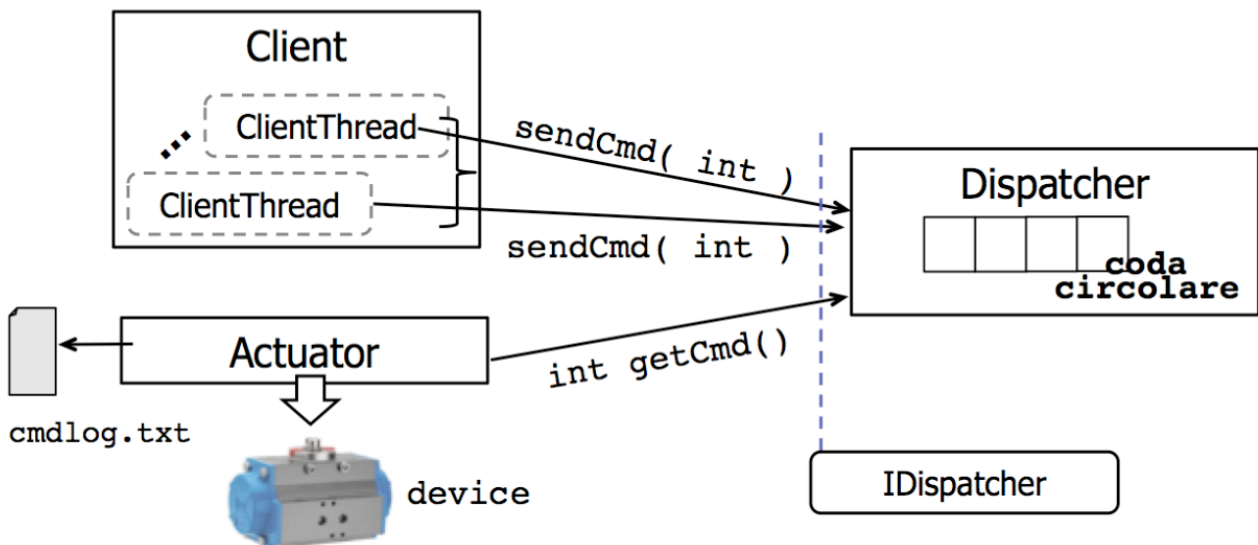


Università degli Studi di Napoli Federico II

Esercizio di Advanced Computer Programming

Il candidato realizzi un server multiprocesso in Python per il dispatching di comandi verso un device remoto. Il sistema è composto da 3 entità, illustrate in Figura:

1. **Dispatcher**, che offre due servizi: **sendCmd** e **getCmd**. Il servizio **sendCmd(command)** consente l'invio al Dispatcher del comando che sarà eseguito sul device; **command** identifica una specifica operazione sul device richiesta da **Client**. Il Dispatcher inserisce il comando ricevuto in una coda (di default la dimensione è pari a 5) nell'attesa che venga successivamente prelevato da **Attuator** tramite l'invocazione di **getCmd()**. L'accesso alla coda deve rispettare i vincoli del problema produttore-consumatore
2. **Client**: genera 5 thread, ciascuno dei quali, allo scadere di un tempo di t secondi (con t scelto a caso tra 2 e 4) invoca **sendCmd**, con **command** scelto a caso tra 0 e 3, avendo come riferimento le seguenti associazioni: 0-*leggi*, 1-*scrivi*, 2-*configura*, 3-*reset*. Ciascun thread effettua 3 richieste.
3. **Attuator**: Invoca **getCmd()** ogni secondo; **getCmd** estrae il comando dalla coda e lo restituisce ad Attuator. Se la coda è vuota, la richiesta **getCmd** è messa in attesa. Il comando restituito da **getCmd** è stampato a video ed in un log file denominato **cmdlog.txt**, specificando data ed etichetta del comando (e.g., 2024-04-12 11:13:06.216420 *scrivi*).



Il candidato implementi il sistema utilizzando le socket ed il pattern proxy/skeleton obbligatorio