

Blaze Builder のヒミツ

2012.4.22

山本和彦

リストの連結

- 右結合だと $O(N)$
"boo" ++ ("foo" ++ ("woo" + "qoo"))
- 左結合だと $O(N^2)$
(("boo" ++ "foo") ++ "woo") + "qoo"

(++) は右結合だけど...

- 簡単に左結合になる

```
foo :: [String] -> String
foo [] = ""
foo (xs:xss) = foo xss ++ xs
```

差分リスト

- 差分リスト呼ばれるが、不適切な名前
- Builder の定義

```
newtype Builder = Builder {  
    runBuilder :: String -> String  
}
```

```
instance Monoid Builder where  
    mempty = Builder id  
    Builder b1 `mappend` Builder b2 =  
        Builder $ b1 . b2
```

```
toBuilder :: String -> Builder  
toBuilder xs = Builder (xs ++)
```

```
toString :: Builder -> String  
toString b = runBuilder b ""
```

```
(<>) :: Monoid a => a -> a -> a -- GHC 7.4 から入る  
(<>) = mappend
```

差分リストを使う

- Builder を使う

```
toString $ toBuilder "boo"  
        <> toBuilder "foo"  
        <> toBuilder "woo"
```

- <> は $O(1)$

- 関数結合だから

- toString は $O(N)$

- $(f . g) . h$ でも h, g, f の順番に作用する！

```
((("boo" ++) . ("foo" ++)) . ("woo" ++)) ""
```

Blaze Builder

- ByteString 用の Builder
 - fromByteString で Builder を作る
 - <> で連結
- toByteStringIO
 - (ByteString -> IO ()) -> Builder -> IO ()
 - 固定長のバッファへコピーしながら入出力を実行する
- 左からコピーしてますよ！
- でも、実装は関数合成ですよ

```
instance Monoid Builder where
  mempty = Builder id
  Builder b1 `mappend` Builder b2 =
    Builder $ b1 . b2
```
- 関数合成でどうやって実現するんですか？

答えは継続でした

- 話を簡単にするためのコード

```
newtype Builder = Builder {
    runBuilder :: IO () -> IO ()
}

instance Monoid Builder where
    mempty = Builder $ \_ -> return ()
    Builder a `mappend` Builder b =
        Builder $ a . b

f,g,h :: Builder
f = Builder $ \b -> putStrLn "f" >> b
g = Builder $ \b -> putStrLn "g" >> b
h = Builder $ \b -> putStrLn "h" >> b
```

走らせてみる

- どう関数結合しても f, g, h の順番に作用する

```
> runBuilder ((f <> g) <> h) $ return ()
```

```
f
```

```
g
```

```
h
```

```
> runBuilder (f <> (g <> h)) $ return ()
```

```
f
```

```
g
```

```
h
```