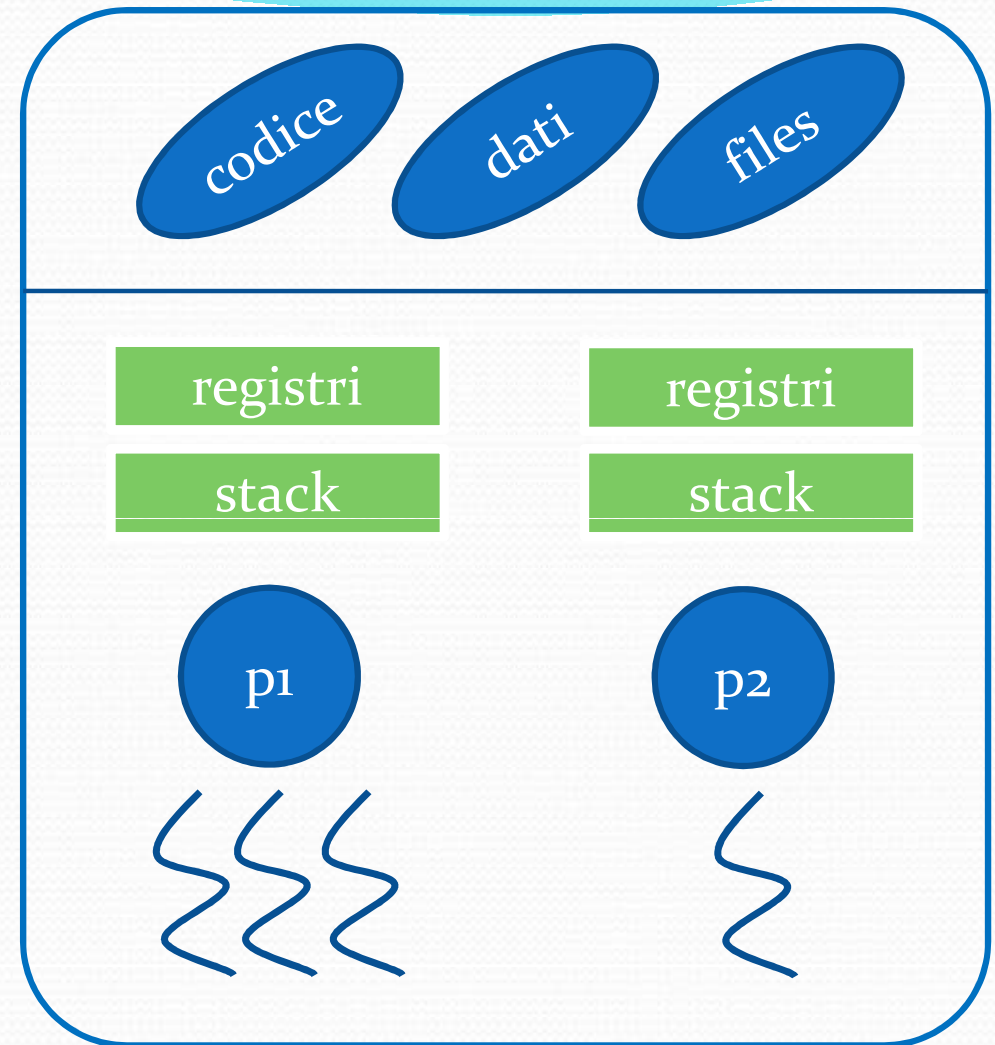


Programmazione concorrente in Java


Processi e Thread


- **Processo**: è l'istanza di un programma in esecuzione in un dato momento su una macchina
- **Thread**: unità di esecuzione all'interno di un processo con cui condivide le risorse (registri, spazio di indirizzamento)



Supporto del Sistema Operativo

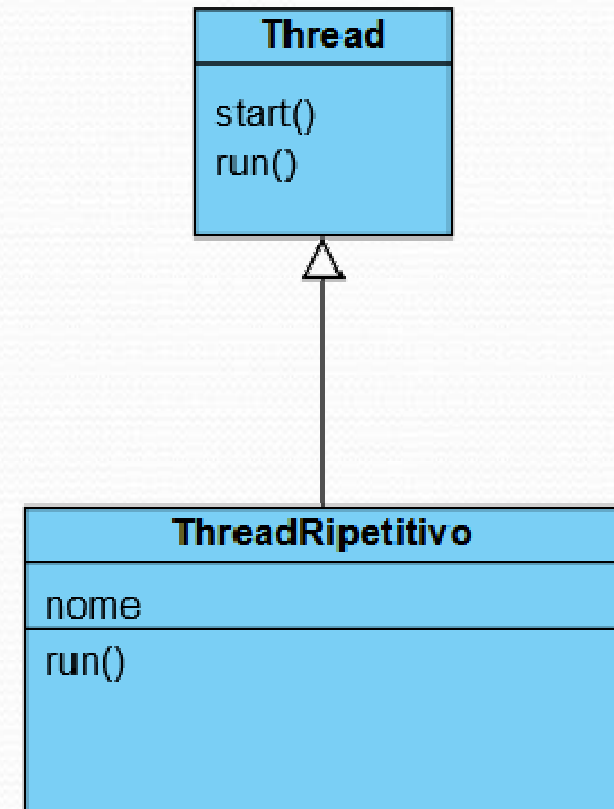
- La possibilità di avere più processi e/o thread in esecuzione dipende dalla funzionalità del sistema operativo installato oltre che dalle caratteristiche dell'Hardware
- I sistemi **monotasking** supportano la gestione di un solo processo/task per volta
- I sistemi **multitasking** supportano più processi, ma ogni processo può avere un solo thread
- Sistemi **multithreading** supportano più processi che possono essere eseguiti con più thread contemporaneamente

- 
- Nei sistemi multitasking il processore può eseguire le istruzioni di un processo per volta
 - Il passaggio da un processo ad un altro:
 - viene stabilito dal Sistema Operativo in base a delle **politiche di scheduling**
 - Comporta un **cambio di contesto** del processore, che è dispendioso in termini di tempo

- 
- I **thread** condividono alcune risorse con il **processo** padre, per cui il processore può essere assegnato a diversi thread dello stesso processo ma il **cambio di contesto** è molto più rapido rispetto al passaggio da un processo ad un altro

La classe Thread in Java

- per creare un thread si può creare una classe che eredita dalla **classe Thread** (vedremo in seguito che esiste anche un altro modo)



Funzionamento

- Il metodo `run()` della classe `Thread` viene sovrascritto della sottoclasse
- In fase di esecuzione, la creazione del thread è demandata al metodo `start()` che imposta la JVM per la creazione del thread ed esegue le istruzioni contenute nel metodo `run()`

