SEGGER Embedded Studio 6 - SEGGER 社独自のリアルタイムメモリアロケータ

SEGGER Embedded studio ver.6 では、リアルタイムメモリ管理機能を強化し、メモリの割り当てと解放効率と応答時間を向上させ、C++ を利用したアプリケーションのリアルタイム性を向上させています。

評価モデルとテストアプリケーション - 自社有志プロジェクトでのチェスエンジン「emChess」の開発

まずは Windows 用のシミュレーションで、GUI ソフトウェア「emWin」と RTOS「embOS」を利用して開発し、SEGGER Embedded Studio で組込ターゲットで動作させる手法をとります。

チェスエンジンでの C++ STL 利用

チェスエンジンの開発には、SEGGER の独自リンカ開発でも利用されている C++ STL を採用しました。

チェスエンジンの開発例

まずチェスボードの位置を記述するために典型的な 8x8 のピース（駒）配列と、タイプ別に素早く検索するためのピースリストをセットしました。このピースリストは、STL 標準のリスト「std::list<Piece>」です。ピースリストは、キャプチャ（駒取り）やプロモーション（成り駒）後、変更され、STL ピースリストは反復可能なコンテナに対する C++ の for ループで簡単に実行できます。

```
1 | for (auto p: Board.Pawns) GenPawnAttacks(p);
```

駒の動きを制御するムーブジェネレータは、ボードの位置を調査し、その位置で、動作できる有効な動きのリストを返しますが、その手の善し悪しの評価は行いません。動きのリストは、「std::list<Move>」として、新しい移動「m」が生成されると次の様に記録します。

```
1 | MoveList.push_back(m);
```

Transposion Table (転置テーブル)

このチェスエンジンでの特定のボード位置での評価、白と黒の優勢評価、どれだけ優れているかを記録する転置テーブルを使用します。これは、高速の検索を実現するために重要なデータ構造で、異なる駒の動きの命令によって、同じ位置に到達することができ、これらの位置表かは、常に同じとなります。

転置テーブルは、ボードの位置、評価をマップします。実際、ボード位置の Zobrist ハッシュを評価として、マップし、ボード全体の位置を比較して、等価性を評価することは、要求の高い事項です。
これは、メッセージの暗号化ハッシュに似た概念で、1 つの部分位置が異なる 2 つのボード位置は、大きく異なるハッシュを生成し、2 つのボード位置が同じ Zobrist ハッシュ (一般に衝突する) を生成する可能性は低くなります。
転置テーブルは STL「std::map<U64, int> 」として記述され、64 ビットの Zobrist ハッシュをセンチポーン差での評価にマッピングします。従って位置が記録されるたびに、マップを更新する必要があります。この動作では、STL マップは一般にツリーとして実装されるため、記録の挿入毎に新しいツリーノードを作成し、関連性がなくなったら解放する必要があります。つまり、多くのヒープ操作が要求されます。

時間制御における不都合

チェスエンジン内では、STL コンテナを操作するたびに、メモリの動的割り当てと解放が必要になります。
ゲームプレイ中、メモリーアロケータは、様々なサイズのブロックをランダムに割り当てたり、解放したりします。これはアプリケーションの動作として、問題につながります。

ラピッドや、ブリッツのような時間制限の短い早指しチェスをプレイする際は、一手に時間かけすぎてはならないため、1 ゲームの手数を初期推定値として 50 手に設定、時間制限を手数で割って、一手に割り当てられる時間を取得、RTOS「embOS」でのタイマーは、検索開始時にスケジュールされ、時間切れになった場合、検索は中止され、「これまで見つかった最良の動き」が選択されます。

テストにおいて、いくつかの動きが想定以上に長くかかっていることを発見しました。原因として、検索回数が大きくなると、標準のメモリアロケータがメモリの割り当てや解放する際に、想定以上の時間がかかることがあるということです。これは異なるサイズのブロックの割り当て要求を管理する時、メモリが断片化されるためです。
標準アロケータでは、単一のフリーリストがあり、そのリストで最適なものを選択するため、すべてを検索します。また解放時には、フリーになった領域を横断合算することで、解放されたブロックをアドレス順に挿入します。(一般的に低いアドレスでブロックを解放する方が、高いアドレスでブロックを解放するより、高速のため)
一般的な組込アプルケーションにおいて、動的にストレージをアロケーションする事はそれほど問題になりませんが、チェスエンジンのようにヒープ操作（割り当てと割り当て解除）を断続的に利用するため、C++ STL を利用する場合は、問題が発生します。

SEGGER 独自開発のアルタイムアロケータ

私たちは、いくつかのツールチェーンのアロケータで、「Doug Lea の dlmalloc」を使用している事を確認しました。これはよく評価されていますが、このアロケータさえ、一定時間の割り当て・解放を保証するものではないため、チェスエンジンのようなヒープ操作の多いリアルタイムシステムにおいては、問題が出る可能性があります。

この問題を解決するために、割り当てと解放にかかる時間を一定とするような新しいリアルタイムアロケータの開発につながりました。これを実現するために SEGGER のヒープマネージャでは、すべての要求に対し、最適 (Best) を目指さずに適合（Good）で対応します。

新しいアロケータは、TLSF(two-level segregate fit) モデルを採用しています。また Cortex-M4 での実装 (初期化、割り当て・解放機能) にわずか 664Byte のコードで対応しています。割り当ての最悪ケースでは、125 個の実行命令と、プロセッサで実行された命令の割り当て解除 95 個となります。

ヒープの断片化について、新しいリアルタイムヒープは、様々なサイズの要求によって過度に断片化されないことも優れた特性として持っています。

他のツールチェーンに搭載されているアロケータとの比較

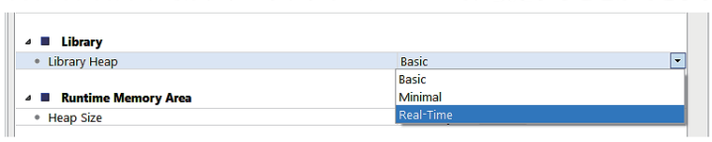
SEGGER のリアルタイムヒープは、様々な商用・フリーのツールチェーンで採用されている「Doug Lea の dlmalloc」と比較することはできません。なぜなら、このアロケータは一定時間の割り当て機能を提供していないためです。dlmalloc は「最悪の場合にどの程度悪いパフォーマンスとなるか。」という検証論文はありますが、あまり良い比較とはいえません。

コードサイズの比較はシンプルで、SEGGER の新しいアロケータの実装と同じ条件、Cortex-M4 での実装（初期化、割り当て・解放機能）で、一つの事例として、他商用ツールチェーンでの dlmalloc の実装サイズは約 8 倍、5,332 Byte となっています。

パフォーマンス

ヒープ・パフォーマンスは、様々な方法で測定でき、最終的にはアプリケーションの割り振りプロファイルと解放プロファイルに依存します。このチェスエンジンでは、リアルタイムアロケータのパフォーマンスの向上を確認できました。

■ Embedded Studio では、ドロップダウンメニューで、設定することができます。



■ 標準ヒープを使用して、168MHz の Cortex-M4 でチェスエンジンのムーブジェネレータ（駒の動き生成）でベンチマークを実行すると、結果は次のとおりです。

```
Starting position

Depth  1:     1.354 ms  OK         20 nodes        14771 nodes/s
Depth  2:    27.591 ms  OK        400 nodes        14497 nodes/s
Depth  3:   592.270 ms  OK       8902 nodes        15030 nodes/s
Depth  4: 13058.626 ms  OK     197281 nodes        15107 nodes/s
```

```
Starting position

Depth  1:     0.944 ms  OK         20 nodes        21186 nodes/s
Depth  2:    18.923 ms  OK        400 nodes        21138 nodes/s
Depth  3:   405.463 ms  OK       8902 nodes        21955 nodes/s
Depth  4:  8949.222 ms  OK     197281 nodes        22044 nodes/s
```

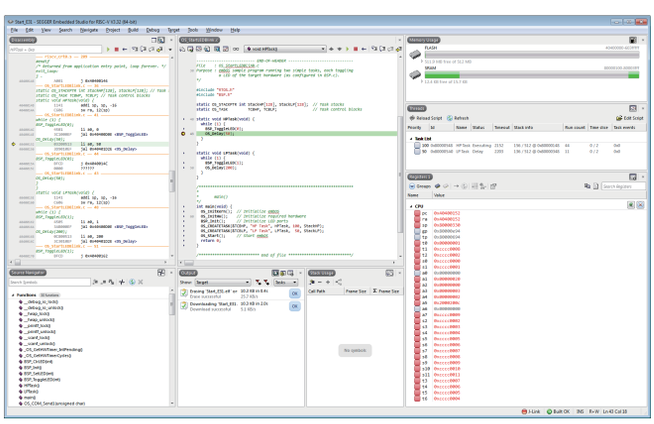
リアルタイムヒープを利用するとパフォーマンスが 46% 程度向上したことを確認できました。

SEGGER の新しいリアルタイムアロケータは小型で、C または C ++ でコーディングされたリアルタイムシステムに優れたパフォーマンスを提供します。リアルタイムシステムで STL を使用することが実用的になりました。
C++ で多くのオブジェクトを使用するアプリケーションでは、SEGGER リアルタイム アロケーターを使用するとアプリケーションによって異なりますが、パフォーマンスが向上します。今回の習作アプリケーション、emChess は期待通りの時間管理を実現しました。

Embedded Studio は、Arm / RISC-V マイコン開発のベースとなる開発ツールです。

Embedded Studio は商用統合開発環境に求められる全てを低コストに実現します。標準的なコンパイラである「GCC/LLVM」と SEGGER 社独自のコンパイラの 3 つのコンパイラを同梱しています。この独自コンパイラと同じく SEGGER 社が開発したリンカにより、お客様アプリケーションのパフォーマンスを最大限に発揮することができます。

また Arm 開発におけるデファクトスタンダードデバッガ「J-Link」の性能・機能を引き出し、ソフトウェア開発の効率性を上げ、品質を向上させます。商用統合開発環境の利点である「マイコンベンダーに依存することのない開発プラットフォーム」
ツールとして、基本性能・使いやすさ・ソフトウェア解析ツールとしての付加価値をコストバランス良く開発者様へ提供します。



Professional Tools FREE for non-commercial use

■ 高性能 SEGGER オリジナルコンパイラ
■ Clang/LLVM 、GCC C/C++ コンパイラ同梱
■ マルチスレッドコンパイル・ビルド対応
■ SEGGER 社により最適化された C ランタイムライブラリ
■ J-Link と統合化されたデバッガ
■ 高機能プロジェクトマネージャ
■ CPU サポートパッケージ、簡単なプロジェクト立ち上げ
■ 非商用利用（評価・学術用途）無償フルパッケージ

株式会社エンビテック
〒 130-0021 東京都墨田区緑 4-8-8 中井ビル 4F
Tel. 03-6240-2655 / Fax. 03-6240-2656
e-mail. sales@embitek.co.jp