

La gestione della memoria centrale – Esercizi risolti

1) Un sistema a memoria paginata ha un tempo di accesso alla memoria primaria di 100 nanosecondi. Per gestire un page fault occorrono mediamente 10 mS se è disponibile una pagina vuota in memoria centrale, o se la pagina da sostituire non è modificata, e 24 mS se si deve sostituire una pagina modificata. Se nel 50% dei casi la pagina da sostituire è modificata, qual'è la probabilità accettabile di page fault per un tempo medio di accesso di 200 nanosecondi? Qual è il tempo medio di accesso per una probabilità di page fault di 10^{-4} ?

Soluzione

La formula generale è la seguente, dove tm è il tempo di accesso medio, p è la probabilità di page fault, tc è il tempo di accesso a memoria centrale, e tf è il tempo di servizio di un page fault (in cui si trascura il tempo di accesso a memoria centrale perché molto piccolo rispetto all'accesso a disco):

$$tm = (1 - p) \times tc + p \times tf$$

Nel nostro caso, essendo la probabilità di sostituire una pagina modificata pari al 50% dei page fault, il tempo medio di servizio di un page fault è 17 mS (media tra 10 mS e 24 mS). Essendo $tm = 200$ nS e $tc = 100$ nS, si ha (esprimendo i tempi in nanosecondi):

$$\begin{aligned} 200 &= (1 - p) \times 100 + p \times 17 \times 10^6 \\ 100 &= p \times 17 \times 10^6 - 100 \times p \end{aligned}$$

arrotondando si ha $p = 1 / 170.000$, cioè un page fault ogni 170.000 accessi a memoria.

Con la stessa formula si calcola il tempo medio di accesso per una probabilità di page fault di 10^{-4} :

$$tm = (1 - 10^{-4}) \times 100 + 10^{-4} \times 17 \times 10^6 \approx 100 + 1.700 = 1,8 \mu S$$

2) La memoria virtuale di un calcolatore è divisa in pagine di 512 byte. Dati i seguenti frammenti di programma in linguaggio C:

```
int x[32][128];
int r, c;
...
for (r = 0; r < 32; r++)
    for (c = 0; c < 128; c++)
        x[r][c] = 0;
```

(a)

```
int x[32][128];
int r, c;
...
for (c = 0; c < 128; c++)
    for (r = 0; r < 32; r++)
        x[r][c] = 0;
```

(b)

dire quanti page fault vengono generati nell'ipotesi che

- per i dati vengano usati 8 page frame, inizialmente non allocati
- i dati di tipo int occupino ciascuno 4 byte
- la matrice sia memorizzata per righe in modo contiguo e compatto
- gli unici dati allocati in memoria paginata siano gli elementi della matrice
- si adotti una politica di sostituzione locale delle pagine LRU

Soluzione

Se ogni dato intero occupa 4 byte, ogni riga della matrice occupa $128 \times 4 = 512$ byte, cioè una pagina.

Nel caso (a) la matrice viene azzerata per righe, quindi una volta caricata in memoria la pagina corrispondente alla prima riga questa può essere azzerata senza ulteriori page fault. Quindi si carica in memoria la pagina che contiene la seconda riga, e così via fino all'ottava riga. A questo punto tutte le pagine fisiche a disposizione sono state occupate, e per azzerare la riga di indice 8 (cioè la nona) è necessario riutilizzare una pagina già utilizzata. In base alla politica LRU si riutilizza la prima pagina fisica allocata, che contiene la riga di indice 0, che è quella utilizzata meno recentemente. Si prosegue quindi riutilizzando la seconda pagina, poi la terza, fino all'ottava in cui viene allocata la riga di indice 15. Proseguendo nello stesso modo, si genera un page fault per ogni riga e quindi in totale si hanno 32 page fault.

Nel caso (b) la matrice viene azzerata per colonne, ma l'allocazione è sempre per righe. Viene quindi caricata la prima riga nella prima pagina, e viene azzerato l'elemento di indice $[0][0]$. Per azzerare l'elemento di indice $[0][1]$ è necessario caricare la seconda riga, che occupa la seconda pagina fisica, e così via fino all'elemento di indice $[0][7]$ che occupa l'ottava pagina. Per l'elemento di indice $[0][8]$ è necessario liberare una pagina fisica. Come prima, per la politica LRU viene liberata la prima pagina allocata. Proseguendo in questo modo, risulta che per ogni elemento della matrice da azzerare deve essere allocata una nuova pagina, per un totale di $32 \times 128 = 4096$ page fault.

3) La tabella delle pagine di un processo in esecuzione è la seguente. Tutte le numerazioni sono decimali, il processo utilizza 3 pagine fisiche di 16 Kbyte l'una, i tempi sono espressi in unità convenzionali ed esprimono il tempo trascorso dal caricamento o dall'ultimo accesso alla pagina fisica.

# pag. virtuale	valida	usata	modificata	# pag. fisica	tempo dal caricamento	tempo dall'ultimo uso
0	1	1	0	12	127	123
1	1	1	0	5	130	130
2	1	1	1	3	143	112
3	0	0	0	-	-	-

Per ognuno dei seguenti indirizzi logici: 17.830, 36.215, 62.512, dire

- se un riferimento ad esso genera o no un page fault;
- se sì, qual è la pagina fisica sostituita da un algoritmo:
 - FIFO
 - LRU
 - Clock, supponendo che le pagine vengano esaminate nell'ordine delle righe della tabella a partire dalla prima riga
- e se la pagina vittima deve essere scritta su disco
- qual è l'indirizzo fisico corrispondente (eventualmente dopo la risoluzione del page fault)

Soluzione

Gli indirizzi indicati si trovano nelle seguenti pagine virtuali:

17.830	pagina virtuale 1, offset	1.446 ($17.830 = 1 \times 16.384 + 1.446$)
36.215	pagina virtuale 2, offset	3.447 ($36.215 = 2 \times 16.384 + 3.447$)
62.512	pagina virtuale 3, offset	13.360 ($62.512 = 3 \times 16.384 + 13.360$)

Le pagine virtuali 1 e 2 hanno il bit di validità impostato a 1, quindi i primi due indirizzi sono presenti in memoria centrale. Le corrispondenze con gli indirizzi fisici sono le seguenti.

- indirizzo virtuale 17.830, pagina virtuale 1, pagina fisica 5: l'indirizzo fisico corrispondente è $5 \times 16.384 + 1.446 = 83.366$.
- indirizzo virtuale 36.215, pagina virtuale 2, pagina fisica 3: l'indirizzo fisico corrispondente è $3 \times 16.384 + 3.447 = 52.599$.

L'indirizzo virtuale 62.512 si trova nella pagina virtuale 3 che non è presente in memoria, come indicato dal bit di validità impostato a 0. Un riferimento a questo indirizzo genera quindi un page fault.

- Adottando una politica di sostituzione FIFO si utilizza la pagina fisica 3, che è stata caricata da più tempo. Essendo modificata deve essere scritta su disco. L'indirizzo fisico dopo la risoluzione del page fault è $3 \times 16.384 + 13.360 = 62.512$ (coincide con l'indirizzo logico perché la pagina logica coincide con quella fisica).

- Adottando una politica di sostituzione LRU si riutilizza la pagina fisica 5, che non viene utilizzata da più tempo. Non essendo modificata non deve essere scritta su disco. L'indirizzo fisico che ne risulta è $5 \times 16.384 + 13.360 = 95.280$.
- Con una politica di sostituzione Clock viene riutilizzata la pagina fisica 12. Infatti poiché tutte le pagine sono state utilizzate recentemente i loro bit di uso vengono rimessi a zero, quindi al secondo passaggio viene selezionata la pagina che occupa la prima posizione della tabella. La pagina non è modificata e quindi non deve essere scritta su disco. L'indirizzo fisico risultante è $12 \times 16.384 + 13.360 = 209.968$.

4) Un processo in esecuzione ha uno spazio di indirizzamento di quattro pagine virtuali di 4 Kbyte ciascuna, e viene eseguito utilizzando solo tre frame della memoria fisica. In un certo momento dell'esecuzione la tabella delle pagine è la seguente: le numerazioni sono decimali, i tempi sono espressi in unità convenzionali.

# pagina virtuale	valida	usata	modificata	# frame fisico	Tempo trascorso dal caricamento	Tempo trascorso dall'ultimo uso
0	1	1	0	12	127	123
1	0	-	-	-	-	-
2	1	1	1	3	143	112
3	1	1	0	5	130	130

Per la sequenza di due riferimenti consecutivi agli indirizzi virtuali 4.560 e 10.052, dire

- se l'indirizzamento genera o no un page fault;
- se sì, qual è la pagina fisica sostituita da un algoritmo:
 - FIFO
 - LRU
- e se la pagina vittima deve essere scritta su disco
- qual è l'indirizzo fisico corrispondente (dopo la risoluzione dell'eventuale page fault)

Soluzione

L'indirizzo 4.560 si trova nella pagina virtuale 1 (la seconda) all'offset $4.560 - 4.096 = 464$. L'indirizzo 10.052 si trova nella pagina virtuale 2 (la terza) all'offset $10.052 - 8.192 = 1.860$. Un accesso all'indirizzo 4.560 genera un page fault perché la pagina virtuale 1 non è valida.

1) Secondo una strategia FIFO di sostituzione delle pagine viene selezionato come frame da riutilizzare il frame 3, che ospita la pagina virtuale 2, presente in memoria da più tempo (143 unità di tempo). La pagina virtuale 2 viene quindi invalidata ed essendo modificata deve essere salvata su disco. L'indirizzo 4.560 si troverà all'indirizzo fisico $(3 \times 4.096) + 464 = 12.752$.

Il successivo accesso all'indirizzo 10.052 genera un page fault perché la pagina virtuale 2 è stata invalidata. Viene selezionato come frame da riutilizzare il frame 5, presente in memoria da 130 unità di tempo. Non essendo modificato il contenuto del frame non deve essere salvato. L'indirizzo fisico corrispondente all'indirizzo virtuale 10.052 è $(5 \times 4.096) + 1.860 = 22.340$.

2) Secondo una strategia LRU di sostituzione delle pagine viene selezionato come frame da riutilizzare il frame 5, che ospita la pagina virtuale 3, inutilizzata da più tempo (130 unità di tempo). La pagina virtuale 3 viene invalidata ma non essendo modificata non deve essere salvata. L'indirizzo 4.560 si troverà in questo caso all'indirizzo fisico $(5 \times 4.096) + 464 = 20.944$.

Il successivo accesso all'indirizzo 10.052 non genera page fault perché la pagina virtuale 2 è ancora valida. L'indirizzo fisico corrispondente è $(3 \times 4.096) + 1.860 = 14.148$.

5) Un processo in esecuzione ha uno spazio di indirizzamento contiguo di 64 Kbyte; la memoria fisica del sistema è organizzata in frame di 16Kbyte ciascuno; al processo vengono assegnati tre frame con una strategia locale di sostituzione delle pagine.

La tabella delle pagine del processo è la seguente, dove tutte le numerazioni sono decimali, i numeri di pagina logica e frame fisico partono da zero, i tempi sono espressi in unità convenzionali e indicano il tempo trascorso dal caricamento del frame o dall'ultimo accesso al suo contenuto:

# pagina virtuale	valida	modificata	# frame fisico	tempo dal caricamento	tempo dall'ultimo accesso
0	1	1	4	250	140
1	1	0	7	160	160
2	1	0	2	190	150
3	0	-	-	-	-

Il programma accede consecutivamente agli indirizzi virtuali 58.912 e 24.192. Si considerino gli algoritmi di sostituzione delle pagine FIFO e LRU, e per ciascuno di essi si dica:

- se la sequenza dei due accessi causa, per uno o entrambi gli accessi, un page fault;
- se sì, qual è il frame utilizzato per la sostituzione, e se il suo contenuto deve essere salvato su disco prima della sostituzione;
- qual è l'indirizzo fisico corrispondente, eventualmente dopo la risoluzione del page fault.

Soluzione

L'indirizzo virtuale 58.912 è nella pagina virtuale 3 all'offset 9.760

L'indirizzo virtuale 24.192 è nella pagina virtuale 1 all'offset 7.808

La pagina virtuale 3 non è valida, quindi l'accesso al primo indirizzo genera un page fault.

1) Secondo una strategia FIFO di sostituzione delle pagine il frame vittima è quello da più tempo residente in memoria (quello con il *tempo dal caricamento* più elevato), cioè il frame 4, occupato dalla pagina virtuale 0. Il frame è modificato quindi deve essere salvato su disco. Dopo la risoluzione del page fault l'indirizzo virtuale 58.912 corrisponde all'indirizzo fisico $75.296 (4 * 16384 + 9.760)$

Il successivo accesso alla pagina virtuale 1 non genera page fault, l'indirizzo fisico corrispondente all'indirizzo virtuale 24.192 è $122.496 (7 * 16384 + 7.808)$

2) Secondo un algoritmo LRU di sostituzione delle pagine il frame vittima è il frame da più tempo non usato (quello con il *tempo dall'ultimo accesso* più elevato), cioè il frame 7, occupato dalla pagina virtuale 1. Il frame non è modificato quindi non deve essere salvato su disco. Dopo la risoluzione del page fault l'indirizzo virtuale 58.912 corrisponde all'indirizzo fisico $124.448 (7 * 16384 + 9.760)$

Il successivo accesso alla pagina virtuale 1 genera un page fault, perché la pagina è stata rimossa dalla memoria fisica durante il passo precedente. In questo caso il frame utilizzato per la sostituzione è quello con il secondo valore di tempo più elevato, cioè il frame 2. Il frame non è modificato quindi non deve essere salvato su disco. Dopo la risoluzione del page fault l'indirizzo virtuale corrispondente è 40.576 ($2 * 16384 + 7.808$).

6) Si consideri un processo che utilizza solo tre pagine fisiche (*frame*) di memoria, inizialmente vuote. Il processo accede a locazioni di memoria contenute in sei pagine virtuali, che vengono referenziate nel seguente ordine:

1, 2, 3, 2, 6, 4, 2, 1, 5, 3, 6, 5

Si illustri il comportamento degli algoritmi LRU (Least Recently Used) e FIFO (First In First Out) per la sostituzione delle pagine fisiche evidenziando i page fault che vengono generati e commentando la ragione della loro scelta.

Si confronti quindi il comportamento dei due algoritmi con un ipotetico algoritmo di sostituzione ottima delle pagine fisiche.

Soluzione

La sequenza di sostituzione delle pagine è rappresentata su una tabella avete una riga per ogni frame, le cui colonne rappresentano le configurazioni della tabella delle pagine a seguito dei riferimenti alla memoria. Sotto la tabella sono riportati i casi in cui si genera un page fault. La tabella delle pagine è inizialmente vuota

LRU: sostituisce la pagina che è inutilizzata da più tempo. Quando viene richiesta la pagina 6 viene sostituita la pagina 1 che non è più stata utilizzata dopo il suo caricamento iniziale. Quando viene richiesta la pagina 4 viene sostituita la pagina 3, perché le pagine 2 e 6 sono state utilizzate negli ultimi due accessi. Quando viene nuovamente richiesta la pagina 1, tra quelle presenti viene scaricata la pagina 6 perché non è utilizzata da più tempo rispetto alle pagine 2 e 4. E così via. Il numero di page fault è elevato anche se la strategia LRU è ritenuta molto efficace perché la sequenza degli accessi non segue una regolarità che ci si attenderebbe in base al principio di località, e perché il numero di pagine logiche accedute è elevato rispetto al numero di pagine fisiche disponibili.

Pagina	1	2	3	2	6	4	2	1	5	3	6	5
Frame												
1	1	1	1	1	6	6	6	1	1	1	6	6
2		2	2	2	2	2	2	2	2	3	3	3
3			3	3	3	4	4	4	5	5	5	5
	F	F	F		F	F		F	F	F	F	

FIFO: sostituisce la pagina che è stata caricata da più tempo. Quando viene richiesta la pagina 6 viene sostituita la pagina 1, prima nell'ordine di caricamento. Successivamente per caricare le pagine 6 e 4 vengono sostituite le pagine 2 e 3, in quest'ordine, perché sono le pagine caricate da più tempo. Quando viene richiesta nuovamente la pagina 2 viene generato un ulteriore page fault rispetto all'algoritmo FIFO.

Pagina	1	2	3	2	6	4	2	1	5	3	6	5
Frame												
1	1	1	1	1	6	6	6	1	1	1	6	6
2		2	2	2	2	4	4	4	5	5	5	5
3			3	3	3	3	2	2	2	3	3	3
	F	F	F		F	F	F	F	F	F	F	

Algoritmo ottimo: è un algoritmo teorico di riferimento, che sostituisce la pagina che sarà utilizzata più avanti nel tempo, posticipando quindi il momento di un successivo page fault per richiamarla. Quando viene richiesta la pagina 6, viene sostituita la pagina 3 perché sarà riutilizzata solo dopo numerosi passi, mentre le pagine 1 e 2 saranno utilizzate nuovamente a breve termine. Quando viene richiesta la pagina 4, per lo stesso motivo viene scaricata la pagina 6. Quando viene richiesta la pagina 5, le pagine presenti in memoria non saranno più utilizzate in futuro quindi ognuna di esse può essere scelta come pagina vittima. Nella soluzione scegliamo di seguire l'ordine sequenziale per numero di pagina fisica.

Pagina	1	2	3	2	6	4	2	1	5	3	6	5
Frame												
1	1	1	1	1	1	1	1	1	5	5	5	5
2		2	2	2	2	2	2	2	2	3	3	3
3			3	3	6	4	4	4	4	4	6	6
	F	F	F		F	F			F	F	F	

7) Un processo in esecuzione utilizza quattro pagine virtuali e tre pagine fisiche di 8 Kbyte ciascuna. La tabella delle pagine è la seguente, dove tutte le numerazioni sono decimali, i tempi sono espressi in unità convenzionali ed esprimono il tempo assoluto del caricamento o dell'ultimo accesso alla pagina fisica:

# pagina virtuale	valida	usata	modificata	# frame fisico	tempo del caricamento	tempo dell' ultimo accesso
0	1	1	0	12	73	76
1	0	0	0	–	–	–
2	1	1	0	3	60	90
3	1	1	1	5	70	70

Per la sequenza di accesso in lettura ai due indirizzi virtuali: 9.118 e 20.096, nell'ordine, dire, per gli algoritmi di sostituzione di pagina FIFO e LRU:

- se l'accesso genera o no un page fault;
- se sì, qual è la pagina vittima individuata, e se deve essere scritta su disco o no.
- qual è l'indirizzo fisico corrispondente, eventualmente dopo la risoluzione del page fault.

Soluzione

I due indirizzi virtuali sono allocati così: 9.118, pagina 1, offset 926; 20.096, pagina 2 offset 3.712

Strategia FIFO

L'accesso all'indirizzo 9.118 causa un page fault perché la pagina virtuale 1 non è valida. Viene scelta come pagina vittima la pagina fisica caricata da più tempo, cioè la pagina fisica 3 (tempo assoluto di caricamento = 60). Non essendo modificata non deve essere scritta su disco. L'indirizzo fisico corrispondente è $(3 \times 8.192) + 926 = 25.502$.

La tabella delle pagine si modifica in questo modo (X è il tempo in cui avviene la sostituzione, $X > 90$):

# pagina virtuale	valida	usata	modificata	# pagina fisica	tempo del caricamento	tempo dell' ultimo accesso
0	1	1	0	12	73	76
1	1	1	0	3	X	X
2	0	0	0	–	–	–
3	1	1	1	5	70	70

Il successivo accesso all'indirizzo 20.096 causa un page fault perché la pagina virtuale 2 non è più valida. Viene scelta come pagina vittima la pagina fisica caricata da più tempo, cioè la pagina fisica 5 (tempo assoluto di caricamento = 70). Essendo modificata, deve essere scritta su disco. L'indirizzo fisico corrispondente è $(5 \times 8192) + 3.712 = 44.672$.

La tabella delle pagine si modifica in questo modo (Y è il tempo in cui avviene la sostituzione, $Y > X$)

# pagina virtuale	valida	usata	modificata	# frame fisico	tempo del caricamento	tempo dell' ultimo accesso
0	1	1	0	12	73	76
1	1	1	0	3	X	X
2	1	1	0	5	Y	Y
3	0	0	0	–	–	–

Strategia LRU

Come sopra, l'accesso all'indirizzo 9.118 causa un page fault. Viene scelta come pagina vittima la pagina fisica non utilizzata da più tempo, cioè la pagina fisica 5 (tempo assoluto di ultimo accesso = 70), che deve essere scritta su disco. L'indirizzo fisico corrispondente è $(5 \times 8.192) + 926 = 41.886$.

La tabella delle pagine si modifica in questo modo (X è il tempo in cui avviene la sostituzione, $X > 90$):

# pagina virtuale	valida	usata	modificata	# frame fisico	tempo del caricamento	tempo dell' ultimo accesso
0	1	1	0	12	73	76
1	1	1	0	5	X	X
2	1	1	0	3	60	90
3	–	–	–	–	–	–

Il successivo accesso all'indirizzo 20.096 non causa un page fault perché la pagina virtuale 2 è valida. L'indirizzo fisico corrispondente è $(3 \times 8192) + 3.712 = 28.288$.

8) Si consideri un processo che fa riferimento a 5 pagine virtuali nel seguente ordine:

1, 2, 3, 2, 3, 4, 2, 3, 5, 1, 3, 5

Si consideri una memoria fisica inizialmente vuota di 3 pagine e si illustri il comportamento dei seguenti algoritmi di sostituzione delle pagine, evidenziando i page fault che vengono generati:

1. Ottimo
2. FIFO (First In First Out)
3. LRU (Least Recently Used)

La sequenza di sostituzione delle pagine è rappresentata su una tabella, le cui righe rappresentano le pagine, le colonne le successive configurazioni della tabella delle pagine a seguito dei successivi riferimenti alla memoria. Sotto la tabella sono riportati i casi in cui si genera un page fault. La tabella delle pagine è inizialmente vuota.

Algoritmo ottimo: sostituisce la pagina che richiederà meno page fault in futuro. Quando viene richiesta la pagina 4, viene sostituita la pagina 1 perché è quella che verrà riutilizzata più avanti nel tempo. Quando viene richiesta la pagina 5, sia la pagina 2 sia la pagina 4 non verranno più utilizzate, quindi entrambe sono candidate alla sostituzione. Si hanno perciò due soluzioni equivalenti:

	1	2	3	2	3	4	2	3	5	1	3	5
	1	1	1	1	1	4	4	4	5	5	5	5
		2	2	2	2	2	2	2	2	1	1	1
			3	3	3	3	3	3	3	3	3	3
	F	F	F			F			F	F		

	1	2	3	2	3	4	2	3	5	1	3	5
	1	1	1	1	1	4	4	4	4	1	1	1
		2	2	2	2	2	2	2	5	5	5	5
			3	3	3	3	3	3	3	3	3	3
	F	F	F			F			F	F		

FIFO: sostituisce la pagina che è stata caricata da più tempo. Quando viene richiesta la pagina 4 viene sostituita la pagina 1, successivamente vengono sostituite la pagina 2 e la pagina 3, che sono le pagine caricate per prime, in quest'ordine. Quando viene richiesta nuovamente la pagina 3 (penultimo passaggio) viene sostituita la pagina 4, presente in memoria da più tempo rispetto alle pagine 1 e 5.

	1	2	3	2	3	4	2	3	5	1	3	5
	1	1	1	1	1	4	4	4	4	4	3	3
		2	2	2	2	2	2	2	5	5	5	5
			3	3	3	3	3	3	3	1	1	1
	F	F	F			F			F	F	F	

LRU: sostituisce la pagina che è inutilizzata da più tempo. Quando viene richiesta la pagina 4 viene sostituita la pagina 1 che non è più stata utilizzata dopo il suo caricamento iniziale. Quando viene richiesta la pagina 5 viene sostituita la pagina 4, perchè le pagine 2 e 3 sono state utilizzate negli ultimi due accessi. Quando viene nuovamente richiesta la pagina 1, tra quelle presenti viene scaricata la pagina 2 perché non è utilizzata da più tempo rispetto alle pagine 3 e 5. La strategia LRU in questo caso coincide con uno dei due casi della strategia ottima.

	1	2	3	2	3	4	2	3	5	1	3	5
	1	1	1	1	1	4	4	4	5	5	5	5
		2	2	2	2	2	2	2	2	1	1	1
			3	3	3	3	3	3	3	3	3	3
	F	F	F			F			F	F		

9) Si consideri un processo che utilizza solo tre pagine fisiche (*frame*) di memoria, inizialmente vuote. Il processo accede a locazioni di memoria contenute in sei pagine virtuali, che vengono referenziate nel seguente ordine:

1, 2, 3, 6, 2, 4, 1, 3, 5, 3, 6, 5

Si illustri il comportamento degli algoritmi FIFO (First In First Out) e LRU (Least Recently Used) per la sostituzione delle pagine fisiche evidenziando i page fault che vengono generati e commentando la ragione della loro scelta. Si confronti quindi il comportamento dei due algoritmi con un ipotetico algoritmo di sostituzione ottima della pagine fisiche.

Soluzione

Le tabelle seguenti mostrano la progressione dell'allocazione della memoria. La strategia FIFO rialloca la pagina presente in memoria da più tempo, la strategia LRU la pagina non utilizzata da più tempo. La strategia ottima di riferimento sostituisce la pagina che per più tempo resterà inutilizzata in futuro. Per la strategia ottima si noti che per gestire il riferimento alla pagina virtuale 1 dopo il riferimento alla pagina 4 si è deciso di riutilizzare il frame che contiene la pagina 4, sarebbe stato equivalente riutilizzare il frame che contiene la pagina 2 perché anch'essa non sarà più referenziata in futuro.

FIFO

Pagina	1	2	3	6	2	4	1	3	5	3	6	5
Frame												
1	1	1	6	6	6	6	3	3	3	3	3	3
2		2	2	2	2	4	4	4	5	5	5	5
3			3	3	3	3	1	1	1	1	6	6
Fault	F	F	F	F		F	F	F	F		F	

LRU

Pagina	1	2	3	6	2	4	1	3	5	3	6	5
Frame												
1	1	1	1	6	6	6	1	1	1	1	6	6
2		2	2	2	2	2	2	3	3	3	3	3
3			3	3	3	4	4	4	5	5	5	5
Fault	F	F	F	F		F	F	F	F		F	

Ottima

Pagina	1	2	3	6	2	4	1	3	5	3	6	5
Frame												
1	1	1	1	1	1	1	1	1	5	5	5	5
2		2	2	2	2	4	4	3	3	3	3	3
3			3	6	6	6	6	6	6	6	6	6
Fault	F	F	F	F		F		F	F			

10) Si consideri un processo che utilizza solo quattro pagine fisiche (*frame*) di memoria, inizialmente vuote. Il processo accede a locazioni di memoria contenute in sei pagine virtuali, che vengono referenziate nel seguente ordine:

1, 2, 3, 6, 4, 2, 3, 1, 5, 3, 6, 5

Si illustri il comportamento degli algoritmi LRU (Least Recently Used) e FIFO (First In First Out) per la sostituzione delle pagine fisiche evidenziando i page fault che vengono generati e commentando la ragione della loro scelta.

Si confronti quindi il comportamento dei due algoritmi con un ipotetico algoritmo di sostituzione ottima della pagine fisiche.

FIFO

[illegible]

OPT

Pagina	1	2	3	6	4	2	3	1	5	3	6	5
Frame												
1	1	1	1	6	4	4	4	1	5	5	5	6
2		2	2	2	2	2	2	2	2	2	6	3
3			3	3	3	3	3	3	3	3	3	5
	F	F	F	F	F			F	F		F	

LRU

[illegible]