



# Università degli Studi di Cassino e del Lazio Meridionale

## Corso di Calcolatori Elettronici *Memoria Virtuale*

Anno Accademico 2015/2016  
Francesco Tortorella

# Requisiti per la gerarchia di memoria

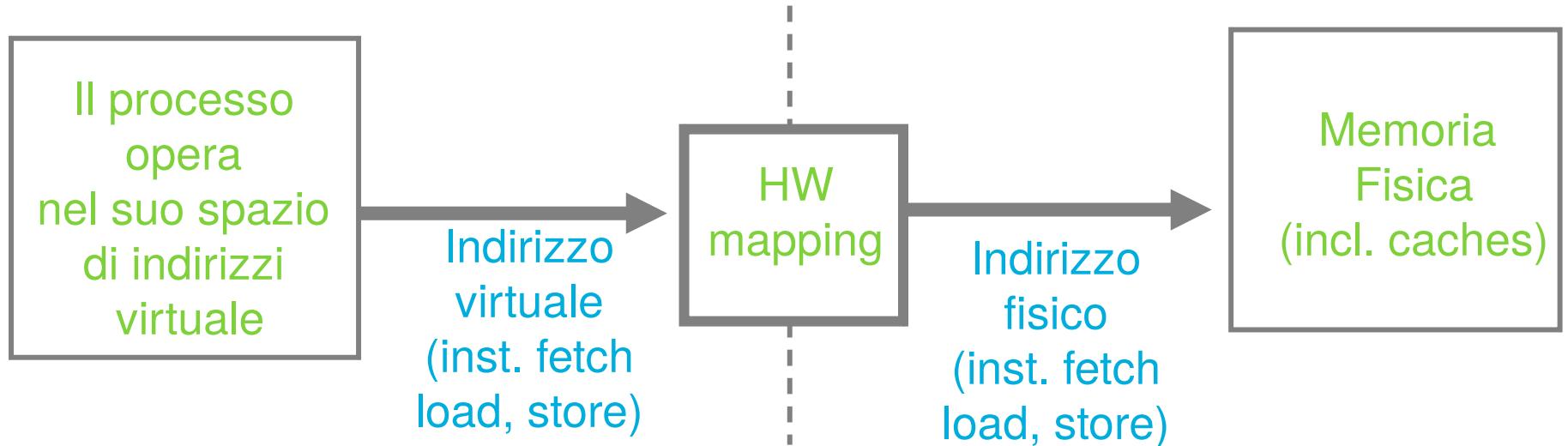
- Se il principio di località permette di ottenere un ottimo rapporto velocità/capacità tra cache e memoria centrale è pensabile di adottare lo stesso approccio tra memoria centrale e memoria di massa?
- Ci sono altri requisiti che è opportuno considerare?

# Memoria Virtuale

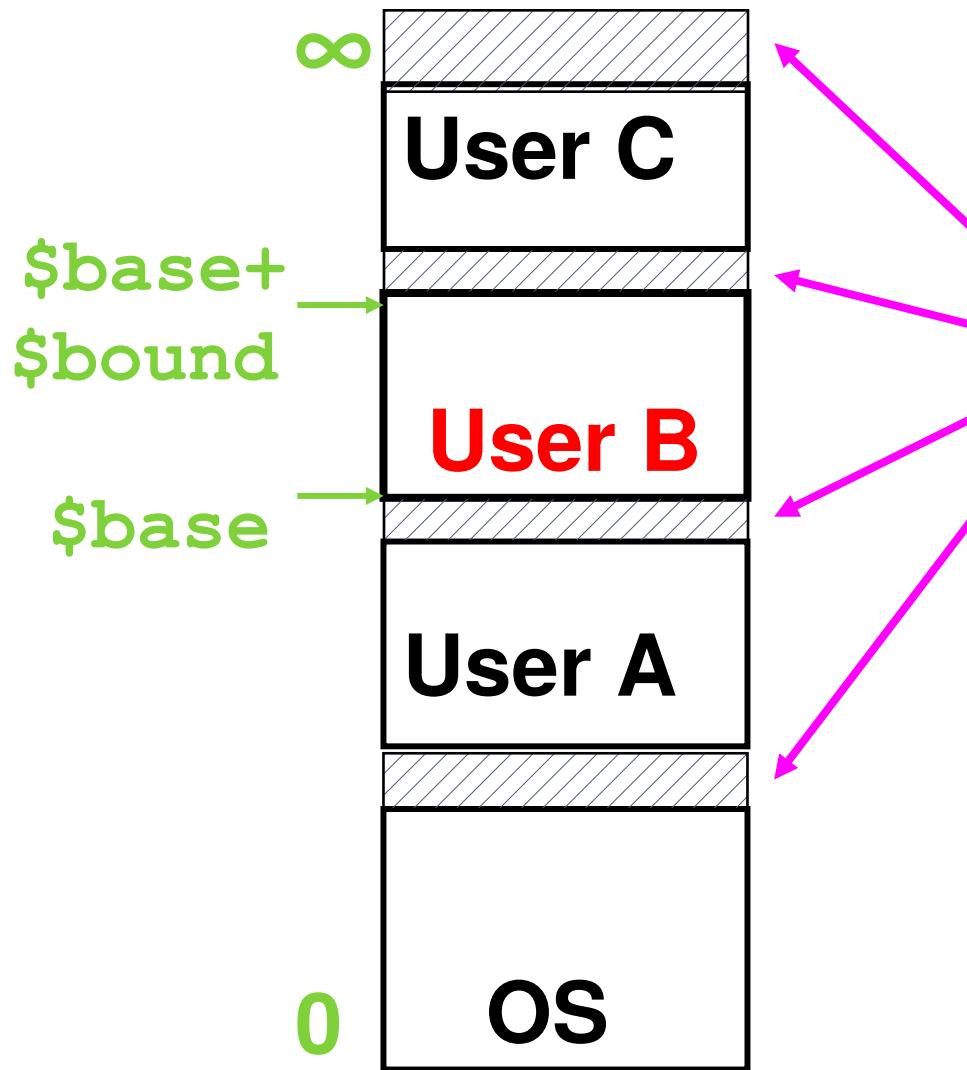
- Successivo livello nella gerarchia di memoria
  - Fornisce ai programmi l'illusione di avere uno spazio di memoria illimitato
  - Distribuisce le “pagine” tra memoria centrale (pagine attive) e disco
- È lo strumento con il quale il Sistema Operativo condivide la memoria e protegge i processi.
- Attualmente, la funzione di protezione è più utile ed importante rispetto alla gerarchia di memoria

# Corrispondenza tra indirizzo virtuale e fisico

- Ogni processo lavora nel suo spazio virtuale di indirizzi.
- Ogni processo è protetto da interventi errati di altri processi.
- Il SO pianifica l'impiego della memoria
- La traduzione dell'indirizzo virtuale in indirizzo fisico avviene via hardware.



# Soluzione semplice: registro limite



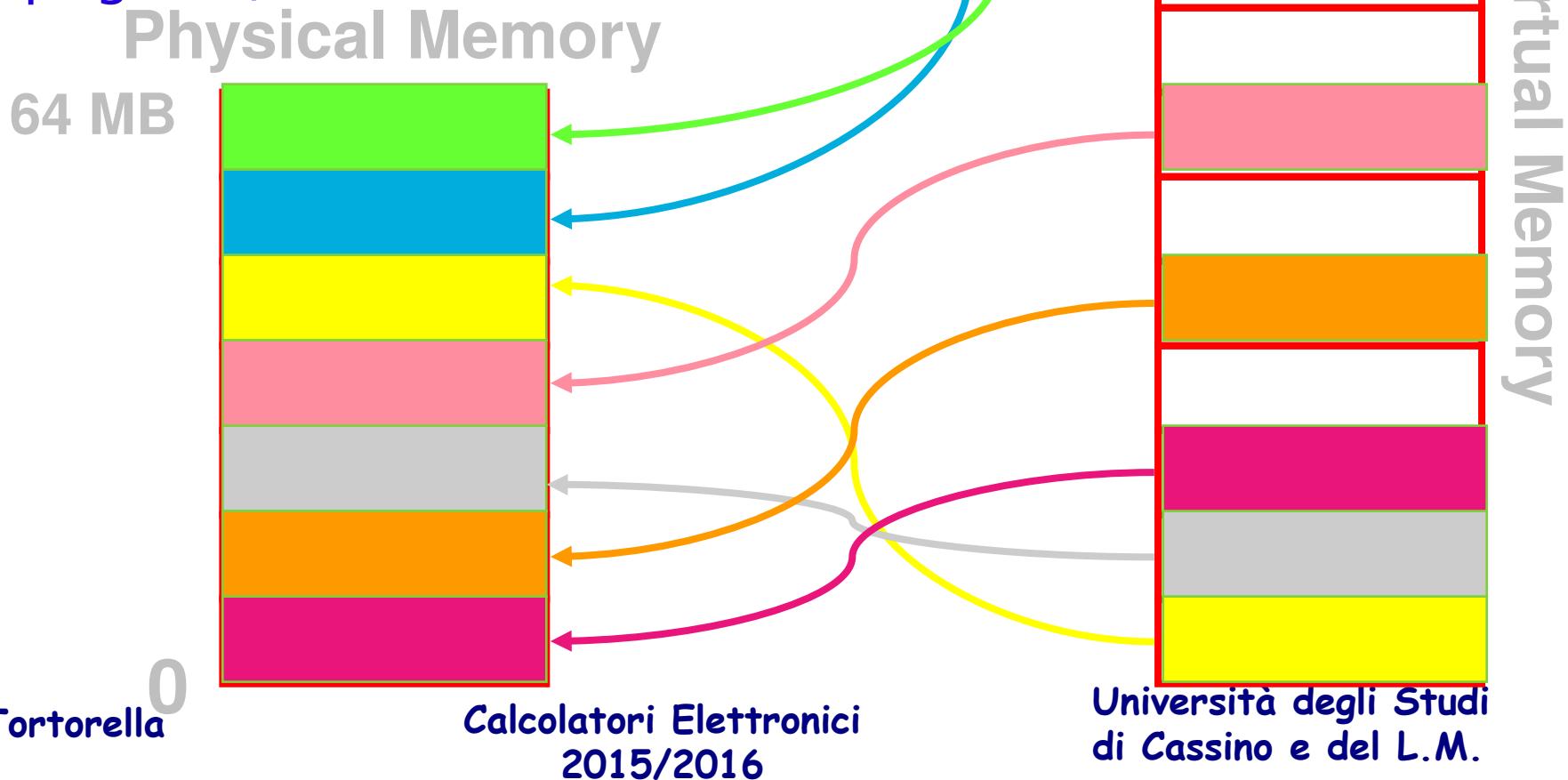
Spazio sufficiente per  
ulteriore processo  
ma discontinuo  
("Frammentazione")

Requisiti:

- discontinuous mapping
- Process size  $>>$  mem

# Corrispondenza memoria virtuale → fisica

- Si divide la memoria in pezzi di uguale dimensione (4 KB - 8 KB)
- Ciascun pezzo di M.V. assegnato ad un pezzo di memoria fisica (“pagina”)



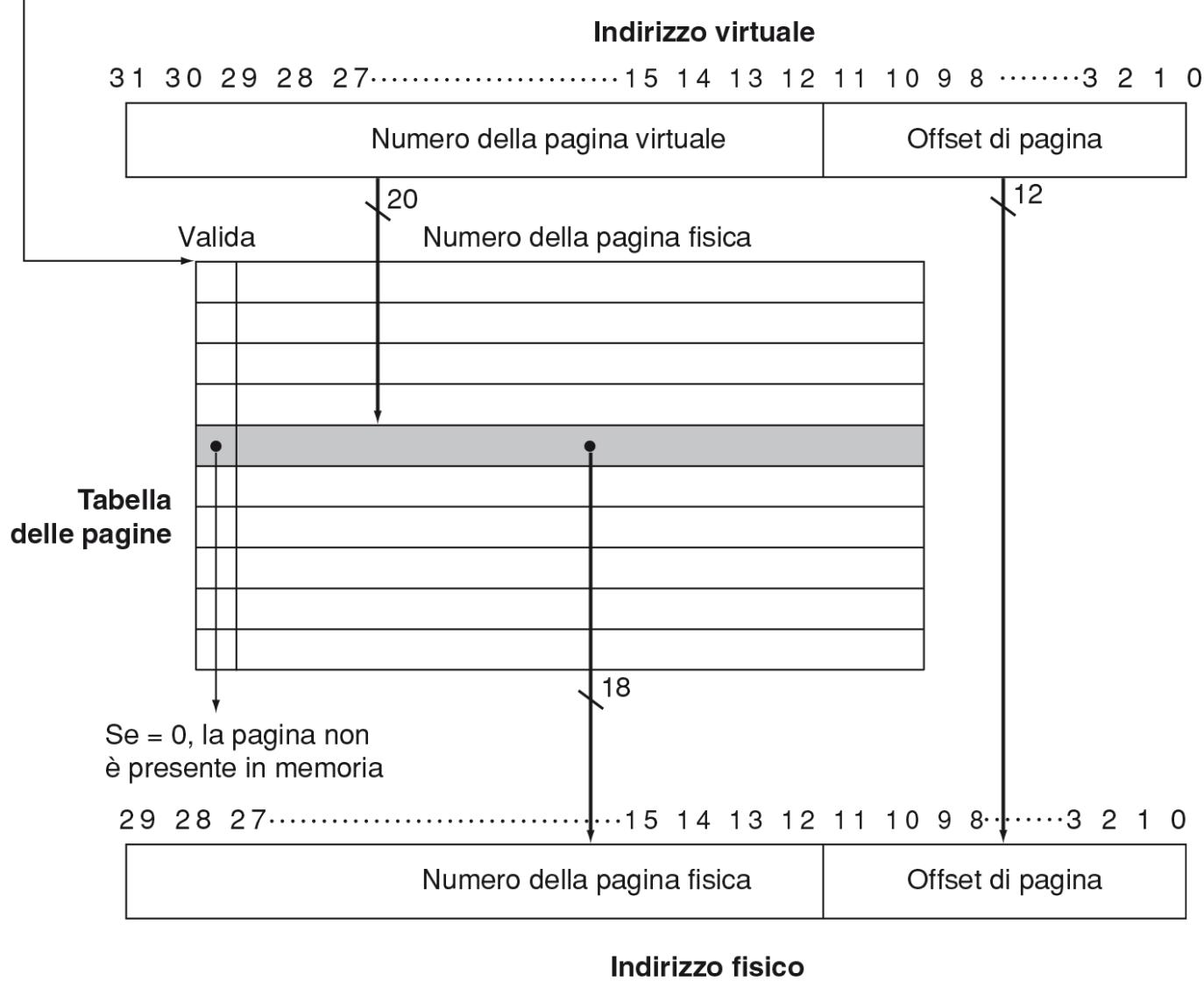
# Funzione di mapping per la M.V.

- Non è possibile definire una funzione semplice per il mapping
- Si impiega una *lookup table*

Page Number	Offset
-------------	--------

- Tabella delle pagine (*Page Table*) per il mapping: indicizzata sul Page number
- Mapping per la M.V.:
  - Offset fisico = Offset virtuale
  - Page Number fisico (*Page frame*)  
= PageTable[Page Number virtuale]

## Registro della tabella delle pagine



# Formato dell'entry della TDP

- Bit di validità: indica se la pagina è effettivamente in memoria; se a 0, il SO forza la ricerca sul disco
- Numero di pagina fisica
- Potrebbe essere presente anche una descrizione dei diritti di accesso alla pagina (Read only, Read/Write, Executable)

# Tabella delle pagine

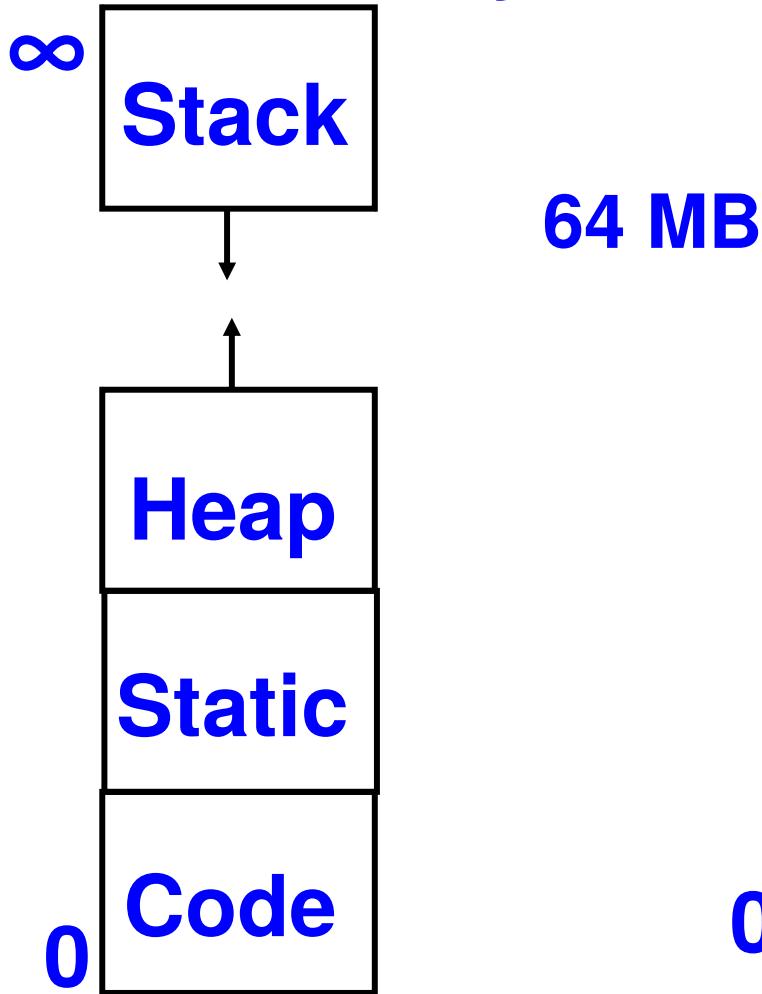
- La tabella delle pagine è una struttura del sistema operativo che contiene la corrispondenza tra indirizzi virtuali e locazioni fisiche.
  - Diversi approcci per organizzare la tabella, tutti gestiti dal sistema operativo
- Ogni processo attivo nel sistema operativo ha una propria tabella delle pagine
  - Lo stato di un processo è costituito dal PC, dai registri del processore e dalla TDP
  - Il SO passa dalla TDP di un processo a quella di un altro modificando il **Registro della tabella delle pagine** (*Page table base register*).

# Protezione degli spazi di indirizzi

- Condivisione della memoria con protezione
  - Differenti pagine fisiche possono essere allocate su diversi processi (condivisione)
  - Un processo può accedere alle sole pagine presenti nella sua TDP (protezione)
- Spazi degli indirizzi separati
  - Siccome i processi usano gli indirizzi virtuali, differenti processi possono avere codice/dati agli stessi indirizzi (virtuali)

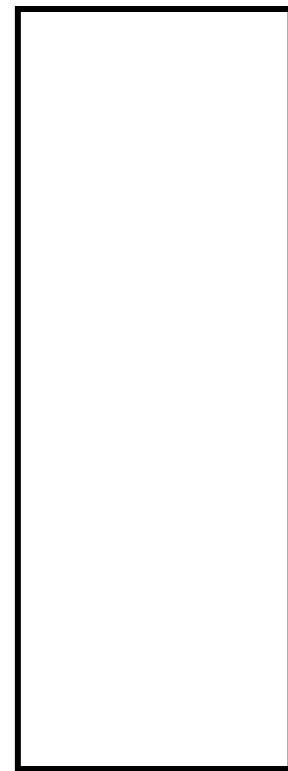
# M.V. con più processi

Processo A:  
Virtual Memory

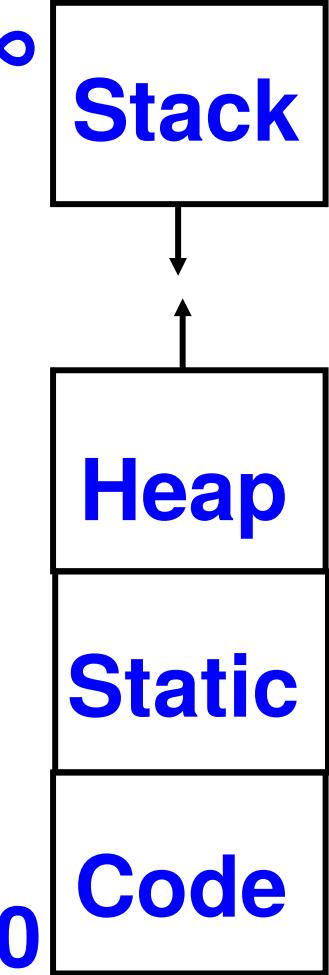


Physical  
Memory

64 MB

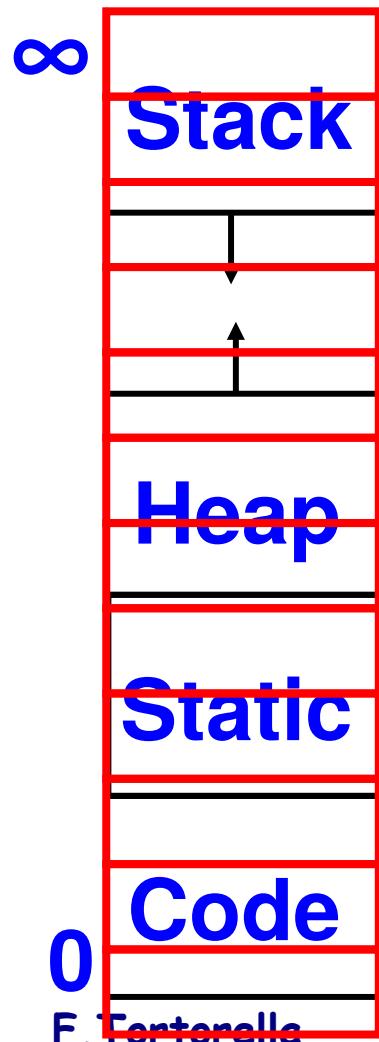


Processo B:  
Virtual Memory



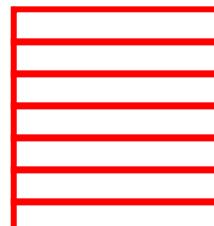
# M.V. con più processi

Processo A:  
Virtual Memory



Physical  
Memory

64 MB

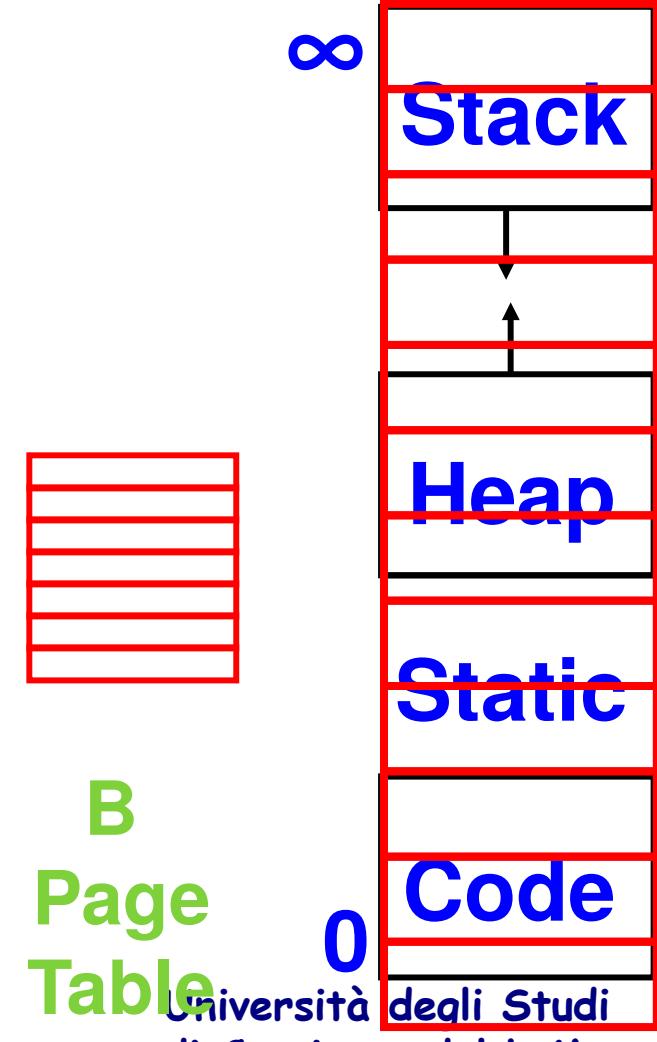


A  
Page  
Table

F. Tortorella

Calcolatori Elettronici  
2015/2016

Processo B:  
Virtual Memory



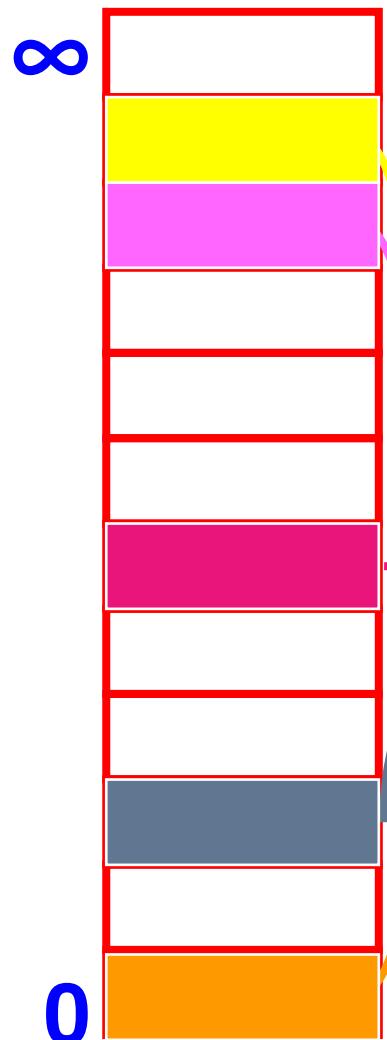
B  
Page  
Table

Università  
degli Studi  
di Cassino e del L.M.

# M.V. con più processi

User A:

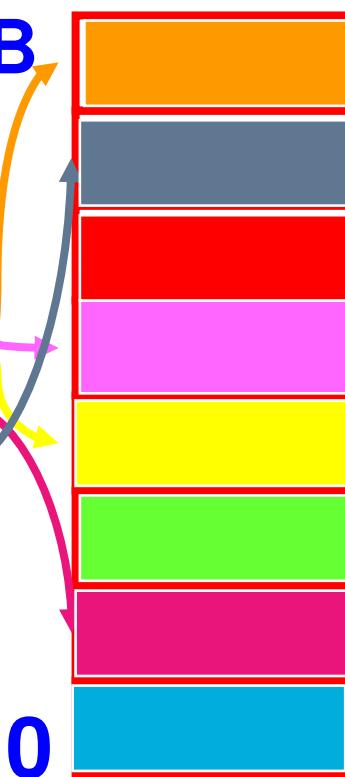
Virtual Memory



A  
Page  
Table

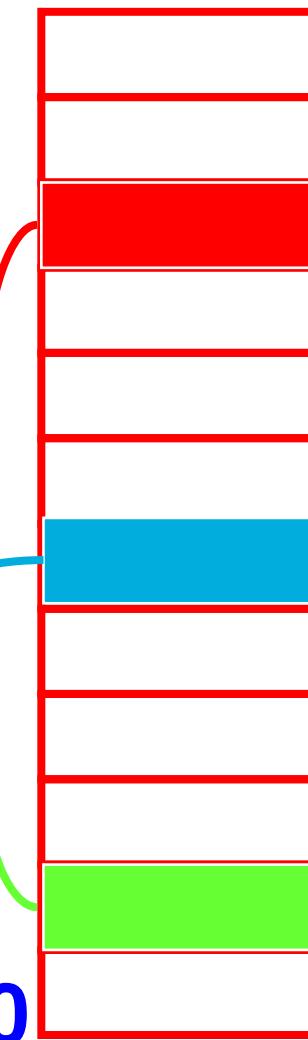
Physical  
Memory

64 MB



B  
Page  
Table

User B:  
Virtual Memory



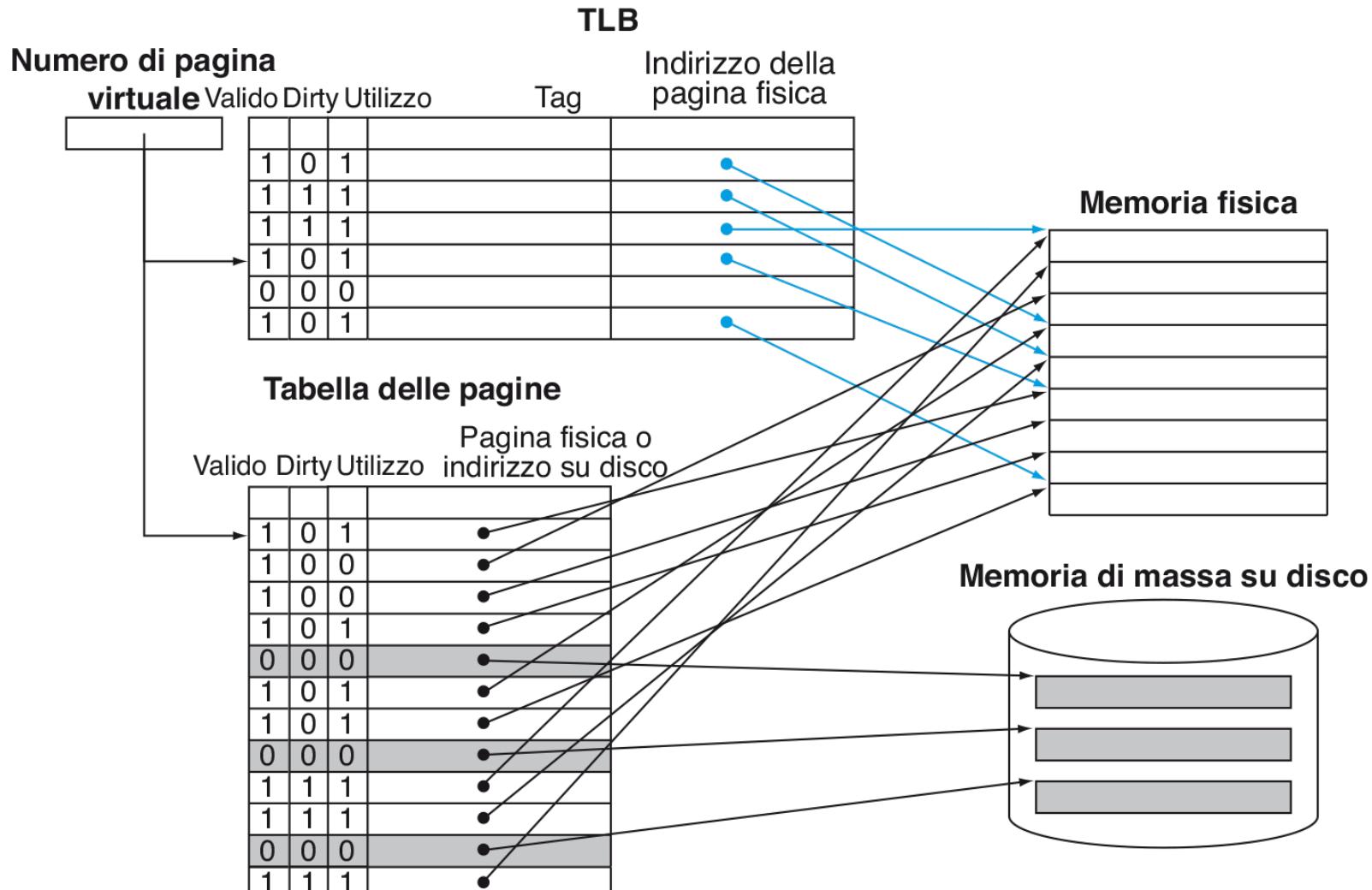
# Caratteristiche dell'approccio con TDP

- Risolve il problema della frammentazione: tutti i pezzi di memoria sono della stessa dimensione, per cui gli spazi liberi possono essere utilizzati senza problema
- Il SO deve riservare dello spazio di swap sul disco **per ogni processo**
- Ogni processo ha la sua TDP
- Il SO può aumentare la memoria di un processo (aree dinamiche) allocando ulteriori pagine:
  - Se ci sono, usa pagine disponibili
  - Altrimenti, scarica (swap) alcune pagine su disco (bit dirty)
  - Metodo LRU per selezionare le pagine da scaricare (bit utilizzo)

# TDP in memoria centrale: problema!

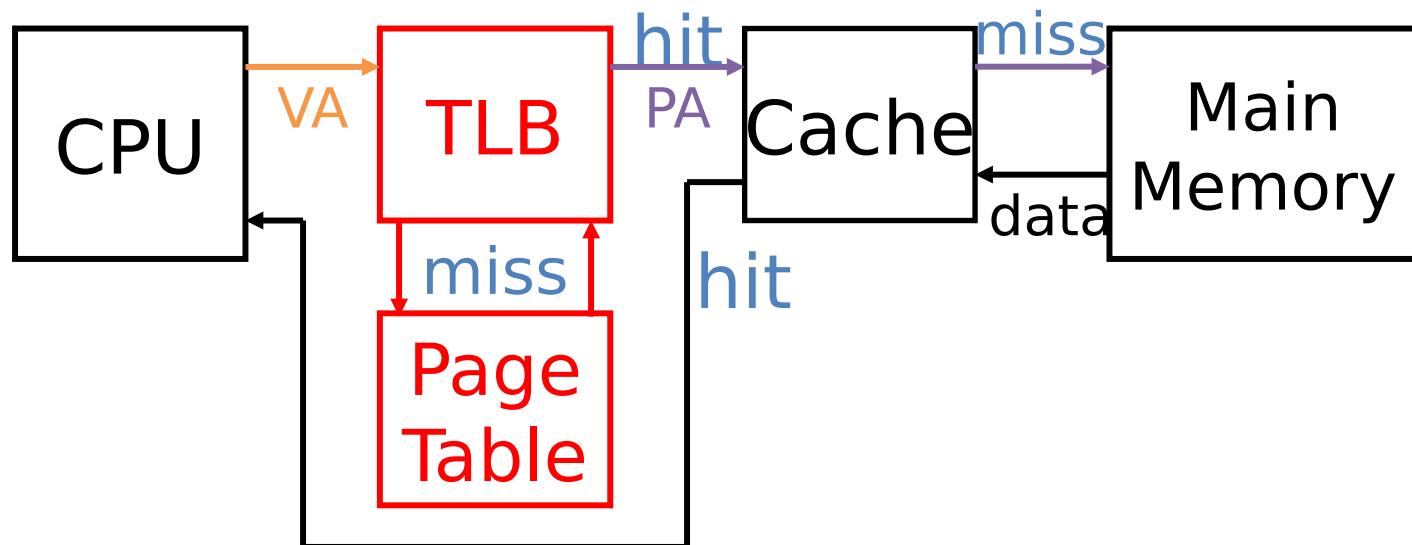
- Per ottenere un dato, sono necessari due accessi in memoria centrale:
  - Uno per la traduzione da indirizzo virtuale a indirizzo fisico
  - Uno per prelevare il dato
- Per velocizzare il processo si può osservare che vale il principio di località anche nel processo di traduzione dell'indirizzo
- Si può quindi considerare una memoria cache per ospitare un certo numero di entry della TDP
- Translation Look-Aside Buffer (TLB)

# Translation Look-Aside Buffer (TLB)

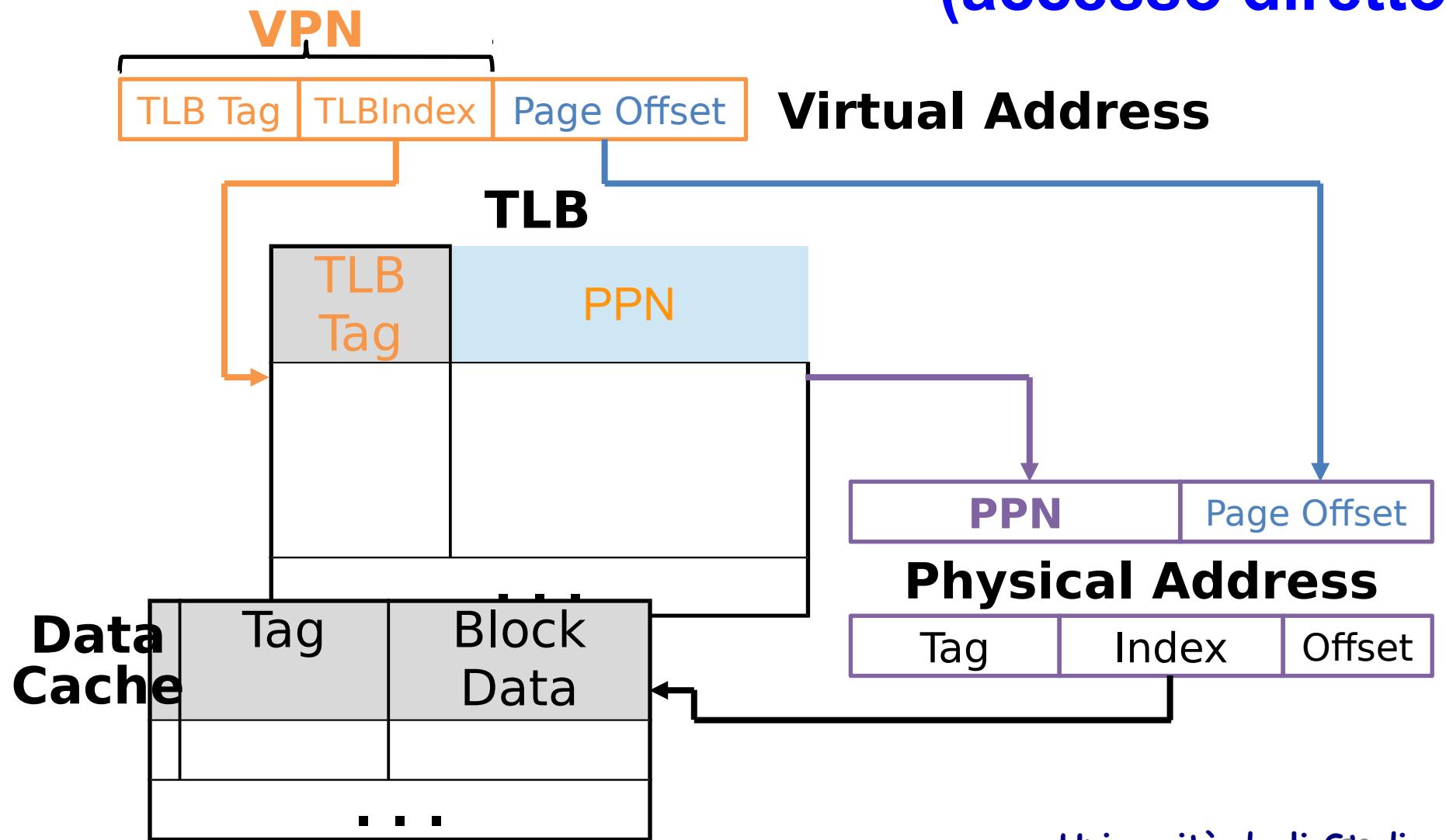


# Translation Look-Aside Buffer (TLB)

- Non molto grande: 128-256 entry
- Può essere direct-mapped, fully o set-associative
- In caso di miss, si preleva l'entry dalla TDP in memoria



# Traduzione dell'indirizzo via TLB (accesso diretto)



# Prelievo di dati in lettura

1) Verifica il TLB (input: VPN, output: PPN)

- *TLB Hit*: realizza la traduzione, fornisci PPN
- *TLB Miss*: verifica la TDP (in memoria)
  - *Page Table Hit*: Carica la page table entry nel TLB
  - *Page Table Miss (Page Fault)*: Trasferisci la pagina dal disco alla memoria, aggiorna la page table entry corrispondente e quindi caricala nel TLB

2) Verifica la cache (input: PPN, output: data)

- *Cache Hit*: Fornisci i dati al processore
- *Cache Miss*: Trasferisci i dati dalla memoria alla cache, fornisci i dati al processore