

クロスコンパイラを作ってみたものの、次に何をしたいか分からなくなったときのメモ書き。
毎回、なんだか思考錯誤してコンパイラの構築を疑いながら開発しているもので ...
binutils、gcc、newlib から、SH のクロスコンパイラを作る方法は結構いろいろなサイトで公開されていますが、実際にコンパイルする話ってあまりでてこないような ... 検索の仕方が悪いのかな？

ちなみに私のつかっているそれぞれのバージョンは

- binutils-2.13
- gcc-3.23
- newlib-1.11.0

です。

なお、下記についてアドバイスや訂正などありましたら、お手数ですが教えてください。

とりあえずプログラムをコンパイルしたい

用意するもの

- スタートアップルーチン
- リンカスクリプト

スタートアップルーチン

C 言語のプログラムは main 関数から呼ばれる ... のはプログラムに対しホストとなる OS のようなシステムが面倒を見てくれる場合である。スタートアップルーチンは、OS がないような組み込みプログラムで、まず最初に実行し、C 言語でいう main 関数を呼び出すためのプログラム。もちろん、アセンブラで記述しなければいけない ...
たとえば、スタートアップルーチンを書かずに、

```
/* hoge.c */
main(){
    printf("fefe?n");
}
```

のようなプログラムを

```
% sh-hms-gcc hoge.c
```

とコンパイルすると次のようなエラーが発生してしまいます

```
/usr/local/sh/lib/gcc-lib/sh-hms/3.2.3/libgcc.a(__main.o): In function `__do_global_dtors':
libgcc2.c:1901: undefined reference to `__EH_FRAME_BEGIN__'
/usr/local/sh/lib/gcc-lib/sh-hms/3.2.3/libgcc.a(__main.o): In function `__do_global_ctors':
libgcc2.c:1921: undefined reference to `__EH_FRAME_BEGIN__'
collect2: ld returned 1 exit status
```

なので、簡単でいいので main 関数を呼ぶまでのスタートアップルーチンは必要。

例えば、こんな感じ。

```
        .text
        .align 2
        .global __main

start_up:    mov.l   L0,r0
            jmp     @r0
            nop

__main:     rts
            nop

        .align 2
        .global _main
L0:        .long   _main

        .end
```

リンカスクリプト

リンカスクリプトは実際に作成したオブジェクトコードを実際にメモリに配置するアドレスなどを指定するもの。きちんと、開発環境が構築できていた場合には、`/usr/local/sh/sh-hms/lib/ldscripts/sh.x` な感じで生成されている。これを参考にして書き直せばいい。

コンパイルの仕方

例えば

```
% sh-hms-gcc -Wl,-T,/usr/local/sh/sh-hms/lib/ldscripts/sh.x,-Ttext,0x900 ?
-oformat,srec -lm -o test.mot hoge.c crt0.s
```

出力を変えたい場合は、例えば、`-oformat,ihex` とか
基本的には、`Makefile` を書いておいた方が便利