

# Zaštita

- **Ciljevi zaštite**
- **Domeni zaštite**
- **Matrica prava pristupa**
- **Implementacija matrice prava pristupa**
- **Poništavanje prava pristupa**
- **Capability-Based Sistemi**
- **Language-Based tip zaštite**

# Zaštita

## ■ Operativni sistem sastoji se iz **kolekcije objekata**:

- ☞ hardverski (CPU, štampač, CD-ROM...)
- ☞ ili
- ☞ softverski (fajl, program, semafor)

## ■ Svaki objekat ima:

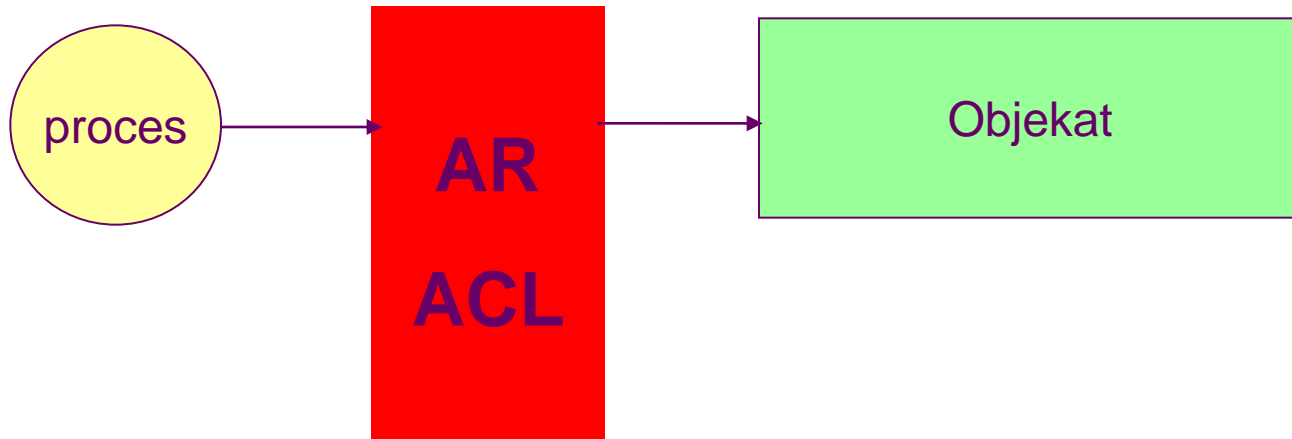
- ☞ jedinstveno **ime**
- ☞ može mu se pristupati kroz
- ☞ dobro definisan **skup operacija**

## ■ Problemi zaštite:

- ☞ Osigurati da se **svakom objektu**
- ☞ pristupa na korektan način
- ☞ i
- ☞ samo od **strane procesa**
- ☞ **koji su ovlašćeni da to rade**

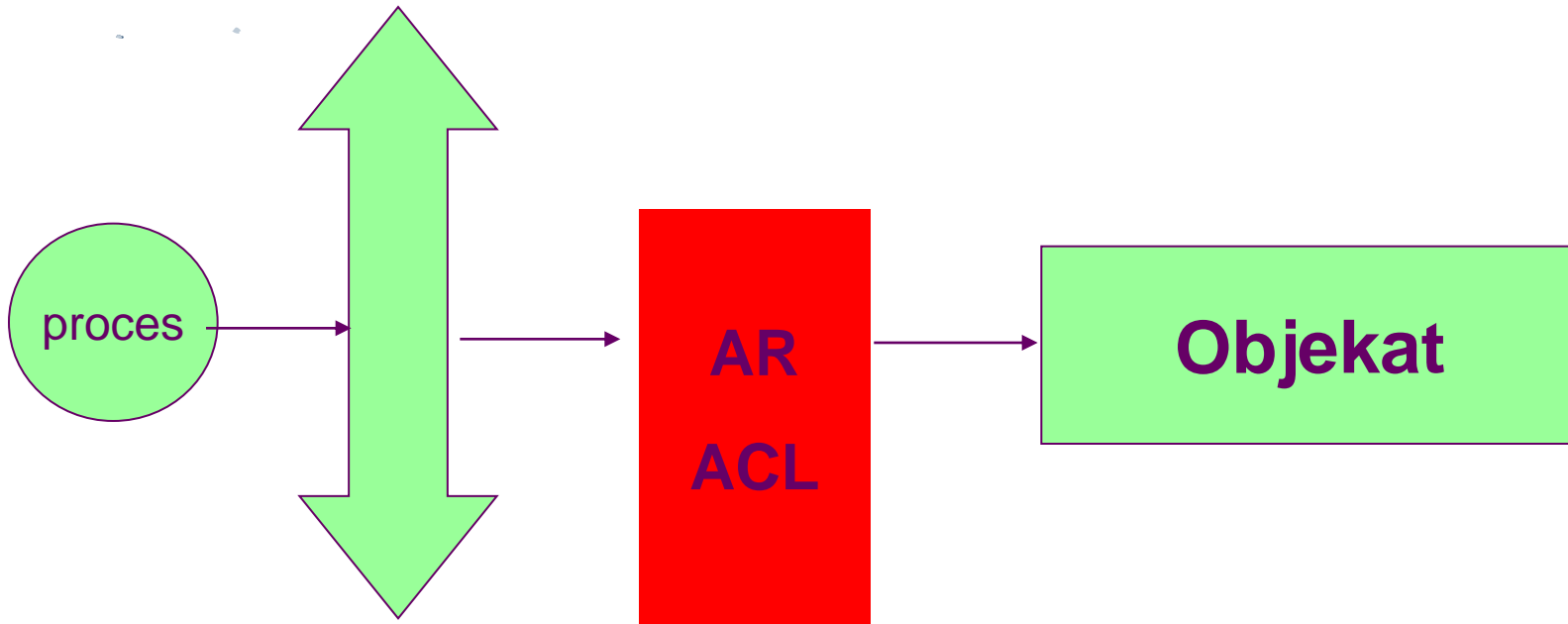
# Zaštita vs. Sigurnost

## ■ Zaštita je **interni** problem



# Sigurnost

- Sigurnost je **eksterni** problem



# Struktura Domena

## ■ Domen zaštite =

☞ skup objekata

☞ operacija dozvoljenih na tim objektima

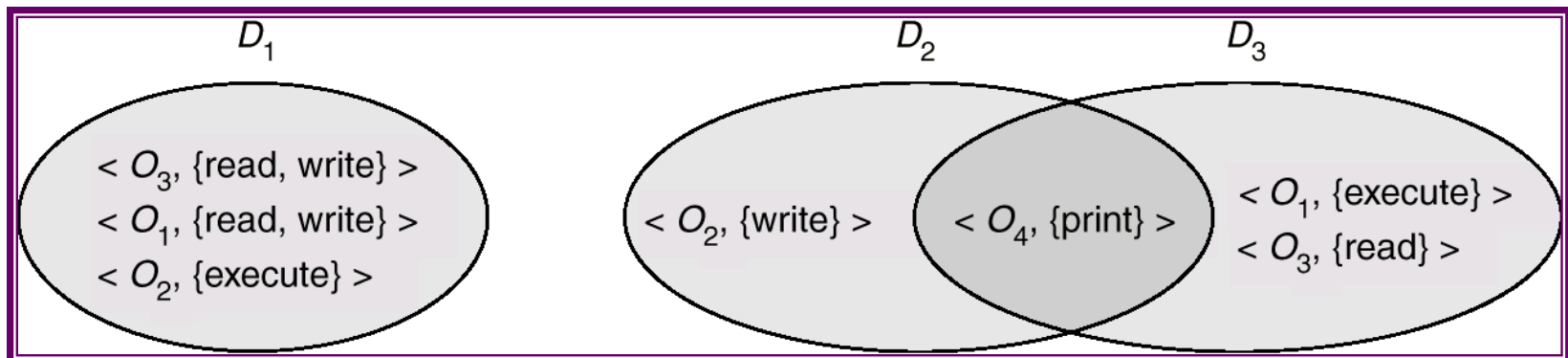
## ■ Pravo pristupa = $\langle$ ime objekta, skup prava $\rangle$

☞ gde skup prava predstavlja

☞ podskup svih važećih operacija

☞ koje mogu biti izvršene na objektu.

## ■ Domen = skup prava pristupa



# Implementacija Domena (UNIX)

## ■ Sistem sadrži 2 klase of domena:

☞ Korisnički domen

☞ Upravljački (kernelški) domen

## ■ UNIX

☞ Domen = korisnički-id

☞ Prebacivanje domena se može realizovati preko fajl-sistema

📄 gde svaki fajl ima domen bit (**setuid bit**).

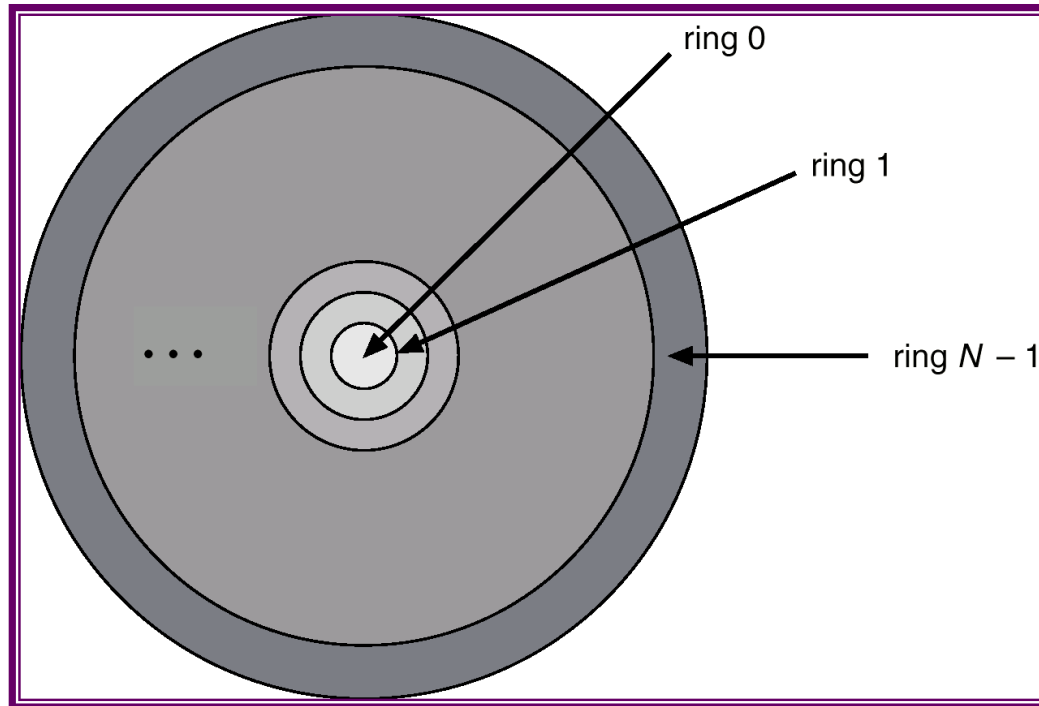
📄 Ako se fajl is izvršava i setuid = on

📄 korisnik dobija id od vlasnika fajla

📄 kada se izvršavanje programa završi, korisnički id je resetovan

# Implementacija Domena (Multics Sistemi)

- Neka su  $D_i$  i  $D_j$  bilo koja dva domenska prstena
- Ako je  $j < i \Rightarrow D_i \subseteq D_j$



## Multics Prstenovi

# Matrica Prava Pristupa

- Posmatrajmo zaštitu kao **matricu** (matricu prava pristupa)
- Redovi predstavljaju **domene**
- Kolone predstavljaju **objekte**

- **Pristup**(*i*, *j*) **Pristup**(*i*, *j*)

☞ predstavlja skup operacija koje

☞ proces iz **Domena<sub>i</sub>**

☞ može da izvrši nad **Objektom<sub>j</sub>**

# Matrica Prava Pristupa

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	

Slika A

# Korišćenje Matrice Prava Pristupa

- Ako proces u **Domenu  $D_i$** 
  - ☞ pokušava da izvrši “op” nad **objektom  $O_j$** ,
  - ☞ tada
  - ☞ “op” mora biti deo matrice prava pristupa
- Može biti proširena na **dinamičku zaštitu**
  - ☞ Operacije prava pristupa:
    - ☞ **add**
    - ☞ **delete**
    - ☞ **switch**
  - ☞ **Specijalna prava pristupa:**
    - ☞ **vlasništvo  $O_i$**
    - ☞ **copy** “op” od  **$O_i$**  ka  **$O_j$**
    - ☞ **control** –  **$D_i$**  može modifikovati  **$D_j$**  prava pristupa
    - ☞ **transfer** – prebacivanje od domena  **$D_i$**  ka  **$D_j$**

# Korišćenje Matrice Prava Pristupa

- Matrica prava pristupa pravi razliku između mehanizma i strategije.

- **Mehanizam**

- ☞ Operativni sistem stvara matricu prava pristupa uz pravila.
- ☞ Osigurava da se matricom rukovodi samo
- ☞ od strane autorizovanih korisnika
- ☞ i
- ☞ da su pravila striktno određena

- **Strategija (Policy)**

- ☞ Korisnik diktira strategiju
- ☞ ko može pristupiti
- ☞ datom objektu
- ☞ i
- ☞ u kom načinu rada

# Matrica Prava Pristupa sa Slike A sa Domenima kao Objektima

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			

Slika B

# Matrica Prava Pristupa sa Pravima *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)

# Matrica Prava Pristupa sa Pravima *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)

# Modifikovana Matrica Prava Pristupa sa Slike B

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			



# Implementacija Matrice Prava Pristupa

## ■ 1. Globalna tabela

☞ trojka <domen, objekat, skup prava>

## ■ 2. ACL od strane objekta (**kolona**)

☞ par <domen, skup prava>

## ■ 3. Domenska lista mogućnosti(red)

☞ par <objekat, skup prava>

## ■ 4. Mehanizam ključeva

☞ zaključavanja objekata

☞ domenskih ključeva

# Implementacija Matrice Prava Pristupa

## ■ ACL

- Svaka kolona = Pristupno-kontrolna lista za jedan objekat
- Definiše ko može izvršavati koju operaciju.

Domen 1 = Read, Write  
Domen 2 = Read  
Domen 3 = Read

- Sažeta lista (korisnik D, grupa D, javni D)

## ■ Lista mogućnosti

- Svaki red = Lista mogućnosti (kao ključ)



Za svaki domen, definiše koja operacija je dozvoljena na kom objektu.

Objekat 1 – Read

Objekat 4 – Read, Write, Execute

Objekat 5 – Read, Write, Delete, Copy

# Poništavanje Prava Pristupa

## ■ Pristupna lista ACL

### ■ **Brisanje prava** pristupa sa pristupne liste:

☞ **Jednostavno**

☞ **Trenutno**

## ■ Lista mogućnosti

### ■ Šema potrebna za nalaženje mogućnosti u sistemu

### ■ pre

### ■ nego što je i sama opozvana

☞ **Reacquisition**

☞ **Back-pointers**

☞ **Indirection**

☞ **Keys**

# Capability-Based Sistemi

## ■ Hydra

- ☞ Unapred određeni skup prava pristupa
  - 📄 poznata i interpretirana od strane sistema.
- ☞ Interpretacija definisanih korisničkih prava
  - 📄 izvedena jedinstveno od strane korisničkih programa;
  - 📄 sistem omogućuje zaštitu pristupa za korišćenje ovih prava.

## ■ Cambridge CAP Sistem

- ☞ Mogućnost podataka –
  - 📄 Omogućuje standardne operacije read, write, execute
  - 📄 individualnih smeštajnih segmenata
  - 📄 dodeljenih objektu.
- ☞ Softverska mogućnost –
  - 📄 interpretacija levo ka podsystemu,
  - 📄 kroz njegove sigurnosne procedure.

# Language-Based Zaštita

- **Specifikacija zaštite u programskom jeziku dozvoljava**
  - ☞ visok nivo opisa polisa
  - ☞ za dodeljivanja i korišćenje resursa
- **Implementacija programskog jezika omogućuje:**
  - ☞ softver za primenu zaštite
  - ☞ kada je automatska hardver-podrška provere nedostupna
- **Tumačenje specifikacije zaštite**
  - ☞ generisanje poziva makro zaštitnog sistema
  - ☞ je omogućeno
  - ☞ od strane hardvera i operativnog sistema