



Κατακερματισμός



Οργάνωση Αρχείων (σύνοψη)

Οργάνωση αρχείων: πως είναι τοποθετημένες οι εγγραφές ενός αρχείου όταν αποθηκεύονται στο δίσκο

1. Αρχεία Σωρού
2. Ταξινομημένα Αρχεία

Φυσική διάταξη των εγγραφών ενός αρχείου με βάση την τιμή ενός από τα πεδία του το οποίο λέγεται **πεδίο διάταξης** (ordering field)



Αρχεία Κατακερματισμού

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



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

Πίνακας κατακερματισμού με M θέσεις - κάδους (buckets)

h : συνάρτηση κατακερματισμού

$$h(k) = i$$

← Σε ποιο κάδο - τιμή από 0 έως $M-1$

Πεδίο αναζήτησης - Πεδίο κατακερματισμού



Εξωτερικός Κατακερματισμός (εφαρμογή σε δεδομένα αποθηκευμένα σε αρχεία)

Στόχος

$$h(k) = i$$

Τιμή του πεδίου κατακερματισμού

Διεύθυνση (αριθμός) block του αρχείου που είναι αποθηκευμένη

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



Κάδος: μια συστάδα από συνεχόμενα blocks του αρχείου

$$h(k) = i$$

Τιμή του πεδίου κατακερματισμού

Σχετική διεύθυνση του κάδου (ποιος κάδος του αρχείου)

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

Εξωτερικός Κατακερματισμός



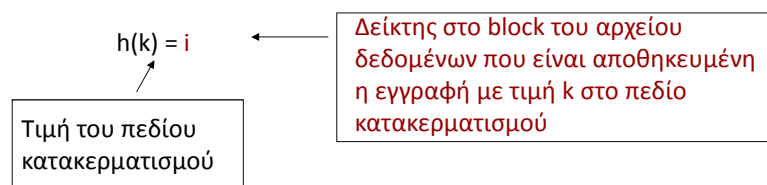
Ένας πίνακας που αποθηκεύεται στην επικεφαλίδα του αρχείου μετατρέπει τον αριθμό κάρου στην αντίστοιχη διεύθυνση block

| | |
|-----|---|
| 0 | διεύθυνση 1ου block του κάρου στο δίσκο |
| 1 | διεύθυνση 1ου block του κάρου στο δίσκο |
| 2 | διεύθυνση 1ου block του κάρου στο δίσκο |
| ... | ... |
| M-1 | διεύθυνση 1ου block του κάρου στο δίσκο |

Εξωτερικός Κατακερματισμός



Πέρα από την οργάνωση αρχείων, ο κατακερματισμός μπορεί να χρησιμοποιηθεί ως ευρετήριο



Εξωτερικός Κατακερματισμός



Πρόβλημα: Στατικός Κατακερματισμός

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

Δυναμικός Κατακερματισμός

- Επεκτατός
- Γραμμικός

Δυναμικός Εξωτερικός Κατακερματισμός



Δυναμικός Εξωτερικός Κατακερματισμός

- Διαδική αναπαράσταση του αποτελέσματος της συνάρτησης κατακερματισμού, δηλαδή ως μια ακολουθία δυαδικών ψηφίων
- Κατανομή εγγραφών με βάση την τιμή των αρχικών (ή τελικών) ψηφίων

Δυναμικός Εξωτερικός Κατακερματισμός



- Το αρχείο ξεκινά με **ένα** μόνο κάδο
- Μόλις γεμίσει ένας κάδος διασπάται σε δύο κάδους με βάση **την τιμή του 1ου (ή τελευταίου) δυαδικού ψηφίου** των τιμών κατακερματισμού -- δηλαδή οι εγγραφές που το πρώτο (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

Δυναμικός Εξωτερικός Κατακερματισμός



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

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

Δυναμικός Εξωτερικός Κατακερματισμός (Παράδειγμα)



Χρήση των τελευταίων bits της δυαδικής αναπαράστασης

| | | | |
|---------------------------------------|----|--------|---------------------|
| Αποτέλεσμα συνάρτησης κατακερματισμού | 1 | 000001 | |
| | 4 | 000100 | |
| | 5 | 000101 | |
| | 7 | 000111 | |
| | 10 | 001010 | |
| | 12 | 001100 | 4 εγγραφές ανά κάδο |
| | 15 | 001111 | |
| | 16 | 010000 | |
| | 19 | 010011 | |
| | 21 | 010101 | |
| | 32 | 100000 | |
| | 13 | 001101 | |
| | 20 | 010100 | |

Δυναμικός Εξωτερικός Κατακερματισμός



Αλγόριθμος αναζήτησης

h := τιμή κατακερματισμού

t := ρίζα του δέντρου

i := 1

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 Εξωτερικός Κατακερματισμός



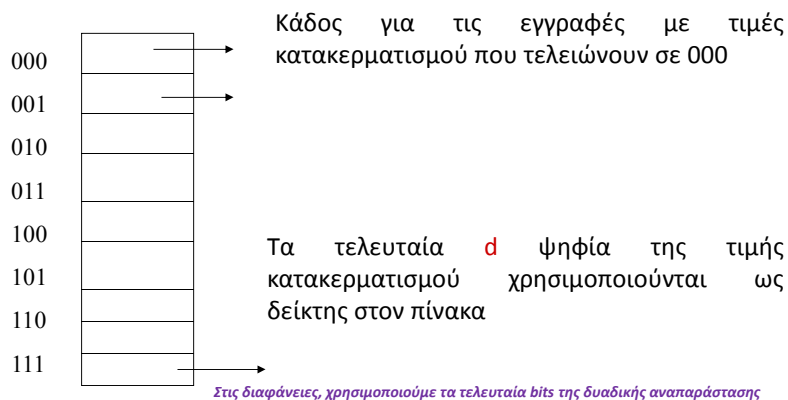
- Που αποθηκεύεται ο κατάλογος
 - στη μνήμη, εκτός αν είναι πολύ μεγάλος
 - τότε στο δίσκο – οπότε θα απαιτούνται επιπρόσθετες προσπελάσεις
- Δυναμική επέκταση αλλά μέγιστος αριθμός επιπέδων (το πλήθος των δυαδικών ψηφίων της συνάρτησης κατακερματισμού)
- Ισοζύγηση
- Συνένωση κάδων (δυναμική συρρίκνωση)

Επεκτατός Εξωτερικός Κατακερματισμός



Extendible hashing

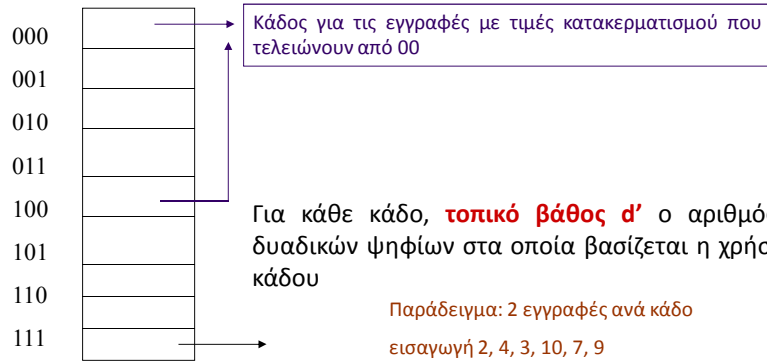
Ο κατάλογος είναι ένας πίνακας με 2^d διευθύνσεις κάδων (**d**: ολικό βάθος του καταλόγου)



Επεκτατός Εξωτερικός 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



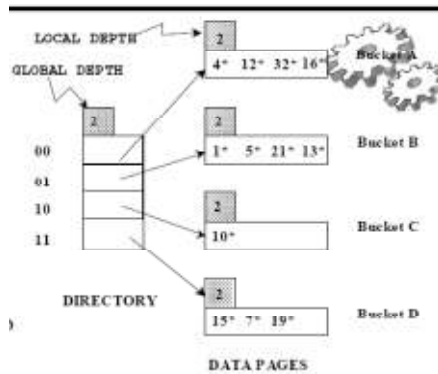
Επεκτατός Εξωτερικός Κατακερματισμός (Παράδειγμα)



Χρήση των τελευταίων bits της δυαδικής αναπαράστασης

| | | |
|----|--------|---------------------|
| 1 | 000001 | 4 εγγραφές ανά κάδο |
| 4 | 000100 | |
| 5 | 000101 | |
| 7 | 000111 | |
| 10 | 001010 | |
| 12 | 001100 | |
| 15 | 001111 | |
| 16 | 010000 | |
| 19 | 010011 | |
| 21 | 010101 | |
| 32 | 100000 | |
| 13 | 001101 | |

Επεκτατός Εξωτερικός Κατακερματισμός (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 |

Επεκτατός Εξωτερικός Κατακερματισμός



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

▪ Αύξηση της τιμής του d

Όταν ένας κάδος με τιμή $d' = d$ υπερχειλίζει

Διπλασιασμός του πίνακα

Δε χρειάζεται rehash (επανα-κερματισμό), διασπάμε κάθε κάδο

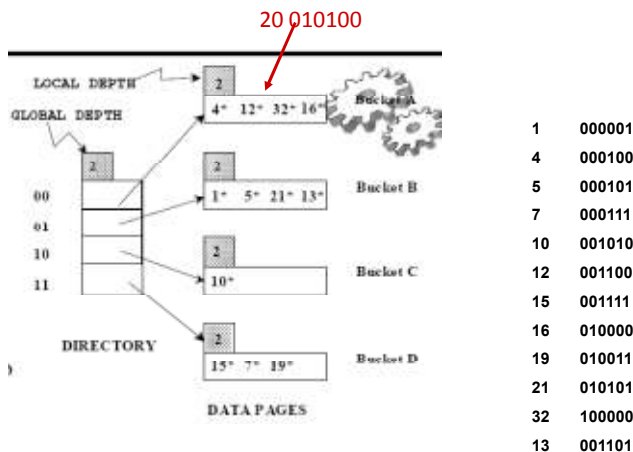
▪ Μείωση της τιμής του d

Όταν για όλους τους κάδους $d' < d$

Μείωση του μεγέθους του πίνακα στο μισό

Επίσης, κάθε φορά μόνο τον κάδο που υπερχειλίζει

Επεκτατός Εξωτερικός Κατακερματισμός (Παράδειγμα)



Βάσεις Δεδομένων 2012-2013

Ευαγγελία Πιπουρά

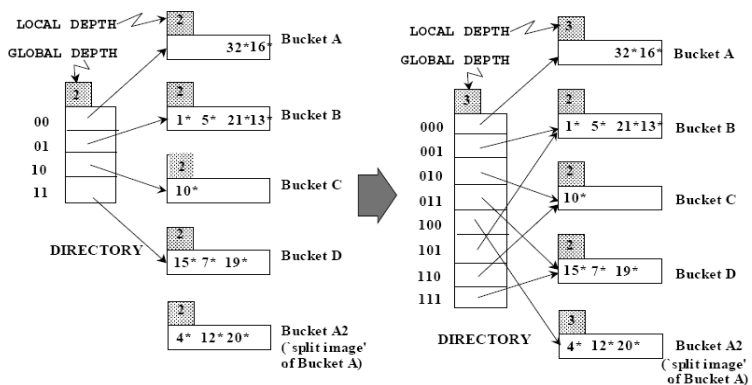
21

Επεκτατός Εξωτερικός Κατακερματισμός



- 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



Βάσεις Δεδομένων 2012-2013

Ευαγγελία Πιπουρά

22



Επανάληψη: 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)
 - Τιμή του πεδίου κατακερματισμού + δείκτη στο υπόλοιπο της εγγραφής; (ως **ευρετήριο**)
- Τι γίνεται αν το πεδίο κατακερματισμού δεν είναι κλειδί (παραπάνω από μια εγγραφή με την ίδια τιμή)



Γραμμικός Κατακερματισμός

Θέλουμε να αποφύγουμε τη χρήση καταλόγου +
Διπλασιασμό μεγέθους του καταλόγου

Προσοχή! Αυτή η μέθοδος:

- ✓ Διατηρεί λίστες υπερχείλισης
- ✓ Δε χρησιμοποιεί τη δυαδική αναπαράσταση

Γραμμικός Εξωτερικός Κατακερματισμός



Χρησιμοποιεί μια **οικογένεια από συναρτήσεις κατακερματισμού**

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

Κάθε συνάρτηση *διπλασιάζει* κάδους από την προηγούμενη:

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

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

Όταν συμβαίνει η πρώτη υπερχείλιση ενός κάδου, πάμε στην επόμενη συνάρτηση μέχρι να διασπαστούν όλοι οι κάδοι με αυτήν τη συνάρτηση

ΠΡΟΣΟΧΗ: δε διασπάμε τον κάδο που υπερχειλίζει, αλλά έναν-έναν τον κάδο με τη σειρά!

Γραμμικός Εξωτερικός Κατακερματισμός



Βασικά σημεία

- Πολλές συναρτήσεις κατακερματισμού (άλλη σε κάθε βήμα)
- Οι κάδοι σε κάθε βήμα διασπώνται με τη σειρά (ο ένας μετά τον άλλο – ανεξάρτητα αν έχουν ή όχι υπερχείλιση)
- *Επίσης, υποθέτουμε ότι κάθε υπερχείλιση, οδηγεί σε διάσπαση*



Αρκούν 2 μεταβλητές

- ✓ **Βήμα Διάσπασης** - ποια συνάρτηση χρησιμοποιούμε
- ✓ **Πλήθος Διασπάσεων** (στο τρέχον βήμα) – ποιος είναι ο επόμενος κάδο που θα διασπαστεί



Αρχικά:

Βήμα Διάσπασης (ποια συνάρτηση χρησιμοποιούμε) αρχικά $j = 0$:

Πλήθος Διασπάσεων (στο τρέχον βήμα) αρχικά $n = 0$,

j -> ποια συνάρτηση χρησιμοποιούμε

n -> ποιο κάδο διασπάμε

Έστω αρχικά M κάρτες αριθμημένους από 0 έως $M - 1$ και αρχική συνάρτηση κατακερματισμού

$$h_0(k) = k \bmod M$$



Γραμμικός Εξωτερικός Κατακερματισμός

j -> ποια συνάρτηση
χρησιμοποιούμε
n -> ποιο κάδο διασπάμε

Όταν συμβεί μια υπερχείλιση σε έναν οποιοδήποτε κάδο, **ο κάδος 0** χωρίζεται σε δύο κάδους: τον αρχικό κάδο 0 και ένα νέο κάδο M στο τέλος του αρχείου με βάση την συνάρτηση $h_1(k) = k \bmod 2M$

Βήμα Διάσπασης (ποια συνάρτηση χρησιμοποιούμε) $j = 1$

Πλήθος Διασπάσεων $n = 1$

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



Γραμμικός Εξωτερικός Κατακερματισμός

Βήμα διάσπασης (ποια συνάρτηση χρησιμοποιούμε) $j = 1$:

Πλήθος Διασπάσεων $n = m - 1$:

Όταν συμβεί μια υπερχείλιση σε έναν οποιοδήποτε κάδο,

ο κάδος $m - 1$ χωρίζεται σε δύο κάδους: τον αρχικό κάδο $m - 1$ και ένα νέο κάδο $m + k - 1$ στο τέλος του αρχείου με βάση την συνάρτηση $h_1(k) = k \bmod 2M$

Δηλαδή, σε κάθε υπερχείλιση χωρίζουμε όλους τους κάδους με τη σειρά ξεκινώντας από τον πρώτο κάδο

Γραμμικός Εξωτερικός 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 $0, 1, \dots$ κ.τ.λ

Γραμμικός Εξωτερικός Κατακερματισμός



Γενικά βήμα διάσπασης j ($j = 0, 1, 2, \dots$)

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

Γραμμικός Εξωτερικός Κατακερματισμός



32
9
44
31
25
5
35
7
36
14
18
10
11
30

Κάθε κάδος 4 εγγραφές

Αρχικά 4 κάδους ($M = 4$)

ΠΡΟΣΟΧΗ: Δε χρησιμοποιούμε τη δυαδική αναπαράσταση

Γραμμικός Εξωτερικός Κατακερματισμός (παράδειγμα)

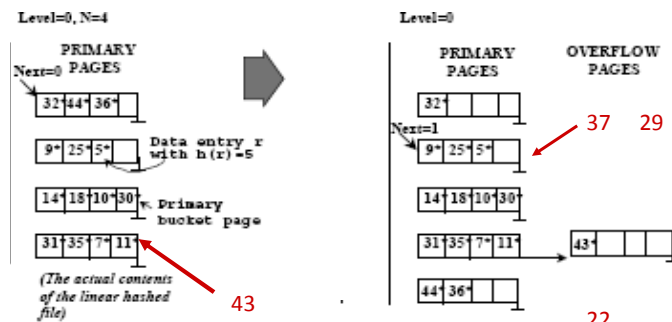


$$h_0(k) = k \bmod 4$$

Για μη διασπασμένους κάδους: παλιά συνάρτηση

$$h_1(k) = k \bmod 8$$

Για διασπασμένους κάδους: νέα συνάρτηση



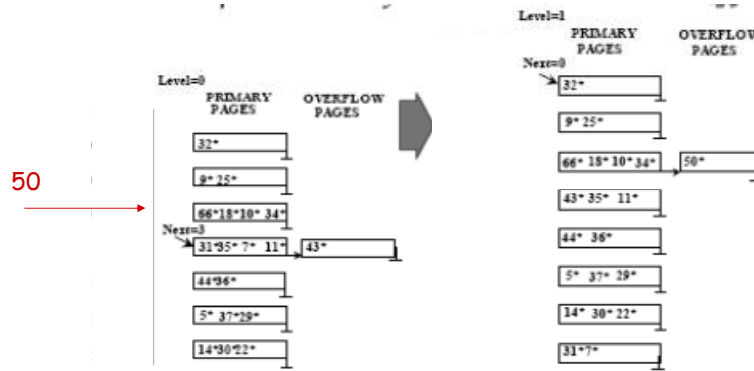
Διασπάμε τον πρώτο κάδο

Βήμα διάσπασης 0 (χρήση h_0)

Πλήθος διασπάσεων = 0

22
66
34

Γραμμικός Εξωτερικός Κατακερματισμός (παράδειγμα)



Βήμα διάσπασης 0 (χρήση h_0)
 Πλήθος διασπάσεων = 0

Γραμμικός Εξωτερικός Κατακερματισμός

Αναζήτηση Εγγραφής (γενικά)

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



Αναζήτηση Εγγραφής

Κρατάμε μια μεταβλητή το πλήθος n των διασπάσεων

Έστω n ο αριθμός διασπάσεων και ότι αναζητούμε το k ,

βρίσκεται στον κάδο $h_0(k)$

τότε αν $n \leq h_0(k)$ ο κάδος δεν έχει διασπαστεί

ενώ αν $n > h_0(k)$ ο κάδος έχει διασπαστεί και εφαρμόζουμε την $h_1(k)$



Αλγόριθμος Αναζήτησης

j : βήμα διάσπασης n : πλήθος διασπάσεων στο βήμα j

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

σημαίνει ότι ο κάδος έχει διασπαστεί