§2. INTRODUCTION

$$\underbrace{\text{program} \xrightarrow{\text{run}} \text{execute } \boxed{\text{instruction}} \quad ① \text{ fetch} \quad ② \text{ decode} \quad ③ \text{ execute}}_{\text{von Neumann.}}$$

2.1
- virtualizain

$$\boxed{\begin{array}{c} \text{physical} \longrightarrow \text{virtual} \\ \\ \text{resource} \end{array}}$$

- role of OS.
  - virtual machine
  - standard library API
  - resourse manger share $\left\{ \begin{array}{l} \text{cpu} \\ \text{memory} \\ \text{device} \end{array} \right.$

• 2.1 virtualize CPU

single CPU ⟶ seemingly infinite num

many progs. seemingly run @ once

↓

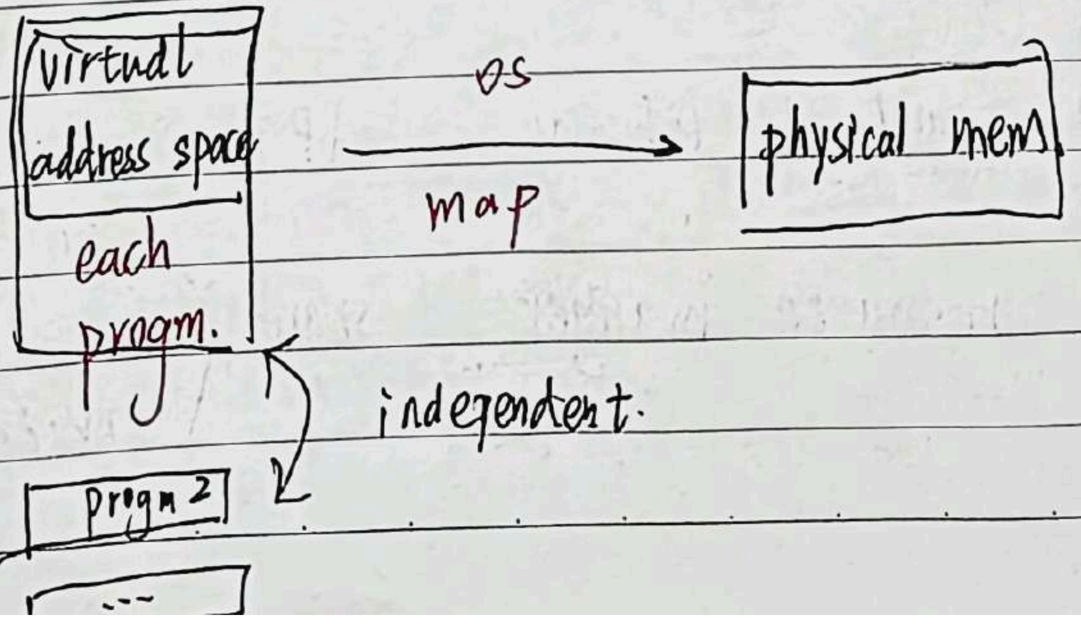
new questions ...

policy ... etc. mechaisms

↓

resource manager

• 2.2 virtualize memory.



• 2.3 concurrency

$$\begin{tikzcd}
\text{shared resource} & & \text{a thread} \arrow[ll, "\text{write}"']
\end{tikzcd}$$

not atomic } ① load to register ② update ③ store

$$\begin{tikzcd}
\text{not atomic} \arrow[d, "\text{context switch}"] \\
\text{(if high concurrency)} \arrow[d, "\text{thus loop many times.}"] \\
\text{error}
\end{tikzcd}$$

• 2.4 persistent

$$\begin{tikzcd}
\text{file system} \arrow[r] & \text{write file} \\
\text{call API} \arrow[u] &
\end{tikzcd}$$

① where on disk

② maintain metadata

③ I/O, device driver

call API
open()
write() close()

performance better → delay. $\boxed{\text{buffer}}$

⇑ large batch ← delay

buffer → if crash during writes

⇓

write protocol

↓

recover afterward

• 2.5 Design Goals

$\boxed{\text{virtualize concurrency presistent to do}}$

→ abstraction.

→ minimize the overheads

→ protection → isolation

→ $\boxed{\text{reliability}}$ (non stop)

→ energy - efficient security

mobility

• 2.6 Somo History

◦ Early OS.

library → API

batch processing (human operator)

◦ Protection

procedure call → system call

~~user mode → trap kernel mode~~

$$\begin{tikzcd}\text{user mode} \arrow[r, "\text{call}"] & \text{kernel mode} \arrow[r, "\text{trap}"] \arrow[l, bend left, "\text{return from trap}"] & \text{I O}\end{tikzcd}$$

o Multi Programming

$$\begin{tikzcd} \text{mini computer (} & \text{shared} \arrow[dl] \arrow[dr] & \text{by workgroup)} \\ \text{memory protect} & & \text{concurrency} \end{tikzcd}$$

o Modern Era.

personal computer.

$$\text{DOS. MacOS} \rightarrow \text{windows NT}$$

lessons in mini computer