

ΕΝΟΤΗΤΑ 7

ΔΙΑΧΕΙΡΙΣΗ ΣΥΣΚΕΥΩΝ Ε/Ε & ΑΠΟΘΗΚΕΥΣΗΣ

Περιεχόμενα

1. Συσκευές εισόδου/εξόδου
2. Απευθείας προσπέλαση μνήμης
3. Ρόλος του Λειτουργικού Συστήματος στη διαχείριση των συσκευών Ε/Ε
4. Διαχείριση δίσκου
5. Άλλα θέματα διαχείρισης δίσκου

1. Συσκευές εισόδου/εξόδου

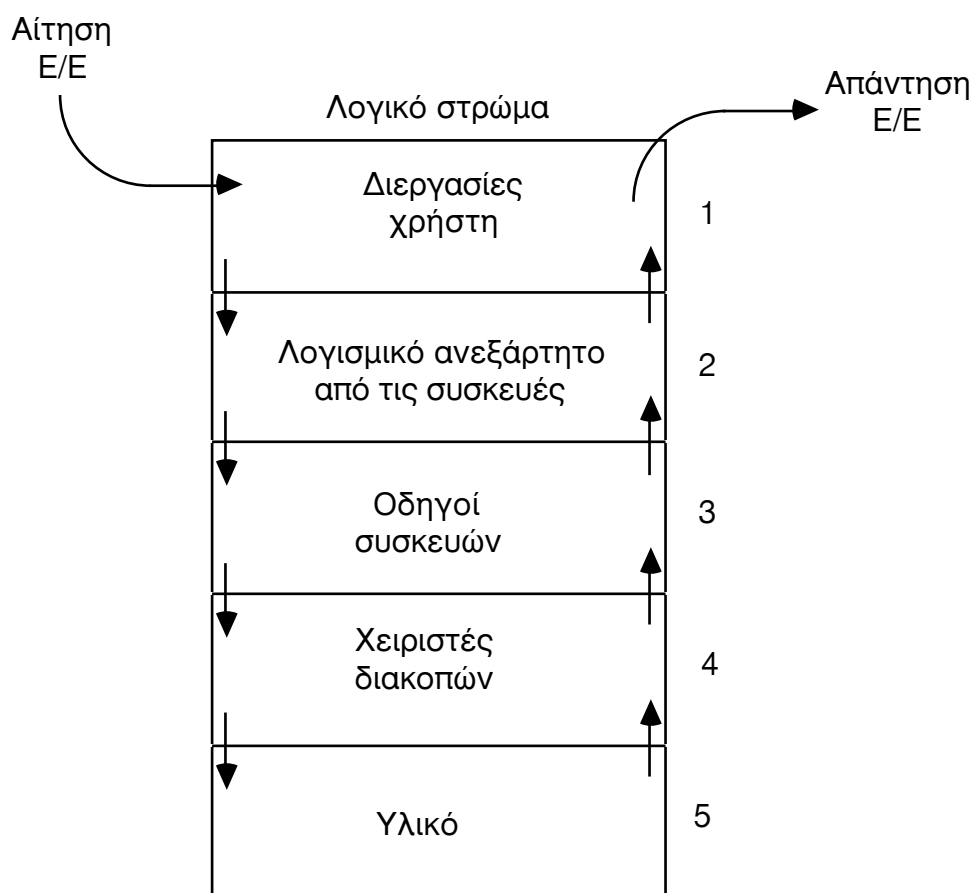
- Οι συσκευές εισόδου/εξόδου που είναι προσαρτισμένες σε ένα σύστημα Η/Υ χωρίζονται σε τρεις κατηγορίες:
 - Αυτές των οποίων τα δεδομένα που διαχειρίζονται είναι σε αναγνώσιμη μορφή από τον άνθρωπο (οθόνες, ποντίκια, πληκτρολόγια).
 - Αυτές των οποίων τα δεδομένα που διαχειρίζονται είναι σε αναγνώσιμη μορφή από τη μηχανή (ελεγκτές, ανιχνευτές).
 - Αυτές που χρησιμοποιούνται κυρίως για επικοινωνίες (μετατροπείς).
- Ανάλογα με το είδος τους διαφέρουν στα ακόλουθα:
 - Ρυθμός μεταφοράς δεδομένων (data rate).
 - Εφαρμογές.
 - Πολυπλοκότητα του ελέγχου τους.
 - Μονάδα μεταφοράς δεδομένων: μπλοκ σταθερού μεγέθους (δίσκος, ταινία) ή ακολουθία (stream) χαρακτήρων (τερματικό, ποντίκι).

2. Απευθείας προσπέλαση μνήμης

- Συνήθως η μεταφορά δεδομένων προς ή από την κύρια μνήμη χρειάζεται τη χρήση του επεξεργαστή. Μία τεχνική που λέγεται *απευθείας προσπέλαση μνήμης* (direct memory access) επιτρέπει την απευθείας μεταφορά των δεδομένων προς/από την κύρια μνήμη χωρίς να χρειασθεί να απασχοληθεί ο επεξεργαστής. Η υπευθυνότητα για τη μεταφορά δίνεται στην ίδια τη συσκευή Ε/Ε που αναβαθμίζεται με προσωρινή μνήμη (buffering) και λογισμικό. Καθίσταται έτσι ένα είδος επεξεργαστού ειδικευμένου για είσοδο και έξοδο.
- Ακόμα και έτσι για τη μεταφορά των δεδομένων χρησιμοποιείται ο δίαυλος δεδομένων (data bus): στη διάρκεια χρήσης του από κάποια συσκευή δεν είναι διαθέσιμος στην ΚΜΕ η οποία πρέπει να αναστείλει τη λειτουργία της για ένα κύκλο μηχανής (cycle stealing). Μία λύση είναι η χρησιμοποίηση ειδικών διαύλων εισόδου-εξόδου (I/O channels).

3. Ρόλος του Λ.Σ. στη διαχείριση των συσκευών Ε/Ε

- Ο ρόλος του Λ.Σ. είναι διπλός:
 - Παροχή εύκολης διαχείρισης των συσκευών που να είναι ανεξάρτητη από τις ιδιαιτερότητες της καθεμιάς τους (device independence), π.χ. αν είναι αφοσιωμένες (dedicated) όπως ο εκτυπωτής ή διαμοιραζόμενες ταυτόχρονα ανάμεσα σε πολλές διεργασίες όπως ο δίσκος.
 - Μεγιστοποίηση της απόδοσης των συσκευών (που από φύση τους είναι αργές).
- Ο πρώτος στόχος μπορεί να επιτευχθεί με τη χρήση λογικών στρωμάτων και αρθρωτού σχεδιασμού όπου οι χαμηλού επιπέδου λεπτομέρειες που είναι διαφορετικές για κάθε συσκευή απομονώνονται στα χαμηλά στρώματα, ενώ τα υψηλότερα παρέχουν γενικές και ομοιογενείς λογικές λειτουργίες όπως άνοιγμα/κλείσιμο συσκευής, διάβασμα/γράψιμο, κλπ.



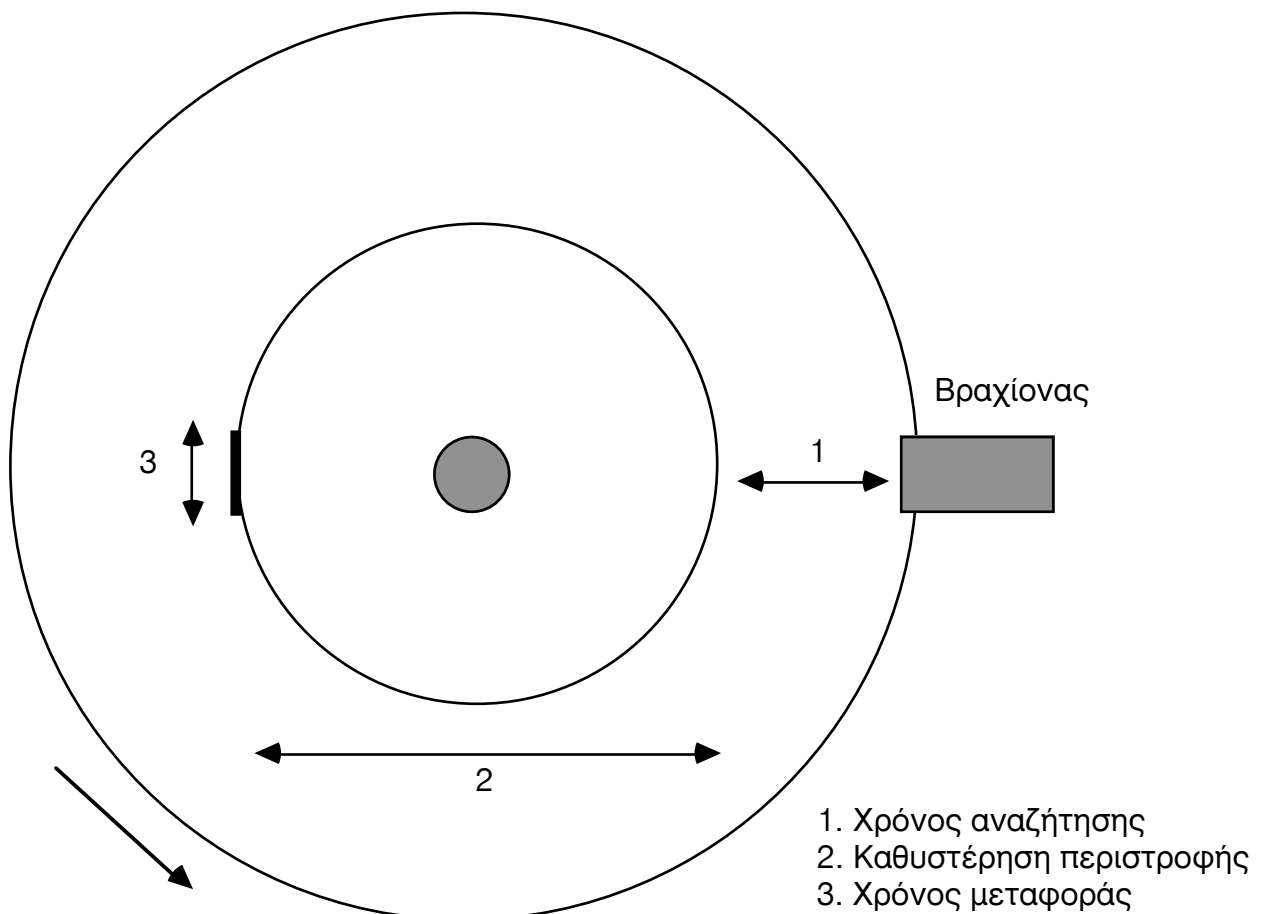
3. Ρόλος του Λ.Σ. στη διαχείριση των συσκευών Ε/Ε (συνέχεια)

- Τα ανωτέρω 5 επίπεδα υλοποιούν τις εξής λειτουργίες:
 1. Δημιουργία κλήσης, μορφοποίηση, ετεροχρονισμός Ε/Ε.
 2. Ονοματολογία (π.χ. /dev/tty0 στο Unix), προστασία, αναστολή, αποθήκευση σε ενδιάμεση μνήμη.
 3. Αρχικοποίηση καταχωρητών συσκευής, έλεγχος κατάστασης.
 4. Αφύπνιση οδηγού όταν ολοκληρωθεί η λειτουργία Ε/Ε.
 5. Εκτέλεση της λειτουργίας Ε/Ε.

- Ο δεύτερος στόχος μπορεί να επιτευχθεί με τη χρήση προσωρινής μνήμης:
 - μονής (single buffering) και για είσοδο και για έξοδο,
 - διπλής (double buffering), μία για είσοδο και μία για έξοδο,
 - πολλαπλής σε κυκλική σειρά (circular buffering) για ακόμα καλύτερη αποδοτικότητα (μόλις γεμίσει η μία χρησιμοποιείται η επόμενη).

4. Διαχείριση δίσκου

- Ακόμα και η πιο γρήγορη περιφερειακή συσκευή αποθήκευσης πληροφοριών, ο (σκληρός) δίσκος, είναι 4 φορές πιο αργή από την ΚΜΕ και την κύρια μνήμη. Ο χρόνος ανάγνωσης ή εγγραφής ενός μπλοκ δίσκου καθορίζεται από 3 παράγοντες:
 - τον *χρόνο αναζήτησης* (seek time), δηλαδή το χρονικό διάστημα που χρειάζεται η κεφαλή ανάγνωσης/εγγραφής να τοποθετηθεί πάνω από τον κατάλληλο δίαυλο εγγραφής (track) του δίσκου·
 - την *καθυστέρηση περιστροφής* (rotational time, latency time), δηλαδή το χρονικό διάστημα που χρειάζεται να φτάσει ο κατάλληλος τομέας κάτω από την κεφαλή·
 - τον πραγματικό χρόνο μεταφοράς (transfer time).

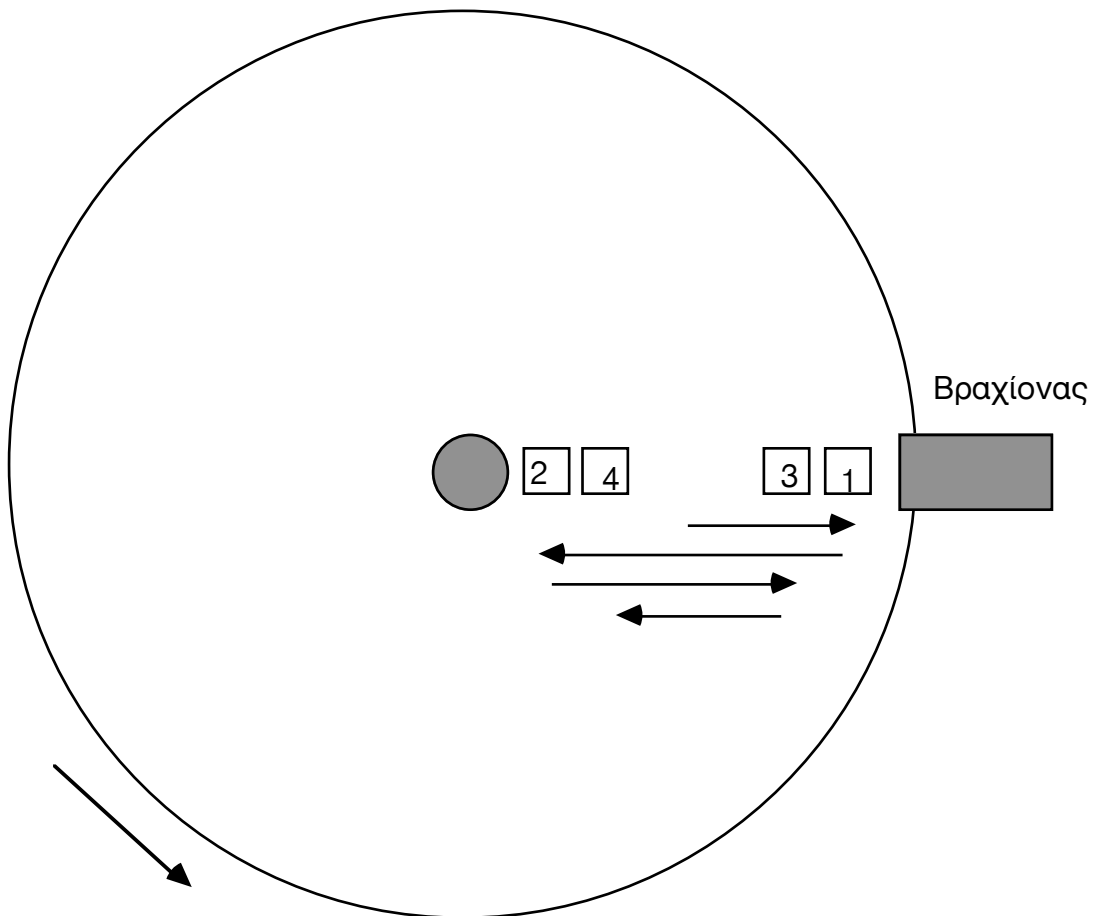


4. Διαχείριση δίσκου (συνέχεια)

- Χρόνος Περιστροφής $TT = b/(r*N)$ Χρόνος Αναζήτησης $T_s = m * n + s$
Μέσος Χρόνος Αναζήτησης $T_a = T_s + 1/(2*r) + TT$ [N=bytes ανά δίαυλο, b=bytes για μεταφορά, r=ταχύτητα περιστροφής δίσκου, m=σταθερά δίσκου, n=αριθμός διαύλων που θα διατρεχθούν, s=καθυστέρηση έναρξης περιστρ.]
- Το άθροισμα του χρόνου αναζήτησης και καθυστέρησης περιστροφής αποτελεί το *χρόνο προσπέλασης* (access time), δηλαδή το χρονικό διάστημα που χρειάζεται η κεφαλή για να φτάσει στο σημείο του δίσκου από το οποίο θα διαβάσει ή στο οποίο θα γράψει.
- Η παράμετρος εκείνη που καθορίζει το ρυθμό απόδοσης του δίσκου είναι ο χρόνος αναζήτησης. Με τον όρο διαχείριση δίσκου επομένως εννοούμε κυρίως την προσπάθεια ελαχιστοποίησης των κινήσεων του βραχίονα του δίσκου ώστε να τοποθετηθεί πάνω από τον κατάλληλο δίαυλο εγγραφής (track), με άλλα λόγια την ελαχιστοποίηση του χρόνου αναζήτησης.
- Υπάρχουν οι ακόλουθες πολιτικές *χρονοδρομολόγησης δίσκου* (disk scheduling) που χωρίζονται σε δύο κατηγορίες ανάλογα με το αν η επιλογή εξυπηρέτησης της εντολής για Ε/Ε γίνεται με βάση τα χαρακτηριστικά της διεργασίας που εκτέλεσε την εντολή ή την περιοχή του δίσκου στην οποία ζητείται προσπέλαση.
- Η μέθοδος της *τυχαίας χρονοδρομολόγησης* (random scheduling) απλά διαλέγει κάποια από τις αιτήσεις για Ε/Ε που εκκρεμούν. Μια και η συμπεριφορά του βραχίονα θα είναι και αυτή τυχαία τα αποτελέσματα θα είναι τα χειρότερα δυνατά. Η μέθοδος αυτή χρησιμοποιείται μόνο σαν σημείο σύγκρισης και εκτίμησης των αποτελεσμάτων που έχουν οι υπόλοιπες μέθοδοι.
- Η μέθοδος των *προτεραιοτήτων* εξυπηρετεί την αίτηση Ε/Ε που ανήκει στη διεργασία με την υψηλότερη προτεραιότητα. Σκοπός της είναι κυρίως η βελτιστοποίηση άλλων παραμέτρων του Λ.Σ. και όχι της χρήσης του δίσκου. Είναι κατάλληλη μόνο για μικρές εργασίες σε δέσμες ή διαλογικές (οι οποίες συνήθως έχουν υψηλή προτεραιότητα). Έχει άσχημα αποτελέσματα σε συστήματα βάσεων δεδομένων.

4. Διαχείριση δίσκου (συνέχεια)

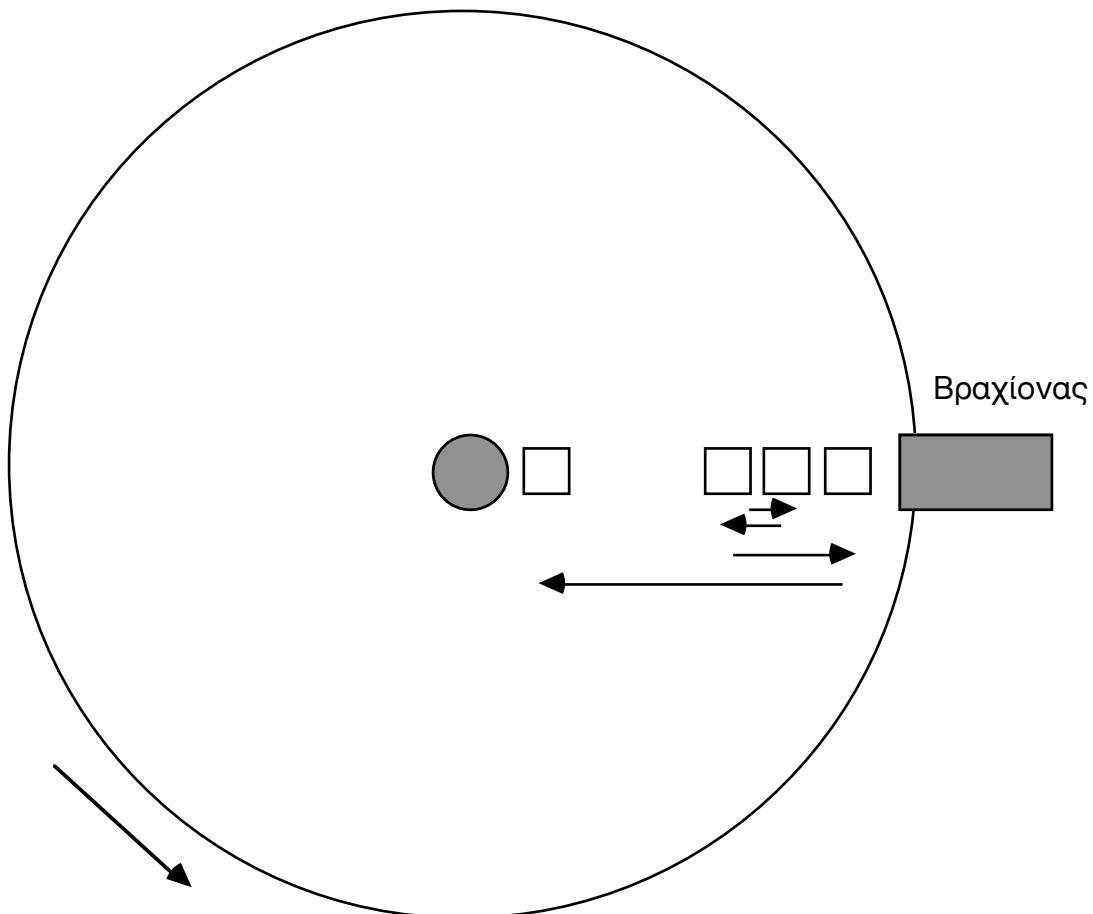
- Η μέθοδος της *Πρώτη Εισερχόμενη - Πρώτη Εξυπηρετούμενη* (first come first served) έχει το πλεονέκτημα ότι είναι δίκαιη προς όλες τις διεργασίες αλλά έχει καλά αποτελέσματα μόνο αν οι αιτήσεις είναι λίγες και αναφέρονται σε χώρους του δίσκου που γειτονεύουν. Στη γενική περίπτωση έχει τα ίδια αποτελέσματα με της τυχαίας επιλογής.



- Μία παραλλαγή της είναι της *Τελευταία Εισερχόμενη - Πρώτη Εξυπηρετούμενη* (last come first served) που βασίζεται στο ότι υπάρχουν πολλές πιθανότητες οι πιο πρόσφατες αιτήσεις να ανήκουν στον ίδιο χρήστη που μπορεί να θέλει να προσπελάσει τον ίδιο χώρο μνήμης (π.χ. σειριακό αρχείο). Έχει καλά αποτελέσματα μόνο για τέτοια σενάρια.

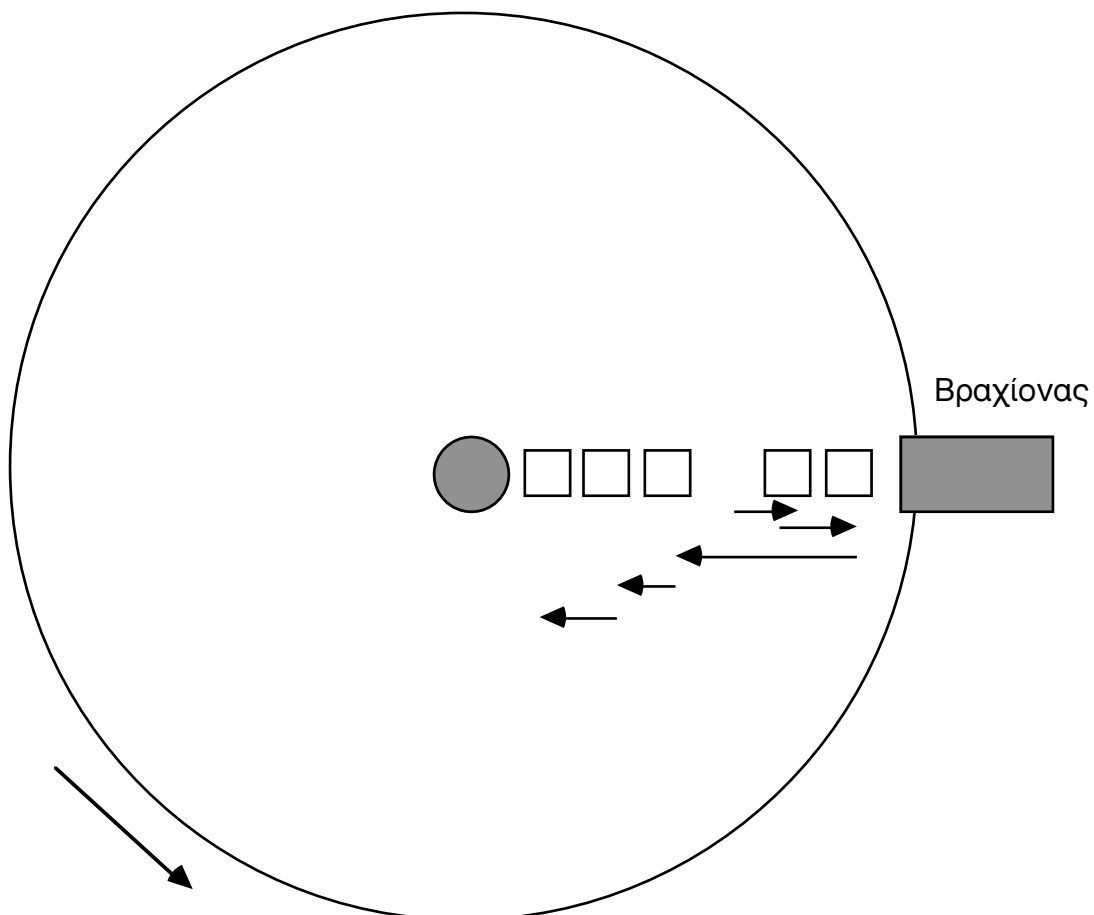
4. Διαχείριση δίσκου (συνέχεια)

- Η μέθοδος της *Συντομότερη Αναζήτηση Πρώτη* (shortest service time first) εξυπηρετεί την πιο κοντινή αίτηση. Ελαχιστοποιεί τον χρόνο αναζήτησης από αίτηση σε αίτηση αλλά όχι κατ' ανάγκη και συνολικά (δηλαδή το μέσο χρόνο αναζήτησης). Στην περίπτωση που υπάρχουν δύο αιτήσεις ίσης απόστασης (μπροστά και πίσω) διαλέγεται τυχαία μία από αυτές. Τείνει να εξυπηρετεί καλύτερα αιτήσεις που αναφέρονται στις μεσαίες περιοχές μεταξύ των εσωτερικών και εξωτερικών διαύλων εγγραφής. Είναι κατά κάποιο τρόπο το αντίστοιχο του αλγόριθμου SPN στη χρονοδρομολόγηση διεργασιών και μπορεί και αυτή να προκαλέσει παρατεταμένη στέρηση μερικών αιτήσεων Ε/Ε αν υπάρχουν συνεχώς καινούργιες αιτήσεις οι οποίες προτιμούνται έναντι των παλαιότερων.



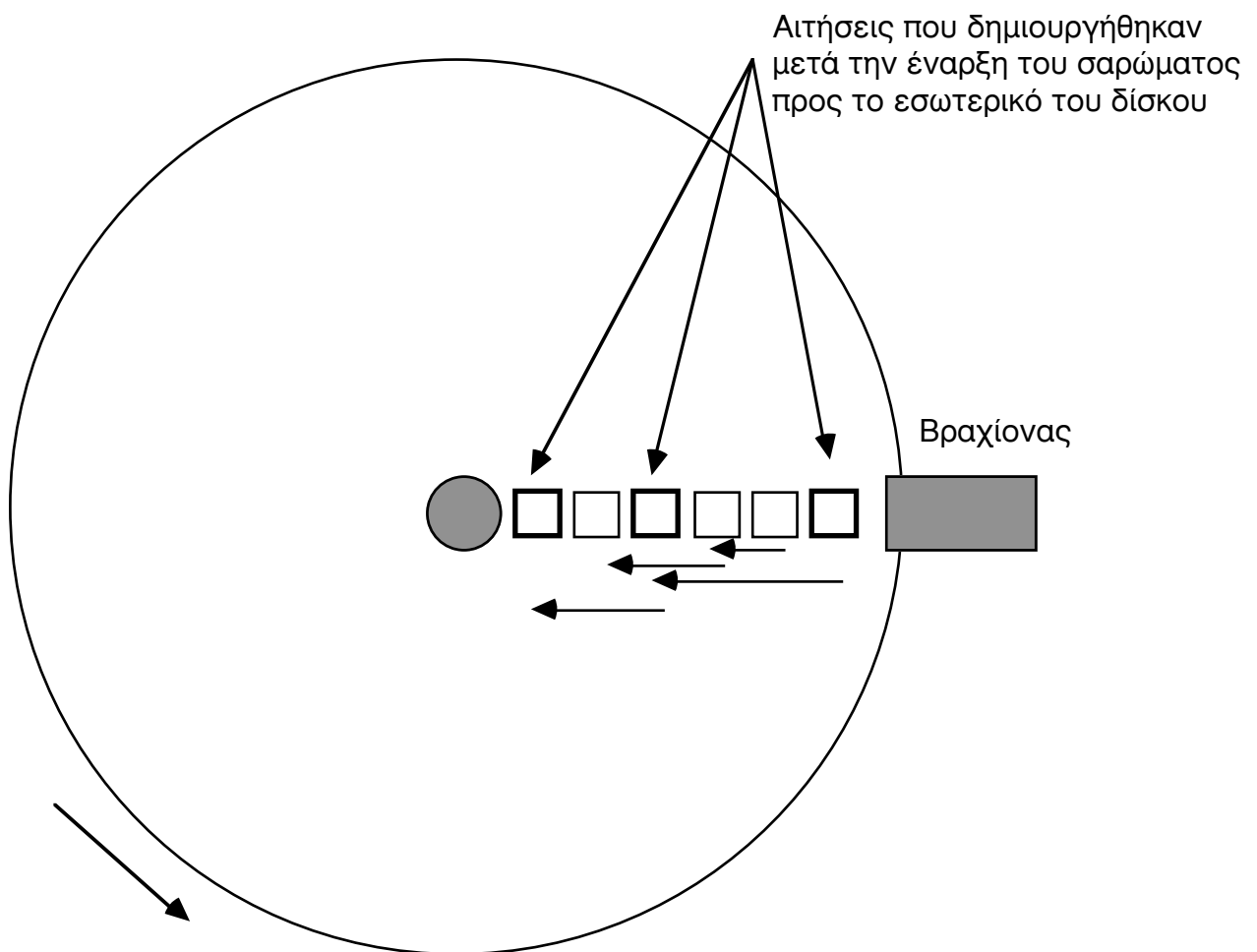
4. Διαχείριση δίσκου (συνέχεια)

- Η μέθοδος της *Σάρωσης Προς την Ίδια Κατεύθυνση (SCAN)* διαλέγει την πιο κοντινή αίτηση που βρίσκεται στην κατεύθυνση προς την οποία σαρώνει (προς το κέντρο ή την περιφέρεια του δίσκου). Μόνο όταν εξυπηρετηθούν όλες οι αιτήσεις προς την ίδια κατεύθυνση, αντιστρέφεται η φορά της σάρωσης. Η μέθοδος αυτή έχει σχεδόν τα ίδια αποτελέσματα με την προηγούμενη χωρίς όμως να δημιουργεί προβλήματα παρατεταμένης στέρησης. Τείνει όμως και αυτή να εξυπηρετεί καλύτερα αιτήσεις που αναφέρονται στις μεσαίες περιοχές μεταξύ των εσωτερικών και εξωτερικών διαύλων εγγραφής καθώς επίσης και τις αιτήσεις που δημιουργήθηκαν πρόσφατα. Η μέθοδος αυτή λέγεται και *αλγόριθμος του ανεγκυστήρα*.



4. Διαχείριση δίσκου (συνέχεια)

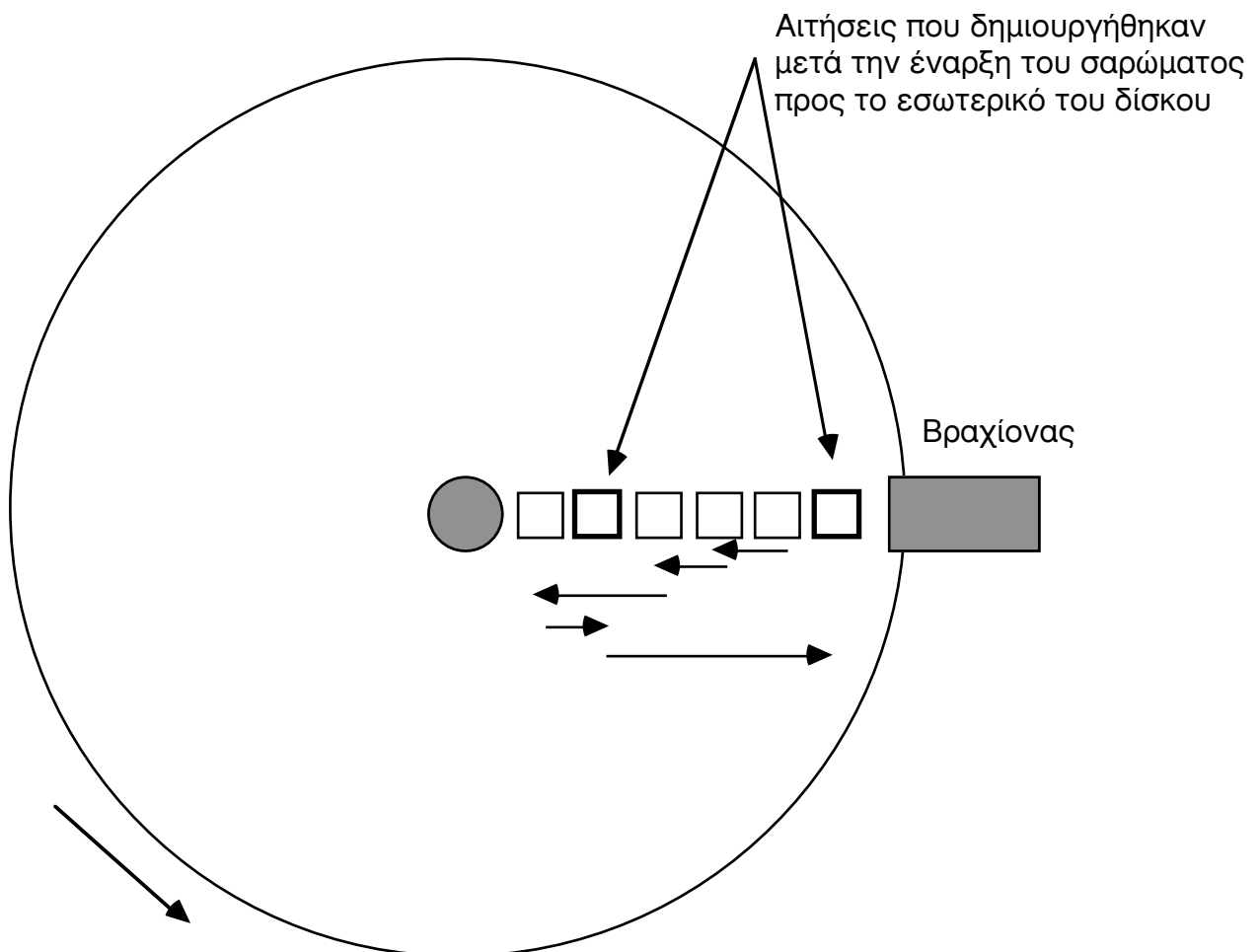
- Η μέθοδος της *Κυκλικής Σάρωσης (C-SCAN)* ουσιαστικά σαρώνει προς μία κατεύθυνση μόνο, συνήθως από έξω προς τα μέσα. Μόλις ο βραχίονας ολοκληρώσει το σάρωμα μετατοπίζεται γρήγορα στον εξωτερικό δίαυλο και επαναρχίζει το σάρωμα. Η μέθοδος αυτή επιλύει το πρόβλημα της μη ικανοποιητικής εξυπηρέτησης των αιτήσεων που αναφέρονται στα άκρα (εσωτερικά και εξωτερικά) του δίσκου.



- Σημειωτέον, ότι θεωρητικά οι αλγόριθμοι SCAN και C-SCAN μετακινούν το βραχίονα μέχρι το τέλος της εσωτερικής ή εξωτερικής επιφάνειας του δίσκου, ανεξαρτήτως από το αν από κάποιο σημείο και μετά υπάρχει κάποια αίτηση για εξυπηρέτηση. Στην πράξη όμως, η μετακίνηση του βραχίονα προς μία κατεύθυνση τερματίζεται στην τελευταία αίτηση προς εκείνη την κατεύθυνση. Αυτές οι παραλλαγές ονομάζονται LOOK και C-LOOK.

4. Διαχείριση δίσκου (συνέχεια)

- Η μέθοδος της Σάρωσης N βημάτων (N-step-SCAN) κάνει χρήση του απλού αλγόριθμου της σάρωσης αλλά για μία ομάδα μόνο από αιτήσεις μεγέθους N . Κατά τη διάρκεια της σάρωσης οι υπόλοιπες αιτήσεις που καταφθάνουν μπαίνουν στην επόμενη ομάδα που θα εξυπηρετηθεί μόλις ολοκληρωθεί η εξυπηρέτηση της τρέχουσας ομάδας για την οποία γίνεται η σάρωση. Αν η επόμενη ομάδα έχει παραπάνω αιτήσεις από N , τότε θα εξυπηρετηθούν μόνο οι πρώτες N . Η μέθοδος αυτή τείνει να είναι δίκαιη προς όλες τις αιτήσεις ανεξάρτητα του χρόνου άφιξής τους. Αν ο αριθμός N είναι μικρός ο αλγόριθμος συμπεριφέρεται σαν τον FCFS ενώ για μεγάλες τιμές η απόδοσή του είναι ανάλογη του SCAN. Η παραλλαγή όπου ο αριθμός N είναι μεταβλητός από ομάδα σε ομάδα λέγεται F-SCAN.



5. Άλλα θέματα διαχείρισης δίσκου

- Χρήση ενός μέρους της πολύ γρήγορης μνήμης (cache memory) για αποθήκευση των περιεχομένων των πιο συχνά χρησιμοποιούμενων τομέων του δίσκου (disk cache).
- Δύο προβλήματα πρέπει να αντιμετωπισθούν εδώ. Το πρώτο είναι πως ακριβώς γίνεται η προσπέλαση των δεδομένων που βρίσκονται εκεί. Υπάρχουν δύο πιθανοί τρόποι:
 - Μεταφορά των δεδομένων αυτών στη “συνηθισμένη” κύρια μνήμη πριν να χρησιμοποιηθούν.
 - Απ’ ευθείας χρήση τους από τη μνήμη cache με τη χρήση δεικτών.Η δεύτερη λύση φαίνεται να είναι καλύτερη λόγω του ότι δεν ξοδεύεται χρόνος στη μεταφορά των δεδομένων από τη μνήμη cache στην κύρια μνήμη. Επίσης, επιτρέπει την κοινή χρήση των δεδομένων από πολλές διεργασίες χρησιμοποιώντας μοντέλλα ταυτοχρονισμού τύπου αναγνωστών συγγραφέων.
- Το δεύτερο πρόβλημα έχει να κάνει με το ποια πολιτική αντικατάστασης τομέων του δίσκου θα υλοποιηθεί (αυτό είναι ένα παρόμοιο πρόβλημα με εκείνο της αντικατάστασης σελίδων). Ο καλύτερος αλγόριθμος φαίνεται να είναι ο LRU λόγω της αρχής της τοπικότητας αναφοράς στα δεδομένα. Ένας άλλος αλγόριθμος με καλά αποτελέσματα είναι ο LFU (Least Frequently Used). Προσοχή όμως πρέπει να δίνεται εδώ στο αν η συχνότητα αναφοράς είναι ομοιογενής χρονικά αλλιώς η εικόνα που παρουσιάζεται είναι πλασματική.
- Η καθυστέρηση περιστροφής μπορεί και αυτή να ελαχιστοποιηθεί με την επιλογή του τομέα εκείνου που βρίσκεται πιο κοντά στον βραχίονα (κάτι ανάλογο της μεθόδου SSTF).
- Περαιτέρω απόδοση μπορεί να επιτευχθεί με τη χρήση πολλαπλών δίσκων που δουλεύουν μαζί. Τα δεδομένα αποθηκεύονται στους δίσκους με τέτοιο τρόπο ώστε οι αιτήσεις για ανάγνωση/εγγραφή να ικανοποιούνται παράλληλα, μειώνοντας έτσι τον απαιτούμενο χρόνο. Ένας τέτοιος σχεδιασμός λέγεται Πλεονάζουσα Διάταξη Ανεξάρτητων Δίσκων (RAID).