

Laboratorio – 1° Appello 2019 – 2/7/2019

Cognome	BRAVO
Nome	STUDENTE
Matricola	SOLUZIONE

Programmazione Socket (3 punti)

Il codice sotto riportato rappresenta una versione semplificata di un sistema di acquisti online. Il client richiede e visualizza il catalogo dei prodotti disponibili e chiede all'utente di indicare il codice del prodotto da comprare. Le rimanenze di un prodotto possono essere terminate, perciò il server deve verificare se il prodotto richiesto è disponibile in magazzino prima di dare conferma al client. Per motivi di sicurezza, ad ogni client è concesso un numero limitato di acquisti consecutivi.

Server

```
from socket import *

serverPort = 12000
listaProdotti = ['1 - Prodotto 1', '2 - Prodotto 2', '3 - Prodotto 3']
disponibilita = [0, 2, 3]
serverSocket = socket(AF_INET, SOCK_STREAM) # Welcome socket
serverSocket.bind(('', serverPort))
serverSocket.listen(1)

while 1:
    print('Server in attesa di connessioni')

    connectionSocket, addr = serverSocket.accept()

    fup = 0
    request = ""
    while request != '':
        if sum(disponibilita) == 0:
            request = connectionSocket.recv(2048)
            reply = 'CLOSE'
            connectionSocket.send(reply.encode('utf-8'))
            break
        else:
            request = connectionSocket.recv(2048)
            request = request.decode('utf-8')
            if request == 'CATALOGO':
                reply = ""
                for pr in listaProdotti:
                    reply += pr + '\n'
                connectionSocket.send(reply.encode('utf-8'))
            elif request.isdigit() and int(request) in range(1, len(listaProdotti)+1):
                pr_num = int(request)
                if disponibilita[pr_num-1] > 0:
                    disponibilita[pr_num-1] -= 1
                    reply = 'OK. Vuoi effettuare un altro acquisto?'
                    connectionSocket.send(reply.encode('utf-8'))
                    follow_up = connectionSocket.recv(2048)
```

```
        follow_up = follow_up.decode('utf-8')
        if follow_up.upper() == 'SI':
            if sum(disponibilita) == 0:
                reply = 'CLOSE'
            else:
                if fup < 1:
                    reply = 'FOLLOW_UP'
                    fup += 1
                else:
                    reply = 'STOP'
            else:
                reply = 'OK'
        else:
            reply = 'NO_STOCK'

        connectionSocket.send(reply.encode('utf-8'))

    elif request != '.':
        reply = 'KO'
        connectionSocket.send(reply.encode('utf-8'))
```

connectionSocket.close()

Client

```
from socket import *

serverName = "localhost"

serverPort = 12000

clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))

clientSocket.send("CATALOGO".encode('utf-8'))
elenco_prodotti = clientSocket.recv(1024)

reply = "KO"
while reply == "KO" or reply == 'NO_STOCK' or reply == 'FOLLOW_UP':

    if elenco_prodotti.decode('utf-8') == 'CLOSE':
        reply = elenco_prodotti.decode('utf-8')
        break
    else:
        print("Elenco dei prodotti disponibili:")
        print(elenco_prodotti.decode('utf-8'))

    request = input("Inserisci la tua scelta: ")

    clientSocket.send(str(request).encode('utf-8'))

    reply = clientSocket.recv(2048)
    reply = reply.decode('utf-8')
    if reply == "KO":
        print("Scelta non valida.")
```

```
elif reply == "NO_STOCK":
    print("Non ci sono rimanenze per il prodotto selezionato")
else:
    print("Acquisto andato a buon fine!")
    print(reply)
    request = input('Risposta (\SI\ per continuare): ')
    clientSocket.send(str(request).encode('utf-8'))
    reply = clientSocket.recv(2048)
    reply = reply.decode('utf-8')

if reply == 'STOP':
    print("Cominciare una nuova sessione per continuare gli acquisti")
if reply == 'CLOSE':
    print('Disponibilita terminate per ogni prodotto.')
print('Termina Sessione')
clientSocket.send(".".encode('utf-8'))
clientSocket.close()
```

Q1. Completare il codice del Client. (1 punto) Vedi Codice

Q2. Cosa succede se alla domanda del server ‘Vuoi effettuare un altro acquisto?’ l’utente risponde con una stringa tra ‘Si’, ‘sI’ oppure ‘si’? (1 punto)

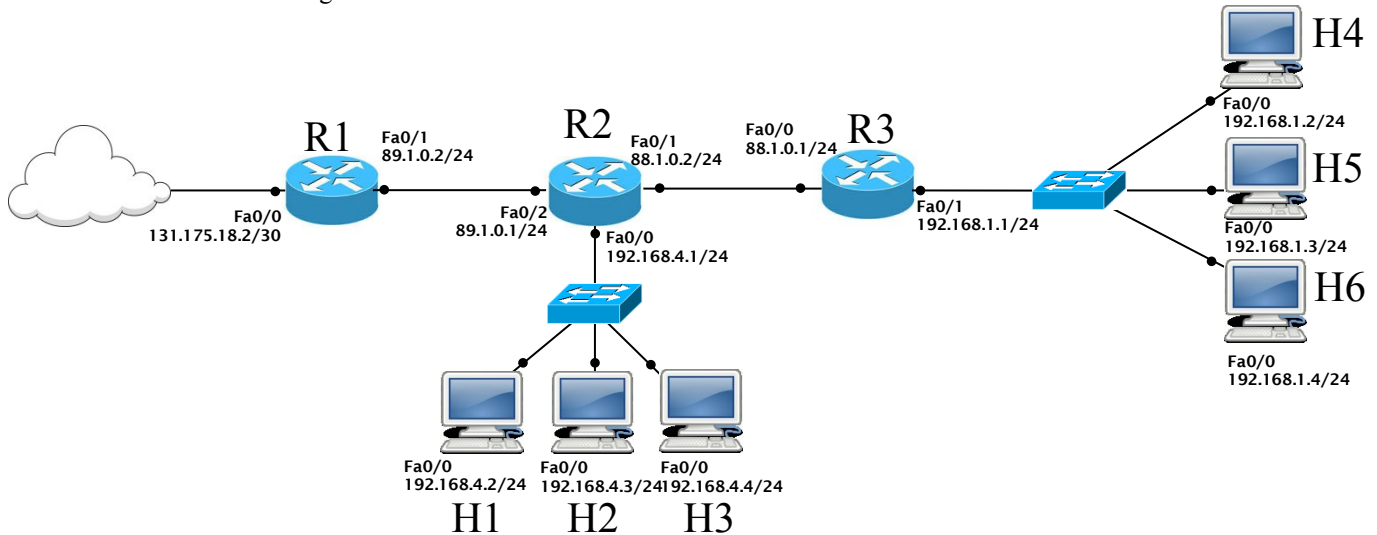
Il server riconosce comunque la scelta, dato che converte il messaggio ricevuto in maiuscolo e lo confronta con la scritta SI.

Q3. Quanti acquisti consecutivi può fare ogni utente? Da cosa si capisce? (1 punto)

Ogni utente può fare al massimo 2 acquisti consecutivi: il controllo è effettuato tramite la variabile ‘fup’, che diventa 1 dopo che l’utente completa il secondo acquisto, determinando perciò la conclusione della sessione.

Packet Tracer (3 punti)

Si consideri la rete in figura



Attenzione:

- Indirizzi IP e gateway sono già stati configurati per gli host H1, H2, H3, H4, H5 e H6
- Tutte le interfacce dei router sono state configurate
- Indicare sempre prima del comando il prompt visualizzato dal sistema, prestando attenzione alla modalità di partenza in ciascuna richiesta

- 1) Configurare il routing statico sul router R2 in modo che gli host H1, H2 e H3 possano raggiungere la rete privata 192.168.1.0 e la rete Internet. (1,5 punti)

```
R2> enable
R2# configure terminal
R2(config)# ip route 192.168.1.0 255.255.255.0 88.1.0.1
R2(config)# ip route 0.0.0.0 0.0.0.0 89.1.0.2
```

Si supponga ora che il routing sia già stato configurato sui rimanenti dispositivi

- 2) Abilitare il NAT sul router R3 per i pacchetti provenienti dalla rete degli host H4, H5, H6 (utilizzare 1 come ID della lista d'accesso). (1,5 punti)

```
R3(config)# interface Fa0/1
R3(config-if)# ip nat inside
R3(config-if)# exit
R3(config)# interface Fa0/0
R3(config-if)# ip nat outside
R3(config-if)# exit
R3(config)# access-list 1 permit 192.168.1.0 0.0.0.255
R3(config)# ip nat inside source list 1 interface Fa0/0 overload
```