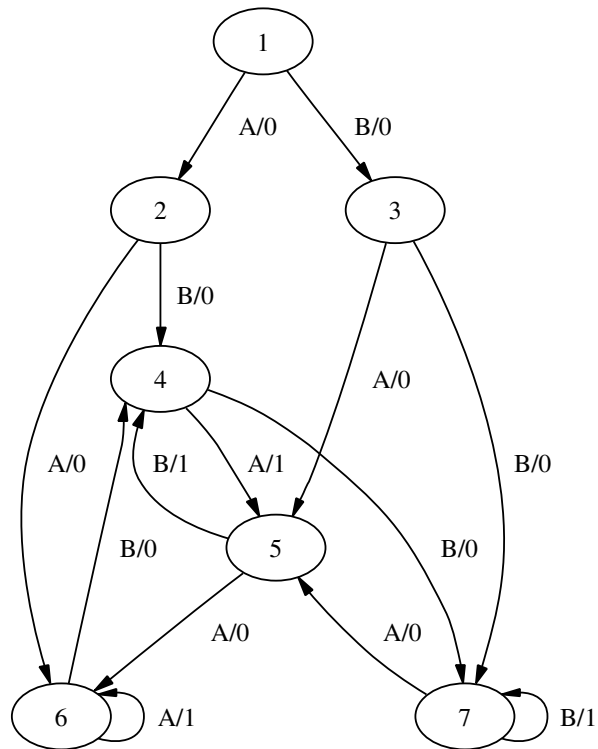


Soluzioni della prova scritta del 25/6/2004

Prof. G. Vaglini, Ing. G. Lettieri

July 12, 2004

Esercizio 1



Esercizio 2

Gli accessi che causano page fault sono: 0, 1, 2, 3, 5, 6, 7, 8. Non è possibile trovare alcun algoritmo che, per questa sequenza di accessi, causi meno di 6 page fault. Se, infatti, proviamo ad usare l'algoritmo di Belady (che rimpiazza

la pagina che verrà richiesta all'istante più lontano nel futuro), otteniamo proprio 6 page fault. Poiché l'algoritmo di Belady è ottimo, non può esistere alcun altro algoritmo che causi meno page fault.

Esercizio 3

- 0 : ($k = 0$) $Cop \rightarrow k$, $M \rightarrow C$, $N \rightarrow D$, 1;
- 1 : ($k = 0$) $0 \rightarrow A$, $A \rightarrow F$, $\bar{C} + 1 \rightarrow C$, 2;
 ($k = 1$) $C + \bar{D} + 1 \rightarrow C$, 4;
- 2 : ($C^n = 1$) $A + F \rightarrow A$, $C + 1 \rightarrow C$, 2;
 ($C^n = 0$) $0 \rightarrow A$, $A \rightarrow F$, $\bar{D} + 1 \rightarrow C$, 3;
- 3 : ($C^n = 1$) $A + F \rightarrow A$, $C + 1 \rightarrow C$, 3;
 ($C^n = 0$) O_1 , 1;
- 4 : ($B^n C^n = 00$) $0 \rightarrow D$, 5;
 ($B^n C^n = 11$) $\bar{B} + 1 \rightarrow B$, $\bar{C} + 1 \rightarrow C$, $0 \rightarrow D$, 5;
 ($B^n C^n = 01$) $0 \rightarrow D$, 6;
 ($B^n C^n = 10$) $\bar{B} + 1 \rightarrow B$, $\bar{C} + 1 \rightarrow C$, $0 \rightarrow D$, 6;
- 5 : ($B^n = 0$) $B + \bar{C} + 1 \rightarrow B$, $D + 1 \rightarrow D$, 5;
 ($B^n = 1$) $D - 1 \rightarrow B$, 0;
- 6 : ($B^n = 0$) $B + C \rightarrow B$, $D + 1 \rightarrow D$, 6;
 ($B^n = 1$) $\bar{D} + 1 + 1 \rightarrow B$, 0;

Esercizio 4

Chiamiamo M la matrice di indirizzo 100 e V il vettore di indirizzo 200. Quando l'esecuzione raggiunge l'etichetta **fine**, il registro AL conterrà il numero di righe della matrice che contengono almeno un elemento diverso da 0. In particolare, il sottoprogramma **conta** conta il numero di elementi diversi da 0 del vettore di 10 elementi memorizzato a partire dall'indirizzo contenuto in EBX, lasciando il risultato nel registro AL. Il programma principale invoca il sottoprogramma **conta** una volta per ogni riga della matrice M , memorizzando i risultati nell'elemento corrispondente del vettore V . Infine, invoca **conta** sul vettore V stesso (alla fine del ciclo, EBX conterrà 200, ovvero l'indirizzo del vettore V).

Si ricorda che le notazioni 200(EBX) e (EBX, ECX) sono casi particolari della notazione generale per l'indirizzamento degli operandi in memoria. Nel primo caso, l'indirizzo dell'operando sarà $200 + ECX$, mentre nel secondo sarà $EBX + ECX$.