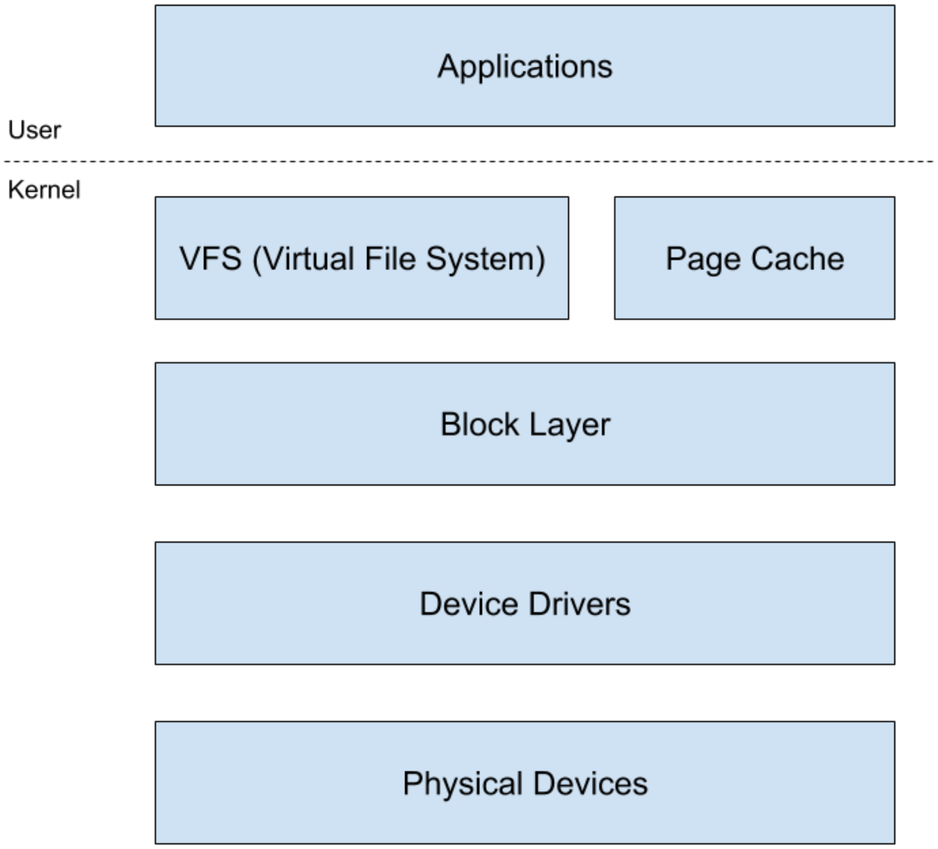
Course Overview :: Meeting 0

Learning Objectives

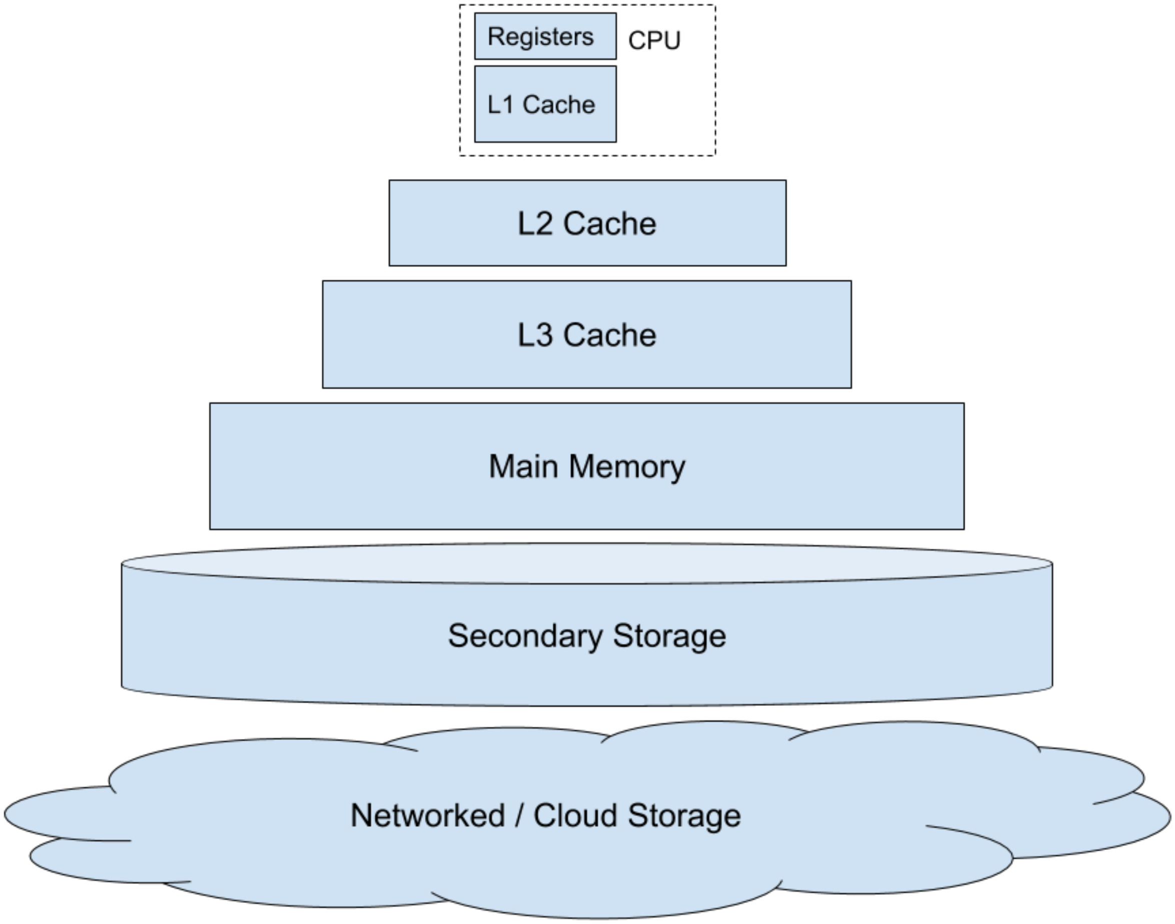
- Holistic understanding of storage landscape
- Start to think about interesting aspects of the course

Overview of the Software Storage Stack (Unix/Linux)



- What are the interfaces at the layer boundaries?
- What are the requirements of each layer (flexible allocation sizes, strict ordering of writes, recovery if power failure, persistence, encryption, etc.)?
- What privileges are required to implement these details (user vs. root permissions, un-changeable hardware)?
- What are the costs of changing the details at that layer (rewrite one app, rewrite all apps, change manufacturing processes)?

Overview of the Memory Hierarchy



- What are (rough) sizes at each level?
- What are (rough) speeds at each level?
- What are the costs ($/GiB) at each level?
    - Are there different choices at any level (e.g., HDD vs. SSD vs. Tape vs. CD-ROM)?
- What is the unit of transfer at each level (byte, word, page, large object)?
- Think about the software storage stack diagram from before. How do the different levels of the cache hierarchy play into each the design of each layer of the software storage stack (persistence, transfer size, interface, etc.)?
- Reminder from CS237: Cache hierarchy purpose and assumptions: locality locality locality. The goal of the memory hierarchy is to present the performance profile of fast expensive memory with the cost and capacity of slow cheap memory.