

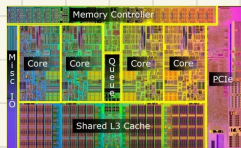
9/27/23

Welcome to 451: Intro to Operating Systems

→ OS: a program that abstracts hardware resources for other programs to use

→ HW Resources

→ CPU

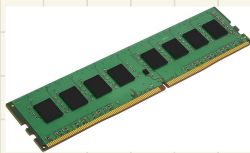


abstract into?

⇒ processes / threads

* single core CPU only in this class

→ memory

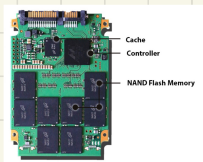


volatile, byte addressable

⇒ virtual memory "infinite" array of addrs → value

→ I/O devices

→ storage



⇒ files / filesystems

→ can also be implemented on top of network (NFS) & memory

→ Network Adapter



⇒ network stack
(ports, sockets, packets, protocols)

→ mouse, keyboard, speakers, microphone

→ HW resources ⇒ software abstractions

What do we gain from ↑ ?

→ ease of use. *illusionist*

→ portable interface (programs independent from HW) *glue*

→ managed access (resource sharing & management) *referee*

→ common interface (enables sharing & communication) *glue*

How does OS provide these abstractions?

★ What we will cover in class!

Our model.

Single computer, single OS, single core/CPU. common abstractions

Beyond
ASI

- more complex hardware
 - multicore, heterogeneous
 - different hw resources
- more complex software
 - containers, virtual machines
- different abstractions
 - designed for specific workloads