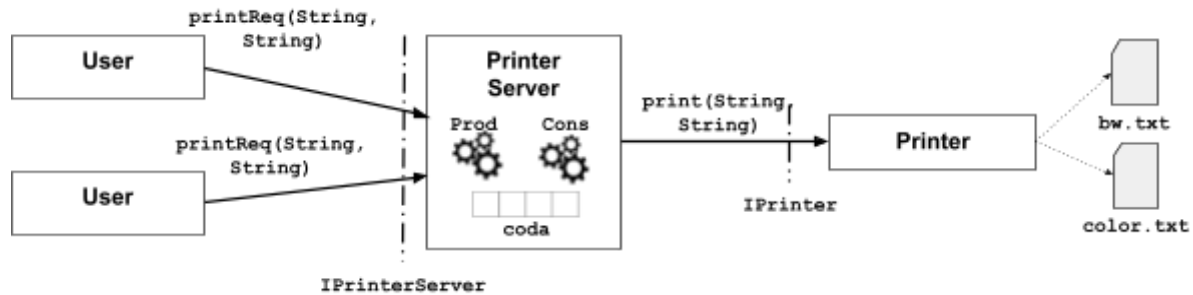


# Università degli Studi di Napoli Federico II

## Advanced Computer Programming



Il candidato implementi un sistema distribuito in **Python** per la gestione di richieste di stampa basato su **Socket**. Il sistema è caratterizzato dai seguenti componenti.

**User.** E' un client utilizzato per la richiesta di job di stampa al **Printer Server**. L'invio di una richiesta consiste nella invocazione del metodo `void print(String, int)` specificato nell'interfaccia **IPrinterServer**. La richiesta è caratterizzata da 1) **pathFile** (*String*), ossia il path del file da stampare, 2) **tipo** (*String*), ossia se la stampa deve essere in bianco/nero (bw), scala di grigi (gs) o a colori (color). Il Client genera 10 richieste di stampa, invocando il metodo `printReq` per ogni richiesta (attendendo 1 secondo tra le invocazioni). Per ciascuna richiesta, *tipo* è generato in maniera casuale scegliendo tra *bw*, *gs* e *color*, mentre *pathFile* è generato in maniera casuale, come `/user/file_{NUM}.{estensione}`, dove `{NUM}` è un valore numerico scelto casualmente tra 0 e 100, ed *estensione* è una stringa scelta a caso tra *doc* e *txt*.

**Printer Server.** Fornisce l'interfaccia **IPrinterServer** e il relativo metodo `printReq( String, String )`. Il metodo `printReq` avvia un processo produttore, il quale inserisce in una coda (process-safe e che implementi il problema del produttore/consumatore) una stringa che concatena sia la stringa del parametro *pathFile* che il parametro *tipo* (ad es., `/user/file_10.txt-bw`). I dati inseriti dalla coda, sono consumati da un processo consumatore avviato al lancio del Printer Server. Quando un nuovo dato è disponibile nella coda, il processo consumatore preleva la stringa, ed estrae i due parametri *pathFile* e *tipo* da utilizzare per l'invocazione del metodo `print` messo a disposizione da **Printer**.

**Printer.** Fornisce l'interfaccia **IPrinter**, ed il relativo metodo `print( String, String )`. Il metodo avvia un nuovo processo che stampa a video i due argomenti *pathFile* e *tipo*, e li scrive su un apposito file, scelto in base al campo *tipo*. Precisamente, la scrittura viene effettuata sul file `bw.txt` nel caso di *tipo* pari a *bw*, sul file `gs.txt` nel caso di *tipo* *gs*, o sul file `color.txt` nel caso di *tipo* pari a *color*.

**Il candidato utilizzi proxy-skeleton con socket TCP per la comunicazione sia tra User e Printer Server, che tra Printer Server e Printer.** A tal fine, il candidato predisponga le opportune interfacce e le classi Proxy-Skeleton. Si utilizzi inoltre skeleton per ereditarietà per il Printer Server e skeleton per ereditarietà per il Printer.