

Ενότητα 8^η: Διακοπές.

Σκοπός Ο σκοπός της ενότητας αυτής είναι να περιγράψει τη λειτουργία των διακοπών και ορισμένες παραμέτρους που τις ρυθμίζουν, καθώς επίσης και να παρουσιάσει τις ομοιότητες και διαφορές των διακοπών με τις εξαιρέσεις, τις υπορουτίνες και τη μέθοδο της εξέτασης.

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



περιγράφετε την έννοια και την λειτουργία των διακοπών,



περιγράφετε την ακολουθία των γεγονότων που συμβαίνουν κατά το χειρισμό μιας αίτησης διακοπής από μία συσκευή,



προσδιορίζετε τον τρόπο με τον οποίο αντιμετωπίζονται οι διαφορετικές προτεραιότητες των συσκευών I/O,



ταξινομείτε τις διακοπές ανάλογα με τον τρόπο που χρησιμοποιούνται,



συγκρίνετε τις διακοπές με τις υπορουτίνες, τις εξαιρέσεις και τη μέθοδο της εξέτασης.



Έννοια και λειτουργία των Διακοπών

Η επιβάρυνση που επιφέρει ένα σχήμα διασύνδεσης εξέτασης (polling interface) αναγνωρίστηκε πολύ καιρό πριν, οδηγώντας στην εφεύρεση των **διακοπών** (interrupts), για να ειδοποιείται ο επεξεργαστής όταν μία συσκευή I/O απαιτεί την προσοχή του. Το **σύστημα εισόδου/εξόδου οδηγούμενο από διακοπές** (interrupt-driven), το οποίο χρησιμοποιείται από όλα σχεδόν τα συστήματα τουλάχιστον για κάποιες συσκευές, χρησιμοποιεί τις διακοπές I/O για να δηλώσει στον επεξεργαστή ότι μία συσκευή I/O χρειάζεται την παρέμβασή του.



Όταν μία συσκευή θέλει να ειδοποιήσει τον επεξεργαστή ότι έχει ολοκληρώσει κάποια λειτουργία ή χρειάζεται την παρέμβασή του, προκαλεί τη διακοπή του επεξεργαστή. Ο μηχανισμός διακοπών απαλείφει την ανάγκη για τον επεξεργαστή να εξετάζει περιοδικά τη συσκευή, και έτσι αυτός είναι ελεύθερος να αφοσιωθεί στην εκτέλεση των υπολοίπων διεργασιών.

Εξέταση: Μπορούμε να χρησιμοποιήσουμε τον επεξεργαστή για να μεταφέρουμε δεδομένα μεταξύ συσκευής και μνήμης βασισμένοι στην εξέταση. Θεωρήστε το παράδειγμα του ποντικιού. Ο επεξεργαστής μπορεί περιοδικά να διαβάζει τις τιμές των μεταβλητών του ποντικιού καθώς και τη θέση των κουμπιών του. Αν η θέση του ποντικιού ή ένα από τα κουμπιά του έχουν μεταβληθεί σε σχέση με την προηγούμενη εξέταση, το λειτουργικό σύστημα μπορεί να ειδοποιήσει το πρόγραμμα που σχετίζεται με την ερμηνεία των μεταβολών του ποντικιού.

Διακοπές και λειτουργικό σύστημα: Ένας εναλλακτικός μηχανισμός είναι να κάνουμε τη μεταφορά των δεδομένων οδηγούμενη από διακοπές (interrupt-driven). Σε αυτή την περίπτωση, και πάλι το λειτουργικό σύστημα θα μετέφερε τα δεδομένα σε μικρά πλήθη bytes από τη συσκευή ή προς αυτή. Αλλά επειδή η λειτουργία I/O είναι οδηγούμενη από διακοπές, το λειτουργικό σύστημα ασχολείται με άλλες εργασίες όσο τα δεδομένα διαβάζονται από τη συσκευή ή γράφονται σε αυτή. Όταν το λειτουργικό σύστημα αναγνωρίσει ότι έγινε μία διακοπή από τη συσκευή, διαβάζει την κατάστασή της για να ελέγξει για λάθη. Αν δεν υπάρχει κανένα λάθος, το λειτουργικό σύστημα μπορεί να παράσχει το επόμενο τμήμα δεδομένων, για παράδειγμα, μέσω μίας ακολουθίας από εγγραφές απεικονισμένες στη μνήμη. Όταν και το τελευταίο byte μίας αίτησης I/O έχει μεταδοθεί και η λειτουργία I/O έχει ολοκληρωθεί, το λειτουργικό σύστημα μπορεί να ενημερώσει το πρόγραμμα σχετικά.



Ο επεξεργαστής και το λειτουργικό σύστημα κάνουν όλη τη δουλειά σε αυτή τη διεργασία, κάνοντας προσπέλαση στη συσκευή και τη μνήμη για κάθε τμήμα δεδομένων που μεταφέρεται.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

Θυμάστε τι είναι η μέθοδος I/O απεικονισμένο στη μνήμη και για ποιο λόγο χρησιμοποιείται; Στην περίπτωση που δυσκολεύεστε να απαντήσετε στην ερώτηση, κρίνεται σκόπιμο να ανατρέξετε στην 7^η ενότητα του κεφαλαίου αυτού και συγκεκριμένα στην υποενότητα με τίτλο: «Δίνοντας ο επεξεργαστής εντολές στις συσκευές I/O».




Παράδειγμα

Θεωρήστε μια διεργασία που απαιτεί να εκτελεστούν κάποιοι υπολογισμοί και να εκτυπωθούν τα αποτελέσματα σε ένα εκτυπωτή γραμμής. Στη συνέχεια ακολουθούν ακόμη περισσότεροι υπολογισμοί και εκτυπώσεις, και ούτω καθεξής. Υποθέτουμε ότι το πρόγραμμα αποτελείται από δύο ρουτίνες, τις **ΥΠΟΛΟΓΙΣΕ** (για συντομία **ΥΠ**) και **ΕΚΤΥΠΩΣΕ** (για συντομία **ΕΚ**). Ας θεωρήσουμε ότι η ΥΠ παράγει η γραμμές εξόδου, οι οποίες θα εκτυπωθούν από τη ρουτίνα ΕΚ.

Απάντηση:

Η απαιτούμενη διεργασία μπορεί να έρθει σε πέρας με την επαναλαμβανόμενη εκτέλεση πρώτα της ρουτίνας ΥΠ και στη συνέχεια της ΕΚ. Ο εκτυπωτής δέχεται μόνο μια γραμμή κειμένου κάθε φορά. Ετσι, η ΕΚ θα πρέπει να στέλνει μια γραμμή κειμένου, να αναμένει ώσπου αυτή να εκτυπωθεί, μετά να στείλει την επόμενη γραμμή μέχρις ότου εκτυπωθούν όλα τα αποτελέσματα.

Το μειονέκτημα αυτής της απλής προσέγγισης είναι ότι ο επεξεργαστής ξοδεύει ένα σημαντικό ποσοστό του χρόνου του περιμένοντας τον εκτυπωτή να ξαναγίνει έτοιμος για εκτύπωση. Αν ήταν δυνατό να εκτελούνταν η ΥΠ κατά τη διάρκεια της εκτύπωσης, θα επιτυγχάναμε μια συνολικά γρηγορότερη ταχύτητα εκτέλεσης. Το παραπάνω είναι δυνατό ως εξής:

Αρχικά, η ΥΠ εκτελείται και παράγει τις η πρώτες γραμμές εξόδου. Στη συνέχεια, εκτελείται η ΕΚ και στέλνει στον εκτυπωτή την πρώτη γραμμή κειμένου. Σε αυτό το σημείο, αντί να περιμένουμε να εκτυπωθεί η γραμμή, η ΕΚ μπορεί να ανασταλεί προσωρινά και να συνεχιστεί η εκτέλεση της ΥΠ. Όταν ο εκτυπωτής γίνει έτοιμος ειδοποιεί τον επεξεργαστή στέλνοντας ένα σήμα αίτησης διακοπής. Σε απάντηση, ο επεξεργαστής διακόπτει την εκτέλεση της ΥΠ και μεταθέτει τον έλεγχο στην ΕΚ. Η ΕΚ στέλνει τη δεύτερη γραμμή στον εκτυπωτή και αναστέλλεται ξανά. Τότε, η ΥΠ που είχε διακοπεί συνεχίζει την εκτέλεση από το σημείο της διακοπής. Αυτή η διεργασία συνεχίζεται μέχρις ότου εκτυπωθούν και οι η γραμμές. Η ρουτίνα που εκτελείται για να ικανοποιήσει μια αίτηση διακοπής λέγεται **ρουτίνα εξυπηρέτησης διακοπών** και είναι η ΕΚ. 

Συνοπτικά, η ακολουθία των γεγονότων που συμβαίνουν κατά το χειρισμό μιας αίτησης διακοπής μιας συσκευής είναι η εξής:

1. Η εντολή ενός προγράμματος επιτρέπει τις διακοπές στον επεξεργαστή.
2. Η συσκευή θέτει μια αίτηση διακοπής.
3. Ο επεξεργαστής διακόπτει το πρόγραμμα που εκτελείται εκείνη τη στιγμή.
4. Απενεργοποιείται η δυνατότητα των συσκευών για διακοπές (ορισμένες φορές ο επεξεργαστής επιτρέπει σε κάποιες τη δυνατότητα διακοπής, ενώ λαμβάνει χώρα μια άλλη). Αυτό συμβαίνει γιατί η ρουτίνα επεξεργασίας διακοπών πρέπει πρώτα να μελετήσει και να προωθήσει την υπάρχουσα διακοπή, και στη συνέχεια να ασχοληθεί με άλλες. Η εμφάνιση διακοπής ενώ ήδη μελετάται μια άλλη μπορεί να είναι καταστροφική για το σύστημα!
5. Η συσκευή πληροφορείται ότι η αίτησή της έχει αναγνωριστεί, και σε απάντηση, απενεργοποιεί το σήμα αίτησης διακοπής.
6. Εκτελείται η λειτουργία που ζητήθηκε από τη διακοπή.
7. Οι διακοπές ενεργοποιούνται ξανά.
8. Συνεχίζεται η εκτέλεση του προγράμματος που είχε αρχικά διακοπεί.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

Να αναφέρετε συνοπτικά την ακολουθία των γεγονότων που συμβαίνουν κατά το χειρισμό μιας αίτησης διακοπής μιας συσκευής. Να συγκρίνετε την απάντησή σας με την παράγραφο που μόλις διαβάσατε.

▣ Μετάδοση πληροφοριών στον επεξεργαστή

Για να μεταδοθούν πληροφορίες στον επεξεργαστή, όπως η ταυτότητα της συσκευής που προκαλεί τη διακοπή, ένα σύστημα μπορεί να χρησιμοποιεί είτε *διανυσματικές διακοπές* (vectored interrupts) είτε ένα *καταχωρητή αιτίας εξαίρεσης* (exception Cause register). Δηλαδή, όταν η διακοπή αναγνωρίζεται από τον επεξεργαστή, η συσκευή μπορεί να στείλει στον επεξεργαστή μέσω της αρτηρίας είτε τη διεύθυνση του διανύσματος διακοπής είτε ένα πεδίο κατάστασης, για να τοποθετηθεί στον καταχωρητή αιτίας εξαίρεσης. Αυτό έχει ως αποτέλεσμα, όταν το λειτουργικό σύστημα λαμβάνει τον έλεγχο να γνωρίζει την ταυτότητα της συσκευής που προκάλεσε τη διακοπή και να μπορεί άμεσα να εξετάσει τη συσκευή.

▣ Προτεραιότητες στις διακοπές

Για να αντιμετωπίσουν τις διαφορετικές προτεραιότητες των συσκευών I/O, οι περισσότεροι μηχανισμοί διακοπών έχουν διάφορα **επίπεδα προτεραιοτήτων**. Αυτές οι προτεραιότητες δηλώνουν τη σειρά με την οποία ο επεξεργαστής θα πρέπει να τις επεξεργαστεί. Έτσι, αν δύο ή περισσότερες διακοπές καταφτάσουν ταυτόχρονα στον

επεξεργαστή, αυτός θα επιλέξει τη σειρά με την οποία θα τις εκτελέσει με βάση την προτεραιότητά τους.

Τόσο οι εσωτερικά παραγόμενες εξαιρέσεις όσο και οι διακοπές I/O έχουν προτεραιότητες· τυπικά οι διακοπές I/O έχουν χαμηλότερη προτεραιότητα από τις εσωτερικές εξαιρέσεις. Μπορεί να υπάρχουν πολλαπλές προτεραιότητες διακοπών I/O, με τις συσκευές υψηλής ταχύτητας να λαμβάνουν τις υψηλότερες προτεραιότητες.

Αν ο μηχανισμός εξαιρέσεων είναι διανυσματικός, η διεύθυνση του διανύσματος διακοπής για μία γρήγορη συσκευή θα αντιστοιχεί στη διακοπή υψηλότερης προτεραιότητας. Αν χρησιμοποιείται ένας καταχωρητής Αιτίας, τότε τα περιεχόμενα του καταχωρητή για μία ταχύτερη συσκευή τίθενται για τη διακοπή υψηλότερης προτεραιότητας.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

Να προσδιορίσετε τον τρόπο με τον οποίο αντιμετωπίζονται οι διαφορετικές προτεραιότητες των συσκευών I/O. Να συγκρίνετε την απάντησή σας με την παράγραφο «Προτεραιότητα στις διακοπές».



Τύποι διακοπών

Μέχρι στιγμής έχουμε υποθέσει ότι μια αίτηση διακοπής προέρχεται από μια συσκευή I/O με τη μορφή ενός σήματος υλικού (hardware signal) που τίθεται στην αρτηρία υπολογιστή. Στην πραγματικότητα όμως, υπάρχουν πολλές άλλες χρήσεις για τις διακοπές, εκτός από αυτή με την οποία ελέγχουμε τις συναλλαγές εισόδου/εξόδου, όπως:

➤ **ΕΠΑΝΑΦΟΡΑ ΑΠΟ ΛΑΘΗ.** Ο επεξεργαστής έχει τη δυνατότητα να διακόψει ένα πρόγραμμα αν, καθώς εκτελεί τις εντολές του, διαγνώσει ένα **λάθος** ή μια **επικίνδυνη κατάσταση**. Για παράδειγμα, το πεδίο του κώδικα εντολής μιας εντολής (OP-code field) μπορεί να μην ανταποκρίνεται σε μια νόμιμη εντολή, ή μπορεί κάποια εντολή να επιχειρήσει διαίρεση με το μηδέν.

Όταν ενεργοποιείται μια διακοπή ως αποτέλεσμα τέτοιων καταστάσεων, ο επεξεργαστής δρα με τον ίδιο ακριβώς τρόπο όπως στην περίπτωση μιας I/O διακοπής: αναστέλλει το πρόγραμμα που εκτελείται και εκκινεί μια ρουτίνα εξυπηρέτησης της διακοπής. Αυτή θα πρέπει να λάβει τα κατάλληλα μέτρα για την επαναφορά από το λάθος, αν αυτή είναι δυνατή, ή τουλάχιστο να ενημερώσει το χρήστη για το λάθος.

Στην περίπτωση μιας διακοπής I/O, ο επεξεργαστής ολοκληρώνει την εκτέλεση της εντολής σε που βρίσκεται εκείνη τη στιγμή σε εξέλιξη πριν επικυρώσει τη διακοπή. Ωστόσο, όταν η διακοπή προκαλείται από λάθος, **δεν είναι δυνατό να ολοκληρωθεί η εκτέλεση της εντολής που διακόπηκε** και ο επεξεργαστής αρχίζει αμέσως την εξυπηρέτηση της διακοπής.

➤ **ΔΙΟΡΘΩΣΗ ΛΑΘΩΝ ΛΟΓΙΣΜΙΚΟΥ.** Μια άλλη σημαντική χρήση των διακοπών είναι η βοήθεια που προσφέρουν στη διόρθωση λαθών σε προγράμματα (debugging). Το λογισμικό του συστήματος συνήθως περιλαμβάνει ένα πρόγραμμα που αποκαλείται **διορθωτής λαθών** (debugger), το οποίο βοηθά τον προγραμματιστή να βρει τα λάθη σε ένα πρόγραμμα. Ο διορθωτής λαθών χρησιμοποιεί διακοπές για να παράσχει δύο σημαντικές ευκολίες στο χρήστη: την αναζήτηση (**trace**) και τα σημεία διακοπής (**breakpoints**).

- Μια λειτουργία αναζήτησης προκαλεί διακοπή σε ένα πρόγραμμα που ελέγχεται για λάθη μετά από **κάθε** εντολή που εκτελείται.
- Τα σημεία διακοπής παρέχουν μια παρόμοια λειτουργία, μόνο που τώρα το πρόγραμμα που ελέγχεται για λάθη διακόπτεται μόνο σε συγκεκριμένα σημεία που επιλέγονται από το χρήστη.

➤ **ΕΠΙΚΟΙΝΩΝΙΑ ΜΕΤΑΞΥ ΠΡΟΓΡΑΜΜΑΤΩΝ.** Οι εντολές διακοπών λογισμικού χρησιμοποιούνται από το λειτουργικό σύστημα για να επικοινωνήσουν ή να ελέγξουν την εκτέλεση άλλων προγραμμάτων.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

Να περιγράψετε συνοπτικά τουλάχιστον άλλους δύο τρόπους με τους οποίους χρησιμοποιούνται οι διακοπές εκτός από τον έλεγχο των συναλλαγών I/O. Να συγκρίνετε την απάντησή σας με την παράγραφο «Τύποι διακοπών».



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5

Να αναφέρετε μια σημαντική διαφορά των διακοπών I/O και των διακοπών που χρησιμοποιούνται για την επαναφορά από λάθος. Για ποιο λόγο θεωρείτε ότι υφίσταται η διαφορά αυτή; Να αιτιολογήσετε την απάντησή σας.



ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5

Στην περίπτωση μιας διακοπής I/O, ο επεξεργαστής ολοκληρώνει την εκτέλεση της εντολής σε που βρίσκεται εκείνη τη στιγμή σε εξέλιξη πριν επικυρώσει τη διακοπή. Ωστόσο, όταν η διακοπή προκαλείται από λάθος, δεν είναι δυνατό να ολοκληρωθεί η εκτέλεση της εντολής που διακόπηκε, και ο επεξεργαστής αρχίζει αμέσως την εξυπηρέτηση της διακοπής. Αυτό συμβαίνει επειδή η εντολή που περιέχει το λάθος, ακριβώς λόγω του λάθους αποτελεί κίνδυνο για το σύστημα και η εκτέλεσή της μέχρι τέλους είναι δυνατό να επιφέρει κάποια μη αποδεκτή κατάσταση. Γι' αυτό το λόγο παρεμβαίνει και η διαδικασία διακοπής, για να αποτρέψει τον κίνδυνο.



Ομοιότητες και διαφορές των διακοπών

■ Σχέση ανάμεσα στις ρουτίνες εξυπηρέτησης διακοπών και στις υπορουτίνες

Οι διακοπές μοιάζουν αρκετά με τις κλήσεις υπορουτίνων (subroutine calls). Και στην περίπτωση των διακοπών, όπως και στις υπορουτίνες, ο έλεγχος μεταφέρεται από την αρχική διαδικασία (η οποία κι αυτή είναι πρόγραμμα, όπως μια υπορουτίνα) σε μια δεύτερη διαδικασία – αυτή που ζήτησε τη διακοπή. Ταυτόχρονα, σώζεται προσωρινά στη μνήμη ο μετρητής εντολών της πρώτης διαδικασίας. Μόλις η δεύτερη διαδικασία ολοκληρωθεί επιστρέφεται ο έλεγχος στην πρώτη και συνεχίζεται η εκτέλεσή της από την επόμενη εντολή από αυτή που δείχνει ο αποθηκευμένος στη μνήμη μετρητής εντολών.



Όμως, θα πρέπει να σημειώσουμε μια σημαντική διαφορά. Μια υπορουτίνα εκτελεί μια λειτουργία που απαιτείται από το πρόγραμμα από το οποίο καλείται. Η ρουτίνα εξυπηρέτησης διακοπών αντίθετα, μπορεί να μην έχει καμία σχέση με το πρόγραμμα το οποίο εκτελείται τη στιγμή που λαμβάνεται η αίτηση διακοπής. Στη πραγματικότητα πολλές φορές συμβαίνει τα δυο προγράμματα να ανήκουν ακόμα και σε διαφορετικούς χρήστες.

■ Διαφορές διακοπών- εξαιρέσεων

Μία διακοπή I/O είναι ακριβώς όπως οι εξαιρέσεις, με δύο σημαντικές διαφορές:

1. Μία διακοπή I/O είναι **ασύγχρονη** σε σχέση με την εκτέλεση των εντολών. Δηλαδή, η διακοπή δε σχετίζεται με καμία εντολή και δεν εμποδίζει την ολοκλήρωση των εντολών. Αυτό είναι πολύ διαφορετικό είτε από τις εξαιρέσεις λάθους σελίδας (page fault exception) ή εξαιρέσεις όπως η αριθμητική υπερχείλιση (arithmetic overflow). Η μονάδα ελέγχου χρειάζεται μόνο να ελέγχει για μία εκκρεμή διακοπή I/O τη στιγμή που ξεκινά μία νέα εντολή.

2. Εκτός από το γεγονός ότι πραγματοποιείται μια διακοπή I/O, θα θέλαμε να στείλουμε **περαιτέρω πληροφορίες**, όπως την ταυτότητα της συσκευής που παρήγαγε τη διακοπή. Επιπλέον, οι διακοπές αντιπροσωπεύουν συσκευές που μπορεί να έχουν διαφορετικές προτεραιότητες και γι' αυτό και οι αιτήσεις διακοπών τους να έχουν **διαφορετική κρισιμότητα**.

■ Ομοιότητες μεταξύ της εξέτασης και των διακοπών

Έχουμε δει δύο διαφορετικές μεθόδους που επιτρέπουν μία συσκευή να επικοινωνεί με τον επεξεργαστή. Αυτές οι δύο τεχνικές, η **εξέταση** και οι **διακοπές I/O**, αποτελούν τη βάση για δύο μεθόδους υλοποίησης της μεταφοράς δεδομένων ανάμεσα στη συσκευή I/O και τη μνήμη.

- Και οι δύο αυτές τεχνικές εργάζονται καλύτερα με συσκευές χαμηλότερου εύρους ζώνης στις οποίες ενδιαφερόμαστε περισσότερο για την ελάττωση του κόστους του

ελεγκτή (controller) της συσκευής και της διασύνδεσης παρά για την παροχή μίας μεταφοράς υψηλού εύρους ζώνης.

- Τόσο οι μεταφορές με εξέταση, όσο και οι οδηγούμενες από διακοπές μεταφορές θέτουν την ευθύνη της μετακίνησης δεδομένων και της διαχείρισης της μεταφοράς στον επεξεργαστή.

Στο παρακάτω παράδειγμα φαίνεται πώς μια διασύνδεση I/O οδηγούμενη από διακοπές λειτουργεί για μια δισκέτα.



Παράδειγμα

Υποθέστε ότι έχουμε την ίδια δισκέτα και τον ίδιο επεξεργαστή που χρησιμοποιήσαμε στο παράδειγμα της ενότητας 7. Υποθέστε πως θέλει να μεταφέρει ποσότητες των 16 bit με ένα ρυθμό 50 KB το δευτερόλεπτο. Η επιβάρυνση για κάθε μεταφορά, περιλαμβάνοντας τη διακοπή, είναι 100 κύκλοι ρολογιού. Βρείτε το ποσοστό του επεξεργαστή που καταναλώνεται όταν η δισκέτα είναι ενεργή.

Απάντηση:

Ο ρυθμός με τον οποίο πρέπει να διακόπτουμε τον επεξεργαστή όταν ο δίσκος εκτελεί μεταφορά είναι:

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

Συνεπώς,

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

Ποσοστό του επεξεργαστή που καταναλώνεται κατά τη διάρκεια μία μεταφοράς

$$= \frac{25.6 \times 10^5}{50 \times 10^6} = 5\%$$

Φυσικά, οι δισκέτες δεν απασχολούνται στο 100% του χρόνου και ο επεξεργαστής παραμένει ανενόχλητος όταν ο δίσκος είναι ανενεργός, αφού δεν απαιτείται καμία εξέταση. Για παράδειγμα, αν η δισκέτα εκτελεί μεταφορές μόνο κατά το 10% του χρόνου, η επιβάρυνση είναι μόνο 0,5%.



Η απουσία επιβάρυνσης όταν το I/O είναι ανενεργό, αποτελεί το κύριο πλεονέκτημα μίας διασύνδεσης οδηγούμενης από διακοπές έναντι της εξέτασης.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 6

Τόσο στο παράδειγμα της ενότητας 7, όσο και στο παρόμοιο παράδειγμα αυτής της ενότητας διαπιστώνουμε ότι το ποσοστό του επεξεργαστή που καταναλώνεται για τον έλεγχο μιας ανάγνωσης από τη δισκέτα είναι και στις δυο μεθόδους (εξέτασης και διακοπών I/O) το ίδιο, 5%. Μπορείτε να εξηγήσετε πως αυτό συμβαδίζει με τη θεωρητική μελέτη ότι το με το σχήμα διακοπών πετυχαίνουμε ο επεξεργαστής να απασχολείται πολύ λιγότερο με τις διαδικασίες I/O; Να αιτιολογήσετε την απάντησή σας.



ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 6

Στην πραγματικότητα, το ποσοστό απασχόλησης του επεξεργαστή με τη μέθοδο των διακοπών είναι πολύ καλύτερο από 5%, αφού η δισκέτα είναι αυτονόητο ότι δε λειτουργεί όλη την ώρα. Έτσι, αν λειτουργεί μόνο κατά το 10% του χρόνου λειτουργίας του συστήματος – ένα αρκετά λογικό ποσοστό - τότε το ποσοστό απασχόλησης του επεξεργαστή πέφτει στο 0,5% , ποσοστό πολύ ικανοποιητικό. Δηλαδή, με το σχήμα διακοπών I/O ο επεξεργαστής απασχολείται μόνο όταν πραγματοποιείται I/O, ενώ με τη μέθοδο της εξέτασης ο επεξεργαστής πρέπει να ασχολείται συνέχεια με τις συσκευές I/O (είτε ενεργές ή ανενεργές), εξετάζοντάς τις περιοδικά, για να γνωρίζει σε ποια κατάσταση βρίσκονται και αν χρειάζονται την επέμβασή του.



Ανακεφαλαιώνοντας λοιπόν...



Με τη μέθοδο των διακοπών είναι δυνατόν μια συσκευή να επικοινωνήσει με τον επεξεργαστή προκαλώντας τη διακοπή του προγράμματος που εκτελεί αυτός εκείνη τη στιγμή, αναγκάζοντάς τον έτσι να την εξυπηρετήσει.



Όταν συμβαίνει μια διακοπή την ευθύνη φέρουν πλέον ο επεξεργαστής και το λειτουργικό σύστημα, το οποίο είναι υπεύθυνο για την εναλλαγή των διεργασιών και τη μελέτη της αίτησης διακοπής.



Για να λάβει ο επεξεργαστής πληροφορίες σχετικά με τη διακοπή, ένα σύστημα μπορεί να χρησιμοποιεί είτε διανυσματικές διακοπές είτε ένα καταχωρητή αιτίας εξαίρεσης.



Και στις διακοπές υπάρχουν προτεραιότητες, αφού οι συσκευές οι οποίες τις χρησιμοποιούν έχουν διαφορετικές προτεραιότητες εξυπηρέτησης.



Οι διακοπές, εκτός από τις συσκευές I/O , χρησιμοποιούνται και σε άλλες περιπτώσεις, όπως για την επαναφορά ενός προγράμματος μετά από λάθος, για τη διόρθωση λαθών ενός προγράμματος, ή για την επικοινωνία μεταξύ προγραμμάτων.



Η λειτουργία των διακοπών είναι παρόμοια με αυτή των υπορουτίνων , με τη διαφορά ότι το πρόγραμμα που διακόπτεται είναι συνήθως άσχετο με το νέο πρόγραμμα που καλείται.



Οι διακοπές είναι παρόμοιες με τις εσωτερικές εξαιρέσεις και οι στόχοι που επιτυγχάνονται χρησιμοποιώντας διακοπές είναι παρόμοιοι με αυτούς που ικανοποιούνται με τη μέθοδο της εξέτασης. Και στις δύο μεθόδους ενδιαφερόμαστε περισσότερο για την απλότητα της υλοποίησης και όχι για μια επικοινωνία υψηλού εύρους ζώνης.



Η απουσία επιβάρυνσης όταν το I/O είναι ανενεργό, αποτελεί το κύριο πλεονέκτημα μίας διασύνδεσης οδηγούμενης από διακοπές έναντι της εξέτασης.