Esame Sistemi Operativi - Operating Systems Exam
2021/06/18

Ex 1 (2.0 points)

Italiano

Dei thread possono richiedere l'allocazione di un sottoinsieme di R risorse (0 ≤ r < R) chiamando la funzione `void req(int nr, int rv[])` nel seguente modo:

```
int rv[3] = {5, 2, 7};
req(3, rv);
```

al fine di richiedere le 3 risorse identificate dagli interi 5, 2, 7.

Ogni thread chiama la funzione req una sola volta per allocare tutte le risorse di cui necessita. Per liberare invece tutte le risorse di cui necessita deve chiamare la funzione `void rel(int nr, int rv[])` con esattamente gli stessi argomenti usati per req.

Si consideri la seguente soluzione in pseudocodice e si indichi quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Threads can request the allocation of a subset of Resource R (0 ≤ r < R) by calling the `void` function `req(int nr, int rv[])` as follows:

```
int rv[3] = {5, 2, 7};
req(3, rv);
```

in order to request the 3 resources identified by integers 5, 2, 7.

Each thread calls the req function once to allocate all the resources it needs. Instead, to free all the resources it needs, it must call the `void rel(int nr, int rv[])` function with exactly the same arguments used for req.

Consider the following solution in pseudocode and indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

```
#define R 5
sem_t sem[R] = {1, ... 1};
void req(int nr, int rv[]) {
      int i;
      for(i=0; i<nr; i++)
            wait(&sem[rv[i]]);
}
void rel(int nr, int rv[]) {
      int i;
      for(i=0; i<nr; i++)
            post(sem[rv[i]]);
}
```

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

1. ✅ Se ogni thread prima di richiedere le risorse ordina il vettore rv in ordine crescente il deadlock non è possibile. If each thread before requesting resources sorts the rv vector in ascending order, deadlock is not possible.
2. ✅ Se ogni thread prima di richiedere le risorse ordina il vettore rv in ordine decrescente il deadlock non è possibile. If each thread before requesting resources sorts the rv vector in descending order, deadlock is not possible.
3. ⬜ Non vi è deadlock in quanto la condizione necessaria di "hold and wait" non è possibile. There is no deadlock as the necessary condition of "hold and wait" is not possible.
4. ⬜ Non vi è deadlock in quanto la condizione necessaria di "circular wait" non è possibile. There is no deadlock as the necessary condition of "circular wait" is not possible.

5. ✅ Relativamente allo pseudocodice sopra riportato, se c'è un ciclo nel grafo di allocazione delle risorse si ha deadlock e vice versa. With regard to the pseudocode above, if there is a cycle in the resource allocation graph you have deadlock and vice versa.
6. ✅ Ci può essere deadlock. There may be deadlock.

Ex 2 (1.5 points)

Italiano

Si supponga di eseguire il seguente tratto di codice:
Si indichi quale delle seguenti affermazioni è corretta

English

Suppose you run the following segment of code:
Please indicate which of the following statements is correct.

```
if (fork()) {
      sleep (10);
      exit (1);
} else {
      exit (1);
}
```

Scegli UNA SOLA alternativa: Choose JUST ONE option:

1. ⬜ Il processo figlio diventerà orfano. The child process will become an orphan.
2. ⬜ Il processo padre diventerà zombie. The parent process will become zombie.
3. ⬜ Il processo figlio diventerà zombie quando il padre termina. The child process will become zombie when the parent terminates.
4. ⬜ Il processo padre diventerà orfano. The parent process will become an orphan.
5. ✅ Il processo figlio diventerà zombie.The child process will become zombie.

Ex 3 (2.5 points)

Italiano

Si analizzi il seguente tratto di codice.
Si indichi quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale

English

Analyze the following segment of code.
Please indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

```
void* myfunc(void* ptr) {
      int *tidP, data;
      data = (int *) ptr;
      printf("%d ", data);
      return NULL;
}
int main() {
      int i, x;
      void *retval;
      pthread_t tid[10];
      for (i=0; i<10; i++) {
            pthread_create(&tid[i], NULL, myfunc, (void *)i);
      }
      for (x=0; x<10; x++) {
```

```
            pthread_join (tid[x], &retval);
        }
        pthread_exit(NULL);
}
```

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

1. ⬜ Il codice può visualizzare la sequenza: 1 2 3 4 5 6 7 8 9 11 The code can display the sequence: 1 2 3 4 5 6 7 8 9 11
2. ✅ Il codice contiene una race-condition. The code contains a race condition
3. ⬜ Il codice può visualizzare la sequenza: 1 2 3 4 The code can display the sequence: 1 2 3 4
4. ⬜ Il codice non contiene una race-condition. The code does not contain a race condition
5. ✅ Il codice può visualizzare la sequenza: 1 2 3 4 5 6 7 8 9 10 The code can display the sequence: 1 2 3 4 5 6 7 8 9 10
6. ✅ Il codice può visualizzare la sequenza: 1 7 8 8 8 8 8 8 8 10 The code can display the sequence: 1 7 8 8 8 8 8 8 8 10
7. ✅ Il codice può visualizzare la sequenza: 0 1 2 3 4 5 6 7 8 9 The code can display the sequence: 0 1 2 3 4 5 6 7 8 9 .

Ex 4 (1.5 points)

Italiano

Si supponga che dei file siano allocati su un hard disk usando l'allocazione di tipo contiguo.
Si indichi quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale

English

Suppose files are allocated on a hard disk using contiguous allocation.
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. ✅ Essa soffre di frammentazione interna. It suffers from internal fragmentation.
2. ⬜ Ogni blocco contiene un puntatore al blocco successivo. Each block contains a pointer to the next block.
3. ✅ Essa soffre di frammentazione esterna. It suffers from external fragmentation.
4. ✅ L'accesso sequenziale è immediato, facile da realizzare. Sequential access is immediate, easy to achieve.
5. ⬜ I file possono crescere di dimensione fintantoché c'è spazio sull'hard disk. Files can grow in size as long as there is space on the hard disk

Ex 5 (3.0 points)

Italiano

Se si esegue il programma successivo, quanti caratteri 'P' saranno visualizzati su standard output?

English

If the following program is run, how many characters 'P' will be displayed on standard output?

```
int main () {
        int i;
        i=0;
        while (i<3 && fork()) {
                fork ();
                i++;
        }
```

```
        printf ("P");
        return 1;
}
```

Scegli UNA SOLA alternativa: Choose JUST ONE option:

1. ⬜ 9
2. ⬜ 16
3. ✅ 15
4. ⬜ 8
5. ⬜ 7

Ex 6 (2.5 points)

Italiano

Si consideri il precedente 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 Job First (SJF). 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, ..., P6 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. Errori di formato verranno considerati alla stregua degli altri errori. Esempio di risposta corretta: 20 23 11 7 45 67 30.5

English

Consider the previous set of scheduled processes with a time quantum of 10 units of time.
Represent the execution of these processes using the Shortest Job First (SJF) algorithm by Gantt chart. Calculate the termination time of each job and the average wait time.
Please report the answer on a single line, indicating the termination times of P1, ..., P6 followed by the average waiting time. Separate the numbers with a single space. Report the average wait time with only 1 decimal place. Do not insert any other character into the answer. Format errors will be treated as other errors. Example of correct answer: 20 23 11 7 45 67 30.5

Processo Process	TempoArrivo ArrivalTime	BurstTime	Priorità Priority
1	0	23	5
2	0	17	1
3	0	15	4
4	0	21	3
5	0	19	2
6	0	12	6

Risposta: Answer:
107 44 27 84 63 12 38.3

Ex 7 (1.5 points)

Italiano

Quali delle seguenti affermazioni sulle pipe sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Which of the following pipe 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. ✅ Le operazioni di scrittura su una pipe sono atomiche fino alla dimensione della costante PIPE_BUF. Write operations on a pipe are atomic up to the size of the constant PIPE_BUF.
2. ⬜ Un'operazione di lettura su una pipe è sempre bloccante. A read operation on a pipe is always blocking.
3. ⬜ Si può scrivere da ambo il lati della pipe contemporaneamente. It is possible to write on both sides of the pipe at the same time
4. ✅ Una pipe si riempie quando lo scrittore scrive troppo sulla pipe senza che il lettore legga nulla. A pipe gets filled up when the writer writes too much to the pipe without the reader reading any of it.
5. ✅ Un'operazione di scrittura è tipicamente bloccante su una pipe piena. A write operation is typically blocking on a full pipe.

Ex 8 (4.5 points)

Italiano

Scrivere uno script bash in grado di calcolare la somma delle lunghezze delle parole presenti sulla diagonale di una matrice quadrata che contiene solo stringhe.
La matrice quadrata viene memorizzata in un file di testo, specificato come parametro da linea di comando. Verificare il corretto passaggio di tale parametro.

English

Write a bash script able to compute the sum of lengths of the words appearing on the diagonal of a square matrix that contains just strings.
The square matrix is stored in a text file, specified as a command line argument to the script. Check that such an argument is correctly passed to the script.

Example

```
word11 word12 word13 word14
word21 word22 word23 word24
word31 word32 word33 word34
word41 word42 word43 word44
```

Result: 24

Risposta: Answer:

```
#!/bin/bash
################################################################################
Exercise 11 of exam 18/06/2021 # # Version A #
################################################################################
Check correct number of arguments passed
if [ $# -ne 1 ]; then
      echo "Usage: ./ex11a.sh "
      exit 1
fi

# Check input file existence
if [ ! -e $1 ]; then
      echo "File $1 not found!"
      exit 1
fi

# Scan matrix and sum lengths
i=0
j=0
tot=0
while read line; do
      for word in $line; do
            if [ $i -eq $j ]; then
```

```
                l=$(echo -n $word | wc -m)
                let "tot+=$l"
            fi
            let "j+=1"
      done
      let "j=0"
      let "i+=1"
done < $1
# Print result
echo "Result: $tot"


#!/bin/bash
################################################################################
Exercise 11 of exam 18/06/2021 # # Version B #
################################################################################
Check correct number of arguments passed
if [ $# -ne 1 ]; then
      echo "Usage: ./ex11a.sh "
      exit 1
fi

# Check input file existence
if [ ! -e $1 ]; then
      echo "File $1 not found!"
      exit 1
fi

# Retrieve square matrix dimension
N=$(cat $1 | wc -l)

# Compute sum of lengths
tot=0
for((i=1; i<=N; i++)); do
      word=$(head -n $i $1 | tail -n 1 | tr -s " " | cut -d " " -f $i)
      l=$(echo -n $word | wc -m)
      let "tot+=$l"
done
# Print result
echo "Result: $tot"


#!/bin/bash
################################################################################
Exercise 11 of exam 18/06/2021 # # Version C #
################################################################################
Check correct number of arguments passed
if [ $# -ne 1 ]; then
      echo "Usage: ./ex11a.sh "
      exit 1
fi

# Check input file existence
if [ ! -e $1 ]; then
      echo "File $1 not found!"
      exit 1
fi
```

```
# Scan matrix and sum lengths
i=1
tot=0
while read line; do
      word=$(echo -n $line | tr -s " " | cut -d " " -f $i)
      l=$(echo -n $word | wc -m)
      let "tot+=$l"
      let "i+=1"
done < $1
# Print result
echo "Result: $tot"
```

Ex 9 (3.0 points)

Italiano

Se il programma seguente viene eseguito con un singolo parametro uguale al valore intero 5, cioè
`./pgrm 5`
quanti caratteri 'E' e quanti caratteri 'e' vengono visualizzati su standard output?

English

If the following program is run with a single parameter equal to the integer value 5, i.e.,
`./pgrm 5`
how many characters 'E' and how many characters ‘e’ will be displayed on standard output?

```
int main (int argc, char *argv[]) {
      char str[20];
      int n = atoi (argv[1]);
      setbuf (stdout, 0);
      if (n>0) {
            printf ("E");
            sprintf (str, "%d", n-1);
            execlp (argv[0], argv[0], str, NULL);
      }
      printf ("e");
      return 1;
}
```

Scegli UNA SOLA alternativa: Choose JUST ONE option:

1. ⬜ 5 'E', 5 'e'.
2. ⬜ 1 'E', 5 'e'.
3. ⬜ 1 'E', 1 'e'.
4. ✅ 5 'E', 1 'e'.
5. ⬜ 6 'E', 6 'e'.

Ex 10 (4.5 points)

Italiano

Usando esclusivamente comandi di shell, effettuare le seguenti operazioni:

1. estrarre la 50-ma riga di un file di nome dato.
2. salvare in un file di nome “list.txt” i nomi di tutti i file con estensione “.h”, che sono contenuti nell’albero di direttori con radice “/home/foo” e che contengono la stringa “define” oppure la stringa “include”.

3. ordinare alfabeticamente in ordine crescente tutte le righe del file di nome “bar.txt”, eliminare le righe duplicate e prelevare solo la seconda stringa di ciascuna riga. Memorizzare il risultato nel file “/home/tmp/list.txt”.

Si indichino 3 comandi separati utilizzando eventuali operazioni di pipe o ridirezione.

English

Using shell commands only, do the following:

1. extract the 50-th row of a given file name.
2. save in a file named "list.txt" the names of all the files with extension ".h", which are contained in the root "/home/foo" directory tree and contain the string "define" or the string "include".
3. sort alphabetically in ascending order all lines in the file named "bar.txt", delete the duplicate lines, and pick only the second string of each line. Store the result in the file "/home/tmp/list.txt".

Indicate 3 separate commands using pipes or redirection operations.

Risposta: Answer:

```
1.
cat "file.txt" | head -n 50 | tail -n 1
head -n 50 "file.txt" | tail -n 1

2.
find "/home/foo" -type f -name "*.h" -exec grep -l -E "(define)|(include)" \{} \; >
"list.txt"
find "/home/foo" -type f -name "*.h" -exec grep -l -e "define" -e "include" \{} \; >
"list.txt"
find "/home/foo" -type f -name "*.h" -exec grep -H -e "define" -e "include" \{} \; |
cut - d ":" -f 1 > "list.txt"
grep -l -E "(define)|(include)" $(find "/home/foo" -name "*.h" -type f)

3.
cat "bar.txt" | sort | uniq | cut -d " " -f 2 > "/home/tmp/list.txt"
sort "bar.txt" | uniq | cut -d " " -f 2 > "/home/tmp/list.txt"
sort -u "bar.txt" | cut -d " " -f 2 > "/home/tmp/list.txt"
```

Ex 11 (1.5 points)

Italiano

Si supponga un processo effettui una "waitpid". Si indichi quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale

English

Suppose a process does a waitpid. 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 processo riceverà un SIGCHLD non appena un suo figlio termina. The process will receive a SIGCHLD as soon as one of its children ends
2. ⬜ Il processo potrà attendere un massimo numero di secondi. The process can wait a maximum number of seconds
3. ⬜ Il processo uscirà dalla waitpid alla terminazione del suo primo figlio. The process will exit from waitpid at the end of its first child
4. ✅ Il processo potrà rimanere bloccato sulla waitpid anche dopo la terminazione di un figlio. The process may get stuck on the waitpid even after a child has terminated

Ex 12 (3.0 points)

Italiano

Si implementino le system calls sem_init, sem_post e sem_wait utilizzando le proprietà di sincronizzazione delle pipe. Si indichi qual è la logica della soluzione.

English

Implement the system_calls sem_init, sem_post, and sem_wait exploiting the synchronization properties of the pipes. Indicate what the logic of the solution is

Risposta: Answer:

```
void sem_init (int *S) {
	if(pipe(S)==-1) {
		printf("Error\n");
		exit(-1);
	}
	return;
}

void sem_wait (int *S) {
	char car;
	if(read(S[0], &car, sizeof(char))!=1) {
		printf("Errore");
		exit(-1);
	}
	return;
}

void sem_post (int *S) {
	char car='X';
	if(write(S[1], &car, sizeof(char))!=1) {
		printf("Error");
		exit(-1);
	}
	return;
}
```

Ex 13 (5.0 points)

Italiano

In un sistema esistono due thread: uno di tipo A e uno di tipo B. Il thread di tipo A può eseguire la funzione funz_a(), mentre il thread di tipo B può eseguire la funzione funz_b().
Il thread che esegue per primo una delle due funzioni è bloccato, mentre il secondo (quello che esegue l'altra funzione) non è bloccato e sblocca il processo precedentemente bloccato dall'altra funzione.
Tali funzioni possono essere riutilizzate dai thread A e B un numero indefinito di volte.
Si realizzino le funzioni funz_a() e funz_b() con il numero minimo di semafori POSIX, definendone le relative variabili condivise e i valori di inizializzazione dei semafori.
Esempio:

- A chiama la funzione funz_a() e si blocca
- B chiama la funzione funz_b() che sblocca A
- B chiama la funzione funz_b() e si blocca
- A chiama la funzione funz_a() che sblocca B

English

In a system there are two threads: one of type A and one of type B. The type A thread can call the funz_a() function, while the B-type thread can call the funz_b() function.
The thread that performs one of the two functions first is blocked, while the second (the one running the other function) is not blocked and unlocks the thread previously blocked by the other function.
These functions can be reused by threads A and B an indefinite number of times.
Realize the funz_a() and funz_b() functions with the minimum number of POSIX semaphores, defining their shared variables and semaphore initialization values.

Example:

- A calls the function funz_a() and it blocks
- B calls the function funz_b() and it unlocks A
- B calls the function funz_b() and it blocks
- A calls the function funz_a() and unlocks B

Risposta: Answer:

```
// Solution 1
init(m, 1);
init(s, 0);
int flag=0;
funz_a() {
	wait(m);
	if (flag==0) {
		flag=1;
		signal(m);
		wait(s);
	} else {
		flag=0;
		signal(m);
		signal(s);
	}
}
funz_b() {
	wait(m);
	if (flag==0) {
		flag=1;
		signal(m);
		wait(s);
	} else {
		flag=0;
		signal(m);
		signal(s);
	}
}

// Solution 2
init(a, 0);
init(b, 0);
funz_a() {
	signal(b);
	wait(a);
}
funz_b() {
	signal(a);
	wait(b);
}
```