



UNIVERSITÀ DEGLI STUDI DI MESSINA

FACOLTÀ DI MATEMATICA E INFORMATICA

Manuale d'uso del progetto:
Politiche di sostituzione delle pagine (FIFO e LRU)



Descrizione generale del progetto

Obiettivo:

L'obiettivo è l'implementazione di un applicativo che analizzi gli algoritmi di sostituzione delle pagine FIFO (First-in, First-out), LRU (Least Recently Used) e algoritmi che approssimano LRU. Supponiamo vi siano n processi e che la memoria venga assegnata in maniera proporzionale alla dimensione del processo.

Criteri generali:

Il progetto è stato realizzato utilizzando il linguaggio di programmazione Python.

La versione di Python utilizzata è 2.7.10

Per utilizzare il programma `memory.py` è necessaria la libreria `"prettytable.py"` (nella stessa directory dell'applicativo).

La libreria `"prettytable.py"` viene utilizzata per stampare i dati in output in una tabella (Fonte: <https://github.com/dprince/python-prettytable>)

E' consigliabile allargare la finestra del terminale almeno a 160x50

Manuale utente

L'applicativo va lanciato seguendo la seguente sintassi:

\$ python memory.py <Numero Max di Frame> <Numero Min Frame per Processo> <Numero di Processi> <Lunghezza della Successione dei riferimenti>

Nel caso in cui la sintassi non venga rispettata, viene visualizzato un messaggio a schermo:

```
Host-01:sistemi_operativi salvatorenitopi$ python memory.py

Parametri non validi:
python memory.py 0 0 0 0

Parametri richiesti:

<max_frame>          Numero massimo di frame
<p_min_frame>        Numero minimo di frame per processo
<n_processi>         Numero dei processi
<l_riferimenti>      Lunghezza della successione dei riferimenti

Esempio:
python memory.py <max_frame> <p_min_frame> <n_processi> <l_riferimenti>

-----
Progetto realizzato da: Salvatore Nitopi
```

Successivamente viene richiesto all'utente di inserire la dimensione di ogni singolo processo e di scegliere l'algoritmo da utilizzare.

```
Host-01:sistemi_operativi salvatorenitopi$ python memory.py 3 1 2 5
Inserisci la dimensione del processo 0: 2
Inserisci la dimensione del processo 1: 3

1)Usa FIFO

2)Usa LRU

Quale Algoritmo vuoi usare? █
```

In seguito il programma esegue l'algoritmo scelto e stampa a schermo l'output.

```
Host-01:sistemi_operativi salvatorenitopi$ python memory.py 3 1 2 5
Inserisci la dimensione del processo 0: 2
Inserisci la dimensione del processo 1: 3

1)Usa FIFO
2)Usa LRU

Quale Algoritmo vuoi usare? 1

Calcolo dimensione totale processi: 5
```

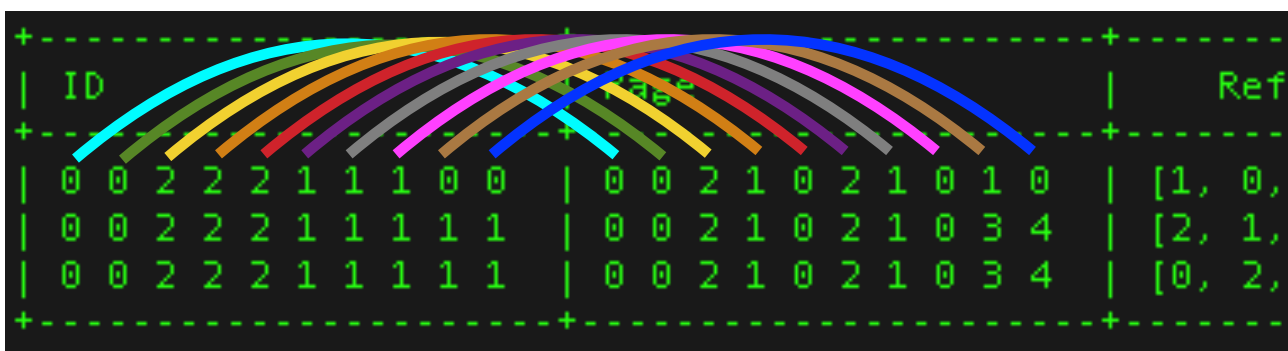
ID	Page	Referenze	Processo	PageFault	Frames	Conclusioni
1 0 0	0 1 0	[1, 1, 1, 1, 0]	0	0	3	PageFault processo 0: 0 con 3 frames
1 1 1	0 1 2	[0, 0, 1, 0, 2]	1	2	3	PageFault processo 1: 2 con 3 frames
1 1 0 0	1 0 1 0	[1, 1, 1, 1, 0]	0	0	4	PageFault processo 0: 0 con 4 frames
1 1 1 0	1 0 2 0	[0, 0, 1, 0, 2]	1	1	4	PageFault processo 1: 1 con 4 frames
0 1 1 0 0	0 1 0 1 0	[1, 1, 1, 1, 0]	0	0	5	PageFault processo 0: 0 con 5 frames
0 1 1 1 0	0 1 0 2 0	[0, 0, 1, 0, 2]	1	1	5	PageFault processo 1: 1 con 5 frames

```
Totale PageFault: 4
```

Ad ogni esecuzione viene stampata una tabella (vedi esempio) dove vengono elencati: Gli id dei processi, le pagine possedute da ciascun processo, referenze, processo corrente, PageFault, numero di frames, Commento conclusivo.

Una copia dell'output viene salvata in un file denominato: test_000.txt dove 000 sarà sostituito al timestamp dell'esecuzione.

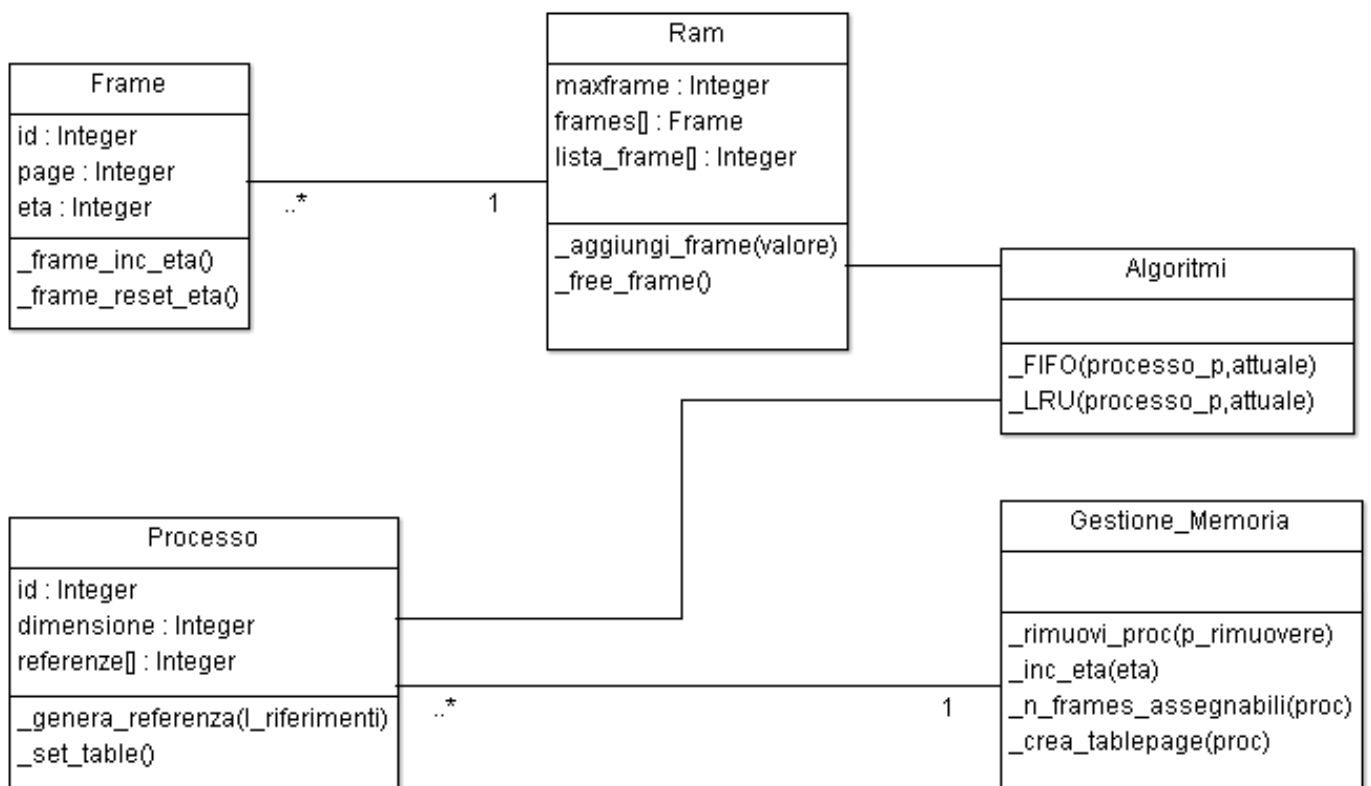
Attenzione: le prime due colonne rappresentano lo stato della memoria, quindi gli id dei processi e le pagine possedute da ciascun processo devono essere interpretati come segue:



Linea ciano: al processo con ID 0 corrisponde la pagina 0
 Linea verde: al processo con ID 0 corrisponde la pagina 0

Linea gialla:	al processo con ID 2 corrisponde la pagina 2
Linea arancione:	al processo con ID 2 corrisponde la pagina 1
Linea rossa:	al processo con ID 2 corrisponde la pagina 0
Linea viola:	al processo con ID 1 corrisponde la pagina 2
Linea grigia:	al processo con ID 1 corrisponde la pagina 1
Linea magenta:	al processo con ID 1 corrisponde la pagina 0
Linea marrone:	al processo con ID 0 corrisponde la pagina 1
Linea blu:	al processo con ID 0 corrisponde la pagina 0

Class Diagram



Analisi e Commenti

Lanciando l'applicativo più volte utilizzando i parametri:

Numero massimo di frame: 1

Numero minimo di frame per processo: 3

Numero processi: 2

Lunghezza della successione dei riferimenti: 11

Dimensione processi: 5

e aumentando il numero di frame (come richiesto) incrementando esclusivamente il parametro: `max_frame` (Numero massimo di frame) è possibile generare la seguente tabella ed il seguente grafico:

Numero Massimo di Frame	FIFO	LRU
1	59	71
2	47	63
3	39	47
4	44	39
5	23	18
6	14	14
7	11	11
8	7	7
9	4	2
10	2	2

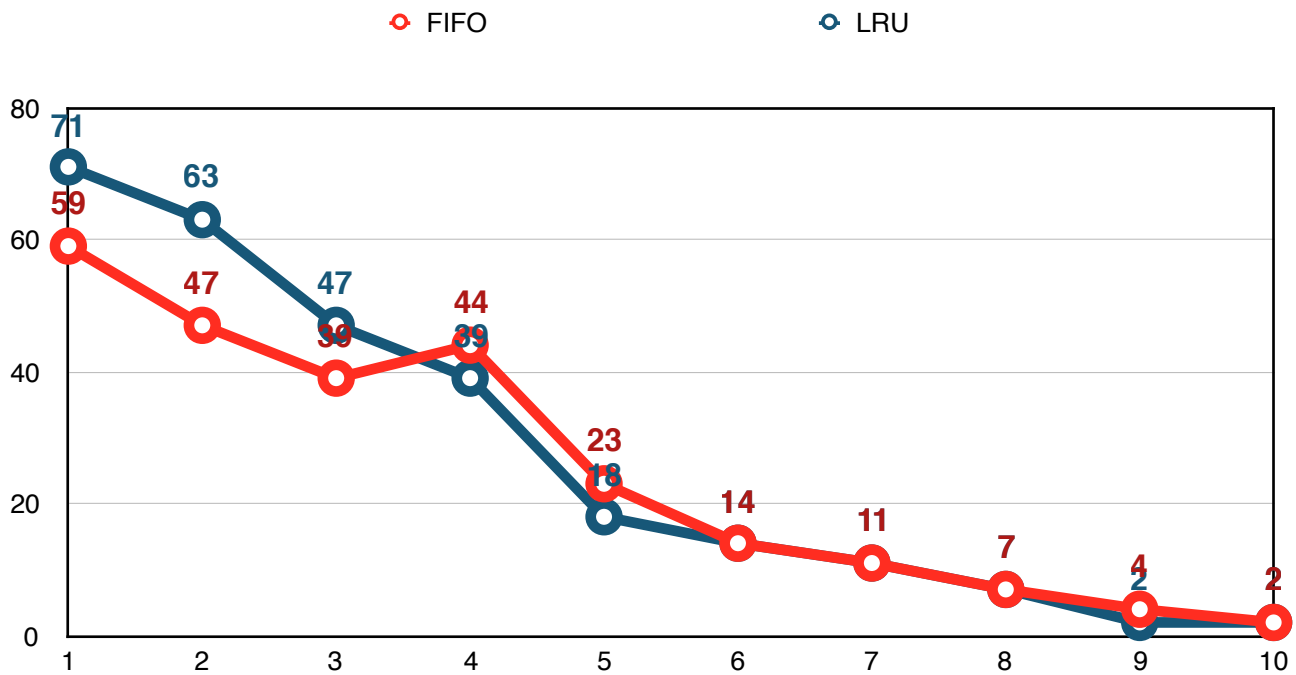


Grafico Teorico FIFO

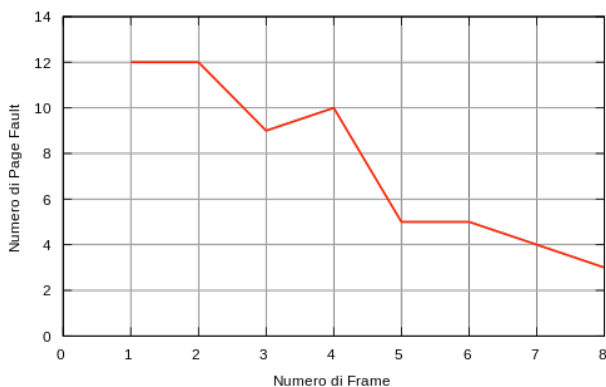
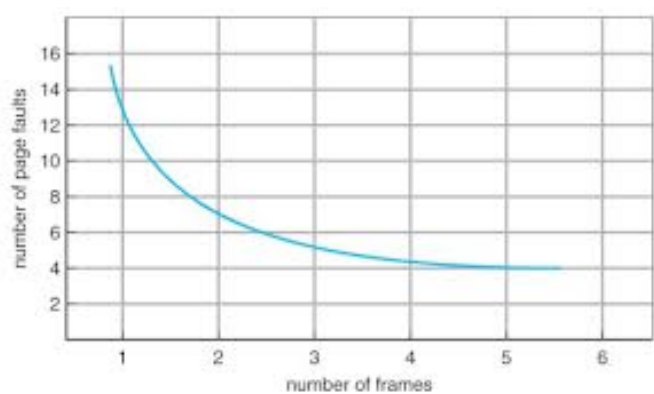


Grafico Teorico LRU



Si può quindi affermare che entrambi sono ottimi algoritmi per la gestione della memoria ma dipendono dalla sequenza delle reference del processo. Infatti il grafico risultante è molto simile a quelli teorici, ma per evidenziare l'anomalia di Belady sono state necessarie numerose prove visto che le reference vengono generate casualmente dall'applicativo.