

MECCANISMI E POLITICHE DI PROTEZIONE

Protezione

- Obiettivi della Protezione
- Dominio di Protezione
- Matrice di Accesso
- Implementazione della Matrice di Accesso
- Revoca dei Diritti di Accesso
- Sistemi basati su Abilitazioni
- Protezione basata sul Linguaggio

Protezione

- Un sistema operativo deve definire meccanismi e politiche per controllare gli accessi a tutte le risorse che esso gestisce.
- Utenti, programmi e processi devono essere controllati nell'accesso a dispositivi, file, programmi, informazioni.
- Le politiche di protezione definiscono cosa controllare, i meccanismi definiscono i modi per farlo.
- Protezione e sicurezza sono due aspetti di uno stesso problema.

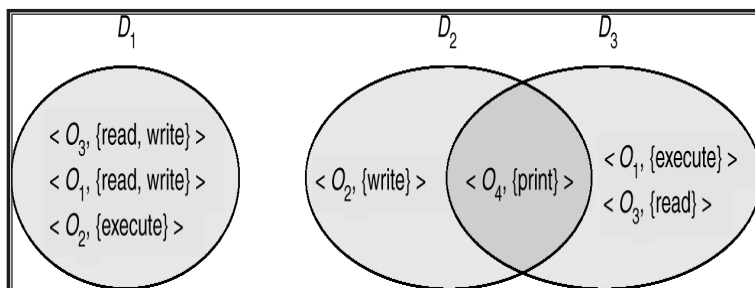
Protezione

- Un sistema di elaborazione consiste di un insieme di oggetti hardware (CPU, memoria, mouse, ...) o software (file, compilatori, programmi utente, ...).
- Ogni oggetto ha un identificatore unico e può essere acceduto/usato tramite un insieme di operazioni ben definite.
- *Problema della **Protezione*** - assicurare che ogni oggetto sia acceduto/usato correttamente e solo dai processi che hanno il permesso di farlo.

Dominio di protezione

- **Privilegio minimo:** un processo può accedere solo alle risorse che sono necessarie per svolgere il suo compito.
- **Dominio di protezione D :** insieme delle risorse a cui un processo può accedere.
- In un dominio per ogni oggetto usato sono indicati:
<nome_oggetto, diritti_di_accesso>
- **Diritti di accesso :** sottoinsieme delle operazioni possibili.
- **Dominio di protezione D = insieme di oggetti con i i diritti di accesso.**

Dominio di protezione



- Associazione statica o dinamica.
- Diversi livelli di dominio: utente, applicazione, processo, metodo.

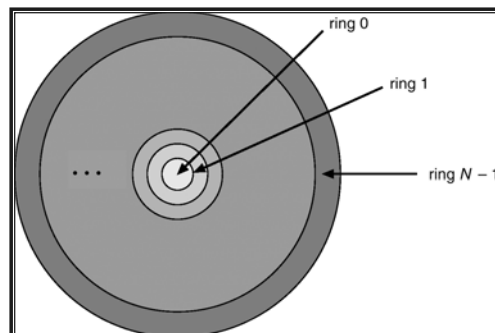
Esempio di domini in UNIX

- Il sistema consiste di due domini:
 - *User*
 - *Supervisor*

- UNIX
 - Dominio = *user-id*
 - Nell'accesso ai file il cambio di dominio è svolto dal file system.
 - ❖ Ad ogni file è associato un bit di dominio (*setuid bit*).
 - ❖ Quando un file viene acceduto e il *setuid = on*, allora lo *user-id* è modificato assegnandolo al valore del proprietario del file. Al termine delle operazioni lo *user-id* viene riportato al valore precedente.

Esempio di domini in Multics

- I domini sono gerarchicamente organizzati ad anelli.
- Siano D_i e D_j due anelli di dominio.
- Se $j < i \Rightarrow D_i \subseteq D_j$: **gli anelli più interni hanno maggiori privilegi.**



Anelli in Multics

Esempio di domini in Multics

- Il cambio di dominio in MULTICS avviene passando da un anello ad un altro: ad esempio nell'invocazione di un processo o di una procedura di un livello diverso.
- La protezione ad anelli di MULTICS non implementa la politica del privilegio minimo.
- Il meccanismo di protezione di MULTICS è complesso e non molto efficiente.
- Se gli anelli sono solo due è simile al modello di UNIX.

Matrice di Accesso

- In questo modello la protezione viene realizzata tramite una matrice detta **Matrice di accesso**.
- Le **righe** della matrice rappresentano i **domini**.
- Le **colonne** della matrice rappresentano gli **oggetti**.
- Un elemento **$access(i, j)$** è l'insieme delle operazioni che il processo che esegue nel dominio **D_i** può eseguire sull'oggetto **O_j** .

Matrice di Accesso

| object domain | F_1 | F_2 | F_3 | printer |
|------------------|---------------|-------|---------------|---------|
| D_1 | read | | read | |
| D_2 | | | | print |
| D_3 | | read | execute | |
| D_4 | read write | | read write | |

Se un processo in D_i tenta di operare su un oggetto O_j , l'operazione, per poter essere eseguita, deve essere presente nel dominio.

Uso della Matrice di Accesso

- L'uso della matrice di accesso permette di separare i meccanismi di protezione dalle politiche di protezione.
 - Meccanismi
 - ❖ Il sistema operativo fornisce matrice di accesso + regole.
 - ❖ Deve assicurare che i diritti di accesso specificati nella matrice di accesso non vengano osservati.
 - Politiche
 - ❖ Gli utenti e il supervisore definiscono le politiche.
 - ❖ Quando un oggetto viene creato o diviene attivo possono essere definite gli opportuni diritti di accesso.

Uso della Matrice di Accesso

- Può essere estesa per una protezione dinamica.
 - Definizione dei passaggi da un dominio all'altro (*switch*).
 - Operazioni sulla matrice di accesso per aggiungere e cancellare diritti di accesso.
 - Diritti per operazioni speciali sulla matrice di accesso:
 - ❖ *owner di O_i*
 - ❖ *copy operazione da O_i a O_j*
 - ❖ *control – D_i può modificare i diritti di accesso di D_j*
 - ❖ *switch – cambio di dominio da D_i a D_j*

Matrice di Accesso con domini come oggetti

| object \ domain | F_1 | F_2 | F_3 | laser printer | D_1 | D_2 | D_3 | D_4 |
|-----------------|---------------|-------|---------------|---------------|--------|--------|--------|--------|
| D_1 | read | | read | | | switch | | |
| D_2 | | | | print | | | switch | switch |
| D_3 | | read | execute | | | | | |
| D_4 | read write | | read write | | switch | | | |

- Dal dominio **D_1** si può passare al dominio **D_2** .
- Dal dominio **D_2** si può passare ai domini **D_3** e **D_4** .
- Dal dominio **D_4** si può passare al dominio **D_1** .

Matrice di Accesso con diritti di Copy

| object \ domain | F_1 | F_2 | F_3 |
|-----------------|---------|-------|---------|
| D_1 | execute | | write* |
| D_2 | execute | read* | execute |
| D_3 | execute | | |

(a)

| object \ domain | F_1 | F_2 | F_3 |
|-----------------|---------|-------|---------|
| D_1 | execute | | write* |
| D_2 | execute | read* | execute |
| D_3 | execute | read | |

(b)

Matrice di Accesso con diritti di Owner

| object \ domain | F_1 | F_2 | F_3 |
|-----------------|------------------|----------------|--------------------------|
| D_1 | owner execute | | write |
| D_2 | | read* owner | read* owner write* |
| D_3 | execute | | |

(a)

| object \ domain | F_1 | F_2 | F_3 |
|-----------------|------------------|--------------------------|--------------------------|
| D_1 | owner execute | | |
| D_2 | | owner read* write* | read* owner write* |
| D_3 | | write | write |

(b)

Matrice di Accesso con diritti di *Control*

| object domain \ | F_1 | F_2 | F_3 | laser printer | D_1 | D_2 | D_3 | D_4 |
|--------------------|-------|-------|---------|------------------|--------|--------|--------|-------------------|
| D_1 | read | | read | | | switch | | |
| D_2 | | | | print | | | switch | switch control |
| D_3 | | read | execute | | | | | |
| D_4 | write | | write | | switch | | | |

Es.: Dal dominio D_2 si può modificare i diritti del dominio D_4 .

Implementazione della Matrice di Accesso

- La matrice di accesso è sparsa (contiene molti elementi vuoti).
- Metodi più efficienti della tabella globale:
- Ogni colonna è una **Lista di accesso per oggetti**
 - Definisce i domini che possono usare gli oggetti
Esempio: per F_1
 $D_1 = Read$
 $D_4 = Write$
- Ogni riga è una **Lista di abilitazioni per domini**
 - Per ogni dominio sono elencati gli oggetti con i diritti
Esempio: per D_3
 $F_2 = Read$
 $F_3 = Execute$

Revoca dei Diritti di Accesso

Se si usa :

- *Lista di accesso* – Si cancellano i diritti di accesso dalla lista.
 - Semplice
 - Immediato

- *Lista di abilitazioni* – Occorre trovare le abilitazioni che sono distribuite sui domini prima di poterli revocare.
 - Riacquisizione
 - Puntatori all'indietro
 - Indirizione
 - Chiavi

Sistemi basati su Abilitazioni

- **Hydra**
 - Insieme fissato di diritti di accesso conosciuti e interpretati dal sistema.
 - Definizione di diritti user-defined per essere usati da programmi utente; il sistema fornisce la protezione di accesso nell'uso di questi diritti.

- **Cambridge CAP System**
 - *Abilitazioni sui dati* -fornisce diritti standard *read, write, execute* di segmenti di memoria associati agli oggetti.
 - *Abilitazioni software* -interpretazione effettuata tramite procedure protette.

Protezione basata sul linguaggio

- La specifica della protezione in un linguaggio di programmazione permette la descrizione di alto livello di accesso ed uso di risorse.
- L'implementazione di un linguaggio permette di realizzare protezioni software quando altri meccanismi non sono disponibili.
- Si possono generare chiamate a meccanismi di protezione di basso livello (s.o. o hardware) quando queste sono disponibili.

Protezione in Java

- In Java la protezione è gestita dalla Java Virtual Machine (JVM).
- Ad ogni classe è assegnato un dominio di protezione caricato dalla JVM.
- Il dominio di protezione indica quali operazioni una classe può (e non può) eseguire.
- Se un metodo viene invocato per eseguire una operazione privilegiata, lo stack viene controllato per verificare che l'operazione possa essere eseguita dal metodo.

Controllo dello Stack

| | | | |
|--------------------|--|--|---|
| protection domain: | untrusted applet | URL loader | networking |
| socket permission: | none | *.lucent.com:80, connect | any |
| class: | gui: ... get(url); open(addr); ... | get(URL u): ... doPrivileged { open('proxy.lucent.com:80'); } <request u from proxy> ... | open(Addr a): ... checkPermission(a, connect); connect (a); ... |

Domande

- Descrivere l'importanza dei meccanismi di protezione in un ambiente mono-utente.
- Spiegare perché i domini in MULTICS non permettono l'implementazione del privilegio minimo.
- Definire una variante della matrice di accesso che realizzi la definizione del cambio di dominio senza l'aggiunta di elementi di switch.
- Discutere le differenze tra liste di abilitazione e liste di accesso.
- Descrivere come può avvenire l'utilizzo della matrice di accesso quando in un programma si eseguono operazioni di lettura da tastiera.