CS 423 Operating System Design: Final Exam Overview

Xuhao Luo

“Take-Home” Final

- **12/11 (Sun), 9:00am –12/11, 9:00pm**

  You will have **12 hours**

  No late submission

- **Release:**

  A PDF will be posted on Piazza at 9:00 am, 12/11.

  Don’t stay all night (it should only take a few hours).

- **Submission**:

  A Google Form link will be posted on Piazza

  Form submission will be closed at 9:00 pm 12/11

Final Exam Policies

- **Open book and Internet**: Textbooks, paper notes, printed sheets allowed.
  - We don’t have any control any way.
- **Discussion are not allowed.**
  - We don’t have control but we will know if your answer is the same or similar.
  - **Consequence is severe, so don’t do it.**

Final Format

- **Four or five Open-ended Problems (each with multiple sub-problems)**
  - Present you an OS problem and the design of a potential solution (e.g., a file system)
    - Ask you to explain the design (whether it is a solution or not; what are the potential issues)
    - Ask you to implement the design
    - Ask you to design solutions for a few potential issues in the original design.

Final Exam Content

- Everyone we have discussed in the class, **including materials before the midterm.**
  - Kernel Abstraction
  - Process and Threat Management
  - Synchronization
  - Scheduling
  - Memory Management
  - Virtualization
  - Storage and File System
  - Distributed Systems
  - Reliability and Security

Content before Midterm

- Please refer to
- https://cs423-uiuc.github.io/fall22/slides/final-overview.pdf

Memory Management

- How virtual memory works?
  - What is a TLB?
  - What is a page table?
  - What is a translation cache?
  - How is address translation done?
  - How to handle a page fault?
  - How to handle a TLB miss?
  - How to support huge pages?
  - What is the page replacement algorithm used in Linux and how does it work?

Virtualization

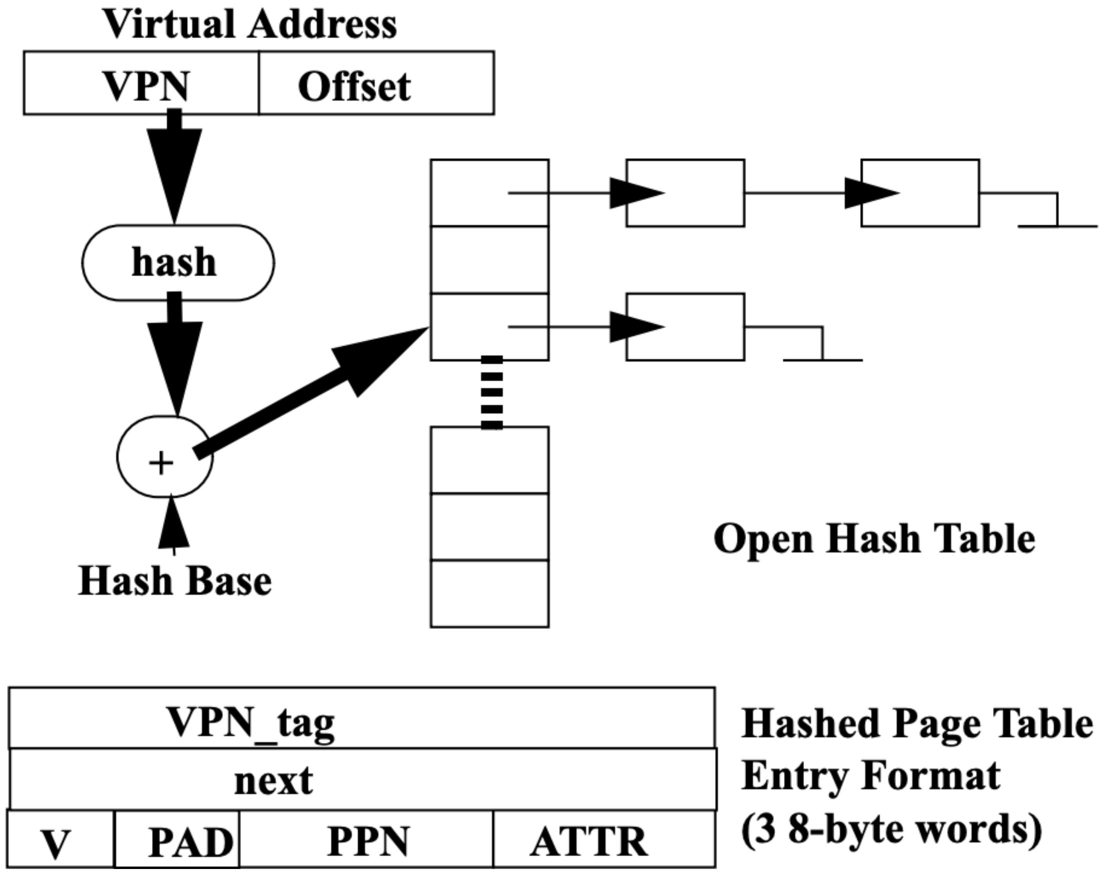
- What is the difference between “hypervisor” versus “containers”?
- How do hypervisors virtualize CPUs?
- How do hypervisors virtualize memory?
- What is the difference between full-virtualization versus para-virtualization?
  - What are the design tradeoffs?
  - How does it work?
- How to implement a container?
- What are the security mechanisms for containers and how does each of them work?

Storage and File Systems

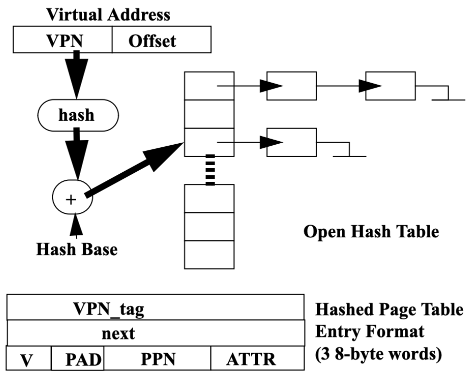
- How does a disk work? What are the main bottlenecks?
- What are different disk scheduling algorithms and what are the tradeoffs?
- How is the file abstraction implemented?
  - What are inodes?
  - What are directories?
  - What are the disk layout for different filesystems?
- How does FAT work?
- How does FFS work?
  - What are the key optimizations to be “fast”?
- How does LFS work?
- How does GFS work?

Example Problem (10 pts)

- An alternative design to multi-level radix page table discussed in class is hashed page table (HPT). HPT is supported by a few architecture such as Intel Itanium and IBM Power. A HPT is illustrated as follows.

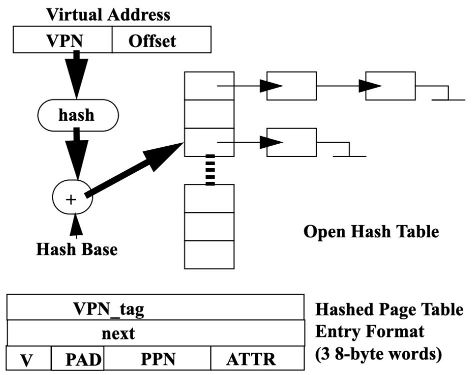


In HPT, the Virtual Page Number (VPN) is hashed into a key. The key can be used (together with the Base) to index the HPT to find the PPN (Physical Page Number).



Example Problem (10

1. Please describe how a TLB miss is handled by a virtual memory system using HPT? We assume a perfect hash in which there is no collision. (3pt)
2. No hash function is perfect – hash collisions are inevitable, which means two different VPNs will be hashed into the same index pointing to the same PPN. How to address collisions? (2pt)
3. Please integrate your hash collision resolution in your answer for #1 (TLB miss handling). (2pt)
4. Please annotate in your answer for #3 which step is done by MMU and which is done by the kernel? (1pt)



Example Problem (10

5. In your answer for #4, when the kernel needs to be involved, please describe the process in detail, e.g., how to notify the kernel and what should the kernel do? (1pt)

6. One problem of HPT is the loss of locality, compared with radix page table, because hashing breaks the spatial locality which impairs prefetching. What is a potential solution or a workaround? (1pt)