

FORTHによるLISPインタファクタの作成

池野 信一 (電気通信大学)

1. はじめに

LISPなどの言語処理系をマイコン上に実現し、自宅で人に気がねなく自由に使えるようにしたいとは、誰しも考えとるであろう。事実、8ビットマイコン上には数々のLISP処理系が作られている。ただ、8ビットマイコンでは、セル領域も限られるし、十分な機能を望むのは無理であろう。それほどセルを必要としない肉題、練習用、教育用などにはもちろんこれでも十分である。

16ビットプロセッサを用いれば、アドレス空間はメガバイトのオーダーになるし、十分なセル領域がとれて、実用的なシステムが期待できる。すでにそのような処理系が作られているようであるが、筆者もFORTHの上にもひとつ実現して見たので、以下簡単に紹介することにした。

2. ハードウェア

筆者はこれまで、8008, 6800, 280などのプロセッサを使ってシステムを作ってきたが⁽¹⁾、16ビットでは68000のアーキテクチャが大変気に入って、システムを作りあげた。⁽²⁾

作り方は非組織的で、まず最小限の動くものを作り、それが動くのを確認してから少しずつ機能を追加し拡張して行ったのである。昨年末以来あまり生長してないが、現状は右の通りである。RAMの32KBは現在つながないが、128KBに改造予定のものである。

RS-232Cは別のマイコンPC-8001 MK2 につながつているが、最近68000を使い512KBのメモリを付けた改良マイコンを、CP/M 68Kを動かすために入手したので、いずれこれもつなげたいと考えている。

RTTYは、無線を通して他のマイコンとデータ通信をしたいと考えてインタフェースを作ったが、現在のところ、まだメッセージの通信しかやっていない。

CPU	Mc68000
ROM	4KB
RAM	128KB
	4KB (メモ用)
	(32KB)

ミニプロセ	
デジタルカセット (MT2)	
プリンタ	
ディスプレイ	
キーボード	
RS-232C	} インタフェース
D/A	
ROMライター	
RTTY (アマ無線)	

3. FORTH

ソフトに因りては、主に入出力の制御と、簡単なデバッグ機能をもつたモニタをマシンコードで作った。

次のステップは普通アセンブラということになるであろうが、アセンブラ作りはあまりおもしろくもなまそうだし、いかに、どこかのものを移植しなおすにたどり着くかという点で、マシンコードで直接 FORTH を作ることにした。⁽⁵⁾

FORTH は

1. 核の部分は比較的簡単にできる。
2. コマンドを追加してやれば機能はいろいろと拡張できる。
3. コマンド列というプログラムの形式は、OS の機能に含まれるのに便利であるなどという特徴がある。

こうして、主に入出力の核を依り、これにワード定義機能を加えて、必要と思われる機能をコマンドとして加えていった。従って標準的な FORTH とは十分違ったものになっている。

FORTH の要素はワードである。普通ワードには定数、変数、コマンドがあるが、さらにテキストファイル、LISP 用テーブルなども含めて、すべて同じワード辞書に登録するようにした。

ワード名は5文字以内とし、32ビットに圧縮し、ワード辞書内はほぼソースのままの形で右図のように入れている。

ワード名のリストをとると、下図のようになっている。これは LISP 処理系を合入している。

各ワードは、ED コマンドで自由にエディットできる。またファイルワード

ワード辞書	
0	コマンド
1	コマンド(機械語)
2	変数、配列
3	テキストファイル
4	LISP 用テーブル

```

* TEN ! #C0FC000A #4E75FFFF ;
* PTEN CR , .. TEN "#*10#" .. ;
#12 PTEN
12*10=120
#V X
    
```

```

008540 FFFF ) コマンド区切り
008542 FFFF )
008544 8001 ) コマンド名(TEN)
008546 414E )
008548 0010 ) リンク
00854A 0001 ) 機械語の表示
00854C 00FC )
00854E 000A ) 内容
008550 4E75 )
008552 FFFF )
008554 FFFF ) 区切り
008556 FFFF )
008558 8041 ) コマンド名(PTEN)
00855A 414E )
00855C 0024 ) リンク
00855E 0000 ) コマンドの表示
008560 8000 )
008562 0002 ) CR
008564 8000 )
008566 002C )
008568 8000 )
00856A 000E )
00856C 8001 ) TEN 内容
00856E 414E )
008570 002A )
008572 3130 )
008574 003D )
008576 FFFF )
008578 8000 )
00857A 000E )
00857C FFFF )
00857E FFFF ) 区切り
008580 8000 ) 変数名(X)
008582 0018 )
008584 000C )
008586 0002 ) 変数の表示
008588 0000 ) 変数
00858A 0000 )
00858C FFFF )
00858E FFFF ) 区切り
008590 0000 )
008592 0000 ) 辞書最後の表示
    
```

※

```

. H CR ' P L E D + ^ , A L M A+ D E N A N D P S P H R N R C P C A R R A Y
B C R T L L L D D- S K I F E L S E T H E N C A S E ( ) < R O R R O L O R F S
A++ S M L L A D1- D O L O O P T R C D D+4 P R I N T N P S S S T R G D1- E X
C L L I N T B L J F I F S M L 2 S U B A S1 D1+ N O P S W P E X - B A R A B S R T F
L A 2 S F U N N B A R A S R A S R 3 A S L 3 R O L 1 D 2 L D 2 C O N S 1 A C O P D+ A T O M
E Q P O P X O V E R A P A R F P A R C C P A T & B % B Q U O T E A T O M 1 Z E R O P = 0
S U B 1 N O R M N U L L * P A D P M M P F M M M T I M E M M N T & C A --- P S T R G
L M L C M P L X P F U N K F L I S T T 4 S T K N M A 0 L A+4 S A 2+4 A 2 > G R T P
S T R T C M P C P F M F-M A 2++ R S T R G L M B+ A1 L A1 S A T O % & C O N S
R A T O S E R C F R S M B C P Y 1 C O P Y E D I T A S D F I L A D .. L I S T P S E 1 O F I L E
I F I L E M F I L E M T W B E X C G B G I N C M F S N N T R C C D R 1 R E T U N R E T N V
C A R C A R 1 C D R E X Z A D D 1 A S T G O , C A R E Q 1 A- S M I N D D E F I N L M L 2
A T O M X D L F L E C S O R T 2 J M P S E T A C L A M C A A R C A D R G B C C S B R 1 G Q
P O P P U S P U S 1 I N I T C M P L P A T O M P S E P S E 2 P A T 1 G B C T 2 T 3 Q
R P L C A R P L C D N C O N C P P P S U B R M I N U P M I N U S < 0 E V A L 1 T O R 1 E Q U A L
E Q L M E M B Q U N C M P I L E L S D E V A L C E L E F I L E T T F I L E C O N D D I F
T I M E S P L U S C O N S 2 E V L I S F U N C M A R K P R O G P U S X // E V A L 0 F L A G
I F D B S E T B I S B T S T B I T S T B I C B C L R Z 0 Q U N 1 L A B L A Z G O T O H L T
Z X X X P F I L E L G T G D 1 G D 2 P D F U N P N U M M T R A P P
    
```

は EDIT でエディットできる。しかしの場合イワードはエディット領域にコピーされてエディットされる。

フロッピーは、辞書全体をファイルとして save できるし、またひとつのワードだけをファイルとすることもできる。

ファイルは辞書のトップからロードできるが、右下にあるイワの後にジャンプができる。

フロッピーは1台だけであるが、ディスタルカセットが2台を挿入している。

#DIR	RTTY	GDISP	RS	LISP	DASM	FORTH1	FORTH	LISP1
.	003000	0035F0	020000	02067C	003000	003100	020000	02472C
.	020000	021A24	020000	020A84	020000	0209A0	020000	024768

CONSI	0 D DO , CASE(1 ^ 0 EX) LD CONSI D LOOP LD
ACOP	264024DB 24DB24DB 201C4E75
D+	52804E75
ATOM	4A006A04 42804E75 70024E75
EQ	909C6704 42804E75 70024E75
POPX	, 12 + POP SWP &
FPAR	00000000
CC	C0000003 00000000 00000000 00000000

機械語

変換

処理系は一応インタプリタである。

コマンドはそのたびに辞書の頭からリンクをたどっていき、これが機械語であれば即実行、コマンド列になる、それ以外のコマンドによって同じことを繰り返すというわけである。

ソースのまま実行できるのが BASIC と同様デバッグには便利であるが、スピードは極めておそい。

コマンド CMPIL はコマンド列をマシンコードに変換する。ただし長さを変えない。

- 通常のコマンドは BSR (ブランチ+フェッチ) 命令ではおまかえる。
- 4バイト以下のマシン語にするものは、これを直接うめこむ。
- 小文字定数はスタックにプッシュする命令をうめこむ。

ソースとオブジェクトが全く同じ場所にあるからコンパイルは高速で、LISP インタプリタも2秒でコンパイルできる。

変換されたプログラムもインタプリタで動かされる。しかし今度、トップのコマンド以外は BSR のネストでつながれているから、ひとつのマシンコードプログラムとして動く。

例えば LISP インタプリタの場合、コンパイルにより約 100 倍 スピードアップされる

4. LISPインタプリタ

前記 FORTH の上に
簡単な LISP (pure
LISP より やや 遅い) の
インタプリタを作
ってみた。

右に示したのはその
一部である。

セルは 64 ビット、
頭の 3 ビットはタグ
および ガーベジマーク
用である。

セル領域は現在 64K
3 程の スタックに 16K
をあてている。

右下は SUBR 関数と
FORTH コマンドとの
対応表で、これも FORTH
ワードであるが、コン
パイル時に特殊の処理
をするので、戻性を別
にした。LISP リソースファイルも
当然 FORTH ワードである。

インタプリタであるから、速
度はあまり速くない。

.TARAI	4	82秒
SORT	2	17秒
8 QUEENS		143秒

というところである。

最近 512K バイト、CP/M-68K
付きのシステムを手に入れたので
こゝろに移植し、もう少し整備し
たいと考えている。

```
*P EVAL
C % D2 IFD EVAL0 ELSD EVAL1 THEN
*P EVAL0
, ATOM D2 IFD ATOMX RTF THEN , CAR FLAG 1 - D2 IFD ELSD
% IND RTF THEN ^ , CDR EVLIS SWP CAR FLAG 2 - D2 IFD ELSD
% IND RTF THEN FLAG 4 - D2 IFD ELSD CAR FUNC RTF THEN Q
, ATOM IF "*" ^ RTF THEN , CAR 3 - IF "#" ^ RTF THEN CDR
FUNC
*P EVLIS
, D2 IFD , CAR EVAL SWP CDR EVLIS CONS THEN
*P FUNC
, CAR FPAR & SWP APAR & DO APAR , CAR EXZ POP FPAR POP
, PUSX & LOOP ^ , CAR SWP CDR CAR EVAL SWP FPAR & DO FPAR
, CAR EXZ POP POPX LOOP ^
*P DEFIN
CDR , CDR , D SWP CAR , 4 + 4 SWP &B & LD
*P COND
DO CDR , EXZ , CAR , CAR EVAL D2 IFD CDR CAR EVAL SWP ^
0 EX THEN ^ LOOP
*P SETQ
CDR , CDR CAR EVAL , D SWP CAR & LD
*P CAR
25402013
4E75FFFF
*P TIMES
, CAR SWP CDR CAR * NORM
```

1 QUOTE QUOTE	3 NIL
3 T B	3 LAMBDA C
1 COND COND	1 DE DEFIN
1 PROG PROG	2 RETURN RETURN
1 GO GO	2 CAR CAR1
2 CDR CDR1	2 CAAR CAAR
2 CADR CADR	2 CONS CONS2
2 RPLACA RPLCA	2 RPLACD RPLCD
2 NCONC NCONC	2 ATOM ATOM1
2 EQ EQ1	2 EQUAL EQUAL
2 LIST LIST	1 OR OR1
1 SETQ SETQ	2 PRINT PSE1
2 PNUM PNUM	2 ZEROP ZEROP
2 MINUSP MINUP	2 MINUS MINUS
2 ADD1 ADD1	2 SUB1 SUB1
2 NULL NULL	2 GREATERP GRTP
2 DIFFERENCE DIF	2 PLUS PLUS
2 TIMES TIMES	2 MEMBER MEMB
1 BITSET BIS	1 BITCLR BIC
1 BITTEST BITST	

-
- (1) 池野：手作りマイコン bit 増刊 1980, 1
 - (2) 池野：68000 のすゝめ bit 1981, 6
 - (3) 池野：68000 システムと FORTH の試作 129-フェース 1982, 1