

# メモリ領域

Maple のメモリ割り当て機能の大きな改良点として、複数メモリ領域が追加されました。

メモリ領域が複数になったことで、以下のような利点があります。

- メモリの割り当てが効率的になりました。これにより、予約領域のサイズを最初に指定しておかなくても、より多くのメモリを使用して、より大規模な問題を解けるようになります。
- 同じマシンで実行している別のプロセス (Maple のマルチコアおよびグリッド内で並列ノードとして起動されている複数の Maple カーネルを含む) とのリソースの共有が改善されました。
- 断片化の軽減とローカルでのキャッシュによる処理効率の向上に加えて、並列コード実行時の処理効率も向上しました。

従来は、メモリ内に連続した大きなヒープ領域を確保しておいて、それを必要に応じて割り当てたり、ガベージコレクションを実行したりすることにより管理していました。最初は、メモリ割り当て要求を処理するために、この確保されたメモリの小さな断片が割り当てられます。所定の割り当てが使い果たされると、ガベージコレクションが実行され、使用されていないメモリが解放されます。計算でより多くのメモリが要求されると、メモリ管理システムがヒープ内のメモリをより多く割り当てます。

メモリ領域を追加することで、単一の大きな連続メモリ領域 (ヒープ) による制限がなくなります。代わって、要求に応じて追加領域を組み込むことができるようになりました。さまざまな種類のメモリ領域が用意されています。たとえば、スレッドローカル領域 (512 ワード以下) による小さな割り当て、グローバルスレッド共有領域 (1MB 未満) の中規模の割り当て、および独自の領域として個別に割り当てられる大きなメモリブロックがあります。これにより、利用可能なメモリを Maple がより効率的に管理できるようになりました。具体的には、データの局所参照性の改善や断片化の減少によって、キャッシュ機能が向上しました。

Maple の従来リリースでは、大量のメモリが割り当てられた場合、そのメモリは残りのセッションのために Maple の管理下に置かれていました。個別に調整されたメモリ領域を割り当てて 1MB を超えるメモリ要求を処理することにより、Maple 17 ではこの大きなメモリブロックを必要がなくなりしたいオペレーティングシステムに返すことができるようになりました。

たとえば、以下のコード例は 2 つの大きな行列 A および B を割り当て、何らかの操作を行ってから A と B を上書きし、割り当てられたメモリガベージをレンダリングします。ガベージコレクションを明示的にコールした後でメモリの使用量が低下していることに注目してください。

```
> restart;
> size := 8192 :
A := Matrix(size, size) :
B := Matrix(size, size) :
# Do some work. For example, C := LinearAlgebra:-MatrixMatrixMultiply(A, B):
kernelopts(bytesalloc) ,
1024
# Report how many KB Maple is using.
```

1057160

(1)

```
> A := 0 : B := 0 : C := 0 : gc( );  
    $\frac{\text{kernelopts}(\text{bytesalloc})}{1024}$ ;  
   # Now how much memory is in use?
```

57728

(2)

\* ここでは、Maple のメモリ管理システムの具体的な挙動を説明するために `gc()` を明示的にコールしています。通常は、ガベージコレクションの周期を開始するタイミングは Maple が自動決定するように設定しておきます。

## 参照

[gc](#)、[kernelopts](#)