

C言語でアプリケーションを開発できるフレームワーク Android NDK

花田 政弘

Android は Java ライクな言語でプログラムを開発する。しかし組み込み機器では、既存資産のライブラリを使用したり、性能上の要請から C/C++ 言語でプログラムを記述したい。そこで導入されたのが Android NDK (Native Development Kit) だ。NDK を使えば、Android 上から C/C++ 言語で記述したプログラムを呼び出せる。
(編集部)

Android プラットホームは、Linux カーネルを中心に持つ OS として構成されています。しかし、そこで動作するアプリケーションを作成するためのコードは Java ベースの仮想マシン Dalvik 用として記述する必要があり、既存の C や C++ といった言語で記述されたプログラムをそのまま流用することができません。

Android NDK (Native Development Kit) は、こうした C や C++ で書かれコンパイルされたオブジェクト・コード、いわゆる「ネイティブ・コード」を Android アプリケーションから使用できるようにするための開発ツールです。

ここでは、この Android NDK を使用してネイティブ・コードを埋め込んだアプリケーションを開発し、Android が動作する実機「Armadillo-500 FX」上で実行するまでの手順を中心に紹介します。

1. Android NDK の特徴

前述したように、Android NDK を使うとアプリケーションへネイティブ・コードで作成されたコンポーネントを埋め込むことができます。こうして作成したアプリケーションは、ただ動作するというだけではありません。ネイティブ・コードは実機上でそのまま動作する（仮想マシンを使用しない）ため、Dalvik 上だけで動作するアプリケーションと比べて速度面でメリットがある場合もあります。

Android Developers サイトの NDK ページ^{注1}にある記

注1: Android 1.6 NDK, Release 1,

http://developer.android.com/sdk/ndk/1.6_r1/index.html

事「What is the Android NDK?」にも注記されているように、NDK を使ってもネイティブ・コードのみによって構成された Android アプリケーションを作れるわけではありません。メインとなる実行環境はあくまでも Dalvik 仮想マシンのままなので、この点では NDK の使用により複雑化します。そうしたことから、常に必ずパフォーマンスが上がるわけではないことに注意してください。

同記事にあるように、NDK の使用が向くといえるのは「単体で完結し、メモリ・アロケーションを伴わず、CPU を酷使するもの」のような例でしょう。

2. Android NDK で使用できるライブラリ

現在の Android 1.6 NDK には、以下のヘッダ・ファイルが含まれています。

- libc (C ライブラリ) ヘッダ
- libm (算術ライブラリ) ヘッダ
- JNI インターフェース・ヘッダ
- libz (zlib 圧縮ライブラリ) ヘッダ
- liblog (Android ログ記録ライブラリ) ヘッダ
- OpenGL ES 1.1 (3D グラフィックス・ライブラリ) ヘッダ
- C++ サポートのための最小構成ヘッダ

Android に用意された C ライブラリは (たとえば glibc : GNU C Library) 標準的な libc ではなく、Android 独自の機能限定版 C ライブラリである Bionic であることに注意する必要があります。Bionic には ANSI C 標準の関数が存