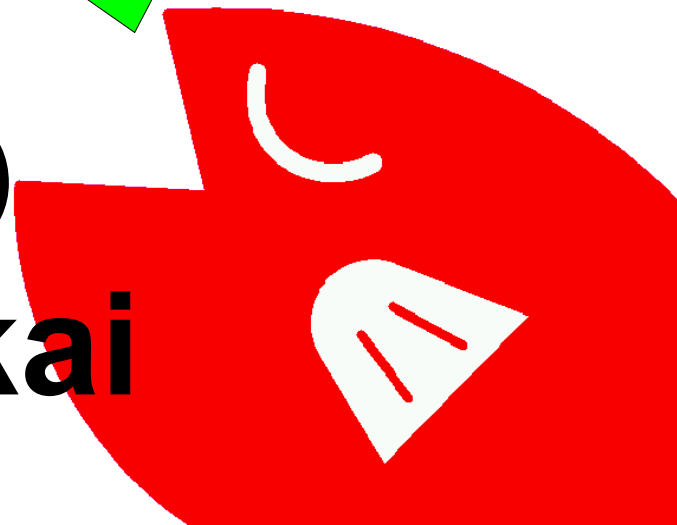


自作組込みOSを
エミュレータで
動かしてみた

坂井弘亮

(KOZOSプロジェクト)

Twitter ID:kozossakai



略歴

1997年頃 プログラミングに傾倒

X Window System プログラミング
オブジェクト指向プログラミング
OSに興味を持つ

1999年 就職、組込みシステムに興味を持つ

2007年 自作組込みOS「KOZOS」の開発を開始

2009年 OSC出展、KOZOSを実ボードに移植

2010年 OSCなどのイベントに多数参加

組込みOS自作の書籍を出版

セキュリティ&プログラミングキャンプ講師

組込み勉強会を主催

OSを自作しています

KOZOSプロジェクトとは…

マイコンボード上で動作する
独自組込みOS「KOZOS」を
フルスクラッチで
作っているプロジェクトです

↓ここで動くOSです！

秋月電子:H8/3069Fマイコンボード



KOZOSプロジェクトの構成

代表...坂井

企画担当...坂井

設計担当...坂井

コーディング担当...坂井

テスト担当...坂井

営業担当...坂井

広報担当...坂井

会計担当...坂井

事務担当...坂井

カメラマン...坂井

運転手...坂井

ひとりプロジェクトです

KOZOSの特徴は

ブートローダ...自作

シンプルなOSカーネル ...自作

デバイスドライバ ...自作

libcっぽいライブラリ ...自作

シェルっぽいもの ...自作

なんちゃってTCP/IP ...自作

超単純なwebサーバ ...自作

完全フルスクラッチ!

個人の学習向け、
いじって遊ぶための
OSです

で、作っている
わけなんですが

前から思っていたこと

自作OSに
TCP/IPを実装して、
Webサーバとか
やらせてみたい!

フルスクラッチの
Webサーバを
動かしてみたい!

作り始めてみたが...
(もちろんフルスクラッチで)

**TCPはステートフルなので、
デバッグがたいへん！**

そもそも今までどうやって
デバッグしていたのか？

自作組込みOSでは、
デバuggなんて
当然使えない！
(自分で対応すれば別)

**printf()デバッグは
できないの？**

そんなの**超**贅沢!

自作OSで
数値を表示
したかったら
どうするか？

まずは
「数値表示ライブラリ」を
自作する必要がある

こんな感じですかね

```
int putxval(unsigned long value, int column)
{
    char buf[9];
    char *p;

    p = buf + sizeof(buf) - 1;
    *(p--) = '\0';

    if (!value && !column)
        column++;

    while (value || column) {
        *(p--) = "0123456789abcdef"[value & 0xf];
        value >>= 4;
        if (column) column--;
    }

    puts(p + 1);

    return 0;
}
```

デバッグ環境も
自分でなんとかする
必要があります!

なんとかするのが
めんどろくさい
場合には
どうなるの？

デバッグ環境
無しでがんばる
ことになります

というわけで
組込みのデバッグは

基本、「勘」です

「勘デバ」のいいところ

- ・特別な機材やアプリが不要
- ・準備作業が不要
- ・どこででもデバッグ可能
- ・一瞬で解決できることあり
(運が良ければ)
- ・運が良ければ、**最高の効率**

しかし、まあそうも
言っていない

TCPの開発には、
デバッガが使いたい!

他にも...

最近暇が無い！

出先や移動中に開発したい！

電車でハックしたい！

飛行機でハックしたい！

エミュレータが使えれば
一石二鳥だ！

ということで、
まずは**エミュレータ**で
動作するようにしよう

エミュレータには 何を使うか？

- ・ HEW...CPUメーカー純正の
統合開発環境
- ・ GDB...超有名なフリーの
デバッガ
(H8のエミュレーション機能有り)

HEWならば...

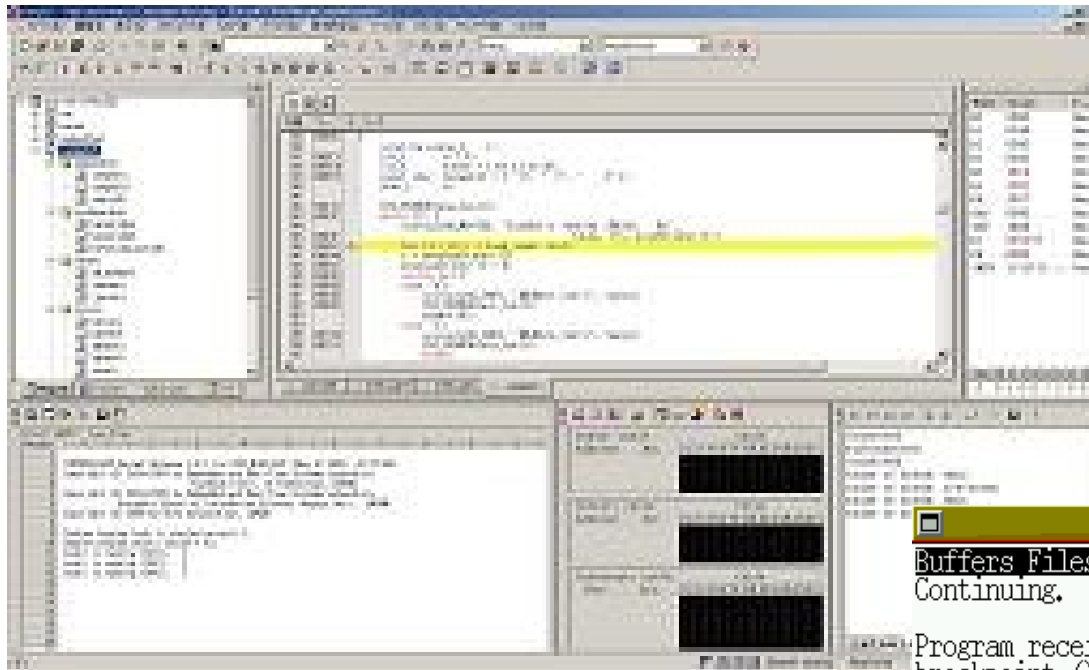
- ・CPUの品種ごとに、細かい動作
- ・メーカ純正の安心感
- ・きらびやかな統合開発環境
- ・もちろんGUI
- ・日本語マニュアルあり

GDBだと...

- ・自分でいろいろ対応が必要そう
- ・CPUのコア部分しかシミュレートしてくれない
(I/Oのエミュレートは自分でなんとかする)
- ・割り込み処理が未実装(自分でなんとかする)
- ・バグがあるかも(自分でなんとかする)
- ・マニュアルなんか当然無い(なんとかする)

まあ、
GDBかな、と

HEW



GDB+emacs

```
kterm
Buffers Files Tools Edit Search Mule Gud Complete In/Out Signals Help
Continuing.

Program received signal SIGTRAP, Trace/breakpoint trap.
breakpoint () at i386-stub.c:973
(gdb) break dummy_func
Breakpoint 1 at 0x804aa7e: file telnetd.c, line 21.
(gdb) continue
Continuing.

Breakpoint 1, dummy_func (n=10) at telnetd.c:21
(gdb) █

-かな-EEE:~*F1 *gud-koz* (Debugger:run Encoded-kbd)--L23--C6--Bot---
int telnetd_id;
int telnetd_dummy;

static int dummy_func(int n)
{
    int i, sum;
=>sum = 0;
    for (i = 1; i < n; i++) {
        sum += i;
    }
    return sum;
}
-かな-EEE:---F1 telnetd.c (C Encoded-kbd)--L21--C0-- 8%-----
```

で、まあGDBを
選んだわけですよ

では、実際に
「自作webサーバ
with自作TCP/IP
on自作OS
from自作ブートローダー」
を動かしてみましよう

おしまい