

Università di Ferrara

# Architettura di Reti

## Lab 2

Carlo Giannelli

carlo.giannelli@unife.it

<http://www.unife.it/scienze/informatica/insegnamenti/architettura-reti/>

<http://docente.unife.it/carlo.giannelli>

# Esercizio 1

Si realizzi, utilizzando la socket API, un'applicazione distribuita Client/Server che permetta di visualizzare i processi attivi sul server indicato. Il Client deve presentare la seguente interfaccia:

**rps host <option>**

dove "host" è il nome del server e il parametro "option" una singola opzione da passare al comando ps di Linux.

Il Client si dovrà connettere al Server, comunicandogli il parametro desiderato per la visualizzazione dei processi attivi (si veda «man ps» per la lista di tutti i parametri disponibili).

A sua volta, il Server si occuperà di lanciare il comando ps con il parametro ricevuto e di restituire al Client il risultato che dovrà visualizzarlo sullo standard output.

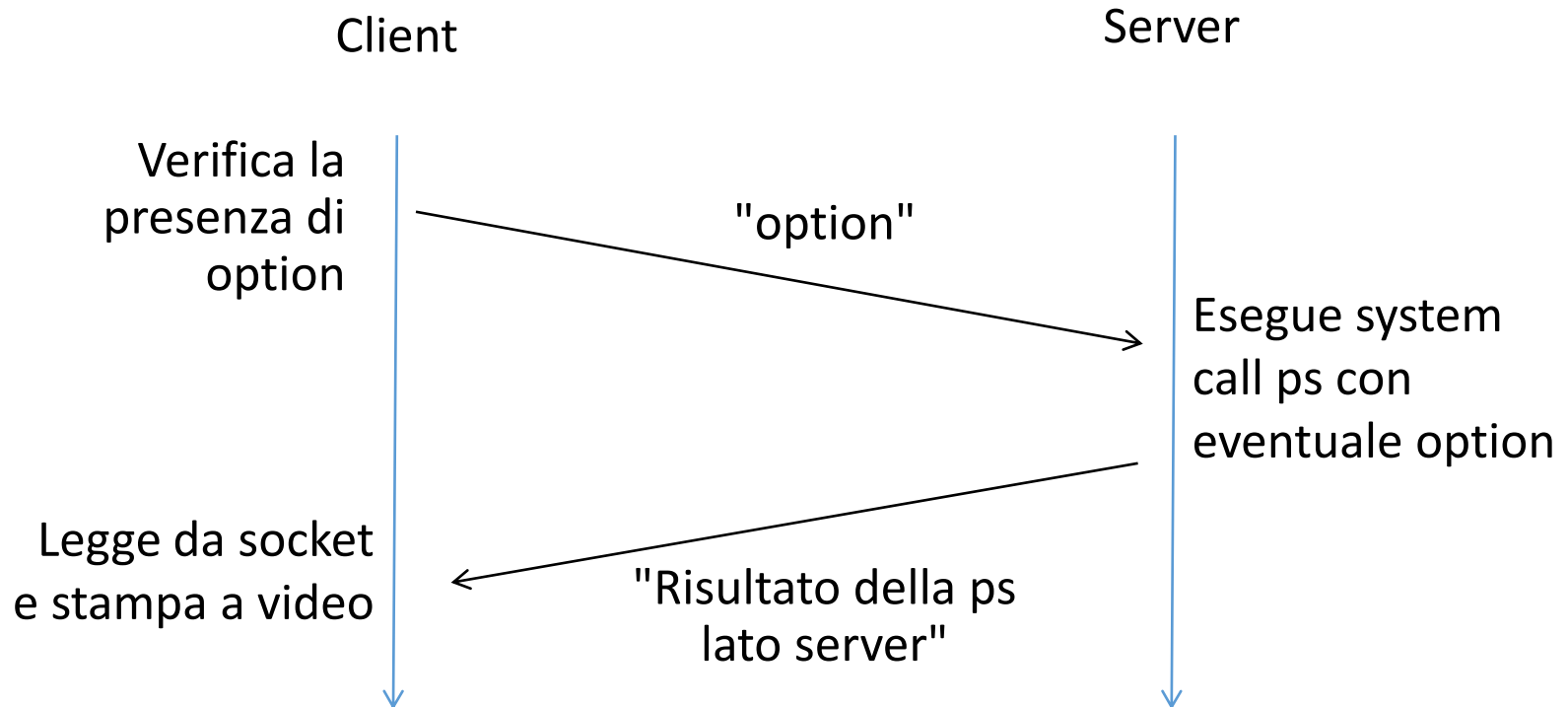
# Esercizio 1 (cont.)

L'output prodotto dal Client sarà del tipo:

```
luca@ubuntu: ~/Desktop/rps
luca@ubuntu:~/Desktop/rps$ ./c 127.0.0.1
connect riuscita al tentativo 1
Server 127.0.0.1:
  PID TTY          TIME CMD
 2150 pts/0        00:00:00 bash
 2857 pts/0        00:00:00 s
 2895 pts/0        00:00:00 ps
luca@ubuntu:~/Desktop/rps$ █
```

Si realizzi l'applicazione utilizzando il linguaggio C, sia per il Client che per il Server. Si ipotizzi che le stringhe siano sempre inviate in un unico segmento TCP.

# rps protocollo



**Attenzione: redirection, exec**

# Esercizio 2

Si realizzi, utilizzando la socket API, un'applicazione distribuita Client/Server che permetta di controllare la presenza di una specifica stringa di testo all'interno di uno specifico file nel filesystem del Server.

Il Client deve offrire la seguente interfaccia:

**rgrep hostname porta stringa nomefile**

dove "hostname" e "porta" sono rispettivamente il nome dell'host e il numero di porta su cui il processo Server è in ascolto e "stringa" e "nomefile" sono rispettivamente la stringa e il nome del file sul filesystem del Server a cui l'utente è interessato.

## Esercizio 2 (cont.)

Per prima cosa, il Client si dovrà connettere al Server, comunicandogli il nome del file a cui l'utente è interessato.

Il Server dovrà quindi verificare la presenza del file nel filesystem locale e terminare la connessione nel caso quest'ultimo non esista.

Il Client dovrà quindi comunicare la stringa al Server, che a sua volta dovrà restituire al Client il contenuto delle righe del file in cui compare la stringa richiesta. Una volta ricevute le informazioni dal Server, il Client dovrà stamparle a video e quindi terminare.

Si realizzi l'applicazione utilizzando il linguaggio Unix/C sia per il Client che per il Server.

# rgrep protocollo

