

名刺サイズ・コンピュータにもよく使うしIoT向き

やり直しのための シェル入門

中村 和敬

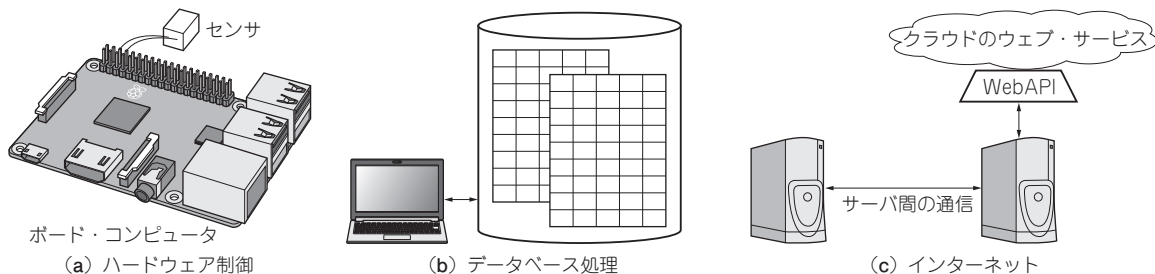


図1 伝統的なシェルができることは意外と(?)IoT時代向き!

本稿では、UNIXの中でのシェルの位置づけ、他のプログラム言語との違い、シェルでできること、シェル・スクリプトでシステムを作る場合の考え方などを紹介します。

メリット

近年高性能なCPUを搭載し、組み込み用途向けのUNIX系OSが動作するボード・コンピュータ^{注1}が注目されています。ラズベリー・パイではLinuxのディストリビューションのDebianをアレンジしたRaspbianが動作します。

UNIXのCUI(Character-based User Interface)はシェルです。

● 全てのUNIXシステムで使える

シェルは最初にUNIXのユーザ・インターフェースとして開発され、現在も全てのUNIX系OSにシェルが搭載されています。現在のUNIX系OSではリッチなGUI(Graphical User Interface)も利用できるようになっていますが、OSの設定や、システム開発でよく使われるソフトウェアの設定などでは、シェルを通じた操作が必要です。

注1：インテルEdisonボードに搭載されているLinuxなどは、busyboxという機能が制限されたプログラムを利用しているため、本稿の内容は当てはまらない部分がある。

また、シェルは本格的なプログラム言語であり、UNIXはプログラムの起動やファイルの操作といった抽象度の高い処理から、デバイスの直接的な操作のような抽象度の低い処理までできるように作られています(図1)。シェルの使い方を知ることで、スーパーコンピュータから組み込み用のボード・コンピュータまで、UNIXの動作するさまざまなコンピュータでシステムを作ることができます。

シェルで記述されたプログラムは、シェル・スクリプトと呼ばれます。

● シンプルで理解しやすい

シェル・プログラミングの特徴はUNIXのパイプという機能を多用する点です。パイプはプログラムを連係動作させるための仕組みです。

パイプを用いたシェル・プログラムはシンプルで理解しやすく、開発しやすいことが利点です。

● 他のプログラム言語で作成されたライブラリが使える

Python、Rubyなどの他のプログラム言語で作成されたライブラリも、UNIXのコマンドという形に加工することで、組み合わせる利用しやすいという利点があります。

● 並列処理を実現しやすい

近年のCPUは複数の命令実行ユニットの搭載され