Structure of an operating system

1. The operating system:

A. It coincides with the kernel
B. It constitutes the interface between the physical machine (hardware) and the user applications
C. It is subject to scheduling policies
D. It resides in main memory even after system shutdown machine

2. In a microkernel operating system:

A. 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 protection mechanisms provided

3. In an operating system structured according to a microkernel approach

A. It does not need to have two CPU usage modes (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 for core functionality, implement everything else in user space

4. The machine level instruction set:

A. They are composed 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 of the above answers are correct

5. The CPU's internal 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. The transition from user to kernel mode occurs when:
A. A program executes a function call
B. The computer starts (bootstrap)
C. The first statement of a program is executed
D. The quantum of time allotted to the running process expires

7. The device controller of an I/O device:
A. It contains registers 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
D. All of the above answers are correct

8. System calls:
A. They are always blockers
B. They cause the termination of the current process and the start of a new one process
C. They must be implemented in user space
D. They must be implemented in kernel space

9. A blocking system call:
A. Move the process that executes it to the ready queue
B. Permanently kills the process running it
C. Temporarily stops the process that executes it
D. Need for the process that executes it to periodically verify it 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. It 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. It updates dynamically with each interrupt
B. It is a data structure that contains pointers to the various handlers of the interruptions
C. It is a data structure that is associated with each process
D. It is a data structure that contains pointers to error codes

13. The system-call table: (already exited 2 times)
A. It contains as many entries as there are system calls to support
B. Contains as many entries as there are interrupts to support
C. Contains as many entries as there are I/O devices present in the system
D. 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 by the operating system kernel and 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 modify the user code that uses it, as long as you also change the interface (API) of the system call
D. It is not necessary to change the user code that uses it, as long as you also do not change the interface (API) of the system call

16.A processor takes 5 clock cycles to execute an instruction (CPI = 5),
that is, to complete the entire fetch-decode-execute cycle. Assuming that the
processor clock rate is 5 MHz, how many instructions is in
able to run 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 m units(cores), the number of processes/threads that
at a certain moment they are in the "running" state:
A. It can be greater than m
B. It is exactly equal to m
C. The data is insufficient to answer
D. It is at most equal to m

The Processes

18.Creation of a new process by a process occurs
via:(released 2 times)
A. A system call
B. A function call
C. Sending an interrupt
D. None of the above answers are correct

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

20.A process running on the CPU goes into ready state when:
A. Receives an interrupt signal from an I/O device
B. Requests input from the user

C. Requests a page that is not present in main memory
D. Executes a function call

21.A process running on the CPU goes into waiting state when:
A. Receives a signal from an I/O device
B. Finish the allotted amount of time
C. Opens a network connection (e.g., a TCP socket)
D. Executes a function call

22.A process running on the CPU goes into waiting state when:
A. Requests user input
B. Executes a function call
C. Finish the allotted amount of time
D. Receives an interrupt signal from an I/O device

23.A process running on the CPU goes into waiting state when:
A. Finish the allotted amount of time
B. The user drags the pointing device (e.g. mouse)
C. Executes 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
D.3

25.Given the portion of code in the figure, indicate what the value of the variable will be value that will be printed on line 18: (GOOD LUCK LOL)

```c
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int value = 5;

int main() {
    pid_t pid;

    pid = fork();

    if (pid == 0) { /* child process */
        value += 15;
        return 0;
    }
    else if (pid > 0) { /* parent process */
        wait(NULL);
        printf("PARENT: value = %d", value); /* print `value` */
        return 0;
    }
}
```

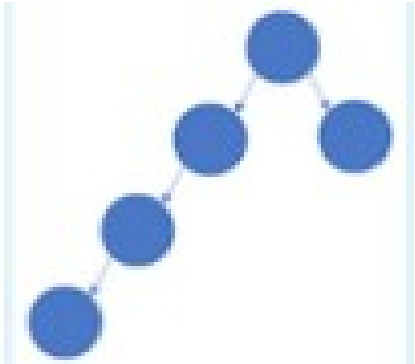
A. 5
B. 20
C. 15

D. The data is insufficient to answer the question

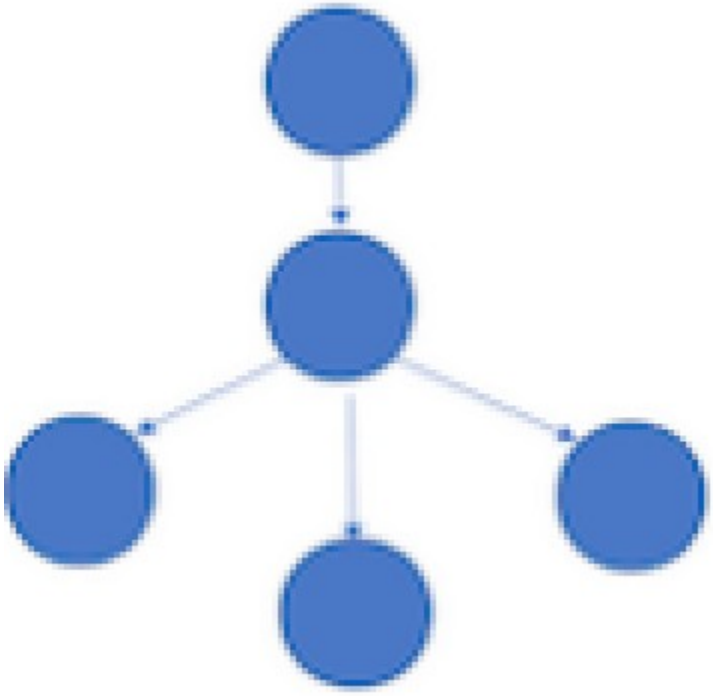
26.Given the portion of code in the figure, indicate the corresponding tree of generated processes:

```
int pid = fork();

if(pid == 0) {

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

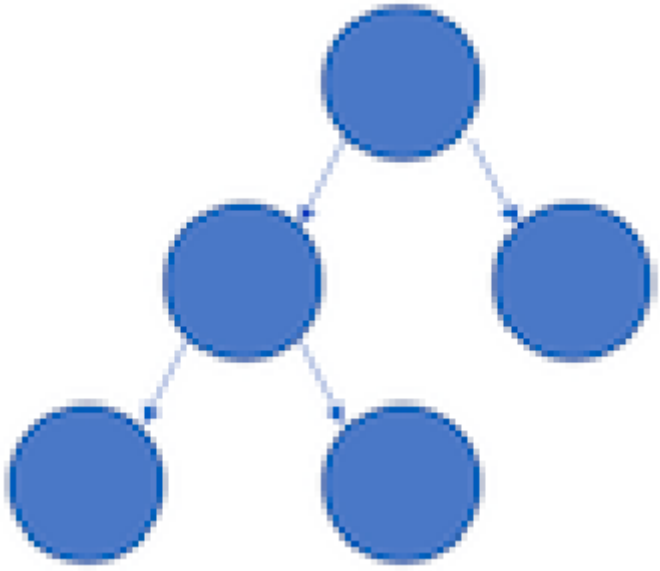
A.

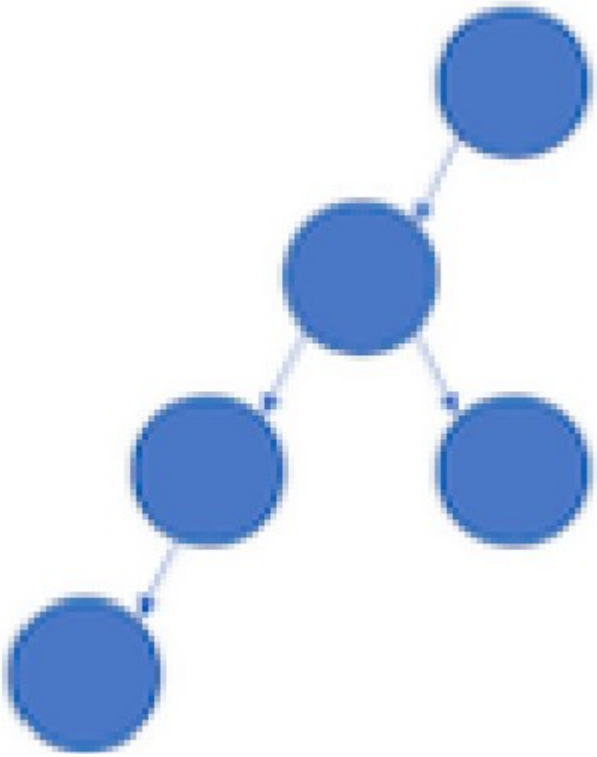


B.



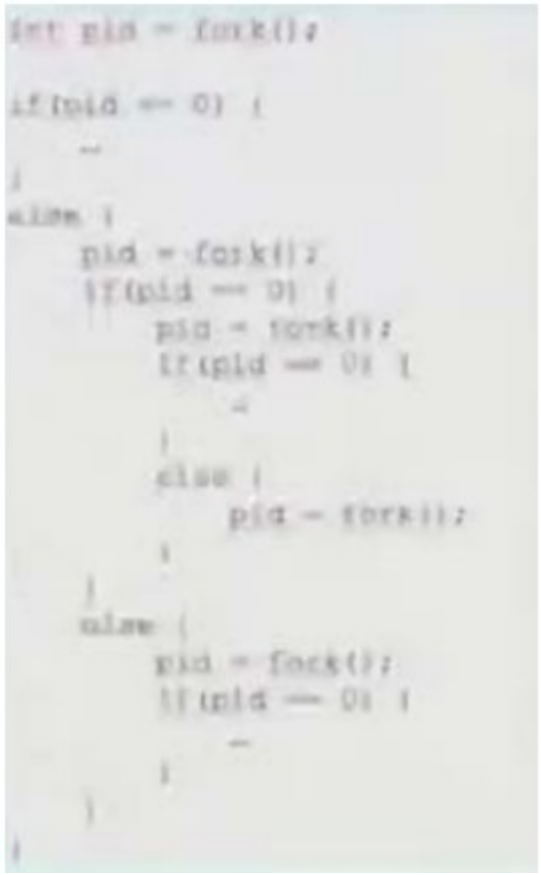
C.





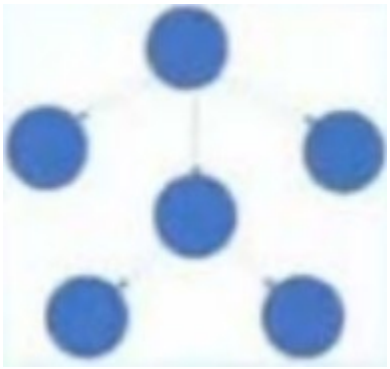
D.

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



A.

B.



C.

D.

28. CPU-bound processes that do not perform I/O requests:
A. They have a high priority
B. They have a low priority
C. They are on average short processes

D. They may never release the CPU voluntarily

Scheduling Policies

29.The CPU scheduler activates:
A. When a process attempts to write to discord
B. When program code performs division by zero
C. When the quantum of time expires
D. All of the above answers are correct

30. Preemptive scheduling (based on time slice or temporal quantum):
A. Give priority to CPU-bound processes
B. It is activated only when the time quantum expires (time slice)
C. It is activated only upon a system call
D. Provides an upper limit on the CPU time allocated to each process

31.In a uniprocessor (single core) time-sharing system in which the processes in execution are all purely CPU-bound:(already exited 2 times)
A. The use of multi-threading allows you to improve the latency of the system
B. The use of multi-threading allows you to reduce the time completely of each process
C. The use of multi-threading improves the throughput of the system
D. There is no advantage to using multi-threading

32.In case of preemptive scheduling, the scheduler intervenes:
A. When a process goes from the running state to the waiting state
B. When a process goes from the running state to the ready state
C. When a process goes from the waiting state to the ready state
D. All of the above answers are correct

33.If a process arrives in the ready queue at time t=2 and terminates at time t=15, its turnaround time equals:
A. 13
B.2
C. The data is insufficient to answer the question
D. 15

34.If a process arrives in the ready queue at time t=3 and terminates at time t=25, the waiting time is equivalent to:
A. 3
B. 22
C. 25
D. There is insufficient data 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.
Which is the policy that guarantees the shortest waiting time (queued ready) at

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 give process C the same amount of time wait

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 slices q= 4. In the calculation, consider the time necessary 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 processes, assuming a Shortest Job First preemptive scheduling policy (SJF). In the calculation, consider the time necessary to carry out the context switch procedure to be negligible:

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

A. 6
B. 5.75
C.4.5
D.5

39.Calculate the average waiting time of the following processes, assuming a First Come First Served scheduling policy (FCFS) and that process A executes at time t=2 an I/O call that is will complete after 4 units of time, i.e. at instant t=6. In calculation, consider the time needed to execute the context switch 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
D.6.5

40.Calculate the average waiting time of the following processes, assuming a First Come First Served scheduling policy (FCFS) and that process B executes at time t=6 an I/O call that yes will complete after 3 units of time, i.e. at instant t=9. In calculation, yes consider the time needed to execute the context switch negligible:(exit 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
D. 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. It requires the support of an appropriate thread table at the kernel level
B. It is the smallest unit that can be scheduled on the CPU by the operating system
C. It is managed in user space via a specific library
D. It always coincides with one and only one kernel thread

43.In the so-called one-to-one thread mapping model:
A. It allows you to manage threads through a specific user-level library
B. It can only be implemented on multiprocessor systems
C. Causes all threads in a process to block, even just one these threads execute 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 distributed across multiple CPUs (if any)

B. The effect of a blocking call from a non-user thread blocks the other threads that make up the process
C. Many user threads are mapped to a single kernel thread
D. Many kernel threads are mapped to a single user thread

45.The thread mapping model considered many-to-many
A. There is no limit on the number of kernel threads
B. It can only be implemented on multiprocessor systems
C. Causes all threads in a process to block, even just one these threads execute a blocking system call
D. It is the compromise between a purely user thread implementation level and a purely kernel level

46. We speak of parallelism when: (exit 3 times already)
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
D. All of the above answers are correct

47.We talk about competition when: (already exited 2 times)
A. 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
D. Multi-threaded processes run on multicore CPUs

48.Communication between threads of the same process compared to that between different processes:
A. a. It is slower because the threads are handled by high-level libraries
B. It is faster because threads do not perform context switches
C. It is faster since the threads share the same space as addressing
D. There is no material 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 via a specific 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 system's own processes operational (i.e., not user processes)

50.The use of a lock synchronization primitive requires 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 the above conditions must be verified

51.The acquisition of a lock:
A. It must happen atomically, preventing the scheduler from interrupting the acquisition

B. Requires hardware instruction support atomic
C. It is mandatory for the operating system to disable them interruptions
D. None of the above answers are correct

52.A semaphore can be used for: (already exited 2 times)
A. Force scheduling policies between processes/threads
B. Access the kernel code
C. The exchange of messages between processes/threads
D. Handle interrupts coming to the CPU

53.Invoking the wait() method on a semaphore whose value is equal to 2:
A. Leave the semaphore value at 2 unchanged and continue the process who executed the invocation (net of scheduling policies)
B. Decrement the value of the semaphore to 1 and block the process it has performed the invocation
C. Increase the value of the semaphore to 3 and continue the process executed the invocation (net of scheduling policies)
D. Decrease the value of the semaphore to 1 and continue the process 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 that allows you to disable interrupts
C. It is an atomic instruction that allows multiple values to be updated register simultaneously
D. It is an atomic instruction that allows you to reset the value of a semaphore

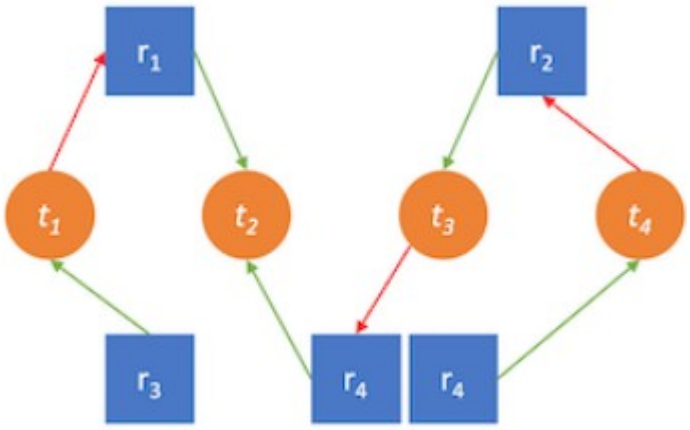
55.The difference between deadlock and starvation lies in the fact that:
A. Refer to user code and system code (respectively)
B. In the case of starvation the entire system is completely blocked
C. There is no difference
D. In the case of deadlock the entire system is completely blocked

56.We consider 2 processes/threads 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 that R1 and R2 are internal registers and val is one variable in main memory initially equal to 73, what will be the value stored in val at the end of the execution of P and C if the 6 instructions total (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. There is insufficient data to answer the question

57.We consider 2 processes/threads 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 that R1 and R2 are internal registers and that val is a variable in main memory initially equal to 24, what will be the value stored in val at the end of the execution of P and C if the 6 instructions in total (of P and C) will they be scheduled in the following order: C1, P1, P2, C2, C3, P3?
A. 23
B. 24
C. 25
D. There is insufficient data to answer the question

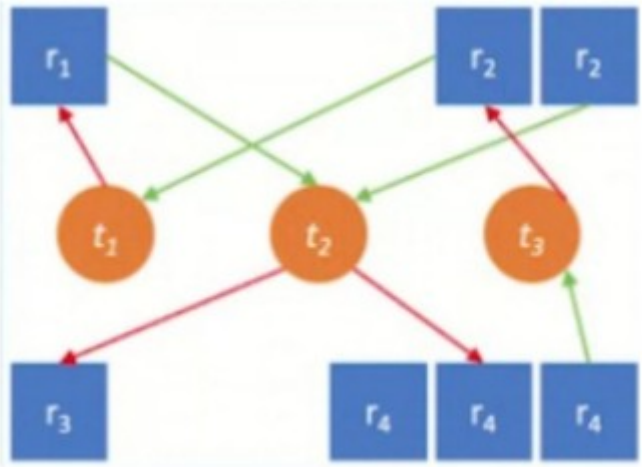
58.We consider 2 processes/threads Producer (P) and Consumer (C) whose code consists of the following instructions:
P (P1) R1 = val; P(2) R1 += 1; (P3) val= R1;
C (C1) R2 = val; (C2) R2 -= 1; (C3) val = R2;
Assuming that R1 and R2 are internal registers and that val is a variable in main memory initially equal to 19, what will be the value stored in val at the end of the execution of P and C if the 6 instructions in total (of P and C) will they be scheduled in the following order: C1, P1, C2, P2, P3, C3?
A. 18
B. 19
C. 20
D. There is insufficient data to answer the question

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



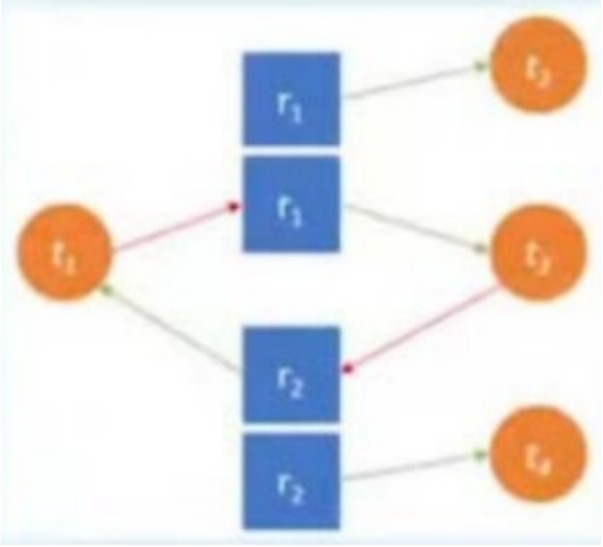
A. It depends on the operating system's scheduler choices
B. It has deadlocks
C. It has no deadlocks
D. It's impossible to answer

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



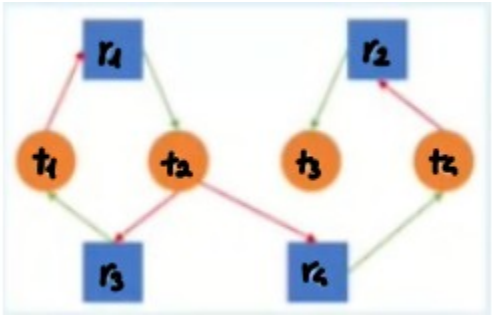
A. It depends on the operating system's scheduler choices
B. Deadlock present
C. It has no deadlocks
D. It's impossible to answer

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



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

Main Memory

63.The term address binding means:
A. The process of translating logical addresses to physical addresses
B. The process of initializing a program's global variables
C. The process of linking compiled code to any external libraries
D. None of the above answers are correct

64. Swapping allows you to:
A. Implement dynamic code relocation of a process
B. Solve the external fragmentation problem
C. Temporarily move processes that are not currently running to disk
D. Swap memory areas occupied by two or more processes

65.The "paged" management of memory (paging):
A. Expects the logical address space of a process to be non-contiguous and divided into blocks of fixed dimensions (pages)
B. It does not require any hardware support to be implemented efficient way
C. Expects the physical address space of a process to be non-contiguous and divided into blocks of fixed dimensions (frames)
D. It solves the internal fragmentation problem

66.The TLB (Translation Look-aside Buffer) cache
A. It is shared among all processes in the system
B. It allows a faster translation of logical addresses on average
C. Contains a subset of the page table entries
D. All of the above answers are correct

67.The size (i.e., 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 each person's memory access requests process

C. It depends on the (fixed) size of the pages
D. Varies dynamically depending on the process

68.The size (i.e., the number of entries) of the page table:
A. It varies dynamically depending on the process
B. It is directly proportional to the (fixed) size
C. It is inversely proportional to the (fixed) size of the pages
D. It adapts according to each person's memory access requests
process

69.A compiler generates the logical address 576 to refer to a certain
physical memory location. Assuming address translation
occurs via static relocation with base physical address = 24, whichever it will be
the corresponding physical address?
A. 576
B. 552
C. 600
D. There is insufficient data to answer the problem

70.Consider a process of size equal to 2,488 bytes and a block of
free memory of size equal to 2,699 bytes. In this case, hiring
the contiguous memory allocation constraint, the most convenient choice
è:(already released 2 times)
A. Allocate the entire block to the process, wasting 211
bytes(internal fragmentation)
B. Allocate the portion of the block needed for the process and add
the remaining 211 bytes to the list of free blocks (fragmentation
external)
C. Wait until there is a block of multiple sizes than that
of the process
D. Wait until there is a smaller block size suitable for
contain the process

71.Consider a process of size 4,996 and a block of memory
free with a size of 5,016 bytes. In this case, assuming the constraint
of contiguous memory allocation, the most convenient choice is:
A. Wait until there is a smaller size block suitable for
contain the process
B. Allocate the entire block to the process, wasting 20
bytes(internal fragmentation)
C. Wait until there is a block of multiple sizes than that
of 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 contiguously allocated in main memory. If the

list of free memory blocks contains the following elements: A, B, C, D le whose dimensions are respectively 102 KiB, 99 KiB, 256 KiB and 128 KiB, 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 contiguously allocated in main memory. If the list of free memory blocks contains the following elements: A, B, C, D, E, F whose dimensions are respectively 300 KiB, 600 KiB, 350 KiB, 200 KiB, 750 KiB and 125 KiB, which block will be allocated for P assuming a policy Worst-Fit?
A. block B
B. The request cannot be satisfied, 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 contiguously in main memory. If the list of free memory blocks contains the following elements: A, B, C, D le whose dimensions are respectively 105 KiB, 916 KiB, 129 KiB and 80 KiB, 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 contiguously in main memory. If the list of free memory blocks contains the following elements: A, B, C, D,E,F whose dimensions are respectively 300 KiB, 600 KiB, 350 KiB, 200 KiB,750 KiB and 125 KiB 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 contiguously in main memory. If the list of free memory blocks contains the following elements: A, B, C, D,E,F whose dimensions 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 contiguously in main memory. If the
list of free memory blocks contains the following elements: A, B, C, D le
whose dimensions are respectively 36 KiB, 90 KiB, 42 KiB and 35 KiB, 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 you have a memory M with a capacity of 4 KiB, i.e. 4,096
bytes. Assuming that addressing occurs with word length
(word size) equal to 2 bytes and that M uses a 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 (internal to the page), respectively?
A. p=6; offset=5
B. b.p=7; offset=5
C. p=5; offset=7
D. p=5; offset=6

79.Consider a memory M with a capacity of 512 bytes with frames of
size equal to 16 bytes. Given the address of byte 197, what will the address be
of page (p) and the offset (internal to the page): (happened twice)
A. p=5; offset=12
B. The 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 of 100 bytes with frames of
size equal to 10 bytes. Given the address of byte 37, what will the address be
of page (p) and the offset (internal to the page).
A. p=3; offset=7
B. The data is insufficient to answer the question
C. p=7; offset=3
D. p=0; offset=37

81.Consider a process of size equal to 2,097 bytes and a block of
free memory of size equal to 2,104 bytes. In this case, hiring
the contiguous memory allocation constraint, the most convenient choice is:
A. Wait until there is a block of multiple sizes than that
of the process
B. Allocate the entire block to the process, wasting 7 bytes
(internal fragmentation)

C. Wait until there is a smaller size block suitable for
contain 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 you have a memory M with a capacity of 2 KiB, i.e. 2,048
bytes. Assuming that addressing occurs with word length
(word size) equal to 4 bytes, how many bits are necessary to address the words
contained in M?
A. 2
B. 9
C. 11
D. There is insufficient data to answer the problem

83.Suppose you have a memory M with a capacity of 4 KiB or 4.096
bytes. Assuming that addressing occurs with word length
(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. There is insufficient data to answer the question

84.Suppose you have a memory M with a capacity of 8 KiB, i.e. 8,192
bytes. Assuming that addressing occurs 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
entry number) has the corresponding page table T?
A. There is insufficient data to answer the question
B. 13
C. 64
D.7

85.Suppose you have a memory M with a capacity of 8 KiB, i.e. 8,192
bytes. Assuming that addressing occurs with word length
(word size) equal to 4 bytes and that M uses a paged management with blocks of
size equal to S = 256 bytes, what will be the number of entries in the
corresponding page table T?
A. 32
B. 2,048
C. 8
D.5

86.Suppose you have a memory M with a capacity of 16 KiB, i.e. 16,384
bytes. Assuming that addressing occurs with word length
(word size) equal to 4 bytes and that M uses a paged management with blocks of
size equal to S = 64 bytes, what will be the number of entries in the

corresponding page table T?
A. 8
B. 14
C. 256
D. There is insufficient data to answer the problem

87.Consider an operating system that uses 21-bit logical addresses, address 16-bit physical and paged memory where each page has size 2 KiB(2048 bytes). What is the maximum size of physical memory supported from the system? (exit 2 times already)
A. 32 KiB
B. 64 KiB
C. 2 MiB
D. 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. Run a process directly from storage devices secondary (e.g., disk)

89.If an idempotent instruction generates a page fault:
A. The process of which the statement is a part ends
B. Idempotent instructions cannot generate page faults
C. The statement will no longer be executed once returned from page fault management
D. The statement will be executed again when you return from managing the page fault

90.The problem of external fragmentation:
A. It needs hardware support to fix
B. It is not fixable unless the system is rebooted
C. It is a consequence of the contiguous memory allocation constraint
D. Causes a hardware interruption

91.The problem of external fragmentation
A. It is not fixable unless the system is rebooted
B. Causes a hardware interrupt
C. Requires hardware support to resolve
D. It is due to the allocation/deallocation of contiguous blocks of memory

92.The working set is:
A. Fixed for every 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 replacement of
frame through the use of a timestamp. For each access request to a
certain frame is needed:
A. increment a counter variable
B. Update the timestamp value with the current one
C. Set a validity bit
D. None of the above answers are correct

94.Given a memory composed of 3 physical frames and a process composed of 5
virtual pages: A, B, C, D, E, calculate the number of page faults that occur
in response to the following requests from the process: B, C, C, B, A, E, B, A,
E, D, B. Assume that no page of the process is initially
loaded into memory and that an LRU replacement algorithm is used
pages.
A.4
B.5
C. 6
D.7

95.Given a memory composed of 3 physical frames and a process composed of 5
virtual pages: A, B, C, D, E, calculate the number of page faults that occur
in response to the following requests from the process: D, B, A, C, C, E, A, D,
B, E, D, A. Assume that no page of the process is initially
loaded into memory and that an LRU replacement algorithm is used
pages.
A. 10
B. 7
C. 9
D.6

96.Given a memory composed of 3 physical frames and a process composed of 5
virtual pages: A, B, C, D, E, calculate the number of page faults that occur
in response to the following requests from the process: C,B,C,B,A,E,B,A. Yes
assume that no process pages are initially loaded into
memory and use an LRU page replacement algorithm.
A. 2
B.4
C.5
D.1

97.Given a physical memory composed of 3 physical frames and a compound process
from 5 virtual pages: A, B, C, D, E, calculate the number of page faults that occur

verify against the following requests from the process: A, B, E, C,
E, D, D, A, B. Assume that no process page is initially
loaded into memory and that a FIFO replacement algorithm is used
pages.
A. 6
B. 7
C. 4
D.8

98.Given a memory composed of 3 physical frames and a process composed of 5
virtual pages: A, B, C, D, E, calculate the number of page faults that occur
in response to the following requests from the process: D, A, C, B, B, A, C, B,
D, E, A. Assume that no page of the process is initially
loaded into memory and that a FIFO replacement algorithm is used
pages.
A. 6
B. 7
C.5
D.4

99. Given a memory composed of 3 physical frames and a process composed of 5
virtual pages: A, B, C, D, E calculate the number of page faults that occur
in response to the following requests from the process: E, B, E, C, D, E, A, B,
E. Assume that no process pages are initially loaded into
memory and use a FIFO page replacement algorithm.
A. 7
B.8
C. 6
D.5

Secondary Memory

100. The contiguous allocation of a file on disk: (exit 4 times already)
A. It is great for both direct (random) and sequential access
B. It presents the problem of fragmentation
C. Requires the maintenance of free blocks within an appropriate one
data structure
D. All of the above answers are correct

101. Contiguously allocating 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 one

specific sector
B. Includes transfer time to main memory
C. It is the time it takes for the disk to position its heads on one specific cylinder
D. It is negligible compared to the entire time needed to transfer the data

103. A disk is made up of 15 cylinders, each with a capacity of 500 MB. What is the total capacity of the disk?
A. 7.5GB
B. b. 75GB
C. c. 750 MB
D. d. The data is insufficient to answer the problem

104. Suppose the physical memory access time is = 50 nsec. and that the time for managing a page fault is equal to 15 msec. Assuming that the probability of a page fault occurring is = 0.0002, what is the expected overall 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 = 25 nsec. and that the time for managing a page fault is equal to 30 msec. Assuming that the probability of a page fault occurring is = 0.005, what is the expected overall memory access time?
A. ~150.025 microsec
B. ~15.025 nsec
C. ~150.025 nsec
D. ~15.025 microsec

106. Suppose the physical memory access time is = 50 nsec. and that the time for managing a page fault is equal to 25 msec. Assuming that the average memory access time is 0.5 microsec, what is the probability of a page fault occurring?
A. ~0.02%
B. ~0.2%
C. ~0.002%
D. ~0.0002%

107. Suppose the physical memory access time is = 60 nsec. and that the time for managing a page fault is equal to 5 msec. What should be the value of the probability that a fault occurs ( ) if you want to guarantee that the expected memory access time is at most 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 outermost cylinder/track with respect to the center of the disk), whose head is initially located on cylinder 42. Calculate the number of cylinders/tracks crossed by the disk head, assuming that the sequence of requests: 74, 50, 32, 55, 81 is managed by a scheduling algorithm SSTF (Shortest Seek Time First) 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 outermost cylinder/track with respect 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 managed by a scheduling algorithm FCFS (First Come First Served) and neglecting the turnover time.
A. 290
B. 240
C. 238
D. 265

110. Consider a magnetic disk composed of 200 cylinders/tracks, numbered from 0 to 199 (0 index of the outermost cylinder/track with respect 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 managed by an algorithm FCFS (First Come First Served) scheduling and neglecting the time of rotation.
A. 595
B. 558
C. 650
D. 638

111. Consider a magnetic disk composed of 200 cylinders/tracks, numbered from 0 to 199 (0 index of the outermost cylinder/track with respect 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 managed by an algorithm FCFS (First Come First Served) scheduling and neglecting the time of rotation.
A. 650
B. 522

C. 638
D. 595

112. Consider a magnetic disk composed of 100 cylinders/tracks, numbered from 0 to 99 (0 index of the outermost cylinder/track with respect 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 scheduling algorithm SCAN (non-optimized), that the head is moving outwards (i.e., towards the cylinders with lower numbers) and neglecting the rotation time.
A. 76
B. 87
C. 46
D. 93

113. Consider a magnetic disk composed of 100 cylinders/tracks, numbered from 0 to 99 (0 index of the outermost cylinder/tracks with respect 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 scheduling algorithm C-SCAN (non-optimized), that the head is moving outwards (i.e., towards the cylinders with lower numbers) and neglecting the time of rotation.(already released 2 times)
A. 189
B. 193
C. 187
D. 196

114. The total transfer time for a disk I/O operation magnetic is equal to 30 msec. Knowing that: the overall seek time is equal to 18 msec, the overall 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)(exit already 3 times)
A. 9,375 MB
B. b. 7.5 MB
C. 937.5 KB

D. There is insufficient data to answer the question
115. The total transfer time for a disk I/O operation magnetic is equal to 40 msec. Knowing that: the overall seek time is equal to 18 msec, the overall 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. 9,375 MB
B. 70 MB
C. 70 KB
D. There is insufficient data to answer the question

116. The total transfer time for a disk I/O operation
magnetic is equal to 36 msec. Knowing that the overall seek time is equal to
13 msec and that 2MB were transferred at a speed of 1 Gbit/sec qual
is the rotational delay of the disk? (Remember that 1 B = 1 byte = 8 bits)
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 one entry for each file in use
C. Maintains a counter for each file in use
D. All of the above 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 location on disk of each file referenced by
a process
C. Contains a pointer to the global table of open files for each file
reported by a trial
D. It is shared among multiple processes

119. A possible example of an application that requires sequential access
to a file is:
A. A compiler
B. A search system within a database
C. A telephone contact tracing system
D. None of the above answers are correct

120. Indexed file allocation is preferable when the file in question:
A. It is small in size, regardless of which way it comes
accessed
B. It is large, regardless of which way it comes
accessed
C. It is large and is typically accessed sequentially
D. It is large and is typically accessed in a manner
casual(direct)

121. File allocation based on linked lists is preferable
when the file in question
A. It is small in size, regardless of which way it comes
accessed

B. It is large in size and is typically accessed in a manner
casual(direct)
C. It is large, regardless of which way it comes
accessed
D. It is large and is typically accessed sequentially

122. In a UNIX-like system, a file that has the following privileges: 101000000:
A. Allows only the owner of the file to exercise reading rights and
writing (to the file)
B. Allows only the owner of the file to exercise reading rights
execution (on file)
C. Allows only the owner of the file to exercise write and rights
execution (on file)
D. It does not give any rights to the owner of the file

123. In a UNIX-like system, a file that has the following privileges: 011000000:
A. Allows only the owner of the file to exercise reading rights and
writing (to the file)
B. Allows only the owner of the file to exercise reading rights
execution (on 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 and edit rights
execution (on file)

124. In a UNIX-like system, a file that has the following privileges: 111101101:
A. Allows the file owner to exercise all rights (to the file) e
gives other users only write and execute rights
B. Allows the owner of the file to exercise all rights (on the file) e
provides other users with only read and write rights
C. Allows the owner of the file to exercise all rights (on the file) e
gives other users only read and execute rights
D. allows anyone to exercise all rights (on the file)

125. The UNIX command ln file_1 file_2
A. Create a hard link with the file file_2(source) where name is
file_1(destination)
B. Create a hard link with the file file_1(source) whose name is
file_2(destination)
C. Create a soft link with the file file_1(source) whose name is
file_2(destination)
D. Create a soft link with the file file_2(source) whose name is
file_1(destination)

126. The UNIX command ln -s file_1 file2:
A. Create a hard link with the file file_2(source) where name is
file_1(destination)
B. Create a hard link with the file file_1(source) whose name is

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

127. Consider an organized file system with indexed file descriptors multi-level (multi-level indexed files), containing direct references to 10 blocks to which a 100 block indirection level is added and a further double level of indirect reference, again of 100 blocks each. Assuming that each block has a size of 2 KiB, qual is the maximum file size supported?
A. ~20.2 KB
B. ~20.2 MB
C. ~20.7 KB
D. ~20.7 MB