

第1章

LinuxベースのシステムにGUIメニュー画面を実装する各種方法

LinuxのGUIシステムとAndroid採用のメリット

片岡 啓明 Hiroaki Kataoka

スマートフォンが普及した昨今では、タッチパネル機能付きLCDモニタを装備した機器に対してもリッチなGUI(Graphical User Interface)によるメニューが要求されるようになりました。Linuxベースの組み込みシステムに、GUIによる操作画面を実装する方法にはいろいろなものがあります。ここではまず、Linux上でGUIを構築する各種の方法について解説します。後半ではAndroidのGUIシステムの構成について解説します。

1. LinuxにおけるGUI構成

本特集のテーマであるAndroidについて述べる前に、まずはLinuxにおけるGUIのシステム構成について触れておきましょう。一般的な構成を示したものが図1になります。

さまざまな切り口があるかと思いますが、ここではLinux内部をGUIフロントエンドとバックエンドの大きく二つの層に分けてみました。

● GUIのバックエンド

図1の下側に位置するバックエンド層は、よりハードウェアに近い基本的な処理を行います。通常、この部分は特定のハードウェアに依存した実装が行われる層と特定のハードウェアに依存しない抽象化された層(HAL, Hardware Abstraction Layer)を持ちます。

まず出力であるグラフィックスに関していうと、こ

こではよりプリミティブな描画の処理、つまり単純な矩形や線の描画から、画像の転送、変形、合成などが行われます。これらの処理を行うソフトウェアにもさまざまなものがあります。代表的なものから順に見ていきましょう。

(1) Linux フレーム・バッファ

Linux フレーム・バッファは、Linux カーネル2.1から正式に採用された描画のためのドライバ・インターフェースです。通常2D描画のアクセラレーションとビデオ・モード(解像度、リフレッシュ・レートなど)を変更する機能を持っています。C言語で書いたソフトウェアからグラフィックスを扱う場合、Linuxでは最もシンプルな方法といえるでしょう。

フレーム・バッファの特徴は、グラフィックス・デバイスの持つビデオ・メモリをプロセスの仮想メモリ空間にマップすることによって、C言語のmemcpyなどの一般的なメモリ・アクセスで描画が行える点にあります。また、IOCTLシステム・コールからデバイス特有の機能を使用することも可能です。

(2) DirectFB

DirectFBは、2Dのグラフィックス向けのコンパクトなライブラリです。描画にハードウェア・アクセラレーションを使用することを目的としてデザインされており、組み込みLinuxではよく利用されます。

このLinuxフレーム・バッファが描画向けの最小限のインターフェースしか用意していないのに対して、DirectFBはもう少し高レベルなAPIを提供します。例えば、ウィンドウ・システムやレイヤ合成、画像、テキストのレンダリングなど2Dの描画で必要になる機能を一通り備えています。

(3) Window System

X Window System(以下X)は、クライアント・サーバ方式で動作するようにデザインされたウィンドウ・システムです。PC Linuxでは、GUIのバックエンドとしてXは欠かせない存在です。クライアントと

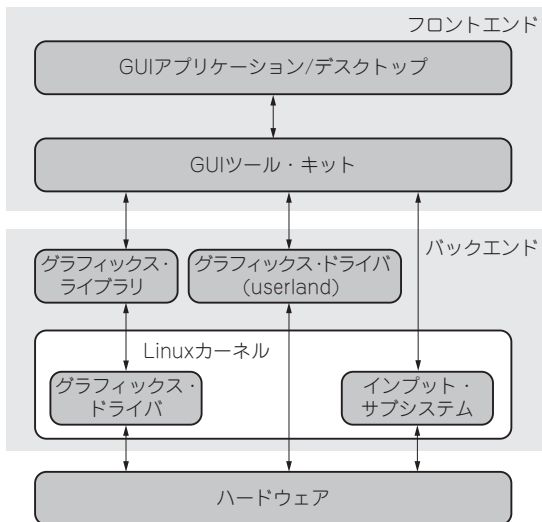


図1 LinuxにおけるGUI