

# 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プラットフォームは、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<sup>注1</sup>にある記

注1: Android 1.6 NDK, Release 1,

[http://developer.android.com/sdk/ndk/1.6\\_r1/index.html](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