Esame Sistemi Operativi - Operating Systems Exam
2021/09/07

Ex 1 (3.0 points)

Italiano

Si descriva che cosa è un processo orfano e che cosa è un processo zombie.
Si riportino quindi due segmenti di codice: il primo che genera un processo orfano e il secondo che genera un processo zombie.

English

Describe what is an orphan process and then what is a zombie process.
Report two code segments: the first one generating an orphan process and the second one generating a zombie process.

Soluzione. Solution.

Un processo orfano è un processo figlio in cui il padre è terminato prima del processo figlio. Quando questo accade, i processi orfani sono "adottati" da uno speciale processo del sistema operativo (tipicamente il processo init). Un processo zombie è un processo figlio che termina prima che il suo padri chiami la system call wait().
An orphan process is a child process in which the parent is terminated before the child process. When this happens, orphan processes are "adopted" by a unique of process (typically the init process). A zombie process is a child process that terminates before its parent called the system called wait().

```
// Orphan process
if(fork()){
	exit(0);
} else{
	// Orphan process
	sleep(1); // This orphan process will wait for a second so that we are sure
that the parent terminated
	exit(0);
}

// Zombie process
if(fork()){
	sleep(1); // Same as before but this time for the parent
	exit(0);
} else{
	//zombie process
	exit(0);
}
```

Ex 2 (1.5 points)

Italiano

Si supponga un thread esegua la seguente istruzione:
`pthread_detach (pthread_self ());`
Si indichino quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Suppose a thread executes the following instruction:
`pthread_detach (pthread_self ());`
Which of the following observations are correct (possibly more than one)? Note that incorrect answers may imply a penalty on the final score.

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

1. ⬜ Il thread che ha eseguito tale istruzione NON POTRÀ effettuare una pthread_join. The thread that executed this instruction CANNOT perform a pthread_join.
2. ✅ Il thread che ha generato tale thread NON POTRÀ effettuare una pthread_join (con tid relativo al thread che ha eseguito l'istruzione pthread_detach). The thread that created this thread CANNOT perform a pthread_join (with tid related to the thread that executed the instruction pthread_detach).
3. ⬜ Il thread che ha eseguito tale istruzione NON POTRÀ effettuare una pthread_create. The thread that executed this instruction CANNOT perform a pthread_create.
4. ✅ Le informazioni di stato saranno perse alla terminazione del thread. The status information will be lost at the termination of the thread.
5. ⬜ Il thread che ha eseguito tale istruzione NON POTRÀ effettuare una pthread_exit. The thread that executed this instruction CANNOT perform a pthread_exit.

Ex 3 (3.0 points)

Italiano

Si analizzi il seguente tratto di codice. Si ipotizzi che in pochi millisecondi tutte le operazioni di output siano completate. Si indichi qual è l'output generato dal programma. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Analyze the following segment of code. Assume that in a few milliseconds all output operations are completed. Indicate which is the output generated by the program. Note that incorrect answers imply a penalty in the final score.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main(){
      int i;
      for (i=0; i<4 && !fork(); i++){
            if (fork()) {
                  sleep (1);
                  system ("echo i+");
            } else
                  execlp ("echo", "system", "i++", NULL);
      }
}
```

Scegli UNA SOLA alternativa. Choose JUST ONE option.

1. ⬜ `i++ 0+ i++ 1+ i++ 2+ i++ 3+`
2. ⬜ `0++ 0+ 1++ 1+ 2++ 2+ 3++ 3+`
3. ⬜ `0++ i+ 1++ i+ 2++ i+ 3++ i+`
4. ✅ `i++ i+ i++ i+ i++ i+ i++ i+`
5. ⬜ `i+ i++ i+ i++ i+ i++ i+ i++`

Ex 4 (1.5 points)

Italiano

Si considerino i metodi di codifica delle informazioni su file. Si indichino quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Consider the methods for encoding information in a file. Indicates which ones among the following observations are correct (possibly more than one). Note that incorrect answers may imply a penalty on the final score.

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

1. ⬜ I caratteri in UNICODE sono sempre salvati su 32 bits / UNICODE characters are always stored on 32 bits.
2. ✅ UNICODE è un estensione di ASCII usata per rappresentare più di 256 caratteri / UNICODE is an extension of ASCII used to represent more than 256 characters.
3. ✅ I file ASCII sono spesso più compatti rispetto ai file codificati in UNICODE / ASCII files are often more compact than UNICODE files.
4. ✅ Ci sono molte versioni della tabella ASCII, per tenere traccia delle differenti lingue e notazioni / There are several versions of the ASCII table, to keep track of the different languages and notations.
5. ⬜ La codifica UNICODE è usata per salvare file in formato binario / The UNICODE encoding is used to store files in binary format.
6. ✅ I file binari sono spesso più compatti rispetto ai file codificati in UNICODE / Binary files are often more compact than UNICODE files.
7. ⬜ Le codifiche ASCII e UNICODE differiscono sempre / The ASCII and UNICODE encodings always differ

Ex 5 (2.0 points)

Italiano

Il seguente script, di nome script.sh:

```
echo -n $#
for x in $* ; do echo -n $x ; done
```

viene eseguito con la seguente riga di comando:

```
bash script.sh ci sono '$0' parole
```

Che cosa viene visualizzato su standard output? Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

The following script, named script.sh:

```
echo -n $#
for x in $* ; do echo -n $x ; done
```

is run with the following command line:

```
bash script.sh ci sono '$0' parole
```

What does it display on standard output? Note that incorrect answers imply a penalty in the final score.

Scegli UNA SOLA alternativa. Choose JUST ONE option.

1. ⬜ `4 ci sono $0 parole`
2. ⬜ `4 ci sono ci sono parole parole`
3. ✅ `4cisono$0parole`
4. ⬜ `4cisonoparole`
5. ⬜ `4cisonocisonoparoleparole`

Ex 6 (3.0 points)

Italiano

Si analizzi il seguente tratto di codice.

Si indichi qual è l'output generato dal programma. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Analyze the following segment of code.

Indicate which is the output generated by the program. Note that incorrect answers imply a penalty in the final score.

```
#include <stdio.h>
#include <pthread.h>

void *t2 ();
```

```
int n = 5;
int m = 3;
void *t1 (){
	pthread_t thread;
	if (n>0) {
		printf ("%d ", n--);
		pthread_create (&thread, NULL, t2, NULL);
	}
	pthread_join (thread, NULL);
	pthread_exit (NULL);
}

void *t2 (){
	pthread_t thread;
	if (m>0) {
		printf ("%d ", m--);
		pthread_create (&thread, NULL, t1, NULL);
	}
	pthread_join (thread, NULL);
	pthread_exit (NULL);
}

int main (int argc, char *argv[]) {
	pthread_t thread;
	setbuf (stdout, 0);
	pthread_create (&thread, NULL, t1, &n);
	pthread_join (thread, NULL);
	return 1;
}
```

Scegli UNA SOLA alternativa. Choose JUST ONE option.

1. ⬜ 3 2 1 5 4 3 2
2. ⬜ 5 3 4 2 3
3. ⬜ 5 3 4 2 3 1 2 0 1
4. ✅ 5 3 4 2 3 1 2
5. ⬜ 5 4 3 2 3 2 1

Ex 7 (1.5 points)

Italiano

Si consideri la seguente affermazione: “Se un thread finisce, tutti gli altri thread del processo necessariamente terminano”. Si indichi in quali delle seguenti circostanze quanto affermato si verifica. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Consider the following statement: "If one thread terminates, all other threads in the process will necessarily terminate." Indicate in which circumstances the previous statement is correct, among the following ones. Note that incorrect answers imply a penalty in the final scor.e

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

1. ⬜ Quando il thread effettua una return dalla sua funzione iniziale. When the thread performs a return from its start function.
2. ✅ Quando il thread effettua una exit. When the thread performs and exit.
3. ✅ Quando il thread effettua una return dal main. When the thread performs a return from the main.
4. ⬜ Quando il thread riceve una pthread_cancel da un altro thread. When the thread receives a pthread_cancel from another thread.
5. ⬜ Quando il thread effettua una pthread_exit. When the thread performs a pthread_exit.

Ex 8 (3.0 points)

Italiano

Si analizzi il seguente tratto di codice. Si indichi qual è l'output generato dal programma. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Analyze the following segment of code. Indicate which is the output generated by the program. Note that incorrect answers imply a penalty in the final score.

```
#include <stdio.h>
#include <stdlib.h>
#include <semaphore.h>
#include <pthread.h>

sem_t *sa, *sb, *sc, *me, *ss;
int n;
static void *TA ();
static void *TB ();
static void *TC ();

int main (int argc, char **argv) {
      pthread_t th;
      sa = (sem_t *) malloc (sizeof(sem_t));
      sb = (sem_t *) malloc (sizeof(sem_t));
      sc = (sem_t *) malloc (sizeof(sem_t));
      me = (sem_t *) malloc (sizeof(sem_t));
      ss = (sem_t *) malloc (sizeof(sem_t));
      n = 0;
      sem_init (sa, 0, 0);
      sem_init (sb, 0, 0);
      sem_init (sc, 0, 0);
      sem_init (me, 0, 1);
      sem_init (ss, 0, 2);
      setbuf(stdout, 0);
      pthread_create (&th, NULL, TA, NULL);
      pthread_create (&th, NULL, TB, NULL);
      pthread_create (&th, NULL, TC, NULL);
      pthread_exit(0);
}

static void *TA () {
      pthread_detach (pthread_self ());
      while (1) {
            sem_wait (ss);
            sem_wait (me);
            printf ( "A");
            sem_post (me);
            n++;
            if (n==2)
                  sem_post (sc);
            sem_wait (sa);
      } return 0;
}

static void *TB () {
```

```
	pthread_detach (pthread_self ());
	while (1) {
		sem_wait (ss);
		sem_wait (me);
		printf ( "B");
		sem_post (me);
		n++;
		if (n==2)
			sem_post (sc);
		sem_wait (sb);
	}
	return 0;
}

static void *TC () {
	pthread_detach (pthread_self ());
	while (1) {
		sem_wait (sc);
		n = 0;
		printf ("C\n");
		sem_post (sa);
		sem_post (sb);
		sem_post (ss);
		sem_post (ss);
	}
	return 0;
}
```

Scegli UNA SOLA alternativa. Choose JUST ONE option.

1. ⬜ Una sequenza di stringhe “ACB”. A sequence of strings "ACB".
2. ✅ Una sequenza di stringhe "ABC" e "BAC". A sequence of strings "ABC" and "BAC".
3. ⬜ Una sequenza di stringhe “ABC”. A sequence of strings "ABC".
4. ⬜ Una sequenza di stringhe “BCA”. A sequence of strings "BCA".
5. ⬜ Una sequenza di stringhe “BAC”. A sequence of strings "BAC".

Ex 9 (3.0 points)

Italiano

Si descriva che cosa è una pipe e come può essere utilizzata.
Perché è buona pratica chiudere i terminali inutilizzati di una pipe?
Si riporti il codice C che illustri come trasferire un insieme di stringhe di diversa dimensione, in modo che sia posssibile leggerle correttamente dall'altro lato della pipe.

English

Describe what is a pipe and how it can be used.
Why is it a good practice to close the unused terminals of a pipe?
Report a C code snippet to show how to transfer a set of strings of variable size, such that is is possible to correctly read them on the other side of the pipe.

Soluzione. Solution.

Una pipe è un metodo di comunicazione tra processi. Può essere utilizzata per spedire dati o per la sincronizzazione tra processi. La pipe deve essere creata prima della fork per fare in modo che entrambi i processi, padre e figlio, possano accedere ai descrittori della pipe. Una volta creata, essa può essere utilizzato in modo simile ai file attraverso le system calls "read" e "write". La read è un operazione bloccante se la pipe è vuota. Se tutti i descrittori di file dei lettori sono chiusi ed è eseguita un'operazione di scrittura, un segnale di tipo SIGPIPE è spedito. Invece se un lettore cerca di leggere su una pipe in cui non esiste nessuno scrittore, è restituito il numero "0". Anche se le pipes sono un metodo di comunicazione di tipo half-duplex (cioè possono

essere spedite informazioni in entrambe le direzioni, ma solo in una direzione per volta), è buona pratica chiudere uno dei due estremi (0 è tipicamente usato per la lettura, mentre 1 per la scrittura) al fine di evitare di scrivere in entrambi i lati della pipe allo stesso tempo. Per una comunicazione di tipo full-duplex dovrebbero essere utilizzate due pipe.
A pipe is an interprocess communication method. It can be used to send data and for synchronization between processes. The pipe must be created before the fork so that both parent and child can access the pipe descriptors. Once created, it can be used similarly to files with "read" and "write" system calls. Read is a blocking operation if the pipe is empty. If all the file descriptors of the readers are closed and a write operation is performed, a SIGPIPE signal is sent, while if a reader tries to read from a pipe with no writers, "0" is returned. Even if Pipes are a Half-duplex communication method (i.e., they could send information in each direction, but only one direction at a time), it is a good practice to close one of the two ends in its process (0 is commonly used for reading while 1 for writing) to avoid writing on both sides at the same time. For full-duplex communication, two pipes should be used.

```
#define N ...

int main () {
	int fd[2];
	char line[N];

	pipe (fd);

	if (fork()) {
		close (fd[0]);
		while (1) {
			scanf ("%s", line);
			write (fd[1], line, (strlen(line)+1)*sizeof(char));
		}
	} else {
		close (fd[1]);
		while (1) {
			read (fd[0], line, N*sizeof(char));
			printf ("%s\n", line);
		}
	}
	return (0);
}
```

Ex 10 (2.5 points)

Italiano

Si consideri il seguente insieme di processi schedulati con un quantum temporale di 20 unità di tempo. Rappresentare mediante diagramma di Gantt l'esecuzione di tali processi utilizzando l'algoritmo Round Robin (RR). Calcolare il tempo totale di attesa di ciascun processo. Riportare tali tempi su un'unica riga, indicando i tempi totali di attesa in ordine per P1, P2, P3, P4, P5 e P6. Si separino i valori con un singolo spazio e non si inserisca nessun altro carattere nella risposta. Errori di formato saranno considerati come ogni altro errore. Esempio di risposta corretta: 20 23 11 45 67 102

English

Consider the previous set of scheduled processes with a time quantum of 20 units of time. Represent the execution of these processes using the Round Robin (RR) algorithm by Gantt chart. Compute the total waiting time of each process. Please report the answer on a single line, indicating the total waiting times of P1, P2, P3, P4, P5, and P6. Separate the numbers with a single space and do not insert any other character into the answer. Format errors will be treated as other errors. Example of correct answer: 20 23 11 45 67 102

Processo Process	TempoArrivo ArrivalTime	BurstTime	Priorità Priority
P1	0	12	3
P2	2	27	1
P3	4	15	4
P4	6	13	5
P5	8	29	2
P6	10	12	6

Fasi operative. Workflow.

```
RR:
  P1: from=  0 to  = 11 end = 12 waitingTime =  0
  P2: from= 12 to  = 31 end = 32 waitingTime = 10
  P3: from= 32 to  = 46 end = 47 waitingTime = 28
  P4: from= 47 to  = 59 end = 60 waitingTime = 41
  P5: from= 60 to  = 79 end = 80 waitingTime = 52
  P6: from= 80 to  = 91 end = 92 waitingTime = 70
  P2: from= 92 to  = 98 end = 99 waitingTime = 60
  P5: from= 99 to  = 107 end = 108 waitingTime = 19
  EndTime: 12 99 47 60 108 92
  AverageWaitingTime: [(  0 ) + ( 10 + 60 ) + ( 28 ) + ( 41 ) + ( 52 + 19 ) + ( 70 )
] / 6 = 46.67
  AverageTournaroudTime: ( 12 + 97 + 43 + 54 + 100 + 82 ) / 6 = 64.67
  GanttChart:
111111111111222222222222222222233333333333333344444444444445555555555555555555666666666666622222225555555555
```

Risposta. Answer.

```
0 70 28 41 71 70
```

Ex 11 (1.5 points)

Italiano

Si supponga che il disco rigido di un piccolo sistema embedded sia costituito da 24 blocchi di 1 MByte, che tali blocchi siano numerati da 0 a 23, che il sistema operativo mantenga traccia dei blocchi liberi (occupati) indicandoli in un vettore con il valore 0 (1), e che la situazione attuale del disco sia rappresentata dal seguente vettore:

```
0 1 1 0 0 0 1 0 0 1 1 1 1 0 0 0 1 0 1 0 0 1 0 0
```

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

English

Suppose that the hard disk of a small embedded system is composed of 24 blocks of 1 MByte each, which are 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 disk is represented by the following vector:

```
0 1 1 0 0 0 1 0 0 1 1 1 1 0 0 0 1 0 1 0 0 1 0 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. ⬜ Dopo aver allocato mediante allocazione contigua e algoritmo FIRST-FIT due file da 1.9 MByte, un ulteriore file da 2.6 MByte NON può essere allocato con la stessa strategia. After allocating with the continuous allocation strategy based on the FIRST-FIT algorithm two files with dimension 1.9 MByte, an additional file of 2.6 MByte CANNOT be allocated with the same strategy.

2. ✅ Con la strategia di allocazione concatenata un file di dimensione 13.5 MByte PUO' essere allocato. With the linked allocation strategy a file of dimension 13.5 MBytes CAN be allocated.
3. ⬜ Con la strategia di allocazione indicizzata un file di dimensione 13.5 MByte PUO' essere allocato. With the indexed allocation strategy a file of dimension 13.5 MBytes CAN be stored.
4. ✅ Due file con dimensione 2.9 MByte POSSONO essere allocati utilizzando una strategia di allocazione concatenata. Two files with dimension 2.9 MByte CAN be allocated using a linked allocation strategy.

Ex 12 (5.0 points)

Italiano

Scrivere uno script BASH che riceva come argomenti un elenco di file e, per ogni file, se esistente, stampi le informazioni seguenti:

1. Se il file è un file regolare, stampa il nome, la dimensione e indica se l'utente che esegue lo script dispone delle autorizzazioni di lettura e di scrittura sul file.
2. Se il file è un direttorio, stampa il nome e il numero di sottodirettori in esso contenuti.

Si noti che l'output del comando `ls -l` ha il seguente formato:

```
drwx------ 12 user user 408 Oct 30 19:09 Desktop
-rw-r--r-- 1 user user 192 Jul 13 00:03 pgrm
-rwxr-xr-x 1 user user 74 Nov 3 10:02 ex.awk
drwxrwxrwx 22 user user 408 Oct 30 12:09 tmp
```

English

Write a BASH script that receives as arguments a list of files, and for each file, if it exists, it prints the following information: 1. If the file is a regular file, it prints its name, its dimension, and if the user that runs the script has read and write permissions for the file. 2. If the file is a directory, it prints its name, and how many sub-directories it contains. Notice that the output of command `ls -l` is like the following one:

```
drwx------ 12 user user 408 Oct 30 19:09 Desktop
-rw-r--r-- 1 user user 192 Jul 13 00:03 pgrm
-rwxr-xr-x 1 user user 74 Nov 3 10:02 ex.awk
drwxrwxrwx 22 user user 408 Oct 30 12:09 tmp
```

Soluzione. Solution.

```
#!/bin/bash
################################################################################
Exercise 15 of exam 07/09/2021 - Version A #
################################################################################

for path in $*; do
	if [ -f $path ]; then
		dim=$(ls -l $path | tr -s " " | cut -d " " -f 5)
		read_perm=""
		if [ -r $path ]; then
			read_perm="R"
		fi
		write_perm=""
		if [ -w $path ]; then
			write_perm="W"
		fi
		echo "FILE: $path $dim $read_perm $write_perm"
	elif [ -d $path ]; then
		subdirs=$(find $path -maxdepth 1 -mindepth 1 -type d | wc -l)
		echo "DIR: $path $subdirs"
	fi
done
```

```
#!/bin/bash
################################################################################
Exercise 15 of exam 07/09/2021 - Version B #
################################################################################for
r path in $@; do
      if [ -f $path ]; then
            dim=$(ls -l $path | tr -s " " | cut -d " " -f 5)
            read_perm=""
            if [ -r $path ]; then
                  read_perm="R"
            fi
            write_perm=""
            if [ -w $path ]; then
                  write_perm="W"
            fi
            echo "FILE: $path $dim $read_perm $write_perm"
      elif [ -d $path ]; then
            subdirs=$(ls -l $path | grep -e "^d" | wc -l)
            echo "DIR: $path $subdirs"
      fi
done
```

Ex 13 (2.5 points)

Italiano

Si analizzi il seguente tratto di codice. Si indichi quanti caratteri 'X' vengono visualizzati. Si osservi che risposte errate implicano una penalità nel punteggio finale

English

Analyze the following segment of code. Indicate how many characters 'X' are displayed. Note that incorrect answers imply a penalty in the final score

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main () {
      int i;
      i = 0;
      setbuf(stdout,0);
      while (i<=2 && fork()) {
            if (fork ()) {
                  printf ("X");
            }
            i++;
      }
      return 1;
}
```

Scegli UNA SOLA alternativa. Choose JUST ONE option.

1. ⬜ 14
2. ⬜ 8
3. ⬜ 6
4. ⬜ 12
5. ✅ 7

Ex 14 (1.5 points)

Italiano

Si supponga un processo esegua l'istruzione:
`signal(SIGCHLD, SIG_IGN);`
Si indichino quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.
English
Suppose a process executes the following instruction:
`signal(SIGCHLD, SIG_IGN);`
Indicate which ones among the following observations are correct (possibly more than one). Note that incorrect answers may imply a penalty on the final score.

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

1. ✅ Il processo gestirà con la funzione di default SIG_IGN (definita tramite macro in signal.h) i segnali di tipo SIGCHLD. The process will handled signals of type SIGCHLD with the default function SIG_IGN (which is defined as a macro in signal.h).
2. ✅ Il processo non dovrà eseguire una wait o una waitpid. The process will not have to execute a wait or a waitpid.
3. ⬜ Il processo si comporterà in maniera standard alla ricezione di un SIGCHLD. The process will behave in a standard way when a SIGCHLD is received.
4. ✅ Il processo ignorerà segnali di tipo SIGCHLD. The process will ignore signals of type SIGCHLD.
5. ✅ Se il processo eseguirà una wait questa restituirà un codice di errore. If the process executes a wait, it will return an error code.
6. ⬜ Il processo figlio potrà diventare zombie. The child process can become zombie.

Ex 15 (1.5 points)

Italiano
Si supponga T1 e T2 siano due thread, che m1 e m2 siano due mutex inizializzati a 1, e che v sia una variabile globale inizializzata a 0. I due thread T1 e T2 includono i tratti di codice riportati di seguito. Si indichino quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.
English
Suppose T1 and T2 are two threads, m1 and m2 are two mutexes initialized to 1, and v is a global variable initialized to 0. The two threads T1 and T2 include the following code snippets. Please indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

```
// Thread 1
pthread_mutex_lock(&m1);
v++;
pthread_mutex_unlock(&m1);

// Thread 2
pthread_mutex_lock(&m2);
v++;
pthread_mutex_unlock(&m2);
```

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

1. ✅ A seguito dell'esecuzione di un'istanza di T1 e di un'istanza di T2 la variabile v può valere 1. After the execution of an instance of T1 and an instance of T2, the variable v can be equal to 1.
2. ✅ La mutua esclusione sulla variabile v non è organizzata correttamente. Mutual exclusion on variable v is not organized correctly.
3. ✅ A seguito dell'esecuzione di un'istanza di T1 e di un'istanza di T2 la variabile v può valere 2. After the execution of an instance of T1 and an instance of T2, the variable v can be equal to 2.
4. ⬜ Occorre che uno dei due mutex sia inizializzato a 0 e l'altro a 1. One of the two mutexes must be initialized to 0 and the other to 1.
5. ⬜ La variabile v è protetta correttamente. The variable v is protected correctly.

6. ⬜ A seguito dell'esecuzione di un'istanza di T1 e di un'istanza di T2 la variabile v può valere 0. After the execution of an instance of T1 and an instance of T2, the variable v can be equal to 0.
7. ✅ La variabile v può essere modificata contemporaneamente dai due thread. The variable v can be modified simultaneously by the two threads.