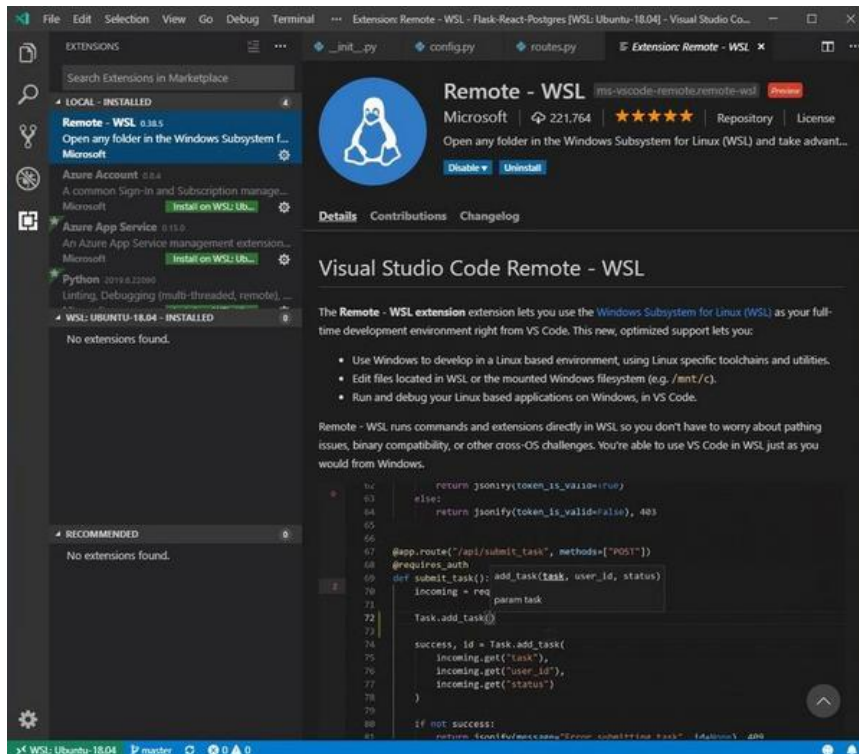
```
daichi@PARANT10-ONGS: /mnt/c/Users/daich/Documents/Flask-React-Postgres$ code-insiders .
daichi@PARANT10-ONGS: /mnt/c/Users/daich/Documents/Flask-React-Postgres$ neofetch
daichi@PARANT10-ONGS
-----
OS: Ubuntu 18.04.1 LTS on Windows 10 x86_64
Kernel: 4.4.0-18362-Microsoft
Uptime: 8 mins
Packages: 751
Shell: bash 4.4.19
Terminal: /dev/tty2
CPU: Intel i7-4770HQ (2) @ 2.195GHz
Memory: 2087MiB / 3071MiB

daichi@PARANT10-ONGS: /mnt/c/Users/daich/Documents/Flask-React-Postgres$ code-insiders .
daichi@PARANT10-ONGS: /mnt/c/Users/daich/Documents/Flask-React-Postgres$ code-insiders .
daichi@PARANT10-ONGS: /mnt/c/Users/daich/Documents/Flask-React-Postgres$ code-insiders .
Installing VS Code Server fe0c3e785c22c3ed2a9caa7178488c92d62bdb08
Downloading: 100%
Unpacking: 100%
daichi@PARANT10-ONGS: /mnt/c/Users/daich/Documents/Flask-React-Postgres$
```

WSLからWindowsのVisual Studio Codeを起動

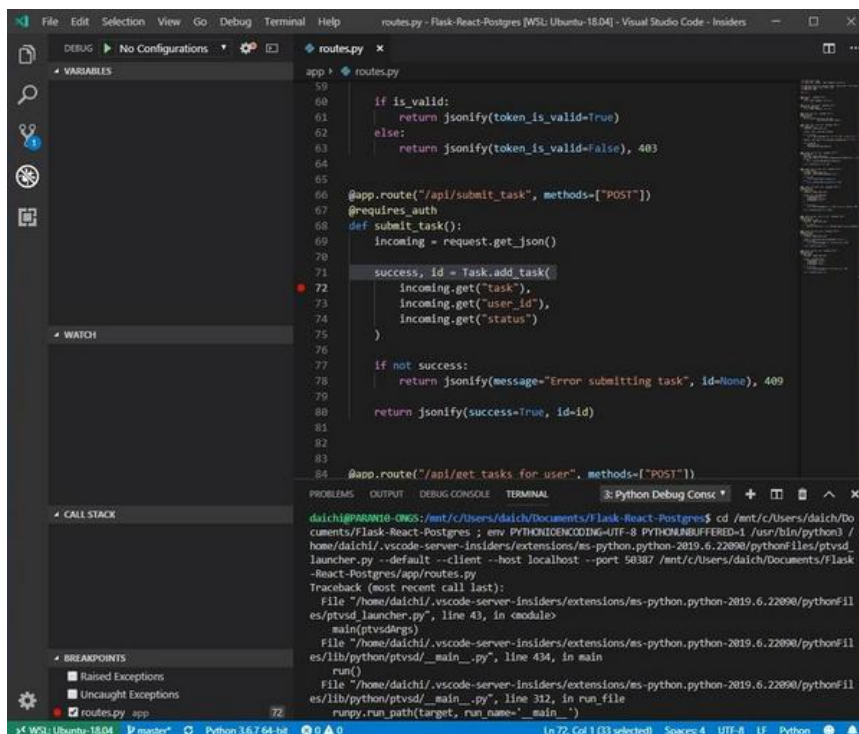
さらに、ブログではコマンドプロンプトからプロジェクトのフォルダへ移動し、そこからWSLを起動、さらにそこから開発版のVisual Studio Codeを起動するといった手段を踏んでいる。WindowsとLinuxを行ったり来たりしながら、アプリケーションを起動している点も特徴的なところだ。

Visual Studio Codeには「Visual Studio Code Remote - WSL」といったVisual Studio CodeでWSLを利用するためのエクステンションと、Python開発を行うためのエクステンションがいくつかインストールされている。



Visual Studio Code Remote - WSL

以下のスクリーンショットは、開発版のVisual Studio CodeでPythonスクリプトを開発している場面だが、デバッグ用に実行している環境がWSL上のpython3コマンドを使っている。このスクリーンショットを撮ったWindows 10にはPythonがインストールされていない。実行に必要なPythonはWSL上のUbuntuにインストールされたものが使われている。



WSLを使って動作するVisual Studio Code開発版

Python 3.7はまだベータという位置づけだが、Windowsにも移植が行われており、Windows版のPythonを使って同じようなことができる状況にはなっている。しかし、最終的にLinuxで動作させることを考えると、開発段階でもLinuxを使うというのは意味のあることだ。

Windows 10でLinuxアプリケーション開発

デスクトッププラットフォームとしてWindowsは圧倒的なシェアを持っている。Net Applicationsの調査によれば、2019年6月におけるデスクトップオペレーティングシステムのシェアはWindowsが87%を超えている。GUI環境としてWindowsを使うことは、数の上から言っても一般的だろう。

これに対し、サーバプラットフォームとしてのWindowsのシェアは限られたものだ。エッジサーバではLinuxが圧倒的なシェアを持っており、サーバで動作するアプリケーションの開発はLinuxを使うというのが自然な流れになっている。

ここで、Visual Studio CodeとWSLの出番というわけだ。Visual Studio CodeとWSLを利用すると、GUI環境として広く使われているWindows 10を使いながら、Linux向けのソフトウェア開発をネイティブに行うことができる。WSLで動作するLinuxはUbuntu 18.04 LTSだし、開発環境としては文句のない状況になっている。

前回、Windows Terminalを取り上げるとともに、WindowsとLinuxの技術が混ざり始めているといった話をしたが、開発環境においてもそうした取り組みが進んできている。ノートPCのようなデバイスを考えると、やはりベアメタルで動作するオペレーティングシステムはWindows 10かmacOSか、そのプロダクト向けに提供されているオペレーティングシステムを使っておくのが無難だ。ここにLinuxをインストールすると何らかの問題が生じることもあり、Linuxは仮想環境などを経由して使うのが安全な選択肢となる。

今年、Windows InsidersでWSL 2が利用可能になったことが[発表](#)されたが、これによりファイルシステム性能の向上も期待できる。Visual Studio CodeとWSLを使った開発手法は現時点で広く使われていないが、今後利用が進む可能性が高く、注目しておきたい技術である。

※本記事は掲載時点の情報であり、最新のものとは異なる場合があります。予めご了承ください。

連載一覧

- 第28回 [Visual Studio CodeとWSLでPython開発に挑戦してみよう](#)
- 第27回 [ここでおさらい、WSL 1とWSL 2の違いとは？](#)
- 第26回 [Visual Studio CodeとWSLを使って開発してみよう](#)
- 第25回 [Windows Terminalプレビュー版登場、その注目点は？](#)
- 第24回 [Vimより簡単なエディタ「nano」をカスタマイズしてみよう](#)

[目次ページへ](#)