Protezione nei sistemi operativi

Lezione 14 di Sicurezza dei sistemi informatici 1

Docente: Giuseppe Scollo

Università di Catania, sede di Comiso (RG)
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Studi in Informatica applicata, AA 2006-7

Indice

problemi di sicurezza di sistemi operativi

alcune categorie di programmi presentano problemi specifici di sicurezza che hanno rilevanza **critica**, perché da essi dipende la sicurezza di (molti) altri programmi e dati

esempi precipui di tali categorie di programmi:

- sistemi operativi
- sistemi di gestione di basi di dati

studiamo la sicurezza nei **sistemi operativi** in due stadi successivi, corrispondenti a due diversi punti di vista:

- l'**uso** del sistema: oggetti della protezione e metodi per realizzarla
- la **progettazione** di sistemi operativi sicuri

i sistemi operativi odierni supportano **multiprogrammazione e time sharing** delle risorse fra **più utenti:**

- i meccanismi di protezione necessari riguardano dunque non solo l'interferenza fra utente e sistema, ma anche anche quella fra utenti diversi, o fra programmi diversi di uno stesso utente

oggetti della protezione

multiprogrammazione e time sharing hanno dato origine alla necessità di protezione di varie categorie di risorse da parte di un sistema operativo, delle quali esso deve permettere una **condivisione controllata:**

- memoria
- periferiche condivisibili (dischi, etc.)
- periferiche riutilizzabili in serie (nastri, stampanti, etc.)
- risorse di comunicazione, reti
- software (di sistema, utilità, etc.)
- dati condivisibili

metodi di protezione

base della protezione è la **separazione** degli oggetti di un utente da quelli degli altri utenti (e del sistema); la separazione può essere:

- **fisica**
- **temporale**
- **logica**
- **crittografica**

in ordine di complessità crescente, rigidità decrescente, e per le prime tre di sicurezza decrescente

i metodi di protezione offerti dal sistema operativo sono inoltre, e ortogonalmente, da classificare in base al supporto che forniscono alla **condivisione** di oggetti:

- nessuna protezione: condivisione incontrollata
- isolamento: nessuna condivisione
- condivisione sì/no per oggetti (pubblici/privati)
- condivisione mediante controllo degli accessi di soggetti a oggetti
- condivisione mediante lista delle capacità (ingl. capabilities) dei soggetti
- condivisione mediante limitazione dell'uso di oggetti: generalizza le precedenti

protezione della memoria

problema: confinamento delle azioni di ciascun programma alla propria area di memoria

soluzioni: di vario tipo, tutte dotate di **supporto hardware**; eccone alcune, in ordine di complessità crescente e rigidità decrescente:

- **recinto:** un indirizzo che separa lo spazio di sistema dallo spazio di utente, adatto solo a sistemi con un solo utente
- **registro recinto:** come sopra, ma il recinto è il contenuto di un registro, dunque modificabile dinamicamente
- **riposizionamento:** il caricamento dei programmi in memoria può porli in qualsiasi posizione, poiché gli indirizzi (di dati e istruzioni) sono **offset**, cioè aggiunti a un indirizzo costante determinato dal caricatore e memorizzato in un registro recinto di **base**
- **registri di base e di confine:** il registro recinto di base impedisce lo sconfinamento di un programma ad un indirizzo inferiore; l'aggiunta di un **registro di confine** serve ad impedire lo sconfinamento oltre il limite superiore dell'area assegnatagli
  un controllo più fine è possibile se si dispone di **due coppie** di registri di base e di confine per ciascun programma in esecuzione: una coppia per i dati, l'altra per le istruzioni; in tal modo si può impedire ad es. che, per un errore di codifica, il programma effettui operazioni di scrittura nella propria area istruzioni

architettura con etichetta

l'uso di registri di base e di confine per la protezione della memoria, sebbene sia efficace, limita la **condivisione** di dati fra programmi in esecuzione

- un problema è costituito dalla **contiguità** dell'area di memoria individuata dai suddetti registri: se dati in quest'area devono essere condivisibili, dev'esserlo tutta l'area, per la quale vale dunque una forma di condivisione sì/no

l'**architettura con etichetta** (ingl. **tagged architecture**) permette forme più articolate di condivisione, in quanto **ogni parola di memoria** è dotata di uno o più bit di etichetta, il cui valore ne specifica i diritti di accesso

- l'etichetta è modificabile solo da **istruzioni privilegiate** (del sistema operativo)

in alcune varianti si è adoperata un'etichetta per un blocco contiguo di indirizzi

- nell'architettura Intel I960 si aveva un'etichetta di un bit per distinguere le parole "normali", contenenti istruzioni o dati, da parole "speciali", che codificano i diritti di accesso per il susseguente blocco di parole normali

le architetture con etichetta non hanno avuto molta diffusione, per problemi di compatibilità con codice già sviluppato, e perché superate dagli approcci descritti appresso

segmentazione e paginazione

segmentazione: suddivisione del programma in **segmenti**, cioè parti, ciascuna delle quali è una sua unità logica (il codice di un sottoprogramma, una struttura dati, etc.) e ha memorizzazione contigua

- segmenti diversi hanno in genere lunghezza diversa

la segmentazione può dividere un programma in qualsiasi numero di parti con diritti di accesso diversi

altri **vantaggi della segmentazione:**

- i segmenti sono **rilocabili:** gestione più efficiente della memoria
- i segmenti non in uso corrente possono essere trasferiti in memoria di massa: **virtualizzazione della memoria**
- i riferimenti a indirizzi nel segmento sono offset, che richiedono l'intervento del sistema operativo per essere completati: opportunità di **controllo e protezione**

paginazione: suddivisione del programma in **pagine** di lunghezza fissa, indipendenti dal contenuto logico

- vantaggiosa per la semplicità ed efficienza di gestione dell'allocazione di memoria
- conveniente per l'allocazione dinamica di memoria a strutture dati di dimensione variabile a tempo di esecuzione

i vantaggi della segmentazione e della paginazione possono essere **combinati** in approcci misti, ad es. con la **paginazione sulla segmentazione**, cioè allocando pagine di memoria a segmenti di programma