Virtual Memory

15-213 / 18-213: Introduction to Computer Systems
10th Recitation, March 24th, 2014

Ankur Sharma
Section C
Today

- Management and Stuff
- Shell Lab
- Malloc Lab
- Git primer
- Virtual Memory
- TLB
Management and Stuff

- Shell Lab due Thursday, March 27th 2014, 11:59 PM
- Malloc Lab out Thursday, March 27th 2014, 11:59 PM
Today

- Management and Stuff
- Shell Lab
- Malloc Lab
- Git primer
- Virtual Memory
- TLB
FAQs

Adding sigsuspend breaks my code (was working fine with a while loop)?

- Read the documentation for sigsuspend again, especially the example.
FAQs

- **trace_xx passes intermittently?**
  - You still have race conditions in your code.
  - Reproduce the failed scenario and compare with reference shell.
  - Understand the trace file.
FAQs

- time out error?
  - Driver says that it timed out while waiting for shell prompt.
  - Most likely the fg job was not reaped.
Today

- Management and Stuff
- Shell Lab
- Malloc Lab
- Git primer
- Virtual Memory
- TLB
Malloc Lab

- Out March 27th
- Due April 15th
- Start early
- Ask questions
Today

- Management and Stuff
- Shell Lab
- Malloc Lab
- Git primer
- Virtual Memory
- TLB
Git primer

- Afraid of losing files but too confused/lazy to learn Git and set up an account?
- Make a local repository
  - No account required
  - `cd tshlab-handout`
    - `git init`
    - `git add (files)`
    - `git commit`
Git primer

- **git init**
  - Creates a git repository.
  - Directory named `.git` will be created.

- **git status**
  - Shows the status of repository.

- **git add file_name**
  - Stages file for commit.
  - `git add .`
    - Stages all the files in current directory for commit.
Git primer

- **git commit**
  - Commits the files to repository.
  - `git commit -m "commit_Msg"`

- **git push**
  - Pushes the local repository to remote location.

- **git clone**
  - Copies a remote repository
    - `git clone git://github.com/path/file_name.git`

- **git pull**
  - Merges remote repository with local.
Git primer

- **Remote repositories**
  - Github, Bitbucket
  - **Beware**, do not make your repository public.
  - Public repositories are indexed by google.

- **Help**
  - [http://www.contrib.andrew.cmu.edu/~cakrivou/98174/](http://www.contrib.andrew.cmu.edu/~cakrivou/98174/)
Today

- Management and Stuff
- Shell Lab
- Malloc Lab
- Git primer
- Virtual Memory
- TLB
Virtual Memory Abstraction

Virtual memory is layer of indirection between processor and physical memory providing:

- Caching
  - Memory treated as cache for much larger disk
- Memory management
  - Uniform address space eases allocation, linking, and loading
- Memory protection
  - Prevent processes from interfering with each other by setting permission bits
Virtual Memory Implementation

- Virtual memory implemented by combination of hardware and software
  - Operating system creates page tables
    - Page table is array of Page Table Entries (PTEs) that map virtual pages to physical pages
  - Hardware Memory Management Unit (MMU) performs address translation
Address Translation and Lookup

On memory access (e.g., mov 0xdeadbeef, %eax)

- CPU sends virtual address to MMU
- MMU uses virtual address to index into in-memory page tables
- Cache/memory returns PTE to MMU
- MMU constructs physical address and sends to mem/cache
- Cache/memory returns requested data word to CPU
Recall: Address Translation With a Page Table

Virtual address

Virtual page number (VPN)  Virtual page offset (VPO)

Page table

Valid  Physical page number (PPN)

Valid bit = 0: page not in memory (page fault)

Page table address for process

Physical address

Physical page number (PPN)  Physical page offset (PPO)
Translating with a k-level Page Table
x86 Example Setup

- Page size 4KB (2^12 Bytes)
- Addresses: 32 bits (12 bit VPO, 20 bit VPN)
- Consider a one-level page table with:
  - Base address: 0x01000000
  - 4-byte PTEs
    - 4KB aligned (i.e., lowest 12 bits are zero)
    - Lowest 3 bits used as permissions
      - Bit 0: Present?
      - Bit 1: Writeable?
      - Bit 2: UserAccessible?
- How big overall?
  - 2^20 entries
Example

- Given the setup from the previous slide, what are the VPN (index), PPO, and VPO of address: 0xdeadbeef?
Example

- **Answers:**
  - VPN (index) = 0xdeadb (1101 1110 1010 1101 1011)
  - VPO = PPO = 0xeef

- **Consider a page table entry in our example PT:**
  - Location of PTE = base + (size * index)
    - 0x0137ab6c = base + 4 * index
  - PTE: 0x98765007
  - Physical address: 0x98765eef
Example: 2 level page table

Use the first VPN to index into the page directory. This gives the address of the start of the page table.
Use the second VPN to index into the page table. This gives the address of the start of the page frame. Add the offset to obtain the location in physical memory.
Today

- Management and Stuff
- Shell Lab
- Malloc Lab
- Git primer
- Virtual Memory
- TLB
TLB

- TLB (Translation Lookaside Buffer) is cache for page table entries.
  - Each PTE lookup leads to a memory access.
  - TLB caches the Page Table entries.
TLB (HIT)

1 VA

2 VPN

3 PTE

4 PA

PROCESSOR

TLB

TRANSLATION

CACHE/MEMORY
TLB (MISS)

1. VA

2. VPN

3. PTEA

4. PTE

5. PA

PROCESSOR  \rightarrow  TRANSLATION  \rightarrow  CACHE/MEMORY
Questions?