Esame Sistemi Operativi - Operating Systems Exam
2025/02/17

Ex 01 (2.5 points)

Italiano

Quali affermazioni riguardanti i file di testo, ASCII, Unicode e binari sono corrette? Si noti che le risposte errate comportano una penalità nel punteggio finale.

English

Which statements regarding text, ASCII, Unicode, and binary files are correct? Note that incorrect answers imply a penalty in the final score.

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

1. ✅ I file di testo, come i file binari, sono sequenze di byte. La codifica dei file testuali è gestita da standard specifici, mentre per i file binari, nel caso della serializzazione, potrebbe non essere definito uno standard. ----- Text files like binary files are sequences of bytes. The encoding of textual files is managed by specific standards while, for binary files, in the case of serialization, a standard might not be defined.
2. ⬜ La codifica ASCII supporta tutti i caratteri delle lingue moderne ed è quindi lo standard di codifica del testo più universale. ----- ASCII coding supports all characters in modern languages, making it the most universal text encoding standard.
3. ✅ La codifica Unicode include ASCII come sottoinsieme. ----- Unicode encoding includes ASCII as a subset.
4. ⬜ Un file memorizzato nella codifica UTF-16 è sempre compatibile con i programmi che si aspettano un testo codificato in ASCII. ----- A file stored in UTF-16 encoding is always compatible with programs expecting ASCII-encoded text.
5. ✅ I file binari possono contenere testo leggibile ma anche dati grezzi come immagini, codice eseguibile e informazioni compresse. ----- Binary files can contain readable text but they can also include raw data such as images, executable code, and compressed information.
6. ✅ UTF-8 è una codifica a lunghezza variabile in grado di memorizzare in modo efficiente tutti i caratteri Unicode mantenendo la compatibilità con ASCII. ----- UTF-8 is a variable-length encoding that can store all Unicode characters efficiently while maintaining compatibility with ASCII.
7. ⬜ L'ASCII è uno schema di codifica a lunghezza variabile, che rende difficile l'analisi corretta del testo. ----- ASCII is a variable-length encoding scheme, making it difficult to parse text correctly.
8. ⬜ Unicode e UTF-8 sono la stessa cosa. ----- Unicode and UTF-8 are the same thing.
9. ⬜ I file binari sono utilizzati solo per memorizzare immagini e file multimediali, mentre i file di testo sono utilizzati per tutto il resto. ----- Binary files are only used for storing images and multimedia, while text files are used for everything else.
10. ✅ Se un file di testo codificato in UTF-8 viene letto come ASCII, alcuni caratteri possono apparire confusi o sostituiti con punti interrogativi (o altri caratteri). ----- If a UTF-8 encoded text file is read as ASCII, some characters may appear garbled or replaced with question marks (or other characters).

Ex 02 (3.5 points)

Italiano

Si supponga che il seguente programma venga eseguito mediante l'istruzione `./pgrm 3`.

Si indichi quante volte viene visualizzata la stringa "[3]", quante la stringa "[2]" e quante la stringa "[1]" in quest'ordine (ovvero si riportino tre interi separati da un singolo spazio dove il primo intero rappresenta il numero di stringhe "[3]"). Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Suppose the following program is executed with the command `./pgrm 3`.

Indicate how many times the string “[3]”, the string “[2]”, and the string “[1]” are displayed, in this order (i.e., provide three integer numbers separated by a single space where the first integer represents the number of strings “[3]”). Note that incorrect answers imply a penalty in the final score.

```
int main (int argc, char *argv[]) {
  int n;
  char str[100];

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

  fprintf (stdout, "[%d]", n);

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

  return 1;
}
```

Risposta. Answer.
1 2 5

Ex 03 (2.0 points)

Italiano
Si consideri il tratto di codice riportato successivamente. Si indichino quali delle seguenti affermazioni sono vere. Si osservi che risposte errate implicano una penalità nel punteggio finale.
English
Consider the following code. Indicate which of the following statements are true. Note that incorrect answers imply a penalty in the final score.

```
signal (SIGCHLD, S);
if (fork() == 0) {
  i = 2;
  printf ("i=%d PID=%d\n", i, getpid());
  exit (i);
} else {
  pid = wait (&code);
  printf ("Wait: ret=%d code=%x\n", pid, code);
}
```

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

1. ✅ Se S è SIG_IGN si disattiva la gestione del segnale SIGCHLD tramite la wait() da parte del padre. ----- If S is SIG_IGN, the management of the SIGCHLD signal by means of the wait() in the parent is disabled.
2. ✅ Se S è SIG_IGN il figlio non diventerebbe zombie anche se terminasse prima che il padre facesse la wait(). ----- If S is SIG_IGN, the child would not become a zombie even if it would be terminated before the wait() in the parent.
3. ⬜ La variabile 'code' all'interno della printf() vale 2 quando la printf() viene eseguita. ----- The variable 'code' within printf() is equal to 2 at the time of execution of the printf().
4. ✅ Se S è il nome di una funzione istanziata per gestire il segnale, il segnale verrà gestito in maniera asincrona e la wait() non è necessaria. ----- If S is the name of a function instantiated to handle the signal, the signal will be handled asynchronously, and wait() is not required.
5. ⬜ Se S è SIG_IGN la variabile pid all'interno della printf() sarà il PID del processo figlio. ----- If S is SIG_IGN, the variable pid within the printf() function will be the PID of the child process.
6. ✅ Se S è SIG_DFL il valore restituito dalla funzione getpid() e la variabile pid all'interno della printf() hanno lo stesso valore. ----- If S is SIG_DFL, the value returned by the getpid() function and the pid variable within printf() have the same value.

Ex 04 (2.0 points)

Italiano

Si consideri il seguente output del comando ls -l in Linux. Si indichino quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Consider the following output of the ls -l command in Linux. Indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

```
-rw-r--r--  1 user1 user1 1234 Feb  8 12:34 file1.txt
drwxr-xr-x  2 user2 user2 4096 Feb  8 12:35 dir1
-rwxr-xr--  1 user3 user3 9876 Feb  8 12:36 script.sh
```

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

1. ✅ Il file file1.txt è leggibile da tutti gli utenti del sistema. ----- The file file1.txt is readable by all users of the system.
2. ⬜ La directory dir1 non può essere aperta da utenti diversi da user2. ----- The directory dir1 cannot be accessed by users other than user2.
3. ⬜ La directory dir1 è eseguibile da tutti gli utenti. ----- The directory dir1 is executable by all users.
4. ✅ Il numero 1234 indica la dimensione del file file1.txt in byte. ----- The number 1234 represents the size of the file file1.txt in bytes.
5. ⬜ Il file script.sh non è eseguibile da nessuno. ----- The file script.sh cannot be executed by any user.
6. ✅ Il file file1.txt non è modificabile dagli utenti diversi da user1. ----- The file file1.txt cannot be modified by users other than user1.
7. ✅ Il proprietario di script.sh può modificarlo ed eseguirlo. ----- The owner of script.sh can modify and execute it.
8. ⬜ Il numero 1 vicino a file1.txt rappresenta il numero di link simbolici che puntano a quello specifico file. ----- The number 1 near to file1.txt represents the number of symbolic links pointing to that specific file.

Ex 05 (3.5 points)

Italiano

Si scriva uno script BASH che riceve il nome di una directory e un prefisso come parametri. Esso dovrà unire tutti i file di testo (con estensione ".txt") presenti nella directory che hanno il prefisso inserito da linea di comando nel nome del file (prima del primo underscore "_") in un singolo file il cui nome è dato dal prefisso, seguito da "_merged.txt".

esempio: prefisso = hello

fa match con i file:

- *hello_hello.txt*
- *hello_bye_bye.txt*

***NON** fa match con i file:*

- *bye_hello.txt*
- *_hello.txt*
- *hellobye_bye.txt*

English

Write a BASH script that receives the name of a directory and a prefix as parameters. It should concatenate all the text files (with the extension '.txt') present in the directory that have the prefix passed from the command line in their filename (before the first underscore '_') into a single file, whose name is the prefix followed by '_merged.txt'.

example: prefix = hello

matches with files:

- *hello_hello.txt*
- *hello_bye_bye.txt*

*does **NOT** match with files:*

- *bye_hello.txt*
- *_hello.txt*
- *hellobye_bye.txt*

Soluzione. Solution.

```
#!/bin/bash

if [[ $# -lt 2 ]]; then
        echo "usage $0 <dir> <prefix>"
        exit -1
fi

dir="$1"
pre="$2"
out="${pre}_merged.txt"

for file in $(find $dir -maxdepth 1 -type f -name "${pre}_*.txt"); do
        cat $file >> $out
done
```

Ex 06 (2.5 points)

Italiano

Considerando la gestione dei thread e la relazione tra thread utente e thread kernel, quale delle seguenti affermazioni sono corrette? Nota che le risposte errate comportano una penalità nel punteggio finale.

English

Consider thread management and the relationship between user and kernel threads. 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. ✅ Un processo con cinque kernel thread può avere dieci thread utente mappati su di essi. ----- A process with five kernel threads can have ten user threads mapped onto them.
2. ⬜ È sempre meglio usare i kernel thread invece di quelli utente per il multithreading. ----- It is always better to use kernel threads instead of user threads for multithreading.
3. ⬜ I meccanismi di sincronizzazione dei thread, come i mutex e i semafori, sono necessari solo per i kernel thread. ----- Thread synchronization mechanisms such as mutexes and semaphores are only required for kernel threads.
4. ✅ Un programma single-threaded in user-space può essere eseguito su un processore multi-core senza modifiche al suo codice volte a prevenire le race conditions. ----- A single-threaded user-space

program can run on a multi-core processor without modifications on its code aimed at preventing race conditions.

5. ⬜ Se un sistema consente solo i thread utente (senza i thread del kernel), essi possono essere eseguiti parallelamente (cioè simultaneamente) su una CPU multicore. ----- If a system allows only user threads (without kernel threads), they can still be executed in a parallel way (i.e., simultaneously) on a multi-core CPU.
6. ⬜ I programmi multi-thread che utilizzano i kernel thread vengono sempre eseguiti più velocemente dei programmi single-thread. ----- Multi-threaded programs using kernel threads always run faster than single-threaded programs.
7. ✅ I thread utente hanno generalmente un overhead di cambio di contesto inferiore a quello dei kernel thread. ----- User threads generally have typically lower context-switching overhead than kernel threads.

Ex 07 (6.0 points)

Italiano

Si scriva un programma in C che utilizzi la libreria pthread per creare due thread, TA e TB.
Il thread TA genera un numero casuale uguale a 1 o 2; nel primo caso, esegue una istanza di TB; nel secondo caso, esegue due istanze di TB.
Il thread TB fa la stessa cosa, ma esegue 1 o 2 istanze di TA.
Il programma deve fermarsi quando sono state create (globalmente) 10 istanze del thread TA o 10 istanze del thread TB. Per estrarre un numero casuale si può usare la funzione che segue.

English

Write a program in C using the pthread library that creates two threads TA and TB.
Thread TA generates a random number equal to 1 or 2; in the first case, it runs one instance of TB; in the second one, it runs two instances of TB.
Thread TB does the same but runs 1 or 2 instances of TA.
The program must stop when either 10 instances of thread TA or 10 instances of TB have been created (globally). To extract a random number, you can use the following function.

```
// Random number generator helper function
int generate_random(int lower, int upper) {
    return (rand() % (upper - lower + 1)) + lower;
}
```

Soluzione. Solution.

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

#define MAX_INSTANCES 10

// Shared counters and mutex
int ta_count = 0;
int tb_count = 0;
pthread_mutex_t lock;

// Function declarations
void *thread_TA(void *arg);
void *thread_TB(void *arg);

// Random number generator helper function
int generate_random(int lower, int upper) {
    return (rand() % (upper - lower + 1)) + lower;
}

int main() {
    // Initialize the mutex
```

```
    pthread_mutex_init(&lock, NULL);

    srand(time(NULL)); // Seed for random number generation

    // Start with the first instance of thread TA
    pthread_t initial_ta;
    pthread_create(&initial_ta, NULL, thread_TA, NULL);
    pthread_join(initial_ta, NULL);

    // Destroy the mutex
    pthread_mutex_destroy(&lock);

    printf("Program finished: TA instances = %d, TB instances = %d\n", ta_count,
tb_count);

    return 0;
}

void *thread_TA(void *arg) {
    pthread_mutex_lock(&lock);
    if (ta_count >= MAX_INSTANCES || tb_count >= MAX_INSTANCES) {
        pthread_mutex_unlock(&lock);
        return NULL;
    }
    ta_count++;
    printf("TA instance %d started.\n", ta_count);
    pthread_mutex_unlock(&lock);

    // Generate 1 or 2 TB threads
    int tb_instances = generate_random(1, 2);
    pthread_t tb_threads[tb_instances];
    for (int i = 0; i < tb_instances; i++) {
        pthread_create(&tb_threads[i], NULL, thread_TB, NULL);
    }

    for (int i = 0; i < tb_instances; i++) {
        pthread_join(tb_threads[i], NULL);
    }
    return NULL;
}

void *thread_TB(void *arg) {
    pthread_mutex_lock(&lock);
    if (tb_count >= MAX_INSTANCES || ta_count >= MAX_INSTANCES) {
        pthread_mutex_unlock(&lock);
        return NULL;
    }
    tb_count++;
    printf("TB instance %d started.\n", tb_count);
    pthread_mutex_unlock(&lock);

    // Generate 1 or 2 TA threads
    int ta_instances = generate_random(1, 2);
    pthread_t ta_threads[ta_instances];
    for (int i = 0; i < ta_instances; i++) {
        pthread_create(&ta_threads[i], NULL, thread_TA, NULL);
```

```
    }

    for (int i = 0; i < ta_instances; i++) {
        pthread_join(ta_threads[i], NULL);
    }

    return NULL;
}
```

Ex 08 (2.5 points)

Italiano

Si consideri il problema della sezione critica e la sua risoluzione attraverso il tratto di codice riportato successivamente. Si indichino quali delle seguenti affermazioni sono vere. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Consider the critical section problem and its resolution through the code provided below. Indicate which of the following statements are true. Note that incorrect answers imply a penalty in the final score.

```
char lock = FALSE;

char TestAndSet (char *lock) {
  char val;
  val = *lock;
  // LINE 1
  // LINE 2
}

while (TRUE) {
   while (TestAndSet (&lock));
   SC
   lock = FALSE;
   sezione non critica
}
```

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

1. ⬜ Al posto del commento 'LINE 1' deve essere messo: `lock = TRUE;` ----- The comment with 'LINE 1' must be changed to: `lock = TRUE;`
2. ✅ Al posto del commento 'LINE 2' deve essere messo: `return val;` ----- The comment with 'LINE2' must be changed to: `return val;`
3. ✅ La funzione TestAndSet deve essere implementata in maniera atomica, ovvero viene eseguita in modo indivisibile senza interruzioni da altri processi. ----- The function TestAndSet must be implemented atomically, meaning it has to be indivisible in its execution without interruptions from other processes.
4. ⬜ I processi in attesa sono in "attesa attiva"; questo costituisce un metodo efficiente per gestire le sezioni critiche in termini di utilizzo della CPU. ----- The waiting processes are in 'busy waiting'; which it is an efficient method to handle critical sections in terms of CPU usage.
5. ✅ L'implementazione di TestAndSet deve essere supportata a livello hardware. ----- The implementation of TestAndSet must be supported at hardware level.
6. ⬜ L'uso di test-and-set è preferibile rispetto ai semafori in termini di prestazioni e semplicità di implementazione su sistemi multi-core. ----- The use of test-and-set is preferable to semaphores in terms of performance and ease of implementation on multi-core systems.
7. ✅ Per evitare la starvation, si potrebbero introdurre meccanismi di fairness (come code di attesa) nella gestione del lock, ma ciò complicherebbe l'implementazione di TestAndSet. ----- To avoid starvation,

one could introduce fairness mechanisms (such as wait queues) in the lock management, but this would complicate the implementation of TestAndSet.

Ex 09 (3.5 points)

Italiano

Un programma definisce 4 semafori (sa, sc, sb, and sac) e una variabile globale n, e li inizializza nel modo seguente:

```
n = 0;
sem_init (sa,0,1);
sem_init (sc,0,1);
sem_init (sac,0,1);
sem_init (sb,0,0);
```

Poi esegue 3 thread TA, TB e TC, con il seguente codice (vedi fine domanda).
Specificare quali possono essere i possibili ordini di esecuzione (anche più di uno) tra le seguenti possibilità.Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

A program defines 4 semaphores (sa, sc, sb, and sac) and a global variable n, and it initializes them as follows:

```
n = 0;
sem_init (sa,0,1);
sem_init (sc,0,1);
sem_init (sac,0,1);
sem_init (sb,0,0);
```

Then it runs 3 threads TA, TB and TC, with the following code. Specify which are the possible execution orders (possibly more than one) among the following ones. Note that incorrect answers imply a penalty in the final score.

```
static void *TA () {
  sem_wait (sa);
  sem_wait (sac);
  printf ( "X ");
  n++;
  if (n==1) {
    sem_post (sb);
  } else {
    n=0;
    sem_post(sa);
    sem_post(sc);
    sem_post(sac);
  }
  pthread_exit();
}
```

```
static void *TC () {
  sem_wait(sc);
  sem_wait(sac);
  printf ( "Z ");
  n++;
  if(n == 1) {
    sem_post(sb);
  } else {
    n=0;
    sem_post(sa);
    sem_post(sc);
    sem_post(sac);
  }
  pthread_exit();
}
```

```
static void *TB () {
  sem_wait(sb);
  printf ("Y ");
  sem_post(sac);
}
```

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

1. ✅ X Y Z
2. ⬜ Y X Z
3. ⬜ Z X Y
4. ✅ Z Y X
5. ⬜ Y Z X
6. ⬜ X Z Y

Ex 10 (2.0 points)

Italiano

Si indichino quali sono le caratteristiche dell'implementazione del paradigma produttore/consumatore riportato alla fine della domanda, ed eseguito concorrentemente da più produttori e consumatori. Segnare tutte le affermazioni vere. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Indicate the characteristics of implementing the producer/consumer paradigm reported at the end of the question and executed concurrently by many producers and consumers. Mark all the true answers. Note that incorrect answers imply a penalty in the final score.

```
Producer () {
	int val;
	while (TRUE) {
		produce (&val);
		wait (empty);
		enqueue (val);
		signal (full);
	}
}

Consumer () {
	int val;
	while (TRUE) {
		wait (full);
		wait (mutex);
		dequeue (&val);
		signal (mutex);
		signal (empty);
		consume (val);
	}
}
```

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

1. ⬜ L'implementazione è corretta nel caso di più produttori e più consumatori. ----- Considering the case of multiple producers and consumers, the implementation is correct.
2. ✅ Nel caso diano presenti più produttori, l'implementazione è sicuramente errata in quanto anche nella funzione Producer la funzione enqueue va protetta dal mutex. ----- With more than one Proucer, the implementation is surely wrong because in the function Producer, the enqueue function must be protected by the mutex.
3. ✅ Il mutex utilizzato nella funzione Consumer limita la concorrenza dei consumatori relativamente all'esecuzione della funzione dequeue. ----- The mutex used in the function Consumer limits the concurrency of the consumers regarding the execution of the function dequeue.
4. ⬜ La soluzione è soggetta a deadlock. ----- The solution is subject to deadlock.
5. ⬜ Il semaforo full serve per evitare l'inserimento di dati in un buffer pieno. ----- The full semaphore is used to prevent the insertion of data into a full buffer.
6. ✅ Il semaforo empty va inizialilzzato alla dimensione della coda. ----- The empty sempahore must be initialized at the size of the queue.
7. ⬜ Il semaforo full va inizialilzzato alla dimensione della coda. ----- The full sempahore must be initialized at the size of the queue.

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 Priority Scheduling (PS). 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, P5 e P6 seguiti dal tempo di attesa medio. Separare numeri e stringhe 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 82 30.5

English

Consider the following processes, scheduled with a temporal quantum of 15 units. Using a Gantt diagram, represent the execution of these processes using the Priority Scheduling (PS) scheduling algorithms. Compute the termination time of each process and the average waiting time. Report your answer on a single line, indicating P1, P2, P3, P4, P5, and P6 termination times, 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 82 30.5

Processo Process	TempoArrivo ArrivalTime	BurstTime BurstTime	Priorità Priority
P1	0	13	5
P2	0	17	4
P3	10	14	1
P4	10	10	6
P5	20	9	3
P6	20	12	2

Fasi operative. Workflow.

```
PS:
  P2: from=  0 to  = 16 end = 17 waitingTime =  0 tournaroundTime = 17
  P3: from= 17 to  = 30 end = 31 waitingTime =  7 tournaroundTime = 31
  P6: from= 31 to  = 42 end = 43 waitingTime = 11 tournaroundTime = 33
  P5: from= 43 to  = 51 end = 52 waitingTime = 23 tournaroundTime = 42
  P1: from= 52 to  = 64 end = 65 waitingTime = 52 tournaroundTime = 45
  P4: from= 65 to  = 74 end = 75 waitingTime = 55 tournaroundTime = 55
  EndTime: 17 31 43 52 65 75
  AverageWaitingTime: (  0 +  7 + 11 + 23 + 52 + 55 ) / 6 = 24.67
  AverageTournaroudTime: ( 17 + 31 + 33 + 42 + 45 + 55 ) / 6 = 74.33
GanttChart:
222222222222222233333333333333666666666666555555551111111111114444444444
```

Risposta. Answer.

```
65 17 31 75 52 43 24.6
```

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 P3 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 reported at the end of the question. Suppose that process P3 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	0 0 1	4 4 2	3 2 3

P2	0 2 0	3 4 2	
P3	0 1 0	3 2 2	
P4	0 0 0	4 3 5	
P5	1 0 1	2 5 4	

Fasi operative. Workflow.

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

Risposta. Answer.

YES 3 2 5 1 4