【１】主記憶管理　　　　　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

　限られた主記憶領域を効率よく、より多くのプログラムに提供する為に、実記憶方式から仮想記憶方式へと時代とともに次のように改善されてきた。

実記憶方式

単一区画方式

固定区画方式

可変区画方式

スワッピング方式

オーバレイ方式

ページング方式

セグメント方式

仮想記憶方式

【２】実記憶方式　　　　　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

１．固定区画方式（[①　　　　　　　　　　]方式）

|  |  |
| --- | --- |
| 方法 | 領域を固定的な区画に分割して、分割した区画ごとに複数のプログラムを記憶する。 |
| 利点 | 多重プログラミングが可能となる。 |
| 欠点 | ・[②　　　　　　　　　　　　　]（断片化）が区画ごとに発生する。（もったいない）  ・区画サイズより大きなプログラムは格納することができない。 |

図：固定区画方式

未使用領域

処理プログラム

主記憶

ＯＳ領域

処理プログラム

処理プログラム

未使用領域

未使用領域

区画1

区画2

区画3

２．可変区画方式（[③　　　　　　　　　　]方式）

|  |  |
| --- | --- |
| 方法 | 区画を取り払い自由に空いている領域に格納できる。 |
| 利点 | ・区画サイズより大きなプログラムを格納できる。  ・フラグメンテーションに対してコンパクションをおこなう。 |
| 欠点 | ・コンパクションによるオーバヘッド(ＯＳの動作時間)の増大となる。  ・主記憶より大きなプログラムが格納できない。 |

図：可変区画方式のコンパクション

未使用領域

処理プログラム

主記憶

ＯＳ領域

処理プログラム

処理プログラム

未使用領域

未使用領域

処理プログラム

主記憶

ＯＳ領域

処理プログラム

処理プログラム

未使用領域

３．[④　　　　　　　　　]方式

|  |  |
| --- | --- |
| 方法 | プログラムは補助記憶装置に格納しておき、必要に応じて主記憶内にスワップインしたり、逆に補助記憶へスワップアウトしたりする方式である。 |
| 欠点 | 主記憶より大きなプログラムが格納できない。 |

　 ※スワップイン＝ロールイン

　 ※スワップアウト＝ロールアウト

スワップイン

処理プログラム１

主記憶

ＯＳ領域

処理プログラム２

処理プログラム２

補助記憶

図：スワッピング方式

スワップアウト

≪範例≫

フラグメンテーションに関する記述のうち、適切なものはどれか。

ア　可変長ブロックのメモリプール管理方式では、いろいろな大きさのメモリ領域の獲得や返却を行ってもフラグメンテーションは発生しない。

イ　固定長ブロックのメモリプール管理方式では、可変長ブロックのメモリプール管理方式よりもメモリ領域の獲得と返却を速く行えるが、フラグメンテーションが発生しやすい。

ウ　フラグメンテーションの発生によって、合計としては十分な空きメモリ領域があるのに、必要とするメモリ領域を獲得できなくなることがある。

エ　メモリ領域の獲得と返却の頻度が高いシステムでは、メモリ領域返却のたびにガーベジコレクションを行う必要がある。

≪解答≫　ウ

【３】仮想記憶方式：主記憶装置の容量不足を補う為に用意されている仮想記憶を管理する

　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

１．[①　　　　　　　　　　　]方式

|  |  |
| --- | --- |
| 方法 | ・プログラムを物理的な一定の大きさの**ページ**単位（2K～4Kバイト）に分割し､主記憶領域をページ枠単位に分割し､必要時に自動的にスワッピングをおこない実行させる方式である。  ・仮想記憶装置から実記憶装置にページを転送することを**ページイン**､その逆に不要なページを実記憶から追い出す動作を**ページアウト**という。  ・**ページフォールト**と呼ばれる割り込みによって､ページアウト及びページインの動作がおこなわれる。 |
| 利点 | ①主記憶領域の制約を受けない。（ただし、補助記憶領域の制約は受ける）  ②一定の大きさのページでのスワッピングの為､フラグメンテーションが起こらない。 |
| 欠点 | ①スワッピングが激しくなると処理効率が低下する。（[②　　　　　　　　　]） |

図：ページング方式；ページイン・ページアウト

ページ１

補助記憶（ＤＡＳＤ）

ページ4

ページ2

ページ3

ページ5

ページ6

仮想記憶装置

ページ単位で格納

巨大プログラム

【静的再配置】

ページ1

主記憶

ＯＳ領域

ページ4

ページ3

ページ枠３

【動的再配置】

実記憶装置

ＤＡＴ(ページテーブル)

|  |  |
| --- | --- |
| ページ1 | 1100番地 |
| ページ4 | 1400番地 |
| ページ3 | 1300番地 |

ページイン

ページアウト

（１）**ＤＡＴ（動的アドレス変換機構）**

　仮想記憶装置の番地（仮想アドレス）と実記憶装置の番地（実アドレス）を変換するハードウェアをいい、ページテーブルによって管理している。

（２）**動的再配置**

　主記憶領域にロードされたプログラム（ページ）が、主記憶装置の異なる領域で実行されるように、実行中にこのプログラムに対して、新たな絶対アドレスを割り当てる方法をいう。

（３）**静的再配置**

　動的再配置に対して、格納したプログラムが記憶領域の位置を変えないやり方を静的再配置という。

（４）**ページリプレースメント**

　実記憶装置に空きがなくページインできないとき、現在常駐しているページをページアウトしなければならない。ここで、システムの処理効率を考えて、使用頻度の高いページを実記憶装置に常駐させておき、使用頻度の低いページは仮想記憶領域に退避させ、必要なときだけ実記憶装置に移す。これをページリプレースメントという。

このとき、実記憶装置のページを追い出す方法として、次の方法がある。

|  |  |
| --- | --- |
| [③　　　　　　]法  (Least Recently Used) | 主記憶装置のページ枠にあるページのうち、最後に参照されたときから最も長い時間が経過したページを補助記憶領域に追い出す。 |
| [④　　　　　　]法  (First-In First-Out) | 主記憶装置のページのうち、最も古くからあるページを補助記憶領域に追い出す。 |

　　　ページリプレースメントの手順

　ページフォールト→置き換え対象ページの決定→ページアウト→ページイン

※注意点

　仮想記憶方式は、主記憶領域の有効利用にはなるが、スワッピングをおこなうため処理速度は必然的に低下するのが前提であることを抑えておこう！！

≪範例1≫

ページング方式の仮想記憶において、ページ置換えアルゴリズムにLRU方式を採用する。主記憶に割り当てられるページ枠が4のとき、ページ1、2、3、4、5、2、1、3、2、6の順にアクセスすると、ページ6をアクセスする時点で置き換えられるページはどれか。ここで、初期状態では主記憶にどのページも存在しないものとする。

ア　1　　　　　　　　　イ　2　　　　　　　　ウ　４　　　　　　　　エ　５

　≪解答≫　エ

　ページングの状況を図に示すと次のようになります。

３

２

１

６

２

３

１

２

５

４

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ① |  |  |  | ⑤ |  |  |  |  | ⑥ |
|  | ② |  |  |  | 2 |  |  | ２ |  |
|  |  | ③ |  |  |  | ① |  |  |  |
|  |  |  | ④ |  |  |  | ③ |  |  |

　　１

　　２

　　３

　　４

　≪範例2≫

　　仮想記憶におけるページ置換えアルゴリズムとしてFIFO方式を採用する。主記憶のページ枠が3で、プログラムが参照するページ番号の順序が、4→3→2→1→3→5→2のとき、ページインは何回行われるか。ここで、初期状態では、主記憶には何も読み込まれていないものとする。

　　ア　2　　　　　　　　　イ　3　　　　　　　　ウ　5　　　　　　　　エ　6

　≪解答≫　ウ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ④ |  |  | ① |  |  |  |
|  | ③ |  |  | ３ | ⑤ |  |
|  |  | ② |  |  |  | ２ |

　　１

　　２

　　３

　　丸付きの数字がページインされたページ番号になりますので、ページインの回数は５回です。