Αρχεία και ευρετήρια κατακερματισμού

Αρχεία Κατακερματισμού

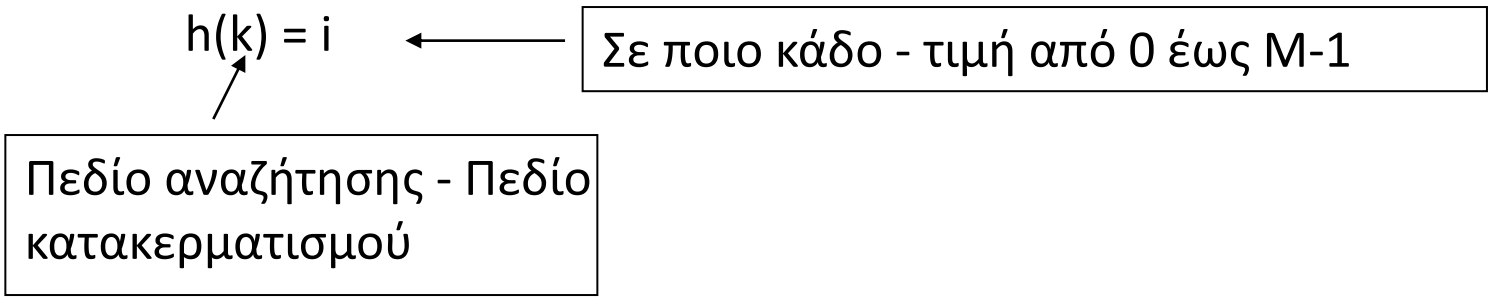
Βασική ιδέα: η τοποθέτηση των εγγραφών στα blocks του αρχείου γίνεται εφαρμόζοντας μια συνάρτηση κατακερματισμού σε κάποιο από τα πεδία της

Εσωτερικός Κατακερματισμός

Εσωτερικός Κατακερματισμός (τα δεδομένα είναι στη μνήμη, όπως στις δομές δεδομένων)

Πίνακας κατακερματισμού με M θέσεις - κάδους (buckets)

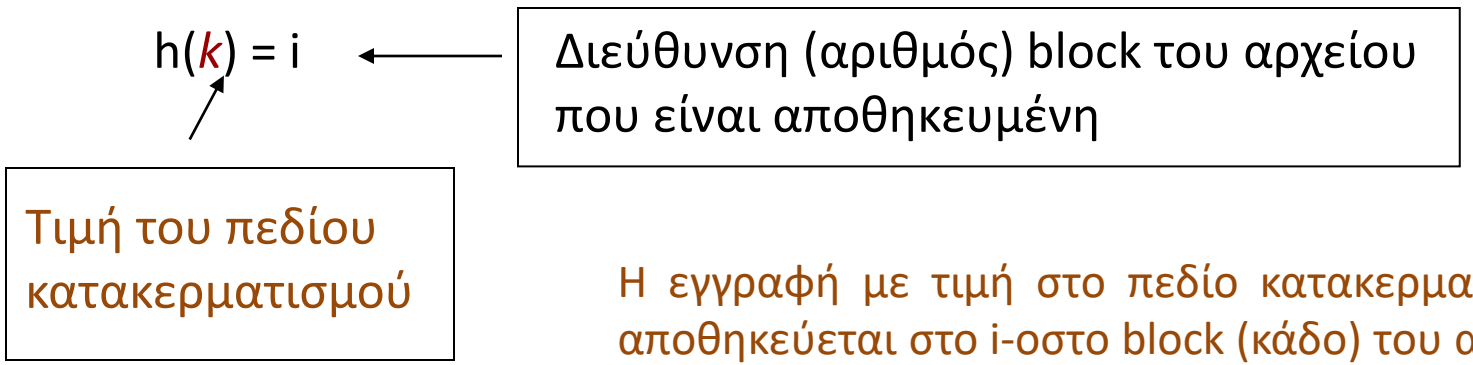
h: *συνάρτηση κατακερματισμού*



Αρχεία Κατακερματισμού

Εξωτερικός Κατακερματισμός (εφαρμογή σε δεδομένα αποθηκευμένα σε αρχεία)

Στόχος



Η εγγραφή με τιμή στο πεδίο κατακερματισμού *k* αποθηκεύεται στο i-οστο block (κάδο) του αρχείου

Κατακερματισμός

h: συνάρτηση κατακερματισμού

(Στόχος) Ομοιόμορφη κατανομή των κλειδιών στους κάδους (blocks)

- Συνηθισμένη συνάρτηση κατακερματισμού:

$$h(k) = k \bmod M$$

Κατακερματισμός

- Σύγκρουση (collision): όταν μια νέα εγγραφή κατακερματίζεται σε μία ήδη γεμάτη θέση

- Καλή συνάρτηση κατακερματισμού: κατανέμει τις εγγραφές ομοιόμορφα στο χώρο των διευθύνσεων (ελαχιστοποίηση συγκρούσεων και λίγες αχρησιμοποίητες θέσεις)

- Ευριστικοί:

  -- αν r εγγραφές, πρέπει να επιλέξουμε το M ώστε το r/M να είναι μεταξύ του 0.7 και 0.9

  -- όταν χρησιμοποιείται η mod τότε είναι καλύτερα το M να είναι πρώτος

Κατακερματισμός

Επίλυση Συγκρούσεων

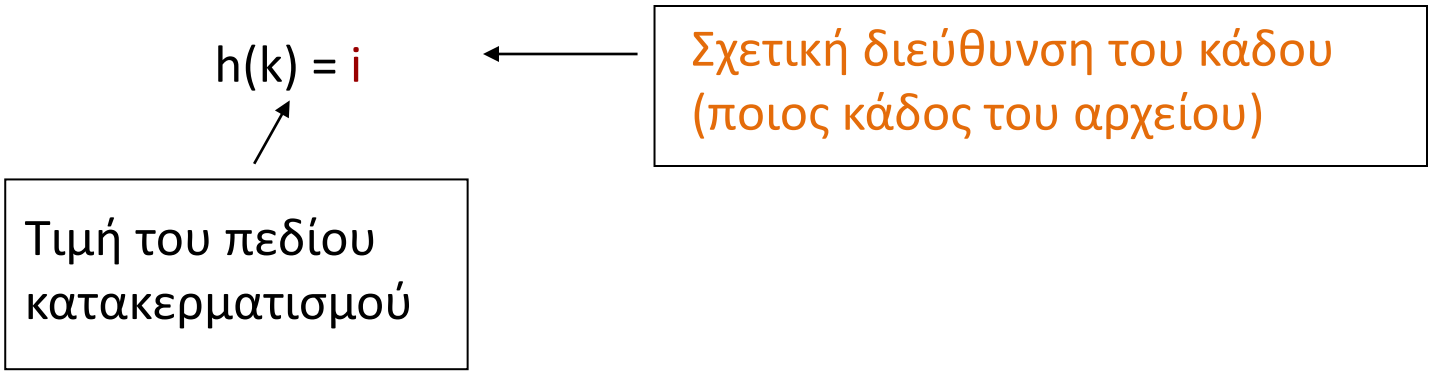
1. *Ανοιχτή Διευθυνσιοδότηση* (open addressing): χρησιμοποίησε την επόμενη κενή θέση

2. *Αλυσιδωτή Σύνδεση* (chaining): για κάθε θέση μια συνδεδεμένη λίστα με εγγραφές υπερχείλισης

3. *Πολλαπλός Κατακερματισμός* (multiple hashing): εφαρμογή μιας δεύτερης συνάρτησης κατακερματισμού

Εξωτερικός Κατακερματισμός

Κάδος: μια συστάδα από συνεχόμενα blocks του αρχείου



Ο κατακερματισμός είναι πολύ αποδοτικός για *επιλογές (αναζητήσεις) ισότητας*

Εξωτερικός Κατακερματισμός

Ένας πίνακας που αποθηκεύεται στην επικεφαλίδα του αρχείου μετατρέπει τον αριθμό κάδου στην αντίστοιχη διεύθυνση block

0	διεύθυνση 1ου block του κάδου στο δίσκο
1	διεύθυνση 1ου block του κάδου στο δίσκο
2	διεύθυνση 1ου block του κάδου στο δίσκο
…	…
M-1	διεύθυνση 1ου block του κάδου στο δίσκο

Εξωτερικός Κατακερματισμός

Συγκρούσεις - αλυσιδωτή σύνδεση - εγγραφές υπερχείλισης ανά κάδο

1. Ανάγνωση όλου του αρχείου (scan)

Έστω ότι διατηρούμε κάθε κάδο γεμάτο κατά 80% άρα ένα αρχείο με μέγεθος B blocks χρειάζεται 1.25 B blocks

$1.25 * B * (T_D + R * T_C)$

2. Αναζήτηση

Συνθήκη ισότητας και μόνο ένα block ανά κάδο: $T_D + R * C$

Αν συνθήκη περιοχής (διαστήματος): scan!

Δυναμικός Κατακερματισμός

Κατακερματισμός

Πρόβλημα στατικού κατακερματισμού:

Έστω M κάδους και r εγγραφές ανά κάδο - το πολύ M * r εγγραφές (αλλιώς μεγάλες αλυσίδες υπερχείλισης)

Δυναμικός κατακερματισμός

- Επεκτατός
- Γραμμικός

Δυναμικός Εξωτερικός Κατακερματισμός

- Δυαδική αναπαράσταση του αποτελέσματος της συνάρτησης κατακερματισμού, δηλαδή ως μια ακολουθία δυαδικών ψηφίων

- Κατανομή εγγραφών με βάση την τιμή των *τελευταίων* (*ή αρχικών)* ψηφίων

- Θα χρησιμοποιήσουμε τα *τελευταία ψηφία*

Δυναμικός Κατακερματισμός (εισαγωγή)

- Το αρχείο ξεκινά με ένα μόνο κάδο

- Μόλις γεμίσει ένας κάδος διασπάται σε δύο κάδους με βάση την τιμή του τελευταίου δυαδικού ψηφίου των τιμών κατακερματισμού -- δηλαδή οι εγγραφές που το τελευταίο ψηφίο της τιμής κατακερματισμού τους είναι 1 τοποθετούνται σε ένα κάδο και οι άλλες (με 0) στον άλλο

- Νέα υπερχείλιση ενός κάδου οδηγεί σε διάσπαση του με βάση το αμέσως επόμενο δυαδικό ψηφίο κοκ

Παράδειγμα

Χρήση των τελευταίων bits της δυαδικής αναπαράστασης

Αποτέλεσμα συνάρτησης κατακερματισμού

1	**000001**
4	**000100**
5	**000101**
7	**000111**
10	**001010**
12	**001100**
15	**001111**
16	**010000**
19	**010011**
21	**010101**
32	**100000**
13	**001101**
20	**010100**

4 εγγραφές ανά κάδο

Δυναμικός Κατακερματισμός

Έτσι δημιουργείται μια δυαδική δενδρική δομή που λέγεται κατάλογος (directory) ή ευρετήριο (index) με δύο ειδών κόμβους

- εσωτερικούς: που καθοδηγούν την αναζήτηση
- εξωτερικούς: που δείχνουν σε ένα κάδο

Δυναμικός Κατακερματισμός (αναζήτηση)

Αλγόριθμος αναζήτησης

```
h := τιμή κατακερματισμού
t  := ρίζα του δέντρου
i := d      /* d πλήθος bit
while (t εσωτερικός κόμβος)
        if (i-οστό bit του h είναι 0)
                t := αριστερά του t
        else t := δεξιά του t
        i := i - 1
```

Δυναμικός Κατακερματισμός

- *Που αποθηκεύεται ο κατάλογος*
  - στη μνήμη, εκτός αν είναι πολύ μεγάλος
  - τότε στο δίσκο – οπότε απαιτούνται επιπρόσθετες προσπελάσεις

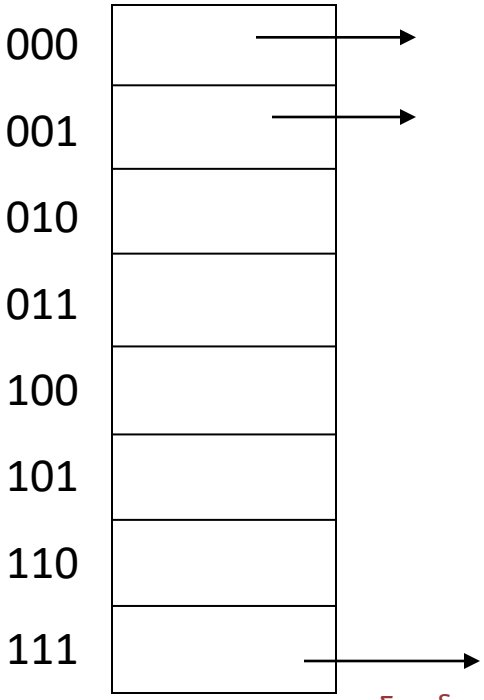
- Δυναμική επέκταση αλλά *μέγιστος αριθμός* επιπέδων (το πλήθος των δυαδικών ψηφίων της συνάρτησης κατακερματισμού)

- Ισοζύγιση

- Συνένωση κάδων (δυναμική συρρίκνωση)

Επεκτατός Κατακερματισμός (extendible hashing)

Ο κατάλογος είναι ένας πίνακας με 2^d διευθύνσεις κάδων (d: *ολικό βάθος* του καταλόγου)



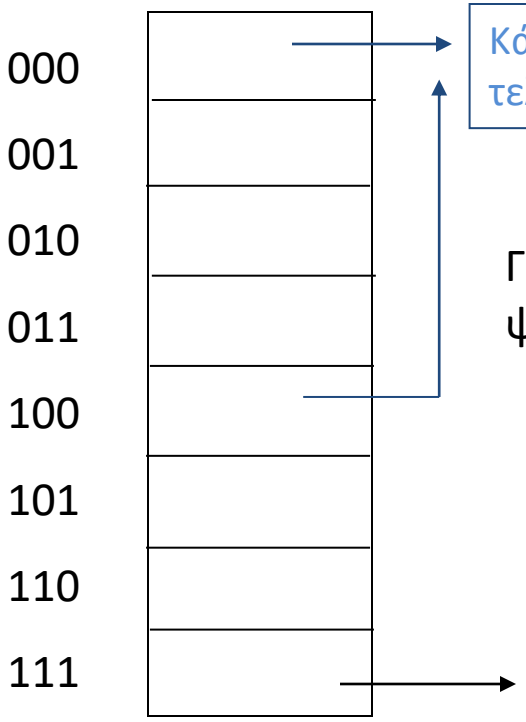
Κάδος για τις εγγραφές με τιμές κατακερματισμού που τελειώνουν σε 000

Τα τελευταία *d* ψηφία της τιμής κατακερματισμού χρησιμοποιούνται ως δείκτης στον πίνακα

Στις διαφάνειες, χρησιμοποιούμε τα τελευταία bits της δυαδικής αναπαράστασης

Επεκτατός Κατακερματισμός

Δε χρειάζεται ένας διαφορετικός κάδος για κάθε μία από τις 2^d θέσεις - μπορεί η θέση του πίνακα να δείχνει στη διεύθυνση του ίδιου κάδου αν αυτές χωράνε σε ένα κάδο



Κάδος για τις εγγραφές με τιμές κατακερματισμού που τελειώνουν από 00

Για κάθε κάδο, τοπικό βάθος d' ο αριθμός των δυαδικών ψηφίων στα οποία βασίζεται η χρήση του κάδου

Παράδειγμα: 2 εγγραφές ανά κάδο

εισαγωγή 2, 4, 3, 10, 7, 9

0010, 0100, 0011, 1010, 0111, 1001

Παράδειγμα

Χρήση των τελευταίων bits της δυαδικής αναπαράστασης

Αποτέλεσμα συνάρτησης κατακερματισμού	
1	000001
4	000100
5	000101
7	000111
10	001010
12	001100
15	001111
16	010000
19	010011
21	010101
32	100000
13	001101

4 εγγραφές ανά κάδο

Επεκτατός Κατακερματισμός

Η τιμή του d μπορεί να αυξάνεται (μέχρι 2^{κ}, κ: αριθμός δυαδικών ψηφίων της τιμής κατακερματισμού) ή να μειώνεται

- Αύξηση της τιμής του d

  Όταν ένας κάδος με τιμή d' = d υπερχειλίσει

  Διπλασιασμός του πίνακα

  Δε χρειάζεται rehash (επανα-κερματισμό),

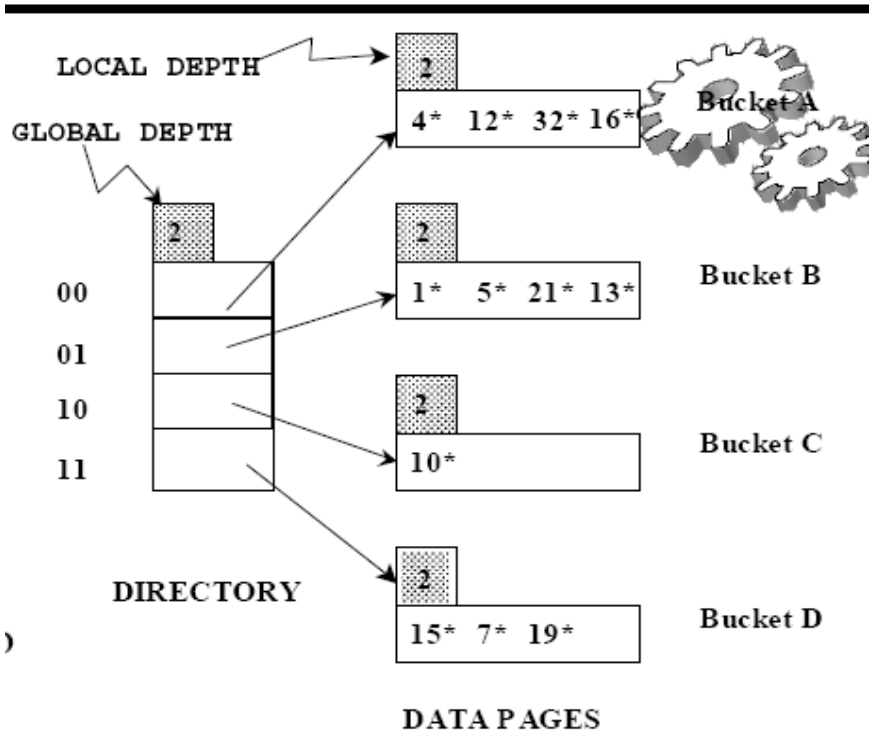
  Μοιράζουμε μόνο τις εγγραφές του κάδου που υπερχείλισε

- Μείωση της τιμής του d

  Όταν για όλους τους κάδους d' < d

  Μείωση του μεγέθους του πίνακα στο μισό

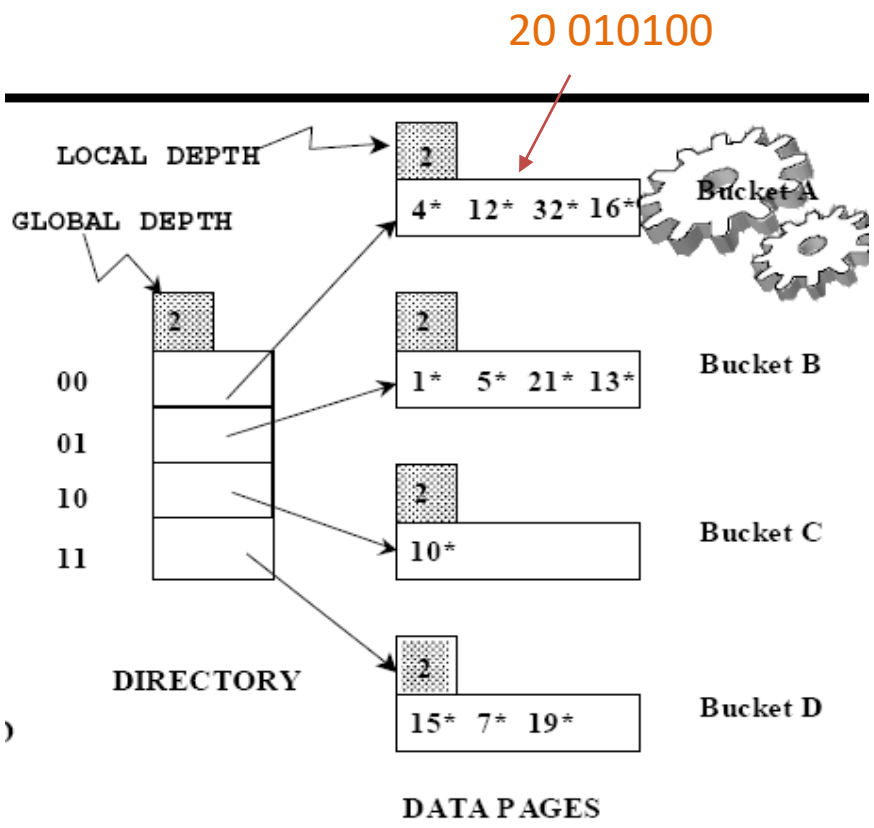
Παράδειγμα



Χρήση των τελευταίων bits της δυαδικής αναπαράστασης

1	000001
4	000100
5	000101
7	000111
10	001010
12	001100
15	001111
16	010000
19	010011
21	010101
32	100000
13	001101
20	010100

Παράδειγμα



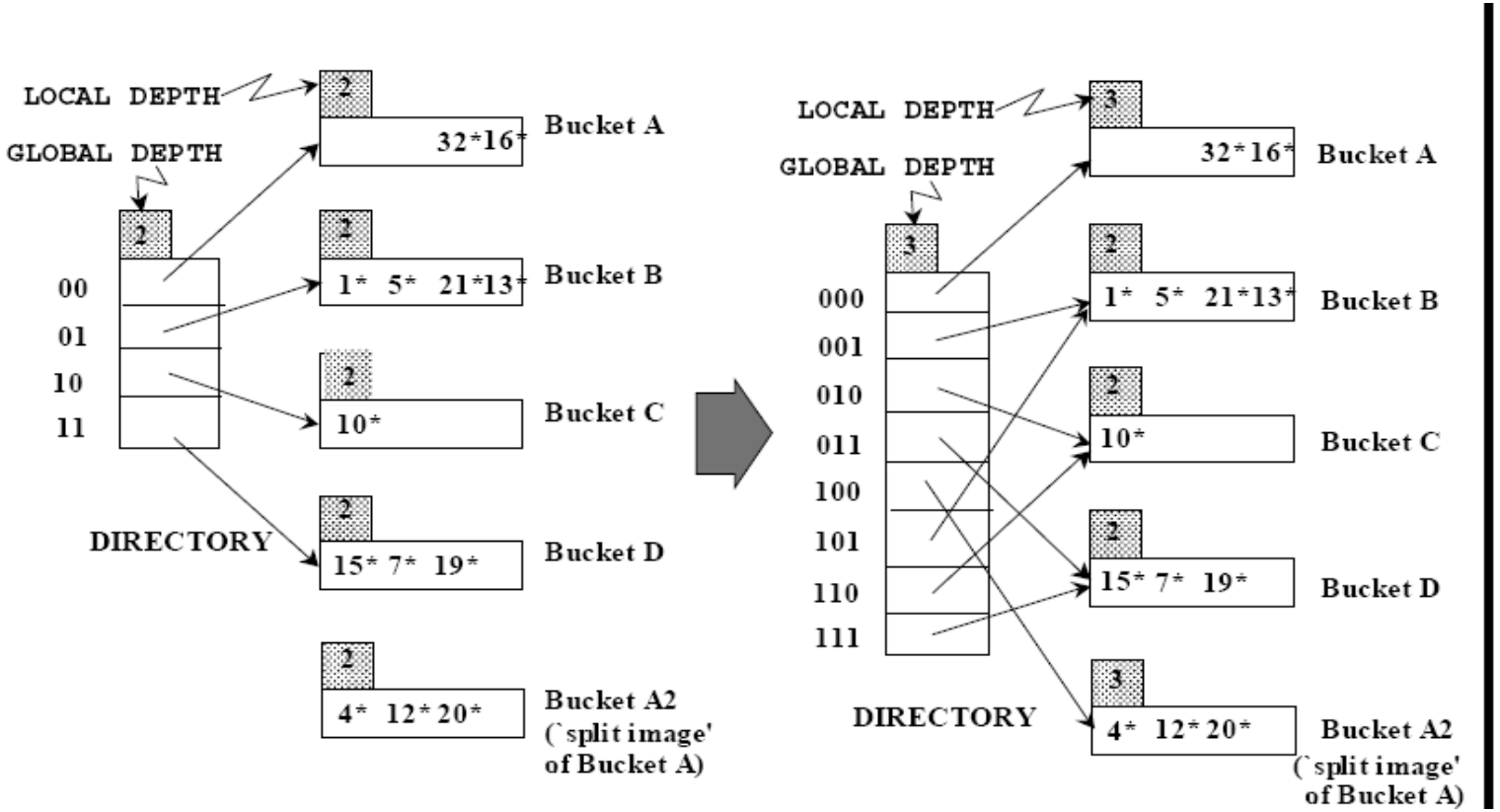
1	000001
4	000100
5	000101
7	000111
10	001010
12	001100
15	001111
16	010000
19	010011
21	010101
32	100000
13	001101
20	010100

Διάσπαση-
νέο ολικό
βάθος 3

Παράδειγμα

1	000001
4	000100
5	000101
7	000111
10	001010
12	001100
15	001111
16	010000
19	010011
21	010101
32	100000
13	001101
20	010100

4 12 32 16 20 -> διάσπαση



Γραμμικός Κατακερματισμός

Θέλουμε να αποφύγουμε τη χρήση καταλόγου και το κόστος διπλασιασμού του μεγέθους του καταλόγου

Προσοχή! Αυτή η μέθοδος:

- Διατηρεί *λίστες υπερχείλισης*
- *Δεν* χρησιμοποιεί τη δυαδική αναπαράσταση

Γραμμικός Κατακερματισμός

Έστω αρχικά M κάδους

Χρησιμοποιεί μια οικογένεια από συναρτήσεις κατακερματισμού

$$h_0(k), h_1(k), \ldots, h_d(k)$$

Κάθε συνάρτηση έχει *διπλάσιους κάδους* από την προηγούμενη:

$$h_0(k) = k \bmod M,\ h_1(k) = k \bmod 2M,\ h_2(k) = k \bmod 4M, \ldots,$$

$$h_j(k) = k \bmod 2^j M$$

Γραμμικός Κατακερματισμός (εισαγωγή)

Έστω M = 2
Ξεκινάμε από την πρώτη συνάρτηση κατακερματισμού (h_0)
Όταν συμβεί *η πρώτη υπερχείλιση (συνθήκη διάσπασης) ενός κάδου*, γίνεται διάσπαση με χρήση της h_1 αλλά όχι του κάδου που υπερχείλισε αλλά του κάδου **0**
Στη συνέχεια, *κάθε κάδος διασπάται με τη σειρά* (δηλαδή, κάδος **1**, **2**, **3**)
Στο στάδιο αυτό χρησιμοποιούνται η h_0 και η h_1
Μέχρι να διασπαστούν και οι 4 κάδοι

Όταν *διασπαστούν όλοι* οι κάδοι,
Οι διασπάσεις θα ξεκινούν από τον κάδο **0** με χρήση της h_2
Πάλι η διάσπαση των κάδων γίνεται *με τη σειρά* (δηλαδή **0**, **1**, **2**, …, **7)**
Στο στάδιο αυτό χρησιμοποιούνται η h_1 και η h_2
Μέχρι να διασπαστούν και οι 8 κάδοι

κοκ

Γραμμικός Κατακερματισμός

Βασικά σημεία

- *Πότε γίνεται διάσπαση;*

Θα θεωρήσουμε ότι γίνεται διάσπαση όταν δημιουργείται ένας κάδος υπερχείλισης (όταν γίνεται εισαγωγή σε ένα γεμάτο κάδο για πρώτη φορά)

- Οι κάδοι σε κάθε βήμα διασπώνται με τη σειρά (ο ένας μετά τον άλλο – ανεξάρτητα αν είναι αυτοί που έχουν ή όχι υπερχειλίσει)

- Πολλές συναρτήσεις κατακερματισμού (δύο σε κάθε βήμα)

Νέα συνάρτηση, όταν διασπαστούν όλοι οι κάδοι με την προηγούμενη συνάρτηση

Γραμμικός Κατακερματισμός

Αρκούν δύο μεταβλητές

- Βήμα Διάσπασης (j) - ποια συνάρτηση χρησιμοποιούμε
- Πλήθος Διασπάσεων (n) – ποιος είναι ο επόμενος κάδος που θα διασπαστεί

Αρχικοποίηση

j = 0; n = 0

Όταν συμβεί μια υπερχείλιση, πρώτη διάσπαση κάδου n = 0, αύξηση n <- n+1

Συνεχίζουμε γραμμικά, διασπώντας με τη σειρά τους κάδους 1, 2, 3, ...

μέχρι να διασπαστούν όλοι οι «παλιοί» κάδοι

η μεταβλητή n («πλήθος διασπάσεων») κρατάει ποιος κάδος έχει σειρά για διάσπαση

Παράδειγμα

32

9

44

31

25

5

35

7

36

14

18

10

11

30

43

M = 4

Κάθε κάδος μέχρι 4 εγγραφές

Αρχικά 4 κάδους

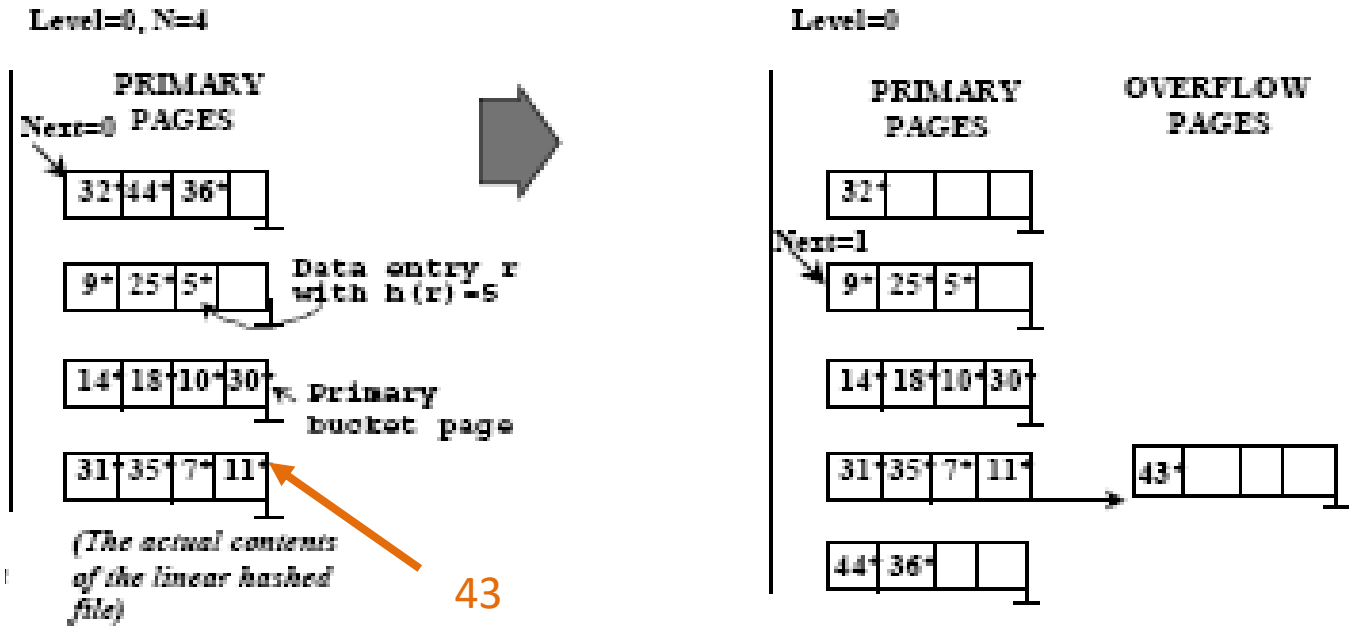
Παράδειγμα

$h_0(k) = k \bmod 4$

$h_1(k) = k \bmod 8$

Για μη διασπασμένους κάδους: παλιά συνάρτηση

Για διασπασμένους κάδους: νέα συνάρτηση



43

Διασπάμε τον πρώτο κάδο

Βήμα διάσπασης 0 (χρήση h_0)

Πλήθος διασπάσεων = 0

Γραμμικός Κατακερματισμός

Όταν συμβεί μια υπερχείλιση σε έναν οποιοδήποτε κάδο,

ο κάδος *n* χωρίζεται σε δύο κάδους:

τον αρχικό κάδο *n* και

ένα νέο κάδο *n + k - 1* στο τέλος του αρχείου

με βάση την συνάρτηση $\mathbf{h_1(k) = k \bmod 2M}$

Δηλαδή, σε κάθε υπερχείλιση χωρίζουμε τον επόμενο στη σειρά κάδο

Γραμμικός Κατακερματισμός

Συνεχίζουμε ...

Όλοι οι κάδοι έχουν διασπαστεί όταν: n = M

Τότε έχουμε 2M κάδους

Όταν **n = M**,

μηδενίζουμε το n, n = 0

και για οποιαδήποτε νέα διάσπαση εφαρμόζουμε την

$$h_2(k) = k \bmod 4M$$

Διασπώντας πάλι τον κάδο 0, 1, ... κ.τ.λ

Γραμμικός Κατακερματισμός

Γενικά βήμα διάσπασης **j** (j = 0, 1, 2, …)

$h_j(k) = k \bmod 2^j M$, και την $h_{j+1}(k)$ για διασπάσεις

Δηλαδή, σε κάθε βήμα έχουμε ένα ζεύγος συναρτήσεων (j, j+1):
η πρώτη χρησιμοποιείται για τους μη διασπασμένους κάδους (δηλαδή, με αριθμό μεγαλύτερο του n) και η δεύτερη για τους διασπασμένους

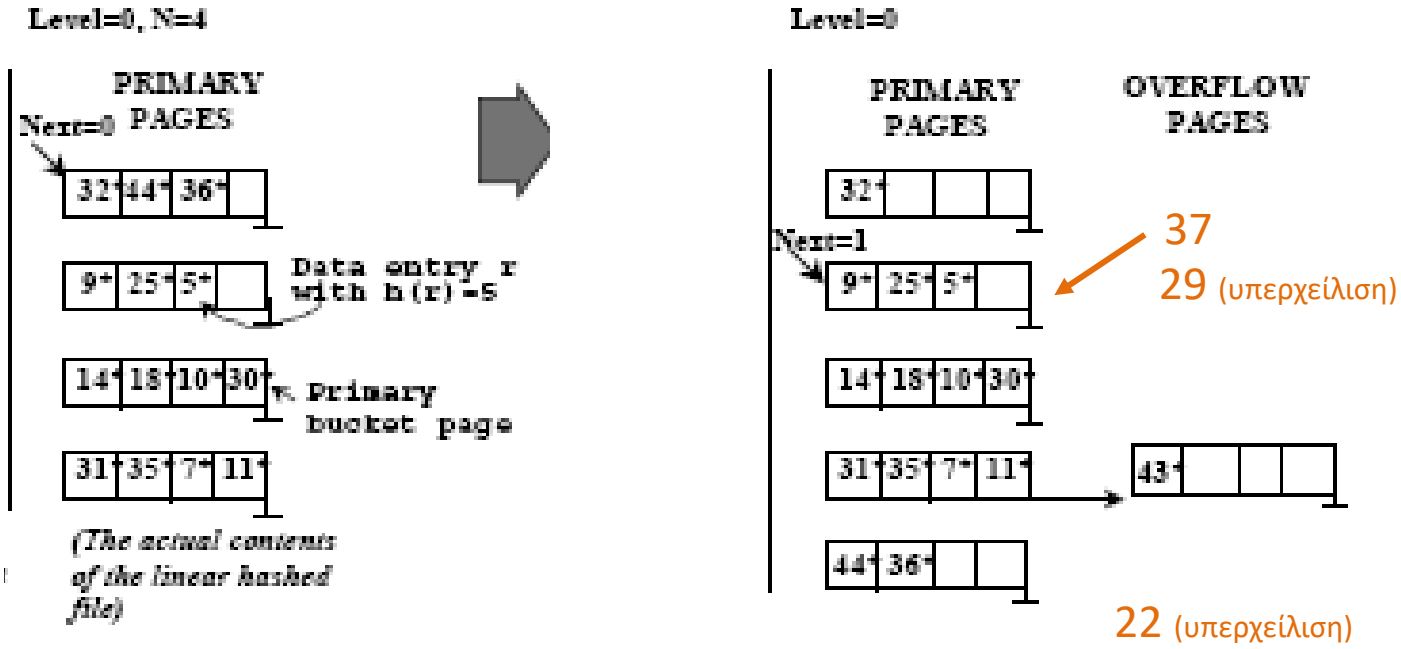
Παράδειγμα

$h_0(k) = k \bmod 4$

$h_1(k) = k \bmod 8$

Για μη διασπασμένους κάδους: παλιά συνάρτηση

Για διασπασμένους κάδους: νέα συνάρτηση

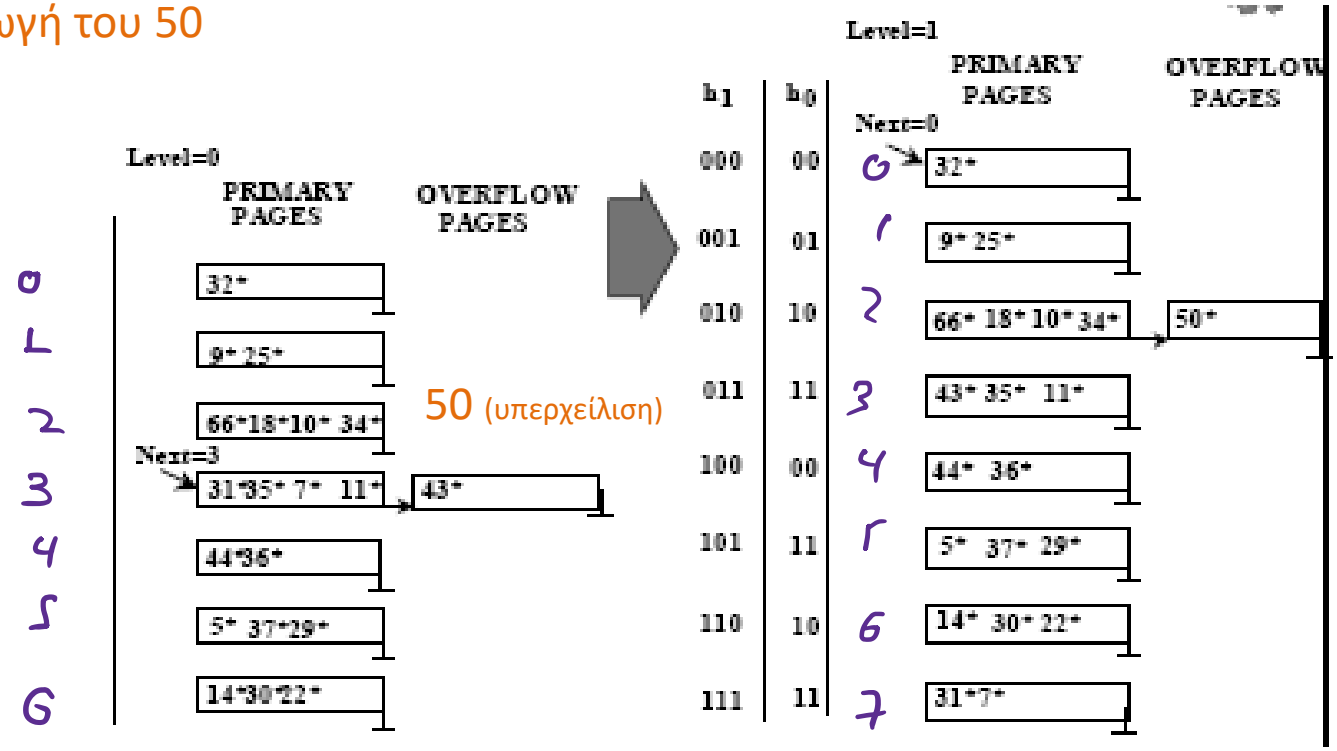


Βήμα διάσπασης 0 (χρήση h_0) 66

Πλήθος διασπάσεων = 1 34

Παράδειγμα

Εισαγωγή του 50



Βήμα διάσπασης 0 (χρήση h_0)

Πλήθος διασπάσεων = 3

Γραμμικός Κατακερματισμός (αναζήτηση εγγραφής)

Τι χρειάζεται να ξέρουμε για να βρεθεί ο κάδος της εγγραφής k που ψάχνουμε;

- ποια συνάρτηση χρησιμοποιούμε (δηλαδή, το j)
- σε ποια διάσπαση βρισκόμαστε (δηλαδή το n)

Έστω ότι είμαστε στο βήμα j,

Τότε θα πρέπει να κοιτάξουμε είτε το

$h_j(k)$ αν ο κάδος δεν έχει διασπαστεί

ή το

$h_{j+1}(k)$ αν έχει διασπαστεί

Πως θα ελέγξουμε αν ο κάδος έχει διασπαστεί ή όχι

Γραμμικός Κατακερματισμός (αναζήτηση)

Έστω n ο αριθμός διασπάσεων και ότι αναζητούμε το k,

βρίσκεται στον κάδο $\mathbf{h_j(k)}$

τότε αν $\mathbf{n \leq h_j(k)}$ ο κάδος δεν έχει διασπαστεί

ενώ αν $\mathbf{n > h_j(k)}$ ο κάδος έχει διασπαστεί και εφαρμόζουμε την $h_{j+1}(k)$

Γραμμικός Κατακερματισμός (αναζήτηση)

Αλγόριθμος Αναζήτησης

j : βήμα διάσπασης n : πλήθος διασπάσεων στο βήμα j

```
if (n = 0)
    then m := h_j(k);
else {
    m := h_j(k);
    if (m < n) then m := h_{j+1}(k)
}
```

σημαίνει ότι ο κάδος έχει διασπαστεί

Κατακερματισμός

Τι αποθηκεύουμε στους κάδους;

Στα παραδείγματα δείχνουμε μόνο την τιμή του πεδίου κατακερματισμού

- Την ίδια την εγγραφή (ως τρόπος *οργάνωσης αρχείου*)
  - μέγεθος κάδου -> 1 block (ή συστοιχία από συνεχόμενα blocks)

Ένα bucket = block (σελίδα)

Στη συνέχεια και ως τρόπος οργάνωσης ευρετηρίου

Ερωτήσεις;

Ασκήσεις

Άσκηση 1

Θεωρείστε ένα ευρετήριο επεκτατού κατακερματισμού, όπου κάθε κάδος (bucket/block) μπορεί να χωρέσει έως *3 εγγραφές*.

(i) Θεωρείστε ότι κάποια στιγμή ο κατάλογος του ευρετηρίου έχει *ολικό βάθος 3*. Ποιο είναι το *μικρότερο* και ποιο το *μεγαλύτερο* δυνατό *τοπικό βάθος*; Δώστε ένα *παράδειγμα* τιμών των οποίων η εισαγωγή οδηγεί σε ένα ευρετήριο όπου κάποιες θέσεις έχουν αυτό το μικρότερο δυνατό τοπικό βάθος.

(ii) Θεωρείστε ότι κάποια στιγμή το ευρετήριο έχει *200 κάδους*. Ποιο είναι το μικρότερο δυνατό *ολικό βάθος* για αυτόν τον κατάλογο;

Ασκήσεις

Άσκηση 2

Θεωρείστε το *ευρετήριο γραμμικού κατακερματισμό* της παρακάτω εικόνας, όπου υπάρχουν 5 κάδοι (και 1 κάδος υπερχείλισης) και ο επόμενος προς διάσπαση κάδος είναι ο 1 (δηλαδή, ο δεύτερος κάδος). Κάθε κάδος χωρά 4 εγγραφές.

(α) Εισάγετε στο ευρετήριο το 4 και μετά το 15 και δώστε το αποτέλεσμα μετά από κάθε εισαγωγή.

(β) Εισάγετε στο αρχικό ευρετήριο της εικόνας το 2 και μετά το 15 και δώστε το αποτέλεσμα μετά από κάθε εισαγωγή.

(γ) Ποια είναι η μικρότερη αριθμητικά τιμή της οποίας η εισαγωγή στο ευρετήριο της εικόνας μπορεί να οδηγήσει στην υπερχείλιση ενός κάδου (δηλαδή, σε ένα κάδο με λίστα υπερχείλισης);

(δ) Υποθέστε ότι μετά από έναν αριθμό από εισαγωγές, υπάρχουν 25 κάδοι (χωρίς τους κάδους υπερχείλισης). Ποιος θα είναι ο επόμενος κάδος προς διάσπαση;