クロス・コンパイルで ビルド待ち時間を減らす

山田 英伸

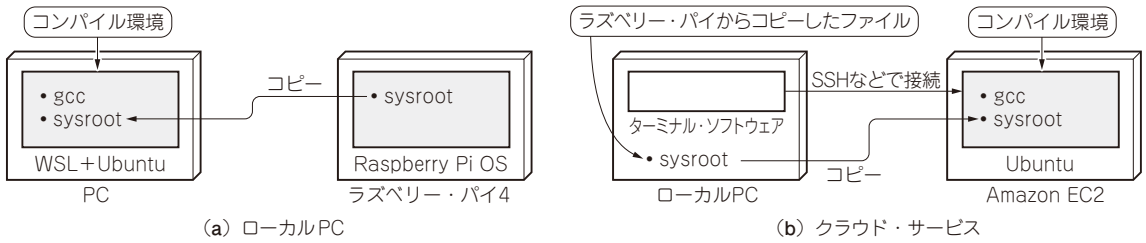

図1　クロス・コンパイル環境の準備

ラズベリー・パイのようなシングルボード・コンピュータを装置に組み込み，制御や通信処理に利用する場合があります．それらの開発作業においては，ラズベリー・パイで動くLinux上に開発環境を整備して，プログラミングやデバッグを行えます．しかし，大きなプログラムを扱う場合，ラズベリー・パイのリソースではビルドに時間がかかってしまいます．

ビルド時間を短縮するため，複数のラズベリー・パイを使ってビルドするシステムを2023年11月号で紹介しました．今回は，PCやクラウド・サービスを使ってクロス・コンパイルし，ラズベリー・パイで動くバイナリを生成する方法を紹介します．
（編集部）

PCやクラウド・サービスを使って，ラズベリー・パイ向けにOpenCVをクロス・コンパイルし，かかった時間を比較します．

クロス・コンパイルとは，実行環境とは異なる開発環境を使って，実行環境で実行可能なオブジェクトを生成することを指します．例えば，WindowsやLinuxなどのPCを使ってソースコードをコンパイルし，ラズベリー・パイで動く実行可能ファイルを生成します．

クロス・コンパイルできるコンパイラとして有名なのが，gccやclangです．本稿では，gccをPCのLinux上で使用します．

ハードウェアとしてPCを使う方法と，クラウド・サービスのAWSを使う方法をそれぞれ環境構築から説明します．その後，実際にクロス・コンパイルし，ビルド時間を比べます．

環境構築1：PCでクロス・コンパイル

● WSL＋Ubuntu上にツールを準備

Linuxの実行環境として，Windows 10以降ではWindows Subsystem for Linux（WSL）を使えます．今回は，WSL上にUbuntu 20.04をインストールして，クロス・コンパイラをはじめとしたツール・チェインを用意します［**図1**（a）］．クロス・コンパイラをインストールするために次のコマンドを実行します（前回のラズベリー・パイ上のgccとメジャー・バージョンが合うように選択している）．

```
$ sudo apt update↵
$ sudo apt install gcc-10-aarch64-
linux-gnu g++-10-aarch64-linux-gnu↵
```

• ファイルのコピー

ラズベリー・パイ用の実行可能ファイルをビルドするには，ラズベリー・パイ用のインクルード・ファイルなどが必要です．そのため，Raspberry Pi OSから必要なファイルをコピーして，クロス・コンパイルに必要となるsysrootディレクトリを構築します．

Raspberry Pi OSが起動しているラズベリー・パイが，raspi4.localというネットワーク・アドレスでアクセスできる場合を例に手順を説明します．**リスト1**のコマンドによってPC環境のsysrootディレクトリにラズベリー・パイからファイルを取得します．ファイルをコピーした直後では，シンボリック・リンクが不正な場所を示しています．例えば，**リスト2**のようなリンクになっています．正しくは，