System and device programming

Iniziato	lunedì, 12 settembre 2022, 09:43
Stato	Completato
Terminato	lunedì, 12 settembre 2022, 09:43
Tempo impiegato	13 secondi
Valutazione	**0,00** su un massimo di 15,00 (**0**%)

Domanda 1

Risposta non data

Punteggio max.: 1,00

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

```
typedef struct cond_s {
   pthread_mutex_t lock;
   pthread_cond_t cond;
   int count;
   int flag;
} cond_t;

static void *TA (void *args) {
  cond_t *cond_d = (cond_t *) args;
  while (1) {
     pthread_mutex_lock (&cond_d->lock);
     cond_d->count++;
     printf ("%d ", cond_d->count);
     if (cond_d->count >= 10) {
        cond_d->flag = 1;
        pthread_cond_signal (&cond_d->cond);
        pthread_mutex_unlock (&cond_d->lock);
        break;
     }
     pthread_mutex_unlock (&cond_d->lock);
  }
  pthread_exit(0);
}

static void *TB (void *args) {
   cond_t *cond_d = (cond_t *) args;

   pthread_mutex_lock (&cond_d->lock);
   while (cond_d->flag == 0) {
      pthread_cond_wait (&cond_d->cond, &cond_d->lock);
      printf ("[%d]\n", cond_d->count);
   }
   pthread_mutex_unlock (&cond_d->lock);
   pthread_exit(0);
}

int main () {
  cond_t cond_d;
  pthread_t tid1, tid2;

  setbuf (stdout, 0);

  pthread_mutex_init (&cond_d.lock, NULL);
  pthread_cond_init (&cond_d.cond, NULL);
  cond_d.count = 0;
  cond_d.flag = 0;

  pthread_create (&tid1, NULL, TA, (void *) &cond_d);
  pthread_create (&tid2, NULL, TB, (void *) &cond_d);

  pthread_join (tid1, NULL);
  pthread_join (tid2, NULL);

  pthread_exit(0);
}
```

Scegli una o più alternative:

- [ ] (a) 1 2 3 4 5 6 7 8 9 [10]
- [ ] (b) 1 2 3 4 5 6 7 8 9
- [ ] (c) [10] 1 2 3 4 5 6 7 8 9 10
- [ ] (d) 1 2 3 4 5 6 7 8 9 10

☐ (e) 1 2 3 4 5 6 7 8 9 10 [10]

☐ (f) 0 1 2 3 4 5 6 7 8 9 [10]

Risposta errata.

La risposta corretta è: 1 2 3 4 5 6 7 8 9 10 [10]
, 1 2 3 4 5 6 7 8 9 10

Domanda 2

Risposta non data

Punteggio max.: 1,50

In which line(s) of the main, copy constructor is called? Why?

```
#include <iostream>
using namespace std;

class Y {

public: //the five copy-control members
   //constructors
   Y()  { std::cout << "dc " << std::endl; } //default constructor dc
   Y(const Y &) {  std::cout << "cc" << std::endl; } //copy constructor cc
   Y(Y &&) noexcept { std::cout << "mc" << std::endl; }; //move constructor mc
   //assignments
   Y &operator=(const Y &) { std::cout << "ca" << std::endl; } //copy assignment ca
   Y &operator=(Y &&) {std::cout << "ma" << std::endl;} //move assignment ma
   //destructor
   ~Y() { std::cout << "d" << std::endl; } //destructor d
};

Y f_a(Y y) { Y new_y = move(y); return new_y; };
Y f_b(Y& y) { Y new_y = move(y); return new_y;};

int main() {
    Y y0, y1; //LINE 1
    Y y2 = f_a(y0); //LINE 2
    Y y3 = f_b(y1); //LINE 3
    return 0;  //LINE 4
}
```

Domanda 3

Risposta non data

Non valutata

If you want to withdraw from this part (Quer/Vetrò) of the exam, please select true/vero/yes. Otherwise, i.e., you want to take the exam, select false/falso/no.

Notice: It is also possible to withdraw once the exam has been completed, sending an e-mail to the instructors.

Scegli una risposta:

- ○ Vero
- ○ Falso

La risposta corretta è 'Falso'.

Domanda 4

Risposta non data

Punteggio max.: 1,00

Analyze the following segment of code. Indicate which are the possible outputs generated by the program when 5 is passed as an argument. Note that more than one response can indeed be correct and that incorrect answers may imply a penalty on the final score.

```
#define N 100

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

  n = atoi (argv[1]);
  setbuf(stdout,0);
  if (n>0 && !fork()) {
    if (!fork()) {
      fprintf (stdout, "E");
      sprintf (str, "%d", n-1);
      execlp (argv[0], argv[0], str, NULL);
    } else {
      sprintf (str, "echo -n S");
      system (str);
    }
  }
  return 1;
}
```

Scegli una o più alternative:

- [ ] (a) SESESESESE
- [ ] (b) ESESESESES
- [ ] (c) SSSEEESESE
- [ ] (d) SE
- [ ] (e) SESESESESESESESESESE
- [ ] (f) ES

Risposta errata.

La risposta corretta è: SESESESESE, ESESESESES, [illegible]

Domanda 5

Risposta non data

Punteggio max.: 2,50

A program can execute four different threads, namely TP (thread plus), TM (thread minus), TS (thread star), and TNL (thread newline).
Each thread is organized through an **infinite cycle** containing synchronization instructions but a **single I/O instruction**:

- Thread TP includes instruction printf ("+")
- Thread TM includes instruction printf ("-")
- Thread TS includes instruction printf ("*")
- Thread TNL includes instruction printf ("\n").

Synchronize the four threads to print the following sequence of lines:

```
++++++++++
----------
**********
++++++++++
----------
**********
etc.
```

where the number of characters on each row is equal to a constant value N (that is, 10 in the previous example).

The students who followed the course before the academic year 2021-2022 can write the C code adopting the UNIX or the Windows API notation.

Domanda 6

Risposta non data

Punteggio max.: 3,50

Cheat Sheet UNIX.

A process P orchestrates the following operations:

- It receives real values from two different FIFOs. The first FIFO connects P with P1, and the second connects P with P2. P1 and P2 send real values to P at different speed.
- It uses I/O multiplexing to read the data from the two FIFOs.
- It stores all real values it receives in a memory-mapped file. The process P terminates when it receives a value equal to zero from both P1 and P2 (even if the two zeros are not contiguous).

Implement the process P in C language, whereas the implementation of P1 and P2 is not required.

Domanda 7

Risposta non data

Punteggio max.: 2,50

Write a C++ program for the parallelization of a high number of computations (100).

Each computation depends on two parameters, namely *i* and *p*:

- *i* assume values 1,2,…100 (e.g., you could iterate in a for cycle);
- p is set by a separate thread that takes a number from the console, whenever the user introduces it.

Each computation adds to the value of *i* its %*p*. For example, if the user inputs 10:

- for i=1, it is added 0,1 and i becomes 1,1 ;
- for i=2, it is added 0,2 and i becomes 2,2 ;
- for i=3, it is added 0,3 and i becomes 3,3 ;
- …

The parameter *p* must be between 1 and 100 (both included): computations are not done until the value of *p* satisfies this requirement.
The final goal of the program is to compute the sum of all the updated values of i (from 1 to 100).
For the computation tasks, you are free to use either threads or a structure of advanced parallelism at your choice (e.g., promises, asynchronous tasks, or packaged tasks).

If you do not remember the exact syntax of C++ synchronization primitives, you can write down a mock version (with the same meaning). Correctness is strictly required in the template syntax which must be right, as well as any basic C++ syntax.

Domanda 8

Risposta non data

Punteggio max.: 2,00

Cheat Sheet UNIX.

Describe how it is possible to implement the function **pthread_barrier_wait** with semaphores, mutexes, and counters. Explain why it is necessary to discriminate between an acyclic and a cyclic behavior and report the entire C code segment to describe the two cases.
Indicate the logic difference (and report the code) between a standard barrier and a turnstile barrier.

The students who followed the course before the academic year 2021-2022 can write the C code adopting the UNIX or the Windows API notation.

Domanda 9

Risposta non data

Punteggio max.: 1,00

What is the final value of a1->a ?

Note that a wrong answer might imply a negative score

```
#include <iostream>
using namespace std;
class A {
   private:
      int a = 0;
      friend class B;
   public:
     int get_a( ){return this->a;};
     void set_a( int value){ this->a = value;};
};
class B {
private:
   int b=5;
public:
   int get_b(){return this->b;}
   void set_b(int value){this->b = value;}
   void set_a(A& obj_a, int value){obj_a.a = value;}
};
int main(){
   A a1 ;
   B b1 ;
   a1.set_a(3);
   b1.set_a(a1, b1.get_b());
   b1.set_b(1);
}
```

- 3
- 0
- 1
- there is run time error, because field a of A is private and it is accessed directly in function B.set_a;
- 5
- undefined behaviour;
- there is a compilation error, because field a of A is private and it is accessed directly in function B.set_a;

Risposta errata.

La risposta corretta è: 5