

# 学問への扉

## 第4回（高野分）

大阪大学 高野祐輝

[ytakano@cy2sec.comm.eng.osaka-u.ac.jp](mailto:ytakano@cy2sec.comm.eng.osaka-u.ac.jp)

# 資料配布URL

- <https://github.com/ytakano-lecture/Science2021>

表記

# 表記

表記	意味
0xaff12	aff12という16進数値
100 B	100 bytes
3 KiB	$3 \text{ KiB} = 3 \times 10^{10}$
5 MiB	$5 \text{ MiB} = 5 \times 10^{20}$
6 GiB	$6 \text{ GiB} = 6 \times 10^{30}$
7 TiB	$7 \text{ TiB} = 7 \times 10^{40}$

コンピュータアーキテクチャ

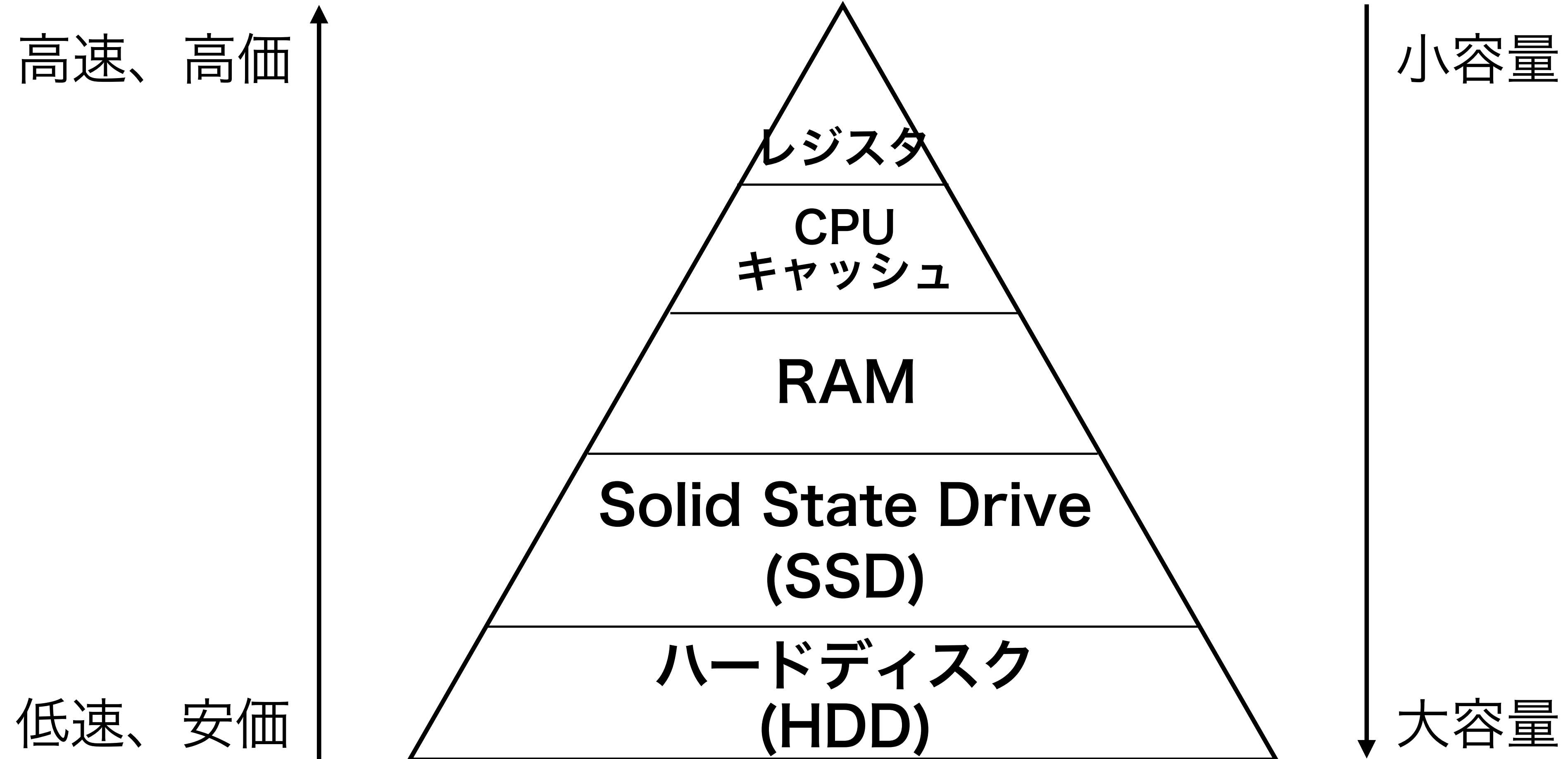
# コンピュータの主なコンポーネント

- ・ メモリ: データを一時的に保存しておくデバイス。主記憶装置とも呼ばれる
- ・ HDD or SSD: データを永続的に保存しておくデバイス。補助記憶装置とも呼ばれる
- ・ CPU: 計算を実行するためのデバイス。人間の脳に喩えられる

# CISCとRISC

- Complex Instruction Set CPU (CISC)
  - Intel x86, AMD x86-64  
(PC、PlayStation 4と5)
- Reduced Instruction Set CPU (RISC)
  - ArmのCPU (スマートフォン、タブレットPC、Nintendo Switchなど)
  - Power PC (PlayStation 3など)
  - 他にも、MIPSやRISC-Vなどがある

# メモリストレージピラミッド





# メモリとストレージのサイズ

- レジスタ  
数十から数百バイト
- CPUキャッシュ  
数十から数百MiB (L3)
- Random Access Memory (RAM)  
数十GiB
- Solid State Drive (SSD)  
数百GiB
- Hard Disk Drive (HDD)  
数百GiBから数TiB

$$1 \text{ MiB} = 2^{20} \text{ B} = 1,048,576 \text{ B}$$

$$1 \text{ GiB} = 2^{30} \text{ B} = 1,073,741,824 \text{ B}$$

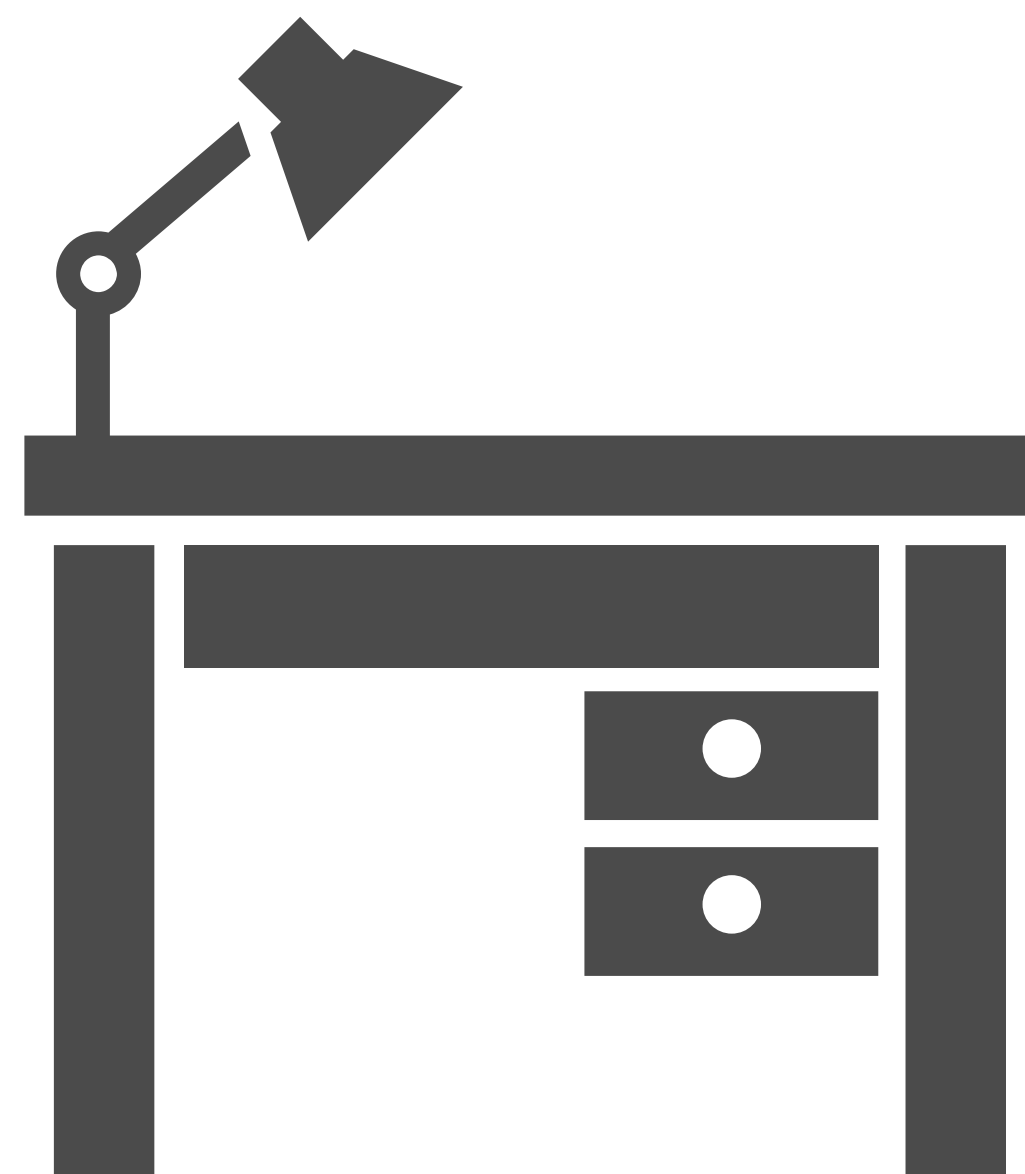
$$1 \text{ TiB} = 2^{40} \text{ B} = 1,099,511,627,776 \text{ B}$$

# 揮発性と不揮発性メモリ

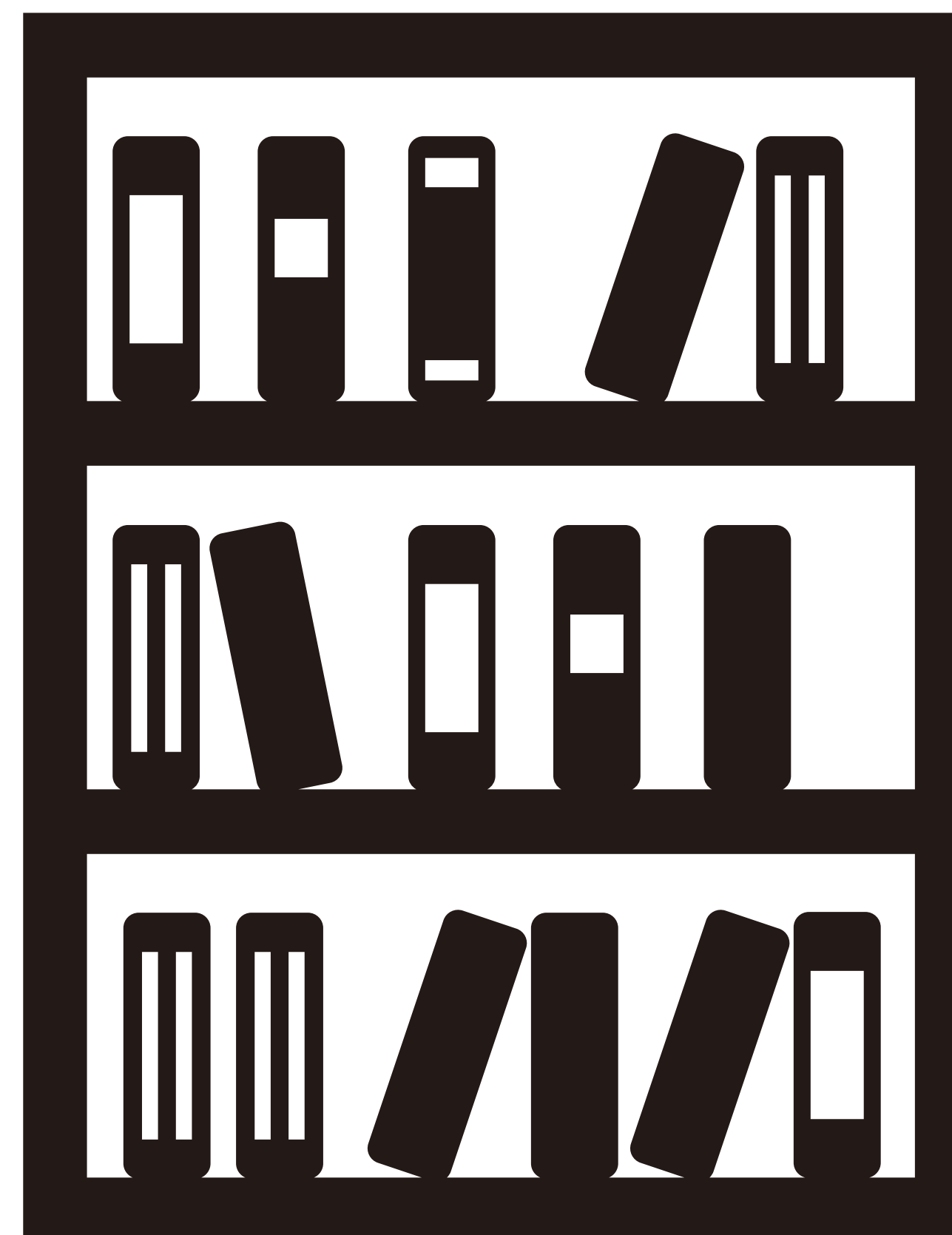
- ・ 揮発性メモリ
  - ・ 電源供給が途絶えると、保存されているデータは消える
  - ・ RAM、CPUキャッシュ、CPUレジスタ
- ・ 不揮発性メモリ
  - ・ 永続的にデータが利用可能
  - ・ SSD、HDD

# 喩えるなら

メモリは喩えるなら机の広さ  
机が広いと多くの資料を同時に手早く  
アクセス可能



HDD, SSDは喩えるなら本棚  
作業を終えてもデータが保管されている  
が、アクセス速度は遅い

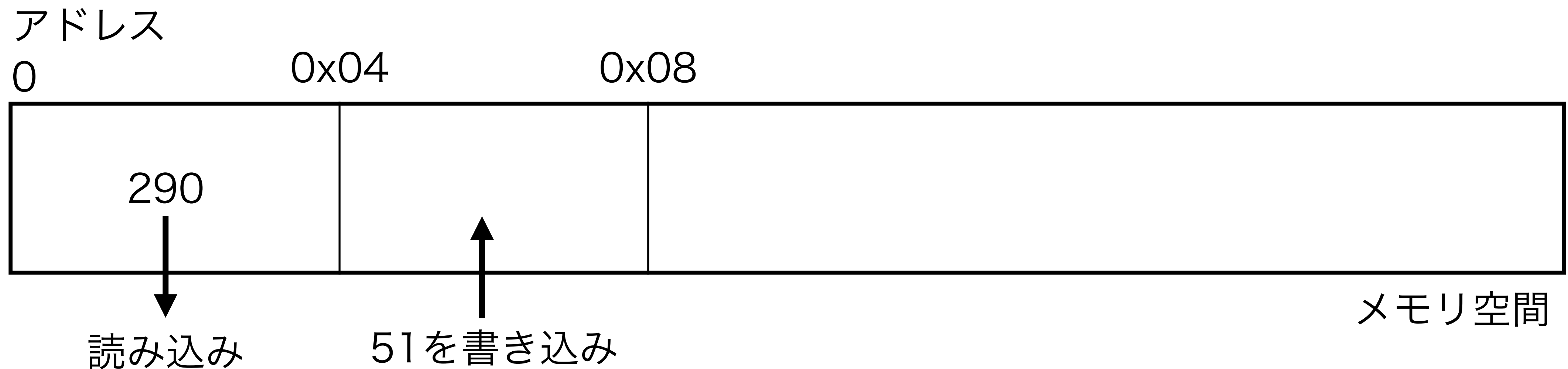


# レジスタ

- CPUの内部にある保存領域
- Arm (AArch64)の場合、x0からx30までの汎用64ビットレジスタを持つ
- x86-64は16の汎用64ビットレジスタを持つ

# RAM

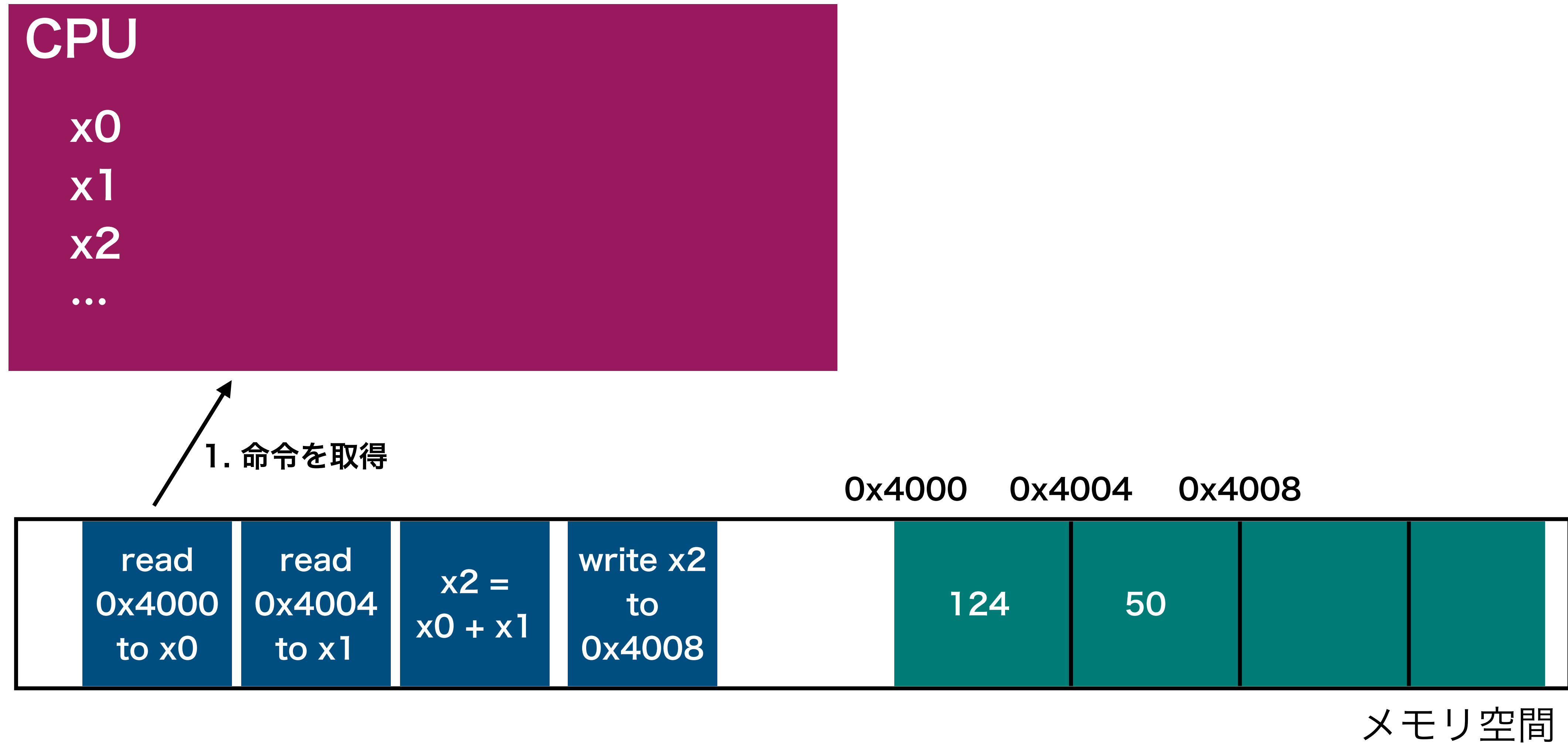
- メモリへは数字でアクセス可能
- 例：
  - 0アドレスから4バイト読み込む
  - 0x04アドレスへ51（4バイトの値）を書き込む



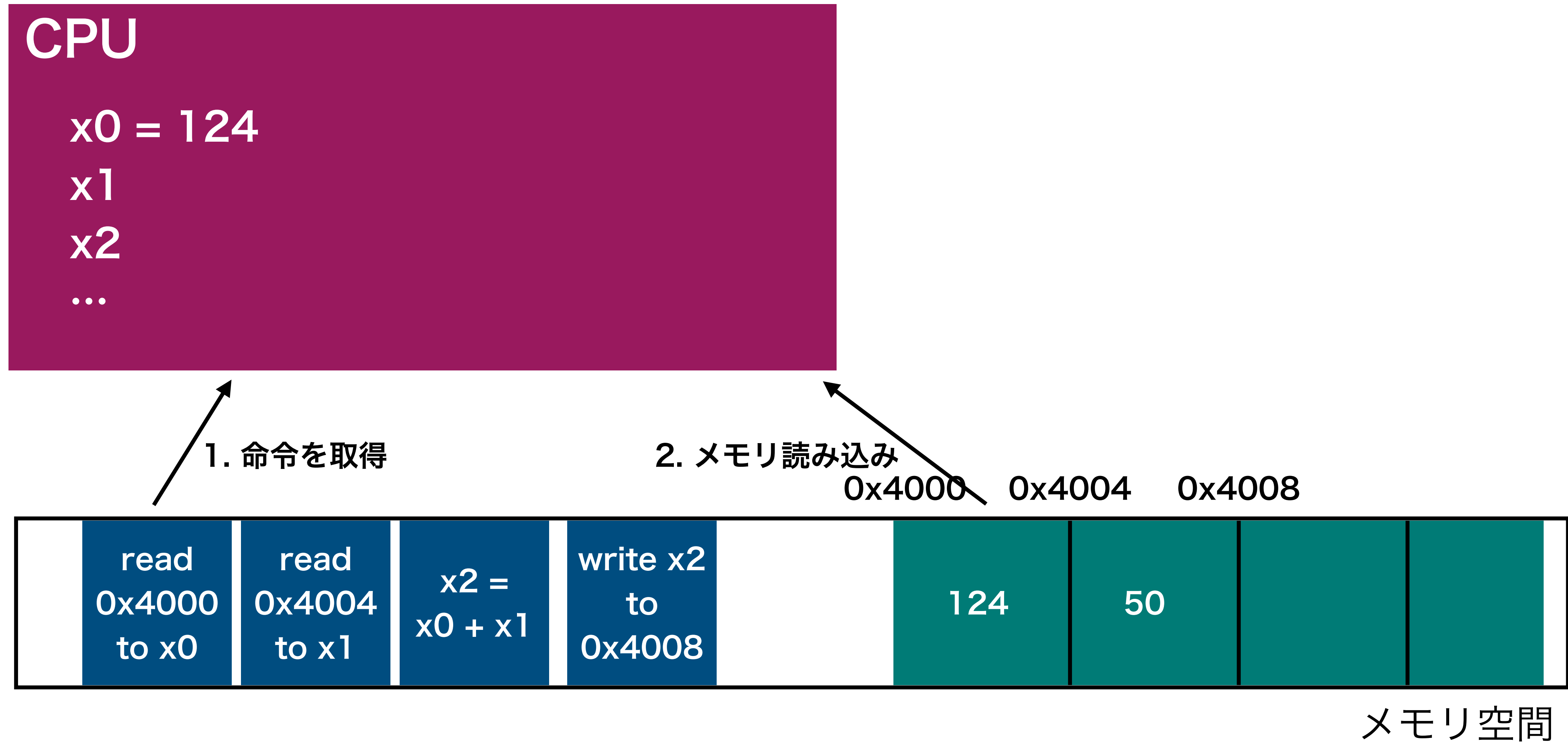
# コンピュータによる計算の流れ

1. CPUがメモリから命令を読み込み
2. CPUは読み込んだ命令を実行
3. 必要なら、CPUは2の結果をメモリに書き込む
4. 以上を繰り返す

# 計算実行例 (1)



# 計算実行例 (2)





# 計算実行例 (3)

CPU

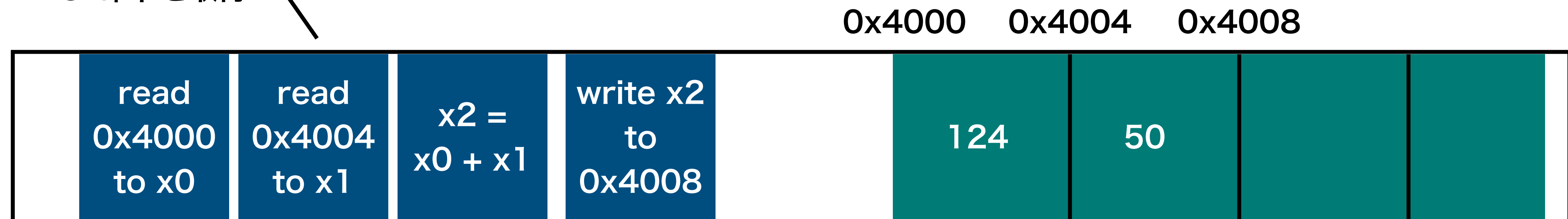
x0 = 124

x1

x2

...

3. 命令を取得



メモリ空間

# 計算実行例 (4)

CPU

x0 = 124

x1 = 50

x2

...

3. 命令を取得

4. メモリ読み込み

0x4000

0x4004

0x4008



メモリ空間

# 計算実行例 (5)

CPU

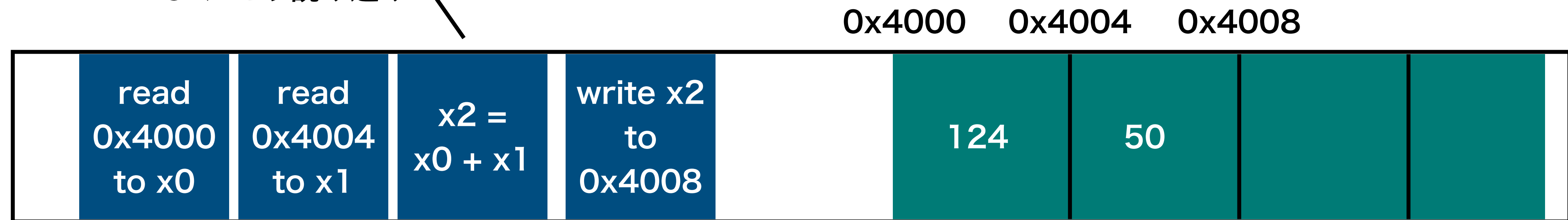
x0 = 124

x1 = 50

x2

...

5. メモリ読み込み



メモリ空間

# 計算実行例 (6)

CPU

x0 = 124

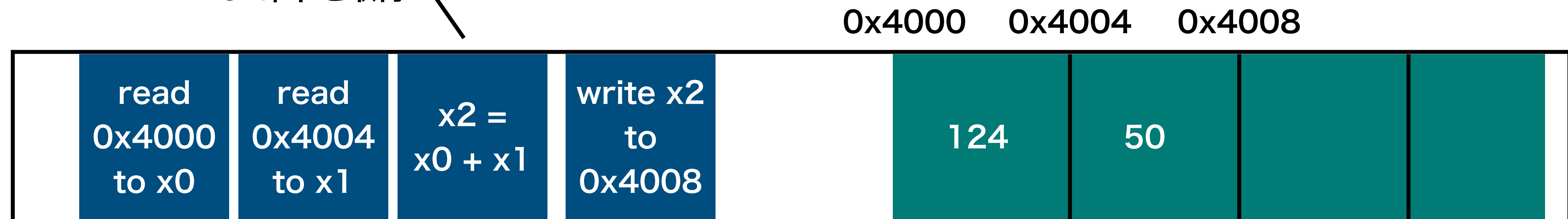
x1 = 50

x2 = 174

...

6. 計算を実行

5. 命令を取得



メモリ空間

# 計算実行例 (7)

CPU

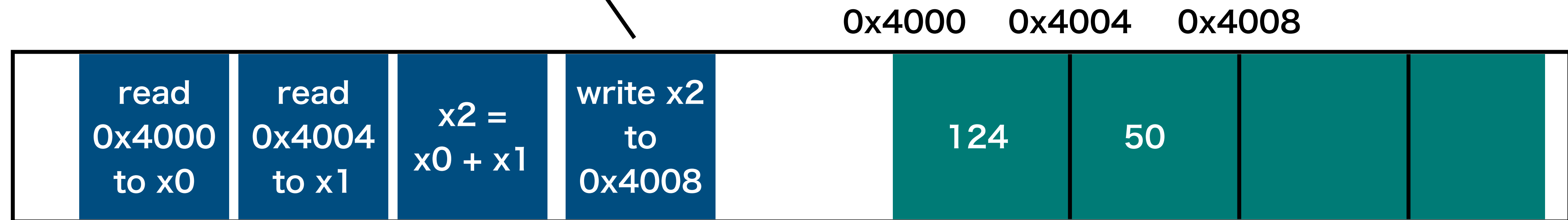
x0 = 124

x1 = 50

x2 = 174

...

7. 命令を取得



メモリ空間

# 計算実行例 (8)

CPU

x0 = 124

x1 = 50

x2 = 174

...

7. 命令を取得

8. メモリ書き込み

0x4000

0x4004

0x4008

read  
0x4000  
to x0

read  
0x4004  
to x1

x2 =  
x0 + x1

write x2  
to  
0x4008

124

50

174

メモリ空間

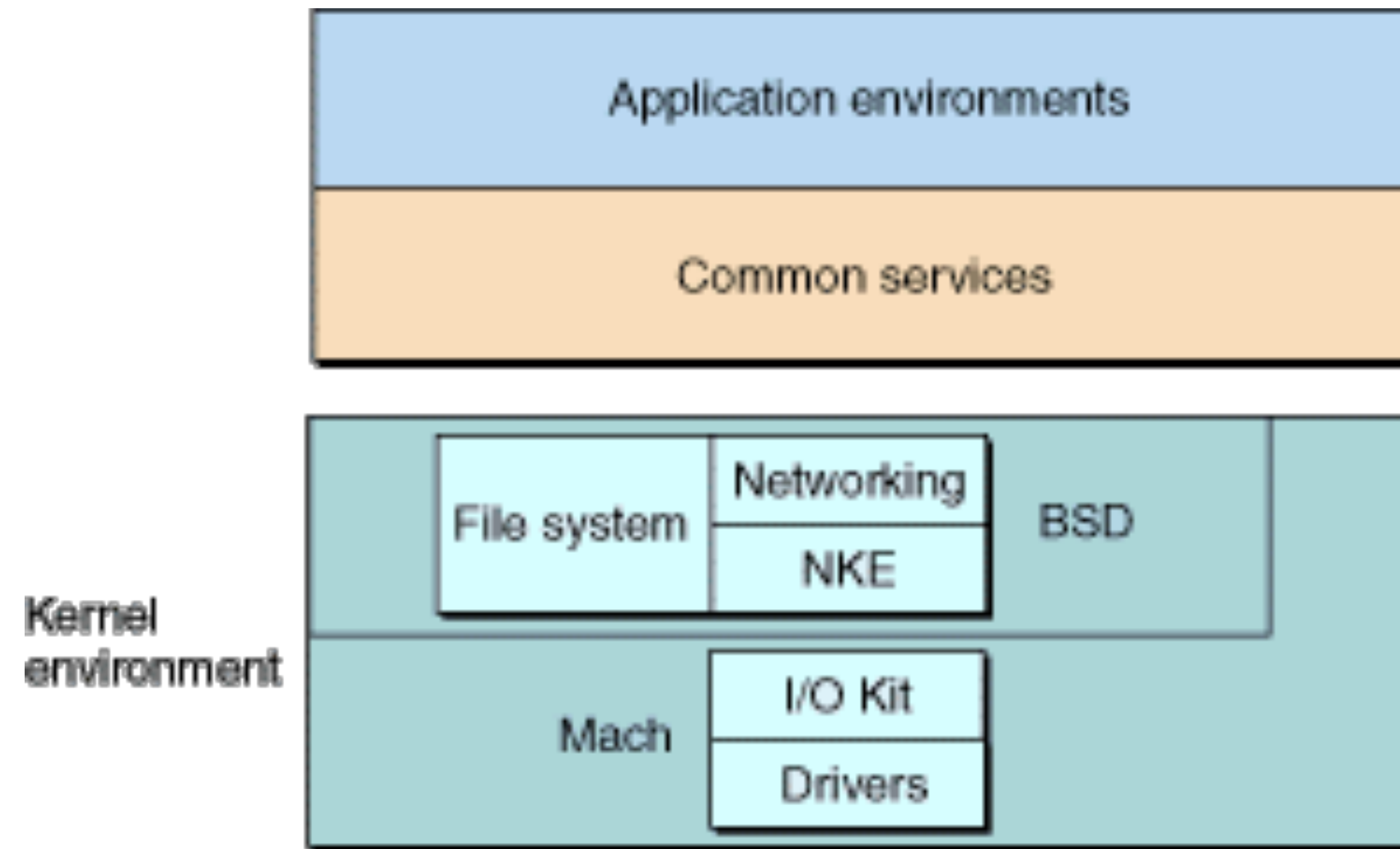
NextSTEP

# NextSTEPとは？

- iPhone、iPadなどのiOSの先祖
- 1985年にAppleコンピュータを追い出されたスティーブジョブスが、1986年にNext社を設立し、そこで作れたOS
- 1994年にオープン化される（OpenSTEPと呼ばれる）
- 1996年にNextはAppleに買収され、NextSTEPがMacOSの基盤に



# OS Xの構成



<https://developer.apple.com/library/archive/documentation/Darwin/Conceptual/KernelProgramming/Architecture/Architecture.html>

# いまでも動くOpenSTEP

