

Presentation on your terminal

@syuu1228

in OSC京都2014



んん？

- ・ Xが無いのにmikutterが走っている
- ・ 単にフレームバッファに背景画像を貼っているだけではない、**アイコンと画像が表示出来ている**

注目



What...?

- ・ 主犯 展示者の@tsutsuiさんに仕組みを伺ってみた
- ・ mikutterのデーモンモード（GUIなし）に terminalへTLを表示するプラグインを追加、そこに**Sixelで画像を表示している**との事
- ・ 詳しくは <http://www.ceres.dti.ne.jp/tsutsui/osc2013kyoto/img00.html>

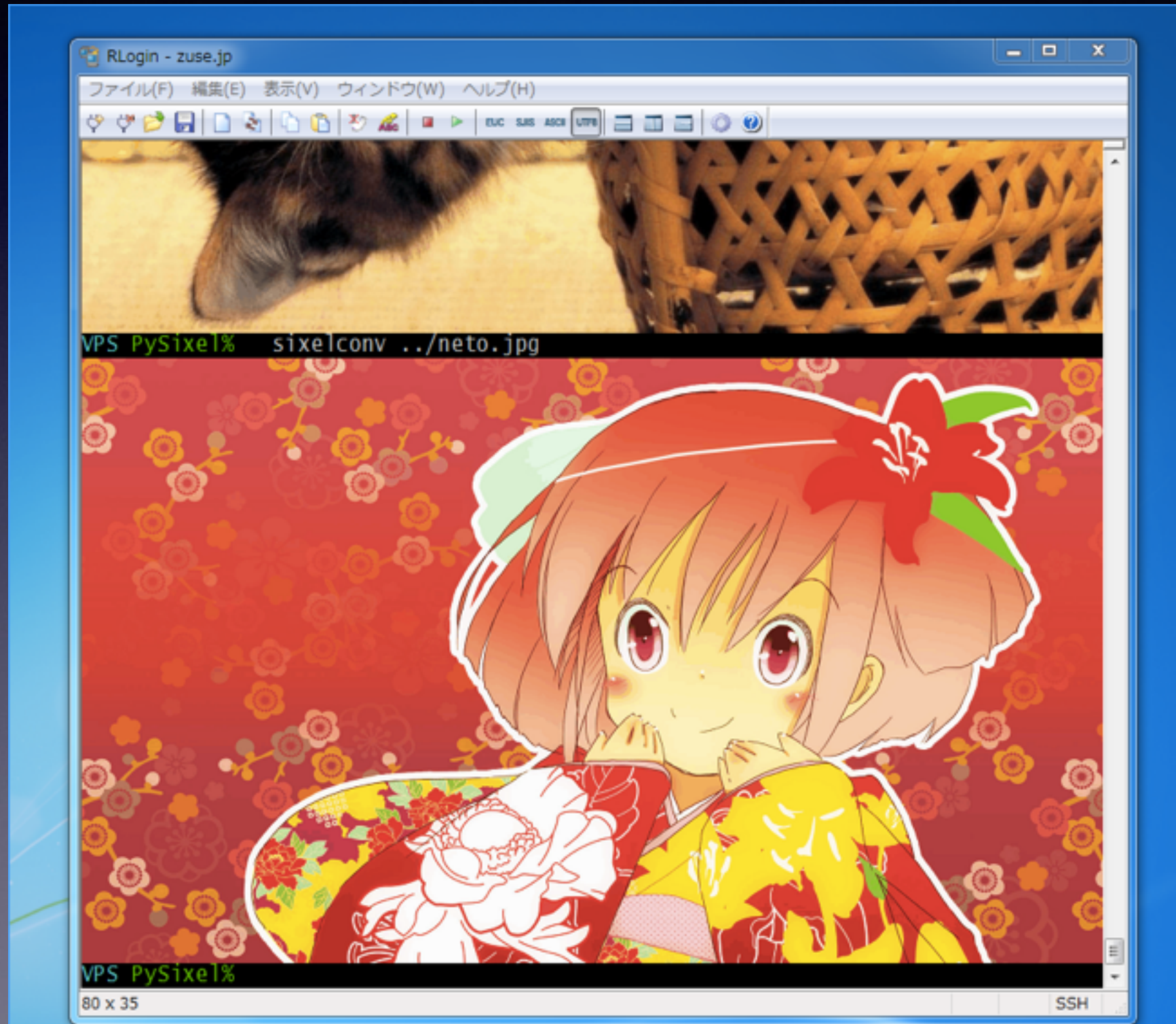
terminalに画像表示…？

- ・ 可能です
- ・ その為のエスケープシーケンスが用意されています→Sixel（ビットマップ画像用）
- ・ DEC VT2xx-VT3xx（物理的な端末）の機能
 - ・ もちろんTERM=vt100では出来ない
 - ・ terminal emulatorが対応しているとは限らない
- ・ **最近terminal emulatorでSixel対応した例が増えて来ている！**

Sixel対応terminal emulator

- mlterm (Windows, X, Framebuffer)
 - luna68kではfb版で動いていた
- RLogin (Windows)
- XTerm (X)
- tansinn (Firefox)

Windowsでも



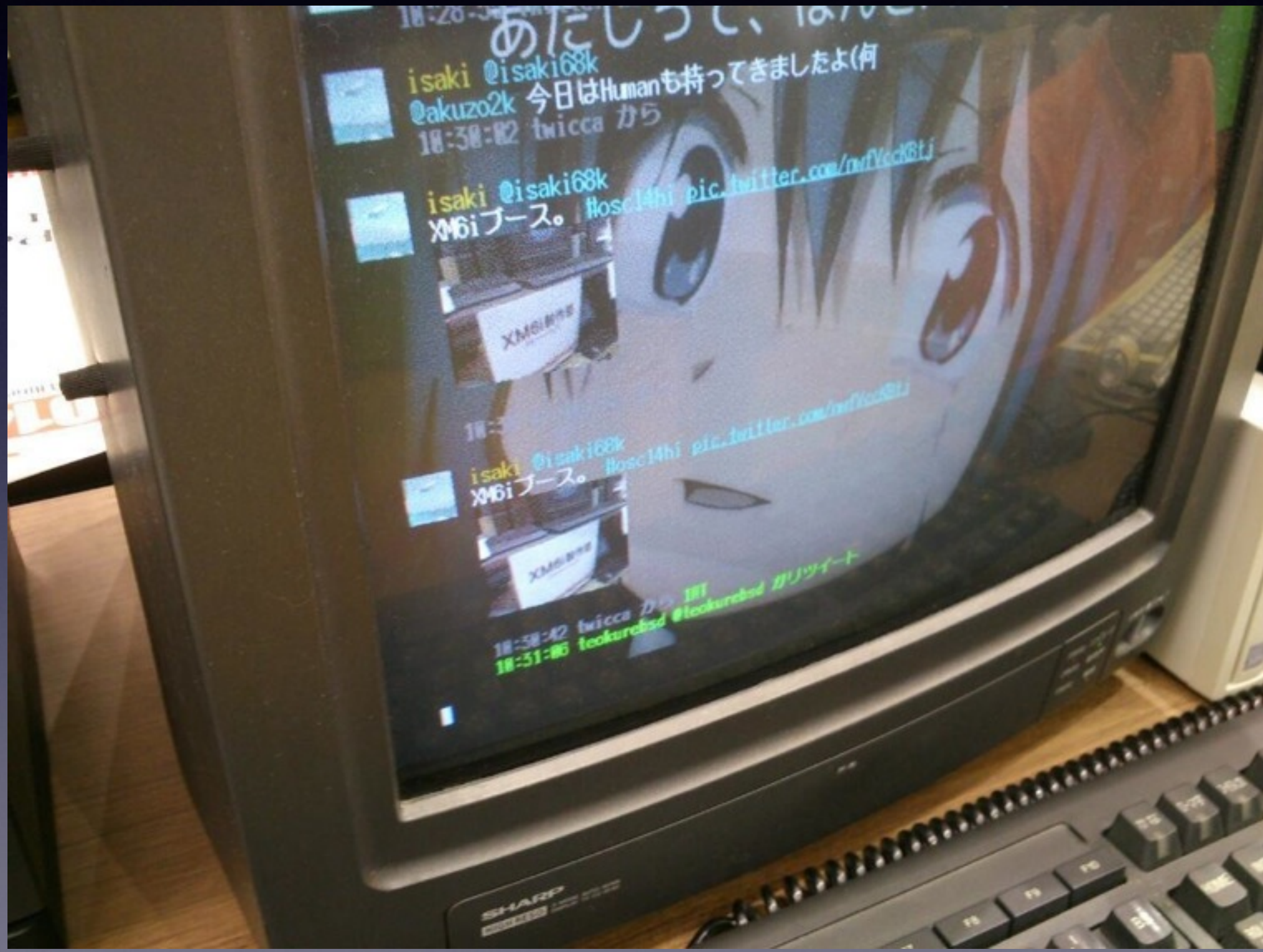
Macでも



luna68kでも



x68kでも



Sixelエスケープシーケンスの 例

<ESC>Pq

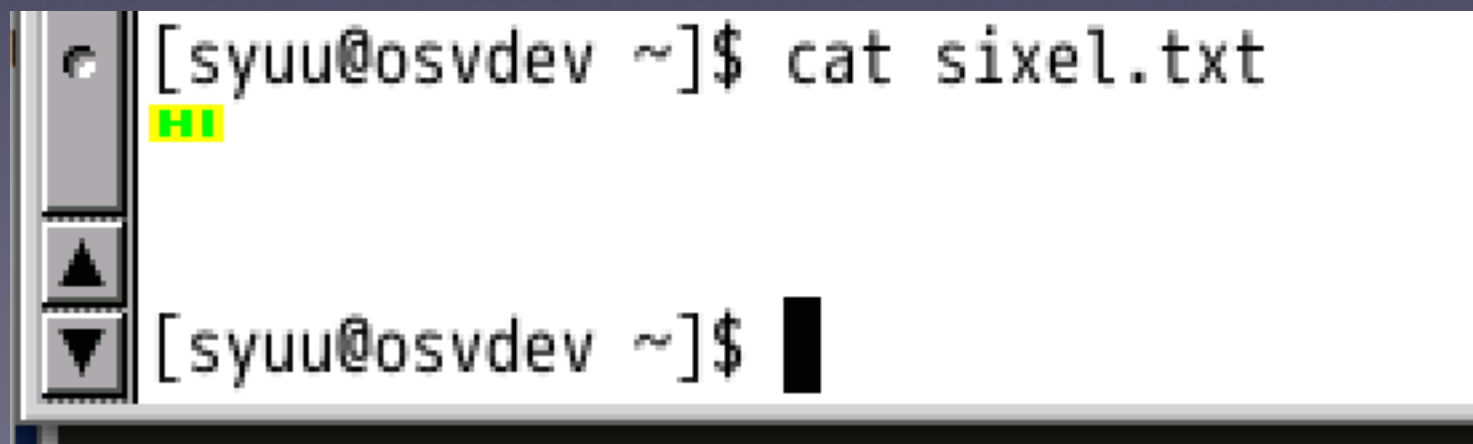
#0;2;0;0;0#1;2;100;100;0#2;2;0;100;0

#1~~@@vv@@~~@@~~\$

#2??}}GG}}??}}??-

#1!14@

<ESC>\



```
[syuu@osvdev ~]$ cat sixel.txt
HI
[syuu@osvdev ~]$
```

The screenshot shows a terminal window with a title bar. The prompt is [syuu@osvdev ~]\$ and the command cat sixel.txt has been entered. The output of the command is displayed on the next line: HI. The 'H' is in green and the 'I' is in yellow. Below the output, the prompt [syuu@osvdev ~]\$ is shown again with a black cursor block.

僕も何か作りたい！

- ・ でも、ただLinuxやMacの上でSixelで画像表示しても簡単すぎて何もおもしろくない
- ・ **じゃあOS無しでやろうか**

sixelでプレゼン？



スライドデータ
+
シリアル出力
+
ページ操作

シリアル接続



mlterm

作ってみた

- sixelslide

<https://github.com/syuu1228/sixelslide>

ターゲットデバイス

- ・ 取り敢えずqemu-system-i386向けに書いてみた
- ・ x86(32bit) multiboot specification ELF
- ・ 叩くデバイスはCOM1のみ
 - ・ COM1へのwrite：Sixelでスライド出力
 - ・ COM1からのread：ページ操作
 - ・ HDD, NIC, クロックなど他デバイスは使わない
- ・ メモリ管理はしない、全て静的に確保

スライドをELFバイナリに 埋め込む

```
convert slide.pdf slide.png
```

```
for i in slide-*.png; do  
img2sixel $i ${i/.png/.txt}; done
```

```
for i in slide-*.txt; do  
ld -r -b binary -o ${i/.txt/.o} $i  
done
```

```
ld -o kernel kernel.o slide-*.o
```

プログラムからの参照

```
char *p = (char *)  
    &_binary_slide_0_txt_start;
```

```
char *end = (char *)  
    &_binary_slide_0_txt_end;
```

```
while (p < end)  
    com_putc(*p++);
```


参照を配列にしたい

- ・ このままだと出来ない
- ・ ELFパーサを実装すれば
_binary_slide_**n**_txt_startをイテレートできる
- ・ 今はめんどくさい
- ・ やっつけRubyスクリプトで配列生成

やっつけ感

```
num_slides = Dir.glob("slide-*.png").count
puts "char *slide_start[] = {"
(0...num_slides).each do |i|
    puts "(char *)&_binary_slide_#{i}"
    _txt_start, "
end

puts "};"
```


スライド操作

- ・ 良く分からないけどviっぽく
 - ・ k = 上 (1ページ戻る)
 - ・ j = 下 (1ページ進む)
 - ・ h = 左 (1ページ戻る)
 - ・ l = 右 (1ページ進む)
- ・ q = 終了
- ・ 1 = 1ページ目へ戻る

コード量

```
$ cat main.c io.c io.h power.c power.h  
boot.s com.c com.h sixelslide.ld|wc -l
```

195

```
$ cat genslide*.rb|wc -l
```

26

```
$ cat Makefile |wc -l
```

46

DEMO

- ・ このプレゼンがDEMOです。

TODO

- ・ 動きそうなスペックのボードに移植する → Intel Galileo?
 - ・ スライドデータがデカくてスペック低い板じゃ無理そう
圧縮？ 減色？
- ・ 単一のオブジェクトに全スライドを入れたい
- ・ データはELFバイナリビルド後に埋め込めるようにしたい
or ファームウェア経由でファイルシステムからロード
- ・ 物理ボタンでスライド操作したい