

Κεφάλαιο 7

ΑΡΧΕΙΑ ΣΕΙΡΙΑΚΑ ΜΕ ΔΕΙΚΤΗ

7.1 Εισαγωγή

7.2 Μέθοδος των καταλόγων κυλίνδρων και επιφανειών

7.3 Μέθοδος των σελίδων καταλόγου και δεδομένων

7.4 Ασκήσεις

Κεφάλαιο 7

ΣΕΙΡΙΑΚΑ ΑΡΧΕΙΑ ΜΕ ΔΕΙΚΤΗ

7.1 Εισαγωγή

Τα σειριακά αρχεία (ταξινομημένα ή αταξινομήτα) δεν αποτελούν αποτελεσματικές δομές για την αναζήτηση μίας συγκεκριμένης εγγραφής. Η οργάνωση των **σειριακών αρχείων με δείκτη** (index sequential files) μοιάζει με τα ταξινομημένα σειριακά αρχεία κατά το ότι τα δεδομένα αποθηκεύονται με τον ίδιο τρόπο στο δίσκο, δηλαδή σειριακά και ταξινομημένα ως προς το πρωτεύον κλειδί. Συνεπώς, αν από τη δομή αυτή ζητηθεί η ανάκτηση όλων των εγγραφών, τότε δεν αλλάζει τίποτε ως προς την αντίστοιχη επεξεργασία των σειριακών αρχείων. Ωστόσο, η νέα αυτή δομή συνοδεύεται και από έναν **κατάλογο ή δείκτη** (index), που αποτελεί ένα μικρότερο αρχείο με σκοπό την ταχύτερη υποστήριξη της αναζήτησης μίας συγκεκριμένης εγγραφής.

Στο σημείο αυτό μία παρένθεση κρίνεται απαραίτητη για την αποφυγή παρεξήγησης. Στην αγγλική ορολογία υπάρχει ο όρος pointer, που εύλογα μεταφράζεται ως δείκτης, αλλά υπάρχει και ο όρος index που μεταφράζεται ως κατάλογος ή ως δείκτης κατά περίπτωση. Στο εξής, ο όρος index θα αποδίδεται ως κατάλογος, αλλά τα αντίστοιχα αρχεία θα ονομάζονται σειριακά με δείκτη, γιατί αυτός ο όρος έχει επικρατήσει στην ελληνική βιβλιογραφία (καθώς επίσης και ο όρος σειριακά δεικτοδοτημένα αρχεία).

Εδώ και πολλά χρόνια κυκλοφορούν εμπορικά πακέτα που υλοποιούν αυτήν την οργάνωση. Η ανάπτυξη νέων κομψότερων μεθόδων έχει μειώσει τη ζήτηση των πακέτων αυτών, που ωστόσο ακόμη και σήμερα χρησιμοποιούνται αρκετά πλατιά (λόγω αδράνειας). Τα κυριότερα πακέτα είναι τα αρχεία ISAM (Index Sequential Access Method), τα αρχεία VSAM (Virtual

Storage Access Method) και τα αρχεία SIS (Scope Indexed Sequential). Τα δύο πρώτα πακέτα παρουσιάστηκαν με το σύστημα 370 της IBM, ενώ το τρίτο χρησιμοποιείται σε μεγάλα συστήματα υπολογιστών της CDC. Τα αρχεία ISAM χρησιμοποιούν τη μέθοδο των καταλόγων κυλίνδρων και επιφανειών, ενώ τα αρχεία VSAM και SIS χρησιμοποιούν τη μέθοδο των σελίδων καταλόγου και δεδομένων.

7.2 Μέθοδος των καταλόγων κυλίνδρων και επιφανειών

Η μέθοδος των καταλόγων κυλίνδρων και επιφανειών (cylinder and surface indexing) χρησιμοποιείται κυρίως σε συστήματα της IBM και είναι ευρύτερα γνωστή ως οργάνωση ISAM. Η οργάνωση ISAM στο παρελθόν αποτέλεσε πρότυπο. Για το λόγο αυτό, σύγχρονα συστήματα διαχείρισης βάσεων δεδομένων, όπως για παράδειγμα η CA-INGRES, παρ' ότι χρησιμοποιούν άλλες πιο αποτελεσματικές μεθόδους (όπως τα B⁺-δένδρα, που θα αναπτυχθούν στη συνέχεια), αποκαλούν τη δεικτοδοτημένη τους οργάνωση, μέθοδο ISAM. Η οργάνωση αυτή στηρίζεται στα φυσικά χαρακτηριστικά του δίσκου, δηλαδή είναι άμεσα εξαρτώμενη από το υλικό, σε αντίθεση με τις οργανώσεις VSAM και SIS. Στη συνέχεια δίνεται μία σύντομη περιγραφή της δομής αυτής και θα ακολουθήσει ένα παράδειγμα για την καλύτερη κατανόησή της. Τέλος, θα δοθούν αναλυτικοί τύποι εκτίμησης του κόστους των διαφόρων διεργασιών.

Κατά τη δημιουργία ενός αρχείου ISAM, οι εγγραφές ταξινομούνται κατά αύξουσα τάξη μεγέθους του κλειδιού τους και αποθηκεύονται σειριακά στις διαδοχικές ατράκτους ενός κυλίνδρου, αφήνοντας συνήθως κενές την πρώτη και μερικές από τις τελευταίες ατράκτους. Επίσης δεσμεύεται κενός χώρος σε κάθε άτρακτο με σκοπό την μελλοντική αποθήκευση νέων εγγραφών. Ακόμη, ο πρώτος και μερικοί τελευταίοι κύλινδροι αφήνονται κενοί. Όταν ο δεύτερος κύλινδρος γεμίσει, τότε η φόρτωση συνεχίζεται στους επόμενους κυλίνδρους. Ο μέχρι στιγμής κατειλημμένος χώρος ονομάζεται κύρια περιοχή δεδομένων (prime data area).

Όταν όλες οι εγγραφές έχουν φορτωθεί στους κυλίνδρους, τότε το σύστημα δημιουργεί τον κατάλογο, που είναι μία στατική δενδρική δομή. Η δομή αυτή συνήθως αποτελείται από τρία επίπεδα: τον κύριο κατάλογο

(main index), τον κατάλογο κυλίνδρων (cylinder index) και τον κατάλογο ατράκτων (track index).

Ο κατάλογος κυλίνδρων αποτελείται από έναν αριθμό ζευγών στοιχείων ίσο με τον αριθμό των κυλίνδρων που καταλαμβάνει το αρχείο. Κάθε ζεύγος αποτελείται από την τιμή του μεγαλύτερου κλειδιού για κάθε κύλινδρο και τη διεύθυνση του αντίστοιχου κυλίνδρου. Η θέση όπου θα αποθηκευθεί ο κατάλογος κυλίνδρων είναι θέμα βελτιστοποίησης σε συνάρτηση με τα χαρακτηριστικά της χρήσης του αρχείου. Αν δεν είναι γνωστή η πιθανότητα προσπελάσεων στους κυλίνδρους, τότε ο κατάλογος αυτός αποθηκεύεται στο μεσαίο κύλινδρο του αρχείου, ειδάλλως αποθηκεύεται μεταξύ των κυλίνδρων με τη μεγαλύτερη πιθανότητα προσπέλασης.

Αν ο κατάλογος κυλίνδρων είναι ογκώδης (και συνεπώς δαπανηρός κατά την επεξεργασία του), τότε δημιουργείται ο κύριος κατάλογος που αποτελείται από τόσα ζεύγη στοιχείων, όσες είναι οι άτρακτοι που καταλαμβάνει ο κατάλογος κυλίνδρων. Το λογισμικό της IBM δίνει τη δυνατότητα δημιουργίας μέχρι πέντε επιπέδων καταλόγου επάνω από το επίπεδο του καταλόγου κυλίνδρων για υπερβολικά μεγάλα αρχεία. Πολλές φορές αυτά τα επίπεδα καταλόγου αποθηκεύονται σε δίσκους ημιαγωγών (δες Κεφάλαιο 2.5) για την επιτάχυνση της διαδικασίας αναζήτησης. Κάθε ζεύγος του κύριου καταλόγου δίνει την τιμή του μεγαλύτερου κλειδιού για κάθε άτρακτο του καταλόγου κυλίνδρων και την αντίστοιχη διεύθυνση. Όσο το αρχείο είναι ανοικτό για επεξεργασία, ο κύριος κατάλογος είναι αποθηκευμένος στην κύρια μνήμη και συνεπώς προσπελάζεται με πρακτικά μηδαμινό κόστος.

Στο χαμηλότερο επίπεδο των καταλόγων βρίσκεται ο κατάλογος ατράκτων. Στην πρώτη άτρακτο κάθε κυλίνδρου αποθηκεύεται ο αντίστοιχος κατάλογος ατράκτων και αποτελείται από δύο ζεύγη στοιχείων. Το πρώτο ζεύγος ονομάζεται κανονική είσοδος (normal entry) και αποτελείται από την τιμή του μεγαλύτερου κλειδιού για κάθε άτρακτο και την αντίστοιχη διεύθυνση της άτρακτου. Το δεύτερο ζεύγος ονομάζεται είσοδος υπερχείλισης (overflow entry) και αποτελείται από την τιμή του μεγαλύτερου κλειδιού, που προέρχεται από την άτρακτο αυτή αλλά είναι αποθηκευμένο στην περιοχή υπερχείλισης (θα εξετασθεί στη συνέχεια) και την αντίστοιχη διεύθυνση. Αν N_{key_i} και O_{key_i} είναι αντίστοιχα η τιμή του κλειδιού της κανονικής εισόδου και της εισόδου υπερχείλισης της άτρακτου i , τότε ισχύει:

$$N_{key_1} \leq O_{key_1} < N_{key_2} \leq O_{key_2} < \dots < N_{key_n} \leq O_{key_n}$$

Αν $N_{key_i} = O_{key_i}$, τότε έπεται ότι δεν υπάρχουν εγγραφές υπερχείλισης.

Λόγω των εισαγωγών η κύρια περιοχή αργά ή γρήγορα εξαντλείται. Τότε αποδίδεται στο αρχείο η περιοχή υπερχειλίσης (overflow area), που διακρίνεται στην τοπική ή κυλινδρική ή ενσωματωμένη (local, cylinder, embedded) και στην ολική ή ανεξάρτητη (global, independent) περιοχή υπερχειλίσης. Η τοπική περιοχή υπερχειλίσης αποτελείται από τις τελευταίες ατράκτους κάθε κυλίνδρου, ενώ η ολική περιοχή υπερχειλίσης αποτελείται από κάποιους κυλίνδρους στο τέλος του αρχείου. Έτσι για την προσπέλαση στην τοπική περιοχή υπερχειλίσης δεν υπάρχει επιπλέον επιβάρυνση με κόστος εντοπισμού κυλίνδρου.

Η αναζήτηση γίνεται με διαδοχικές προσπελάσεις στα επίπεδα του καταλόγου, στην κύρια περιοχή, και (ίσως) στην περιοχή υπερχειλίσης. Μία νέα εγγραφή ανάλογα με την τιμή του κλειδιού της εισάγεται στην κύρια ή στην περιοχή υπερχειλίσης εκτοπίζοντας πιθανώς άλλες εγγραφές. Στην περίπτωση διαγραφής, η εγγραφή δεν απομακρύνεται φυσικά από το αρχείο, αλλά ένα ιδιαίτερο πεδίο της, μία 'σημαία', 'ανεβαίνει' για να το δηλώσει λογικά. Έτσι ο χώρος δεν αποδίδεται και πάλι για χρήση αλλά μένει ανεκμετάλλετος προσωρινά. Είναι δυνατόν ο χώρος διαγραφείσας εγγραφής από την κύρια περιοχή να καταληφθεί και πάλι από μελλοντική εισαγωγή με ανάλογη διευθέτηση των εγγραφών της κύριας περιοχής. Αν κάποια εγγραφή διαγραφεί από την περιοχή υπερχειλίσης, τότε ενημερώνεται σχετικά η συνδεδεμένη λίστα. Σε περίπτωση μελλοντικής εισαγωγής στην περιοχή υπερχειλίσης, αυτή η λίστα κατευθύνει την αποθήκευση της νέας εγγραφής σε θέση όπου ήταν αποθηκευμένη προηγουμένως μία άλλη εγγραφή. Προφανώς οι διαγραφές δεν αντιστοιχούν μία προς μία προς τις εισαγωγές, άρα είναι βέβαιο ότι θα υπάρχει χώρος ανεκμετάλλετος. Με τη διαδικασία της αναδιοργάνωσης (reorganization) μπορεί να γίνει ταχτοποίηση του αρχείου και οι εγγραφές υπερχειλίσης να μεταφερθούν στην κύρια περιοχή. Στη συνέχεια παρουσιάζεται ένα παράδειγμα για την καλύτερη κατανόηση των εννοιών και διαδικασιών που αναφέρθηκαν.

Στο Σχήμα 7.1 παρουσιάζεται δείγμα αρχείου που καταλαμβάνει δεκατέσσερις κυλίνδρους. Οι κύλινδροι αυτοί είναι αριθμημένοι από 100 ως 113. Ο πρώτος κύλινδρος (υπ' αριθμό 100) χρησιμεύει για την αποθήκευση του καταλόγου κυλίνδρων, ενώ ο τελευταίος (υπ' αριθμό 113) ως ολική περιοχή υπερχειλίσης. Άρα, η κύρια περιοχή δεδομένων αποτελείται από δώδεκα κυλίνδρους. Οι κύλινδροι έχουν 6 ατράκτους χωρητικότητας τριών εγγραφών. Σε κάθε κύλινδρο της κύριας περιοχής, η πρώτη και η τελευταία άτρακτος χρησιμεύουν για την αποθήκευση του καταλόγου άτρακτων και της τοπικής περιοχής υπερχειλίσης, αντίστοιχα.

Κύλινδρος C100	Κύλινδρος C101	...	Κύλινδρος C112	Κύλινδρος C113
Κατάλογος κυλίνδρων	Κατάλογος ατράκτων	...	Κατάλογος ατράκτων	Ολική περιοχή υπερχείλισης
	4 5 10		291 302 316	
	12 19 52		346 351 356	
	56 71 90		382 401 411	
	100 113 142		425 431 437	
	Τοπική περιοχή υπερχείλισης		Τοπική περιοχή υπερχείλισης	

Σχήμα 7.1: Σειριακό αρχείο με δείκτη σε 14 κυλίνδρους.

Μεγαλύτερο κλειδί	Διεύθυνση κυλίνδρου	Κανονική είσοδος	Είσοδος υπερχείλισης
142	C101	10 S1	10 S1
⋮	⋮	52 S2	52 S2
437	C112	90 S3	90 S3
		142 S4	142 S4

(α) (β)

Σχήμα 7.2: Κατάλογος κυλίνδρων και κατάλογος ατράκτων κυλίνδρου 101.

Στο Σχήμα 7.2α παρουσιάζεται ο κατάλογος κυλίνδρων με τις τιμές των μεγαλύτερων κλειδιών, που είναι αποθηκευμένα στους δώδεκα κυλίνδρους. Στο Σχήμα 7.2β το ενδιαφέρον εστιάζεται στον κατάλογο ατράκτων του κυλίνδρου 101. Όταν γίνεται αναζήτηση μίας εγγραφής με βάση την τιμή του πρωτεύοντος κλειδιού, τότε τα κλειδιά αυτά σαρώνονται σειριακά μέχρι να βρεθεί ένα μεγαλύτερο κλειδί από εκείνο για το οποίο γίνεται η αναζήτηση.

Έστω ότι στο αρχείο εισάγεται μία εγγραφή με κλειδί 8. Είναι γνωστό βέβαια ότι σε όλους τους τύπους αρχείων κάθε εισαγωγή ενεργοποιεί τη ρουτίνα αναζήτησης της συγκεκριμένης εγγραφής για δύο λόγους. Πρώτον, για να αποφευχθεί η διπλή αποθήκευση, και δεύτερον, για να εντοπισθεί η θέση όπου τελικά θα γίνει η εισαγωγή. Σαρώνοντας τον κατάλογο του Σχήματος 7.2α διαπιστώνεται ότι η εγγραφή αυτή ανήκει στον κύλινδρο 101.

Με δεύτερη σάρωση του καταλόγου ατράκτων, όπως φαίνεται στο Σχήμα 7.2β, διαπιστώνεται ότι η εγγραφή 8 πρέπει να αποθηκευθεί στην άτρακτο 1. Ωστόσο, η άτρακτος αυτή είναι πλήρης αφού ήδη περιέχει τρεις εγγραφές. Για να διατηρηθεί η αύξουσα σειρά των κλειδιών παρεμβάλλεται η εγγραφή 8 στη θέση μεταξύ των εγγραφών 5 και 10. Έτσι η εγγραφή 10 εκτοπίζεται προς την τοπική περιοχή υπερχειλίσσης, δηλαδή στην άτρακτο 5, θέση 0. Ενημερώνεται, λοιπόν, ανάλογα η κανονική είσοδος και η είσοδος υπερχειλίσσης της ατράκτου 1 του κυλίνδρου 101. Το αποτέλεσμα της εισαγωγής παρουσιάζεται στο Σχήμα 7.3α.

Έστω ότι στη συνέχεια πρέπει να εισαχθεί η εγγραφή 14. Με παρόμοιο τρόπο διαπιστώνεται ότι η θέση της είναι στον κύλινδρο 101 και στην άτρακτο 2. Η εγγραφή 14 εισάγεται μεταξύ των εγγραφών 12 και 19 εκτοπίζοντας την εγγραφή 52 στην περιοχή υπερχειλίσσης. Πιο συγκεκριμένα, η εγγραφή 52 τοποθετείται στη θέση 1 της ατράκτου 5, οπότε ενημερώνεται ανάλογα η αντίστοιχη κανονική είσοδος του καταλόγου, όπως φαίνεται στο Σχήμα 7.3β.

Στη συνέχεια εισάγεται η εγγραφή 6, που αποθηκεύεται στην άτρακτο 1 του κυλίνδρου 101. Έτσι η εγγραφή 8 εκτοπίζεται στη θέση 2 της ατράκτου 5, και ενημερώνεται σχετικά ο κατάλογος. Η φυσική σειρά των εγγραφών υπερχειλίσσης δεν έχει καμία σημασία. Οι εγγραφές αυτές με τη βοήθεια των καταλόγων προσπελάζονται κατ' ευθείαν από τις ατράκτους της κύριας περιοχής. Μάλιστα, από κάθε άτρακτο ξεκινά μία συνδεδεμένη λίστα που συντηρεί τις εγγραφές της ατράκτου διατεταγμένες κατά αύξουσα τάξη. Το αποτέλεσμα της εισαγωγής της εγγραφής 6 παρουσιάζεται στο Σχήμα 7.3γ.

Από αυτή τη διαδικασία εισαγωγών που περιγράφηκε εξάγονται τα εξής συμπεράσματα. Η στήλη των μεγαλύτερων κλειδιών της κανονικής εισόδου είναι πιθανόν να αλλάξει με την είσοδο των νέων εγγραφών, οι στήλες της διεύθυνσης της κανονικής εισόδου και του μεγαλύτερου κλειδιού της εισόδου υπερχειλίσσης δεν αλλάζουν ποτέ, ενώ η στήλη της διεύθυνσης της εισόδου υπερχειλίσσης είναι βέβαιο ότι θα αλλάξει αν κάποια εγγραφή μεταφερθεί στην περιοχή υπερχειλίσσης.

Έστω, λοιπόν, ότι πρέπει να διαγραφεί η εγγραφή 14. Το αποτέλεσμα της διαγραφής φαίνεται στο Σχήμα 7.3δ, όπου ο χώρος της εγγραφής έχει γραμμοσκιασθεί. Στη συνέχεια ακολουθεί μαθηματική ανάλυση μερικών ποσοτικών στοιχείων της οργάνωσης αυτής, που αφορούν στο χρονικό κόστος των διαφόρων διεργασιών επί της δομής.

(α) εισαγωγή 8

Κανονική είσοδος		Είσοδος υπερχείλισης	
8	S1	10	S5R0
52	S2	52	S2
90	S3	90	S3
142	S4	142	S4

S1	4	5	8
S2	12	19	52
S3	56	71	90
S4	100	113	142
S5	10		

(β) εισαγωγή 14

Κανονική είσοδος		Είσοδος υπερχείλισης	
8	S1	10	S5R0
19	S2	52	S5R1
90	S3	90	S3
142	S4	142	S4

S1	4	5	8
S2	12	14	19
S3	56	71	90
S4	100	113	142
S5	10	52	

(γ) εισαγωγή 6

Κανονική είσοδος		Είσοδος υπερχείλισης	
6	S1	10	S5R2
19	S2	52	S5R1
90	S3	90	S3
142	S4	142	S4

S1	4	5	6
S2	12	14	19
S3	56	71	90
S4	100	113	142
S5	10	52	8

(δ) διαγραφή 14

Κανονική είσοδος		Είσοδος υπερχείλισης	
6	S1	10	S5R2
19	S2	52	S5R1
90	S3	90	S3
142	S4	142	S4

S1	4	5	6
S2	12	14	19
S3	56	71	90
S4	100	113	142
S5	10	52	8

Σχήμα 7.3: Κατάλογος ατράχτων και κύρια περιοχή κυλίνδρου 101 μετά την εισαγωγή των εγγραφών 8, 14, 6 και τη διαγραφή της εγγραφής 108.

7.2.1 Προσπέλαση εγγραφής

Για την προσπέλαση μίας εγγραφής σε ένα σειριακό αρχείο με δείκτη πρέπει, προφανώς, να προσπελασθεί πρώτα ο κατάλογος. Υποτίθεται ότι ο κύριος κατάλογος είναι αποθηκευμένος στην κύρια μνήμη, οπότε ο αντίστοιχος χρόνος αναζήτησης (c) είναι αμελητέος, ενώ τα άλλα επίπεδα του καταλόγου είναι αποθηκευμένα στο δίσκο. Αν δεν υπάρχουν εγγραφές υπερχειλίσις, τότε το χρονικό κόστος είναι:

$$T_{\text{προσ,κυρ}} = c + 2 \times (s + r + btt) + (r + btt)$$

όπου το γινόμενο $2 \times (r + btt)$ δίνει το χρόνο επεξεργασίας του καταλόγου κυλίνδρων και του καταλόγου ατράκτων, ενώ η ποσότητα $r + btt$ παριστά το χρόνο περιστροφής και μεταφοράς της κατάλληλης σελίδας από την κύρια περιοχή του αρχείου.

Ωστόσο, είναι πολύ πιθανό να υπάρχουν εγγραφές υπερχειλίσις στο αρχείο. Αν το αρχείο έχει n εγγραφές στην κύρια περιοχή και o στην περιοχή υπερχειλίσις, τότε η πιθανότητα μία εγγραφή να βρίσκεται στην περιοχή υπερχειλίσις είναι:

$$P_{\text{υπερ}} = \frac{o}{n + o}$$

Οι εγγραφές υπερχειλίσις συνδέονται με μία αλυσίδα από την κύρια σελίδα από όπου προήλθαν. Προσεγγιστικά το μήκος της αλυσίδας ισούται με:

$$L_{\text{αλυσ}} = P_{\text{υπερ}} \times Bfr$$

Αν οι εγγραφές υπερχειλίσις είναι αποθηκευμένες σε άλλον κύλινδρο, τότε υπεισέρχεται νέος χρόνος αναζήτησης s' , τουλάχιστο για την προσπέλαση της πρώτης εγγραφής της αλυσίδας. Ο χρόνος αυτός είναι:

$$T_{\text{προσ,υπερ}} = P_{\text{υπερ}} \times (s' + r + btt)$$

Είναι πολύ πιθανό ότι οι υπόλοιπες εγγραφές της αλυσίδας ανήκουν σε διαφορετικές σελίδες. Μία εκτίμηση του αριθμού των εγγραφών της αλυσίδας υπερχειλίσις (και συνεπώς των σελίδων) που θα προσπελασθούν είναι $\frac{(L_{\text{αλυσ}} - 1) + 1}{2}$. Πρέπει, βέβαια, να ληφθεί υπ' όψη ότι προσπέλαση στην αλυσίδα θα γίνει μόνο με την προϋπόθεση ότι αναζητείται εγγραφή υπερχειλίσις. Συνεπώς ισχύει:

$$T_{\text{προσ,αλυσ}} = P_{\text{υπερ}} \times \frac{L_{\text{αλυσ}}}{2} \times (r + btt)$$

Τελικά, ο χρόνος προσπέλασης μίας εγγραφής είναι:

$$T_{\text{προσ}} = T_{\text{προσ,κυρ}} + T_{\text{προσ,υπερ}} + T_{\text{προσ,αλυσ}}$$

7.2.2 Προσπέλαση επόμενης εγγραφής

Για την προσπέλαση της επόμενης εγγραφής δεν χρειάζεται και πάλι επεξεργασία του καταλόγου. Ωστόσο, η προσπέλαση της επόμενης εγγραφής είναι μία αρκετά σύνθετη διαδικασία και υποδιαιρείται σε πολλές περιπτώσεις:

- η τρέχουσα εγγραφή είναι αποθηκευμένη σε μία κύρια σελίδα και η επόμενη εγγραφή βρίσκεται στην ίδια σελίδα,
- η τρέχουσα εγγραφή είναι η τελευταία της κύριας σελίδας, δεν υπήρξαν εισαγωγές και η επόμενη εγγραφή είναι αποθηκευμένη στην επόμενη σελίδα του ίδιου κυλίνδρου,
- η τρέχουσα εγγραφή είναι η τελευταία της κύριας σελίδας, δεν υπήρξαν εισαγωγές, αλλά η επόμενη εγγραφή είναι αποθηκευμένη στην πρώτη σελίδα του επόμενου κυλίνδρου,
- η τρέχουσα εγγραφή είναι η τελευταία της κύριας σελίδας, έχουν υπάρξει εισαγωγές και η επόμενη εγγραφή είναι αποθηκευμένη στην περιοχή υπερχειλίσης,
- η τρέχουσα εγγραφή είναι μία περίπτωση εισαγωγής και η επόμενη εγγραφή είναι αποθηκευμένη σε άλλη σελίδα στον ίδιο κύλινδρο, και τέλος,
- η τρέχουσα εγγραφή είναι μία περίπτωση εισαγωγής και η επόμενη εγγραφή είναι αποθηκευμένη σε άλλη σελίδα.

Κάθε μία από τις περιπτώσεις αυτές πρέπει να αντιμετωπισθεί ιδιαίτερα. Έτσι εισάγονται νέες μεταβλητές:

$$\begin{aligned} P_{\text{κυρ}} \text{ (τρέχουσα εγγραφή σε κύρια σελίδα)} &= 1 - P_{\text{υπερ}} \\ P_{\text{δία,σελ}} \text{ (επόμενη εγγραφή στην ίδια σελίδα)} &= 1 - 1/Bfr \\ P_{\text{οχι,εισ}} \text{ (μη εισαγωγές στην ίδια σελίδα)} &= 1 - P_{\text{υπερ}} \\ P_{\text{δίο,κυλ}} \text{ (επόμενη σελίδα στον ίδιο κύλινδρο)} &= 1 - 1/C \\ P_{\text{οχι,αλυσ}} \text{ (τρέχουσα εγγραφή υπερχειλίσης δεν είναι} \\ &\text{τελευταία της αλυσίδας)} &= 1 - 1/L_{\text{αλυσ}} \end{aligned}$$

Τελικά, ο χρόνος προσπέλασης της επόμενης εγγραφής είναι το άθροισμα που ακολουθεί, όπου κάθε όρος αντιστοιχεί στις έξι περιπτώσεις που αναφέρθηκαν:

$$T_{\text{επομ}} = \begin{cases} P_{\text{χυρ}} P_{\text{ιδια,σελ}} c + \\ P_{\text{χυρ}} (1 - P_{\text{ιδια,σελ}}) P_{\text{οχι,εισ}} P_{\text{ιδιο,κυλ}} (r + btt) + \\ P_{\text{χυρ}} (1 - P_{\text{ιδια,σελ}}) P_{\text{οχι,εισ}} (1 - P_{\text{ιδιο,κυλ}}) (s + r + btt) + \\ (1 - P_{\text{χυρ}}) P_{\text{οχι,αλυσ}} (r + btt) + \\ (1 - P_{\text{χυρ}}) (1 - P_{\text{οχι,αλυσ}}) (s' + r + btt) \end{cases}$$

Όταν οι περιοχές υπερχειλίσης βρίσκονται στους ίδιους κυλίνδρους με τα δεδομένα, τότε ο όρος s' μηδενίζεται. Επίσης αν αγνοηθεί ο χρόνος επεξεργασίας του κύριου καταλόγου και αν αγνοηθούν οι χρόνοι αναζήτησης των κυλίνδρων, τότε ο προηγούμενος τύπος απλοποιείται ως εξής:

$$T_{\text{επομ}} = (r + btt) \times \frac{n + o \times Bfr}{(n + o) \times Bfr}$$

7.2.3 Εισαγωγή εγγραφής

Η εισαγωγή μίας εγγραφής πρέπει να αντιμετωπισθεί ως ένα σύνολο δύσκολα μοντελοποιήσιμων περιπτώσεων. Για απλοποίηση, εδώ υποτίθεται ότι η νέα εγγραφή είτε αποθηκεύεται στην περιοχή υπερχειλίσης, είτε ωθεί μία άλλη εγγραφή στην περιοχή υπερχειλίσης.

Η διαδικασία εισαγωγής αρχίζει με την ανάγνωση των προηγούμενων δεδομένων ή μίας σελίδας υπερχειλίσης. Επομένως αρχικά υπάρχει ένα κόστος ίσο με $T_{\text{προσ}}$. Η πιθανότητα η νέα εγγραφή με τη λογικά προηγούμενή της να βρίσκονται στην ίδια σελίδα είναι πρακτικά πολύ μικρή για μεγάλα αρχεία. Ωστόσο, υποτίθεται ότι βρίσκονται στον ίδιο κύλινδρο, άρα χωρίς χρόνο εντοπισμού απαιτείται $r + btt$ για να προσπελασθεί η νέα σελίδα, ενώ $2r$ απαιτούνται για την επανα-αποθήκευση της σελίδας. Συνεπώς τελικά ισχύει:

$$T_{\text{εισ}} = T_{\text{προσ}} + 5r + btt$$

Ας σημειωθεί ότι η επίδραση του μήκους της αλυσίδας υπερχειλίσης συμπεριλαμβάνεται στον όρο $T_{\text{προσ}}$.

7.2.4 Ανανέωση εγγραφής

Προκύπτει εύκολα ότι το χρονικό κόστος ανανέωσης μίας εγγραφής (χωρίς να αλλάζει το μήκος της ή η τιμή του κλειδιού της) ισούται με:

$$T_{\text{αναν}} = T_{\text{προσ}} + 2r$$

Στη γενική περίπτωση πρέπει να γίνει αποδεκτό ότι η εγγραφή μαρκάρεται ως διαγραμμένη και ότι εισάγεται μία νέα εγγραφή. (Εξήγηση της διαγραφής ακολουθεί.) Συνεπώς ισχύει:

$$T_{\text{αναν}} = T_{\text{προσ}} + 2r + T_{\text{εισ}} = 2T_{\text{προσ}} + 7r + btt$$

7.2.5 Διαγραφή εγγραφής

Το χρονικό κόστος για τη διαγραφή μίας εγγραφής ισούται με τον πρώτο όρο $T_{\text{αναν}}$, γιατί η περιοχή απλώς μαρκάρεται με μία σημαία χωρίς η εγγραφή πραγματικά να απομακρυνθεί. Άρα:

$$T_{\text{διαγ}} = T_{\text{προσ}} + 2r$$

7.2.6 Εξαντλητική ανάγνωση αρχείου

Η περίπτωση αυτή παρουσιάζεται όταν η αναζήτηση δεν γίνεται με βάση το πρωτεύον κλειδί ή με βάση το κλειδί ενός δευτερεύοντος καταλόγου. Η ανάγνωση του αρχείου μπορεί να γίνει είτε ακολουθώντας τη λογική σειρά των εγγραφών, είτε λαμβάνοντας τους κυλίνδρους με τη σειρά και αδιαφορώντας για το αν οι εγγραφές δίνονται ταξινομημένες στο χρήστη. Και στις δύο περιπτώσεις ο κατάλογος αγνοείται. Όλα σχεδόν τα συστήματα προσφέρουν την πρώτη δυνατότητα. Έτσι προκύπτει ότι:

$$\begin{aligned} T_{\text{εξαν}} &= T_{\text{προσ}} + (n + o - 1) \times T_{\text{επομ}} \approx (n + o) \times T_{\text{επομ}} \\ &= \frac{(n + o \times Bfr) \times (r + btt)}{Bfr} \end{aligned}$$

7.2.7 Αναδιοργάνωση αρχείου

Η αναδιοργάνωση είναι μία διαδικασία που συνίσταται στη σειριακή ανάγνωση του κύριου αρχείου αγνοώντας τον κατάλογο και τις διαγραμμένες εγγραφές. Στη συνέχεια, το αρχείο επανα-αποθηκεύεται σε άλλη έκταση του δίσκου χωρίς περιοχές υπερχειλίσης. Στη φάση της νέας φόρτωσης δημιουργείται ο νέος κατάλογος με τη βοήθεια απομονωτικών μνημών. Κατόπιν ελευθερώνεται η έκταση που καταλαμβάνονταν από το παλιό αρχείο. Προκύπτει ότι το κόστος της αναδιοργάνωσης είναι:

$$T_{\text{αναδ}} = (n + o * Bfr) \times \frac{r + btt}{Bfr} + \frac{(n + o - d) \times R}{t'} + \frac{SI}{t'}$$

όπου d είναι ο αριθμός εγγραφών που έχουν μαρκαρισθεί ως διαγραμμένες, και SI είναι ο χώρος του καταλόγου. Προφανώς, ο πρώτος όρος δίνει το χρόνο ανάγνωσης του αρχείου, ενώ ο δεύτερος και ο τρίτος όρος δίνουν τον απαραίτητο χρόνο αποθήκευσης της κύριας περιοχής και του καταλόγου του αρχείου, αντίστοιχα.

Στη συνέχεια προχωρούμε σε μία εκτίμηση του χώρου που καταλαμβάνει ο κατάλογος, SI . Για κάθε σελίδα της κύριας περιοχής υπάρχει μία είσοδος στον κατάλογο ατράχτων. Άρα ο αριθμός αυτών των εισόδων είναι:

$$i_1 = \frac{n}{bfr}$$

Ο αριθμός των εισόδων που περιέχονται σε μία σελίδα του καταλόγου ατράχτων λέγεται λόγος διακλάδωσης (fanout ratio) ή παράγοντας διακλάδωσης (branching factor) και ισούται με:

$$y = \left\lfloor \frac{B}{K + P} \right\rfloor$$

όπου K είναι το μέγεθος του κλειδιού, και P είναι το μέγεθος του δείκτη σε bytes. Ο αριθμός των εισόδων στον κατάλογο κυλίνδρων που θα δεικτοδοτούν σελίδες του καταλόγου ατράχτων είναι:

$$i_2 = \frac{i_1}{y}$$

Θεωρητικά τα επίπεδα των καταλόγων είναι τόσα ώστε στο ανώτερο επίπεδο όλες οι εισοδοί να χωρούν σε μία σελίδα. Συνεπώς, ο αριθμός των σελίδων στο i -οστό επίπεδο ισούται με:

$$b_{i_i} = \left\lfloor \frac{i_i}{y} \right\rfloor = i_{i+1}$$

Άρα χρησιμοποιώντας την τιμή 1 για την ποσότητα b_{i_x} της σελίδας της ρίζας ισχύει:

$$SI = (b_{i_1} + b_{i_2} + \dots + 1) \times B = (i_2 + i_3 + \dots + 1) \times B$$

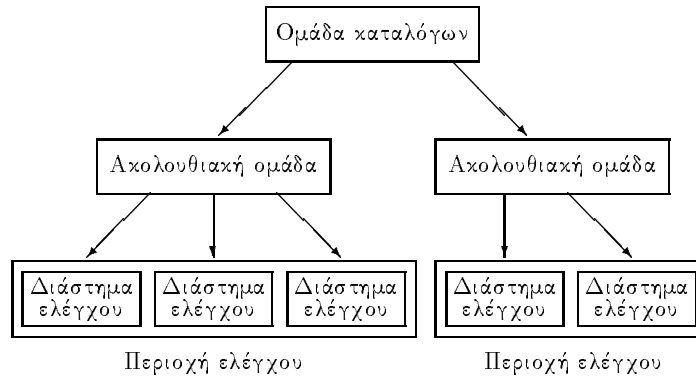
7.3 Μέθοδος των σελίδων καταλόγου και δεδομένων

Η μέθοδος των σελίδων καταλόγου και δεδομένων (index and data block) χρησιμοποιείται στα αρχεία VSAM της IBM και SIS της CDC. Οι δομές αυτές μοιάζουν με τα B⁺-δένδρα που θα αναπτυχθούν σε επόμενο κεφάλαιο. Στη συνέχεια θα παρουσιασθεί η μέθοδος VSAM που εμφανίστηκε το 1972 και εφαρμόστηκε σε μηχανές με υπερβατική μνήμη.

Η οργάνωση αυτή δεν στηρίζεται στα φυσικά χαρακτηριστικά του δίσκου (σε αντίθεση με την οργάνωση ISAM) και απαρτίζεται από τις εξής τέσσερις περιοχές:

- διαστήματα ελέγχου (control intervals), που περιέχουν τις εγγραφές με τα δεδομένα,
- περιοχές ελέγχου (control areas), που περιέχουν πολλά διαστήματα ελέγχου,
- ακολουθιακές ομάδες (sequence sets), που είναι κατάλογοι περιοχών ελέγχου, και
- ομάδα καταλόγων (index set), που αντιστοιχεί σε ένα διάστημα ελέγχου και είναι μία δενδρική δομή με το πολύ τρία επίπεδα. Οι σελίδες του κατωτέρου επιπέδου είναι κατάλογοι ακολουθιακών ομάδων.

Στο Σχήμα 7.4 παρουσιάζεται η δομή ενός VSAM αρχείου, ενώ το Σχήμα 7.5 παρουσιάζει τη δομή ενός διαστήματος ελέγχου. Κάθε διάστημα ελέγχου περιέχει στην αρχή μία ή περισσότερες εγγραφές, ελεύθερο χώρο για κατοπινές εισαγωγές εγγραφών, πεδία ορισμού εγγραφών (record definition fields, RDF) για κάθε εγγραφή του διαστήματος και ένα πεδίο ορισμού του διαστήματος ελέγχου (control interval definition fields, CIDF). Κάθε πεδίο ορισμού εγγραφών περιέχει τη σχετική διεύθυνση σε χαρακτήρες (relative byte address, RBA) της εγγραφής από την αρχή του αρχείου. Το



Σχήμα 7.4: Δομή αρχείου VSAM.

Εγγραφή 1	Εγγραφή 2	Εγγραφή 3	Εγγραφή 4
Εγγραφή 5	Εγγραφή 6	Ελεύθερος χώρος	
Ελεύθερος χώρος		RDF6	RDF5
RDF4	RDF3	RDF2	RDF1
			CIDF

Σχήμα 7.5: Δομή διαστήματος ελέγχου.

πεδίο ορισμού του διαστήματος ελέγχου περιέχει δύο τιμές: το μέγεθος και τη σχετική θέση του ελεύθερου χώρου του διαστήματος.

Κατά τη δημιουργία του αρχείου VSAM οι εγγραφές ταξινομούνται και αποθηκεύονται στα διαστήματα ελέγχου. Η πρώτη εγγραφή αποθηκεύεται στην αρχή, στην πρώτη θέση του πρώτου διαστήματος και το αντίστοιχο πεδίο ορισμού εγγραφών αποθηκεύεται στα αριστερά του πεδίου ορισμού διαστήματος ελέγχου. Στη συνέχεια η δεύτερη εγγραφή αποθηκεύεται δεξιά της πρώτης εγγραφής και το αντίστοιχο πεδίο ορισμού της εγγραφής στα αριστερά του πεδίου ορισμού της πρώτης εγγραφής. Έτσι, οι εγγραφές και τα αντίστοιχα πεδία ορισμού τους αποθηκεύονται στα διαστήματα ως δύο στοίβες, που μεγαλώνουν προς αντίθετες κατευθύνσεις. Σε περιπτώσεις εισαγωγών και διαγραφών, οι επεμβάσεις γίνονται στα δύο άκρα του διαστήματος, και συνεπώς έτσι διασφαλίζεται ότι ο ελεύθερος χώρος είναι συνεχής και δεν διασπάται.

Η περιοχή ελέγχου και τα διαστήματα ελέγχου είναι έννοιες παρόμοιες προς τις φυσικές έννοιες του κυλίνδρου και των ατράκτων. Το μεγαλύτερο κλειδί κάθε διαστήματος μίας περιοχής ελέγχου αποθηκεύεται στο κατώτερο επίπεδο της ομάδας καταλόγων. Τα διαστήματα ελέγχου μίας περιοχής δεν είναι αναγκαίο να είναι αποθηκευμένα φυσικά κατά αύξουσα τάξη των κλειδιών. Αυτό συμβαίνει επειδή η ακολουθιακή ομάδα έχει μία διατεταγμένη σειρά από τα μεγαλύτερα κλειδιά των διαστημάτων ελέγχου μαζί με τις αντίστοιχες διευθύνσεις στο δίσκο. Η ακολουθιακή ομάδα περιέχει επίσης και δείκτες για τη σειριακή ανάκτηση όλων των διαστημάτων με αύξουσα τάξη των κλειδιών.

Η ομάδα καταλόγων είναι μία δομή δένδρου που περιέχει σελίδες με ζεύγη κλειδιών-δεικτών. Κάθε ζεύγος αποτελείται από το μεγαλύτερο κλειδί της σελίδας του καταλόγου του κατώτερου επιπέδου και τον αντίστοιχο δείκτη. Η σειρά των ζευγών είναι κατά αύξουσα τάξη των κλειδιών, έτσι ώστε να διευκολύνεται η σειριακή αναζήτηση στο αρχείο. Ας σημειωθεί ότι εφαρμόζοντας τεχνικές συμπίεσης είναι δυνατόν τα κλειδιά στην ομάδα των καταλόγων να αποθηκευθούν σε μικρότερο χώρο και έτσι το μέγεθος του καταλόγου να ελαττωθεί.

Η υλοποίηση του VSAM αρχείου που διατηρεί όλες τις εγγραφές των διαστημάτων ελέγχου, καθώς και όλα τα ζεύγη όλων των επιπέδων της ομάδας των καταλόγων, ταξινομημένες κατά αύξουσα τάξη των κλειδιών πρέπει να φορτωθεί έτσι από την αρχή. Η δομή αυτή λέγεται αρχείο VSAM με ακολουθία κλειδιών (key-sequenced). Κατά την αρχική φόρτωση του αρχείου αφήνεται κατανεμημένος ελεύθερος χώρος (distributed free space, embedded space) για μελλοντικές εισαγωγές, είτε στο τέλος των διαστημάτων ελέγχου, είτε διατηρώντας κάποια διαστήματα απολύτως ελεύθερα στο τέλος κάθε περιοχής ελέγχου. Υπάρχει, λοιπόν, μία στενή αναλογία με την περιοχή υπερχείλισης των αρχείων ISAM.

Αναζήτηση σε αρχείο VSAM μπορεί να γίνει κατά τρεις τρόπους:

- Η σειριακή αναζήτηση αρχίζει με διάσχιση των επιπέδων του καταλόγου, ώστε να εντοπισθεί η σχετική διεύθυνση σε bytes του πρώτου διαστήματος ελέγχου του αρχείου. Στη συνέχεια ακολουθώντας τη συνδεδεμένη λίστα της ακολουθιακής ομάδας διατρέχεται ολόκληρο το αρχείο. Η σειριακή αναζήτηση δεν είναι απαραίτητο να αρχίσει από το πρώτο διάστημα ελέγχου, αλλά μπορεί να αρχίσει από κάποια συγκεκριμένη εγγραφή και να συνεχίσει ως το τέλος του αρχείου. Αυτού

του είδους η αναζήτηση γίνεται σε περίπτωση ερώτησης διαστήματος με προσδιορισμένη την κάτω τιμή.

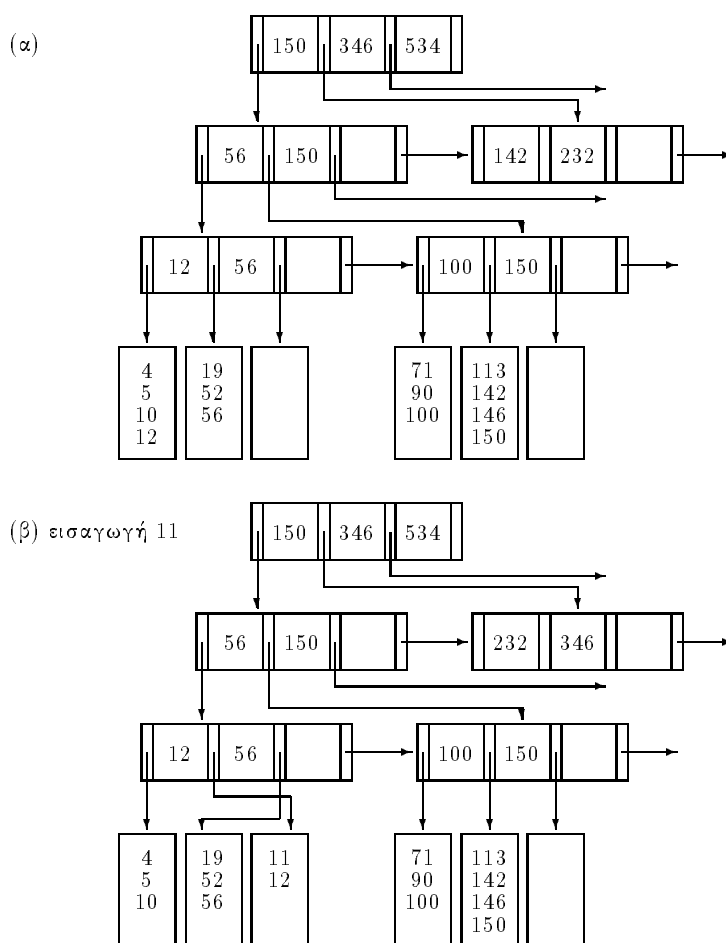
- Σειριακή αναζήτηση με άλμα εκτελείται όταν η ερώτηση διαστήματος έχει ορισμένες και τις δύο ακραίες τιμές.
- Τυχαία αναζήτηση γίνεται διασχίζοντας τη δενδρική δομή του καταλόγου μέχρι του σημείου να εντοπισθεί η σχετική διεύθυνση σε bytes της σελίδας που την περιέχει. Η διαδικασία αυτή είναι παρόμοια με τη διάσχιση ενός B⁺-δένδρου, όπως θα φανεί στο επόμενο κεφάλαιο.

Γενικά, πάντως, ο μηχανισμός προσπέλασης του αρχείου VSAM είναι πολύ αποτελεσματικός.

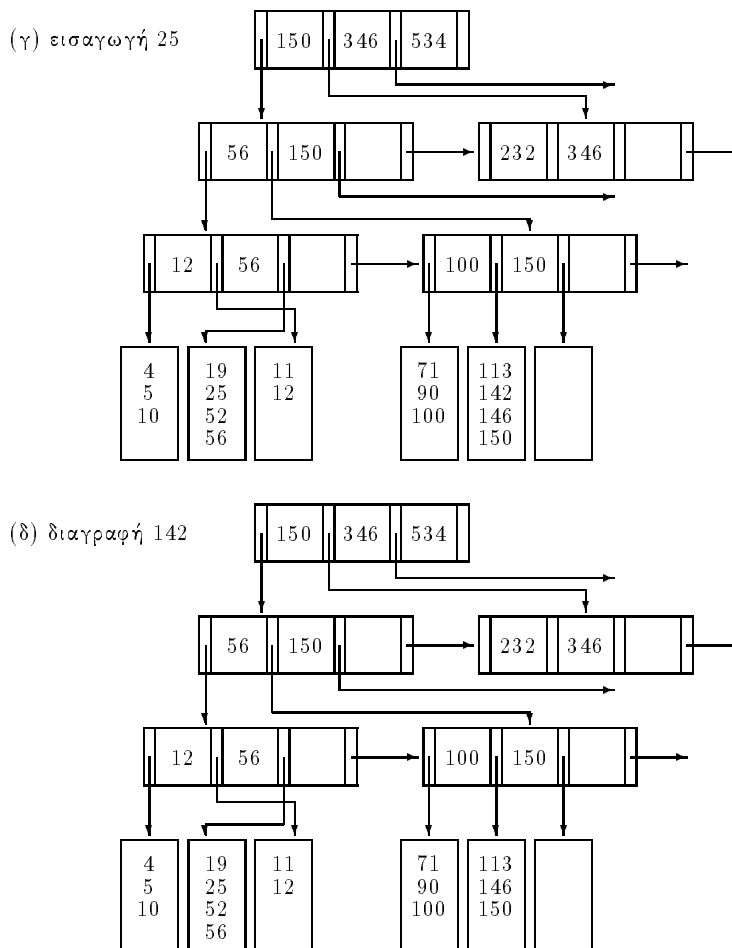
Κάθε νέα εγγραφή καταλαμβάνει μία θέση ανάλογα με την τιμή του κλειδιού της εξωθώντας τις υπόλοιπες. Το μέγιστο μέγεθος ενός διαστήματος ελέγχου είναι 32.768 bytes. Έτσι όταν λόγω μίας εισαγωγής γεμίσει ένα διάστημα ελέγχου, τότε συμβαίνει **διάσπαση** (splitting) του διαστήματος, οπότε οι μισές εγγραφές του μεταφέρονται σε ένα κενό διάστημα ελέγχου, που έχει δεσμευθεί στο τέλος της περιοχής ελέγχου. Με τον τρόπο αυτό οι εγγραφές δεν παραμένουν αποθηκευμένες σε αύξουσα τάξη. Εν τούτοις μπορεί να γίνει σειριακή ανάκτηση, επειδή η ομάδα των καταλόγων ενημερώνεται σχετικά. Αν μετά από διαδοχικές εισαγωγές γεμίσουν όλα τα διαστήματα μίας περιοχής ελέγχου, τότε η συγκεκριμένη περιοχή διασπάται και δημιουργείται μία νέα περιοχή στο τέλος του αρχείου, όπου μεταφέρονται τα μισά διαστήματα ελέγχου. Και πάλι η ομάδα των καταλόγων ενημερώνεται σχετικά. Η αυτόματη αυτή επέκταση ενός αρχείου VSAM μπορεί να φθάσει μέχρι το μέγιστο μέγεθος των 4,29 GB. Αυτή η διαδικασία είναι δαπανηρή και γι' αυτό ο σχεδιαστής του αρχείου πρέπει να είναι πολύ προσεκτικός στις τιμές που θα δώσει στις παραμέτρους του αρχείου.

Σε περίπτωση διαγραφής, ο χώρος ελευθερώνεται (σε αντίθεση με τα αρχεία ISAM) και διατίθεται προς χρήση. Οι εγγραφές του διαστήματος ελέγχου με κλειδιά μεγαλύτερα από αυτό που διαγράφεται ολισθαίνουν κατά μία θέση, ώστε ο διατιθέμενος χώρος να ενωθεί με τον ελεύθερο χώρο του διαστήματος. Αυτή η διαδικασία λέγεται **δυναμική ανάκτηση χώρου** (dynamic space reclamation). Αν κατά την ενημέρωση μίας εγγραφής το μήκος της δεν αλλάξει, τότε δεν υπάρχει πρόβλημα, αν όμως το μήκος της ελαττωθεί ή αυξηθεί, τότε το σύστημα λειτουργεί σαν να πρόκειται για διαγραφή ή εισαγωγή, αντίστοιχα.

Όλες οι έννοιες και οι διαδικασίες που αναπτύχθηκαν θα επεξηγηθούν με τη βοήθεια ενός παραδείγματος. Έστω ότι στο δείγμα αρχείου VSAM του Σχήματος 7.6α εισάγεται η εγγραφή με κλειδί 11. Η διαδικασία αρχίζει με διάσχιση της ομάδας των καταλόγων με στόχο το αμέσως μεγαλύτερο κλειδί από το 11. Το κλειδί αυτό είναι το 150 που συνοδεύεται από ένα δείκτη προς το κατώτερο επίπεδο του καταλόγου. Νέα σάρωση για το αμέσως μεγαλύτερο κλειδί από το 11 καταλήγει στον εντοπισμό του κλειδιού 56 με τον αντίστοιχο δείκτη προς την ακολουθιακή ομάδα.



Σχήμα 7.6: Εισαγωγές και διαγραφές σε αρχείο VSAM.



Σχήμα 7.6: Εισαγωγές και διαγραφές σε αρχείο VSAM (συνέχεια).

Και πάλι η σάρωση της ακολουθιακής ομάδας για το αμέσως μεγαλύτερο κλειδί από το 11 καταλήγει στο κλειδί 12 και τον αντίστοιχο δείκτη προς το κατάλληλο διάστημα ελέγχου, όπου δεν υπάρχει διαθέσιμος χώρος για την εγγραφή 11. Συνεπώς στην περίπτωση αυτή το διάστημα διασπάται. Οι εγγραφές 4, 5 και 10 παραμένουν στο ίδιο διάστημα ελέγχου, ενώ οι εγγραφές 11 και 12 αποθηκεύονται σε νέο διάστημα ελέγχου, που βρίσκεται στο τέλος της περιοχής ελέγχου. Η νέα μορφή του αρχείου φαίνεται στο Σχήμα 7.6β. Σχετικά έχει ενημερωθεί και η ακολουθιακή ομάδα.

Έστω ότι εισάγεται η εγγραφή 25. Αναζήτηση διά μέσου της ομάδας των καταλόγων καταλήγει στην ίδια περιοχή ελέγχου. Το διάστημα ελέγχου, όπου η εγγραφή 25 πρέπει να αποθηκευθεί, έχει διαθέσιμο χώρο. Στην περίπτωση αυτή οι εγγραφές 52 και 56 ολισθαίνουν μία θέση προς τα πίσω, ώστε η εγγραφή 25 να καταλάβει το δημιουργούμενο κενό χώρο χωρίς να διαταραχθεί η λογική σειρά των εγγραφών. Δεν χρειάζονται αλλαγές στην ακολουθιακή ομάδα. Η νέα μορφή του αρχείου φαίνεται στο Σχήμα 7.6γ.

Έστω ότι πρέπει να διαγραφεί η εγγραφή 142 του Σχήματος 7.6γ. Με την απομάκρυνση της εγγραφής 142, οι εγγραφές 146 και 150 ολισθαίνουν μία θέση προς τα εμπρός, ώστε να ενοποιηθεί ο ελεύθερος χώρος του διαστήματος (όπως φαίνεται στο Σχήμα 7.6δ). Αν στη συνέχεια διαγραφεί το κλειδί 150, τότε πρέπει να ενημερωθεί και η ακολουθιακή ομάδα και η ομάδα των καταλόγων, ώστε να καταγραφεί το γεγονός ότι πλέον το μεγαλύτερο κλειδί του διαστήματος είναι το 146.

Το αρχείο VSAM υποστηρίζει και δύο άλλες υλοποιήσεις εκτός από την υλοποίηση με ακολουθία κλειδιών.

- Η υλοποίηση του αρχείου με ακολουθία εισόδων (entry-sequenced file) μοιάζει με τα σειριακά αρχεία, επειδή κάθε νέα εγγραφή παρατίθεται στο τέλος του αρχείου. Το σύστημα δεν κτίζει αυτόματα την ομάδα καταλόγου, αλλά είναι στη διάθεση του προγραμματιστή να κτίσει έναν κατάλογο χρησιμοποιώντας τη σχετική διεύθυνση σε bytes που επιστρέφεται από το σύστημα και παραμένει σταθερή. Προφανώς, η εκδοχή αυτή του αρχείου VSAM είναι χρήσιμη σε περιπτώσεις, όπου δεν είναι απαραίτητη η ταξινόμηση των εγγραφών αλλά αντίθετα είναι απαραίτητη η αποθήκευση των εγγραφών με τη σειρά άφιξής τους (όπως για παράδειγμα σε αρχεία συναλλαγών).
- Η υλοποίηση του αρχείου σχετικών εγγραφών (relative record file) δεν έχει κατάλογο και μπορεί να διαχειρισθεί μόνο εγγραφές σταθερού μήκους. Το αρχείο αυτό μπορεί να αποθηκεύσει μέχρις ένα συγκεκριμένο αριθμό εγγραφών, οπότε δεσμεύεται ανάλογος χώρος. Η εκδοχή αυτή στηρίζεται στον κατακερματισμό και έχει αρκετές ομοιότητες με τα τυχαία αρχεία που θα εξετασθούν σε επόμενο κεφάλαιο.

Είναι προφανές ότι οι δύο αυτές εκδοχές δεν διακρίνονται από δυναμική ανάκτηση χώρου.

Συμπερασματικά, τα αρχεία VSAM είναι πιο αποτελεσματικά από τα αρχεία ISAM για τους εξής βασικούς λόγους:

- δεν κάνουν διάκριση μεταξύ κύριας περιοχής και περιοχής υπερχειλίσης,
- συνδυάζουν αυτόματα το χώρο που προκύπτει από τις διαγραφές με τον ελεύθερο χώρο του διαστήματος,
- εκτελούν αυτόματα τις απαραίτητες διασπάσεις διαστημάτων και περιοχών ελέγχου καθώς και τη δυναμική ανάκτηση χώρου, και
- διαχειρίζονται εγγραφές μεταβλητού μήκους.

7.4 Ασκήσεις

<1> Αρχείο ISAM αποθηκεύεται σε συσκευές μαγνητικών δίσκων με τα εξής χαρακτηριστικά: 200 κύλινδροι, 19 άτρακτοι/κύλινδρο, 14000 bytes/άτρακτο, 2000 bytes/σελίδα. Το αρχείο αποτελείται από 1.000.000 εγγραφές, όπου το μέγεθος του κλειδιού είναι 14 bytes, ενώ του δείκτη είναι 4 bytes. Πόσες συσκευές δίσκων χρειάζονται;

<2> Δίνεται αρχείο ISAM με 10.000 εγγραφές των 160 bytes. Το μήκος του κλειδιού είναι 16 bytes και του δείκτη 4 bytes. Το αρχείο είναι αποθηκευμένο σε δίσκο HP7925 με χαρακτηριστικά: 256 bytes/σελίδα, 64 σελίδες/άτρακτο, 9 άτρακτοι/κύλινδρο, ενώ η συσκευή αποτελείται από 815 κυλίνδρους. Να βρεθεί ο βέλτιστος παράγοντας ομαδοποίησης, ώστε να ελαχιστοποιηθεί το βάθος του καταλόγου. Πόσο χώρο σε κυλίνδρους καταλαμβάνει συνολικά το αρχείο και ποιός είναι ο συνολικός παράγοντας χρησιμοποίησης;

<3> Η προηγούμενη άσκηση να λυθεί και πάλι για τα εξής δεδομένα: 1000 bytes/σελίδα, 24 σελίδες/άτρακτο και 12 άτρακτοι/κύλινδρο.

<4> Δίνεται αρχείο VSAM με χαρακτηριστικά ίδια με το αρχείο της Άσκησης 2 που είναι αποθηκευμένο σε δίσκο HP7925. Να βρεθεί το βάθος του καταλόγου, ο χώρος σε σελίδες που καταλαμβάνει συνολικά το αρχείο και ο συνολικός παράγοντας χρησιμοποίησης; Τι θα συμβεί στο βάθος του καταλόγου αν το αρχείο αποτελείται από 100.000 εγγραφές; Να εξετασθεί η συνέπεια της αύξησης του μεγέθους της σελίδας.

<5> Η προηγούμενη άσκηση να λυθεί και πάλι για τα εξής δεδομένα: 1000 bytes/σελίδα, 24 σελίδες/άτρακτο και 12 άτρακτοι/κύλινδρο.

<6> Δίνεται αρχείο ISAM με τα εξής χαρακτηριστικά: μήκος εγγραφής 100 bytes, μέγεθος σελίδας 1000 bytes, μέγεθος κλειδιού 38 bytes, μέγεθος δείκτη 4 bytes, ενώ ο αριθμός των εγγραφών είναι 100.000. Το αρχείο είναι αποθηκευμένο σε HP7925. Πόσα επίπεδα καταλόγου απαιτούνται; Να υπολογισθεί ο απαιτούμενος χρόνος:

- για μία δυαδική αναζήτηση μόνο στα δεδομένα του αρχείου, και
- για μία αναζήτηση στο αρχείο ως σύνολο.

<7> Αν δεν επιτρέπεται οι εγγραφές να αποθηκεύονται σε 2 σελίδες, τότε να βρεθούν αναλυτικές εκφράσεις (ως συνάρτηση των παραμέτρων n , R , K και B) για:

- το χώρο που το αρχείο καταλαμβάνει σε σελίδες,
- πόσες σελίδες καταλαμβάνει το πρώτο επίπεδο ενός στατικού καταλόγου, αν ο δείκτης χρειάζεται 2 bytes,
- πόσα επίπεδα καταλόγου απαιτούνται;
- πόσος είναι ο συνολικός χώρος που απαιτείται από όλα τα επίπεδα του καταλόγου;

<8> Είναι δυνατόν ο κατάλογος να είναι αποθηκευμένος σε διαφορετική συσκευή από το κυρίως αρχείο. Έστω ότι ένα αρχείο αποτελείται από 10.000 εγγραφές των 80 bytes με κλειδί των 8 bytes. Το μέγεθος της σελίδας είναι 1024 bytes, ενώ το μέγεθος του δείκτη είναι 2 bytes. Χρησιμοποιώντας τους τύπους της προηγούμενης άσκησης να βρεθεί ο αριθμός των επιπέδων του καταλόγου και ο συνολικός χώρος (σε bytes) που απαιτείται για την αποθήκευση του καταλόγου, αν το μέγεθος της σελίδας μεταβάλλεται από 256 bytes ως 3072 bytes με βήματα των 256 bytes.

<9> Σειριακό αρχείο με δείκτη έχει τρία επίπεδα καταλόγων. Από το διαχειριστή του συστήματος αποφασίζεται ότι αναδιοργάνωση πρέπει να γίνεται όταν η περιοχή υπερχείλισης, που έχει μέγεθος 20% της κύριας περιοχής, είναι πλήρης κατά 80%. Να υπολογισθεί ο μέσος χρόνος προσπέλασης μίας εγγραφής θεωρώντας ότι $Bfr=10$.