

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

2η Προαιρετική Εργασία

Προσομοίωση αλγορίθμων αντικατάστασης σελίδων

ΜΙΜΟΓΛΟΥ ΓΕΩΡΓΙΟΣ - Π17073
2018-19

ΕΚΦΩΝΗΣΗ

Να υλοποιήσετε ένα πρόγραμμα το οποίο να προσομοιώνει μία από τις παρακάτω επιλογές αλγορίθμων:

- Αλγόριθμος της γήρανσης ή
- Αλγόριθμος WSClock.

Περιγραφή υλοποίησης:

1. Θεωρείστε ότι ο πίνακας σελίδων έχει 128 σελίδες (Σ0-Σ127) και ότι η μνήμη έχει 32 πλαίσια σελίδας (Π0-Π31). Στην αρχή όλα τα πλαίσια είναι κενά.
2. Το πρόγραμμα θα παράγει με μία ακολουθία **10.000 αναφορών σελίδας** για τις υπάρχουσες σελίδες μνήμης. Δηλαδή, δημιουργήστε μία τυχαία ακολουθία 10.000 αναφορών για τις 128 σελίδες που υπάρχουν.
3. Ο αλγόριθμος δημιουργίας της ακολουθίας αναφορών θα λειτουργεί ως εξής:
 - ο Ορίστε ως χρονικό διάστημα «διακοπής» κάθε 20 αναφορές (δηλαδή θα έχετε συνολικά 500 διακοπές ελέγχου).
 - ο Για κάθε χρονικό σημείο ελέγχου, δημιουργήστε τις 20 αναφορές ως εξής:
 - ο Κάθε σελίδα που βρίσκεται ήδη σε ένα πλαίσιο από την προηγούμενη διακοπή, θα παραμένει στη μνήμη με πιθανότητα 60%.
 - Επεξήγηση: με βάση τον παραπάνω αλγόριθμο, από τις 20 σελίδες που βρίσκονταν στη μνήμη τη στιγμή T_x , θα πρέπει να παραμείνουν (περίπου) οι 12, και την επόμενη χρονική στιγμή T_{x+1} . Προσέξτε ότι δεν θα παραμένουν πάντα ακριβώς 12 (60%) από τις προηγούμενες 20 για παραμονή, εφόσον κάθε σελίδα παραμένει με πιθανότητα 60% και αντικαθίσταται με πιθανότητα 40%.
 - ο Οι σελίδες που δεν επανεπιλέχθηκαν για παραμονή στη μνήμη από το προηγούμενο βήμα θα αντικατασταθούν με τυχαία επιλογή που θα παράγει το πρόγραμμά σας. Ο αλγόριθμος συνεχίζει μέχρι να συμπληρωθούν με αυτό τον τρόπο οι 20 αναφορές.
 - Επεξήγηση: στο προηγούμενο παράδειγμα, θα πρέπει για τη χρονική στιγμή T_{x+1} να επιλέξετε τυχαία ποια σελίδα αντικαθιστά κάθε μία από τις σελίδες που θα φύγουν από τη μνήμη. Προσέξτε ώστε να εξαιρεί ο αλγόριθμος τυχαίας επιλογής, εκείνες τις σελίδες που παραμένουν στη μνήμη από το προηγούμενο βήμα.
4. Στην αρχή κάθε σελίδα προκαλεί ένα σφάλμα και μετά φορτώνεται σε ένα πλαίσιο. Μόλις γεμίσουν τα πλαίσια, θα γίνεται η αντικατάσταση σελίδας, με βάση τον εκάστοτε αλγόριθμο που υλοποιείτε και τη ροή σελίδων που παράγονται.
5. Για κάθε σελίδα θα πρέπει να διατηρείτε τα bit (A)ναφοράς και (T)ροποποίησης, το χρόνο φόρτωσης και το χρόνο τελευταίας αναφοράς στη σελίδα. Οι χρόνοι μπορεί να οριστούν και με βάση έναν καθολικό μετρητή εντολών που θα χρησιμοποιείτε.
6. Υπολογίστε τον αριθμό σφαλμάτων σελίδας που προκύπτουν για τον/τους αλγόριθμους που έχετε υλοποιήσει.

Πίνακας Περιεχομένων

1. [Εκφώνηση](#)
2. [Πίνακας Περιεχομένων](#)
3. [Η κλάση Page](#)
4. [Η κλάση Form](#)
5. [Η λογική](#)
6. [Παρατηρήσεις](#)
7. [Screenshots](#)

Εισαγωγή

Η εργασία υλοποιήθηκε με το εργαλείο Visual Studio και με τη γλώσσα C#, όπου χρησιμοποιούνται γραφικά για την καλύτερη παρακολούθηση των αποτελεσμάτων. Επίσης, για τις ανάγκες του project, πέρα τον auto-generated κώδικα της φόρμας, δημιουργήθηκε και μια κλάση Page, η οποία περιγράφει μια ένα πλαίσιο σελίδας.

Ως αλγόριθμος αντικατάστασης σελίδας, επιλέχθηκε ο αλγόριθμος της γήρανσης (aging).

Η κλάση Page

Η κλάση αυτή, αποτελείται από 6 properties, 1 constructor και 3 μεθόδους.

Ενδεικτικά:

Properties:

1. **bool Reference**: δείχνει αν αναφέρθηκε η σελίδα στην τελευταία διακοπή.
2. **int LoadTime**: ο εικονικός χρόνος όπου φορτώθηκε στη μνήμη.
3. **int LastReferenceTime**: ο εικονικός χρόνος τελευταίας αναφοράς.
4. **int ID**: ένας ακέραιος αριθμός (0-127) ταυτοποίησης της σελίδας, καθώς και θέση της στον πίνακα σελίδων
5. **BitArray¹ Counter**: αντιπροσωπεύει το μετρητή M.
6. **int Position**: η θέση της σελίδας στη μνήμη (-1 αν δεν βρίσκεται σε αυτή).

Methods:

- A. **Page(int ID)**: ο constructor της κλάσης, αρχικοποιεί όλα τα properties.
- B. **Page Copy()**: αντιγράφει το Page instance σε ένα νέο και το επιστρέφει.
- C. **static void Fill(Page[] disk)**: γεμίζει το array disk με νέες σελίδες.
- D. **void ShiftCounter()**: μετακινεί όλα τα 'bit' του counter μια θέση δεξιά και τοποθετεί το 0 (false) στην πρώτη θέση.

1) [BitArray](https://docs.microsoft.com/en-us/dotnet/api/system.collections.bitarray?view=netframework-4.7.2) είναι μια κλάση της C#, η οποία χρησιμοποιεί ένα bool array για την αναπαράσταση και διαχείριση δυαδικών αριθμών.
<https://docs.microsoft.com/en-us/dotnet/api/system.collections.bitarray?view=netframework-4.7.2>

Η κλάση Form

Η κλάση αυτή, αποτελείται από 8 μεταβλητές, 1 constructor και 4 μεθόδους. Στο γραφικό κομμάτι, αποτελείται από 4 controls. Ενδεικτικά:

Graphics:

1. `StartBtn`: το κουμπί με το οποίο ξεκινάει η προσομοίωση.
2. `richTextBox1`: το παράθυρο όπου εμφανίζονται τα αποτελέσματα.
3. `progressBar1`: δείχνει την πρόοδο της προσομοίωσης.
4. `label1`: δείχνει τα faults που έγιναν κατά την προσομοίωση.

Methods:

- A. `Form1()`: ο constructor της κλάσης.
- B. `void Form1_Load(...)`: τοποθετεί τη φόρμα και το `richTextBox1`, όταν αυτή φορτώνει.
- C. `void GenerateSeq()`: Παράγει την ακολουθία αναφορών.
- D. `Page FindMin()`: Βρίσκει και επιστρέφει την σελίδα που βρίσκεται στη μνήμη και έχει το μικρότερο counter.
- E. `void StartBtn_Click(...)`: ξεκινάει την προσομοίωση.

Variables:

1. `Page[] disk`: προσομοιώνει τον πίνακα σελίδων (ή τον δίσκο για ευκολία).
2. `Page[] memory`: προσομοιώνει την μνήμη.
3. `int[] sequence`: η ακολουθία αναφορών.
4. `int totalFaults`: τα συνολικά σφάλματα που έγιναν.
5. `int currentFaults`: τα σφάλματα που έγιναν στην τελευταία διακοπή.
6. `int currentTime`: ο εικονικός χρόνος.
7. `int pagesInMem`: ο αριθμός των σελίδων στη μνήμη.
8. `string text`: το τελικό κείμενο των αποτελεσμάτων.

Η λογική

- Κατά την εκκίνηση της προσομοίωσης αρχικοποιούνται οι απαραίτητες μεταβλητές.
- Καλείται η συνάρτηση `GenerateSeq()` ώστε να παραχθεί η εκάστοτε ακολουθία. Ουσιαστικά, αν η μνήμη είναι μερικώς γεμάτη, επιλέγεται μια τυχαία σελίδα (συγκεκριμένα ένα ID), ώστε προστεθεί στην ακολουθία, με πιθανότητα 60%. Διαφορετικά, προστίθεται μια σελίδα που δεν βρίσκεται στην μνήμη. Σε κάθε περίπτωση γίνεται έλεγχος ώστε να μην υπάρχουν 2 αναφορές στην ίδια σελίδα.
- Για όλες τις σελίδες στην μνήμη, το bit αναφοράς γίνεται 0 (false), και γίνεται shift στα bit του counter.

- Για κάθε ID σελίδας που βρίσκεται στην ακολουθία αναφορών, ανακτάται η αντίστοιχη σελίδα από την μνήμη.
- Αν η θέση της σελίδας (Position) αντιστοιχεί σε μια θέση μνήμης, δηλαδή η σελίδα βρίσκεται στη μνήμη, τότε ανανεώνονται το bit αναφοράς (Reference), ο μετρητής M (Counter) και ο χρόνος τελευταίας αναφοράς (LastReferenceTime).
- Διαφορετικά, δηλαδή αν $Position == -1$, υπάρχουν 2 περιπτώσεις:
 - Η μνήμη δεν είναι γεμάτη, οπότε τοποθετείται η εκάστοτε σελίδα που κλήθηκε.
 - Η μνήμη είναι γεμάτη, οπότε καλείται η συνάρτηση FindMin και η επιστρεφόμενη σελίδα φεύγει από τη μνήμη, ανανεώνοντας τις πληροφορίες στο δίσκο για τη σελίδα αυτή και στη θέση της τοποθετείται η σελίδα που κλήθηκε.
- Τέλος και στις 2 περιπτώσεις, ανανεώνονται οι απαραίτητες πληροφορίες στη μνήμη αλλά και οι μετρητές σφαλμάτων.
- Αφού γίνουν τα παραπάνω για όλες τις αναφορές, ανανεώνεται και το text με τις πληροφορίες.

Παρατηρήσεις

- Επέλεξα να μην υλοποιήσω το bit τροποποίησης, διότι πιστεύω πως δεν υπάρχει ουσιαστική χρήση του, έστω και πληροφοριακή, στη συγκεκριμένη προσομοίωση.
- Κατά τις δοκιμαστικές εκτελέσεις, παρατήρησα πως, όταν “φορτωνόταν” μια σελίδα (instance Page) από το δίσκο/πίνακα σελίδων (disk[]) στη μνήμη (memory[]), ουσιαστικά περνούσε η πραγματική διεύθυνση μνήμης του instance στο memory[]. Οπότε ό,τι αλλαγές γινόταν σε ένα instance Page στο memory[], “περνούσαν” παράλληλα και στο disk[]. Πρακτικά τα αποτελέσματα ήταν σωστά. Όμως για την θεωρητική ορθότητα της προσομοίωσης, δημιούργησα την μέθοδο Copy(), ώστε να δημιουργείται ένα αντίγραφο του instance με διαφορετική διεύθυνση μνήμης (πραγματική) και οι αλλαγές σε ένα Page πχ. στο memory να μην επηρεάζει το ίδιο Page στο disk.

- Ο χρόνος περάτωσης της προσομοίωσης κυμαίνεται στα 5'' - 9''. Αυτό οφείλεται στο τεράστιο text όπου εμφανίζεται στο richTextBox1. Οι χρόνοι αυτοί επιτεύχθηκαν μετά από προσθήκη κώδικα ο οποίος εμφανίζει τα αποτελέσματα κάθε 50 διακοπές.
- Τα τελικά σφάλματα κυμαίνονται στα 3.900 - 4.600, αφού η συγκεκριμένη υλοποίηση επιλέγει με πιθανότητα 60% σελίδες από τη μνήμη και με 40% σελίδες που δεν βρίσκονται σε αυτή, άρα προκαλούν σφάλμα ($10.000 * 0,4 = 4.000$).
- Περισσότερες τμηματικές πληροφορίες σχετικά με την υλοποίηση υπάρχουν σε σχόλια στον κώδικα.

Screenshots

Form1

Start

TOTAL FAULTS: 4

INTERRUPT: 1

SEQUENCE: 103 125 32 38 102 108 85 18 73 14 62 84 99 16 114 115 83 11 71 100

FRAME 0	ID: 103	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 1	ID: 125	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 2	ID: 32	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 3	ID: 38	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 4	ID: 102	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 5	ID: 108	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 6	ID: 85	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 7	ID: 18	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 8	ID: 73	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 9	ID: 14	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 10	ID: 62	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 11	ID: 84	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 12	ID: 99	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 13	ID: 16	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 14	ID: 114	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 15	ID: 115	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 16	ID: 83	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 17	ID: 11	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 18	ID: 71	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 19	ID: 100	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 10000000
FRAME 20	ID: XX	REFERENCE BIT: XXXX	LOAD TIME: XXXX	LAST REFERENCE TIME: XXXX	COUNTER: XXXXXXXX
FRAME 21	ID: XX	REFERENCE BIT: XXXX	LOAD TIME: XXXX	LAST REFERENCE TIME: XXXX	COUNTER: XXXXXXXX
FRAME 22	ID: XX	REFERENCE BIT: XXXX	LOAD TIME: XXXX	LAST REFERENCE TIME: XXXX	COUNTER: XXXXXXXX
FRAME 23	ID: XX	REFERENCE BIT: XXXX	LOAD TIME: XXXX	LAST REFERENCE TIME: XXXX	COUNTER: XXXXXXXX
FRAME 24	ID: XX	REFERENCE BIT: XXXX	LOAD TIME: XXXX	LAST REFERENCE TIME: XXXX	COUNTER: XXXXXXXX
FRAME 25	ID: XX	REFERENCE BIT: XXXX	LOAD TIME: XXXX	LAST REFERENCE TIME: XXXX	COUNTER: XXXXXXXX
FRAME 26	ID: XX	REFERENCE BIT: XXXX	LOAD TIME: XXXX	LAST REFERENCE TIME: XXXX	COUNTER: XXXXXXXX
FRAME 27	ID: XX	REFERENCE BIT: XXXX	LOAD TIME: XXXX	LAST REFERENCE TIME: XXXX	COUNTER: XXXXXXXX
FRAME 28	ID: XX	REFERENCE BIT: XXXX	LOAD TIME: XXXX	LAST REFERENCE TIME: XXXX	COUNTER: XXXXXXXX
FRAME 29	ID: XX	REFERENCE BIT: XXXX	LOAD TIME: XXXX	LAST REFERENCE TIME: XXXX	COUNTER: XXXXXXXX
FRAME 30	ID: XX	REFERENCE BIT: XXXX	LOAD TIME: XXXX	LAST REFERENCE TIME: XXXX	COUNTER: XXXXXXXX
FRAME 31	ID: XX	REFERENCE BIT: XXXX	LOAD TIME: XXXX	LAST REFERENCE TIME: XXXX	COUNTER: XXXXXXXX

CURRENT FAULTS: 20 TOTAL FAULTS: 20

INTERRUPT: 2

SEQUENCE: 18 61 115 11 39 55 38 14 62 80 9 32 13 125 83 68 85 114 16 108

FRAME 0	ID: 103	REFERENCE BIT: False	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 01000000
FRAME 1	ID: 125	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 1	COUNTER: 10100000
FRAME 2	ID: 32	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 1	COUNTER: 10100000
FRAME 3	ID: 38	REFERENCE BIT: True	LOAD TIME: 0	LAST REFERENCE TIME: 1	COUNTER: 10100000
FRAME 4	ID: 102	REFERENCE BIT: False	LOAD TIME: 0	LAST REFERENCE TIME: 0	COUNTER: 01000000

Εικόνα 1. Εκκίνηση προσομοίωσης και εμφάνιση των πρώτων αποτελεσμάτων.

Form1

Start

TOTAL FAULTS: 4361

FRAME: 20	ID: 49	REFERENCE BIT: True	LOAD TIME: 147	LAST REFERENCE TIME: 147	COUNTER: 10000000
FRAME: 21	ID: 120	REFERENCE BIT: False	LOAD TIME: 145	LAST REFERENCE TIME: 146	COUNTER: 01010000
FRAME: 22	ID: 113	REFERENCE BIT: True	LOAD TIME: 147	LAST REFERENCE TIME: 147	COUNTER: 10000000
FRAME: 23	ID: 16	REFERENCE BIT: True	LOAD TIME: 147	LAST REFERENCE TIME: 147	COUNTER: 10000000
FRAME: 24	ID: 68	REFERENCE BIT: True	LOAD TIME: 140	LAST REFERENCE TIME: 147	COUNTER: 10010101
FRAME: 25	ID: 10	REFERENCE BIT: True	LOAD TIME: 147	LAST REFERENCE TIME: 147	COUNTER: 10000000
FRAME: 26	ID: 89	REFERENCE BIT: False	LOAD TIME: 142	LAST REFERENCE TIME: 146	COUNTER: 01001001
FRAME: 27	ID: 117	REFERENCE BIT: True	LOAD TIME: 147	LAST REFERENCE TIME: 147	COUNTER: 10000000
FRAME: 28	ID: 85	REFERENCE BIT: True	LOAD TIME: 145	LAST REFERENCE TIME: 147	COUNTER: 10101000
FRAME: 29	ID: 13	REFERENCE BIT: False	LOAD TIME: 145	LAST REFERENCE TIME: 146	COUNTER: 01010000
FRAME: 30	ID: 4	REFERENCE BIT: False	LOAD TIME: 146	LAST REFERENCE TIME: 146	COUNTER: 01000000
FRAME: 31	ID: 75	REFERENCE BIT: True	LOAD TIME: 146	LAST REFERENCE TIME: 147	COUNTER: 10100000

CURRENT FAULTS: 10

TOTAL FAULTS: 1303

INTERRUPT: 149

SEQUENCE: 16 46 80 113 93 67 109 35 95 32 0 73 10 64 30 82 120 24 97 90

FRAME: 0	ID: 93	REFERENCE BIT: True	LOAD TIME: 148	LAST REFERENCE TIME: 148	COUNTER: 10000000
FRAME: 1	ID: 24	REFERENCE BIT: True	LOAD TIME: 148	LAST REFERENCE TIME: 148	COUNTER: 10000000
FRAME: 2	ID: 15	REFERENCE BIT: False	LOAD TIME: 146	LAST REFERENCE TIME: 147	COUNTER: 01010000
FRAME: 3	ID: 67	REFERENCE BIT: True	LOAD TIME: 148	LAST REFERENCE TIME: 148	COUNTER: 10000000
FRAME: 4	ID: 64	REFERENCE BIT: True	LOAD TIME: 148	LAST REFERENCE TIME: 148	COUNTER: 10000000
FRAME: 5	ID: 127	REFERENCE BIT: False	LOAD TIME: 146	LAST REFERENCE TIME: 147	COUNTER: 01010000
FRAME: 6	ID: 109	REFERENCE BIT: True	LOAD TIME: 148	LAST REFERENCE TIME: 148	COUNTER: 10000000
FRAME: 7	ID: 80	REFERENCE BIT: True	LOAD TIME: 147	LAST REFERENCE TIME: 148	COUNTER: 10100000
FRAME: 8	ID: 105	REFERENCE BIT: False	LOAD TIME: 144	LAST REFERENCE TIME: 147	COUNTER: 01010010
FRAME: 9	ID: 115	REFERENCE BIT: False	LOAD TIME: 146	LAST REFERENCE TIME: 147	COUNTER: 01010000
FRAME: 10	ID: 8	REFERENCE BIT: False	LOAD TIME: 146	LAST REFERENCE TIME: 147	COUNTER: 01010000
FRAME: 11	ID: 35	REFERENCE BIT: True	LOAD TIME: 148	LAST REFERENCE TIME: 148	COUNTER: 10000000
FRAME: 12	ID: 97	REFERENCE BIT: True	LOAD TIME: 148	LAST REFERENCE TIME: 148	COUNTER: 10000000
FRAME: 13	ID: 107	REFERENCE BIT: False	LOAD TIME: 147	LAST REFERENCE TIME: 147	COUNTER: 01000000
FRAME: 14	ID: 90	REFERENCE BIT: True	LOAD TIME: 142	LAST REFERENCE TIME: 148	COUNTER: 10100100
FRAME: 15	ID: 30	REFERENCE BIT: True	LOAD TIME: 148	LAST REFERENCE TIME: 148	COUNTER: 10000000
FRAME: 16	ID: 32	REFERENCE BIT: True	LOAD TIME: 147	LAST REFERENCE TIME: 148	COUNTER: 10100000
FRAME: 17	ID: 20	REFERENCE BIT: False	LOAD TIME: 147	LAST REFERENCE TIME: 147	COUNTER: 01000000
FRAME: 18	ID: 46	REFERENCE BIT: True	LOAD TIME: 146	LAST REFERENCE TIME: 148	COUNTER: 10101000
FRAME: 19	ID: 95	REFERENCE BIT: True	LOAD TIME: 148	LAST REFERENCE TIME: 148	COUNTER: 10000000
FRAME: 20	ID: 49	REFERENCE BIT: False	LOAD TIME: 147	LAST REFERENCE TIME: 147	COUNTER: 01000000
FRAME: 21	ID: 82	REFERENCE BIT: True	LOAD TIME: 148	LAST REFERENCE TIME: 148	COUNTER: 10000000
FRAME: 22	ID: 113	REFERENCE BIT: True	LOAD TIME: 147	LAST REFERENCE TIME: 148	COUNTER: 10100000
FRAME: 23	ID: 16	REFERENCE BIT: True	LOAD TIME: 147	LAST REFERENCE TIME: 148	COUNTER: 10100000
FRAME: 24	ID: 68	REFERENCE BIT: False	LOAD TIME: 140	LAST REFERENCE TIME: 147	COUNTER: 01001010
FRAME: 25	ID: 10	REFERENCE BIT: True	LOAD TIME: 147	LAST REFERENCE TIME: 148	COUNTER: 10100000
FRAME: 26	ID: 73	REFERENCE BIT: True	LOAD TIME: 148	LAST REFERENCE TIME: 148	COUNTER: 10000000
FRAME: 27	ID: 117	REFERENCE BIT: False	LOAD TIME: 147	LAST REFERENCE TIME: 147	COUNTER: 01000000

Εικόνα 2. Τέλος προσομοίωσης και εμφάνιση όλων των αποτελεσμάτων.

ΤΕΛΟΣ ΠΑΡΟΥΣΙΑΣΗΣ

ΕΥΧΑΡΙΣΤΩ ΠΟΛΥ.