Sistemi Operativi 1

Struttura di un sistema operativo

1. Il sistema operativo:

   B Costituisce l'interfaccia tra la macchina fisica (hardware) e le applicazioni utente

2. In un sistema operativo *microkernel:*
   A. Alcune delle funzionalità sono implementate in spazio utente anziché all'interno del kernel

3. In un sistema operativo strutturato secondo un approccio microkernel
   D Ad eccezione delle funzionalità fondamentali, implementa tutto il restoin spazio utente

4. L'insieme di istruzioni del livello macchina:
   D Tutte le risposte precedenti sono corrette
5. I registri interni della CPU e la cache sono unità di memoria:
   B Gestite interamente dall'architettura a livello hardware

6. La transizione da user a kernel mode avviene quando:
   D Scade il quanto di tempo assegnato al processo in esecuzione
7. Il device controller di un dispositivo di I/O:
   D. Tutte le risposte precedenti sono corrette

8. Le chiamate di sistema:
   D. Devono essere implementate in spazio kernel

9. Una chiamata di sistema bloccante:
   C. Interrompe temporaneamente il processo che la esegue

10. Il system call handler:
    D Gestisce le chiamate di sistema tramite la system call table

11. Il codice generico del system call handler:
    B È indicizzato tramite la interrupt vector table (IVT)

12. L'interrupt vector table(IVT):
    B E'una struttura dati che contiene puntatori ai vari gestori(handler) delleinterruzioni
13. La system-call table:(uscita già 2 volte)
    A. Contiene tante entry quanto sono le chiamate di sistema supportar

14. La system-call table è una struttura dati gestita:
    D. Dal kernel del sistema operativo

15. Se si cambia l'implementazione di una chiamata di sistema esistente:
    D Non è necessario modificare il codice utente che ne fa uso, a patto che **non** cambi anche l'interfaccia (API) della chiamata di sistema

16. Un processore impiega 5 cicli di clock per eseguire un'istruzione (CPI = 5), ossia per completare l'intero ciclo fetch-decode-execute. Assumendo che la frequenza di clock del processore sia pari a 5 MHz, quante istruzioni è in grado di eseguire in un secondo? (Si ricordi che 1 MHz = 1*10^6 cicli al secondo)
    D. 1*10^6

17. Data una CPU multicore con $m$unità(cores), il numero di processi/thread che ad un certo istante si trovano nella "coda" di esecuzione(running):
    D. E' al massimo pari a m

I Processi

18. La creazione di un nuovo processo da parte di un processo avviene tramite:(uscito 2 volte)
    A. Una chiamata di sistema

19. Il sistema operativo tiene traccia dello stato di un processo tramite:
    D Un apposito campo all'interno del process control block (PCB)
20. Un processo in esecuzione sulla CPU passa in stato ready quando:
    A. Riceve un segnale di interruzione da parte di un dispositivo di I/O

21. Un processo in esecuzione sulla CPU passa in stato waiting quando:
    C Apre una connessione di rete (ad es., un socket TCP)

22. Un processo in esecuzione sulla CPU passa in stato waiting quando:
    A. Fa richiesta di input da parte dell'utente

23. Un processo in esecuzione sulla CPU passa in stato waiting quando:

    B L'utente trascina il dispositivo di puntamento(e.g. mouse)
24. Quanti processi saranno presenti nel sistema a seguito di queste chiamate: pid_1 = fork(); pid_2 = fork(); pid_3 = fork();?
    A. 8

25. Data la porzione di codice in figura, indicare quale sarà il valore della variabile value che verrà stampato alla line 18:

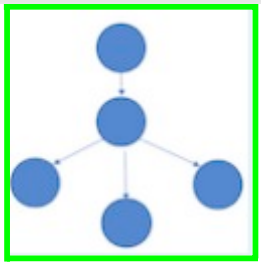
A. 5

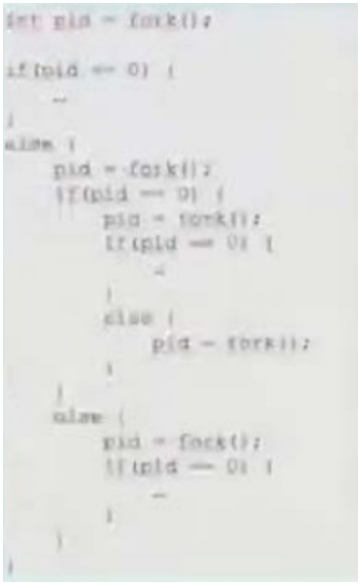
26. Data la porzione di codice in figura, indicare il corrispondente albero dei processi generati:

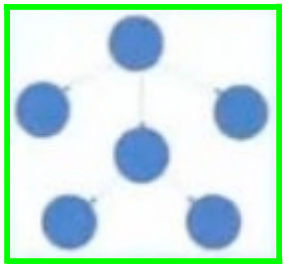
```
int pid = fork();

if(pid == 0) {

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

B. 

27. Data la porzione di codice in figura, indicare il corrispondente albero dei processi generati  :

B. 

28. I processi CPU-bound che non eseguono richieste di I/O:
    D Possono non rilasciare mai la CPU volontariamente

Politiche di Scheduling

29. Lo scheduler della CPU si attiva:
    D Tutte le risposte precedenti sono corrette
30. Lo scheduling preemptive(basato su time slice o quanto temporale):
    D Fornisce un limite superiore al tempo di CPU assegnato a ciascunprocesso
31. In un sistema uniprocessore (single core) time-sharing in cui i processi in esecuzione sono tutti **puramente** CPU-bound:(già uscita 2 volte)
    D L'impiego dei multi-threading non costituisce alcun vantaggio
32. In caso di scheduling preemptive, lo scheduler interviene:
    D. Tutte le risposte precedenti sono corrette

33. Se un processo arriva nella coda dei pronti all'istante $t_0 = 2$e termina all'istante $t_f = 15$, il suo tempo di turnaround equivale a:
    A. 13
34. Se un processo arriva nella coda dei pronti all'istante $t_0 = 3$ e termina all'istante $t_f = 25$, il tempo di attesa equivale a:

35. Si considerino i 5 processo della figura seguente e 3 politiche di scheduling: FCFS, SJF (non-preemptive) e RR con time slice pari a 2 unità di tempo. Qual è la politica che garantisce il minor tempo di attesa (in coda pronti) al processo C?

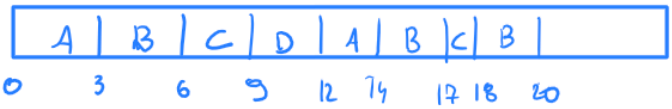
Job	$T_{arrival}$	T_{burst}
A	0	2
B	0	1
C	0	8
D	0	4
E	0	5

A. FCFS

36. Calcolare il tempo medio di attesa (average waiting time) dei seguenti processi, assumendo una politica di scheduling round robin con time slice = 3, nessuna attività di I/O e context switch trascurabile:

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

B. 6.75



$t_w^A = 14 - 0 - 5 = 9$

$t_w^B = 20 - 2 - 8 = 10$

$t_w^C = 18 - 7 - 4 = 7$

$t_w^D = 12 - 3 - 8 = 1$

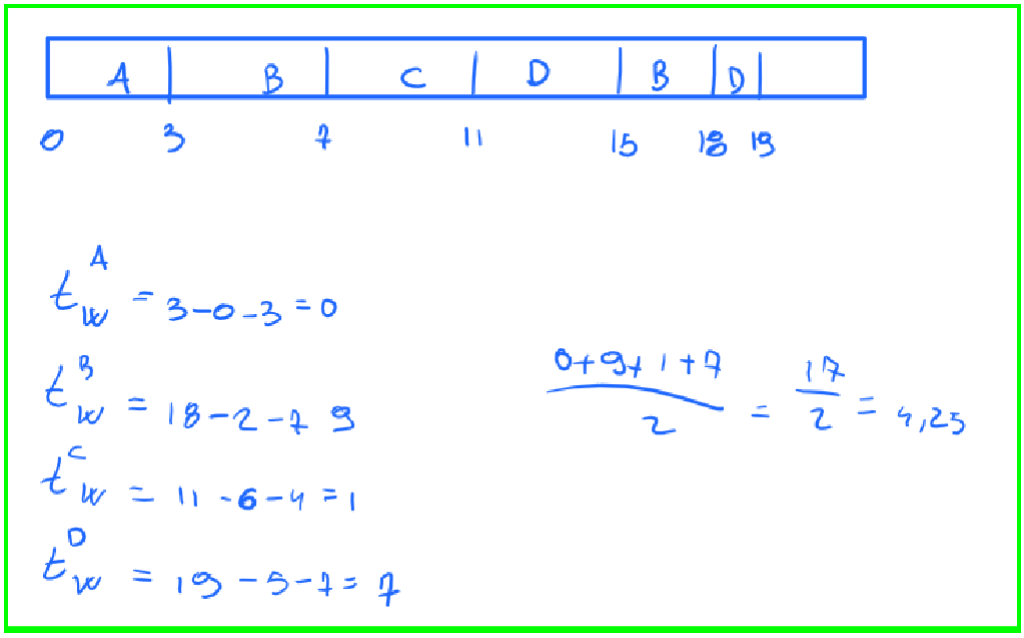
$\frac{9+10+7+1}{4} \quad \frac{27}{4} = 6{,}75$

37. Calcolare il tempo medio di attesa (average waiting time) dei seguenti processi, assumendo una politica di scheduling Round Robin con time slice q= 4. Nel calcolo, si consideri il tempo necessario ad eseguire il context

switch trascurabile:

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

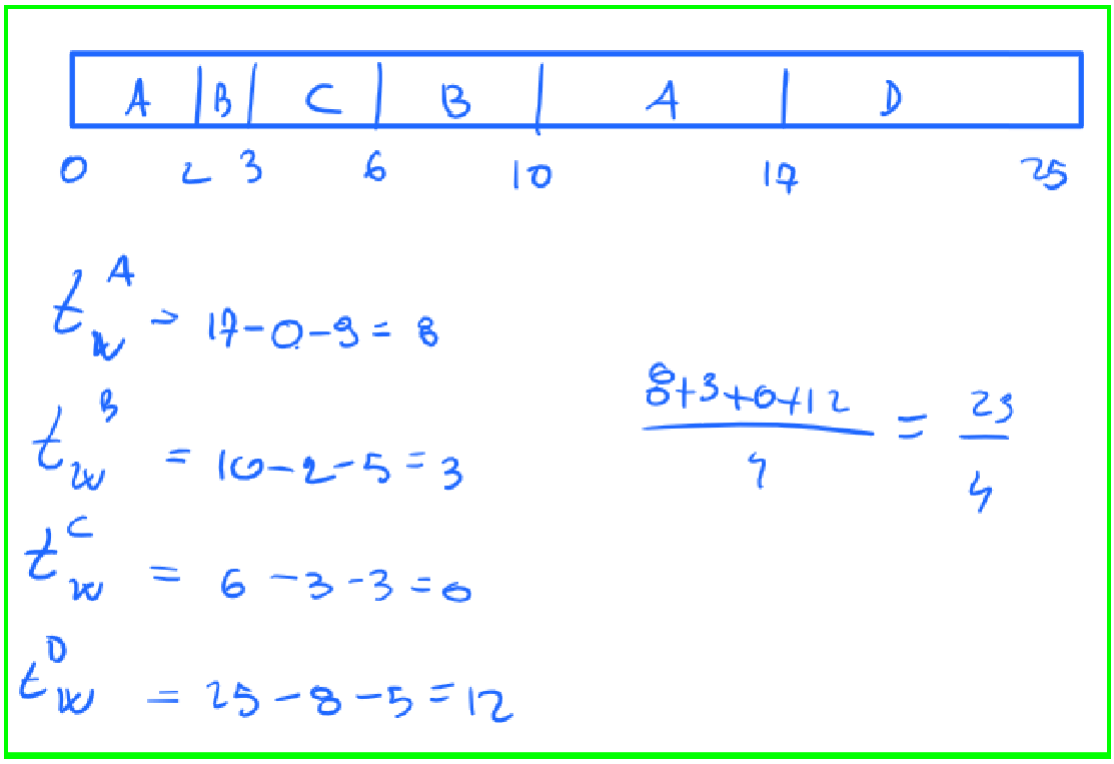
B. 4.25



38. Calcolare il tempo medio di attesa (average waiting time) dei seguenti processi, assumendo una politica di scheduling Shortest Job First preemptive (SJF). Nel calcolo, si consideri trascurabile il tempo necessario ad eseguire il context switch:

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

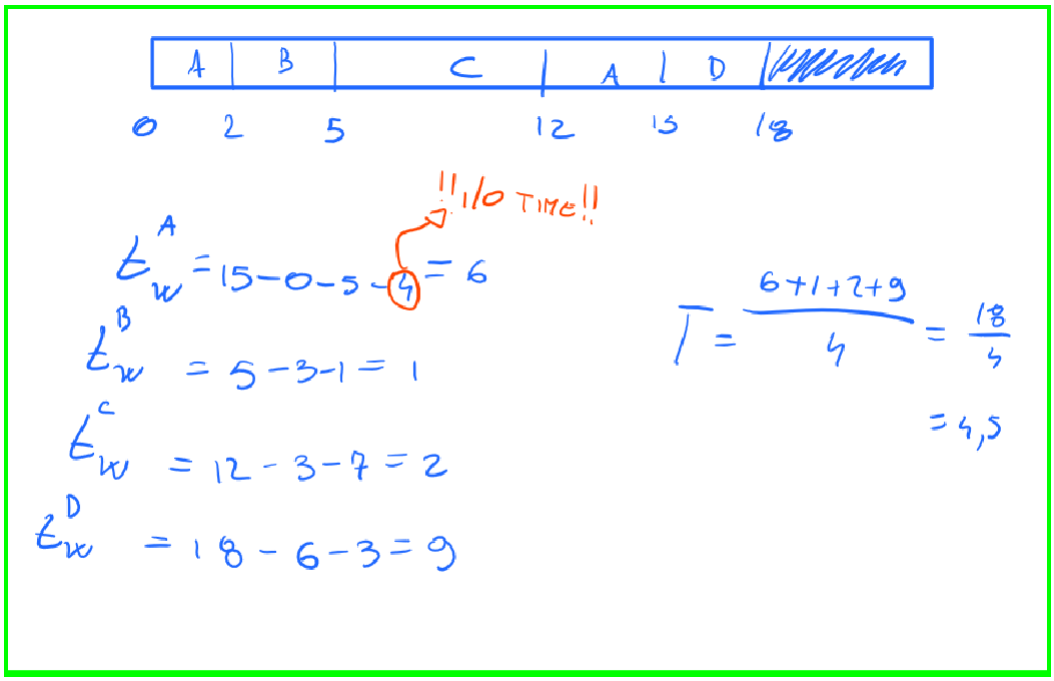
B. 5.75



39. Calcolare il tempo medio di attesa (average waiting time) dei seguenti processi, assumendo una politica di scheduling First Come First Served (FCFS) e che il processo A esegua all'istante t=2 una chiamata di I/O che si completerà dopo 4 unità di tempo, ossia all'istante t=6. Nel calcolo, si consideri trascurabile il tempo necessario ad eseguire il context switch:

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

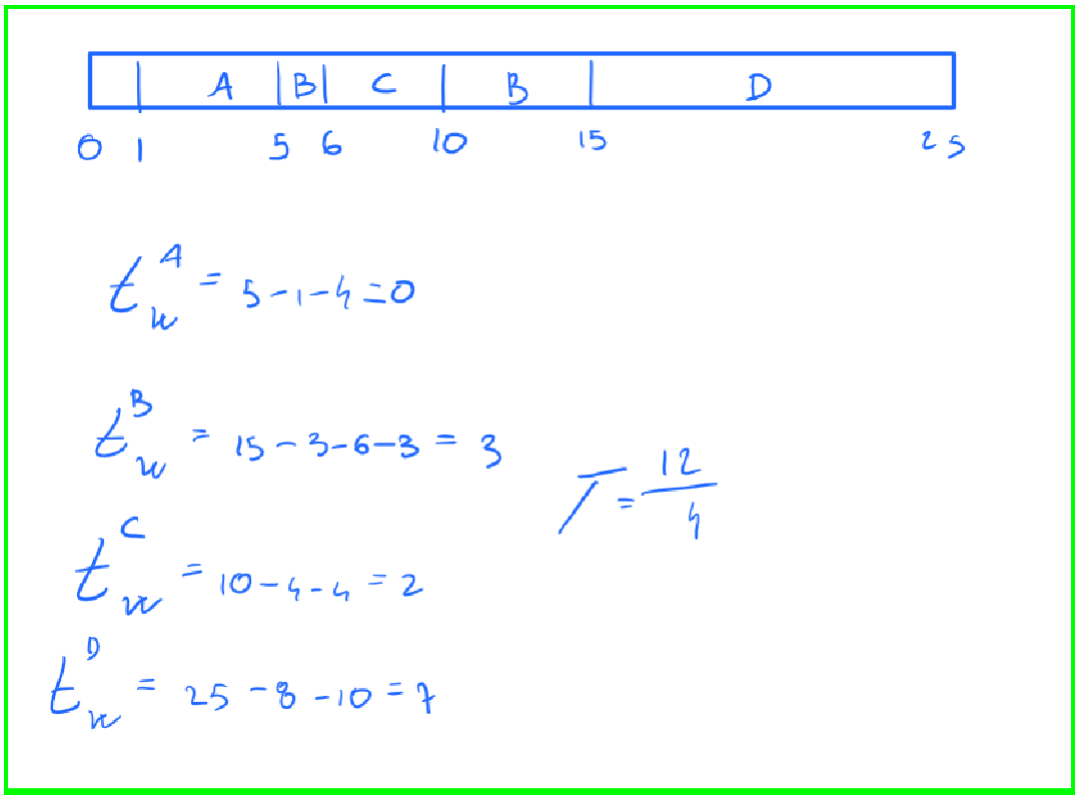
A. 4.5



40. Calcolare il tempo medio di attesa (average waiting time) dei seguenti processi, assumendo una politica di scheduling First Come First Served (FCFS) e che il processo B esegua all'istante t=6 una chiamata di I/O che si completerà dopo 3 unità di tempo, ossia all'istante t=9. Nel calcolo, si consideri trascurabile il tempo necessario ad eseguire il context switch:(uscita 2 volte)

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

C. 4



I Threads e la Sincronizzazione

41. I thread di uno stesso processo condividono:
    B Le variabili globali
42. Lo user thread:
    C. E' gestito in spazio utente tramite un'apposita libreria

43. Nel modello di thread mapping cosiddetto one-to-one:
    D Consente di gestire i thread a livello del kernel del sistema operativo
44. Nel modello di thread mapping cosiddetto many-to-one:
    C Molti user thread sono mappati su un singolo kernel thread
45. Il modello di thread mapping considerato many-to-many
    D E' il compromesso tra un'implementazione dei thread puramente user level e una puramente kernel level

46. Si parla di parallelismo quando:(uscita già 3 volte)
    C Vengono eseguiti processi multi-threaded su CPU multicore
47. Si parla di concorrenza quando:(uscita già 2 volte)
    A. Vengono eseguiti processi multi-threaded su CPU single core
48. La comunicazione tra thread dello stesso processo rispetto a quella tra processi diversi:
    C. È più veloce poiché i thread condividono lo stesso spazio di

49. Il kernel thread:
    C È la più piccola unità schedulabile sulla CPU dal sistema operativo

50. L'uso di una primitiva di sincronizzazione lock prevede che:
    D. Tutte le condizioni precedenti devono essere verificate

51. L'acquisizione di una lock:
    A. Deve avvenire in modo atomico, evitando che lo scheduler interrompa l'acquisizione

52. Un semaforo può essere utilizzato per:(uscito già 2 volte)
    A. Forzare le politiche di scheduling tra processi/thread

53. L'invocazione del metodo wait() su un semaforo il cui valore è pari a 2:
    D Decrementa il valore del semaforo a 1 e fa proseguire il processo che ha eseguito l'invocazione (al netto delle politiche di scheduling)

54. L'istruzione test-and-set:
    A. È un'istruzione atomica che consente di implementare le primitive di sincronizzazione

55. La differenza tra deadlock e starvation risiede nel fatto che:
    D Nel caso di deadlock tutto il sistema è completamente bloccato

56. Si considerano 2 processi/thread **Produttore** (**P**) e **Consumatore** (**C**) il cui codice è composta dalle seguenti istruzioni:

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

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

    Assumendo che R1 e R2 siano registri interni e che val sia una variabile in memoria principale inizialmente pari a 73, quale sarà il valore memorizzato in val al termine dell'esecuzione di **P** e di **C** se le 6 istruzioni complessive (di **P** e **C**) verranno schedulate secondo il seguente ordine: P1,P2,P3,C1,C2,C3?
    B. 73

57. Si considerano 2 processi/thread **Produttore** (**P**) e **Consumatore** (**C**) il cui codice è composta dalle seguenti istruzioni:

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

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

    Assumendo che R1 e R2 siano registri interni e che val sia una variabile in memoria principale inizialmente pari a 24, quale sarà il valore memorizzato in val al termine dell'esecuzione di **P** e di **C** se le 6 istruzioni complessive (di **P** e **C**) verranno schedulate secondo il seguente ordine: C1, P1, P2, C2, C3, P3?
    C. 25

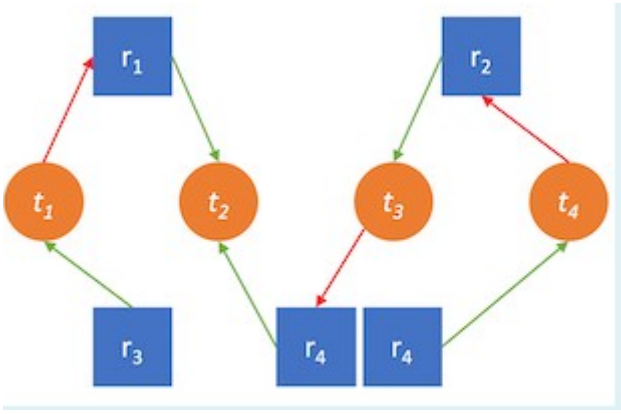
58. Si considerano 2 processi/thread **Produttore** (**P**) e **Consumatore** (**C**) il cui codice è composta dalle seguenti istruzioni:

    P (P1) R1 = val; P(2) R1 += 1; (P3) val= R1;

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

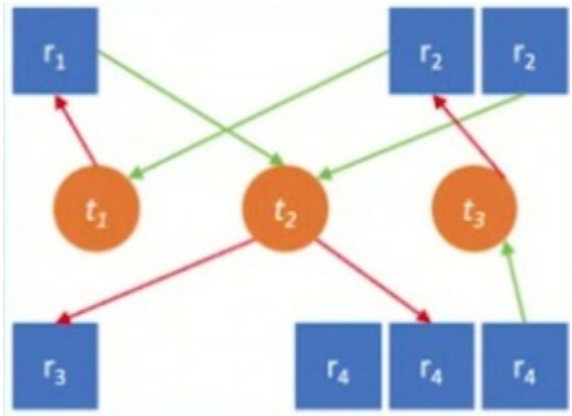
    Assumendo che R1 e R2 siano registri interni e che val sia una variabile in memoria principale inizialmente pari a 19, quale sarà il valore memorizzato in val al termine dell'esecuzione di **P** e di **C** se le 6 istruzioni complessive (di **P** e **C**) verranno schedulate secondo il seguente ordine: C1, P1, C2, P2, P3, C3?

59. Il seguente Resource Allocation Graph (RAG) mostra un sistema il cui stato:



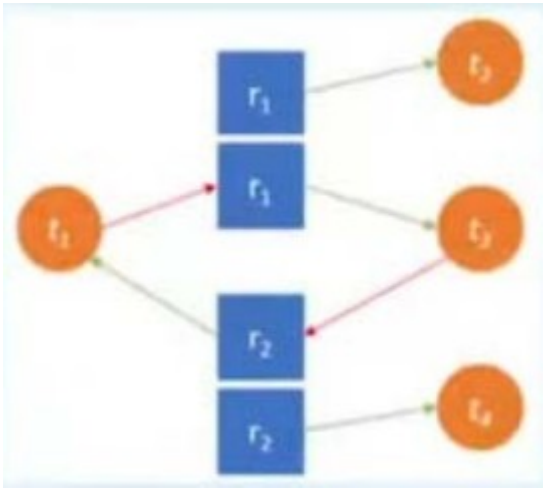
Non presenta deadlock

60. Il seguente Resource Allocation Graph (RAG) mostra un sistema il cui stato:



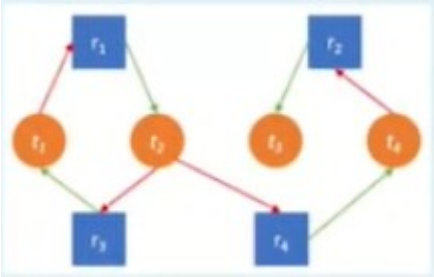
A. Non presenta deadlock

61. Il seguente Resource Allocation Graph (RAG) mostra un sistema il cui stato:



Sicuramente non presenta deadlock

62. Il seguente Resource Allocation Graph(RAG) mostra un sistema che:



A. Sicuramente presenta deadlock

La Memoria Principale

63. Con il termine address binding si intende:
    A. Il processo di traduzione da indirizzi logici a indirizzi fisici
64. Lo swapping consente di:
    C Trasferire temporaneamente su disco i processi che non sono attualmente in esecuzione
65. La gestione "paginata" della memoria (paging):
    C Prevede che lo spazio di indirizzamento fisico di un processo sia non-contiguo e suddiviso in blocchi di dimensioni fissate (frames)
66. La cache TLB (Translation Look-aside Buffer)
    D. Tutte le risposte precedenti sono corrette

67. La dimensione (i.e., il numero di entry) della page table:
    C Dipende dalla dimensione (fissata) delle pagine
68. La dimensione (i.e., il numero di entry) della page table:
    C E' inversamente proporzionale alla dimensione (fissata) delle pagine

69. Un compilatore genera l'indirizzo logico 576 per riferirsi ad una certa locazione di memoria fisica. Assumendo che la traduzione degli indirizzi avvenga tramite rilocazione statica con indirizzo fisico base = 24, quale sarà l'indirizzo fisico corrispondente?
    C. 600
70. Si consideri un processo di dimensione pari a 2,488 bytes e un blocco di memoria libero di dimensione pari a 2,699 bytes. In questo caso, assumendo il vincolo di allocazione contigua della memoria, la scelta più conveniente è:(uscito già 2 volte)
    A. Allocare l'intero blocco al processo, sprecando 211

71. Si consideri un processo di dimensione pari a 4,996 e un blocco di memoria libero di dimensione pari a 5,016 bytes. In questo caso, assumendo il vincolo di allocazione contigua della memoria, la scelta più conveniente è:
    B Allocare l'intero blocco al processo, sprecando 20 bytes(frammentazione interna)

72. Si supponga che un processo P necessiti di un'area di memoria libera pari a 99 KiB per essere allocato in modo **contiguo** in memoria principale. Se la lista dei blocchi di memoria libera contiene i seguenti elementi: A, B, C, D le cui dimensioni sono rispettivamente 102 KiB, 99 KiB, 256 KiB e 128 KiB, quale blocco verrà allocato per P assumendo una politica Worst-Fit?
    B. blocco C
73. Si supponga che un processo P necessiti di un'area di memoria libera pari a 99 KiB per essere allocato in modo **contiguo** in memoria principale. Se la lista dei blocchi di memoria libera contiene i seguenti elementi: A, B, C, D, E, F le cui dimensioni sono rispettivamente 300 KiB, 600 KiB, 350 KiB, 200 KiB, 750 KiB e 125 KiB, quale blocco verrà allocato per P assumendo una politica Worst-Fit?
    D. blocco E
74. Si supponga che un processo P necessiti di un'area di memoria libera pari a 128 KiB per essere allocato in modo **contiguo** in memoria principale. Se la lista dei blocchi di memoria libera contiene i seguenti elementi: A, B, C, D le cui dimensioni sono rispettivamente 105 KiB, 916 KiB, 129 KiB e 80 KiB, quale blocco verrà allocato per P assumendo una politica First-Fit?
    C. blocco B
75. Si supponga che un processo P necessiti di un'area di memoria libera pari a 115 KiB per essere allocato in modo **contiguo** in memoria principale. Se la lista dei blocchi di memoria libera contiene i seguenti elementi: A, B, C, D,E,F

le cui dimensioni sono rispettivamente 300 KiB, 600 KiB, 350 KiB, 200 KiB,750 KiB e 125 KiB quale blocco verrà allocato per P assumendo unapolitica First-Fit?
    A. blocco A

76. Si supponga che un processo P necessiti di un'area di memoria libera pari a 375 KiB per essere allocato in modo **contiguo** in memoria principale. Se la lista dei blocchi di memoria libera contiene i seguenti elementi: A, B, C, D,E,F le cui dimensioni sono rispettivamente 300 KiB, 600 KiB, 350 KiB, 200 KiB,750 KiB e 125 KiB quale blocco verrà allocato per P assumendo una politica Best-Fit?
    A. blocco B
77. Si supponga che un processo P necessiti di un'area di memoria libera pari a 34 KiB per essere allocato in modo **contiguo** in memoria principale. Se la lista dei blocchi di memoria libera contiene i seguenti elementi: A, B, C, D le cui dimensioni sono rispettivamente 36 KiB, 90 KiB, 42 KiB e 35 KiB, quale blocco verrà allocato per P assumendo una politica Best-Fit?

78. Si supponga di avere una memoria M di capacità pari a 4 KiB, ossia 4,096 bytes. Assumendo che l'indirizzamento avvenga con lunghezza di parola (word size) pari 2 bytes e che M utilizzi una gestione paginata con blocchi di dimensione pari a S = 128 bytes, quanti bit sono necessari per identificare l'indice di pagina (p) e l'offset (interno alla pagina), rispettivamente?
    D. p=5; offset=6

79. Si consideri una memoria M di capacità pari a 512 bytes con frame di dimensione pari a 16 bytes. Dato l'indirizzo del byte 197, quale sarà l'indirizzo di pagina (p) e l'offset (interno alla pagina):(capitato due volte)
    D p=12; offset=5
80. Si consideri una memoria M di capacità pari a 100 bytes con frame di dimensione pari a 10 bytes. Dato l'indirizzo del byte 37, quale sarà l'indirizzo di pagina (p) e l'offset (interno alla pagina).
    A. p=3; offset=7

81. Si consideri un processo di dimensione pari a 2,097 bytes e un blocco di memoria libero di dimensione pari a 2,104 bytes. In questo caso, assumendo il vincolo di allocazione contigua della memoria, la scelta più conveniente è:
    B Allocare l'intero blocco al processo, sprecando 7 bytes (frammentazione interna)

82. Si supponga di avere una memoria M di capacità pari a 2 KiB, ossia 2,048 bytes. Assumendo che l'indirizzamento avvenga con lunghezza di parola (word size) pari a 4 bytes, quanti bit sono necessari ad indirizzare le parole contenute in M?
    B. 9
83. Si supponga di avere una memoria M di capacità pari a 4 KiB ossia 4,096 bytes. Assumendo che l'indirizzamento avvenga con lunghezza di parola (word size) pari a 2 bytes, quanti bit sono necessari ad indirizzare le parole contenute in M?
    B. 11
84. Si supponga di avere una memoria M di capacità pari a 8 KiB, ossia 8,192 bytes. Assumendo che l'indirizzamento avvenga con lunghezza di parola (word size) pari al singolo byte e che M utilizzi una gestione paginata con blocchi di dimensione pari a S = 128 bytes, quale dimensione (intesa come numero di entry) ha la corrispondente page table T?
    C. 64

85. Si supponga di avere una memoria M di capacità pari a 8 KiB, ossia 8,192 bytes. Assumendo che l'indirizzamento avvenga con lunghezza di parola (word size) pari a 4 bytes e che M utilizzi una gestione paginata con blocchi di dimensione pari a S = 256 bytes, quale sarà il numero di entry della corrispondente page table T?
    A. 32
86. Si supponga di avere una memoria M di capacità pari a 16 KiB, ossia 16,384 bytes. Assumendo che l'indirizzamento avvenga con lunghezza di parola (word size) pari a 4 bytes e che M utilizzi una gestione paginata con blocchi di dimensione pari a S = 64 bytes, quale sarà il numero di entry della corrispondente page table T?
    B. 14
    C. 256

87. Si consideri un sistema operativo che utilizza indirizzi logici da 21 bit, indirizzo fisico da 16 bit e memoria paginata in cui ciascuna pagina ha dimensione 2 KiB(2048 bytes). Qual è la dimensione massima di memoria fisica supportata dal sistema?(uscita già 2 volte)
    A. 64 KiB

La memoria Virtuale

88. La memoria virtuale consente di:
    B Mantenere allocate in memoria fisica solo alcune pagine dello spazio diindirizzamento logico di un processo

89. Se un'istruzione idempotente genera un page fault:
    D L'istruzione verrà nuovamente eseguita al ritorno dalla gestione delpage fault
90. Il problema della frammentazione esterna:
    C E' una conseguenza del vincolo di allocazione contigua della memoria
91. Il problema della frammentazione esterna
    D E' dovuto all' allocazione/deallocazione di blocchi contigui di memoria
92. Il *working set* è:
    C Relativamente piccolo rispetto all'intero spazio di indirizzamento di unprocesso

93. Si consideri un sistema che implementa la politica LRU per la sostituzione dei frame mediante l'uso di un timestamp. Ad ogni richiesta di accesso ad un determinato frame occorre:
    B Aggiornare il valore del timestamp con quello corrente

94. Data una memoria composta da 3 frame fisici e un processo composto da 5 pagine virtuali: A, B, C, D, E, si calcoli il numero di page fault che si verificano a fronte delle seguenti richieste da parte del processo: B, C, C, B, A, E, B, A

caricata in memoria e che si utilizzi un algoritmo LRU di sostituzione delle pagine.

C. 6

95. Data una memoria composta da 3 frame fisici e un processo composto da 5 pagine virtuali: A, B, C, D, E, si calcoli il numero di page fault che si verificano a fronte delle seguenti richieste da parte del processo: D, B, A, C, C, E, A, D, B, E, D, A. Si assuma che **nessuna** pagina del processo sia inizialmente caricata in memoria e che si utilizzi un algoritmo LRU di sostituzione delle pagine.

C. 9

96. Data una memoria composta da 3 frame fisici e un processo composto da 5 pagine virtuali: A, B, C, D, E, si calcoli il numero di page fault che si verificano a fronte delle seguenti richieste da parte del processo: C,B,C,B,A,E,B,A. Si assuma che **nessuna** pagina del processo sia inizialmente caricata in memoria e che si utilizzi un algoritmo LRU di sostituzione delle pagine.

B. 4

97. Data una memoria fisica composta da 3 frame fisici e un processo composto da 5 pagine virtuali: A, B, C, D, E, si calcoli il numero di page fault che si verificano a fronte delle seguenti richieste da parte del processo: A, B, E, C, E, D, D, A, B. Si assuma che **nessuna** pagina del processo sia inizialmente caricata in memoria e che si utilizzi un algoritmo FIFO di sostituzione delle pagine.

B. 7

98. Data una memoria composta da 3 frame fisici e un processo composto da 5 pagine virtuali: A, B, C, D, E, si calcoli il numero di page fault che si verificano a fronte delle seguenti richieste da parte del processo: D, A, C, B, B, A, C, B, D, E, A. Si assuma che **nessuna** pagina del processo sia inizialmente caricata in memoria e che si utilizzi un algoritmo FIFO di sostituzione delle pagine.

B. 7

99. Data una memoria composta da 3 frame fisici e un processo composto da 5 pagine virtuali: A, B, C, D, E si calcoli il numero di page fault che si verificano a fronte delle seguenti richieste da parte del processo: E, B, E, C, D, E, A, B, C. Si assuma che **nessuna** pagina del processo sia inizialmente caricata in memoria e che si utilizzi un algoritmo FIFO di sostituzione delle pagine.

A. 7

La Memoria Secondaria

100. L'allocazione contigua di un file su disco:(uscita già 4 volte)
   D Tutte le risposte precedenti sono corrette
101. L'allocazione contigua di un file su un disco è la scelta preferibile quando il disco è:
   A. Un CD/DVD-ROM in sola lettura

102. In un disco magnetico, il seek time:(capitato già 2 volte)
   C È il tempo necessario al disco per posizionare le proprie testine su unospecifico cilindro

103. Un disco è composto da 15 cilindri, ciascuno di capacità pari a 500 MB. Qual è la capacità totale del disco?
   A. 7.5 GB

104. Si supponga che il tempo di accesso alla memoria fisica sia t_{MA} = 50 nsec. e che il tempo per la gestione di un page fault t_{FAULT} sia pari a 15 msec. Assumendo che la probabilità che si verifichi un page fault sia p = 0.0002, qual è il tempo complessivo atteso di accesso alla memoria?
   C ~3.05 microsec
105. Si supponga che il tempo di accesso alla memoria fisica sia t_{MA} = 25 nsec. e che il tempo per la gestione di un page fault t_{FAULT} sia pari a 30 msec. Assumendo che la probabilità che si verifichi un page fault sia p = 0.005, qual è il tempo complessivo atteso di accesso alla memoria?
   A. ~150.025 microsec
106. Si supponga che il tempo di accesso alla memoria fisica sia t_{MA} = 50 nsec. e che il tempo per la gestione di un page fault t_{FAULT} sia pari a 25 msec. Assumendo che il tempo medio di accesso alla memoria sia pari a 0.5 microsec, qual è la probabilità $p$che si verifichi un page fault?
   C. ~0.002%

107. Si supponga che il tempo di accesso alla memoria fisica sia t_{MA} = 60 nsec. e che il tempo per la gestione di un page fault t_{FAULT} sia pari a 5 msec. Quale dovrà essere il valore della probabilità che si verifichi un fault (p) se si vuole garantire che il tempo atteso di accesso alla memoria sia al più il 20% più lento di t_{MA}? (Si ricordi che 1 msec = 10^3 microsec = 10^6 nsec)
   B. ~0,00024%

108. Si consideri un disco magnetico composto da 128 cilindri/tracce, numerati da 0 a 127 (0 indice del cilindro/traccia più esterno/a rispetto al centro del disco), la cui testina si trova inizialmente sul cilindro 42. Si calcoli il numero di cilindri/tracce attraversate dalla testina del disco, assumendo che la sequenza

di richieste: 74, 50, 32, 55, 81 venga gestita da un algoritmo di scheduling SSTF (Shortest Seek Time First) e trascurando il tempo di rotazione.

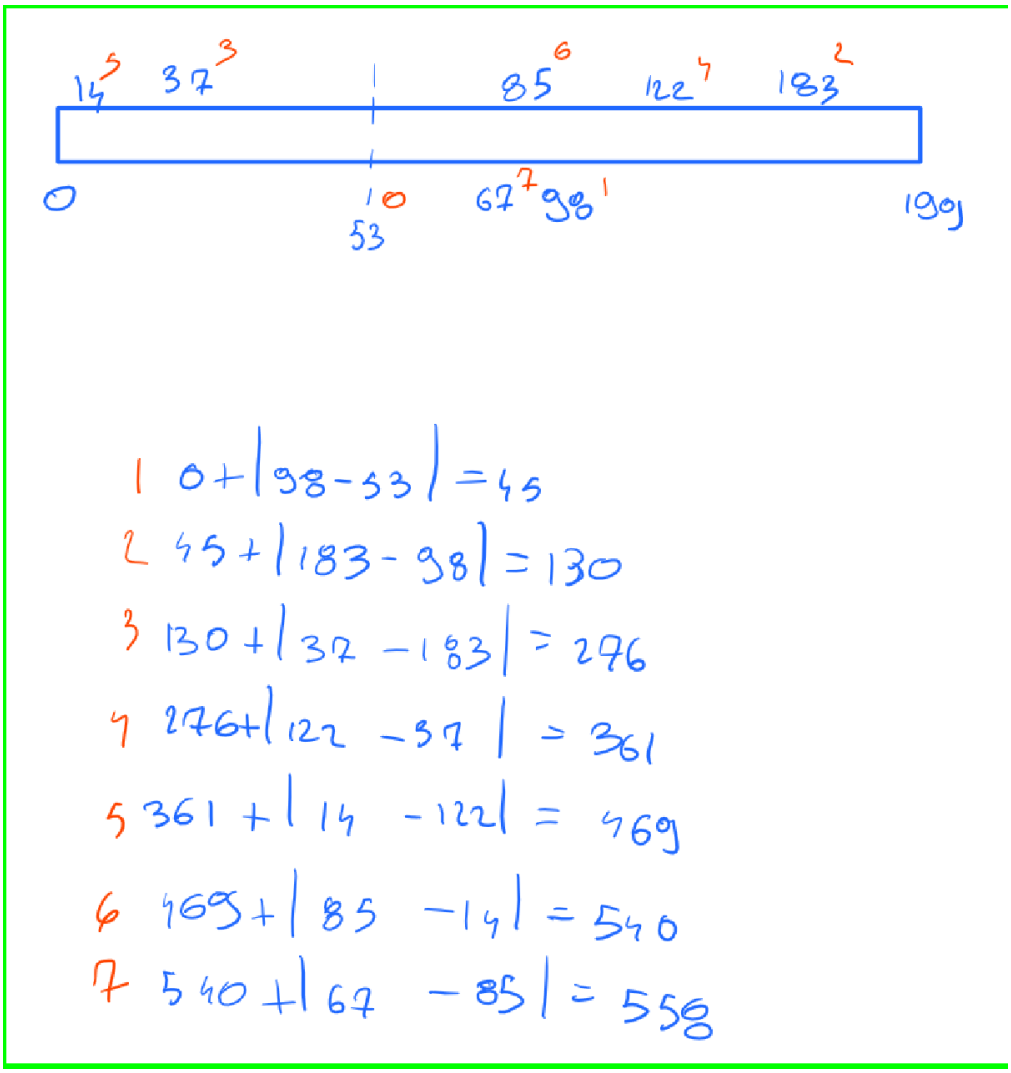
D. 88

109. Si consideri un disco magnetico composto da 128 cilindri/tracce, numerati da 0 a 127 (0 indice del cilindro/traccia più esterno/a rispetto al centro del disco), la cui testina si trova inizialmente sul cilindro 87. Si calcoli il numero di cilindri/tracce attraversate dalla testina del disco, assumendo che la sequenza di richieste: 43, 81, 36, 25, 127 venga gestita da un algoritmo di scheduling FCFS (First Come First Served) e trascurando il tempo di rotazione.

B. 240

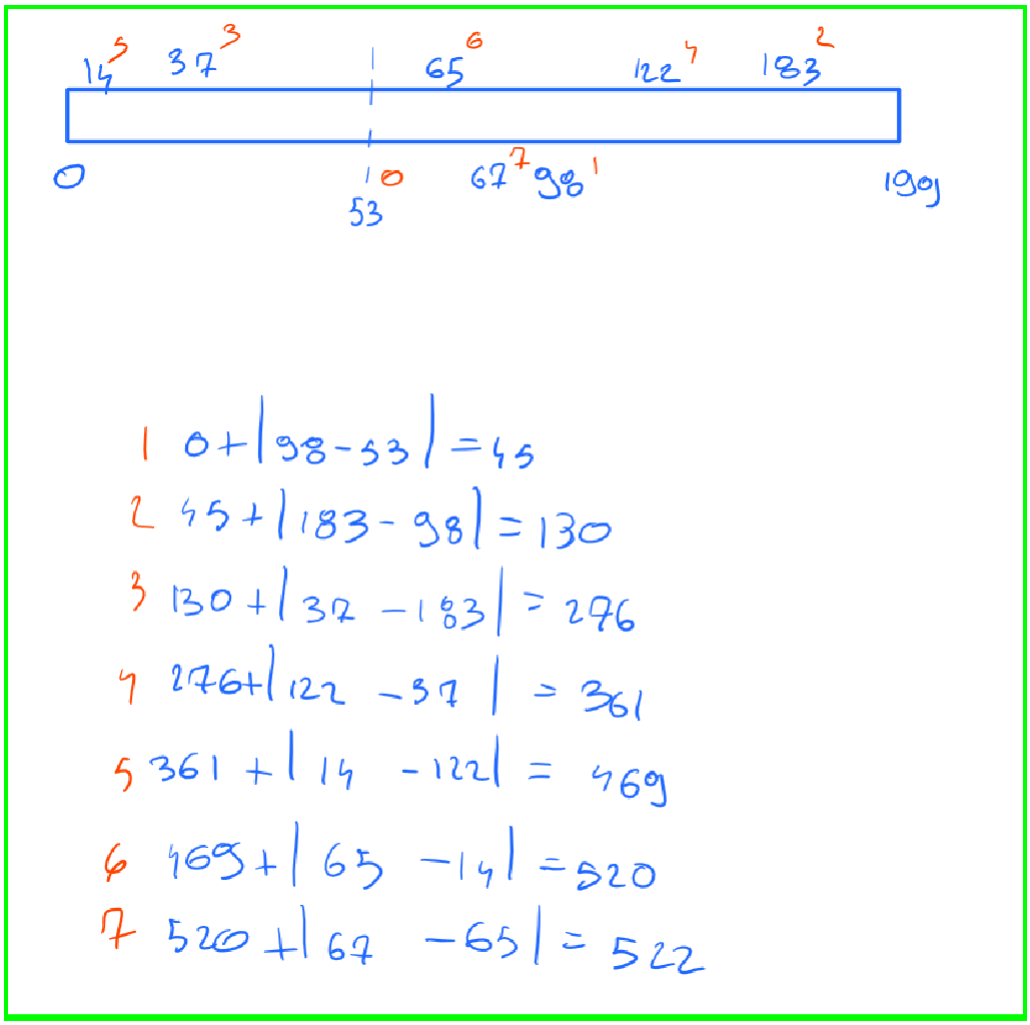
110. Si consideri un disco magnetico composto da 200 cilindri/tracce, numerati da 0 a 199(0 indice del cilindro/traccia più esterno/a rispetto al centro del disco), la cui testina si trova inizialmente sul cilindro 53. Si calcoli il numero di cilindri/tracce attraversate dalla testina del disco, assumendo che la sequenza di richieste: 98,183,37,122,14,85,67 venga gestita da un algoritmo di scheduling FCFS (First Come First Served) e trascurando il tempo di rotazione.

B. 558



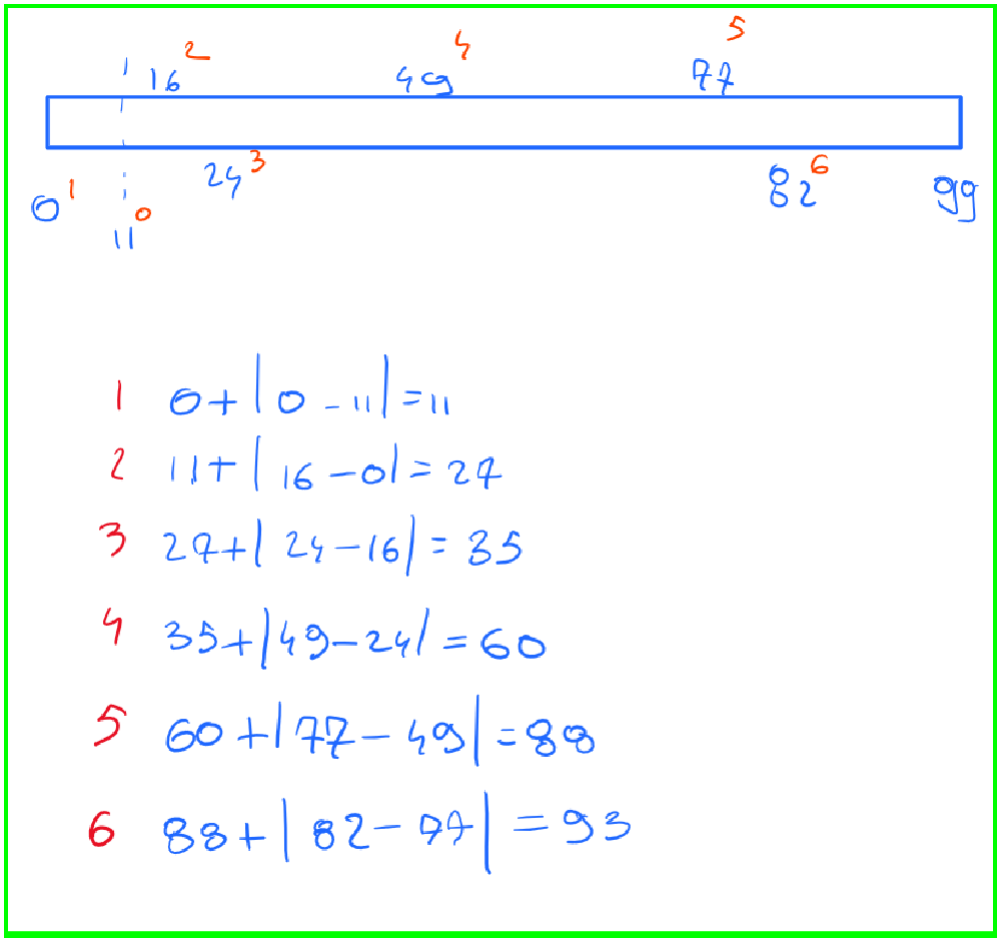
111. Si consideri un disco magnetico composto da 200 cilindri/tracce, numerati da 0 a 199(0 indice del cilindro/traccia più esterno/a rispetto al centro del disco), la cui testina si trova inizialmente sul cilindro 53. Si calcoli il numero di cilindri/tracce attraversate dalla testina del disco, assumendo che la sequenza di richieste: 98,183,37,122,14,65,67 venga gestita da un algoritmo di scheduling FCFS (First Come First Served) e trascurando il tempo di rotazione.

B. 522



112. Si consideri un disco magnetico composto da 100 cilindri/tracce, numerati da 0 a 99 (0 indice del cilindro/traccia più esterno/a rispetto al centro del disco), la cui testina si trova inizialmente sul cilindro 11. Si calcoli il numero di cilindri/tracce attraversate dalla testina del disco, assumendo che la sequenza di richieste: 24, 16, 77, 49, 82 venga gestita da un algoritmo di scheduling SCAN (**non**-ottimizzato), che la testina si stia muovendo verso l'esterno (i.e., verso i cilindri con numeri più bassi) e trascurando il tempo di rotazione.

D 93



113. Si consideri un disco magnetico composto da 100 cilindri/tracce, numerati da 0 a 99(0 indice del cilindro/tracci più esterno/a rispetto al centro del disco), la cui testina si trova inizialmente sul cilindro 46. Si calcoli il numero di cilindri/tracce attraversate dalla testina del disco,assumendo che la sequenza di richieste: 4, 96,51,29,18 venga gestita da un algoritmo di scheduling C-SCAN (**non**-ottimizzato), che la testina si stia muovendo verso l'esterno (i.e., verso i cilindri con numeri più bassi) e trascurando il tempo di rotazione.(uscito già 2 volte)

B. 193

114. Il tempo di trasferimento totale per un'operazione di I/O da disco magnetico è pari a 30 msec. Sapendo che: il seek time complessivo è pari a 18 msec, il rotational delay complessivo è pari a 7 msec e che il transfer rate è pari a 1.5 Gbit/sec, qual è la quantità totale di dati trasferita? (Si ricordi che 1 B = 1 byte = 8 bit e 1 MB = 10^3 KB = 10^6 B)(uscita già 3 volte)

C. 937.5 KB

115. Il tempo di trasferimento totale per un'operazione di I/O da disco magnetico è pari a 40 msec. Sapendo che: il seek time complessivo è pari a 18 msec, il rotational delay complessivo è pari a 7 msec e che il transfer rate è pari a 5 Gbit/sec, qual è la quantità totale di dati trasferita? (Si ricordi che 1

A. 9,375 MB

116. Il tempo di trasferimento totale per un'operazione di I/O da disco magnetico è pari a 36 msec. Sapendo che il seek time complessivo è pari a 13 msec e che sono stati trasferiti 2MB ad una velocità pari a 1 Gbit/sec qual è il rotational delay del disco?(Si ricordi che 1 B = 1 byte = 8 bit)
   A. 7 msec

Gestione File System

117. La tabella globale dei file aperti (global open file table):
   D. Tutte le risposte precedenti sono corrette
118. La tabella locale dei file aperti (local open file table):
   C Contiene un puntatore alla tabella globale dei file aperti per ciascun fileriferito da un processo

119. Un possibile esempio di applicazione che necessita accesso sequenziale ad un file è:
   A. Un compilatore
120. L'allocazione di un file indicizzata è preferibile quando il file in questione:
   D. E' di grandi dimensioni ed è tipicamente acceduto in modo casuale(diretto)

121. L'allocazione di un file basata su linked list(liste puntate) è preferibile quando il file in questione
   D E' di grandi dimensioni ed è tipicamente acceduto in modo sequenziale

122. In un sistema UNIX-like, un file che ha i seguenti privilegi: 101000000:
   B Consente al solo proprietario del file di esercitare diritti di lettura edesecuzione (sul file)
123. In un sistema UNIX-like, un file che ha i seguenti privilegi: 011000000:
   D Consente al solo proprietario del file di esercitare diritti di scrittura edesecuzione (sul file)
124. In un sistema UNIX-like, un file che ha i seguenti privilegi: 111101101:
   C Consente al proprietario del file di esercitare tutti i diritti(sul file) efornisce agli altri utenti solo diritti di lettura ed esecuzione

125. Il comando UNIX ln file_1 file_2
   B Crea un hard link con il file file_1(sorgente) il cui nome èfile_2(destinazione)
126. Il comando UNIX ln -s file_1 file2:
   C Crea un soft link con il file file_1(sorgente) il cui nome èfile_2(destinazione)

127. Si consideri un file system organizzato con file descriptor indicizzati multi-livello (multi-level indexed files), contenente i riferimenti diretti a 10

ulteriore doppio livello di riferimento indiretto, sempre da 100 blocchi ciascuno. Assumendo che ciascun blocco abbia dimensione pari a 2 KiB, qual è la dimensione massima del file supportata?

D. ~20.7 MB