

並列処理のアップデート:

- [並列ガベージコレクタ](#)
- [スレッドローカル記憶テーブル](#)
- [スレッドローカルデータ](#)

Maple 17 では、メモリ管理システムに並列処理が導入されました。これにより、大規模な計算に役立つローレベルでの処理が向上しました。さらに、新しいプログラミング構造が追加され、並列コードをより簡単かつ安全に記述できるようになりました。

並列ガベージコレクタ

Maple 17 では、ガベージコレクタがマルチコアを活用できるようになり、順次操作も高速になりました。また、並列コードを実行している場合、スレッド別のメモリ管理を行えるようになり、並列処理全体が改善されました。

詳細は、[並列ガベージコレクタ](#)を参照してください。

スレッドローカル記憶テーブル

Maple 17 では、プロシージャが実行されているスレッドごとに、異なる記憶テーブルを持つようになりました。スレッドローカル記憶テーブルを持つプロシージャが並列で実行される場合に、記憶テーブルの変更によってスレッドの安全性に関する問題が生じることがなくなりました。この変更によって、記憶テーブルを使用するスレッドセーフな関数を簡単に記述できるようになりました (特に、記憶テーブルの直接操作が必要な場合など)。

プロシージャに共有オプションを定義することにより、スレッド間で共有する共有記憶テーブルを使用するプロシージャも引き続き作成できます。これは、従来の Maple バージョンと同じ機能です。

詳細は、[スレッドローカル記憶テーブル](#)を参照してください。

スレッドローカルデータ

Maple 17 では、モジュール内のローカル変数をスレッドローカルとして宣言できるようになりました。これは、変数にアクセスするスレッドごとに異なる値を格納することを意味します。これにより、状態を保持しておく必要がある複雑なアルゴリズムをスレッドセーフな方法で記述できます。

詳細は、[スレッドローカルデータ](#)を参照してください。

参照

[Maple 17 の新コマンドと新パッケージの一覧](#)、[スレッドローカル状態の保持](#)、[プロシージャの remember オプション](#)