

Hachioji.pm #46

Jan. 31, 2015

About me

Hideaki Ohno

**A.K.A. `hide_o_55` / `hideo55`
Perl / C++ / Node.js / Go**

宣伝

WEB+DB Press vol.85 「Perl Hackers Hub」 に
「Perlで自然言語処理入門」を寄稿しています。

Go書いてたら、
いつのまにかasm書いてた話

Go

- Googleが開発した言語
- 構文はC言語に近い感じ
- goroutineによる並行/並列処理

Go  PopCount

PopCount

- Population Count
- ビット列内の1の数をカウントする
- 省メモリなデータ構造の構築、検索で多用する

Go 🚀 PopCount

```
func popcnt64Go(x uint64) uint64 {  
    x = (x & 0x5555555555555555) + ((x & 0xAAAAAAAAAAAAAAAA) >> 1)  
    x = (x & 0x3333333333333333) + ((x & 0xCCCCCCCCCCCCCCCC) >> 2)  
    x = (x & 0x0F0F0F0F0F0F0F0F) + ((x & 0xF0F0F0F0F0F0F0F0) >> 4)  
    x *= 0x0101010101010101  
    return ((x >> 56) & 0xFF)  
}
```


これでもいいんだけど、メインターゲットの**x86_64**環境向けに最適化したい(←**C++er**の悪癖)

アセンブリ書いた

```
// func popcnt64ASM(x uint64) uint64
TEXT · popcnt64ASM(SB),4,$0-8
XORQ AX, AX
MOVQ x+0(FP), SI
BYTE $0xf3; BYTE $0x48; BYTE $0x0f; BYTE $0xb8; BYTE $0xc6; //POPCNT (SI), AX
MOVQ AX, ret+8(FP)
RET
```

Goのアセンブリサポート

- Plan9のアセンブリが使える
- `go tool 6a`
 - Plan 9 assembler
- 通常は暗黙的に呼び出される

Goでアセンブリ

.goで関数を宣言

```
// +build amd64
package popcount
func havePOPCNT() bool
func popcnt64ASM(x uint64) uint64
var asm = havePOPCNT()
func popcnt64(x uint64) uint64 {
    if asm {
        return popcnt64ASM(x)
    }
    return popcnt64Go(x)
}
```

Goでアセンブリ

- .sファイルにアセンブリを書く
- POPCNTのニーモニックはサポートされていないのでBYTEで機械語のバイト列を書く

```
// func popcnt64ASM(x uint64) uint64
TEXT · popcnt64ASM(SB),4,$0-8
XORQ AX, AX
MOVQ x+0(FP), SI
BYTE $0xf3; BYTE $0x48; BYTE $0x0f; BYTE $0xb8; BYTE $0xc6;//POPCNT (SI), AX
MOVQ AX, ret+8(FP)
RET
```

```
// func havePOPCNT() bool
TEXT · havePOPCNT(SB),4,$0
XORQ AX, AX
INCL AX
CPUID
SHRQ $23, CX
ANDQ $1, CX
MOVB CX, ret+0(FP)
RET
```

GOでアセンブリ

x86_64以外の環境向け

```
// +build !amd64
```

```
package popcount
```

```
func popcnt64(x uint64) uint64 {  
    return popcnt64Go(x)  
}
```

go-popcount

<https://github.com/hideo55/go-popcount>