

Εισαγωγή

1.1. Υλικό (Hardware)

Ο υπολογιστής ανεξάρτητα από το μέγεθός του, είναι μια ηλεκτρονική συσκευή, και μπορεί να θεωρηθεί σαν ένα εργαλείο (όπως ο μοχλός ή ο τροχός) το οποίο μπορεί να παράγει έργο. Σε θεμελιώδες επίπεδο λειτουργεί με δυαδικά ψηφία (bits) ή σειρές από μονάδες (1) και μηδενικά (0). Τα bits συνθέτουν μεγαλύτερες ομάδες που ονομάζονται bytes, συνήθως 8 bits = 1 byte. Τα bytes μπορούν πάλι να χρησιμοποιηθούν σαν ομάδες από 1, 2, ή 4 bytes (8, 16 ή 32 bits). Κάθε τέτοια ομάδα χαρακτηρίζεται σαν λέξη (word).

Οι περισσότεροι υπολογιστές αποτελούνται από τα παρακάτω:

- **Κ.Μ.Ε. (CPU)** ή Κεντρική Μονάδα Επεξεργασίας: Αυτή είναι η καρδιά του μηχανήματος και σε αυτήν γίνεται όλη η βασική εργασία.
- **Μνήμη:** Είναι μια συμπαγής σταθερή διάταξη, που περιλαμβάνει μια συλλογή από bits ή bytes και η οποία μπορεί να γραφτεί ή να διαβαστεί από την Κ.Μ.Ε. Οι υπολογιστές μπορούν να περιγραφούν ανάλογα με το μέγεθος της μνήμης τους σαν αυτούς των 32K, των 64K, ..., των 512K κ.λπ. Το «Κ» αντιπροσωπεύει την λέξη και η ακριβέστερα 1024, αφού στην πραγματικότητα είναι ίσο με το 2^{10} . Τόσο τα προγράμματα όσο και τα δεδομένα βρίσκονται στη μνήμη. Είναι σημαντικό να κατανοήσουμε ότι τόσο τα περιεχόμενα ενός κομματιού μνήμης, όσο και η διεύθυνση αυτού του συνδυασμού είναι μάλλον δύσκολες έννοιες. Η μνήμη ενός Η/Υ συχνά καλείται RAM ή «μνήμη τυχαίας προσπέλασης» (Random Access Memory). Αυτό απλά σημαίνει ότι στη μνήμη αυτή έχουμε τυχαία προσπέλαση. Έτσι για παράδειγμα προκειμένου να εξετάσουμε το περιεχόμενο της υπ' αριθμόν 97 διεύθυνσης της μνήμης αυτής, δεν χρειάζεται να εξετάσουμε προηγουμένως τις διευθύνσεις 1-96. Στην πραγματικότητα δεν υπάρχει τίποτε το τυχαίο σχετικά με την όλη διαδικασία. Όλα αυτά είναι από πριν καθορισμένα.

● **Διαυλος επικοινωνίας (Bus):** Ο συνδεδετικός κρίκος μεταξύ Κ.Μ.Ε. και των άλλων εξαρτημάτων, συμπεριλαμβανομένης και της μνήμης. Ο διαυλος μπορεί να χρησιμοποιηθεί για πολλούς σκοπούς. Είναι το κανάλι επικοινωνίας μεταξύ Κ.Μ.Ε. και μνήμης, και μπορεί να χρησιμοποιηθεί για να μεταφέρει οδηγίες (προγράμματα ή μέρη προγραμμάτων, οδηγίες προς τα άλλα εξαρτήματα) ή δεδομένα.

Αυτά τα τρία εξαρτήματα είναι μάλλον θεμελιώδη. Υπάρχουν ωστόσο και άλλες διατάξεις που χρησιμοποιούνται συνήθως κατά τη λειτουργία ενός Η/Υ. Πιο συγκεκριμένα:

● **Δίσκοι:** Αν χρησιμοποιείτε μικροϋπολογιστή θα είσατε ενήμεροι σχετικά με τους δίσκους. Θα διαθέτετε τουλάχιστον έναν οδηγό δισκέτας ή ακόμα καλύτερα, ένα σκληρό δίσκο (Winchester). Οι δίσκοι χρησιμοποιούνται για τη φύλαξη προγραμμάτων ή δεδομένων. Διαφέρουν από τη μνήμη στο ότι ο χρόνος προσπέλασης για την ανάκληση πληροφοριών δεν είναι σταθερός. Λαμβάνοντας υπόψη το γεγονός ότι οι δίσκοι είναι ηλεκτρομηχανικά εξαρτήματα, είναι φυσικό να είναι πολύ πιο αργοί σε σχέση με τη σταθερή συμπαγή μνήμη. Μερικοί μικροϋπολογιστές διαθέτουν δίσκους από σιλικόνη, που αποτελούν μεγάλα κομμάτια μνήμης, κατασκευασμένα να λειτουργούν και να χρησιμοποιούνται σαν ένας δίσκος. Σε αυτές τις περιπτώσεις έχουμε πολύ μικρό χρόνο αποθήκευσης σε σχέση με ένα κανονικό δίσκο.

Ένας απλός οδηγός δισκέτας μπορεί να αποθηκεύσει σε μια δισκέτα από το 1/3 έως το 1/2 του ενός Megabyte (1.000.000. bytes) σε πληροφορίες. Ενώ ένας σκληρός δίσκος μπορεί να φυλάξει πάνω από 30 Mb. Η δισκέτα έχει το πλεονέκτημα ότι μπορεί να μετακινηθεί πολύ εύκολα και έτσι να είμαστε σε θέση να μεταφέρουμε τα προγράμματά μας από ένα μηχάνημα σε άλλα. Κάτι τέτοιο δεν συμβαίνει με τους σκληρούς δίσκους ή με τους δίσκους σιλικόνης.

● **Ταινίες:** Ανέκαθεν ήταν ότι καλύτερο για αποθήκευση πληροφοριών μια και μπορούν να φυλάξουν τεράστιες ποσότητες δεδομένων πολύ φτηνά. Όλα τα μεγάλα συστήματα διαθέτουν ταινίες, ενώ ελάχιστοι μικροϋπολογιστές είναι σε θέση να κάνουν το ίδιο. Πιο συγκεκριμένα μερικοί multi-users μικροϋπολογιστές έχουν ταινία στην οποία κρατούνται κόπιες από τα περιεχόμενα των δίσκων για λόγους αρχειοθέτησης. Κάτι παρόμοιο μπορεί να γίνει με το κασετοφώνο σας.

● **Λοιπά:** Υπάρχουν πολλές άλλες διατάξεις που μπορούν να συνδεθούν στους Η/Υ. Ένα παράδειγμα είναι οι εκτυπωτές, οι οποίοι ποικίλουν σε είδη, όπως οι λέιζερ εκτυπωτές, οι matrix, οι εκτυπωτές μαργαρίτας, ακόμα και οι γραφομηχανές. Ακόμα και οι μικρότεροι μικροϋπολογιστές είναι σε θέση να χειριστούν penplotters, και να σχεδιάσουν αρκετά εντυπωσιακά σχέδια. Μερικοί Η/Υ θα «επέτρεπαν» τον έλεγχο ηλεκτρονικών συσκευών που ρυθμίζουν την κεντρική θέρμανση, ανοίγουν την είσοδο του γκαράζ κλπ. Όλα αυτά είναι εξωτερικές διατάξεις. Υπάρχουν και εσωτερικές διατάξεις όπως βίντεο, κάμερες, βιομηχα-

νικες διατάξεις, «ποντίκια», χειριστήρια για παιχνίδια, τηλέφωνα, στυλό φωτοκυτταρου κλπ.

Μερικοί μικροϋπολογιστές έχουν επίσης την δυνατότητα-ικανότητα να χρησιμοποιούν πρόσθετους μικροεπεξεργαστές πιο ειδικευμένους σε συγκεκριμένες εργασίες. Για παράδειγμα τους μαθηματικούς συν-επεξεργαστές, που έχουν την ικανότητα να εκτελούν τις αριθμητικές πράξεις πολύ γρηγορότερα από τους "standard" επεξεργαστές. Αυτό δίνει ένα πλεονέκτημα στην εκτέλεση προγραμμάτων με πολλούς αριθμητικούς υπολογισμούς.

Μια από τις πιο βασικές διατάξεις που δεν αναφέρθηκε μέχρι τώρα, είναι η απλούστερη απ' τις συσκευές εισόδου/εξόδου: το πληκτρολόγιο και η οθόνη. Αυτά αποτελούν τώρα πια, αναπόσπαστο μέρος-κομμάτι κάθε μικροϋπολογιστή. Από πολλούς τα μέρη αυτά αναφέρονται και σαν μικροκομπιούτερ, από άλλους VDU (Video Pisplay Unit), ή ακόμη και σαν τερματικό. Μπορεί να υπάρχει και ένας εκτυπωτής αντί μιας οθόνης αν και οι οθόνες είναι πιο φθηνές από τους εκτυπωτές. Η επικοινωνία με τον Η/Υ γίνεται μέσω του πληκτρολογίου, πράγμα που απαιτεί μια σχετική εξοικείωση με το πλαίσιο του.

Είναι πολύ εκνευριστικό να ξέρεις τι θέλεις να πληκτρολογήσεις, αλλά να μην μπορείς να βρεις τα σύμβολα στο πληκτρολόγιο. Το πιο σημαντικό ίσως πλήκτρο εκτός από το κουμπί που ανάβει και σβήνει τον Η/Υ) είναι το πλήκτρο που επιστρέφει τον «κέρσορα» στην αρχή της επόμενης γραμμής, και είναι γνωστό σαν "ENTER". Είναι κάπου στην δεξιά πλευρά του πληκτρολογίου, και πρέπει να είναι μεγαλύτερο από τα υπόλοιπα πλήκτρα. Με την πίεση του πλήκτρου αυτού, στέλνουμε στον υπολογιστή την εντολή που έχουμε γράψει σε μια γραμμή της οθόνης.

1.2. Λογισμικό (Software)

Λογισμικό (software) είναι ένα γενικό όνομα που δίνεται στα προγράμματα που «τρέχουν» από το υλικό (hardware). Ένα από τα πιο σπουδαία μέρη του λογισμικού είναι το λειτουργικό σύστημα το οποίο δίνεται μαζί με το μηχάνημα. Το λειτουργικό σύστημα είναι απλώς ένα πρόγραμμα, αν και τυπικά είναι ένα πολύπλοκο πρόγραμμα. Ένας συγκεκριμένος Η/Υ μπορεί να υποστηρίζει αρκετά διαφορετικά λειτουργικά συστήματα (αλλά σπάνια ταυτόχρονα).

Οι περισσότερες από τις ορατές εργασίες του λειτουργικού συστήματος αφορούν την επεξεργασία αρχείων. Ένα αρχείο είναι μια συλλογή από πληροφορίες που αναφερόμαστε σ' αυτές με κάποιο όνομα.

Από τα αρχεία, εκείνα που θα μας απασχολήσουν κατά κύριο λόγο είναι τα αρχεία προγραμμάτων. Σε ειδικό κεφάλαιο θα αναφερθούμε με λεπτομέρειες σ' ένα άλλο σημαντικό είδος αρχείων. Στα αρχεία δεδομένων.

Κάθε πρόγραμμα είναι ένα σύνολο εντολών προς τον υπολογιστή, γραμμένες σ' ένα συγκεκριμένο τρόπο σε μια γλώσσα προγραμματισμού. Οι γλώσσες προγραμματισμού μπορούν για τους σκοπούς μας, να χωρισθούν σε γλώσσες υψηλού επιπέδου και γλώσσες χαμηλού επιπέδου. Μια γλώσσα χαμηλού επιπέδου

είναι η γλώσσα που συνεργάζεται στενά με την φύση του υλικού (hardware) του Η/Υ. Τυπικά υπάρχει μία προς μία αντιστοιχία μεταξύ των εντολών χαμηλού επιπέδου και των εντολών της μηχανής. Ο σύμβολο-μεταφραστής (assembler) είναι μια τέτοια γλώσσα. Με μια γλώσσα υψηλού επιπέδου (όπως η FORTRAN) υπάρχει αντιστοιχία μιας εντολής υψηλού επιπέδου σε πολλές εντολές μηχανής.

Στο λογισμικό ενός υπολογιστή περιλαμβάνεται συνήθως, εκτός από το λειτουργικό σύστημα, ένας διορθωτής (editor) καθώς και μερικοί μεταφραστές (compilers), μια assembler και ένα loader ή link editor.

Ένας compiler μεταφράζει δηλώσεις υψηλού επιπέδου στις σωστές εντολές επιπέδου μηχανής. Ένας Loader ή link editor, παίρνει τα αποτελέσματα από τον compiler και ολοκληρώνει την εργασία της δημιουργίας ενός προγράμματος έτοιμου προς εκτέλεση, προσθέτοντας σ' αυτό μερικές επιπλέον δυνατότητες.

Μια assembler μεταφράζει δηλώσεις γλώσσας χαμηλού επιπέδου στις αντίστοιχες της μηχανής.

Ένας editor παρέχει στο χρήστη τη δυνατότητα αλλαγών σε άλλα προγράμματα ή δεδομένα. Όλα τα συστήματα υπολογιστών γενικού σκοπού, έχουν τουλάχιστον έναν editor για τη δημιουργία και την τροποποίηση αρχείων προγραμμάτων.

Η μετατροπή των εντολών υψηλού επιπέδου σε εντολές μηχανής, οι οποίες είναι και οι μόνες κατανοητές απ' τον υπολογιστή, γίνεται μέσω ενός ειδικού προγράμματος που λέγεται μεταφραστής (compiler). Για κάθε γλώσσα προγραμματισμού δηλαδή, υπάρχει και ένας μεταφραστής. Για να γίνει επομένως κατανοητό από τον υπολογιστή ένα πρόγραμμα FORTRAN, πρέπει ο υπολογιστής αυτός να διαθέτει το ειδικό αυτό πρόγραμμα-μεταφραστή της FORTRAN.

Ο πιο κοινός ίσως τύπος editor είναι ο editor οθόνης, ο οποίος επιτρέπει τη μεταφορά μέρους του προγράμματος στην οθόνη, και την μετακίνηση μέσα σ' αυτό με ειδικά πλήκτρα, συνήθως εκείνα με τα βέλη. Άλλα πλήκτρα επιτρέπουν την αλλαγή, την προσθήκη ή την απαλοιφή χαρακτήρων.

Ο άλλος τύπος editor είναι ο editor γραμμής, ο οποίος επιτρέπει την επέμβαση στο πρόγραμμα μόνο κατά γραμμές. Μεταφέρεται δηλαδή στην οθόνη, όχι μέρος του προγράμματος, αλλά συγκεκριμένη γραμμή του, που ορίζεται με αριθμό γραμμής. Το είδος αυτό του editor είναι συχνά αρκετά ισχυρό, αλλά σπάνια απλό στη χρήση, όπως ο editor οθόνης.

Αν και η σημασία του editor για το γράψιμο και τη διόρθωση ενός προγράμματος είναι αρκετά σημαντική, πολλές φορές η επιλογή του δεν εξαρτάται από τον χρήστη αλλά από το τι είναι διαθέσιμο για τον συγκεκριμένο υπολογιστή.

Καθώς δημιουργούμε αρχεία προγραμμάτων και ιδιαίτερα όταν τα μεταφράζουμε, όλο και περισσότερο μέρος του δίσκου καταλαμβάνεται, χωρίς κάτι τέτοιο να είναι πάντοτε απαραίτητο για τις επιδιώξεις μας.

Σε μερικά συστήματα όταν διορθώνουμε ένα πρόγραμμα, ένα νέο αρχείο προγράμματος δημιουργείται, χωρίς να σβύνεται το παλιό. Περισσότερα αρχεία προγραμμάτων δεν σημαίνει μόνο περισσότερο χώρο αλλά και περισσότερο χρόνο ανίχνευσης για την εύρεση του ζητούμενου.

1.3. Γλώσσες προγραμματισμού

Είναι γνωστό ότι ένας αλγόριθμος είναι μια ακολουθία εντολών, που σχεδιάστηκαν για τη λύση ενός μέρους ή ενός ολόκληρου προβλήματος. Ένα πρόγραμμα είναι η πραγματοποίηση βημάτων του αλγόριθμου σε κάποια από τις γλώσσες Προγραμματισμού. Η ανάπτυξη αυτών των γλωσσών προγραμματισμού είναι ένα από τα σπουδαιότερα επιτεύγματα της τελευταίας τριακονταετίας.

Έτσι όπως διάφορες γλώσσες (Ελληνική, Αγγλική, κλπ.) τις μαθαίνουμε για να τις χρησιμοποιούμε στην επικοινωνία μας με ανθρώπους από διαφορετικές χώρες, έτσι και για την «επικοινωνία» μεταξύ ανθρώπων και ηλεκτρονικών υπολογιστών χρησιμοποιούμε κάποια γλώσσα, όπου αποδέκτης είναι μια υπολογιστική μηχανή που είναι κατάλληλα εξοπλισμένη ώστε να «καταλαβαίνει» αυτή τη γλώσσα ή κάποια άλλη.

Για να εκτελέσει κάποια εργασία ένας υπολογιστής πρέπει να πάρει συγκεκριμένες εντολές-οδηγίες οι οποίες να είναι μέσα στο σύνολο των εντολών που διαθέτει, διαφορετικά δεν τις εκτελεί επειδή δεν τις αναγνωρίζει. Μια σειρά εντολών είναι και το Πρόγραμμα. Πρέπει εδώ να παρατηρήσουμε ότι ο υπολογιστής δεν μπορεί να αναγνωρίσει αν μια σειρά εντολών είναι στην λογική τους θέση ή όχι και το μόνο που κάνει είναι να εκτελέσει τις εντολές πιστά με την σειρά που του δόθηκαν. Έτσι αντιλαμβανόμαστε και πρέπει να το έχουμε αυτό πάντα υπόψη μας ότι ο υπολογιστής δεν μπορεί να «σκεφτεί», αφού δεν διαθέτει λογική, παρά μόνο να εκτελέσει οδηγίες που καταλήγουν σε λάθος ή σωστό αποτέλεσμα ανάλογα με το αν χρησιμοποιήσαμε σωστά τη λογική ακολουθία για την επίλυση ενός προγράμματος και την σωστή γραφή του προγράμματος σε κάποια από τις γλώσσες προγραμματισμού.

Στις γλώσσες Προγραμματισμού διακρίνουμε δύο κατηγορίες: α) χαμηλού επιπέδου γλώσσες (Low level Languages ή ASSEMBLY) πολύ συγγενική με την γλώσσα μηχανής, όπου συνδυασμοί δυαδικών αριθμών έχουν αντικατασταθεί με «μνημονικούς» συμβολισμούς. β) Υψηλού επιπέδου γλώσσες (High Level Languages). Οι γλώσσες αυτές έχουν αναπτυχθεί για τον προγραμματισμό ηλεκτρονικών υπολογιστών, είναι πιο εύκολες στην εκμάθηση τους από τη γλώσσα μηχανής και την ASSEMBLY και είναι ανεξάρτητες από τα διάφορα συστήματα υπολογιστών.

Η δημιουργία των γλωσσών αυτών ξεκίνησε για την ικανοποίηση διάφορων αναγκών σε διάφορα προβλήματα με κατεύθυνση όπως εμπειρικές τεχνικές, επιστημονικές εφαρμογές.

Όμως σήμερα με την εξέλιξη των υπολογιστών, όπως ήταν φυσικό, ακολούθησε και η εξέλιξη και ο εμπλουτισμός των διάφορων γλωσσών προγραμματισμού, με αποτέλεσμα να χρησιμοποιούνται με τις δυνατότητες που παρουσιάζουν σε όλα τα είδη των προβλημάτων. Για παράδειγμα πριν λίγα χρόνια πιστεύαμε, ότι με την γλώσσα FORTRAN δεν είναι εύκολο να διαχειρισθούμε αρχεία. Όμως αυτό δεν ισχύει και μπορούμε να πούμε, πράγμα που θα διαπιστώσετε και στα σχετικά κεφάλαια του βιβλίου αυτού, ότι μπορούμε να διαχειριστούμε αρχεία και να αναπτύξουμε προγράμματα εμπορικών εφαρμογών καλύτερα

ακόμα και από την COBOL, που ειδικά χρησιμοποιείται στις περιπτώσεις αυτές
Οι κυριότερες γλώσσες είναι:

- FORTRAN (FORMULA TRANSLATION)
- BASIC (BEGINNERS ALL-PURPOSE SYMBOLIC INSTRUCTION CODE)
- ALGOL (ALGORITHM LANGUAGE)
- COBOL (COMMON BUSINESS ORIENTED LANGUAGE)
- PL 1 (PROGRAMMING LANGUAGE 1)
- PASCAL
- LISP
- FORTH κτλ.

1.3.1. Η γλώσσα προγραμματισμού FORTRAN

Η ονομασία της λέξης FORTRAN προέρχεται από τις λέξεις FORMULA TRANSLATION (τύπος μετάφρασης).

Η γλώσσα FORTRAN έχει τη δυνατότητα έκφρασης κάθε προβλήματος που απαιτεί αριθμητικές πράξεις. Έτσι χρησιμοποιείται κύρια στην επίλυση μαθηματικών, επιστημονικών και τεχνικών προβλημάτων χωρίς να αποκλείεται η χρήση της και για εμπορικές και οικονομικές εφαρμογές.

Η πρώτη μορφή της FORTRAN, η FORTRAN II εμφανίστηκε το 1955 από την IBM και τα κύρια χαρακτηριστικά της ήταν:

περιείχε μόνο πραγματικές και ακέραιες σταθερές και μεταβλητές, περιορισμένες εντολές εισόδου-εξόδου και περιορισμένους κωδικούς FORMAT και διαθέτει λίγες συναρτήσεις βιβλιοθήκης. Με την εξάπλωση και την ανάπτυξη των Ηλεκτρονικών Υπολογιστών όπως ήταν φυσικό άρχισαν να επεκτείνονται και οι δυνατότητες της γλώσσας. Έτσι το 1966 παρουσιάζεται η νέα εμπλουτισμένη μορφή της FORTRAN II, γνωστή σαν FORTRAN IV, ή 66 που προτάθηκε από το Αμερικανικό Ινστιτούτο Προτύπων ANSI (AMERICAN NATIONAL STANDARDS INSTITUTE).

Η FORTRAN IV παρουσιάζει βελτιώσεις στην αποδοχή σταθερών, μεταβλητών, μιγαδικών, διπλής ακρίβειας και λογικών, διαθέτει πολλές συναρτήσεις βιβλιοθήκης, περισσότερες δυνατότητες στις εντολές εισόδου-εξόδου καθώς και περισσότερων κωδικών FORMAT.

Η εξέλιξη στην FORTRAN συνεχίστηκε και μετά το 1966 και έτσι μετά από 11 χρόνια αναγγέλθηκε το 1978 η FORTRAN 77 η οποία τυποποιήθηκε το 1978. Στην πορεία της αυτή η FORTRAN βελτιώνεται σημαντικά και εισάγονται ουσιαστικές δυνατότητες κυριότερες από τις οποίες είναι:

α) Οι δύο τελείες και η απόστροφος ' , περιλαμβάνονται στα σύμβολα της FORTRAN. Τα σχόλια μπορούν να αρχίζουν και με αστερίσκο στη στήλη 1 (εκτός από το c).

- β) Εισάγεται το σύνθετο λογικό IF με τις εντολές IF (M) THEN, ELSE IF (L) THEN, ELSE, ENDIF.
- γ) Εισάγονται οι σταθερές και μεταβλητές χαρακτήρων.
- δ) Τροποποιείται η εντολή DO. Τα m_1 , m_2 , m_3 μπορούν να είναι μεταβλητές ή εκφράσεις ακέραιες, πραγματικές, ή διπλής ακριβείας, το βήμα της μπορεί να είναι αρνητικό, κλπ.
- ε) Υπάρχει η δυνατότητα να ρυθμίζει ο ηλεκτρονικός υπολογιστής την FORMAT στις εντολές εισόδου-εξόδου (με την χρήση αστερισκων στη θέση του αριθμού της FORMAT).
- στ) Επιτρέπεται η ανάμιξη διάφορων τύπων σταθερών και μεταβλητών σε μια εκφραση.
- ζ) Εισάγονται οι εντολές PARAMETER, PROGRAM.
- η) Δίνεται η δυνατότητα εκτέλεσης ενός προγράμματος από ενδιάμεσο σημείο.
- θ) Γενικεύονται τα όρια μέσα στα οποία μεταβάλλονται οι δείκτες ενός πίνακα. Γίνονται δεκτοί πίνακες μέχρι επτά δείκτες.
- ι) Αυξάνονται οι δυνατότητες των εντολών εισόδου-εξόδου.
- κ) Εισάγονται οι εντολές OPEN, CLOSE, INQUIRE για την χρήση αρχείων, με αποτέλεσμα να μειώνεται η ανάγκη εντολών του λειτουργικού συστήματος του υπολογιστή και να γίνονται τα προγράμματα ανεξάρτητα από το σύστημα κλπ.

Η FORTRAN 77 προβλέπεται ότι θα παραμείνει για μερικά χρόνια η γλώσσα με τη μεγαλύτερη εφαρμογή όχι μόνο στα επιστημονικά και τεχνικά προβλήματα, αλλά και στα οικονομικο-εμπορικά όπου έχει ήδη κατακτήσει τις προτιμήσεις πολλών προγραμματιστών.

Σήμερα γίνονται προσπάθειες για βελτιώσεις και οι διάφορες μέθοδοι εφαρμόζονται στους υπερυπολογιστές (Supercomputers), CRAY και HITACHI.

1.4. Επίλυση ενός προβλήματος με τον Υπολογιστή

Είναι γνωστό ότι ένας Η/Υ δεν μπορεί να σκεφθεί, μπορεί μόνο να ακολουθεί προρισμένες εντολές κατεύθυνσης.

Προκειμένου να ελαχιστοποιήσουμε τα σφάλματα που παρουσιάζονται κατά την διαδικασία επίλυσης ενός προβλήματος με τον Η/Υ πρέπει να ακολουθούνται τα εξής στάδια:

1. Μαθηματική ανάλυση του προβλήματος.
2. Δημιουργία του αλγόριθμου.
3. Σχηματική παρουσίαση του αλγόριθμου.
4. Μεταφορά της διαδικασίας του αλγόριθμου στην γλώσσα προγραμματισμού.
5. Τρέξιμο του προγράμματος στον Η/Υ.
6. Έλεγχος αποτελεσμάτων.
7. Διόρθωση σφαλμάτων.

Κάθε ένα από τα βήματα αυτά περιγράφεται με λεπτομέρεια στις παραγράφους που ακολουθούν σ' αυτό το κεφάλαιο.

1.4.1. Μαθηματική ανάλυση του προβλήματος

Προκειμένου να επιλύσουμε οποιοδήποτε πρόβλημα όσο σύνθετο και αν είναι αυτό, πρέπει πρώτα πρώτα να λάβουμε υπόψη μας τα δεδομένα, κατόπιν να βρούμε τις εξισώσεις που πρέπει να χρησιμοποιηθούν και να καταστρωθούν όλες οι αριθμητικές διαδικασίες έτσι ώστε, να ακολουθήσουν χωρίς εμπλοκή τα αποτελέσματα.

Για παράδειγμα έστω ότι θέλουμε να βρούμε το εμβαδόν ενός τριγώνου χρησιμοποιώντας τις πλευρές του.

Τα βήματα της ανάλυσής μας θα είναι:

α) a, β, γ είναι τα δεδομένα

β) $t = a + \beta + \gamma / 2$ και

$E = \sqrt{t(t-a)(t-\beta)(t-\gamma)}$ είναι οι τύποι που πρόκειται να χρησιμοποιήσουμε για την επίλυση του προβλήματος και

γ) E είναι το ζητούμενο αποτέλεσμα.

1.4.2. Δημιουργία του αλγόριθμου

Μία σειρά οδηγιών που παρουσιάζονται σε μορφή βημάτων κατά την επίλυση ενός προβλήματος δημιουργούν τον αλγόριθμο. Τα περισσότερα προβλήματα

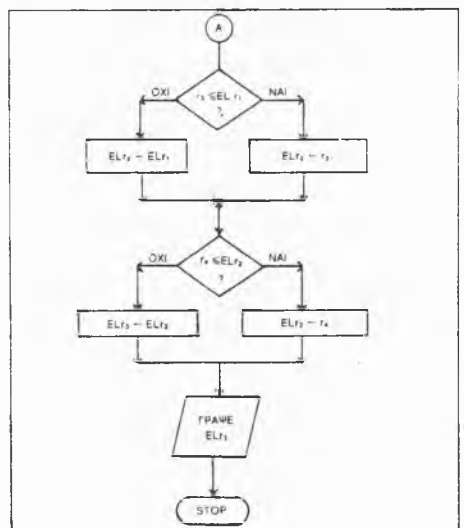
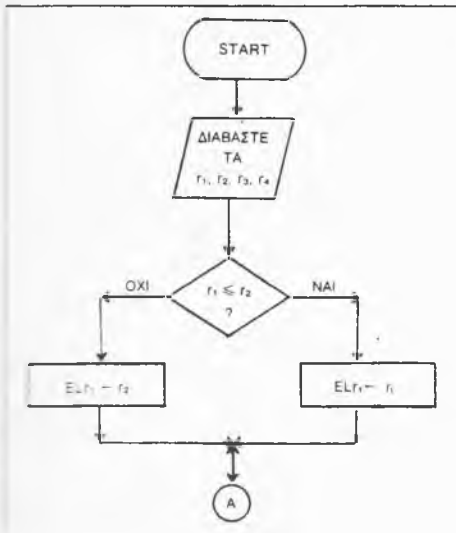
χύνονται με την χρήση αλγορίθμων απλών, σε μερικές περιπτώσεις περισσότερο συνθέτων κύρια σε προβλήματα επιστημονικά. Για παράδειγμα ο αλγόριθμος για τον προσδιορισμό του 631 ως πρώτου αριθμού είναι σχετικά απλός. Αντίθετα ο αλγόριθμος για τον υπολογισμό των στατικών ενός ουρανοξύστη είναι σύνθετος και περιέχει πολλές εκατοντάδες βήματα.

Η δημιουργία του αλγορίθμου είναι μια ολόκληρη τεχνική, γνωστή με το όνομα «Αλγοριθμική», και ασχολείται με τους τρόπους προσέγγισης ενός προβλήματος για τη λύση του με έναν υπολογιστή. Δεν θα ασχοληθούμε αναλυτικά με το θέμα αυτό, αφού άλλωστε αποτελεί ιδιαίτερο αντικείμενο μελέτης. Θέλουμε απλώς να επισημειώσουμε ότι η Αλγοριθμική είναι το κλειδί του καλού προγραμματισμού ανεξάρτητα από τη χρησιμοποιούμενη γλώσσα.

1.4.3. Σχηματική παρουσίαση του αλγορίθμου.

Ένας τρόπος παράστασης ενός αλγορίθμου είναι η μέθοδος της σχηματικής του παρουσίασης με τη βοήθεια του «Λογικού Διαγράμματος». Η μέθοδος αυτή παρουσιάζει, με τη χρήση διαφόρων σχημάτων, τη λογική σειρά που πρέπει να ακολουθήσουμε για την επίλυση ενός προβλήματος. Κανόνες για την κατασκευή ενός Λ.Δ. δεν υπάρχουν διότι ο σκοπός είναι υποβοηθητικός, κύρια σ' αυτούς τους λιγότερο έμπειρους στον προγραμματισμό. Τα χρησιμοποιούμενα σχήματα συνδέονται μεταξύ τους με βέλη που δείχνουν την πορεία της λύσης του προβλήματος. Τα Λ.Δ. διαβάζονται από πάνω προς τα κάτω, και από αριστερά προς τα δεξιά.

Το σχήμα που ακολουθεί παρουσιάζει τα βήματα καθώς και την σειρά εκτέλεσής τους, με τη βοήθεια Λ.Δ., ενός προβλήματος που βρίσκει και τυπώνει την μεγαλύτερη από τις 4 θερμοκρασίες $t_1, t_2, t_3,$ και t_4 αντίστοιχων πόλεων.



Υπάρχουν αρκετοί τύποι βημάτων που ακολουθούνται προκειμένου να λύσουμε ένα πρόβλημα, και σ' ένα λογικό διάγραμμα χρησιμοποιούνται διάφορα σχήματα για διαφορετικούς τύπους βημάτων. Αυτά τα σχήματα έχουν τυποποιηθεί από το American National Standards Institute (ANSI). Η πλήρης σειρά συμβόλων είναι πολύ μεγάλη αλλά θα χρησιμοποιήσουμε και θα παρουσιάσουμε τα συνηθέστερα που είναι τα παρακάτω:

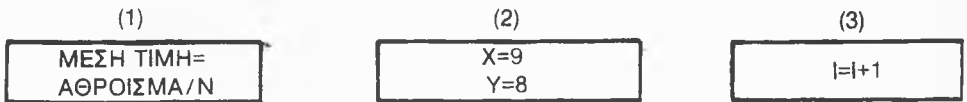
α) Το οβάλ σύμβολο χρησιμοποιείται για την αρχή και το τέλος ενός προγράμματος.



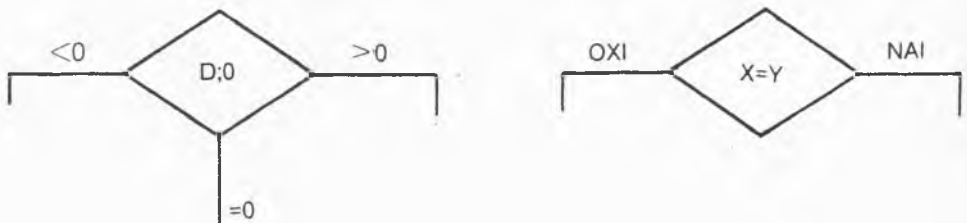
β) Το παραλληλόγραμμο χρησιμοποιείται για να διαβάσουμε τα δεδομένα και να τυπώσουμε τα ζητούμενα ενός προβλήματος.



γ) Το ορθογώνιο παραλληλόγραμμο χρησιμοποιείται για να δηλώσει μια ενδιάμεση πράξη ή κάποιο μαθηματικό τύπο π.χ.



δ) Ο ρόμβος χρησιμοποιείται για να δηλώσει την πορεία που θα ακολουθήσει μετά από τη λήψη κάποιας απόφασης.



ε) Ένας μικρός κύκλος χρησιμοποιείται όταν θέλουμε να συνδυάσουμε δύο τμήματα κάποιου λογικού διαγράμματος που τοποθετήσαμε σε διαφορετικές σελίδες επειδή είχαμε έλλειψη χώρου ή για να αποφύγουμε τις διακλαδώσεις βελών κατά μήκος ενός λογικού διαγράμματος.

Παράδειγμα: το λογικό διάγραμμα του προβλήματος του εμβαδού που αναφέραμε προηγούμενα θα μπορούσε να σχηματιστεί όπως παρακάτω.