Structure of an operating system

1. The operating system:
   A. Coincides with the kernel
   B. It forms the interface between the physical machine (hardware) and le user applications
   C. Is subject to scheduling policies
   D. It resides in main memory even after the shutdown of the machine

2. In an operating system *microkernels:*
   TO. Some of the features are implemented in userspace instead inside the kernel
   B. User processes can interact directly with the system, avoiding the use of system calls
   C. Communication between the various components of the system is more efficient
   D. There are no security mechanisms

3. In an operating system structured according to a microkernel approach
   A. It doesn't need to have two modes of CPU usage (user vs. kernel mode)
   B. It does not require communication mechanisms between different portions of the operating system
   C. It is more efficient than a monolithic system
   D. Except core features, implement everything else in user space

4. The machine level instruction set:
   A. They consist of an opcode and zero or more operands
   B. They are defined by a specific machine language
   C. They are an abstraction of the hardware architecture
   d. All previous answers are correct

5. Internal CPU registers and cache are units of memory:
   A. Non-volatile
   B. Managed entirely by the architecture at the hardware level
   C. Managed entirely by the operating system
   D. Very economical and highly performing

6. Transition from user to kernel mode occurs when:
   A. A program makes a function call

B. The computer boots (bootstrap)
C. The first statement of a program is executed
D. The amount of time assigned to the running process expires

7. The device controller of an I/O device:
A. Contains logs that indicate its status
B. Contains registers that allow it to be controlled by the CPU
C. Contains registers for exchanging data with the CPU
Q. All previous answers are correct

8. System calls:
A. They are always blockers
B. Cause the current process to terminate and a new process to start

C. Must be implemented in user space
D. They must be implemented in kernel space

9. A blocking system call:
A. Moves the process running it to the ready queue
B. Permanently terminates the process that executes it
C. Temporarily kills the process that executes it
D. Need for the process that runs it to periodically check the outcome (polling)

10. The system call handler:
A. It is invoked by the operating system scheduler
B. It is invoked when the time quantum expires
C. Runs in user space
D. Handles system calls via the system call table

11. The generic system call handler code:
A. Runs in user space
B. It is indexed via the interrupt vector table (IVT)
C. It is invoked when the time quantum expires
D. It is invoked by the operating system scheduler

12. The interrupt vector table(IVT):
A. Dynamically updates on every interruption
B. It is a data structure that contains pointers to the various file handlers interruptions
C. It is a data structure that is associated with each process
D. It is a data structure which contains pointers to error codes

13. The system-call table: (already exited 2 times)
A. Contains as many entries as there are system calls to support

B. Contains as many entries as there are interrupts to support
C. It contains as many entries as there are I/O devices in the system
D. It contains as many entries as there are running processes

14. The system-call table is a managed data structure:
A. From I/O devices
B. From the user process
C. Both from the operating system kernel and from the user process
D. From the operating system kernel

15.If you change the implementation of an existing system call:
A. It is always necessary to change the user code that uses it
B. It is never necessary to change the user code that uses it
C. It is not necessary to change the user code that uses it, provided that the interface (API) of the system call also changes
D. It is not necessary to change the user code that uses it, provided that **not** you also change the interface (API) of the system call

16.A processor takes 5 clock cycles to execute an instruction (CPI = 5), i.e. to complete the entire fetch-decode-execute cycle. Assuming the processor's clock frequency is 5 MHz, how many instructions can it execute in one second? (Remember that 1 MHz = 1*10^6 cycles per second)

A. 1*10^3
B. I decide NOT to answer this question
c. 25*10^3
D. 1*10^6
E. 25*10^6

17. Given a multicore CPU with unit(cores), the number of processes/threads that at a certain moment are in the execution "queue":
A. Can be greater than
B. It is exactly equal to
C. Data is insufficient to answer the question
D. It is at most equal to

Processes

18. Creating a new process from a process is done via: (exited 2 times)

TO.A system call

B. A function call
C. Sending an interrupt
Q. None of the above answers are correct

19. The operating system tracks the state of a process by:
A. A special dedicated and protected area of the main memory
B. A dedicated internal register of the CPU
C. A special dedicated and protected area of the cache memory
d. A dedicated field within the process control block (PCB)

20.A process running on the CPU enters the ready state when:
A. Receives an interrupt signal from an I/O device
B. Request input from the user
C. Request a page that is not in main memory
D. Make a function call

21.A process running on the CPU enters the waiting state when:
A. Receives a signal from an I/O device
B. The quantum of time allotted to it ends
C. Open a network connection (e.g., a TCP socket)
D. Make a function call

22.A process running on the CPU enters the waiting state when:
TO.Request input from the user
B. Make a function call
C. The quantum of time allotted to it ends
D. Receives an interrupt signal from an I/O device

23.A process running on the CPU enters the waiting state when:
A. The quantum of time allotted to it ends
B. User drags pointing device(eg mouse)
C. Make a function call
D. Receives an interrupt signal from an I/O device

24.How many processes will be present in the system following these calls: pid_1 = fork(); pid_2 = fork(); pid_3 = fork();?
A.8
B.7
C.4
Q. 3

25.Given the portion of code in the figure, indicate what the value of the variable will be valuewhich will be printed on the line18:

A.5

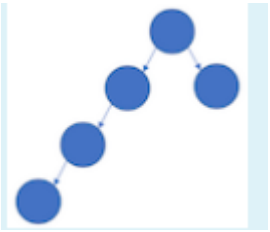
B. 20

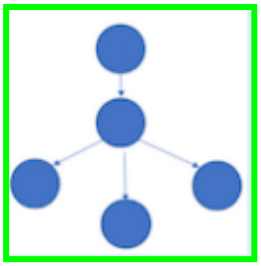
C.15

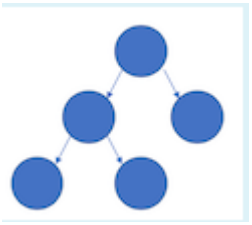
D. The data is insufficient to answer the question

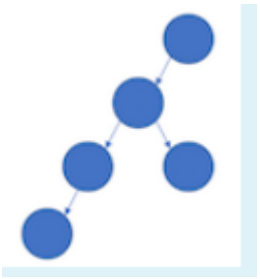
26.Given the portion of code in the figure, indicate the corresponding tree of the generated processes:

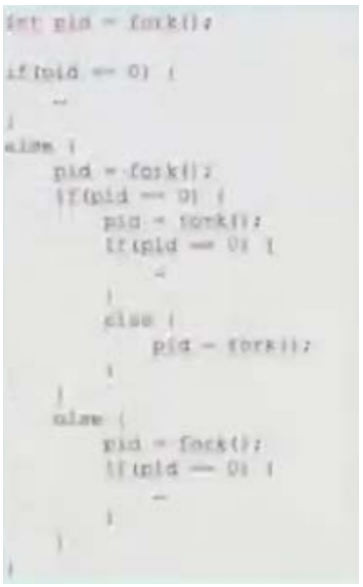
```
int pid = fork();

if(pid == 0) {

    pid = fork();
    if (pid == 0) {
        …
    }
    else {
         pid = fork();
         if (pid == 0) {
            …
         }
         else {
            pid = fork();
                if (pid == 0) {
                    …
                }
            }
         }
    }
}
```

TO. 

b. 

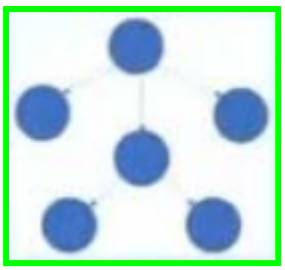
c. 

d. 

27.Given the portion of code in the figure, indicate the corresponding tree of generated processes 

:

TO. 

b. 

c. 

d. 

28. CPU-bound processes that do not make I/O requests:
    A. They have a high priority
    B. They have low priority
    C. Processes are on average short
    d. They can never release the CPU voluntarily

Scheduling Policies

29. The CPU scheduler activates:
    A. When a process tries to write to discord
    B. When program code divides by zero
    C. When the quantum of time expires
    Q. All previous answers are correct
30. Preemptive scheduling (based on time slice or quantum of time):
    A. Prioritize CPU-bound processes
    B. It is activated only when the time slice expires
    C. It is activated only in response to a system call
    d. Provides an upper limit on the CPU time allotted to each process
31. In a time-sharing uniprocessor (single core) system where the running processes are all**purely**CPU-bound:(already exited 2 times)
    A. The use of multi-threading helps improve system latency

B. The use of multi-threading allows you to decrease the time to complete of each process

C. The use of multi-threading improves system throughput

d. There is no benefit to using multi-threading

32. In case of preemptive scheduling, the scheduler intervenes:

A. When a process transitions from the running state to the waiting state

B. When a process transitions from the running state to the ready state

C. When a process goes from the waiting state to the ready state

Q. All previous answers are correct

33.If a job arrives in the ready queue instantly = 2and ends instantly = 15,its turnaround time equals:

A. 13

B.2

C. Data is insufficient to answer the question

Q. 15

34.If a job arrives in the ready queue instantly = 3and ends instantly = 25,the waiting time is equivalent to:

A.3

B. 22

C.25

D. The data is insufficient to answer the question

35. Consider the 5 processes in the following figure and 3 scheduling policies: FCFS, SJF (non-preemptive) and RR with time slice equal to 2 time units. What is the policy that guarantees the shortest waiting time (in the ready queue) for process C?

Job	$T_{arrival}$	T_{burst}
A	0	2
B	0	1
C	0	8
D	0	4
E	0	5

A.FCFS

B.RR

C.SJF

D. All three policies guarantee process C the same wait time

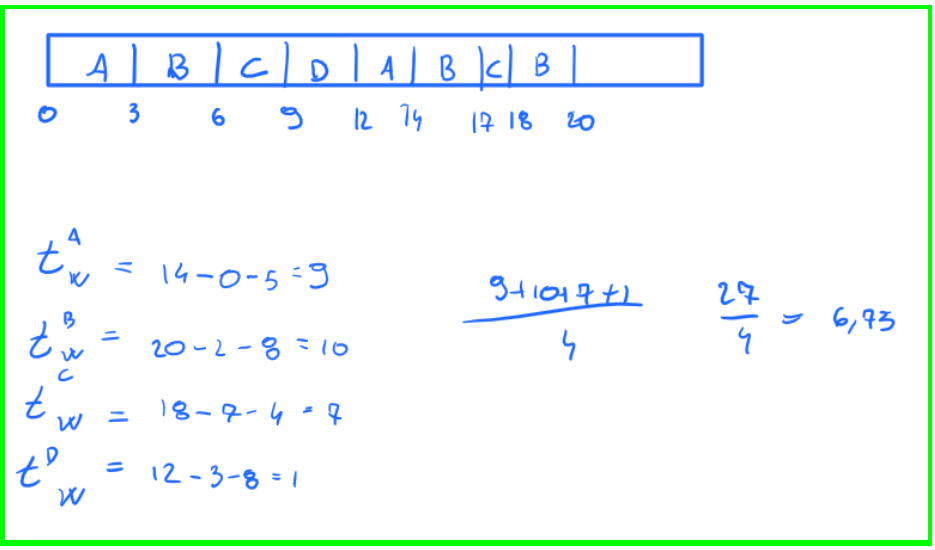
36. Calculate the average waiting time of the following processes, assuming a round robin scheduling policy with time slice = 3, no I/O activity and negligible context switch:

Job	$T_{arrival}$	T_{burst}
A	0	5
B	2	8
C	7	4
D	8	3

Choose an alternative:

A. 6.5

B. 6.75



C. 7.15

D. 5.85

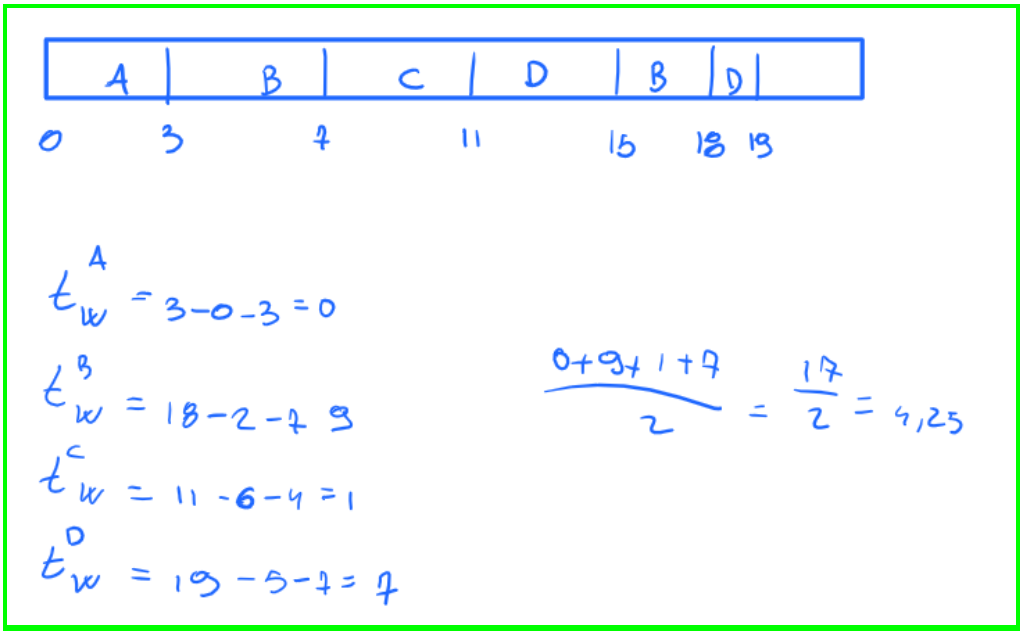
37. Calculate the average waiting time of the following processes, assuming a Round Robin scheduling policy with time slice q= 4. In the calculation, consider the time needed to execute the context

negligible switch:

Job	$T_{arrival}$	T_{burst}
A	0	3
B	2	7
C	6	4
D	7	5

A. 4.85

B.4.25



C.4.5

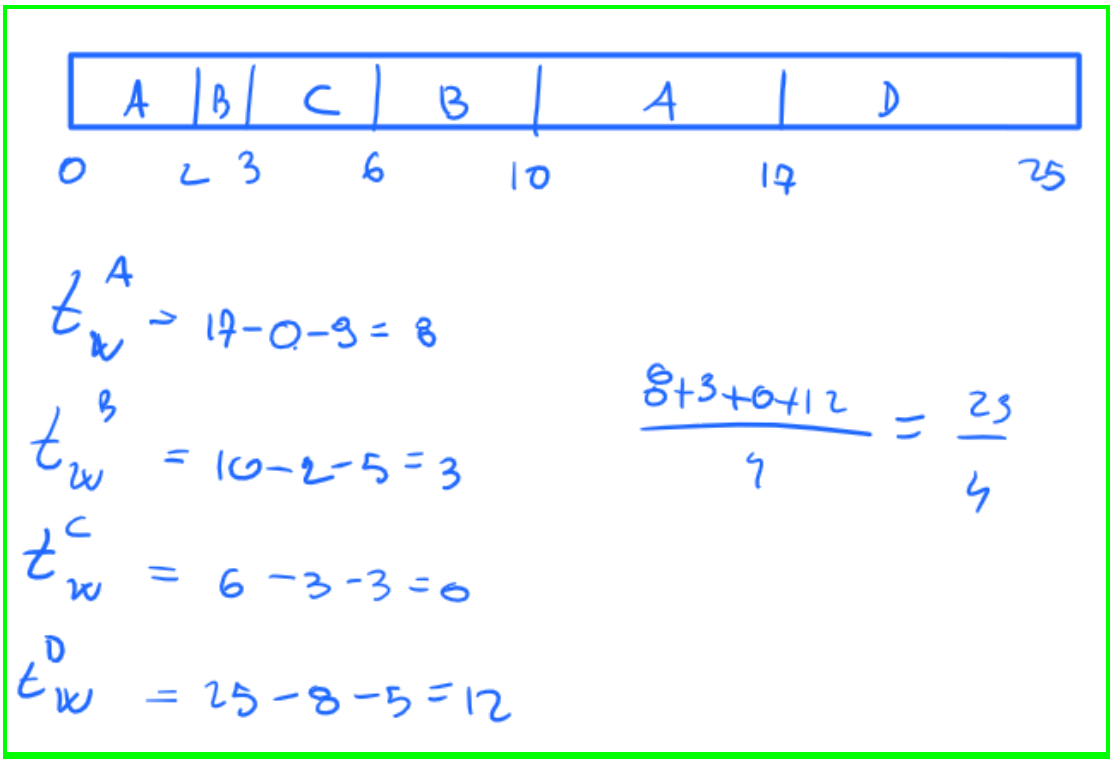
D. 4.75

38.Calculate the average waiting time of the following jobs, assuming a Shortest Job First preemptive (SJF) scheduling policy. In the calculation, the time needed to execute the context switch is considered negligible:

Job	$T_{arrival}$	T_{burst}
A	0	9
B	2	5
C	3	3
D	5	8

A. a. 6

B. 5.75



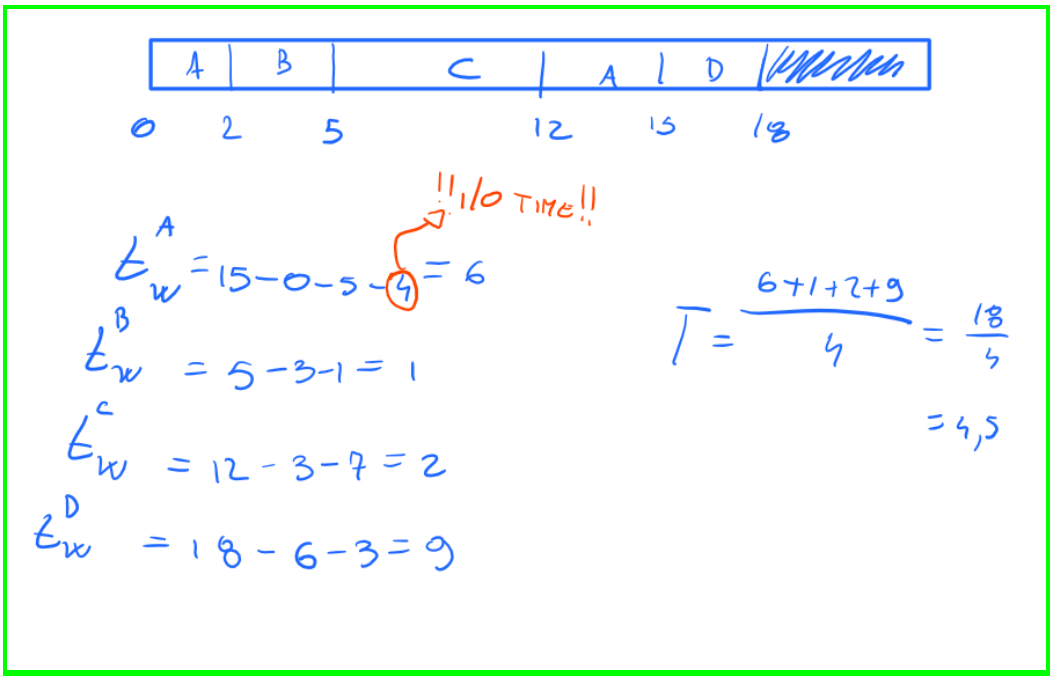
C.4.5

Q. 5

39.Calculate the average waiting time of the following processes, assuming a First Come First Served (FCFS) scheduling policy and that process A executes at time t=2 an I/O call that will complete after 4 time units, i.e. at instant t=6. In the calculation, the time needed to execute the context switch is considered negligible:

Job	$T_{arrival}$	T_{burst}
A	0	5
B	1	3
C	3	7
D	6	3

A.4.5



B.5.5

C.7.5

Q.6.5

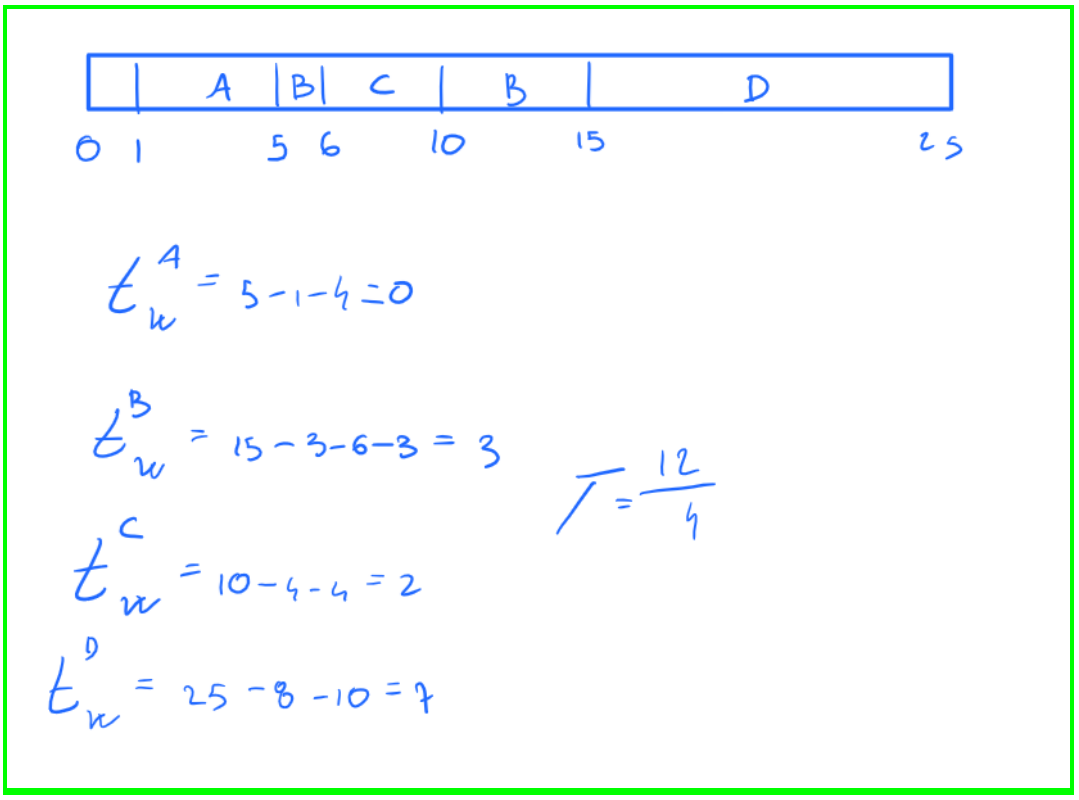
40. Calculate the average waiting time of the following processes, assuming a First Come First Served (FCFS) scheduling policy and that process B executes an I/O call at time t=6 which will complete after 3 time units, i.e. at instant t=9. In the calculation, consider negligible the time needed to execute the context switch:(output 2 times)

Job	$T_{arrival}$	T_{burst}
A	1	4
B	3	6
C	4	4
D	8	10

A.4.5

B. 5.25

c. 4



Q.4.25

Threads and Synchronization

41. Threads of the same process share:
    A. The stack
    B. Global variables
    C. The values of the CPU registers
    D. None of the information listed above
42. The user thread:
    A. Requires kernel-level thread table support
    B. It is the smallest unit that can be scheduled on the CPU by the operating system
    C. It is managed in user space through a special library
    D. It always coincides with one and only one kernel thread

43. In the so-called one-to-one thread mapping model:
    A. Allows you to manage threads through a user-level thread management library
    B. Can only be implemented on multiprocessor systems
    C. Causes all threads in a process to block if even one of those threads makes a blocking system call
    D. Allows you to manage threads at the operating system kernel level

44. In the so-called many-to-one thread mapping model:
    A. Many user threads can be spread across multiple CPUs (if any)
    B. The effect of a blocking call by a user thread does not block the other threads of which the process is composed
    c. Many user threads are mapped to a single kernel thread
    D. Many kernel threads are mapped to a single user thread
45. The many-to-many thread mapping model
    A. There is no limit on the number of kernel threads
    B. Can only be implemented on multiprocessor systems
    C. Causes all threads in a process to block if even one of those threads makes a blocking system call
    Q. It's the compromise between a purely user thread implementation level and a purely kernel level

46. We talk about parallelism when: (already exited 3 times)
    A. Single-threaded processes run on multicore CPUs
    B. Multi-threaded processes run on single core CPUs
    c. Multi-threaded processes run on multicore CPUs
    Q. All previous answers are correct

47. There is talk of competition when: (already released 2 times)
    TO.Multi-threaded processes run on single core CPUs
    B. Single-threaded processes run on single core CPUs
    C. Single-threaded processes run on multicore CPUs
    Q. Multi-threaded processes are running on multicore CPUs

48. Communication between threads of the same process versus between different processes:
    A. a. It is slower since the threads are managed by high level libraries
    B. It's faster since threads don't context switch
    C. It's faster since the threads share the same space as addressing
    Q. There is no substantial difference in terms of performance
49. The kernel thread:
    A. It always coincides with one and only one user thread
    B. It is managed in user space through a special library
    C. It is the smallest unit that can be scheduled on the CPU by the operating system
    D. It is the term used to identify the operating system's own processes (ie, not user processes)

50. Using a lock synchronization primitive expects that:

A. The lock is initially free
B. The lock is acquired before entering the critical section
C. The lock is released after exiting the critical section
D. All previous conditions must be verified

51. Acquiring a lock:
A. It must happen atomically, without the scheduler interrupting the acquisition
B. Mandatory support of atomic hardware instructions

C. It is mandatory for the operating system to disable interrupts

Q. None of the above answers are correct

52.A semaphore can be used for:(exited 2 times already)
A. Enforce scheduling policies between processes/threads
B. Access kernel code
C. Message passing between processes/threads
D. Handle interrupts to the CPU

53. Invoking the wait() method on a semaphore whose value is equal to 2:
A. Leave the semaphore value unchanged at 2 and continue the process that executed the invocation (net of scheduling policies)
B. Decrement the semaphore value to 1 and block the invoking process
C. Increase the value of the semaphore to 3 and continue the process that executed the invocation (net of scheduling policies)
D. Decrease the value of the semaphore to 1 and continue the process che executed the invocation (net of scheduling policies)

54. The test-and-set statement:
A. It is an atomic instruction that allows you to implement the primitives of synchronization
B. It is an atomic instruction which allows interrupts to be disabled
C. It is an atomic instruction that allows you to update the values of multiple registers simultaneously
D.e. It is an atomic instruction that allows you to reset the value of a semaphore

55. The difference between deadlock and starvation lies in the fact that:
A. Refer to user code and system code (respectively)
B. In case of starvation the whole system is completely blocked
C. There is no difference
D. In the event of a deadlock the whole system is completely blocked

56. 2 processes/threads are considered**Producer**(**P**) And**Consumer**(**c**) whose code consists of the following instructions:

P(P1) R1 = val; (P2)R1 += 1; (P3)val= R1;

c(C1) R2 = val; (C2)R2 -= 1; (C3)val = R2;

Assuming thatR1AndR2are internal registers and thatvalis a variable in main memory initially equal to73, what will be the value stored invalat the end of the execution of**P**and of**c**if the total 6 instructions (of**P**And**c**) will be scheduled according to the following order: P1,P2,P3,C1,C2,C3?

A. 72

B. 73

C.74

D. The data is insufficient to answer the question

57. 2 processes/threads are considered**Producer**(**P**) And**Consumer**(**c**) whose code consists of the following instructions:

P(P1) R1 = val; **c**((P2)R1 += 1; (P3)val= R1;
C1) R2 = val; (C2)R2 -= 1; (C3)val = R2;

Assuming thatR1AndR2are internal registers and thatvalis a variable in main memory initially equal to24, what will be the value stored in valat the end of the execution of**P**and of**c**if the total 6 instructions (of**P**And **c**) will be scheduled according to the following order: C1, P1, P2, C2, C3, P3?

A. 23

B. 24

C.25

D. The data is insufficient to answer the question

58. 2 processes/threads are considered**Producer**(**P**) And**Consumer**(**c**) whose code consists of the following instructions:

P(P1) R1 = val; **c**(P(2)R1 += 1; (P3)val= R1;
C1) R2 = val; (C2)R2 -= 1; (C3)val = R2;

Assuming thatR1AndR2are internal registers and thatvalis a variable in main memory initially equal to19, what will be the value stored in valat the end of the execution of**P**and of**c**if the total 6 instructions (of**P**And **c**) will be scheduled according to the following order: C1, P1, C2, P2, P3, C3?

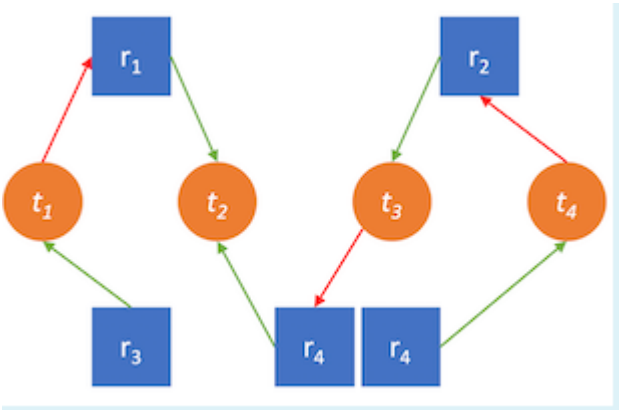
A. 18

B. 19

C.20

D. The data is insufficient to answer the question

59. The following Resource Allocation Graph (RAG) shows a system whose state:



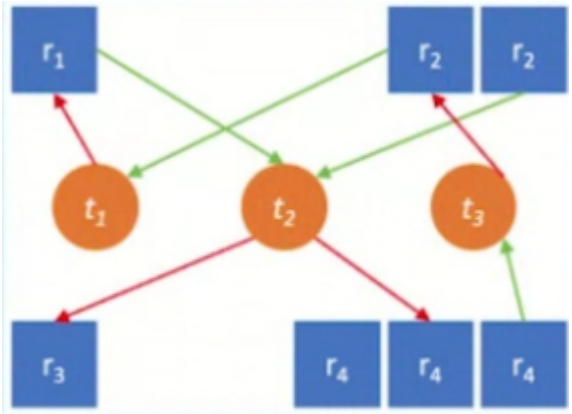
A. It depends on the choices of the operating system scheduler

B. Has deadlocks

C. Has no deadlocks

Q. It is impossible to answer

60. The following Resource Allocation Graph (RAG) shows a system whose state:



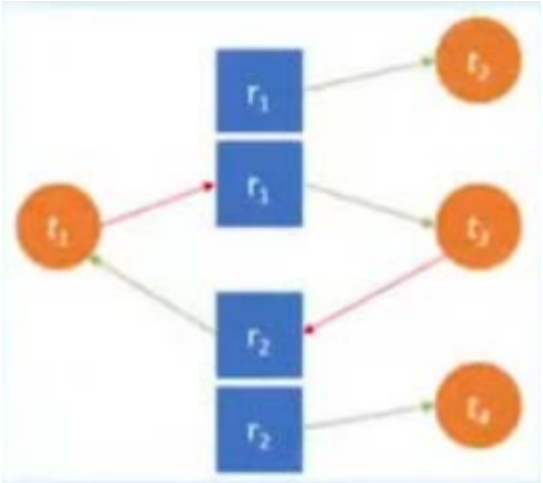
A. It depends on the choices of the operating system scheduler

B. Deadlock present

C. Has no deadlocks

Q. It's impossible to answer

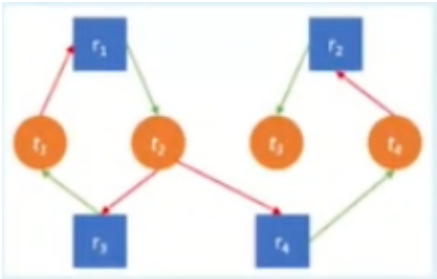
61. The following Resource Allocation Graph (RAG) shows a system whose state:



A. It definitely has deadlocks

B. May have deadlocks
C. Definitely no deadlocks
Q. It's impossible to answer

62. The following Resource Allocation Graph(RAG) shows a system that:



A. It definitely has deadlocks
B. May have deadlocks
C. Definitely no deadlocks
Q. It's impossible to answer

The Main Memory

63. The term address binding means:
    A. The process of translating from logical to physical addresses
    B. The process of initializing the global variables of a program
    C. The process of linking the compiled code to any external libraries

    Q. None of the above answers are correct
64. Swapping allows you to:
    A. Implement dynamic code relocation of a process
    B. Solve the problem of external fragmentation
    C. Temporarily move to disk processes that are not currently running
    D. Exchange areas of memory occupied by two or more processes
65. "Paged" memory management (paging):
    A. It expects the logical address space of a process to be non-contiguous and divided into blocks of fixed size (pages)
    B. It does not require any hardware support to be implemented efficiently

C. Expects the physical address space of a process to be non-contiguous and divided into blocks of fixed size (frames)

D. It solves the problem of internal fragmentation

66. The Translation Look-aside Buffer (TLB) cache

A. It is shared among all processes on the system

B. Allows an average faster translation of logical addresses

C. Contains a subset of the page table entries

Q. All previous answers are correct

67. The size (ie, the number of entries) of the page table:

A. It is directly proportional to the (fixed) size of the pages

B. It adapts according to the memory access requests of each process

C. It depends on the (fixed) size of the pages

D. It varies dynamically depending on the process

68. The size (ie, the number of entries) of the page table:

A. It varies dynamically depending on the process

B. It is directly proportional to the (fixed) dimension

C. It is inversely proportional to the (fixed) size of the pages

D. Scales according to each process's memory access requests

69. A compiler generates the logical address 576 to refer to a certain physical memory location. Assuming that address translation occurs via static relocation with base physical address = 24, what will be the corresponding physical address?

A. 576

B. 552

c.600

Q. There is insufficient data to answer your question

70.Consider a process of size 2.488 bytes and a block of free memory of size 2.699 bytes. In this case, assuming the contiguous memory allocation constraint, the most convenient choice is: (already exited 2 times)

A. Allocate the entire block to the process, wasting 211 bytes(internal fragmentation)

B. Allocate the necessary portion of the block to the process and add the remaining 211 bytes to the list of free blocks (external fragmentation)

C. Wait until there is a block that is multiple in size than the process

D. Wait for a smaller block size to fit the process

71. Consider a process of size 4.996 and a free block of memory of size 5.016 bytes. In this case, assuming the contiguous memory allocation constraint, the most convenient choice is:
    A. Wait for a smaller block size to fit the process

    B. Allocate the entire block to the process, wasting 20 bytes(internal fragmentation)
    C. Wait until there is a block that is multiple in size than the processes

    D. Allocate the portion of the block needed by the process and add the remaining 20 bytes to the list of free blocks (external fragmentation)

72. Suppose that a process P needs a free memory area equal to 99 KiB to be allocated in such a way**contiguous**in main memory. If the list of free memory blocks contains the following elements: A, B, C, D whose sizes are 102 KiB, 99 KiB, 256 KiB and 128 KiB respectively, which block will be allocated for P assuming a Worst-Fit policy?
    A. block A
    B. block C
    C. block B
    D. block D
73. Suppose that a process P needs a free memory area equal to 99 KiB to be allocated in such a way**contiguous**in main memory. If the list of free memory blocks contains the following elements: A, B, C, D, E, F whose sizes are 300 KiB, 600 KiB, 350 KiB, 200 KiB, 750 KiB and 125 KiB respectively, which block will be allocated for P assuming a Worst-Fit policy?

    A. block B
    B. The request cannot be fulfilled, so P will have to wait
    C. C and the remaining 25 KiB are allocated to A
    D. block E
74. Suppose that a process P needs a free memory area equal to 128 KiB to be allocated in such a way**contiguous**in main memory. If the list of free memory blocks contains the following elements: A, B, C, D whose sizes are 105 KiB, 916 KiB, 129 KiB and 80 KiB respectively, which block will be allocated for P assuming a First-Fit policy?
    A. block A
    B. block D
    C. block B
    D. block C
75. Suppose that a process P needs a free memory area equal to 115 KiB to be allocated in such a way**contiguous**in main memory. If the list of free memory blocks contains the following elements: A, B, C, D,E,F

whose sizes are 300KiB, 600KiB, 350KiB, 200KiB, 750KiB and 125KiB respectively which block will be allocated for P assuming a First-Fit policy?

A. block A
B. block F
C. block E
D. block D

76. Suppose that a process P needs a free memory area equal to 375 KiB to be allocated in such a way**contiguous**in main memory. If the list of free memory blocks contains the following elements: A, B, C, D,E,F whose sizes are respectively 300 KiB, 600 KiB, 350 KiB, 200 KiB,750 KiB and 125 KiB which block will be allocated for P assuming a Best-Fit policy?

    A. block B
    B. block C and the remaining 25 Kib are allocated to A
    C. block E
    D. The request cannot be fulfilled, so P will have to wait

77. Suppose that a process P needs a free memory area equal to 34 KiB to be allocated in such a way**contiguous**in main memory. If the list of free memory blocks contains the following elements: A, B, C, D whose sizes are 36 KiB, 90 KiB, 42 KiB and 35 KiB respectively, which block will be allocated for P assuming a Best-Fit policy?

    A. block A
    B. block B
    C. block C
    D. block D

78. Suppose we have a memory M with a capacity of 4 KiB, or 4.096 bytes. Assuming that the addressing takes place with word length (word size) equal to 2 bytes and that M uses paged management with blocks of size equal to S = 128 bytes, how many bits are needed to identify the page index (p) and the offset (inside the page), respectively?

    A. p=6; offset=5
    B. bp=7; offset=5
    C. p=5; offset=7
    D. p=5; offset=6

79. Consider a memory M with a capacity equal to 512 bytes with a frame size equal to 16 bytes. Given the address of byte 197, what will be the page address (p) and the offset (inside the page):(happened twice)

A. p=5; offset=12
B. Data is insufficient to answer the question
C. p=13; offset=0
d. p=12; offset=5

80. Consider a memory M with a capacity equal to 100 bytes with a frame size equal to 10 bytes. Given the address of byte 37, what will be the page address (p) and the offset (inside the page).
A. p=3; offset=7
B. Data is insufficient to answer the question
C. p=7; offset=3
D. p=0; offset=37

81. Consider a process of size 2.097 bytes and a block of free memory of size 2.104 bytes. In this case, assuming the contiguous memory allocation constraint, the most convenient choice is:
A. Wait for a block that is multiple in size than the process

B. Allocate the entire block to the process, wasting 7 bytes (internal fragmentation)
C. Wait for a smaller block size to fit the process

D. Allocate the portion of the block needed by the process and add the remaining 7 bytes to the list of free blocks (external fragmentation)

82. Suppose we have a memory M with a capacity of 2 KiB, or 2.048 bytes. Assuming that the addressing takes place with a word size equal to 4 bytes, how many bits are needed to address the words contained in M?

A. 2
B. 9
C.11
Q. There is insufficient data to answer your question

83. Suppose we have a memory M with a capacity of 4 KiB or 4.096 bytes. Assuming that the addressing occurs with a word size equal to 2 bytes, how many bits are needed to address the words contained in M?

A. 10
b.11
C.12
D. The data is insufficient to answer the question

84. Suppose we have a memory M with a capacity of 8 KiB, or 8.192 bytes. Assuming that the addressing takes place with word length (word size) equal to a single byte and that M uses paged management with blocks of size equal to S = 128 bytes, what size (understood as number of entries) does the corresponding page table have T?
    A. The data is insufficient to answer the question
    B. 13
    C.64
    Q. 7

85. Suppose we have a memory M with a capacity of 8 KiB, or 8.192 bytes. Assuming that the addressing takes place with a word length (word size) equal to 4 bytes and that M uses paged management with blocks of size equal to S = 256 bytes, what will be the number of entries of the corresponding page table T?

    A. 32
    B. 2.048
    C.8
    Q. 5

86. Suppose we have a memory M with a capacity of 16 KiB, or 16.384 bytes. Assuming that the addressing takes place with a word length (word size) equal to 4 bytes and that M uses paged management with blocks of size equal to S = 64 bytes, what will be the number of entries of the corresponding page table T?

    A.8
    B. 14
    C. 256
    Q. There is insufficient data to answer your question

87. Consider an operating system that uses 21-bit logical addresses, 16-bit physical addresses, and paged memory where each page is 2 KiB(2048 bytes) in size. What is the maximum physical memory size supported by the system?(exit already 2 times)
    A. 32 KiB
    b. 64KiB
    C. 2 MiB
    Q. There is no physical limit to the memory supported by the system

Virtual memory

88. Virtual memory allows you to:
    A. Increase the efficiency of I/O operations

B. Keep only some pages of the space allocated in physical memory logical addressing of a process

C. Decrease the degree of multiprogramming of the system

D. Running a process directly from secondary storage devices (eg, disk)

89.If an idempotent instruction generates a page fault:

A. The process that the statement is part of ends

B. Idempotent instructions cannot generate page faults

C. The instruction will no longer be executed after the page fault handling returns

D. Will the instruction re-execute when I return from handling the page fault

90. The problem of external fragmentation:

A. It needs hardware support to fix

B. It is not fixable short of a system reboot

C. It is a consequence of the contiguous memory allocation constraint

D. It causes a hardware outage

91. The problem of external fragmentation

A. It is not fixable without a system reboot

B. It causes a hardware interrupt

C. It needs hardware support to fix

D. It is due to allocating/deallocating contiguous blocks of memory

92. The*working set*is:

A. Fixed for each quantum of time

B. Relatively large compared to the entire address space of a process

C. Relatively small compared to the entire address space of a process

D. Fixed for the entire execution of a process

93. Consider a system that implements the LRU policy for replacing frames using a timestamp. For each request for access to a specific frame it is necessary to:

A. increment a counter type variable

B. Update the timestamp value with the current one

C. Set a validity bit

Q. None of the previous answers are correct

94. Given a memory made up of 3 physical frames and a process made up of 5 virtual pages: A, B, C, D, E, calculate the number of page faults that occur following the following requests from the process: B ,C,C,B,A,E,B,A, E, D, B. Assume that**none**process page is initially

loaded into memory and that an LRU page replacement algorithm is used.

A.4
B.5
C.6
Q. 7

95. Given a memory made up of 3 physical frames and a process made up of 5 virtual pages: A, B, C, D, E, calculate the number of page faults that occur following the following requests from the process: D , B, A, C, C, E, A, D, B, E, D, A. Assume that**none**process page is initially loaded into memory and that an LRU page replacement algorithm is used.

A. 10
B.7
C.9
Q. 6

96. Given a memory made up of 3 physical frames and a process made up of 5 virtual pages: A, B, C, D, E, calculate the number of page faults that occur following the following requests from the process: C ,B,C,B,A,E,B,A. Assume that**none**process page is initially loaded into memory and that an LRU page replacement algorithm is used.

A. 2
B.4
C.5
Q.1

97. Given a physical memory made up of 3 physical frames and a process made up of 5 virtual pages: A, B, C, D, E, calculate the number of page faults that occur following the following requests from the process: A, B, E, C, E, D, D, A, B. Assume that**none**process page is initially loaded into memory and that a FIFO page replacement algorithm is used.

A. 6
B.7
C.4
Q. 8

98. Given a memory made up of 3 physical frames and a process made up of 5 virtual pages: A, B, C, D, E, calculate the number of page faults that occur following the following requests from the process: D , A, C, B, B, A, C, B, D, E, A. Assume that**none**process page is initially loaded into memory and that a FIFO page replacement algorithm is used.

A. 6
B.7
C.5
Q. 4

99. Given a memory made up of 3 physical frames and a process made up of 5 virtual pages: A, B, C, D, E, compute the number of page faults that occur following the following requests from the process: E, B, E, C, D, E, A, B, E. Assume that**none**process page is initially loaded into memory and that a FIFO page replacement algorithm is used.
A. 7
B.8
C.6
Q. 5

The Secondary Memory

100. The contiguous allocation of a file on disk:(output already 4 times)
A. It is great for both direct (random) and sequential access
B. Has the problem of fragmentation
C. It requires the maintenance of free blocks within a suitable data structure
d. All previous answers are correct

101. Contiguous allocation of a file on a disk is the preferable choice when the disk is:
A. A read-only CD/DVD-ROM
B. A magnetic disk
C. A solid state drive
D. In none of the above cases

102. In a magnetic disk, the seek time: (happened 2 times already)
A. It is the time it takes for the disk to position its heads on a specific sector
B. Includes transfer time to main storage
C. It is the time it takes for the disk to position its heads on one specific cylinder
D. It's negligible compared to the entire time it takes to transfer the data

103. A disk consists of 15 cylinders, each with a capacity of 500MB. What is the total disk capacity?
A. 7.5GB
b.b. 75GB

c.c. 750MB
D. d. The data is insufficient to answer the question

104. Suppose the physical memory access time is t_{MA} = 50 nsec. and that time for handling a page fault t_{FAULT} is equal to 15 msec. Assuming that the probability of a page fault occurring is p = 0.0002, what is the expected total memory access time?
A. ~30.5 nsec
B. ~30.5 microsec
c. ~3.05 microsec
D. ~305 nsec

105. Suppose the physical memory access time is t_{MA} = 25 nsec. and that time for handling a page fault t_{FAULT} is equal to 30 msec. Assuming that the probability of a page fault occurring is = 0.005, what is the expected total memory access time?
A. ~150.025 microsec
B. ~15.025 nsec
C. ~150.025 nsec
D. ~15.025 microsec

106. Assume that the physical memory access time is t_{MA} = 50 nsec. and that time for handling a page fault t_{FAULT} is equal to 25 msec. Assuming the average memory access time is 0.5 microsec, what is the probability that a page fault will occur?
A. ~0.02%
B. ~0.2%
C ~0.002%
D. ~0.0002%

107. Suppose the physical memory access time is t_{MA} = 60 nsec. and that time for handling a page fault t_{FAULT} is equal to 5 msec. What should be the value of the probability that a fault occurs ( ) if we want to guarantee that the expected memory access time is at most the 20% slower than ? (Remember that 1 msec = 10^3 microsec = 10^6 nsec)
A. a. The data is insufficient to answer the question
B. ~0.00024%
C. ~0.000024%
D. ~0.0000024%

108. Consider a magnetic disk composed of 128 cylinders/tracks, numbered from 0 to 127 (0 index of the cylinder/track most external to the center of the disk), whose head is initially located on cylinder 42. Calculate the number of cylinders/tracks traversed by the disk head, assuming the sequence

of requests: 74, 50, 32, 55, 81 is managed by an SSTF (Shortest Seek Time First) scheduling algorithm and neglecting the rotation time.

A. 86
B. 49
c.123
D. 88

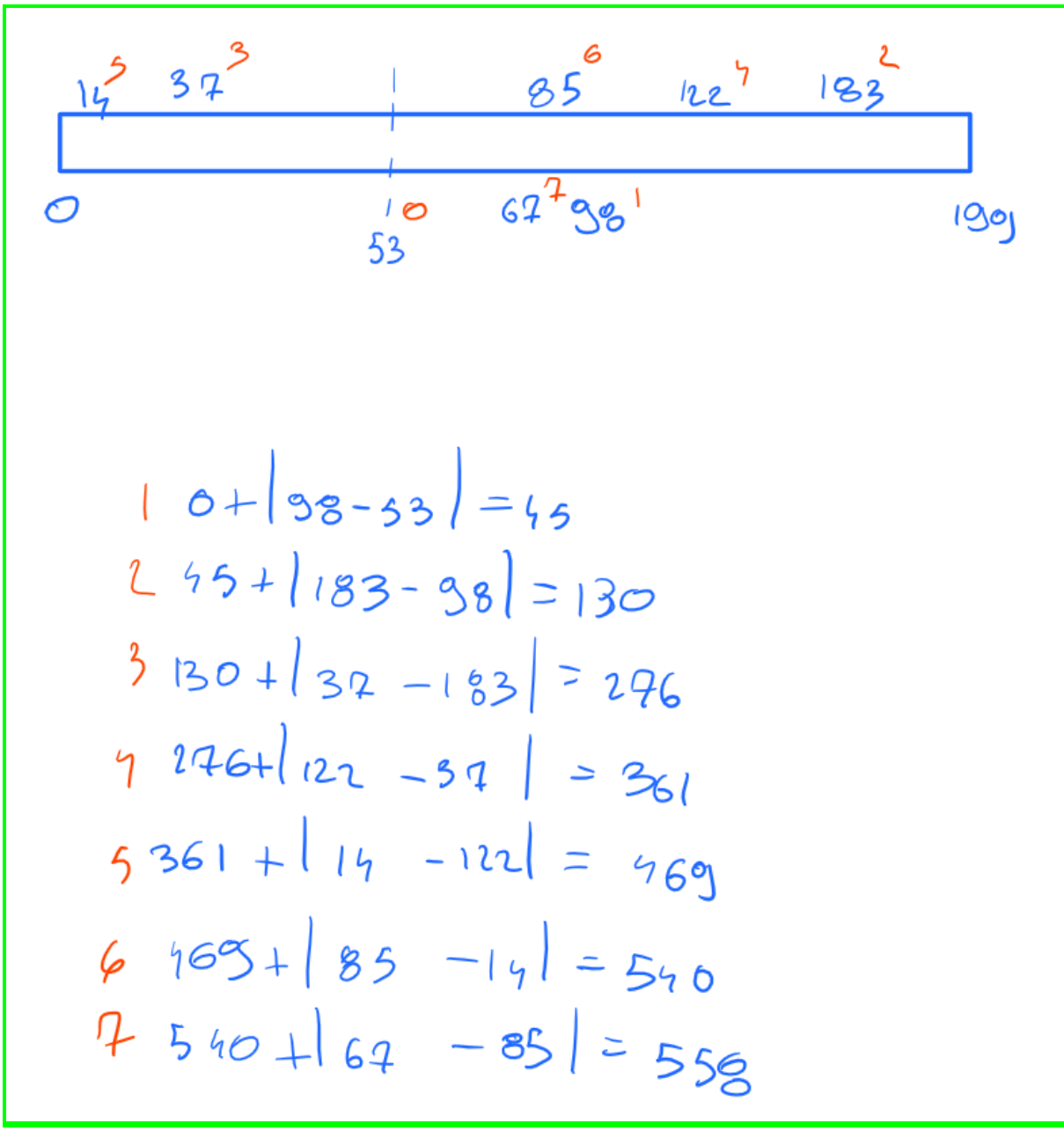
109. Consider a magnetic disk composed of 128 cylinders/tracks, numbered from 0 to 127 (0 index of the cylinder/track most external to the center of the disk), whose head is initially located on cylinder 87. Calculate the number of cylinders/ tracks crossed by the disk head, assuming that the sequence of requests: 43, 81, 36, 25, 127 is handled by a FCFS (First Come First Served) scheduling algorithm and neglecting the rotation time.

A. 290
B. 240
C. 238
D. 265

110. Consider a magnetic disk made up of 200 cylinders/tracks, numbered from 0 to 199 (0 index of the cylinder/track most external to the center of the disk), whose head is initially located on cylinder 53. Calculate the number of cylinders/tracks crossed by the disk head, assuming that the sequence of requests: 98,183,37,122,14,85,67 is handled by a FCFS (First Come First Served) scheduling algorithm and neglecting the rotation time.

A.595

b. 558



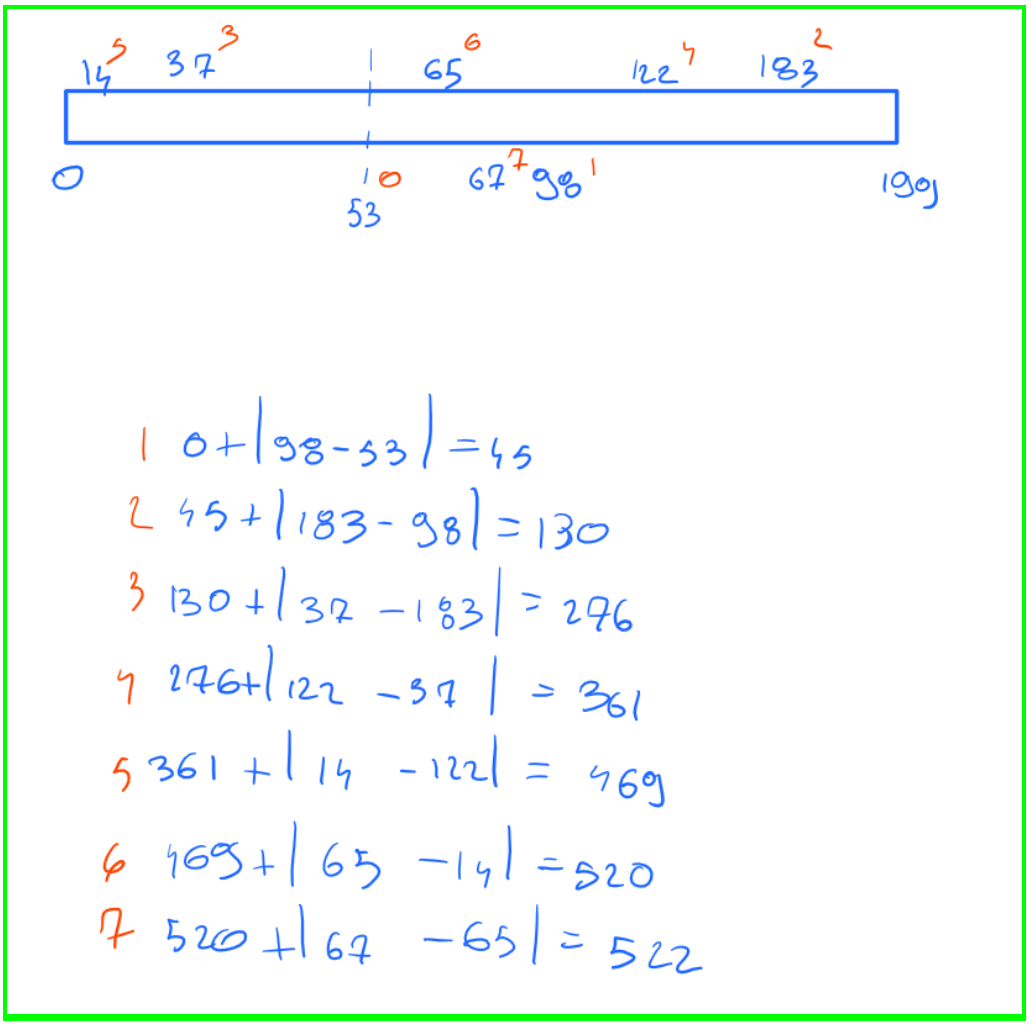
C.650

D. 638

111. Consider a magnetic disk made up of 200 cylinders/tracks, numbered from 0 to 199 (0 index of the cylinder/track most external to the center of the disk), whose head is initially located on cylinder 53. Calculate the number of cylinders/tracks crossed by the disk head, assuming that the sequence of requests: 98,183,37,122,14,65,67 is handled by a FCFS (First Come First Served) scheduling algorithm and neglecting the rotation time.

A.650

B. 522



C.638

D. 595

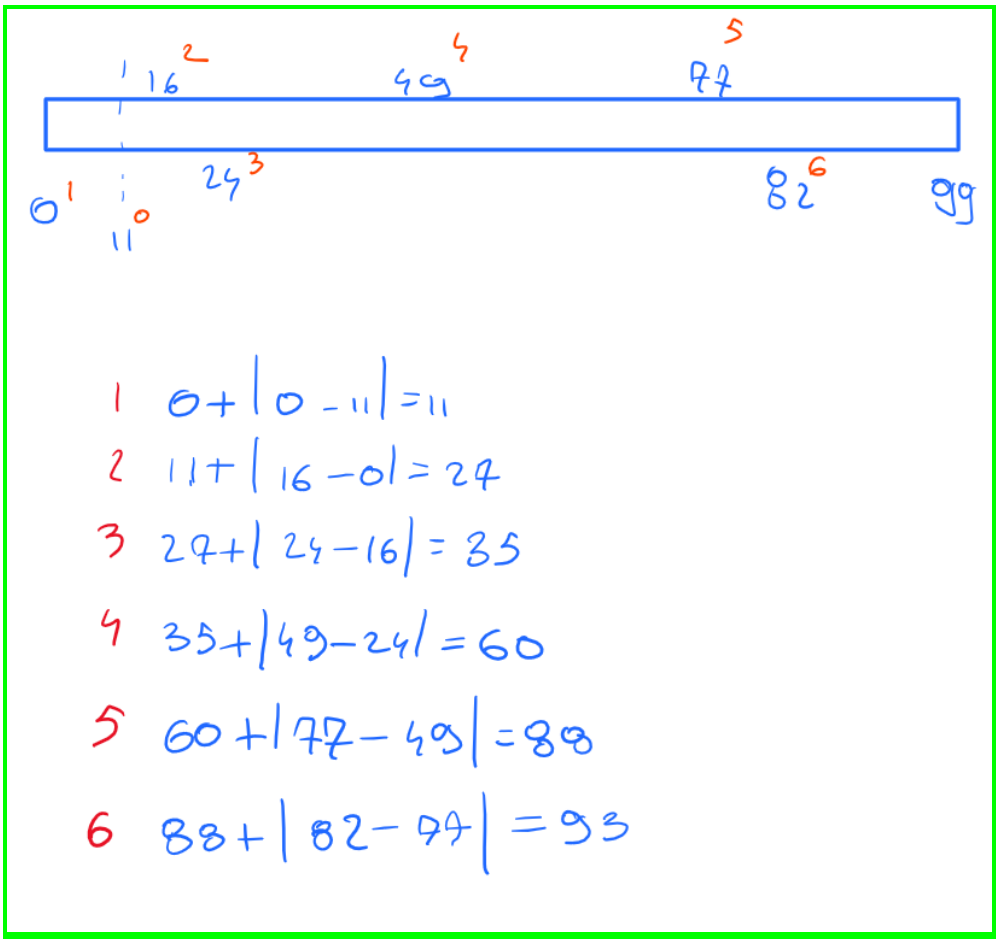
112. Consider a magnetic disk made up of 100 cylinders/tracks, numbered from 0 to 99 (0 index of the cylinder/track most external to the center of the disk), whose head is initially located on cylinder 11. Calculate the number of cylinders/tracks crossed by the disk head, assuming that the sequence of requests: 24, 16, 77, 49, 82 is managed by a SCAN scheduling algorithm (**not** -optimized), that the head is moving outwards (ie, towards lower numbered cylinders) and neglecting the rotation time.

A. 76

B. 87

C.46

D. 93



113. Consider a magnetic disk made up of 100 cylinders/tracks, numbered from 0 to 99 (0 index of the cylinder/tracks most external to the center of the disk), whose head is initially located on cylinder 46. Calculate the number of cylinders/ tracks crossed by the disk head, assuming that the sequence of requests: 4, 96,51,29,18 is managed by a C-SCAN scheduling algorithm (**not**-optimized), that the head is moving outward (ie, towards lower numbered cylinders) and disregarding the rotation time. (exited 2 times already)

A. 189
B. 193
c.187
D. 196

114. The total transfer time for a magnetic disk I/O operation is 30 msec. Knowing that: the total seek time is equal to 18 msec, the total rotational delay is equal to 7 msec and that the transfer rate is equal to 1.5 Gbit/sec, what is the total amount of data transferred? (Remember that 1 B = 1 byte = 8 bits and 1 MB = 10^3 KB = 10^6 B)(output already 3 times)

A. 9,375 MB
b.b. 7.5MB
c. 937.5KB
D. The data is insufficient to answer the question

115. The total transfer time for a magnetic disk I/O operation is 40 msec. Knowing that: the total seek time is equal to 18 msec, the total rotational delay is equal to 7 msec and that the transfer rate is equal to 5 Gbit/sec, what is the total amount of data transferred? (Remember that 1 B = 1 byte = 8 bits and 1 MB = 10^3 KB = 10^6 B)
A. 9.375MB
B. 70MB
C. 70KB
D. The data is insufficient to answer the question

116. The total transfer time for a magnetic disk I/O operation is 36 msec. Knowing that the total seek time is equal to 13 msec and that 2MB have been transferred at a speed equal to 1 Gbit/sec what is the rotational delay of the disk? (Remember that 1 B = 1 byte = 8 bit)
A. 7 msec
B. 2 msec
C. 16 msec
D. The data is insufficient to answer the question

File System Management

117. The global open file table:
A. It is shared among all processes
B. Contains an entry for each file in use
C. Maintains a counter for each file in use
Q. All previous answers are correct
118. The local open file table:
A. Contains security information of each file reported by a process

B. Contains a pointer to the disk location of each file referenced by a process

C. Contains a pointer to the global open files table for each file reported by a process

D. It is shared among multiple processes

119. A possible example of an application that needs sequential access to a file is:

A. A compiler

B. A search system within a database

C. A telephone contact tracing system

Q. None of the above answers are correct

120. An indexed file allocation is preferable when the file in question:

A. It is small, regardless of how it is accessed

B. It is large, regardless of how it is accessed

C. It is large in size and is typically accessed sequentially

D. It is large in size and is typically accessed in a mode random(direct)

121. Allocating a file based on linked lists is preferable when the file in question

A. It is small in size, regardless of how it is accessed

B. It is large and is typically accessed randomly (directly)

C. It is large, regardless of how it is accessed

D. It is large and is typically accessed sequentially

122. On a UNIX-like system, a file that has the following privileges: 101000000:

A. Allows only the owner of the file to exercise read and write rights (on the file)

B. Allows only the owner of the file to exercise rights to read ed execution (on file)

C. Allows only the owner of the file to exercise write and execute rights (on the file)

D. It does not give any rights to the owner of the file

123. On a UNIX-like system, a file that has the following privileges: 011000000:

A. Allows only the owner of the file to exercise read and write rights (on the file)

B. Allows only the owner of the file to exercise read and execute rights (on the file)

C. It does not give any rights to the owner of the file

D. Allows only the owner of the file to exercise write rights ed execution (on file)

124. On a UNIX-like system, a file that has the following privileges: 111101101:

A. Allows the owner of the file to exercise all rights (on the file) and gives other users only write and execute rights

B. Allows the owner of the file to exercise all rights (on the file) and gives other users read and write rights only

C. Allows the owner of the file to exercise all rights (on the file) e it gives other users only read and execute rights

D. allows anyone to exercise all rights (on the file)

125. The UNIX commandln file_1 file_2

A. Create a hard link with file file_2(source) where name is file_1(target)

B. Create a hard link with the file file_1(source) whose name is file_2(destination)

C. Create a soft link with file file_1(source) whose name is file_2(target)

D. Create a soft link with file file_2(source) whose name is file_1(target)

126. The UNIX command ln -s file_1 file2:

A. Create a hard link with file file_2(source) where name is file_1(target)

B. Create a hard link with the file file_1(source) whose name is file_2(target)--

C. Create a soft link with the file file_1(source) whose name is file_2(destination)

D. Create a soft link with file file_2(source) whose name is file_1(target)

127. Consider a file system organized with multi-level indexed file descriptors, containing 10-block direct references to which is added a 100-block level of indirection and a further double level of indirection , always of 100 blocks each. Assuming each block is 2 KiB in size, what is the maximum file size supported?

A. ~20.2 KB

B. ~20.2MB

C. ~20.7KB

Q. ~20.7 MB