

ゲーム開発のためのメモリアロケータ

– ヒープ、固定メモリ、デバッグ情報を効率的に扱う –

2014 年 2 月 24 日 初稿

板垣 衛

■ 改訂履歴

稿	改訂日	改訂者	改訂内容
初稿	2014 年 2 月 24 日	板垣 衛	(初稿)

■ 目次

■ 概略	1
■ 目的	3
■ 要件定義	3
▼ 基本要件	3
▼ 要求仕様／要件定義	3
■ 仕様の依存関係	4
■ データ仕様	4
■ 処理仕様	4

■ 概略

本書は、。

(メモ)

- ・メモリ断片化や確保数増加に対してパフォーマンスが発揮できる構造を第一とする。

※メモリマネージャのコンパクト化は、どちらかと言えば重視しない。

全ノードの双方向リンクリスト、割り当て済みノードの双方リンクリスト(不要かも?)、

空ノードのサイズ比較レッドブラックツリー、割り当て済みノードのアドレス

比較レッドブラックツリー(不要かも?)をサポートできると高速化できる。

- ・64bit アドレスを考慮。

- ・双方向

- ・断片化情報を常に管理。

- ・スモールヒープ連動を考慮。ただし、メモリマネージャとしては完全に独立。

アロケータによって、スモールヒープを活用するかどうか判断させる仕組み。

- ・【可能なら】PS3の1MB単位のメモリマップなどを考慮し、ページ単位で物理メモリを返却できるような構造を取る。

(例えば、128MBの論理メモリ空間に対して、16MBの物理メモリが1MBごとにまばらに割り当てられているような状態。1MBのメモリが空けば、その物理メモリを再割り当てして、2MBの連続領域として使用できるようにするなどの組み換えが必要に応じて自動で行われるようにする。)

- ・【可能なら】GPUメモリの管理に対応。

- ・メモリ再配置(コンパクション)に標準対応。

- ・【可能なら】参照カウント、ガベージコレクションに標準対応。

- ・【可能なら】固定サイズメモリプールを任意の目的で使用可能にする。

- ・スタック式メモリ管理とは完全独立。

ただし、スタック式メモリ管理の対応として、範囲メモリ確保(メモリ確保の最小と最大サイズを指定し、可能な限り最大サイズを確保しようとする)仕組みと、

リサイズに対応する。(リサイズは基本的に縮小方向のみ)

<追加>

- ・スラバアロケータ/Buddy Systemを参考に

- ・ピークメモリ計測標準化

- ・メモリカテゴリ別集計・リミット標準化

- ・スモールヒープの最適メモリ計測標準化 ※ピークメモリの割り当てサイズと各割り当てがピークの時のそれぞれの割り当て状態を保存

- ・メモリ汚し考慮

- ・動的スモールヒープと、スモールヒープの自動統計。

- ・ 2 段階ページ管理⇒1MB（仮想アドレスへの物理メモリ割り当て）、2KB（ページ）

簡易スラバロケータ

- ・ 1～4kb ページ管理
- ・ メモリ要求に合わせて、メモリオブジェクトを割り当てる
- ・ メモリ要求サイズは、 $(\max(\text{要求サイズ}, \text{アラインメントサイズ}) + (4-1)) \& \sim(4-1)$ で算出
- ・ 要求されたメモリブロックが無ければページからメモリブロックを割り当てる
- ・ 要求されたメモリブロックに対するページが無ければ/足りなければページを割り当てる(割り当て時の連続ページ数を考慮)
- ・ 要求されたページの空きが無ければ未使用ページを破棄して割り当てる(割り当て時の連続ページ数を考慮)
- ・ ページの確保ができなかった場合、一つ上のサイズのメモリブロックとして確保する
- ・ 全てのサイズのメモリ確保に失敗した場合、メモリアロケータに失敗する
- ・ メモリマネージャは、予め下記の事を定義しておく
 - 全体メモリサイズとメモリ
 - 確保可能なメモリサイズの最大値(4096 まで)
 - アロケータに失敗した場合の上位メモリブロック使用の制限値(通常、64(ヘッダーサイズ)+要求メモリサイズ(アラインメントを考慮)が使用ブロックサイズ+4(参照カウンタのサイズ)を超える場合、ヒープメモリの方が効率がいいので、そこで打ち止めとする)⇒例えば、8 バイトのメモリ確保の際、スラバロケータでは $8+4=12$ バイトを要し、ヒープメモリでは $64+8=72$ バイトを要する。この時、68 バイトブロックの確保までならスラバロケータの方がメモリ効率が良いが、それ以上はヒープメモリの方が良いので、そこでメモリ確保失敗を判断する。
- ・ メモリ確保/破棄をスレッドセーフにするかどうか(ライトウェイトミューテックス使用)
- ・ 各メモリサイズ(4～4096 の 4 バイトずつ 1023 段階)に対して、メモリブロック確保の許可(1bit)、参照カウンタの有無(1bit)、メモリブロック確保時の必要連続ページ数(2bytes)、初期メモリブロック確保数(2bytes)、最小メモリブロック確保数を定義(2bytes)をあらかじめ定義
 - 管理情報①:メモリサイズ、管理情報② の先頭インデックス
 - 管理情報②:メモリページ位置、参照カウンタページ位置、デバッグ情報ページ位置、連続ページ数、確保可能なブロック数、確保中のブロック数、ブロック使用状態ビットマップインデックス、同じメモリサイズの次の管理情報②のインデックス
 - 管理情報②用の空きノード情報(ビットマップインデックス)

※管理情報②は全ページ数分予め確保されている。また、仮に 1 ページを 8192 バイトとした場合、最小である 4 バイトのメモリなら 2048 個確保できる。2048 個のビットマップインデックスは、256 バイトで表現できる。更にこの 256 バイトのビットマップインデックスは、32 バイトで表現でき、更にそのビットマップインデックスは 4 バイトで表現できる。つまり、2048 個分の階層ビットマップインデックスは $256+32+4=292$ バイト (=73 個の 32 ビット値) で表すことができる。このビットマップインデックスの領域を効率的に確保することが課題。

- デバッグ情報: 全体使用中ページ数、全体メモリサイズ、各メモリサイズ or アライメントサイズに対する確保要求数合計、各メモリブロックサイズに対する割当中ページ数、使用中ページ数、使用中ブロック数、ミスアロケート数(上位ブロック使用)合計、アロケート失敗数合計 + 全体メモリサイズ更新時、初回・最終ミスアロケート時、初回・最終アロケート失敗時のデバッグ情報のコピー

※ミューテックスのような同期オブジェクトはプラットフォームに応じて実装を変えられるように、ラップしたクラスをライブラリに用意する

■ 目的

本書は、を目的とする。
色んな要件だけまとめておく。
残念

■ 要件定義

▼ 基本要件

- ・ を規定する。

▼ 要求仕様／要件定義

- ・ する。
 - する。

■ 仕様の依存関係



本書の仕様は、である。

■ データ仕様

あ

■ 処理仕様

あ

■■以上■■

■ 索引

索引項目が見つかりません。

ゲーム開発のためのメモリアロケータ

以 上