# 基礎実験1 UNIX・アセンブラ実習 第2回

2007年4月11日(水)

### 実習内容

68000 実習用の各種のソフトを使ってみます.(1) アセンブラソースプログラムファイルの作成,
 (2) アセンブラソフト(m68k-as)による言語処理,(3) エミュレータ(m68k-emu)による実行,(4)
 バグの発見とデバッグという一連の流れを体験してみます.

#### 1. ファイル(add.s)の準備

今日の実習では、アセンブラの方法と、エミュレータの使い方について学びます。 まず、アセンブラソースプログラムファイルを用意します。次の命令を Terminal 上で実行しましょう。

\$ /u/matsuki/jishu2-setup 自分のユーザ名 <Enter キー>

ファイル add.s の中身を emacs で確認しましょう . /\*と\*/で囲まれた文はコメントを表し, 実行時には 影響を与えませんので, 省略しても構いません.

```
/*
      sample program add.s */
      0 \times 0000
.org
.dc.1 0x5000
.dc.l start
.org
      0 \times 0400
start:
                       /* a 番地に 3 を格納 */
      move.w #3,a
                        /* b番地に4を格納 */
      move.w #4,b
                        /* a 番地の値をレジスタ d0 に格納 */
      move.wa, %d0
      add.w b, %d0
                        /* b番地の値をレジスタ d0 に加える */
                        /* レジスタ d0 の値を c 番地に格納 */
      move.w %d0,c
                        /* BSVC の break 相当 */
      .dc.w 0x4848
                        /* 終了 */
      stop #0
a:
      .ds.w 1
      .ds.w 1
b:
c:
      .ds.w 1
```



図1.今日の実習の流れ

### 2. アセンブラ

アセンブラソースプログラムファイルが準備できたので,次に,アセンブラソフト m68k-as を起動し ます.Terminal 上で次のコマンドを実行しましょう。

\$ m68k-as add.s < Enter +->

アセンブラは、68000 という CPU やそのエミュレータで実行可能なファイルに変換する言語処理プロ グラムです。そして、上記の m68k-as という命令は、引数で指定されたファイルをアセンブルするコ マンドです。

うまく、アセンブルが出来ると、ディレクトリ中に次のようなファイル add.LIS, add.abs, add.map が生成されていることが分かります。

<Enter +-> % **ls** add.LIS add.abs add.map

add.s

add.LIS : エミュレータで参照できるリスティングファイル add.abs :エミュレータや実機で実行可能なファイル add.map:メモリーマップのファイル add.s :ソースファイル

課題1.次のファイルの中身について報告せよ.報告は簡単でよいが、「実際に中身を見た」ことが分 かるような報告であること

(1) add.LIS (2) add.abs (3) add.map

## 3. エミュレータ (BSVC)

教育用計算機の CPU は 68000 ではありません。そのため、この計算機の CPU では実行ファイル (add.abs)を直接動作させることはできません。しかしながら、エミュレータと呼ばれるプログラム (BSVC)を用いることによって、68000CPU 上で動作を「再現」することが可能です。このエミュレ ータは、m68k-emu というコマンドによって起動します。

\$ m68k-emu &	<pre>Enter +-&gt;</pre>			
💙 BSVC: Version 2.1				///// 👝 🗖 🗙
File Edit Window				Help
Registers		Trace		
D0 = 00000000 D1 = 00000000 D2 = 00000000 D3 = 00000000 D4 = 00000000 D5 = 00000000 D6 = 00000000 A0 = 00000000 A1 = 00000000 A2 = 00000000 A3 = 00000000 A4 = 00000000 A5 = 00000000 A6 = 00000000 A7 = 00000000 PC = 00000000				
SR = 2700	Breakpoints	Single Step	Run	Reset

この BSVC(エミュレータ)のウィンドウの左側には D0~D7(データレジスタ) A0~A7、A7'(A0 ~A6 がアドレスレジスタで, A7 と A7' はスタックポインタ) PC(プログラムカウンタ) SR(ステ ータスレジスタ)の値が表示されます。

このウィンドウ上部にある File メニューから、「Load Program」を実行し、実行したいプログラムを 選びます。ここでは、add.absを選択してください。

Directory:	/u/matsuki/add	-	<u> </u>
🗈 add.abs			
🖹 add.LIS			
🖹 add.map			
i add.s			
1			
4			(,
J File <u>n</u> ame: add	l.abs		). Open
J File <u>n</u> ame:  add	I.abs		() Open

さらに、「Window」メニュー中の、「Memory Viewer」および「Program Listing」を実行すると、ア センブラプログラムの実行の様子を詳しく観察することが出来ます。それぞれの Memory Viewer, Program Listingのウィンドウを開いて,見やすく配置しておきましょう.準備が出来たら,いよいよ プログラムを実行してみましょう.実行は右下の「Run」ボタンを押します.

このプログラムは a 番地に 3 を、b 番地に 4 を格納し、それぞれのアドレスに格納された値をレジスタ d0 を使って足し合わせ、最終的に c 番地にその結果を格納する (「アドレス」や「番地」という言葉に ついては,授業の方で説明があったので思い出して下さい)。今回の実習では,「BSVC: Programming Listing add.LIS」の次の部分に注目してください.

000424 0000	15 a: .ds.w 1
000426 0000	16 b: .ds.w 1
000428 0000	17 c: .ds.w 1

この3行には,データa、b、cを確保する擬似命令「a:.ds.w.1」,「b:.ds.w.1」,「c:.ds.w1」が並ん でいるのですが,それらの行の一番左側に表示されいるアドレス「000424」、「000426」、「000428」が データa、b、cのメモリアドレスを表しています。

Memory Viewer の値で,メモリの中身を観察できるので,「000424」、「000426」、「000428」番地の中 身を観察してみましょう. Memory Viewer のメニュー「View」から「From Address...」を選択し, 任意のアドレスを入力します.これにより,そのアドレスにあるメモリ内容を確認することが出来ます. ここでは,キリの良い000420 からのメモリを見てみましょう(エディットボックスに 420 と入力する). Memory Viewer を使うと 000424 番地に 00,00425 番地に 03 が格納されています (ds.w という命令 が1ワード(2バイト)分のデータ領域を確保する命令であるため2バイト使用する).同様に,0000426 番地からの2バイトに 0004,0000428 番地からの2バイトに計算結果の 0007 が格納されているのを 確認できるはずです(注: a, b, c のメモリアドレスは 000424、000426、000428 に固定されているわ けではありません.これらのアドレス値はこの例に限ったもので、異なる場合もあります)

000424 番地:	00
000425 番地:	03
000426 番地:	00
000427 番地:	04
000428 番地:	00
000429 番地:	07

また、レジスタ d0 にも 0007 が格納されていることを確認しましょう (メインウィンドウ左側)。

SVC:	Mem	ory	Vie	ver		17/2					777		777	7777	2.	. X
File E	dit	Viev	н										-	\$		5
000420	: 4e	72	00	00	00	03	00	04	00	07	00	00	00	00	00	00
000430	: 00	00	00	00	00	00	100	00	00	00	00	00	00	00	00	00
000440	: 00	00	00	00	00	00/	00	00	00	00	00	00	00	00	00	00
000450	: 00	00	00	00	~~,		<u></u>	00	00	00	00	00	00	00	00	00
000460	: 00	00	00	00	Â	<b>吉</b> 果	D	00	00	00	00	00	00	00	00	00
000470	: 00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000480	: 00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000490	: 00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0004a0	: 00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000450	: 00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0004c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000440	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000460	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000410	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000500	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000510	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000520	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000540	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000550	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000560	: 00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

4. ステップ実行

エミュレータでは、命令毎に実行を止めて、段階的にプログラムを実行する方法(ステップ実行)があ ります。

プログラムの再実行の前に,実行結果をクリアしておくべきです(そうしないと,前の実行結果が残っ てしまい,プログラムの振る舞いを観察するときに混乱します).まず、実行した結果が000424番地か ら 000429番地まで残っているのでそれをクリアしましょう。Memory Viewer の Edit メニューの「Fill Memory Block...」を実行しましょう。Starting Address に 000424を Length に 6を Value に 0を入 れて「OK」をクリックします。

~	BSVC:	Mem	ory	Vie	ver		11/2	1112		7777		1111		7777	777		. X
Fi	le E	dit	Viev	v								4				10	5
00	0420	: 4e	72	00	00	00	03	00	04	00	07	00	00	00	00	00	00
00	0430	: 00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	0440	: 00								last				60	00	00	00
00	0450	00				FIII	Mei	nor	ув	IUCK				10	00	00	00
ÕÕ	0470	00				10222						_	-	10	ÕÕ	00	00
00	0480	: 00	1	star	ting	Ad	dre	ss:	104	124	Q.			00	00	00	00
00	0490	: 00		_		_								00	00	00	00
00	04aU	: 00		enc	th:	6								10	00	00	00
00	04.00	: 00				10								10	00	00	00
00	04d0	00					_	_	_	_	_	_	- 1	10	00	00	00
00	04e0	00	1	/alu	e:	0								00	00	00	00
00	04f0	: 00	1										- 12	00	00	00	00
00	0500	: 00			Ok		- 1			Ca	nce	1	-	00	00	00	00
00	0510	: 00										_	1	10	00	00	00
00	0520	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	0540	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	0550	: 00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	0560	: 00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

次に、BSVC のメインウィンドウで Reset ボタンを押します。すると PC(プログラムカウンタ)が 00000400 になり Program Listingのウィンドウの水色の部分が 000400 番地に戻ります。この水色で 選択されている行が,次に実行される命令です(PC は次に実行する命令のアドレスを表す).

ー回だけ Single Step ボタンを押しましょう。すると Trace (灰色の領域)に実行した命令が現れ、PC が 00000408 になり、Program Listing の水色の部分も移動します。最初の命令は 000424 番地からの 2 バイトに 0003 を書き込むという命令です。Memory Viewer で確認しましょう。

PC が 00000420 になるまで (Trace で 00000422 BREAK がでるまで ) Single Step を押し、実行の様 子を一**行ずつ**確認しましょう。

5.エミュレータの終了 エミュレータを終了するには File メニューの Quit をクリックしてください。

# 今日の実習はここまでです。

参考 Web ページ: http://www.db.is.kyushu-u.ac.jp/kaneko/as/index.html