

ca-2. メモリとアドレス, ダンプリスト

(コンピュータ・アーキテクチャ演習)

URL: <https://www.kkaneko.jp/cc/ca/index.html>

金子邦彦



アウトライン



2-1 メモリとアドレス

2-2 メモリへの操作

2-3 16進数の表記

2-4 ダンプリスト

2-1 メモリとアドレス

メモリとは



- **メモリ**は、データの記憶を行うチップ
- データを覚えさせたり (**書き込み**) , 取り出したり (**読み出し**) の機能がある

メモリとアドレス



- **メモリはバイト (8ビット) 単位に区切られている**
- **各バイトには0から始まる通し番号**が付けられている。これを**アドレス**という (番地ともいう)

メモリ内のデータは

01	00	00	00	02	00	00	00	03	00
----	----	----	----	----	----	----	----	----	----

0 1 2 3 4 5 6 7 8 9

アドレス

メモリアドレス

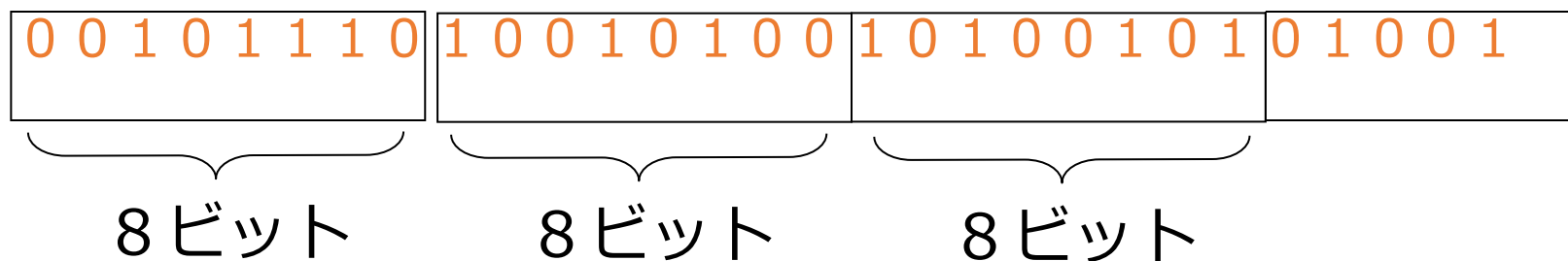


- メモリアドレスは、読み書きすべきデータの「場所」を示す

(例) 0065FDF0 (16進)

→ メモリ先頭から0065FDF0 (16進数) 番目
という意味

- メモリ内のデジタルデータは、8ビットずつ区切られて、メモリアドレスが付けられている



メモリアドレスは、ふつう16進数表記する



```
C [dropdown] Enter a title here
Main.c [x] [+]
1 #include <stdio.h>
2 int main(void){
3     double teihen = 3;
4     double takasa = 4;
5     double menseki;
6     menseki = teihen * takasa / 2.0;
7     printf("&teihen %x\n", &teihen);
8     printf("&takasa %x\n", &takasa);
9     printf("&menseki %x\n", &menseki);
10 }
11
```

実行 (Ctrl-Enter)

出力 コンパイルエラー 入力 コメント 0

```
&teihen cf323990
&takasa cf323988
&menseki cf323980
```

16進数

実行結果の例

なぜ 16 進数なのか



- メモリアドレスそのものもデジタル（「0」, 「1」の列）
- といって, メモリアドレスを, 「0」, 「1」の並びで書くのは, 長すぎて人間にとって分かりづらい

(例)

0 0 1 0 1 1 1 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 1 0 1 0 0 1

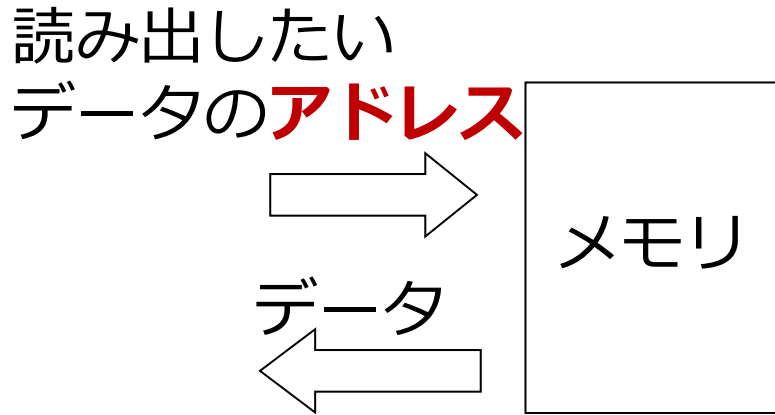
そこで, 「16 進数」を使う

2-2 メモリへの操作

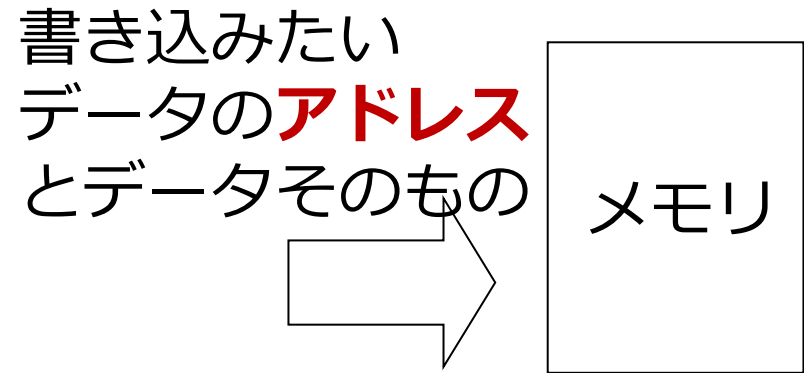
メモリへの操作



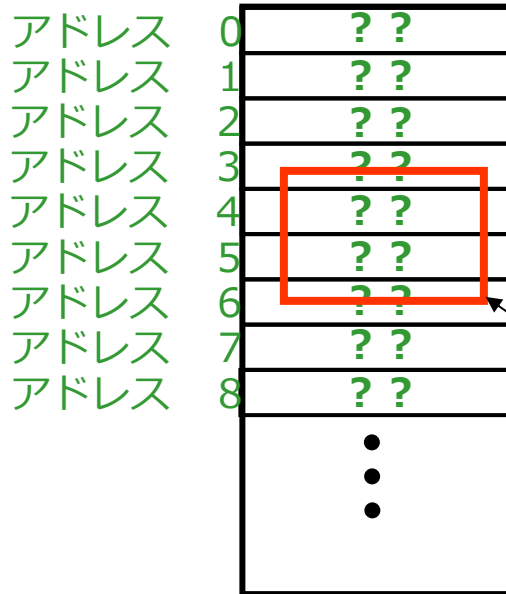
• 読み出し



• 書き込み



読み出し



アドレス4番地, 5番地
から2バイト分
読み出すとき

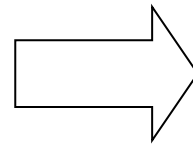
メモリの値は変化
しない

メモリの各区画は1バイト
(16進数で2桁)

書き込み

アドレス 6 番地,
7 番地に
「0400」を
書き込むと

アドレス	0	??
アドレス	1	??
アドレス	2	??
アドレス	3	??
アドレス	4	??
アドレス	5	??
アドレス	6	??
アドレス	7	??
アドレス	8	??
		⋮



	??
	??
	??
	??
	??
	??
	??
	0 4
	0 0
	??
	⋮

メモリの各区画は 1 バイト
(16 進数で 2 桁)

前の値は消える

2-3 16進数の表記

16進数の表記



- 4000

「4000」と書いただけでは、
10進数なのか 16進数なのか分からない

2-3 16進数の表記

- 4000h

10進数

16進数

16384



0x4000

同じ値

何も付けない

末尾に h を付けたり。
(主にアセンブリ言語での習慣)

頭に 0x を付ける。
(主にC言語での習慣)

16進数であることを示す目印

2-4 ダンプリスト

メモリとアドレス



- **メモリはバイト (8ビット) 単位に区切られている**
- **各バイトには0から始まる通し番号**が付けられている。これを**アドレス**という (番地ともいう)

メモリ内のデータは

01	00	00	00	02	00	00	00	03	00
0	1	2	3	4	5	6	7	8	9

アドレス

ダンプリストの例



0x002F8130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x002F813E	00	00	00	00	00	00	01	00	00	00	00	00	00	00	00
0x002F814C	00	00	00	00	00	00	00	00	01	00	00	00	88	ddh)	
0x002F815A	1b	00	b0	de	1b	00	00	00	00	00	00	00	00	00	00	..-^.....
0x002F8168	00	00	00	00	00	00	00	00	00	00	00	00	60	91	00
0x002F8176	28	57	00	00	00	00	00	00	00	00	00	00	00	00	00	(W.....
0x002F8184	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

アドレス

メモリの中身

バイト単位で区切られて表示.

表示は16進数

ダンプリストの例



アドレスは
0x002f8130

アドレスは
0x002f8131

アドレスは
0x002f8132

0x002F8130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x002F813E	00	00	00	00	00	00	01	00	00	00	00	00	00	00	00
0x002F814C	00	00	00	00	00	00	00	00	01	00	00	00	68	ddh)	
0x002F815A	1b	00	b0	de	1b	00	00	00	00	00	00	00	00	00	..-^.....	
0x002F8168	00	00	00	00	00	00	00	00	00	00	00	00	60	91	
0x002F8176	28	57	00	00	00	00	00	00	00	00	00	00	00	00	(W.....	
0x002F8184	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

確認クイズ



このアドレス
は？

このアドレス
は？

0x002F8130	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x002F813E	00	00	00	00	00	00	01	00	00	00	00	00	00	00
0x002F814C	00	00	00	00	00	00	00	01	00	00	00	68	ddh)	
0x002F815A	1b	00	b0	de	1b	00	00	00	00	00	00	00	00)	
0x002F8168	00	00	00	00	00	00	00	00	00	00	00	60	91)	
0x002F8176	28	57	00	00	00	00	00	00	00	00	00	00	00	(W.....	
0x002F8184	00	00	00	00	00	00	00	00	00	00	00	00	00	

アドレス

メモリの中身が**バイト単位**
で区切られて表示

メモリの中身
を文字に置き換え
て表示

確認クイズの答え



0x002F814C

左のアドレス
表示で分かる

0x002F8150

これは 0x002F814C に 4 足した値

0x002F8130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x002F813E	00	00	00	00	00	00	01	00	00	00	00	00	00	00	00
0x002F814C	00	00	00	00	00	00	00	01	00	00	00	68	dd	h)	
0x002F8154	1b	00	00	00	00	00	00	00	00	00	00	00	00	
0x002F8168	00	00	00	00	00	00	00	60	91	
0x002F8176	28	57	00	00	(W	
0x002F8184	00	00	00	00	

4 つ右隣
り

アドレス

メモリの中身がバイト単位
で区切られて表示

メモリの中身
を文字に置き換え
て表示

実行型ファイルのダンプリストの例



アドレス

```
#include "stdafx.h"
#include <math.h>

int main()
{
    double d = 2.0;
    printf("%f, %f %n", d, sqrt(d));
    return 0;
}
```



ビルド

00000000	4D 5A 90 00 03 00 00 00	04 00 00 00 FF FF 00 00	MZ.....
00000010	B8 00 00 00 00 00 00 00	40 00 00 00 00 00 00 00@.....
00000020	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00000030	00 00 00 00 00 00 00 00	00 00 00 00 F0 00 00 00
00000040	0E 1F BA 0E 00 B4 09 CD	21 88 01 4C CD 21 54 88!..L.!Th
00000050	69 73 20 70 72 6F 67 72	61 6D 20 63 61 6E 6E 6F is program canno
00000060	74 20 82 65 20 72 75 6E	20 69 6E 20 44 4F 53 20 t be run in DOS
00000070	6D 6F 64 65 2E 0D 0D 0A	24 00 00 00 00 00 00 00 mode....\$......
00000080	FF FA 97 C6 B8 98 F9 95	BB 9B F9 95 BB 9B F9 95
00000090	FD CA 24 95 B8 98 F9 95	FD CA 26 95 BA 9B F9 95\$......&.....
000000a0	FD CA 19 95 A8 98 F9 95	FD CA 18 95 BC 9B F9 95
000000b0	66 64 32 95 B9 98 F9 95	BB 9B F8 95 83 9B F9 95fd2.....
000000c0	B6 C9 18 95 B9 98 F9 95	B6 C9 22 95 BA 9B F9 95
000000d0	B6 C9 27 95 BA 98 F9 95	52 69 63 68 BB 9B F9 95Rich.....
000000e0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
000000f0	50 45 00 00 4C 01 07 00	A8 13 37 55 00 00 00 00PE..L.....7U.....
00000100	00 00 00 00 E0 00 02 01	0B 01 0C 00 00 3C 00 00<.....

実行型ファイル
※ マシン語

ソースファイル
※ C++言語