

マイコン・プログラマのための
Linux 超入門

第8回

これからのキーテクノロジー！

進化が著しいグラフィックス描画&表示のメカニズム

宗像 尚郎

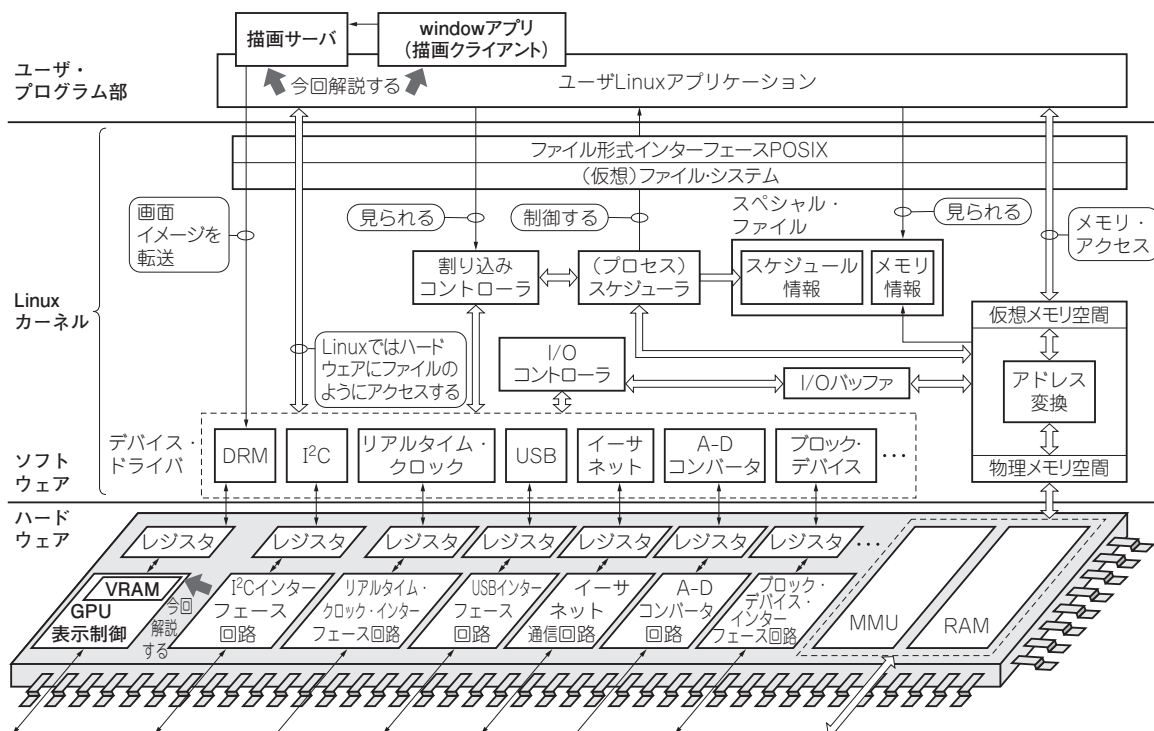


図1 今回解説すること…進化中! いくつかのアプリからのグラフィックス描画&表示要求を整理してくれるしくみ

● 複数アプリが動かしてもディスプレイは一つしかない

画面出力を持つ複数のLinuxアプリケーションを同時に利用する場合、それぞれのアプリケーションは、オフスクリーン・バッファと呼ばれるRAM領域に、画面のピクセル・イメージ(ビットマップ)を描画します。この画面イメージをカーネルが管理する表示コントローラに渡すことで、最終的にLCDパネルなどの表示デバイスに画像が表示されます。

LinuxはマルチタスクOSなので、アプリケーションはそれぞれに確保したオフスクリーン・バッファに非同期に画面イメージを書き出しますが、画面のあるアプリケーション(=描画クライアント)が複数ある場合には、共用している表示ディスプレイ装置にどの

アプリケーションの画面を表示するのかを管理するために、描画サーバ機能が必要になります(図1)。

ディスプレイに合わせた画面描画をレンダリングと呼びます。描画サーバはディスプレイの数や解像度、画面更新レート、表示可能な色数、ローテーション(横長か縦長か)などディスプレイ装置の仕様に合わせたレンダリングを行う必要があります(図2)。

組み込みLinuxボードの起動/停止やコマンド実行などを制御するコンソール端末は文字表示だけのキャラクター画面の場合が多いです。

Linuxパソコンではグラフィック画面上に複数のアプリケーションを並べてマウスなどでアプリを切り替えながら操作するマルチ・ウィンドウの利用が一般的でしょう。