Politecnico di Milano
FACOLTÀ DI INGEGNERIA DELL'INFORMAZIONE

Corso di Piattaforme Software per la rete – MODULO 2
anno accademico 2013-2014

Prof. William FORNACIARI

Traccia di soluzione del 1.06.2015

Quesito D1

Si descriva come funziona uno scheduler base per processi in sistemi UNIX-like, dove le si hanno code a priorità multipla variabile ciascuna gestita in Round Robin.

- Quali sono gli obiettivi di tale scheduler?
- Come si comportata per processi CPU-bound e come per quelli I/O bound interattivi?
- Quali potrebbero essere utili ausili hardware/Instruction Set a supporto del calcolo di tali politiche di scheduling in modo da ridurre l'overhead del kernel del Sistema Operativo?

Vedi appunti lezione e libro

- *Reattività, fairness, assenza di starvation*
- *Code in modo user e modo sistema*
- *….*

Quesito D2

Con riferimento all' implementazione del sistema di segnali di Linux, si risponda alle seguenti domande argomentando brevemente la risposta :

1. E' corretto dire che il codice del gestore del segnale nel processo che lo riceve viene eseguito non appena esso viene inviato?
2. E' corretto dire che il codice del gestore del segnale nel processo che lo riceve viene eseguito non appena esso notificato dal sistema operativo, ovvero non appena il sistema operativo marca l' esistenza all' interno del signal vector del processo?
3. E' possibile che vi sia un ritardo tra la notifica di un segnale KILL e l' effettiva terminazione del processo? Se sì, dare un' esempio di situazione in cui questo accade

Risposte:

1. No. Il gestore del segnale viene eseguito non appena il processo ricevente viene schedulato
2. No. Come nel caso precedente, il gestore viene eseguito non appena il processo viene schedulato. Questo non accade necessariamente appena dopo la consegna.
3. Sì. Nel caso in cui il processo sia bloccato in attesa della terminazione di un' azione di input/output da periferica, il segnale di KILL non ha effetto immediato.

Quesito D3

Realizzare in C un client QOTD (quote of the day) TCP. Il servizio QOTD (porta 17)
invia a qualunque client si connetta una stringa di al più 512 caratteri contenente una citazione famosa. Il client QOTD deve connettersi a un indirizzo specificato come primo parametro da commandline, e stampare a video la citazione ricevuta.

```
#include <unistd.h>
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/ip.h>
#include <netinet/tcp.h>

int main(int argc, char *argv[])
{
    int sock;
    int received = 0;
    struct sockaddr_in dest_addr;
    char buffer[512];

    if ( (sock = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
        perror("Socket creation error");
        return 1;
    }

    memset((void *) &dest_addr, 0, sizeof(dest_addr));
    dest_addr.sin_family = AF_INET;
    dest_addr.sin_port = htons(17);

    if ( (inet_pton(AF_INET, argv[1], &dest_addr.sin_addr)) <= 0) {
        perror("Address creation error");
        return -1;
    }

    if ( connect(sock,(struct sockaddr*) &dest_addr
                sizeof(dest_addr)) < 0){
         perror("Error connecting to server : ");
         return 1;
    }

    received =  recv(sock, &buffer, 512, 0);
    buffer[received] ='\0';
    puts(buffer);
    return 0;
}
```

Quesito D4

Dato un host con 2 interfacce di rete:

- eth0: esposta con indirizzo pubblico 131.175.1.2/24
- eth1: verso una rete interna con indirizzo 192.168.0.1/24 .

in cui il forwarding dei pacchetti ip è stato già abilitato, si indichino i comandi necessarie per configurare il sistema di filtraggio dei pacchetti in modo tale da:

1. Impostare correttamente le policy per ogni tabella
2. Consentire l' accesso al server SSH (porta 22/TCP) presente sull' host da ovunque
3. Far sì che l' host agisca come source NAT per tutti quelli connessi alla rete interna, e dunque appartenenti alla sottorete 192.168.0.0/24
4. Consentire al traffico proveniente dall' esterno di accedere al server HTTPS (porta 443/TCP) presente su un host con indirizzo 192.168.0.13 effettuando traduzione di indirizzo di destinazione (destination NAT)

Risposta:

```
   1. iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
   2. iptables -A INPUT -p tcp --dport 22 -j ACCEPT
   3. iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -j SNAT --to
      131.175.1.2
   4. iptables -t nat -A PREROUTING -p tcp –dport 443 -j DNAT --to
      192.168.0.13:443
```