Introduction to Operating Systems
CS 1550

Spring 2023

Sherif Khattab

ksm73@pitt.edu

(Some slides are from **Silberschatz, Galvin and Gagne ©2013)**

Announcements

- Upcoming deadlines
    - **All deadlines moved to Monday May 1st at 11:59 pm**
        - But please don't wait to last minute!
    - Homework 11, 12, Bonus Homework
    - Lab 4 and Lab 5
    - Quiz 3 and Quiz 4
    - Project 4 **(no late deadline)**

Final Exam

- **Wednesday 4/26 8:00-9:50**
  - same classroom
  - coffee will be served!
- Same format as midterm
- **Non-cumulative**
- Study guide and practice test on Canvas
- **Review Session** during Finals' Week
  - Date and time TBD
  - recorded

Bonus Opportunities

- **Bonus Homework**
  - worth up to 1%
  - lowest two homework assignments still dropped
- **Post-Course Quiz on Canvas**
  - worth 1%
- bonus point for class when

  OMETs response rate >= 80%
  - Currently at 28%
  - Deadline is Sunday 4/23

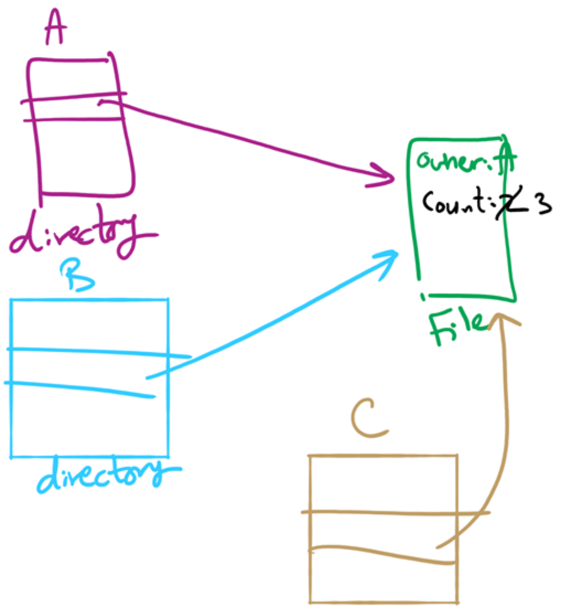
Previous Lecture …

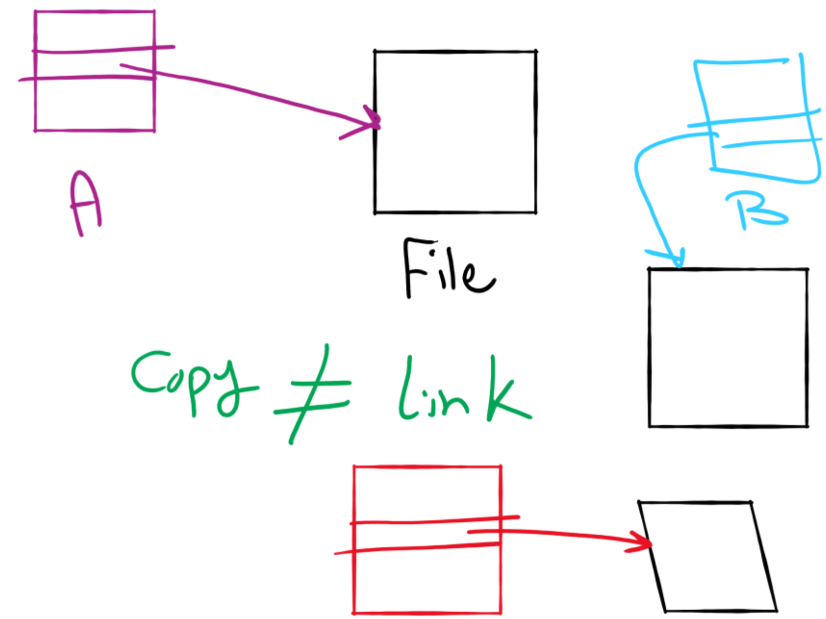
- How does a file system hide disk access delays?
- How do device drivers program I/O devices?
- Disk arm scheduling
  - FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK

Today …

- Miscellaneous issues in File Systems
- Protection in operating systems

Hard Linking vs. Copying





hard linking

- doesn't create a new i-node
- increments the **link count** inside original i-node
- **e.g., ln** command in Linux **without -s**

soft (symbolic linking)

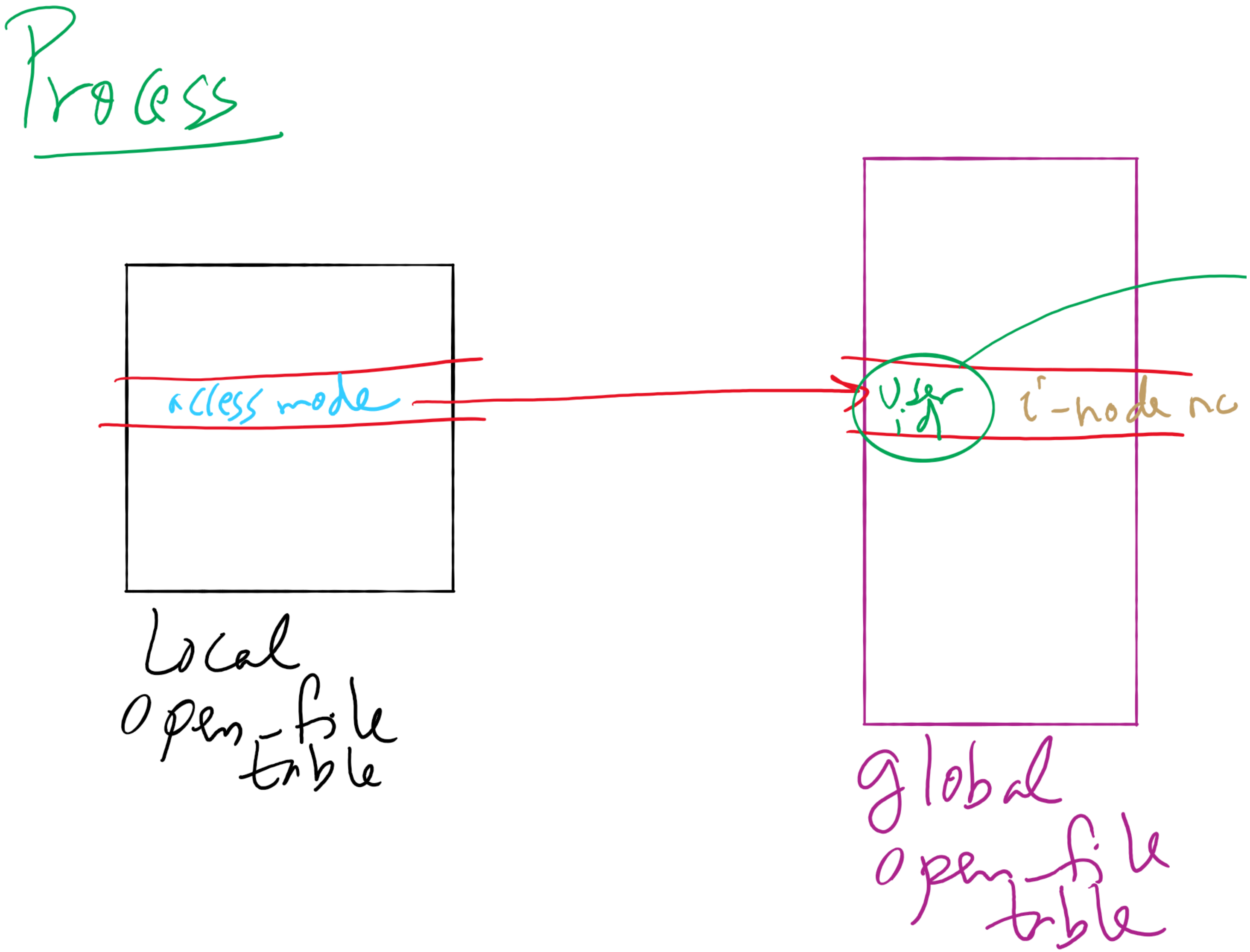
creates a new i-node that contains **path** of original file

e.g., ln -s command in Linux

copying

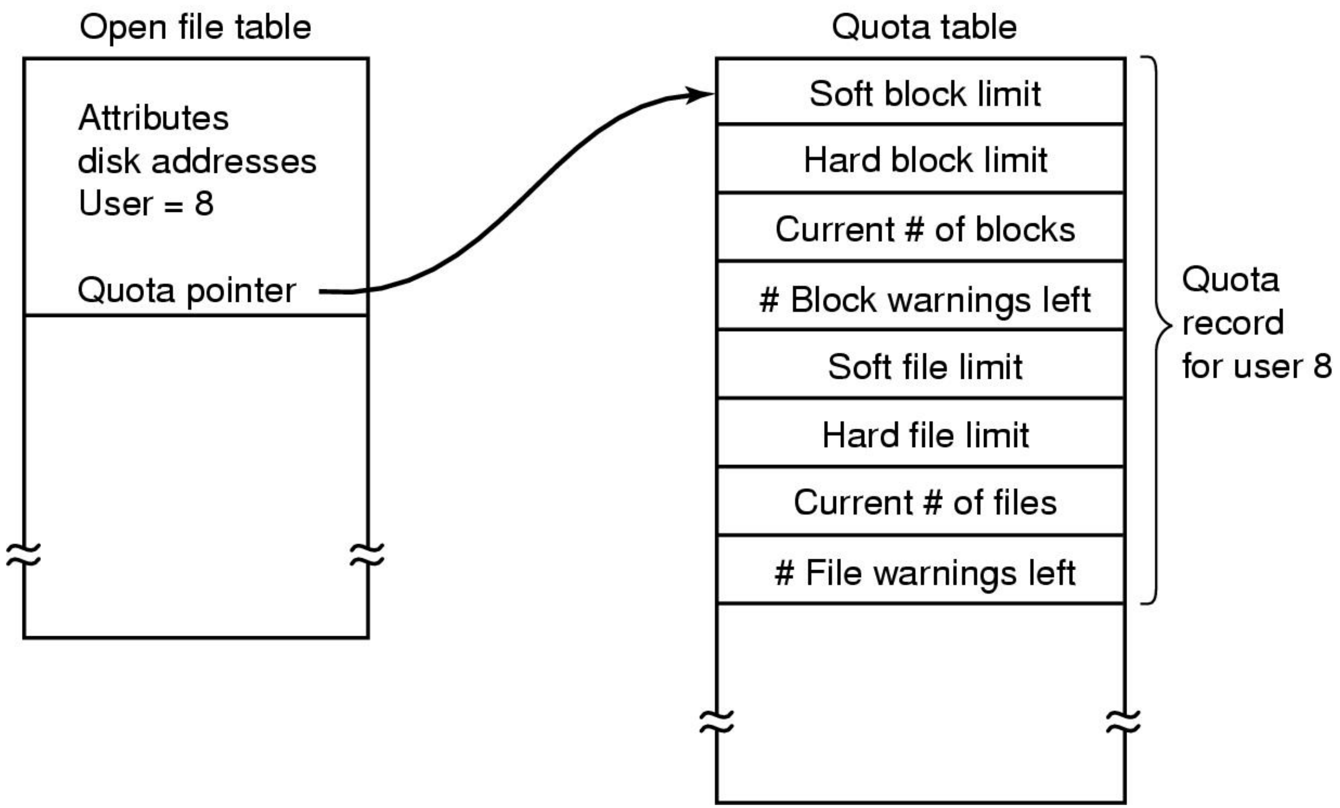
- creates a new i-node
- **e.g., cp** command in Linux

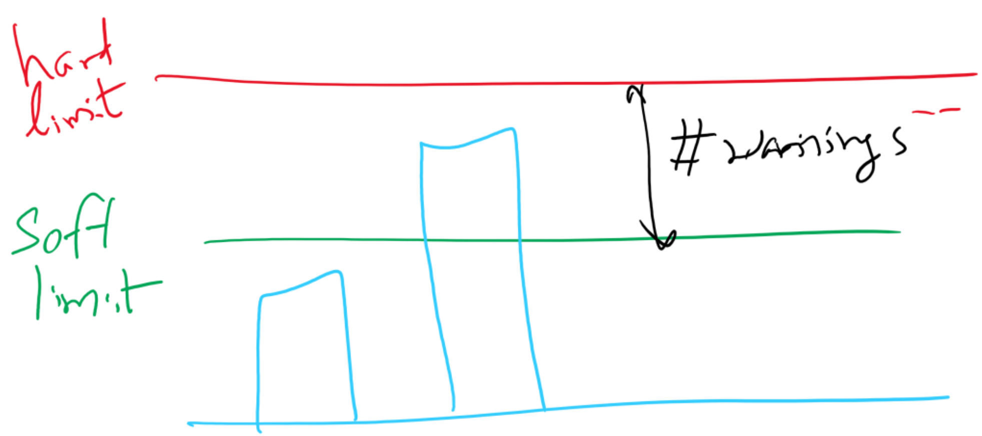
File-related kernel structures: open-file tables



per-process and global open-file tables

File-related kernel structures: quota table





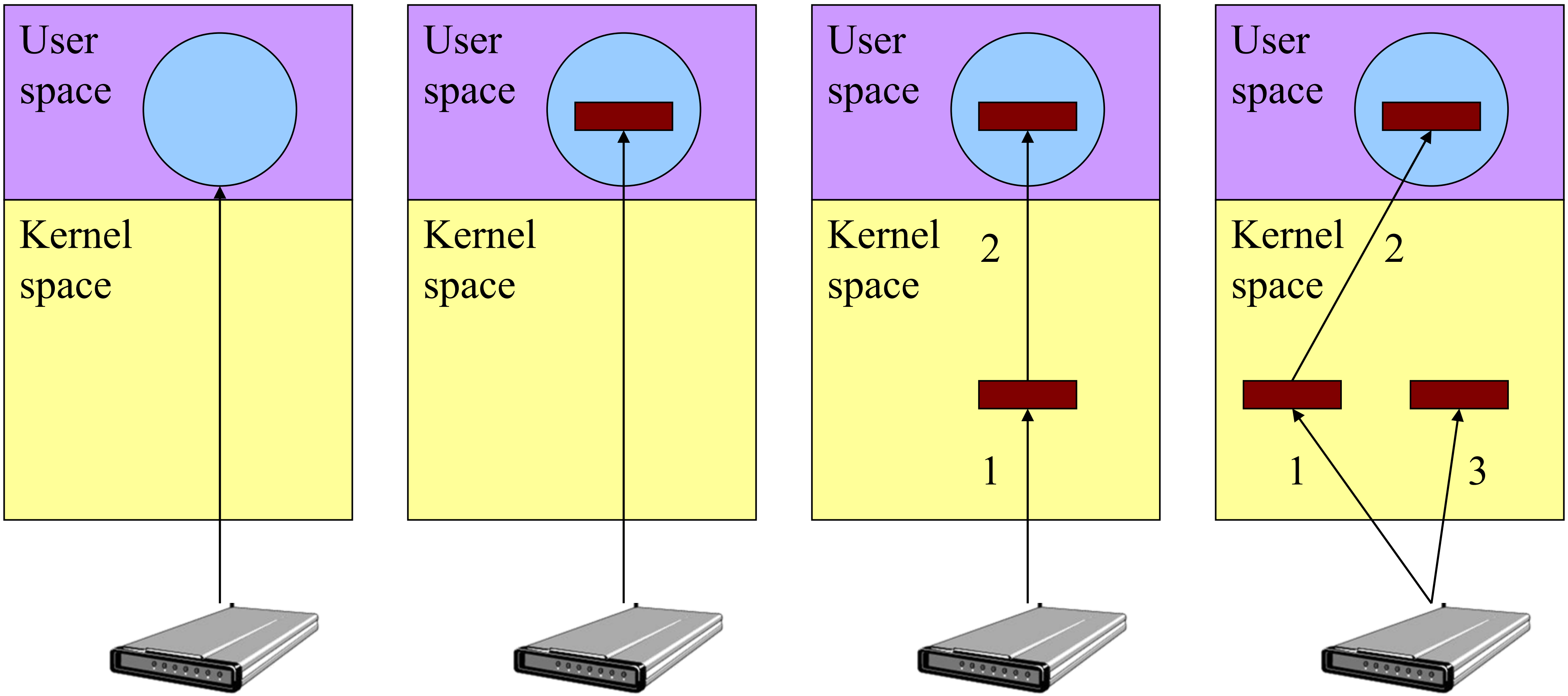
Journaling File System

- **Problem**: regular file system: changes to files and directories result in **multiple separate writes** to disk
  - e.g., deleting a file → three writes
  - **power failures** result in file system **inconsistency**
- **Solution**: Write the changes **twice**
  - first to an on-disk ***journal***
    - for efficiency, journal can be put on **SSD or NVRAM**
  - then **commit** changes to main part
  - atomic operations
- modified **data** may or may not be written to the journal
  - implications?
- Examples
  - Windows NTFS
  - Linux ext3, ext4

Journaling File System

- Interaction with disk arm scheduling?
  - may cause a different write order than wanted
  - **solution**: flush write cache at certain points (barriers)
    - ext3 and ext4 file systems
- Journaling vs. Log-structured file system
  - main file system itself is a journal
  - journaling is not needed in LFS

Buffering device input



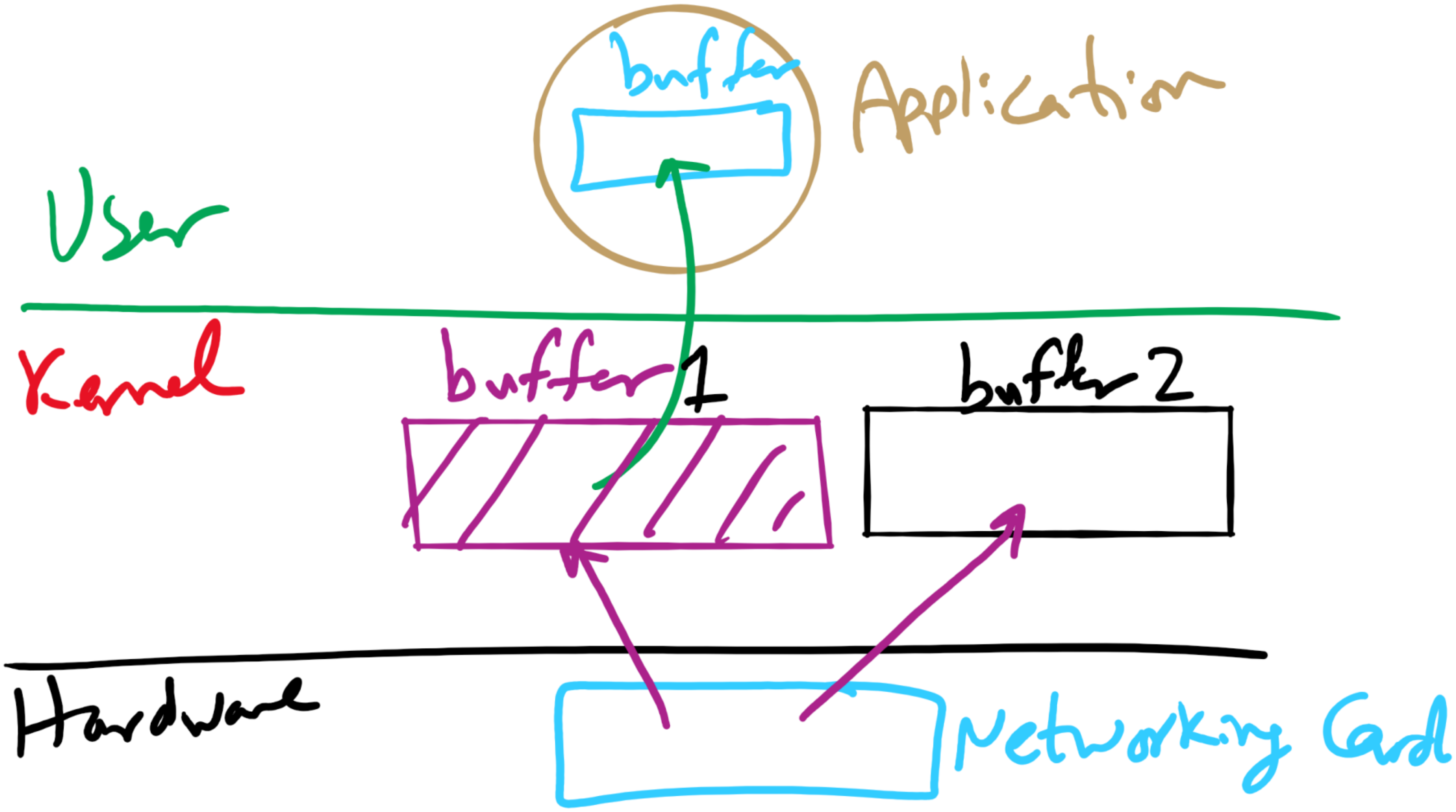
Unbuffered input

Buffering in user space

Buffer in kernel
Copy to user space

Double buffer in kernel

Double Buffering



Max Partition Size

Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

Max Partition Size

Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

32-bit block no.

(block no. size)

Max. Partition size $= 2^{\text{(block no. size)}} * \text{block size}$

FAT-12 : $2^{12} * \frac{1}{2}$ KB

$= 2^{12} * 2^{9} = 2^{21} = 2$MB

FAT_16 : $2^{16} * 2$ kB

$= 2^{16} * 2^{11} = 2^{27} = 128$MB

FAT-32 : $2^{28} * 4$ kB

$= 2^{28} * 2^{12} = 2^{40} = 1$TB

$2^{\text{sector no. size}} * 512 = 2^{32} * 512$

$= 2^{41} = 2$TB

Max Partition Size

Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

32-bit block no.

Max. Partition Size = $2^{(\text{block no. size})}$ * block size

Max Partition Size

Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

32-bit block no.

Max. Partition Size $= 2^{(\text{block no. size})} * \text{block size}$

FAT-12 : $2^{12} * \frac{1}{2}$ KB

$= 2^{12} * 2^{9} = 2^{21} = 2$MB

Max Partition Size

Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

FAT_16 : $2^{16} * 2 KB$

$= 2^{16} * 2^{11} = 2^{27} = 128 MB$

Max Partition Size

Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

FAT-32 : $2^{28} * 4 KB$

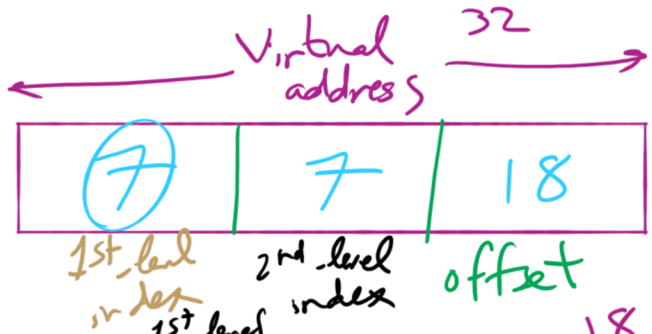
$= 2^{28} * 2^{12} = 2^{40} = 1 TB$

sector no.
size

$2 * 512 = 2^{32} * 512$

$= 2^{41} = 2 TB$

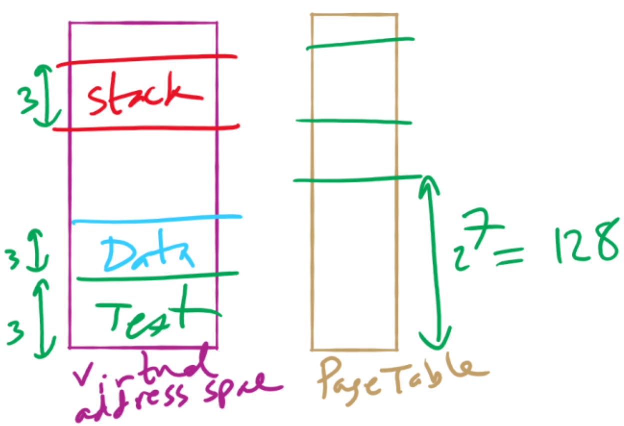
HW 10: Q 2-4



$\# PDEs = 2^{\text{1st\_level index size}} = 2^7$

$2^8 KB = 2^{18}$ bytes

$\# PTEs$ in 2^{nd} level Page Table $= 2^{\text{2nd\_level index size}} = 2^7$



HW 10: Q 10-13

single-level Page Table

$EAT = h(a+m) + (1-h)(a+m+m)$

2-level Page Table

$EAT = h(a+m) + (1-h)(a+m+m+m)$

0x1200

$\underbrace{1}_{\text{Page No}}\underbrace{200}_{\text{offset}}$

(PID, VPN)

(30, 1)

$4KB = 2^{12}$

$\Rightarrow$ offset : 12 bits

Frame no. : 3

~~3~~~~1~~200

$\boxed{3200}$

PID

(10, ~~7~~) 7

0x~~4~~500 7

frame no.

offset

frame no. : 4

$\boxed{0x7500}$

Effective Disk Access Time

$$EDAT = h * hit\,time + (1-h) * miss\,time$$

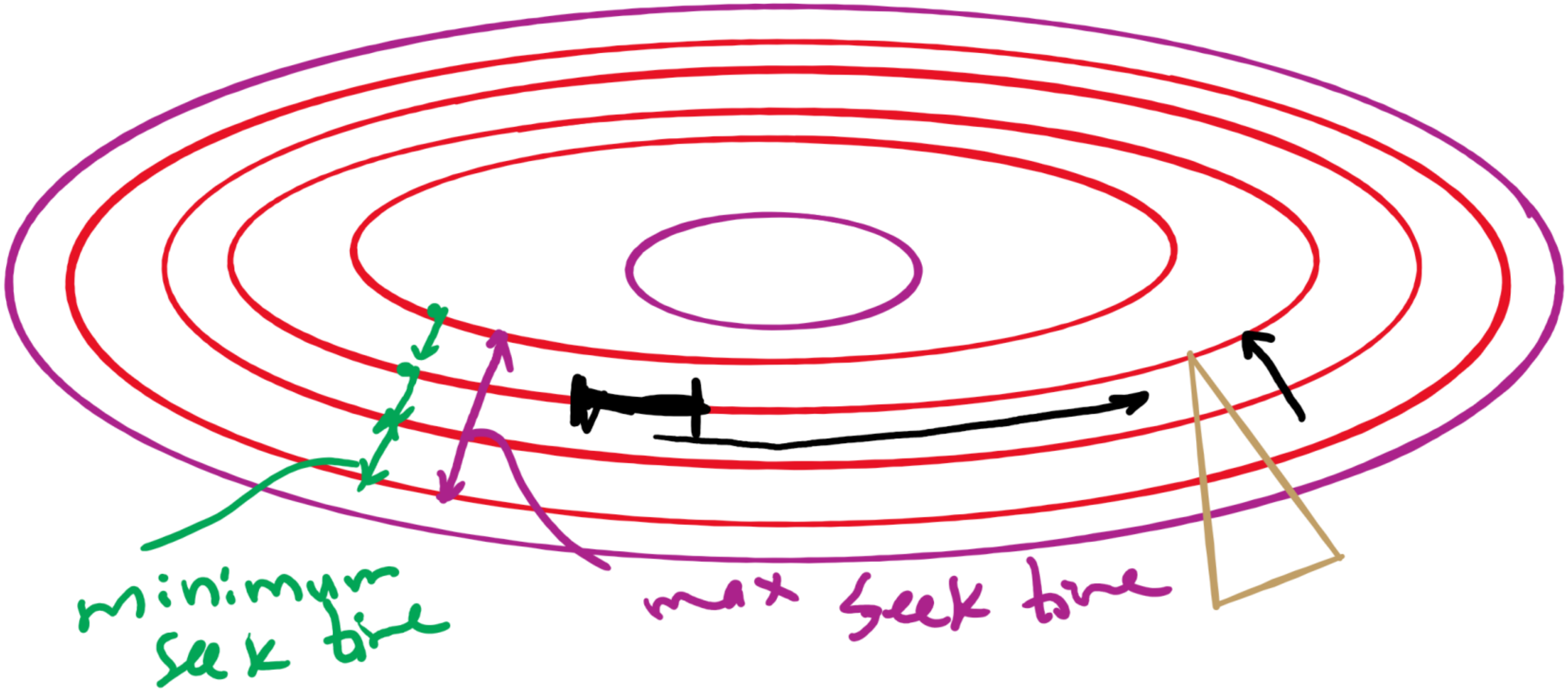
0.93

0.01

0.07

0.01 + 3 * 10

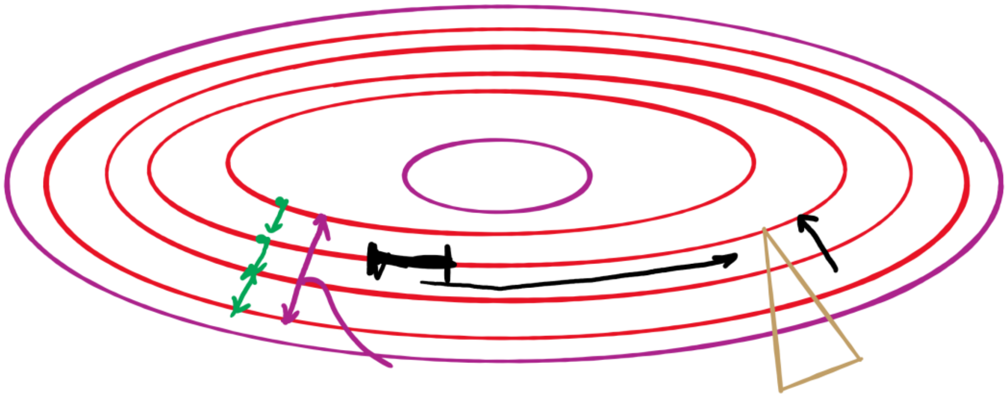
Minimum and Maximum Seek Time



Max Seek time = min Seek time * (# tracks-1)

#cylinders

Average Rotational Delay



Average rotational delay = $\dfrac{\text{one full rotation time}}{2}$

$= \dfrac{1/\text{rotation speed}}{2}$

$= \dfrac{(1/4800) * 60 * 1000}{2}$ ms