

Sincronizzazione (concorrenza)

I programmi paralleli utilizzano meccanismi di sincronizzazione per prevenire interleaving indesiderati:

busy waiting il consumatore testa periodicamente il valore di una variabile condivisa (inefficiente);

sleep/wakeup il sottoprocesso va in *sleep* attendendo un segnale di *wakeup* da un altro sottoprocesso, tramite servizi messi a disposizione dal runtime/SO;

send/receive primitive di message passing/IPC – *receive* blocca il consumatore in attesa di un dato, *send* lo sblocca con il dato fornito;

lock/mutex mutua esclusione: la primitiva *lock* blocca/acquisisce una risorsa se è libera, altrimenti sospende il sottoprocesso finché non viene liberata (tramite *free/release*).

L'implementazione può avvenire a livello del runtime del linguaggio nel caso di thread, ma deve necessariamente essere gestita dal sistema operativo per i processi (che sono isolati).