DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

COMPUTER ARCHITECTURES

Exercises: I/O Peripherals

Gábor Lencse, lencse@hit.bme.hu

Budapest,
2023. 03. 23.

MŰEGYETEM 1782

TASK 1.

The clock frequency of the CPU is 100MHz. A keyboard is connected to the computer. On average, 10 keys are pressed in a second. Theoretically, two key presses can follow each other in 50ms, key presses closer than this can not be distinguished by the keyboard. 500 clock cycles are necessary to obtain the status of the keyboard. The interrupt processing needs additional 100 clock cycles.

a) How many times do we need to poll the keyboard not to miss any events?

b) What is the (relative) load of the CPU devoted to the keyboard when the key press events are handled by polling?

c) What is the (relative) load of the CPU devoted to the keyboard when the key press events are handled by interrupts?

SOLUTION OF TASK 1.

- The parameters:
  - CPU clock frequency: 100MHz
  - On average, 10 keys are pressed in a second
  - Minimum time between 2 key presses: 50ms
  - Keyboard status read: 500 clock cycles
  - Interrupt processing overhead: 100 clock cycles

SOLUTION OF TASK 1.

- The parameters:
  - CPU clock frequency: 100MHz
  - On average, 10 keys are pressed in a second
  - Minimum time between 2 key presses: 50ms
  - Keyboard status read: 500 clock cycles
  - Interrupt processing overhead: 100 clock cycles

a) How many times do we need to poll the keyboard not to miss any events?

SOLUTION OF TASK 1.

- The parameters:
  - CPU clock frequency: 100MHz
  - On average, 10 keys are pressed in a second
  - Minimum time between 2 key presses: 50ms
  - Keyboard status read: 500 clock cycles
  - Interrupt processing overhead: 100 clock cycles

a) How many times do we need to poll the keyboard not to miss any events?
   - 1000ms / 50ms = 20 times per second

SOLUTION OF TASK 1.

- The parameters:
  - CPU clock frequency: 100MHz
  - On average, 10 keys are pressed in a second
  - Minimum time between 2 key presses: 50ms
  - Keyboard status read: 500 clock cycles
  - Interrupt processing overhead: 100 clock cycles

a) How many times do we need to poll the keyboard not to miss any events?
   - 1000ms / 50ms = 20 times per second

b) What is the (relative) load of the CPU devoted to the keyboard when the key press events are handled by polling?

SOLUTION OF TASK 1.

- The parameters:
  - CPU clock frequency: 100MHz
  - On average, 10 keys are pressed in a second
  - Minimum time between 2 key presses: 50ms
  - Keyboard status read: 500 clock cycles
  - Interrupt processing overhead: 100 clock cycles

a) How many times do we need to poll the keyboard not to miss any events?
   - 1000ms / 50ms = 20 times per second

b) What is the (relative) load of the CPU devoted to the keyboard when the key press events are handled by polling?
   - 20*500 / 100,000,000 = 0.0001 = 0.01%

SOLUTION OF TASK 1.

- The parameters:
  - CPU clock frequency: 100MHz
  - On average, 10 keys are pressed in a second
  - Minimum time between 2 key presses: 50ms
  - Keyboard status read: 500 clock cycles
  - Interrupt processing overhead: 100 clock cycles

a) How many times do we need to poll the keyboard not to miss any events?
   - 1000ms / 50ms = 20 times per second

b) What is the (relative) load of the CPU devoted to the keyboard when the key press events are handled by polling?
   - 20*500 / 100,000,000 = 0.0001 = 0.01%

c) What is the (relative) load of the CPU devoted to the keyboard when the key press events are handled by interrupts?

SOLUTION OF TASK 1.

- The parameters:
  - CPU clock frequency: 100MHz
  - On average, 10 keys are pressed in a second
  - Minimum time between 2 key presses: 50ms
  - Keyboard status read: 500 clock cycles
  - Interrupt processing overhead: 100 clock cycles

a) How many times do we need to poll the keyboard not to miss any events?
   - 1000ms / 50ms = 20 times per second

b) What is the (relative) load of the CPU devoted to the keyboard when the key press events are handled by polling?
   - 20*500 / 100,000,000 = 0.0001 = 0.01%

c) What is the (relative) load of the CPU devoted to the keyboard when the key press events are handled by interrupts?
   - 10*(500+100) / 100,000,000 = 0.00006 = 0.006%

TASK 2.

Assume that a hard disk drive has 3 double-sided platters. There are 100000 tracks on each recording surface. The disk uses ZBR with two zones: there are 2000 sectors on tracks 1-50000 (zone 1) and 1000 sectors on tracks 50001-100000 (zone 2). The size of the sectors is 500 byte. The speed of the interface of the HDD is $250 \cdot 10^6$ byte/s. The command processing time is 1 ms. The average seek time is 5 ms, the rotation speed is 6000 RPM.

a) What is the total capacity of the HDD in bytes?

b) What is the revolution time of the discs?

c) Compute the service time of a read request of a single sector, if the sector falls onto track 25000 (zone 1).

d) Compute the service time of a read request of a single sector, if the sector falls onto track 75000 (zone 2).

e) On which zone would you put your frequently used data?

TASK 2.

- The parameters of the HDD are:
  - 3 disks, 2 recording surfaces on each
  - 100,000 tracks on each recording surface
  - ZBR with 2 zones:
    - zone 1: 1 – 50,000 tracks: 2000 sectors/track,
    - zone 2: 50,001 – 100,000 tracks: 1000 sectors/track
  - Sector size: 500 bytes
  - Interface speed: $250 \cdot 10^6$ byte/s
  - Command processing time: 1ms
  - Average seek time: 5ms
  - Rotation speed: 6,000RPM (revolutions per minute)

SOLUTION OF TASK 2.

a) The capacity of the disk:

SOLUTION OF TASK 2.

a) The capacity of the disk:
   - 3 disks * 2 surfaces * (50,000*2000 sectors + 50,000*1000 sectors) * 500 bytes = $450 \cdot 10^9$ bytes

SOLUTION OF TASK 2.

a) The capacity of the disk:
   - 3 disks * 2 surfaces * (50,000*2000 sectors + 50,000*1000 sectors) * 500 bytes = $450 \cdot 10^9$ bytes

b) The revolution time of the disk:

SOLUTION OF TASK 2.

a) The capacity of the disk:
   - 3 disks * 2 surfaces * (50,000*2000 sectors + 50,000*1000 sectors) * 500 bytes = $450 \cdot 10^9$ bytes

b) The revolution time of the disk:
   - T = (60,000 ms/min) / (6,000 revolution/min) = 10 ms/revolution

SOLUTION OF TASK 2.

c) Request service times for a sector on track 25,000:

SOLUTION OF TASK 2.

c) Request service times for a sector on track 25,000:
   - Average rotational latency: T/2 = 10ms / 2 = 5ms

SOLUTION OF TASK 2.

c) Request service times for a sector on track 25,000:
   - Average rotational latency: T/2 = 10ms / 2 = 5ms
   - Media transfer time for 1 sector on track 25,000: T / 2000sector = 10ms / 2000 sector = 0.005ms/sector

SOLUTION OF TASK 2.

c) Request service times for a sector on track 25,000:

- Average rotational latency: T/2 = 10ms / 2 = 5ms
- Media transfer time for 1 sector on track 25,000: T / 2000sector = 10ms / 2000 sector = 0.005ms/sector
- Interface data transfer time: (500 bytes/sector) / ($250 \cdot 10^6$bytes/s) = $2 \cdot 10^{-6}$ s/sector = 0.002ms/sector

SOLUTION OF TASK 2.

c) Request service times for a sector on track 25,000:

- Average rotational latency: T/2 = 10ms / 2 = 5ms
- Media transfer time for 1 sector on track 25,000: T / 2000sector = 10ms / 2000 sector = 0.005ms/sector
- Interface data transfer time: (500 bytes/sector) / ($250 \cdot 10^6$bytes/s) = $2 \cdot 10^{-6}$ s/sector = 0.002ms/sector
- Seek time is given: 5ms, command processing time is given: 1ms

SOLUTION OF TASK 2.

c) Request service times for a sector on track 25,000:

- Average rotational latency: T/2 = 10ms / 2 = 5ms
- Media transfer time for 1 sector on track 25,000: T / 2000sector = 10ms / 2000 sector = 0.005ms/sector
- Interface data transfer time: (500 bytes/sector) / ($250 \cdot 10^6$bytes/s) = $2 \cdot 10^{-6}$ s/sector = 0.002ms/sector
- Seek time is given: 5ms, command processing time is given: 1ms
- Service time for a single sector

SOLUTION OF TASK 2.

c) Request service times for a sector on track 25,000:

- Average rotational latency: T/2 = 10ms / 2 = 5ms
- Media transfer time for 1 sector on track 25,000: T / 2000sector = 10ms / 2000 sector = 0.005ms/sector
- Interface data transfer time: (500 bytes/sector) / ($250 \cdot 10^6$bytes/s) = $2 \cdot 10^{-6}$ s/sector = 0.002ms/sector
- Seek time is given: 5ms, command processing time is given: 1ms
- Service time for a single sector
  - 1 + 5 + 5 + 0.005 + 0.002 = **11.007ms**

SOLUTION OF TASK 2.

c) Request service times for a sector on track 75,000:

- Average rotational latency: T/2 = 10ms / 2 = 5ms
- Media transfer time for 1 sector on track 75,000: T / 1000sector = 10ms / 1000 sector = 0.01ms/sector
- Interface data transfer time: (500 bytes/sector) / ($250 \cdot 10^6$bytes/s) = $2 \cdot 10^{-6}$ s/sector = 0.002 ms/sector
- Seek time is given: 5ms, command processing time is given: 1ms
- Service time for a single sector
  - 1 + 5 + 5 + 0.01 + 0.002 = **11.012ms**

SOLUTION OF TASK 2.

d) On which zone would you put your frequently used data?

SOLUTION OF TASK 2.

d) On which zone would you put your frequently used data?

- Zone 25,000: **11.007ms**

SOLUTION OF TASK 2.

d) On which zone would you put your frequently used data?

- Zone 25,000: **11.007ms**
- Zone 75,000: **11.012ms**

SOLUTION OF TASK 2.

d) On which zone would you put your frequently used data?

- Zone 25,000: **11.007ms**
- Zone 75,000: **11.012ms**
- The difference is negligible!

SOLUTION OF TASK 2.

d) On which zone would you put your frequently used data?

- Zone 25,000: **11.007ms**
- Zone 75,000: **11.012ms**
- The difference is negligible!
- The costs of putting the frequently used data into the somewhat faster zone is likely higher than the gain.

TASK 3.

Assume that a hard disk drive has 3 double-sided platters. There are 20,000 tracks on each recording surface, with 1000 sectors on each track (there is no ZBR). The sector size is 500byte. The average seek time is 4ms. By measurements, we have determined that the average service time of read requests aiming a single sector is 10ms.

a) What is the rotation speed (in RPM) if we assume that the data transmission through the interface and command processing are infinitely fast?

b) How long does it take to read a sector from the recording surface, once the head and the disk are at the appropriate position?

c) Assume that the command processing time is 0.1ms and the transmission speed of the interface is $50 \cdot 10^6$byte/s. What is the (random) throughput measured by 2000byte long read requests? What is the (sequential) throughput measured by $50 \cdot 10^6$byte long read requests?

SOLUTION OF TASK 3.

- The parameters of the HDD are:
  - 3 disks, 2 recording surfaces on each
  - 20,000 tracks on each recording surface
  - No ZBR, 1000sectors/track
  - Sector size: 500bytes
  - Interface speed: $250 \cdot 10^6$byte/s
  - Average seek time: 4ms
  - Average service time of a single sector read request: 10ms
  - Command processing time: negligible
  - Interface transmission time: negligible
  - Rotation speed is to be determined!

SOLUTION OF TASK 3.

a) What is the rotation speed (in RPM) if we assume that the data transmission through the interface and command processing are infinitely fast?

SOLUTION OF TASK 3.

a) What is the rotation speed (in RPM) if we assume that the data transmission through the interface and command processing are infinitely fast?
   - Let us denote the time of a single rotation by T.

SOLUTION OF TASK 3.

a) What is the rotation speed (in RPM) if we assume that the data transmission through the interface and command processing are infinitely fast?

- Let us denote the time of a single rotation by T.

  0 + 4ms + T/2 + T/1000 + 0 = 10ms

SOLUTION OF TASK 3.

a) What is the rotation speed (in RPM) if we assume that the data transmission through the interface and command processing are infinitely fast?

- Let us denote the time of a single rotation by T.

$0 + 4\text{ms} + T/2 + T/1000 + 0 = 10\text{ms}$

$T/2 + T/1000 = 6\text{ms}$

SOLUTION OF TASK 3.

a) What is the rotation speed (in RPM) if we assume that the data transmission through the interface and command processing are infinitely fast?

- Let us denote the time of a single rotation by T.

$$0 + 4\text{ms} + T/2 + T/1000 + 0 = 10\text{ms}$$

$$T/2 + T/1000 = 6\text{ms}$$

$$500T + T = 6000\text{ms}$$

SOLUTION OF TASK 3.

a) What is the rotation speed (in RPM) if we assume that the data transmission through the interface and command processing are infinitely fast?

- Let us denote the time of a single rotation by T.

$$0 + 4\text{ms} + T/2 + T/1000 + 0 = 10\text{ms}$$

$$T/2 + T/1000 = 6\text{ms}$$

$$500T + T = 6000\text{ms}$$

$$501T = 6000\text{ms}$$

SOLUTION OF TASK 3.

a) What is the rotation speed (in RPM) if we assume that the data transmission through the interface and command processing are infinitely fast?

- Let us denote the time of a single rotation by T.

$$0 + 4\text{ms} + T/2 + T/1000 + 0 = 10\text{ms}$$

$$T/2 + T/1000 = 6\text{ms}$$

$$500T + T = 6000\text{ms}$$

$$501T = 6000\text{ms}$$

$$T = 6000/501\text{ms} \approx 11.976\text{ms}$$

SOLUTION OF TASK 3.

a) What is the rotation speed (in RPM) if we assume that the data transmission through the interface and command processing are infinitely fast?

- Let us denote the time of a single rotation by T.

$$0 + 4\text{ms} + T/2 + T/1000 + 0 = 10\text{ms}$$

$$T/2 + T/1000 = 6\text{ms}$$

$$500T + T = 6000\text{ms}$$

$$501T = 6000\text{ms}$$

$$T = 6000/501\text{ms} \approx 11.976\text{ms}$$

The rotation speed (in RPM) is:

SOLUTION OF TASK 3.

a) What is the rotation speed (in RPM) if we assume that the data transmission through the interface and command processing are infinitely fast?

- Let us denote the time of a single rotation by T.

$$0 + 4\text{ms} + T/2 + T/1000 + 0 = 10\text{ms}$$

$$T/2 + T/1000 = 6\text{ms}$$

$$500T + T = 6000\text{ms}$$

$$501T = 6000\text{ms}$$

$$T = 6000/501\text{ms} \approx 11.976\text{ms}$$

The rotation speed (in RPM) is:

60,000ms/minute / 6000/501ms/rotation = **5010 rotation/minute**

SOLUTION OF TASK 3.

b) How long does it take to read a sector from the recording surface, once the head and the disk are at the appropriate position?

SOLUTION OF TASK 3.

b) How long does it take to read a sector from the recording surface, once the head and the disk are at the appropriate position?

- The reading time of a single sector is:

  6000/501ms / 1000 = 6/501ms ≈ 0.011976ms

SOLUTION OF TASK 3.

c) Assume that the command processing time is 0.1 ms and the transmission speed of the interface is $50 \cdot 10^6$ byte/s.
   - What is the (random) throughput measured by 2000 byte long read requests?

SOLUTION OF TASK 3.

c) Assume that the command processing time is 0.1 ms and the transmission speed of the interface is $50 \cdot 10^6$ byte/s.
- What is the (random) throughput measured by 2000 byte long read requests?

- First, we calculate the transmission time of a single sector:

  500byte/sector / $50 \cdot 10^6$ byte/s = 10^{-5}s = 0.01ms

SOLUTION OF TASK 3.

c) Assume that the command processing time is 0.1 ms and the transmission speed of the interface is $50 \cdot 10^6$ byte/s.
- What is the (random) throughput measured by 2000 byte long read requests?

- First, we calculate the transmission time of a single sector:

  500byte/sector / $50 \cdot 10^6$ byte/s = 10^{-5}s = 0.01ms
- As this value is less then the reading time of a sector (0.011976ms), they can be overlapped!

SOLUTION OF TASK 3.

c) Assume that the command processing time is 0.1 ms and the transmission speed of the interface is $50 \cdot 10^6$ byte/s.
- What is the (random) throughput measured by 2000 byte long read requests?

- First, we calculate the transmission time of a single sector:

  500byte/sector / $50 \cdot 10^6$ byte/s = 10^{-5}s = 0.01ms
- As this value is less then the reading time of a sector (0.011976ms), they can be overlapped!
- We need to read and transmit 2000byte / 500byte/sector = 4 sectors

SOLUTION OF TASK 3.

c) Assume that the command processing time is 0.1 ms and the transmission speed of the interface is $50 \cdot 10^6$ byte/s.
- What is the (random) throughput measured by 2000 byte long read requests?

- First, we calculate the transmission time of a single sector:

  500byte/sector / $50 \cdot 10^6$ byte/s = 10^{-5}s = 0.01ms
- As this value is less then the reading time of a sector (0.011976ms), they can be overlapped!
- We need to read and transmit 2000byte / 500byte/sector = 4 sectors
- We calculate with the reading time of 4 sectors and transmission time of the last sector.

SOLUTION OF TASK 3.

c) Assume that the command processing time is 0.1 ms and the transmission speed of the interface is $50 \cdot 10^6$ byte/s.
- What is the (random) throughput measured by 2000 byte long read requests?

- First, we calculate the transmission time of a single sector:

  500byte/sector / $50 \cdot 10^6$ byte/s = 10^{-5}s = 0.01ms
- As this value is less then the reading time of a sector (0.011976ms), they can be overlapped!
- We need to read and transmit 2000byte / 500byte/sector = 4 sectors
- We calculate with the reading time of 4 sectors and transmission time of the last sector.
- The service time is:

  0.1ms + 4ms + 6000/501ms / 2 + 4*0.011976ms +0.01ms = 10.146ms

SOLUTION OF TASK 3.

c) Assume that the command processing time is 0.1 ms and the transmission speed of the interface is $50 \cdot 10^6$ byte/s.
- What is the (random) throughput measured by 2000 byte long read requests?

- First, we calculate the transmission time of a single sector:

  500byte/sector / $50 \cdot 10^6$ byte/s = 10^{-5}s = 0.01ms
- As this value is less then the reading time of a sector (0.011976ms), they can be overlapped!
- We need to read and transmit 2000byte / 500byte/sector = 4 sectors
- We calculate with the reading time of 4 sectors and transmission time of the last sector.
- The service time is:

  0.1ms + 4ms + 6000/501ms / 2 + 4*0.011976ms +0.01ms = 10.146ms
- Thus the disk is able to process 1000ms/10.146ms=98.561 requests per second (IOPS), therefore, the data transmission speed is: 98.561reques/s * 2000byte/request = **197122 byte/s**

SOLUTION OF TASK 3.

c) Assume that the command processing time is 0.1ms and the transmission speed of the interface is $50 \cdot 10^6$ byte/s.
   - What is the (sequential) throughput measured by $50 \cdot 10^6$ byte long read requests?

SOLUTION OF TASK 3.

c) Assume that the command processing time is 0.1ms and the transmission speed of the interface is $50 \cdot 10^6$ byte/s.
   - What is the (sequential) throughput measured by $50 \cdot 10^6$ byte long read requests?
     - The calculation is very similar, just the numbers are different:

SOLUTION OF TASK 3.

c) Assume that the command processing time is 0.1ms and the transmission speed of the interface is $50 \cdot 10^6$ byte/s.
- What is the (sequential) throughput measured by $50 \cdot 10^6$ byte long read requests?

  - The calculation is very similar, just the numbers are different:
  - We need to read and transmit $50 \cdot 10^6$ byte / 500byte/sector = 10^5 sectors

SOLUTION OF TASK 3.

c) Assume that the command processing time is 0.1ms and the transmission speed of the interface is $50 \cdot 10^6$ byte/s.
- What is the (sequential) throughput measured by $50 \cdot 10^6$ byte long read requests?

- The calculation is very similar, just the numbers are different:
- We need to read and transmit $50 \cdot 10^6$ byte / 500byte/sector = 10^5 sectors
- We calculate with the reading time of 10^5 sectors and transmission time of the last sector.

SOLUTION OF TASK 3.

c) Assume that the command processing time is 0.1ms and the transmission speed of the interface is $50 \cdot 10^6$ byte/s.
- What is the (sequential) throughput measured by $50 \cdot 10^6$ byte long read requests?

- The calculation is very similar, just the numbers are different:
- We need to read and transmit $50 \cdot 10^6$ byte / 500byte/sector = 10^5 sectors
- We calculate with the reading time of 10^5 sectors and transmission time of the last sector.
- The service time is:

  0.1ms + 4ms + 6000/501ms / 2 + 10^5*0.011976ms +0.01ms = 1207.7ms

SOLUTION OF TASK 3.

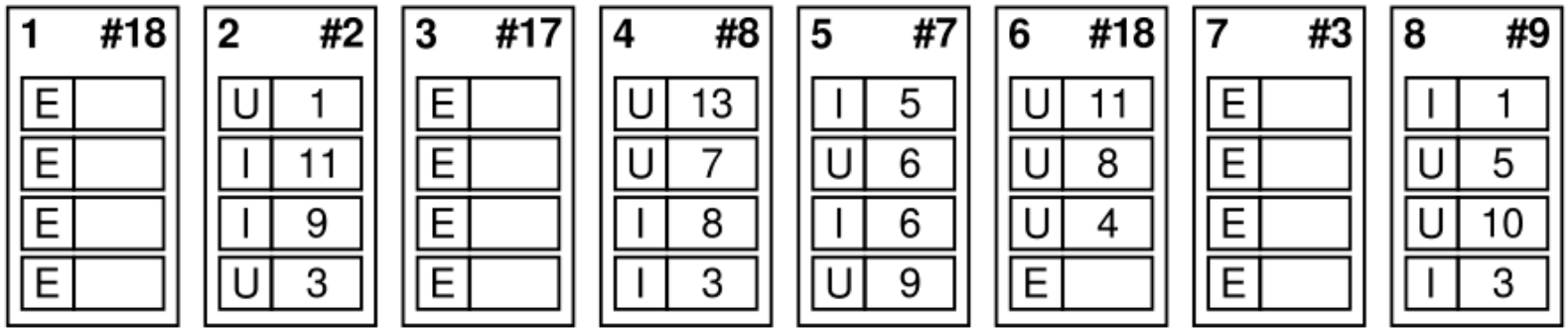
c) Assume that the command processing time is 0.1ms and the transmission speed of the interface is $50 \cdot 10^6$ byte/s.
- What is the (sequential) throughput measured by $50 \cdot 10^6$ byte long read requests?

- The calculation is very similar, just the numbers are different:
- We need to read and transmit $50 \cdot 10^6$ byte / 500byte/sector = 10^5 sectors
- We calculate with the reading time of 10^5 sectors and transmission time of the last sector.
- The service time is:

  0.1ms + 4ms + 6000/501ms / 2 + 10^5*0.011976ms +0.01ms = 1207.7ms
- Thus the disk is able to process 1000ms/1207.7ms=0.828 requests per second (IOPS), therefore, the data transmission speed is: 0.828reques/s * $50 \cdot 10^6$byte/request = **41.4 ·10⁶ byte/s**

TASK 4.

- The current state of an SSD consisting of 8 blocks is depicted in the following figure.



The top left corner of the blocks refer to the number of the block, and the top right corner of the blocks contain the number of programming/erase cycles. Each block consists of 4 pages. For each page, the state of the page (”U”=used, ”I”=invalid, ”E”=erased) and the LBA address of the stored data are provided.

In the initial state, blocks 1, 3 and 7 are erased, and the write frontier is block 6.

TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

1 #18	
E	
E	
E	
E	

2 #2	
U	1
I	11
I	9
U	3

3 #17	
E	
E	
E	
E	

4 #8	
U	13
U	7
I	8
I	3

5 #7	
I	5
U	6
I	6
U	9

6 #18	
U	11
U	8
U	4
E	

7 #3	
E	
E	
E	
E	

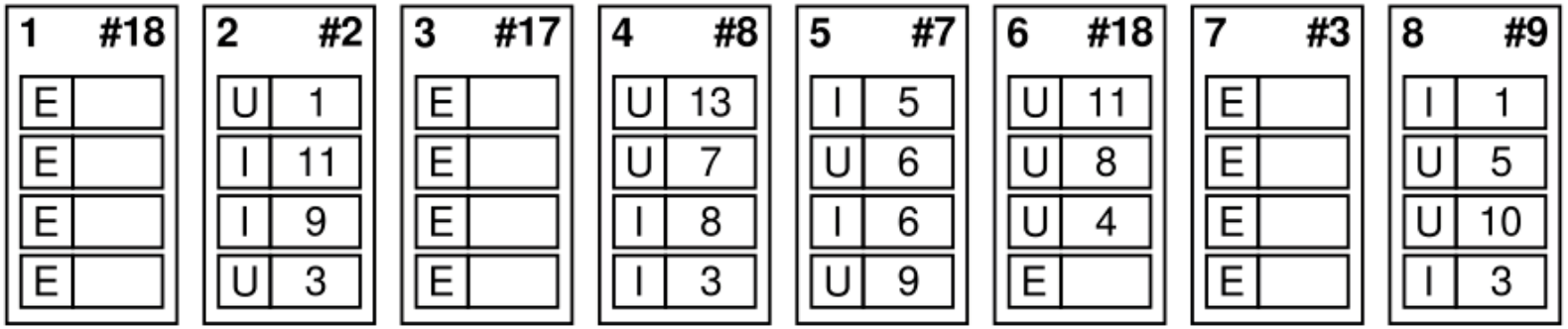
8 #9	
I	1
U	5
U	10
I	3

a) How does the state of the SSD change when write requests arrive to pages 5, 13 and 2, in this order?
If a new write frontier is needed the SSD should select the one that leads to the most balanced wearing of the blocks!

b) How does the state of the SSD change when the garbage collection algorithm is triggered in the initial state, and it does not stop till an extra erased block is created (thus it should have 4 of them)? The garbage collection algorithm should always pick the block with the most invalid pages, in case there are more options, it picks the one with the lower P/E counter value.

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



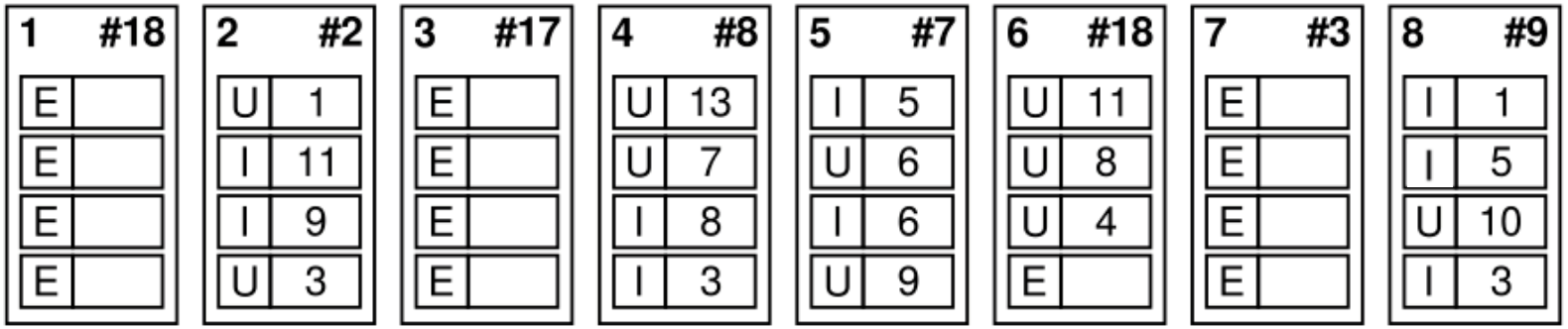
a) How does the state of the SSD change when write requests arrive to pages 5, 13 and 2, in this order?

Page 5:

- it is in block 8, we mark it as invalid

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



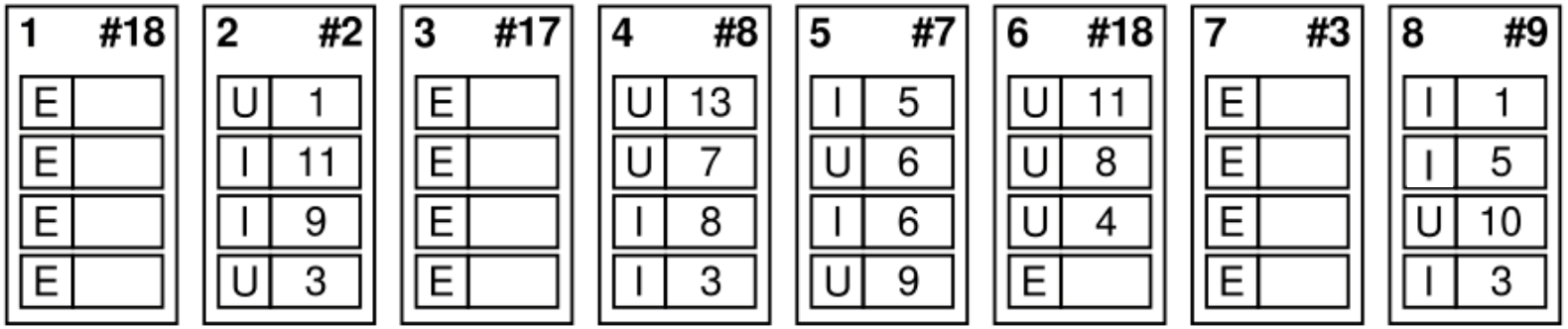
a) How does the state of the SSD change when write requests arrive to pages 5, 13 and 2, in this order?

Page 5:

- it is in block 8, we mark it as invalid

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



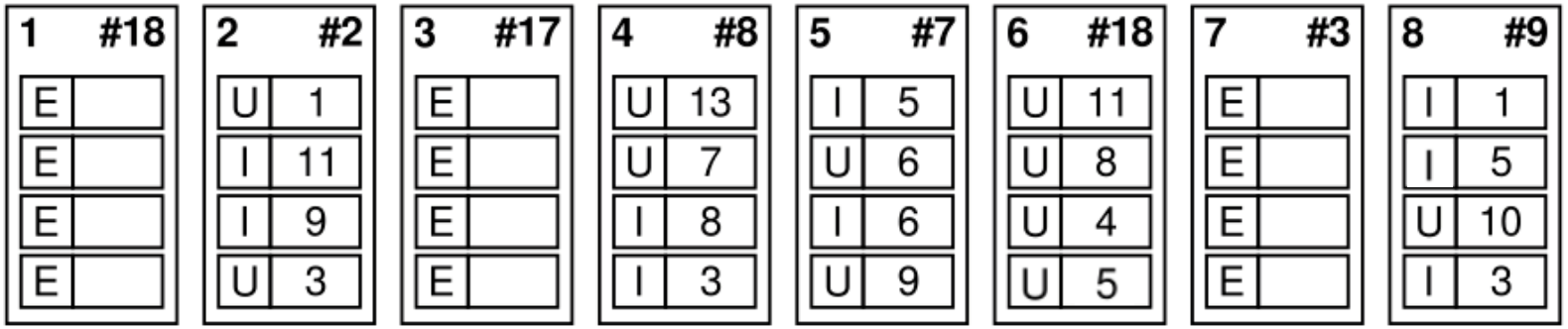
a) How does the state of the SSD change when write requests arrive to pages 5, 13 and 2, in this order?

Page 5:

- it is in block 8, we mark it as invalid
- write frontier is block 6, we use its last page

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



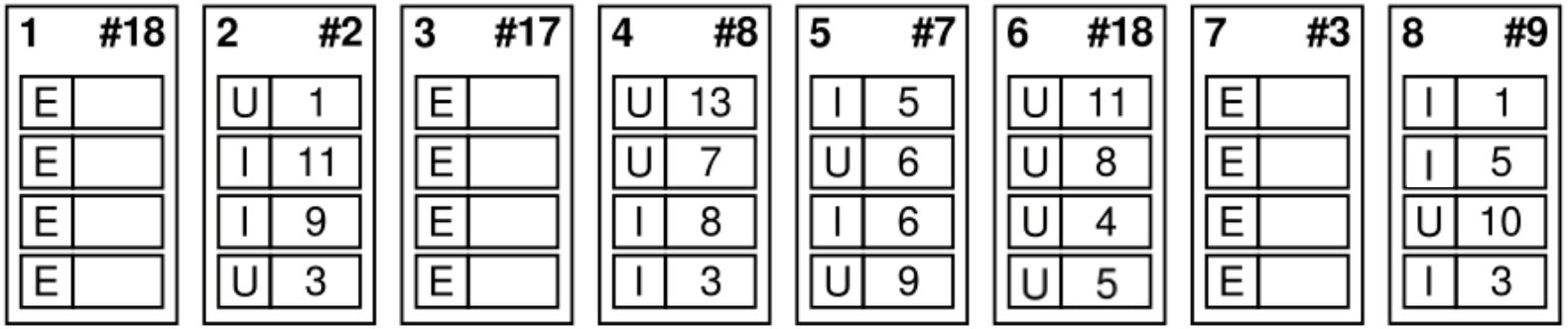
a) How does the state of the SSD change when write requests arrive to pages 5, 13 and 2, in this order?

Page 5:

- it is in block 8, we mark it as invalid
- write frontier is block 6, we use its last page

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



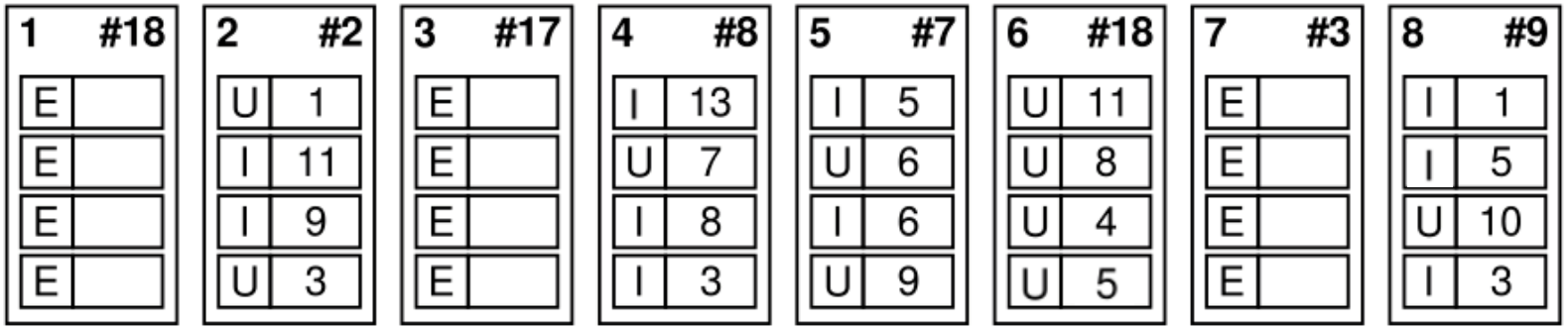
a) How does the state of the SSD change when write requests arrive to pages 5, 13 and 2, in this order?

Page 13:

- it is in block 4, we mark it as invalid

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



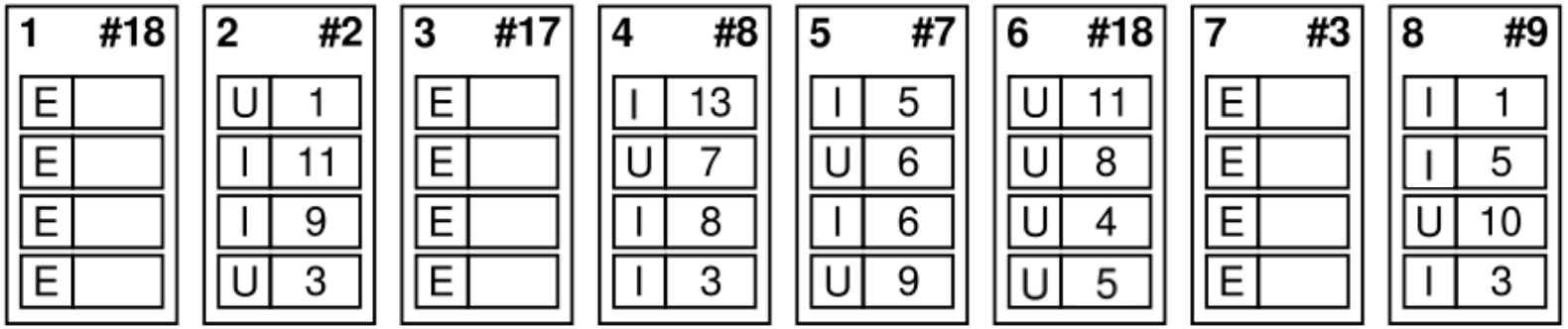
a) How does the state of the SSD change when write requests arrive to pages 5, 13 and 2, in this order?

Page 13:

- it is in block 4, we mark it as invalid

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



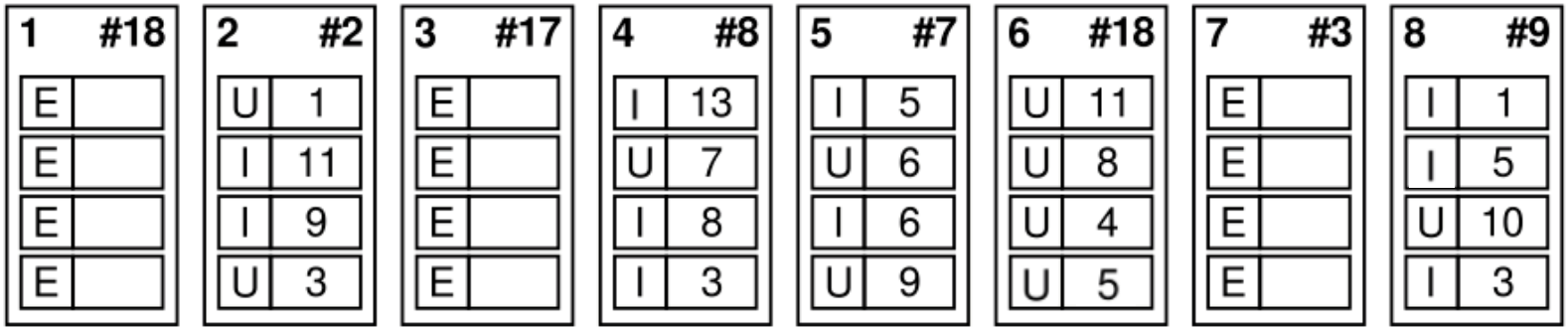
a) How does the state of the SSD change when write requests arrive to pages 5, 13 and 2, in this order?

Page 13:

- it is in block 4, we mark it as invalid
- write frontier is full, we select from erased blocks: 1, 3, 7
  block 7 has the lowest deletion count, we select it

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



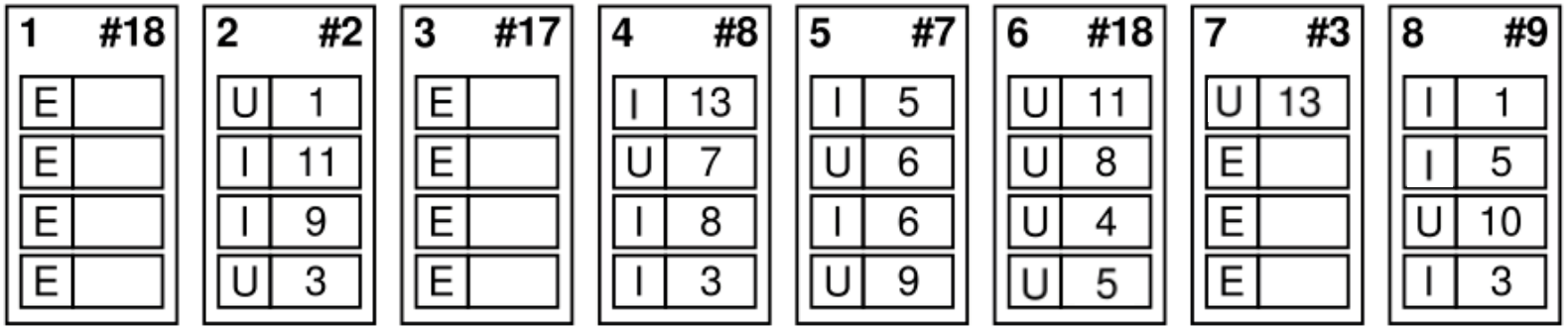
a) How does the state of the SSD change when write requests arrive to pages 5, 13 and 2, in this order?

Page 13:

- it is in block 4, we mark it as invalid
- write frontier is full, we select from erased blocks: 1, 3, 7
  block 7 has the lowest deletion count, we select it
- we write page 13 into its very first page

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



a) How does the state of the SSD change when write requests arrive to pages 5, 13 and 2, in this order?

Page 13:

- it is in block 4, we mark it as invalid
- write frontier is full, we select from erased blocks: 1, 3, 7
  block 7 has the lowest deletion count, we select it
- we write page 13 into its very first page

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

1 #18	
E	
E	
E	
E	

2 #2	
U	1
I	11
I	9
U	3

3 #17	
E	
E	
E	
E	

4 #8	
I	13
U	7
I	8
I	3

5 #7	
I	5
U	6
I	6
U	9

6 #18	
U	11
U	8
U	4
U	5

7 #3	
U	13
E	
E	
E	

8 #9	
I	1
I	5
U	10
I	3

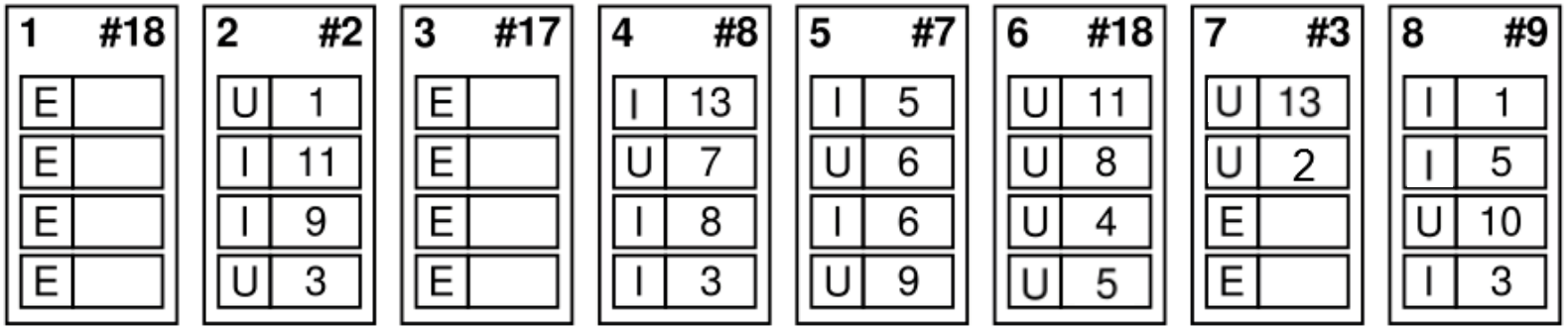
a) How does the state of the SSD change when write requests arrive to pages 5, 13 and 2, in this order?

Page 2:

- it has not been written on the SSD yet, we do not have to invalidate
- we write it on the next free page of the write frontier

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



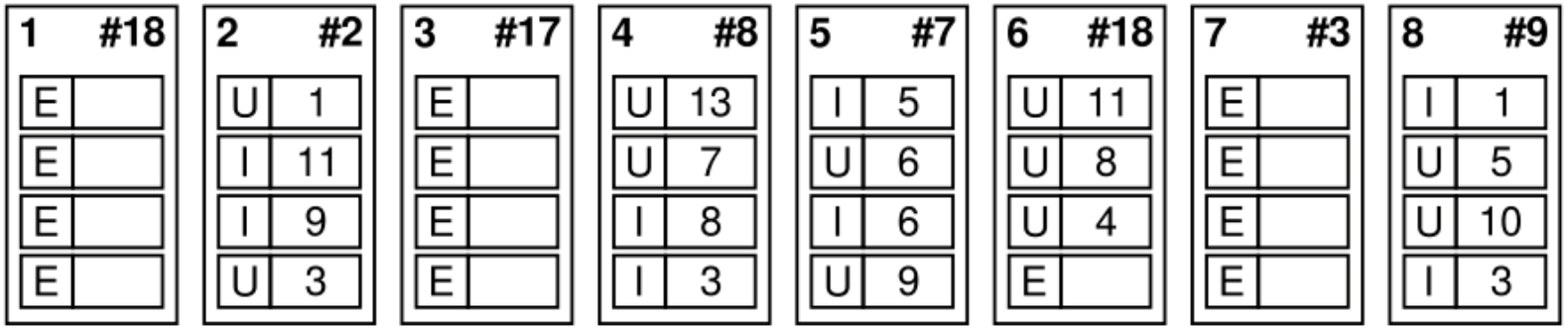
a) How does the state of the SSD change when write requests arrive to pages 5, 13 and 2, in this order?

Page 2:

- it has not been written on the SSD yet, we do not have to invalidate
- we write it on the next free page of the write frontier

SOLUTION OF TASK 4.

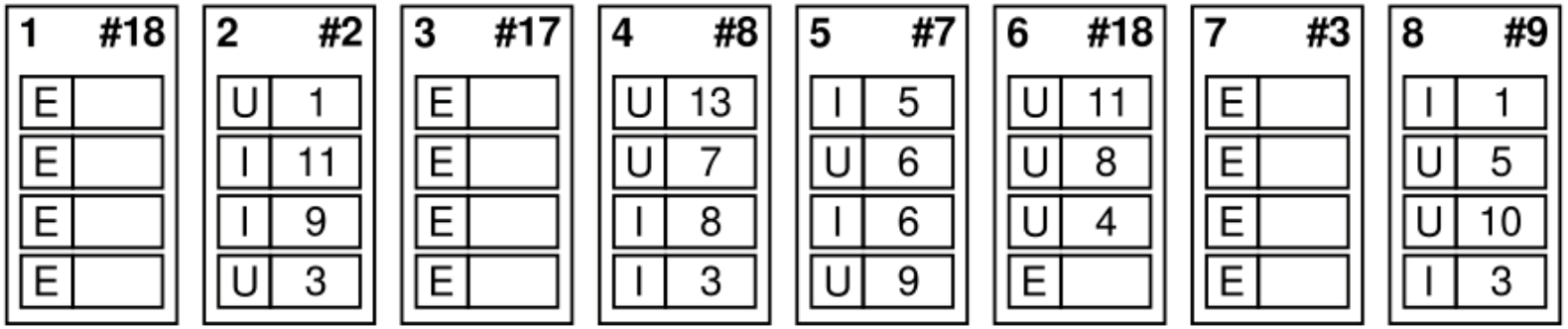
- The state of an SSD consisting of 8 blocks is as follows



b) How does the state of the SSD change when the garbage collection algorithm is triggered in the initial state, and it does not stop till an extra erased block is created (thus it should have 4 of them)? The garbage collection algorithm should always pick the block with the most invalid pages, in case there are more options, it picks the one with the lower P/E counter value.

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

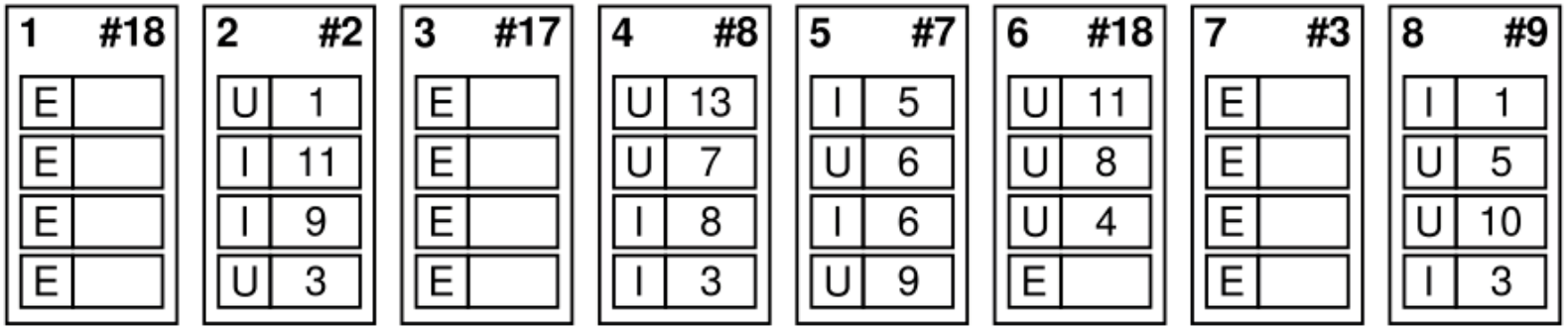


b) Garbage collection:

- There are 4 blocks having each 2 invalid pages, blocks: 2, 4, 5 and 8

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

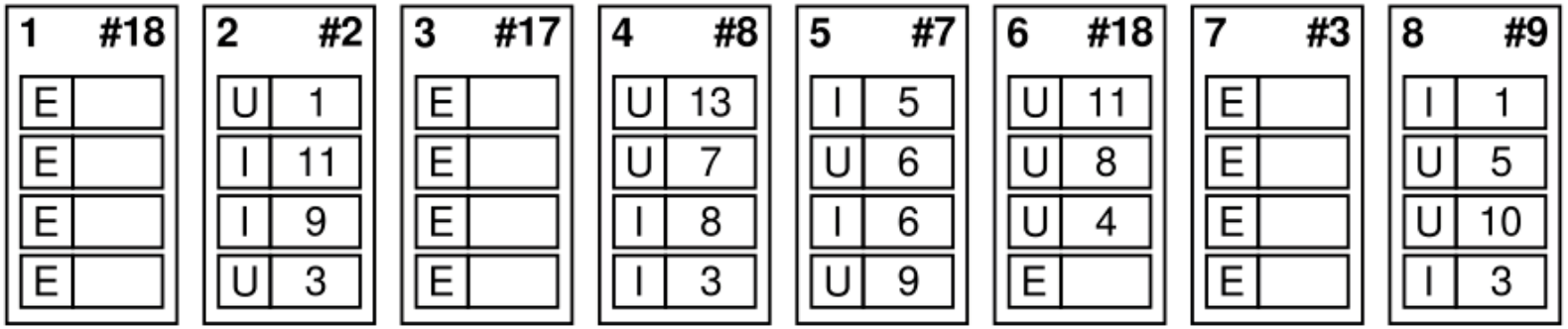


b) Garbage collection:

- There are 4 blocks having each 2 invalid pages, blocks: 2, 4, 5 and 8
- from among them, block 2 has the lowest age → we will delete it.

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

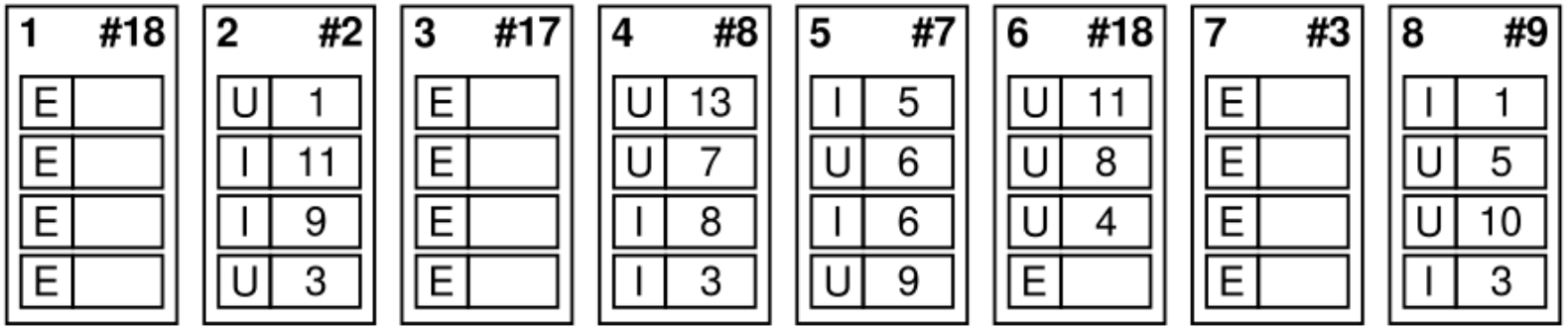


b) Garbage collection:

- There are 4 blocks having each 2 invalid pages, blocks: 2, 4, 5 and 8
- from among them, block 2 has the lowest age → we will delete it.
- to that end, we first have to save its used blocks

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

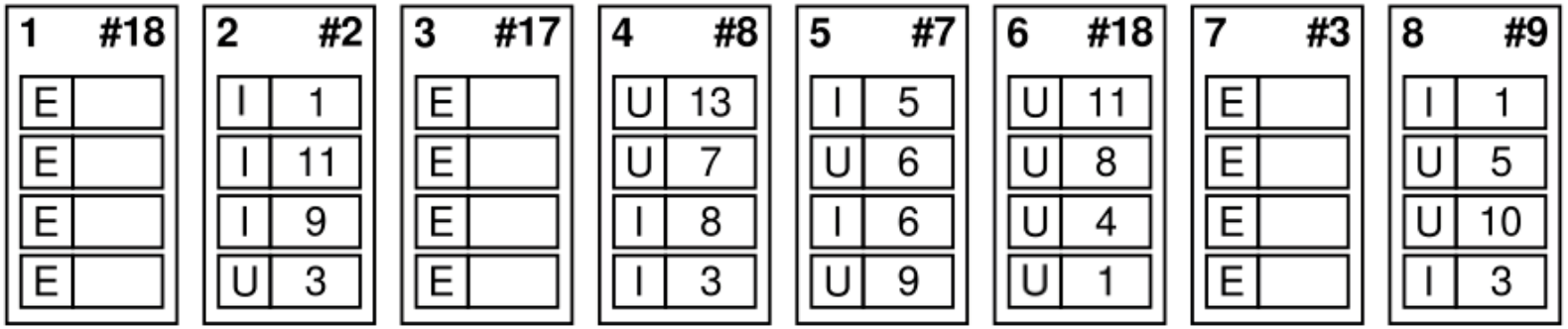


b) Garbage collection:

- There are 4 blocks having each 2 invalid pages, blocks: 2, 4, 5 and 8
- from among them, block 2 has the lowest age → we will delete it.
- to this end, we first have to save its used blocks
  - we copy page 1 to the current write frontier

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



b) Garbage collection:

- There are 4 blocks having each 2 invalid pages, blocks: 2, 4, 5 and 8
- from among them, block 2 has the lowest age → we will delete it.
- to this end, we first have to save its used blocks
  - we copy page 1 to the current write frontier

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

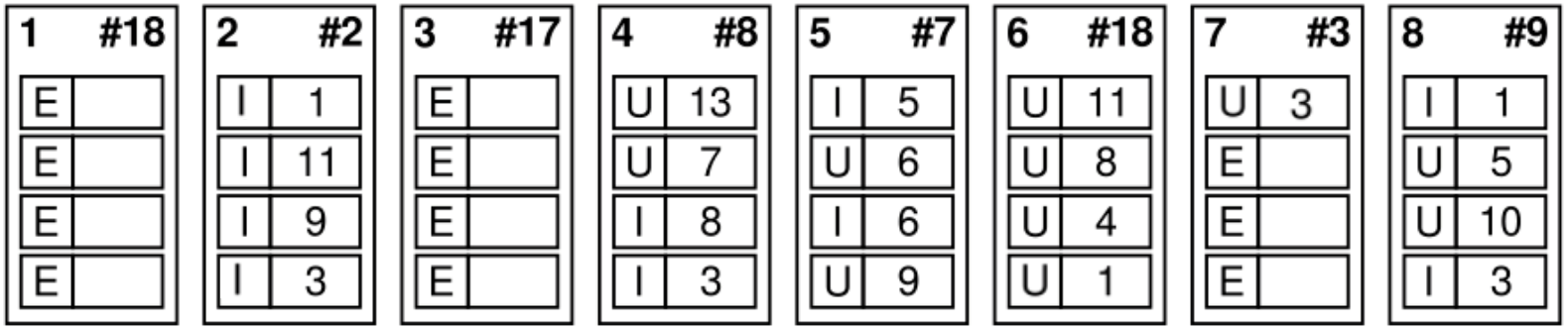
1 #18		2 #2		3 #17		4 #8		5 #7		6 #18		7 #3		8 #9	
E		I	1	E		U	13	I	5	U	11	E		I	1
E		I	11	E		U	7	U	6	U	8	E		U	5
E		I	9	E		I	8	I	6	U	4	E		U	10
E		U	3	E		I	3	U	9	U	1	E		I	3

b) Garbage collection:

- There are 4 blocks having each 2 invalid pages, blocks: 2, 4, 5 and 8
- from among them, block 2 has the lowest age → we will delete it.
- to this end, we first have to save its used blocks
  - we copy page 1 to the current write frontier
  - we copy page 3 to the new write frontier, that is into block 7

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



b) Garbage collection:

- There are 4 blocks having each 2 invalid pages, blocks: 2, 4, 5 and 8
- from among them, block 2 has the lowest age → we will delete it.
- to this end, we first have to save its used blocks
  - we copy page 1 to the current write frontier
  - we copy page 3 to the new write frontier, that is into block 7

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

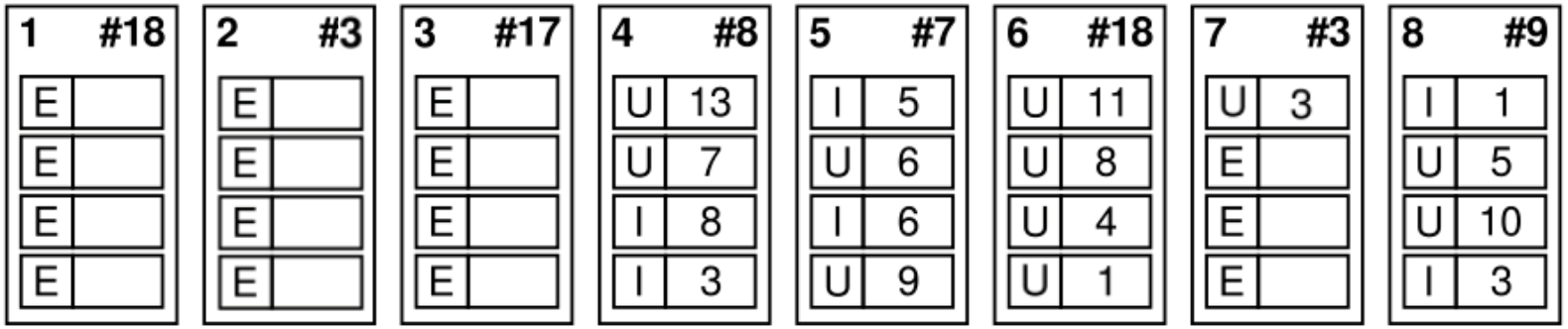
1 #18		2 #2		3 #17		4 #8		5 #7		6 #18		7 #3		8 #9	
E		I	1	E		U	13	I	5	U	11	U	3	I	1
E		I	11	E		U	7	U	6	U	8	E		U	5
E		I	9	E		I	8	I	6	U	4	E		U	10
E		I	3	E		I	3	U	9	U	1	E		I	3

b) Garbage collection:

- There are 4 blocks having each 2 invalid pages, blocks: 2, 4, 5 and 8
- from among them, block 2 has the lowest age → we will delete it.
- to this end, we first have to save its used blocks
  - we copy page 1 to the current write frontier
  - we copy page 3 to the new write frontier, that is into block 7
- Now we erase block 2.

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



b) Garbage collection:

- There are 4 blocks having each 2 invalid pages, blocks: 2, 4, 5 and 8
- from among them, block 2 has the lowest age → we will delete it.
- to this end, we first have to save its used blocks
  - we copy page 1 to the current write frontier
  - we copy page 3 to the new write frontier, that is into block 7
- Now we erase block 2. But we still have only 3 erased blocks. :-(

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

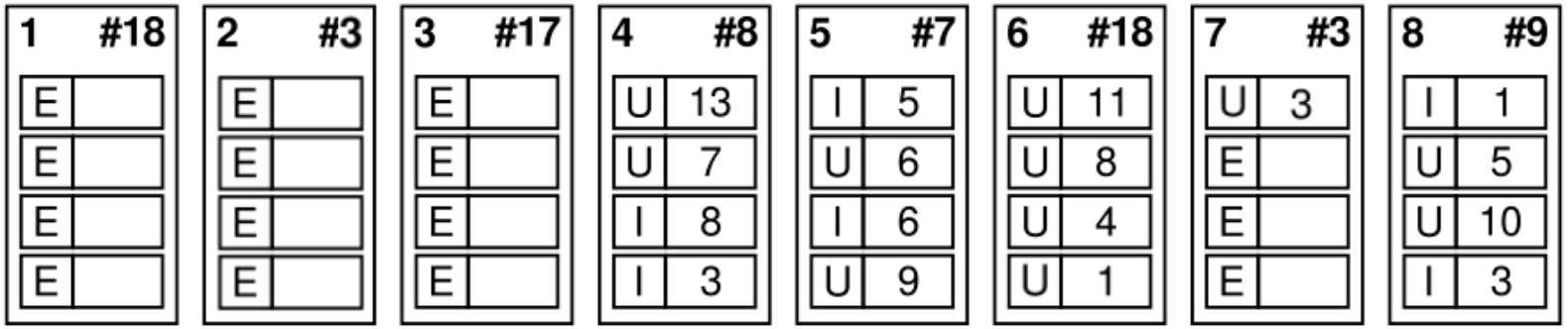
1 #18		2 #3		3 #17		4 #8		5 #7		6 #18		7 #3		8 #9	
E		E		E		U	13	I	5	U	11	U	3	I	1
E		E		E		U	7	U	6	U	8	E		U	5
E		E		E		I	8	I	6	U	4	E		U	10
E		E		E		I	3	U	9	U	1	E		I	3

b) Garbage collection must be continued on:

- There are 3 blocks having each 2 invalid pages, blocks: 4, 5 and 8

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



b) Garbage collection must be continued on:

- There are 3 blocks having each 2 invalid pages, blocks: 4, 5 and 8
- from among them, block 5 has the lowest age → we will delete it.

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

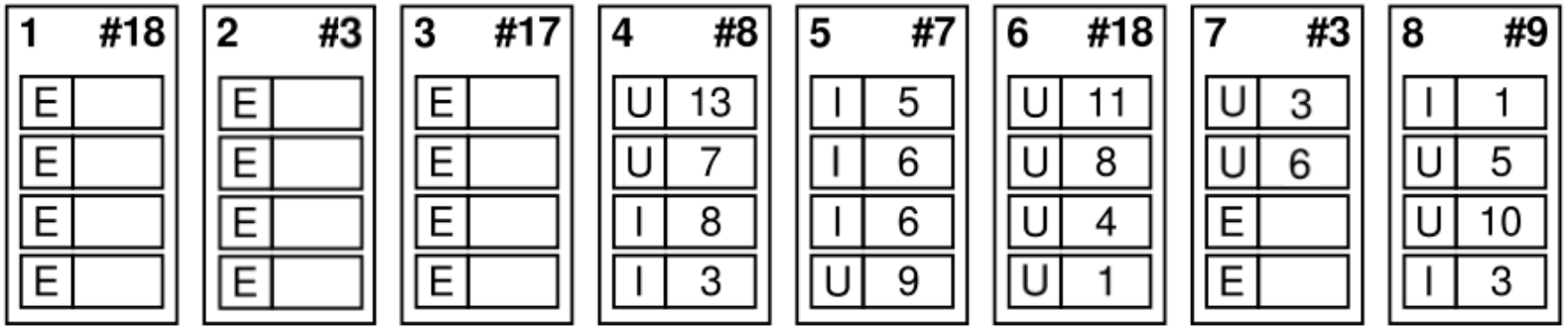
1 #18		2 #3		3 #17		4 #8		5 #7		6 #18		7 #3		8 #9	
E		E		E		U	13	I	5	U	11	U	3	I	1
E		E		E		U	7	U	6	U	8	E		U	5
E		E		E		I	8	I	6	U	4	E		U	10
E		E		E		I	3	U	9	U	1	E		I	3

b) Garbage collection must be continued on:

- There are 3 blocks having each 2 invalid pages, blocks: 4, 5 and 8
- from among them, block 5 has the lowest age → we will delete it.
- to this end, we first have to save its used blocks
  - we copy page 6 to the current write frontier

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

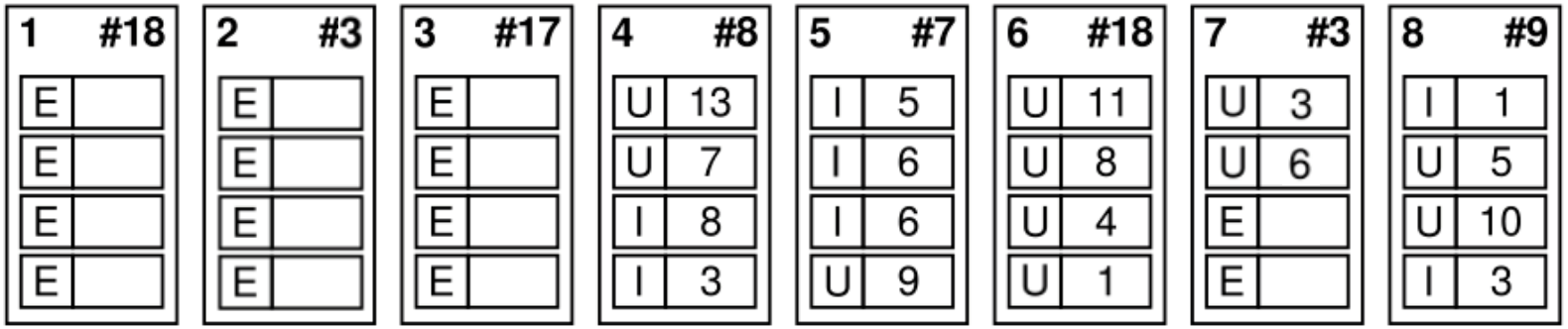


b) Garbage collection must be continued on:

- There are 3 blocks having each 2 invalid pages, blocks: 4, 5 and 8
- from among them, block 5 has the lowest age → we will delete it.
- to this end, we first have to save its used blocks
  - we copy page 6 to the current write frontier

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

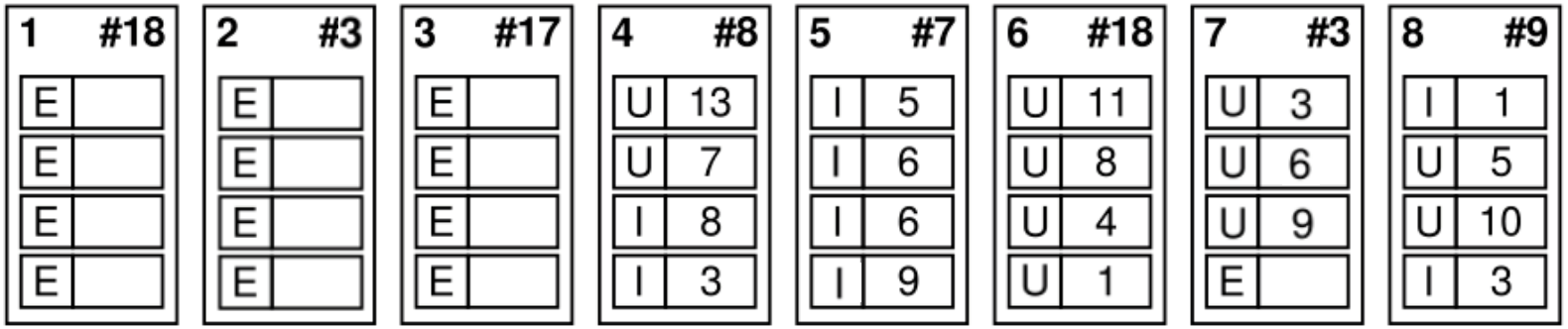


b) Garbage collection must be continued on:

- There are 3 blocks having each 2 invalid pages, blocks: 4, 5 and 8
- from among them, block 5 has the lowest age → we will delete it.
- to this end, we first have to save its used blocks
  - we copy page 6 to the current write frontier
  - we copy page 9 to the current write frontier

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

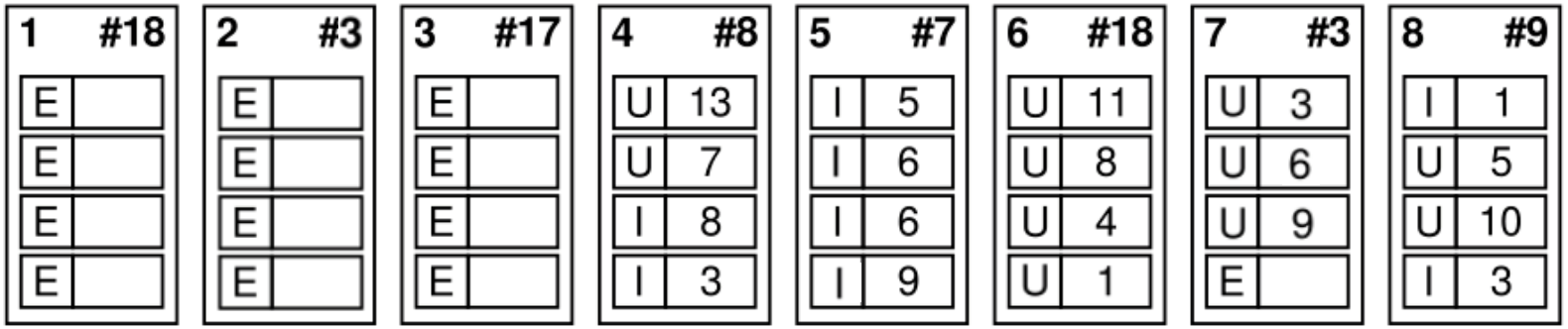


b) Garbage collection must be continued on:

- There are 3 blocks having each 2 invalid pages, blocks: 4, 5 and 8
- from among them, block 5 has the lowest age → we will delete it.
- to this end, we first have to save its used blocks
  - we copy page 6 to the current write frontier
  - we copy page 9 to the current write frontier

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

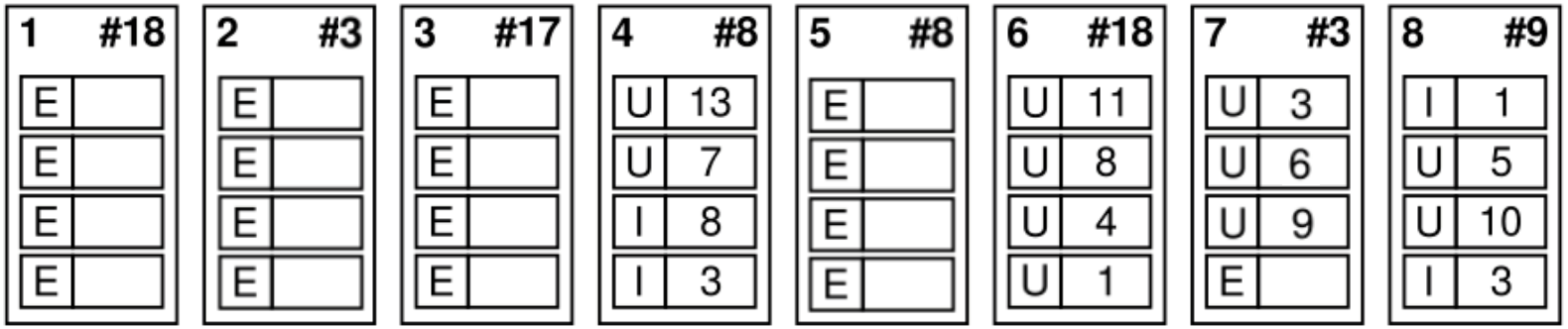


b) Garbage collection must be continued on:

- There are 3 blocks having each 2 invalid pages, blocks: 4, 5 and 8
- from among them, block 5 has the lowest age → we will delete it.
- to this end, we first have to save its used blocks
  - we copy page 6 to the current write frontier
  - we copy page 9 to the current write frontier
- Erase block 5.

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows

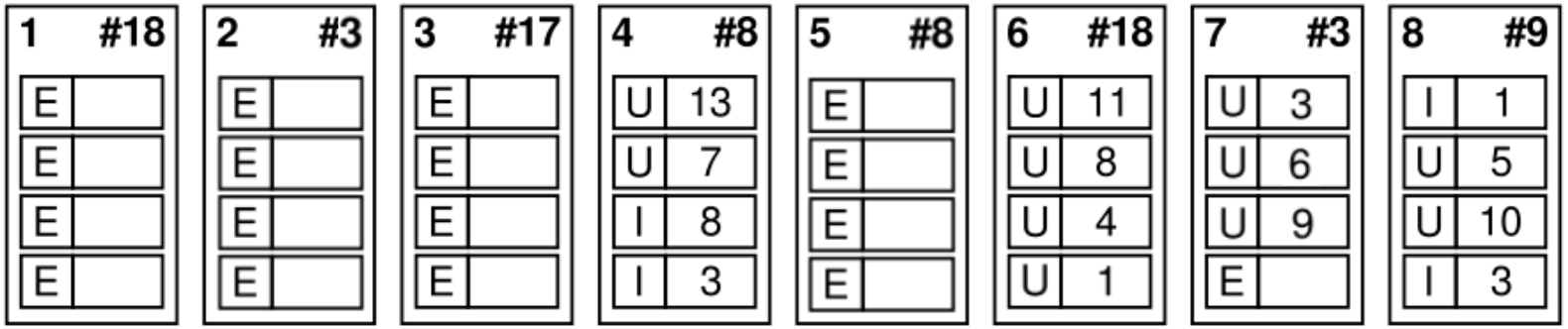


b) Garbage collection must be continued on:

- There are 3 blocks having each 2 invalid pages, blocks: 4, 5 and 8
- from among them, block 5 has the lowest age → we will delete it.
- to this end, we first have to save its used blocks
  - we copy page 6 to the current write frontier
  - we copy page 9 to the current write frontier
- Erase block 5.

SOLUTION OF TASK 4.

- The state of an SSD consisting of 8 blocks is as follows



b) Garbage collection must be continued on:

- There are 3 blocks having each 2 invalid pages, blocks: 4, 5 and 8
- from among them, block 5 has the lowest age → we will delete it.
- to this end, we first have to save its used blocks
  - we copy page 6 to the current write frontier
  - we copy page 9 to the current write frontier
- Erase block 5. Having 4 erased blocks, we have finished. :-)

DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES