

## Piattaforme Software (Modulo 2) 29/06/2011

### Esercizio 1-a

Determinare l'output del seguente script bash, invocato a partire dalla cartella `/home/cognome/nome` e con i parametri *cognome* e *matricola* (dove le parti in corsivo sono sostituite con il proprio nome, cognome o numero di matricola).

```
#!/usr/bin/bash
bar() {
    while [[ $3 ]] ; do echo test $3; shift ; done
}
IFS="/"
echo `pwd`
bar $2 $(pwd)
```

### Esercizio 1-b

Scrivere la dichiarazione XDR da passare a *rpcgen* per trasformare l'invocazione della funzione `lookup` in chiamata remota, e la traduzione in C delle strutture dati e dei prototipi di funzione definiti da *rpcgen*, data in input la dichiarazione stessa. **Bonus:** implementare anche il server.

```
#include <stdio.h>
#include <stdlib.h>

char* lookup(char *filename, int max, double value){
    FILE *fin = fopen(filename,"r");
    char *current=malloc(sizeof(char)*256);
    double fvalue=0.0;
    int len=0,i=0;
    current[0]=0;
    for(i=0; i<max ; i++){
        fscanf(fin,"%s : %f",current,&fvalue);
        if(((int) value)==((int) fvalue)) break; }
    if (i>=max) strcpy(current,"not found");
    return current;
}

int main(int argc, char **argv){
    printf("%s\n", lookup(argv[1],atoi(argv[2]),atof(argv[3])));
}
```

## Esercizio 2

Progettare un sistema client-server in cui piu' client possono connettersi ad un server noto, ed eseguire la funzionalita' descritta dalla funzione *lookup* dell'**Esercizio 1-b**, comunicando pero' attraverso il protocollo *http*. Il server riceve i parametri con il metodo GET o POST (a scelta dell'implementatore) e invia come risposta la stringa restituita da *lookup*.

Implementare solo il server.

## Esercizio 3

Indicate quali regole sono scattate (inclusi gli hit sulle policy) e quante volte dopo il passaggio dei seguenti pacchetti, assumendo che l' host abbia 2 interfacce di rete, una esposta con indirizzo pubblico 131.175.1.2, l' altra verso una rete interna con indirizzo 10.0.0.42. Il forwarding dei pacchetti ip è abilitato.

```
1 - 10.0.0.7:12943 -> 10.0.0.42:123, TCP
2 - 10.0.0.42:12943 -> 10.0.0.42:80, TCP
3 - 10.0.0.3:12943 -> 8.8.8.8:53, TCP
4 - 10.0.0.3:12943 -> 10.0.0.4:80, TCP
5 - 46.59.1.2:12943 -> 10.0.0.42:80, TCP
6 - 10.0.0.3:12943 -> 46.59.1.2:80, TCP
7 - 10.0.0.3:12943 -> 8.8.8.8:53, UDP
8 - 10.0.0.4:12943 -> 209.85.148.106:80, TCP
9 - 10.0.0.3:12943 -> 8.8.8.8:53, TCP
10 - 10.0.0.4:12943 -> 8.8.8.8:53, TCP
11 - 10.0.0.42:12943 -> 8.8.8.8:53, TCP
12 - 10.0.0.12:12943 -> 209.85.148.106:80, TCP
```

Come bonus, indicate quali regole sono inutili (ovvero quali regole potete rimuovere senza cambiare il comportamento della macchina), indipendentemente dal traffico che transita.

```
iptables -A INPUT -s 10.0.0.3 -j LOG
iptables -A INPUT -i lo -j ACCEPT
iptables -P FORWARD ACCEPT
iptables -A FORWARD -i lo -j ACCEPT
iptables -A INPUT -s 10.0.0.4/24 -j RATEEST
iptables -A FORWARD -d 46.59.1.2 -j ACCEPT
iptables -I INPUT 2 -s 10.0.0.3 -j DROP
iptables -I INPUT 2 -s 10.0.0.3 -j QUEUE
iptables -P OUTPUT ACCEPT
iptables -A INPUT -d 10.0.0.42 -j ACCEPT
iptables -A INPUT -s 10.0.0.7 -j DROP
iptables -A INPUT -s !10.0.0.3 -j ACCEPT
iptables -P INPUT DROP
iptables -A FORWARD -s 10.0.0.4/24 -j RATEEST
iptables -A FORWARD -s 10.0.0.4/24 -j ACCEPT
iptables -A FORWARD -p tcp --dport 40:100 -j LOG
```