

グーグルの「Android」は携帯電話を変えるか？ ケータイの明日を決めるプラットフォーム

グーグルから「Android (アンドロイド)」が発表されたのをきっかけに携帯電話プラットフォームに注目が集まっている。各種プラットフォームを解説すると同時に今後の携帯電話業界を展望する。 文 天田天 (モバイル技術ライター)

2007年11月、米グーグルがオープンソースのモバイルプラットフォーム「Android (アンドロイド)」を発表し、携帯電話プラットフォームがしばらく業界の話題をさらった。

携帯電話のプラットフォームはすでに数多く存在している(図表1)。ユーザーからするとプラットフォームは見えない存在であり、直接メリットに結びつかない。しかし実際には、プラットフォームには携帯電話の今後の進化や将来を左右するほどの影響力があるのだ。

では、携帯電話のプラットフォームとは何だろうか。それはPCにおけるオペレーティングシステム(以下OS)と思ってもらえばよい。誤解のないように言うと、携帯電話にもリアルタイムOS(RTOS)と言われるOSはちゃんと入っている。しかし、PC用のOS(Windows Vistaなど)とは異なり、RTOSが提供できるのはソフトウェアが動作するための必要最低限の処理(イベント、割り込み、メモリー管理など)だけである。画面の描画やアプリケーションの起動、ファイルの入出力などの機能はない。

天田天 (あまだ・たかし)氏
モバイル技術関連のライター。端末の開発を行いながら検定試験委員などにも参加

この、PCのOSと比べて足りない部分をRTOSの上位にあたる「ミドルウェア(ライブラリー)」や「フレームワーク」などで補う(図表2)。

このRTOS、ミドルウェア、フレームワークまでを含めて「プラットフォーム」と呼ぶのだ。場合によっては、ハードウェアとドライバーまで含めてプラットフォームと呼ぶこともある(この場合、アプリケーションプラットフォームとも呼ばれる)。それぞれの用語を解説しておこう。

(1)ハードウェア

CPUやメモリー、GPSなどの部品。

(2)ドライバー

OSからハードウェアをコントロールするために必要なソフトウェア。

(3)ミドルウェア(ライブラリー)

主に画面に関係しない共通処理を行う場所。例えば、音楽再生のようにマルチメディア情報を音に直す処理(コーデック)を行ったり、GPSのようなハードウェアをナビゲーションなど複数のアプリケーションから使う際の調整役として振る舞う。

(4)フレームワーク

アプリケーションの競合を管理し、複数のアプリケーション間で連携させて振る舞いを決める部分。例えば、操作中に着信の割り込みがあった場合、着信を通知して、元のアプリケーションを一時停止し、通話が終了したタイミングで元のアプリケーションに戻るなどの動作を実現する(図表3)。

ちなみにこのフレームワークには、PC用OSにはない仕様/機能が多数

存在する。そのため、PCで利用しているOSをそのまま携帯電話に組み込んでも、携帯電話の操作性は実現しにくい。フレームワークはミドルウェアと並んで携帯電話プラットフォームを構築するための重要要素である。

拡張性と互換性が鍵

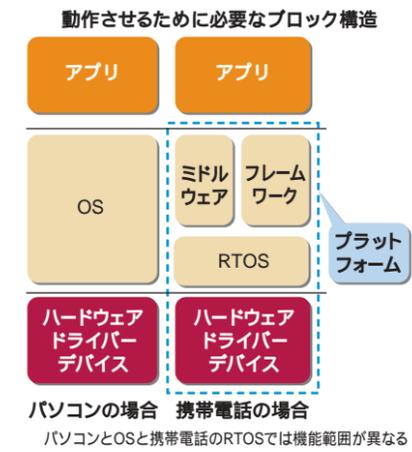
端末メーカーがプラットフォームを選択する上で大切なポイントは「拡張性」「互換性」の有無である。

携帯電話のサービスは日進月歩で進化しており、日々新しいデバイスが端末に実装されている。昨今ではMEMS(微小電気機械システム)の発展により、多くのセンサーデバイスが端末に取り付けられることもある。新たに組み込んだデバイスを動かすには、ドライバーの実装や、ドライバーをアプリケーションから用いるAPI(Application Program Interface)の追加拡張などが必要だ。

また、アプリケーションの互換性もプラットフォームには大切な要素だ。一度作成したアプリケーションが、同じプラットフォームを利用している他の端末で動かないとなるとその度に修正が必要となってしまう。互換性を保つためには、プラットフォームがハードウェアの違いを隠蔽する処理をできるだけ行い、仮想的にアプリケーションを動作させる技術が必要なのだ。

一言にハードウェアの違いと書いたが、OSをハードウェアに実装する際のポーティング(移植)レイヤーや、ドライバーの実装差異などもそこに含まれる。互換性を気にする場合には、これらのポーティングによるばら

図表2 携帯電話とパソコンのOSの違い



ている。これらを活用することで、携帯電話のソフトウェアを既存製品の組み合わせに近い形で実装することができるのも特徴だ。

デメリットは、1台当たりのライセンス費用が数ドルと比較的高い点だ。しかし、その分サポートがしっかりしているし、現在までSymbianでアプリケーションを作ってきたメーカーは資産継承の意味でも使い続けるだろう。当面、Symbianは好調を保つと思われる。

(2) BREW

BREW(Binary Runtime Environment for Wireless)は米クアルコム社のワイヤレスソリューション「MSMシリーズ」に搭載されているプログラム実行/開発環境であり、KDDIの携帯電話で使われている。

BREW単体ではフレームワークが存在しないため、KDDIが作成したKCP+(またはKCP)によりKDDI用のフレームワークとアプリケーションがメーカーに提供され、端末に実装されている。この他にも、クアルコムは独自にアプリケーションとフレームワークのセット「uiOne HDK」などの提供を行っている。

BREWの特徴は、無線通信のハードウェア(チップ)に加えて、その上で動作するソフトウェアを作成する環境が提供されている点だ。実装に楽な面もあるが、自社開発で完結できないため、拡張についてはどうしても難しい部分がある。また、実装依存性が比較的高く、アプリケーションの互換性については苦労する面が多いと言われている。

KDDIでは今後もBREWとKCP+

つきを無くす取り組みも必要だ。互換性が担保されていれば、アプリケーションが流通し、プラットフォームの使いやすさや市場での付加価値向上に貢献する。

4種類のプラットフォーム

プラットフォームにはそれぞれ得意不得意がある。図表4に記載したサポート範囲の違いの他にも、ライセンスや保証範囲の違い、製品実績の有無といった特徴がある。

ここからは、それぞれのプラットフォームの特徴を紹介していこう。

(1) Symbian OS

Symbian OSは現在世界で最も普及しているプラットフォームである。例えば、ノキアの「S60」、NTTドコモの「MOAP(S)」、UIQテクノロジーの「UIQ」など、各社はSymbian OSに独自のカスタマイズを組み合わせ使用している。

製品化の実績も多く、普及度の高さから、Symbianアプリケーションを作成するサードパーティの商品も多数販売され、1つの市場が形成され

図表1 OSとプラットフォームを採用している国内事業者

OS		プラットフォーム		国内採用事業者
OS名	OS会社名	PF名	PF会社名	
Symbian OS	シンビアン	S60	ノキア	NTTドコモ
		UIQ	UIQテクノロジー	
		MOAP(S)	NTTドコモ	
BREW	クアルコム	KCP+	KDDI	KDDI
		uiOne+HDK	クアルコム	
Linux		Android	グーグル	NTTドコモ
		ALP	ACCESS	
		LiMo Foundation Platform	LiMo Foundation	
		Mobilinux	MontaVista Software	
Windows Mobile	マイクロソフト	OSに含まれる		ソフトバンクモバイル NTTドコモ ウィルコム イー・モバイル

OSを含んだ形でプラットフォームは形成される。プラットフォームは事業者の戦略を反映する。