Esame Sistemi Operativi - Operating Systems Exam
2025/01/28

Ex 01 (2.0 points)

Italiano

Si consideri l'algoritmo di schedulazione Round-Robin (RR). Si indichino quali delle seguenti affermazioni sono vere. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Consider the Round-Robin (RR) scheduling algorithm. Indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

Scegli una o più alternative. Choose one or more options.

1. ✅ Più il quantum temporale aumenta e più il comportamento dell'algoritmo RR diventa simile all'algoritmo First-Come First-Served (FCFS). The more the time quantum increases, the more the behavior of the RR algorithm becomes similar to the First-Come First-Served (FCFS) algorithm.
2. ⬜ L'algoritmo RR è senza prelazione. The RR algorithm is without preemption.
3. ⬜ Si può dimostrare che l'algoritmo RR è un algoritmo ottimo utilizzando il tempo di attesa come criterio di valutazione. It can be shown that the RR algorithm is an optimal algorithm by using the waiting time as an evaluation criterion.
4. ✅ L'incremento del quantum temporale porta l'algoritmo RR a essere gestito con maggiore efficienza dallo scheduler. The increase in the time quantum leads the RR algorithm to be managed more efficiently by the scheduler.
5. ⬜ Nell'algoritmo RR, all'aumentare del quantum temporale, la frequenza delle operazioni di context switching aumenta. In the RR algorithm, as the time quantum increases, the frequency of context switching operations increases.
6. ✅ All'aumentare del quantum temporale, la produttività (throughput) dell'algoritmo RR, aumenta. As the time quantum increases, the throughput of the RR algorithm increases.
7. ⬜ Per sistemi senza vincoli real-time e con processi che fanno molto uso di CPU (CPU intensive) è meglio avere quanti corti. For systems without real-time constraints and with CPU-intensive processes, it is better to have short quantum.

Ex 02 (3.5 points)

Italiano

Si supponga di eseguire i tre thread successivi (T1, T2 T3) in parallelo. Si indichino quali sequenze possono venire visualizzate oppure quali affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Suppose to run the following three threads (T1, T2, and T3) in parallel. Indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

```
init(a,0);
init(b,1);
```

T1	**T2**	**T3**

`wait(a);`
`printf("1");`
`signal(a);`
`signal(b);`
	`wait(b);`
	`printf("2");`
	`signal(a);`
	`wait(a);`
	`printf("3");`
	`wait(b);`
	`printf("4");`
	`signal(a);`
		`wait(a);`
		`printf("5");`
		`signal(a);`
		`signal(b);`

Fasi operative. Workflow.

`2 1 3 4 5`
`2 1 5 3 4`
`2 5 3 4 1`
`2 5 1 3 4`
`2 3` deadlock

Scegli una o più alternative. Choose one or more options.

1. ✅ `2 1 3 4 5`
2. ⬜ `2 5 3 1 4`
3. ⬜ `1 5 2 3 4`
4. ✅ `2 5 3 4 1`
5. ✅ `2 1 5 3 4`
6. ✅ Il programma è soggetto a deadlock. The program is subject to deadlock.
7. ⬜ `2 1 3 5 4`
8. ✅ `2 5 1 3 4`

Ex 03 (2.5 points)

Italiano

Sia data la seguente soluzione al problema dei 5 filosofi, nella quale la variabile i ∈ {0, 1, 2, 3, 4} rappresenta l'indice assegnato a ognuno dei filosofi. Indicare quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Given the following solution to the 5 philosophers problem, the variable i ∈ {0, 1, 2, 3, 4} represents the index assigned to each of the philosophers. Please indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

```
sem_t fork[5] = {1,1,1,1,1};

void phy(int i) {
      think();
      wait(fork[i]); wait(fork[(i+1)%5]);
      eat();
      signal(fork[i]); signal(fork[(i+1)%5]); }
```

Scegli una o più alternative. Choose one or more options.

1. ✅ La soluzione potrebbe essere soggetta a deadlock. The solution may be subject to deadlocks.
2. ⬜ Invertendo l'ordine di `wait(fork[i]);` e `wait(fork[(i+1)%5]);` per solo uno dei filosofi, la soluzione potrebbe essere soggetta a deadlock. By reversing the order of wait(fork[i]); and wait(fork[(i+1)%5]); for just one of the philosophers, the solution could be subject to deadlocks.
3. ✅ Invertendo l'ordine di `wait(fork[i]);` e `wait(fork[(i+1)%5]);` per tutti i filosofi, la soluzione potrebbe essere soggetta a deadlock. By reversing the order of wait(fork[i]); and wait(fork[(i+1)%5]); for all philosophers, the solution could be subject to deadlocks.
4. ⬜ Invertendo l'ordine di `wait(fork[i]);` e `wait(fork[(i+1)%5]);` per tutti i filosofi, sicuramente il sistema non è in una condizione di deadlock. By reversing the order of wait(fork[i]); and wait(fork[(i+1)%5]); for all philosophers, surely the system is not in a deadlock condition.
5. ⬜ Invertendo l'ordine di `wait(fork[i]);` e `wait(fork[(i+1)%5]);` per tutti i filosofi, sicuramente il sistema è in una condizione di deadlock. By reversing the order of wait(fork[i]); and wait(fork[(i+1)%5]); for all philosophers, surely the system is in a deadlock condition.
6. ✅ Il deadlock non può verificarsi se viene inizializzato a 2 almeno un elemento del vettore `fork`. The deadlock cannot occur if at least one element of the fork vector is initialized to 2.
7. ⬜ Il deadlock non può verificarsi se si sostituisce l'ultima linea di codice con `signal(fork[i]); signal(fork[i]); signal(fork[(i+1)%5]);`. The deadlock cannot occur if you replace the last line of code with signal(fork[i]); signal(fork[i]); signal(fork[(i+1)%5]);.

Ex 04 (3.0 points)

Italiano

Un sistema ha 4 processi (P1, P2, P3, P4) e 4 tipi di risorse (R1, R2, R3, R4), in cui il numero di istanze per ogni tipologia di risorsa è pari a 1. Lo stato del sistema è il seguente:

- Il processo P1 detiene l'istanza della risorsa di tipo {R1}, e attende l'istanza della risorsa di tipo {R2}.
- Il processo P2 detiene l'istanza della risorsa di tipo {R2}, e attende l'istanza della risorsa di tipo {R4}.
- Il processo P3 detiene l'istanza della risorsa di tipo {R4}, e attende l'istanza della risorsa di tipo {R3}.
- Il processo P4 detiene l'istanza della risorsa di tipo {R3}, e attende l'istanza della risorsa di tipo {R1}.

Si effettui un'analisi della situazione dal punto di vista del deadlock. Si indichi quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

A system has 4 processes (P1, P2, P3, P4) and 4 types of resources (R1, R2, R3, R4), in which the number of instances of each resource type is 1. The state of the system is the following:

- Process P1 holds the instance of resource type {R1} and waits for the instance of resource type {R2}.
- Process P2 holds the instance of resource type {R2} and waits for the instance of resource type {R4}.
- Process P3 holds the instance of resource type {R4} and waits for the instance of resource type {R3}.
- Process P4 holds the instance of resource type {R3} and waits for the instance of resource type {R1}.

Analyze the situation from the point of view of the deadlock. Please indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

Scegli una o più alternative. Choose one or more options.

1. ✅ Analizzando esclusivamente la presenza di cicli nel grafo di allocazione delle risorse, il sistema è SICURAMENTE in una condizione di deadlock. Exclusively analyzing the presence of loops in the resource allocation graph, the system is DEFINITELY in a deadlock condition.
2. ⬜ Il sistema potrebbe NON essere in una condizione di deadlock in quanto le risorse sono unitarie. The system may NOT be in a deadlock condition as the resources are unitary.
3. ✅ In questo caso specifico, se il sistema è in deadlock, basta eliminare qualsiasi arco per uscire dalla condizione di deadlock. In this specific case, if the system is in deadlock, the deletion of any arc lead the system to a condition without deadlock.
4. ✅ L'attesa circolare è una condizione necessaria perché ci sia deadlock. Circular waiting is a necessary condition for deadlock.
5. ⬜ Hold-and-wait è una condizione sufficiente perché ci sia deadlock. Hold-and-wait is a sufficient condition for deadlock.
6. ⬜ L'eliminazione dell'arco R1->P1 può essere gestita più facilmente dal processo P1 rispetto all'eliminazione dell'arco P1->R2. The deletion of arc R1->P1 can be handled more easily by the P1 process than the deletion of arc P1->R2.

Ex 05 (2.5 points)

Italiano

Dato il seguente file `f.txt`, si indichino quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Given the following `f.txt` file, indicate which statements are correct. Note that incorrect answers imply a penalty in the final score.

```
axxxy 20 Giuseppe Rosso
axxxxy 5 Anna Verde
axy 2 Giorgio Giallo
ay 10 Stefano Verde
az 15 Giulio Rosa
```

Scegli una o più alternative. Choose one or more options.

1. ⬜ Il commando `egrep -e "ax(xx)+y" f.txt` seleziona due righe del file f.txt. The command 'egrep -e "ax(xx)+y" f.txt' selects two lines of the f.txt file.

2. ✅ Il comando `egrep -e "ax*y" f.txt` seleziona quattro righe del file f.txt. The command 'egrep -e "ax*y" f.txt' selects four lines of the f.txt file.
3. ✅ Il comando `sort -k4,4r -k3,3 f.txt | head -n 1` restituisce il seguente output "`axxxxy 5 Anna Verde`". The command' sort -k4,4r -k3,3 f.txt | head -n 1' returns the following output "axxxxy 5 Anna Verde".
4. ⚪ Il comando `cat f.txt | tr -s "x" | sort -k2,2nr | head -n 1` restituisce il seguente output "`axxxy 20 Giuseppe Rosso`". The command 'cat f.txt | tr -s "x" | sort -k2,2nr | head -n 1' returns the following output "axxxy 20 John Red".
5. ✅ Il comando `cut -d " " -f 2,4 f.txt | head -n 2 | tail -n 1` restituisce il seguente output "`5 Verde`". The command 'cut -d " " -f 2.4 f.txt | head -n 2 | tail -n 1' returns the following output "5 Green".
6. ⚪ Il comando `cut -d " " -f 1-3 f.txt | tail -n 1` restituisce il seguente output "`axxxy 20 Giuseppe`". The command 'cut -d " " -f 1-3 f.txt | tail -n 1' returns the following output "axxxy 20 Joseph".

Ex 06 (2.0 points)

Italiano

Si faccia riferimento alla gestione dei segnali in un sistema UNIX-like. Si indichino quali delle seguenti affermazioni sono vere. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Consider signal handling in a UNIX-like system. Indicate which of the following statements related to signals are correct. Note that incorrect answers imply a penalty in the final score.

Scegli una o più alternative. Choose one or more options.

1. ✅ La ricezione contemporanea di due (o più) volte lo stesso segnale da parte di un processo, potrebbe comportare la non gestione di alcuni segnali. The simultaneous reception of two (or more) times of the same signal by a process could result in the non-handling of some signals.
2. ⚪ La ricezione di un segnale prima di una pause() rende la pause() non bloccante. Receiving a signal before a pause() makes the pause() non-blocking.
3. ✅ Il comportamento di alcuni segnali può essere ridefinito. The behavior of some signals can be redefined.
4. ⚪ All'interno del signal handler si possono (è utile) usare funzioni non rientranti. Within the signal handler you can (and it is useful) use non-reentrant functions.
5. ✅ Un processo può mandare un segnale sia a se stesso che ad altri processi. A process can send a signal to both itself and other processes.
6. ⚪ Il comportamento di default di un processo alla ricezione di un segnale è la terminazione. The default behavior of a process when receiving a signal is termination.
7. ✅ Alcuni segnali possono essere ignorati e tutti i segnali hanno un comportamento di default. Some signals can be ignored, and all signals have a default behavior.

Ex 07 (2.5 points)

Italiano

Si supponga che la memoria principale di un sistema embedded sia costituita da 24 blocchi di 2 MByte ciascuno, che tali blocchi siano numerati da 0 a 23, che il sistema operativo tenga traccia dei blocchi liberi (occupati) indicandoli in un vettore con il valore 0 (1), e che la situazione attuale della memoria sia rappresentata dal seguente vettore:

0 0 0 0 1 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 1 0

Si indichino quali delle seguenti affermazioni sono vere. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Suppose that the main memory of an embedded system is composed of 24 blocks of 2 MBytes each, numbered from 0 to 23. Suppose that the operating system keeps track of the free (occupied) blocks indicating them in a vector with the value 0 (1), and that the current situation of the memory is represented by the following vector:

0 0 0 0 1 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 1 0

Indicate which of the following statements are correct. Note that wrong answers imply a penalty in the final score.

Scegli una o più alternative. Choose one or more options.

1. ✅ Un file di dimensione 9.5 MByte PUO' essere allocato utilizzando una strategia di allocazione contigua. A file of dimension 9.5 MBytes CAN be allocated using a contiguous allocation strategy.
2. ⬜ Con la strategia di allocazione indexata NON possono essere allocati file più grossi di 10 MByte. With the indexed allocation strategy, CANNOT be allocated files with dimension greater than 10 MBytes.
3. ✅ Con la strategia di allocazione contigua NON possono essere allocati file più grossi di 10 MByte. With the contiguous allocation strategy CANNOT be allocated files with dimension greater than 10 MBytes.
4. ⬜ La strategia di allocazione contigua soffre di frammentazione interna ma non soffre di frammentazione esterna. The contiguous allocation strategy suffers from internal fragmentation, but it does not suffer from external fragmentation.
5. ✅ Nella strategia di allocazione contigua il massimo spazio sprecato per frammentazione interna, per ogni singolo file, è inferiore a 2 MByte. In the contiguous allocation strategy, the maximum space wasted by internal fragmentation, on a per-file basis, is less than 2 MBytes.
6. ✅ Memorizzando 4 file con una allocazione indicizzata, per i dati sono disponibili 24 MBytes. By storing 4 files with an indexed allocation strategy, 24 MBytes are available for data.
7. ✅ Con la strategia di allocazione contigua ottenuta mediante l'algoritmo BEST-FIT possono essere allocati in ordine di richiesta i file F1 di 8.5 MByte, F2 di 5.8 MByte e F3 di 7.6 MByte. With the contiguous allocation strategy based on the BEST-FIT algorithm, it is possible to allocate, in order of request, files F1 of 8.5 MBytes, F2 of 5.8 MBytes, and file F3 of 7.6 MBytes.

Ex 08 (3.0 points)

Italiano

Si consideri il seguente insieme di processi schedulati con un quantum temporale di 10 unità di tempo. Rappresentare, mediante diagramma di Gantt, l'esecuzione di tali processi utilizzando l'algoritmo Shortest-Remaining-Time First (SRTF). Si calcolino i tempi di terminazione di ciascun processo e il tempo di completamento (turnaround time) medio. Si riporti la risposta su un'unica riga, indicando i tempi di terminazione di P1, P2, P3 e P4 seguiti dal tempo di completamento medio. Separare numeri e stringhe con un unico spazio. Riportare il tempo di completamento medio con 1 sola cifra decimale. Non inserire nessun altro carattere nella risposta. Esempio di risposta corretta: 20 11 45 67 30.5

English

Consider the following processes, scheduled with a temporal quantum of 10 units. Represent, using a Gantt diagram, the execution of these processes using the Shortest-Remaining-Time First (SRTF) scheduling algorithms. Compute the termination time of each process and the average turnaround time. Report your answer on a single line, indicating P1, P2, P3, and P4 termination times, followed by the average turnaround time. Separate the numbers with a single space. Report the average turnaround time with a single decimal digit. Do not enter any other character in the response. Example of correct answer: 20 11 45 67 30.5

Processo Process	TempoArrivo ArrivalTime	BurstTime	Priorità Priority
P1	0	27	1
P2	0	13	4
P3	5	3	3
P4	18	15	2

Fasi operative. Workflow.

```
SRTF:
  P2: from=  0 to  =  4 end =  5 waitingTime =  0
  P3: from=  5 to  =  7 end =  8 waitingTime =  0
  P2: from=  8 to  = 15 end = 16 waitingTime =  3
  P1: from= 16 to  = 17 end = 18 waitingTime = 16
```

```
P4: from= 18 to  = 32 end = 33 waitingTime =  0
P1: from= 33 to  = 57 end = 58 waitingTime = 15
EndTime: 58 16  8 33
AverageWaitingTime: [( 16 + 15 ) + (  3 ) + (  ) + (  ) ] / 4 = 8.50
AverageTournaroudTime: ( 58 + 16 +  3 + 15 ) / 4 = 23.00
GanttChart: 22222333222222221144444444444444411111111111111111111111
```

Risposta. Answer.

```
58 16 8 33 23.0
```

Ex 09 (3.5 points)

Italiano

Si supponga di eseguire il seguente programma. Si indichino quanti caratteri 'A' e quanti caratteri 'B' sono stampati su standard output. Riportare due valori interi che indichino il numero di 'A' e il numero di 'B'. Esempio di risposta corretta nel caso di 1 'A' e 2 'B': 1 2

English

Suppose to run the following program. Indicate how many characters 'A' and 'B' are printed on standard output. Indicate two values, the number of 'A's and the one of 'B's. Example of correct answer in case of 1 'A' and 2 'B': 1 2

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

int main(){
  int i, a;
  setbuf(stdout,0);

  a = fork();
  if (fork() == 0) {
    if (a) {
      fork();
      execlp("echo", "B", "A", NULL);
    } else {
      system("echo A");
    }
  } else {
    for (i=0; i<3; i++) {
      if (!fork()) {
        printf("B");
      }
    }
  }
  return 0;
}
```

Risposta. Answer.

3 14

Ex 10 (3.0 points)

Italiano

Si realizzi un programma mediante il linguaggio di scripting BASH che ricerchi nel direttorio `/home/scanzio` e nel suo sottoalbero i file regolari con estensione `.txt` e che appartengono all'utente `scanzio`. Il programma dovrà stampare il nome dei soli file che contengono la stringa "`main`".

English

Create a program using the BASH scripting language that searches in the /home/scanzio directory and in its subtrees for regular files with the extension .txt and that belong to the user ‘scanzio‘. The program must only print the files' names containing the string "main".

Soluzione. Solution.

```
#!/bin/bash

#single command
find /home/scanzio -user "scanzio" -type f -name "*.txt" -exec grep -l "main" {} \;
| rev | cut -d "/" -f1 | rev

#loop
for f in $(find /home/scanzio -user "scanzio" -type f -name "*.txt"); do
      r=$(grep "main" ${f})
      if [[ ! -z "${r}" ]]; then
            echo "$(basename ${f})"
      fi
done
```

Ex 11 (3.0 points)

Italiano

Si realizzi un programma mediante il linguaggio di scripting BASH che legga un file riga per riga e, per ogni riga, stampi il numero di parole che la compongono. Il nome del file è passato come primo parametro della linea di comando. Nel caso in cui il numero di parametri passati allo script sia diverso da uno, il programma dovrà scrivere un messaggio di errore e terminare l’esecuzione.

English

Create a program using the BASH scripting language that reads a file line by line and, for each line, prints the number of words that compose it. The file name is passed as the first parameter of the command line. In case the number of parameters passed to the script is different from one, the program will have to write an error message and terminate the execution.

Soluzione. Solution.

```
#!/bin/bash

if [ $# -ne 1 ]; then
        echo "Usage: $0 <filename>"
        exit 1
fi

# Alternative: word count
while read -r line; do
        echo "${line}" | wc -w
done < $1

# Alternative: array
while read -a words; do
        echo ${#words[*]}
done < $1

# Alternative: loop
while read -r line; do
        count=0
        for word in $line; do
                ((count++))
        done
```

```
        echo $count
done < $1
```

Ex 12 (3.5 points)

Italiano

In riferimento alla seguente soluzione software per la gestione della mutua esclusione, si indichino quali affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Regarding the following software solution for managing mutual exclusion, indicate which statements are correct. Note that incorrect answers imply a penalty in the final score.

```
int flag[2]={0,0};
```

T0	**T1**

`1: while(1) {`
`2:  while(flag[1]);`
`3:  flag[0]=1;`
`4:  // Critical section`
`5:  flag[0]=0;`
`6: }`
	`7: while(1) {`
	`8:  while(flag[0]);`
	`9:  flag[1]=1;`
	`10:  // Critical section`
	`11:  flag[1]=0;`
	`12: }`

Scegli una o più alternative. Choose one or more options.

1. ⬜ Non è corretta perché soffre di livelock. It is not correct because it suffers from livelock.
2. ✅ La mutua esclusione potrebbe non essere garantita. Mutual exclusion may not be guaranteed.
3. ✅ Nel caso in cui sia la sequenza le righe 2 e 3, e le righe 8 e 9, siano eseguite in modo atomico, la soluzione sarebbe corretta dal punto di vista della mutua esclusione. In the case where both the sequence lines 2 and 3, and lines 8 and 9, are executed atomically, the solution would be correct from the point of view of mutual exclusion.
4. ⬜ Un processo in attesa di entrare nella sezione critica non consuma CPU. A process waiting to enter the critical section does not consume CPU.
5. ⬜ Eliminando le linee di codice 5 e 11 dopo un po' il sistema andrà in livelock. Deleting lines 5 and 11 will lead the system to a livelock after a while.
6. ✅ Il codice all'interno della sezione critica deve essere eseguito da un singolo thread (o processo) per volta. The code in the critical section must be executed by a single thread (or process) at a time.

Ex 13 (2.0 points)

Italiano

Si considerino le system call della famiglia **exec.** Indicare quali delle seguenti affermazioni sono vere. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Considering the system calls of the family **exec**. Indicate which statements are correct. Note that incorrect answers imply a penalty in the final score.

Scegli una o più alternative. Choose one or more options.

1. ✅ Exec sostituisce il codice del processo con il codice eseguibile di un altro programma. Exec substitutes the process code with the executable code of another program.
2. ⬜ E' inutile verificare il valore ritornato da exec in quanto il codice dopo l'exec non viene mai eseguito. It is useless to check the value returned by exec because the code after the exec is never executed.
3. ✅ Nel caso non si conoscano in fase di compilazione il numero di argomenti del comando passato all'exec, è indispensabile usare le funzioni con estensione v, e cioè execv(), execvp() o execve(). If you do not know the number of arguments of the command passed to the exec at compilation time, it is essential to use the functions with the extension v, i.e. execv(), execvp() or execve().

4. ⬜ La seguente execlp(“ls”, “-l”, NULL); esegue il comando ls -l. The following execlp("ls", "-l", NULL); executes the command ls -l.
5. ✅ La seguente exec è corretta: execlp(“cp”, “-f”, “./f1.txt”, “./f2.txt”, NULL);. The following exec is correct: execlp("cp", "-f", "./f1.txt", "./f2.txt", NULL);.
6. ⬜ La seguente exec è corretta: execl(“cp”, “-f”, “./f1.txt”, “./f2.txt”, NULL);. The following exec is correct: execl("cp", "-f", "./f1.txt", "./f2.txt", NULL);.
7. ⬜ Le exec() sono implementate al loro interno mediante la system call system(). The exec() are implemented internally, employing the system() system call.