

Simulatore degli algoritmi di sostituzione delle  
pagine della memoria virtuale

Claudia Rizzello

VR015321

2007/2008

# INDICE

<b>INDICE .....</b>	<b>2</b>
<b>1 CENNI DI TEORIA .....</b>	<b>3</b>
1.1 LA MEMORIA VIRTUALE .....	3
1.2 LA PAGINAZIONE SU RICHIESTA .....	3
1.3 GLI ALGORITMI DI SOSTITUZIONE .....	4
<b>2 COMPILAZIONE ED ESECUZIONE.....</b>	<b>6</b>
2.1 IN LINUX .....	6
2.2 IN WINDOWS .....	7
<b>3 INTERFACCIA DEL PROGRAMMA.....</b>	<b>8</b>
3.1 LA FINESTRA DI DIALOGO .....	8
3.2 LA PROCEDURA “STEP BY STEP” .....	10
3.3 LA PROCEDURA “RISULTATO” .....	12

# 1 CENNI DI TEORIA

L'applicazione fornisce una simulazione degli algoritmi di sostituzione delle pagine della memoria virtuale.

Gli algoritmi illustrati sono: LRU, FIFO, LIFO, MFU, LFU, OTTIMALE.

E' possibile scegliere se seguire la procedura passo per passo, osservando cosa accade per ogni riferimento, oppure se visualizzare immediatamente il risultato di tutti i passaggi.

## 1.1 LA MEMORIA VIRTUALE

La memoria virtuale permette di eseguire processi che possono non essere caricati completamente in memoria: in tal modo i programmi possono avere dimensioni maggiori della memoria fisica.

Viene generalmente implementata attraverso *paginazione su richiesta*, o in alternativa con un sistema di segmentazione. E' anche possibile utilizzare un sistema di segmentazione su richiesta, ma in questo caso gli algoritmi di sostituzione dei segmenti sono più complessi rispetto a quelli di sostituzione delle pagine, poiché i segmenti sono di dimensione variabile.

## 1.2 LA PAGINAZIONE SU RICHIESTA

Ogni programma risiede nella memoria secondaria, in genere costituita da un disco, e per poter essere eseguito deve essere caricato in memoria. Anziché trasferire l'intero programma, è possibile trasferire solo le pagine necessarie, nel momento in cui vengono chiamate.

Il processo viene dunque avviato senza che tutte le sue pagine siano effettivamente caricate; ogni volta che tenta di utilizzare una pagina non è presente in memoria, si verifica un page fault; quando ciò avviene, lo stato del processo viene salvato, in modo che si possa riprendere lì dove è stato interrotto, e la procedura di gestione dei page fault verifica in una tabella delle pagine la validità del riferimento. Se il riferimento è valido, la pagina da trasferire viene inserita in un frame libero, se esiste; altrimenti va a sostituire una delle pagine già caricate, individuata da uno specifico algoritmo di sostituzione. Dopo gli opportuni trasferimenti, nonché l'aggiornamento della tabella delle pagine, il processo può riprendere normalmente dall'istruzione che era stata interrotta.

Il programma simula una memoria virtuale implementata tramite paginazione su richiesta *pura*; vale a dire che, inizialmente, nessuna pagina è caricata in memoria, per cui i primi riferimenti generano sempre dei page fault.

## 1.3 GLI ALGORITMI DI SOSTITUZIONE

### - LRU (least recently used)

Sceglie come vittima la pagina che è stata usata meno di recente.

Questo criterio si basa sul fatto che una pagina che non è stata più utilizzata, probabilmente non è più utile.

### - FIFO (first in first out)

Rimuove la pagina che era stata inserita per prima, indipendentemente dalla sua frequenza d'uso: presuppone che la pagina più "vecchia" non debba essere più utilizzata.

### - LIFO (last in first out)

Indica come vittima l'ultima pagina inserita nei frame. Si utilizza questo criterio quando si suppone che le prime pagine inserite siano quelle più utilizzate.

**- OPT (Optimal)**

Quando deve scegliere una vittima, osserva le chiamate successive e va a sostituire la pagina che non verrà utilizzata per il periodo di tempo più lungo: pertanto assicura il minimo numero di page fault.

Purtroppo non è un algoritmo facilmente utilizzabile poiché, solitamente, la stringa dei riferimenti non è nota a priori.

**- LFU (least frequently used)**

Si serve di un contatore dei riferimenti alle pagine, per poter rimuovere la pagina che è stata meno usata; suppone infatti che la pagina con il numero minore di riferimenti, sia quella meno indispensabile.

**- MFU (most frequently used)**

Sceglie come vittima la pagina che è stata utilizzata maggiormente, supponendo che le pagine aventi un numero di riferimenti basso, debbano ancora essere utilizzate.

## 2 COMPILAZIONE ED ESECUZIONE

La cartella “SO\_MvSimul” contiene al suo interno i seguenti file java:

- MvSimul (file per l'avvio del programma)
- DialogMV
- StepFrame
- RisFrame
- Page
- MemoriaVirtuale
- TLBtable

### PREREQUISITI:

E' necessario aver già installato un compilatore Java.

Se non è presente un compilatore Java, è disponibile on line *Java 2 Software Development Kit (SDK)*. Una volta scaricato il file, procedere alla sua installazione.

In Windows per settare il “path” del compilatore, aprire Pannello di Controllo -> Sistema -> Avanzate -> Variabili d'ambiente. Alla voce "path" aggiungere la directory "bin" del compilatore, specificando il percorso nella quale è stata installata (di default C:\Programmi\Java\jdk1.6.0\_03\bin).

(Nell'aggiungere il percorso della directory bin, separarlo dal percorso precedente con ";" ).

### 2.1 IN LINUX

Prima di eseguire il programma è necessario compilare tutti i file, entrando dalla shell nella cartella che li contiene ed inviando il comando:

```
javac *.java
```

A compilazione avvenuta, lanciare il programma nella medesima cartella, tramite il comando:

```
java MvSimul
```

## **2.2 IN WINDOWS**

Aprire il Prompt dei Comandi, entrare nella cartella contenente i file e compilarli utilizzando il comando:

```
javac *.java
```

Infine, lanciare il programma tramite il comando:

```
java MvSimul
```

## 3 INTERFACCIA DEL PROGRAMMA

### 3.1 LA FINESTRA DI DIALOGO

**INSERIRE I DATI RICHIESTI**

Reference String: 3 2 6 5 5 1 4 7 5

Numero di Frame: 3

(Inserire il numero di frame disponibili)

**ALGORITMO**

☒ LRU      ☐ FIFO

☐ LIFO      ☐ OPT

☐ LFU      ☐ MFU

LEAST RECENTLY USED:  
rimuove la pagina usata  
meno di recente.

?    Step by step    Risultato    Esci

All'avvio del programma appare una finestra di dialogo che riceve in input:

- La Reference String, ovvero la sequenza degli indici delle pagine.  
(Deve essere una sequenza di interi separati da spazio).
- Il numero di Frame disponibili.
- Un algoritmo a scelta tra:
  - LRU
  - LIFO
  - FIFO



- OPT
- MFU
- LFU

Selezionando ciascun algoritmo, appare nell'area di testo il relativo criterio di scelta della pagina vittima.

Una volta inseriti i dati, si può scegliere se visualizzare la procedura passo per passo (pulsante STEP BY STEP), che mostra quanto accade in corrispondenza di ogni singolo riferimento, oppure se vedere immediatamente l'intero risultato (pulsante RISULTATO).

Se gli inserimenti sono corretti, il programma creerà le pagine corrispondenti ai riferimenti della reference string, e la relativa tabella delle pagine. Nella finestra successiva procederà al loro inserimento nei frame secondo l'algoritmo e la modalità selezionati.

Se si verifica un errore nei dati immessi dall'utente (nessun inserimento oppure inserimento errato), viene comunicato nell'area di testo. In tal caso per proseguire è sufficiente reinserire correttamente i valori errati, selezionare l'algoritmo (nel caso in cui non sia stato scelto) e premere nuovamente il pulsante che avvia la procedura desiderata.

Il pulsante “?” fornisce un aiuto rapido sull'uso della finestra e sui dati richiesti.

E' possibile uscire dal programma cliccando il pulsante “ESCI”.

### 3.2 LA PROCEDURA “STEP BY STEP”

**STEP BY STEP CON ALGORITMO LRU**

Tempo:       Pagina entrante:

N. Page Fault:       Pagina uscente:

**Situazione dei Frame**

Tempo 0	tempo1	tempo2	tempo3
3PF	3	3	5PF
libero	2PF	2	2
libero	libero	6PF	6

La finestra che appare quando viene scelta questa procedura, mostra quanto avviene per ogni singolo riferimento della stringa.

Ad ogni passo indica:

- Tempo di esecuzione.
- Pagina entrante.
- Pagina uscente.
- Numero complessivo di page fault.

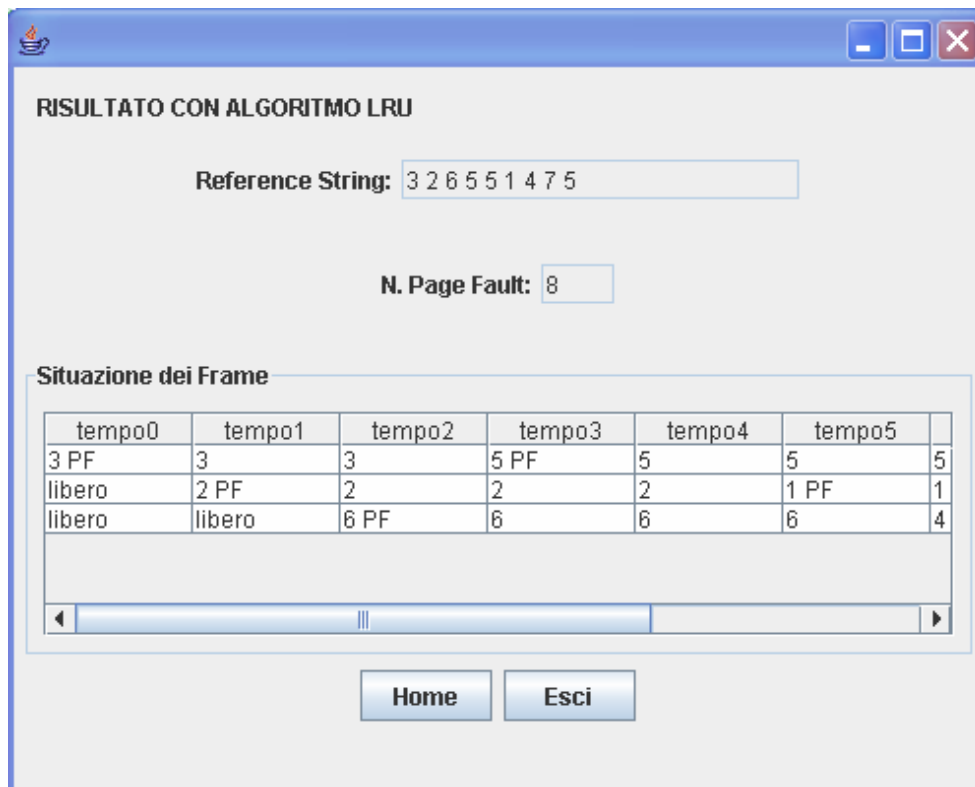
Visualizza, per ogni istante, la situazione dei frame: il nuovo inserimento della pagina, qualora si verifichi un page fault, è indicato nella cella con “PF”.

Per visualizzare il tempo successivo si preme il pulsante “AVANTI”.

Terminati gli indici della Reference String, il pulsante “AVANTI” prende il nome “HOME,” e permette di ritornare alla finestra di inserimento dati per un’ulteriore immissione.

Si può in ogni momento uscire dal programma cliccando il pulsante “ESCI”.

### 3.3 LA PROCEDURA “RISULTATO”



**RISULTATO CON ALGORITMO LRU**

Reference String:

N. Page Fault:

**Situazione dei Frame**

tempo0	tempo1	tempo2	tempo3	tempo4	tempo5	
3 PF	3	3	5 PF	5	5	5
libero	2 PF	2	2	2	1 PF	1
libero	libero	6 PF	6	6	6	4

Home Esci

La procedura “Risultato” permette di vedere immediatamente la tabella con la situazione dei frame per tutti i riferimenti della Reference String.

Fornisce inoltre il numero di Page Fault.

Il pulsante “HOME” permette di ritornare alla finestra di inserimento dati per un ulteriore immissione.

Si può uscire dal programma cliccando il pulsante “ESCI”.