第4章 実行するためのお膳立てを行うホスト・コードの処理内容から

初めてのGPUプログラミング…OpenCLでHello World

本橋 弘臣

表1　処理①…OpenCLのプラットフォームIDとデバイスIDを取得する

項　目	内　容
処理概要	システムに存在するOpenCLデバイスの一覧を取得して，使用するデバイスを決定する

ソースコード	`// OpenCL プラットフォーム ID を取得する`
	`ret = clGetPlatformIDs(MAX_PLATFORM_NUM, platform_ids, &ret_num_platforms);`
	`printf("clGetPlatformIDs(): num_platforms=%d\n", ret_num_platforms);`
	`// OpenCL デバイス ID を取得する`
	`ret = clGetDeviceIDs(platform_ids[0], CL_DEVICE_TYPE_DEFAULT, MAX_GPU_UNIT, device_ids, &ret_num_devices);`
	`printf("clGetDeviceIDs(): num_devices=%d\n", ret_num_devices);`
	`device_id = device_ids[0];`

実行結果	`clGetPlatformIDs(): num_platforms=1` `clGetDeviceIDs(): num_devices=1`
説　明	• プラットフォーム：OpenCLランタイム・ライブラリのこと．もし，システムにランタイム・ライブラリが2組インストールされていれば，プラットフォームIDが2つ得られる • デバイスID：OpenCLデバイスを識別するID．システムに同一メーカのGPUが2枚存在していれば，2個のデバイスIDが得られる

ここからは，第2部 第3章で開発環境を構築したラズベリー・パイ上で動作するGPUプログラムを作成していきます．

本稿では，OpenCL版のHello Worldプログラムを作成します．動作としては，標準出力に「Hello, World」と表示するだけとシンプルですが，GPUで動作させるためのお膳立てが必要です．　（編集部）

ホスト・コードの内容

● ホスト・コードの役割

リスト1のソースコードは，メインCPU側で実行するホスト・コードです．ホスト・コードはGPU上でカーネル・コードを動かすためのお膳立てを行う役目を担っており，具体的には次のような処理を行っています．

- 処理①：OpenCLデバイスIDの取得
- 処理②：使用するデバイスの選択
- 処理③：OpenCLデバイス情報の表示
- 処理④：OpenCLデバイスの実行準備
- 処理⑤：OpenCLカーネル・コードの実行準備
- 処理⑥：OpenCLカーネル関数の実行

● 処理①：OpenCLデバイス情報の取得

次のAPIを呼び出すことにより，OpenCLデバイス情報を取得できます．

- `clGetPlatformIDs()`
  …プラットフォームID取得
- `clGetDeviceIDs()`…デバイスID取得

詳細は**表1**に示します．

● 処理②…使用するデバイスの選択

リスト1では，システムに存在するOpenCLデバイスのうち，単純に1台目のOpenCLデバイスを選択するようにしています．

```
device_id = device_ids[0];
```

もしシステムに複数のOpenCLデバイスが存在している場合には，`clGetDeviceInfo()`を利用し，各OpenCLデバイスのメーカ名や型番情報を得て使用するデバイスを選択するようなコードを書く必要があります．

● 処理③…OpenCLデバイス情報の表示

次のAPIを呼び出すことにより，各OpenCLデバイスのベンダ名やデバイス名を取得できます．

- `clGetDeviceInfo()`
  …OpenCLデバイス情報の取得