

Esame Sistemi Operativi - Operating Systems Exam

2023/04/28

Ex 1 (5.0 points)

Italiano

Si illustri il problema dei Produttore e Consumatore riportandone la soluzione per il caso di un solo produttore e di tre consumatori. Si indichi la funzione dei vari semafori motivandone l'utilizzo.

English

Provide a solution to the Producer and Consumer problem considering the case in which there is a single producer and three consumers. Indicate the purpose of each semaphore motivating how it is used in the solution.

Risposta: Answer:

```
init (full, 0);
init (empty, MAX);
init (MEc, 1);
```

```
Producer () {
    Message m;
    while (TRUE) {
        produce m;
        wait (empty);
        enqueue (m);
        signal (full);
    }
}
```

```
Consumer () {
    Message m;
    while (TRUE) {
        wait (full);
        wait (MEc);
        m = dequeue ();
        signal (MEc);
        signal (empty);
        consume m;
    }
}
```

The empty semaphore counts the number of empty elements, and it blocks the producer in case the buffer is full.

The full semaphore counts the number of full elements, and it blocks the consumer in case the buffer is empty.

The semaphore meC forces mutual exclusion between several consumers.

On the producers, the semaphore is not needed as there is only one producer.

Ex 2 (4.0 points)

Italiano

Si supponga che il seguente programma venga eseguito mediante l'istruzione `./pgrm abc 2`. Si indichi un possibile output del programma.

English

Suppose that the following program is run using the command `./pgrm abc 2`. Indicate a possible output of the program.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#define L 100

int main (int argc, char *argv[]) {
    int i, j;
    char str1[L], str2[L];
    i = atoi (argv[2]);
    for (j=0; j<i; j++) {
        if (fork () == 0) {
            sprintf (str1, "echo system with j=%d", j);
            system (str1);
        } else {
            sprintf (str1, "%s", argv[1]);
            sprintf (str2, "%d", j);
            printf ("exec ./pgrm %s %s\n", str1, str2);
            execlp ("./pgrm", "myPgrm", str1, str2, NULL);
        }
    }
    return (0);
}

```

Risposta: [Answer:](#)

```

exec ./pgrm abc 0
system with j=0
exec ./pgrm abc 1
exec ./pgrm abc 0
system with j=1
system with j=0

```

Ex 3 (5.0 points)

Italiano

Implementare uno script BASH che riceva sulla linea di comando quattro argomenti: `dir`, `dim1`, `dim2` e `canc`.

Lo script deve:

- controllare il corretto passaggio dei parametri
- cercare tutti i file
 - nell'albero di direttori con radice `dir` e profondità compresa tra 3 e 5 (estremi inclusi)
 - con estensione `sh`
 - con dimensione minore di `dim1+dim2`
- cancellare ogni riga dei file trovati che inizia oppure termina con la stringa contenuta nella variabile `canc`.

English

Write a BASH script that receives four arguments from the command line: `dir`, `dim1`, `dim2` and `canc`.

The script should:

- Check that all the arguments are passed correctly
- Find all the files that:
 - Reside in the directory tree rooted in `dir` at a depth between 3 and 5 (boundaries included)
 - Have extension `sh`
 - Whose size is less than `dim1+dim2`
- Delete each line in the files found that starts or terminates with the string defined by `canc`.

Risposta: [Answer](#):

```
#!/bin/bash

# Check if the correct parameters has been passed
if [ $# -lt 4 ]; then
    echo "Usage: ./ex3.sh <dir> <dim1> <dim2> <canc>"
    exit 1
fi

# Compute minimum file size
let "dim=$2+$3"

# Search all files that satisfy the given conditions
find $1 -mindepth 3 -maxdepth 5 -name "*.sh" -size -${dim}k > "list.txt"

# Delete lines starting or ending with the given string
while read file; do
    grep -v "^$4|$4$" $file > /tmp/tmp$$
    mv /tmp/tmp$$ $file
done < "list.txt"

# Remove temporary file
rm "list.txt"
```

Ex 4 (5.0 points)

Italiano

Si supponga di avere a disposizione l'implementazione Pthread dei semafori binari attraverso le funzioni:

- `int pthread_mutex_init (pthread_mutex_t *mutex, const pthread_mutexattr_t *attr);`
- `int pthread_mutex_lock (pthread_mutex_t *mutex);`
- `int pthread_mutex_unlock (pthread_mutex_t *mutex);`

Si mostri come sia possibile implementare un semaforo con conteggio, ovvero non binario, il cui valore iniziale sia uguale a count.

Suggerimento: si ricordino le implementazioni "originali" delle primitive `signal`, `wait` e `init`.

English

Given the implementation of binary semaphores provided by the Pthread library with the following functions:

- `int pthread_mutex_init (pthread_mutex_t *mutex, const pthread_mutexattr_t *attr);`
- `int pthread_mutex_lock (pthread_mutex_t *mutex);`
- `int pthread_mutex_unlock (pthread_mutex_t *mutex);`

Show how it is possible to implement a counting semaphore, i.e. a non-binary semaphore, whose starting value is count.

Suggestion: remember the "original" implementation of the `signal`, `wait` and `init` primitives.

Risposta: [Answer](#):

```
typedef struct {
    int count; /* the counter */
    pthread_mutex_t lock; /* mutex ensuring exclusive access to count */
    pthread_mutex_t s; /* real semaphore */
}
```

```

} Semaphore;

static void semaphore_init (Semaphore *s, int i) {
    pthread_mutex_init (&s->lock, NULL);
    pthread_mutex_init (&s->s, NULL);
    pthread_mutex_lock(&s->s);
    s->count = i;
}

static void semaphore_wait (Semaphore *s) {
    pthread_mutex_lock (&s->lock);
    s->count--;
    if (s->count < 0){
        pthread_mutex_unlock (&s->lock);
        pthread_mutex_lock (&s->s);
    } else
        pthread_mutex_unlock (&s->lock);
}

static void semaphore_signal (Semaphore *s) {
    pthread_mutex_lock (&s->lock);
    s->count++;
    if (s->count <= 0)
        pthread_mutex_unlock (&s->s);
    pthread_mutex_unlock (&s->lock);
}

```

Ex 5 (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 Round Robin (RR), 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, which are scheduled with a temporal quantum of 10 units. Represent using a Gantt diagram the execution of these processes using the Round Robin (RR) 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	19	1
P2	6	12	3
P3	8	18	2
P4	11	14	4
P5	13	15	5

Risposta: [Answer:](#)
39 61 69 73 78 40.8

Ex 6 (2.0 points)

Italiano

Il direttorio `/home/user` contiene i file riportati di seguito, il cui elenco è stato derivato mediante il comando `ls -l`.

```
-rw-r--r-- 1 scanzio root 0 mag 17 11:31 abXcx.txt
-rw-r--r-- 1 root scanzio 0 mag 17 11:38 a.py
-rw-r--r-- 1 scanzio root 0 mag 17 11:33 axcXab.txt
-rw-r-xr-- 1 scanzio root 0 mag 17 11:36 w.py
-rw-r--r-- 1 scanzio root 0 mag 17 11:33 Xcxab.txt
-rw-r--r-- 1 scanzio root 0 mag 17 11:33 x.py
-rw-r--r-- 1 scanzio root 3464324 mag 17 11:34 y.py
drwxr-xr-x 2 scanzio root 4096 mag 17 11:35 z.py
```

Quali file sono selezionati dal comando `find` che segue:

```
find /home/user -type f \ ( -name "*.cpp" -o -name "*.py" -o -name "X?x*.txt" \)
-group root -size -1M ! -perm 654
```

English

The directory `/home/user` contains the following files, which were derived from the output of the command `ls -l`.

```
-rw-r--r-- 1 scanzio root 0 mag 17 11:31 abXcx.txt
-rw-r--r-- 1 root scanzio 0 mag 17 11:38 a.py
-rw-r--r-- 1 scanzio root 0 mag 17 11:33 axcXab.txt
-rw-r-xr-- 1 scanzio root 0 mag 17 11:36 w.py
-rw-r--r-- 1 scanzio root 0 mag 17 11:33 Xcxab.txt
-rw-r--r-- 1 scanzio root 0 mag 17 11:33 x.py
-rw-r--r-- 1 scanzio root 3464324 mag 17 11:34 y.py
drwxr-xr-x 2 scanzio root 4096 mag 17 11:35 z.py
```

What are the files selected by the `find` command that follows:

```
find /home/user -type f \ ( -name "*.cpp" -o -name "*.py" -o -name "X?x*.txt" \)
-group root -size -1M ! -perm 654
```

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

- 1. ☐ z.py
- 2. ☐ y.py
- 3. ☒ x.py
- 4. ☒ Xcxab.txt
- 5. ☐ w.py
- 6. ☐ abXcx.txt
- 7. ☐ a.py
- 8. ☐ axcXab.txt

Ex 7 (2.0 points)

Italiano

Dati i segmenti di pseudo-codice riportati a seguire ed eseguiti in concorrenza dai processi P1 e P2 di PID `pid_P1` e `pid_P2`, rispettivamente, indicare quali tra le seguenti affermazioni sono VERE.

English

Given the following segments of pseudo-code, which are executed concurrently by processes P1 and P2 with PID `pid_P1` and `pid_P2`, respectively, indicate which of the following statements are TRUE.

```
P1
while (1) {
    ...
```

```

        kill (pid_P2, SIGUSR1);
        pause ();
        A();
    }

P2
while (1) {
    pause ();
    B();
    ...
    kill (pid_P1, SIGUSR2);
}

```

Scegli una o più alternative: [Choose one or more options:](#)

- ☐ E' possibile l'esecuzione consecutiva della funzione A() per più di una volta (senza nessuna funzione B() nel mezzo). / [It is possible to execute function A\(\) more than one time \(without the execution of any B\(\) function in between\).](#)
- ☐ La funzione A() può essere eseguita prima della funzione B(). / [Function A\(\) can be executed before function B\(\).](#)
- ☒ Sono soggetti a deadlock. / [They are subject to deadlocks.](#)
- ☒ Sono soggetti a starvation. / [They are subject to starvation.](#)
- ☐ La funzione B() è sicuramente eseguita almeno una volta. / [Function B\(\) is certainly executed at least one time.](#)
- ☒ La prima invocazione della funzione B() è sempre eseguita prima della prima invocazione funzione A(). / [The first call to function B\(\) is always executed before the first call to function A\(\).](#)

Ex 8 (2.0 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 tenga 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 1 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 1 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:](#)

- ☒ Un file di dimensione 3.5 MByte NON può essere allocato utilizzando una strategia di allocazione contigua. / [A file with dimension 3.5 MByte CANNOT be allocated using a contiguous allocation strategy.](#)
- ☐ Un file di dimensione 3.5 MByte NON può essere allocato utilizzando una strategia di allocazione concatenata. / [A file with dimension 3.5 MByte CANNOT be allocated using a linked allocation strategy.](#)
- ☒ Con la strategia di allocazione contigua ottenuta mediante l'algoritmo BEST-FIT possono essere allocati nell'ordine i file F1 di 1.6 MByte, F2 di 1.9 MByte e F3 di 2.6 MByte. / [With the contiguous allocation strategy based on the BEST-FIT algorithm, the following files can be allocated in the following order: F1 of 1.6 MByte, F2 of 1.9 MByte, and F3 of 2.6 MByte.](#)

4. ☐ Con la strategia di allocazione contigua ottenuta mediante l'algoritmo FIRST-FIT possono essere allocati nell'ordine i file F1 di 1.6 MByte, F2 di 1.9 MByte e F3 di 2.6 MByte. / [With the contiguous allocation strategy based on the FIRST-FIT algorithm, the following files can be allocated in the following order: F1 of 1.6 MByte, F2 of 1.9 MByte, and F3 of 2.6 MByte.](#)

Ex 9 (1.5 points)

Italiano

Si faccia riferimento alla system call `kill()` e al comando di shell `kill`. Si indichino quali delle seguenti affermazioni sono vere. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

[Refer to the system call `kill\(\)` and to the `kill` shell command. 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:](#)

- ☐ Il segnale inviato può essere mascherato solo nel caso in cui si usi la system call. / [The sent signal can be masked only in the case of the system call.](#)
- ☒ Tanto il comando di shell quando la system call possono terminare un processo. / [Both the shell command and the system call can terminate a process.](#)
- ☐ La system call termina un processo, il comando di shell no. / [The system call terminates a process, the shell command does not.](#)
- ☒ Entrambe inviano un segnale a un processo. / [Both send a signal to a process.](#)
- ☐ Il comando di shell termina un processo, la system call no. / [The shell command terminates a process, the system call does not.](#)

Ex 10 (1.5 points)

Italiano

Si faccia riferimento all'utilizzo dei segnali in ambiente UNIX/Linux. Si indichi quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

[Refer to the use of signals in a UNIX/Linux environment. 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:](#)

- ☐ Attraverso la system call `signal()` si può decidere di ignorare QUALSIASI tipo di segnale. / [By using the system call `signal\(\)` you can decide to ignore ANY type of signal.](#)
- ☒ All'interno di un signal handler occorre usare solo funzioni rientranti. / [Inside a signal handler, only reentrant functions should be used.](#)
- ☐ La ricezione di un segnale da parte di un processo provoca SEMPRE la terminazione del processo. / [The reception of a signal by a process ALWAYS causes the termination of the process.](#)
- ☐ L'utilizzo della coppia di funzioni `kill()` e `pause()` o delle primitive semaforiche `sem_post()` e `sem_wait()` per la sincronizzazione dei processi sono EQUIVALENTI. / [The use of the pair of functions `kill\(\)` and `pause\(\)` or of the semaphore primitives `sem_post\(\)` and `sem_wait\(\)` are EQUIVALENT from the point of view of process synchronization.](#)
- ☒ La ricezione di alcuni tipi di segnali (ad esempio `SIGKILL`) non può essere ignorata. / [The reception of some types of signal \(for instance `SIGKILL`\) cannot be ignored.](#)
- ☒ L'esecuzione di un signal handler a seguito della ricezione di un segnale da parte di un processo può portare a race conditions. / [The execution of a signal handler after the reception of a signal by a process can lead to race conditions.](#)

Ex 11 (3.0 points)

Italiano

Sia dato un sistema con 4 processi (P1, P2, P3, P4) e 3 tipi di risorse (R1, R2, R3), in cui vi siano 2 istanze di risorse di tipo R1 e 1 singola istanza per le risorse di tipo R2 e R3. Lo stato del sistema è il seguente:

- Il processo P1 detiene una risorsa {R1}
- Il processo P2 detiene una risorse {R1} ed è in attesa delle risorse {R2, R3}
- Il processo P3 detiene la risorsa {R2} ed è in attesa di una risorsa {R1}
- Il processo P4 detiene la risorsa {R3} ed è in attesa della risorsa {R2}

Si effettui un'analisi della situazione dal punto di vista del deadlock basandosi ESCLUSIVAMENTE sulla presenza di cicli nel grafo di allocazione delle risorse. 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 3 types of resources (R1, R2, R3), in which the number of instances of resources of type R1 is 2, and the number of instances for each resources of type R2 and R3 is 1.

The state of the system is the following:

- Process P1 holds resource {R1}
- Process P2 holds resource {R1} and it waits resources {R2, R3}
- Process P3 holds resource {R2} and it waits resource {R1}
- Process P4 holds resource {R3} and it waits resource {R2}

Analyze the situation from the point of view of the deadlock, focusing the analysis ONLY on the presence of cycles in the resource allocation graph. 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. ☐ Il sistema SICURAMENTE NON si è in una condizione di deadlock. / The system SURELY is NOT in a deadlock condition.
2. ☒ Eliminando l'arco R2->P3 si ha la CERTEZZA di NON essere in una condizione di deadlock. / Eliminating the edge R2->P3, SURELY the system is NOT in a deadlock condition.
3. ☐ Eliminando l'arco R3->P4 si ha la CERTEZZA di NON essere in una condizione di deadlock. / Eliminating the edge R3->P4, SURELY the system is NOT in a deadlock condition.
4. ☒ Il deadlock è POSSIBILE. The deadlock is POSSIBLE.
5. ☐ Eliminando l'arco P2->R2 si ha la CERTEZZA di NON essere in una condizione di deadlock. / Eliminating the edge P2->R2, SURELY the system is NOT in a deadlock condition.
6. ☐ Il sistema SICURAMENTE è in una condizione di deadlock. / The system is SURELY in a deadlock condition.

Ex 12 (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 there are '$0' words
```

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 there are '$0' words
```

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 there are \$0 words

- 2. ☐ 4 there are there are words words
- 3. ☒ 4thereare\$0words
- 4. ☐ 4therearewords
- 5. ☐ 4therearetherearewordswords