

革命後のソフトウェア 開発ライフ

土井 伸洋

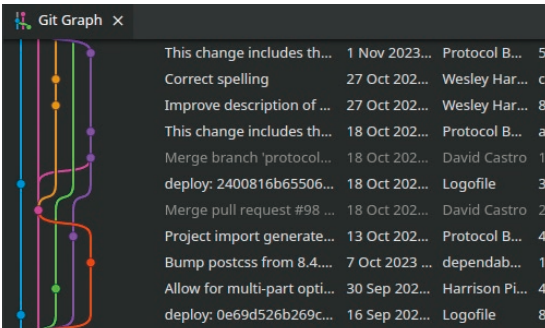


図1 Visual Studio CodeのGitエクステンションGitGraphの表示例

各機能の開発経緯がGUIで確認できる。またcommitやpush, mergeといった操作もGUI上からできる

ソフトウェア開発企業に勤める筆者は、普段は自動運転にかかわる解析・認識処理ソフトウェアの開発に従事しています。主要な利用言語はPythonです。本誌集で取り扱う、Visual Studio Code (VSCode), Git, Docker, Copilotの4大ツールももちろん使っています。むしろ利用することが前提で、空気のように感じている部分があるかもしれません。

そんな筆者がどのように4大ツールを使っているのか、紹介します。

● 開発業務からメモまで…何でもVSCode

VSCodeはマルチプラットフォームなコード・エディタであり、2024年現在、全開発者の中で最も使われています。フル機能のIDE(統合開発環境)に比べ軽量であり、エクステンションを追加することで、同じユーザー・インターフェース(UI)で多様なプログラミング言語に対応できるメリットがあります。

筆者の職場でも以前はEclipseやJetBrains製ツールを使っていましたが、最近ではほとんどのメンバがVSCodeを使用しています(VSCode以外ではVimを使っているメンバがいる)。

業務以外の個人開発においても、マイコンのファームウェア開発(C++)やUnity(C#)向けのコーディン

グ、何かのメモをとる(Markdown)も全てVSCodeで行っています。

これ以降に述べるGit, Docker, Copilotも全てVSCodeのエクステンションを通じて使っています。

● 問題発生時に強い味方になる…Git

Gitとはソースコードの変更履歴を管理するためのシステムです。コードの変更履歴をGitリポジトリに蓄えておくことで、確実に動く状態のコードを残し、またコードの差分を確認することで、問題が起こったときのデバッグを容易にします。特に複数人でコーディング作業するときには欠かせないツールです。

筆者の職場ではコーディング作業を行うメンバはもちろん全員が使っています。加えて筆者の場合、VSCodeに導入できるGitエクステンションも使っています。バージョンの系譜や、各ブランチの開発状況がGUIで表示されるため、特に開発メンバが増えてきたときに助かっています(図1)。

一方で未経験者にとってGitは鬼門であるとも言われます。コマンドを間違えて数時間分の作業を台なしにしたり、間違ったmergeやforce pushでリポジトリを破壊し、チーム・メンバに迷惑をかけた経験が筆者もあります。

そんなときにはコマンド操作にこだわらず、まずはVSCode + Gitエクステンションで提供されるUIを眺めて、何をやっているのかを先につかむのが効果的です。

● 開発環境構築トラブルとサヨナラ…Docker

プログラム開発においてコーディングと並んで重要となるのが環境構築です。これにはLinuxディストリビューションやバージョンだけでなく、利用するライブラリやバージョン、場合によっては環境変数などの設定も含まれます。近年では、これら環境の構築にはDockerを使う方法が主流です。

開発環境の構築時、過去の手順書に従ってインストールしたものの動かなかったり、新しいLinuxバージョンでは特定のパスや環境変数の設定が必要となったり、そもそも手順書が保守されていなかったり(ひ