

090228(safe mode)

Scheme で bluetooth スタックを書く

~ mosh で暮らそう ~

電気通信大学 MMA 奥村ゆうき
(id:mjt)

デモ

デモします

何これ

Wii リモコン震度計

- 赤青緑の 3 本が Wii リモコンの XYZ 軸に対応
- (描画処理が重いので描画はかなり間引きました)

動作環境

mosh + cairo + SDL + libusb。

bluetooth のプロトコル処理は全て Scheme。

デモの内側

skymosh = “Visual mosh” を作った

- cairo/SDL 対応
- libusb 対応
- Win32(Mingw) 対応

そのうちマージします

define-packet-type って構文を考えたよ

あと2分で説明します

define-packet-type の必要性

ネットワークプロトコルは C 言語の struct で実装されている

struct に似たものとして R6RS には define-record-type が有る

でも record は Scheme の世界から外に持ち出せない

C の struct のように物理構造が定義されたレコード定義が必要

define-packet-type とは

- **物理構造**を持つ (= 1 つの bytevector になる)
- C の struct に相当
- エンディアン指定可能
- 可変長フィールドもサポート

従来の LL は意外と弱かった

いままで

pack/unpack

(unpack "CC" <packet>)

=> ' (54 20)

printf のように文字列でフォーマットを指定

プロトコルの処理は OS や外部ライブラリに頼るのが普通

これから

プロトコル記述のための言語機能

```
(define-packet-type two-char  
  (seq (x (byte))  
        (y (byte))))
```

```
(two-char-receive) => (54 20)
```

テンプレートエンジンのようにプログラム構造で指定

パケットさえ投げられればもう OS は要らない

define-packet-type は

R6RS define-record-type に加えて、

- フィールドの型
- enum
- bytevector に/から変換するクロージャ
- bytevector に変換した上でパケットとして送信するクロージャ
- チェックサム、“長さ”フィールドの自動生成

等も提供します。

(エンドユーザから見た) 使いかた

1. 適切なライブラリを `import`。
2. `make-hoge` にパラメタを渡してクロージャを得る。
3. 呼ぶ。

REPL 上で `(ping)` とやれば `ping` が打てる！

```
(import (skymosh proto ip))  
(define (ping)  
  (make-ping-send (ip-addr "localhost")))
```

```
(ping) => #t/#f
```

例: さっきのデモ

1. Bluetooth 用の Packet 定義を 8 つ (library bt)
2. Wii リモコン用の Packet 定義 1 つとヘルパ関数 (library wiiremote)
3. mosh の libusb サポート
4. skymosh のグラフ機能

結論と今後 (1/2)

課題

- 現在は R6RS 形式の library を生成 (静的に変換)
- `bytevector-xx-set!` とか `bytevector-copy!` のカタマリ。。

展望

- 仕様はまだまだ未確定
- 最適化

結論と今後 (2/2) : 未来展望

あなたにもバイナリがしゃべれる

- HTML を書くくらい手軽にバイナリを入出力できる
- 遠くの Web API より近くのデバイス

mosh で暮らせる？

- mosh で USB デバイスを制御できる (許容可能な手間で)
- TCP/IP も実装できる
- (たぶん)JPEG や PNG も実装できる

OS やカーネルの役割って？

つづく？