

省電力デバイスにおける OS カーネルの研究

伊藤 僚
須栗 裕樹

1. 研究目的

本研究の目的は、省電力デバイスで用いるためのオペレーティングシステムのカーネルに必要な機能を設計・実装し、かつユーザーランド及びアプリケーションによって評価実験を行うことである。省電力デバイスとしては、第一に Android を搭載した高機能なマルチメディア端末であるスマートフォンを対象とするが、それ以外のスマートフォン、タブレット、また従来型の組み込みシステムにも対応可能とする。

2. 研究動機

今日、スマートフォンやタブレット等の高機能な省電力デバイスが普及しつつある。これらが従来の組み込みシステムと異なるのは、マルチメディア、PC やクラウドとの連携、GUI など、高度な機能を提供していることである。これらの OS として代表的なものが、iOS と Android の二つである。特に Android 端末は、2011 年 3 月に日本国内におけるシェアで iOS を逆転する [1] など、躍進が目立つ。

iOS と Android に共通しているのは、どちらも Unix ベースの OS だということである。カーネルとしては、iOS が Mach/BSD、Android が Linux であり、どちらもデスクトップやラップトップの Mac OS や Linux ディストリビューションとの連続性があるため、高度な機能を実装するのが容易である。

他方、これらは、数十年に渡る Unix の歴史を引きずっているため、最近の高機能省電力デバイスに最適化されているとは言い難い面もある。例えば、SoC として実装されるヘテロジニアスマルチコアへの対応、省電力とハイパフォーマンスの両立、制限されたユーザーインターフェース、ストリーム抽象化では扱いきれない多種多様な入出力、ライトウェイトなプロセス、従来のファイルシステムを超える概念、異なる粒度のリアルタイム性を担保するスケジューラ等が問題点としてあげられる。

3. 問題点

3.1 SoC として実装されるヘテロジニアスマルチコアへの対応

対象とする端末は SoC として CPU と GPU を一緒に搭載しているが、GPU のプロセスを UNIX のプロセスでは扱えない。

3.2 制限されたユーザーインターフェース

画面が小さく、タッチスクリーンのみであるということ従来 UNIX ではうまく扱えない。

3.3 ストリーム抽象化では扱いきれない多種多様な入出力

現状では一次元のバイトストリームによって処理しているが、マルチメディアデータに関しては対応しきれていない。

3.4 ライトウェイトなプロセス

UNIX はプロセスを新たに作るのも、一度終了した後再度立ち上げるのも大変である。ヘビーウェイトなプロセスは省電力性に悪影響を及ぼす為、ライトウェイトなプロセスにする必要がある。

3.5 従来のファイルシステムを超える概念

従来のファイルシステムはパスで場所を指定している。これを例えば RDB にすることで検索等も簡単になる。

Android においては UNIX のようにパスがそのまま見える。しかし、iPhone の場合はそんなことはない。内部的にはパスを使っているがそれをユーザーには見せないという点で親切である。エンドユーザーとしては iOS の方が親切でわかりやすい。人に分かりやすい UI を素直に作ろうとした場合、従来のファイルシステムを超えるものを作る必要がある。

3.6 異なる粒度のリアルタイム性を担保するスケジューラ

処理によって 1 秒待てるものもあれば、0.01 秒も待てないものもある。そのように桁が違うリアルタイム性に対して UNIX のカーネルは対応できていない。

4. 開発環境

VirtualBox を用いて、Ubuntu10.10 上で作業を行った。

クロスコンパイラとして Sourcery G++ Lite 2009q3-67 を使用した。

また、環境を構築する過程でシェルスクリプトを実行する必要があったのだが、うまく実行できなかったことがあった。結局、パーミッションがうまく設定できていなかったためであり、ターミナルから `chmod a+x` シェルスクリプトのパスで実行権限を与えてやるときちゃんと動くようになった。因みに `a+x` とは、すべてのユーザーに実行権限を与えるという設定である。そのことがあってからは、パーミッションをきちんと意識するようになった。

5. 対象とする端末

Samsung 製 Galaxy S の国内向けモデル SC-02B(DOCOMO)

主な仕様 [2]

- 筐体サイズ: 64.2 x 122.4 x 9.9mm、重さ約 118g
- CPU : Samsung Intrinsicity S5PC110 1GHz
- GPU : PowerVR SGX540
- メモリ : RAM 512MB、ROM 16GB

- OS : Android2.3.3(Gingerbread)
- バッテリー : 1,500mAh、連続待受 510 時間、連続通話 380 時間

6. 評価と結果

5.1 Ver.0.1

標準のソースコードを、前述した開発環境でコンパイルしたバージョンである。SIM カードを挿した状態で数時間運用してみたが、電話の発着信・メールの送受信・ネットワークへの接続等、通常の使用で想定される機能はすべて問題なかった。これにより、コンパイル環境は問題ないと判断した。

5.2 Ver.0.2

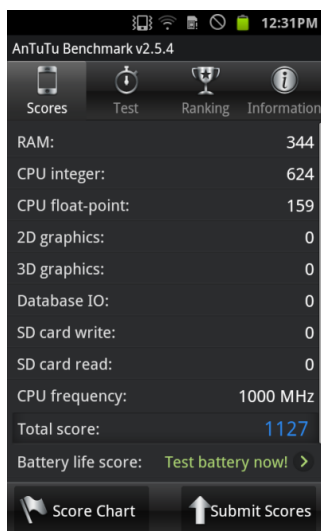
起動と同時に root 権を取得できるよう、initramfs に手を加えたバージョンである。root 権を取得する方法はいくつかあるが、Android の起動スクリプトである init.rc に記述を追加して、必要ないくつかのファイル (BusyBox、SU バイナリ等) を/system 以下のディレクトリにコピーするシェルスクリプトを起動させている。

root 権を取得できているかどうかは、root 権限を必要とするクロック制御用アプリケーション (SetCPU v2.2.4) で確認したが、問題なくアプリに root 権限を付与できた。

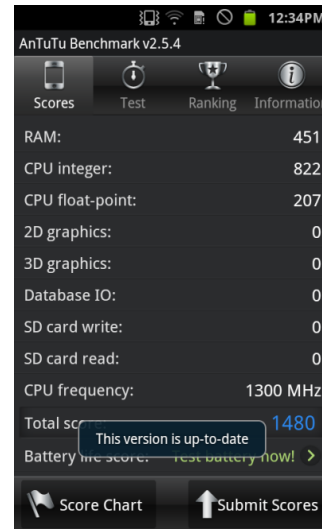
5.3 Ver.1.0

Ver.0.2 に加えて、パフォーマンスの向上のため、1300MHz にオーバークロックできるようなソースコードを一部書き換えたバージョンである。CPU の周波数の設定をするファイルにクロックの定義を変更・追加し、その定義を使用している部分を、新しい定義に対応するように修正した。

このバージョンでは、ベンチマークアプリケーション (AnTuTu v2.5.4) を用いて本当に高速化できているか試した。



1.標準クロック(1000MHz)時



2.オーバークロック(1300MHz)時

スコアは、多少ではあるものの伸びており、目的であったパフォーマンスの向上は達成出来た。

7. 今後の課題

これまでの研究では3で挙げた問題点を解決できていないので、今後はこれらの解決に取り組んでいきたい。特に、①「省電力とハイパフォーマンスの両立」と②「ファイルシステムの改良」の二つの課題に重点を置いていく。

①に関しては、待機時の電力消費を極力抑える方向で改良を加える。具体的には、待機時のデータ通信を一切カットし、定期的に通信をリクエストするという機能を、電波法との関係等も考えつつ実装したい。

②については、ファイルシステムについての理解を深めるためにも、検索を速くすることを当面の目標とする。どのような形になるかは分からないが、Windows でいうインデックスを貼ることによって速くするというようなことを試したい。また、Linux ディストリビューションはファイルシステムを追加するのが容易なので、新しくファイルシステムを追加するというようなこともやってみたい。

参考文献

- [1] スマートフォンの利用が加速！ グーグル・アンドロイド OS がアップル iOS のシェアを抜く！ - comScore, Inc

http://www.comscore.com/jpn/Press_Events/Press_Releases/2011/6/Google_Android_Leads_Acceleration_in_Smartphone_Adoption_in_Japan

2012年6月20日

- [2] GALAXY S SC-02B サポート情報：サービス・機能とスペック | お客様サポート | NTT ドコモ

<http://www.nttdocomo.co.jp/support/utilization/product/sc02b/spec.html>

2012年6月20日