



Politecnico di Milano
FACOLTÀ DI INGEGNERIA DELL'INFORMAZIONE

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

29 Luglio 2013

Prof. William FORNACIARI

TRACCIA DI SOLUZIONE (29 luglio)

Quesito D1

D1.1 Dare una definizione di **deadlock** e descrivere l'algoritmo del **banchiere**, spiegando esplicitamente cosa si intenda per **stato sicuro**.

D1.2 Si supponga di avere un sistema composto da 5 processi (P0, P1, P2, P3, P4), 3 tipi di risorsa (A, B, C) con le seguenti cardinalità: A=10, B=5, C=7.

La situazione all'istante T0 sia la seguente:

T0	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	3	3	2
P1	2	0	0	3	2	2			
P2	3	0	2	9	0	2			
P3	2	1	1	2	2	2			
P4	0	0	2	4	3	3			

Nell'ipotesi che vi siano le seguenti richieste di risorse, **dimostrare** se il sistema si trova o meno in uno **stato sicuro**, **motivando la risposta**

	Richieste		
	A	B	C
P0	7	4	3
P1	1	2	2
P2	6	0	0
P3	0	1	1
P4	4	3	1

D1.3 Facoltativo

- Quali sono le tipologie di applicazioni che risentono di meno dei problemi di utilizzo delle risorse originati dall'algoritmo del banchiere? Fornire un esempio motivato.

Risposta D1

Il sistema si trova in uno stato sicuro poiché esiste almeno la seguente sequenza di esecuzione <P1, P3, P4, P2 e P0> che soddisfa la definizione di stato sicuro.

Quesito D2

Dato un processo attivo con **EUID=1000 RUID=1000 SUID=0**, lanciato dall'utente con **user id 1000** quali sono gli effetti:

1. dell'esecuzione di una chiamata a funzione del tipo **seteuid(300)**
 2. dell'esecuzione, successivamente alla precedente, di una chiamata **seteuid(0)**
 3. del tentativo di leggere il seguente file
-rw----- 1 300 100 0 Jul 13 19:38 readme.txt
- i) tra la prima e la seconda invocazione di funzione;
 - ii) dopo l'esecuzione della seconda

Risposta D2

i) tra la prima e la seconda invocazione di funzione:

L'invocazione di seteuid non va a buon fine in quanto, al momento dell'invocazione, 300 non è il saved-user-id del processo e il processo non ha diritti di cambiare arbitrariamente l'euid. Di conseguenza, il processo non riesce a leggere il file in questo caso

ii) dopo l'esecuzione della seconda:

L'esecuzione della seconda chiamata cambia l'euid del processo con successo in 0, con la conseguenza che la lettura va a buon fine, avendo il processo acquisito diritti di amministrazione.

Quesito D3

Nella configurazione della vostra macchina vi trovate a dover decidere quale di queste due regole di IpTables utilizzare per consentire il traffico in ingresso sull'interfaccia di loopback.

Regola 2 : iptables -A INPUT -i lo -j ACCEPT

Regola 1 : iptables -A INPUT -s 127.0.0.1/8 -j ACCEPT

Regola 1 : iptables -A PREROUTING -s 127.0.0.1/8 -j ACCEPT

Le tre regole sono equivalenti da un punto di vista funzionale?

Qual è il modo corretto di approcciare?

Risposta D3

Le tre regole NON sono equivalenti da un punto di vista funzionale : il modo corretto di approcciare il problema dell'accettazione dei pacchetti provenienti dalla stessa macchina è l'accettare tutto ciò che transita sull'interfaccia di loopback.

Quesito D4

Dato un buffer circolare i cui elementi contengono un nome di persona (rappresentato come una vettore di caratteri da 40 elementi) quale strategia per controllare se il buffer è pieno conviene adottare?

Come conviene cambiare approccio nel caso in cui gli elementi siano della dimensione di qualche centinaio di MB?

Motivare la risposta, eventualmente fornendo esempio di codice se ritenuto utile

Risposta D4

Nel primo caso, gli elementi del buffer circolare sono di dimensioni ridotte, è dunque possibile adottare la politica di lasciare uno slot sempre vuoto per risolvere il problema del controllo dello stato di riempimento del buffer. Nel secondo caso, date le dimensioni dei singoli elementi, la politica precedente non è una buona scelta: una qualunque tra le politiche alternative a costo in spazio costante è una scelta ragionevole.