環境構築・使用方法

事前準備

- MSYS2 （Windows のみ）
- Mercurial を利用中のパッケージシステムからインストール

1. 環境構築

```
$ hg clone https://bitbucket.org/7shi/i8086tools
$ cd i8086tools
$ make
$ sudo make install
```

- MSYS では sudo なしで make install
- BSD 系では gmake を使用

以下のファイルをダウンロードし,/usr/local/minix2/usr に展開します。
http://www.minix3.org/iso.previous/Intel-2.0.4/i86/USR.TAZ
http://www.minix3.org/iso.previous/Intel-2.0.4/src/SYS.TAZ

```
$ mkdir -p /usr/local/minix2/usr
$ sudo tar xvzf USR.TAZ -C /usr/local/minix2/usr
$ sudo tar xvzf SYS.TAZ -C /usr/local/minix2/usr
```

これで環境構築は完了です.

2. 使用方法

m2cc:

c 言語で書かれたソースコードをコンパイルし,minix8086 のバイナリを出力します.

使用例
c 言語
,$ m2cc hello.c
アセンブリ言語
$ m2cc -.o hello.s

ソースコードをコンパイルし a.out という実行ファイルが作成されます.
なお,アセンブラからバイナリを作成する際は -.o オプションをつけてください.
-.o オプションは crt や libc をリンクしないオプションです.

7run:

minix8086 用のバイナリを実行するインタプリタです.

使用例
$ m2cc hello.c
$ 7run a.out
hello

8086 用のバイナリを実行するすることができます.

@7run オプション

7run [-d/-m] cmd [args...]
-d: disassemble mode
-m: memory dump & run

-d:　disassemble mode

バイナリがどのような命令か解釈し,アセンブラに変換します.
なお,この操作をディスアセンブルと呼びます.

使用例
$ 7run -d a.out
0000: 58　　pop ax
0001: 89e2　　mov dx, sp
0003: 52　　push dx
（略）

引数に与えられた実行ファイルを minix8086 のバイナリとして,解釈し,その命令をアセンブラに変換します.

[-m]: memory dump　& run

命令やレジスタなどのログを表示しながら, 実行することができます.
使用例
$7run -m a.out
AX　BX　CX　DX　SP　BP　SI　DI FLAGS IP

0000 0000 0000 0000 fff6 0000 0000 0000 ---- 0000:58
pop ax
…略

レジスタなどの値を一命令ごとに表示しながら, バイナリを実行しています.

hexdump:

今回のツールに付属するものではありませんが, ファイルの中身を16 進数で出力するコマンドです.
バイナリエディタや他のコマンド(od など)を用いても構いません.

使用例
$hexdump a.out
0000000 01 03 20 04 20 00 00 00 10 00 00 00 26 00 00 00
0000010 00 00 00 00 00 00 00 00 00 00 01 00 70 00 00 00
0000020 bb 00 00 cd 20 bb 10 00 cd 20 00 00 00 00 00 00
0000030 01 00 04 00 01 00 06 00 00 00 20 00 00 00 00 00
0000040 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00
0000050 68 65 6c 6c 6f 0a 6d 65 73 73 61 67 65 00 00 00
…(略)

内部のバイナリを簡単に確認するために用います.
-C オプションをつけると内部のバイナリを文字に変換したものも表示してくれます.