

## 日付のシリアル値化

日付を時間、秒までふくめ、uint 型に変換します。  
実装は勢いで書いたので適当です。

uint のサイズは 4 バイトで、その範囲内で表せれる期間は、130 年ちょいです。  
なので、2006 年から数えて、2136 年くらいまでは数えれます。

一ヶ月を 31 日として数えているので、無駄がありますが、面倒なので難しい計算はしてません。

俺変換ですので、世間一般的な日付のシリアル値とは全く異なります。  
ご注意ください。

---

ちなみに、ただ単に、DateTime をシリアル値化したいだけなら、

```
// シリアル値化
long serial = DateTime.Now.ToBinary();
// シリアル値から復元
DateTime dt = DateTime.FromBinary(serial);
```

ってのがあつぽいので、これ使うといいかもしれません。(2.0 からかもしれんけど)

ていうか、オレ変換を使うより、

```
// 基準を決める
DateTime kako = DateTime.Parse("2006/1/1");
// (シリアル値化) 上記から今までの差の秒を求める
int second = (int)((TimeSpan)(DateTime.Now - kako)).TotalSeconds;
// (シリアル値から復元) 過去の日付に秒を足す
DateTime re = kako.AddSeconds(second);
```

これ使ったほうがいいのかも。

以下、俺シリアル値の使い方。

## 使い方

日付を uint に変換

```
uint now = enc(DateTime.Now);
```

uint を日付に変換

```
uint time = 9470952;
DateTime now = dec(time);
```

## コード

```

private int level(int lv)
{
    int[] levels = {12, 31, 24, 60, 60};
    Array.Reverse(levels);

    int res = 1;
    for(int i=0; i<lv; i++)
    {
        res *= levels[i];
    }

    return res;
}

private DateTime dec(uint s)
{
    long serial = s;

    long yearr = (long)((serial) / level(5));
    long monthr = (long)((serial -= level(5) * yearr) / level(4));
    long dayr = (long)((serial -= level(4) * monthr) / level(3));
    long hourr = (long)((serial -= level(3) * dayr) / level(2));
    long minuter = (long)((serial -= level(2) * hourr) / level(1));
    long secondr = (long)((serial -= level(1) * minuter));

    yearr += 2006;
    monthr += 1;
    dayr += 1;

    DateTime result = new DateTime((int)yearr, (int)monthr, (int)dayr, (int)hourr, (int)minuter,
(int)secondr);

    return result;
}

private uint enc(DateTime now)
{
    long year = (now.Year-2006) * level(5);
    long month = (now.Month - 1) * level(4);
    long day = (now.Day - 1) * level(3);
    long hour = now.Hour * level(2);
    long minute = now.Minute * level(1);
    long second = now.Second;
    uint result = (uint)(year + month + day + hour + minute + second);

    return result;
}

```