OS9，GAM

OS9の特徴

Microware 社によりMotorola のMC6809用に作られた

- Module と言う単位をメモリ上にどこに配置し
- Time sharing を採用した並列実行(concur
- Unix like なshell とpipe
- Basic09 というPascal likeな言語を持つ。

Level 1/2

level 1 ROM上のOS9 p1 kernel で動作する。

level 2 MMUで2Mbyteのメモリを使える

- アドレス変換に対応し、512kメモリを使用でき

ME09, FORT

8bit OS。1980年初頭。

ても良い

rrent) (平行(parallel)ではない)

,

きる。

nit

OS9 をdisass

Tandy Coco

ライセンス自

大目に見らね

En

sbc09という

TH, TL/1

tros9

emble したものらしい

上で動いていたらしい

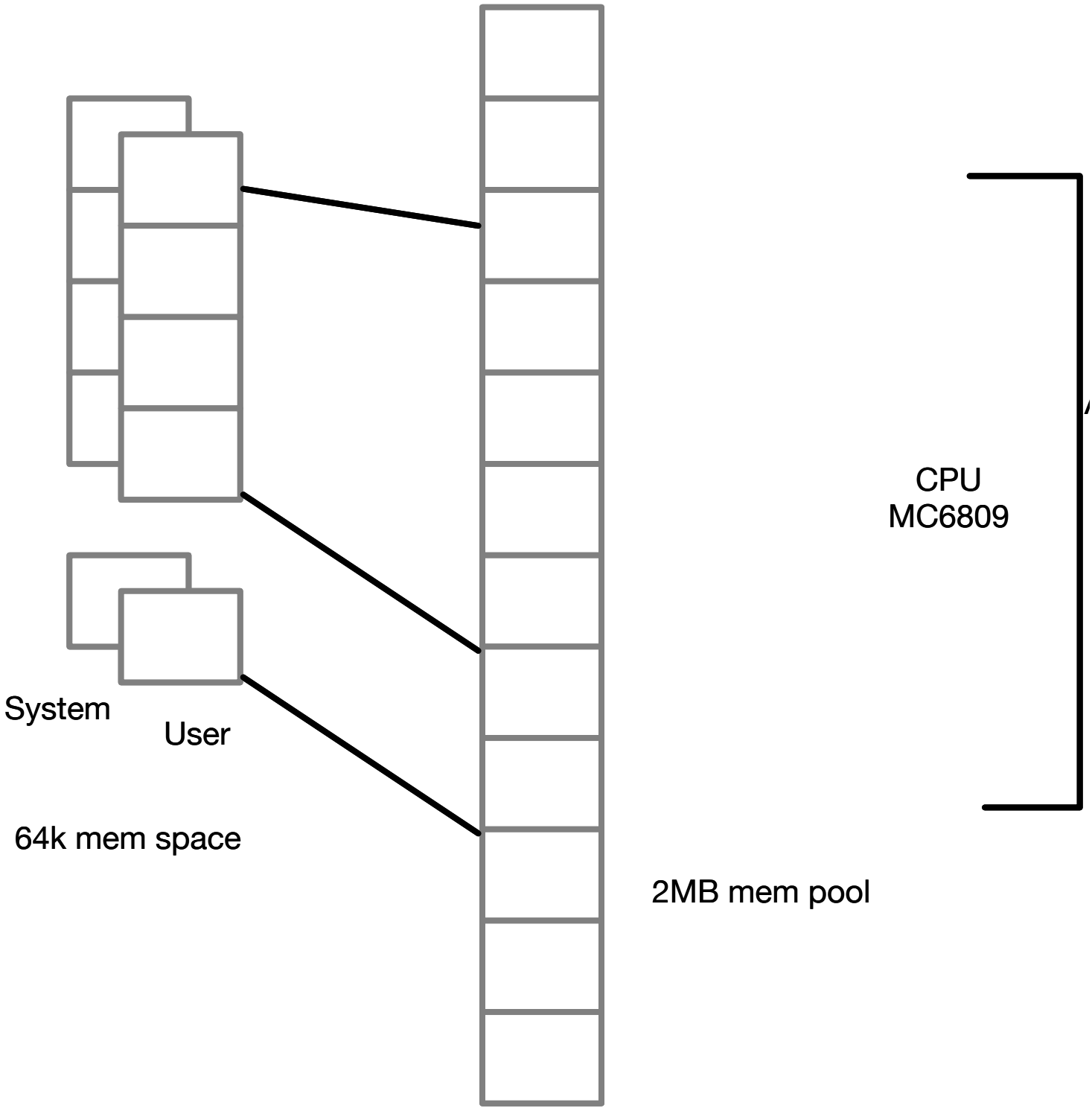
匂にはだめかも

てる?

mulator

アセンブラEmulator上に実装して動作させた

MC6809アーキテク

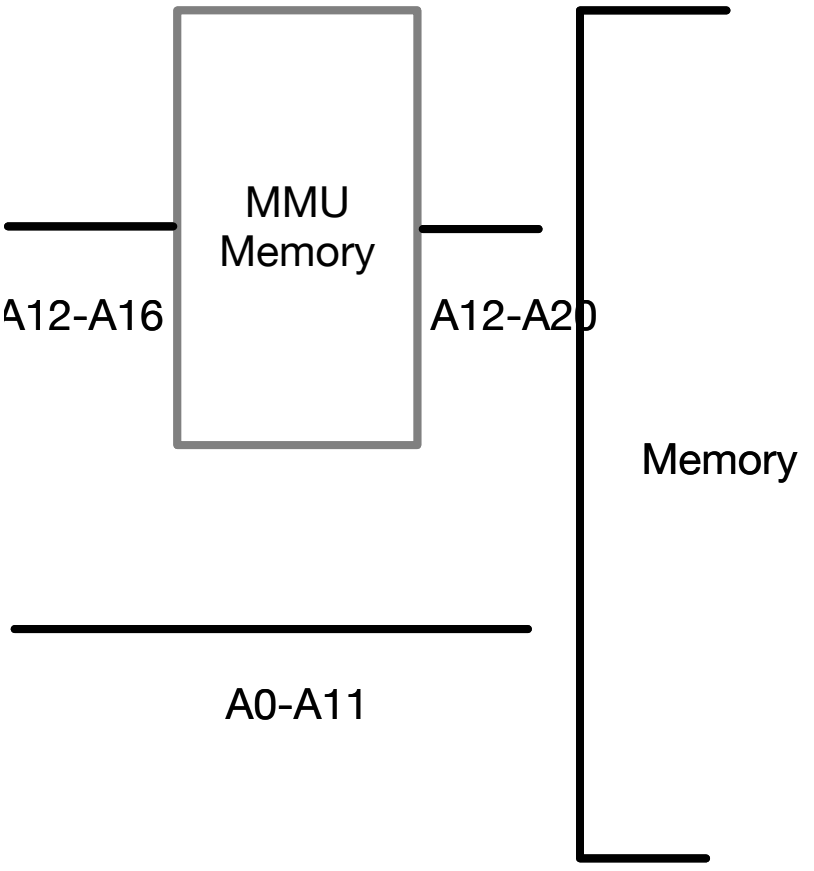


sbc09 を mm

仮想RBF (ra

```
Unix
```

チャ



FC

FORTH

PostScript の

関数単位

```
assembl
```

```
: ACCEP
  CR .
  ABUF
  CR .
  ABUF
```

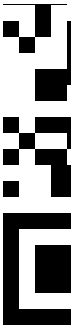
GAME09

VTLの日本で

記号的なBA

u 対応にして level 2 まで動かした。

ndom block filer manager)



上のファイルを Emulator 側からos9のファイルシ

ORTH, GAME09

元になった言語

er を自分でかける

```
T-TEST
" PLEASE TYPE UP TO 80 CHARACTERS:" CR
 80 ACCEPT
" RECEIVED: " [CHAR] " EMIT
 SWAP TYPE [CHAR] " EMIT CR
```

での実装

SIC

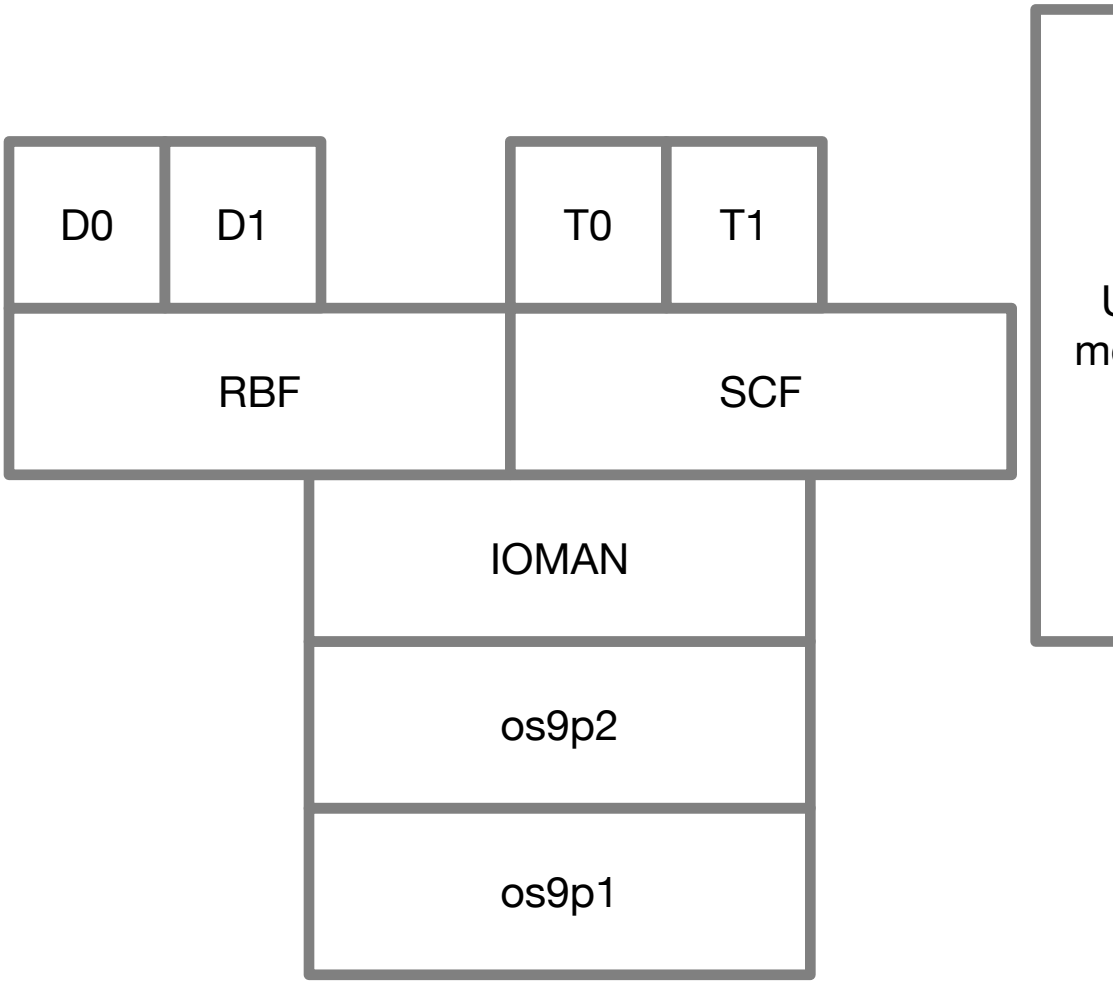
```
100   "ADDRESS=" A=?
110   "TO      =" D=?
120   @=(A>D) #=-1
900   / ??=A .=20 E=A
```

ステムとして見せる

@ !=900

. !=1000

OS9のkernel構成



kernel構成

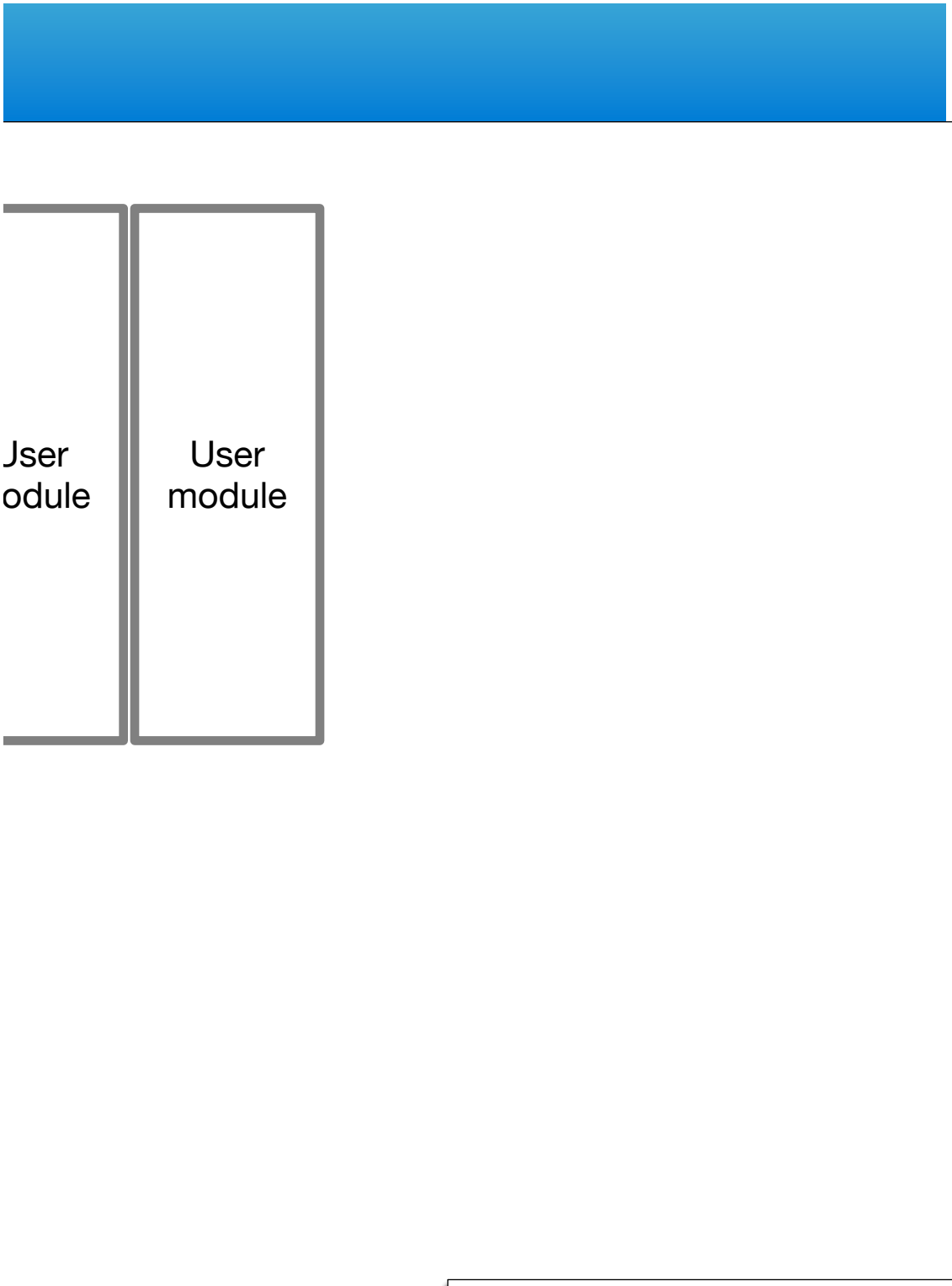
OS9p1

`system call`と割り込み処理

`Module` 発見と管理

OS9p2

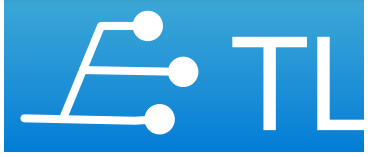
メモリ管理



コンパイラ
行番号なし

mohta氏と手塚

```
04D7:
04D7:
04D7:
04D7: 3
04D9: 3
04DB: E
04DD: 3
04DF: 1
04E2: 3
04E4:
04E4: 3
```



TL/1
大西氏のコンパイラ言語

データ型はByteとByte Arrayのみ。

手続きと関数がある。

とかが書かれた 割とダメ。

り rvtl というのがある。

```
910   $=$D .=5 B=0,5
920   ?$=E:B) " "
```

icroC

氏の作った 6809 用の整数Cコンパイラ。構造体がある。

```
                    * getchar()
                    * {     return getc(stdin);
                    getchar
440                         PSHS    U
3E4                         LEAU    ,S
C20                         LDD     0,Y
406                         PSHS    D
7FFB7                       LBSR    getc
262                         LEAS    2,S
                    * }
5C0                         PULS    U,PC
```

L/1

```
0233: 17 FE C6      LBSR        $00FC
0236: 16 FE D6      LBRA        $010F
0239: AF E3         STX         ,--S
023B: 30 86         LEAX        A,X
023D: 86 04         LDA         #$04
023F: A7 02         STA         +$02,X
```

;=E+B>=A .=3 #=930

呼出側の局所変数の確保

Task管理

Signal

kernel構成2

IOMan

SCF/RBFと device driver とdescriptor

SCF

sequencial file io manager

RBF

randome block file io manager

file system管理

の登録

```
PROC WAIT
FUNC TIME
%--- MAIN ---
VAR MMI,MMJ,MMK
BEGIN
  WRITE(1:ASCII($A))
  MMI:=1 MMJ:=2 MMK:=3
  WAIT(4,5)
  WRITE(1:MMI,CRLF)
END
```

```
0241: 86 06      LDA     #$06
0243: A7 04      STA     +$04,X

0245: 86 07      LDA     #$07
0247: A7 05      STA     +$05,X
0249: 86 04      LDA     #$04
024B: 17 00 24   LBSR    $0272
024E: A7 03      STA     +$03,X
0250: 86 01      LDA     #$01
0252: 97 01      STA     <$01
0254: A6 20      LDA     +$00,Y

0256: 17 FE 14   LBSR    $006D
0259: A6 02      LDA     +$02,X

025B: 17 FE 0F   LBSR    $006D
025E: A6 00      LDA     +$00,X
0260: 17 FE 0A   LBSR    $006D
```

局所変数

大域変数

呼び出した方の引数