Operating Systems Concepts

Final Review

CS 4375, Fall 2025
Instructor: MD Armanuzzaman (***Arman***)
marmanuzzaman@utep.edu
December 1, 2025

Summery

- Protection and Security Concepts
- Security and Security Violation
- Cryptography
  - Symmetric
  - Asymmetric
- Program threats
  - Trojan Horse
  - Trap Door
  - Logic Bomb
  - Stack and Buffer Overflow
  - Viruses
  - Etc.

Agenda

- Final Exam logistics
- Final Topics
- Exercise question

Logistics

- Friday 12/12/2025
    - 10 AM - 12.45 PM
    - One page handwritten note
    - Bring you UTEP ID
    - More instructions to follow

Final Exam

- **Midterm topics: No direct questions**
  - Processes
  - Threads
  - System Calls
  - CPU Scheduling
  - Address translations
- **REVIEW THE BASIC UNDERSTANDING AS A PREPARATION OF THE FINAL**

Final Exam

- Main Memory
- Virtual Memory
- Process Synchronization
- Deadlocks
- File System
- Protection and Security

Final Exam: Main Memory

- Contiguous Allocation
- Dynamic Allocation Algorithms
- Fragmentation
- Address Binding
- Address Protection
- Paging
- Segmentation

Final Exam: Virtual Memory

- Demand Paging
- Page Faults
- Page Replacement
- Page Replacement Algorithms (FIFO, LRU, Clock, LFU, MFU, Optimal)
- Performance of Demand Paging

Final Exam: Process Synchronization

- Race Conditions
- Critical Section Problem
- Semaphores
- Mutex/Locks
- Conditional Variables

Final Exam: Deadlocks

- Deadlock characterization
- Resource Allocation Graph
- Handling of Deadlocks
  - Prevention
  - Detection
  - Avoidance
  - Recovery

Final Exam: File Systems

- Directory structure & implementation
- File allocation methods
  - Contiguous, linked, indexed
- Free space management
  - Bit vectors, linked lists, grouping, counting

Final Exam: Protection and Security

- Protection and Security Concepts
- Security Violation Categories and Methods
- Program & Network Threats
- Cryptography
    - Symmetric
    - Asymmetric

Exercise Questions: Question 1

Provide short answers to the following questions

- Can the segments that are **shared between two or more processes be swapped** out to the disk?

Exercise Questions: Question 1

Provide short answers to the following questions

- Can the segments that are **shared between two or more processes be swapped** out to the disk?

Yes, sharing does not require locking

Exercise Questions: Question 2

Provide short answers to the following questions

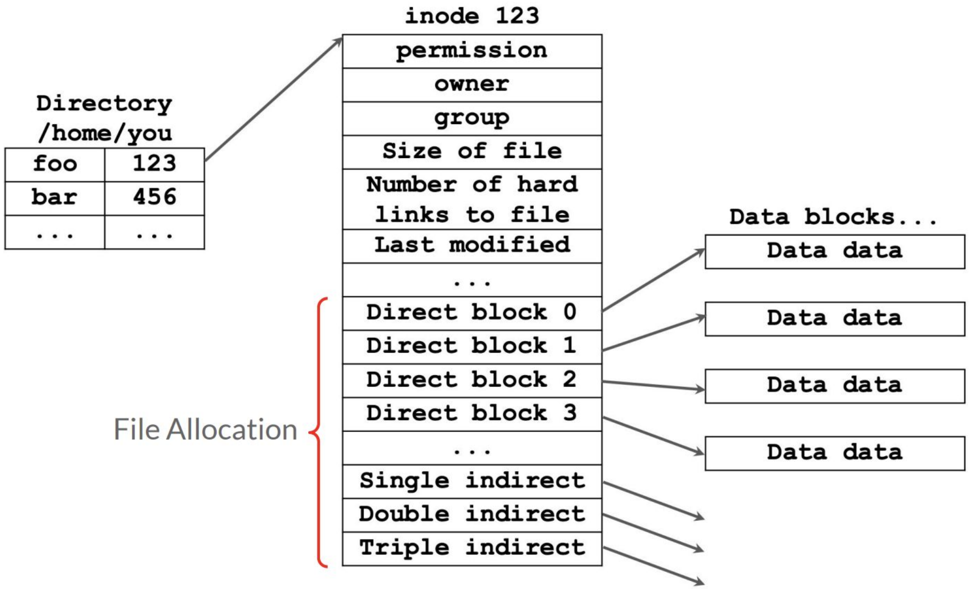
- The layout of disk blocks for a file using Unix inodes is **always/sometimes/never** contiguous on disk?

Exercise Questions: Question 2

Provide short answers to the following questions

- The layout of disk blocks for a file using Unix inodes is **always/sometimes/never** contiguous on disk?

Sometimes



Exercise Questions: Question 3

Provide short answers to the following questions

- Piece of code which is made available to unsuspecting user, that misuses its environment is called:
    1. Trojan Horse
    2. Stealth Virus
    3. Logic Bomb
    4. Trap Door

Exercise Questions: Question 3

Provide short answers to the following questions

- Piece of code which is made available to unsuspecting user, that misuses its environment is called:
  1. **Trojan Horse**
  2. Stealth Virus
  3. Logic Bomb
  4. Trap Door

Exercise Questions: Question 4

Provide short answers to the following questions

- The modified (dirty) bit is used for the purpose of:
    1. Implementing FIFO page replacement algorithm
    2. To reduce the average time required to service page faults
    3. Dynamic allocation of memory used by one process to another
    4. All of the above

Exercise Questions: Question 4

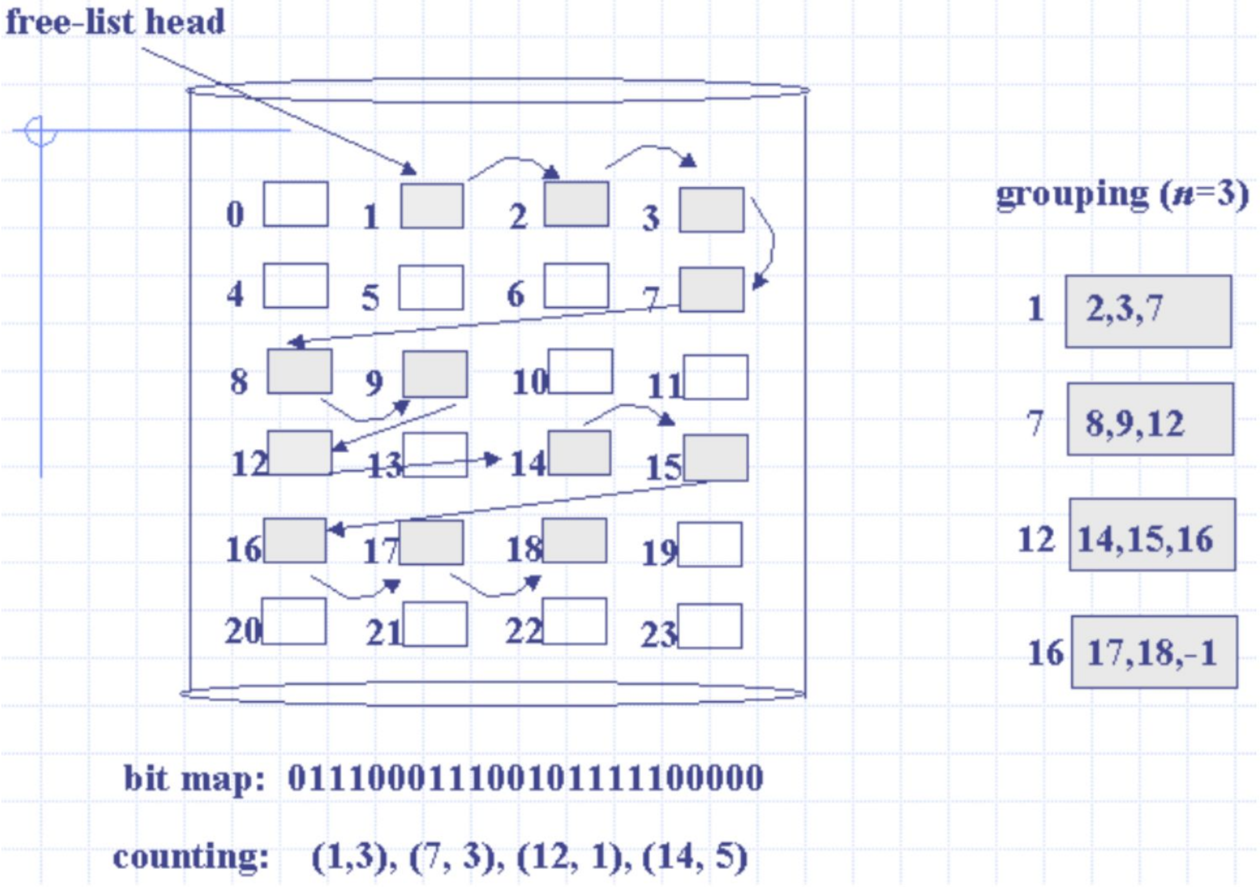
Provide short answers to the following questions

- The modified (dirty) bit is used for the purpose of:
  1. Implementing FIFO page replacement algorithm
  2. To reduce the average time required to service page faults
  3. Dynamic allocation of memory used by one process to another
  4. All of the above

Exercise Questions: Question 5

In terms of reliability and performance, compare bit vector implementation of a free block list with keeping a list of free blocks where the first few bytes of each free block provide the logical sector number of the next free block.

Exercise Questions: Question 5



Exercise Questions: Question 5

In terms of reliability and performance, compare bit vector implementation of a free block list with keeping a list of free blocks where the first few bytes of each free block provide the logical sector number of the next free block.

Performance: Bit vector implementation is more efficient since it allows fast and random access to free blocks, linked list approach allows only sequential access, and each access results in a disk read. Bit vector implementation can also find n consecutive free blocks much faster. (Assuming bit vector is kept in memory)

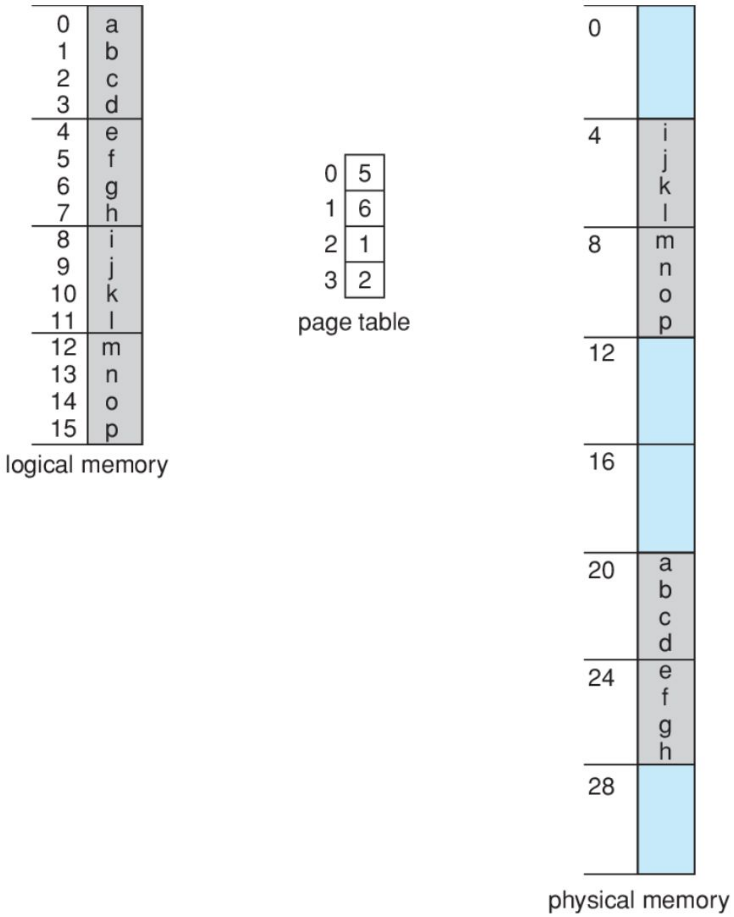
Reliability: If an item in a linked list is lost, you cannot access the rest of the list. With a bit vector, only those items are lost. Also, it's possible to have multiple copies of the bit vector since it is a more compact representation. Although keeping the bit vector in memory seem to be unreliable, you can always keep an extra copy on the disk.

Exercise Questions: Question 6

Consider the paging table on the right.

What are the physical addresses of the following logical addresses [p,d] and the words on them:

1. 0, 0
2. 1, 4
3. 2, 3

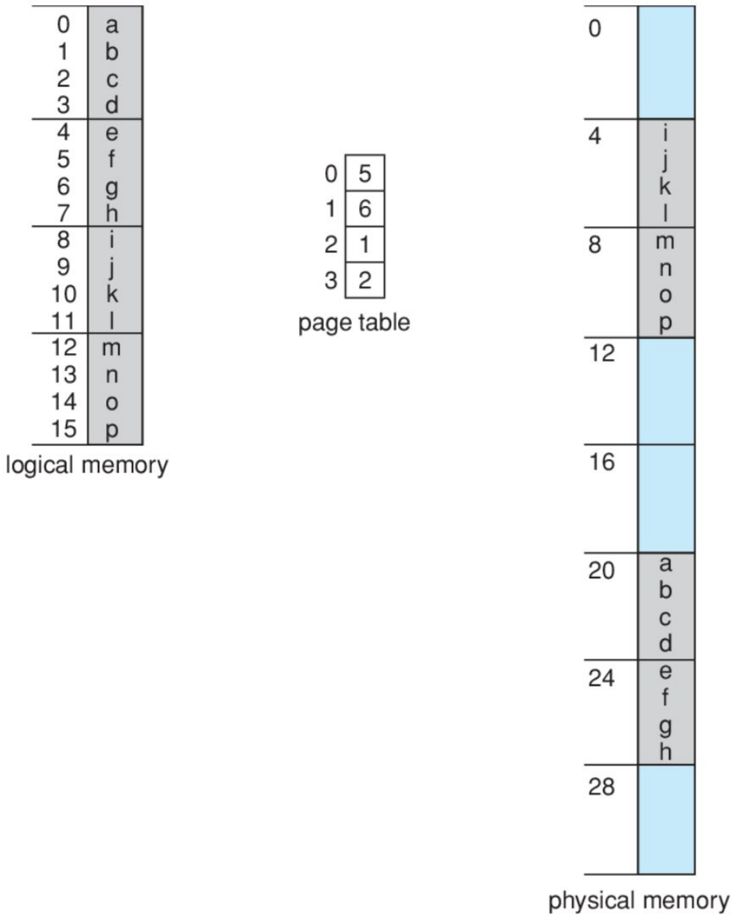


Exercise Questions: Question 6

Consider the paging table on the right.

What are the physical addresses of the following logical addresses [p,d] and the words on them:

1. 0, 0 → 20, a
2. 1, 4 → illegal/does not exist
3. 2, 3 → 7, l



Exercise Questions: Question 7

Consider the following segment table:

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses?

1. 1, 100
2. 2, 0
3. 3, 580

Exercise Questions: Question 7

Consider the following segment table:

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses?

1. 1, 100 ⟶ illegal (2300+100 is not within segment limits)
2. 2, 0 ⟶ physical address = 90 + 0 = 90
3. 3, 580 ⟶ illegal (1327 + 580 is not within segment limits)

Exercise Questions: Question 8

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

a) Install a faster CPU

Exercise Questions: Question 8

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

a) Install a faster CPU

NO, a faster CPU reduces the CPU utilization further since the CPU will spend more time waiting for a process to enter in the ready queue.

Exercise Questions: Question 8

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

b) Install a bigger paging disk

Exercise Questions: Question 8

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

b) Install a bigger paging disk

NO, the size of the paging disk does not affect the amount of memory that is needed

to reduce the page faults.

Exercise Questions: Question 8

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

c) Decrease the degree of multiprogramming

Exercise Questions: Question 8

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

c) Decrease the degree of multiprogramming

LIKELY, by suspending some of the processes, the other processes will have more

frames in order to bring their pages in them, hence reducing the page faults.

Exercise Questions: Question 8

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

d) Install more main memory

Exercise Questions: Question 8

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

d) Install more main memory

Likely, more pages can remain resident and do not require paging to or from the disks (i.e. would depend on the page replacement algorithm you are using and the page reference sequence).

Exercise Questions: Question 9

Consider a demand paging system where 40% of the page table is stored in the registers, and the rest needs to be addressed from the memory. Assume any reference to the memory takes 100 ns, and 20% of the time the page that is being replaced has its dirty bit set to 1. Swapping a page in takes 1000 us and swapping out a page takes 2000 us.

(1 ns ≪ 1 us ≪ 1 ms ≪ 1s)

a) How long does a paged memory reference take?

Exercise Questions: Question 9

Consider a demand paging system where 40% of the page table is stored in the registers, and the rest needs to be addressed from the memory. Assume any reference to the memory takes 100 ns, and 20% of the time the page that is being replaced has its dirty bit set to 1. Swapping a page in takes 1000 us and swapping out a page takes 2000 us.

(1 ns ≪ 1 us ≪ 1 ms ≪ 1s)

a) How long does a paged memory reference take?

Hit + Miss

(0.4 x M) + (0.6 x 2M) = 1.6 M = 160 ns = 0.160 us

or

Extra + Always

(0.6 x M) + M = 60 + 100 = 160 ns = 0.160 us

Exercise Questions: Question 9

Consider a demand paging system where 40% of the page table is stored in the registers, and the rest needs to be addressed from the memory. Assume any reference to the memory takes 100 ns, and 20% of the time the page that is being replaced has its dirty bit set to 1. Swapping a page in takes 1000 us and swapping out a page takes 2000 us.

(1 ns ≪ 1 us ≪ 1 ms ≪ 1s)

b) What is the Effective Access Time (EAT) if the page fault ratio is 0.001?

Exercise Questions: Question 9

Consider a demand paging system where 40% of the page table is stored in the registers, and the rest needs to be addressed from the memory. Assume any reference to the memory takes 100 ns, and 20% of the time the page that is being replaced has its dirty bit set to 1. Swapping a page in takes 1000 us and swapping out a page takes 2000 us.

(1 ns ≪ 1 us ≪ 1 ms ≪ 1s)

b) What is the Effective Access Time (EAT) if the page fault ratio is 0.001?

EAT = (1-p) x (Average Hit) + p x (Average Miss)

= (1-p) x 0.160 + p x ((0.80) x 1000 + (0.2) x 3000 + 0.160)

or = (1-p) x 0.160 + p x ((0.2) x 2000 + 1000 + 0.160)

= 0.160 + p x 1400

= 0.160 + 1/1000 x 1400 = 1.56 us

Exercise Questions: Question 9

Consider a demand paging system where 40% of the page table is stored in the registers, and the rest needs to be addressed from the memory. Assume any reference to the memory takes 100 ns, and 20% of the time the page that is being replaced has its dirty bit set to 1. Swapping a page in takes 1000 us and swapping out a page takes 2000 us.

(1 ns ≪ 1 us ≪ 1 ms ≪ 1s)

b) What is the Effective Access Time (EAT) if the page fault ratio is 0.001?

EAT = M + p ([Average Swapping overhead])

= M + p ([Average Swap out] + Swap in)

= 0.160 + p x ((0.2)x2000 + 1000)

= 0.160 + p x 1400

= 0.160 + 1/1000 x 1400 = 1.56 us

Exercise Questions: Question 9

Consider a demand paging system where 40% of the page table is stored in the registers, and the rest needs to be addressed from the memory. Assume any reference to the memory takes 100 ns, and 20% of the time the page that is being replaced has its dirty bit set to 1. Swapping a page in takes 1000 us and swapping out a page takes 2000 us.

(1 ns ≪ 1 us ≪ 1 ms ≪ 1s)

c) What should be the maximum acceptable page-fault rate if we only want 50% performance degradation?

Exercise Questions: Question 9

Consider a demand paging system where 40% of the page table is stored in the registers, and the rest needs to be addressed from the memory. Assume any reference to the memory takes 100 ns, and 20% of the time the page that is being replaced has its dirty bit set to 1. Swapping a page in takes 1000 us and swapping out a page takes 2000 us.

(1 ns ≪ 1 us ≪ 1 ms ≪ 1s)

c) What should be the maximum acceptable page-fault rate if we only want 50% performance degradation?

$$\text{EAT} / M < 150\% \rightarrow (0.160 + p \times 1400) / 0.160 < 1.5$$

$$0.160 + p \times 1400 < 0.240 \rightarrow p \times 1400 < 0.80 \rightarrow p < 0.80 / 1400$$

Can leave it here

Approx 0.000057

Exercise Questions: Question 9

Consider a virtual memory system with 34-bit addresses. The first 23 bits are used as a page number, and the last 11 bits is the offset. (2^{10} = 1K, 2^{20} = 1M)

a) How many words does a single page frame have?

Exercise Questions: Question 9

Consider a virtual memory system with 34-bit addresses. The first 23 bits are used as a page number, and the last 11 bits is the offset. (2^{10} = 1K, 2^{20} = 1M)

a) How many words does a single page frame have?

Each frame has 2^{11} = 2048 words

Assuming each word is 1 byte, it will be 2048 bytes

Exercise Questions: Question 9

Consider a virtual memory system with 34-bit addresses. The first 23 bits are used as a page number, and the last 11 bits is the offset. (2^{10} = 1K, 2^{20} = 1M)

b) Assuming there are 2^{10} frames in the physical memory, how many bits are needed to address the physical memory?

Exercise Questions: Question 9

Consider a virtual memory system with 34-bit addresses. The first 23 bits are used as a page number, and the last 11 bits is the offset. (2^{10} = 1K, 2^{20} = 1M)

b) Assuming there are 2^{10} frames in the physical memory, how many bits are needed to address the physical memory?

Physical memory size = $2^{10} * 2^{11} = 2^{21}$ bytes = 2 MB

21 bits are needed to address the physical memory

Exercise Questions: Question 9

Consider a virtual memory system with 34-bit addresses. The first 23 bits are used as a page number, and the last 11 bits is the offset. (2^{10} = 1K, 2^{20} = 1M)

c) Assuming the single-level paging and each page-table entry to be 4 bytes, how many Megabytes of memory is needed to store the page table?

Exercise Questions: Question 9

Consider a virtual memory system with 34-bit addresses. The first 23 bits are used as a page number, and the last 11 bits is the offset. (2^{10} = 1K, 2^{20} = 1M)

c) Assuming the single-level paging and each page-table entry to be 4 bytes, how many Megabytes of memory is needed to store the page table?

2^{23} entries * 4 bytes = 2^{25} bytes = 32 MB !

Exercise Questions: Question 10

Consider the following page-reference string:

1, 2, 3, 4, 5, 6, 7, 8, 5, 6, 2, 3, 6, 2, 3, 7, 8, 3, 2, 1, 5, 6, 2, 4

a) Show the page assignments to frames assuming second chance (clock) algorithm is used. Please fill in all frames. Consider the following rules:
   i) When a page is brought to the memory the first time, initialize reference bit to 0
   ii) Advance the next victim pointer only if you need to find a victim page to replace and when you bring a new page in.
b) Calculate the number of page faults, page hits, and page replacements

Exercise Questions: Question 10

Consider the following page-reference string:

1, 2, 3, 4, 5, 6, 7, 8, 5, 6, 2, 3, 6, 2, 3, 7, 8, 3, 2, 1, 5, 6, 2, 4

Referenced Page		1	2	3	4	5	6	7	8	5	6	2	3	6	2	3	7	8	3	2	1	5	6	2	4
Frame-1		1_0	1_0	1_0	1_0	5_0	5_0	5_0	5_0	5_1	5_1	5_0	5_0	5_0	5_0	5_0	7_0	8_0	8_0	8_0	8_0	5_0	5_0	5_0	5_0
Frame-2			2_0	2_0	2_0	2_0	6_0	6_0	6_0	6_0	6_1	6_0	6_0	6_1	6_1	6_1	6_1	6_0	6_0	6_0	1_0	1_0	6_0	6_0	6_0
Frame-3				3_0	3_0	3_0	3_0	7_0	7_0	7_0	7_0	2_0	2_0	2_0	2_1	2_1	2_1	2_0	2_0	2_1	2_1	2_0	2_0	2_1	2_0
Frame-4					4_0	4_0	4_0	4_0	8_0	8_0	8_0	8_0	3_0	3_0	3_0	3_1	3_1	3_0	3_1	3_1	3_1	3_0	3_0	3_0	4_0

# of page faults: 16

# of page hits: 8

# of page replacements: 12

Check every 4-5 references!

Exercise Questions: Question 11

Given the following memory partitions, how would each of the **first-fit, best-fit,** and **worst-fit** algorithms place processes of 292, 522, 138, 770, 162, 418 KB (in order). Which algorithm makes the most efficient usage of memory?

200 KB, 600 KB, 500 KB, 800 KB, 400 KB, 300 KB

Exercise Questions: Question 11

First-fit:	Processes in order:	292	522	138	770	162	418
	Success?						
Memory holes:	200 KB						
	600 KB						
	500 KB						
	800 KB						
	400 KB						
	300 KB						

Exercise Questions: Question 11

First-fit:	Processes in order:	292	522	138	770	162	418
	Success?	✓					
Memory holes:	200 KB						
	~~600 KB~~	308					
	500 KB						
	800 KB						
	400 KB						
	300 KB						

Exercise Questions: Question 11

First-fit:	Processes in order:	292	522	138	770	162	418
	Success?	✓	✓				
Memory holes:	200 KB						
	~~600 KB~~	308					
	500 KB						
	~~800 KB~~		278				
	400 KB						
	300 KB						

Exercise Questions: Question 11

First-fit:	Processes in order:	292	522	138	770	162	418
	Success?	✓	✓	✓			
Memory holes:	~~200 KB~~			62			
	~~600 KB~~	308					
	500 KB						
	~~800 KB~~		278				
	400 KB						
	300 KB						

Exercise Questions: Question 11

First-fit:	Processes in order:	292	522	138	770	162	418
	Success?	✓	✓	✓	✗		
Memory holes:	~~200 KB~~			62			
	~~600 KB~~	308					
	500 KB						
	~~800 KB~~		278				
	400 KB						
	300 KB						

Exercise Questions: Question 11

First-fit:	Processes in order:	292	522	138	770	162	418
	Success?	✓	✓	✓	✗	✓	
Memory holes:	~~200 KB~~			62			
	~~600 KB~~	~~308~~				146	
	500 KB						
	~~800 KB~~		278				
	400 KB						
	300 KB						

Exercise Questions: Question 11

First-fit:	Processes in order:	292	522	138	770	162	418
	Success?	✓	✓	✓	✗	✓	✓
Memory holes:	~~200 KB~~			62			
	~~600 KB~~	~~308~~				146	
	~~500 KB~~						82
	~~800 KB~~		278				
	400 KB						
	300 KB						

Exercise Questions: Question 11

First-fit:

Final result

	Processes in order:	292	522	138	770	162	418
	Success?	✓	✓	✓	✗	✓	✓
Memory holes:	~~200 KB~~			62			
	~~600 KB~~	~~308~~				146	
	~~500 KB~~						82
	~~800 KB~~		278				
	400 KB						
	300 KB						

Exercise Questions: Question 11

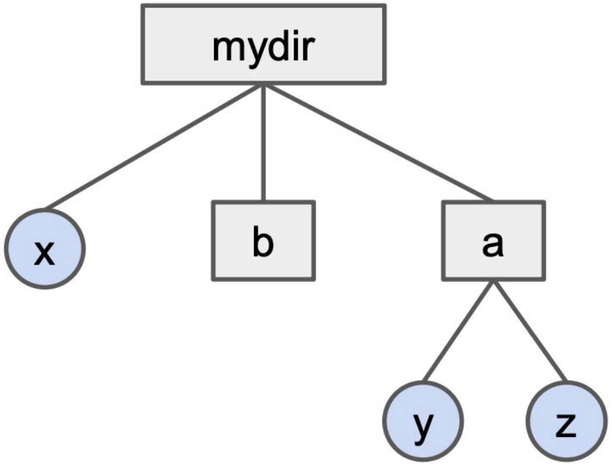
Best-fit:	Processes in order:	292	522	138	770	162	418
Final result	Success?	✓	✓	✓	✓	✓	✓
Memory holes:	~~200 KB~~			62			
	~~600 KB~~		78				
	~~500 KB~~						82
	~~800 KB~~				30		
	~~400 KB~~					238	
	~~300 KB~~	8					

Exercise Questions: Question 11

Worst-fit:	Processes in order:	292	522	138	770	162	418
Final result	Success?	✓	✓	✓	✗	✓	✗
Memory holes:	200 KB						
	~~600 KB~~		78				
	~~500 KB~~					338	
	~~800 KB~~	~~508~~		370			
	400 KB						
	300 KB						

Exercise Questions: Question 12

Consider the following directory structure (user view).



Assume the inode number for mydir's parent is 1 and

the inodes for directories and files shown are:

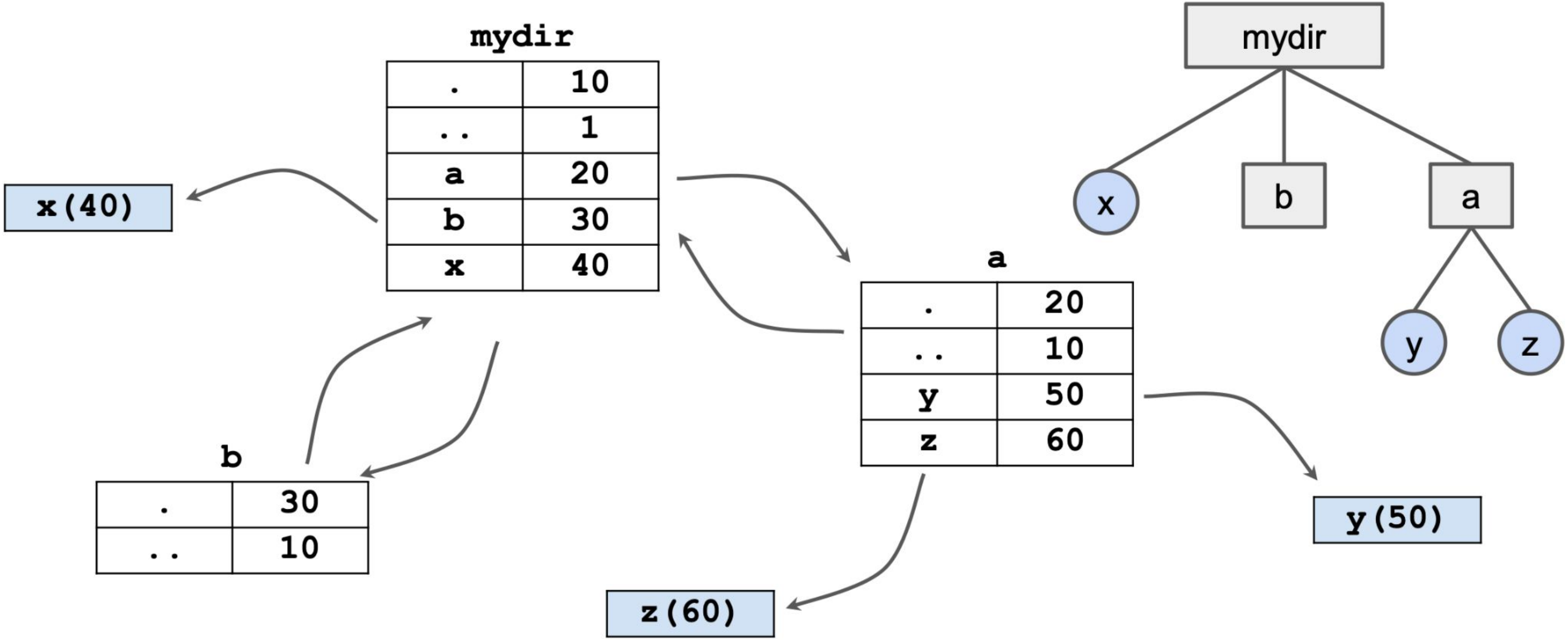
Directories: mydir = 10, a = 20, b = 30

Files: x = 40, y = 50, z = 60

a) Please show the system representation (system view) of this directory tree
b) Show the system representation (system view) after executing all of the following:
   i) `rm mydir/x`
   ii) `cp mydir/a/z mydir`
   iii) `ln mydir/a/y mydir/b/ylink`
   iv) `mv mydir/b mydir/a`

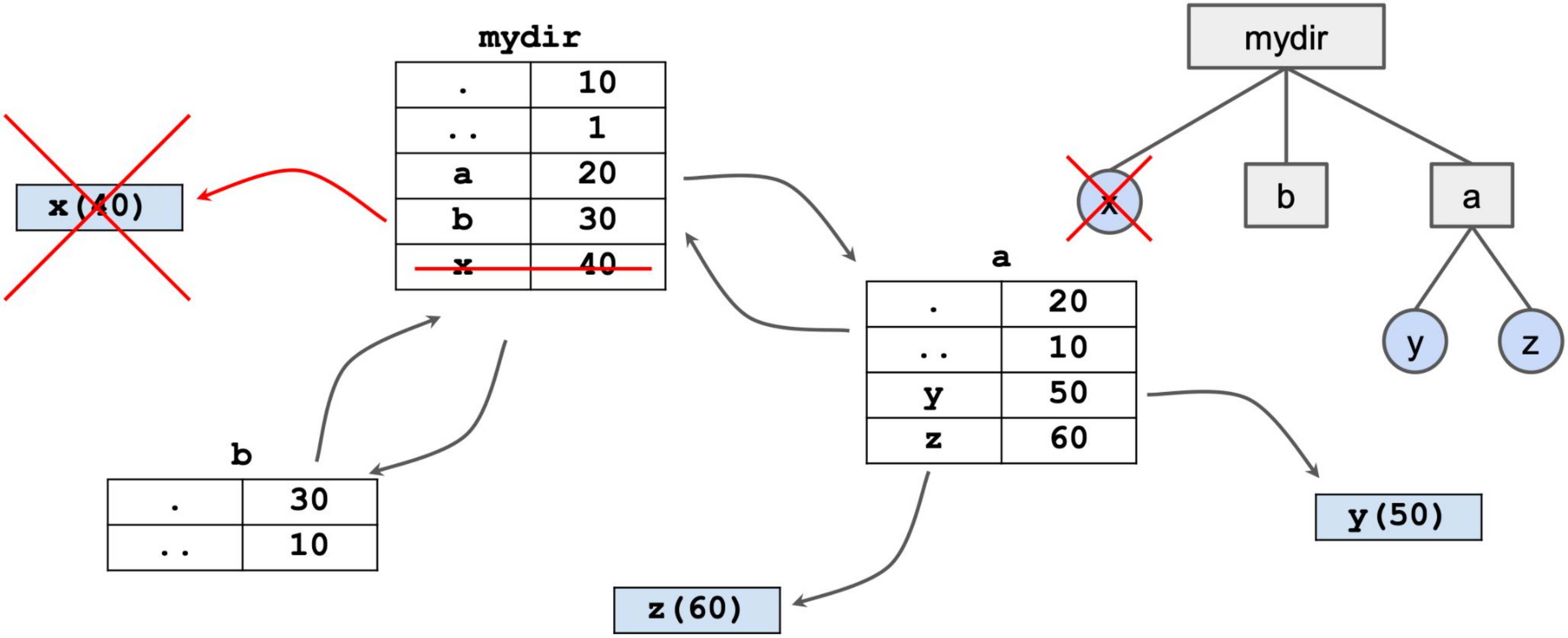
Exercise Questions: Question 12

Please show the system representation (system view) of this directory tree



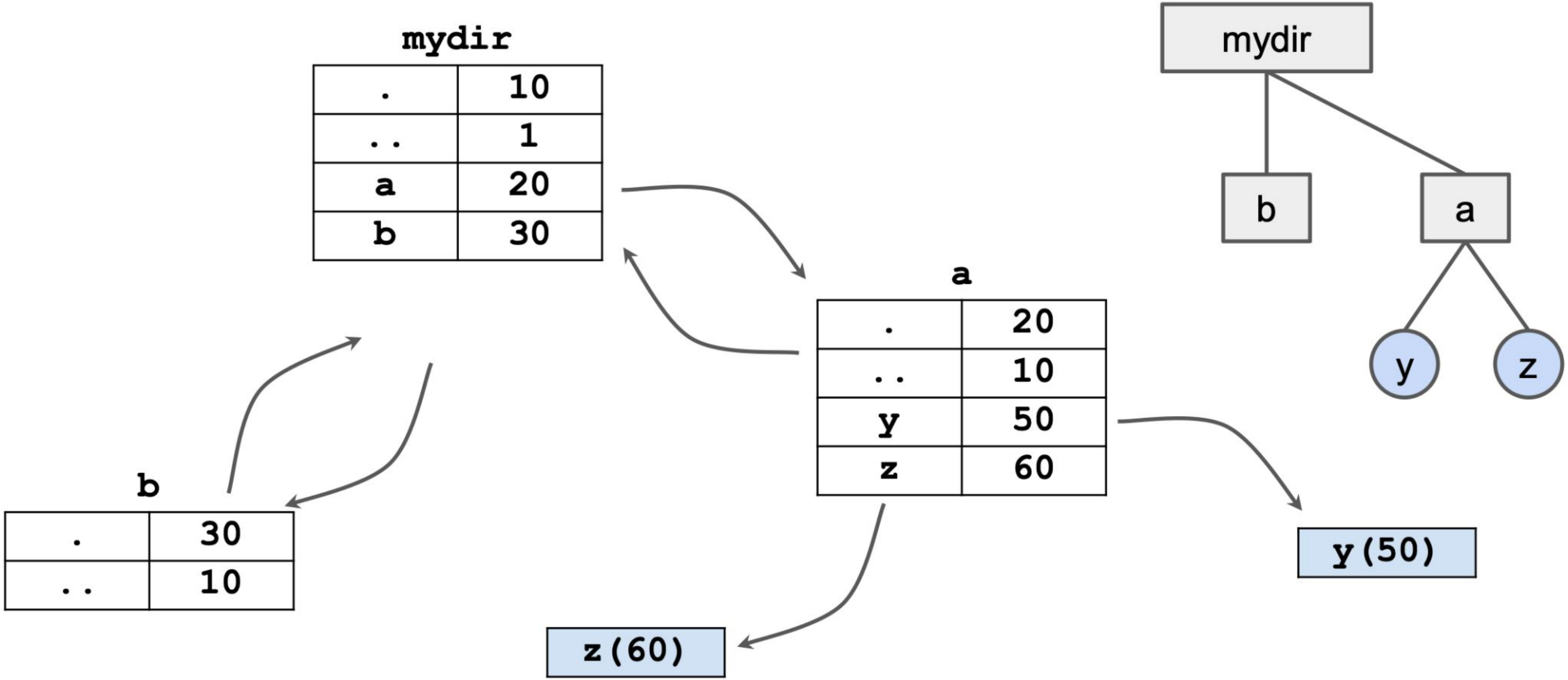
Exercise Questions: Question 12

b) Executing: i) `rm mydir/x`



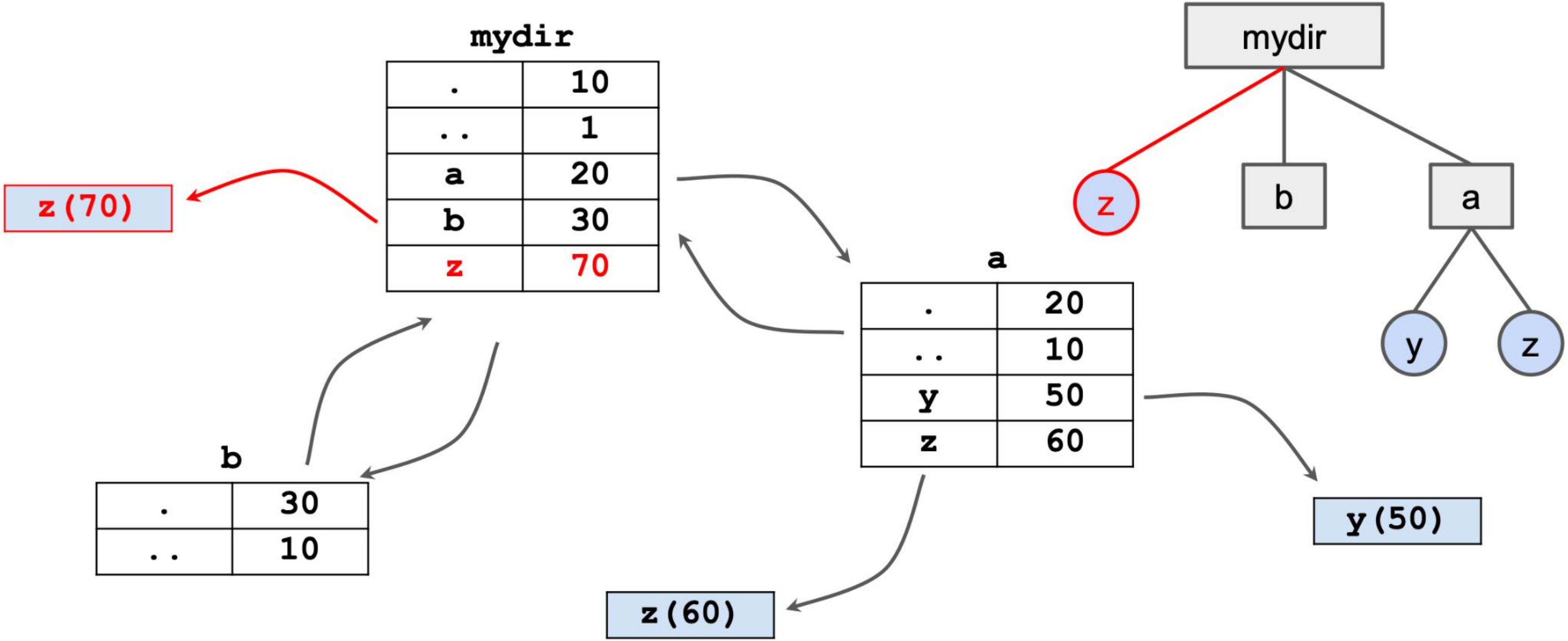
Exercise Questions: Question 12

b) Result: i) `rm mydir/x`



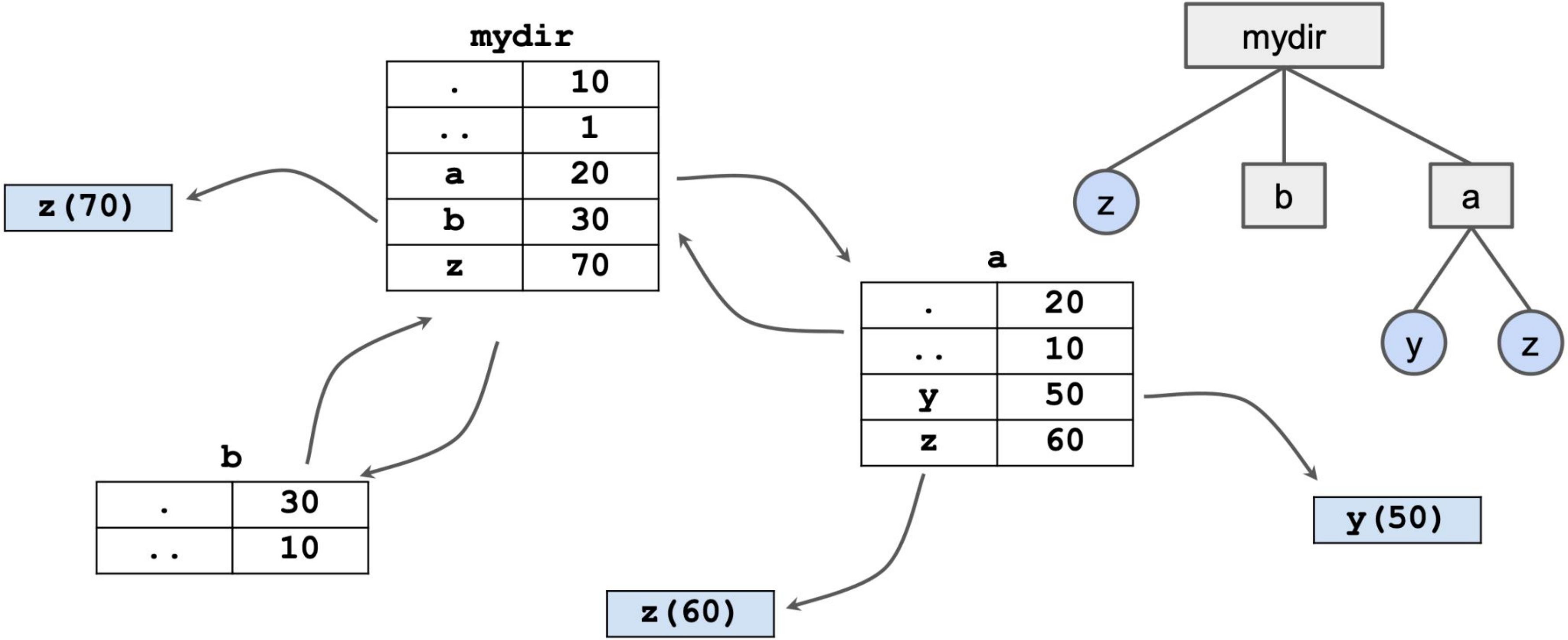
Exercise Questions: Question 12

b) Executing: ii) `cp mydir/a/z mydir`



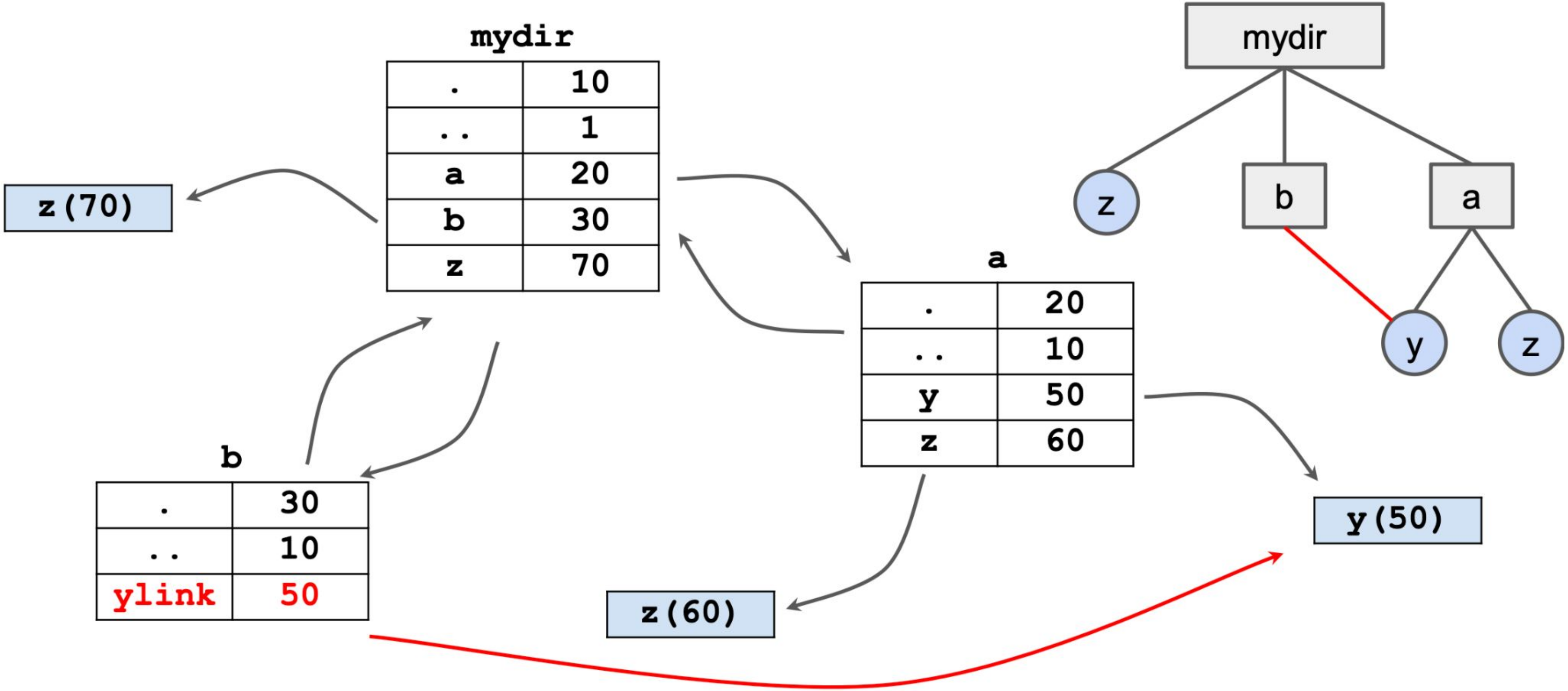
Exercise Questions: Question 12

b) Result: ii) `cp mydir/a/z mydir`



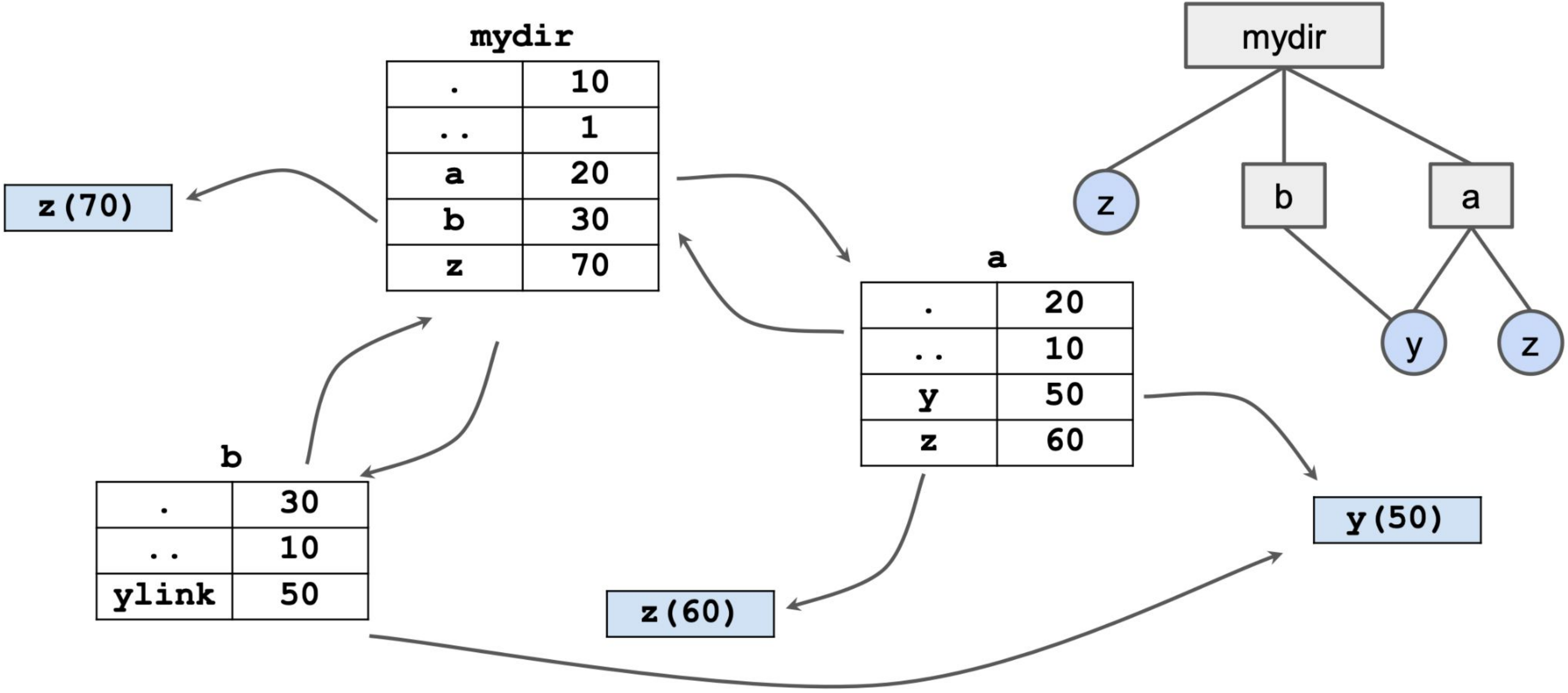
Exercise Questions: Question 12

b) Executing: iii) `ln mydir/a/y mydir/b/ylink`



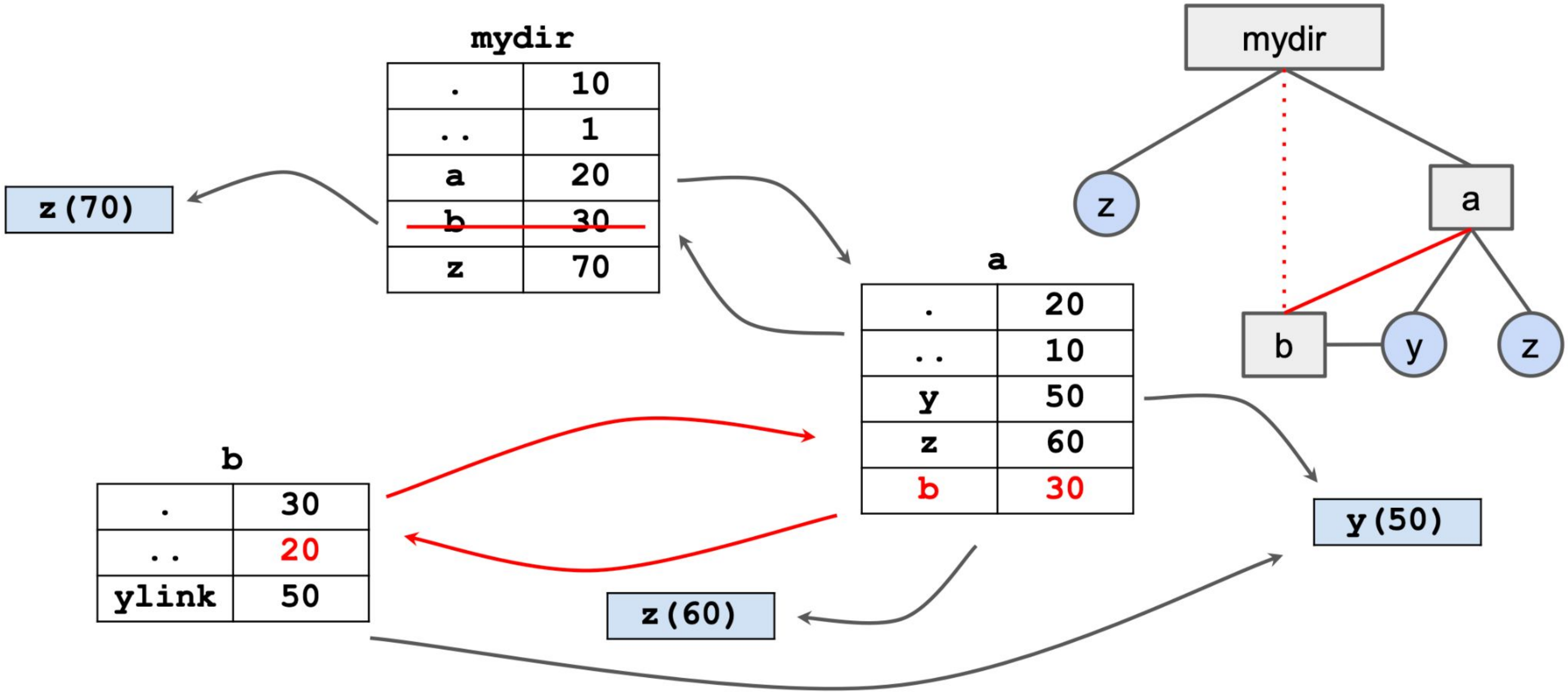
Exercise Questions: Question 12

b) Result: iii) `ln mydir/a/y mydir/b/ylink`



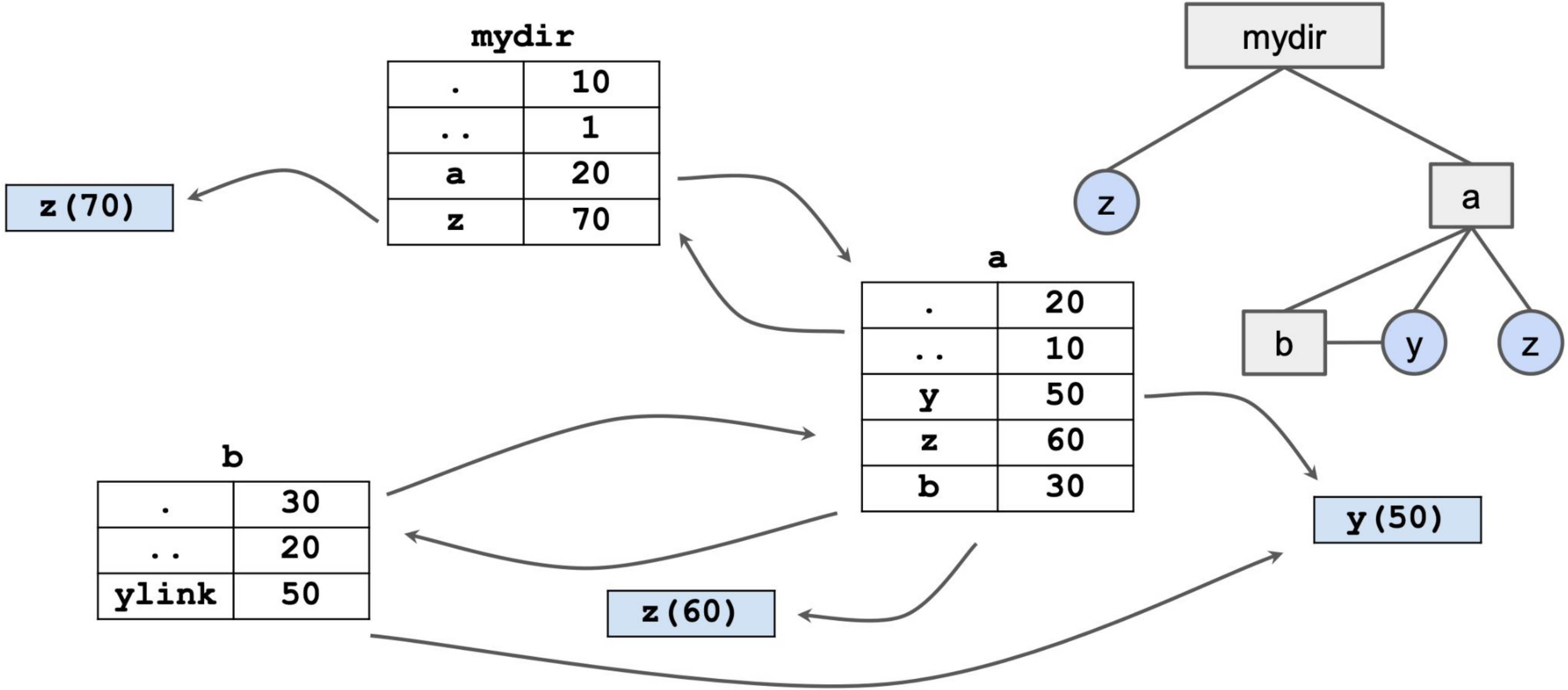
Exercise Questions: Question 12

b) Executing: iv) `mv mydir/b mydir/a`



Exercise Questions: Question 12

b) Result: iv) `mv mydir/b mydir/a`



Acknowledgements

- “Operating Systems Concepts” book and supplementary material by A. Silberschatz, P. Galvin and G. Gagne
- “Operating Systems: Internals and Design Principles” book and supplementary material by W. Stallings
- “Modern Operating Systems” book and supplementary material by A. Tanenbaum
- R. Doursat and M. Yuksel from University of Nevada, Reno
- Farshad Ghanei from Illinois Tech
- T. Kosar and K. Dantu from University at Buffalo

Announcement

- Quiz 8
  - Will be released on blackboard on 12/04
- Homework 5
  - Due on 12/08 - HARD DEADLINE: NO EXTENSION
- Course evaluation
  - 35 out of 59 **(60%)**