

UNIVERSITA' di VERONA
DIPARTIMENTO di INFORMATICA

CORSO DI LAUREA in BIOINFORMATICA

Elementi di architettura e sistemi operativi
2017/2018

CONSEGNA - ELEARNING: entro 23.00 del 18 Giugno 2018

Elaborato: 2 - System Call C

Si vuole implementare un simulatore di calcolo parallelo sincronizzato. Il programma dovrà leggere un file di configurazione *config.txt* il quale avrà la seguente struttura:

```
<NPROC>
<id> <num1> <op> <num2>
.
.
<id> <num1> <op> <num2>
```

- NPROC: numero di processi FIGLIO che andranno ad eseguire i calcoli.
- Id: id-esimo figlio che dovrà eseguire l'operazione corrispondente.
- Num1/2: gli operandi
- Op: l'operazione (+, -, *, /)

Esempio di un file di configurazione:

Config.txt

```
3
2 5 + 1
1 2 - 10
2 4 * 3
3 1 / 2
```

ASSUNZIONI:

- I processi figlio indicati nel file di configurazione non saranno più di 9.
- Numero **massimo** di operazioni per figlio saranno 10.
- Nel file di configurazione ci saranno solo operazioni computabili (NO 1/0 per esempio)
- Qualsiasi altra assunzione non specificata è lecita, indicandola nella relazione e coerentemente con il resto del progetto.

Il processo **PADRE** dovrà eseguire la seguente routine:

- Legge il file di configurazione e inizializza le risorse condivise (code di messaggi, pipe) in maniera opportuna.
- Genera NPROC processi figlio (l'utilizzo anche del comando EXEC non comporterà una penalità)
- Invia TUTTE le operazioni ai processi figlio tramite "code di messaggi" e SENZA attesa dei risultati. Inoltre dovrà inviare un carattere speciale "X" per notificare al figlio la terminazione delle operazioni a lui destinate.
- Terminato l'invio delle operazioni, il processo padre dovrà leggere tramite PIPE i risultati delle operazioni nell'ordine specificato nel file di configurazione.
- Se l'attesa in lettura sulla PIPE supera 10 secondo, il processo dovrà forzare la terminazione di tutti i processi FIGLIO (segnale SIGTERM), liberare le risorse e ritornare un messaggio di ERRORE.
- Se tutto va a buon fine, il processo padre dovrà scrivere i risultati su un file *Output.txt* nello stesso ordine indicato nel file di configurazione.

UNIVERSITA' di VERONA

DIPARTIMENTO di INFORMATICA

- Il processo padre è responsabile della "terminazione dei processi figli" la terminazione avviene nelle seguenti ipotesi:
 - Il processo padre riscontra un errore di qual si voglia natura, o il processo padre riceve un segnale a sua volta di terminazione:
Il processo dovrà far terminare i processi figlio (segnale SIGTERM), liberare le risorse e terminare in maniera corretta.
 - I processi figli hanno completato il loro compito e terminano... il processo padre completato il suo compito in modo corretto si accerta che tutti i processi figli abbiano terminato in modo corretto (wait())

I processi **FIGLI** dovranno eseguire la seguente routine:

- Inizializzare le risorse
- Prenotarsi per intercettare il segnale SIGTERM. Nel momento in cui viene intercettato tale segnale, il processo dovrà liberare tutte le risorse allocate e terminare in maniera corretta.
- Leggere dalla coda di messaggi TUTTE le operazioni a loro assegnate e svolgerle solo DOPO aver ricevuto il carattere speciale "X".
- Terminare la computazione di tutte le operazioni.
- Il processo dovrà inviare al PADRE i risultati in ordine (separati da uno spazio) tramite PIPE.

Il codice è preferibile, strutturato, su più file e commentato.

Consegnare un'archivio contenente il programma e tutti i sorgenti necessari debitamente commentati e indentati, una piccola relazione in formato PDF che illustri il funzionamento delle parti principali del programma, in particolare quelle specificatamente richieste.

I sorgenti consegnati devono potersi compilare senza errori, pena la nullità del progetto, il progetto verrà VALUTATO anche se incompleto di alcune funzionalità richieste.

Tutto quanto non esplicitato in questo documento può essere implementato liberamente. La mancanza di parti indicate nella specifica o sviluppate in maniera diversa da quanto indicato NON comportano l'annullamento del progetto ma una penalità.

L'elaborato è personale, la presentazione di elaborati identici comporta la NULLITÀ degli elaborati UGUALI.

La prima pagina della relazione dovrà riportare le seguenti informazioni:

#Matricola
#Nome e cognome
#Data di realizzazione
#Titolo esercizio

Riferimenti

Nicola Drago: nicola.drago@univr.it

Justin Joita: id755ftj@studenti.univr.it

CONSEGNA TRAMITE E-LEARNING

CONSEGNA – E-LEARNING: **entro 23.00 del 18 Giugno 2018**

UNIVERSITA' di VERONA

DIPARTIMENTO di INFORMATICA

FAQ

1) È obbligatorio scrivere il codice su più file?

No, ma è preferibile e valutato di conseguenza.

2) Se il progetto compila sul pc personale, ma non compila sul pc del tutor cosa succede?

Vi presentate lo stesso all'orale, si verifica l'uguaglianza del codice che eseguirete con quello inviato al tutor, quindi esporrete il vostro progetto (si consiglia fortemente di testare il TUTTO su un pc del laboratorio).

3) Alcune cose non sono ben specificate nell'elaborato. Cosa faccio?

Alcune cose non sono specificate apposta per lasciare liberta' agli studenti di implementarle come preferiscono. In caso di dubbi, si possono comunque contattare il docente o il tutor per eventuali chiarimenti. Eventuali chiarimenti sul testo verranno pubblicati come avviso e sul sito per renderli fruibili a tutti.