

Ενότητα 7^η : Διασυνδέοντας Συσκευές I/O με τη Μνήμη, τον Επεξεργαστή και το Λειτουργικό Σύστημα - Μέθοδος της εξέτασης.

Σκοπός Ο σκοπός της ενότητας αυτής είναι να εξετάσει το ρόλο του λειτουργικού συστήματος και τον τρόπο επικοινωνίας του επεξεργαστή, της μνήμης και των συσκευών I/O.

Προσδοκώμενα Αποτελέσματα Όταν θα έχετε μελετήσει την ενότητα, θα είστε σε θέση να:



περιγράφετε τις αρμοδιότητες και τις λειτουργίες του λειτουργικού συστήματος,



εξηγείτε τον τρόπο επικοινωνίας του επεξεργαστή, της μνήμης και των συσκευών I/O,



αναφέρετε τις μεθόδους που χρησιμοποιούνται για να δώσει ο επεξεργαστής εντολές στις συσκευές I/O,



αιτιολογείτε γιατί η μέθοδος της εξέτασης είναι ο απλούστερος τρόπος για την επικοινωνία του επεξεργαστή με μία συσκευή I/O.



μέθοδος εξέτασης



Ο ρόλος του λειτουργικού συστήματος

Το λειτουργικό σύστημα διαδραματίζει κυρίαρχο ρόλο στη διαχείριση του I/O, δρώντας ως διασύνδεση (interface) ανάμεσα στο υλικό (hardware) και το πρόγραμμα που ζητά να επιτελέσει μια λειτουργία I/O.

Οι ευθύνες του λειτουργικού συστήματος προκύπτουν από τρία βασικά χαρακτηριστικά των συστημάτων I/O:

1. Το σύστημα I/O χρησιμοποιείται από κοινού από πολλά προγράμματα που χρησιμοποιούν τον επεξεργαστή.
2. Τα συστήματα I/O συχνά χρησιμοποιούν **διακοπές** (interrupts) - εξωτερικά παραγόμενες εξαιρέσεις - (πλήρης αναφορά στην ενότητα 8) για να μεταδώσουν πληροφορίες σχετικά με τις λειτουργίες εισόδου/εξόδου. Επειδή οι διακοπές προκαλούν μία μετάβαση από το κατάσταση πυρήνα (kernel mode) στο κατάσταση επόπτη (supervisor mode), η διαχείρισή τους πρέπει να γίνεται από το λειτουργικό σύστημα.
3. Ο **έλεγχος χαμηλού επιπέδου** μίας συσκευής I/O είναι πολύπλοκος, επειδή απαιτεί τη διαχείριση ενός συνόλου ταυτόχρονων γεγονότων και επειδή οι απαιτήσεις για το σωστό έλεγχο της συσκευής είναι συχνά πολυσύνθετες.

Λειτουργίες του λειτουργικού συστήματος

Τα τρία παραπάνω χαρακτηριστικά των συστημάτων I/O οδηγούν σε αρκετές διαφορετικές λειτουργίες που θα πρέπει να παρέχει το λειτουργικό σύστημα.

➤ Το λειτουργικό σύστημα εγγυάται πως **ένα πρόγραμμα χρήστη προσπελαύνει μόνο τα μέρη εκείνα μίας συσκευής I/O στα οποία ο χρήστης έχει δικαίωμα να προσπελάσει**. Για παράδειγμα, το λειτουργικό σύστημα δεν πρέπει να επιτρέπει σε ένα πρόγραμμα να διαβάσει ή να γράψει ένα αρχείο στο δίσκο, αν ο ιδιοκτήτης του αρχείου δεν έχει εκχωρήσει πρόσβαση σε αυτό το πρόγραμμα. Σε ένα σύστημα με κοινόχρηστες συσκευές I/O, δε θα είναι δυνατό να παρέχουμε προστασία αν τα προγράμματα των χρηστών μπορούν να επιτελούν απευθείας I/O.

➤ Το λειτουργικό σύστημα **παρέχει αφαιρετικά σχήματα** (abstractions) για την πρόσβαση των χρηστών στις συσκευές, παρέχοντας ρουτίνες που χειρίζονται τις χαμηλού επιπέδου λειτουργίες των συσκευών. Έτσι η πρόσβαση γίνεται ευκολότερη. Για παράδειγμα, ο χρήστης που φτιάχνει ένα πρόγραμμα το οποίο κάνει πρόσβαση σε ένα συγκεκριμένο αρχείο στο δίσκο δε θα ασχοληθεί με την περιστροφή της πλακέτας ή την τοποθέτηση της κεφαλής στο κατάλληλο σημείο : αυτά είναι δουλειά του λειτουργικού συστήματος.

➤ Το λειτουργικό σύστημα χειρίζεται τις **διακοπές** (interrupts) που παράγονται από τις συσκευές I/O, όπως ακριβώς χειρίζεται τις εξαιρέσεις (exceptions) που παράγονται από ένα πρόγραμμα.

➤ Το λειτουργικό σύστημα προσπαθεί να παρέχει **αμερόληπτη πρόσβαση στους κοινόχρηστους πόρους I/O**, όπως επίσης και να προγραμματίζει τις προσβάσεις έτσι ώστε να βελτιώσει το ρυθμό διαμεταγωγής του συστήματος.

Τύποι επικοινωνίας του λειτουργικού συστήματος

Για να επιτελέσει αυτές τις λειτουργίες για λογαριασμό των προγραμμάτων χρήστη, το λειτουργικό σύστημα πρέπει να μπορεί να επικοινωνεί με τις συσκευές I/O και να εμποδίζει το πρόγραμμα χρήστη να επικοινωνεί απευθείας με αυτές. **Τρεις τύποι επικοινωνίας απαιτούνται:**

1. Το λειτουργικό σύστημα πρέπει να μπορεί να **δίνει εντολές στις συσκευές I/O**. Αυτές οι εντολές δεν περιλαμβάνουν μόνο λειτουργίες όπως ανάγνωση και εγγραφή, αλλά και άλλες λειτουργίες που πρέπει να γίνουν σε μία συσκευή, όπως για παράδειγμα μία αναζήτηση (seek) στο δίσκο.
2. Η συσκευή I/O πρέπει να μπορεί να **ειδοποιήσει** το λειτουργικό σύστημα όταν αυτή έχει ολοκληρώσει μία λειτουργία ή έχει συναντήσει κάποιο λάθος. Για παράδειγμα, όταν ένας δίσκος έχει ολοκληρώσει μία αναζήτηση, θα ειδοποιήσει το λειτουργικό σύστημα.
3. Τα **δεδομένα πρέπει να μεταφέρονται** ανάμεσα στη μνήμη και μία συσκευή I/O. Για παράδειγμα, το μπλοκ που διαβάζεται από ένα δίσκο πρέπει να μετακινηθεί από το δίσκο στη μνήμη.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

Να εξηγήσετε σε 10-15 γραμμές ποιος είναι ο ρόλος του λειτουργικού συστήματος και ποιες οι λειτουργίες που θα πρέπει αυτό να παρέχει για τη σωστή προσπέλαση των συσκευών I/O. Να συγκρίνετε την απάντησή σας με την παράγραφο που μόλις διαβάσατε: «Ο ρόλος του λειτουργικού συστήματος».



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

Έστω ένα υπολογιστικό σύστημα στο οποίο οι συσκευές I/O είναι κοινόχρηστες. Θεωρείτε πως είναι εφικτό να παρέχεται προστασία αν τα προγράμματα των χρηστών μπορούν να επιτελούν απ' ευθείας I/O; Να αιτιολογήσετε την απάντησή σας.



ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 2

Δεν είναι εφικτό να παρέχεται προστασία αν τα προγράμματα των χρηστών μπορούν να επιτελούν απ' ευθείας I/O. Αυτό συμβαίνει επειδή αν τα προγράμματα χρηστών έχουν τη δυνατότητα να επιτελούν απευθείας I/O, σημαίνει ότι παρακάμπτουν το λειτουργικό σύστημα και οποιοδήποτε άλλο πρόγραμμα που είναι ικανό να θέσει όρια στη λειτουργία τους. Έτσι, είναι ικανά να προσπελάσουν οποιαδήποτε συσκευή επιθυμούν και να αποσπάσουν χωρίς περιορισμό οποιοδήποτε μπλοκ δεδομένων, ακόμα και αν δεν έχουν δικαιώματα σε αυτό.



Δίνοντας ο Επεξεργαστής Εντολές στις Συσκευές I/O

Για να δώσει μία εντολή σε μία συσκευή I/O, ο επεξεργαστής πρέπει να μπορεί να επικοινωνεί με τη συσκευή μέσω της κατάλληλης διεύθυνσης και να παρέχει μία ή περισσότερες λέξεις εντολών.

Δύο μέθοδοι χρησιμοποιούνται για να δεχτεί μια συσκευή εντολές:


1. I/O απεικονισμένο στη μνήμη (memory-mapped I/O)
2. Ειδικές εντολές I/O.

I/O απεικονισμένο στη μνήμη

Στο I/O απεικονισμένο στη μνήμη, τμήματα του χώρου διευθύνσεων της μνήμης εκχωρούνται σε συσκευές I/O. Εντολές ανάγνωσης και εγγραφής σε αυτές τις διευθύνσεις ερμηνεύονται ως εντολές στη συσκευή I/O.



Παράδειγμα

Μία λειτουργία εγγραφής μπορεί να χρησιμοποιηθεί, για να στείλει δεδομένα σε μία συσκευή I/O, όπου τα δεδομένα θα ερμηνευθούν ως μία εντολή για τις συσκευές I/O. Όταν ο επεξεργαστής τοποθετήσει τη διεύθυνση και τα δεδομένα στην αρτηρία μνήμης, το σύστημα μνήμης αγνοεί τη λειτουργία, επειδή η διεύθυνση δηλώνει ένα τμήμα του χώρου διευθύνσεων που χρησιμοποιείται για I/O. Ο ελεγκτής της συσκευής όμως, βλέπει τη λειτουργία, καταγράφει τα δεδομένα και τα μεταδίδει στη συσκευή ως μία εντολή. Έτσι, αν INBUF είναι η ενδιάμεση μνήμη αποθήκευσης εισόδου (input buffer) που σχετίζεται με το πληκτρολόγιο, η εντολή του 68000 MOVE.B INBUF, MEM διαβάζει ένα byte από το INBUF και το αποθηκεύει στη μνήμη στη διεύθυνση MEM. 



Τα προγράμματα χρήστη δε μπορούν να κάνουν απευθείας χρήση εντολών I/O, επειδή το λειτουργικό σύστημα δεν παρέχει πρόσβαση στο χώρο διευθύνσεων που έχει εκχωρηθεί στις συσκευές I/O και συνεπώς οι διευθύνσεις αυτές προστατεύονται κατά τη μετάφραση της διεύθυνσης.

Πλεονέκτημα της μεθόδου. Η χρήση του I/O απεικονισμένου στη μνήμη προσφέρει σημαντική ευελιξία στο χειρισμό των λειτουργιών εισόδου/εξόδου, κι αυτό επειδή οι εντολές μηχανής και οι τρόποι διευθυνσιοδότησης που μπορούν να χρησιμοποιηθούν για τους τελεστές μνήμης μπορούν επίσης να αναφερθούν και σε μια συσκευή I/O. Είναι

σύνηθες να εκχωρούνται γειτονικές διευθύνσεις μνήμης για να εξυπηρετήσουν τις συσκευές I/O. Ο PDP-11 και ο 68000 χρησιμοποιούν τη μέθοδο του I/O απεικονισμένου στη μνήμη για να επικοινωνήσουν με τις συσκευές εισόδου/εξόδου.

Ειδικές εντολές I/O


Η εναλλακτική λύση στο I/O απεικονισμένο στη μνήμη είναι η χρήση **ειδικευμένων εντολών I/O του επεξεργαστή**. Αυτές οι εντολές I/O μπορούν να προσδιορίσουν τόσο την ταυτότητα της συσκευής όσο και τη λέξη εντολής (ή τη θέση της λέξης εντολής στη μνήμη). Ο επεξεργαστής μεταδίδει τη διεύθυνση της συσκευής μέσω ενός συνόλου καλωδίων που κανονικά περιλαμβάνονται ως μέρη της αρτηρίας I/O. Η εντολή μπορεί να μεταδοθεί μέσω των γραμμών δεδομένων αρτηρίας. Παραδείγματα υπολογιστών που χρησιμοποιούν ειδικές εντολές I/O είναι ο Intel 80x86 και ο IBM 370.



Παράδειγμα

Μια εντολή επεξεργαστή όπως η :

OUT data , device

μπορεί να χρησιμοποιηθεί για να μεταφερθούν τα δεδομένα που προσδιορίζονται από τον πρώτο τελεστέο στη συσκευή που προσδιορίζεται από το δεύτερο τελεστέο. Η εκτέλεση αυτής της εντολής έχει ως αποτέλεσμα ο επεξεργαστής να εκκινήσει μια λειτουργία εγγραφής μέσω της αρτηρίας I/O , και όχι της αρτηρίας επεξεργαστή-μνήμης. Παρόμοια, μια λειτουργία εισόδου μπορεί να πραγματοποιηθεί με μια **εντολή IN**. 



Απαγορεύοντας στις εντολές I/O να εκτελούνται όταν ο επεξεργαστής δε βρίσκεται σε κατάσταση πυρήνα ή επόπτη, τα προγράμματα χρήστη δεν είναι ικανά να έχουν απευθείας πρόσβαση στις συσκευές.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

Να συγκρίνετε τις μεθόδους με τις οποίες ο επεξεργαστής μπορεί να δώσει εντολές στις συσκευές I/O ως προς την ευκολία της υλοποίησης. Να αιτιολογήσετε την απάντησή σας.



ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 3


Πιο απλή σε υλοποίηση είναι η μέθοδος του I/O απεικονισμένου στη μνήμη, επειδή σε αυτή χρησιμοποιείται απλώς ένα προσυμφωνημένο τμήμα μνήμης για την επικοινωνία επεξεργαστή-συσκευών I/O, δηλαδή δε χρειάζεται να υλοποιήσουμε τίποτα επιπλέον. Αντίθετα, για τη μέθοδο των ειδικευμένων εντολών I/O επεξεργαστή θα πρέπει να υλοποιήσουμε αυτές τις εντολές, οι οποίες ειδικευμένες καθώς είναι δε θα χρησιμεύσουν και πουθενά αλλού, ενώ είναι πολύ πιθανό να χρειάζεται να προσθέσουμε και επιπλέον καλώδια στην αρτηρία για τη μετάδοσή τους.

Πολλαπλές λειτουργίες I/O για μια αίτηση

Η πλήρης διεκπεραίωση μίας ανάγνωσης ή εγγραφής δεδομένων που έχει ζητήσει ένα πρόγραμμα συνήθως απαιτεί αρκετές ξεχωριστές λειτουργίες I/O. Επιπλέον, ο επεξεργαστής μπορεί να χρειάζεται να εξετάζει την κατάσταση της συσκευής κατά τη διάρκεια εκτέλεσης των εντολών για να διαπιστώσει αν η εντολή ολοκληρώθηκε με επιτυχία.



Παράδειγμα

Ο εκτυπωτής γραμμής DEC LP11 έχει δύο καταχωρητές συσκευής I/O – έναν για τις πληροφορίες κατάστασης και έναν για τα δεδομένα προς εκτύπωση. Ο καταχωρητής κατάστασης (Status register) περιέχει ένα bit διεκπεραίωσης (done bit), το οποίο τίθεται από τον εκτυπωτή όταν έχει εκτυπώσει ένα χαρακτήρα και ένα bit λάθους (error bit), που δηλώνει πως ο εκτυπωτής μπλόκαρε ή ξέμεινε από χαρτί. Κάθε byte των δεδομένων προς εκτύπωση τοποθετείται στον καταχωρητή δεδομένων (Data register). Ο επεξεργαστής πρέπει κατόπιν να περιμένει, μέχρι ο εκτυπωτής να θέσει το bit διεκπεραίωσης, προτού να μπορέσει να τοποθετήσει έναν άλλο χαρακτήρα (δηλαδή ένα byte) στην ενδιάμεση μνήμη (buffer). Ο επεξεργαστής πρέπει επίσης να ελέγχει το bit λάθους για να διαπιστώσει αν προέκυψε κάποιο πρόβλημα. Κάθε μία από αυτές τις λειτουργίες απαιτεί μία ξεχωριστή πρόσβαση στη συσκευή I/O. 



Επικοινωνώντας με τον επεξεργαστή / Εξέταση

Η διαδικασία κατά την οποία περιοδικά ελέγχονται τα bits του καταχωρητή κατάστασης μιας συσκευής για να διαπιστωθεί αν είναι ώρα για την επόμενη λειτουργία I/O ή αν χρειάζεται η επέμβαση του επεξεργαστή, ονομάζεται **εξέταση (polling)**.

Στο προηγούμενο παράδειγμα που αναφερόταν στον εκτυπωτή DEC LP11 είδαμε μια περίπτωση εξέτασης.

Η εξέταση είναι ο **απλούστερος** τρόπος για να επικοινωνεί μία συσκευή I/O με τον επεξεργαστή. Η συσκευή I/O απλά τοποθετεί τις πληροφορίες σε έναν καταχωρητή κατάστασης και ο επεξεργαστής απλώς «έρχεται» και τις λαμβάνει. Ο επεξεργαστής έχει τον απόλυτο έλεγχο και κάνει όλη τη δουλειά. Το **ποντίκι**, για παράδειγμα, είναι μία συσκευή μόνο εισόδου η οποία συνήθως προσπελάσσεται μέσω εξέτασης.

Μειονέκτημα: Το μειονέκτημα της εξέτασης είναι ότι μπορεί να **σπαταλήσει αρκετό χρόνο επεξεργαστή** αφού οι επεξεργαστές είναι κατά πολύ ταχύτεροι από τις συσκευές I/O. Ο επεξεργαστής μπορεί να έχει διαβάσει τον καταχωρητή κατάστασης πολλές φορές, για να διαπιστώνει κάθε φορά απλώς ότι η συσκευή δεν έχει ακόμα ολοκληρώσει μία λειτουργία I/O που είναι σχετικά αργή, ή ότι το ποντίκι δεν έχει μετακινηθεί από την τελευταία στιγμή που εξετάστηκε. Δηλαδή η επεξεργαστική ισχύς αναλώνεται σε άσκοπες ενέργειες.



Ακόμα και όταν η συσκευή ολοκληρώσει μία λειτουργία, πρέπει να διαβάσουμε την κατάσταση της για να διαπιστώσουμε αν η λειτουργία ήταν επιτυχής.



Παράδειγμα

Ας προσδιορίσουμε την επιβάρυνση που επιφέρει η διαδικασία της εξέτασης σε τρεις διαφορετικές συσκευές. Υποθέστε ότι το πλήθος των κύκλων ρολογιού για μία λειτουργία εξέτασης είναι 100 και ότι ο επεξεργαστής λειτουργεί με ένα ρολόι των 50 MHz.

Προσδιορίστε το ποσοστό του χρόνου της CPU που καταναλώνεται για τις ακόλουθες τρεις περιπτώσεις, υποθέτοντας ότι η εξέταση γίνεται τόσο συχνά ώστε τα δεδομένα να μη χάνονται ποτέ:

1. Το ποντίκι πρέπει να εξετάζεται 30 φορές το δευτερόλεπτο για να εξασφαλίσουμε ότι δε χάσαμε καμία κίνηση που έγινε από το χρήστη.
2. Η δισκέτα μεταφέρει δεδομένα στον επεξεργαστή σε μονάδες των 16 bit και έχει ένα ρυθμό δεδομένων των 50 KB/sec. Δε μπορεί να χαθεί καμία μεταφορά δεδομένων.

3. Ο σκληρός δίσκος μεταφέρει δεδομένα σε κομμάτια της μίας λέξης και μπορεί να εκτελεί μεταφορές με ρυθμό 2 MB/sec. Ξανά, δε μπορεί να χαθεί καμία μεταφορά δεδομένων.

Απάντηση:

Για το ποντίκι έχουμε:

Κύκλοι ρολογιού ανά sec για την εξέταση = $30 \times 100 = 3000$ κύκλοι ανά sec
Η εξέταση μπορεί προφανώς να χρησιμοποιηθεί για το ποντίκι, χωρίς μεγάλη επίδραση στην απόδοση του επεξεργαστή.

Για τη **δισκέτα**, ο ρυθμός με τον οποίο πρέπει να γίνεται η εξέταση είναι

$$\frac{50 \text{ KB/sec}}{2 \text{ bytes/προσπέλαση εξέτασης}} = 25 \text{ K} \frac{\text{προσπελάσεις}}{\text{sec}} = 25 \times 2^{10} \frac{\text{προσπελάσεις}}{\text{sec}}$$

Συνεπώς, μπορούμε να υπολογίσουμε το πλήθος των κύκλων (μετατρέποντας από το K στο 1024):

$$\text{Κύκλοι ρολογιού ανά sec για την εξέταση} = 25 \times 2^{10} \times 100 = 25.6 \times 10^5 \text{ κύκλοι ρολογιού ανά sec}$$

$$\text{Ποσοστό των κύκλων του επεξεργαστή που καταναλώθηκε} = \frac{25.6 \times 10^5}{50 \times 10^6} = 5\%$$

Αυτή η ποσότητα επιβάρυνσης είναι σημαντική, αλλά θα μπορούσε να γίνει ανεκτή σε ένα σύστημα χαμηλών προδιαγραφών με λίγες μόνο συσκευές I/O όπως αυτή η δισκέτα.

Στην περίπτωση του **σκληρού δίσκου**, πρέπει να κάνουμε εξέταση με ένα ρυθμό ίσο με το ρυθμό δεδομένων σε λέξεις, ο οποίος είναι 500K φορές το δευτερόλεπτο (2 MB ανά δευτερόλεπτο/4 bytes ανά μεταφορά). Συνεπώς, Κύκλοι ρολογιού ανά sec για την εξέταση = $500 \times 2^{10} \times 100 = 51.2 \times 10^6$ κύκλοι ρολογιού ανά sec

$$\text{Ποσοστό των κύκλων του επεξεργαστή που καταναλώθηκε} = \frac{51.2 \times 10^6}{50 \times 10^6} = 100\%$$

Συνεπώς, η ισχύς του επεξεργαστή καταναλώνεται ολοκληρωτικά στην περίπτωση εξέτασης του δίσκου. Προφανώς, η διαδικασία της εξέτασης δε γίνεται δεκτή να χρησιμοποιηθεί για ένα σκληρό δίσκο σε αυτή τη μηχανή.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

Να περιγράψετε συνοπτικά τη μέθοδο της εξέτασης και να αναφέρετε τα πλεονεκτήματα και τα μειονεκτήματα τα οποία έχει. Να συγκρίνετε την απάντησή σας την υποενότητα: «Επικοινωνώντας με τον επεξεργαστή/Εξέταση»



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5

Να αναφέρετε συσκευές I/O όπου τα πλεονεκτήματα της εξέτασης υπερτερούν και συσκευές όπου η εξέταση κρίνεται ακατάλληλη μέθοδος.



ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5

Οι συσκευές οι οποίες θα εξετάσουμε είναι το ποντίκι, η δισκέτα και ο σκληρός δίσκος. Για κάθε ένα από αυτά έχουμε:

Ποντίκι: Η εξέταση μπορεί να χρησιμοποιηθεί για το ποντίκι, χωρίς μεγάλη επίδραση στην απόδοση του επεξεργαστή.

Δισκέτα: Η ποσότητα επιβάρυνσης είναι σημαντική, αλλά θα μπορούσε να γίνει ανεκτή σε ένα σύστημα χαμηλών προδιαγραφών με λίγες μόνο συσκευές I/O όπως αυτή η δισκέτα.

Σκληρός δίσκος: Η ισχύς του επεξεργαστή καταναλώνεται ολοκληρωτικά στην περίπτωση εξέτασης του δίσκου. Προφανώς, η διαδικασία της εξέτασης δε γίνεται δεκτή να χρησιμοποιηθεί για ένα σκληρό δίσκο σε αυτή τη μηχανή.

Συμπερασματικά, η απλότητα της λειτουργίας του ποντικιού κάνει την εξέταση ελκυστική ιδέα, αφού άλλωστε επιφέρει μικρή επιβάρυνση. Αντίθετα, αν η μέθοδος της εξέτασης χρησιμοποιηθεί για την επικοινωνία με το σκληρό δίσκο, αυτή θα έχει ως αποτέλεσμα την κατανάλωση του 100% της υπολογιστικής ισχύος και για αυτό το λόγο κρίνεται ως ακατάλληλη.



Ανακεφαλαιώνοντας λοιπόν...



Το λειτουργικό σύστημα λειτουργεί ως η διασύνδεση ανάμεσα στις συσκευές εισόδου/εξόδου και τα υπόλοιπα μέρη του συστήματος (επεξεργαστής – μνήμη). Επίσης εγγυάται για την ασφάλεια και την «δίκαιη χρήση» των τμημάτων του συστήματος, βοηθά στην ευκολότερη πρόσβαση των συσκευών και χειρίζεται τις διακοπές.



Το λειτουργικό σύστημα είναι υπεύθυνο για τρεις τύπους επικοινωνίας: στέλνει δεδομένα στις συσκευές I/O, δέχεται δεδομένα από αυτές και φροντίζει να ανταλλάσσονται δεδομένα μεταξύ συσκευών και μνήμης.



Δύο μέθοδοι που χρησιμοποιούνται για να δώσει ο επεξεργαστής εντολές στις συσκευές εισόδου/εξόδου είναι το I/O απεικονισμένο στη μνήμη, και οι ειδικές εντολές I/O.



Στο I/O απεικονισμένο στη μνήμη, τμήματα του χώρου διευθύνσεων της μνήμης εκχωρούνται σε συσκευές I/O και οι εντολές ανάγνωσης και εγγραφής σε αυτές τις διευθύνσεις ερμηνεύονται ως εντολές στη συσκευή I/O. Αυτό το σχήμα προσφέρει μεγάλη ευελιξία.



Μια απλή μέθοδος με την οποία ο επεξεργαστής μπορεί να επικοινωνεί με τις συσκευές I/O είναι η εξέταση (polling). Σύμφωνα με αυτή, περιοδικά ελέγχονται τα bits του καταχωρητή κατάστασης μιας συσκευής για να διαπιστωθεί αν είναι ώρα για την επόμενη λειτουργία I/O.



Η εξέταση είναι απλή και αποτελεσματική, έχει όμως το μειονέκτημα ότι σπαταλά μεγάλο μέρος της ισχύος του επεξεργαστή και δεν εκμεταλλεύεται τη μεγάλη του ταχύτητα σε σχέση με τις συσκευές I/O. Ειδικά όταν πρόκειται για το σκληρό δίσκο η εξέταση δεν είναι αποδεκτή μέθοδος.