

Androidを1秒で起動する
QuickBootの概要

橋本 健一

多くのAndroidはバッテリー動作の組み込み機器で動作する。そのときに問題となるのがスリープ時の処理だ。CPUやメモリに通電したままスリープを行うと、スリープ中も電力を消費してしまう。かといってメモリ内容のすべてを不揮発性メモリに保存するハイバネーションでは、スリープからの復帰に時間がかかるという問題がある。これらの問題を解決したのがQuickBootだ。

(編集部)

はじめに

● 既存の起動時間を高速にする手法

OS起動時間の問題は、コンピュータ・システムの永遠の課題です。特に組み込みLinuxを用い、家電に応用するシステムでは大きな問題になっています。また今回の特集でも扱っているAndroidは、組み込み機器でも使われるため、同様の問題を抱えています。

この問題を解決し、高速起動を実現させるために用いられてきた手法は大きく分けて二つあります。

一つはいわゆるLinuxブート・シーケンスのチューニングです。不要なドライバ類を外すことで、起動時間を短縮します。たしかにデモ機レベルならこの方法は効果的なのですが、実際の製品で必要なドライバを組み込み、アプリケーションまで起動することを考えると、期待するほど起動時間は削減できません。

● ハイバネーションによる保存/復帰

もう一つはハイバネーションと呼ばれる技術による方法です。図1はハイバネーションのメモリ・イメージを保存したときの処理です。一度通常のブート・シーケンスに従ってブートし、所望のアプリケーションまで起動します。その後、スナップショット・ドライバと呼ばれるドライバ・ソフトウェアを用い、CPUやI/Oのレジスタ類をRAMに保存し、そのRAMイメージを不揮発性メモリに保存します。

図2はハイバネーションから復帰するときの処理です。不揮発性メモリからRAMへイメージを戻し、保存していたCPUやI/Oのレジスタを復帰させ、プログラム・カウ

ンタを戻します。これらも保存時と同じようにスナップショット・ドライバの利用で、アプリケーションが起動した状態に復帰させます。

● ハイバネーションの弱点

ハイバネーションは効果的な手法ですが、致命的な弱点があります。ソフトウェアが使用するメモリ・サイズが増加するのに比例して、RAMイメージを不揮発性メモリから復帰する時間も増加してしまうのです。つまり、起動時間が使用メモリ・サイズに比例して遅くなるのです。

たとえば、アプリケーションがリッチなGUIをもち、大きなデータを扱うようなシステムでは、巨大なRAMを搭載するため、ハイバネーションによる起動を行っても数十秒を要してしまいます。結果的に通常ブートしたときと比べて起動時間を大きく短縮させることができません。RAMイメージの圧縮を行えば多少は起動時間を短縮できますが、大きく短縮することは困難です。組み込みLinuxのハ

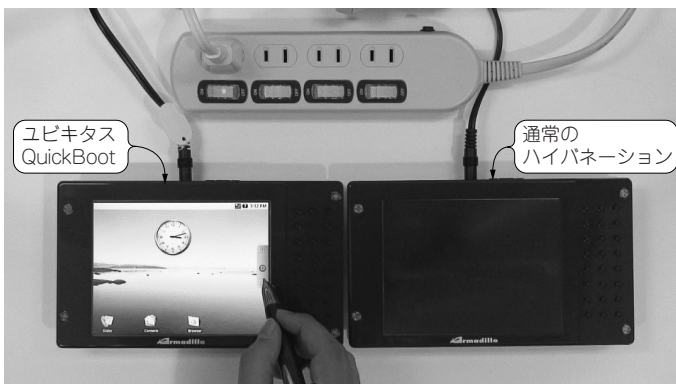


写真1 QuickBootの動画デモ

<http://www.ubiquitous.co.jp/products/middleware/quickboot/>