



付 6 3 話 開発環境の変更は重要だが負担が大きい

プログラム開発を行っているとき、2年から3年に一度、コンパイラなど開発環境を変更する必要が生じる。OSの変更によっても、同様に開発環境を更新する。OSに適合することや、セキュリティなど各種の問題の解決、スピードアップなどの性能向上を目指すために必要だと思うが、コストを含め負担が大きい。使用側にとってもOS更新は、研修や開発に莫大な費用を要するが、何のメリットもない。最早、使い勝手の良さより、セキュリティに強いOSを作って欲しい。

開発環境の変化は、システムを開発する側にとっても、大きな負担を強いる。開発環境が変化しても、元のプログラムが変更なしでビルドされ、正常に動作するときは、更新コストがかかるのみで問題はない。コンパイラや開発環境が変化すると、プログラムエラーが発生し、リンクしたソフトが動作しなくなることがある。一般に、言語のルールが厳しくなり、既定の取り決めが無視される傾向にある。例えば、FORTRANで静的に配列を定義すると、自動的にゼロクリアが行われていたが、いつの時点からかそれがなくなり、全く異なる結果が得られる。配列の大きさを超えてアクセスしても曖昧に動作していたが、これもいつの間にか、突然動作が止まり、システムがクラッシュする。基本的にはプログラムのルール違反、文法エラーではあるが、開発環境が更新されるとそれらが表面化する。ビルドできても元の状態に戻らず、正常に動作しない。「今まで、動いていたのに」と悔しがることになる。

私が使用した開発環境を振り返り、その変遷をたどってみよう。最初のプログラム開発は、大学のミニコンピュータを用いたFORTRANによる数値解析用のソフトである。紙テープによるプログラムと入力データ、最後に、コンパイル・リンクと実行を指示する数行のコマンド、だったと思う。残念だが忘れてしまった。いずれにしてもコンピュータによる開発環境は何もなく、紙と自分の頭で開発を行う。大学の大型計算機でも同様、紙テープがカードに、ドットプリンタがラインプリンタに変わるのみで、開発環境は変化しない。計算機はTSSで動作し、日本語は全く使用できない。カード群の最初に数枚の制御用カードを添付する。

大学間ネットワークが構築され、地方大学の計算センターとも繋がる。大学の計算機センターに登録すると、記憶領域を数メガバイト貸し出される。そこにプログラムを記憶させ、開発と実行を行う。初めて、ソースコードと実行形式が分離されて記憶され、実行する度にコンパイル・

リンクをしなくて良い。プログラムを開発するためには、ソースコードを編集する必要があり、大型計算機にはテキストエディタが付いており、これを利用する。後は、対話形式でコマンドを打ち込み、コンパイル・リンク、続いて実行。ここでも、開発環境は非常にシンプルである。当時は、数値計算で使用する言語は FORTRAN、事務処理用では COBOL が使われており、このプログラミング言語は、計算機に実行させたいことを順番に命令語を記述する手続き型言語という。他に LISP のような関数型言語や、Prolog のような論理型言語などもあり、両者は AI 用言語として使用される。

ミニコンや WS などでは Unix 系の OS になり、開発環境は飛躍的に向上する。テキストエディタは Vi か Emacs が OS に付属しており、特に Vi エディタは 2 つの動作モード、コマンドモードとテキストモードを切り替えて使うなど、私にとっては使いづらいエディタである。対話形式で Unix コマンドを使用し、プログラムを開発する。開発ソフトが徐々に大きくなるとプログラムを分割し、メイクを使用してコンパイル・リンクを行う。メイクはコマンドライン処理では非常に有効であり、コンパイルやリンクを明示的に指示することができる。さらに、Unix には C シェルによって自前でコマンドを作ることができ、開発環境は飛躍的に進歩する。

Windows 用のソフト開発は、開発環境を一変させる。統合環境がないと開発は難しい。高級言語を用いても、一から全て書くのは煩雑で無駄。ライブラリやフレームワークなど各種の機能を持つモジュールが用意される。統合開発環境(IDE)は、アプリケーション開発に必要なソフトをコンパイルし、ライブラリなどとまとめて提供するパッケージである。統合開発環境には、Xcode、Swift、Android Studio、Eclipse などが有名で、言語や OS に適した IDE(Integrated Development Environment)を使用する。特に Visual Studio は Microsoft が Windows 向けのアプリケーション開発用としてリリースしたもので、.NET Framework、Visual Basic、C++、C#などのプログラミング言語で開発が可能である。私自身は、C++ とインテル製 FORTRAN とのバイリンガルで SPACE を開発してきたが、設定や使用方法の学習でかなり苦労した。数度、IDE を更新したが、その都度、使い勝手が変わり、またコンパイル・リンクの設定が変化する。コンパイラも更新し、これまで正常に動作していたプログラムに、文法エラーが発生する。正常に動作させるまでに相当の労力を要し、コストと共に負担が大きい。更新するか否か、迷うところである。