メモリバンクのサンプル

*このサンプルは、2015 年 11 月の Xbox One XDK 以降と互換性があります。*

# 概要

このサンプルは、メモリバンクを管理するために利用できるいくつかの異なる方法を示します。

* ランダム - VirtualAlloc によって割り当てられたベースラインメモリブロック。
* 固定 - 特定のメモリアドレスにメモリを割り当てる方法を示します。
* 読み取り専用 - メモリを読み取り専用に変換する方法を示します。
* バンクの切り換え - ローテーション可能な一連のメモリバンクを作成する方法を示します。
* 共有 - 同じ物理的な場所を共有する複数のメモリバンクを作成する方法を示します。

# サンプルの使用

各デモンストレーションのためにコントローラの対応するボタンを押してください。成功または失敗を報告します。

# 実装上の注意

## ランダム

このデモは、他のデモのベースラインとして機能します。VirtualAlloc を使用してメモリブロックを割り当て、次にディスクからメモリブロックにバイナリツリーを読み込みます。ディスクからのデータは修正する必要があります。内部的に、バイナリツリーはそれが割り当てられているメモリブロックにインデックスを格納します。ベースアドレスはプログラムの実行ごとに変わるため、バイナリツリーの内部ポインタはすべてインデックスに基づいて更新する必要があります。

## 固定

このデモは、ランダムアドレスのデモンストレーションと同じように機能します。主な違いは、ベースアドレスがデータファイルに格納されていることです。これにより、プログラムは、プログラムを実行するたびに、同じメモリ位置に同じブロックを再作成できます。これは、内部インデックスが不要で、ポインタの修正が不要であることを意味します。CPU オーバーヘッドがほぼゼロで、はるかに速くデータをディスクからロードできます。

## バンクの切り替え

このデモは、切り替え可能な複数のメモリバンクを作成する方法を示しています。これにより、2 ブロックの物理メモリの仮想アドレスがスワップされます。3 つ以上のメモリバンクを使用でき、交換することができます。いくつかの用途として、ログデータ、再生の保存に使用するフレームデータ、およびバッファー間でメモリを移動するその他の場所の作成があります。これにより、メモリの移動操作を削除でき、劇的にパフォーマンスが向上します。

## 共有

このデモンストレーションは、同じ物理ブロックを指す複数のメモリブロックを作成する方法を示しています。隣り合う 2 つの仮想ブロックが作成されると、境界のコピーに複数メモリのコピー操作を必要としないリングバッファーが可能になります。データが仮想ブロック側に書き込まれるため、物理ブロックの開始に自動的に合わせられます。

別の用途は、複数の仮想ポインターを、読み取り/書き込みや読み取り専用など、異なる許可で作成することです。write-combine やキャッシュ可能など、キャッシュの用途を調整する特定の種類の許可をミックスする場合は注意が必要です。CPU が 1 つのポインターまたはその他にアクセスするときに異なる方法でデータをキャッシュすることになり、追跡困難なバグが発生する懸念があります。

## 読み取り専用

このデモンストレーションは読み取り専用ページの作成方法を示します。1 つのブロックが読み取り/書き込みとしてマークされ別のブロックが読み取り専用としてマークされている共有デモンストレーションを使用して行います。このパターンの主な用途は、ランダムメモリ破損の追跡です。例えば、タイトルが使用する静的データです。ファイル読み込みシステムが、データの作成に読み取り/書き込みアドレスを使用できるとします。残りのタイトルはアクセスに読み取り専用ポインターを使用します。メモリ破損の問題が起こると、問題が発生しているまさにその場所で例外を引き起こします。

# 更新履歴

初回リリース 2016 年 10 月