
Προηγμένα Θέματα Βάσεων Δεδομένων

Διδάσκων: Άγγελος Μιχάλας

Φυσική Οργάνωση

- Βασικά Τμήματα του ΣΔΒΔ
- Τύποι Εγγραφών
- **Τύποι Σελίδων**
- **Ομαδοποίηση Εγγραφών**
- **Διαχείριση Χώρου Δίσκου**
- **Διαχείριση Απομονωτικής Μνήμης**

Τύποι Σελίδων

Οργάνωση Σελίδων

- Κάθε σελίδα έχει **επικεφαλίδα** (page header), που περιέχει σημαντικές πληροφορίες
 - Π.χ. αριθμός αποθηκευμένων εγγραφών στη σελίδα, διεύθυνση επόμενης σελίδας του αρχείου
- Μία σελίδα έχει ένα **σύνολο θέσεων (slots)** για την αποθήκευση των εγγραφών.
- Κάθε εγγραφή χαρακτηρίζεται από το σύστημα με ένα μοναδικό **κωδικό εγγραφής** (record identifier - rid) που αποτελείται από το:
ζεύγος <αναγνωριστικό σελίδας, αριθμός θέσης>

Page Format: Fixed Length Records

1^η λύση:

- Οι εγγραφές αποθηκεύονται στις πρώτες διαθέσιμες θέσεις της σελίδας.
- Σε περίπτωση διαγραφής η τελευταία εγγραφή καταλαμβάνει τη θέση της διαγραφείσας.

n	
	εγγραφή 1
	εγγραφή 2
	...
	εγγραφή n
	ελεύθερος χώρος

- **Μειονέκτημα:** αν υπάρχουν αναφορές εγγραφών, η τελευταία εγγραφή δεν μπορεί να μετακινηθεί διότι οι αναφορές της θα καταστούν λανθασμένες (π.χ. εγγραφή εκτός σελίδας δείχνει προς τη θέση της τελευταίας εγγραφής)

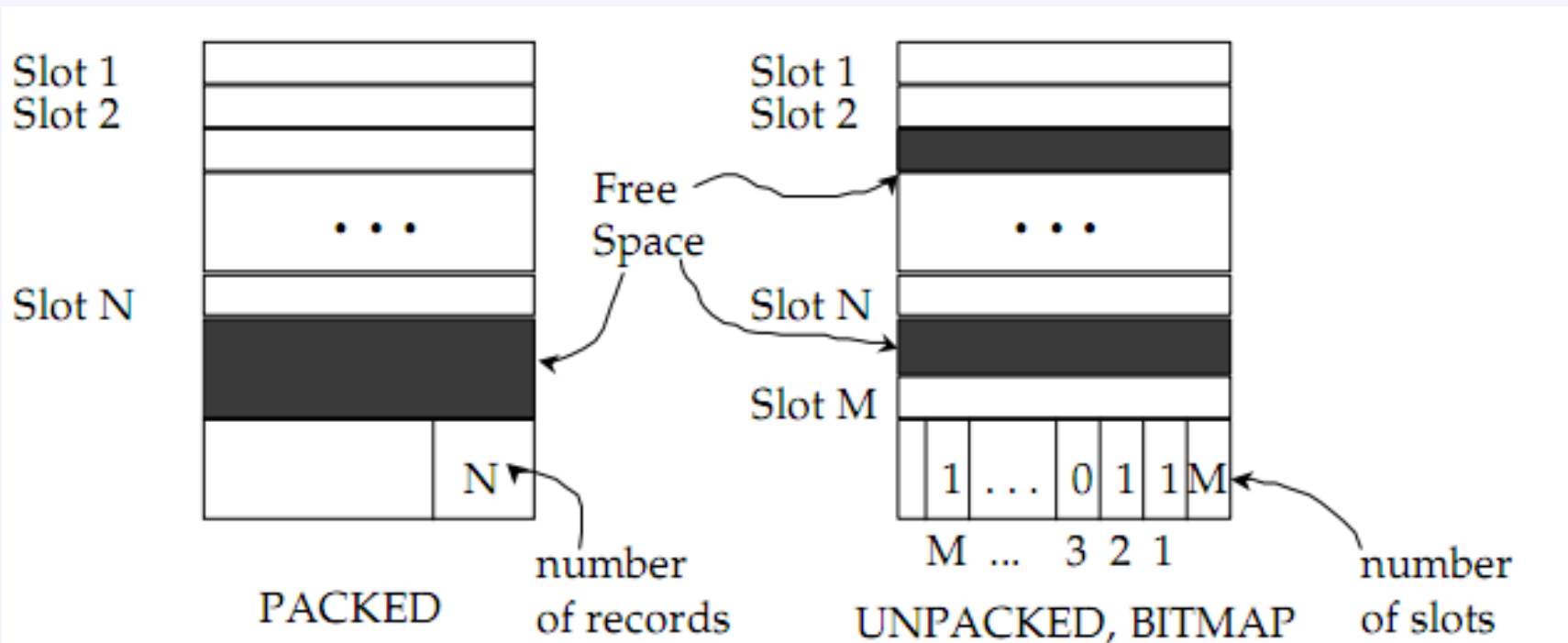
Page Format: Fixed Length Records

2^η λύση:

- Αποθηκεύεται στην επικεφαλίδα ένας δυαδικός πίνακας με **σημαίες (flags)** που δηλώνουν αν μία θέση είναι κατειλημμένη ή όχι.
- Κατά την αναζήτηση εξετάζονται μόνο οι θέσεις με αληθείς σημαίες
- Κατά τη διαγραφή η αντίστοιχη σημαία καθίσταται ψευδής.

m	101...01
	εγγραφή 1
	εγγραφή 3
	...
	εγγραφή m

Page Format: Fixed Length Records



- * Record id = <page id, slot #>. In first alternative, moving records for free space management changes rid; may not be acceptable.

Page Format: Variable Length Records

- Αν οι εγγραφές είναι μεταβλητού μήκους τότε οι **σελίδες δεν περιέχουν σταθερό αριθμό θέσεων**
- Εισαγωγή
 - Χρήση “κατάλληλου” κενού χώρου, ώστε να μην αχρηστευθεί σημαντικό τμήμα
- Διαγραφή
 - Μετακίνηση εγγραφών για την ενοποίηση κενού χώρου
- Αναζήτηση

Page Format: Variable Length Records

- **Κατάλογος θέσεων** (directory of slots), ή **πίνακας τοποθέτησης** (position table) στην επικεφαλίδα της σελίδας που περιέχει:
 - για κάθε εγγραφή το ζεύγος
<απόσταση εγγραφής (record offset), μήκος εγγραφής>
- Περιέχει δύο δείκτες
 - Ένας προς την αρχή της σελίδας από όπου αρχίζει η περιοχή αποθήκευσης των εγγραφών.
 - Ένας προς την αρχή της ενοποιημένης περιοχής που είναι ελεύθερη και διαθέσιμη για αποθήκευση νέων εγγραφών.

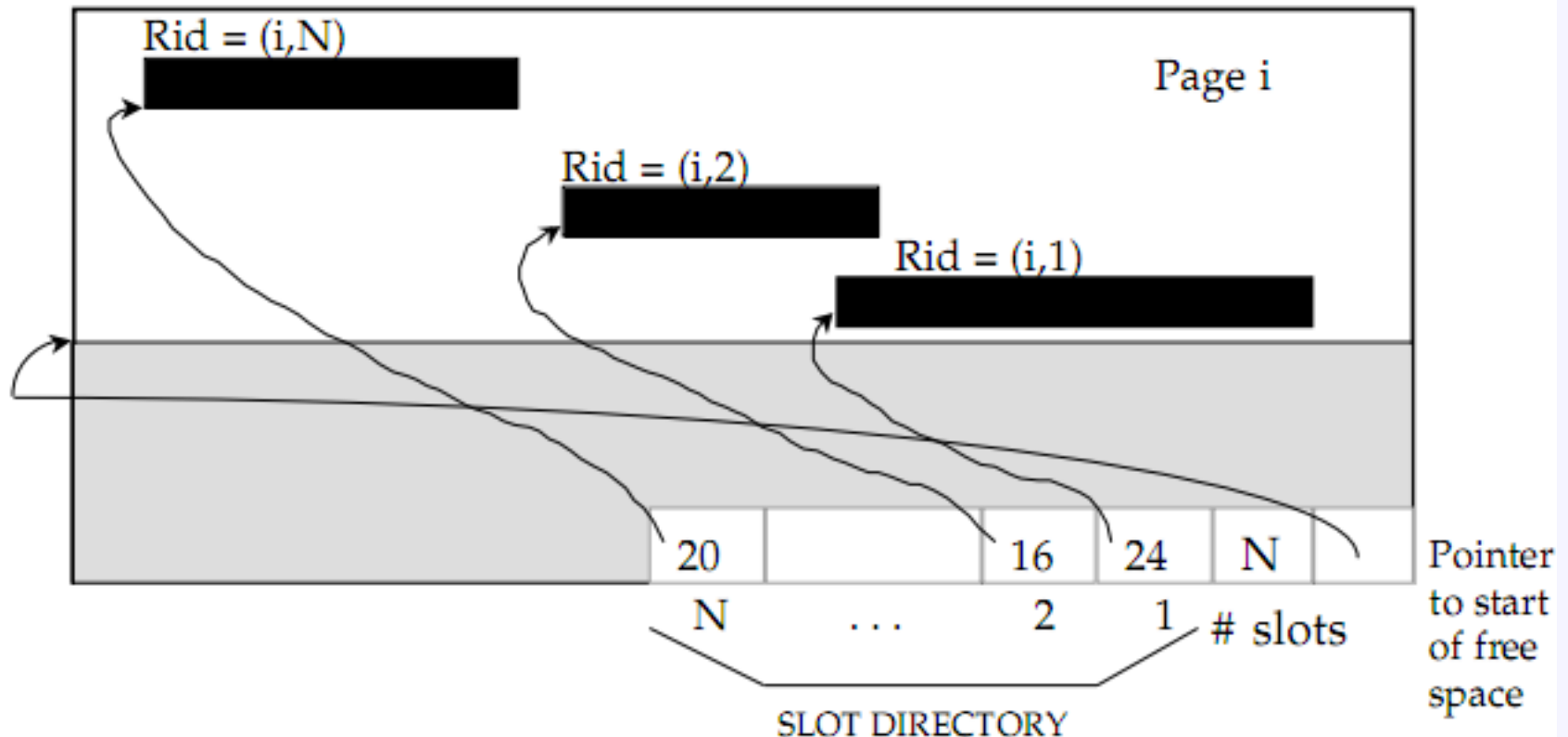
Page Format: Variable Length Records

- Σε περίπτωση διαγραφής δεν μπορεί να γίνει τακτοποίηση του ίδιου του καταλόγου.
 - Η θέση αυτή μπορεί να αποδοθεί σε επόμενη εισαγωγή εγγραφής

Παραλλαγή:

- Ο κατάλογος δεν περιέχει ζεύγη <offset, μήκος εγγραφής> αλλά μόνο το <offset> , ενώ η **ένδειξη μήκους** (length indicator) τοποθετείται στην αρχή της αντίστοιχης εγγραφής.
 - Σε περίπτωση αναζήτησης διευκολύνεται η υπέρβαση μιας εγγραφής

Page Format: Variable Length Records



- * *Can move records on page without changing rid; so, attractive for fixed-length records too.*

Ομαδοποίηση Εγγραφών

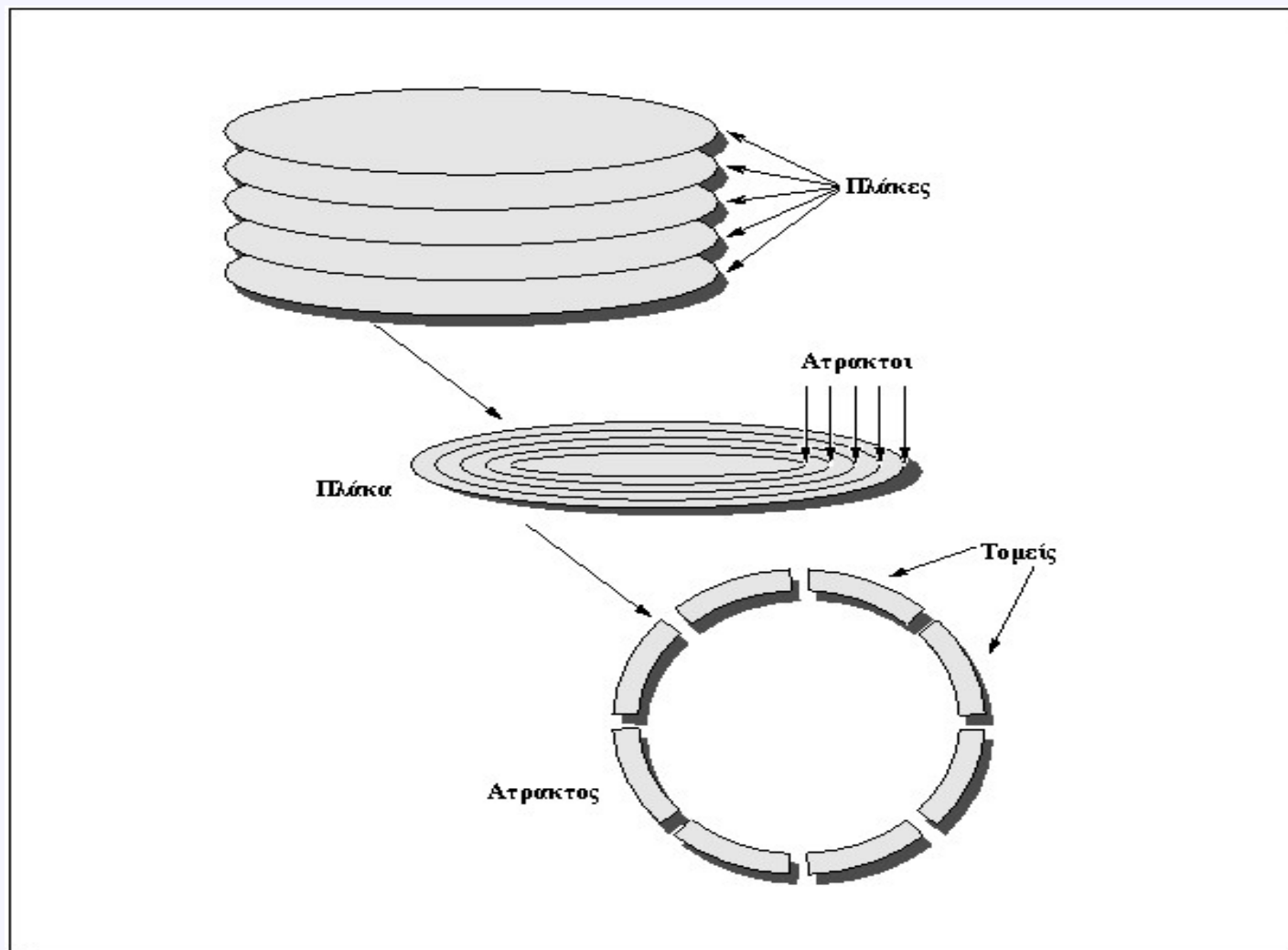
Μαγνητικοί Δίσκοι (1/7)

- Το 1956 η IBM κατασκεύασε τον πρώτο δίσκο RAMAC (Random Access Method for Accounting and Control - 5 MB).
- Ο RAMAC ήταν αρκετά μεγαλύτερος σε μέγεθος σε σχέση με τους σύγχρονους δίσκους, διότι η σύγχρονη τεχνολογία επιτρέπει την αύξηση του αριθμού των bits που μπορούν να κωδικοποιηθούν ανά μονάδα μαγνητικής επιφάνειας.
- Οι σημερινοί δίσκοι είναι ταχύτεροι, μικρότεροι και αποθηκεύουν περισσότερα δεδομένα.

Μαγνητικοί Δίσκοι (2/7)

- Υπάρχουν δύο κύριοι τύποι μαγνητικών δίσκων: οι δισκέτες και οι σκληροί δίσκοι.
- Οι δύο τύποι δίσκων αποτελούνται από μία περιστρεφόμενη πλάκα (platter) επιστρωμένη με μία μαγνητική επιφάνεια και χρησιμοποιούν μία κινητή κεφαλή ανάγνωσης/εγγραφής για να προσπελάσουν το δίσκο.
- Η αποθήκευση στο δίσκο είναι μόνιμη, πράγμα που σημαίνει πως τα δεδομένα παραμένουν ακόμα και όταν διακοπεί η τροφοδοσία ηλεκτρικής ισχύος.

Μαγνητικοί Δίσκοι (3/7)



Μαγνητικοί Δίσκοι (4/7)

- Ένας μαγνητικός δίσκος αποτελείται από μία συλλογή από **πλάκες** (2 με 20), καθεμιά από τις οποίες έχει δύο εγγράψιμες επιφάνειες.
- Η **στοίβα των πλακών περιστρέφεται με 3600 ως 7200 στροφές ανά λεπτό (RPM)** και έχει διάμετρο που κυμαίνεται από κάτι παραπάνω από μία ίντσα έως κάτι παραπάνω από 10 ίντσες.
- Κάθε επιφάνεια του δίσκου διαιρείται σε ομόκεντρους κύκλους που ονομάζονται **άτρακτοι (tracks)**. Τυπικά, υπάρχουν **500 με 2000 άτρακτοι** ανά επιφάνεια.

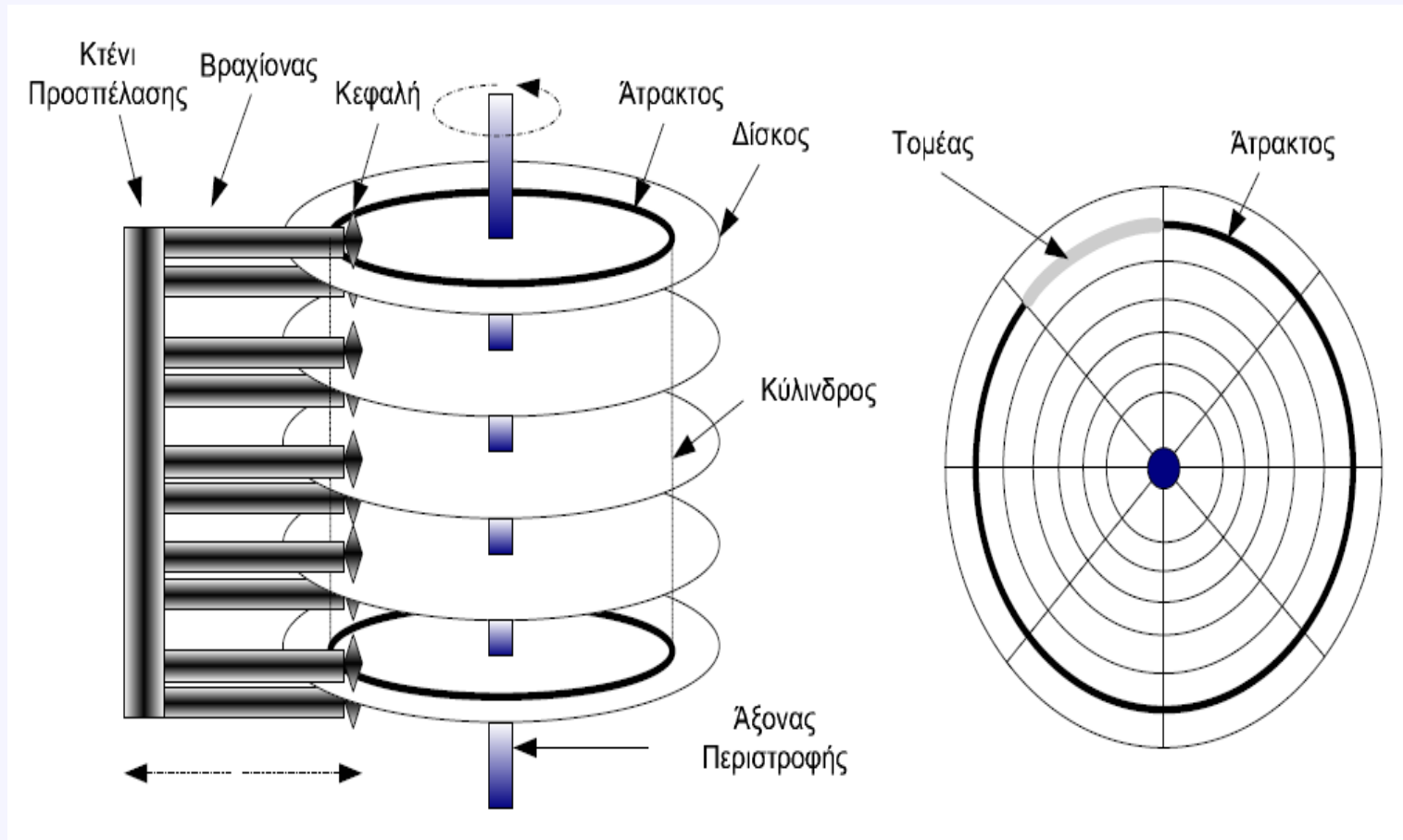
Μαγνητικοί Δίσκοι (5/7)

- Κάθε άτρακτος διαιρείται με τη σειρά της σε **τομείς (sectors)** ή **page** ή **block** αλλιώς που περιέχουν τις πληροφορίες. Κάθε άτρακτος μπορεί να έχει **32 με 128 τομείς**
 - ο τομέας είναι η μικρότερη μονάδα που μπορεί να διαβαστεί ή να γραφτεί.
- *Παραδοσιακά, όλες οι άτρακτοι έχουν το ίδιο πλήθος από τομείς και συνεπώς το ίδιο πλήθος bits.*
- Για να διαβάσουμε και να γράψουμε πληροφορίες, οι κεφαλές ανάγνωσης/εγγραφής πρέπει να μετακινηθούν, ώστε να βρεθούν πάνω από τη σωστή θέση.

Μαγνητικοί Δίσκοι (6/7)

- Οι βραχίονες του δίσκου (disk arms) για κάθε επιφάνεια είναι συνδεδεμένοι μεταξύ τους και μετακινούνται μαζί, έτσι ώστε *κάθε βραχίονας να είναι πάνω από την ίδια άτρακτο σε κάθε επιφάνεια.*
- Ο όρος **κύλινδρος (cylinder)** χρησιμοποιείται για να αναφερόμαστε σε όλες τις άτρακτους κάτω από τους βραχίονες σε ένα δεδομένο σημείο σε όλες τις επιφάνειες.

Μαγνητικοί Δίσκοι (7/7)



Ομαδοποίηση Εγγραφών

- Υποθέτουμε ότι
 1. μονάδα αποθήκευσης είναι ο σκληρός δίσκος,
 2. η εγγραφή είναι η ελάχιστη λογική μονάδα διακίνησης πληροφορίας (για τον άνθρωπο),
 3. η σελίδα είναι το ελάχιστο φυσικό ποσό δεδομένων (για το σύστημα) που μεταφέρεται μεταξύ του δίσκου και της κύριας μνήμης.

Ομαδοποίηση Εγγραφών

- **Seek time:** Ο χρόνος που απαιτείται για την μηχανική τοποθέτηση της κεφαλής ανάγνωσης / εγγραφής στην σωστή άτρακτο
 - Ο χρόνος συνήθως ποικίλλει μεταξύ 10 και 60 msec

Ομαδοποίηση Εγγραφών

- **Χρόνος περιστροφής (rotational delay):** Αναμονή μέχρι να περιστραφεί το block και να βρεθεί κάτω από την κεφαλή ανάγνωσης/εγγραφής
$$rd = (1/2) * (1/p) \text{min} = (60 * 1000) / (2 * p) \text{msec}$$
 - p είναι το πλήθος των περιστροφών που πραγματοποιούνται σε ένα λεπτό.
 - Για p=3600rpm \rightarrow rd=8.33msec
- Εάν έχουμε μια άτρακτο 50KBytes και p 3600rpm τότε ο **ρυθμός μεταφοράς** είναι:
$$tr = (50 * 1000) / (60 * 1000 / 3600) = 3000 \text{ bytes/msec}$$

Ομαδοποίηση Εγγραφών

Παράγοντας ομαδοποίησης (blocking factor - Bfr) είναι το πλήθος των εγγραφών που χωρούν σε μία σελίδα.

Αν υποθεθεί ότι το μέγεθος επικεφαλίδας της σελίδας είναι αμελητέο και ότι οι **εγγραφές** είναι **σταθερού μήκους**, τότε ο παράγοντας ομαδοποίησης ισούται με:

$$Bfr = \left\lfloor \frac{B}{R} \right\rfloor$$

όπου:

- B είναι το μέγεθος της σελίδας,
- R είναι το μέγεθος της εγγραφής

Ομαδοποίηση Εγγραφών

- Αν οι εγγραφές είναι μεταβλητού μήκους, τότε η μεταβλητή R συμβολίζει το μέσο μήκος των εγγραφών.
- Προσεγγιστικά, ο μέσος κενός χώρος ισούται με το μισό της μέσης εγγραφής, οπότε ο παράγοντας ομαδοποίησης γίνεται:

$$Bfr = \frac{B - R/2}{R}$$

Ομαδοποίηση Εγγραφών

- Αν n είναι το πλήθος των εγγραφών, τότε ο αριθμός των σελίδων του αρχείου είναι:
- Αν οι κεφαλές είναι τοποθετημένες στην αρχή της σελίδας, τότε ο χρόνος μεταφοράς της σελίδας ισούται με:
όπου t είναι ταχύτητα διακίνησης των δεδομένων μεταξύ δίσκου και μνήμης και μετράται με bytes/ms.
- Αν (αντί σελίδας) θεωρηθεί μία άτρακτος T , τότε ο χρόνος μεταφοράς ισούται με:
όπου r είναι ο μέσος χρόνος περιστροφής.

$$b = \left\lceil \frac{n}{Bfr} \right\rceil$$

$$btt = \frac{B}{t}$$

$$btt = \frac{T}{2r}$$

Οργάνωση Δίσκου

- Στο σκληρό δίσκο **απαιτείται** κάποια **οργάνωση** με σκοπό το συγχρονισμό κατά τη λειτουργία του και επομένως τη **μείωση του λανθάνοντα περιστροφικού χρόνου**.
- Στην πράξη χρησιμοποιούνται δύο τρόποι για την οργάνωση ενός δίσκου:
 - η ύπαρξη των κενών μεταξύ των σελίδων, και
 - η χρήση της τεχνικής της **παρεμβολής** (interleaving) που επιτυγχάνει καλύτερους χρόνους προσπέλασης αλλά είναι σχετικά περίπλοκη.

Τεχνική της Παρεμβολής

- Η τεχνική της παρεμβολής είναι αναγκαία
- *Ο ελεγκτής του δίσκου απαιτεί κάποιο χρόνο επεξεργασίας των δεδομένων που δέχεται από το δίσκο πριν μπορέσει να δεχθεί άλλα δεδομένα*
- Αν η επόμενη σελίδα τοποθετούνταν στον επόμενο τομέα, τότε θα απαιτούνταν μία πλήρης περιστροφή του δίσκου.

Τεχνική της Παρεμβολής

- Παράδειγμα: Αποθήκευση 10 σελίδων ενός αρχείου στους 10 τομείς μίας ατράκτου.
- Έστω ότι ο ελεγκτής απαιτεί διπλάσιο χρόνο από το χρόνο που απαιτείται για την προσπέλαση του επόμενου τομέα.

Φυσικός τομέας	1	2	3	4	5	6	7	8	9	10
Λογικός τομέας	1	8	5	2	9	6	3	10	7	4

Μέθοδος κενών

Θεωρούμε μια καινούργια μεταβλητή που ονομάζεται **πραγματικός χρόνος μεταφοράς σελίδας** (effective block transfer time) και ισούται με:

$$ebt = \frac{B}{t'}$$

όπου t' είναι η ταχύτητα διακίνησης δεδομένων από φορμαρισμένο δίσκο με κενά στη μνήμη και μετράται με bytes/ms.

Παράμετροι του IBM 3380

Ορισμός - Παράμετρος	Τιμή
Μέγεθος σελίδας,	2400 bytes
Χρόνος μεταφοράς σελίδας, btt	0,8 ms
Σελίδες ανά κύλινδρο, C	600
Πραγματικός χρόνος μεταφοράς σελίδας, ebt	0,84 ms
Αριθμός κυλίνδρων, N	885
Μέσος χρόνος περιστροφής, r	8,3 ms
Μέσος χρόνος αναζήτησης, s	16 ms
Ταχύτητα ανάγνωσης σε δίσκο, t	3000 bytes/ms
Ταχύτητα ανάγνωσης σε φορμαρισμένο δίσκο, t'	2857 bytes/ms

Ταχύτητα Ανάγνωσης

- Ο μέσος χρόνος ανεύρεσης και μεταφοράς μιας σελίδας είναι:
 - $(s + r + ebt)$ msec
 - s μέσος χρόνος αναζήτησης σε msec
 - r μέσος χρόνος περιστροφής σε msec
- Ο χρόνος μεταφοράς k μη συνεχόμενων σελίδων που βρίσκονται στον ίδιο κύλινδρο είναι:
 - $s + (k * (r + ebt))$
- Για συνεχόμενες σελίδες ισχύει:
 - $s + r + k * ebt$

Διαχείριση Χώρου Δίσκου

Διαχείριση Χώρου Δίσκου

- Ο διαχειριστής του χώρου του δίσκου είναι το πλησιέστερο τμήμα ενός ΣΔΒΔ προς το υλικό.
- Ο διαχειριστής δίνει εντολές προς το δίσκο για να
 - δεσμεύσει ή να αποδεσμεύσει μία σελίδα
 - αναγνωσθεί ή να αποθηκευθεί μια σελίδα
- Συχνά είναι χρήσιμο **το σύνολο σελίδων ενός αρχείου να αποθηκεύονται σε διαδοχικές ατράκτους του δίσκου.**
 - Αποτελεσματική σειριακή ανάκτηση.

Διαχείριση Χώρου Δίσκου

- Το πλήθος των προσπελάσιμων σελίδων περιορίζεται από το μέγεθος της απομονωτικής μνήμης,
 - Γνωστό ως **κάδος** (bucket) που είναι η λογική μονάδα προσπέλασης στο δίσκο.
- Το μέγεθος του κάδου μπορεί να είναι 1, 2, 4, 8 ή και παραπάνω σελίδες

Διαχείριση Χώρου Δίσκου

- Η επιλογή κατάλληλου μεγέθους κάδου συνεπάγεται τη βελτιστοποίηση της χρήσης του χώρου του δίσκου.
- Παράδειγμα: αποθήκευση 2 εγγραφών μεγέθους 160 bytes σε σελίδες μεγέθους 256 bytes. Οι επιλογές είναι:
 - Η δεύτερη εγγραφή «σπάει» σε δύο σελίδες (αυξημένο κόστος προσπέλασης),
 - Κάθε εγγραφή αποθηκεύεται σε μια σελίδα (αυξημένο κόστος σε χώρο λόγω ύπαρξης κενού χώρου).
- Το φαινόμενο της ύπαρξης κενού χώρου σε κάθε σελίδα ονομάζεται **εσωτερική τμηματοποίηση**.

Διαχείριση Χώρου Δίσκου

Παράγοντας ομαδοποίησης	Μέγεθος εγγραφών	Σελίδες/ κάδο	Μέγεθος κάδου	Ποσοστό χρήσης
1	160	1	256	63%
2	320	2	512	63%
3	480	2	512	94%
4	640	3	768	83%
5	800	4	1024	78%
6	960	4	1024	94%
7	1120	5	1280	88%
8	1280	5	1280	100%

Διαχείριση Χώρου Δίσκου

- Το ποσοστό χρησιμοποίησης του χώρου δίδεται από:

$$U = \frac{R \times Bfr}{S \times Ppb}$$

όπου Ppb είναι ο αριθμός των σελίδων ανά κάδο

$$Ppb = \left\lfloor \frac{R \times Bfr}{S} \right\rfloor$$

και S είναι το μέγεθος του τομέα σε bytes.

Διαχείριση Χώρου Δίσκου

- **Χρόνος αναζήτησης (seek)**, T_{seek} : μετακίνηση των κεφαλών επάνω από την κατάλληλη άτρακτο.
- **Χρόνος περιστροφικής καθυστέρησης (rotational latency)**, T_{rot} : αναμονή μέχρι ο κατάλληλος τομέας να περάσει κάτω από την κεφαλή.
- **Χρόνος Μεταφοράς (transfer) δεδομένων**, T_{trans} : η κεφαλή διαβάζει ή γράφει δεδομένα.
- Για το **χρόνο προσπέλασης (access)** ισχύει:

$$T_{total} = T_{seek} + T_{rot} + T_{trans}$$

Διαχείριση Χώρου Δίσκου

- Ο διαχειριστής πρέπει να γνωρίζει ποιες σελίδες είναι κατειλημμένες και ποιες ελεύθερες.
- Δύο τεχνικές:
 1. με χρήση μία λίστας για τις διευθύνσεις των ελεύθερων σελίδων,
 2. με χρήση ενός δυαδικού πίνακα, όπου τα 1 και 0 αντιστοιχούν στις κατειλημμένες και στις ελεύθερες σελίδες, αντίστοιχα.

Διαχείριση Χώρου Δίσκου

- Η **τμηματοποίηση** του δίσκου είναι η διαδικασία κατά την οποία υπάρχει συνολικά **ελεύθερος χώρος** όμως είναι **μη πρακτικά εκμεταλλεύσιμος**.
- Τα αρχεία χωρίζονται σε **εκτάσεις** (extents) όπου κάθε έκταση καταλαμβάνει κατά αποκλειστικότητα ένα συγκεκριμένο μέρος του δίσκου
 - οι εκτάσεις είναι ισομεγέθεις και ποικίλουν σε πλήθος από μία μέχρι μερικές δεκάδες.

Διαχείριση Χώρου Δίσκου

Βασικό πλεονέκτημα χρήσης εκτάσεων:

- Αποτελεσματικότερη διαχείριση από άποψη παραλληλισμού.
 - Τα μεγάλα αρχεία επιμερίζονται σε εκτάσεις που αποθηκεύονται σε διαφορετικούς δίσκους. Η κατανομή αυτή λέγεται **ζεύξη** ή **σύνδεση**.

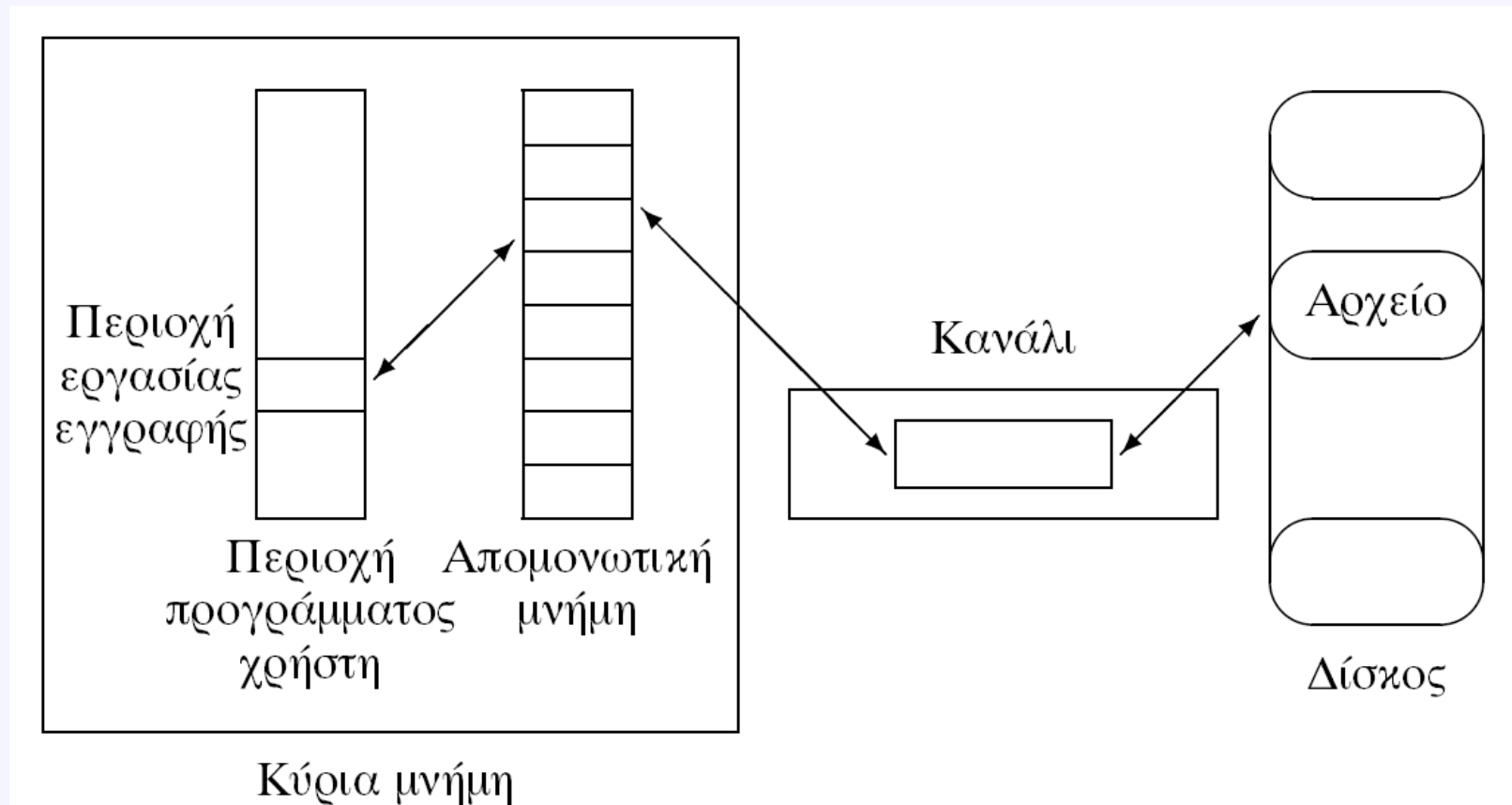
Διαχείριση Απομονωτικής Μνήμης

Διαχείριση Απομονωτικής Μνήμης

Ο σκοπός των **απομονωτικών μνημών** (buffer memories) είναι η ομαλοποίηση της ροής των δεδομένων μεταξύ της γρήγορης κύριας μνήμης και της αργής δευτερεύουσας μνήμης, και γενικά διακρίνονται σε δύο τύπους:

- ο τις **υλικές** (hardware buffers), που χρησιμοποιούνται ευρύτατα για τη διαχείριση των ιδιαίτερα αργών περιφερειακών συσκευών, και
 - ο Π.χ. Στους εκτυπωτές η μνήμη αυτή βρίσκεται στον ελεγκτή τους
- ο τις **λογισμικές** (software buffers), που είναι τμήματα της κύριας μνήμης.

Διαχείριση Απομονωτικής Μνήμης



Διαχείριση Απομονωτικής Μνήμης

Ο διαχειριστής της απομονωτικής μνήμης είναι υπεύθυνος:

1. για τον τεμαχισμό της μνήμης σε ένα σύνολο **πλαισίων (frames)**.

- Το μέγεθος κάθε πλαισίου είναι **ίσο** προς το μέγεθος της **σελίδας**
- Το σύνολο των πλαισίων ονομάζεται **δεξαμενή μνημών (buffer pool)**.

2. για την αποθήκευση των δεδομένων στα πλαίσια μέχρι να γίνει η επεξεργασία τους.

Διαχείριση Απομονωτικής Μνήμης

Επικοινωνία διαχειριστή απομονωτικής μνήμης και ανώτερων στρωμάτων του συστήματος για:

1. Ανάκτηση σελίδας από κάποιο πλαίσιο ή από το δίσκο.
2. Ολοκλήρωση επεξεργασίας σελίδας (μπορεί να απελευθερωθεί το πλαίσιο).
3. Ενημέρωση σελίδας οπότε η ενημέρωση πρέπει να περάσει στην αντίστοιχη σελίδα του δίσκου.

Διαχείριση Απομονωτικής Μνήμης

- Ο διαχειριστής γνωρίζει για **κάθε πλαίσιο πόσες φορές έχει ζητηθεί** χωρίς να ελευθερωθεί από τη στιγμή που τοποθετήθηκε η σελίδα στην απομονωτική μνήμη και αν **το πλαίσιο έχει ενημερωθεί** (βρώμικο).
- Πλήθος καρφωμάτων πλαισίου, δηλαδή πόσες φορές έχει ζητηθεί.
 - **Μετρητής καρφωμάτων** (pin_count).
- Αν το πλαίσιο είναι **βρώμικο** (dirty), δηλαδή αν έχει ενημερωθεί.
 - **Βρώμικο bit**.
- Αρχικά οι δύο μεταβλητές παίρνουν τιμή ίση με μηδέν $0_{10} 0_2$.

Διαχείριση Απομονωτικής Μνήμης

- Λειτουργίες: Το περιεχόμενο μιας σελίδας που βρίσκεται σε ένα πλαίσιο ζητείται (κάρφωμα)
 - **Αυξάνεται** το **pin_count** κατά **1**.
- Λειτουργίες: Το περιεχόμενο μιας σελίδας δεν είναι χρήσιμο (ξεκάρφωμα)
 - **Ελαττώνει** το **pin_count** κατά **1**.
- Λειτουργίες: Ενημέρωση σελίδας
 - Θέτει το **βρώμικο bit** ίσο με **1**.
 - Αλλάζει το περιεχόμενο του πλαισίου ώστε να περιέχει την ενημερωμένη σελίδα.

Διαχείριση Απομονωτικής Μνήμης

Λειτουργίες: Ανάκτηση σελίδας

1. Ελέγχει αν ο χώρος της απομονωτικής μνήμης περιέχει τη σελίδα.
2. Αν ναι, αυξάνει το `pin_count` κατά 1.
3. Αν όχι (και η απομονωτική μνήμη είναι γεμάτη),
 1. επιλέγει ένα πλαίσιο (από αυτά που έχουν `pin_count==0`) προς αντικατάσταση.
 2. αν το βρώμικο bit του προς αντικατάσταση πλαισίου είναι 1, γράφει πρώτα την σελίδα στο δίσκο (τη στέλνει στο διαχειριστή δίσκου) και ελευθερώνει το πλαίσιο,
 3. διαβάζει τη σελίδα και την τοποθετεί στο προς αντικατάσταση πλαίσιο.
4. Επιστρέφει τη διεύθυνση του πλαισίου που περιέχει τη σελίδα.

Διαχείριση Απομονωτικής Μνήμης

- Το πλαίσιο προς αντικατάσταση ονομάζεται **θύμα (victim)**.
- Το **θύμα** επιλέγεται μεταξύ των πλαισίων με μηδενικό μετρητή καρφωμάτων εφαρμόζοντας μία πολιτική **αντικατάστασης σελίδων** (page replacement).
- Όταν δεν υπάρχει σελίδα με μηδενικό μετρητή καρφωμάτων, η νέα αίτηση πρέπει να περιμένει και πιθανώς να απορριφθεί.

Διαχείριση Απομονωτικής Μνήμης

- Τα ΣΔΒΔ μπορεί να προβλέψει τα **πρότυπα ζήτησης σελίδων** ειδικά όταν εκτελούνται συγκεκριμένες διεργασίες (σειριακές σαρώσεις αρχείων, join αρχείων, κ.α.).
 - ➔ **Αποτελεσματικότερη διαχείριση**
 - με την επιλογή του κατάλληλου πλαισίου για το ρόλο του θύματος,
 - με την επιλογή της κατάλληλης στιγμής για την αποθήκευση των βρώμικων σελίδων στο δίσκο, και
 - με την **προ-προσπέλαση** (prefetching) σελίδων από το δίσκο πριν ακόμη αυτές ζητηθούν.

Διαχείριση Απομονωτικής Μνήμης

- Βασικό κόστος η μεταφορά δεδομένων από τη δευτερεύουσα στην κύρια μνήμη και όχι η επεξεργασία των σελίδων στις απομονωτικές μνήμες.
- Η **προ-προσπέλαση** βελτιώνει σημαντικά την επίδοση του συστήματος όταν
 - Οι σελίδες δε ζητούνται ισοπίθانا αλλά οι αιτήσεις εστιάζονται κυρίως σε μερικές περιοχές δεδομένων (**τοπικότητα** - locality) → αύξηση του **λόγου επιτυχίας**

Διαχείριση Απομονωτικής Μνήμης

- **Λόγος επιτυχίας** (hit ratio), ονομάζεται ο λόγος του πλήθους αιτήσεων που απαντώνται χωρίς προσπέλαση στη δευτερεύουσα μνήμη προς τον αριθμό των συνολικών αιτήσεων.
- Η προ-προσπέλαση βελτιώνει σημαντικά την επίδοση του συστήματος όταν οι αιτήσεις αφορούν γειτονικές σελίδες σε περίπτωση σειριακής προσπέλασης, ενώ η επίδοση φθίνει σε περίπτωση τυχαίας προσπέλασης.

Διαχείριση Απομονωτικής Μνήμης

- Προ-προσπέλαση, μέθοδος **single buffering**: κυκλική λίστα με δύο δείκτες
 - Μπορεί να υιοθετηθεί σε περιπτώσεις σειριακής προσπέλασης αρχείου
 - **Πρώτα γεμίζει η μνήμη και μετά ακολουθεί η επεξεργασία**
 - *Μεταφέρεται η επόμενη σελίδα από το δίσκο στην μνήμη μόνο όταν τελειώσει η επεξεργασία της σελίδας της απομονωτικής μνήμης*

Διαχείριση Απομονωτικής Μνήμης

- *single buffering – αποτελεσματικότερη χρήση της μνήμης μεταφέροντας δεδομένα ταυτόχρονα από τη περιφερειακή συσκευή προς τη μνήμη και από τη μνήμη στη περιοχή του προγράμματος του χρήστη.*
- Υλοποιείται με κυκλική λίστα με χρήση δύο δεικτών
 - ο **πρώτος** δείχνει την **πρώτη διαθέσιμη εγγραφή** για το πρόγραμμα εφαρμογής,
 - ο **δεύτερος** δείχνει την **πρώτη διαθέσιμη θέση** για αποθήκευση εγγραφών.

Διαχείριση Απομονωτικής Μνήμης

Προ-προσπέλαση, μέθοδος **double buffering**:

Παράδειγμα

- έστω ότι η απομονωτική μνήμη αποτελείται από 2 πλαίσια και έστω ότι αυτά γεμίζουν με τις πρώτες 2 σελίδες του αρχείου
- μόλις ολοκληρωθεί η επεξεργασία των εγγραφών του πρώτου πλαισίου, τότε αυτό ξαναγεμίζει με το περιεχόμενο της τρίτης σελίδας
- σε κάθε χρονική στιγμή το ένα πλαίσιο δέχεται δεδομένα, ενώ το άλλο συμμετέχει στην επεξεργασία
- Η τεχνική αλλαγής των ρόλων των δύο μνημών ονομάζεται **ανταλλαγή απομονωτικών μνημών (buffer swap)**.

Διαχείριση Απομονωτικής Μνήμης

Προ-προσπέλαση, μέθοδος **double buffering**:

- Αν ο **χρόνος επεξεργασίας** (χρόνος CPU) είναι **μικρότερος** από το **χρόνο προσπέλασης** (χρόνος I/O) τότε μπορεί να θεωρηθεί ότι:

συνολικός χρόνος = χρόνος προσπέλασης

Διαχείριση Απομονωτικής Μνήμης

Παράδειγμα double buffering

I/O	CPU	Πρώτο πλαίσιο	Δεύτερο πλαίσιο
Ανάγνωση A			
Ανάγνωση B	Επεξεργασία A	A	
Ανάγνωση C	Επεξεργασία B	A/C	B
Ανάγνωση D	Επεξεργασία C	C	B/D
⋮	⋮	⋮	⋮

Διαχείριση Απομονωτικής Μνήμης

- Όταν οι υπολογισμοί είναι σύνθετοι, τότε δεν έχουμε θεαματικά αποτελέσματα
- Παράδειγμα: έστω ότι ο χρόνος CPU είναι τριπλάσιος από χρόνο I/O.

I/O	CPU	Πρώτο πλαίσιο	Δεύτερο πλαίσιο
Ανάγνωση A			
Ανάγνωση B	Επεξεργασία A	A	
Αναμονή	Επεξεργασία A	A	B
Αναμονή	Επεξεργασία A	A	B
Ανάγνωση C	Επεξεργασία B	A/C	B
Αναμονή	Επεξεργασία B	C	B
Αναμονή	Επεξεργασία B	C	B
⋮	⋮	⋮	⋮

Αλγόριθμοι Αντικατάστασης Σελίδων

- Οι τεχνικές του single buffering και double buffering μπορούν να εφαρμοσθούν στην περίπτωση σειριακής προσπέλασης αρχείου.
- Όμως σε ένα ΣΔΒΔ μπορούν πολλοί χρήστες ταυτόχρονα να προσπελούν τη ΒΔ, οπότε **η απομονωτική μνήμη μπορεί να έχει σελίδες από πολλά αρχεία**
 - Απαιτείται μία πολιτική αντικατάστασης σελίδων

Αλγόριθμοι Αντικατάστασης Σελίδων

- **Λιγότερο πρόσφατα χρησιμοποιημένης** (least recently used, **LRU**)
 - Θύμα: η σελίδα που δεν έχει χρησιμοποιηθεί για το μεγαλύτερο χρονικό διάστημα.
- **Λιγότερο συχνά χρησιμοποιούμενη** (least frequently used - **LFU**).
 - Θύμα: η σελίδα που έχει τη μικρότερη συχνότητα χρησιμοποίησης.
 - Μειονέκτημα: Μια σελίδα που χρησιμοποιούνταν πολύ στο παρελθόν, θα αργήσει να αντικατασταθεί

Αλγόριθμοι Αντικατάστασης Σελίδων

- Άλλες πολιτικές αντικατάστασης σελίδων είναι:
 - **Τυχαία (random)**,
 - **Περισσότερο πρόσφατα χρησιμοποιημένη**
(most recently used - **MRU**),
 - **Πρώτη ερχόμενη πρώτη αντικαθιστώμενη**
(first-in first out - **FIFO**).

Αλγόριθμοι Αντικατάστασης Σελίδων

- Πιο γνωστή πολιτική: LRU
 - Πρόβλημα με σειριακή προσπέλαση
- Παράδειγμα:
 - έστω απομονωτική μνήμη 10 πλαισίων και προς σειριακή προσπέλαση αρχείο με 11 ή περισσότερες σελίδες,
 - με την LRU κάθε σειριακή προσπέλαση θα πρέπει να διαβάζει όλες τις σελίδες του αρχείου.
- Η περίπτωση αυτή ονομάζεται **σειριακή πλημμύρα** (sequential flood)

Αλγόριθμοι Αντικατάστασης Σελίδων

- **Σειριακή πλημμύρα** (sequential flood): Δυσάρεστη κατάσταση που προκαλείται από LRU + repeated sequential scans
 - # buffers < # pages του αρχείου → Κάθε σελίδα που ζητείται προκαλεί ένα I/O συμβάν.
 - Η τεχνική MRU είναι πολύ καλύτερη σε αυτή την περίπτωση.

Παράδειγμα LRU

Αναφορές

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Buffer

Παράδειγμα LRU

*

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7																			

Παράδειγμα LRU

* *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7																		
	0																		

Παράδειγμα LRU

* * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7																	
	0	0																	
		1																	

Παράδειγμα LRU

* * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2																
	0	0	0																
		1	1																

Η 7 δεν έχει χρησιμοποιηθεί για το περισσότερο διάστημα και επιλέγεται ως θύμα

Παράδειγμα LRU

* * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2															
	0	0	0	0															
		1	1	1															

Παράδειγμα LRU

* * * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2														
	0	0	0	0	0														
		1	1	1	3														

Παράδειγμα LRU

* * * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2													
	0	0	0	0	0	0													
		1	1	1	3	3													

Παράδειγμα LRU

* * * * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4												
	0	0	0	0	0	0	0												
		1	1	1	3	3	3												

Παράδειγμα LRU

* * * * * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4											
	0	0	0	0	0	0	0	0											
		1	1	1	3	3	3	2											

Παράδειγμα LRU

* * * * * * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4										
	0	0	0	0	0	0	0	0	3										
		1	1	1	3	3	3	2	2										

Παράδειγμα LRU

* * * * * * * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0									
	0	0	0	0	0	0	0	0	3	3									
		1	1	1	3	3	3	2	2	2									

Παράδειγμα LRU

* * * * * * * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0								
	0	0	0	0	0	0	0	0	3	3	3								
		1	1	1	3	3	3	2	2	2	2								

Παράδειγμα LRU

* * * * * * * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0							
	0	0	0	0	0	0	0	0	3	3	3	3							
		1	1	1	3	3	3	2	2	2	2	2							

Παράδειγμα LRU

* * * * * * * * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	1						
	0	0	0	0	0	0	0	0	3	3	3	3	3						
		1	1	1	3	3	3	2	2	2	2	2	2						

Παράδειγμα LRU

* * * * * * * * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	1	1					
	0	0	0	0	0	0	0	0	3	3	3	3	3	3					
		1	1	1	3	3	3	2	2	2	2	2	2	2					

Παράδειγμα LRU

* * * * * * * * * * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1				
	0	0	0	0	0	0	0	0	3	3	3	3	3	3	0				
		1	1	1	3	3	3	2	2	2	2	2	2	2	2				

Παράδειγμα LRU

* * * * * * * * * * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1			
	0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0			
		1	1	1	3	3	3	2	2	2	2	2	2	2	2	2			

Παράδειγμα LRU

*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1		
	0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0		
		1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	2	7	

Παράδειγμα LRU

* * * * * * * * * * * * *

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	
	0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	
		1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	7	7	

Παράδειγμα LRU

*	*	*	*		*		*	*	*	*		*		*		*				
7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1	
7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	1	
	0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0	
		1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	2	7	7	7

Συνολικά έχουμε 12 page faults.

Clock Replacement Policy

- Οργάνωσε τα frames σε μια κυκλική λίστα. Κάθε πλαίσιο έχει ένα bit αναφοράς.
 - Referenced = 1 when pin_count is reduced to 0
- Επισκέψου τα πλαίσια ακολουθιακά.
 - If pin_count > 0 move to next frame,
 - If pin_count = 0 and referenced = 1 set referenced to 0 and move to next frame
 - If pin_count = 0 and referenced = 0 choose the current frame
- Εάν έχει διατρεχτεί ολόκληρη η λίστα χωρίς να επιλεγεί κάποιο πλαίσιο, τότε ο buffer δεν έχει κάποιον υποψήφιο για αντικατάσταση.

Παράδειγμα

- Θεωρούμε ότι τα δεδομένα είναι αποθηκευμένα σε 4 σελίδες $\Sigma 1, \Sigma 2, \Sigma 3, \Sigma 4$ και η απομονωτική μνήμη μπορεί να δεχθεί 3 σελίδες. Έστω το σύστημα δημιουργεί την παρακάτω ακολουθία αναφορών στις σελίδες: $\Sigma 1, \Sigma 2, \Sigma 3, \Sigma 4, \Sigma 2, \Sigma 2, \Sigma 1, \Sigma 2, \Sigma 1, \Sigma 1$

	FIFO	LRU						
$\Sigma 1$	<table border="1"><tr><td>$\Sigma 1$</td><td></td><td></td></tr></table>	$\Sigma 1$			<table border="1"><tr><td>$\Sigma 1$</td><td></td><td></td></tr></table>	$\Sigma 1$		
$\Sigma 1$								
$\Sigma 1$								
$\Sigma 2$	<table border="1"><tr><td>$\Sigma 1$</td><td>$\Sigma 2$</td><td></td></tr></table>	$\Sigma 1$	$\Sigma 2$		<table border="1"><tr><td>$\Sigma 1$</td><td>$\Sigma 2$</td><td></td></tr></table>	$\Sigma 1$	$\Sigma 2$	
$\Sigma 1$	$\Sigma 2$							
$\Sigma 1$	$\Sigma 2$							
$\Sigma 3$	<table border="1"><tr><td>$\Sigma 1$</td><td>$\Sigma 2$</td><td>$\Sigma 3$</td></tr></table>	$\Sigma 1$	$\Sigma 2$	$\Sigma 3$	<table border="1"><tr><td>$\Sigma 1$</td><td>$\Sigma 2$</td><td>$\Sigma 3$</td></tr></table>	$\Sigma 1$	$\Sigma 2$	$\Sigma 3$
$\Sigma 1$	$\Sigma 2$	$\Sigma 3$						
$\Sigma 1$	$\Sigma 2$	$\Sigma 3$						
$\Sigma 4$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 2$</td><td>$\Sigma 3$</td></tr></table>	$\Sigma 4$	$\Sigma 2$	$\Sigma 3$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 2$</td><td>$\Sigma 3$</td></tr></table>	$\Sigma 4$	$\Sigma 2$	$\Sigma 3$
$\Sigma 4$	$\Sigma 2$	$\Sigma 3$						
$\Sigma 4$	$\Sigma 2$	$\Sigma 3$						
$\Sigma 2$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 2$</td><td>$\Sigma 3$</td></tr></table>	$\Sigma 4$	$\Sigma 2$	$\Sigma 3$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 2$</td><td>$\Sigma 3$</td></tr></table>	$\Sigma 4$	$\Sigma 2$	$\Sigma 3$
$\Sigma 4$	$\Sigma 2$	$\Sigma 3$						
$\Sigma 4$	$\Sigma 2$	$\Sigma 3$						
$\Sigma 2$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 2$</td><td>$\Sigma 3$</td></tr></table>	$\Sigma 4$	$\Sigma 2$	$\Sigma 3$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 2$</td><td>$\Sigma 3$</td></tr></table>	$\Sigma 4$	$\Sigma 2$	$\Sigma 3$
$\Sigma 4$	$\Sigma 2$	$\Sigma 3$						
$\Sigma 4$	$\Sigma 2$	$\Sigma 3$						
$\Sigma 1$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 1$</td><td>$\Sigma 3$</td></tr></table>	$\Sigma 4$	$\Sigma 1$	$\Sigma 3$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 2$</td><td>$\Sigma 1$</td></tr></table>	$\Sigma 4$	$\Sigma 2$	$\Sigma 1$
$\Sigma 4$	$\Sigma 1$	$\Sigma 3$						
$\Sigma 4$	$\Sigma 2$	$\Sigma 1$						
$\Sigma 2$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 1$</td><td>$\Sigma 2$</td></tr></table>	$\Sigma 4$	$\Sigma 1$	$\Sigma 2$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 2$</td><td>$\Sigma 1$</td></tr></table>	$\Sigma 4$	$\Sigma 2$	$\Sigma 1$
$\Sigma 4$	$\Sigma 1$	$\Sigma 2$						
$\Sigma 4$	$\Sigma 2$	$\Sigma 1$						
$\Sigma 1$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 1$</td><td>$\Sigma 2$</td></tr></table>	$\Sigma 4$	$\Sigma 1$	$\Sigma 2$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 2$</td><td>$\Sigma 1$</td></tr></table>	$\Sigma 4$	$\Sigma 2$	$\Sigma 1$
$\Sigma 4$	$\Sigma 1$	$\Sigma 2$						
$\Sigma 4$	$\Sigma 2$	$\Sigma 1$						
$\Sigma 1$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 1$</td><td>$\Sigma 2$</td></tr></table>	$\Sigma 4$	$\Sigma 1$	$\Sigma 2$	<table border="1"><tr><td>$\Sigma 4$</td><td>$\Sigma 2$</td><td>$\Sigma 1$</td></tr></table>	$\Sigma 4$	$\Sigma 2$	$\Sigma 1$
$\Sigma 4$	$\Sigma 1$	$\Sigma 2$						
$\Sigma 4$	$\Sigma 2$	$\Sigma 1$						

Εξυπηρέτηση Αιτήσεων

- Αναζήτηση της σελίδας στο μέσο αποθήκευσης
- Ο **ελεγκτής δίσκου** διαθέτει μία **ουρά εξυπηρέτησης** αιτήσεων, όπου αποθηκεύονται αιτήσεις προς προσπέλαση.
- Για την εξυπηρέτηση μίας αίτησης απαιτείται συνολικός χρόνος $T_{seek} + T_{rot} + T_{trans}$
 - T_{seek} πολύ μεγαλύτερος από T_{rot} και T_{trans}
- Οι μέθοδοι εξυπηρέτησης προσπαθούν να ελαχιστοποιήσουν το χρόνο T_{seek} .

Εξυπηρέτηση Αιτήσεων

- Οι κυριότερες μέθοδοι εξυπηρέτησης είναι:
 - **Πρώτη ερχόμενη, πρώτη εξυπηρετούμενη** (first come, first served - **FCFS**)
 - Εξυπηρετεί τις αιτήσεις με τη σειρά που έρχονται
 - + Εύκολη στην υλοποίηση μέθοδος
 - Αν διαδοχικές αιτήσεις αναφέρονται σε απομακρυσμένες ατράκτους τότε η κεφαλή διαγράφει μεγάλες αποστάσεις.

Εξυπηρέτηση Αιτήσεων

- Οι κυριότερες μέθοδοι εξυπηρέτησης είναι:
 - **Πρώτα η μικρότερη αναζήτηση** (shortest seek time first - **SSTF**)
 - Βελτίωση της προηγούμενης μεθόδου
 - + Από τις αιτήσεις εξυπηρετείται αυτή που θα προκαλέσει την μικρότερη μετακίνηση
 - Μειονέκτημα: κάποιες αιτήσεις δε θα εξυπηρετηθούν ποτέ ή θα αργήσουν πάρα πολύ (ασχέτως αν ήρθαν πρώτες)

Εξυπηρέτηση Αιτήσεων

- Οι κυριότερες μέθοδοι εξυπηρέτησης είναι:
 - **Σάρωση (Scan)**
 - Η κεφαλή μετακινείται από το ένα άκρο μέχρι το άλλο εξυπηρετώντας αιτήσεις.
 - Όταν η κεφαλή φτάσει στο ένα άκρο, η φορά αντιστρέφεται

Εξυπηρέτηση Αιτήσεων

- Οι κυριότερες μέθοδοι εξυπηρέτησης είναι:
 - **LOOK**
 - Αποτελεί βελτίωση της SCAN
 - + Η κεφαλή δεν μετακινείται μέχρι τέλους αλλά σταματά στην τελευταία αίτηση που αναφέρεται στη μεγαλύτερη ή μικρότερη άτρακτο

Σύνοψη

- Η φυσική οργάνωση των δεδομένων της βάσης σχετίζεται άμεσα με την απόδοση του συστήματος.
- Για το λόγο αυτό, εκτελούνται από το ΣΔΒΔ πολλές λειτουργίες διαχείρισης αρχείων που παραδοσιακά βρίσκονται στη δικαιοδοσία του ΛΣ.

Σύνοψη

- Τα δεδομένα της βάσης αποθηκεύονται σε αρχεία.
- Κάθε αρχείο αποτελείται από ένα σύνολο εκτάσεων.
- Η έκταση διαχωρίζεται σε σελίδες, και τέλος
- Κάθε σελίδα περιέχει ένα σύνολο εγγραφών σταθερού ή μεταβλητού μήκους.
- Η μεταφορά δεδομένων από/προς τη δευτερεύουσα μνήμη πραγματοποιείται με τη βοήθεια απομονωτικών μνημών.

Σύνοψη

- Η μνήμη απομόνωσης είναι τμήμα της κύριας μνήμης, το οποίο χρησιμοποιείται αποκλειστικά από το ΣΔΒΔ.
- Η αναζήτηση δεδομένων πραγματοποιείται πρώτα στη μνήμη απομόνωσης, και μόνο σε περίπτωση αποτυχίας επιχειρείται προσπάθεια στη δευτερεύουσα μνήμη.