

Progetto del Laboratorio di Sistemi Operativi

a.a. 2020-21

Docente della parte di Laboratorio: Fabio Rossi

Data ultima Revisione: 15 Maggio 2021

Descrizione dell'applicazione da realizzare

Lo studente deve realizzare un'applicazione in linguaggio C che verifichi quali tra i primi n numeri naturali siano primi. La verifica è attuata in parallelo da almeno 3 entità di calcolo distinte, che possono essere sia processi single-threaded (anche "imparentati" con `fork()`) che threads di uno stesso processo. Possiamo chiamare queste entità di calcolo "**calcolatori**". Per "in parallelo" si intende che i 3 calcolatori operano contemporaneamente e continuamente su 3 numeri diversi.

La scelta di attuare il calcolo con più processi single-threaded o più threads di uno stesso processo è a discrezione dello studente, come anche la scelta di aggiungere eventuali altri processi/threads che svolgano funzione accessorie di coordinamento/dialogo tra processi/threads.

I numeri risultati primi (e solo questi) devono essere memorizzati in un file uno per ogni riga ed in ordine crescente; tale memorizzazione deve essere attuata da un altro processo completamente separato dai calcolatori che chiameremo "**memorizzatore**" (nemmeno "imparentato" tramite `fork()`). (nдр: modifica del 26/05/2021)

Il memorizzatore viene informato dagli altri processi (calcolatori o accessori) di quali numeri primi registrare tramite almeno una delle seguenti forme di IPC: **pipe**, **socket**, **signal**, **shared memory** (nдр: modifica del 27/05/2021).

Per la verifica della primalità potete utilizzare qualsiasi algoritmo, anche triviale, purché corretto.

Alcuni requisiti

Lo studente deve sviluppare l'applicazione cercando di:

- Parallelizzare il più possibile il calcolo; in altre parole i calcolatori devono avere minori "tempi morti" (cioè di assenza di attività di calcolo) possibili. Per esempio: eventuali attese dovrebbero avvenire solo per il tempo strettamente necessario
- Eventuali attese di processi/threads resi necessarie devono essere "passive", cioè non impegnare la CPU
- n ed il path del file in cui memorizzare i numeri primi devono essere passati da linea di comando o, in alternativa, letti da file esterno
- i processi impiegati devono visualizzare su standard output messaggi che informino l'utente il più possibile su cosa il singolo processo/threads stia facendo: quale numero è in corso di verifica e da parte di quale calcolatore, quale numero primo è in corso di memorizzazione (crescente), l'accesso a meccanismi di IPC, etc. e comunque tutto ciò che possa far capire il "flusso" delle operazioni nell'intera applicazione

Makefile e script di lancio

Lo studente deve fornire anche un *makefile* atto a compilare tutti i sorgenti componenti l'applicazione ed uno *script Bash* che lanci la compilazione e tutti i processi che compongono l'applicazione

Altre indicazioni

Sarà considerata nota di merito l'inserimento di commenti nel codice che ne facilitino la comprensione, così come l'inserimento di commenti atti a motivare le scelte di progetto dell'applicazione fatte.

Il progetto deve essere spedito via email al Docente all'indirizzo fabio.rossi@unipg.it, oppure messo a disposizione su un qualsiasi cloud e reso scaricabile tramite link da inviare sempre al suddetto indirizzo.

La consegna deve avvenire tassativamente almeno una settimana prima della data dell'esame. Il formato può essere tar, tgz o zip.