

Maple 13におけるプログラミング機能の変更点

Maple 13ではプログラミング機能に関して以下の変更が施されています。

▼ タスクプログラミングモデル

[Threads\[Task\]](#) サブパッケージにより、新しいマルチスレッドプログラミングモデルが提供されます。

[Task Programming Model](#)を使用して、拡張性のあるマルチスレッドコードを記述することができます。このモデルは、マルチスレッドコードの記述手順を簡素化するもので、多くの場合、同期化ルーチンを使用する必要がなくなります。いずれは、[Threads\[Create\]](#)を呼び出すのではなく、タスクプログラミングモデルを使用して大部分の並列コードを記述するようになるはずです。

▼ OpenMaple[C]

[OpenMaple\[C\]](#) パッケージに新しいコマンドが追加されました。

- [MaplePushErrorProc](#) および [MaplePopErrorProc](#) により、エラー発生時に実行されるコールバック機能に外部コードを登録することができます。
- [MapleGetInterruptValue](#) 機能は、カーネルが割り込みを受けた場合に `true` を返し、それ以外の場合は `false` を返します。この機能は、割り込みの発生を確認する目的で使用することができますが、外部コードは、[MapleCheckInterrupt](#)を呼び出す前にクリーンアップ作業の実行が必要です。
- 新しい [MaplePushGMPAllocators](#) および [MaplePopGMPAllocators](#) 機能により、スレッドセーフな方法を用いて、GMP メモリ割当て機能の登録および登録削除を外部コードで実行することができます。

▼ プリプロセッサの拡張

プリプロセッサ命令が記述された Maple 言語ファイルを、`read` 文を使用して読むことができるようになりました。これにより、旧バージョンより柔軟な形で `$include` および `$define` などの命令を使用することができます。

▼ マルチラインコメント

お使いのコードにマルチラインコメントを追加できるようになりました。マルチラインコメントは記号(`*` と `*`)で区切ります。マルチラインコメントは、2-D math では使用できません。Maple のコメントに関する詳細については、[comment](#) を参照してください。

▼ 外部プロセスの起動

`system` コマンドに、新しいプロセスで外部アプリケーションを実行できる新オプションが追加されました。Maple はこのプロセスを妨げることなく、完了するまで待機しますが、起動後すぐに再開します。

▼ 参照

[Index of New Maple 13 Features, Language and System Updates in Maple 13](#)