Esame Sistemi Operativi - Operating Systems Exam
2025/06/12

Ex 01 (3.5 points)

Italiano

Si supponga che il seguente programma venga eseguito mediante l'istruzione `./pgrm 3`.
Si indichi, nel seguente ordine, quante volte vengono stampate su standard output la stringa "[3]", quante la stringa "[2]" e quante la stringa "[1]" (ovvero si riporti tre interi separati da un singolo spazio). Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Suppose the following program is executed by means of the instruction `./pgrm 3`.
Indicate, in the following order, how many times the string "[3]", the string "[2]", and the string "[1]" are printed on standard output (i.e., provide three integers separated by a single space). Note that incorrect answers imply a penalty in the final score.

```
#define N 100

int main (int argc, char *argv[]) {
  int n;
  char str[N];

  setbuf(stdout,0);
  n = atoi (argv[1]);

  if (n<=0)
    return 1;

  if (fork()) {
    if (fork()) {
      if (fork()) {
        sprintf (str, "%d", n-1);
        execlp (argv[0], argv[0], str, NULL);
      } else {
        system ("echo [1]");
      }
    } else {
       system ("echo [2]");
    }
  } else {
    system ("echo [3]");
  }

  return 1;
}
```

Risposta. Answer.

3 3 3

Ex 02 (2.0 points)

Italiano

Quali affermazioni riguardanti la strategia di allocazione di tipo indicizzata sono corrette? Si noti che le risposte errate comportano una penalità nel punteggio finale.

English

Which statements regarding the indexed allocation strategy are correct? Note that incorrect answers imply a penalty in the final score.

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

1. ⬜ Con la strategia di allocazione indicizzata, la frammentazione interna avviene solo nel blocco indice. ----- With the indexed allocation strategy, the internal fragmentation takes place only in the index block.
2. ✅ La strategia di allocazione indicizzata non soffre della frammentazione esterna. ----- The indexed allocation strategy does not suffer from external fragmentation.
3. ✅ Blocchi, sia di dati che di indici, di dimensioni ridotte consentono di ridurre lo spreco di spazio. ----- Blocks, both data and index blocks, of smaller size allow for the reduction of waste of space.
4. ⬜ Per accelerare l'accesso ai blocchi di dati si utilizzano blocchi di indici concatenati e a più livelli. ----- Linked and multi-level index blocks are used to speed up the access to data blocks.
5. ✅ Il sistema operativo UNIX/Linux combina l'uso di blocchi indicizzati concatenati e a più livelli. ----- The UNIX/Linux operating system combines both the use of the linked and multi-level index blocks.
6. ⬜ Nel sistema operativo UNIX/Linux, ogni file ha un i-node associato, ma non le directory. ----- In the UNIX/Linux operating system, each file has an associated i-node, but not the directories.

Ex 03 (3.5 points)

Italiano

Dato il seguente output del comando ls -l, indicare quali delle seguenti affermazioni sono corrette. Si noti che le risposte errate comportano una penalità nel punteggio finale.

English

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

```
total 80
drwxr-xr-x  2 matteo  staff     64 May 24 15:54 1bab.txt
-r--r---wx  1 matteo  staff      0 May 24 15:39 678bxb.txt
-rw-rw--wx  1 matteo  staff      0 May 24 15:40 678bxb543.txt
-rwxr-xr-x  1 matteo  staff  33784 May 24 18:45 a.out
-rw-r--r--  1 matteo  staff      0 May 24 15:39 bab.txt
-rw-r--r--  1 matteo  staff    839 May 24 18:45 x.c
-rw-r--r--  1 matteo  staff      0 May 24 15:40 x.txt
drwxr-xr-x  2 matteo  staff     64 May 24 15:40 y.txt
```

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

1. ✅ L'output del comando find . -type f -regex ".*b.b[^/]*\.txt" è: ----- The output of the command find . -type f -regex ".*b.b[^/]*\.txt" is:
   ./bab.txt
   ./678bxb543.txt
   ./678bxb.txt
2. ⬜ L'output del comando find . -type f -regex ".*b.b[^/]*\.txt" è: ----- The output of the command find . -type f -regex ".*b.b[^/]*\.txt" is:
   ./bab.txt
   ./678bxb.txt
3. ⬜ L'output del comando find . -type f -regex ".*b.b[^/]*\.txt" è: ----- The output of the command find . -type f -regex ".*b.b[^/]*\.txt" is:
   ./1bab.txt
   ./bab.txt
   ./678bxb543.txt
   ./678bxb.txt

4. ⬜ L'output del comando find . -type f -perm 443 è: ----- The output of the command find . -type f -perm 443 is:
   ./678bxb543.txt
5. ✅ L'output del comando find . -type f -perm 443 è: ----- The output of the command find . -type f -perm 443 is:
   ./678bxb.txt
6. ⬜ L'output del comando ls -l | egrep -v "^d" | tail -n 1 è: ----- The output of the command ls -l | egrep -v "^d" | tail -n 1 is:
   drwxr-xr-x 2 matteo staff 64 May 24 15:40 y.txt
7. ✅ L'output del comando ls -l | egrep -v "^d" | tail -n 1 è: ----- The output of the command ls -l | egrep -v "^d" | tail -n 1 is:
   -rw-r--r-- 1 matteo staff 0 May 24 15:40 x.txt

Ex 04 (2.0 points)

Italiano

Quali affermazioni riguardanti la famiglia di funzioni exec sono corrette? Si noti che le risposte errate comportano una penalità nel punteggio finale.

English

Which statements regarding the family of exec functions 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 ed execle differiscono solo per il modo in cui il primo argomento viene passato alla funzione. ----- Exec and execle differ only in the way the first argument is passed to the function.
2. ⬜ Nella funzione exec il primo argomento deve essere un percorso assoluto. ----- In the exec function the first argument must be an absolute path.
3. ✅ Le funzioni execv, execvp ed execve devono essere utilizzate se il numero di parametri del processo da sostituire non è noto al momento della compilazione. ----- The execv, execvp, and execve functions must be used if the number of parameters of the process to be replaced is not known at compilation time.
4. ✅ Il terzo argomento di una funzione execl è il primo parametro del processo da sostituire. ----- The third argument of an execl function is the first parameter of the process to be replaced.
5. ⬜ In ogni caso, il codice dopo un exec non viene mai eseguito. ----- In any case, the code after an exec is never executed.
6. ⬜ La funzione exec crea un nuovo processo con il codice di un programma specifico passato come argomento. ----- The exec function creates a new process with the code of a specific program passed as an argument.

Ex 05 (2.0 points)

Italiano

Relativamente alle system call **open()**, **read()**, **write()** e **close()**, si indichino quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Regarding the system calls **open()**, **read()**, **write(),** and **close()**, 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. ⬜ La system call **open()** accetta esattamente tre parametri. ----- The **open()** system call takes exactly three parameters.
2. ✅ Le system call **read()** e **write()** possono essere utilizzate rispettivamente per leggere e scrivere testo (caratteri). ----- The **read()** and **write()** system calls can be used to read and write text (characters), respectively.
3. ✅ La funzione **printf()** può essere implementata mediante la system call **write()**. ----- The function **printf()** can be implemented using the system call **write()**.

4. ✅ La system call **open()** accetta un numero variabile di parametri. ----- The **open()** system call takes a variable number of parameters.
5. ⬜ La system call **read()** ritorna il numero di bits letti. ----- The **read()** system call returns the number of bits read.
6. ✅ Nel caso di creazione di un file, attraverso la system call **open()** si possono impostare i permessi del file creato. ----- When creating a file, the permissions of the newly created file can be set through the **open()** system call.

Ex 06 (3.0 points)

Italiano

Si supponga che il seguente programma venga eseguito con il valore 5 sulla linea di comando. Si riporti esattamente l'output generato. Si prega di riportare la risposta su un'unica riga, indicando i vari messaggi stampati separati da un unico spazio. Non inserire nessun altro carattere nella risposta.

English

Suppose the following program is executed, passing the value 5 on the command line. Report the exact output generated. Please provide your answer on a single line, reporting all printed messages separated by a single space. Do not insert any other character in the response.

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

static void *T1 ();
static void *T2 ();

void *T1 (void *p){
  pthread_t thread;
  int *pn = (int *) p;
  int n = *pn;
  if (n>0) {
      n-=3;
      printf ("[%d]", n+1);
      pthread_create (&thread, NULL, T2, &n);
  }
  pthread_join (thread, NULL);
  pthread_exit (NULL);
}
void *T2 (void *p){
  pthread_t thread;
  int *pn = (int *) p;
  int n = *pn;
  if (n>0) {
      n+=1;
      printf ("(%d)", n);
      pthread_create (&thread, NULL, T1, &n);
  }
  pthread_join (thread, NULL);
  pthread_exit (NULL);
}
int main (int argc, char *argv[]) {
  pthread_t thread;
```

```
  int n = atoi (argv[1]);
  setbuf (stdout, 0);
  pthread_create (&thread, NULL, T1, &n);
  pthread_join (thread, NULL);
  return 1;
}
```

Risposta. Answer.
[3] (3) [1]

Ex 07 (2.0 points)

Italiano
Dato il seguente frammento di codice, indicare quali sono le affermazioni corrette. Si noti che le risposte errate comportano una penalità nel punteggio finale.
English
Given the following piece of code, indicate which statements are correct. Note that incorrect answers imply a penalty in the final score.

```
// lock globale
int lock = 0;

// Test and Set
int testAndSet (int *lock){
  int tmp;
  tmp=*lock;
  *lock=1;
  return tmp;
}

// Protocol
while (1) {
  while(testAndSet(&lock));
  CS
  lock=0;
  nonCS
}
```

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

1. ✅ Se la funzione test-and-set è implementata in hardware con la possibilità di eseguire l'operazione atomicamente, questa soluzione soffre di starvation. ----- If the test-and-set function is implemented in hardware with the possibility of performing the operation atomically, this solution suffers from starvation.
2. ⬜ Questa soluzione non può essere estesa a più di due processi. ----- This solution cannot be extended to more than two processes.
3. ⬜ Poiché si basa su una forma di attesa attiva (busy form of waiting), questa soluzione può essere utilizzata in piccoli dispositivi embedded alimentati a batteria. ----- Since it is based on busy waiting, this solution can be used in small battery-powered embedded devices.
4. ⬜ Questa soluzione, quando la test-and-set è implementata in software, è soggetta a deadlock. ----- This solution, when the test-and-set is implemented in software, it is subject to deadlock.
5. ✅ Questa soluzione, quando la test-and-set è implementata in software, non garantisce la mutua esclusione. ----- This solution, when the test-and-set is implemented in software, does not guarantee mutual exclusion.

Ex 08 (5.0 points)

Italiano

Il formato ELF (Executable and Linkable Format) è lo standard binario per i programmi, librerie, e file oggetto sui sistema operativi Unix-like. Il comando BASH:
readelf -s name.elf
legge e parsifica il file in ingresso name.elf, per generare sullo standard output il seguente:

```
Symbol table '.sym_tab' contains 1397 entries:
Num:    Value  Size Type    Bind   Vis      Ndx Name
0: 12345    2 NOTYPE  LOCAL  DEFAULT    1 .data
1: 34567    4 NOTYPE  LOCAL  DEFAULT    1 .code
2: 40000  512 NOTYPE  LOCAL  DEFAULT    1 .bss
...
```

Si scriva uno script in BASH che riceva in input quattro parametri: un percorso per un file ELF, un percorso per un file di output, due interi S e L. Lo script deve leggere il file ELF con il comando readelf (guarda sopra) e manipolare l'output per salvare tutti i simboli che hanno un indirizzo (colonna Value) maggiore di S e fine indirizzo (somma delle colonne Value + Size) sia minore di S + L (estremi inclusi):

- Value >= S;
- Value + Size <= S + L.

Il file di output deve contenere linee del seguente formato:
Name, Address/Value, Size
ordinate per valore di Address/Value crescenti.
Se nessun simbolo è trovato nell'intervallo [S, S+L], lo script deve mostrare un messaggio di errore sullo standard error senza creare il file.

English

The so-called ELF (Executable and Linkable Format) file is the standard binary format to store executable programs, shared libraries, and object files on Linux and other Unix-like systems. The BASH command:
`readelf -s name.elf`
reads an ELF file `name.elf`, and generates the following output on standard output:

```
Symbol table '.sym_tab' contains 1397 entries:
Num:    Value  Size Type    Bind   Vis      Ndx Name
0: 12345    2 NOTYPE  LOCAL  DEFAULT    1 .data
1: 34567    4 NOTYPE  LOCAL  DEFAULT    1 .code
2: 40000  512 NOTYPE  LOCAL  DEFAULT    1 .bss
...
```

Write a BASH script that receives four parameters on the command line: a path to an ELF file, a path to an output file, and two integers S and L. The script must read the ELF file with the command `readelf` (see above) and manipulate the resulting output to save into the output file all symbols that have an address (i.e., the record `Value`) greater than S and its end address (sum of columns Value + Size) less than S + L:

- Value >= S;
- Value + Size <= S + L.

The output file format must have lines sorted by `Address` and formatted in this way:
`Name, Address/Value, Size;`
If no symbols are present in the ELF file within the range [S, S+L], the script must display an error message on the standard error without creating the output file.

Soluzione. Solution.

```
#!/bin/bash

if [[ $# -lt 4 ]]; then
      echo "usage $0 elf_path out_path START END"
      exit -1
fi

elf=$1
out=$2
str=$3
len=$4

if [[ ! -f $elf ]]; then
      echo "file not found!"
      exit -2
fi

let end=$str+$len
readelf -s $elf | tr -s " " | tail -n +4 | while read line; do
      addr=$(echo $line | cut -d " " -f 2)
      size=$(echo $line | cut -d " " -f 3)
      let end_addr=$addr+$size
      if [[ $addr -ge $str ]]; then
            if [[ $end_addr -le $end ]]; then
                  name=$(echo $line | cut -d " " -f 8)
                  echo "$addr, $size, $name"
            fi
      fi
done > $out

if [[ $(cat $out | wc -l) -gt 0 ]]; then
      sort -o $out -k 2 $out
else
      echo "No symbol was found in range [$str, $end]" 1>&2
fi
```

Ex 09 (5.0 points)

Italiano

Si supponga di avere a disposizione l'implementazione Pthread dei mutex (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 e wait e il loro utilizzo di una variabile di conteggio.

English

Assume to have the Pthread implementation of mutexes (binary semaphores) available through 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's possible to implement a counting semaphore (i.e., non-binary) whose initial value is equal to count.
Hint: Recall the "original" implementations of the signal and wait primitives and their use of a counting variable.

Soluzione. Solution.

```
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 10 (2.0 points)

Italiano
Si supponga un processo diventi "zombie".
Si indichino quali delle seguenti affermazioni sono vere. Si osservi che risposte errate implicano una penalità nel punteggio finale.
English
Suppose that a process becomes a "zombie".
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. ⬜ Il PCB del processo è stato cancellato. ----- The PCB of the process has been deleted.
2. ⬜ Il processo ha eseguito una wait. ----- The process executed a wait.
3. ⬜ Il processo non ha processo padre. -----The process does not have a parent process.
4. ✅ Il processo è terminato. ----- The process has terminated.

5. ⬜ Il processo prima di terminare aveva quale processo padre il processo "init". ----- Before terminating, the process had the "init" process as the parent process.
6. ⬜ Il processo è stato adottato dal processo "init". ----- The process was adopted by the process “init”.
7. ✅ Il suo PCB verrà cancellato solo dopo che il padre effettuerà una wait o una waitpid. ----- Its PCB will be deleted only after its parent performs a wait or a waitpid.

Ex 11 (3.0 points)

Italiano

Si consideri il seguente insieme di processi schedulati con un quantum temporale di 15 unità di tempo. Rappresentare, mediante diagramma di Gantt, l'esecuzione di tali processi utilizzando l’algoritmo Round Robin (RR). Si calcolino i tempi di terminazione di ciascun processo e il tempo di attesa medio. Si riporti 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, scheduled with a temporal quantum of 15 units. Using a Gantt diagram, represent the execution of these processes using the Round Robin (RR) scheduling algorithm. Compute the termination times of each process and the average waiting time. Report your answer on a single line, indicating P1, P2, P3, P4, and P5 termination times, followed by the average waiting time. Separate all numbers with a single space. Report the average waiting time with a single decimal digit. Do not include any other characters in the response. Example of correct answer: 20 23 11 45 67 30.5

Processo Process	TempoArrivo ArrivalTime	BurstTime	Priorità Priority
P1	0	13	4
P2	3	15	3
P3	6	12	5
P4	9	22	2
P5	12	26	1

Risposta. Answer.

13 28 40 77 88 25.6

Ex 12 (3.0 points)

Italiano

Lo stato di un sistema con 5 processi e 3 tipologie di risorse è definito dalle matrici riportate alla fine della domanda. Si supponga il processo P1 effettui una richiesta per le risorse {1, 1, 1}. Si stabilisca se la richiesta può essere soddisfatta permettendo al sistema di rimanere in uno stato sicuro. In caso affermativo si risponda “YES” e si riporti la sequenza di esecuzione dei processi. Ad esempio, si risponda “YES 5 4 3 2 1” nel caso in cui la sequenza sicura sia P5, P4, P3, P2, P1. In caso contrario, si risponda “NO”.

English

The state of a system with 5 processes and 3 types of resources is defined by the matrices provided at the end of the question. Suppose that process P1 requests the resources {1, 1, 1}. Indicate whether or not the request can be granted, allowing the system to remain in a safe state. In the affirmative case, respond with “YES” and report the sequence in which the processes will be executed. For instance, respond “YES 5 4 3 2 1” in the case the safe sequence is P5, P4, P3, P2, P1. Otherwise, respond with “NO”.

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

P4	0 1 0	3 2 2	
P5	0 0 0	4 3 5	

Fasi operative. Workflow.

```
Check Request (process 1 - row 0)
Richiesta soddisfacibile; nuovo stato
  nProc  5
  nRis  3
  Processo Fine Assegnate Massimo Necessita' Disponibili
  |  1  |  0  |  2  1  2  |  2  5  4  |  0  4  2  |  3  2  2  |
  |  2  |  0  |  0  0  1  |  4  4  2  |  4  4  1  |
  |  3  |  0  |  0  1  0  |  3  4  2  |  3  3  2  |
  |  4  |  0  |  0  1  0  |  3  2  2  |  3  1  2  |
  |  5  |  0  |  0  0  0  |  4  3  5  |  4  3  5  |
  RequestByProcP  1 (row  0)
  RequestOf  1  1  1
------------------------------------------------------------
Soluzione Iterativa (una soluzione)
  Step  0;                                                Disponibili:  3  2  2
  Step:  1   RunProc:  4   Necessita':  3  1  2    -->    Disponibili:  3  3  2
  Step:  2   RunProc:  3   Necessita':  3  3  2    -->    Disponibili:  3  4  2
  Step:  3   RunProc:  1   Necessita':  0  4  2    -->    Disponibili:  5  5  4
  Step:  4   RunProc:  2   Necessita':  4  4  1    -->    Disponibili:  5  5  5
  Step:  5   RunProc:  5   Necessita':  4  3  5    -->    Disponibili:  5  5  5
    Stato sicuro
```

Risposta. Answer.

YES 4 3 1 2 5