Esame Sistemi Operativi - Operating Systems Exam
2022/02/10

Ex 1 (2.5 points)

Italiano

Dato il seguente comando di shell, si indichino quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Given the following shell command, indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

```
find . -type f -regextype posix-extended -regex
".*/((XXXXX(XX)*)|((BB|CCC){2,4}))([8-9][0-9]|1[0-4][0-9]|15[0-7])?\.txt" -exec cat
\{} \; | sort -k1,1nr| uniq
```

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

1. ✅ Il file regolare ./BBCCCBB.txt è selezionato dal comando find. The regular file ./BBCCCBB.txt is selected by the find command.
2. ⬜ L'espressione regolare codificata nel comando find può riconoscere solo file presenti nella directory di lavoro corrente. The regular expression coded in the find command can match only files located in the current working directory.
3. ✅ Il file regolare ./BBCCCBBBB148.txt è selezionato dal comando find. The regular file ./BBCCCBBBB148.txt is selected by the find command.
4. ⬜ Il file regolare ./XXXXXBBBB85.txt è selezionato dal comando find. The regular file ./XXXXXBBBB85.txt is selected by the find command.
5. ⬜ Per OGNI file selezionato il comando find esegue un ordinamento numerico delle sue righe dal numero più grande al più piccolo e se ci sono righe uguali elimina i duplicati. For EACH selected file the find command performs a numerical sorting of its lines from the biggest to the smallest number, and if there are equal lines it deletes duplicates.

Ex 2 (2.0 points)

Italiano

Dato il seguente elenco di comandi, si indichino quali delle seguenti affermazioni sono corrette dopo averli eseguiti. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Given the following list of commands, indicate which of the following statements are correct after executing them. Note that incorrect answers imply a penalty in the final score.

```
touch f1
ln f1 f2
ln f1 f3
ln -s f1 f4
rm f1
echo “hello” > f4
echo “_world” >>  f2
```

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

1. ⬜ Il file f1 è stato rimosso e quindi non è presente nella directory. File f1 was removed, and therefore it is not present in the directory.
2. ✅ I file f2 e f3 hanno lo stesso contenuto. Files f2 and f3 have the same content.
3. ✅ La dimensione del file f4 è di 2 bytes. The dimension of file f4 is 2 bytes.
4. ✅ Il numero di hard link del file f2 è pari a 2. The number of hard links of file f2 is 2.

5. ⬜ Il file f2 contiene la stringa “hello_world”. File f2 contains the string “hello_world”.

Ex 3 (4.0 points)

Italiano

Scrivere uno script BASH di nome parse che prenda i seguenti parametri da linea di comando:
parse fileIn str1 str2 N str3 fileOut
Lo script deve leggere fileIn e scriverne una copia (di nome fileOut) effettuando le seguenti operazioni:

- Rimpiazzare ogni occorrenza della stringa str1 con la stringa str2.
- Aggiungere un padding destro e sinistro ad ogni riga: ogni riga del file di output deve iniziare e finire con N ripetizioni della stringa str3 (separate da un singolo spazio), come riportato in seguito.

Si supponga che nessuna delle stringhe str1, str2 e str3 contenga spazi bianchi. Lo script deve inoltre verificare che tutti i parametri attesi siano passati da linea di comando.

English

Write a BASH script called parse that takes the following parameters from the command line:
parse fileIn str1 str2 N str3 fileOut
The script must read fileIn and write a copy of it (called fileOut) performing the following two operations:

- Replace all instances of string str1 with string str2.
- Perform a left and right padding of each line: each line of the output file should begin and end with N repetitions of string str3 (separated by a single space), as reported in the following.

Suppose that none of the strings str1, str2 and str3 contains whitespaces. The script should also check that all the expected parameters are passed via the command line.

str3 str3 str3 … (N times) **<line 1 of the file fileOut>** str3 str3 str3 … (N times)
str3 str3 str3 … (N times) **<line 2 of the file fileOut>** str3 str3 str3 … (N times)
…
str3 str3 str3 … (N times) **<last line of the file fileOut>** str3 str3 str3 … (N times)

Soluzione. Solution.

```
# Solution 1
#!/bin/bash

# Check corret number of parameters passed
if [ $# -lt 6 ]; then
    echo "Usage: parse fileIn str1 str2 N str3 fileOut"
    exit 1
fi

# Check input file exists (optional)
if [ ! -e $1 ]; then
    echo "Input file $1 does not exists"
    exit 1
fi

# Collect input parameters
fileIn=$1
str1=$2
str2=$3
N=$4
str3=$5
fileOut=$6

# Read file line-by-line
while read line; do

    # Print padding left
```

```
    i=0
    while [ $i -lt $N ]; do
        echo -n "$str3 "
        let i=i+1
    done

    # Print line subsituting str1 for str2
    for word in $line; do
        if [ $word == $str1 ]; then
            echo -n "$str2 "
        else
            echo -n "$word "
        fi
    done

    # Print padding right
    i=0
    while [ $i -lt $N ]; do
        echo -n "$str3 "
        let i=i+1
    done

    # Print newline
    echo ""
done < $fileIn > $fileOut
```

 # Solution 2

```
#!/bin/bash

# Check corret number of parameters passed
if [ $# -lt 6 ]; then
    echo "Usage: parse fileIn str1 str2 N str3 fileOut"
    exit 1
fi

# Check input file exists (optional)
if [ ! -e $1 ]; then
    echo "Input file $1 does not exists"
    exit 1
fi

# Collect input parameters
fileIn=$1
str1=$2
str2=$3
N=$4
str3=$5
fileOut=$6

# Read file line-by-line
while read line; do

    # Print padding left
    for i in $(seq 0 $N); do
        echo -n "$str3 "
    done
```

```
    # Print line subsituting str1 for str2
    for word in $line; do
        if [ $word == $str1 ]; then
            echo -n "$str2 "
        else
            echo -n "$word "
        fi
    done

    # Print padding right
    for i in $(seq 1 $N); do
        echo -n "$str3 "
    done

    # Print newline
    echo ""
done < $fileIn > $fileOut
```

 # Solution 3

```
#!/bin/bash

# Check corret number of parameters passed
if [ $# -lt 6 ]; then
    echo "Usage: parse fileIn str1 str2 N str3 fileOut"
    exit 1
fi

# Check input file exists (optional)
if [ ! -e $1 ]; then
    echo "Input file $1 does not exists"
    exit 1
fi

# Collect input parameters
fileIn=$1
str1=$2
str2=$3
N=$4
str3=$5
fileOut=$6

# Read file line-by-line
while read line; do

    # Print padding left
    for((i=0; i<$N; i++)); do
        echo -n "$str3 "
    done

    # Print line subsituting str1 for str2
    for word in $line; do
        if [ $word == $str1 ]; then
            echo -n "$str2 "
        else
            echo -n "$word "
```

```
        fi
    done

    # Print padding right
    for((i=0; i<$N; i++)); do
        echo -n "$str3 "
    done

    # Print newline
    echo ""
done < $fileIn > $fileOut
```

 # Solution 4

```
#!/bin/bash

# Check correct number of parameters passed
if [ $# -lt 6 ]; then
    echo "Usage: parse fileIn str1 str2 N str3 fileOut"
    exit 1
fi

# Check input file exists (optional)
if [ ! -e $1 ]; then
    echo "Input file $1 does not exists"
    exit 1
fi

# Collect input parameters
fileIn=$1
str1=$2
str2=$3
N=$4
str3=$5
fileOut=$6

# Compose padding
i=0
padding=""
while [ $i -lt $N ]; do
    padding=$padding"$str3 "
    let i=i+1
done

# Read file line-by-line
while read line; do

    # Print padding left
    echo -n "$padding"

    # Print line subsituting str1 for str2
    for word in $line; do
        if [ $word == $str1 ]; then
            echo -n "$str2 "
        else
            echo -n "$word "
        fi
```

```
    done

    # Print padding right
    echo "$padding"
done < $fileIn > $fileOut
```

Ex 4 (2.5 points)

Italiano

Si analizzi il seguente tratto di codice. Si ipotizzi che in pochi millisecondi tutte le operazioni di output siano completate. Si indichi quanti caratteri 'E' sono visualizzati su standard output quando il programma è eseguito passando 3 come primo argomento. Indicare un unico intero nella risposta.

English

Analyze the following segment of code. Assume that in a few milliseconds all output operations are completed. Indicate how many characters 'E' are displayed on standard output when the program is executed passing 3 as its first argument. Report a single integer value in your response.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
int main (int argc, char *argv[]) {
  char str [50];
  int i;
  setbuf (stdout, 0);
  i = atoi (argv[1]);
  if (i<0)
    exit (0);
  if (fork () > 0) {
    if (fork () > 0) {
      sprintf (str, "echo -n E");
      system (str);
    }
    sprintf (str, "%d", i-2);
    printf ("E");
    execlp (argv[0], argv[0], str, NULL);
  }
  exit (0);
}
```

Risposta. Answer.

9

Ex 5 (3.0 points)

Italiano

Dati tre processi P1, P2 e P3 il cui codice è riportato a seguire e i cui pid siano pid_P1, pid_P2, e pid_P3, rispettivamente. Si indichino quali dei seguenti output sono corretti. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Given three processes P1, P2 and P3 whose code is reported in the following and whose pids are pid_P1, pid_P2, and pid_P3, respectively. Indicate which of the following outputs are correct. Note that incorrect answers imply a penalty in the final score.

```
P1
kill(pid_P2, SIG…);
pause();
```

```
printf("A");
P2
other_code();
pause();
printf("B");
kill(pid_P1, SIGUSR1);
kill(pid_P3, SIGUSR1);
P3
other_code();
pause();
printf("C");
```

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

1. ✅ Nessun output. No output.
2. ✅ B
3. ✅ BC
4. ✅ BA
5. ✅ BCA
6. ✅ BAC

Ex 6 (2.0 points)

Italiano

Dire quali delle seguenti affermazioni relative ai threads sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Indicate which of the following statements related to threads are correct. Note that incorrect answers imply a penalty in the final score.

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

1. ✅ Per comunicare i thread possono usare le variabili global. To communicate threads can use global variables.
2. ✅ La funzione pthread_join() attende la terminazione di uno specifico thread il cui identificatore è passato come parametro alla funzione. The function pthread_join() waits the termination of a specific thread whose identifier is passed as a parameter to the function.
3. ⬜ Ogni thread con l'eccezione del thread con tid=0 ha un padre. Each thread with the exception of the thread with tid=0 has a parent.
4. ⬜ L'esecuzione parallela di un algoritmo per mezzo di più di un thread porta SICURAMENTE alla riduzione del suo tempo di esecuzione. The parallel execution of an algorithm by means of more than one thread leads IN ANY CASE to a reduction on its execution time.
5. ✅ La funzione pthread_exit() eseguita da un thread porta alla terminazione del solo thread che la esegue. The function pthread_exit() executed by one thread leads to the termination of the only thread that executes it.
6. ⬜ L'istruzione return eseguita da un thread porta SICURAMENTE alla terminazione dell'intero processo. The instruction return executed by one thread leads IN ANY CASE to the termination of the whole process.

Ex 7 (2.0 points)

Italiano

Nel seguente frammento di codice, il programma principale crea N thread che eseguono la funzione tF. Lo scopo del programma è quello di eseguire i thread passando ad ognuno di essi un valore intero che identifichi univocamente il thread, cioè un numero intero nell'intervallo da 0 a N-1

Indicare quali delle seguenti osservazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

In the following code snippet, the main program creates N threads that execute the function tF. The purpose of the program is to execute the threads passing to each one an integer value that uniquely identifies the thread, i.e., an integer number ranging from 0 to N-1.
Indicate which of the following observations are correct. Note that incorrect answers imply a penalty in the final score.

```
#define N 10
void *tF (void *par) {
  int *tidP, tid;
  ...
  tidP = (int *) par;
  tid = *tidP;
  ...
  pthread_exit (NULL);
}
int main () {
  pthread_t t[N];
  int rc, i;
  for (i=0; i<N; i++) {
      rc = pthread_create (&t[i], NULL, tF, (void *) &i);
      if (rc) {
      fprintf (stderr, "Error.\n");
      exit (1);
   }
 }
  pthread_exit (NULL);
}
```

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

1. ✅ La variabile tid può contenere il valore 10 per un thread. The variable tid can contain the value 10 for one thread.
2. ⬜ La variabile tid contiene un valore intero diverso per ogni thread. The variable tid contains a different integer value for each thread.
3. ✅ La variabile tid può contenere il valore 10 per tutti i thread. The variable tid can contain the value 10 for all the threads.
4. ⬜ La variabile tid può contenere il valore 7 per tutti i thread. The variable tid can contain the value 7 for all the threads.
5. ⬜ Ad ogni thread è passata una copia della variabile i. A copy of the variable is passed to each thread.

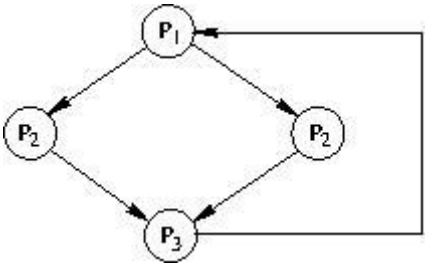
Ex 8 (4.0 points)

Italiano

Dato il seguente grafo di precedenza, realizzarlo utilizzando il minimo numero possibile di semafori. Si utilizzino le primitive semaforiche implementate tramite init, signal, wait e destroy realizzate in pseudo-codice.
I processi rappresentati devono essere processi ciclici (con corpo del tipo while(1)).
Si osservi che il grafo prevede l'esecuzione di 3 processi {P1, P2, P3} e che esistono due istanze (ovviamente identiche) del processo P2 ciascuna delle quali deve essere eseguita una volta (è errato eseguire due volte la stessa istanza per ogni ciclo principale).

English

Given the following precedence graph, implement it using the minimum number of semaphores. Use semaphore primitives implemented through the pseudo-code functions init, signal, wait, and destroy.
The processes represented must be cyclical processes (with body of the type while(1)).
Note that the graph involves the execution of 3 processes {P1, P2, P3} and that there are two instances (obviously identical) of P2 each of which must be executed once (it is incorrect to run the same instance twice per main cycle).



Soluzione. Solution.

```
Initialization:
init (s1, 1);
init (s2, 0);
init (s3, 0);
init (s, 0);
n = 0;

Termination:
destroy (s1);
..
destroy (s);

P1
while (1) {
  wait (s1);
  printf ("P1\n");
  signal (s2);
}

P2
while (1) {
  wait (s2);
  n++;
  if (n==1) {
      signal (s2);
  } else {
      n = 0;
      signal (s);
      signal (s);
  }
  wait (s);
  printf ("P2\n");
  signal (s3);
}

P3
while (1) {
  wait (s3);
  wait (s3);
  printf ("P3\n");
  signal (s1);
}
```

Ex 9 (2.0 points)

Italiano

Dato il seguente codice, si indichi quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Given the following code, indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

```
typedef struct lock_s {
  int ticketNumber;
  int turnNumber;
} lock_t;
int increment (int *var) {
  int tmp = *var;
  *var = tmp + 1;
  return (tmp);
}
void init (lock_t lock) {
  lock.ticketNumber = 0;
  lock.turnNumber = 0;
}
void lock (lock_t lock) {
  int myTurn = increment (&lock.ticketNumber);
  while (lock.turnNumber != myTurn);
}
void unlock (lock_t lock) {
  increment (&lock.turnNumber);
}
```

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

1. ⬜ La funzione increment() chiamata da più processi concorrentemente soffre di race condition sulla variabile tmp. The function increment() called concurrently by many processes suffers of race conditions on the variable tmp.
2. ⬜ Un processo o thread bloccato sulla funzione lock() è in stato di attesa e non usa la CPU. A process or a thread blocked on the function lock() is in a waiting state and does not use the CPU.
3. ✅ A seguito dell'esecuzione da parte di due processi della funzione lock(), la variabile lock.ticketNumber potrebbe essere aumentata di una sola unità (cioè se il suo valore iniziale è 0, quello finale potrebbe essere 1). After the execution by means of two threads of the function lock(), the variable lock.ticketNumber could be incremented of only one unit (i.e., if its initial value is 0, the final value could be 1).
4. ✅ Le funzioni lock() e unlock() possono essere utilizzate per gestire l'accesso a una sezione critica nel caso in cui increment() sia eseguita in modo atomico. Functions lock() and unlock() can be used to manage the access to a critical section in the case the function increment() is executed in an atomic way.

Ex 10 (2.0 points)

Italiano

Nel seguente frammento di codice, due gruppi di entità (uno riportato a destra e l'altro riportato a sinistra) si sincronizzano per accedere alla sezione critica CS utilizzando i semafori s1, s2 e busy. I semafori sono inizializzati come indicato all'inizio del frammento di codice. Indicare quali delle seguenti osservazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

In the following code snippet, two groups of entities (one on the left and one on the right-hand side) synchronize to enter a critical section CS using semaphores s1, s2, and busy. These semaphores are initialized as represented on top. Indicate which ones of the following observations are correct. Note that incorrect answers imply a penalty in the final score.

```
Initialization
```

```
n1 = n2 = 0;
init (s1, 1);
init (s2, 1);
init (busy, 1);
```

First group of N (>1) entities	**Second group of M (>1) entities**

`wait (s1);`
`  n1++;`
`  if (n1 == 1)`
`  wait (busy);`
`signal (s1);`
`CS`
`wait (s1);`
`  n1--;`
`  if (n1 == 0)`
`  signal (busy);`
`signal (s1);`
	`wait (s2);`
	`  n2++;`
	`  if (n2 == 1)`
	`  wait (busy);`
	`signal (s2);`
	`CS`
	`wait (s2);`
	`  n2--;`
	`  if (n2 == 0)`
	`  signal (busy);`
	`signal (s2);`

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

1. ⬜ Solo un'entità del primo gruppo può entrare nella sezione critica. Only one entity of the first group can enter the critical section.
2. ✅ N entità del primo gruppo possono entrare nella sezione critica. N entities of the first group can enter the critical section.
3. ✅ M entità del secondo gruppo possono entrare nella sezione critica. M entities of the second group can enter the critical section.
4. ⬜ Solo un'entità del secondo gruppo può entrare nella sezione critica. Only one entity of the second group can enter the critical section.
5. ⬜ Il semaforo busy deve essere inizializzato a zero per realizzare il problema classico del "Tunnel a senso alternato". The semaphore busy must be initialized to zero to implement the classical problem of the "Alternate direction tunnel".
6. ✅ N entità del primo gruppo o M entità del secondo gruppo possono entrare nella sezione critica ma non allo stesso tempo. N entities of the first group or M entities of the second one can enter the critical section, but not at the same time.

Ex 11 (2.0 points)

Italiano

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

English

Given the following code, indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

```
1 : init (full, 0); init (empty, SIZE); init (mutex, 1); int in=0, out=0,
buff[SIZE];
2 : in (int val) {
3 :   wait (empty);
4 :   wait (mutex);
5 :   buff[in]=val;
6 :   in=(in+1)%SIZE;
7 :   signal (mutex);
8 :   signal (full);
9 : }
10: int out () {
11:   int res;
12:   wait (full);
```

```
13:   wait (mutex);
14:   res=buff[out];
15:   out=(out+1)%SIZE;
16:   signal (mutex);
17:   signal (empty);
18:   return(res);
19: }
```

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

1. ✅ L'inversione dell'ordine delle due istruzioni signal (signal(mutex) and signal(empty)) in linee 16 e 17 non modifica sostanzialmente il comportamento di questo codice. Inverting the order of the two signal instructions (signal(mutex) and signal(empty)) in lines 16 and 17 does not substantially modify the behavior of this code.
2. ✅ L'uso di due mutex diversi per le due funzioni (usando cioè wait(mutex1) e signal(mutex1) per in(), e wait(mutex2) e signal(mutex2) per out()) incrementa il livello di concorrenza. / The use of two different mutexes for the two functions (i.e., using wait(mutex1) and signal(mutex1) for in(), and wait(mutex2) and signal(mutex2) for out()) increases the level of concurrency.
3. ⬜ Questo codice risolve il problema dei lettori e degli scrittori. This code solves the problem of readers and writers.
4. ⬜ L'istruzione wait (empty); è utilizzata per bloccare i processi quando il buffer è vuoto. The instruction sem_wait (empty); is used to block processes when the buffer is empty.
5. ✅ L'inversione dell'ordine delle due istruzioni wait (wait(empty) and wait(mutex)) in linee 3 e 4 può generare stallo/deadlock. Inverting the order of the two wait instructions (wait(empty) and wait(mutex)) in lines 3 and 4 may lead to deadlock.

Ex 12 (3.0 points)

Italiano

Si consideri il seguente insieme di processi. Rappresentare mediante diagramma di Gantt l'esecuzione di tali processi utilizzando l'algoritmo Shortest Remaining Time First (SRTF), al fine di calcolare il tempo di terminazione di ciascun processo e il tempo di attesa medio.
Si prega di riportare la risposta su un'unica riga, indicando i tempi di terminazione di P1, P2, P3, P4 e P5 seguiti dal tempo di attesa medio. Separare i numeri con un unico spazio. Riportare il tempo di attesa medio con 1 sola cifra decimale. Non inserire nessun altro carattere nella risposta. Esempio di risposta corretta: 20 23 11 45 67 30.5

English

Consider the following set of processes. Represent using a Gantt diagram the execution of these processes using the Shortest Remaining Time First (SRTF) scheduling algorithm, in order to compute the termination time of each process, and the average waiting time.
Please, write your answer on a single line, indicating the termination times of P1, P2, P3, P4 and P5 followed by the average waiting time. Separate the numbers with a single space. Report the average waiting time with a single decimal digit. Do not enter any other character in the response. Example of correct answer: 20 23 11 45 67 30.5

Processo Process	TempoArrivo ArrivalTime	BurstTime	Priorità Priority
P1	0	25	1
P2	3	21	3
P3	6	17	5
P4	9	13	4
P5	12	15	2

Fasi operative. Workflow.

```
SRTF:
  P1: from=  0 to  =  2 end =  3 waitingTime =  0
  P2: from=  3 to  =  5 end =  6 waitingTime =  0
```

```
  P3: from=  6 to  =  8 end =  9 waitingTime =  0
  P4: from=  9 to  = 21 end = 22 waitingTime =  0
  P3: from= 22 to  = 35 end = 36 waitingTime = 13
  P5: from= 36 to  = 50 end = 51 waitingTime = 24
  P2: from= 51 to  = 68 end = 69 waitingTime = 45
  P1: from= 69 to  = 90 end = 91 waitingTime = 66
  EndTime: 91 69 36 22 51
  AverageWaitingTime: [( 66 ) + ( 45 ) + ( 13 ) + (  ) + ( 24 ) ] / 5 = 29.60
  AverageTournaroudTime: ( 91 + 66 + 30 + 13 + 39 ) / 5 = 47.80
  GanttChart:
11122233344444444444443333333333333335555555555555552222222222222222211111111111111
1111111
```

Risposta. Answer.

```
91 69 36 22 51 29.6
```

Ex 13 (3.0 points)

Italiano

Lo stato di un sistema con 6 processi e 3 tipologie di risorse è definito dalle matrici riportate alla fine della domanda.
Si esegua l'algoritmo del banchiere e si riporti il vettore "Disponibili" che si ottiene a seguito dell'esecuzione dell'algoritmo.
Riportare la soluzione come tre numeri separati da spazi.

English

The state of a system with 6 processes and 3 types of resources is defined by the matrices reported at the end of the question.
Execute the Banker's Algorithm, and report the vector "Available" that is obtained after the executing of the algorithm.
Report the solution as three numbers separated by spaces.

Processo Process	Assegnate Allocation	Massimo Max	Disponibile Available
P1	1 0 1	4 3 4	4 2 2
P2	0 0 1	4 4 2	
P3	0 1 1	2 3 3	
P4	0 1 0	5 4 5	
P5	0 1 0	1 4 4	
P6	0 0 0	2 5 5	

Risposta. Answer.

```
5 5 5
```

Ex 14 (2.0 points)

Italiano

Quali delle seguenti affermazioni sulle pipe sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Which of the following pipe 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. ⬜ Un'operazione di lettura su una pipe è sempre bloccante. A read operation on a pipe is always blocking.

2. ⬜ Su una pipe non si possono effettuare operazioni di lettura/scrittura di dati strutturati (e.g., una struct) se non utilizzando una sequenza di operazioni di read/write successive. On a pipe it is not possible to execute read/write operations on a composite data structure (such as a struct) if not using a sequence of read/write operations.
3. ✅ Una pipe si riempie quando lo scrittore scrive troppo sulla pipe senza che il lettore legga nulla. A pipe gets filled up when the writer writes too much to the pipe without the reader reading any of it.
4. ✅ Un'operazione di scrittura è tipicamente bloccante su una pipe piena. A write operation is typically blocking on a full pipe.
5. ✅ Le operazioni di scrittura su una pipe sono atomiche fino alla dimensione della costante PIPE_BUF. Write operations on a pipe are atomic up to the size of the constant PIPE_BUF.
6. ⬜ Si può scrivere da ambo il lati della pipe contemporaneamente. It is possible to write on both sides of the pipe at the same time.