スタックマシン仕様ver1

**メモリ領域**

・命令を格納する（参照で格納、実体は別で）

・定数を格納する

・疑似レジスタを格納する

・余った部分はスタック領域として使う

・容量は65536バイト

・メモリ一つのサイズは1バイト

**扱える型**

・int(4バイト)

・double(8バイト),

・bool(1バイト)

・char(1バイト)

・pointer(4バイト)

**疑似レジスタ**

・SPレジスタ（4バイト）

　スタック領域の一番上のデータを指す

・BPレジスタ（4バイト）

　SP退避用

・PRレジスタ（4バイト）

　次の命令の場所を指す

・AXレジスタ（8バイト）

　計算結果を保存するときに使う

・GR1～8レジスタ（8バイト）

　汎用レジスタ

**命令**

実際に実行する命令

**・用語定義**

i:int

d:double

b:bool

c:char

p:pointer

<type>: i,d,b,c,pのいずれか

adr:アドレス

x:自身の左となりにあるadrの引数に加算する値。(xアドレスの中身を使って加算)。Xアドレスはint\*

**・Load命令**

　adr2の中身をadr1に入れる。adrは<type>型にキャストして扱う。

<type>LD adr1,adr2

<type>LD adr1,adr2,x2

<type>LD adr1,x1,adr2

<type>LD adr1,x1,adr2,x2

**・Jump命令**

PRレジスタにadrを代入する

JMP adr

JMP adr,x

**・分岐命令**

adr2の中身がtrueならば、adr1をPRレジスタに代入する。adr2はbool型にキャストして扱う。

JMPB adr1,adr2

JMPB adr1,adr2,x2

JMPB adr1,x1,adr2,x2

JMPB adr1,x1,adr2

**・算術・論理演算命令**

adr1とadr2を足したものを、adr1に代入。adrは<type>型にキャストして扱う。

ただし、typeがpointerならadr2はint型にキャストして扱う

<type>ADD adr1,adr2

<type>ADD adr1,adr2,x2

<type>ADD adr1,x1,adr2

<type>ADD adr1,x1,adr2,x2

**・比較演算命令**

adr1とadr2の中身を比較し、等しかったらAXレジスタにtrue、違うならfalseを代入する。

　adrは<type>にキャストして扱う。

<type>CPA adr1,adr2

<type>CPA adr1,adr2,x2

<type>CPA adr1,x1,adr2

<type>CPA adr1,x1,adr2,x2

**・シフト演算命令**

**・スタック操作命令**

adr1の中身をスタックに溜める

<type>PUSH adr1,x1

<type>POP adr1,x1

**・プリセット関数呼び出し命令**

PCALL 即値

**・型変換命令**

adr1<type1>を<type2>にキャストして、adr2に入れる

<type1>CAST<type2> adr1,adr2

　<type1>CAST<type2> adr1,adr2,x2

<type1>CAST<type2> adr1,x1,adr2

<type1>CAST<type2> adr1,x1,adr2,x2

**・参照命令**

adr1にadr2のアドレスを入れる。

REF adr1,adr2

REF adr1,adr2,x2

REF adr1,x1,adr2

　REF adr1,x1,adr2,x2

**・逆参照命令**

adr1にadr2の中身を<type\*>にキャストして\*したものを入れる

<type>DREF adr1,adr2

<type>DREF adr1,adr2,x2

<type>DREF adr1,x1,adr2

　<type>DREF adr1,x1,adr2,x2

**・逆参照Load命令**

adr1の中身を<type\*>にキャストして\*したものにadr2<type\*>の中身を入れる

<type>LDR adr1,adr2

<type>LDR adr1,adr2,x2

<type>LDR adr1,x1,adr2

　<type>LDR adr1,x1,adr2,x2

**マクロ命令**

スタックマシン実行前に行う前処理

**・ラベルについて**

いわゆる#defineです。

**・静的領域確保命令**

領域を確保して、即値を入れる。ラベルにその領域の先頭アドレスが入る。

　即値の型は<tyep>で、型推論。

ラベル　DC 即値

　ラベル　DC 即値,即値,即値.......

　即値の数だけ領域を確保する。ラベルにその領域の先頭アドレスが入る。

　即値の型はintとして扱う

ラベル　DS 即値

**・命令マーク**

　命令のアドレスをラベルに入れる

ラベル 命令

**メモリ領域配置図**

SPレジスタ(4byte)

BPレジスタ(4byte)

AXレジスタ(8byte)

PRレジスタ(4byte)

命令格納領域

静的領域

残りは全てスタック領域

GR1～GR8レジスタ(8byte)