Seminario di Sicurezza Informatica

Reference Monitors

Studente:

FALLERI FILIPPO

Docente:

STEFANO BISTARELLI

Reference Monitor (RM)

- **Reference monitor:** Concetto di controllo degli accessi, che fà riferimento ad una macchina astratta la quale controlla tutti gli accessi.
- **Security Kernel:** L'insieme delle componenti hardware firmware e software del TCB che implementano il concetto di RM.

Dove mettere il RM

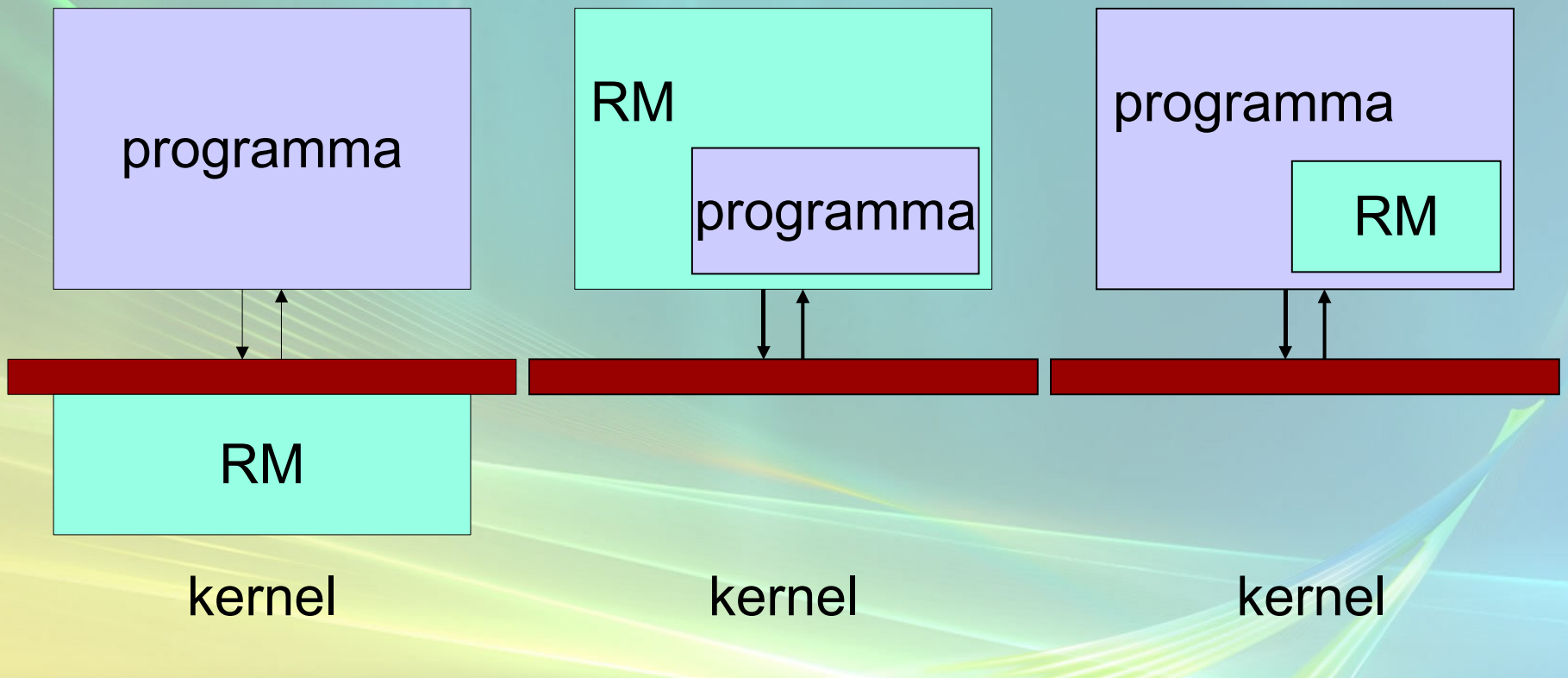
- **HARDWARE:** Meccanismi di controllo degli accessi nel microprocessore.
- **KERNEL SISTEMA OPERATIVO:** Es: Hypervisor, che emula una macchina virtuale e puo' essere usata per separare utentio applicazioni.
- **SISTEMA OPERATIVO:** Es: Cotrollo degli accessi in Unix e Windows2000.
- **LIVELLO SERVIZI:** Es: Controllo degli accessi nella gestione dei database.
- **LIVELLO APPLICAZIONE:** Sviluppatori possono decidere di inserire nelle applicazioni meccanismi di sicurezza per particolari richieste.

RM rispetto all'applicazione

Supporto del kernel (e.g. in O/S)

Interprete

Modifica dell'applicazione



Trusted Computing Base (TCB)

- L'insieme dei meccanismi di protezione nel calcolatore, la combinazione dei quali è responsabile di garantireuna politica di sicurezza.
- Un TCB consiste di 1 o più componenti che, uniti, provvedono al mantenimento della sicurezza di un sistema o prodotto.
- La capacità di un TCB di applicare una corretta politica di sicurezza dipende esclusivamente dai meccanismi all'interno del TCB e dall'inserimento di parametri corretti da parte del personale amministrativo.

Integrità del Sistema Operativo

Integrity problem: Anche il S.O. Stesso è oggetto di richieste di accessi, quindi necessita di meccanismi di protezione.

3 richieste principali:

- Utenti *non* devono essere in grado di modificare il S.O.

- Utenti devono essere in grado di usare(Invocare) il S.O.

- Utenti *non* devono poter abusare del S.O.

Concetti comunemente usati per soddisfare le tali richieste:

- **Status information**
- **Controlled invocation** (restricted privilege)

Modalità delle Operazioni
(Status Information)

- Per proteggersi, il S.O. E'in grado di distinguere i suoi processi,dai processi dell'utente.

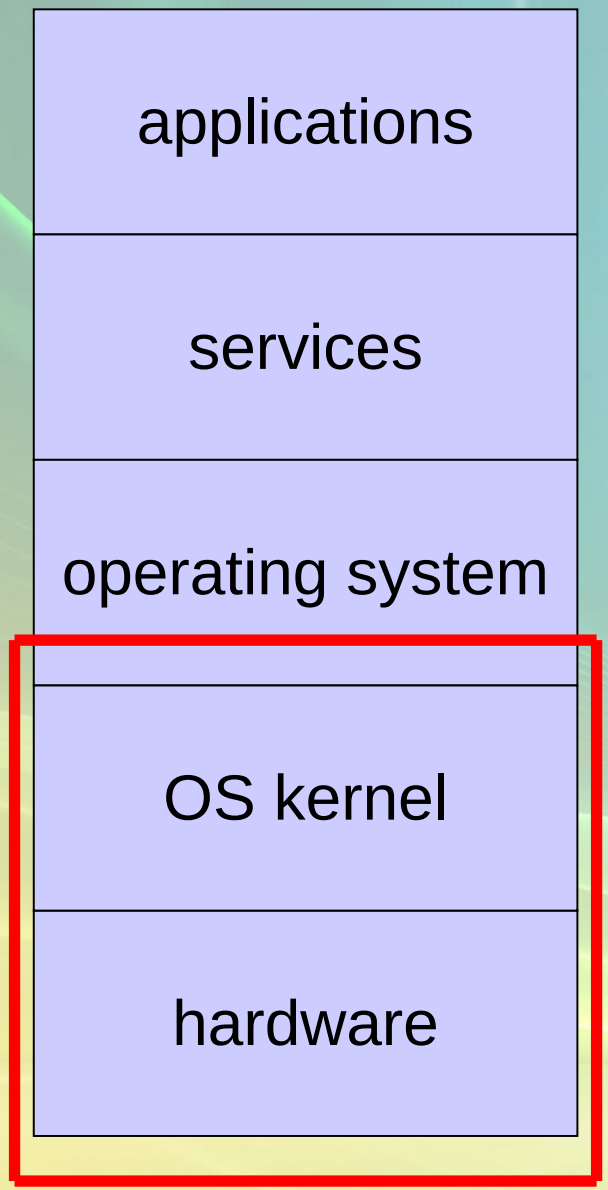
- **Status flag** permette al sistema di lavorare in modi differenti.
  - Intel 80x86: 2 bit di stato e 4 modalità
  - Unix distingue tra user e superuser (root)

- Es:: Per fermare un utente dallo scrivere in memoria, il S.O. Può concedere tale possibiltà solo se il processore è in modalità SuperUser.

Controlled Invocation

- Es: Un utente vuole scrivere una cella di memoria (è richiesta la modalità SuperUser)
- Il sistema deve cambiare modalità.
- Semplicemente cambiando lo **status bit** per dare all'utente tutti I privilegi del superuser.
- Così il sistema permette l'esecuzione solo di un predefinito insieme di operazioni in modalità superuser, ritornando in modalità user prima di restituire il controllo all'utente.
- Questo procedimento è chiamato:
  - Controlled Invocation

Meccanismi di Sicurezza del Core



Sapendo che un meccanismo di sicurezza in un determinato livello può essere compromesso da un attacco ad un livello inferiore.

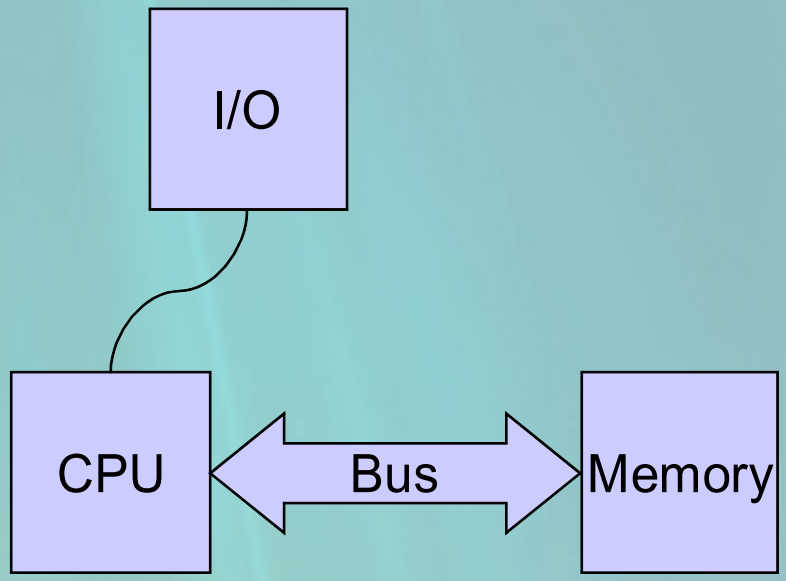
- *E' possibile valutare I sistemi di sicurezza in maniera più approfondita*

  (Strutture ragionevolmente semplici per l'analisi e i test)

- *Riduzione dei ritardi causati dalla sicurezza*

  (Scelgo dei meccanismi di protezione più usati e li metto nel Core)

Architettura del Calcolatore



Principali componenti di un calcolatore ():

- Central processing unit (**CPU**)
  - *Registri (dedicati , generali)*
  - *Aritmetic Logic Unit (ALU)*
- Memoria
  - *Volatile / Non Volatile*
  - *Es: RAM , ROM , EPROM , WROM , etc...*
- Bus connette CPU e memoria
- Dispositivi input/output

Processi e Threads

PROCESSO: un programma in esecuzione, che consiste di *codice eseguibile*, *dati*, ed un *contesto di esecuzione(Es:* il contenuto di alcuni registri della CPU).

- Un processo ha un proprio spazio di indirizzi e comunica con gli altri solo richiamando primitive del SO.
- Separazione logica dei processi è la base della sicurezza.

THREADS: filoni di esecuzione all'interno di un processo. Threads condividono uno spazio di indirizzi al fine di evitare i ritardi causati dai passaggi tra contesti differenti e potenziali controlli di sicurezza.

Processi e thread sono importanti unità di controllo per il S.O. e per la sicurezza. Essi sono i 'soggetti' del controllo degli accessi.

Interrupts - Traps

CPU si occupa di sollevare interruzioni delle esecuzioni create da errori nel programma, da richieste di utenti, da guasti hardware, etc, con le *EXCEPTIONS, INTERRUPTS e TRAPS.*

Questi termini si riferiscono a diversi tipi di eventi.

Una **Trap** è un input per la CPU che include un indirizzo (*Interrupt Vector*) in una *Interrupt Vector Table* con la posizione del programma (*Interrupt Handler*) che si occupa della gestione dell'evento specificato nella Trap.

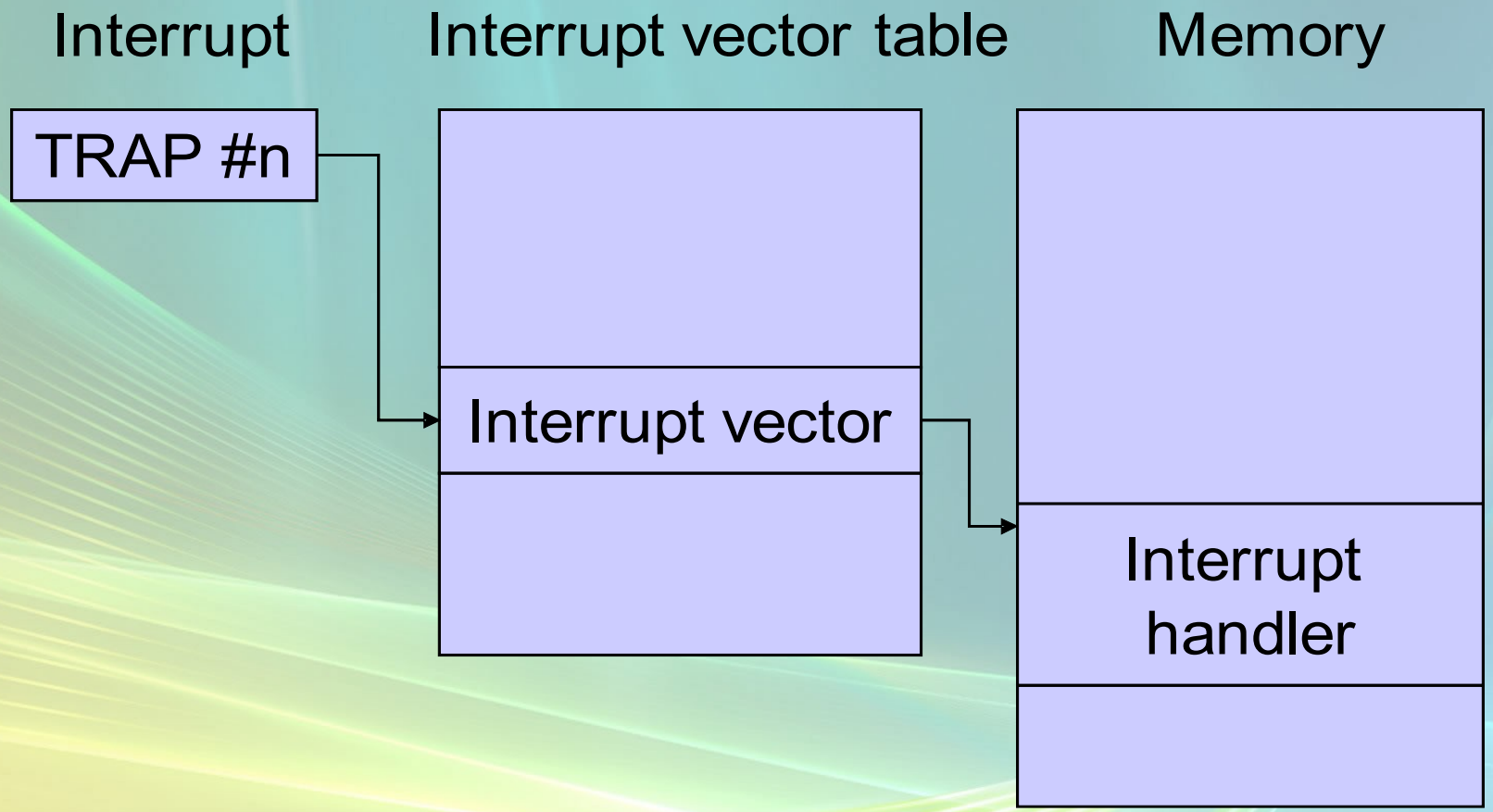
- Il SO salva lo il suo stato nello stack

- Viene eseguito l' Interrupt Handler

- E' ripristinato lo stato della CPU, prima di restituire il controllo all'utente

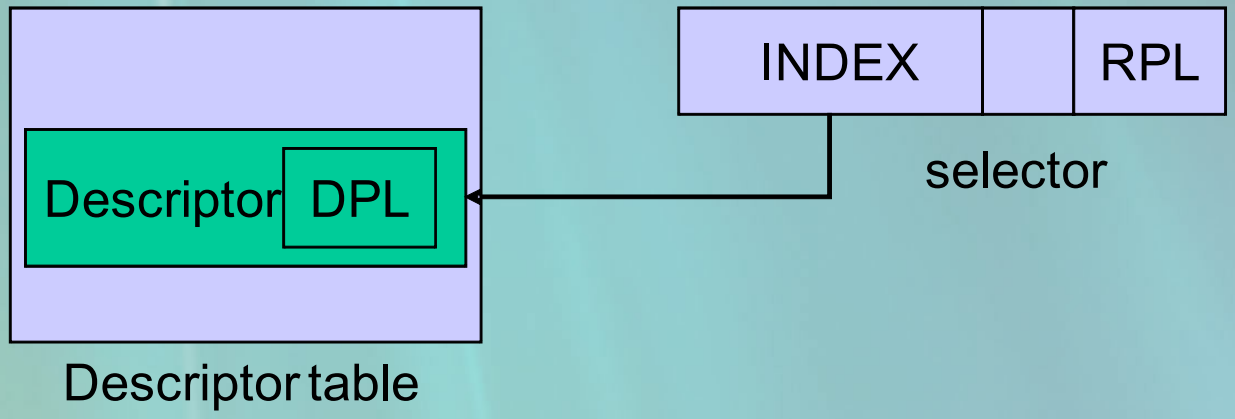
Schema Traps

Schema del funzionamento del TRAP



Es: Intel 80x86

Controllo degli Accessi



DESCRITTORI: Contengono informazioni sugli oggetti del sistema come segmenti di memoria, tabelle di controllo degli accessi, Gates, etc...

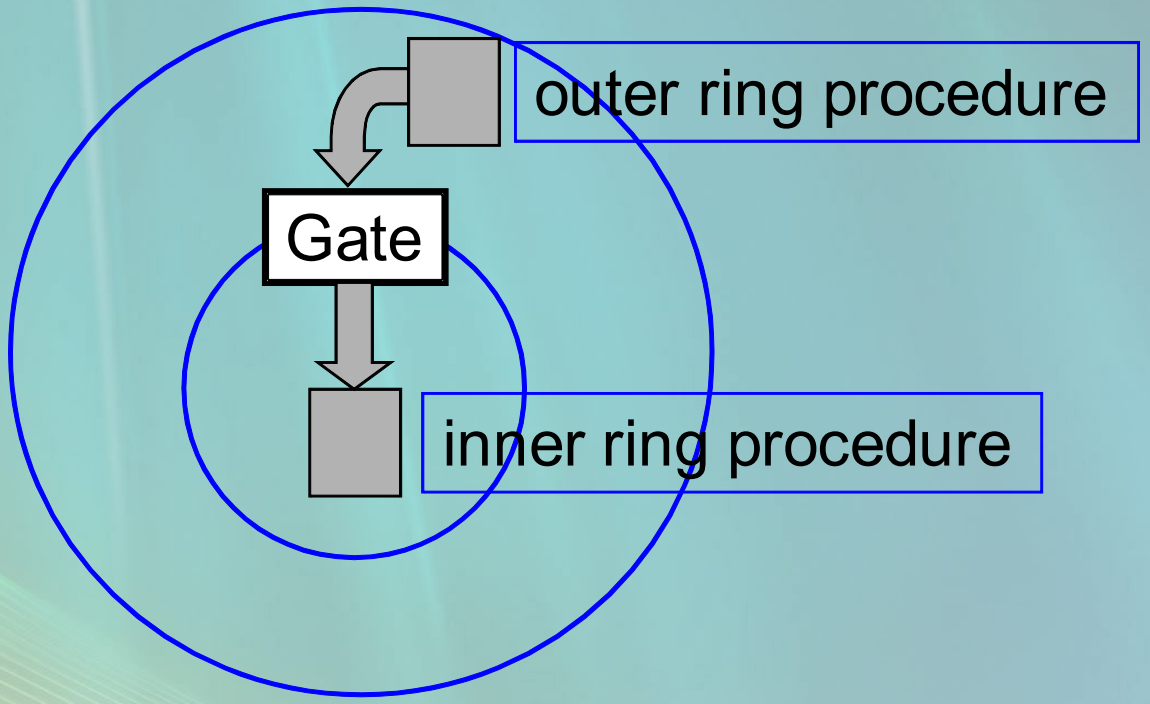
Il livello di privilegi di un oggetto è contenuto nel campo *DPL* del suo decrittore

SELETTORI: Composto da un campo di 16 bit contenente l'indice per l'entry dell'oggetto interessato nella Descriptor Table ed un campo(*RPL*) contenente il livello di privilegio richiesto.

Solo il S.O. Può avere accesso ai selettori

Es: Intel 80x86

Controlled Invocation



GATE: Oggetto di sistema che punta ad una procedura, dove il gate ha un livello di privilegi diverso da quello della procedura a cui punta.

- Consente l'accesso a procedure ad un livello più interno

Confused Deputy Problem

Nel momento in cui una sub-routine viene invocata attraverso un Gate, il CPL del processo chiamante viene cambiato e impostato con i permessi del codice che deve essere eseguito; una volta completata l'esecuzione, il CPL è riportato al suo valore iniziale.

Il processo chiamante può richiedere di copiare oggetti di sistema a livelli più accessibili in modo da potervi accedere anche con un livello di permessi minore.

Questo tipo di attacco viene chiamato:

- **Confused Deputy Problem** (*Luring Attack*)

RIMEDIO:

- Utilizzo dell'istruzione **ARPL** (*Adjust Privilege Level*)

Protezione della Memoria

Il SO gestisce tutti gli accesi a risorse e dati.
Durante esecuzioni di processi appartenenti ad utenti diversi esso deve garantire:

- Separazione dello spazio dell'utente dallo spazio dedicato al SO
- Separazione logica degli utenti
- Limitazione degli oggetti di memoria a cui un processo può accedere

Separazione logica degli utenti a 2 livelli:

- *File Management*: si occupa di oggetti di memoria logica
- *Memory Management*: si occupa di oggetti di memoria fisica

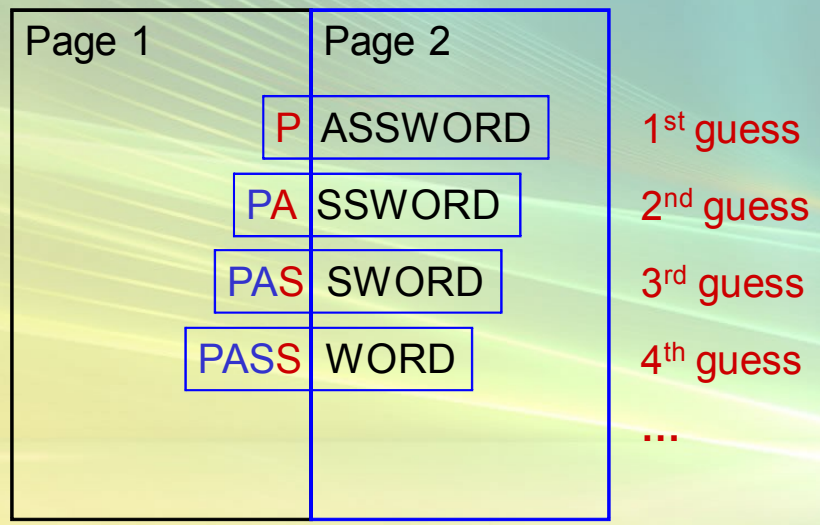
SEGMENTAZIONE e **PAGINAZIONE**

Paginazione

Paginazione: divide la memoria in pagine di uguale lunghezza.

+ Unità di lunghezza fissa consentono una gestione efficiente della memoria.
- Non è una buona base per il controllo di accesso, non essendo le pagine unità logiche. Una pagina può contenere diversi oggetti che richiedono differenti protezioni.

Un *Page Fault* può creare un *Covert Channel*(canale occulto)



Es:
Se una password è memorizzata in una pagina di confine, è possibile tramite l'osservazione di un PageFault ricreare la password indovinando lettera per lettera tutti i caratteri.
Se l'intruso è in grado di controllare se la password viene memorizzata sulla pagina, diventa facile indovinarla.

Segmentazione (1/2)

Segmentazione: divide la memoria in unità logiche di lunghezza variabile.

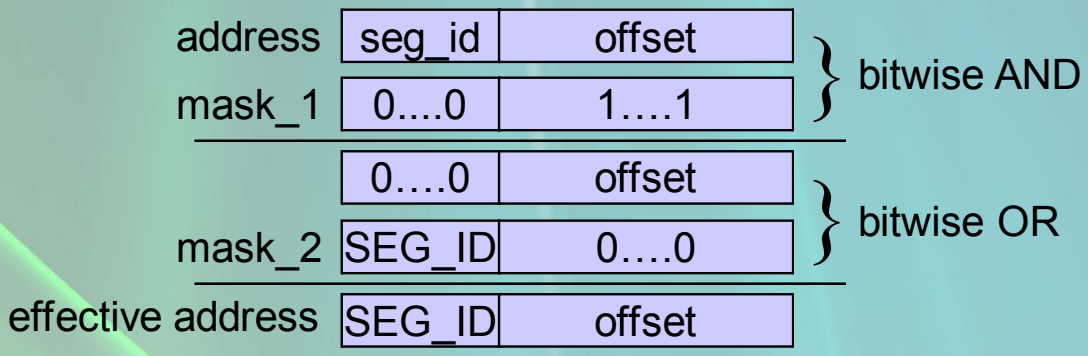
+ La divisione in unità logiche è una buona base per l'applicazione di una politica di sicurezza.

- Unità di lunghezza variabile rendono più difficile la gestione della memoria.

Tre opzioni per controllare l'accesso alla memoria:

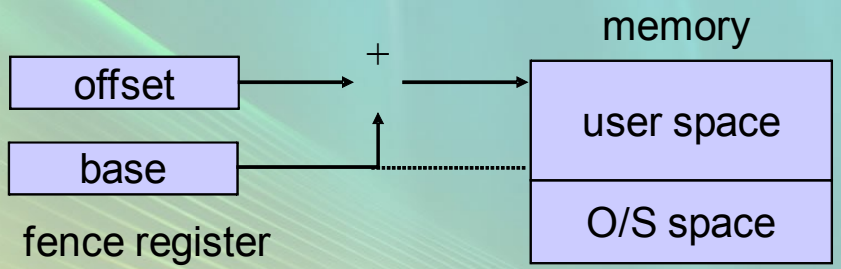
- SO modifica l'indirizzo che riceve dai processi utente
- SO costruisce l'effettivo indirizzo dall'indirizzo relativo che riceve dai processi utente;
- SO controlla se l'indirizzo che riceve da un processo utente si trova all'interno dei limiti predisposti.

Segmentazione (2/2)



Es. 1° Approccio:

Address Sandboxing



Es. 2° Approccio:

Utilizzo *Fence Registrer*

FC2	FC1	FC0	
0	0	0	(undefined,reserved)
0	0	1	user data
0	1	0	user program
0	1	1	(undefined,reserved)
1	0	0	(undefined,reserved)
1	0	1	supervisor data
1	1	0	supervisor program
1	1	1	interrupt acknowledge

Es. 3° Approccio:

Motorola 68000

Function Codes

Bibliografia

- Reference Monitor , TCB , Security Kernel , Confused Deputy Problem:
  http://en.wikipedia.org/

- Computer Security Planning Study (October 1972)
  James P. Anderson

- Interrupts , Exceptions , Traps
  http://www.internals.com/articles/protmode/interrupts.htm

GRAZIE
per la cortese attenzione...

...THE END