◎　ToUnderstandFORTHの簡単な解説

FORTHに関しての文献で、私が知っているものは１９９０年ごろに書かれた文献が多いのですが、いくつか今でも読むことができます。ここでは簡単な説明にとどめますので、詳しくはそちらを参照願います。

・標準FORTH　（日本語のみ。日本の図書館のインターネットライブラリから見ることができます。使用には登録が必要です。）

https://ndlsearch.ndl.go.jp/

（日本語画面）

https://ndlsearch.ndl.go.jp/search?cs=bib&collapse=null&display=panel&from=0&size=20&keyword=%E6%A8%99%E6%BA%96FORTH&f-ht=ndl&f-ht=library

（英語画面）

https://ndlsearch.ndl.go.jp/en/search?cs=bib&collapse=null&display=panel&from=0&size=20&keyword=forth+inoue&f-ht=ndl

・Starting FORTH　（WEB版なので各国語で読むことができます）

https://www.forth.com/starting-forth/0-starting-forth/

・A Beginner's Guide to Forth　　（英語のみ）

https://galileo.phys.virginia.edu/classes/551.jvn.fall01/primer.htm

あと、MSX用に開発されたFORTHのWEBもあります（事情があって、今は私のGitHub上に移動しています）

・H-FORTH\_MSX

https://github.com/MIN0/H-FORTH\_MSX

◎簡単な逆ポーランド記法とワードの説明

・通常の算術演算はポーランド記法で、以下のように行われます。

5 + 4 - 3 = 6

・FORTH言語はスタックを使って、逆ポーランド記法で行われます。

　FORTHのスタック操作による数値演算は以下になります。

　　１）数値ｎ１と数値ｎ２がスタックに積まれる。

　　２）ワード「＋」が数値ｎ１と数値ｎ２を取り出して演算する。数値ｎ１と数値ｎ２はスタックから消えた状態になります。

　　３）ワード「＋」の演算で求められた数値ｎ３がスタックに積まれる。

　その時のスタックの状態を図で示します。一つ数値がスタックに積み上げられると、図では最上位の数値の上にその数値が上に載って、その数値が最上位となります。

　　　注意１）　数値に（）がかぶさっている場合はそれがスタックの最上位であることを示す。

　　　注意２）　-----はスタックの底辺で、スタックをすべて吐き出すと空になることを表す。

　　　注意３）　ここでの数値以外の「＋」や「ー」、「＊」はワードと呼ばれる。ワードはスタック上のデータの処理や様々な処理を実行し、定義することができる。

a)

4 5 + であれば、５と４を足して結果は９。

(5)

(4) --> 4 --> + --> (9)

----- ----- -----

b)

3 9 - であれば、９から３を引いて結果は６。

(9)

(3) --> 3 --> - --> (6)

----- ----- -----

c)

3 4 5 + - であれば、処理は左から右に実行されます。数値がスタックに積まれる。ワード「＋」で一番上から演算されて、数値５と数値４が消えた後に演算結果の数値９がスタックに積まれます。ワード「ー」で一番上から演算されて、数値９と数値３が消えた後に演算結果の数値６がスタックに積まれます。これが通常のポーランド記法での５＋４－３の結果です。

(5)

(4) 4 (9)

(3) --> 3 --> 3 --> + --> 3 --> - --> (6)

----- ----- ----- ----- -----

結果がするために処理はスタックにデータを積み上げてはスタックの一番上の内容と次の内容を演算処理、その結果をまたスタックの一番上に積み上げます。

・これは算術演算の一部のワードの話であり、ワードによって実行時に消費されるスタック数とそのあとに積み上げられるスタックの数はさまざまである。

　FIG-FORTHでは理解を助けるために次のような表示の仕方をしている。"("と")"の間に書かれた文字列はコメントとして扱われて、実行されません。

( n11 n12 n13 --- n21 n22 )

(n13)

n12

n11 --> 数値はスタックから n13 --> n12 --> n11 の順に取り出される。

-----

実行前のスタックの状態

(n22)

n21 --> 数値は n21 --> n22 の順にスタック上に積み上げられる。

-----

実行後のスタックの状態

◎ToUnderstandFORTHを動かしてみる。

・使えるPCはWindowsです。「今までの開発状況と現在の課題」を参考に、Visual Studio 2022を立ち上げて、「デバッグに開始（S)」をクリックする。少し時間が経ってからコマンドプロンプトが表示され、FORTHの初期メッセージの表示後にキー入力待ちとなる。。

ー入力待ちとなる。

・そこにFORTHのワードを並べてキー入力を行い、最後にENTERキーを押す。

・キー入力したFORTHのワード列が実行され、画面に文字を表示したりとワード列が実行される。

・すべてのキー入力したFORTHのワード列の実行が終了すると、画面に「OK」が表示される。もしも実行の途中ででエラーが発生すると、何らかのメッセージを表示して、通常は画面に「OK」が表示される。

例えば、「＋」や「ー」、「＊」を使って数値計算をすると以下のようになります。

［．］は計算した結果を画面に表示します。

ここでの計算には通常のポーランド記法でなくて、逆ポーランド記法で行われます。

4 5 + . [ENTER] --> 9 OK ４と５を足す

3 4 5 + - . [ENTER] --> 6 OK ４と５を足す。その結果から３を引く。

2 3 4 5 + - \* . [ENTER] --> 12 OK ４と５を足す。その結果から３を引く。さらにその結果に２を掛ける。

◎現在、ToUnderstandFORTHで正常に実行することが確認されているワード

　今のところ、正常に実行できるワードは以下のとおりである。

　　１）数値、「＋」や「ー」、「＊」、「．」（ドット）を使った場合は正常な結果が表示されている。

例） 入力内容：1 2 + . [enter]

出力内容： 3 OK

　　２）文字列が画面に表示される。

入力内容：." Hello! " [enter]

出力内容：Hello! OK

　　３）ワード TRACE\_ON、TRACE\_OFFは新規に作成したワードで、今実行されているワード名や使用しているレジスタ等の情報を画面に「表示する／表示しない」の切り替えを行います。

　　　　例） 入力内容：TRACE\_ON 1 . TRACE\_OFF [enter]

　　　　　　　出力内容：　（画面に表示される量が多すぎるので省略するが、1 . を各ワード（WORDやPFIND等）で解析し、実行していく状況が表示される）

◎上記以外はおかしな動作をするか、忙しくてまだチェックができていないかのどちらかの状態です。まだ開発途上にあります。次はそうした例の一つです。

　　１）辞書に新しいワードを登録する（「：」と「；」で囲まれたFORTH文を実行する）際に、処理の途中でToUnderstandFORTHが確保している範囲のアドレス以外をアクセスするためにエラーが発生する。処理はリセットされて初期状態に戻る。

入力内容： : TEST\_NAME 1 2 + . ; [enter]

　　　　　　 出力内容：　WORD\_POINTER\_ADDRESS(R8): : =2032203120454d2f

WRITE\_REGISTER\_NUMBER(R9): : =103

7ff7b6ac2bb0 = 7ff7b6ac8e27

7ff7b6adc000 = 7ff7b6af5fdf

BREAK\_POINT\_Stack 1st : =7ff7b6ac721f

BREAK\_POINT\_Stack 2nd : =4e5f54534554203a

BREAK\_POINT\_NUMBER : =10403

PUSH 'Y' KEY TO COLD START