

Esame Sistemi Operativi - Operating Systems Exam

2024/09/18

Ex 01 (2.0 points)

Italiano

Si consideri un file memorizzato utilizzando tre codifiche diverse: ASCII, UNICODE (es., UCS, UTF,...) e binaria. Si indichi quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Consider a file stored using three different coding systems: ASCII, UNICODE (e.g., UCS, UTF,...), and binary. 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.

- ☒ Ci sono diverse versioni delle tabelle ASCII per tenere in considerazione le diverse lingue e opzioni. *There are several ASCII table versions to keep track of the different languages and options.*
- ☐ Gli encoding ASCII e Unicode sono sempre diversi. *The ASCII and Unicode encodings always differ.*
- ☐ I caratteri Unicode sono sempre memorizzati su 32 bit. *Unicode characters are always stored on 32 bits.*
- ☒ Unicode è un superset di ASCII, il che significa che tutti i caratteri ASCII sono inclusi in Unicode. *Unicode is a superset of ASCII, meaning all ASCII characters are included in Unicode.*
- ☐ I file binari sono sempre comprensibili senza software specializzato. *Binary files are always human-readable without specialized software.*
- ☒ Modificare un file binario con un editor di testo può corrompere il file. *Editing a binary file with a text editor can corrupt the file.*
- ☐ I caratteri Unicode possono tutti essere rappresentati utilizzando solo due byte. *All Unicode characters can be represented using just two bytes.*
- ☒ Un file di testo salvato in Unicode può essere più grande in termini di dimensioni rispetto allo stesso testo salvato in ASCII. *A text file saved in Unicode may be larger in size than the same text saved in ASCII.*
- ☒ Se memorizziamo solo valori (interi o reali) in un file, il formato binario può essere più esteso del formato ASCII. *If we store only (integer or real) numbers in a file, a binary format can be larger than an ASCII format.*
- ☒ La codifica binaria comprende file che memorizzano dati in formati binari, inclusi immagini, video e programmi eseguibili. *Binary encoding encompasses files that store data in binary formats, including images, videos, and executable programs.*

Ex 02 (2.0 points)

Italiano

Quali delle seguenti affermazioni sulle system call `fork()` sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Which statements about the `fork()` system call are correct? Note that incorrect answers imply a penalty in the final score.

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

- ☒ `fork()` restituisce il PID del figlio nel processo padre, zero nel processo figlio, e -1 in caso di errore. *`fork()` returns the PID of the child in the parent process, zero in the child process, and -1 in case of error.*
- ☒ `fork()` crea un nuovo processo duplicando il processo chiamante. Al nuovo processo è assegnato un nuovo PID. *`fork()` creates a new process by duplicating the calling process. To the new process is assigned a new PID.*

3. ☐ Una fork() crea un processo figlio che condividerà lo stesso spazio di indirizzamento e lo stesso valore iniziale delle variabili del processo padre. *A fork() creates a child process that will share the same address space and the same initial value of the variable as the parent process.*
4. ☐ Le istruzioni che seguono la fork() vengono eseguite o dal processo figlio o dal processo padre. *Instructions following fork() are executed by the child process or the parent process.*
5. ☒ Le pipe possono essere usate per comunicare tra il processo padre e il processo figlio creato mediante la system call fork(). *Pipes can be used to communicate between the parent process and the child process created by the system call fork().*
6. ☐ Il processo padre inizia la sua esecuzione dal punto in cui è stata chiamata la fork(). *The parent process starts its execution from the point where fork() was called.*
7. ☐ A seguito di una fork(), tutte le risorse del processo padre sono condivise con il processo figlio. *As a result of a fork(), all resources of the parent process are shared with the child process.*

Ex 03 (3.5 points)

Italiano

Si supponga di eseguire il seguente programma. Si indichino quali affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Suppose to run the following program. Indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

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

int main(){
    int i;
    setbuf (stdout, 0);
    for (i=1; i<=3; i++) {
        switch (i) {
            case 1: printf ("X"); break;
            case 2: fork(); system ("echo Y"); break;
            case 3: fork(); fork(); execlp ("echo", "myPgrm", "Z", NULL); break;
        }
    }
    return (0);
}
```

Fasi operative. Workflow.

XYZZYZZZZZZZ

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

1. ☒ Il programma visualizza una X. *The program displays one X.*
2. ☐ Il programma visualizza due X. *The program displays two Xs.*
3. ☐ Il programma visualizza tre X. *The program displays three Xs.*
4. ☐ Il programma visualizza una Y. *The program displays one Y.*
5. ☒ Il programma visualizza due Y. *The program displays two Ys.*
6. ☐ Il programma visualizza tre Y. *The program displays three Ys.*
7. ☐ Il programma visualizza due Z. *The program displays two Zs.*
8. ☐ Il programma visualizza quattro Z. *The program displays four Zs.*
9. ☐ Il programma visualizza sei Z. *The program displays six Zs.*
10. ☒ Il programma visualizza otto Z. *The program displays eight Zs.*

Ex 04 (2.0 points)

Italiano

Si considerino la system call **signal(SIGCHLD, SIG_IGN)** e la system call **wait**. Si indichino quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Consider the system call **signal(SIGCHLD, SIG_IGN)** and the call **wait**. Indicate which of the following statements are correct. Note that incorrect answers will result in a penalty in the final score.

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

- ☒ La system call **signal(SIGCHLD, SIG_IGN)** impedisce la generazione di processi zombie. **The signal(SIGCHLD, SIG_IGN) system call prevents the creation of zombie processes.**
- ☒ La system call **wait** blocca il processo chiamante finché uno dei suoi processi figli termina. **The wait system call blocks the calling process until one of its child processes terminates.**
- ☒ Se un processo padre imposta **signal(SIGCHLD, SIG_IGN)**, i processi figli terminati verranno raccolti automaticamente dal sistema operativo senza che il processo padre debba chiamare **wait**. **If a parent process sets signal(SIGCHLD, SIG_IGN), the operating system will automatically manage/reap the terminated child processes without the parent process needing to call wait.**
- ☐ Utilizzare **signal(SIGCHLD, SIG_IGN)** garantisce che il processo padre riceva una notifica ogni volta che un processo figlio termina. **Using signal(SIGCHLD, SIG_IGN) ensures that the parent process receives a notification every time a child process terminates.**
- ☒ Se **signal(SIGCHLD, SIG_IGN)** è stata impostata, la chiamata **wait** su un processo figlio non restituirà mai perché i processi figli vengono raccolti automaticamente. **If signal(SIGCHLD, SIG_IGN) has been set, the wait call on a child process will never return because the child processes are automatically managed/reaped.**
- ☒ La system call **wait** può essere usata per recuperare il PID di un processo figlio terminato e il suo stato di uscita. **The wait system call can be used to retrieve the PID of a terminated child process and its exit status.**
- ☐ Utilizzare **signal(SIGCHLD, SIG_IGN)** in combinazione con **wait** è una pratica comune per gestire la terminazione dei processi figli. **Using signal(SIGCHLD, SIG_IGN) in combination with wait is a common practice for handling the termination of child processes.**
- ☐ Quando viene usata **signal(SIGCHLD, SIG_IGN)**, il processo padre deve comunque chiamare **wait** per evitare la creazione di processi zombie. **When signal(SIGCHLD, SIG_IGN) is used, the parent process must still call wait to avoid the creation of zombie processes.**

Ex 05 (2.0 points)

Italiano

Relativamente all'implementazione delle pipe in un sistema operativo Unix, si indichino quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Regarding implementing pipes in a Unix operating system, indicate which statements are correct. Note that wrong answers imply a penalty in the final score.

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

- ☒ La scrittura su una pipe in cui tutti gli estremi di lettura sono stati chiusi provoca la generazione di un segnale di tipo **SIGPIPE**. **The write operation on a pipe, in which all reading extremes have been closed, causes the generation of a SIGPIPE signal.**
- ☐ Il tentativo di scrittura su una pipe piena restituisce errore. **The write attempt on a full pipe returns an error.**
- ☒ Il tentativo di scrittura su una pipe piena è normalmente bloccante. **The write attempt on a full pipe is normally blocking.**
- ☒ Una costante indica quanti byte possono essere scritti in una pipe in modo atomico. **A constant indicates how many bytes can be atomically written on a pipe**
- ☐ L'operazione di scrittura su una pipe è sempre eseguita in modo atomico. **The write operation on a pipe is always performed atomically.**

6.  La lettura su una pipe in cui tutti gli estremi di scrittura sono stati chiusi provoca la generazione di un segnale di tipo SIGPIPE. [The read operation on a pipe, in which all writing extremes have been closed, causes the generation of a SIGPIPE signal.](#)

Ex 06 (3.5 points)

Italiano

Si supponga di eseguire il seguente programma che visualizza un insieme infinito di righe contenenti i caratteri "A", "B", "C", "D", e "E". Si indichino quali tipologie di righe sono ammissibili. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

[Suppose to run the following program, which displays an infinite number of rows with the symbols "A", "B", "C", "D", and "E". Indicate which of the possible row format is admissible. Note that wrong answers imply a penalty in the final score.](#)

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

sem_t *sa, *sbc, *sde1, *sde2;
int na, nde;

static void *TA ();
static void *TB ();
static void *TC ();
static void *TD ();
static void *TE ();

int main (int argc, char **argv) {
    pthread_t th;
    sa = (sem_t *) malloc (sizeof(sem_t));
    sbc = (sem_t *) malloc (sizeof(sem_t));
    sde1 = (sem_t *) malloc (sizeof(sem_t));
    sde2 = (sem_t *) malloc (sizeof(sem_t));
    na = nde = 0;
    sem_init (sa, 0, 1);
    sem_init (sbc, 0, 0);
    sem_init (sde1, 0, 0);
    sem_init (sde2, 0, 0);
    setbuf(stdout, 0);
    pthread_create (&th, NULL, TA, NULL);
    pthread_create (&th, NULL, TB, NULL);
    pthread_create (&th, NULL, TC, NULL);
    pthread_create (&th, NULL, TD, NULL);
    pthread_create (&th, NULL, TE, NULL);
    pthread_exit(0);
}

static void *TA () {
    pthread_detach (pthread_self ());
    while (1) {
        sem_wait (sa);
        na++;
        if (na==1) {
            printf ("A-");
```

```

        sem_post (sbc);
    } else {
        na = 0;
        printf ("-A\n");
        sem_post (sa);
    }
}
return 0;
}

static void *TB () {
    pthread_detach (pthread_self ());
    while (1) {
        sem_wait (sbc);
        printf ( "B-");
        sem_post (sdel);
    }
    return 0;
}

static void *TC () {
    pthread_detach (pthread_self ());
    while (1) {
        sem_wait (sbc);
        printf ( "C-");
        sem_post (sdel);
    }
    return 0;
}

static void *TD () {
    pthread_detach (pthread_self ());
    while (1) {
        sem_wait (sdel);
        nde++;
        printf ("D");
        if (nde==1) {
            sem_post (sdel);
            sem_wait (sde2);
        } else {
            nde = 0;
            sem_post (sde2);
            sem_post (sa);
        }
    }
    return 0;
}

static void *TE () {
    pthread_detach (pthread_self ());
    while (1) {
        sem_wait (sdel);
        nde++;
        printf ("E");
        if (nde==1) {
            sem_post (sdel);

```

```

        sem_wait (sde2);
    } else {
        nde = 0;
        sem_post (sde2);
        sem_post (sa);
    }
}
return 0;
}

```

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

1. ☐ A-B-D-A
2. ☒ A-B-ED-A
3. ☐ A-B-C-ED-A
4. ☒ A-C-DE-A
5. ☒ A-B-DE-A
6. ☐ A-B-DE
7. ☒ A-C-ED-A
8. ☐ A-C-DE

Ex 07 (5.0 points)

Italiano

Un file include il calendario di un mese con il formato rappresentato in seguito. Il formato è simile, ma semplificato, a quello ottenuto con il comando di shell **cal**; in particolare tutti i campi sono costituiti da due caratteri e sono separati da un singolo spazio.

```

Luglio 2024
do lu ma me gi ve sa
XX 01 02 03 04 05 06
07 08 09 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31 XX XX XX

```

Si scriva uno script BASH che, una volta ricevuto il nome di tale file sulla riga di comando, visualizzi:

- Il giorno della settimana con cui inizia il mese; nell'esempio "lu", i.e., lunedì.
- Il giorno della settimana con cui finisce il mese; nell'esempio "me", i.e., mercoledì.
- Il numero di giorni della settimana all'interno del mese; nell'esempio luglio 2024 include 4 domeniche, 5 lunedì, 5 martedì, ..., 4 sabati.

English

A file includes the calendar of a month in the format shown below. The format is similar to, but simplified from that obtained with the **cal** shell command; in particular, all fields consist of two characters and are separated by a single space.

```

July 2024
su mo tu we th fr sa
XX 01 02 03 04 05 06
07 08 09 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31 XX XX XX

```

Write a BASH script that, once it receives the name of such a file on the command line, displays:

- The day of the week on which the month starts; in the above example, "mo", i.e., Monday.

- The day of the week on which the month ends; in the above example, “we”, i.e., Wednesday.
- The number of days of the week within the month; in the above example, July 2024 includes 4 Sundays, 5 Mondays, 5 Tuesdays, ..., and 4 Saturdays.

Solution. Solution.

```
#!/bin/bash

if [[ $# -lt 1 ]]; then
    echo "Missing file name"
    exit 1
fi

cnt=0
days=()
first=""
last=""
declare -A stats
while read -r line; do
    if [ $cnt -eq 0 ]; then
        month=$(echo $line | cut -d" " -f1)
        year=$(echo $line | cut -d" " -f2)
    elif [ $cnt -eq 1 ]; then
        for day in ${line}; do
            days+=($day)
            stats[$day]=0
        done
    else
        i=0
        for day in ${line}; do
            if [[ $day != "XX" ]]; then
                if [[ $first == "" ]]; then
                    first=${days[$i]}
                fi
                last=${days[$i]}
                ((stats[${days[$i]}]++))
            fi
            ((i++))
        done
    fi
    ((cnt++))
done < $1

echo "First day is $first"
echo "Last day is $last"

for day in ${!stats[@]}; do
    echo "$day appears ${stats[$day]} times"
done
```

Ex 08 (2.0 points)

Italiano

I segmenti di pseudo-codice inseriti di seguito sono eseguiti in concorrenza dai processi P1 e P2 di PID pid_P1 e pid_P2, rispettivamente. Si indichino quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

The following pseudo-code segments are executed concurrently by processes P1 and P2 with PID `pid_P1` and `pid_P2`, respectively. Indicate which statements are correct. Note that wrong answers imply a penalty in the final score.

```
P1
while (1) {
    pause ();
    A();
    ...
    kill (pid_P2, SIGUSR2);
    C();
}

P2
while (1) {
    ...
    kill (pid_P1, SIGUSR1);
    pause ();
    B();
}
```

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

- ☒ Nelle varie possibili sequenze di esecuzione si può avere la funzione B() seguita dalla funzione C() o la funzione C() seguita dalla funzione B(). *In the various possible sequences of execution, both function B() followed by function C() or function C() followed by function B() are possible.*
- ☐ E' possibile l'esecuzione consecutiva della funzione B() per più di una volta (senza nessuna funzione A() o C() nel mezzo). *It is possible to execute function B() more than one time (without the execution of any A() or C() functions in between).*
- ☒ I processi sono soggetti a deadlock. *Processes are subject to deadlocks.*
- ☒ I processi sono soggetti a starvation. *Processes are subject to starvation.*
- ☐ La funzione A() è sicuramente eseguita almeno una volta. *Function A() is certainly executed at least one time.*
- ☐ La funzione B() è sicuramente eseguita almeno una volta. *Function B() is certainly executed at least one time.*

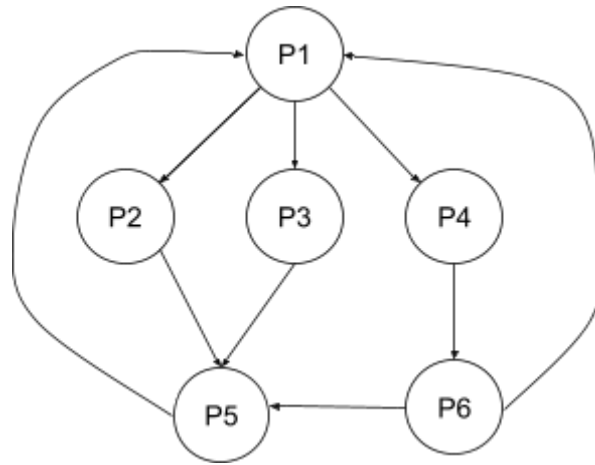
Ex 09 (5.0 points)

Italiano

Realizzare il seguente grafo di precedenza, utilizzando il minimo numero di semafori. I processi rappresentati devono essere processi ciclici (con corpo del tipo `while(1)`). Utilizzare le primitive `init`, `signal`, `wait` e `destroy`. Indicare gli eventuali archi superflui e riportare il corpo dei processi (`P1`, ..., `P6`) e l'inizializzazione dei semafori.

English

Implement the following precedence graph, using the minimum number of semaphores. The processes represented must be cyclical (with a body of type `while(1)`). Use the primitives `init`, `signal`, `wait`, and `destroy`. Indicate any unnecessary arcs and report the body of the processes (`P1`, ..., `P6`), and the initialization of the semaphores.



Soluzione. Solution.

// arc P6->P1 is unnecessary

```

init(p1_sem, 1);
init(p2_sem, 0);
init(p3_sem, 0);
init(p4_sem, 0);
init(p5_sem, 0);
init(p6_sem, 0);

void P1() {
    while (1) {
        wait(p1_sem);
        // P1
        signal(p2_sem);
        signal(p3_sem);
        signal(p4_sem);
    }
}

void P2() {
    while (1) {
        wait(p2_sem);
        // P2
        signal(p5_sem);
    }
}

void P3() {
    while (1) {
        wait(p3_sem);
        // P3
        signal(p5_sem);
    }
}

void P4() {
    while (1) {
        wait(p4_sem);
        // P4
        signal(p6_sem);
    }
}

```

```

    }
}

void P5() {
    while (1) {
        wait(p5_sem);
        wait(p5_sem);
        wait(p5_sem);
        // P5
        signal(p1_sem);
    }
}

void P6() {
    while (1) {
        wait(p6_sem);
        // P6
        signal(p5_sem);
    }
}

destroy(p1_sem);
destroy(p2_sem);
destroy(p3_sem);
destroy(p4_sem);
destroy(p5_sem);
destroy(p6_sem);

```

Ex 10 (4.0 points)

Italiano

Con riferimento al problema del “Produttore e Consumatore” lo si specializzi riferendosi al seguente contesto. Due code, di nome Q1 e Q2 e di dimensione uguale a N1 e N2, rispettivamente, mettono in comunicazione produttori e consumatori.

Esistono due insiemi di produttori denominati P1 (P1_1, P1_2, P1_3,...) e P2 (P2_1, P2_2, P2_3,...).

- P1 producono a intervalli di tempo casuali ma di massimo 4 secondi un oggetto sulla coda Q1.
- P2 producono a intervalli di tempo casuali ma di massimo 6 secondi un oggetto sulla coda Q2.

Esiste un unico insieme di consumatori C (C_1, C_2, C_3,...). Ogni consumatore consuma un oggetto da Q1 oppure Q2 a intervalli di tempo casuali di massimo 5 secondi. Con una probabilità del 60% consuma un oggetto dalla coda C1 e con una probabilità del 40% consuma dalla coda C2. Si può usare il codice “random_value = (double)rand() / RAND_MAX;” per generare un numero casuale nell’intervallo tra 0 e 1.

Si rappresenti un possibile schema di funzionamento in **pseudo-codice** e si indichi la funzione dei vari semafori motivandone l'utilizzo.

English

Concerning the “Producer and Consumer” problem, specialize it by referring to the following context.

Two queues, Q1 and Q2, with sizes equal to N1 and N2, respectively, connect producers and consumers.

There are two sets of producers, named P1 (P1_1, P1_2, P1_3,...) and P2 (P2_1, P2_2, P3_3,...).

- P1 processes produce an item on queue Q1 at random intervals of up to 4 seconds.
- P2 processes produce an item on queue Q2 at random intervals of up to 6 seconds.

There is a single set of consumers C (C_1, C_2, C_3,...). Each consumer consumes an item from Q1 or Q2 at random intervals of up to 5 seconds. With a 60% probability, it consumes an item from queue Q1, and with a 40% probability, it consumes from queue Q2. The code “random_value = (double)rand() / RAND_MAX;” can be used to generate a random number in the interval between 0 and 1.

Represent a possible operation scheme in **pseudo-code**, and explain the function of the various semaphores motivating their usage.

Soluzione. Solution.

```

init (full1, 0);
init (full2, 0);
init (empty1, N1);
init (empty2, N2);
init (MEp1, 1);
init (MEp2, 1);
init (MEc1, 1);
init (MEc2, 1);

P1()
item m;
while (1) {
    sleep(((double)rand()/RAND_MAX)*4);
    m = produce ();
    wait (empty1);
    wait (MEp1);
    enqueue1 (m);
    signal (MEp1);
    signal (full1);
}

P2()
item m;
while (1) {
    sleep(((double)rand()/RAND_MAX)*6);
    m = produce ();
    wait (empty2);
    wait (MEp2);
    enqueue1 (m);
    signal (MEp2);
    signal (full2);
}

C()
item m;
while (1) {
    sleep (((double)rand()/RAND_MAX)*5);
    if ( (double)rand()/RAND_MAX) <= 0.6) {
        wait (full1);
        wait (MEc1);
        m = dequeue1 ();
        signal (MEc1);
        signal (empty1);
        consume (m);
    } else {
        wait (full2);
        wait (MEc2);
        m = dequeue2 ();
        signal (MEc2);
        signal (empty2);
        consume (m);
    }
}

```

Ex 11 (2.5 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 Shortest-Remaining-Time First (SRTF). Si calcolino i tempi di terminazione di ciascun processo e il tempo di attesa medio. Si riportino i dati relativi all'algoritmo di scheduling con tempo medio di attesa. 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 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 30.5

English

Consider the following processes, scheduled with a temporal quantum of 10 units. Represent, using a Gantt diagram, the execution of these processes using the Shortest-Remaining-Time First (SRTF) scheduling algorithms. Compute the termination time of each process and the average waiting time. Report 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	31	5
P2	5	24	4
P3	11	15	3
P4	17	9	2
P5	25	11	1

Fasi operative. Workflow.

SRTF:

P1: from= 0 to = 4 end = 5 waitingTime = 0

```
P2: from= 5 to  = 10 end = 11 waitingTime = 0
```

```
P3: from= 11 to  = 25 end = 26 waitingTime = 0
```

P4: from= 26 to = 34 end = 35 waitingTime = 9

P5: from= 35 to = 45 end = 46 waitingTime = 10

P2: from= 46 to = 63 end = 64 waitingTime = 35

P1: from= 64 to = 89 end = 90 waitingTime = 59

EndTime: 90 64 26 35 46

AverageWaitingTime: $[(59) + (35) + () + (9) + (10)] / 5 = 22.60$

AverageTournaroudTime: $(90 + 59 + 15 + 18 + 21) / 5 = 40.60$

GanttChart:

[illegible]

Risposta. Answer.

90 64 26 35 46 22.6

Ex 12 (2.5 points)

Italiano

Un sistema ha 4 processi (P1, P2, P3, P4) e 4 tipi di risorse (R1, R2, R3, R4), in cui il numero di istanze di risorse del tipo R1 è 1, il numero di istanze per ogni risorsa del tipo R2 è 1, il numero di istanze per ogni risorsa del tipo R3 è 2 e il numero di istanze per ogni risorsa del tipo R4 è 1. Lo stato del sistema è il seguente:

- Il processo P1 detiene l'istanza della risorsa di tipo {R1}, detiene un'istanza della risorsa di tipo {R3} e attende un'istanza della risorsa di tipo {R2}.
- Il processo P2 detiene l'istanza della risorsa di tipo {R2}, attende un'istanza della risorsa di tipo {R3} e attende l'istanza della risorsa di tipo {R4}.
- Il processo P3 detiene un'istanza della risorsa di tipo {R3} e attende l'istanza della risorsa di tipo {R1}.
- Il processo P4 detiene l'istanza della risorsa di tipo {R4} e attende un'istanza della risorsa di tipo {R3}.

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 4 types of resources (R1, R2, R3, R4), in which the number of instances of resources of type R1 is 1, the number of instances for each resource of type R2 is 1, the number of instances for each resource of type R3 is 2, and the number of instances for each resource of type R4 is 1. The state of the system is the following:

- Process P1 holds the instance of resource type {R1}, holds one instance of resource type {R3}, and waits for an instance of resource type {R2}.
- Process P2 holds the instance of resource type {R2}, waits for an instance of resource type {R3}, and wait for the instance of resource type {R4}.

- Process P3 holds one instance of resource type {R3} and waits for the instance of resource type {R1}.
- Process P4 holds the instance of resource type {R4} and waits for an instance of resource type {R3}.

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 è in una condizione di deadlock. The system SURELY is NOT in a deadlock condition.
2. ☒ Eliminando l'arco P1->R2 si ha la CERTEZZA di NON essere in una condizione di deadlock. Eliminating the edge P1->R2, SURELY the system is NOT in a deadlock condition.
3. ☒ Il deadlock è POSSIBILE. The deadlock is POSSIBLE.
4. ☒ Eliminando l'arco R2->P2 si ha la CERTEZZA di NON essere in una condizione di deadlock. Eliminating the edge R2->P2, SURELY the system is NOT in a deadlock condition.
5. ☐ Eliminando l'arco P4->R3 si ha la CERTEZZA di NON essere in una condizione di deadlock. Eliminating the edge P4->R3, 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