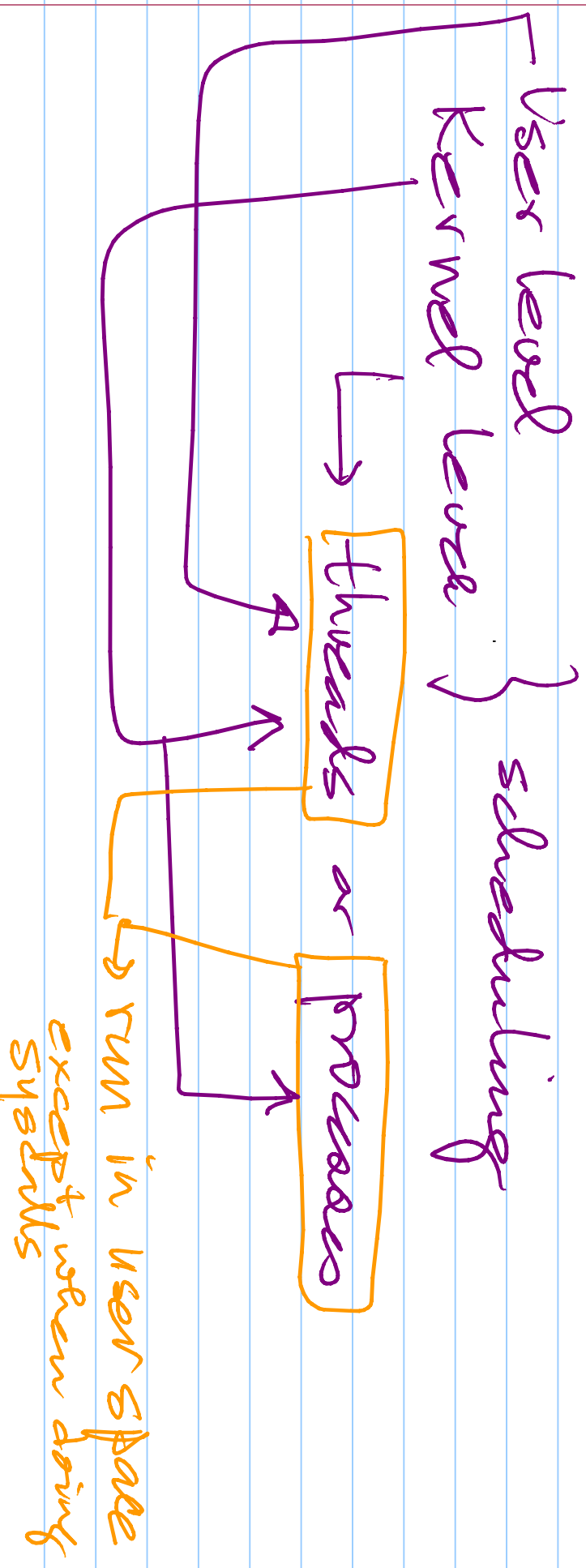




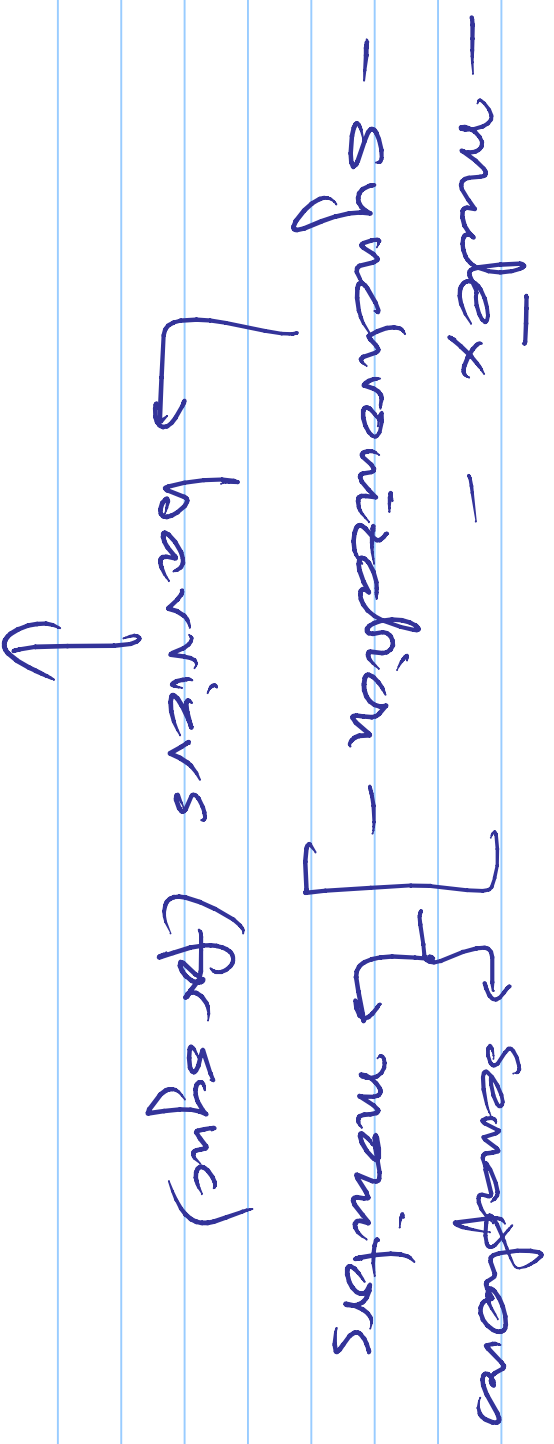
user threads

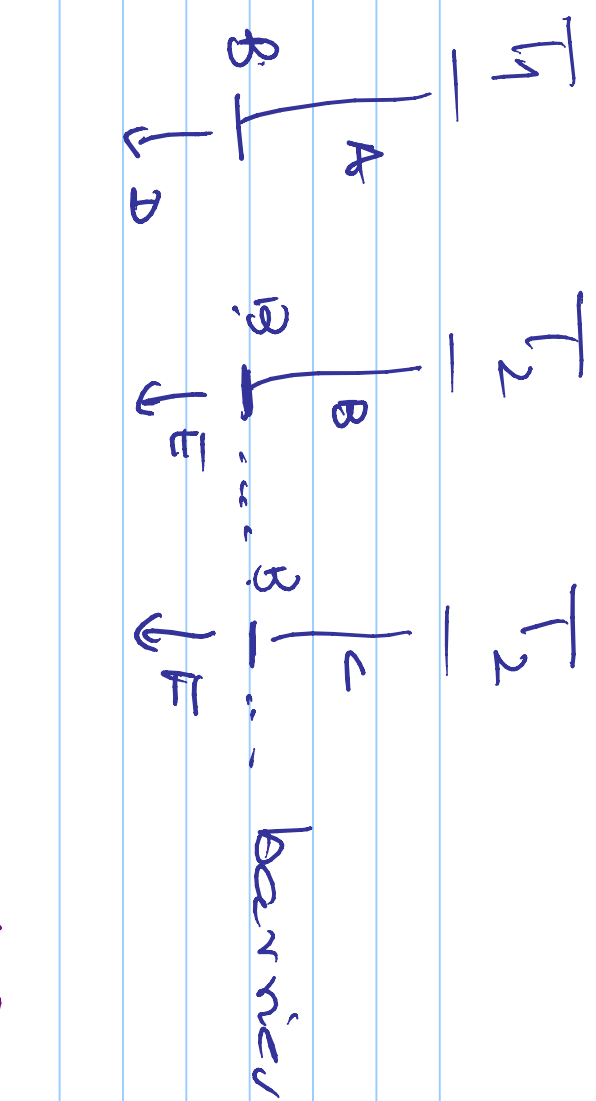
kernel threads

— [pthread, windows threads

→ a thread running in
the kernel

threads need

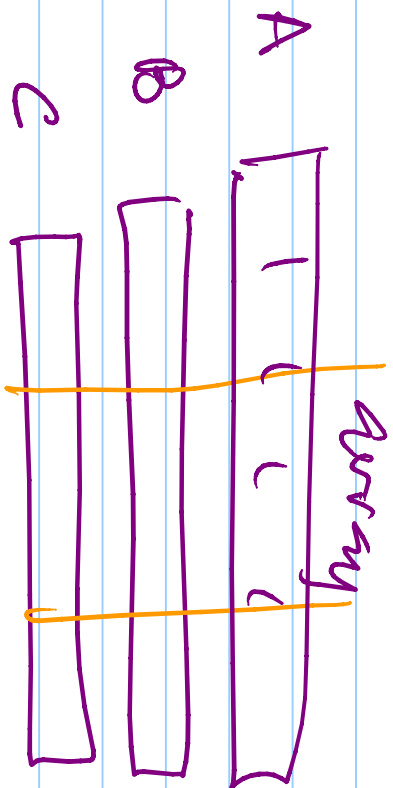




A and B and C leave to complete
before any 1 or more of D, E, F start

Parallel programs

- Split a task into several subtasks, all almost identical



$T_1 \rightarrow \text{does } 1^{\text{st}} \frac{1}{3}$
 $T_2 \rightarrow \text{'' } 2^{\text{nd}} \frac{1}{3}$
 $T_3 \rightarrow \text{'' } 3^{\text{rd}} \frac{1}{3}$

$$C = A + B$$

→ D = something C & another Array

3 parts each

$$C = A + B$$

$$D = m$$

$T_1 T_2 T_3$

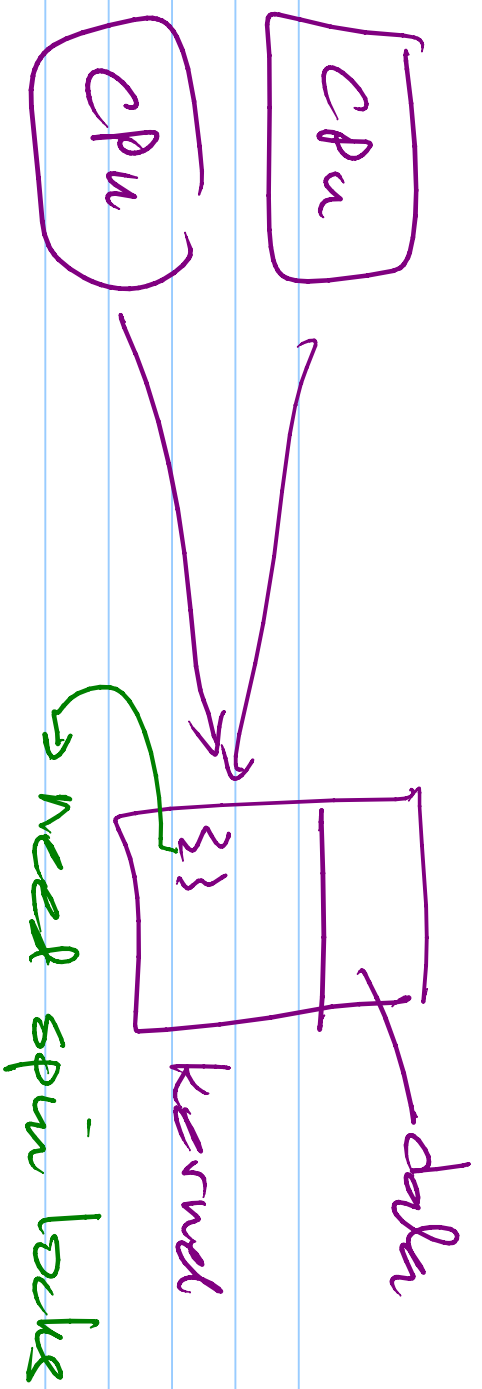
barrier

Multi-core OS

- Symmetric kernels

→ most common

↓
all CPUs run the OS (& same OS) & 1 copy (shared) of kernel exists in memory →



- each cpa independently does Scheduling & I/O

Master Worker Kernel (old)

- ↳ only 1 CPU has access to kernel (master) *runs a process*
- all other CPUs send requests to master for performing syscalls

Multi kernel (new)

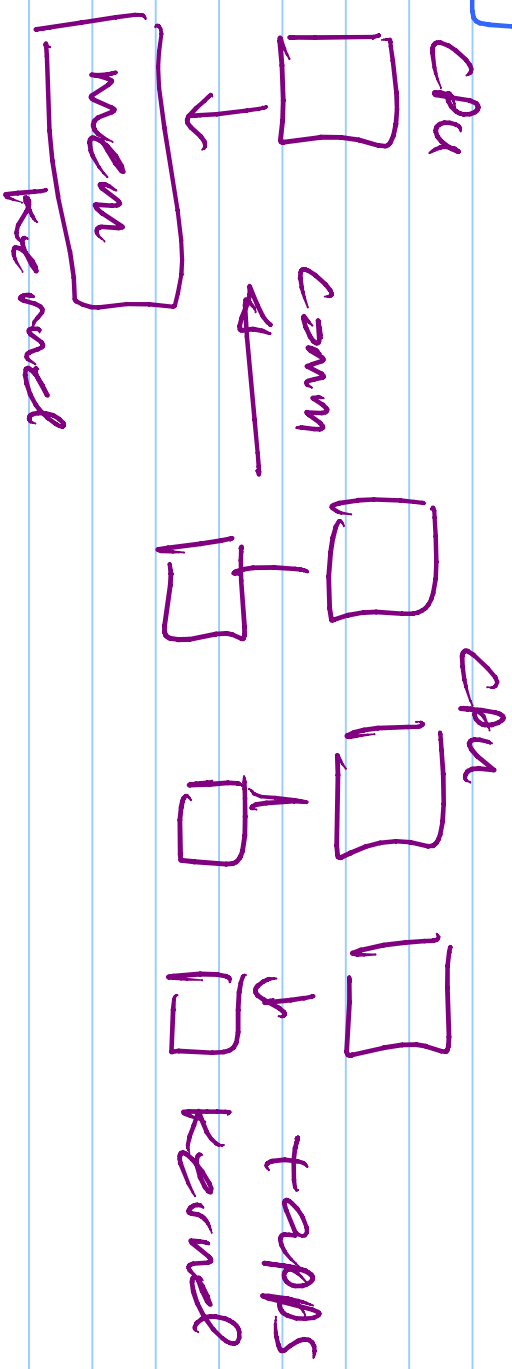
↳ a variety of architectures →

→ one processor runs the OS

→ other processors run

smaller kernels

BARREL FISH



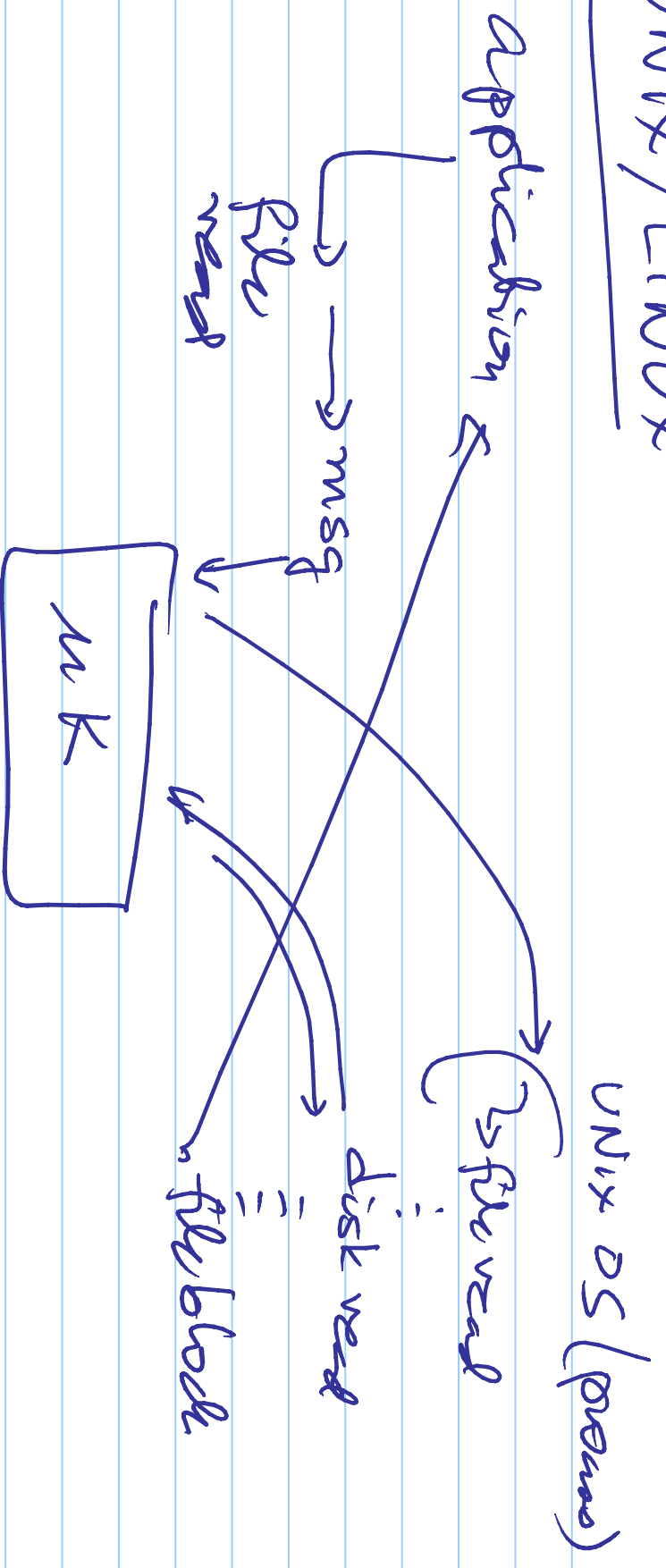
microkernel (NORMA / DISTRIBUTED)

- the OS is not in the kernel
- the kernel is "small" and handles only a few operations
- all other functionality is done by 1 (or more) "services"
→ SERVERS

kernel in a microkernel

- scheduling (context switch)
- memory allocation
- basic I/O
- message passing - interpreter communication

UNIX/Linux



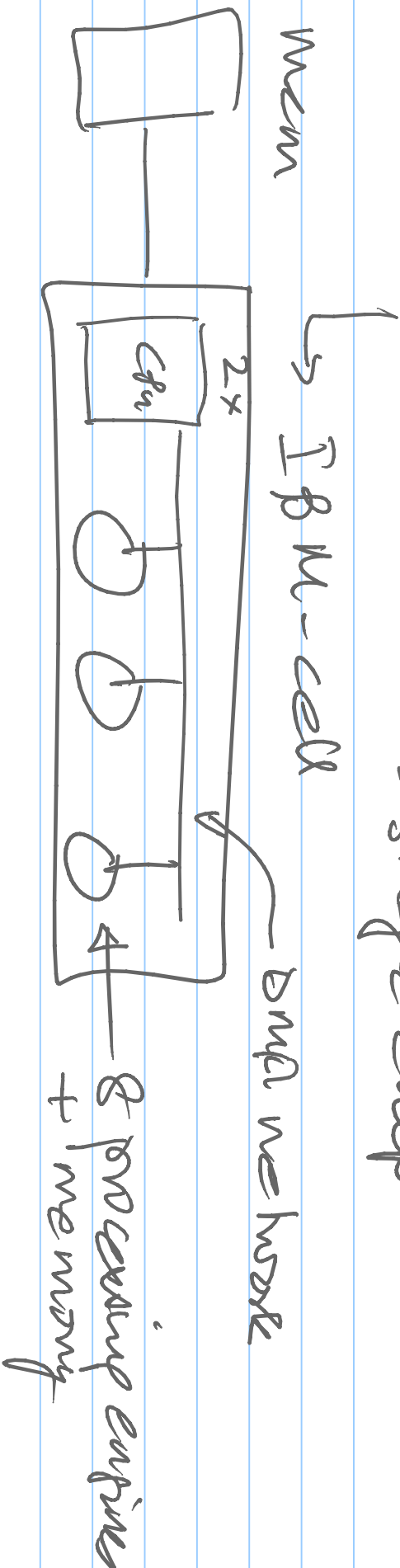
UNIX - LINUX - (macos) kernel
(symmetric)

MACH → Apple OSX → microkernel

windows - microkernel

Multicore architectures

- symmetric, shared memory
- distributed memory multicores
→ single chip



Intel-SCC - Single Chip Supercomputer

↳ 40-80 cores

all have separate memory

↳ large set of components

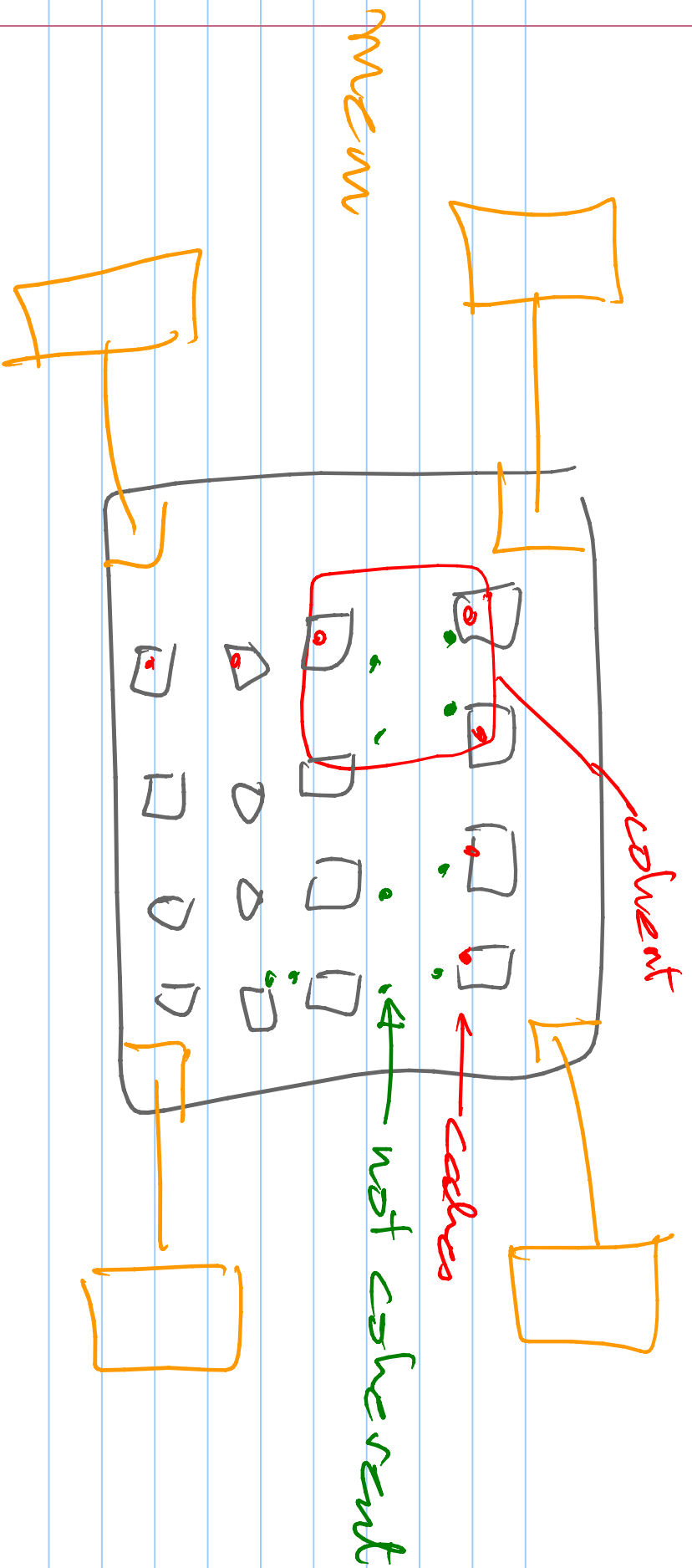
Intel - XEON - P41

- ↳ large # of processors

- ↳ 4 memory units

- ↳ some coherent caches

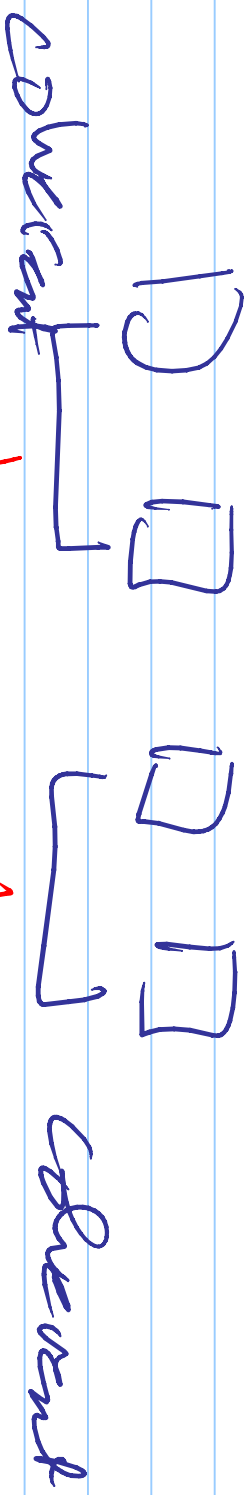
- ↳ some non-coherent caches



CC-NUMA

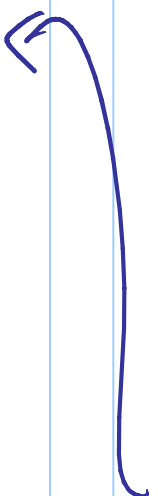
cache coherent

non-uniform ... numa



low level - pthreads

high level - languages



- OpenMP

- MPI

- Linda

OpenMP

- language → compatible with C/C++
- shared memory parallelism
- supports threads, barriers
 - can construct