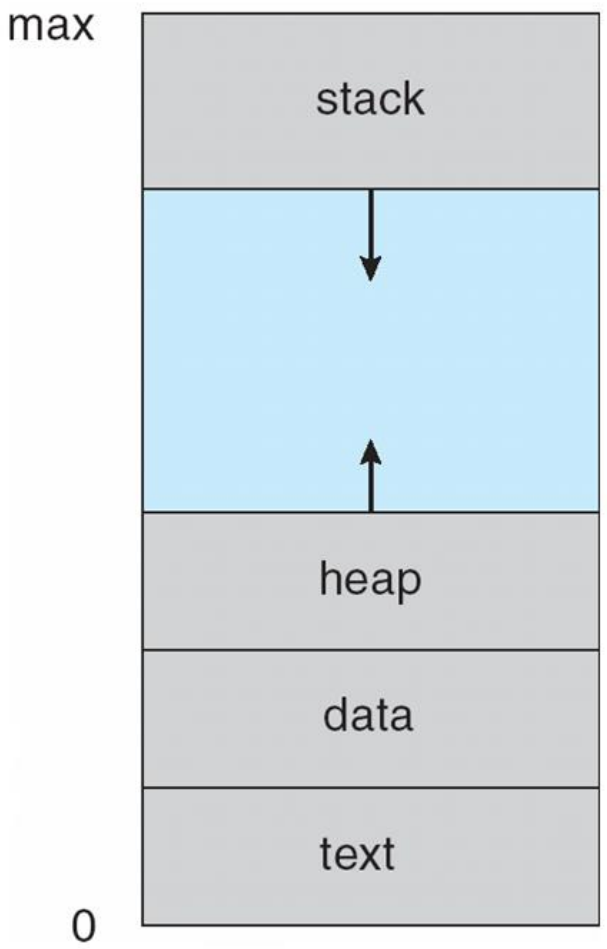
# Final Review

# 2nd Half

- Memory management
- Disk management
- Network and Security
- Virtual machine

# Virtual Memory (VM)

- Abstraction
  - 4GB (32bit) linear address space for each process
- Reality
  - 1GB of actual physical memory shared with 20 other processes



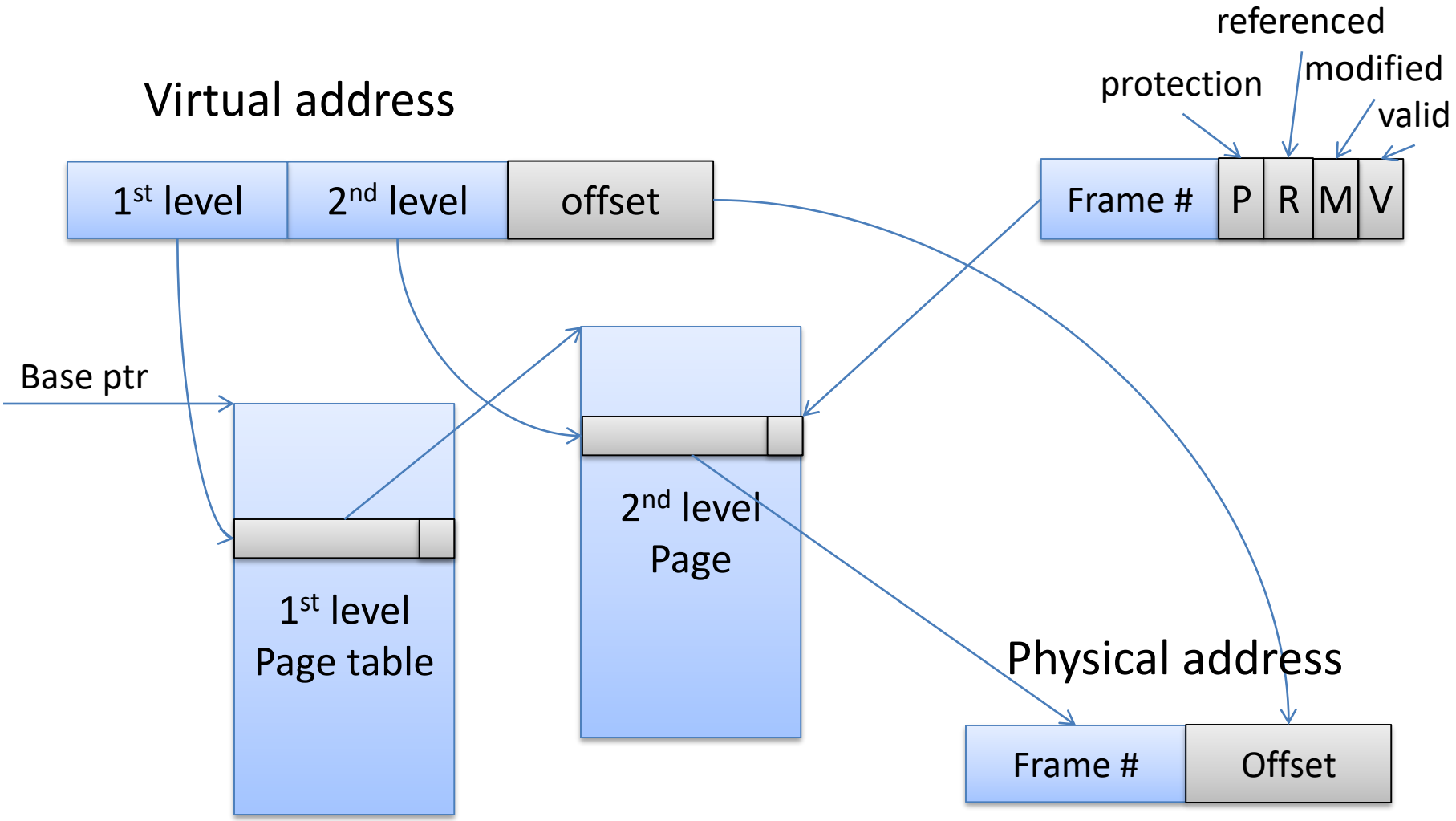
# Paging

- MMU (memory management unit)
  - Segmentation (base+limit),
  - Paging (page table)
  - TLB (translation lookaside buffer)
- Page table
  - Virtual addr → physical address translation table
- Other concepts to know
  - Fragmentation

https://blog.csdn.net/xieyihua1994/article/details/52868872
区分虚拟地址和物理地址
page table 提前被初始化存在物理内存
MMU是硬件，不需要额外花时间进行地址转化
利用nm + 执行文件名可以看到虚拟地址

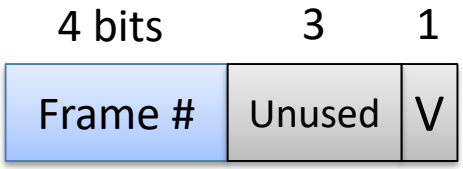
相同实行机制：base+limit
将程序发出的虚拟地址加上基址得到物理地址
https://blog.csdn.net/qq_39755395/article/details/78380942

# Two Level Address Translation



# Quiz: Address Translation


Virtual address format (24bits)

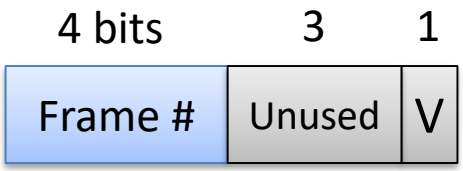



Page table entry (8bit)

- What is the size of a single page?
  - 256 bytes
- What is the size of the virtual address space?
  - 2^24 = 16MB
- What is the size of the physical address space?
  - 2^12 = 4KB

# Quiz: Address Translation


Virtual address format (24bits)




Page table entry (8bit)

Vaddr: 0x0703FE
Paddr: 0x3FE

Vaddr: 0x072370
Paddr: ???

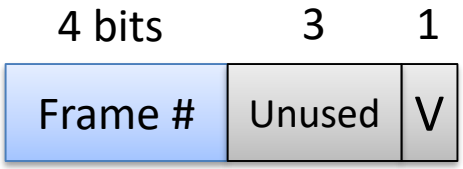
Vaddr: 0x082370
Paddr: ???

Page-table base address = 0x100

| Addr | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +A | +B | +C | +D | +E | +F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x000 | | | | 31 | | | | | | | | | | | |
| 0x010 | | | | | | | | | | | | | | | |
| 0x020 | | | | 41 | | | | | | | | | | | |
| .. | | | | | | | | | | | | | | | |
| 0x100 | 00 | 01 | | | | | | 01 | 00 | | | 01 | | | |
| .. | | | | | | | | | | | | | | | |
| 0x200 | | | | | | | | | | | | | | | |

# Quiz: Address Translation


Virtual address format (24bits)




Page table entry (8bit)

Vaddr: 0x0703FE
Paddr: 0x3FE

Vaddr: 0x072370
Paddr: **0x470**

Vaddr: 0x082370
Paddr: **invalid**

**Page-table base address = 0x100**

| Addr | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +A | +B | +C | +D | +E | +F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x000 | | | | 31 | | | | | | | | | | | |
| 0x010 | | | | | | | | | | | | | | | |
| 0x020 | | | | 41 | | | | | | | | | | | |
| .. | | | | | | | | | | | | | | | |
| 0x100 | 00 | 01 | | | | | | 01 | 00 | | | 01 | | | |
| .. | | | | | | | | | | | | | | | |
| 0x200 | | | | | | | | | | | | | | | |

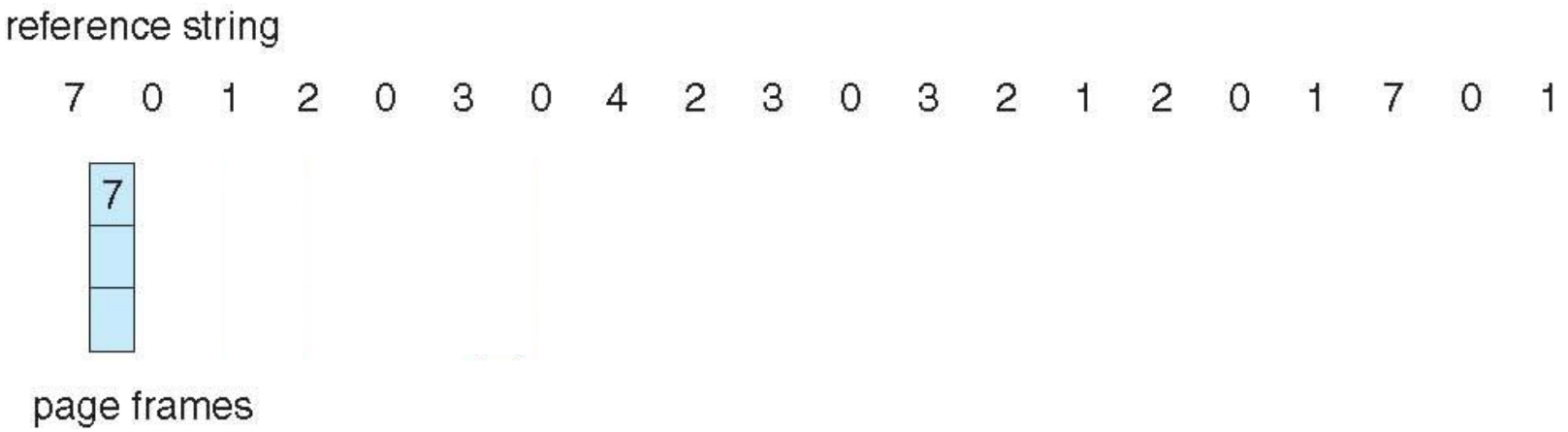
# Demand Paging

- Key idea
  - no need to load all pages on memory.
- Page fault
  - Occurs when there's no valid virt -> phys addr translation
  - Allocate a free frame [and load a page from the disk]
- Copy-on-write
  - Fork()/exec() semantic
- Other concepts to know
  - Page Table Entry (PTE) bits: valid/invald, …

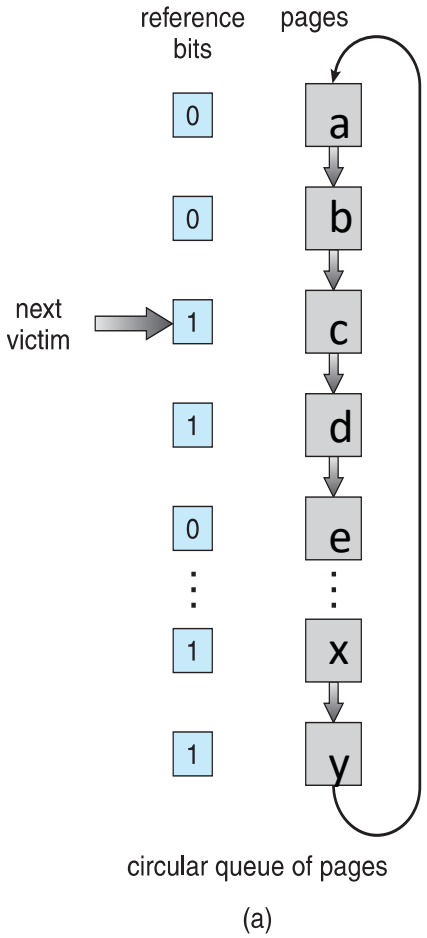
# Page Replacement & Swapping

- Least Recently Used (LRU)
  - Approximation: clock algorithm, N-clock algorithm
  - When swapping occurs in the page fault handler?
- Other concepts to know
  - Thrashing
  - FIFO, MIN (optimal) replacement algorithms

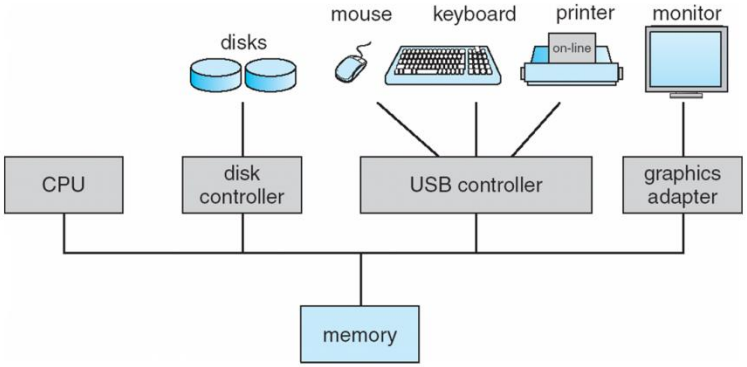
# Exercise: LRU Replacement



# Second Chance Algorithm



# I/O Devices




- Block devices
  - E.g., disk, cd-rom, USB stick
  - High speed, block (sector) level accesses
- Character devices
  - E.g., keyboard, mouse, joystick
  - Low speed, character level accesses
- Network devices
  - E.g., ethernet, wifi, bluetooth
  - Socket interface

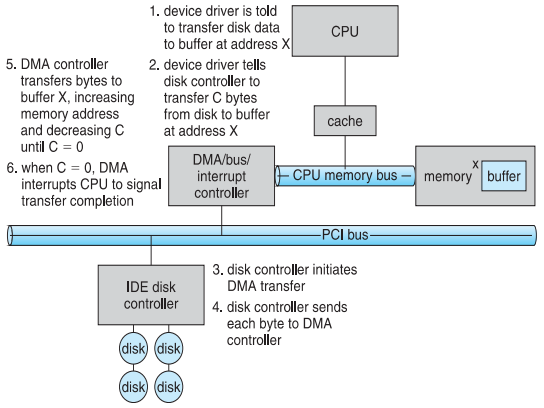
# Disk

- Magnetic disk
  - Key characteristic: long seek time
- Solid state disk (SSD)
  - Key characteristic: zero seek time
- Performance metrics
  - Access latency, price/storage space

# I/O Mechanisms

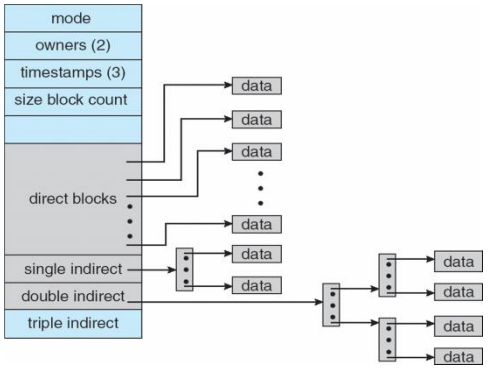
- Memory-mapped I/O vs. I/O instructions
  - MMIO: Address space → registers in the hardware
  - I/o instructions: *inb* or *outb* in x86

- Programmed I/O (PIO) vs. DMA
  - PIO: Reads/writes by CPU
  - DMA: DMA hw operation steps



# Filesystem

- Disk block allocation methods
  - Continuous allocation
  - **Linked allocation**
  - **Indexed allocation**
- Other concepts to know
  - Metadata, inode, directory
  - **Direct/single/double/triple indirect blocks**
    - How they differs. Why different levels?



# Quiz

- Consider an ext2 like file system. Each block in the file system is **2048** bytes and the size of each block pointer is **32bit**. Each *inode* has **10 direct pointers**, **one indirect pointer**, and **one doubly-indirect pointer**.
  - What is the maximum disk size that this filesystem can support?
    - 2^32 * 2048 = 8TB
  - What is the maximum file size of a file?
    - 10 * 2K + 2048 / 4 * 2K + (2048/4) ^2 * 2K

# Name Resolution

- How many disk accesses to resolve "/usr/bin/top"?
  - Read "/" directory inode
  - Read first data block of "/" and search "usr"
  - Read "usr" directory inode
  - Read first data block of "usr" and search "bin"
  - Read "bin" directory inode
  - Read first block of "bin" and search "top"
  - Read "top" file inode
  - Total 7 disk reads!!!
    - This is the minimum. Why? Hint: imagine 10000 entries in each directory

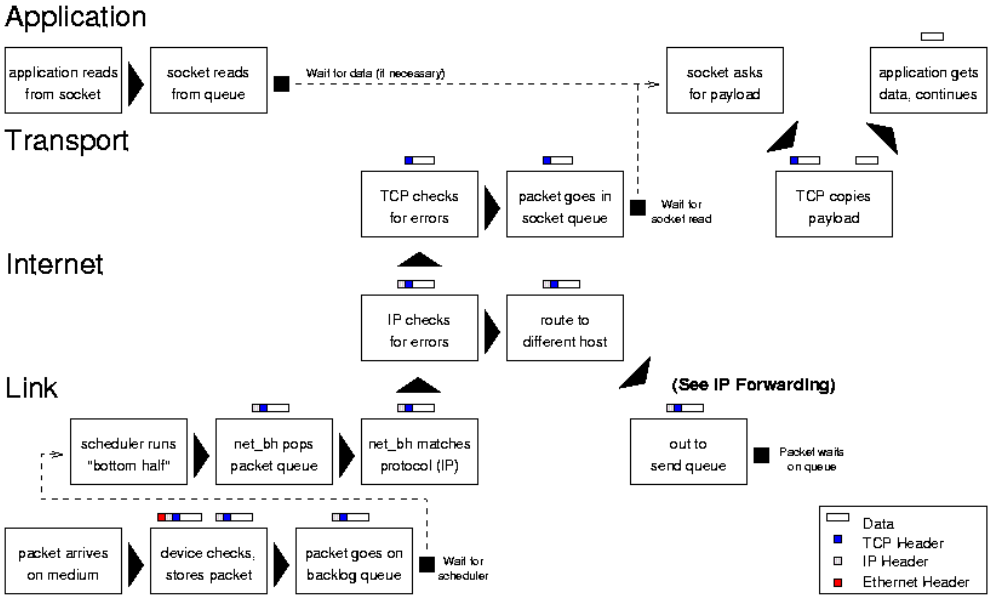
# Storage caches

- Directory cache
  - For faster name resolution
  - Steps to resolve a file name to inode
- Buffer cache
  - Frequently used disk blocks

- Other concepts to know
  - Unified buffer and page cache
  - Journaling

# Network

- OSI 7 layers vs. TCP/IP 5 layers
  - Layering: pros and cons.
- Steps for sending/receiving a packet



# Security & Virtual Machine

- Buffer overflow bugs
  - Look at the example code
- Virtual machine monitor (VMM)
- Native VMM vs. hosted VMM
- Full virtualization vs. para virtualization

# 1st Half

- OS Structure
- Process
- IPC
- Thread
- Synchronization
- Deadlock
- CPU Scheduling

# OS Structure

- User mode/ kernel mode
  - Memory protection, privileged instructions
- System call
  - Definition, examples, how it works?

- Other concepts to know
  - Monolithic kernel vs. Micro kernel

# Process/Thread

- Address space layout
  - Code, data, heap, stack
- Process states
  - new, ready, running, waiting, terminated

- Other concepts to know
  - Process Control Block
  - Context switch
  - Zombie, Orphan
  - Communication overheads of processes vs. threads

# Synchronization

- Spinlock
  - Semantics of test & set, compare & swap
  - Implementation using compare & swap
- Blocking synchronization primitives
  - Busy waiting vs. blocking
  - Mutex, semaphore, monitor
- Example problems
  - Bounded buffer,  read/writer

# Deadlock

- Deadlock conditions
  - mutual exclusion, no preemption, hold and wait, circular wait
- Deadlock prevention
- **Banker's algorithm**

- Other concepts to know
  - Starvation vs. deadlock

# Scheduling

- Three main schedulers
  - FCFS, SJF/SRTF, RR
  - Gant chart examples

- Other concepts to know
  - Fair scheduling (CFS)
  - Fixed priority scheduling
  - Multi-level queue scheduling
  - Load balancing and multicore scheduling

Thank you!