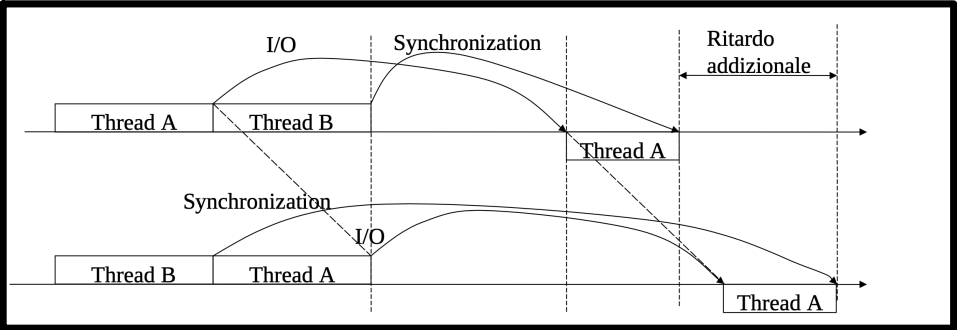


Scheduling di threads

Se ci muoviamo a schedulare in CPU percorsi d'esecuzione che vivono all'interno della stessa applicazione poiché toccano strutture dati che sono all'interno dello stesso address space, abbiamo una gestione diversa rispetto alla gestione che avevamo con processi.

Abbiamo un'applicazione basata su due threads, un thread A e un thread B, e sotto abbiamo un'altra applicazione con i thread A e B in ordine differente d'assegnazione della CPU: sopra abbiamo una timeline in cui assegniamo la CPU prima al thread A e poi al thread B. Il thread B eventualmente nel momento in cui il thread A è in esercizio, può andare in qualche modo a chiedere il rilascio della CPU per attendere che eventualmente sia eseguito qualche cosa.



Il thread A chiama un servizio di I/O e rilascia la CPU, e nel mentre il thread B esegue e finisce, dopo A riprende il controllo per magari effettuare il JOIN con le attività del thread B. Se prima mettiamo il thread B e dopo il thread A con le medesime operazioni, il tempo di turnaround è più grande, dobbiamo stare attenti anche in che ordine schedulerli. Quindi sopra abbiamo avuto un parallelismo. Una grande efficienza. I thread possono avere un'esecuzione correlata fra di loro, quindi schedulare i thread è più complicato.