

Pthread

- **Create:** Crea un nuovo thread.
- **Join:** Attende che un thread specificato termini.
- **Lock e Unlock:** Gestiscono l'accesso esclusivo a una sezione di codice tramite mutex.
- **Wait:** Attende che una condizione specificata diventi vera.
- **Barrier:** Coordina i thread, facendoli attendere finché tutti non raggiungono un punto specifico nel codice.
- **Condition:** Variabili di condizione utilizzate per la sincronizzazione tra i thread.
- **Mutex:** Fornisce il locking mutualmente esclusivo per proteggere la sezione critica.

MPI

- **Mpi_Send e Mpi_Recv:** Permettono rispettivamente l'invio e la ricezione di messaggi tra processi.
- **Mpi_Init e Mpi_Finalize:** Inizializzano e terminano l'ambiente MPI.
- **Mpi_Comm_size e Mpi_Comm_rank:** Restituiscono il numero di processi in un communicator e il rank (identificativo) del processo chiamante all'interno del communicator.
- **Mpi_Bcast:** Distribuisce un dato dal processo root a tutti gli altri processi nel communicator.
- **Mpi_Barrier:** Sincronizza tutti i processi in un communicator, facendo attendere ciascuno finché tutti non arrivano a questo punto.
- **Mpi_Reduce:** Combina i dati da tutti i processi e li riduce ad un singolo risultato, utilizzando un'operazione specificata (come somma, max, min, ecc.).
- **Mpi_Scatter e Mpi_Gather:** Distribuiscono (scatter) e raccolgono (gather) dati tra tutti i processi in un communicator.
- **Mpi_Isend e Mpi_Irecv:** Versioni non bloccanti di Mpi_Send e Mpi_Recv, che permettono al processo di continuare l'esecuzione mentre si attende il completamento dell'operazione.
- **Mpi_Ssend:** Variante di Mpi_Send che attua un invio sincrono.
- **Mpi_Wait:** Utilizzato con operazioni non bloccanti per attendere il loro completamento.

Argomenti d'esame

1. **Calcolo dell'Overhead e Efficienza in Ambienti Paralleli:** Domande su come calcolare l'efficienza e la relazione tra tempo di esecuzione seriale e parallelo.
2. **Problematiche di False Sharing in Memoria Condivisa:** Analisi di codice in contesti di memoria condivisa e identificazione di false sharing.
3. **Tempo di Comunicazione in Reti:** Calcolo del tempo di comunicazione in specifiche configurazioni di rete.
4. **Programmazione Multithreading con Pthread:** Comprendere il comportamento e il tempo di esecuzione di programmi che utilizzano thread, con particolare attenzione all'uso della libreria Pthread.
5. **Realizzazione di Funzionalità Specifiche con Pthread:** Implementazione di funzionalità come barrier con timeout, utilizzando la libreria Pthread.

6. **Comunicazione MPI e Utilizzo di MPI_Type_vector:** Domande sull'uso di MPI per lo scambio di dati, inclusa la creazione di datatype personalizzati con MPI_Type_vector.
7. **Scambio di Dati e Trasposizione di Matrici in MPI:** Implementazione di scambi di dati in MPI, con particolare attenzione alla trasposizione di matrici.