

計算方法リスト

総資産回転率

資産をどれだけ有効に活用して売上を上げているかを表す財務指標で、以下の式で算出される。

$$\text{売上高} \div \text{総資産}$$

総資産営業利益率(ROA)

保有する資産に対してどれだけの利益を上げているかを表す財務指標で、以下の式で算出される。

$$\text{営業利益} \div \text{総資産}$$

売上高営業利益率

売上高に対してどれだけの利益を上げているかを表す財務指標で、以下の式で算出される。

$$\text{営業利益} \div \text{売上高}$$

営業利益

$$\begin{aligned} & [\text{売上高}] - [\text{売上原価}] - [\text{販売費・一般管理費}] \\ & [\text{売上総利益}] - [\text{固定費}] \end{aligned}$$

ホストコンピューターの処理時間

$$\begin{aligned} & (\text{毎秒当たりの送信量} - \text{処理時間}) \times \text{右辺との差分} = \\ & (\text{毎秒当たりの送信量} - \text{処理時間}) \end{aligned}$$

2 段階エディット法

共通バグの発見率から全体の総エラー数を推定する手法

総エラー数のうちグループA, Bのバグ検出確率" $P(A)$ ", " $P(B)$ "は、

$$P(A) = NA / N \quad \dots(1)$$

$$P(B) = NB / N \quad \dots(2)$$

と定義することができ、そこから $P(A)$, $P(B)$ が共に発生する" $P(AB)$ "を、

$$P(AB) = P(A) \times P(B) = NAB / N \quad \dots(3)$$

と導くことができます。

$$NA \text{の検出率} = NA / N$$

$$NB \text{の検出率} = NB / N$$

$$NAB \text{の検出率} = NAB / N$$

(3)の式に(1)と(2)を代入すると、

$$(NA / N) \times (NB / N) = NAB / N$$

$$\rightarrow N = NA \times NB / NAB$$

ページイン方式の割合の求め方

ページインだけの処理の割合を" P "とすると、ページアウトを伴う処理の割合は" $1 - P$ "で表すことができます。

それぞれの処理時間と平均処理時間の関係を表す次の式を解くと

$$(\text{ページアウトを伴わない場合のページインの処理時間}) = 20 \text{ミリ秒}$$

$$(\text{ページアウトを伴う場合、置換えページの選択、ページアウト、ページインの合計処理時間}) = 60 \text{ミリ秒}$$

$$(\text{1回のページフォルトの平均処理時間}) = 30 \text{ミリ秒}$$

$$20 \times P + 30 \times (1 - P) = 30$$

$$20P + 60 - 60P = 30$$

$$-40P = -30$$

$$P = 0.75$$

損益分岐点売上高

損益分岐点売上高 = 固定費 / (1 - 変動費率)
= 固定費 ÷ { (売上高 - 変動費) ÷ 売上高 }
変動費率 = 変動費 / 売上高

※

損益分岐点売上高：係る費用を収益でカバーでき、損益が「0」になって
これ以降は利益が出る、という売上高

変動費：原材料費、仕入原価、外注費、販売手数料など

固定費：人件費、地代家賃、リース料、広告宣伝費など

10 進数の値を 26 進数に直す

10進数の値 = 123

10進数123を26で割ると、商と余りは以下ようになります。

$123 \div 26 = 4 \text{ 余り } 19$

英字と数字は対応しているため、各桁を4→E, 19→Tと置き換えると26進数「ET」になります。

稼働率

$MTBF / (MTBF + MTTR)$

B+木インデックスを格納するノードへのアクセス回数のオーダを表す式

B+木インデックスでは探索範囲を1/nに狭めながら検索していきますが、B+木の深さはどの葉でも一定であるため、どの値を探索する場合でもほぼ同じアクセス回数になります。この深さは各節点を持つエントリ数(次数)で決まり、深さがh、次数がbであるB+木における葉の最大数(x)は次の式で表せます。

$bh = x$ (上の例でいえば $33 = 27$)

つまりx件のデータを探索する際のアクセス回数を示す深さhは、

$h = \log_b X$

このため、アクセス回数のオーダはXの対数「 $O(\log X)$ 」になります。

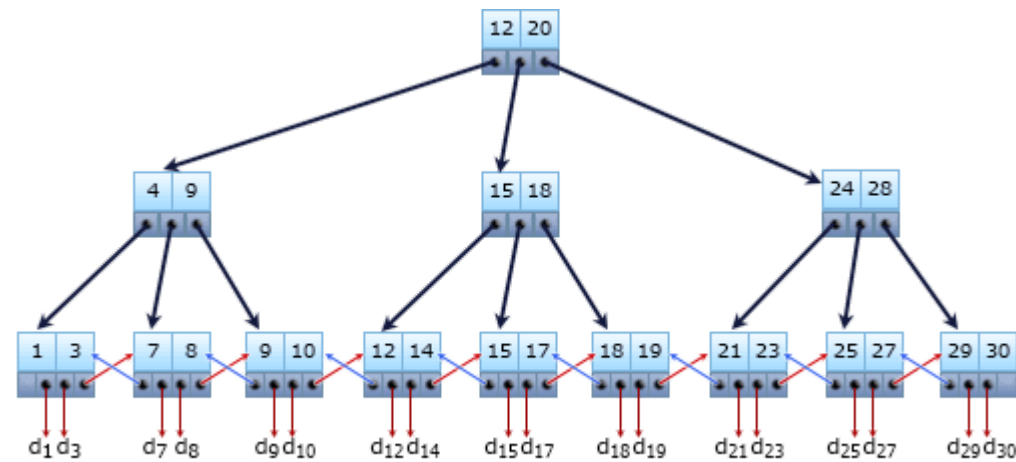


図 B+木インデックス

パイプライン制御

CPU処理を高速化させるため、1命令を、命令読出し(フェッチ)、解釈(デコード)、アドレス計算、オペランド呼出し、実行 というような複数のステージに分け、各ステージを少しずつずらしながら独立した処理機構で並列に実行することで、処理時間全体を短縮させる技法です。
またパイプラインの処理時間を求める公式「 $(I + D - 1) \times P$ 」を用いても計算することができます。

命令数(I)が20、深さ(D)が5、ピッチは1サイクルなので、

$(20 + 5 - 1) \times 1 = 24$ サイクル

$(I + D - 1) \times P$

- I : 実行する命令数
- D : パイプラインの深さ(命令を分割するステージ数)
- P : パイプラインのピッチ(ステージ毎の実行時間)

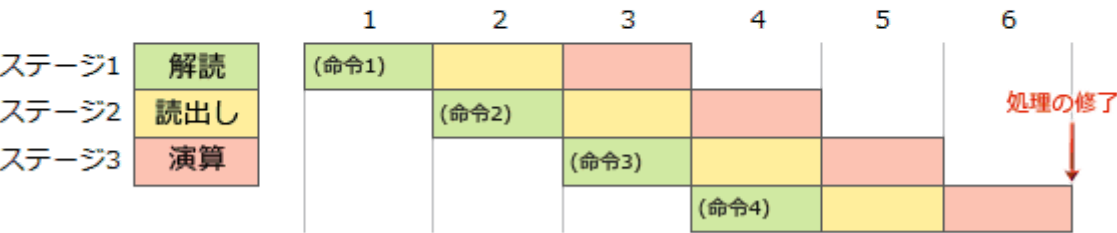


図 パイプラインの処理時間の公式

ページフォールト

ページング方式において要求されたページが主記憶上に存在しないときに発生する割込みです。ページアウト／ページインの処理は、ページフォールトの発生を契機に行われます。

[ページ0]

0は主記憶上にないためページフォールトが発生し、0がページインします。

0--

[ページ1]

1は主記憶上にないためページフォールトが発生し、1がページインします。

01-

[ページ2]

2は主記憶上にないためページフォールトが発生し、2がページインします。

012

[ページ3]

3は主記憶上にないためページフォールトが発生します。最終参照時刻が最も古いページ0がページアウトし、3がその位置にページインします。

312

[ページ4]

4は主記憶上にないためページフォールトが発生します。最終参照時刻が最も古いページ1がページアウトし、4がその位置にページインします。

342

[ページ0]

0は主記憶上にないためページフォールトが発生します。最終参照時刻が最も古いページ2がページアウトし、0がその位置にページインします。

340

[ページ2]

2は主記憶上にないためページフォールトが発生します。最終参照時刻が最も古いページ3がページアウトし、2がその位置にページインします。

240

[ページ4]

4は主記憶上にあるためページフォールトは発生しません。

240

[ページ3]

3は主記憶上にないためページフォールトが発生します。最終参照時刻が最も古いページ0がページアウトし、3がその位置にページインします。

243

[ページ1]

1は主記憶上にないためページフォールトが発生します。最終参照時刻が最も古いページ2がページアウトし、1がその位置にページインします。

143

[ページ4]

4は主記憶上にあるためページフォルトは発生しません。
143

[ページ5]
5は主記憶上にないためページフォルトが発生します。最終参照時刻が最も古いページ3がページアウトし、5がその位置にページインします。
145

したがって12回のページ読み込み中ページフォルトが発生するのは「10回」です。

| 割当て ステップ | 参照する 仮想ページ番号 | 実記憶ページの状態 | | |
|-------------|-----------------|-----------|------|-------|
| 1 | 0 | 0 1回 | - | - |
| 2 | 1 | 0 | 1 2回 | - |
| 3 | 2 | 0 | 1 | 2 3回 |
| 4 | 3 | 3 4回 | 1 | 2 |
| 5 | 4 | 3 | 4 5回 | 2 |
| 6 | 0 | 3 | 4 | 0 6回 |
| 7 | 2 | 2 7回 | 4 | 0 |
| 8 | 4 | 2 | 4 | 0 |
| 9 | 3 | 2 | 4 | 3 8回 |
| 10 | 1 | 1 9回 | 4 | 3 |
| 11 | 4 | 1 | 4 | 3 |
| 12 | 5 | 1 | 4 | 5 10回 |