

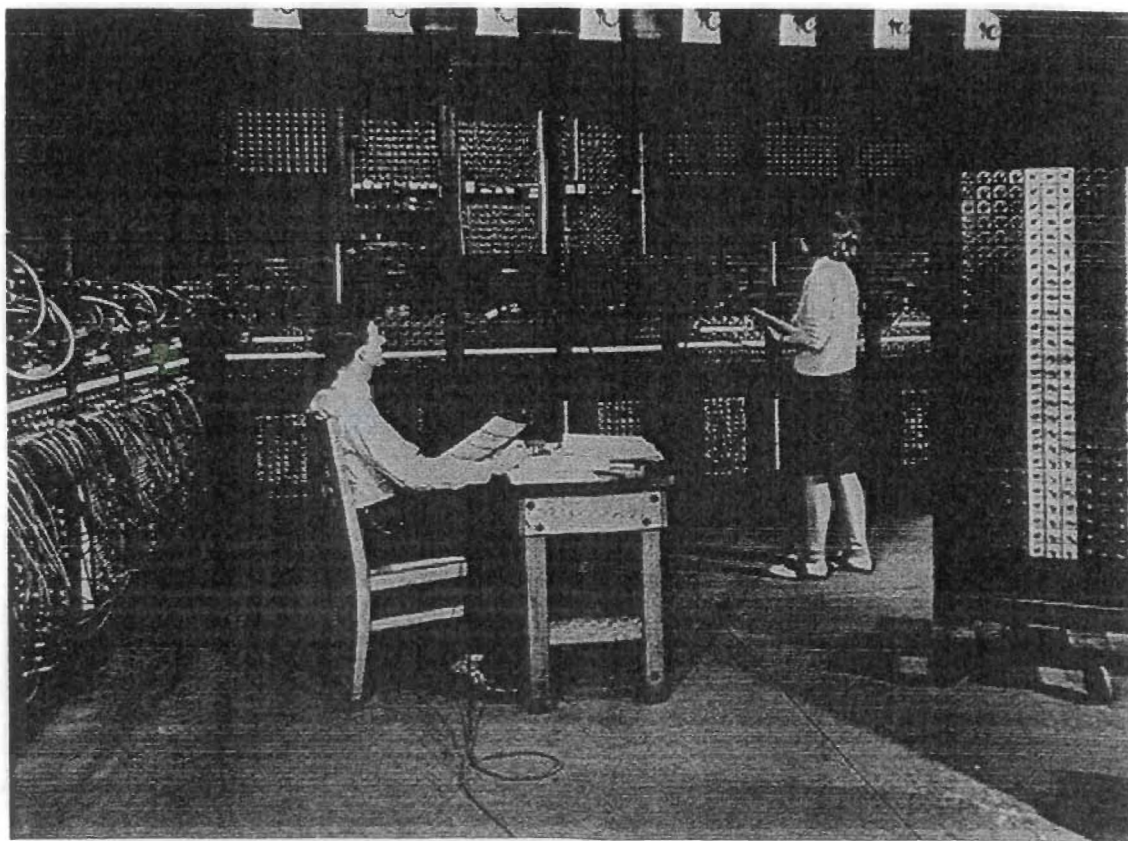


## ΤΕΙ ΜΕΣΟΛΟΓΓΙΟΥ

ΣΧΟΛΗ: ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ  
ΤΜΗΜΑ : ΕΦΑΡΜΟΓΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΗ  
ΔΙΟΙΚΗΣΗ ΚΑΙ ΣΤΗΝ ΟΙΚΟΝΟΜΙΑ

### ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ: Η ΙΣΤΟΡΙΑ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & Η ΕΞΕΛΙΞΗ ΤΗΣ  
ΜΕΧΡΙ ΚΑΙ ΤΙΣ ΗΜΕΡΕΣ ΜΑΣ



ΚΑΘΗΓΗΤΡΙΑ : Κα ΠΑΛΙΑΤΣΑ

ΟΝΟΜΑ: ΑΝΑΣΤΑΣΙΑ ΦΑΚΑ  
Α.Μ. 9971  
e-mail: natasa9971@yahoo.gr



**ΠΕΡΙΕΧΟΜΕΝΑ**

**Πρόλογος**

**Εισαγωγή: Μία σύντομη ιστορία της Πληροφορικής.**

**Μέρος Α : Από τον πρώτο Η/Υ στον σύγχρονο προσωπικό Η/Υ**

**Κεφάλαιο 1 :** Αναδρομή των υπολογιστικών συστημάτων

- 1.1 Μηχανές Πρώτης γενιάς
- 1.2.Μηχανές Δεύτερης γενιάς
- 1.3 Μηχανές Τρίτης γενιάς
- 1.4.Μηχανές Τέταρτης γενιάς
- 1.5 Μηχανές Πέμπτης γενιάς

**Κεφάλαιο 2 :** Δομή του προσωπικού υπολογιστή

- 2.1 Κεντρική μονάδα επεξεργασίας
- 2.2 Κεντρική μνήμη
- 2.3 Οργάνωση μνήμης
- 2.4 Μέτρηση της χωρητικότητας της μνήμης
- 2.5 Τοποθέτηση μνήμης-Μνήμες
- 2.6 Θύρες επικοινωνίας
- 2.7 Μητρική πλακέτα
- 2.8 Το κουτί της κεντρικής μονάδας

**Μέρος Β : Από την κωδικοποίηση της πληροφορικής στην ολοκληρωμένη αρχιτεκτονική των υπολογιστικών συστημάτων**

**Κεφάλαιο 3 :** Κωδικοποίηση της πληροφορικής, θεωρήματα και πράξεις για τα διάφορα συστήματα κωδικοποίησης.

- 3.1 Αριθμητικά συστήματα
  - 3.1.1 Συστήματα αρίθμησης - Σχέση bit και χαρακτήρα
- 3.2 Κωδικοποίηση της πληροφορικής
  - 3.2.1 Πληροφορία
- 3.3 Λογικά κυκλώματα
  - 3.3.1 Αποκωδικοποιητής και Πολυπλέκτης με πύλες
  - 3.3.2 Μεθοδολογία Αποσφαλμάτωσης Κυκλωμάτων

**Κεφάλαιο 4 :** Αρχιτεκτονική των υπολογιστικών συστημάτων

- 4.1 Κεντρική μνήμη
  - 4.1.1 Τρόποι διαχωρισμού μνημών
  - 4.1.2 Οργάνωση της Μνήμης
- 4.2 Αριθμητική λογική μονάδα
- 4.3 Καταχωριτές
- 4.4 Μετρητές
- 4.5 Μονάδα ελέγχου
- 4.6 Μνήμη συστήματος
  - 4.6.1 Μνήμη ανάγνωσης μόνο (ROM)
  - 4.6.2 Τυπική ROM
  - 4.6.3 Προγραμματιζόμενη μνήμη ανάγνωσης



- 4.6.4 Ελεγκτής μνήμης
- 4.6.5 Δίαυλος μνήμης
- 4.7 Μέγεθος μνήμης
  - 4.7.1 Μέγεθος μνήμης και απόδοση του συστήματος
  - 4.7.2 Πραγματική και Εικονική μνήμη
- 4.8 Φυσική και λογική οργάνωση μνήμης

## **Μέρος Γ: Προγραμματισμός Η/Υ και Λειτουργικά συστήματα (Windows XP)**

### **Κεφάλαιο 5 : Προγραμματισμός Η/Υ, γλώσσες και είδη προγραμματισμού**

- 5.1 : Προγραμματισμός
  - 5.1.1 Ορισμός Προγραμματισμού
  - 5.1.2 Γλώσσες Προγραμματισμού
  - 5.1.3 Ιεραρχικός προγραμματισμός
  - 5.1.4 Τμηματικός προγραμματισμός
- 5.2 : Γλώσσες Μηχανής
  - 5.2.1 Συμβολικές γλώσσες
  - 5.2.2 Γλώσσες υψηλού επιπέδου
  - 5.2.3 Γλώσσες τέταρτης γενιάς
- 5.3 : Είδη προγραμματισμού
  - 5.3.1 Διαδικασιακός
  - 5.3.2 Δομημένος
  - 5.3.3 Παράλληλος
  - 5.3.4 Αντικειμενοστραφής
  - 5.3.5 Συναρτησιακός
  - 5.3.6 Λογικός

### **Κεφάλαιο 6 : Σύγχρονα Λειτουργικά συστήματα – Windows XP**

- 6.1 Δίκτυα
  - 6.1.1 Γενική περιγραφή-Ορισμός δικτύου
  - 6.1.2 Διαχωρισμός δικτύων
  - 6.1.3 Είδη δικτύων
- 6.2 Εξυπηρετητές
- 6.3 Πρωτόκολλα επικοινωνίας
- 6.4 Σημαντικές υπηρεσίες του Internet
  - 6.4.1 Υπηρεσία του Παγκόσμιου Ιστού (WWW) και συνοπτική περιγραφή αυτού
  - 6.4.2 Υπηρεσία του Ηλεκτρονικού ταχυδρομείου
  - 6.4.3 Υπηρεσία Μεταφοράς αρχείων
  - 6.4.4 Υπηρεσία telnet
  - 6.4.5 Υπηρεσία συνομιλιών με άλλους χρήστες
  - 6.4.6 Υπηρεσία συζητήσεων
  - 6.4.7 Υπηρεσία Αναζήτησης πληροφοριών
- 6.5 : Internet και λειτουργία του
  - 6.5.1 Εταιρίες παροχής υπηρεσιών (ISP)
  - 6.5.2 Φυλλομετρητής
  - 6.5.3 Περιήγηση στο Web
- 6.6 Μηχανές Αναζήτησης
  - 6.6.1 Λειτουργία μιας μηχανής
  - 6.6.2 Κριτήρια ταξινόμησης αποτελεσμάτων σε μηχανές

αναζήτησης

6.7 Περιγραφή συγκεκριμένων πακέτων

6.7.1 Word

6.7.2 Excel

6.7.3 Access

6.7.4 PowerPoint

6.7.5 Outlook

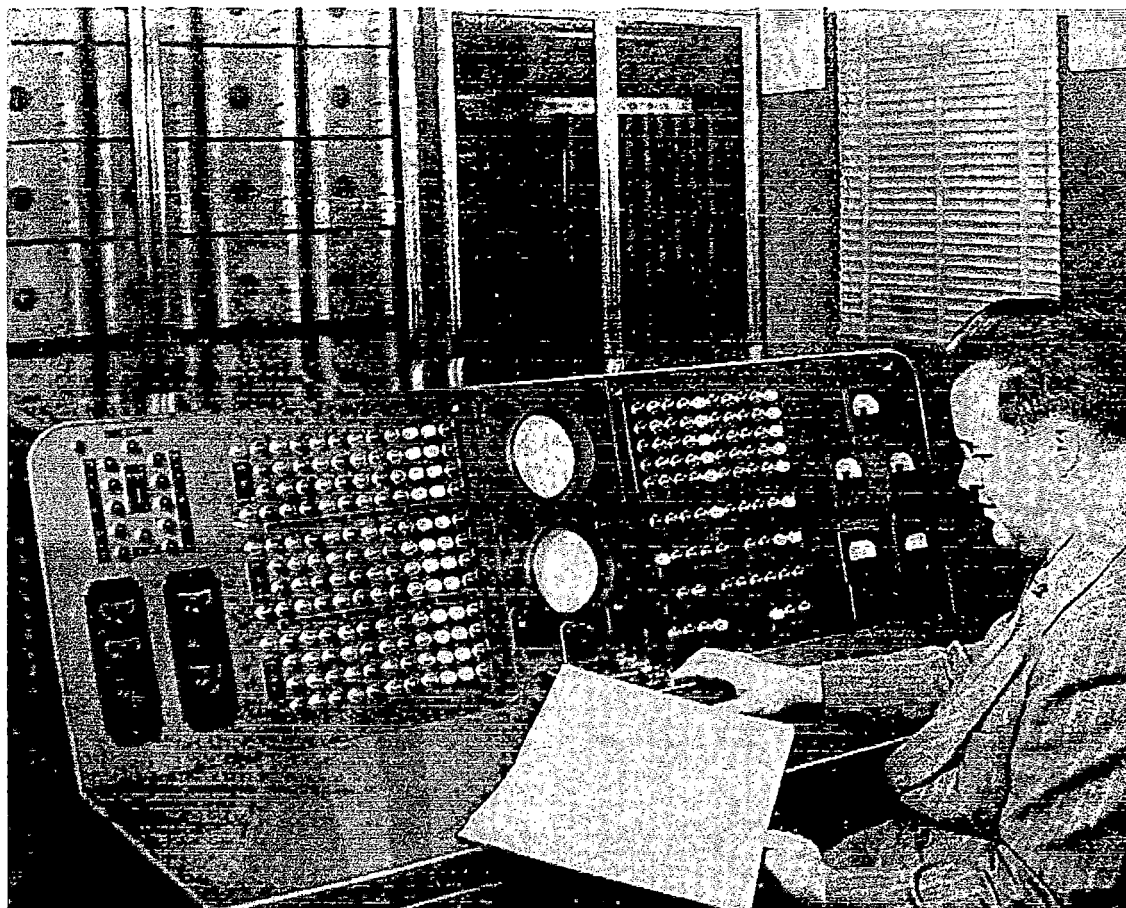
6.7.6 Frontpage

**Επίλογος**

---

## Πρόλογος

Παρά τους σημερινούς ξέφρενους και ιλιγγιώδεις ρυθμούς εξέλιξης στο χώρο της μικροηλεκτρονικής τεχνολογίας, η ανάπτυξη της πληροφορικής δεν ήταν "κεραυνός εν αιθρία", αλλά προϊόν μιας σταδιακής ιστορικής εξέλιξης αρκετών εκατονταετιών. Ας παρακολουθήσουμε λοιπόν χρονολογικά αυτή την εξέλιξη της πληροφορικής από την αρχαιότητα μέχρι σήμερα.



Εάν για τη δημοσιογραφία ισχύει το γνωμικό ότι "είδηση είναι το δάγκωμα ανθρώπου σε σκύλο και όχι το αντίθετο", στην ιστορία ισχύει ότι "ιστορικό γεγονός είναι αυτό που αποδεδειγμένα επηρέασε θετικά ή αρνητικά την εξέλιξη του ανθρώπινου πολιτισμού

## Εισαγωγή:

# Μια Σύντομη Ιστορία της Πληροφορικής

---

## *Πριν τα 1900*

Οι άνθρωποι χρησιμοποιούν μηχανικά εργαλεία για να κάνουν υπολογισμούς για χιλιάδες χρόνια. Για παράδειγμα, ο άβακας πιθανόν υπήρχε στην Βαβυλωνία (το σημερινό Ιράκ) περίπου το 3000 π.Χ. Οι αρχαίοι Έλληνες είχαν αναπτύξει κάποιους πολύ σύνθετους αναλογικούς υπολογιστές. Το 1901, ανακαλύφθηκε ένα αρχαίο Ελληνικό ναυάγιο ανοικτά της νήσου των Αντικυθήρων. Μέσα σ' αυτό βρέθηκε ένα όργανο που είχε διαβρωθεί από το αλάτι (ονομαζόμενο ο μηχανισμός των Αντικυθήρων) που αποτελείτο από σκουριασμένα μεταλλικά γρανάζια και δείκτες. Όταν αυτό το When, όργανο του 80 π.Χ. περίπου ανακατασκευάστηκε, παρήγαγε ένα μηχανισμό για την πρόβλεψη των κινήσεων των αστερών και των πλανητών.

Ο John Napier (1550-1617), ο Σκωτσέζος εφευρέτης των λογαρίθμων, ανακάλυψε τις ράβδους του Napier γύρω στο 1610 για να απλοποιήσει το έργο του πολλαπλασιασμού.

Το 1641 ο Γάλλος μαθηματικός και φιλόσοφος Blaise Pascal (1623-1662) κατασκεύασε μια μηχανή πρόσθεσης. Παρόμοια δουλειά είχε γίνει από τον Gottfried Wilhelm Leibniz (1646-1716). Ο Leibniz επίσης υποστήριζε την χρήση του δυαδικού συστήματος για να γίνονται σ' αυτό οι υπολογισμοί.

Πρόσφατα ανακαλύφθηκε ότι ο Wilhelm Schickard (1592-1635), απόφοιτος του Πανεπιστημίου του Tübingen (Γερμανία), είχε κατασκευάσει ένα τέτοιο όργανο το 1623-4, πριν κι από τον Pascal και τον Leibniz. Μια σύντομη περιγραφή του οργάνου αυτού περιέχεται σε δυο γράμματα προς τον Johannes Kepler. Δυστυχώς, ένα αντίγραφο της μηχανής αυτής κήκε σε μια πυρκαγιά.

Ο Charles Babbage (1791-1871) δούλεψε σε δυο μηχανικά όργανα: τη Μηχανή της Διαφοράς (Difference Engine) και τον πολύ περισσότερο φιλόδοξη Αναλυτική Μηχανή (έναν πρόδρομο του σύγχρονου ψηφιακού υπολογιστή), αλλά κανένα δεν δούλεψε ικανοποιητικά.

Η Ada Augusta Byron, Κοντέσα του Lovelace (1815-1852), μερικές φορές ονομάζεται ως ο "πρώτος προγραμματιστής" εξ αιτίας μιας αναφοράς που έγραψε στη μηχανή του Babbage. Από εκεί πήρε το όνομα η γλώσσα προγραμματισμού Ada.

Ο William Stanley Jevons (1835-1882), ένας Βρετανός οικονομολόγος και μελετητής της λογικής, έφτιαξε μια μηχανή το 1869 για να λύσει προβλήματα

λογικής. Ήταν "η πρώτη τέτοια μηχανή με επαρκή ισχύ να λύσει ένα πολύπλοκο πρόβλημα γρηγορότερα από όσο το πρόβλημα θα μπορούσε να λυθεί χωρίς την βοήθεια της μηχανής." (Gardner) Η μηχανή αυτή βρίσκεται στο Μουσείο Ιστορίας της Επιστήμης στην Οξφόρδη.

Ο Herman Hollerith (1860-1929) ανεκάλυψε τη σύγχρονη διάτρητη κάρτα, την οποία χρησιμοποίησε σε μια μηχανή που σχεδίασε για να τον βοηθήσει να βάλει σε πίνακες τα αποτελέσματα της απογραφής του 1890.

---

### ***1900 - 1939: Η Άνοδος των Μαθηματικών***

Η δουλειά στις μηχανές που κάνουν υπολογισμούς συνεχίστηκε. Κάποιες υπολογιστικές μηχανές ειδικών σκοπών κατασκευάστηκαν. Για παράδειγμα, το 1919, ο E.O. Carissan (1880-1925), αξιωματικός του Γαλλικού πεζικού, σχεδίασε και κατασκεύασε μια εξαιρετική μηχανική συσκευή για την παραγοντοποίηση ακεραίων και τον έλεγχο του αν είναι πρώτοι. Ο Ισπανός Leonardo Torres y Quevedo (1852-1936) κατασκεύασε κάποιες ηλεκτρομηχανικές υπολογιστικές συσκευές, περιλαμβανομένης μιας που έπαιζε απλά παιχνίδια σκακιού.

Το 1928, ο Γερμανός μαθηματικός David Hilbert (1862-1943) έδωσε μια σημαντική ομιλία στο Διεθνές Συνέδριο των Μαθηματικών. Έθεσε τρία ερωτήματα: (1) Είναι τα μαθηματικά πλήρη, δηλαδή, μπορεί κάθε μαθηματική πρόταση είτε να αποδειχθεί είτε να διαψευσθεί; (2) Είναι τα μαθηματικά συνεπή, δηλαδή, αληθεύει ότι προτάσεις όπως " $0 = 1$ " δεν μπορούν να αποδειχθούν με ισχύουσες μεθόδους; (3) Είναι τα μαθηματικά δυνατά να αποφασισθούν (decidable), δηλαδή, υπάρχει μια αυτόματη μέθοδος που μπορεί να εφαρμοσθεί σε οποιαδήποτε μαθηματική πρόταση και (τουλάχιστον κατ' αρχήν) να μπορεί να αποφανθεί τελικά αν η πρόταση αυτή είναι αληθής ή όχι; Αυτό το τελευταίο ερώτημα ονομάστηκε το *Entscheidungsproblem*.

Το 1931, ο Kurt Gödel (1906-1978) απάντησε δυο από τα ερωτήματα του Hilbert. Έδειξε ότι κάθε αρκετά ισχυρό τυπικό σύστημα είναι είτε μη συνεπές ή μη πλήρες. Επίσης, αν ένα αξιωματικό σύστημα είναι συνεπές, η συνέπεια αυτή δεν μπορεί να αποδειχθεί μέσα στο σύστημα αυτό. Το τρίτο ερώτημα παρέμεινε ανοικτό, με την αντικατάσταση του 'αληθούς' από το 'αποδείξιμο.'

Το 1936, ο Alan Turing (1912-1954) έδωσε μια λύση στο Entscheidungsproblem του Hilbert κατασκευάζοντας ένα τυπικό σύστημα ενός υπολογιστή -- και δείχνοντας ότι υπάρχουν προβλήματα που ούτε μια μηχανή μπορούσε να λύσει. Ένα τέτοιο πρόβλημα είναι το ονομαζόμενο "πρόβλημα της περάτωσης" (halting problem): δοθέντος ενός προγράμματος του Pascal, περατώνεται με κάθε είσοδο (input);

---

## ***Η δεκαετία 1940: Τα χρόνια του πολέμου φέρνουν την γένεση του ηλεκτρονικού ψηφιακού υπολογιστή***

Οι υπολογισμοί που απαιτούντο κατά την διάρκεια του Δεύτερου Παγκόσμιου Πολέμου προκάλεσαν την ανάπτυξη του ηλεκτρονικού ψηφιακού υπολογιστή γενικής χρήσης. Στο Harvard, ο Howard H. Aiken (1900-1973) κατασκεύασε τον ηλεκτρομηχανικό υπολογιστή Mark I το 1944, με την βοήθεια της IBM.

Επίσης η αποκρυπτογράφηση των στρατιωτικών κωδίκων οδήγησε σε υπολογιστικά πρότζεκτ. Ο Alan Turing ήταν ανακατεμένος στο σπάσιμο του κώδικα της Γερμανικής μηχανής, το Enigma, στο Bletchley Park στην Αγγλία. Οι Βρετανοί έφτιαξαν μια υπολογιστική μηχανή, το Colossus, για να βοηθήσουν το σπάσιμο των κωδίκων.

Στο Πανεπιστήμιο Iowa State το 1939, ο John Vincent Atanasoff (1904-1995) και ο Clifford Berry σχεδίασαν και κατασκεύασαν έναν ηλεκτρονικό υπολογιστή για την επίλυση συστημάτων γραμμικών εξισώσεων, που όμως ποτέ δεν δούλεψε ικανοποιητικά.

Ο Atanasoff συζήτησε την ανακάλυψή του με τον John William Mauchly (1907-1980), που αργότερα, μαζί με τον J. Presper Eckert, Jr. (1919-1995), σχεδίασαν και κατασκεύασαν τον ENIAC, έναν ηλεκτρονικό υπολογιστή γενικής χρήσης αρχικά προορισμένο για υπολογισμούς του πυροβολικού. (Ακριβώς ποιές ιδέες ο Mauchly πήρε από τον Atanasoff δεν είναι πλήρως αποσαφηνισμένο και το αν ο Atanasoff ή ο Mauchly κι ο Eckert αξίζουν την αναγνώριση ως οι δημιουργοί του ηλεκτρονικού ψηφιακού υπολογιστή απετέλεσα το θέμα νομικών διαμαχών και συνεχιζόμενων ιστορικών διαφωνιών.) Ο ENIAC άρχισε να κατασκευάζεται στη Σχολή Moore του Πανεπιστημίου της Pennsylvania και τέλειωσε το 1946.

Το 1944, οι Mauchly, Eckert και ο John von Neumann (1903-1957) ήδη δούλευαν για το σχεδιασμό ενός ηλεκτρονικού υπολογιστή αποθηκευμένου-προγράμματος, του EDVAC. Η αναφορά του von Neumann, "First Draft of a Report on the EDVAC", είχε μεγάλη επιρροή και περιείχε πολλές από τις ιδέες που ακόμη χρησιμοποιούνται στους περισσότερους σύγχρονους ψηφιακούς υπολογιστές. Οι Eckert και Mauchly συνέχισαν με την κατασκευή του UNIVAC.

Στο μεταξύ, στην Γερμανία, ο Konrad Zuse (1910-1995) είχε κατασκευάσει τον πρώτο λειτουργικό, γενικής-χρήσης, ελεγχόμενο-με-πρόγραμμα υπολογιστή, τον Z3, το 1941.

Το 1945, ο Vannevar Bush δημοσίευσε ένα εκπληκτικά προφητικό άρθρο στο περιοδικό *Atlantic Monthly* για τους τρόπους που η επεξεργασία της πληροφορίας θα επηρέαζε την κοινωνία του μέλλοντος.

Ο Maurice Wilkes, εργαζόμενος στο Cambridge της Αγγλίας, έφτιαξε τον EDSAC, έναν υπολογιστή βασισμένο στον EDVAC. Ο F. C. Williams και άλλοι στο Πανεπιστήμιο του Manchester κατασκεύασαν τον υπολογιστή Manchester Mark I, μια εκδοχή του οποίου λειτουργούσε τόσο παλιά όσο τον Ιούνιο του 1948. Μερικές φορές η μηχανή αυτή θεωρείται ως ο πρώτος ψηφιακός υπολογιστής με αποθηκευμένα προγράμματα.



Η ανακάλυψη του τρανζίστορ το 1947 από τους John Bardeen (1908-1991), Walter Brattain (1902-1987) και William Shockley (1910-1989) μετασημάτισε τον υπολογιστή και καθίστησε δυνατή την επανάσταση των μικρο-επεξεργαστών. Για την ανακάλυψή τους αυτή κέρδισαν το Βραβείο Nobel στην φυσική το 1956.

Ο Jay Forrester (γεννηθείς το 1918) ανακάλυψε τη μαγνητική μνήμη γύρω στο 1949.

---

### **Η δεκαετία του 1950**

Ο Grace Murray Hopper (1906-1992) ανακάλυψε την έννοια του μεταγλωττιστή (compiler), στο Remington Rand το 1951. Νωρίτερα το 1947, ο Hopper βρήκε το πρώτο "bug" (κοριό ή λάθος) του υπολογιστή -- ένα πραγματικό κοριό -- ένα έντομο που είχε μπει μέσα στον υπολογιστή Harvard Mark II. (Στην πραγματικότητα, το "bug-κοριός" είχε χρησιμοποιηθεί να σημαίνει "ελάττωμα" τουλάχιστον από το 1889.)

Ο John Backus και άλλοι ανέπτυξαν τον πρώτο μεταγλωττιστή (compiler) της FORTRAN τον Απρίλιο του 1957. Η LISP, μια γλώσσα επεξεργασίας καταλόγων για προγραμματισμό τεχνητής νοημοσύνης, ανακαλύφθηκε από τον John McCarthy γύρω στο 1958. Οι Alan Perlis, John Backus, Peter Naur και άλλοι ανέπτυξαν την Algol.

Ως προς το υλικό (hardware) του υπολογιστή, ο Jack Kilby (των Texas Instruments) κι ο Robert Noyce (της Fairchild Semiconductor) ανακάλυψαν το ολοκληρωμένο κύκλωμα το 1959.

Ο Edsger Dijkstra ανακάλυψε έναν αποτελεσματικό αλγόριθμο των συντομότερων δρόμων σε γράφους σαν μια επίδειξη των δυνατοτήτων του υπολογιστή ARMAC το 1956. Επίσης ανακάλυψε έναν αποτελεσματικό αλγόριθμο για το ελάχιστο εκτεταμένο δένδρο για να ελαχιστοποιήσει την καλωδίωση που απαιτείται για τον υπολογιστή X1

Σε μια φημισμένη εργασία που εμφανίστηκε στο περιοδικό *Mind* το 1950, ο Alan Turing εισήγαγε το Τεστ του Turing, μια από τις πρώτες προσπάθειες στο πεδίο της τεχνητής νοημοσύνης. Πρότεινε έναν ορισμό της "σκέψης" ή της "συνείδησης" χρησιμοποιώντας ένα παιχνίδι: αυτός που κάνει το τεστ πρέπει να αποφασίσει, στη βάση γραπτού διαλόγου, αν το ον στο διπλανό δωμάτιο που απαντά στις ερωτήσεις του είναι άνθρωπος ή υπολογιστής. Αν αυτή η διάκριση δεν μπορεί να γίνει, τότε θα μπορούσε να πει κανείς δικαιολογημένα ότι ο υπολογιστής "σκέπτεται."

---

### **Η δεκαετία του 1960**

Στην δεκαετία του 1960, η επιστήμη των υπολογιστών ή πληροφορική αναδείχθηκε σαν ένας ξεχωριστός κλάδος. Πράγματι, ο όρος (επιστήμη των υπολογιστών) κατασκευάστηκε από τον George Forsythe, έναν αριθμητικο-

αναλύστα. Το πρώτο πανεπιστημιακό τμήμα της επιστήμης των υπολογιστών έγινε το 1962 στο Πανεπιστήμιο Purdue.

Τα λειτουργικά συστήματα είχαν μεγάλες προόδους. Ο Fred Brooks στην IBM σχεδίασε το System/360, μια οικογένεια διαφορετικών υπολογιστών με την ίδια αρχιτεκτονική και τους ίδιους κανόνες λειτουργίας, από τη μικρή μηχανή ως την μεγαλύτερη στην οικογένεια αυτή. Ο Edsger Dijkstra στο Eindhoven σχεδίασε το σύστημα πολυπρογραμματισμού THE.

Στο τέλος της δεκαετίας αυτής, το ARPAnet, ο πρόδρομος του σημερινού Internet, άρχισε να λειτουργεί.

Πολλές νέες γλώσσες προγραμματισμού ανακαλύφθηκαν, όπως η BASIC (που αναπτύχθηκε γύρω στο 1964 από τον John Kemeny (1926-1992) και τον Thomas Kurtz).

Στην δεκαετία του 1960 ήρθε επίσης στο φως η θεωρία των αυτομάτων και των τυπικών γλωσσών. Τα μεγάλα ονόματα εδώ περιλαμβάνουν τους Noam Chomsky και Michael Rabin. Ο Chomsky έγινε αργότερα πολύ γνωστός για την θεωρία του ότι η γλώσσα αποτελεί το μηχανικό μέρος της ανθρώπινης νόησης και για τις επικρίσεις του στην Αμερικανική εξωτερική πολιτική.

Η απόδειξη της ορθότητας των προγραμμάτων μέσω τυπικών μεθόδων άρχισε επίσης να γίνεται πιο σημαντική την δεκαετία αυτή. Η δουλειά του Tony Hoare έπαιξε ένα σημαντικό ρόλο. Ο Hoare ανακάλυψε επίσης την Quicksort.

Ο Ted Hoff και ο Federico Faggin της Intel σχεδίασαν τον πρώτο μικρο-επεξεργαστή (υπολογιστή σ' ένα chip) κατά το 1969-1971.

Μια αυστηρή μαθηματική βάση για την ανάλυση των αλγορίθμων άρχισε με τον Donald Knuth, συγγραφέα ενός 3-τομου έργου με την ονομασία. *Η Τέχνη του Προγραμματισμού Υπολογιστών (The art of Computer Programming)*.

---

### ***Η δεκαετία του 1970***

Η θεωρία των βάσεων δεδομένων είδε την κύρια πρόοδό της με την δουλειά του Edgar F. Codd σε συσχετικιστικές βάσεις δεδομένων. Ο Codd κέρδισε το βραβείο Turing το 1981.

Το Unix, ένα πολύ σημαντικό λειτουργικό σύστημα, αναπτύχθηκε στα Εργαστήρια Bell από τον Ken Thompson και τον Dennis Ritchie. Μαζί οι Brian Kernighan και Ritchie ανέπτυξαν την C, μια σημαντική γλώσσα προγραμματισμού.

Άλλες νέες γλώσσες προγραμματισμού, όπως η Pascal (που ανακαλύφθηκε από τον Niklaus Wirth) και η Ada (που αναπτύχθηκε από μια ομάδα υπό την καθοδήγηση του Jean Ichbiah), εμφανίσθηκαν την δεκαετία αυτή.

Η πρώτη αρχιτεκτονική RISC άρχισε από τον John Cocke το 1975, στα Εργαστήρια Thomas J. Watson της IBM. Παρόμοια πρότζεκτ ξεκίνησαν στο Berkeley και το Stanford γύρω στο ίδιο διάστημα.

Στην δεκαετία του 1970 είχαμε επίσης την εμφάνιση των υπερυπολογιστών (supercomputers). Ο Seymour Cray σχεδίασε τον CRAY -1, που βγήκε για πρώτη φορά το Μάρτιο του 1976. Μπορούσε να κάνει 160 εκατομμύρια πράξεις το δευτερόλεπτο.

Επιπλέον, υπήρξαν σημαντικές πρόοδοι στους αλγορίθμους και την υπολογιστική πολυπλοκότητα. Το 1971, ο Steve Cook δημοσίευσε την θεμελιώδη εργασία του στην πληρότητα-NP και μετά από λίγο ο Richard Karp έδειξε ότι πολλά φυσικά συνδυαστικά προβλήματα είναι πλήρη-NP. Οι Whit Diffie και Martin Hellman δημοσίευσαν μια εργασία που εισήγαγε στην κρυπτογραφία δημοσίων-κλειδιών κι ένα κρυπτοσύστημα δημοσίου-κλειδιού γνωστό ως RSA ανακαλύφθηκε από τους Ronald Rivest, Adi Shamir και Leonard Adleman.

Το 1979, τρεις μεταπτυχιακοί φοιτητές στη Βόρεια Carolina ανέπτυξαν έναν εξυπηρετητή (server) κατανεμημένων νέων που τελικά εξελίχθηκε στο Usenet.

---

### ***Η δεκαετία 1980***

Στην δεκαετία αυτή είχαμε την εμφάνιση του προσωπικού υπολογιστή από τους Steve Wozniak και Steve Jobs, ιδρυτές της Apple Computer.

Οι πρώτοι ιοί υπολογιστών εμφανίστηκαν γύρω στο 1981. Ο όρος οφείλεται στον Leonard Adleman.

Το 1981, ο πρώτος πραγματικά πετυχημένος φορητός υπολογιστής βγήκε στην αγορά, ο Osborne I. Το 1984, η Apple πρωτο-έβγαλε τον υπολογιστή Macintosh.

Το 1987, το Εθνικό Ίδρυμα Επιστημών (National Science Foundation ή NSF) των ΗΠΑ εγγαίωσε το NSFnet, πρόδρομο μέρους του σημερινού Internet.

---

### ***Η δεκαετία του 1990 και μετά***

Οι παράλληλοι υπολογιστές συνεχίζουν να αναπτύσσονται.

Η βιολογική πληροφορική, με τη πρόσφατη δουλειά του Len Adleman σε υπολογισμούς μέσω DNA, υπόσχεται πολλά. Το Πρότζεκτ του Ανθρώπινου Γονιδιόματος (Human Genome Project) επιχειρεί να καταγράψει όλο το DNA του ανθρώπου.

Η κβαντική πληροφορική παίρνει μια ιδιαίτερη ώθηση με την ανακάλυψη του Peter Shor για την παραγοντοποίηση των ακεραίων που μπορεί να εκτελεσθεί αποτελεσματικά σ' έναν (θεωρητικό) κβαντικό υπολογιστή.

Οι "Λεωφόροι των Πληροφοριών" (Information Superhighway) συνδέουν όλο και περισσότερους υπολογιστές σ' όλο τον κόσμο.

Οι υπολογιστές γίνονται όλο και πιο μικροί και γεννάται η νανο-τεχνολογία.

---

Γραμμένη από τον Jeffrey Shallit για το μάθημα CS 134 του Πανεπιστημίου του Waterloo το καλοκαίρι του 1995.

## **Μέρος Α : Από τον πρώτο Η/Υ στον σύγχρονο προσωπικό υπολογιστή**

### Κεφάλαιο 1: Ιστορική αναδρομή των υπολογιστικών συστημάτων

#### **1.1 Μηχανές πρώτης γενεάς (1951-1958)**

Από τον Ιούνιο του 1951 και μέχρι το 1958 κατασκευάζονται οι υπολογιστικές μηχανές της Πρώτης Γενεάς, που χαρακτηρίζονται από τη χρησιμοποίηση λυχνιών κενού στην κατασκευή των κυκλωμάτων τους. Πρώτη υπολογιστική μηχανή της κατηγορίας αυτής ήταν η UNIVAC I, (UNIVersal Automatic Computer), που κατασκευάστηκε από την ίδια ομάδα ατόμων που συμμετείχαν στην κατασκευή της μηχανής ENIAC.

Η κεντρική τους μνήμη χρησιμοποιεί μαγνητικά τύμπανα. Στην περιφερειακή, η δευτερεύουσα μνήμη χρησιμοποιεί επίσης μαγνητικά τύμπανα, ή μαγνητικές ταινίες. Στη θέση της μονάδας εισόδου έχουν συσκευές ανάγνωσης διάτρητων καρτών, ή χαρτοταινίες και στη θέση της μονάδας εξόδου συσκευές διάτρησης καρτών ή εκτυπωτικές συσκευές. Η μηχανή UNIVAC I ήταν η πρώτη ηλεκτρονική υπολογιστική μηχανή, που είχε "μαζική" παραγωγή και πώληση. Πωλούνταν από την εταιρεία Remington Rand και η πρώτη πώληση έγινε για λογαριασμό του Γραφείου Απογραφών των Η.Π.Α. Χαρακτηριστικό είναι ότι την εποχή εκείνη, για το ευρύ κοινό, η λέξη "univac" υπονοούσε "υπολογιστική μηχανή".

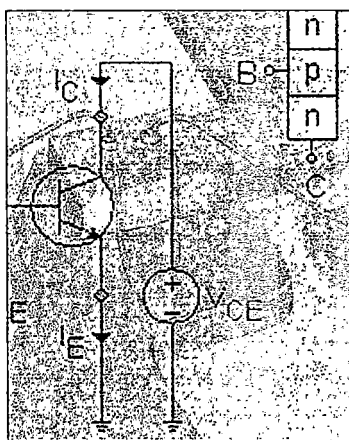
Η εταιρία IBM (International Business Machines) ήταν, κατά την περίοδο αυτή, μια μικρή σχετικά εταιρία, που δεν είχε ενδιαφέροντα στην κατασκευή υπολογιστικών μηχανών. Από το 1953 όμως, ασχολήθηκε με την κατασκευή

υπολογιστικών μηχανών και άρχισε με τη μηχανή IBM-701, που ήταν μια απλή προγραμματιζόμενη μηχανή.

Άλλες μηχανές της σειράς αυτής, όπως η IBM-305 RAMAC και η πολύ επιτυχημένη IBM-650 ακολούθησαν, κθιάρχησαν για μια ολόκληρη πενταετία. Η τελευταία μηχανή της οικογένειας αυτής ήταν η IBM-709, που κατασκευάστηκε το 1958. Όλες οι μηχανές της σειράς IBM κατασκευάζονται με λυχνίες κενού, (χαρακτηριστικό της γενεάς αυτής). Η μηχανή IBM-709 πάντως, είναι η τελευταία, που κατασκευάζει η IBM, με λυχνίες κενού. Η μηχανή IBM-305 RAMAC, είναι η πρώτη μηχανή εφοδιασμένη με σύστημα δίσκου.

## 1.2 Μηχανές Δεύτερης Γενεάς (1959-1964)

Το έτος 1948 πραγματοποιείται, στα εργαστήρια της εταιρείας Bell, η ανακάλυψη της τριόδου ηλεκτρονικής λυχνίας, γνωστής ως τρανζίστορ ( transistor). Ως γνωστό, οι κατασκευαστές του τρανζίστορ ήταν οι John Bardeen, Walter Brattain και William Shockley, στους οποίους το 1956 απενεμήθη, για το λόγο αυτό, το βραβείο Nobel. Το τρανζίστορ είναι περισσότερο αξιόπιστο, κατά πολύ πιο φθηνό, έχει πολύ χαμηλές απαιτήσεις τροφοδοσίας, παράγει χαμηλή θερμότητα κατά την λειτουργία του και είναι κατά πολύ μικρότερο από τις πιο μικρές λυχνίες κενού.



Τρανζίστορ και οι λειτουργίες του

Μετά την ανακάλυψη του τρανζίστορ αρχίζει η ουσιαστική παραγωγή των υπολογιστικών μηχανών της Δεύτερης Γενεάς (1959-1964). Οι μηχανές αυτές έχουν ως κύριο χαρακτηριστικό τη χρησιμοποίηση του τρανζίστορ στα κυκλώματα της κεντρικής μνήμης, αντί των μαγνητικών δακτυλίων. Οι υπόλοιπες μονάδες τους έχουν την ίδια τεχνολογία με αυτή της πρώτης γενεάς.

Η πρώτη ηλεκτρονική υπολογιστική μηχανή με τρανζίστορ κατασκευάστηκε στο MIT και ονομάστηκε TX-O (Transistorized experiment computer O). Σκοπός της κατασκευής της ήταν ο έλεγχος της μηχανής TX-2, που προοριζόταν για εκμετάλλευση. Τελικά η μηχανή TX-2 ουδέποτε ολοκληρώθηκε. Το 1957, ένας από

τους μηχανικούς που συνεργάστηκαν στην κατασκευή της μηχανής TX-2, ιδρύει την εταιρεία Digital Equipment Corporation (DEC). Η εταιρεία DEC κατασκευάζει αρχικά μια μηχανή παρόμοια με την TX-0, ενώ το έτος 1961 κατασκευάζει την πολύ αξιόλογη μηχανή PDP-1, πρώτη στη σειρά των γνωστών PDP μηχανών. Η μηχανή PDP-1 κόστιζε την εποχή εκείνη 120,000 δολάρια.

Η μηχανή PDP-1 ήταν ο πρώτος υπολογιστής κατηγορίας mini, που κατασκευάστηκε και επιπλέον είχε σημαντικές πωλήσεις. Μετά την κατασκευή της μηχανής PDP-1, ένα πλήθος καινοτομιών άρχισαν να εμφανίζονται στο χώρο των υπολογιστικών μηχανών. Ανάμεσα στις καινοτομίες αυτές, είναι και η δημιουργία οθονών σε μορφή παρόμοια με αυτή των σημερινών.

Η IBM κατασκεύασε, εν τω μεταξύ, τη μηχανή IBM-7094, που διαδέχτηκε την IBM-709 και περιείχε τρανζίστορ αντί λυχνιών κενού. Η μηχανή IBM-7094 ήταν η ταχύτερη μηχανή, κατά την εποχή εκείνη και το κόστος της μερικά εκατομμύρια δολάρια. Από τη στιγμή που οι μηχανές άρχισαν να πωλούνται και η κατασκευή τους να έχει εμπορικό ενδιαφέρον, τόσο οι εταιρείες DEC και IBM, όσο και άλλες εταιρείες όπως οι Control Data Corporation και η Univac αλλά και πολλές ακόμη, άρχισαν να κατασκευάζουν και να πωλούν ηλεκτρονικές υπολογιστικές μηχανές.

### 1.3 Μηχανές Τρίτης Γενεάς (1965-1970)

Η Τρίτη Γενεά (1965-1970) χαρακτηρίζεται από τη χρησιμοποίηση ολοκληρωμένων κυκλωμάτων μεγάλης ολοκλήρωσης (LSI-Large Scale Intergration) στα κυκλώματα των μηχανών. Χαρακτηρίζεται επίσης από την κατασκευή κεντρικών μνημών με μαγνητικούς δακτυλίους. Τα ολοκληρωμένα κυκλώματα επινοήθηκαν από τον Jack Kilby το έτος 1958, στα εργαστήρια της εταιρείας Texas Instruments. Ένα σύνηθες ολοκληρωμένο κύκλωμα αποτελείται από ένα μικρό ορθογώνιο παραλληλεπίπεδο, με διάφορες διαστάσεις. Είναι συνηθισμένη αυτή με 25,4 χιλιοστά μήκος, 0,5 χιλιοστά πλάτος και 0,125 χιλιοστά πάχος. Στο εσωτερικό του περιέχεται ένα μονολιθικό κύκλωμα πυριτίου (τσιπ) με πολλά αλληλοσυνδεδεμένα τρανζίστορ και άλλα στοιχεία. Τα πρώτα ολοκληρωμένα κυκλώματα (Intergrated Circuits-IC) περιείχαν περιορισμένο πλήθος στοιχείων, ενώ τα μεταγενέστερα διαρκώς αυξανόμενο πλήθος στοιχείων.

Η τεχνολογία των μαγνητικών δίσκων έχει αναπτυχθεί και αντικαθιστά τις μονάδες μαγνητικής ταινίας, οι οποίες χρησιμοποιούνται πλέον μόνο για διαδικασίες back up. Οι τερματικές συσκευές με πληκτρολόγιο και οθόνη αντικαθιστούν σταδιακά τις μονάδες διάτρησης καρτών και χρησιμοποιούνται ως μηχανισμοί εισόδου και εξόδου αντίστοιχα.

Η τρίτη γενιά είναι δυνατό να υποστηριχθεί ότι χαρακτηρίζεται από την κυριαρχία των μηχανών της IBM και ιδιαίτερα των μηχανών τύπου IBM System/360. Η οικογένεια των μηχανών System/360 περιλάμβανε μια μικρή μηχανή, τη

System/360-30, με κύρια χρήση τις εμπορικές εφαρμογές, τη μηχανή System/360-65, ένα ισχυρότερο τύπο μηχανής, καθώς και άλλες μηχανές για εμπορικές και επιστημονικές εφαρμογές. Στις τελευταίες περιλαμβάνονται οι μηχανές όπως οι System/360-40, System/360-50 και System/360-75.

Η οικογένεια των μηχανών της IBM System/360 χρησιμοποιούσε ολοκληρωμένα κυκλώματα. Η μηχανή αυτή περιλάμβανε έξι διαφορετικά μοντέλα, τα οποία ήταν συμβατά μεταξύ τους και είχαν διαδοχικά αυξανόμενη υπολογιστική ισχύ, από το χαμηλότερο μοντέλο προς το υψηλότερο. Το γεγονός αυτό ήταν πολύ σημαντικό για τις εταιρείες, που χρησιμοποιούσαν μοντέλα μηχανών IBM System/360. Η εξήγηση είναι απλή.

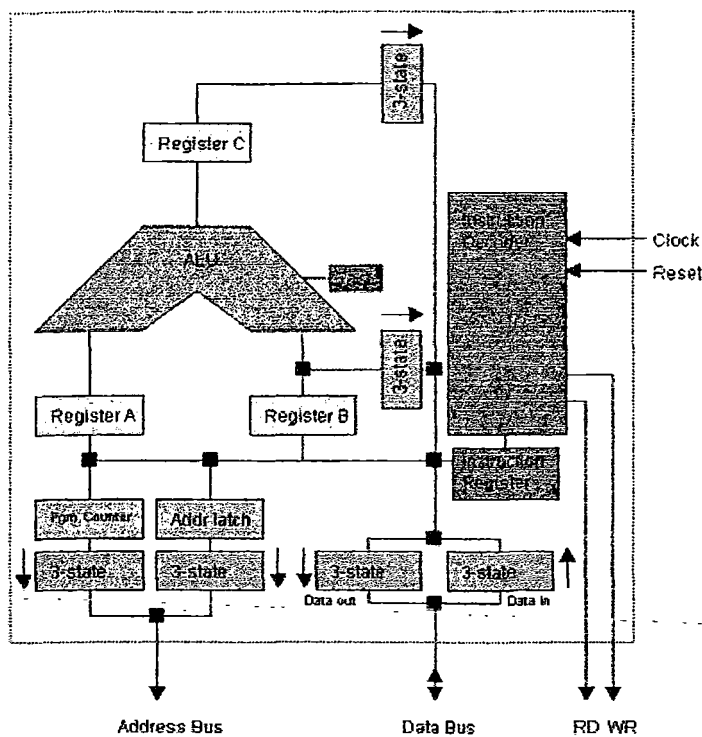
Οι εταιρείες είχαν τη δυνατότητα να καλύπτουν σταδιακά τις αυξανόμενες υπολογιστικές τους απαιτήσεις, με απλή αντικατάσταση του μοντέλου που χρησιμοποιούσαν, με ένα ισχυρότερο, χωρίς να απαιτείται παράλληλη αντικατάσταση ολόκληρου του εξοπλισμού τους και των προγραμμάτων τους. Το έτος 1970 η IBM άρχισε να κατασκευάζει μια νέα σειρά υπολογιστικών συστημάτων, που έγινε γνωστή με το όνομα System/370 και παρουσίασε επίσης πολύ μεγάλη επιτυχία.

Υποστηρίζεται, πάντως, ότι η τρίτη γενιά των υπολογιστικών συστημάτων διαρκεί μέχρι το τέλος της δεκαετίας του 1970 (1965-1980) και περιλαμβάνει την εξέλιξη των μηχανών System/360 της IBM. Γενικά, η τεχνολογία προχώρησε τόσο γρήγορα, ώστε σήμερα πλέον να μην είναι εύκολο να προσδιορισθεί ακριβώς το τέλος της τρίτης γενιάς και η αρχή της τέταρτης. Υπάρχουν πολλά νέα τεχνολογικά επιτεύγματα που θα ήταν δυνατό να χρησιμοποιηθούν ως κριτήρια και να προσδιορίσουν το τέλος της μιας γενιάς και την αρχή της άλλης, όπως το μέγεθος της ολοκλήρωσης, οι δυνατότητες των μηχανών σε ταχύτητα και μνήμη, τα προγράμματα λειτουργίας των υπολογιστικών συστημάτων και άλλα.

Η εποχή των υπολογιστικών συστημάτων, που χαρακτηρίστηκαν ως «mini» άρχισε το 1965 και περιλαμβάνεται μερικώς στην τρίτη γενιά και μερικώς στην τέταρτη γενιά. Οι εταιρίες digital equipment corporation (DEC), Hewlett Packard, Data General ήταν οι κατ' εξοχήν κατασκευαστές υπολογιστικών συστημάτων της κατηγορίας αυτής.

#### **1.4 Μηχανές Τέταρτης Γενιάς (1971- Σήμερα)**

Η Τέταρτη Γενιά που αποτελεί τη σημερινή μορφή των υπολογιστικών μηχανών, αρχίζει από το έτος 1971, ενώ υποστηρίζεται επίσης ότι αρχίζει από το έτος 1978. η γενιά αυτή χαρακτηρίζεται από τη χρησιμοποίηση κυκλωμάτων πολύ μεγάλης ολοκλήρωσης (VLSI-Very Large Scale Integration) και την κατασκευή μονάδων επεξεργασίας με μονολιθικά κυκλώματα πυριτίου. Τα κυκλώματα αυτά έχουν το όνομα Microprocessors (Μικροεπεξεργαστής) και χρησιμοποιούνται κυρίως για την κατασκευή μικροϋπολογιστών.



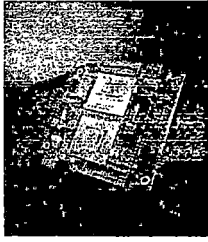
Διάγραμμα λειτουργίας Μικροεπεξεργαστή

Οι υπολογιστικές μηχανές της κατηγορίας αυτής χρησιμοποιούν ως κεντρική μνήμη κυκλώματα ημιαγωγών (Flip-Flops), ενώ χρησιμοποιούν επίσης πολλά από τα επιτεύγματα της νέας τεχνολογίας στις μονάδες τους. Γενικά πάντως, η διάκριση μεταξύ των μηχανών τρίτης και τέταρτης γενεάς δεν είναι τόσο εμφανής, όσο μεταξύ των προηγούμενων γενεών.

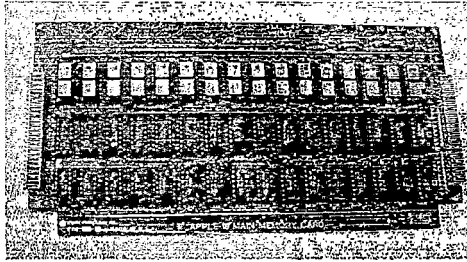
### 1.5 Μηχανές Πέμπτης Γενεάς

Οι μηχανές της Πέμπτης Γενεάς δεν έχουν ακόμη καθορισμένες ιδιότητες, αλλά θεωρείται ότι θα διαφέρουν ουσιαστικά από το μοντέλο της μηχανής του John Von Neumann. Ήδη από τα μέσα της δεκαετίας του 1980 ξεκίνησε στην Ιαπωνία ένα μεγαλόπνοο σχέδιο σχεδίασης και κατασκευής υπολογιστικών μηχανών της αποκαλούμενης πέμπτης γενεάς, οι οποίες αναμένεται να διαφέρουν στη λογική από τις σημερινές μηχανές που ονομάζονται «συμβατικές».

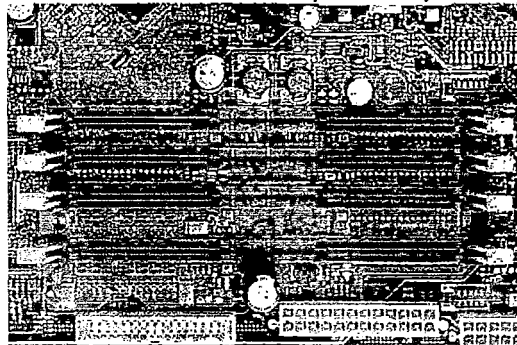




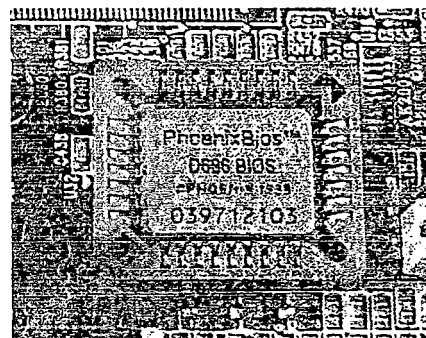
- Μονάδες στις οποίες ο υπολογιστής μπορεί να αποθηκεύσει δεδομένα κατά τη διάρκεια της λειτουργίας του και αποκαλούνται **κύρια μνήμη (main memory)**.



- Μονάδες στις οποίες ο υπολογιστής μπορεί να αποθηκεύσει δεδομένα και στις οποίες θα παραμείνουν και μετά τη λήξη της λειτουργίας του. Οι μονάδες αυτές αποκαλούνται **μονάδες βοηθητικής μνήμης**. Τρόπους με τους οποίους επικοινωνούν οι παραπάνω μονάδες μεταξύ τους.

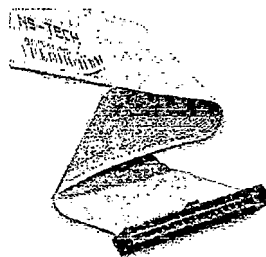


RAM



ROM

- Η επικοινωνία μεταξύ των μονάδων γίνεται χρησιμοποιώντας τους **διαδρόμους (buses)**.



Πληροφορίες (δεδομένα και σήματα ελέγχου) μπορούν να μεταφερθούν:

- Από τη μονάδα εισόδου προς την ΚΜΕ και την κύρια μνήμη,

- από την ΚΜΕ προς την κύρια μνήμη, και το αντίστροφο,
- από την ΚΜΕ ή την κύρια μνήμη στη μονάδα εξόδου,
- από την ΚΜΕ ή την κύρια μνήμη στη μονάδα αποθήκευσης, και το αντίστροφο,
- από τη μονάδα αποθήκευσης προς την ΚΜΕ και την κύρια μνήμη, και το αντίστροφο.

Είναι πολύ σημαντικό να τονίσουμε ότι κάθε χρονική στιγμή μόνο δύο συσκευές μπορούν να επικοινωνούν μέσω του διαδρόμου. Έτσι, αν κάποια στιγμή επικοινωνεί μέσω του διαδρόμου η ΚΜΕ με τη μνήμη, η μονάδα εισόδου δεν μπορεί να στείλει δεδομένα, αλλά πρέπει να περιμένει να ολοκληρωθεί η επικοινωνία μεταξύ της ΚΜΕ και της κυρίας μνήμης.

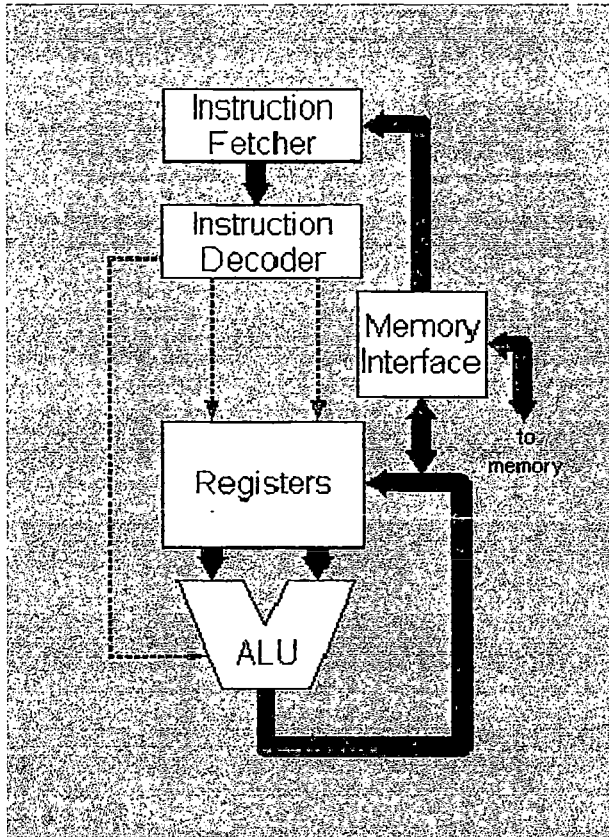
Για παράδειγμα, στην περίπτωση του υπολογισμού του αποτελέσματος μιας αριθμητικής πράξης, ο χρήστης εισάγει τα δεδομένα (αριθμούς και σύμβολο πράξης) από το πληκτρολόγιο. Τα δεδομένα μεταφέρονται μέσω της ΚΜΕ στην κύρια μνήμη.

Στη συνέχεια, η ΚΜΕ ανακαλεί τα δεδομένα από την κύρια μνήμη, εκτελεί την αριθμητική πράξη και τοποθετεί το αποτέλεσμα στην κύρια μνήμη. Το αποτέλεσμα της πράξης εμφανίζεται στην οθόνη. Τέλος, είναι δυνατό το αποτέλεσμα της πράξης να αποθηκευτεί για μελλοντική χρήση στο σκληρό δίσκο του υπολογιστή.

## 2.1 Κεντρική Μονάδα Επεξεργασίας

Η Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) κάνει επεξεργασία δεδομένων. Η επεξεργασία των δεδομένων γίνεται σε μια σειρά από βήματα, τις εντολές. Οι εντολές που εκτελούνται από την ΚΜΕ είναι εντολές σε γλώσσα μηχανής. Μια εντολή σε γλώσσα μηχανής είναι μια σειρά από δυαδικά ψηφία όπου είναι κωδικοποιημένο το είδος της εντολής. Οι εντολές της γλώσσας μηχανής είναι αποθηκευμένες στην κύρια μνήμη, από όπου τις ανακαλεί και τις εκτελεί η ΚΜΕ. Η ΚΜΕ αποτελείται από τρία τμήματα:

- Την αριθμητική και λογική μονάδα (Arithmetic and Logic Unit, ALL/), η οποία εκτελεί τις αριθμητικές και λογικές πράξεις.
  - Την μονάδα ελέγχου (Control Unit), η οποία συντονίζει την εκτέλεση των εντολών και την εκτέλεση των πράξεων στην αριθμητική και λογική μονάδα.
  - Τους καταχωρητές (registers), οι οποίοι χρησιμεύουν ως χώροι αποθήκευσης.
- Οι φάσεις εκτέλεσης μιας εντολής είναι οι εξής:
- Τη φάση ανάκλησης (fetch cycle), κατά την οποία η μονάδα ελέγχου αποφασίζει ποια εντολή πρέπει να ανακληθεί από τη μνήμη και την ανακαλεί. Στη φάση αυτή, η μονάδα ελέγχου αποκωδικοποιεί την εντολή (βρίσκει, δηλαδή, ποια εντολή είναι).
  - Τη φάση εκτέλεσης (execution cycle), κατά την οποία η ΚΜΕ εκτελεί την εντολή. Αν, για παράδειγμα, η εντολή είναι για την εκτέλεση αριθμητικής ή λογικής πράξης, η αριθμητική και λογική μονάδα εκτελεί την πράξη.



*Instruction Fetcher: Ανακλητής Εντολών*  
*Instruction Decoder: Αποκωδικοποιητής Εντολών*  
*Registers: Καταχωρητές*  
*ALU: Αριθμητική και Λογική Μονάδα*  
*Memory Interface: Διεπαφή Μνήμης*

Οι εντολές που συναντάμε στους υπολογιστές είναι:

- Ανάγνωση ενός δεδομένου από τη μνήμη.
- Εκτέλεση πράξεων (π.χ. πρόσθεση, πολλαπλασιασμός κ.λ.π.)
- Εισαγωγή ενός δεδομένου από μονάδα εισόδου.
- Εξαγωγή ενός δεδομένου από μονάδα εξόδου.

Για να γίνει κατανοητός ο τρόπος που λειτουργεί ο υπολογιστής, ας υποθέσουμε

ότι έχουμε την εκτέλεση μιας εντολής πολλαπλασιασμού δύο αριθμών που

βρίσκονται στη μνήμη και πως το αποτέλεσμα πρέπει να αποθηκευτεί στη μνήμη.

Χρειάζονται οι ακόλουθες εντολές:

- 1 Ανάγνωση του πρώτου αριθμού και μεταφορά στον καταχωρητή A.
- 2 Ανάγνωση του δεύτερου αριθμού και μεταφορά στον καταχωρητή B.
- 3 Πολλαπλασιασμός του αριθμού που βρίσκεται στους καταχωρητές A και B

- 4 και αποθήκευση του αποτελέσματος στον καταχωρητή A.  
Εγγραφή του αποτελέσματος στη μνήμη.

*Η εκτέλεση κάθε εντολής περιλαμβάνει δυο φάσεις:*

- Τη φάση ανάκλησης, όπου ανακαλείται από τη μνήμη το είδος της εντολής, το οποίο αποθηκεύεται στη μονάδα ελέγχου (CU).
- Τη φάση εκτέλεσης όπου η μονάδα ελέγχου δίνει εντολή στην αριθμητική και λογική μονάδα η οποία εκτελεί τον πολλαπλασιασμό μεταξύ των περιεχομένων των καταχωρητών A και B και αποθηκεύει το αποτέλεσμα στον καταχωρητή A όπως φαίνεται από τα βέλη κατεύθυνσης του σχήματος.

## 2.2 Κεντρική Μνήμη

Η κεντρική μνήμη (main memory) που αναφέρεται επίσης και ως πρωτεύουσα μνήμη (primary memory), δίνει σε ένα υπολογιστικό σύστημα τη δυνατότητα καταχώρησης και ανάκλησης:

**α.** Των υπό εκτέλεση προγραμμάτων, που καταχωρούνται ως ακολουθίες πράξεων.

**β.** Των πληροφοριακών δεδομένων επί των οποίων τα εκτελούμενα προγράμματα εφαρμόζονται.

Πιο απλά, η κεντρική μνήμη των υπολογιστικών συστημάτων θεωρούμε ότι χρησιμοποιείται για τους ακόλουθους τέσσερις βασικούς λόγους:

- 1.** Για να καταχωρούνται τα πληροφοριακά δεδομένα, τα οποία εισάγονται στο υπολογιστικό σύστημα για άμεση επεξεργασία.
- 2.** Για να καταχωρούνται τα αποτελέσματα πράξεων που εκτελούνται επί των πληροφοριακών δεδομένων, που ήδη είναι καταχωρημένα στη μνήμη.
- 3.** Για να καταχωρούνται αντίγραφα των πληροφοριακών δεδομένων που ήδη υπάρχουν στην μνήμη, αλλά απαιτείται η αναπαραγωγή τους.
- 4.** Για να καταχωρούνται οι ακολουθίες των πράξεων, που πρόκειται να εκτελεστούν. Οι πράξεις καταχωρούνται σε μια κωδικοποιημένη μορφή που μετά την αποκωδικοποίησή της είναι άμεσα εκτελέσιμη από την αριθμητική και λογική μονάδα του υπολογιστικού συστήματος. Η μορφή με την οποία καταχωρούνται οι πράξεις αποτελεί μια τεχνική γλώσσα. Η γλώσσα αυτή, την οποία θεωρητικά “αντιλαμβάνεται” το υπολογιστικό σύστημα (δηλαδή η μηχανή), ονομάζεται για το λόγο αυτό “γλώσσα μηχανής” (machine language). Τα προγράμματα αποτελούνται από οδηγίες (ή εντολές) προς τη μηχανή, για τις υπό εκτέλεση πράξεις. Αποτελούνται λοιπόν από ακολουθίες εντολών που καθορίζουν τις πράξεις, τις οποίες η μηχανή πρόκειται να εκτελέσει.

Η μνήμη του ηλεκτρονικού υπολογιστικού συστήματος αποτελείται από ένα οργανωμένο πλήθος κελιών ή στοιχείων. Κάθε στοιχείο της μνήμης μπορεί να βρίσκεται σε μία από τις δύο δυνατές καταστάσεις ή μεταφορικά να περιέχει μία από τις δύο δυνατές τιμές.

Παλαιότερα, η υλοποίηση των στοιχείων της μνήμης πραγματοποιούνταν με μικροσκοπικούς μαγνητικούς δακτυλίους. Οι τελευταίοι μαγνητίζονταν είτε δεξιόστροφα, για να αποδώσουν τη μία κατάσταση, είτε αριστερόστροφα για να

αποδώσουν την άλλη. Για τη μαγνήτισή του χρησιμοποιούνταν ηλεκτροφόροι αγωγοί, που περνούσαν από κέντρο τους. Η μαγνήτιση, δεξιόστροφη ή αριστερόστροφη ήταν συνάντηση της φοράς του ηλεκτρικού ρεύματος που διαπερνούσε τους αγωγούς.

Η μνήμη, που αποτελείται από μαγνητικούς δακτυλίους, ονομάζεται μαγνητική μνήμη (core). Η ανάκληση της τιμής που περιέχει ένας μαγνητικός δακτύλιος πραγματοποιείται μέσω ενός δεύτερου αγωγού αλλά αυτό έχει σαν αποτέλεσμα την απώλεια της τιμής του και για το λόγο αυτό απαιτείται ένας επιπλέον αγωγός, για να επαναφέρει στο δακτύλιο τη προηγούμενη τιμή του.

Ο μαγνητισμός των μαγνητικών δακτυλίων ήταν μόνιμος και μπορούσε να μεταβληθεί με αλλαγή της φοράς του ρεύματος του αγωγού που τον διαπερνούσε. Δεν ήταν συνεπώς αναγκαία η διατήρηση της μαγνητικής μνήμης με συνεχή τροφοδοσία ηλεκτρικού ρεύματος. Απλώς, η μαγνητική μνήμη ήταν δυνατόν να παραμένει αμετάβλητη και να κρατά συνεπώς πληροφοριακά δεδομένα για όσο διάστημα ήταν επιθυμητό. Αυτό είχε σαν αποτέλεσμα την επάνοδο στην προηγούμενη κατάσταση, στην περίπτωση που η μηχανή, για οποιοδήποτε λόγο και κυρίως μετά τη διακοπή της τροφοδοσίας της με ηλεκτρικό ρεύμα, διέκοπτε την λειτουργία της. Για το λόγο αυτό η μαγνητική μνήμη και γενικά κάθε είδος μνήμης που δεν απαιτεί τροφοδοσία για να διατηρήσει τα στοιχεία που περιέχει, ονομάζεται “σταθερή μνήμη” (non volatile storage).

Σήμερα, τα στοιχεία μνήμης υλοποιούνται μέσω ημιαγωγικών (semi-conductor) ολοκληρωμένων κυκλωμάτων πυριτίου (τσιπ) που έχουν σαν ελάχιστο στοιχείο το flip-flop. Το χαμηλό δυναμικό (τάση) υλοποιεί την μία κατάσταση και το υψηλό την άλλη. Το flip-flop θεωρείται, όπως βέβαια και ο μαγνητικός δακτύλιος, ως το ελάχιστο στοιχείο μνήμης.

Βασική διαφορά των flip-flops από τους μαγνητικούς δακτυλίους, είναι η μεγάλη τους ταχύτητα και ο μικρός τους όγκος, που συνεχώς μειώνεται λόγω της αύξησης της ολοκλήρωσης. Ωστόσο, η ημιαγωγική μνήμη δεν είναι σταθερή και για να διατηρηθεί απαιτεί διαρκή τροφοδοσία με ηλεκτρικό ρεύμα. Είναι δηλαδή ασταθής μνήμη (volatile storage). Σήμερα χρησιμοποιούνται διάφορες τεχνολογίες ημιαγωγικών μνημών, από τις οποίες οι περισσότερες γρηγορές βρίσκονται στην αριθμητική και λογική μονάδα των ηλεκτρονικών υπολογιστικών συστημάτων, ενώ οι λιγότερες γρηγορές που κατασκευάζονται από MOS (Metal-Oxide Semiconductor, ημιαγωγός οξειδίων μετάλλου) βρίσκονται στη κεντρική μνήμη.

Τα ολοκληρωμένα κυκλώματα που χρησιμοποιούνται στην κεντρική μνήμη των υπολογιστικών συστημάτων, διακρίνονται στα δυναμικά και τα στατικά και η αντίστοιχη μνήμη σε δυναμική (dynamic memory) και στατική (static memory). Τα στοιχεία από τα οποία αποτελείται η δυναμική μνήμη περιέχουν ένα τρανζίστορ, που ενεργεί σαν διακόπτης και ένα πυκνωτή στον οποίο καταχωρείται ηλεκτρικό φορτίο. Ο διακόπτης κάθε στοιχείου, δηλαδή το τρανζίστορ, καθορίζει αν ο πυκνωτής έχει φορτίο ή όχι, όπως περίπου ένας διακόπτης μιας ηλεκτρικής λυχνίας καθορίζει αν η λυχνία φωτίζει ή όχι. Η δυναμική μνήμη είναι ασταθής μνήμη, εξαιτίας της απώλειας του φορτίου του πυκνωτή και απαιτεί περιοδική αναπαραγωγή του φορτίου (refresh, φρεσκάρισμα) για να διατηρείται. Η στατική μνήμη είναι και αυτή ασταθής, αλλά δεν

απαιτεί αναπαραγωγή των τιμών που περιέχουν τα στοιχεία της. Απαιτεί όμως τροφοδοσία με ρεύμα. Τα στοιχεία της στατικής μνήμης είναι περισσότερο πολύπλοκα, περιέχουν περισσότερα από ένα τρανζίστορ και είναι μεγαλύτερα σε όγκο. Για το λόγο αυτό, η στατική μνήμη χρησιμοποιείται σε ειδικές περιπτώσεις, ενώ η δυναμική μνήμη χρησιμοποιείται κυρίως στη κεντρική μνήμη των ηλεκτρονικών υπολογιστικών συστημάτων.

### 2.3 Οργάνωση της Μνήμης

Οι δύο διαφορετικές καταστάσεις, στις οποίες είναι δυνατόν να βρίσκεται το ελάχιστο στοιχείο μνήμης, συμβολίζονται με τα δυαδικά ψηφία 0 και 1. Ο συμβολισμός είναι ιδιαίτερα πρακτικός και έχει επικρατήσει. Θεωρούμε ότι τα δυαδικά ψηφία 0 και 1, που αποτελούν το σύνολο  $\{0,1\}$  είναι ένα αλφάβητο δύο συμβόλων. Με το αλφάβητο αυτό είναι δυνατό να κωδικοποιηθεί οποιοδήποτε πληροφοριακό δεδομένο και να εισαχθεί και καταχωρηθεί σε ένα υπολογιστικό σύστημα.

#### Bit:

Συνεπώς, το ελάχιστο στοιχείο μνήμης του ηλεκτρονικού υπολογιστή, μπορεί να αποδώσει και να αποθηκεύσει δύο απλές καταστάσεις που αντιστοιχούν σε ένα δυαδικό ψηφίο. Για το λόγο αυτό το ελάχιστο στοιχείο μνήμης ονομάζεται bit. Το όνομα bit προέρχεται από το 1ο και τα δύο τελευταία γράμματα του όρου Binary digIT, που σημαίνει δυαδικό ψηφίο.

#### Λέξη:

Μια λέξη (word) είναι η ακολουθία από bit. Το πλήθος  $N$  των bit που περιέχονται σε μια λέξη, είναι χαρακτηριστικό για κάθε είδος μηχανής. Κάθε ένα από τα bit μιας λέξης μπορεί να βρίσκεται στη κατάσταση 0 ή στη κατάσταση 1.

Για το λόγο αυτό, μια λέξη μπορεί να αποδώσει μέχρι  $2^N$  το πολύ διαφορετικές καταστάσεις, οι οποίες αντιστοιχούν στους δυνατούς συνδυασμούς των καταστάσεων των  $N$  bit. Με τον τρόπο αυτό μια λέξη είναι δυνατόν να παραστήσει και  $2^N$  διαφορετικά πληροφοριακά στοιχεία, αν κάθε ένα αποδοθεί με ένα διαφορετικό συνδυασμό καταστάσεων των bit της λέξης.

Ο χαρακτηριστικός αριθμός  $N$ , που αποδίδει το πλήθος των bit ανά λέξη, ποικίλλει στους διάφορους τύπους μηχανών. Υπάρχουν μηχανές διάφορων κατηγοριών με 4, 8, 12, 16, 18, 24, 32, 36, 48, 64 bit ανά λέξη.

#### Byte:

Μια τυπική υποδιαίρεση των λέξεων, που αποτελεί την πιο συνηθισμένη ομαδοποίηση των bit, είναι μια ακολουθία από 8 bit, που ονομάζεται byte. Σύμφωνα με αυτό, μια λέξη των 8 bit αποτελείται από 1 byte, μια λέξη των 16 bit αποτελείται από 2 bytes, ενώ μια λέξη των 32 bit αποτελείται από 4 bytes.

Η χρήση του όρου «byte» έχει ελαχιστοποιήσει τη χρήση του όρου «λέξη», που δεν είναι σαφώς καθορισμένη και το μέγεθός της ποικίλλει στους διάφορους τύπους μηχανών. Η έννοια της λέξης θεωρείται επίσης ως περισσότερο κατανοητή στους ειδικούς, απ' ό,τι στους απλούς χρήστες των υπολογιστικών συστημάτων. Αντίθετα, το byte έχει επικρατήσει ως όρος και αποδίδει μια ακολουθία από 8 bit.

## 2.4 Μέτρηση της χωρητικότητας της μνήμης

Το byte είναι η μονάδα μέτρησης της χωρητικότητας της μνήμης. Πολλαπλάσιο του byte είναι ο αριθμός 1024, που αποδίδεται με το γράμμα K και προφέρεται Kilo. Το πολλαπλάσιο K, που δεν σημαίνει 1000, αλλά 1024, προκύπτει από την ύψωση του αριθμού 2 στη δεκάτη δύναμη ( $2^{10}$ ). Το γράμμα K τοποθετείται πριν από τη λέξη byte (Kbyte), ή πριν από το γράμμα b (Kb) και προφέρεται "kilobyte".

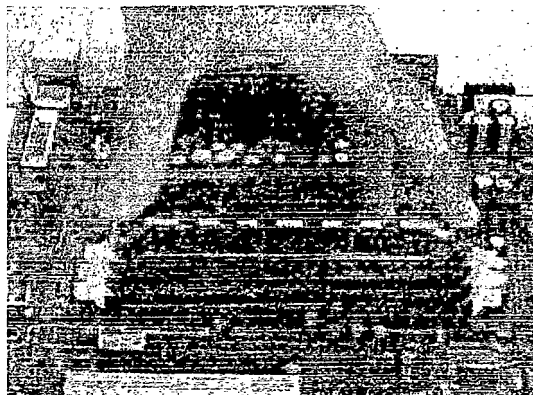
Το γράμμα M (Mega), που σε άλλες περιπτώσεις χρησιμοποιείται για να αποδώσει εκατομμύρια, όταν αναφέρεται σε μεγέθη μνήμης υπολογιστικών συστημάτων δεν σημαίνει  $1.000 \times 1.000$ , αλλά  $1024 \times 1024$ , δηλαδή  $2^{20}$  ή 1.048.576. Όπως και στην περίπτωση του K έτσι και με το M, τα πολλαπλάσια του byte γράφονται Mbyte, ή Mb και προφέρονται "Megabyte".

Τέλος, το γράμμα G είναι το πρώτο γράμμα της λέξης Giga (1 Giga = 1.024 Mega = 1.073.741.824) και σημαίνει 1024 Mega και χρησιμοποιείται όταν γίνεται αναφορά σε μνήμες πολύ μεγάλης χωρητικότητας, όπως οι περιφερειακές μνήμες π.χ. μαγνητικοί δίσκοι ή άλλα μέσα. Τα πολλαπλάσια του byte γράφονται Gbyte, ή Gb και προφέρονται "Gigabyte".

Σύμφωνα με αυτά η χωρητικότητα μνήμης 65.536 byte γράφεται 65 Kb ή 65 Kbyte. Αντιστοίχως, η χωρητικότητα μνήμης 16.777.216 byte γράφεται 16 Mb, ή 16 Mbyte.

## 2.5 Τοποθέτηση μνήμης

Κάθε συσκευασία μνήμης έχει διαφορετικό τρόπο σύνδεσης με το υπόλοιπο υπολογιστικό σύστημα. Έτσι, η μνήμη DIP μπορεί είτε να κολληθεί απευθείας πάνω στην πλακέτα του υπολογιστή είτε να τοποθετηθεί πάνω σε ειδική βάση, η οποία με τη σειρά της είναι κολλημένη πάνω στην πλακέτα. Η βάση αυτή μας επιτρέπει να τοποθετούμε και να αφαιρούμε εύκολα τη μνήμη DIP.

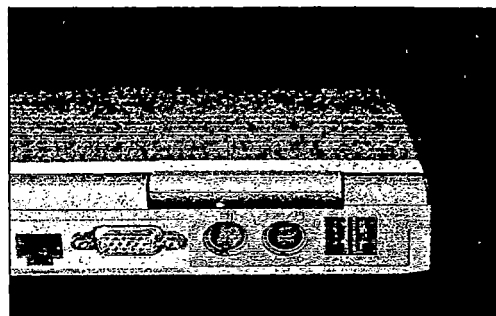


Οι μνήμη SIMM τοποθετείται πάντα πάνω στην ειδική βάση. Στην αριστερή πλευρά της μνήμης αυτής υπάρχει μια εγκοπή, η οποία δείχνει την πλευρά στην οποία βρίσκεται η επαφή 1 της μνήμης. Αντίστοιχα, στη βάση της υπάρχει ένα εξόγκωμα στην ίδια πλευρά, το οποίο εμποδίζει τη λανθασμένη τοποθέτηση της. Η βάση αυτή διαθέτει επίσης στις άκρες της δύο ελάσματα, τα οποία χρησιμεύουν για την ακινητοποίηση της μνήμης μετά την τοποθέτηση της.

Παρόμοια είναι και η διαδικασία τοποθέτησης της μνήμης DIMM στην αντίστοιχη βάση. Η μνήμη DIMM 168 επαφών έχει στο κάτω μέρος της δύο εγκοπές, μία στο κέντρο και μία ασύμμετρα τοποθετημένη προς την πλευρά που βρίσκεται η επαφή 1, ώστε να αποκλείεται η λανθασμένη τοποθέτηση της στη βάση.

## Μνήμες

Οι μνήμες χρησιμοποιούνται στους υπολογιστές για την προσωρινή ή τη μόνιμη αποθήκευση του κώδικα του προγράμματος που εκτελείται και των δεδομένων του. Οι βασικότερες κατηγορίες μνήμης είναι δυο, η μνήμη Ram (Random Access Memory- μνήμη τυχαίας προσπέλασης) και η μνήμη Rom (Read Only Memory- μνήμη μόνο ανάγνωσης).





## 2.6 Θύρες επικοινωνίας

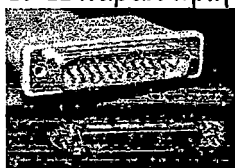
Ένα μεγάλο μέρος της λειτουργικότητας των προσωπικών υπολογιστών οφείλεται στην ποικιλία των περιφερειακών συσκευών που μπορούμε να συνδέσουμε σε αυτούς. Εξάλλου η μεγάλη ποικιλία περιφερειακών συσκευών επιτρέπει τη χρήση των προσωπικών υπολογιστών σε μεγάλο αριθμό διαφορετικών εφαρμογών.

Ο προσωπικός υπολογιστής διαθέτει θύρες (ports), για να επικοινωνεί με τις περιφερειακές συσκευές ή με άλλες ηλεκτρονικές συσκευές και υπολογιστές.

Οι θύρες επικοινωνίας του προσωπικού υπολογιστή είναι τυποποιημένες. Με άλλα λόγια, ο τρόπος σύνδεσης, η ταχύτητα, τα σήματα που ανταλλάσσονται και γενικώς τα χαρακτηριστικά μίας θύρας επικοινωνίας του υπολογιστή είναι αυστηρώς προδιαγεγραμμένα. Αυτό δίνει τη δυνατότητα να μπορούν να συνδέονται πάνω στην ίδια θύρα - διαφορετικές περιφερειακές συσκευές και μάλιστα διαφορετικών κατασκευαστών, αρκεί να είναι σχεδιασμένες σύμφωνα με το πρότυπο που απαιτεί η θύρα αυτή. Έτσι, για παράδειγμα, στην παράλληλη θύρα του υπολογιστή μπορούμε να συνδέσουμε έναν εκτυπωτή ή ένα σαρωτή (scanner) ή ακόμη και μια ψηφιακή κάμερα.

Υπάρχουν τρεις τυποποιημένες θύρες, που έχουν επικρατήσει στο χώρο των προσωπικών υπολογιστών:

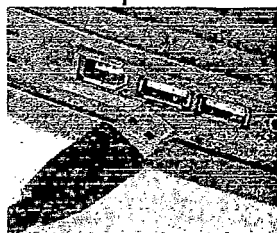
### 1. Η παράλληλη θύρα



### 2. Η σειριακή θύρα και



### 3. Η θύρα USB.



Καθεμιά από αυτές διαθέτει ένα ειδικό προσαρμοστικό κύκλωμα (interface) για την επικοινωνία με την ΚΜΕ μέσω του διαδρόμου του υπολογιστή.

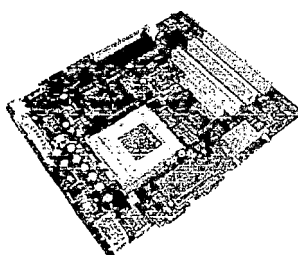
Τα προσαρμοστικά κυκλώματα παλιότερα ήταν σε μορφή κάρτας επέκτασης και συνδέονταν πάνω σε μια υποδοχή επέκτασης του υπολογιστή. Εδώ και αρκετό καιρό για λόγους οικονομίας χώρου τα προσαρμοστικά κυκλώματα έχουν συμπεριληφθεί στα κυκλώματα υποστήριξης της μητρικής πλακέτας. Βεβαία διατίθενται ακόμα κάρτες με προσαρμοστικά κυκλώματα θυρών, στην περίπτωση που θέλουμε να προσθέσουμε επιπλέον θύρες πέρα από αυτές που διαθέτει η μητρική, ή να αντικαταστήσουμε μια κατεστραμμένη θύρα.

Η επικοινωνία της ΚΜΕ με το προσαρμοστικό κύκλωμα γίνεται μέσω διευθύνσεων θύρας (port address). Μέσω των διευθύνσεων αυτών η ΚΜΕ μπορεί:

1. Να δέχεται και να στέλνει δεδομένα μέσω της θύρας.
2. Να ελέγχει την ταχύτητα και γενικότερα τα χαρακτηριστικά της επικοινωνίας της θύρας.

Στο προσαρμοστικό κύκλωμα της θύρας διαθέτει επίσης μία διακοπή, με την οποία ειδοποιεί την ΚΜΕ, για παράδειγμα, ότι έχει έτοιμα δεδομένα, ή ότι υπάρχει κάποιο λάθος κατά την αποστολή των δεδομένων κ.ο.κ.

## 2.7 Η μητρική πλακέτα

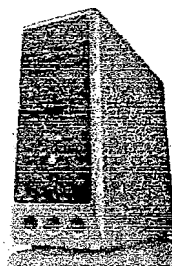


Η μητρική πλακέτα (motherboard) είναι μια παραλληλόγραμμη πλακέτα διαστάσεων περίπου 22x25 cm, πάνω στην οποία είτε είναι τοποθετημένες είτε συνδέονται με τη βοήθεια καλωδίων όλες οι μονάδες του προσωπικού υπολογιστή. Στην ουσία είναι ένα σχετικά μεγάλο τυπωμένο κύκλωμα, πάνω στο οποίο είναι κολλημένα ολοκληρωμένα κυκλώματα που υλοποιούν τις διάφορες μονάδες του υπολογιστή. Επίσης, πάνω στη μητρική πλακέτα είναι τοποθετημένες οι υποδοχές επέκτασης (expansion slots) για τη σύνδεση άλλων περιφερειακών μονάδων.

Η αρχιτεκτονική της μητρικής πλακέτας έχει άμεση σχέση με το είδος του επεξεργαστή του προσωπικού υπολογιστή. Έτσι, για παράδειγμα, η μητρική πλακέτα πρέπει να έχει διαφορετικά κυκλώματα, όταν χρησιμοποιείται ένας επεξεργαστής της οικογένειας Pentium, και άλλα, όταν χρησιμοποιείται ένας επεξεργαστής της οικογένειας Pentium II. Επίσης, διαφορετική είναι η βάση πάνω στην οποία τοποθετείται ο επεξεργαστής καθώς και το είδος της μνήμης που χρησιμοποιείται. Επομένως, η επιλογή της μητρικής πλακέτας του προσωπικού υπολογιστή γίνεται έχοντας πάντα υπόψη τον επεξεργαστή που θα χρησιμοποιηθεί. Συνήθως οι μητρικές πλακέτες σχεδιάζονται, ώστε να μπορούν να τοποθετηθούν σε αυτές διάφοροι επεξεργαστές, οι οποίοι όμως πρέπει να έχουν παρόμοια αρχιτεκτονική. Έτσι, σε μια μητρική πλακέτα ενδεχομένως να μπορεί να τοποθετηθεί ένας από τους επεξεργαστές της οικογένειας Pentium II ή Celeron, αλλά δεν μπορεί να τοποθετηθεί ένας επεξεργαστής Pentium, γιατί έχουν διαφορετική αρχιτεκτονική.

## 2.8 Το κουτί της κεντρικής μονάδας

Τα τμήματα της κεντρικής μονάδας ενός προσωπικού υπολογιστή (η μητρική πλακέτα, το τροφοδοτικό, οι σκληροί δίσκοι, οι οδηγόι δισκέτας και οι κάρτες επέκτασης) τοποθετούνται σε ειδικά διαμορφωμένους χώρους σ' ένα μεταλλικό κουτί.



Κατά καιρούς έχουν κυκλοφορήσει διάφοροι τύποι κουτιών με διαφορετικές ιδιότητες και σχεδιασμένα για διαφορετικές χρήσεις. Για παράδειγμα, υπάρχουν κουτιά που τοποθετούνται οριζόντια και άλλα που τοποθετούνται κάθετα, υπάρχουν κουτιά που χρησιμοποιούνται σε βιομηχανικό περιβάλλον και άλλα που δε χρησιμοποιούν καθόλου βίδες για τη στήριξη της μητρικής πλακέτας και των περιφερειακών.

**Μέρος Β : Από την κωδικοποίηση της πληροφορικής στην ολοκληρωμένη αρχιτεκτονική των υπολογιστικών συστημάτων**

Κεφάλαιο 3 : Κωδικοποίηση της πληροφορικής ,θεωρήματα και πράξεις για τα διάφορα συστήματα κωδικοποίησης.

**3.1 Αριθμητικά συστήματα**

**3.1.1 Συστήματα Αρίθμησης – Σχέση bit και χαρακτήρα**

Ονομάζουμε σύστημα αρίθμησης τη μέθοδο παράστασης των αξιών που επιδέχονται μέτρηση, έτσι ώστε σε κάθε μια από αυτές να αντιστοιχεί μία και μοναδική "εικόνα", την οποία ονομάζουμε παράσταση της αριθμητικής αξίας στο σύστημα αρίθμησης. Τα συστήματα αρίθμησης διακρίνονται σε θεσιακά και μη θεσιακά.

Ένα από τα μεγαλύτερα επιτεύγματα του πολιτισμού μας θεωρείται η επινόηση των θεσιακών συστημάτων αρίθμησης. Στα θεσιακά συστήματα αρίθμησης (Positional Number Systems) μια αριθμητική αξία παριστάνεται σαν μια αλυσίδα από ψηφία έτσι ώστε η θέση κάθε ψηφίου μέσα στην αλυσίδα να έχει το δικό της αριθμητικό βάρος.

Στα μη θεσιακά συστήματα δεν ισχύει η παραπάνω ιδιότητα και η τοποθέτηση των ψηφίων μέσα στην παράσταση γίνεται με εμπειρικό τρόπο.

Σε ένα θεσιακό σύστημα αρίθμησης κάθε αριθμητική αξία  $X$  μπορεί να παρασταθεί σαν μια αλυσίδα από ψηφία τα οποία ανήκουν στο σύνολο  $S$  των ψηφίων του συστήματος. Αν το  $R$  είναι μικρότερο από το 10, τότε σύνολο των ψηφίων, είναι το  $S = [0,1,2,...,R-1]$ . Αν το  $R$  είναι μεγαλύτερο από το 10 τότε σαν ψηφία επιλέγουμε πρώτα τα ψηφία του δεκαδικού συστήματος και μετά γράμματα του λατινικού αλφαβήτου και μέχρι το σύνολο αποκτήσει  $R$  διαφορετικά στοιχεία. Η αλυσίδα αυτή έχει τη μορφή:

$$X_R = X_{N-1} X_{N-2} \dots X_0 \cdot X_{-1} X_{-2} \dots X_{-M}$$

όπου  $X_i$  είναι ψηφίο του συστήματος για  $i = N-1, N-2, \dots, -1, 0, 1, \dots, -M$  και εκφράζεται από τη σχέση :

$$X_R = \sum_{N-1}^0 X_i R^i + \sum_{-1}^{-M} X_i R^i \quad (1)$$

Στην παραπάνω ιδιότητα οφείλεται και ο χαρακτηρισμός του συστήματος σαν θεσιακό. Το πρώτο μέρος του αθροίσματος αποτελεί το ακέραιο μέρος της αριθμητικής ποσότητας και το δεύτερο το κλασματικό μέρος. Το  $R$  ονομάζεται βάση του συστήματος αρίθμησης. Το σημείο μεταξύ ακεραίου παράστασης και της

κλασματικής ονομάζεται "βασικό" σημείο. Σημειώνουμε ότι το  $X_{N-1}$  ονομάζεται το πρώτο (MSD – Most Significant digit) σημαντικό ψηφίο και το  $X_{-M}$  το τελευταίο (LSD – Last Significant Digit) σημαντικό ψηφίο της παράστασης του αριθμού  $X$ , η οποία αποτελείται από  $N+M$  σημαντικά ψηφία.

**Παράδειγμα :** Στο δεκαδικό σύστημα είναι  $R=10$  και οι αριθμητικές ποσότητες στο σύστημα αυτό παριστάνονται σαν αλυσίδες ψηφίων τα οποία ανήκουν στο σύνολο  $S = [0,1,2,3,\dots,9]$ . Στο σύστημα αυτό, το βασικό σημείο ονομάζεται δεκαδικό. Ένας τετραψήφιος αριθμός  $X_3X_2X_1X_0$  εκφράζεται από τη σχέση:

$$X_3X_2X_1X_0 = X_3 10^3 + X_2 10^2 + X_1 10^1 + X_0 10^0$$

Κάθε ψηφίο  $X_i$  έχει βάρος  $10^i$ , όπου (i) η θέση του ψηφίου.

Τα θεσιακά αριθμητικά συστήματα έχουν εύκολη αριθμητική σε σχέση με τα μη θεσιακά όπου η αριθμητική είναι πολύπλοκη. Αυτό θα το διαπιστώσουμε αργότερα όταν θα μελετήσουμε την αριθμητική του δυαδικού συστήματος. Ονομάζουμε αριθμητική ενός συστήματος, τις τέσσερες αριθμητικές πράξεις μεταξύ των παραστάσεων του συστήματος. Ένα σύστημα αρίθμησης με βάση  $R=2$  ονομάζεται δυαδικό, με  $R=8$  οκταδικό και με  $R=16$  δεκαεξαδικό.

**Παράδειγμα :** Αν μια αριθμητική ποσότητα  $X$  έχει την παράσταση 363.321 στο δεκαδικό σύστημα, τότε:

$$\begin{aligned} 363.531_{10} &= 3 \times 10^2 + 6 \times 10^1 + 3 \times 10^0 + 5 \times 10^{-1} + 3 \times 10^{-2} + 1 \times 10^{-3} = \\ &= 300 + 60 + 3 + 5/10 + 3/100 + 1/1000 \end{aligned}$$

παρατηρούμε ότι το ψηφίο 3 έχει διαφορετικό βάρος (πολλαπλασιάζεται με διαφορετικό συντελεστή που είναι συνάρτηση της θέσης του) στις θέσεις 2, 1 και -2.

### Δυαδικό, Οκταδικό και δεκαεξαδικό σύστημα αρίθμησης

Το δυαδικό σύστημα έχει βάση  $R=2$  και κάθε αριθμός σε αυτό μπορεί να παρασταθεί σαν μια αλυσίδα από ψηφία τα οποία ανήκουν στο σύνολο  $[0,1]$ . Το δυαδικό είναι το σύστημα με τη μικρότερη βάση που μπορεί να μας φανεί χρήσιμο. Πράγματι, συστήματα με βάση 1 και 0 μόνο θεωρητικά μπορούν να υπάρξουν.

Η παράσταση  $1011_2$  είναι η παράσταση ενός αριθμού στο δυαδικό σύστημα. Ο δείκτης 2 είναι απαραίτητος όταν δεν είναι σαφές ότι δουλεύουμε σ' ένα μόνο σύστημα. Και αυτό γιατί οι παραστάσεις  $1011_{10}$  και  $1011_2$ , αν έχουν την ίδια εικόνα-παράσταση, παριστάνουν εντελώς διαφορετικές αξίες.

Το οκταδικό σύστημα έχει σαν βάση το 8 και τα ψηφία του ανήκουν στο σύνολο  $[0,1,2,3,4,5,6,7]$ , ενώ το δεκαεξαδικό έχει βάση το 16 και τα ψηφία του ανήκουν στο σύνολο  $[0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F]$ . Η παράσταση  $623_8$  είναι η παράσταση ενός αριθμού στο οκταδικό σύστημα ενώ η  $6AF3.6AB_{16}$  είναι η παράσταση ενός αριθμού στο δεκαεξαδικό.

Το δυαδικό χρησιμοποιείται για την άμεση παράσταση των αριθμών μέσα στον υπολογιστή. Τα άλλα δύο συστήματα (οκταδικό και δεκαεξαδικό) χρησιμοποιούνται για τη συνοπτική παράσταση των δυαδικών ποσοτήτων εκτός του υπολογιστή. Και αυτό γιατί έχουν βάσεις που είναι δυνάμεις του δύο. Αυτή η ιδιότητα διευκολύνει, όπως θα δούμε παρακάτω, την μετατροπή της παράστασης ενός αριθμού από το ένα σύστημα στο άλλο.

**Παράδειγμα** παρουσιάζουμε τους αριθμούς από το 1 ως 16 του δεκαδικού συστήματος και τις αντίστοιχες παραστάσεις τους στο δυαδικό, οκταδικό και δεκαεξαδικό. Παρατηρούμε ότι τα ψηφία του οκταδικού συστήματος χρειάζονται 3 bit για να παρασταθούν στο δυαδικό σύστημα ενώ τα ψηφία του δεκαεξαδικού χρειάζονται 4. Κρατήστε στο μυαλό σας την αντιστοιχία του οκταδικού με το 3 και του δεκαεξαδικού με το 4. Θα χρειαστεί στις μετατροπές των παραστάσεων από το ένα σύστημα στο άλλο. Οι δύο τελευταίες στήλες του πίνακα 1.2 μας δείχνουν τη δυνατότητα παράστασης αριθμών στα 3 και 4 bits αντίστοιχα. Παρατηρήστε ότι το 8 δεν μπορεί να παρασταθεί στα 3 bits. Αυτό σημαίνει ότι, ένας υπολογιστή με μήκος θέσης 3 bits δεν μπορεί να δεχθεί σαν δεδομένο τον αριθμό 8 γιατί απλούστατα δεν μπορεί να τον αποθηκεύσει.

Δεκαδικό	Δυαδικό	Οκταδικό	Δεκαεξαδικό	3 Bits	4Bits	
0	0	0	0	0	000	0000
1	1	1	1	1	001	0001
2	10	2	2	2	010	0010
3	11	3	3	3	011	0011
4	100	4	4	4	100	0100
5	101	5	5	5	101	0101
6	110	6	6	6	110	0110
7	111	7	7	7	111	0111
8	1000	10	10	8	-	1000
9	1001	11	11	9	-	1001
10	1010	12	12	A	-	1010
11	1011	13	13	B	-	1011
12	1100	14	14	C	-	1100
13	1101	15	15	D	-	1101
14	1110	16	16	E	-	1110
15	1111	17	17	F	-	1111
16	10000	20	20	10	-	--

Αντιστοιχίες συστημάτων

### 3.2 Κωδικοποίηση της πληροφορικής

#### 3.2.1 Πληροφορία

Στην καθημερινή μας ζωή χρησιμοποιούμε σχεδόν αποκλειστικά το δεκαδικό σύστημα αρίθμησης. Είναι λοιπόν φυσικό να έχουμε ταυτίσει την αξία (value) μιας ποσότητας, με τη παράσταση της στο σύστημα αυτό. Στη πραγματικότητα δεν είναι έτσι. Οι αξίες μπορούν να παρασταθούν με πολλούς τρόπους. Η αξία παραμένει η ίδια, εκείνο που αλλάζει κάθε φορά είναι το σύστημα αρίθμησης. Η μέθοδος παράστασης των αριθμητικών ποσοτήτων, ονομάζεται αριθμητικό σύστημα αρίθμησης (Number System).

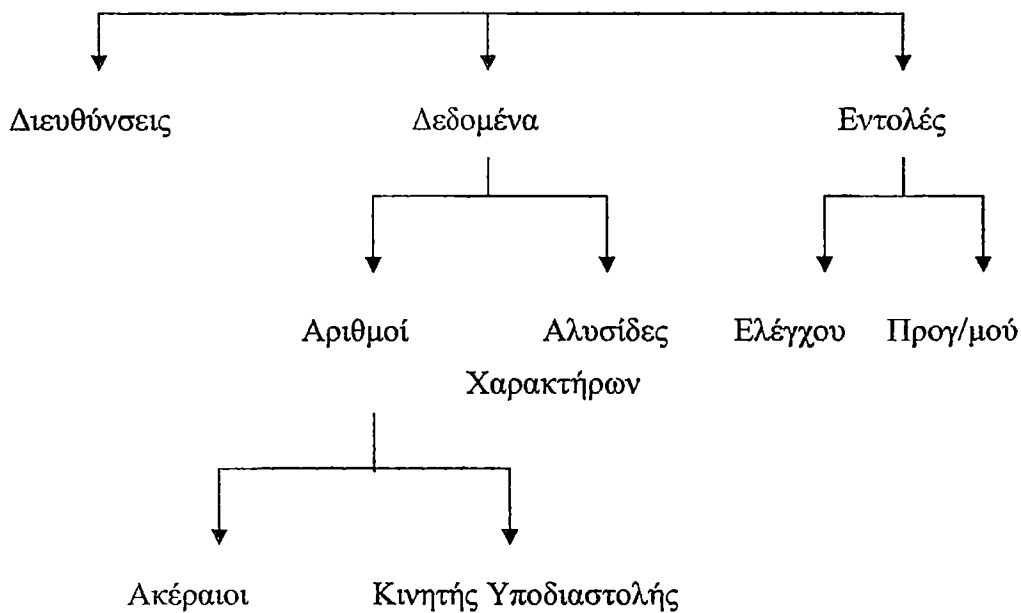
Στους υπολογιστές, η χρήση του δεκαδικού συστήματος θα δημιουργούσε προβλήματα κόστους και πολυπλοκότητας, γιατί θα έπρεπε να κατασκευαστούν στοιχειώδης διακόπτες με δέκα διαφορετικές καταστάσεις. Θα έπρεπε για παράδειγμα, να κατασκευαστούν τρανζίστορ με δέκα διαφορετικές καταστάσεις αντί για δύο. Για τον λόγο αυτό προτιμήθηκε το δυαδικό σύστημα, όπου η κατασκευή μηχανισμών με δύο καταστάσεις (on, off) ήταν πιο εύκολη και φθηνή.

Για παράδειγμα,

Από έναν αγωγό περνά ή δεν περνά ρεύμα, ένας πυκνωτής είναι φορτισμένος ή δεν είναι, ένας μαγνητικός δακτύλιος είναι μαγνητισμένος προς την μια ή την άλλη διεύθυνση, μια μονάδα είναι απασχολημένη ή δεν είναι, μία λυχνία ή ένα τρανζίστορ επιτρέπουν ή όχι την διέλευση ρεύματος κ.λ.π.

Όλες οι πληροφορίες που συνυπάρχουν μέσα σε ένα υπολογιστή αναλύονται σε μια ακολουθία από δυαδικά ψηφία.

Αν και η έννοια της πληροφορίας συνδέεται (λανθασμένα) με τα δεδομένα που επεξεργάζεται ένας υπολογιστής, πληροφορία είναι και οι εντολές που δέχεται ο υπολογιστής από το περιβάλλον. Με τις εντολές αυτές ο υπολογιστής ενεργεί πάνω στα δεδομένα και παράγει νέα. Οι εντολές είναι και τα σήματα ελέγχου που εκπορεύονται από την μονάδα ελέγχου προς τις υπόλοιπες μονάδες. Άλλη μια μορφή πληροφορίας είναι οι διευθύνσεις των δεδομένων και των εντολών μέσα στον υπολογιστή. Σε κάθε δεδομένο ή εντολή αντιστοιχεί πάντα και μια διεύθυνση. Αυτό είναι φυσικό, και έτσι συμβαίνει και στη καθημερινή μας ζωή.



### Τύποι Πληροφορίας

Υπάρχουν δύο τύποι δεδομένων. Οι αριθμοί κάθε τύπου (ακέραιοι, πραγματικοί ή κινητής υποδιαστολής, κ.λ.π.) και οι αλυσίδες χαρακτήρων. Ονομάζουμε αλυσίδα χαρακτήρων (String) μια συνεχή ακολουθία από απλούς χαρακτήρες (Characters)

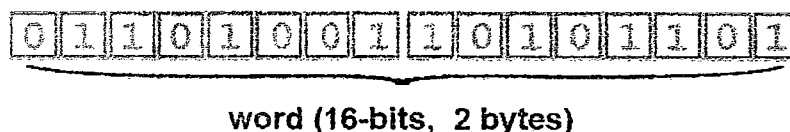
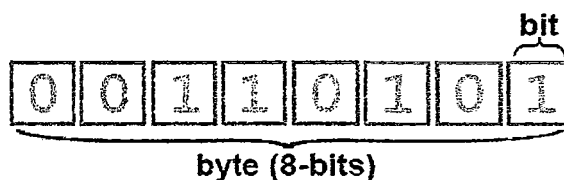
Οι βασικές μονάδες πληροφορίας δεδομένων, όπως άλλωστε και κάθε άλλο είδος πληροφορίας, πρέπει να μεταφερθούν στο δυαδικό σύστημα αρίθμησης, αφού είναι

το μοναδικό σύστημα αρίθμησης που "καταλαβαίνει" ο υπολογιστής. Η μεταφορά αυτή γίνεται κατά την φάση της εισόδου των δεδομένων. Πράγματι, τα δεδομένα πληκτρολογούνται στο δεκαδικό σύστημα αρίθμησης μέσω του πληκτρολογίου και πριν ο υπολογιστής τα επεξεργαστεί, το λειτουργικό σύστημα τα μεταφέρει στο δυαδικό σύστημα αρίθμησης. Το ίδιο γίνεται, αλλά με την αντίθετη λογική, κατά τη φάση της εξόδου των δεδομένων που παράχθηκαν από τον υπολογιστή.

Η πιο μικρή πληροφορία που μπορεί να παρασταθεί ηλεκτρονικά, μαγνητικά ή ηλεκτρομηχανικά είναι το bit (binary digit). Κάθε μορφής πληροφορία παριστάνεται εσωτερικά σαν μια αλυσίδα από διαδοχικά bits. Ένα πλήθος από bits συγκροτούν μεγαλύτερες μονάδες πληροφορίας, πιο περιεκτικές σε αξία. Μια τέτοια μονάδα πληροφορίας, από τις πιο βασικές, είναι ο χαρακτήρας (αν ο χαρακτήρας σχηματίζεται από 8 bits, ονομάζεται byte). Το μήκος ενός χαρακτήρα σε bits συνήθως είναι το οκτώ (άλλωστε byte σημαίνει οκτάδα). Ένα πλήθος από bytes συγκροτούν μια αμέσως επόμενη μονάδα, η οποία ονομάζεται λέξη (Word). Και εδώ το πλήθος των Bytes ανά λέξη ποικίλουν από αρχιτεκτονική σε αρχιτεκτονική (επόμενος πίνακας).

### Μήκος λέξης σε bits

Επειδή οι εντολές, τα δεδομένα και οι διευθύνσεις αναφέρονται στη μνήμη του υπολογιστή θα κάνουμε μια μικρή αναφορά σε αυτήν. Η κεντρική μνήμη δεν είναι τίποτε άλλο παρά μια αποθήκη που αποτελείται από ομοιόμορφους χώρους. Κάθε χώρος ονομάζεται θέση (Location) και στη θέση αυτή αντιστοιχεί μια διεύθυνση (Address). Κάθε θέση σχηματίζεται από ένα πλήθος από bits τα οποία υποδιαιρούνται σε bytes. Ένα πράγμα που πρέπει να συγκρατήσει κανείς είναι ότι κάθε θέση μνήμης μπορεί να περιέχει είτε ένα δεδομένο, είτε μία εντολή. Επίσης σε κάθε θέση μνήμης αντιστοιχούν δύο ποσότητες, το περιεχόμενο και η διεύθυνση. Αν κάποιος "κοιτάξει" σε μια δεδομένη στιγμή το περιεχόμενο μια τυχαίας διεύθυνσης θα "δει" μια σειρά από "0" και "1". Σε καμιά περίπτωση δεν θα μπορεί διαγνώσει αν πρόκειται για δεδομένο ή εντολή. Αυτό μπορεί να το κάνει αν εφοδιαστεί και με άλλες πληροφορίες.



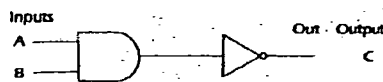
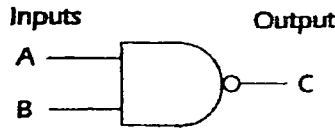
Πέρα όμως από το δυαδικό σύστημα χρησιμοποιούνται, αλλά σε βοηθητικό ρόλο, το οκταδικό και το δεκαεξαδικό σύστημα. Μοιραία λοιπόν θα συγκεντρώσουμε τη προσοχή μας σ' αυτά τα τρία συστήματα..



### 3.3 Λογικά Κυκλώματα

Πρόκειται για τα βασικά διακοπτόμενα κυκλώματα ή «πύλες» (gates) που χρησιμοποιούνται στους ψηφιακούς υπολογιστές και σε άλλες ψηφιακές ηλεκτρονικές συσκευές. Το σήμα εξόδου, που χρησιμοποιεί ένα δυαδικό σύστημα αρίθμησης (binary notation), ελέγχεται από το λογικό κύκλωμα σύμφωνα προς το σύστημα εισόδου (input). Τα τρία βασικά λογικά κυκλώματα είναι το “and”, το “or”

TRUTH TABLE FOR “NAND” GATE



και το “not”.

Το κύκλωμα “and” δίνει μια έξοδο (output) δυαδικού 1, αν κάποιο δυαδικό 1 υπάρχει σε κάθε κύκλωμα εισόδου. Διαφορετικά η έξοδος είναι ένα δυαδικό 0. Το κύκλωμα “or” παρέχει ένα δυαδικό 1 (στην είσοδο), αν ένα δυαδικό 1 υπάρχει τουλάχιστον σε ένα κύκλωμα εισόδου. Αλλιώς η έξοδος είναι δυαδικό 0. Τέλος το κύκλωμα “not” μετατρέπει το σήμα εισόδου, δίνοντας μια έξοδο δυαδικού 1 για μια είσοδο δυαδικού 0 ή μια έξοδο 0 για μια είσοδο 1.

LOGIC SYMBOL FOR “NAND” GATE

Inputs		Output
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

Συχνά αυτά τα βασικά λογικά κυκλώματα χρησιμοποιούνται σε συνδυασμό π.χ. ένα κύκλωμα “nand” αποτελείται από τα κυκλώματα “not” και “and”. Στα πλαίσια του ηλεκτρονικού εξοπλισμού τα λογικά κυκλώματα έχουν σχεδόν εντελώς ενσωματωθεί στα ολοκληρωμένα κυκλώματα.

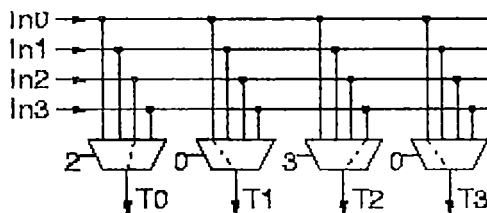
#### 3.3.1 Αποκωδικοποιητής και Πολυπλέκτης με Πύλες

Ο αποκωδικοποιητής είναι ένα ψηφιακό κύκλωμα που έχει μία έξοδο για κάθε ένα, χωριστό συνδυασμό τιμών των εισόδων του. Η κάθε έξοδος ανάβει (γίνεται 1) μόνον όταν οι τιμές των εισόδων βρίσκονται στον αντίστοιχο συνδυασμό. Επομένως, κάθε φορά, μία και μόνο μία έξοδος είναι αναμένη --αυτή η οποία αντιστοιχεί στον παρόντα συνδυασμό τιμών των εισόδων. Έτσι προκύπτει ότι ο πίνακας αληθείας της κάθε εξόδου, σε μορφή χάρτη Karnaugh, θα είναι παντού 0 εκτός ενός και μόνου

τετραγώνου όπου θα είναι 1. Άρα, η συνάρτηση της κάθε εξόδου αντιστοιχεί στο λογικό ΚΑΙ όλων των μεταβλητών εισόδου --είτε αυτών καθεαυτών, είτε των συμπληρωμάτων τους· κάθε έξοδος θα έχει διαφορετικό συνδυασμό αληθών/συμπληρωμάτων για τις μεταβλητές εισόδου.

Πολυπλέκτης (multiplexer, ή εν συντομία mux) είναι ένα κύκλωμα που επιλέγει μία από τις πολλές εισόδους του, και μεταφέρει (αντιγράφει) την τιμή (κατάσταση) της στην έξοδό του. Οι πολλαπλές υποψήφιες εισοδοί, των οποίων η εκάστοτε τιμή (ή κατάσταση, ή δεδομένα - data) μεταφέρονται και εμφανίζονται στην έξοδο, όταν και αν αυτές επιλεγούν, λέγονται εισοδοί δεδομένων (data inputs).

Η είσοδος που επιλέγεται μπορεί να αλλάζει δυναμικά, την ώρα της λειτουργίας του κυκλώματος, και για να γίνεται η επιλογή αυτή ο πολυπλέκτης έχει άλλες, επιπλέον εισόδους, τις εισόδους ελέγχου (control inputs) ή εισόδους επιλογής (select inputs). Το σύμβολο του πολυπλέκτη είναι ένα τραπέζιο, όπως φαίνεται δεξιά

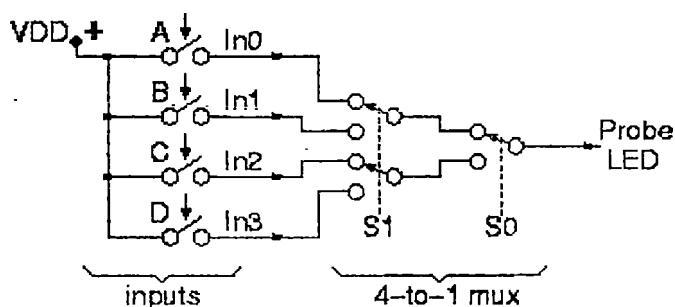


Οι πολυπλέκτες παίζουν κεντρικό ρόλο στα ψηφιακά συστήματα. Σημαντικές εφαρμογές τους είναι στην καρδιά κάθε δικτύου επικοινωνίας, εκεί όπου επιλέγεται το ποιός θα μιλήσει με ποιόν κάθε στιγμή. Αυτό μπορεί να είναι στο εσωτερικό ενός υπολογιστή ή άλλου ψηφιακού συστήματος, ή σ' έναν δρομολογητή (router) ή μεταγωγέα (switch) ενός δικτύου υπολογιστών ή τηλεπικοινωνιών. Στο σχήμα δεξιά φαίνεται ένα απλό παράδειγμα μεταγωγέα 4x4 τύπου crossbar (διασταυρούμενες ράβδοι) εμπνευσμένου από την κλασική τηλεφωνία. Είναι κατασκευασμένος από 4 πολυπλέκτες, καθένας μεγέθους 4-σε-1. Μπορούμε να φανταστούμε ότι οι 4 εισοδοί, In0 έως In3, είναι τα μικρόφωνα των τηλεφώνων τεσσάρων σπιτιών, οι δε 4 εξοδοί, T0 έως T3, τροφοδοτούν τα ακουστικά τεσσάρων άλλων τηλεφώνων. Ελέγχοντας κατάλληλα τα σήματα επιλογής των πολυπλεκτών πραγματοποιούμε τις τηλεφωνικές συνδέσεις που μας ζητώνται. Στο παράδειγμα, η είσοδος In2 έχει ζητήσει να μιλήσει στην έξοδο T0, και η In3 στην T2. Η είσοδος In0 μεταδίδεται σε περισσότερους από έναν ακροατές --στους T1 και T3-- κάτι γνωστό σαν "multicasting" στα δίκτυα. Η είσοδος In1 είναι αδρανής προς στιγμήν --ή τουλάχιστο κανείς δεν την ακούει.

Μιά άλλη σημαντική εφαρμογή των πολυπλεκτών είναι στις μνήμες. Οι μνήμες αποτελούνται από ένα μεγάλο πλήθος στοιχείων αποθήκευσης, που το καθένα τους "θυμάται" μία πληροφορία. Η ανάγνωση από τη μνήμη γίνεται μέσω ενός τεραστίου πολυπλέκτη ο οποίος έχει μία είσοδο δεδομένων συνδεδεμένη σε κάθε στοιχείο αποθήκευσης· έτσι, π.χ. αν η μνήμη έχει 1024 στοιχεία αποθήκευσης, ο πολυπλέκτης αυτός θα είναι μεγέθους 1024-σε-1. Όταν θέλουμε να διαβάσουμε μία από τις

πληροφορίες που βρίσκονται στη μνήμη, ρυθμίζουμε κατάλληλα τις εισόδους επιλογής του πολυπλέκτη ούτως ώστε αυτός να μας μεταφέρει στην έξοδο του την επιθυμητή πληροφορία.

Ακόμα μία εφαρμογή των πολυπλεκτών είναι στη "σειριακή" μετάδοση δεδομένων, στα δίκτυα επικοινωνίας. Στο πρώτο σχήμα παραπάνω, έστω ότι έχουμε τις τέσσερις δυαδικές τιμές In0, In1, In2, και In3 (δηλαδή έχουμε 4 bits), και θέλουμε να τις στείλουμε σε κάποιον μακρυνά, μέσω ενός και μόνο σύρματος, του Out (τα 4 σύρματα θα κόστιζαν υπερβολικά, λόγω απόστασης). Για να το πετύχουμε, αρκεί να αλλάξουμε διαδοχικά τα σήματα επιλογής του πολυπλέκτη από το 0 στο 1, στο 2, και στο 3, οπότε τα 4 bits από τις 4 εισόδους θα τοποθετηθούν διαδοχικά πάνω στο σύρμα Out και θα μεταδοθούν σειριακά (με τη σειρά) στον παραλήπτη.



### 3.3.2 Μεθοδολογία Αποσφαλμάτωσης (Debugging) Κυκλωμάτων

Καθώς αυξάνει η πολυπλοκότητα των κυκλωμάτων για υλοποίηση, αυξάνει και οι κίνδυνοι σφαλμάτων. Πρώτο σας μέλημα πρέπει πάντα να φτιάχνονται κυκλώματα σωστά εκ κατασκευής: σ' ένα πολύπλοκο σύστημα είναι ευκολότερο να προληφθούν τα λάθη από το να γίνει εκ των υστέρων προσπάθεια να βρεθεί σε τι οφείλονται λανθασμένες συμπεριφορές του συστήματος σε σπάνιες και παράξενες περιπτώσεις εισόδων του. Επειδή όμως τα λάθη είναι ανθρώπινα, μερικές φορές τυχαίνει να γίνουν μερικά, οπότε πρέπει να γίνεται μια μεθοδολογία ελέγχου (test) του αποτελέσματος (είναι το κύκλωμα σωστό;) και αν ο έλεγχος αυτός διαπιστώσει λάθη χρειάζεται μια μεθοδολογία εύρεσής τους (αποσφαλμάτωση - debugging). Επιπλέον, υπάρχουν και άλλοι αστάθμητοι παράγοντες, όπως π.χ. ότι ενδέχεται μερικά chips ή μερικοί ακροδέκτες από chips να είναι καμένα/καμένοι, ή μερικοί ακροδέκτες ή σύρματα να μην κάνουν καλή επαφή στην πλακέτα.

Η κατασκευή ξεκινάει πάντα με ένα καθαρό, κατανοητό σχεδιάγραμμα του κυκλώματός. Οι συνδέσεις γίνονται με την τροφοδοσία κλειστή. Μόλις ενεργοποιηθεί η τροφοδοσία, γίνεται έλεγχος υπερθέρμανσης. Αν κάποιο chip ζεσταίνεται απότομα, πρέπει να διακοπεί αμέσως η τροφοδοσία! Επίσης πρέπει να γίνει έλεγχος της τάσης. Πρέπει να δούμε αν είναι συνδεδεμένα στους ακροδέκτες.

Όταν γίνεται έλεγχος της τιμής (0 ή 1) ενός ακροδέκτη, είναι καλύτερα να γίνεται πάνω στον ίδιο ακροδέκτη του chip (αντί πάνω σε κάποιο σύρμα).

## Κεφάλαιο 4 : Αρχιτεκτονική υπολογιστικών συστημάτων

### 4.1 Κεντρική μνήμη

Η Κεντρική μνήμη ( main memory ), που αναφέρεται επίσης και ως πρωτεύουσα μνήμη ( primary memory ), δίνει σε ένα υπολογιστικό σύστημα τη δυνατότητα καταχώρησης και ανάκλησης :

A . των υπό εκτέλεση προγραμμάτων , που καταχωρούνται ως ακολουθίες πράξεων .

B . των πληροφοριακών δεδομένων επί των οποίων τα εκτελούμενα προγράμματα εφαρμόζονται .

Η κεντρική μνήμη των υπολογιστικών συστημάτων θεωρούμε ότι χρησιμοποιείται για τους ακόλουθους τέσσερις βασικούς λόγους :

- 1 . Για να καταχωρούνται τα πληροφοριακά δεδομένα , τα οποία εισάγονται στο υπολογιστικό σύστημα για άμεση επεξεργασία .
- 2 . Για να καταχωρούνται τα αποτελέσματα πράξεων , που εκτελούνται επί των πληροφοριακών δεδομένων , που ήδη είναι καταχωρημένα στη μνήμη .
- 3 . Για να καταχωρούνται αντίγραφα των πληροφοριακών δεδομένων , που ήδη υπάρχουν στην μνήμη , αλλά απαιτείται η αναπαραγωγή τους .
- 4 . Για να καταχωρούνται οι ακολουθίες των πράξεων , που πρόκειται να εκτελεστούν . Οι πράξεις καταχωρούνται σε μια κωδικοποιημένη μορφή , που μετά την αποκωδικοποίηση της , είναι άμεσα εκτελέσιμοι από την αριθμητική και λογική μονάδα του υπολογιστικού συστήματος . Η μορφή με την οποία καταχωρούνται οι πράξεις αποτελεί μια τεχνική γλώσσα . Η γλώσσα αυτή , την οποία θεωρητικά " αντιλαμβάνεται " το υπολογιστικό σύστημα ( δηλαδή η μηχανή ) , ονομάζεται , για το λόγο αυτό , " γλώσσα μηχανής " ( machine language ) . Τα προγράμματα , όπως έχει αναφερθεί , αποτελούνται από οδηγίες ( η εντολές ) προς τη μηχανή , για τις υπό εκτέλεση πράξεις . Αποτελούνται λοιπόν από ακολουθίες εντολών που καθορίζουν τις πράξεις , τις οποίες η μηχανή πρόκειται να εκτελέσει .

#### 4.1.1 Τρόποι Διαχωρισμού Μνημών

1. Μόνο ανάγνωσης -Ανάγνωσης & Εγγραφής
2. Στατικές -Δυναμικές
3. Τυχαίας -Ακολουθιακής Προσπέλασης
4. Πρόσκαιρες (volatile) ή Μη (non-volatile)

Η μνήμη ενός ηλεκτρονικού υπολογιστικού συστήματος μοιάζει με μια κυψέλη , που αποτελείται από ένα οργανωμένο πλήθος κυψελίδων , ή κελιών , ή στοιχείων . Κάθε στοιχείο της μνήμης μπορεί να βρίσκεται σε μια από δυο δυνατές καταστάσεις , ή μεταφορικά , να περιέχει μια από δυο δυνατές τιμές .

Παλαιότερα , η υλοποίηση των στοιχείων της μνήμης πραγματοποιείται με μικροσκοπικούς μαγνητικούς δακτυλίους . Οι οποίοι εμαγνητίζοντο είτε δεξιόστροφα , για να αποδώσουν τη μια κατάσταση , είτε αριστερόστροφα για αποδώσουν την άλλη .

Η μνήμη που αποτελείται από μαγνητικούς δακτυλίους , ονομάζεται μαγνητική μνήμη ( core ) . Η ανάκληση της τιμής που περιέχει ένας μαγνητικός δακτύλιος πραγματοποιείται μέσω ενός δευτέρου αγωγού , αλλά αυτό έχει σαν αποτέλεσμα την απώλεια της τιμής του και για το λόγο αυτό απαιτείται ένας επί πλέον αγωγός , για να επαναφέρει στο δακτύλιο την προηγούμενη τιμή του .

Ο μαγνητισμός των μαγνητικών δακτυλίων ήταν μόνιμος και μπορούσε να μεταβληθεί με αλλαγή της φοράς του ρεύματος του αγωγού που τον διαπερνούσε . Δεν ήταν συνεπώς αναγκαία η διατήρηση της μαγνητικής μνήμης με συνεχή τροφοδοσία ηλεκτρικού ρεύματος . Απλώς η μαγνητική μνήμη ήταν δυνατόν να παραμένει αμετάβλητη και να κρατά συνεπώς πληροφοριακά δεδομένα για όσο διάστημα ήταν επιθυμητό .

Η μαγνητική μνήμη και γενικά κάθε είδους μνήμη που δεν απαιτεί τροφοδοσία για να διατηρήσει τα στοιχεία που περιέχει , ονομάζεται " σταθερή μνήμη " .

Σήμερα τα στοιχεία μνήμης υλοποιούνται μέσω ημιαγωγικών ολοκληρωμένων κυκλωμάτων πυριτίου που έχουν σαν ελάχιστο στοιχείο το " flip flop " . Το flip flop θεωρείται όπως και ο μαγνητικός δακτύλιος , ως το ελάχιστο στοιχείο μνήμης .

Βασική διαφορά των **flip flops** από τους μαγνητικούς δακτυλίους , είναι :

- A . Η μεγάλη τους ταχύτητα
- B . Ο μικρός τους όγκος
- Γ . Δεν είναι σταθερά ( δηλαδή η ημιαγωγική μνήμη είναι ασταθής )

### Τυχαία και ακολουθιακή προσπέλαση

Οι μονάδες μνήμης μπορούν να διαχωριστούν σε μονάδες τυχαίας ( ή άμεσης ) προσπέλασης και μονάδες ακολουθιακής ( ή σειριακής ) προσπέλασης . Ο τρόπος προσπέλασης μιας μονάδας μνήμης καθορίζεται από τον τρόπο με τον οποίο μπορούν να αναζητηθούν και να ανακληθούν οι πληροφορίες που περιέχει η συγκεκριμένη μονάδα μνήμης . Ο χρόνος προσπέλασης ( access time ) μιας μνήμης είναι ο χρόνος που

περνάει από τη στιγμή που επιλέγεται μια θέση μνήμης, μέχρι τη στιγμή που το περιεχόμενο της θέσης αυτής έχει διαβαστεί η γραφτεί.

Μνήμες τυχαίας προσπέλασης: είναι κατασκευασμένες από ολοκληρωμένα κυκλώματα ημιαγωγών, δεν διαθέτουν κινούμενα μέρη και ο χρόνος προσπέλασης μιας θέσης μνήμης είναι πάντα ο ίδιος και είναι ανεξάρτητος από τη συγκεκριμένη θέση της μνήμης.

Μνήμες ακολουθιακής προσπέλασης: οι θέσεις μνήμης δεν είναι άμεσα διαθέσιμες και η πρόσβαση σε αυτές απαιτεί τη χρήση κινούμενων μερών (κεφαλές ανάγνωσης/εγγραφής). Για το λόγο αυτό, ο χρόνος προσπέλασης σε μια θέση μνήμης δεν είναι σταθερός, αλλά εξαρτάται από τη θέση της συγκεκριμένης θέσης μνήμης ως προς την κεφαλή.

### Στατικές –Δυναμικές

Τα ολοκληρωμένα κυκλώματα , που χρησιμοποιούνται στην κεντρική μνήμη των υπολογιστικών συστημάτων διακρίνονται στα :

1. δυναμικά
2. στατικά

και η αντίστοιχη μνήμη διακρίνεται σε δυναμική και στατική .

Η δυναμική μνήμη είναι ασταθής εξαιτίας της απώλειας του φορτίου του πυκνωτή και απαιτεί περιοδική αναπαραγωγή του φορτίου για να διατηρείται .

Η στατική μνήμη είναι και αυτή ασταθής αλλά δεν απαιτεί αναπαραγωγή των τιμών που περιέχουν τα στοιχεία της . Απαιτεί όμως τροφοδοσία με ρεύμα . Τα στοιχεία της στατικής μνήμης είναι πιο πολύπλοκα . Για το λόγο αυτό η στατική μνήμη χρησιμοποιείται σε ειδικές περιπτώσεις , ενώ η δυναμική μνήμη χρησιμοποιείται κυρίως στην κεντρική μνήμη των υπολογιστικών συστημάτων .

### **Πρόσκαιρη και μη πρόσκαιρη αποθήκευση**

Οι μνήμες τα περιεχόμενα των οποίων χάνονται όταν πάψουμε να τις τροφοδοτούμε με ηλεκτρικό ρεύμα ονομάζονται "πρόσκαιρες" (*volatile*). Τα ολοκληρωμένα κυκλώματα των στατικών και δυναμικών RAM ανήκουν σε αυτή την κατηγορία αφού τα κύτταρα μνήμης τους χρειάζονται εξωτερική ηλεκτρική ισχύ για να διατηρήσουν την αποθηκευμένη πληροφορία.

Αντίθετα οι "μη πρόσκαιρες μνήμες" (*no volatile*) όπως είναι οι βοηθητικές μνήμες, διατηρούν τις πληροφορίες τους και χωρίς την ύπαρξη ηλεκτρικού ρεύματος.

Αυτό συμβαίνει διότι στις μνήμες αυτές οι πληροφορίες αποθηκεύονται με μαγνητικό ή οπτικό και δεν επηρεάζονται από την διακοπή της ηλεκτρικής ισχύος.

Άλλο παράδειγμα μη πρόσκαιρης μνήμης είναι η μνήμη μόνο-ανάγνωσης (ROM).

Οι μη πρόσκαιρες μνήμες είναι απαραίτητες στους υπολογιστές για να μπορούμε να αποθηκεύουμε προγράμματα και δεδομένα που δεν χρησιμοποιούνται άμεσα αλλά θα τα χρειαστούμε στο μέλλον. Προγράμματα και δεδομένα που είναι απαραίτητα κατά τη διαδικασία εκκίνησης ενός υπολογιστή και δεν πρέπει να μεταβληθούν ποτέ αποθηκεύονται σε ROM. Ο υπολογιστής μπορεί να χρησιμοποιήσει τα περιεχόμενα της ROM όταν παρέχεται σε αυτόν ισχύ. Όταν ο υπολογιστής είναι σε λειτουργία και όταν όταν χρειαστεί τα υπόλοιπα προγράμματα και δεδομένα που είναι αποθηκευμένα σε βοηθητικές μνήμες μπορούν να μεταφερθούν στην κύρια μνήμη (RAM) του υπολογιστή. Επίσης πριν την διακοπή της λειτουργίας του υπολογιστή τα περιεχόμενα της κύριας μνήμης, θα πρέπει να μεταφερθούν σε μια βοηθητική μνήμη εάν η πληροφορία αυτή πρέπει να φυλαχτεί.



Η κύρια και η κρυφή μνήμη είναι τύπου τυχαίας προσπέλασης

ενώ οι βοηθητικές μνήμες είναι τύπου ακολουθιακής προσπέλασης.

#### 4.1.2 Οργάνωση της Μνήμης

Οι δύο διαφορετικές καταστάσεις που είναι δυνατό να βρίσκεται το ελάχιστο στοιχείο μνήμης συμβολίζεται με τα δυαδικά ψηφία 0 και 1. Θεωρούμε ότι τα δυαδικά ψηφία που αποτελούν το σύνολο { 0, 1 } είναι ένα αλφάβητο δύο πρακτικών συμβόλων .

Διάφορα στοιχεία μνήμης είναι :

- A . To bit
- B . Η λέξη
- Γ . To byte

#### Bit

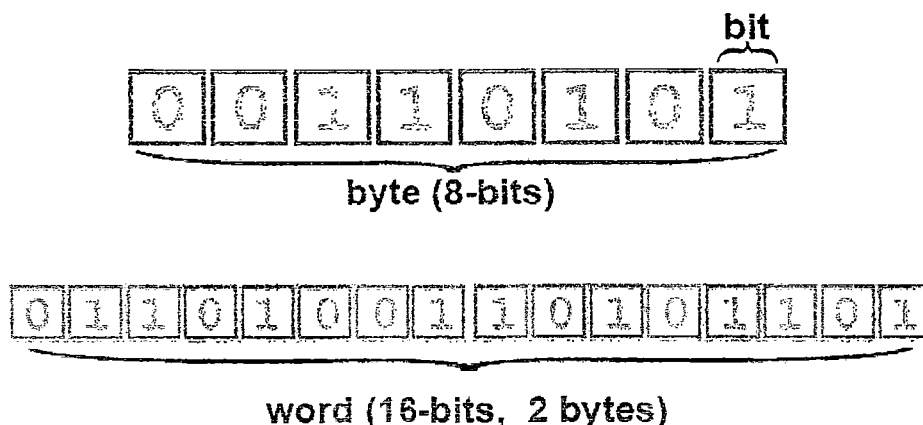
Το ελάχιστο στοιχείο μνήμης του pc μπορεί να αποδώσει και να αποθηκεύσει δύο απλές καταστάσεις όπου αντιστοιχούν σε ένα δυαδικό ψηφίο . Για το λόγο αυτό το ελάχιστο στοιχείο μνήμης ονομάζεται bit . Τα όνομα bit προέρχεται από το πρώτο και τα δύο τελευταία γράμματα του όρου binary digit που σημαίνει δυαδικό ψηφίο .

### Λέξη

Μία λέξη είναι μια ακολουθία από bit . Το πλήθος  $N$  των bit που περιέχονται σε μια λέξη είναι χαρακτηριστικό για κάθε είδος μηχανής . Μία λέξη μπορεί να αποδώσει μέχρι  $2^N$  το πολύ διαφορετικές καταστάσεις οι οποίες αντιστοιχούν στους δυνατούς συνδυασμούς των καταστάσεων των  $N$  bit . Ο χαρακτηριστικός αριθμός  $N$  που αποδίδει το πλήθος των bit ανά λέξη ποικίλλει στους διάφορους τύπους μηχανών .

### Byte

Είναι μια τυπική υποδιαίρεση των λέξεων που αποτελεί την πιο συνηθισμένη ομαδοποίηση των bit , είναι μια ακολουθία από bit που ονομάζεται byte . Σύμφωνα με αυτό μια λέξη των 8 bit αποτελείται από 1 byte .



## 4.2 Αριθμητική Λογική Μονάδα

Ένα ηλεκτρονικό υπολογιστικό σύστημα έχει τη δυνατότητα να εκτελεί αριθμητικές και λογικές πράξεις , να καταχωρεί στην κεντρική του μνήμη



δεδομένα και ακολουθίες εντολών να μεταφέρει πληροφοριακά δεδομένα και να εισάγει πληροφοριακά δεδομένα από και προς το περιβάλλον του .

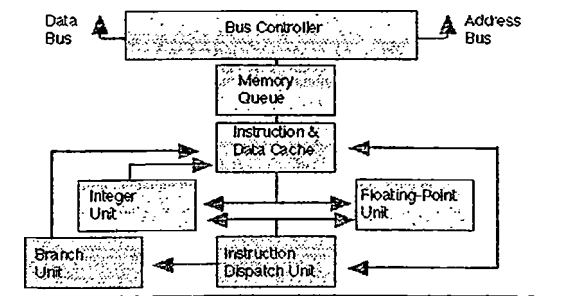
Οι συνηθισμένες αριθμητικές πράξεις όπως η πρόσθεση η αφαίρεση ο πολλαπλασιασμός και η διαίρεση θεωρούνται ως απλές . Υπάρχουν όμως και περισσότερο πολύπλοκες όπως οι υπολογισμοί των τιμών των διαφόρων συναρτήσεων δηλαδή εξαγωγή τετραγωνικής ρίζας , ύψωση σε δύναμη καθώς και άλλες τριγωνομετρικές και μαθηματικές συναρτήσεις . Στις λογικές πράξεις περιλαμβάνονται οι συγκρίσεις , οι συνδέσεις λογικών παραστάσεων με τους λογικούς τελεστές και η πραγματοποίηση επιλογών μεταξύ ακολουθιών εντολών . Ο απλός χρήστης έχει την εντύπωση ότι όλες οι πράξεις εκτελούνται άμεσα από το υπολογιστικό σύστημα . Στην πραγματικότητα οι διαδικασίες εκτέλεσης των απλών ή και πολύπλοκων αριθμητικών και λογικών πράξεων και γενικά όλων των υπολογισμών διατυπώνονται πάντα πριν από την εκτέλεση τους στη γλώσσα της συγκεκριμένης μηχανής στην οποία πρόκειται να εκτελεστούν . Η γλώσσα αυτή είναι στοιχειώδης , περιλαμβάνει πολύ απλές εντολές οι οποίες εκτελούνται με την απλή αυτή μορφή και οδηγούν στους επιθυμητούς υπολογισμούς . Όλες οι πράξεις όμως , είτε αριθμητικές , είτε λογικές , πραγματοποιούνται στην Αριθμητική και Λογική μονάδα ( Arithmetic and Logical Unit ) του ηλεκτρονικού υπολογιστικού συστήματος .

### **Μονάδες εκτέλεσης ακεραίων**

Η περισσότερη από τη δουλειά που γίνεται σε ένα υπολογιστή γίνεται με ακέραιες πληροφορίες, κάτι που σημαίνει ότι ακέραια νούμερα και δεδομένα αναπαρίστανται από ακέραιους αριθμούς. Οι ακέραιοι περιλαμβάνουν ακέραιους αριθμούς, χαρακτήρες και άλλα παρόμοια δεδομένα. Οι μη ακέραιοι αριθμοί ονομάζονται αριθμοί κινητής υποδιαστολής και χειρίζονται διαφορετικά από τη μονάδα κινητής υποδιαστολής. Η μονάδα ακεραίων μπορεί σε ορισμένους επεξεργαστές να χειριστεί και λειτουργίες κινητής υποδιαστολής, αλλά πολύ πιο αργά συγκρινόμενη με τη μονάδα κινητής υποδιαστολής.

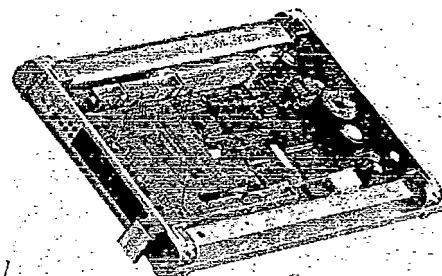
Η μονάδα εκτέλεσης ακεραίων είναι εκεί που τελικά εκτελούνται οι εντολές. Παλαιότεροι επεξεργαστές είχαν μόνο μια τέτοια μονάδα και οι εντολές εκτελούνταν σειριακά. Οι πιο καινούριοι έχουν διάφορες τέτοιες μονάδες, επιτρέποντας σε περισσότερες από μια εντολές να εκτελούνται ταυτόχρονα, επιτρέποντας έτσι αύξηση της απόδοσης. Οι επεξεργαστές που μπορούν να το κάνουν αυτό ονομάζονται υπερβαθμωτοί (superscalar). Πιο εξελιγμένοι επεξεργαστές μπορεί να έχουν εξειδικευμένες τέτοιες μονάδες, σχεδιασμένες για την εκτέλεση συγκεκριμένων τύπων εντολών.

**Μονάδα κινητής υποδιαστολής (Floating Point Unit - FPU) και συνεπεξεργαστής (coprocessor)**



Η μονάδα κινητής υποδιαστολής είναι μια εξειδικευμένη μονάδα εκτέλεσης πράξεων μεταξύ αριθμών κινητής υποδιαστολής. Αριθμός κινητής υποδιαστολής είναι ο οποιοσδήποτε αριθμός εκτός από ακέραιος. Κάθε αριθμός με δεκαδική τελεία αναπαρίστανται σαν αριθμός κινητής υποδιαστολής.

Οι πρώτοι επεξεργαστές μπορούσαν να χειριστούν άμεσα, μόνο ακέραιους αριθμούς. Ο χειρισμός των πραγματικών αριθμών γινόταν μέσω υπορουτίνων που φυσικά επιβάρυναν την ταχύτητα επεξεργασίας. Όταν χρειάζονταν περισσότερη μαθηματική δύναμη αναγκάζονταν να χρησιμοποιήσουν ένα μαθηματικό συνεπεξεργαστή (math coprocessor), μια μονάδα κινητής υποδιαστολής (Float Point Unit), που αναλάμβανε την εκτέλεση πολύπλοκων μαθηματικών υπολογισμών με μεγάλη ταχύτητα. Αργότερα ο συνεπεξεργαστής ήταν ενσωματωμένος στον επεξεργαστή.



Συσκευή FPU

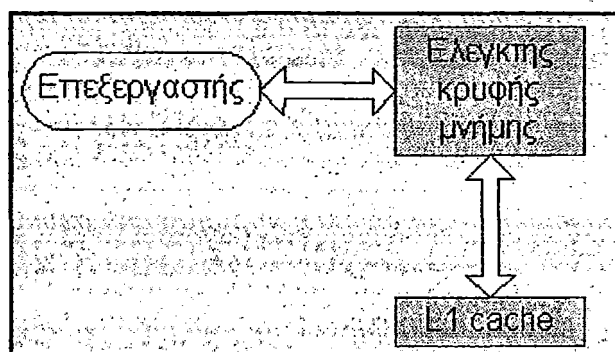
Η χρησιμοποίηση του συνεπεξεργαστή αυξάνει την απόδοση του υπολογιστή όταν εκτελούνται προγράμματα που χρησιμοποιούν αριθμούς κινητής υποδιαστολής, όπως είναι τα προγράμματα τρισδιάστατων γραφικών, σαν το AutoCad. Στην περίπτωση όμως που εκτελούνται προγράμματα γραφείου, τα οποία δε χρησιμοποιούν αριθμούς κινητής υποδιαστολής, έτσι ώστε να χρησιμοποιείται ο συνεπεξεργαστής, η παρουσία του δεν αυξάνει την απόδοση του συστήματος.

**Πρώτου επιπέδου κρυφή μνήμη (L1 cache) και ελεγκτής κρυφής μνήμης**

Όλοι οι σύγχρονοι επεξεργαστές έχουν μια μικρή, υψηλής ταχύτητας κρυφή μνήμη κοντά στο chip, για να κρατάει τα πιο πρόσφατα χρησιμοποιημένα δεδομένα και εντολές από τη μνήμη. Μια αρχή της επιστήμης των υπολογιστών που αποκαλείται τοπικότητα ισχυρίζεται ότι αν ο επεξεργαστής αναφέρθηκε σε μια περιοχή της μνήμης, είναι πολύ πιθανό να αναφερθεί ξανά σε αυτή πολύ σύντομα. Χρησιμοποιώντας μια κρυφή μνήμη για να κρατάει τις πιο πρόσφατα

χρησιμοποιημένες τιμές της μνήμης, ο επεξεργαστής δε χρειάζεται κάθε φορά να πηγαίνει σε αυτή, για να τις φορτώσει. Αυτό παρέχει μια σημαντική αύξηση της απόδοσης, καθώς η μνήμη του συστήματος είναι πολύ πιο αργή από την κρυφή μνήμη του επεξεργαστή.

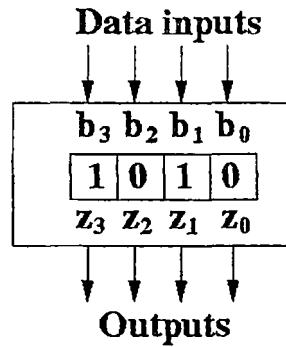
Η κρυφή μνήμη του επεξεργαστή ονομάζεται βασική ( ή πρώτου επιπέδου) καθώς είναι η πιο κοντινή μνήμη στον επεξεργαστή. Κάθε φορά που ο επεξεργαστής απαιτεί πληροφορίες από τη μνήμη ο ελεγκτής της κρυφής μνήμης χρησιμοποιεί ειδικά κυκλώματα για να ελέγξει αν τα δεδομένα είναι ήδη στην κρυφή μνήμη. Αν είναι, ο επεξεργαστής εξοικονομεί το χρόνο της πρόσβασης στην κύρια μνήμη. Οι περισσότεροι επεξεργαστές χρησιμοποιούν επίσης μια δευτερεύουσα (δευτέρου επιπέδου) κρυφή μνήμη, για να κρατάει κάποια δεδομένα που δεν χωρούν στην βασική κρυφή μνήμη.



Η κρυφή μνήμη του επεξεργαστή

Η τοπική βασική κρυφή μνήμη του επεξεργαστή ποικίλλει σε μέγεθος από 8KB μέχρι 64KB, με μεγαλύτερα μεγέθη στους πιο καινούριους επεξεργαστές. Αυτή η μνήμη είναι ταχύτερη καθώς τρέχει με την ταχύτητα του επεξεργαστή και είναι ενσωματωμένη μέσα σε αυτόν. Επιπρόσθετα, πολλές βασικές κρυφές μνημές είναι συσχετίσιμες συνόλων, που βελτιώνει την πιθανότητες παρουσίας στην κρυφή μνήμη. Υπάρχουν δύο τρόποι που ο επεξεργαστής μπορεί να οργανώσει τη βασική του κρυφή μνήμη. Κάποιοι επεξεργαστές έχουν μια απλή κρυφή μνήμη για να διαχειρίζονται και τις εντολές και τα δεδομένα των προγραμμάτων. Αυτή ονομάζεται ενιαία κρυφή μνήμη. Άλλοι επεξεργαστές έχουν διαφορετική κρυφή μνήμη για τα δεδομένα και τις εντολές. Σε κάποιες περιπτώσεις οι δύο αυτές διαφορετικές μορφές μπορεί να έχουν τελείως διαφορετικές ικανότητες.

### 4.3 Καταχωρητές ( registers ) της ALU



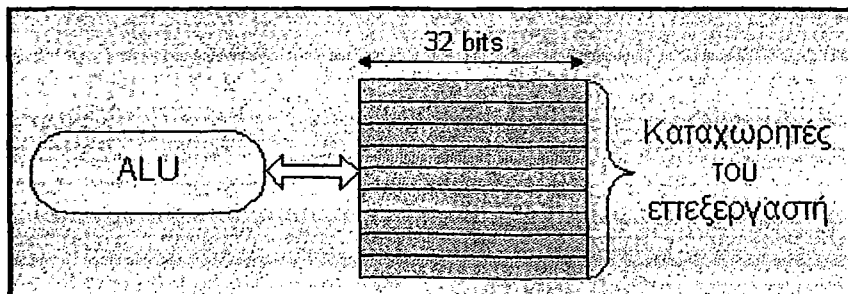
Βασικό Διάγραμμα Λειτουργίας ενός Καταχωρητή

Καταχωρητής: είναι μία ομάδα από δυαδικά κύτταρα αποθήκευσης και από λογικές πύλες που διεκπεραιώνουν τη μεταφορά πληροφοριών.

Οι καταχωρητές είναι τοπικές περιοχές μνήμης μέσα στον επεξεργαστή, που χρησιμοποιούνται για να κρατούν δεδομένα, τα οποία επεξεργάζονται εκείνη τη στιγμή. Κάθε επεξεργαστής έχει μερικούς καταχωρητές, κάποιους αφοσιωμένους σε μια συγκεκριμένη λειτουργία και άλλους διαθέσιμους για γενική χρήση από τους προγραμματιστές. Οι καταχωρητές είναι η ταχύτερη μνήμη, διαθέσιμη για χρήση στους υπολογιστές, ακόμα πιο γρήγορη και από την L1 cache.

Οι περισσότερες λειτουργίες γίνονται στους καταχωρητές. Ο επεξεργαστής δεν μπορεί για παράδειγμα να εκτελεί πράξεις στη μνήμη. Αν θέλουμε να προσθέσουμε τον αριθμό 1 σε μια περιοχή της μνήμης, ο επεξεργαστής κανονικά θα το κάνει αυτό, φορτώνοντας την αρχική τιμή από τη μνήμη στον καταχωρητή, προσθέτοντας τον αριθμό 1 στον καταχωρητή και σώζοντας μετά την τιμή πίσω στη μνήμη. Αυτό φυσικά γίνεται πάρα πολύ γρήγορα και χωρίς να το καταλαβαίνει ο χρήστης.

Το πλάτος (σε bits) των καταχωρητών του επεξεργαστή καθορίζει πόσα δεδομένα μπορεί να επεξεργαστεί σε μια χρονική στιγμή. Αυτό ορισμένες φορές χρησιμοποιείται για να προσδιορίσει το "μέγεθος" του επεξεργαστή. Για παράδειγμα, μπορεί να ακούσετε να μιλούν για "16 bit επεξεργαστή" ή "32 bit επεξεργαστή". Αυτός ο όρος συνήθως αναφέρεται στο μέγεθος του καταχωρητή μέσα στον επεξεργαστή. Ωστόσο, αυτός ο όρος κάποιες φορές χρησιμοποιείται λανθασμένα και κάποιος αναφέροντας το μέγεθος του επεξεργαστή, εννοούν το εύρος του διαύλου του.



Οι καταχωρητές του επεξεργαστή

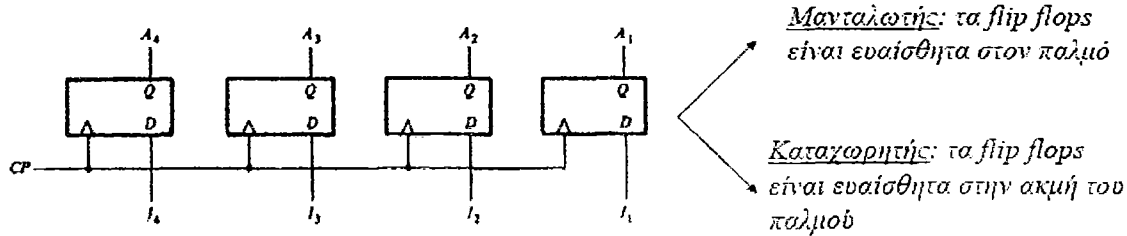
Όσους περισσότερους καταχωρητές έχει ο επεξεργαστής, τόσο μεγαλύτερη ευκινησία έχουν οι προγραμματιστές να γράψουν καλύτερο κώδικα. Όμως, αυτό αυξάνει την πολυπλοκότητα του επεξεργαστή.

Στην ALU περιέχονται ένας μικρός αριθμός θέσεων μνήμης που λέγονται καταχωρι τές, καθώς και εκείνα τα ηλεκτρονικά κυκλώματα που απαιτούνται για την εκτέλεση των πράξεων. Οι καταχωρι τές που βρίσκονται στην ALU χρησιμοποιούνται για την προσωρινή καταχώριση των τιμών, που πρόκειται να χρησιμοποιηθούν στους υπό εκτέλεση υπολογισμούς. Χρησιμοποιούνται επίσης και για προσωρινή καταχώριση των αποτελεσμάτων των υπολογισμών, πριν από τις καταχωρίσεις τους στην κεντρική μνήμη. Συνήθως, ένας ή περισσότεροι από τους καταχωρι τές της ALU, που χρησιμοποιούνται στην εκτέλεση πράξεων, αλλά έχουν μεγαλύτερες δυνατότητες από τους υπόλοιπους καταχωρι τές, ονομάζονται συσσωρευτές.

- Το μοντέλο καταχωρητών ειδικής χρήσης
- Καταχωρητές δεδομένων
  - EAX (AX (AH/AL)): Συσσωρευτής (*Accumulator*)
  - EBX (BX (BH/BL)): Βάση μνήμης
  - ECX (DX (CH/CL)): Μετρητής
  - EDX (DX (DH/DL)): Πολλαπλασιασμός, διαίρεση
- Καταχωρητές διευθύνσεων
  - SP: (stack pointer) Δείκτης στοίβας (*Stack pointer*)
  - BP: (base pointer) Βάση μνήμης
  - SI: (source index) Δείκτης προέλευσης
  - DI: (desination index) Δείκτης προορισμού
- Καταχωρητές τμημάτων μνήμης
  - CS: (code segment) Τμήμα εντολών
  - DS: (data segment) Τμήμα δεδομένων
  - SS: (stack segment) Τμήμα στοίβας
  - ES: (extra segment) Πρόσθετο τμήμα
  - FS: (extra segment) Πρόσθετο τμήμα
  - GS: (extra segment) Πρόσθετο τμήμα
- Καταχωρητές ελέγχου
  - IP: (instruction pointer) Μετρητής εντολής
  - FLAGS: Ενδείκτες (*Flags*)

## Καταχωρητής

Ο απλούστερος δυνατός τύπος καταχωρητή αποτελείται μονάχα από flip-flops χωρίς εξωτερικές πύλες.



#### 4.4 Μετρητές

Οι μετρητές είναι καταχωρητές που παίρνουν διαδοχικά μια προκαθορισμένη σειρά καταστάσεων.

Μετρητής Ριπής: Οι εισοδοί CP όλων των flip flops εκτός του πρώτου πυροδοτούνται από τις ακμές των κυματομορφών που βγαίνουν από τα άλλα flip flops

Σύγχρονος Μετρητής: Οι εισοδοί CP όλων των flip flops τροφοδοτούνται από τους ίδιους παλμούς ρολογιού.

#### 4.5 Μονάδα ελέγχου

Η μονάδα ελέγχου είναι ένα κύκλωμα που ελέγχει τη ροή των πληροφοριών προς τον επεξεργαστή. Κατά κάποιο τρόπο είναι το "μυαλό μέσα στο μυαλό", καθώς ελέγχει ότι γίνεται μέσα στον επεξεργαστή, ο οποίος ελέγχει τον υπόλοιπο υπολογιστή.

Οι λειτουργίες που εκτελούνται από τη μονάδα ελέγχου ποικίλλουν δραστικά από την εσωτερική αρχιτεκτονική του επεξεργαστή, καθώς η μονάδα ελέγχου εκτελεί αυτή την αρχιτεκτονική. Σε ένα κανονικό επεξεργαστή, που εκτελεί τις x86 εντολές η μονάδα ελέγχου εκτελεί τις λειτουργίες της ανάκλησης, της αποκωδικοποίησης, της εκτέλεσης των εντολών και μετά την αποθήκευση των αποτελεσμάτων. Σε ένα επεξεργαστή με πυρήνα RISC, η μονάδα ελέγχου έχει σαφώς περισσότερο έργο να κάνει. Διαχειρίζεται τη μετατροπή των x86 εντολών σε RISC μικροεντολές και διαμοιράζει τις μικροεντολές μεταξύ των διαφόρων μονάδων εκτέλεσης. Σε ένα τέτοιο επεξεργαστή η μονάδα ελέγχου μπορεί να είναι διαχωρισμένη σε μικρότερες μονάδες εξαιτίας της πολυπλοκότητας της εργασίας που πρέπει να εκτελέσει.

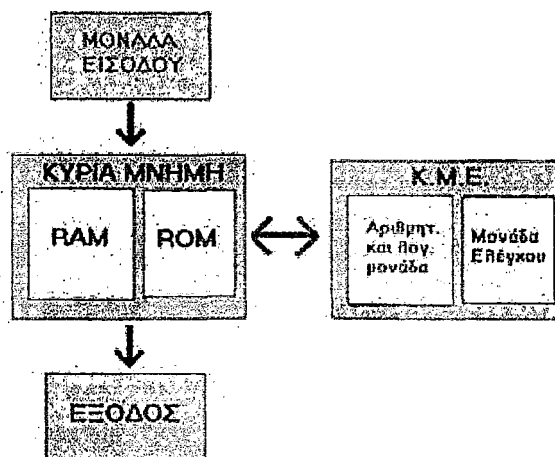
Η μονάδα ελέγχου και η αριθμητική και λογική μονάδα αποτελούν μαζί την κεντρική μονάδα επεξεργασίας ενός ηλεκτρονικού υπολογιστικού συστήματος.

Οι βασικές λειτουργίες της μονάδας ελέγχου είναι :

- Ο συντονισμός όλων των λειτουργιών με σήματα ελέγχου
- Ο προσδιορισμός της εκάστοτε επόμενης προς εκτέλεση εντολής
- Η μεταφορά της υπό εκτέλεση εντολής στη ΜΕ
- Η αποκωδικοποίηση της υπό εκτέλεση εντολής
- Η επίβλεψη των λειτουργιών κατά την εκτέλεση της εντολής
- Ο έλεγχος των διαδικασιών μνήμης, καταχώρησης και ανάκλησης.

Η ομάδα ελέγχου είναι η οργανωτική δύναμη στον Η/Υ και αποτελείται από δύο κύριους καταχωρητές.

1. Καταχωρητής εντολών (instruction register): Είναι ένας καταχωρητής ο οποίος δέχεται όλες τις εντολές του προγράμματος από την μνήμη μια προς μια όπου αναγνωρίζονται και αναλύονται σε επί μέρους εργασίες και τέλος εκτελούνται.
2. Μετρητής προγράμματος (program counter): Ο μετρητής διαθέτει τη διεύθυνση της επόμενης εντολής προς εκτέλεση. Η εντολή μεταφέρεται στον καταχωρητή εντολών (IR) και μετά ο μετρητής προγράμματος θα αυξηθεί κατά 1 για να υποδείξει την επόμενη εντολή προς εκτέλεση.



Ο ρόλος της Μονάδας Ελέγχου στον Ηλεκτρονικό Υπολογιστή

Κατά την εκτέλεση ενός προγράμματος, η ΜΕ λαμβάνει από τη μνήμη ορισμένα κωδικοποιημένα ηλεκτρικά σήματα, τα οποία αντιστοιχούν σε εντολές. Τα κυκλώματα της μονάδας ελέγχου πρώτα "αποκωδικοποιούν" τις εντολές αυτές και στη συνέχεια εκπέμπουν σήματα ελέγχου προς:

- Την μνήμη για την εξαγωγή πληροφοριών προς τις άλλες μονάδες του υπολογιστή ή την λήψη πληροφοριών απ'αυτές και την αποθήκευσή τους στη μνήμη
- Την Αριθμητική & Λογική μονάδα για την εκτέλεση των επιθυμητών υπολογιστικών πράξεων
- Τις μονάδες εισόδου/εξόδου, για τη μεταφορά πληροφοριών από τον υπολογιστή προς το εξωτερικό του περιβάλλον, και αντιστρόφως

Κατά την εκτέλεση ενός προγράμματος που σύμφωνα με τα γνωστά , αποτελείται από μια ακολουθία από εντολές , πραγματοποιούνται τα εξής . Κάθε εντολή σύμφωνα με τη σειρά της μέσα στο πρόγραμμα μεταφέρεται στην κεντρική μονάδα επεξεργασίας . Στη συνέχεια αποκωδικοποιείται και αρχίζει να εκτελείται . Κατά την εκτέλεση παράγονται τα κατάλληλα σήματα ελέγχου τα οποία είναι διαφορετικά για κάθε εντολή και οδηγούν τη μηχανή στις αντίστοιχες ενέργειες .

Η διαδικασία αυτή επαναλαμβάνεται μέχρι και της τελευταίας εντολής του προγράμματος .

#### 4.6 Μνήμη Συστήματος

Η μνήμη του συστήματος (**system memory**), είναι ο χώρος στον οποίο αποθηκεύονται τα προγράμματα που "τρέχουν" σε μια δεδομένη χρονική στιγμή, καθώς και τα δεδομένα που χρησιμοποιούνται και παράγονται από αυτά τα προγράμματα. Ο όρος μνήμη (**memory**) είναι γενικός και αναφέρεται σε κάθε τμήμα του υπολογιστή, που μπορεί να αποθηκεύσει προσωρινά ή μόνιμα κάποιες πληροφορίες. Ωστόσο, ο όρος "μνήμη" αναφέρεται κυρίως στην **κύρια μνήμη του συστήματος (main system memory)**, δηλαδή στο χώρο που αποθηκεύονται εντολές προς εκτέλεση από τον επεξεργαστή, καθώς και τα αντίστοιχα δεδομένα, που χρειάζονται για να εκτελεστούν σωστά οι παραπάνω εντολές. Η μνήμη του συστήματος παίζει σημαντικό ρόλο στη σωστή λειτουργία του υπολογιστή.

Η μνήμη συνεισφέρει θετικά στις παρακάτω τέσσερις παραμέτρους:

- **Απόδοση:** Το είδος και η ποσότητα της μνήμης του συστήματος επηρεάζει τη συνολική απόδοση του υπολογιστή. Για παράδειγμα, ανεπάρκεια στη μνήμη του συστήματος, μπορεί να αναγκάσει τον επεξεργαστή να εργάζεται μέχρι και 50% κάτω από την πραγματική του απόδοση.
- **Υποστήριξη Λογισμικού:** Τα νέα προγράμματα απαιτούν πολύ περισσότερη μνήμη σε σχέση με τα παλαιότερα. Έτσι, μεγαλύτερα ποσά μνήμης μας δίνουν τη δυνατότητα να "τρέξουμε" τα σύγχρονα προγράμματα που κυκλοφορούν



- **Αξιοπιστία και Σταθερότητα:** Ανεπαρκής ποσότητα μνήμης καθώς και λανθασμένος τύπος μπορεί να προκαλέσουν την εμφάνιση διαφόρων μυστήριων προβλημάτων στο σύστημα. Αν χρησιμοποιήσουμε μνήμη υψηλής ποιότητας στην κατάλληλη ποσότητα, το σύστημα θα λειτουργεί ομαλά και τα προβλήματα που σχετίζονται με τη μνήμη σχεδόν θα εκλείψουν. Ωστόσο, στην περίπτωση που το σύστημά μας δεν υποστηρίζει κάποιον τύπο μνήμης υψηλής ποιότητας, είναι σίγουρο ότι θα εμφανιστούν σημαντικά προβλήματα. Επομένως θα πρέπει να χρησιμοποιούμε τον κατάλληλο τύπο μνήμης και στην κατάλληλη ποσότητα, προκειμένου να λειτουργεί η μνήμη σύμφωνα με τις προδιαγραφές της.
- **Αναβάθμιση:** Στην αγορά κυκλοφορούν διάφορων τύπων μνήμες, από τις οποίες άλλες είναι πάρα πολύ δημοφιλείς και άλλες λιγότερο. Αν υπάρχει πρόθεση για μελλοντική αναβάθμιση του συστήματος, προτείνεται η επιλογή της μνήμης να είναι τέτοια που θα μπορεί να χρησιμοποιηθεί και στο νέο σύστημα.

#### 4.6.1 Μνήμη Ανάγνωσης Μόνο (ROM)

Η μνήμη **ROM (Read Only Memory- Μνήμη Μόνο Ανάγνωσης)** είναι ένας ειδικός τύπος μνήμης, τα περιεχόμενα της οποίας δε μεταβάλλονται. Έτσι, η μνήμη ROM είναι μνήμη μόνο ανάγνωσης, ενώ η κύρια μνήμη είναι μνήμη ανάγνωσης αλλά και γραφής. Υπάρχουν δυο κυρίως λόγοι εξαιτίας των οποίων χρησιμοποιείται η ROM στους υπολογιστές:

- **Μονιμότητα:** Σε πολλές εφαρμογές, όπως στους υπολογιστές (BIOS), στις ηλεκτρικές συσκευές, στα ηλεκτρονικά παιχνίδια κάποια δεδομένα πρέπει να παραμείνουν αποθηκευμένα ακόμα και μετά τη διακοπή της τροφοδοσίας. Για παράδειγμα, μέσα στο BIOS (Basic Input Output System) κάθε υπολογιστή είναι αποθηκευμένες δυαδικές πληροφορίες, που είναι απαραίτητες για την εκκίνησή του καθώς και για τον έλεγχο της σωστής λειτουργίας του. Οι πληροφορίες αυτές πρέπει να διατηρούνται πάντοτε, ανεξάρτητα από το αν ο υπολογιστής τροφοδοτείται ή όχι με ρεύμα. Η ROM, εξαιτίας του γεγονότος ότι διατηρεί τα περιεχόμενα της ονομάστηκε **μη πτητική μνήμη (non-volatile storage)**. Οι σκληροί δίσκοι είναι επίσης μη πτητική μνήμη, ενώ η κύρια μνήμη δεν είναι, αφού δε διατηρεί τα περιεχόμενά της μετά τη διακοπή της τροφοδοσίας του υπολογιστή με ρεύμα.
- **Ασφάλεια:** Το γεγονός ότι δεν μπορούμε να μεταβάλλουμε τα περιεχόμενα μιας μνήμης ROM, έχει ως αποτέλεσμα να αποτελεί ένα είδος προστασίας έναντι των εσκεμμένων (ή τυχαίων) απόπειρων μεταβολής των περιεχομένων της. Έτσι, υπάρχει προστασία έναντι των ιών (τεχνικά υπάρχει περίπτωση να μολυνθεί από ιό μόνο ένα είδος μνήμης ROM: η μνήμη EPROM).

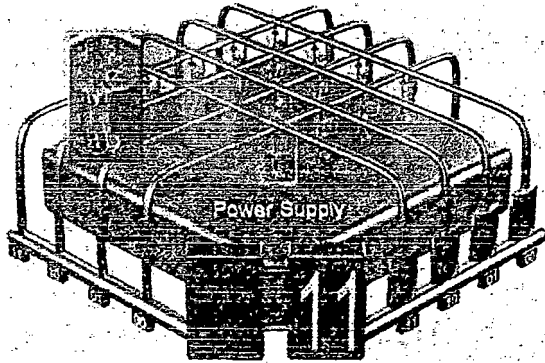
Όπως ειπώθηκε παραπάνω, η μνήμη ROM χρησιμοποιείται για την αποθήκευση προγραμμάτων χαμηλού επιπέδου, που πρέπει να είναι διαθέσιμα κάθε στιγμή. Το πιο χαρακτηριστικό παράδειγμα είναι το BIOS program, που είναι αποθηκευμένο σε ένα ειδικό chip μνήμης ROM, το οποίο είναι γνωστό και ως system BIOS ROM. Αν και το βασικό κριτήριο διαχωρισμού των μνημών ROM από τα υπόλοιπα είδη μνημών, είναι η μονιμότητα των περιεχομένων τους, υπάρχουν κάποιοι τύποι ROMs στους οποίους μπορούμε, ακολουθώντας ειδικές διαδικασίες, να μεταβάλλουμε τα περιεχόμενά τους. Τέλος, θα πρέπει να γίνει κατανοητό, ότι η ταχύτητα μεταφοράς δεδομένων από μια μνήμη ROM είναι σαφώς μικρότερη από αυτή μιας μνήμης RAM, γι' αυτό και τα περιεχόμενα της ROM αντιγράφονται στην κύρια μνήμη πριν χρησιμοποιηθούν για οποιοδήποτε σκοπό.

#### 4.6.2 Τυπική ROM

Τα δεδομένα μιας **τυπικής ROM (Standard ROM)** εισάγονται κατά την κατασκευή της, με την έκθεση ενός φωτοευαίσθητου υλικού μέσω μιας μάσκας που περιέχει το επιθυμητό σχέδιο των bits και με την αφαίρεση στην συνέχεια της εκτεθειμένης (ή της μη εκτεθειμένης) επιφάνειας. Με τον τρόπο αυτό "καίγονται" κάποιοι σύνδεσμοι και αποτυπώνονται οι δυαδικές πληροφορίες. Από τη στιγμή που γίνει η εγγραφή της ROM, οι πληροφορίες δεν μπορούν να αλλάξουν, ενώ αν παρουσιαστεί τέτοια ανάγκη πρέπει να αντικατασταθεί το chip της ROM. Οι ROMs είναι πολύ φθηνότερες από τις RAMs, όταν γίνεται παραγγελία μεγάλων ποσοτήτων για την κάλυψη του κόστους κατασκευής της μάσκας. Δεν είναι όμως καθόλου ευέλικτες, αφού δεν μπορούν να τροποποιηθούν μετά την εγγραφή τους.

#### 4.6.3 Προγραμματιζόμενη μνήμη ανάγνωσης

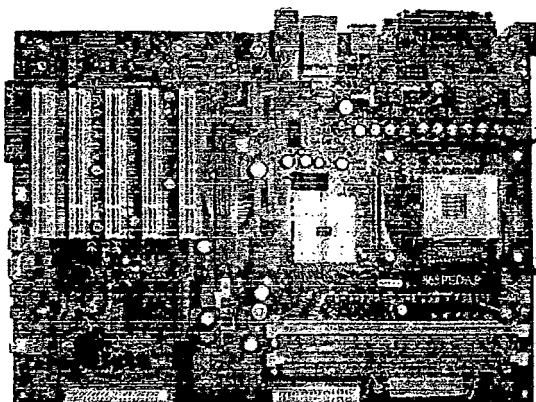
Για να διευκολυνθεί η ανάπτυξη νέων εφαρμογών, αναπτύχθηκαν οι PROMs (Programmable Read Only Memories - Προγραμματιζόμενες Μνήμες Μόνο Ανάγνωσης), που είναι πιο ευέλικτες, γιατί επιτρέπουν τον προγραμματισμό τους όχι μόνο από τον κατασκευαστή, αλλά και από το χρήστη. Ο χρήστης μπορεί να "γράψει" το πρόγραμμα του, χρησιμοποιώντας μια ειδική μηχανή γνωστή ως "προγραμματιστής PROM". Η μνήμη τοποθετείται στη μηχανή αυτή που έχει ήδη διαβάσει το πρόγραμμα και βραχυκυκλώνει (καίει) τους συνδέσμους (fusible links), που στη PROM παριστάνουν "1". Από τη στιγμή αυτή, οι πληροφορίες που αποθηκεύτηκαν παραμένουν αναλλοίωτες. Ο προγραμματισμός της PROM είναι γνωστός και ως burning και κατά κάποιο τρόπο είναι μια διαδικασία παρόμοια με την εγγραφή των CD-R.



*Το κάψιμο των συνδέσμων σε μια μνήμη PROM*

#### 4.6.4 Ελεγκτής μνήμης

Σε κάθε υπολογιστή υπάρχει ένα ολοκληρωμένο κύκλωμα που ονομάζεται **ελεγκτής μνήμης (memory controller)** και ελέγχει τις λειτουργίες που σχετίζονται με τη μνήμη. Παράγει τα απαραίτητα σήματα για να γίνονται σωστά οι λειτουργίες της ανάγνωσης / γραφής στη μνήμη και συνδέει τη μνήμη με τα υπόλοιπα μέρη του υπολογιστή. Συνήθως, ο ελεγκτής μνήμης είναι ενσωματωμένος στο chipset συστήματος.



#### 4.6.5 Δίαυλος μνήμης

Ο δίαυλος της μνήμης (memory bus), είναι ένα σύνολο καλωδίων που χρησιμοποιούνται για τη μεταφορά της διεύθυνσης προς τη μνήμη, καθώς και για τη μεταφορά των δεδομένων από και προς τη κύρια μνήμη. Σε πολλούς υπολογιστές, ο δίαυλος της μνήμης είναι κοινός με το δίαυλο του επεξεργαστή.

Ο δίαυλος της μνήμης αποτελείται από δυο τμήματα: το δίαυλο των δεδομένων (data bus) και το δίαυλο των διευθύνσεων (address bus). Συνήθως, όταν αναφερόμαστε στο δίαυλο της μνήμης, εννοούμε το δίαυλο των δεδομένων, ο οποίος χρησιμοποιείται για την πραγματική μεταφορά των δεδομένων από και προς τη μνήμη. Ο δίαυλος των διευθύνσεων, χρησιμοποιείται για τη μεταφορά της διεύθυνσης στη μνήμη, όπου θα πραγματοποιηθεί μια προσπέλαση για ανάγνωση ή γραφή. Όσο μεγαλύτερος είναι ο δίαυλος των δεδομένων (περισσότερα καλώδια), τόσα περισσότερα δεδομένα μπορούν να μεταφερθούν ταυτόχρονα, με αποτέλεσμα να βελτιώνεται η απόδοση όλου του συστήματος. Το εύρος ζώνης (bandwidth) του διαύλου των δεδομένων, μετρά την ποσότητα της πληροφορίας που μπορεί να μεταφερθεί μέσω των καλωδίων του και είναι συνάρτηση του αριθμού των bits που μπορούν να μεταφερθούν ταυτόχρονα και της ταχύτητας μεταφοράς των δεδομένων.



*Ο δίαυλος της μνήμης*

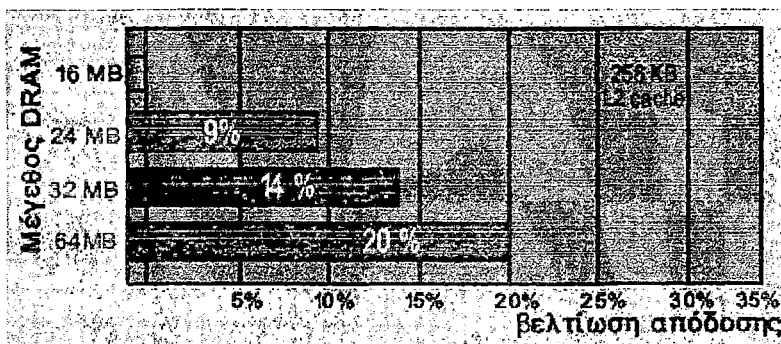
Θα μπορούσαμε να φανταστούμε τον δίαυλο των δεδομένων σαν ένα δρόμο ταχείας κυκλοφορίας. Το εύρος του είναι το πλήθος των διαφορετικών λωρίδων που έχει, ενώ η ταχύτητά του, καθορίζει το πόσο γρήγορα μπορούν να κινηθούν τα αυτοκίνητα. Το εύρος ζώνης του αυτοκινητόδρομου, είναι ο μέγιστος αριθμός των αυτοκινήτων που θα περάσουν σε ένα ορισμένο χρονικό διάστημα, υπό το καθεστώς των περιορισμών στον αριθμό των διαθέσιμων λωρίδων και στην επιτρεπόμενη ταχύτητα.

Το εύρος του διαύλου των διευθύνσεων καθορίζει το μέγιστο αριθμό διαφορετικών διευθύνσεων που μπορεί να αναγνωρίσει ο επεξεργαστής.

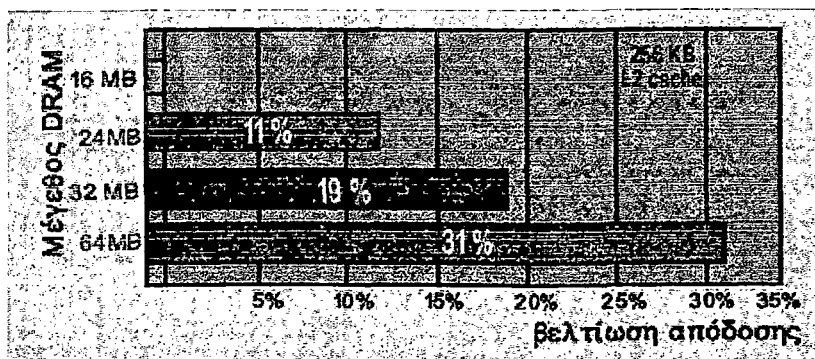
## 4.7 Μέγεθος Μνήμης

### 4.7.1 Μέγεθος μνήμης και απόδοση συστήματος

Το μέγεθος της κύριας μνήμης, που διαθέτει ένας υπολογιστής, παίζει σημαντικό ρόλο στη διαμόρφωση της συνολικής απόδοσης του συστήματος. Επομένως, ανάλογα με το σύστημα που διαθέτουμε και το σκοπό για τον οποίο θα το χρησιμοποιούμε (είδος προγραμμάτων που χρησιμοποιούμε), πρέπει να έχουμε εγκαταστήσει τον κατάλληλο τύπο μνήμης στην κατάλληλη ποσότητα.



*Η βελτίωση της απόδοσης σε ένα σύστημα Pentium 100MHz με λειτουργικό σύστημα Windows 95 για διάφορα μεγέθη DRAM*



*Η βελτίωση της απόδοσης σε ένα σύστημα Pentium Pro 200MHz με λειτουργικό σύστημα Windows 95 για διάφορα μεγέθη DRAM*

Το μέγεθος της μνήμης δεν επηρεάζει την ταχύτητά της, ούτε επίσης την ταχύτητα του επεξεργαστή, των chipsets, της μητρικής πλακέτας καθώς και άλλων εξαρτημάτων του υπολογιστή. Αυτό ισχύει μόνο στην περίπτωση που όλα τα προγράμματα που "τρέχουν" ταυτοχρόνως χωράνε στην κύρια μνήμη του συστήματος. Στην περίπτωση συστημάτων πολυεργασίας, χρησιμοποιούμε εικονική μνήμη. Αν το μέγεθος της κύριας μνήμης είναι πολύ μικρό σε σχέση με τις πραγματικές απαιτήσεις, θα έχουμε πολλές μεταφορές σελίδων μεταξύ του σκληρού δίσκου και της κύριας μνήμης, οπότε η επιπλέον καθυστέρηση θα ρίχνει σημαντικά την απόδοση του συστήματος. Ο καλύτερος τρόπος για να εκτιμήσουμε τη σχέση μεταξύ του μεγέθους της κύριας μνήμης και της συνολικής απόδοσης του συστήματος, είναι να συγκρίνουμε ίδια συστήματα που εκτελούν τις ίδιες εργασίες και έχουν διαφορετικές ποσότητες μνήμης (του ίδιου τύπου).

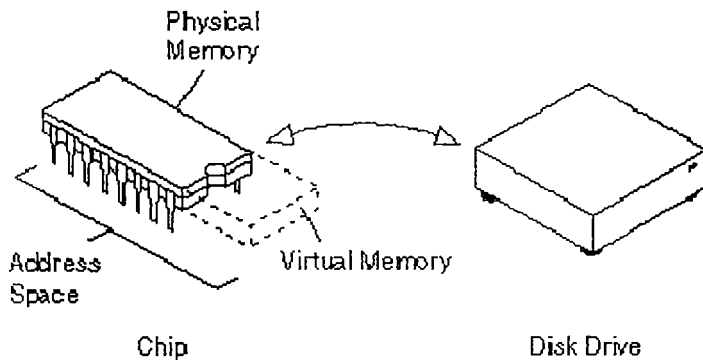
#### 4.7.2 Πραγματική και Εικονική μνήμη

Η **Real memory**, είναι η **πραγματική** μνήμη που έχουμε εγκαταστήσει στο σύστημά μας. Ωστόσο, μερικές φορές είναι καλό να "ξεγελούμε" τον υπολογιστή και να τον κάνουμε να "πιστεύει" ότι έχει στη διάθεσή του περισσότερη μνήμη, από αυτή που πραγματικά διαθέτει. Ιδιαίτερα σε περιπτώσεις όπου τρέχουμε μεγάλα προγράμματα ή σε περιπτώσεις πολυεργασίας (multitasking), η μέθοδος αυτή είναι χρήσιμη και έχει ονομαστεί **εικονική μνήμη (virtual memory)**.

Τρόπος λειτουργίας της εικονικής μνήμης:

Η εικονική μνήμη λειτουργεί κατά τον εξής απλό τρόπο: Ας υποθέσουμε ότι το λειτουργικό σύστημα χρειάζεται 80 MB για να κρατήσει όλα τα προγράμματα που τρέχουν, αλλά υπάρχουν μόνο 32 MB πραγματικής κύριας μνήμης. Το λειτουργικό σύστημα υποθέτει ότι είναι διαθέσιμα 80 MB εικονικής μνήμης, οπότε "καλεί" ένα πρόγραμμα που διαχειρίζεται την εικονική μνήμη και είναι γνωστό ως **διαχειριστής εικονικής μνήμης (virtual memory manager)**. Στη συνέχεια, δημιουργείται ένα **αρχείο ανταλλαγής (swap file)** στο σκληρό δίσκο, με μέγεθος  $80-32=48$  MB, που ανταλλάσει τα περιεχόμενά του με αυτά της πραγματικής κύριας μνήμης. Το λειτουργικό σύστημα πλέον "βλέπει" 80 MB μνήμης, ενώ ο διαχειριστής εικονικής μνήμης είναι επιφορτισμένος με τη διαχείριση της πραγματικής μνήμης των 32 MB.

Φυσικά, από τα 80 MB που απαιτούνται, μόνο τα 32 MB είναι αποθηκεύμενα στην κύρια μνήμη, ενώ τα υπόλοιπα 48 MB βρίσκονται στο αρχείο ανταλλαγής του δίσκου. Όταν το λειτουργικό σύστημα χρειάζεται ένα τμήμα της μνήμης που όμως βρίσκεται στο αρχείο του δίσκου, ο διαχειριστής εικονικής μνήμης διαβάζει αυτό το τμήμα από τα αρχείο και το αντιγράφει στην κύρια μνήμη, στη θέση ενός block δεδομένων που δεν έχουν χρησιμοποιηθεί πρόσφατα. Τα blocks των δεδομένων που ανταλλάσσονται μεταξύ σκληρού δίσκου και κύριας μνήμης, συνήθως ονομάζονται **σελίδες (pages)**.

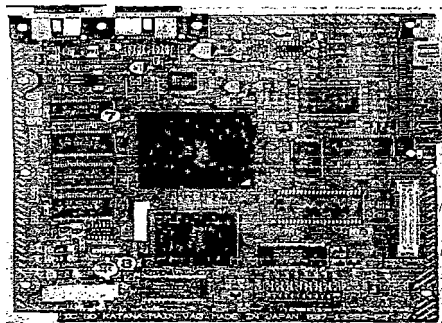


### Λειτουργία της εικονικής μνήμης

Η εικονική μνήμη είναι μια σημαντική επινόηση στο χώρο των υπολογιστών, με δεδομένους τους περιορισμούς στη ποσότητα της κύριας μνήμης που μπορούμε να εγκαταστήσουμε στο σύστημά μας. Χωρίς τη χρησιμοποίηση εικονικής μνήμης δεν θα μπορούσαμε να τρέχουμε πολλά προγράμματα ταυτοχρόνως (multitasking), αφού δε διαθέτουμε τις απαιτούμενες ποσότητες κύριας μνήμης για να ικανοποιήσουν ταυτοχρόνως όλα τα προγράμματα. Τα Windows 95, είναι μια χαρακτηριστική περίπτωση λειτουργικού συστήματος, το οποίο χρησιμοποιεί εικονική μνήμη, αφού παρέχει τη δυνατότητα στους χρήστες για πολυεργασία. Θα πρέπει να σημειωθεί ότι η ιδέα για την χρησιμοποίηση εικονικής μνήμης τοποθετείται χρονολογικά πολύ πριν από την εμφάνιση των Windows, καθώς είχε σχεδιαστεί για λειτουργικά συστήματα κεντρικών υπολογιστών (UNIX, VMS, κλπ).

Η χρησιμοποίηση εικονικής μνήμης έχει όμως και κάποια μειονεκτήματα. Αν το μέγεθος του αρχείου ανταλλαγής στο δίσκο είναι κατά πολύ μεγαλύτερο από το μέγεθος της κύριας μνήμης, τότε θα συμβαίνουν πολλές ανταλλαγές μεταξύ σκληρού δίσκου και κύριας μνήμης. Όμως ο σκληρός δίσκος είναι πολύ πιο αργός από την RAM, με αποτέλεσμα να υπάρχει καθυστέρηση κατά την ανταλλαγή, οπότε μειώνεται η απόδοση της μνήμης και γενικότερα του συστήματος. Οι συνεχείς ανταλλαγές δεδομένων μεταξύ σκληρού δίσκου και κύριας μνήμης είναι γνωστές ως thrashing. Το πρόβλημα thrashing επιδεινώνεται σε συστήματα που είναι ιδιαίτερα επιβαρυνμένα και δεν έχουν σχετικά μεγάλη ποσότητα κύριας μνήμης. Για παράδειγμα, αν χρησιμοποιήσουμε το λειτουργικό σύστημα Windows NT, σε έναν υπολογιστή με 16 MB RAM, τότε είναι σίγουρο ότι θα έχουμε συνεχείς ανταλλαγές σελίδων μεταξύ σκληρού δίσκου και κύριας μνήμης. Αυτό συμβαίνει γιατί τα Windows NT θέλουν 12-16 MB από μόνα τους, με αποτέλεσμα για κάθε επιπλέον εφαρμογή να έχουμε thrashing.

#### 4.8 Φυσική και λογική οργάνωση μνήμης



Όπως οι επεξεργαστές, έτσι και οι μνήμες είναι κατασκευασμένες από μικροσκοπικούς ημιαγωγούς. Όλοι αυτοί οι ημιαγωγοί καθώς και τα λοιπά κυκλώματα πρέπει να ενοποιηθούν σε ένα chip, έτσι ώστε να συνδεθούν με τα υπόλοιπα μέρη του υπολογιστή. Πολλές φορές συνδέουμε μεταξύ τους πολλά chips μνήμης και δημιουργούμε μεγαλύτερες μονάδες οργάνωσης.

### Μέρος Γ: Προγραμματισμός Η/Υ και Λειτουργικά συστήματα- (Windows XP)

#### Κεφάλαιο 5 : Προγραμματισμός Η/Υ, γλώσσες και είδη προγραμματισμού

##### 5.1. Προγραμματισμός

##### 5.1.1 Ορισμός Προγραμματισμού

Προγραμματισμός είναι η διαδικασία ανάπτυξης προγράμματος, δηλαδή συνόλου εντολών κατανοητών από τον υπολογιστή, που υλοποιούν ένα αλγόριθμο. Είναι η μέθοδος με την οποία κάθε χρήστης υπολογιστών, καθοδηγεί έναν υπολογιστή στην επίλυση προβλημάτων.

Από την αρχή της εμφάνισης των υπολογιστών γίνονται προσπάθειες μεθοδολογιών και τεχνικών προγραμματισμού, που θα εξασφάλιζαν τη δημιουργία απλών και κομψών προγραμμάτων.

Βασικά στοιχεία του προγράμματος είναι τα δεδομένα του καθώς και οι δομές δεδομένων πάνω στα οποία και ενεργεί.

Στο στάδιο που βρίσκεται η Επιστήμη των Υπολογιστών σήμερα, ο προγραμματισμός δεν είναι ακόμη μια συγκεκριμένη αυστηρή μεθοδολογία, που βασίζεται σε παγκόσμιες αρχές και αξιώματα. Παρότι έχουν αναπτυχθεί δεκάδες διαφορετικές «τεχνικές» και «μέθοδοι» προγραμματισμού, ο προγραμματισμός αποτελεί ακόμη μία «τέχνη», στην οποία σημαντικό ρόλο παίζει η οξυδέρκεια, αλλά



και η πείρα. Συνέπεια αυτού είναι η εξέλιξη της τεχνολογίας ανάπτυξης λογισμικού να είναι πολύ βραδύτερη από την αντίστοιχη εξέλιξη του υλικού των υπολογιστών.

Ο προγραμματισμός είναι μία διαδικασία από επτά βήματα:

– 1<sup>ο</sup>: Διατύπωση και Ανάλυση του προβλήματος.

Για να λύσουμε ένα πρόβλημα, όσο σύνθετο και αν είναι πρέπει πρώτα να λάβουμε υπόψη μας τα δεδομένα και τις τον τρόπο που θα τα συνδέσουμε μεταξύ τους, ώστε να γίνουν οι διαδικασίες χωρίς εμπλοκές.

– 2<sup>ο</sup>: Δημιουργία του Αλγορίθμου.

Μία σειρά διαδικασιών που παρουσιάζονται με μορφή βημάτων κατά την επίλυση ενός προβλήματος, δημιουργούν τον αλγόριθμο. Τα περισσότερα προβλήματα λύνονται με τη χρήση απλών ή και περισσότερο πολύπλοκων αλγορίθμων.

– 3<sup>ο</sup>: Σχηματική παρουσίαση του Αλγορίθμου.

Είναι ένας τρόπος με τον οποίο επιτυγχάνεται η λογική σειρά που πρέπει να ακολουθηθεί ώστε να φτάσουμε στην επίλυση. Αυτό βέβαια, με τη βοήθεια διαφόρων κωδικοποιημένων σχημάτων.

– 4<sup>ο</sup>: Μεταφορά της διαδικασίας του Αλγορίθμου στη γλώσσα προγραμματισμού.

Μετά την οριστική διατύπωση του αλγορίθμου, ακολουθεί το στάδιο της μεταφοράς του σε μία συγκεκριμένη γλώσσα προγραμματισμού. Για να γίνει σωστά αυτό, πρέπει να υπάρχει καλή γνώση της γλώσσας που πρόκειται να χρησιμοποιήσουμε. Το παραμικρό λάθος στο στάδιο αυτό, αποκλείει την εκτέλεση προγράμματος από τον υπολογιστή.

– 5<sup>ο</sup>: Τρέξιμο του προγράμματος στον Υπολογιστή.

Το πρόγραμμα όταν είναι γραμμένο στη γλώσσα προγραμματισμού που επιλέξαμε ονομάζεται «πηγαίο ή αρχικό πρόγραμμα». Δε μπορεί να γίνει αμέσως κατανοητό από τον υπολογιστή. Αυτός καταλαβαίνει μόνο τη «γλώσσα μηχανής», στην οποία μετατρέπεται μέσω ενός ειδικού προγράμματος, που ονομάζεται μεταφραστής (compiler). Έτσι, το πηγαίο πρόγραμμα μετατρέπεται σε ένα «αντικειμενικό ή τελικό πρόγραμμα», εφόσον το πηγαίο πρόγραμμα δεν έχει συντακτικά λάθη.

– 6<sup>ο</sup>: Έλεγχος των Αποτελεσμάτων.

Τα λογικά λάθη ενός προγράμματος δε μπορεί να τα εντοπίσει ο υπολογιστής. Έτσι, δεδομένου ότι ο υπολογιστής θα εξάγει κάποια αποτελέσματα λανθασμένα, ο προγραμματιστής υποχρεούται να ελέγξει πάλι το πηγαίο πρόγραμμα. Το σημαντικότερο σημείο για σωστό προγραμματισμό με τη χρήση υπολογιστή είναι η σωστή δημιουργία του λογικού διαγράμματος και κατ' επέκταση του σχετικού αλγορίθμου.

– 7<sup>ο</sup>: Διόρθωση σφαλμάτων.

### 5.1.2 Γλώσσες Προγραμματισμού

Οι άνθρωποι χρησιμοποιούν διάφορες γλώσσες για την επικοινωνία μεταξύ τους. Έτσι και για την «επικοινωνία» μεταξύ ανθρώπων και υπολογιστών, χρησιμοποιούμε κάποια γλώσσα, όπου αποδέκτης είναι μία υπολογιστική μηχανή κατάλληλα εξοπλισμένη ώστε να καταλαβαίνει αυτή τη γλώσσα ή κάποια άλλη.

Ένα πρόγραμμα είναι η πραγματοποίηση των βημάτων του αλγορίθμου σε κάποια από τις γλώσσες προγραμματισμού. Για να εκτελεστεί κάποια εργασία, ένας υπολογιστής πρέπει να πάρει συγκεκριμένες εντολές- οδηγίες. Οι εντολές αυτές θα

πρέπει να είναι μέσα στο σύνολο των εντολών που διαθέτει, διαφορετικά δεν εκτελούνται, αφού δεν αναγνωρίζονται. Μια σειρά εντολών είναι και το **Πρόγραμμα**.

Ο υπολογιστής δε μπορεί να αναγνωρίσει αν μία σειρά εντολών είναι στη λογική τους θέση ή όχι. Αυτό που κάνει είναι να εκτελεί τις εντολές πιστά στη σειρά που δίνονται. Πρέπει πάντα να θυμόμαστε ότι ο υπολογιστής δε μπορεί να «σκεφτεί», αφού δε διαθέτει λογική. Η δουλειά του είναι να εκτελεί οδηγίες που οδηγούν σε λάθος ή σωστό αποτέλεσμα, ανάλογα με το αν χρησιμοποιήσαμε σωστά τη λογική ακολουθία και τη σωστή γραφή για την επίλυση ενός προγράμματος σε κάποια γλώσσα προγραμματισμού.

Στις γλώσσες προγραμματισμού διακρίνουμε τις εξής κατηγορίες:

α) **Χαμηλού επιπέδου γλώσσες** (Low Level Languages ή ASSEMBLY), πολύ συγγενική με τη γλώσσα μηχανής. Συνδιασμοί δυαδικών αριθμών έχουν αντικατασταθεί με «μνημονικούς» συμβολισμούς.

β) **Υψηλού επιπέδου γλώσσες** (High Level Languages).

Οι γλώσσες αυτές έχουν αναπτυχθεί για τον προγραμματισμό υπολογιστών. Είναι πιο εύκολες στην εκμάθηση από τη γλώσσα μηχανής και την ASSEMBLY και είναι ανεξάρτητες από τα διάφορα συστήματα υπολογιστών. Οι γλώσσες αυτές χρησιμοποιήθηκαν για διάφορες ανάγκες σε προβλήματα εμπειρικών τεχνικών, επιστημονικών εφαρμογών. Η εξέλιξη των γλωσσών προγραμματισμού οδήγησε στην χρήση τους σε όλα τα είδη προβλημάτων.

Ο προγραμματισμός με τη χρήση συμβολικών γλωσσών αποτέλεσε μεγάλο βήμα σε σχέση με τον προγραμματισμό σε κώδικα μηχανής.

Όμως, προβλήματα υπάρχουν, όπως:

α) Οι εντολές των συμβολικών γλωσσών εκτελούν στην πλειοψηφία τους εντελώς στοιχειώδεις λειτουργίες.

β) Στον προγραμματισμό σε συμβολική γλώσσα ο προγραμματιστής πρέπει να λαμβάνει συνεχώς υπόψη του τις ιδιαιτερότητες του υπολογιστή. Πρέπει να απασχολείται με έννοιες όπως η διάταξη της μνήμης, οι διαθέσιμοι τρόποι προσπέλασης δεδομένων κτλ.

γ) Κάθε υπολογιστής έχει τη δική του συμβολική γλώσσα και κώδικα μηχανής. Έτσι, προγράμματα που είναι γραμμένα σε συμβολική γλώσσα για έναν υπολογιστή, είναι σχεδόν αδύνατον να εκτελεσθούν σε οποιονδήποτε άλλον.

Ο μόνος τρόπος προγραμματισμού των ηλεκτρονικών κυκλωμάτων ενός υπολογιστή είναι η χρήση του κώδικα μηχανής του. Για να χρησιμοποιήσουμε συμβολική γλώσσα έχουμε ανάγκη της ενδιάμεσης χρήσης ενός συμβολομεταφραστή, που να μεταφράζει ένα συμβολικό πρόγραμμα στην αντίστοιχη ακολουθία εντολών κώδικα μηχανής.

Για τους ίδιους λόγους η χρησιμοποίηση μιας γλώσσας υψηλού επιπέδου γίνεται με τη βοήθεια προγραμμάτων μεταφραστών και μόνο. Τα προγράμματα αυτά μεταφράζουν μία τέτοια γλώσσα σε κώδικα μηχανής.

Υπάρχουν δύο κατηγορίες προγραμμάτων μεταφραστών:

1) Οι **μεταγλωττιστές (compilers)**. Δέχονται ένα ολόκληρο αρχικό πρόγραμμα (source program), γραμμένο σε μία γλώσσα υψηλού επιπέδου και παράγουν ένα ισοδύναμο τελικό πρόγραμμα (object program) σε κώδικα μηχανής. Το τελικό πρόγραμμα μπορεί στη συνέχεια να εκτελεστεί όσες φορές επιθυμούμε και είναι τελείως ανεξάρτητο από το αρχικό κείμενο του προγράμματος στη γλώσσα υψηλού επιπέδου. Ο μεταγλωττιστής καλείται μόνο μία φορά για κάθε πρόγραμμα. Δεν είναι

απαραίτητο να βρίσκεται στη μνήμη κατά τη διάρκεια των εκτελέσεων του τελικού προγράμματος.

- 2) Οι **διερμηνευτές (interpreters)**. Καλούνται κάθε φορά που επιθυμούμε να εκτελέσουμε ένα πρόγραμμα γραμμένο σε γλώσσα υψηλού επιπέδου. Ένας διερμηνευτής δε μεταφράζει το πρόγραμμά μας παράγοντας κάποιο ισοδύναμο τελικό πρόγραμμα κώδικα μηχανής. Διαβάζει μία-μία τις εντολές του και εκείνη τη στιγμή εκτελεί κάθε μία από αυτές δίνοντας μία ισοδύναμη ακολουθία εντολών κώδικα μηχανής. Συνεπώς, μας δίνεται η ψευδαίσθηση ότι ο υπολογιστής καταλαβαίνει απευθείας τη γλώσσα μηχανής που χρησιμοποιούμε.

Θεωρητικά, η υλοποίηση κάθε γλώσσας προγραμματισμού, μπορεί να γίνει είτε με μεταγλωττιστή, είτε με διερμηνευτή. Στην πράξη ωστόσο κάποιες γλώσσες είναι σχεδιασμένες για να δέχονται ευκολότερη μεταγλώττιση, ενώ άλλες γλώσσες ενώ άλλες για ευκολότερη διερμηνεία.

Στην πρώτη κατηγορία ανήκουν οι: FORTRAN, COBOL, PASCAL, C κτλ.

Στη δεύτερη οι: BASIC, LISP, APL, PROLOG κτλ.

Η χρήση ενός μεταγλωττιστή έχει το μειονέκτημα ότι πρώτου μπορεί να εκτελέσει ένα πρόγραμμα, πρέπει να περάσει από την ενδιάμεση φάση της εταγλώττισης. Ανάλογα όμως, με το μέγεθος του προγράμματος, μπορεί να διαρκέσει από δευτερόλεπτα μέχρι και ώρες. Επίσης, αν κατ'ατην εκτέλεση του προγράμματος προκύψουν σφάλματα, τότε το διορθωμένο αρχικό πρόγραμμα πρέπει να ξαναμεταγλωττιστεί από την αρχή πρώτου ξαναχρησιμοποιηθεί.

Η χρήση διερμηνευτή έχει το αποτέλεσμα της άμεσης εκτέλεσης και άμεσης διόρθωσης σφαλμάτων. Προτιμάται από γλώσσες που αλληλεπιδρούν στενά με το χρήστη (interactive languages). Όμως, η διαδικασία της διερμηνείας καθιστά την εκτέλεση του προγράμματος πιο αργή από αυτή του μεταγλωττιστή, αφού σε κάθε εκτέλεση παρεμβάλλεται σα μεσάζων το πρόγραμμα του διερμηνευτή.

Συμπερασματικά, η υλοποίηση μιας γλώσσας υψηλού επιπέδου με μεταγλωττιστή, υπερτερεί στην ταχύτητα εκτέλεσης. Η υλοποίησή της ωστόσο με διερμηνευτή υπερτερεί στην ευκολία χρήσης. Σε περιβάλλοντα μικροϋπολογιστών, έχουν εμφανιστεί μικτές υλοποιήσεις γλωσσών.

### 5.1.3 Ιεραρχικός Προγραμματισμός

Η σχεδίαση προγράμματος πρέπει να προχωρεί από πάνω προς τα κάτω. Ξεκινάμε με μια πολύ απλή και σύντομη δήλωση για το τι κάνει το πρόγραμμα, π.χ. με τον υπολογισμό του μέσου όρου βαθμολογίας. Αυτό γίνεται στο αρχικό επίπεδο σχεδίασης του προγράμματος, ενώ στο επόμενο επίπεδο εμφανίζονται με περισσότερες λεπτομέρειες οι επεξεργασίες που εκτελεί το πρόγραμμα. Καθεμιά απ' αυτές δεν είναι τελείως συμπληρωμένη, αλλά περιγράφεται με περισσότερες λεπτομέρειες σε κατώτερο επίπεδο του προγράμματος.

Η διαδικασία αυτή επαναλαμβάνεται βήμα προς βήμα και, σε κάθε κατώτερο επίπεδο, εμφανίζονται περισσότερες λεπτομέρειες. Η σε κατώτερα επίπεδα ανάλυση σταματάει, όταν η επεξεργασία περιέχει τόσες λεπτομέρειες ώστε να μπορεί να κωδικοποιηθεί σε μια γλώσσα προγραμματισμού.

Τα παραπάνω συνιστούν την τεχνική του ιεραρχικού προγραμματισμού όπου, ξεκινώντας από τα γενικά σε ανώτερα επίπεδα , προχωρούμε προσθέτοντας λεπτομέρειες σε κατώτερα επίπεδα

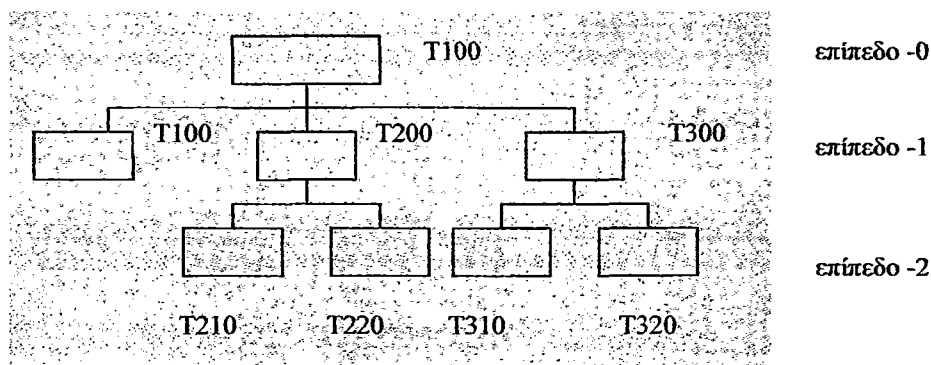
### 5.1.4 Τμηματικός Προγραμματισμός

Οι διάφορες επεξεργασίες που εμφανίζονται κατά την προηγούμενη ανάπτυξη είναι τμήματα προγράμματος λογικά, πλήρη και ανεξάρτητα μεταξύ τους. Τα τμήματα αυτά έχουν μια είσοδο και μια έξοδο και αποτελούν λογικά αυτοτελείς προγραμματικές ενότητες, οι οποίες περιέχουν τις βασικές δομές του Δομημένου Προγραμματισμού. Επειδή είναι μικρότερα προγράμματα, είναι δυνατό να κωδικοποιηθούν, να δοκιμαστούν και να διορθωθούν ανεξάρτητα και ο συνδυασμός τους να δώσει το πλήρες πρόγραμμα.

Ο χωρισμός του προγράμματος σε ανεξάρτητες - λογικές ενότητες - τμήματα αποτελεί τον **Τμηματικό Προγραμματισμό**. Τα μεμονωμένα τμήματα πρέπει να είναι όσο το δυνατόν μικρότερα, ώστε να διευκολύνεται ο καταμερισμός της λογικής ενότητας του προγράμματος σε μικρότερες, που είναι ευκολότερο να διορθωθούν.

Η τεχνική που χρησιμοποιεί τις αρχές του **Ιεραρχικού** και του **Τμηματικού** προγραμματισμού για την σχεδίαση ενός αλγορίθμου και του αντίστοιχου προγράμματος καλείται **Δομημένος Προγραμματισμός**. Με το συνδυασμό των δυο τεχνικών , κάθε επίπεδο προγράμματος υποστηρίζεται από το αμέσως κατώτερο επίπεδο. Το πρόγραμμα στο κατώτερο επίπεδο επεξεργάζεται τα δεδομένα και συγχρόνως δημιουργεί την πληροφορία η οποία απαιτείται από το υψηλότερο επίπεδο.

Η παράσταση αυτών των τεχνικών γίνεται με τα διαγράμματα **HIPO** (Hierarchical Input Output Processing), που στα Ελληνικά μεταφράζεται ως **Ιεράρχηση Εισόδου – Επεξεργασίας-Εξόδου**.)

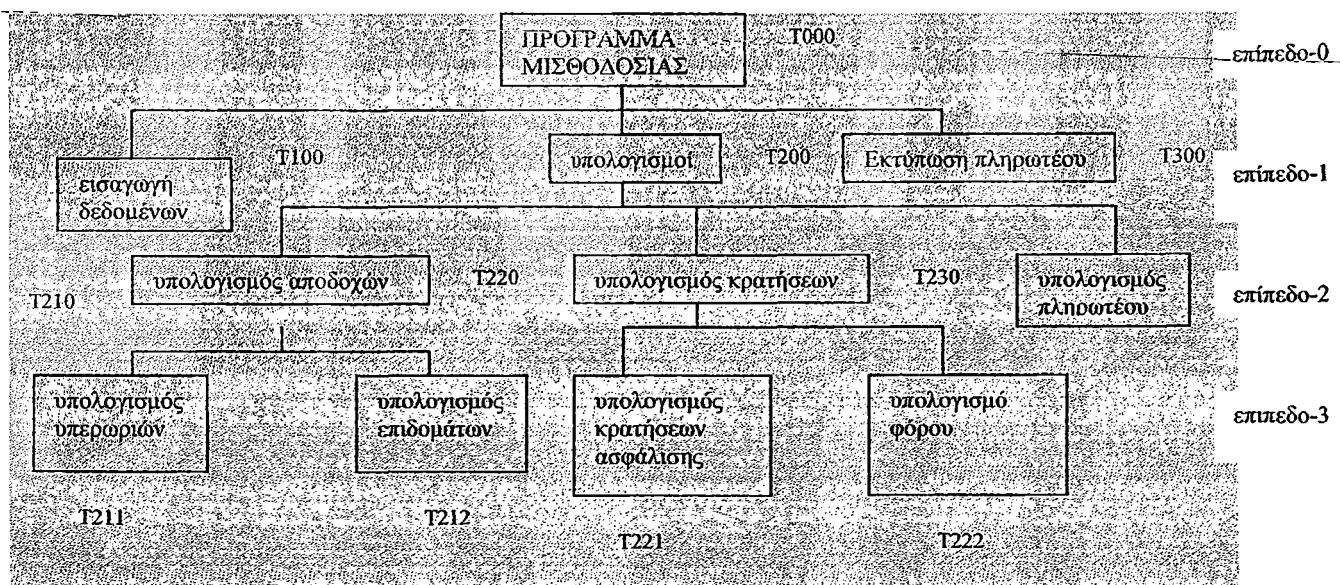


Ένα διάγραμμα HIPO αποτελείται από τρία βασικά τμήματα, τα οποία περιγράφουν την είσοδο δεδομένων , την επεξεργασία και τα αποτελέσματα που απαιτεί ένας αλγόριθμος και το αντίστοιχο πρόγραμμα. Στα διαγράμματα αυτά εμφανίζονται τα διαφορά επίπεδα ιεραρχίας και τα τμήματα του αλγορίθμου – προγράμματος με αριθμό σύμφωνα με τη σειρά εκτέλεσης τους. Στο ανώτερο επίπεδο υπάρχει μια περιγραφή της όλης διαδικασίας που αναπαρίσταται από το διάγραμμα

HIPO. Στο αμέσως επόμενο επίπεδο διακρίνονται , σε τμήματα, οι τρεις βασικές λειτουργίες (είσοδος, επεξεργασία και έξοδος) οι οποίες μπορεί να αναλύονται περαιτέρω σε αλλά υποτιμήματα.

Το παρακάτω διάγραμμα HIPO, αφορά τον υπολογισμό μισθοδοσίας υπαλλήλων.

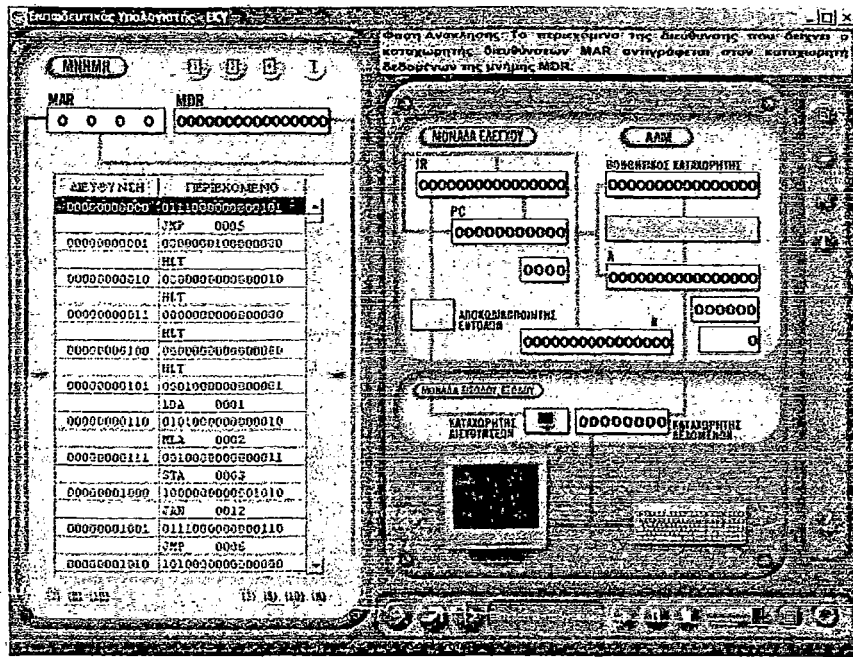
### ΥΠΟΛΟΓΙΣΜΟΣ ΜΙΣΘΟΔΟΣΙΑΣ



### 5.2 Γλώσσες Μηχανής

Γλώσσα μηχανής είναι η γλώσσα προγραμματισμού που αντιλαμβάνεται ο υπολογιστής. Οι εντολές της είναι ακολουθίες δυαδικών ψηφίων 0, 1. Ο τρόπος προγραμματισμού σε γλώσσα μηχανής είναι επίπονος, ακατανόητος για πολλούς και χρονοβόρος.

Αλλά ακόμα και σήμερα οι εντολές των σύγχρονων προγραμμάτων μετατρέπονται από τον σύγχρονο υπολογιστή σε γλώσσα μηχανής, προκειμένου να εκτελεστούν.



Αναπαράσταση λειτουργίας γλώσσας μηχανής

Τα προγράμματα που είναι γραμμένα σε γλώσσα μηχανής ενός υπολογιστή είναι εξαρτημένα από τη δομή του συγκεκριμένου υπολογιστή και μπορούν να τρέξουν μόνο σε υπολογιστές, που είναι όμοιοι ή έχουν συμβατές ΚΜΕ. Ένα πρόγραμμα γραμμένο σε γλώσσα μηχανής είναι εύκολα κατανοητό από τον υπολογιστή για τον οποίο γράφτηκε αλλά δύσκολα κατανοητό από τον άνθρωπο, ακόμη και από αυτόν που το έγραψε, μετά από κάποιο χρονικό διάστημα.

Ο προγραμματισμός σε γλώσσα μηχανής ήταν ο πρώτος τρόπος προγραμματισμού των υπολογιστών. Ο προγραμματισμός στη γλώσσα αυτή θεωρείται σήμερα ένας άθλος για τον προγραμματιστή της εποχής εκείνης. Οι πρώτοι υπολογιστές προγραμματιζόταν σε γλώσσα μηχανής με τη χρήση διακοπών, των οποίων η μία θέση αντιπροσώπευε το 0 (μηδέν) και η άλλη το 1 (ένα).

Εάν προσπαθήσετε να προγραμματίσετε με γλώσσα μηχανής, θα συναντήσετε πολλές δυσκολίες.

- Η γλώσσα και ο κώδικας είναι δύσκολο να διαβαστούν και να μεταφραστούν.
- Οι εντολές αναφέρονται σε συγκεκριμένες θέσεις μνήμης, που μπορεί να αλλάζουν εάν αλλάξει το πρόγραμμα. Ο προγραμματιστής πρέπει να κανονίσει τις θέσεις μνήμης για την αποθήκευση των δεδομένων.
- Ο κώδικας ο οποίος εξαρτάται από την μηχανή θα πρέπει να ξαναγραφεί εάν πρόκειται να εκτελεσθεί σε κάποιον διαφορετικό υπολογιστή. Τα περισσότερα προγράμματα περιέχουν πολλές γραμμές από εντολές σε κώδικα μηχανής.

Επειδή η γλώσσα μηχανής είναι δυσανάγνωστη για τον άνθρωπο αφού οι εντολές της είναι ακολουθίες από δυαδικά ψηφία, είναι χρήσιμος ο ορισμός μιας συμβολικής μορφής των εντολών της ως μνημονικών κωδίκων, αν επιθυμεί κάποιος να γράψει προγράμματα σε αυτή τη γλώσσα. Έτσι ορίστηκαν ευανάγνωστες στον άνθρωπο εκδόσεις γλωσσών μηχανής που ονομάστηκαν συμβολικές γλώσσες (assembly languages). Έτσι μία ακολουθία από bits που στη γλώσσα μηχανής εκφράζει την

εντολή που προσθέτει το περιεχόμενο δύο κυκλωμάτων της κεντρικής μονάδας επεξεργασίας, μπορεί σε συμβολική γλώσσα να συμβολίζεται από το A (ADD).

### 5.2.1 Συμβολικές Γλώσσες

Ο Προγραμματισμός σε γλώσσα μηχανής ήταν και είναι μία πολύ δύσκολη δουλειά για τον προγραμματιστή. Έτσι, πολύ σύντομα οι προγραμματιστές άρχισαν να αντιστοιχίζουν τις διάφορες εντολές του κώδικα μηχανής με συντομογραφίες λέξεων της Αγγλικής γλώσσας. Ταυτόχρονα εκτός από τα μνημονικά ονόματα σε εντολές, χρησιμοποιήθηκαν και μνημονικά ονόματα σε διευθύνσεις μνήμης και έτσι μειώθηκαν οι πιθανότητες λάθους από απροσεξία, τόσο στον καθορισμό των εντολών, όσο και στον καθορισμό των διευθύνσεων που χειριζόταν το πρόγραμμα.

Έτσι δημιουργήθηκε μία άλλη κατηγορία γλωσσών, οι συμβολικές γλώσσες, οι οποίες, επειδή είναι στενά συνδεδεμένες με την αρχιτεκτονική της μηχανής, ονομάστηκαν και χαμηλού επιπέδου γλώσσες. Επειδή όμως αναγνωρίζει μόνο τη γλώσσα μηχανής έπρεπε και τα προγράμματα που γραφόταν σε συμβολική γλώσσα, να μεταφραστούν σε γλώσσα μηχανής. Αρχικά αυτό γινόταν χειρογραφικά από ανθρώπους που έπαιρναν ένα πρόγραμμα γραμμένο σε συμβολική γλώσσα και το μετέφραζαν σε γλώσσα μηχανής. Οι άνθρωποι αυτοί ονομάστηκαν *assemblers*, (συναρμολογητές).

Γύρω στα 1950 διαπιστώθηκε ότι η μηχανική εργασία της μετάφρασης θα μπορούσε να γίνει από τους υπολογιστές πιο γρήγορα και με περισσότερη ακρίβεια απ' ότι θα γινόταν από τον άνθρωπο. Έτσι γράφτηκαν τα πρώτα μεταφραστικά προγράμματα που μετέτρεπαν ένα πρόγραμμα από συμβολική γλώσσα σε γλώσσα μηχανής.

Τα προγράμματα σε συμβολική (*assembly*) γλώσσα είναι άμεσα συνδεδεμένα με τον υπολογιστή για τον οποίο γράφτηκαν και δεν μπορούν να μεταφερθούν σε διαφορετικό υπολογιστή. Η κύρια χρήση της γλώσσας χαμηλού επιπέδου προορίζεται για τη συγγραφή προγραμμάτων διαχείρισης του συστήματος, όπως λειτουργικά συστήματα, βοηθητικά προγράμματα (*utilities*), συστήματα αυτόματου ελέγχου με υπολογιστή κ.λ.π.

Οι συμβολικές γλώσσες είναι το χαμηλότερο επίπεδο στο οποίο ο άνθρωπος μπορεί να προγραμματίσει τον υπολογιστή, λέγονται δε και γλώσσες χαμηλού επιπέδου.

Η χρήση των γλωσσών χαμηλού επιπέδου είναι για τη συγγραφή οδηγών συσκευών, τμημάτων του λειτουργικού συστήματος ή τμημάτων εφαρμογών για τα οποία ο προγραμματιστής επιθυμεί να έχει τον απόλυτο έλεγχο της λειτουργίας του επεξεργαστή. Υπάρχουν εργαλεία αυτόματης μετάφρασης προγραμμάτων συμβολικής γλώσσας σε γλώσσα μηχανής που ονομάζονται **συμβολομεταφραστές (*assemblers*)**.

Τέλος, υπάρχουν διαφορετικές γλώσσες μηχανής και συμβολικές γλώσσες για διαφορετικές οικογένειες επεξεργαστών. Ένας προγραμματιστής που γράφει κώδικα

σε συμβολική γλώσσα ζει σε έναν κόσμο που έχει αρκετή επαφή με την αρχιτεκτονική του επεξεργαστή του υπολογιστή στον οποίο απευθύνεται.

### 5.2.2 Γλώσσες Υψηλού Επιπέδου

Για να απαλλαγεί ο προγραμματιστής από όλες τις δυσκολίες της συμβολικής γλώσσας, δημιουργήθηκαν γλώσσες προγραμματισμού υψηλού επιπέδου, που είναι προσανατολισμένες προς την διευκόλυνση του ανθρώπου και όχι του υπολογιστή. Μια γλώσσα υψηλού επιπέδου χρησιμοποιεί μόνο μια απλή γραμμή οδηγιών για να αντιπροσωπεύσει πολλές γραμμές συμβολικής γλώσσας ή γλώσσας μηχανής. Αυτό ελαττώνει τον χρόνο ανάπτυξης του προγράμματος και το καθιστά λιγότερο εξαρτημένο από τον συγκεκριμένο επεξεργαστή. Χρησιμοποιούν ως εντολές απλές λέξεις της αγγλικής γλώσσας και ακολουθούν αυστηρούς συντακτικούς κανόνες. Οι εντολές αυτές είναι περισσότερο κατανοητές και μεταφράζονται από τον υπολογιστή σε γλώσσα μηχανής από ένα ειδικό πρόγραμμα, που ονομάζεται Μεταγλωττιστής ή Μεταφραστής (Compilers) ή Διερμηνευτής (Interpreter), ανάλογα με το προγραμματιστικό περιβάλλον. Τα προγράμματα όμως που είναι γραμμένα σε γλώσσες υψηλού επιπέδου απαιτούν γενικά περισσότερη μνήμη και εκτελούνται με λιγότερη ταχύτητα από τα προγράμματα που είναι γραμμένα σε συμβολική γλώσσα. Δεν χρησιμοποιούν το σύστημα τόσο αποδοτικά όσο τα προγράμματα σε συμβολική γλώσσα.

Επειδή κάθε υπολογιστής μια γλώσσα μόνο «καταλαβαίνει» κάθε πρόγραμμα που έχει γραφεί σε μια γλώσσα προγραμματισμού (C, BASIC, Fortran κλπ, ή ακόμη και σε γλώσσα assembly) πρέπει πρώτα να μεταγλωττιστεί σε γλώσσα μηχανής προτού μπορέσει να εκτελεστεί. Η διαδικασία αυτή της μεταγλώττισης συναντάται σε τρεις, λίγο-πολύ, διαφορετικές μορφές:

1. Προγράμματα που έχουν γραφεί σε γλώσσα assembly «περνάνε» από το λεγόμενο *assembler*, που δεν είναι τίποτε άλλο από ένα εκτελέσιμο (δηλ. ήδη μεταφρασμένο σε γλώσσα μηχανής) πρόγραμμα που διαβάζει το πρόγραμμα assembly και παράγει ένα λειτουργικά ισοδύναμο εκτελέσιμο πρόγραμμα.
2. Προγράμματα που έχουν γραφεί σε μια γλώσσα υψηλού επιπέδου περνάνε από το λεγόμενο *compiler* (γίνονται *compiled*, όπως λέμε).
3. Υπάρχει και μια τρίτη μορφή μεταγλώττισης ενός προγράμματος, αυτή στην οποία δεν δημιουργείται κάποιο εκτελέσιμο πρόγραμμα το οποίο στη συνέχεια εκτελείται, αλλά ο κύκλος μεταγλώττιση-εκτέλεση είναι πολύ πιο βραχύς, και συνήθως εκτελείται μια «εντολή» της γλώσσας υψηλού επιπέδου αμέσως μόλις μεταγλωττιστεί και προτού μεταγλωττιστεί η «επόμενη» εντολή. Προγράμματα που κάνουν αυτού του είδους τη μεταγλώττιση-εκτέλεση λέγονται *interpreters* και χαρακτηριστικό τους είναι ότι συνήθως δε δημιουργούν κάποια μορφή εκτελέσιμου προγράμματος που μπορεί να εκτελεστεί χωρίς τη βοήθεια του interpreter και μόνο από τη μηχανή.



Οι Γλώσσες Προγραμματισμού που έχουν αναπτυχθεί μέχρι σήμερα είναι εκατοντάδες. Οι πιο σημαντικές κατηγορίες γλωσσών ταξινομούνται ως εξής:

A) Με βάση τις ιδέες πάνω στις οποίες βασίζονται.

- 1) Διαδικασιακές (Procedural)
- 2) Αντικειμενοστραφείς (Object - Oriented)
- 3) Συναρτησιακές (Functional Applicative)
- 4) Μη Διαδικαστικές (Non- Procedural)

B) Με βάση την περιοχή χρησιμοποίησης τους.

- 1) Γλώσσες Επιστημονικής Κατεύθυνσης (Science Oriented)
- 2) Εμπορικής Κατεύθυνσης (Business Oriented)
- 3) Προγραμματισμού Συστημάτων (System Programming Language)
- 4) Τεχνητής Νοημοσύνης (Artificial Intelligence)
- 5) Γλώσσες Ειδικής Χρήσης

#### *ΟΙ ΠΙΟ ΔΗΜΟΦΙΛΕΙΣ ΓΛΩΣΣΕΣ ΥΨΗΛΟΥ ΕΠΙΠΕΔΟΥ ΕΙΝΑΙ:*

**FORTRAN (1955):** *Μία γλώσσα για επιστήμονες και μηχανικούς.*

Η ονομασία της λέξης FORTRAN προέρχεται από τις λέξεις FORMULA TRANSLATION (τύπος μετάφρασης). Έχει τη δυνατότητα της έκφρασης κάθε προβλήματος που απαιτεί αριθμητικές πράξεις. Είναι ακόμη δημοφιλής και χρησιμοποιείται κυρίως στην επίλυση μαθηματικών, επιστημονικών και τεχνικών προβλημάτων. Χρησιμοποιείται, όμως και στις εμπορικές και οικονομικές συναλλαγές.

Το γεγονός ότι υπάρχουν ήδη πολλά προγράμματα σε αυτές τις γλώσσες βεβαιώνει ότι θα υπάρχουν για πολύ καιρό ακόμη.

**COBOL (1959):** *Μία γλώσσα για επιχειρήσεις και εμπόριο.*

Η COBOL (Common Business Oriented Language) είναι αντίθετη με τη FORTRAN σε κάθε σχεδόν όψη. Είναι και αυτή από τις πιο ευρέως χρησιμοποιούμενες γλώσσες υψηλού επιπέδου. Σχεδιάστηκε για να αποκριθεί στην άμεση ανάγκη για μία γλώσσα κατάλληλη για τεράστιες απαιτήσεις επεξεργασίας δεδομένων Τραπεζών, Κυβερνητικών Υπηρεσιών και μεγάλων Εταιρειών. Βασίζεται στην αναγνωσιμότητα και διαύγεια και έτσι τα πράγματα μοιάζουν λιγότερο με σαφή και απόκρυφα μαθηματικά και περισσότερο με ανθρώπινες συναλλαγές.

**BASIC (1964):** *Μία πολυχρησιμοποιούμενη γλώσσα.*

Η λέξη BASIC προέρχεται από τα αρχικά των λέξεων «Beginners All – purpose Symbolic Instruction Code» και αποδίδεται ως «συμβολικός κώδικας γενικού τύπου για αρχάριους». Πρόκειται για μία γλώσσα προορισμένη να εισαγάγει τους αρχάριους στον προγραμματισμό των ηλεκτρονικών υπολογιστών. Σήμερα είναι η πιο πλατιά χρησιμοποιούμενη γλώσσα σε μικροϋπολογιστές. Διαδικασιακά γλώσσα, εύκολη

στην εκμάθηση και στο χειρισμό, προσφέρεται για κάθε τύπο προγραμμάτων. Η δομή της είναι επηρεασμένη από τη FORTRAN.

Τα προγράμματα basic όμως είναι γενικά αντιπαραγωγικά και η γλώσσα δεν ενθαρρύνει πολλές εφαρμογές προγραμματισμού.

**PASCAL (1968):** *Η γλώσσα γενικού σκοπού.*

Αναπτύχθηκε από τον Dr. Nicklaus Wirth στο πανεπιστήμιο της Ζυρίχης και πήρε το όνομά της από τον Γάλλο μαθηματικό και φιλόσοφο Blaise Pascal. Είναι απόγονος της ALGOL 60, δημιουργήθηκε κυρίως για διδακτικούς σκοπούς και σχεδιάστηκε για να διδάξει καλές τεχνικές προγραμματισμού. Στην πορεία δέχτηκε μετατροπές και αναθεωρήσεις μέχρι τα τέλη της δεκαετίας του '70 οπότε εμφανίστηκαν μορφές της γλώσσας με δυνατότητα εφαρμογής σε μικροϋπολογιστές. Η πιο γνωστή ήταν η -UCSD Pascal - η χρήση της οποίας αρχικά περιοριζόταν σε mini-υπολογιστές και αργότερα επεκτάθηκε σε μικροϋπολογιστές. Η σπουδαιότερη από τις εκδόσεις της γλώσσας είναι η Turbo Pascal η οποία εμφανίστηκε το 1983. Είναι γλώσσα προγραμματισμού για ένα μεγάλο μέρος της κοινότητας της επιστήμης των Υπολογιστών. Είναι η «φυσική γλώσσα» με την οποία δουλεύουν βασικά όλοι οι σπουδαστές και ερμηνευτές. Είναι μια γλώσσα γενικής κατεύθυνσης και χαρακτηρίζεται από μεγάλο πλούτο εφαρμογών.

**C (1972):** *Μία διαδικαστική γλώσσα.*

Είναι μία δομημένη διαδικασιακή γλώσσα γενικής χρήσης. Χαρακτηρίζεται από τη λιτότητα στην έκφραση, πλούσια συλλογή τελεστών και τύπων δεδομένων. Αναπτύχθηκε στα εργαστήρια Bell της αμερικανικής εταιρείας AT&T. Αποτελεί έργο ενός μόνο ανθρώπου, του Dennis Ritchie. Η C γνωρίζει τεράστια επιτυχία και χρησιμοποιείται στον εμπορικό και ακαδημαϊκό κόσμο. Καθιερώθηκε ως η κύρια γλώσσα για προγραμματισμό λειτουργικών συστημάτων.

**ADA(1970):** *Μια άλλη γλώσσα γενικού σκοπού.*

Το όνομά της δόθηκε προς τιμήν της κόρης του Λόρδου Βύρωνα Augusta Ada Lovelace. Έχει γραφτεί για ενσωματωμένα συστήματα, ωστόσο έχει αποδειχθεί ότι είναι εξίσου καλή για επιστημονικά προβλήματα. Υποστηρίζει μεταφερσιμότητα (portability), λογική αφαίρεση (abstraction), γενικεύσεις (generics) και ταυτόχρονο προγραμματισμό (concurrent programming).

**SNOBOL(1960):** *Μία γλώσσα για διαχείριση κειμένου και συμβολοσειρών.*

Η SNOBOL (StriNgs Oriented symBOL Language) είναι η ιδέα ενός μοπντέλου που μπορεί να προσδιοριστεί από τον προγραμματιστή και μπορεί ακόλουθα να ταιριάξει με ένα κείμενο – που είναι μια σειρά συμβόλων.

**Η LISP (1960) και η PROLOG(1970)** χρησιμοποιούνται για εφαρμογές τεχνητής νοημοσύνης.

**LISP:** *Μία γλώσσα βασισμένη στην Επεξεργασία Λιστών.*

-Η LISP (LISt Processing) είναι πολύ διαφορετική γλώσσα από τις περισσότερες γλώσσες υψηλού επιπέδου. Είναι εφαρμοστική, δηλαδή βασίζεται σχεδόν αποκλειστικά πάνω στην εφαρμογή των συναστίσεων σε ορίσματα. Τα αποτελέσματα των ορισμάτων αποτελούν ορίσματα για άλλες συναρτήσεις. Η LISP έχει αποδειχθεί πολύ αποτυχημένη σε εφαρμογές που περιέχουν συμβολική διαχείριση όπως παιχνίδια, επεξεργασία φυσικής γλώσσας, λήψη αποφάσεων. Τα προγράμματα LISP δε μεταγλωττίζονται συνήθως, αλλά διερμηνεύονται και

εκτελούνται εντολή – εντολή. Το γεγονός αυτό κάνει τη LISP πολύ κατάλληλη ως μία διαλογική γλώσσα, με τον προγραμματιστή ή τον χρήστη να έχει τη δυνατότητα να παρακολουθεί την εκτέλεση.

**PROLOG:** *Μία γλώσσα βασισμένη στην προτασιακή λογική.*

-Η PROLOG (PROgramming in LOGic) είναι βασισμένη στις προτάσεις που προσδιορίζουν ότι ορισμένα γεγονότα ακολουθούνται λογικά από σύνολα άλλων γεγονότων. Είναι μια ταιριαστή γλώσσα για αναζήτηση σε βάσεις δεδομένων. Είναι η πιο κατάλληλη γλώσσα για αναζητήσεις και διαχείριση βάσεων γνώσης. Έχει ένα φάσμα εφαρμογών παρόμοιο με της LISP.

**Η APL (1960):** *Η λιγότερο συμβατική από τις κοινές γλώσσες.*

Η σχεδίασή της περιέχει ανώτερη συμπαγότητα, αντικαθιστώντας μεγάλα τμήματα προγραμμάτων, που μπορεί να απαιτούσαν διάφορους επικαλυπτόμενους βρόγχους, από απλά σύμβολα. Χρησιμοποιείται ευρέως σε επιστημονικές και τεχνολογικές εφαρμογές. Έχει διάφορες γρήγορες και απλές υλοποιήσεις. Έχει παγιωθεί σαν μία ισχυρή αριθμομηχανή γραφείου. Το πιο δυνατό χαρακτηριστικό της είναι το πλούσιο σύνολο τελεστών της.

Τα προγράμματα σε APL διερμηνεύονται – δε μεταγλωττίζονται.

Έτσι λοιπόν μπορούμε να αναφέρουμε την ακόλουθη αντιστοιχία γλωσσών προγραμματισμού και εφαρμογών τους:

Στις επιστημονικές εφαρμογές οι γλώσσες που χρησιμοποιούνται είναι οι FORTRAN, C και APL

Σε εκπαιδευτικές εφαρμογές οι BASIC, C και LOGO

Σε εμπορικές εφαρμογές οι COBOL και RPG

Σε ειδικές εφαρμογές οι LISP και PROLOG

Και τέλος σε

Επιστημονικές σε συνεργασία με Εμπορικές εφαρμογές οι PL-1, PASCAL, ADA κ.λ.π.

## ΜΙΑ ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΓΛΩΣΣΩΝ C++ ΚΑΙ JAVA

C++

Στα μέσα της 10ετίας του 1980 έχει κάνει την εμφάνισή της η C++, που είναι μια επέκταση της C προς την κατεύθυνση του *object-oriented* προγραμματισμού. Το μερίδιο της C++ στους προγραμματιστές αυξάνει διαρκώς, μια και προσφέρεται ιδιαίτερα για δουλειές όπως την κατασκευή φιλικών *user interfaces* (δηλ. περιβαλλόντων αλληλεπίδρασης με το χρήστη), και το *object oriented programming* προσφέρει ένα πολύ καλό πρότυπο οργάνωσης μεγάλων προγραμμάτων που έχουν να κάνουν όχι τόσο με αριθμητικούς υπολογισμούς αλλά με διαχείριση πολλών διαφορετικών τύπων δεδομένων πάνω στα οποία θέλουμε κάπως να κάνουμε ίδιες εργασίες.

Τα διάφορα εικονίδια δηλ. ανταποκρίνονται στο ίδιο ``σήμα" (open) με διαφορετικό τρόπο, που εν γένει δεν είναι γνωστός στο περιβάλλον σύστημα αλλά υλοποιείται από το κάθε αντικείμενο (παράθυρο, modem, πίνακα ελέγχου, κλπ) με δικό του ξεχωριστό τρόπο.

Εναλλακτικά θα έπρεπε το περιβάλλον σύστημα να γνωρίζει τα ``εσωτερικά" του κάθε αντικειμένου που ``φιλοξενεί", πράγμα ανέφικτο. Αυτού του είδους η προσέγγιση είναι χαρακτηριστική των object oriented συστημάτων.

Το παρακάτω πρόγραμμα τυπώνει "hello, world" στην οθόνη.

```
#include <stdio.h>

main()
{
    printf("hello, world\n");
}
```

Στοιχεία του προγράμματος

Τα προγράμματα της C αποτελούνται από ορισμένες βασικές τάξεις στοιχείων: Τα κενά, όπως ο χαρακτήρας ' ' και η αλλαγή της γραμμής. Αυτά γενικά δεν επηρεάζουν καθόλου τη συμπεριφορά του προγράμματος. Για παράδειγμα μπορούσαμε να γράψουμε το πρόγραμμα ως εξής:

```
#include <stdio.h>
main(){printf("hello, world\n");}
```

Χρησιμοποιούμε κενά (*spaces*) για να κάνουμε το πρόγραμμα πιο εύληπτο.

Οι εντολές του προεπεξεργαστή. Αυτές είναι όλες οι γραμμές που αρχίζουν με το χαρακτήρα #. Απλώς τις γράφουμε στην αρχή του προγράμματος για να μπορέσουμε να εκτελέσουμε ορισμένες βασικές λειτουργίες, όπως το να τυπώσουμε τα αποτελέσματά μας.

Τα ονόματα των συναρτήσεων (*functions*) και των μεταβλητών. Αυτά αρχίζουν με έναν λατινικό αλφαβητικό χαρακτήρα (a-z) και μπορούν να ακολουθούνται από άλλους ή/και ψηφία.

Οι δηλώσεις των συναρτήσεων. Αυτές γράφονται ως το όνομα τη συνάρτησης ακολουθούμενο από παρενθέσεις και το περιεχόμενο της συνάρτησης μέσα σε άγκιστρα ( { } ). Η συνάρτηση main έχει ειδικό νόημα. Ορίζει το σημείο από το οποία θα αρχίσει να εκτελείται το πρόγραμμα.

Το περιεχόμενο των συναρτήσεων. Αυτό αποτελείται από εντολές. Κάθε εντολή *τερματίζεται* με μια λατινική άνω τελεία (*semicolon*) (; - ελληνικό ερωτηματικό).

Οι εντολές μέσα στη συνάρτηση εκτελούνται με τη σειρά από πάνω προς τα κάτω εκτός να έχουμε ορίσει κάτι διαφορετικό.

Οι κλήσεις άλλων συναρτήσεων. Αυτές αποτελούν ένα βασικό είδος εντολής.

Η κλήση μιας συνάρτησης αποτελείται από το όνομα της συνάρτησης ακολουθούμενο από το όρισμα (*argument*) της συνάρτησης μέσα σε παρενθέσεις.

Αν η συνάρτηση δέχεται πολλαπλές παραμέτρους, τότε αυτές χωρίζονται με κόμματα.

Ορισμός απλών συναρτήσεων

Τη συνάρτηση `main` την ορίζουμε εμείς και τη συνάρτηση `printf` την καλούμε.

Σε ένα πρόγραμμα είναι δυνατός ο ορισμός (*definition*) και η κλήση (*call*) και άλλων συναρτήσεων. Το παρακάτω πρόγραμμα τυπώνει "hello, world" στην οθόνη.

```
#include <stdio.h>

hello()
{
    printf("hello, world\n");
}

main()
{
    hello();
}
```

Με τον τρόπο αυτό μπορούμε να ομαδοποιήσουμε τα τμήματα του προγράμματός μας και να χρησιμοποιήσουμε ορισμένα τμήματα πολλές φορές.

Όταν καλείται μια συνάρτηση, αυτή εκτελείται βήμα προς βήμα μέχρι να τελειώσουν οι εντολές που την απαρτίζουν. Τότε η εκτέλεση του προγράμματος επιστρέφει (*returns*) στην επόμενη εντολή από αυτή που κάλεσε τη συνάρτηση.

Τυπική περιγραφή γλωσσών

### Σύνταξη

Είναι ο τρόπος με τον οποίο τοποθετούνται στη σειρά τα συστατικά στοιχεία της γλώσσας για να αποτελέσουν ένα πρόγραμμα.

### Σημασιολογία (*Semantics*)

Η σημασία που αποδίδεται στα συστατικά στοιχεία ενός προγράμματος κατά τη μετάφραση και την εκτέλεσή του.

## Java

### Τα χαρακτηριστικά της Java

Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας.

Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh (πλέον τρέχουν και σε Playstation καθώς και σε άλλες παιχνιδιομηχανές) χωρίς να χρειαστεί να ξαναγίνει διάταξη αποδελτίωσης (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα.

Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή αναξάρτητα του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, Unix, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας μπορεί και «καταλαβαίνει» διαφορετικό *assembly* κώδικα.

Ο συναρμολογούμενος (*assembly*) κώδικας που τρέχει σε Windows είναι διαφορετικός από αυτόν που τρέχει σε ένα Macintosh. Η λύση δόθηκε με την ανάπτυξη της *Εικονικής Μηχανής* (*Virtual Machine* ή VM ή EM στα ελληνικά).

### Μεταγλώττιση και εκτέλεση

Για να μεταγλωττίσουμε το πρόγραμμα πρέπει να το αποθηκεύσουμε σε ένα αρχείο με όνομα ίδιο με το όνομα της κλάσης που περιέχει τη συνάρτηση *main*. Το επίθεμα του αρχείου πρέπει να είναι *.java*.

Με την εντολή *javac όνομα αρχείου* μεταγλωττίζουμε το πρόγραμμα από Java σε μορφή που να μπορεί να εκτελεστεί.

Για να εκτελέσουμε το πρόγραμμα χρησιμοποιούμε την εντολή *java όνομα κλάσης*.

Παράδειγμα:

```
C:\>javac Hello.java

C:\src>java Hello
Hello world.

C:\>
```

### Μηχανισμός μεταγλώττισης και εκτέλεσης

Ο μεταγλωττιστής της Java (javac) μετατρέπει το πηγαίο πρόγραμμα από Java σε εντολές της ιδεατής μηχανής Java (*Java virtual machine*) (JVM)

Το περιβάλλον εκτέλεσης της Java (java) φορτώνει την κλάση που ορίζουμε, φορτώνει όποιες ακόμα κλάσεις απαιτούνται, τις συνδέει με την κλάση που ζητήσαμε να εκτελεστεί και εκτελεί τις εντολές JVM αρχίζοντας από τη μέθοδο main.

### Συνάρτηση σε Java

```
// Return n!
static public int factorial(int n) {
    int result;
    int counter;

    counter = n;
    result = 1;
    while (counter > 0) {
        result = result * counter;
        counter = counter - 1;
    }
    return (result);
}
```

## Στοιχεία του προγράμματος

Τα προγράμματα της Java αποτελούνται από ορισμένες βασικές τάξεις στοιχείων:

Τα κενά, όπως ο χαρακτήρας ' ' και η αλλαγή της γραμμής. Αυτά γενικά δεν επηρεάζουν καθόλου τη συμπεριφορά του προγράμματος. Για παράδειγμα μπορούσαμε να γράψουμε το πρόγραμμα ως εξής:

```
class Hello { public static void main(String args[]) {  
System.out.println("Hello, world."); } }
```

Χρησιμοποιούμε κενά (*spaces*) για να κάνουμε το πρόγραμμα πιο κατανοητό.

Τα ονόματα των συναρτήσεων, κλάσεων και των μεταβλητών. Αυτά αρχίζουν με έναν λατινικό αλφαβητικό χαρακτήρα και μπορούν να ακολουθούνται από άλλους ή/και ψηφία.

Η δήλωση της κλάσης. Αυτή γράφεται ως το όνομα τη κλάσης ακολουθούμενο από το περιεχόμενο της κλάσης μέσα σε άγκιστρα ( { } ). Το όνομα του αρχείου πρέπει να είναι ίδιο με το όνομα της κλάσης. Οι συναρτήσεις που ορίζονται στην κλάση αυτή είναι στην πραγματικότητα μέθοδοι.

Οι δηλώσεις των συναρτήσεων. Γράφονται ως το όνομα τη συνάρτησης ακολουθούμενο από παρενθέσεις και το περιεχόμενο της συνάρτησης μέσα σε άγκιστρα ( { } ). Αν μια συνάρτηση δέχεται ορίσματα, αυτά δηλώνονται μέσα στις παρενθέσεις που ακολουθούν το όνομα τις συνάρτησης.

Η συνάρτηση *main* έχει ειδικό νόημα. Ορίζει το σημείο από το οποία θα αρχίσει να εκτελείται το πρόγραμμα.

Το περιεχόμενο των συναρτήσεων. Αυτό αποτελείται από εντολές. Κάθε εντολή *τερματίζεται* με μια λατινική άνω τελεία (*semicolon*) (; - ελληνικό ερωτηματικό).

Οι εντολές μέσα στη συνάρτηση εκτελούνται με τη σειρά από πάνω προς τα κάτω εκτός να έχουμε ορίσει κάτι διαφορετικό.

Οι κλήσεις άλλων συναρτήσεων. Αυτές αποτελούν ένα βασικό είδος εντολής.

Η κλήση μιας συνάρτησης αποτελείται από το όνομα της συνάρτησης ακολουθούμενο από το όρισμα (*argument*) της συνάρτησης μέσα σε παρενθέσεις. Αν η συνάρτηση δέχεται πολλαπλές παραμέτρους, τότε αυτές χωρίζονται με κόμματα.

## Ορισμός απλών συναρτήσεων

Σε ένα πρόγραμμα είναι δυνατός ο ορισμός (*definition*) και η κλήση (*call*) και άλλων συναρτήσεων.



Το παρακάτω πρόγραμμα τυπώνει και αυτό "hello, world" στην οθόνη.

```
class HelloFun {
    static void hello() {
        System.out.println("Hello, world.");
    }
    public static void main(String args[]) {
        hello();
    }
}
```

Με τον τρόπο αυτό μπορούμε να ομαδοποιήσουμε τα τμήματα του προγράμματός μας και να χρησιμοποιήσουμε ορισμένα τμήματα πολλές φορές.

Όταν καλείται μια συνάρτηση, αυτή εκτελείται βήμα προς βήμα μέχρι να τελειώσουν οι εντολές που την απαρτίζουν. Τότε η εκτέλεση του προγράμματος επιστρέφει (*returns*) στην επόμενη εντολή από αυτή που κάλεσε τη συνάρτηση.

Στην Java οι διαδικασίες ορίζονται ως συναρτήσεις που δεν επιστρέφουν τίποτα. Αυτό γίνεται γράφοντας ως τύπο που επιστρέφει η συνάρτηση τη λέξη *void* (κενό).

### 5.2.3 Γλώσσες Τέταρτης Γενιάς

Οι γλώσσες 4ης γενιάς δίνουν στον χρήστη ένα νέο δομημένο τρόπο για την ανάπτυξη Πληροφοριακών Συστημάτων. Είναι γλώσσες υψηλής παραγωγικότητας, γιατί με τη χρήση τους απαιτείται λιγότερος χρόνος για την ανάπτυξη των εφαρμογών απ' ό τι με τις κλασικές γλώσσες προγραμματισμού.

Επιπλέον η χρήση τους είναι δυνατή από άτομα που δεν έχουν γνώσεις προγραμματισμού, αλλά απλώς διαθέτουν μία εξοικείωση με τη λειτουργία του υπολογιστή.

Οι γλώσσες 4ης γενιάς έχουν μία κατηγορία *δυναμικών εντολών*, που δίνουν στον χρήστη τη δυνατότητα εξαιρετική ευκολία στο χρήστη, για να δημιουργεί αρχεία, να ενημερώνει, να σχεδιάζει οθόνες για την εκμετάλλευσή τους και να δημιουργεί εκτυπώσεις. Τα προηγούμενα επιτυγχάνονται με τη χρήση πολύ λίγων εντολών, η κάθε μια από τις οποίες θα αντιστοιχούσε σε ολόκληρο πρόγραμμα σε μία κλασική γλώσσα προγραμματισμού.

Στις γλώσσες 4ης γενιάς ο χρήστης καθορίζει το είδος της επεξεργασία, που απαιτεί η εφαρμογή, και όχι τον τρόπο εκτέλεσής της από τον υπολογιστή.

Οι γλώσσες 4ης γενιάς είναι διαλογικές (interactive) και έχουν την δυνατότητα εμφάνισης βοηθητικών μηνυμάτων που έχουν σχέση με τη χρήση των διάφορων εντολών της γλώσσας από το χρήστη. Οι γλώσσες αυτές συνήθως αποτελούν εργαλεία πακέτων για Συστήματα Διοίκησης Βάσεων Δεδομένων, όπως π.χ. INGRES, ORACLE κλπ.

Η υλοποίηση ενός πληροφοριακού συστήματος είναι ταχύτερη με τη χρήση γλωσσών 4<sup>ης</sup> γενιάς. Οι εντολές των γλωσσών 4<sup>ης</sup> γενιάς είναι λιγότερες, αλλά περισσότερο περιεκτικές από τις αντίστοιχες μίας συμβατικής γλώσσας προγραμματισμού (COBOL, FORTRAN κλπ.).

Αυτό έχει ως αποτέλεσμα μικρότερα προγράμματα, πράγμα το οποίο σημαίνει λιγότερα λάθη, καθώς επίσης και μικρότερο χρόνο για την ανεύρεσή τους και για την μελλοντική συντήρησή τους.

### 5.3 Είδη Προγραμματισμού

Στη διαρκώς επιταχυνόμενη εξέλιξη υλικού και λογισμικού εντάσσεται και η ανάπτυξη του προγραμματισμού. Οι απαιτήσεις για αξιοπιστία, ποιότητα και χαμηλό κόστος στις προσφερόμενες υπηρεσίες λογισμικού, είχαν ως αποτέλεσμα τη συστηματοποίηση του προγραμματισμού.

Τις τελευταίες δεκαετίες δημιουργούνται συνεχώς νέες μέθοδοι, τεχνικές και πρότυπα στον προγραμματισμό, καθώς και νέες εξειδικευμένες γλώσσες προγραμματισμού.

#### 5.3.1 Διαδικασιακός Προγραμματισμός

Στο Διαδικασιακό Προγραμματισμό (Procedural Programming) το πρόγραμμα είναι γραμμένο σε κάποια γλώσσα προγραμματισμού εξαρτημένη από τη διαδικασία (procedure oriented). Η διαδικασία είναι το χαρακτηριστικό των περισσότερων γλωσσών της τρίτης γενιάς.

Μια γλώσσα που είναι εξαρτημένη από τη διαδικασία, δίνει έμφαση στον προγραμματισμό των υπολογιστικών και λογικών διαδικασιών οι οποίες απαιτούνται για την επίλυση του προβλήματος. Χαρακτηριστικά παραδείγματα γλωσσών που είναι εξαρτημένες από τη διαδικασία είναι: COBOL, FORTRAN, PASCAL κλπ.

Προβλήματα που επιλύονται αλγοριθμικά, προσφέρονται για Διαδικασιακό Προγραμματισμό. Παραδείγματα τέτοιων προγραμμάτων είναι τα προγράμματα λογιστικών εφαρμογών, εμπορικών εφαρμογών, διοικητικών εφαρμογών (προσωπικού, μισθοδοσίας), μεταφορών (κράτηση θέσεων, έκδοση εισιτηρίων) κλπ.

Στο διαδικασιακό προγραμματισμό το πρόγραμμα είναι μια διαδικασία η οποία εκτελεί ένα προς ένα τα βήματα του αλγόριθμου επίλυσης του προβλήματος. Οι εντολές του προγράμματος εκτελούνται διαδοχικά, εκτός αν υπάρχουν συνθήκες, οπότε κάποιες εντολές παραλείπονται και δημιουργούνται διακλαδώσεις ή κάποιες εντολές επαναλαμβάνονται.

Οι γλώσσες της τρίτης γενιάς κατάφεραν να απομακρύνουν τον προγραμματισμό από το επίπεδο της μηχανής προς το επίπεδο του ανθρώπου. Το αποτέλεσμα ήταν να γίνει ο προγραμματισμός μια ευχάριστη και δημιουργική εργασία, στην οποία οι προγραμματιστές απελευθερωμένοι από δουλειές ρουτίνας πέτυχαν να δημιουργήσουν νέα εργαλεία και τεχνικές και η εξέλιξη της Πληροφορικής να συνεχίζεται από τότε με γρήγορους ρυθμούς.

### 5.3.2 Δομημένος Προγραμματισμός

- Ο δομημένος προγραμματισμός (*structured programming*) βασίζεται στην οργάνωση του προγράμματος σε μικρές, αυτόνομα κατανοητές, οντότητες.
- Για να επιτευχθεί αυτό κάθε οντότητα έχει ένα σημείο εισόδου και ένα σημείο εξόδου.
- Η δομή του προγράμματος βασίζεται σε τρεις μόνο λογικές δομές:
  1. ακολουθία (*sequence*)
  2. επιλογή (*condition*)
  3. επανάληψη (*repetition*)
- Με τη χρήση των παραπάνω δομών μπορεί να εκφραστεί οποιοσδήποτε αλγόριθμος.
- Σε περιπτώσεις που η σχολαστική τήρηση των παραπάνω κάνει το πρόγραμμα δυσανάγνωστο πρέπει οι εξαιρέσεις να κωδικοποιούνται δομημένα, τεκμηριωμένα, με οικονομία στη χρήση τους, και με τάξη.

### 5.3.3 Παράλληλος Προγραμματισμός

Όπως στις περισσότερες περιπτώσεις εξέλιξης του λογισμικού, έτσι και στον Παράλληλο προγραμματισμό (*Parallel Programming*) η καθιέρωσή του οφείλεται στην εξέλιξη του υλικού. Η εμφάνιση της αρχιτεκτονικής των πολλών επεξεργαστών οι οποίοι χρησιμοποιούν κοινή μνήμη είχε ως συνέπεια την ανάπτυξη των παραλλήλων αλγορίθμων οι οποίοι επέβαλαν την καθιέρωση του παράλληλου προγραμματισμού.

Ο Παράλληλος Προγραμματισμός εκτός από τις δομές του παραδοσιακού διαδικασιακού προγραμματισμού διαθέτει δομές που επιτρέπουν την ταυτόχρονη εκτέλεση διαδικασιών από διαφορετικούς επεξεργαστές.

Έτσι δίνεται η δυνατότητα ώστε διάφορα υποπρογράμματα ενός προγράμματος να εκτελούνται παράλληλα από δύο ή περισσότερους επεξεργαστές του υπολογιστή.

Η ανάγκη της ταυτόχρονης εκτέλεσης καθώς και της επικοινωνίας μεταξύ των εκτελουμένων διεργασιών καθορίζει και τα πλαίσια των απαιτήσεων στο χώρο του προγραμματισμού. Για την κάλυψη των νέων αναγκών έγινε επέκταση είτε σε κάποιες κλασσικές γλώσσες διαδικασιακού προγραμματισμού όπως οι Ada, Modula 2 Concurrent C είτε στο σχεδιασμό νέων γλωσσών.

Επίσης, δόθηκε έμφαση στον παράλληλο προγραμματισμό για πιο αποτελεσματική εκμετάλλευση των νέων δυνατοτήτων στα πλαίσια της αρχιτεκτονικής των παραλλήλων επεξεργασιών. Αντιπροσωπευτική γλώσσα σχεδιασμένη για παράλληλη επεξεργασία είναι η γλώσσα Occam.

#### 5.3.4 Αντικειμενοστρεφής Προγραμματισμός

Ο Αντικειμενοστρεφής Προγραμματισμός (Object Oriented Programming) είναι μια τεχνική στην οποία υπάρχει ενσωμάτωση των δεδομένων και του τρόπου χειρισμού αυτών μέσα από την έννοια του αντικειμένου.

Ένα αντικείμενο αποτελείται από μια σειρά δεδομένων που αποτελούν τα χαρακτηριστικά του και μία σειρά μεθόδων ή ενεργειών που σχετίζονται με την επεξεργασία των δεδομένων και καθορίζουν τη συμπεριφορά του αντικειμένου στο πρόγραμμα.

Οι μέθοδοι χειρισμού των δεδομένων μπορεί να είναι διαδικασίες ή συναρτήσεις του χρήστη στο κυρίως πρόγραμμα.

Τα αντικείμενα σε ένα αντικειμενοστρεφές πρόγραμμα μπορεί να σχηματίζουν κλάσεις ιεραρχικά δομημένες. Με την ιεραρχική δόμηση των κλάσεων οι υποκλάσεις κληρονομούν χαρακτηριστικά και ιδιότητες των κλάσεων από τις οποίες προήλθαν, έχοντας επιπλέον την δυνατότητα να προστεθούν σε αυτές (τις υποκλάσεις) νέες ιδιότητες και χαρακτηριστικά.

#### 5.3.5 Συναρτησιακός Προγραμματισμός

Ο συναρτησιακός προγραμματισμός (*functional programming*) βασίζεται στην αποτίμηση συναρτήσεων αντί για την εκτέλεση εντολών. Προγράμματα βασισμένα στο συναρτησιακό προγραμματισμό είναι γραμμένα με βάση συναρτήσεις που αποτιμούν βασικές τιμές. Μια συναρτησιακή γλώσσα προγραμματισμού υποστηρίζει και ενθαρρύνει το συναρτησιακό προγραμματισμό.

Βασικά χαρακτηριστικά του συναρτησιακού προγραμματισμού είναι:

- οι συναρτήσεις ως παράμετροι,
- ο ράθυμος υπολογισμός (*lazy evaluation*),
- η αναφορική διαφάνεια (*referential transparency*).

Ορισμένες γνωστές συναρτησιακές γλώσσες προγραμματισμού είναι οι:

- Erlang
- Haskell
- Hope
- Lisp
- Miranda
- ML
- Scheme

Οι βασικές ιδέες της γλώσσας Lisp αναπτύχθηκαν από τον John McCarthy το 1956 σε ένα ερευνητικό πρόγραμμα για τεχνητή νοημοσύνη. Στόχος του McCarthy ήταν η ανάπτυξη μιας αλγεβρικής γλώσσας επεξεργασίας λιστών για έρευνα στην τεχνητή νοημοσύνη. Ανάμεσα στα έτη 1960 και 1965 υλοποιήθηκε η διάλεκτος Lisp 1.5 η οποία τη δεκαετία του 1970 είχε οδηγήσει στις διαλέκτους MacLisp και InterLisp. Στα μέσα της δεκαετίας του 1970 οι Sussman και Steele Jr. ανέπτυξαν τη διάλεκτο Scheme με βάση ιδέες από τη σημασιολογία των γλωσσών προγραμματισμού. Στη δεκαετία του 1980 έγινε προσπάθεια για ενοποίηση των διαλέκτων της Lisp κάτω από το πρότυπο της Common Lisp.

### 5.3.6 Λογικός Προγραμματισμός

Ο Λογικός Προγραμματισμός (Programming in Logic) διαφέρει από τον διαδικασιακό προγραμματισμό ως προς το εξής: Στο πρόγραμμα δεν περιγράφεται ο αλγόριθμος επίλυσης του σχετικού προβλήματος, αλλά η γνώση που σχετίζεται με το πρόβλημα. Έτσι ο Λογικός Προγραμματισμός, σε αντίθεση με το διαδικασιακό, στηρίζεται στη γνώση.

Η γνώση περιγράφεται υπό μορφή γεγονότων και κανόνων ενώ ορίζεται μία πρόταση – στόχος προς απόδειξη. Για την απόδειξη αυτής της πρότασης – στόχου, το πρόγραμμα εφαρμόζει τους κανόνες πάνω στα γεγονότα.

Ο Λογικός προγραμματισμός εμφανίστηκε στα τέλη της δεκαετίας του 1970. Υπήρξε το αντικείμενο ερευνών σε περιοχές της επιστήμης, όπως η αυτόματη απόδειξη θεωρημάτων και η Τεχνητή Νοημοσύνη. Αποτέλεσμα των ερευνών αυτών ήταν και η δημιουργία κατάλληλων γλωσσών λογικού προγραμματισμού με τυπική εκπρόσωπο την Prolog.

Στο Διαδικασιακό Προγραμματισμό ο προγραμματιστής προσδιορίζει τα βήματα του αλγόριθμου για την επίλυση κάποιου προβλήματος. Στο Λογικό Προγραμματισμό ο προγραμματιστής περιγράφει τη γνώση του σχετικά με κάποιο θέμα και προσδιορίζει τους στόχους.

Ένα πρόγραμμα σύμφωνα με το Λογικό Προγραμματισμό περιλαμβάνει ένα σύνολο γνωστών γεγονότων και κανόνων και μία πρόταση – στόχο η οποία πρέπει να αποδειχθεί αληθής με τη χρήση των κανόνων. Η εφαρμογή των κανόνων για την απόδειξη του στόχου δεν αποτελεί μέρος του προγραμματισμού αλλά εκτελείται αυτόματα.

## Κεφάλαιο 6: Σύγχρονα Λειτουργικά Συστήματα- Windows XP

Τα Windows XP είναι η τελευταία έκδοση του δημοφιλέστατου λειτουργικού συστήματος (operating system) της Microsoft, των Microsoft Windows.

Κυκλοφορούν τα:

- Windows XP Home Edition και απευθύνεται σε όσους χρησιμοποιούν τον υπολογιστή στο περιβάλλον του σπιτιού τους και
- Windows XP Professional Edition, και απευθύνεται σε χρήστες που χρησιμοποιούν τον υπολογιστή στην εργασία τους.

Οι βασικές μέθοδοι χρήσης του υπολογιστή και διαχείρισης των αρχείων δε διαφέρουν μεταξύ των δύο αυτών εκδόσεων.

Για να μπορέσουμε να χρησιμοποιήσουμε έναν υπολογιστή, πρέπει στην κύρια μνήμη RAM του υπολογιστή να εκτελείται ένα λειτουργικό σύστημα. Όταν ο υπολογιστής μας είναι σβηστός, το λειτουργικό σύστημα του (π.χ. τα Windows XP) και τα άλλα προγράμματα που έχουμε εγκαταστήσει βρίσκονται αποθηκευμένα στο σκληρό δίσκο. Με την εκκίνηση του υπολογιστή μας, το λειτουργικό σύστημα φορτώνεται αυτόματα από το σκληρό δίσκο στην κύρια μνήμη RAM. Αν για οποιοδήποτε λόγο η φόρτωση του λειτουργικού συστήματος αποτύχει ή παρουσιαστεί κάποιο πρόβλημα, ο υπολογιστής δε θα είναι δυνατόν να λειτουργήσει ή θα υπολειτουργεί.

### **Λειτουργικό σύστημα**

Το λειτουργικό σύστημα είναι το βασικότερο στοιχείο λογισμικού ενός υπολογιστικού συστήματος. Επιτρέπει την επικοινωνία μεταξύ του υπολογιστή και του χρήστη και χειρίζεται την υλοποίηση αυτής της αλληλεπίδρασης. Ταυτόχρονα ελέγχει, διαχειρίζεται και εξασφαλίζει την ομαλή λειτουργία όλων των συσκευών του υπολογιστή.

- Αποτελεί μια τυποποιημένη πλατφόρμα, στην οποία ο χρήστης μπορεί να εκτελεί τις τις εφαρμογές που θέλει, αρκεί αυτές να έχουν σχεδιαστεί για το συγκεκριμένο λειτουργικό σύστημα. Ακόμα εκτελεί διάφορες διεργασίες στο παρασκήνιο (όπως εκτύπωση), ώστε να εξυπηρετεί τις ανάγκες των εφαρμογών που χρησιμοποιούμε.
- Παρέχει τη διασύνδεση (Interface) για την επικοινωνία του χρήστη με το υλικό και το λογισμικό ενός υπολογιστή.

Άλλα λειτουργικά συστήματα είναι:

- MS-DOS
- LINUX
- UNIX .

## 6.1 Δίκτυα

### 6.1.1 Γενική περιγραφή-ορισμός δικτύου

Η επικοινωνία μεταξύ των ανθρώπων αποτελεί έναν από τους κυριότερους παράγοντες για την πρόοδο της ανθρωπότητας. Μέσω αυτής επιτυγχάνεται η συνεργασία και η διασπορά των πόρων και των μέσων σε όλο τον κόσμο. Η πληροφορία είναι ένα από τα πολυτιμότερα αγαθά στα οποία πρέπει να έχει πρόσβαση όποιος το επιθυμεί. Ο αιώνας που διανύουμε μπορεί να χαρακτηριστεί ως ο αιώνας της Πληροφορίας. Οι κάτοχοι των πληροφοριών αποτελούν την ηγεσία της σημερινής κοινωνίας. Ένα μεγάλο πρόβλημα είναι η ύπαρξη μεγάλου όγκου πληροφοριών και η αναζήτηση εκείνων οι οποίες είναι χρήσιμες. Η διάδοση των πληροφοριών επιτυγχάνεται μέσω των Δικτύων.

Δίκτυο είναι μια ομάδα υπολογιστών συνδεδεμένων μεταξύ τους είτε ενσύρματα είτε ασύρματα, η οποία επιτρέπει σε πολλούς ανθρώπους να ανταλλάσσουν πληροφορίες και να διαμοιράζονται εξοπλισμό.

### 6.1.2 Διαχωρισμός δικτύων

Ανάλογα με την τοποθεσία των υπολογιστών που συνδέονται, ένα δίκτυο χαρακτηρίζεται ως:

- Τοπικό Δίκτυο (Local Area Network – LAN)
- Μητροπολιτικό Δίκτυο (Metropolitan Area Network – MAN)
- Δίκτυο Ευρείας Περιοχής (Wide Area Network – WAN)
- Διαδίκτυο

Δύο υπολογιστές που βρίσκονται στον ίδιο χώρο μπορούν να συνδεθούν με ένα καλώδιο. Όταν όμως η απόσταση μεταξύ των υπολογιστών μεγαλώνει, χρησιμοποιούνται διάφοροι τρόποι σύνδεσης: κοινές τηλεφωνικές γραμμές, ISDN γραμμές, μισθωμένες τηλεπικοινωνιακές γραμμές διαφόρων τεχνολογιών, ασύρματες ζεύξεις, ακόμη και συνδέσεις μέσω τηλεπικοινωνιακών δορυφόρων, όταν απαιτείται μετάδοση δεδομένων σε πολύ μεγάλες αποστάσεις.

### 6.1.3 Είδη Δικτύων

Τα δίκτυα υπολογιστών διακρίνονται με βάση:

- ❖ Τη γεωγραφική κατανομή (τοπικά δίκτυα, ευρείας περιοχής δίκτυα, διαδίκτυο).
- ❖ Τη φυσική τοπολογία.
- ❖ Τα πρωτόκολλα δικτύων.
- ❖ Τον τρόπο σύνδεσης.
- ❖ Τον τρόπο επεξεργασίας (συγκεντρωτικά δηλ. κλασικά δίκτυα που λειτουργούν με εξυπηρετητή και πελάτες, καταμεμημένα και αποκεντρωμένα).
- ❖ Τοπικά δίκτυα



**Τοπικό Δίκτυο (Local Area Network - LAN)** λέγεται το δίκτυο στο οποίο οι υπολογιστές που το συγκροτούν βρίσκονται σε σχετικά κοντινή απόσταση μεταξύ τους. Χαρακτηριστικό των δικτύων αυτών είναι ότι για τη σύνδεση τους χρησιμοποιούνται ιδιωτικά μέσα μετάδοσης. Ένα κλασικό παράδειγμα τοπικού δικτύου είναι αυτό των σχολικών μονάδων.

❖ **Δίκτυα ευρείας περιοχής**

Τα **Δίκτυο Ευρείας Περιοχής (Wide Area Network - WAN)**, καλύπτουν ανάγκες δικτύωσης υπολογιστών σε μεγάλες αποστάσεις. Ένα τέτοιο δίκτυο μπορεί να εκτείνεται σε μια ή και περισσότερες πόλεις, χώρες ή, ακόμα, και ηπείρους. Χαρακτηριστικό των δικτύων αυτών είναι ότι χρησιμοποιούνται καλώδια τηλεφωνικών γραμμών ή τηλεπικοινωνιακοί δορυφόροι.

Ένα Δίκτυο Ευρείας Περιοχής συνδέει περισσότερα από ένα Τοπικά Δίκτυα. Μπορεί όμως να συνδέει απλώς αυτόνομους υπολογιστές που βρίσκονται σε μεγάλες αποστάσεις. Επίσης, ένα Δίκτυο Ευρείας Περιοχής μπορεί να αποτελείται από ένα ή περισσότερα Τοπικά Δίκτυα, από ομάδες Τοπικών Δικτύων και από αυτόνομους υπολογιστές που λειτουργούν σε διαφορετικά περιβάλλοντα και ανήκουν σε διαφορετικές κατηγορίες.

Σημαντικά πλεονεκτήματα σε ένα τοπικό δίκτυο προσφέρουν οι δρομολογητές (routers) οι κατανεμητές και οι γέφυρες. Με τους δρομολογητές συνδέονται επίσης μεταξύ τους περισσότερα Δίκτυα Ευρείας Περιοχής και έτσι σχηματίζονται ακόμα μεγαλύτερα δίκτυα αυτής της κατηγορίας. Αυτή η τεχνική σύνδεσης χρησιμοποιείται παγκοσμίως, και έτσι, δημιουργήθηκε και το γνωστό διαδίκτυο Internet.

- ❖ **Μητροπολιτικό Δίκτυο (Metropolitan Area Network - MAN)**, που συνδέει Η/Υ που βρίσκονται μέσα σε μια ευρεία περιοχή όπως μία πόλη, νομός κτλ.
- ❖ **Διαδίκτυο** είναι το δίκτυο των δικτύων και είναι συνδεδεμένοι παρά πολλοί υπολογιστές και επιμέρους δίκτυα, τα οποία μπορούν να χρησιμοποιούν διαφορετική τεχνολογία και λειτουργικά συστήματα.

## 6.2 Εξυπηρετητές

Οι **εξυπηρετητές (servers)** που χρησιμοποιούνται στο Internet είναι διαφόρων κατηγοριών. Ο κάθε ένας από αυτούς προσφέρει στους χρήστες διαφορετικές υπηρεσίες. Οι εξυπηρετητές αποτελούν τον απαραίτητο εξοπλισμό σχεδόν όλων των παροχέων. Μερικοί από αυτούς χρησιμοποιούνται και από τους κόμβους Internet. Ο πίνακας που ακολουθεί περιέχει τις κυριότερες κατηγορίες.

	<b>Εξυπηρετητές Internet</b>
<b>Κατηγορία</b>	<b>Λειτουργία</b>
<b>Web server</b>	Παρέχει πρόσβαση στο WWW.
<b>Mail server</b>	Υποστηρίζει το ηλεκτρονικό ταχυδρομείο.
<b>FTP server</b>	Επιτρέπει τη μεταφορά αρχείων.
<b>CHAT server</b>	Παρέχει υποστήριξη συνομιλιών.

<b>NEWS server</b>	<b>Υποστηρίζει τις ομάδες ειδήσεων.</b>
<b>PROXYserver seserver86ΓΥ6Γ</b>	<b>Υποστηρίζει την ενδιάμεση μεταφορά ιστοσελίδων.</b>

Πίνακας

**Web server.** Πρόκειται για τον κυριότερο εξυπηρετητή και υπάρχει σε όλους τους παροχείς και στην πλειοψηφία των κόμβων. Στον εξυπηρετητή αυτό τοποθετούνται οι ιστοσελίδες (**web pages**) μέσω των οποίων παρέχονται τα δεδομένα προς τους συνδρομητές με τη χρήση των **φυλλομετρητών (browsers)**.

**Mail server.** Είναι ο εξυπηρετητής που υποστηρίζει την υπηρεσία του ηλεκτρονικού ταχυδρομείου. Τον διαθέτουν απαραίτητα όλοι οι παροχείς Internet.

**FTP server.** Ο εξυπηρετητής αυτός περιέχει αρχεία που είναι διαθέσιμα προς τους χρήστες του Internet. Τον διαθέτουν όλοι οι παροχείς, αλλά και αρκετοί κόμβοι.

**CHAT server.** Είναι ο εξυπηρετητής που υποστηρίζει τις συνομιλίες ανάμεσα στους χρήστες του Internet. Αυτό το είδος εξυπηρετητή το διαθέτουν αρκετοί παροχείς αλλά και κόμβοι.

**NEWS server.** Είναι ο εξυπηρετητής που υποστηρίζει την υπηρεσία των ειδησεογραφικών ομάδων και διατίθεται συνήθως μόνο από τους παροχείς.

**PROXY server.** Τέτοιους εξυπηρετητές διαθέτουν σχεδόν όλοι οι παροχείς. Παίζουν τον ρόλο ενδιάμεσων σταθμών δεδομένων του Internet και κρατούν ιστοσελίδες που "κατεβάζουν" συχνά οι χρήστες. Έτσι, την επόμενη φορά, τα δεδομένα αυτά να μεταφέρονται προς το χρήστη από τον εξυπηρετητή και όχι από τη βασική τους πηγή, κερδίζοντας αρκετό χρόνο.

### 6.3 Πρωτόκολλα Επικοινωνίας Internet

Τα γνωστότερα πρωτόκολλα που χρησιμοποιούνται από τις υπηρεσίες του είναι τα παρακάτω:

#### † Πρωτόκολλο IP

Το **IP (Internet Protocol)** είναι το απαραίτητο πρωτόκολλο για το Internet. Σχεδιάστηκε και ορίστηκε ως πρωτόκολλο το 1981, για χρήση σε διασυνδεδεμένα συστήματα επικοινωνιακών δικτύων υπολογιστών.

#### † Πρωτόκολλο TCP

Το πρωτόκολλο **TCP (Transmission Control Protocol)** μαζί με το **IP** συνιστούν το βασικό πρωτόκολλο του Internet, το **TCP/IP**. Το πρωτόκολλο TCP αποτελεί πρότυπο, παρέχει μεγάλη αξιοπιστία και χρησιμοποιείται σε επικοινωνιακά δίκτυα υπολογιστών και σε διασυνδεδεμένα συστήματα παρόμοιων δικτύων. Το TCP παρέχει διεργασιακές επικοινωνιακές υπηρεσίες σε πολυδικτυακό περιβάλλον και προσφέρεται για ταυτόχρονη κοινή χρήση σε πολλαπλά δίκτυα.

#### ✚ Πρωτόκολλο HTTP

Το πρωτόκολλο **HTTP (Hyper Text Transport Protocol)** χρησιμοποιείται από τους φυλλομετρητές. Είναι ένα αντικειμενοστραφές πρωτόκολλο που υποστηρίζει υπερκείμενα. Ένα από τα χαρακτηριστικά του είναι η καταγραφή και η επεξεργασία παρουσίας δεδομένων, γεγονός που επιτρέπει στα συστήματα να δομούνται ανεξάρτητα από τα δεδομένα που μεταφέρονται.

#### ✚ Πρωτόκολλο POP

Σκοπός του πρωτοκόλλου **POP (Post Office Protocol)** είναι να επιτρέπει στο σταθμό εργασίας του χρήστη να λαμβάνει την αλληλογραφία του από έναν εξυπηρετητή ηλεκτρονικού ταχυδρομείου (**mail server**). Πρόκειται για το πρωτόκολλο εισερχόμενης αλληλογραφίας. Η αλληλογραφία ταχυδρομείται από το σταθμό εργασίας προς τον εξυπηρετητή χρησιμοποιώντας το πρωτόκολλο **SMTP**.

#### ✚ Πρωτόκολλο SMTP

Το πρωτόκολλο **SMTP (Simple Mail Transfer Protocol)** χρησιμοποιείται για τη μετάδοση ηλεκτρονικού ταχυδρομείου των χρηστών μέσω των εξυπηρετητών ταχυδρομείου (**mail servers**) των παροχέων. Πρόκειται για το πρωτόκολλο εξερχόμενης αλληλογραφίας.

#### ✚ Πρωτόκολλο FTP

Το πρωτόκολλο **FTP (File Transfer Protocol)** χρησιμεύει για τη μεταφορά αρχείων μεταξύ των υπολογιστών που είναι συνδεδεμένοι στο Internet. Παρέχει έλεγχο πρόσβασης και διαπραγμάτευση των παραμέτρων των αρχείων.

#### ✚ Πρωτόκολλο TELNET

Το πρωτόκολλο **TELNET** χρησιμοποιείται για πρόσβαση σε απομακρυσμένους υπολογιστές. Ο σκοπός αυτού του πρωτοκόλλου είναι να παρέχει μια διπλής κατεύθυνσης επικοινωνιακή διευκόλυνση τερματικών συσκευών και τερματικών διεργασιών μεταξύ τους.

## 6.4 Σημαντικές υπηρεσίες του Internet

- ✚ *Αναζήτηση πληροφοριών – (search engines)*
- ✚ *Εκτέλεση προγραμμάτων σε άλλους υπολογιστές – (telnet)*
- ✚ *Ηλεκτρονικό ταχυδρομείο – (e-mail)*
- ✚ *Μεταφορά αρχείων – (ftp)*
- ✚ *Ομάδες συζητήσεων – (newsgroups, mailing lists, chatlines)*
- ✚ *Παγκόσμιος ιστός – περιήγηση (surfing) σε πολυμορφικές πληροφορίες*
- ✚ *Συνομιλία με άλλους χρήστες – (Talk, irc, videoconferencing)*

#### 6.4.1 Υπηρεσία του Παγκόσμιου Ιστού (WWW)

Ο Παγκόσμιος Ιστός ή *World Wide Web (WWW)* είναι η πιο δημοφιλής υπηρεσία του δικτύου, λόγω της εύκολης χρήσης της και του πλήθους της πληροφορίας που παρέχει. Η πληροφορία παρουσιάζεται στο χρήστη με τη μορφή ιστοσελίδων (*web pages*), οι οποίες προσφέρονται από έναν εξυπηρετητή (*web server*) και γίνονται ορατές μέσα από ένα πρόγραμμα φυλλομετρητή (*web browser*) του πελάτη (*client*), όπως ο *Internet Explorer* της Microsoft ή ο *Navigator*, της εταιρείας Netscape.

Επιτρέπει την πρόσβαση στη μεγαλύτερη συλλογή εγγράφων τα οποία είναι αποθηκευμένα σε εκατομμύρια υπολογιστές στον κόσμο. Μας επιτρέπει από τον προσωπικό μας υπολογιστή να έχουμε πρόσβαση σε διαφορετικά είδη αρχείων, ανεξάρτητα από το πού βρίσκονται στο δίκτυο.

Το WWW αναπτύχθηκε αρχικά στο CERN (Ευρωπαϊκό Κέντρο Πυρηνικής Έρευνας) της Γενεύης το 1989 από την ομάδα του Tim Berners – Lee και σήμερα έχει καθιερωθεί ως η κεντρική υπηρεσία του Internet, αφού μέσω αυτής παρέχεται στον απλό χρήστη η δυνατότητα αναζήτησης, εντοπισμού, περιήγησης και μεταφοράς των δεδομένων του Internet με ιδιαίτερα φιλικό τρόπο. Επιτρέπει την εμφάνιση των δεδομένων σε μορφή κειμένου, εικόνας, κινούμενης εικόνας, ήχου και βίντεο. Οι πληροφορίες στο WWW είναι οργανωμένες σε μορφή υπερκειμένου (*hypertext*) Ιστοσελίδα. Η ιστοσελίδα (*Web page*) είναι ένα έγγραφο του ιστού που περιλαμβάνει πληροφορίες σε πολλές μορφές: κείμενο, εικόνα, κινούμενη εικόνα, ήχος, Video.

#### 6.4.2 Υπηρεσία του Ηλεκτρονικού ταχυδρομείου

Με τον όρο *ηλεκτρονικό ταχυδρομείο (e-mail)*, εννοούμε την αποστολή και τη λήψη μηνυμάτων σε ηλεκτρονική μορφή μέσω ηλεκτρονικού υπολογιστή. Η διαδικασία σύνταξης, αποστολής και λήψης ενός μηνύματος ηλεκτρονικού ταχυδρομείου γίνεται σε ψηφιακή μορφή.

Το ηλεκτρονικό ταχυδρομείο λειτουργεί ασύγχρονα. Επιτρέπει στους χρήστες να στέλνουν και να λαμβάνουν ηλεκτρονικά μηνύματα χωρίς να είναι συνδεδεμένοι ταυτόχρονα στο δίκτυο. Τα μηνύματα αποθηκεύονται σε μία ηλεκτρονική θυρίδα ενός διακομιστή (*server*), που είναι μόνιμα συνδεδεμένος στο δίκτυο. Στη συνέχεια, με τη βοήθεια μιας ειδικής εφαρμογής, ο παραλήπτης πρέπει να συνδεθεί στην ηλεκτρονική θυρίδα του για να παραλάβει τα εισερχόμενα μηνύματα.

### 6.4.3 Υπηρεσία Μεταφοράς αρχείων

Το FTP είναι το εξειδικευμένο πρωτόκολλο μεταφοράς αρχείων στα σύγχρονα δίκτυα δεδομένων όπως το Διαδίκτυο. Τα αρχεία στο Internet μπορούν να διακινούνται από τον ένα υπολογιστή στον άλλο, επικοινωνώντας με μια κοινή γλώσσα (*πρωτόκολλο*) που ονομάζεται File Transfer Protocol (FTP). Στο Διαδίκτυο υπάρχει πλήθος από τοποθεσίες FTP (FTP sites) από τα οποία μπορεί κάποιος να «κατεβάσει» (Download) αρχεία, δηλαδή, να τα μεταφέρει από τον απομακρυσμένο υπολογιστή στον υπολογιστή του ή να «ανεβάσει» αρχεία, δηλαδή να τα στείλει στον απομακρυσμένο υπολογιστή.

Το FTP ήταν ο βασικός τρόπος μεταφοράς αρχείων, αλλά σήμερα χρησιμοποιείται ευρέως και ο παγκόσμιος ιστός. Υπάρχουν πολλές τέτοιες μεγάλες «αποθήκες αρχείων» και έχουν σχεδιαστεί, έτσι ώστε να μπορούν οι χρήστες να αναζητούν και να βρίσκουν τα προγράμματα που τους ενδιαφέρουν. Για παράδειγμα μπορούμε να βρούμε έναν screen saver, ένα νεότερο οδηγό (driver) για μια συσκευή μας ή ένα πρόγραμμα ελέγχου ιών. Αυτές οι «αποθήκες» αρχείων ονομάζονται FTP sites. Όταν δεν χρειάζεται ο χρήστης να δηλώσει τα στοιχεία του, για να έχει πρόσβαση σ' αυτές ονομάζονται ανώνυμες (anonymous). Υπάρχουν εκατοντάδες anonymous FTP sites στον κόσμο που προσφέρουν έναν πλούτο πληροφοριών και προγραμμάτων.

### 6.4.4 Υπηρεσία telnet

Όπως αναφέραμε ήδη μπορούμε μέσω του Internet να εκτελέσουμε προγράμματα σε άλλους υπολογιστές. Η διαδικασία αυτή έχει ως εξής: Κάνουμε κλικ στο μενού *Έναρξη* ► *Εκτέλεση*. Στο παράθυρο διάλογου *Εκτέλεση*, στο πεδίο *Άνοιγμα*., πληκτρολογούμε *telnet* και πατάμε το κουμπί *OK*. Στο παράθυρο διαλόγου που ακολουθεί επιλέγουμε το μενού *Σύνδεση Απομακρυσμένο σύστημα....*. Στο πλαίσιο *Όνομα κεντρικού υπολογιστή*, πληκτρολογούμε ή επιλέγουμε το όνομα του απομακρυσμένου συστήματος με το οποίο θέλουμε να συνδεθούμε. Κατόπιν κάνουμε κλικ στο κουμπί *Σύνδεση*.

Το **Telnet** είναι η υπηρεσία του Internet που μας επιτρέπει να συνδεόμαστε με έναν απομακρυσμένο υπολογιστή και να δουλεύουμε αλληλεπιδραστικά στον υπολογιστή αυτόν χρησιμοποιώντας τα προγράμματά του σαν να είμαστε άμεσα συνδεδεμένοι μαζί του. Ο υπολογιστής μας, μετατρέπεται σε *τερματικό του απομακρυσμένου υπολογιστή*, ο οποίος ανταποκρίνεται στις εντολές μας. Μέσω του Telnet, μπορούμε να συνδεόμαστε με υπολογιστές σε ολόκληρο τον κόσμο και να εκμεταλλευόμαστε την ισχύ τους και τις υπηρεσίες που μας προσφέρουν. Έτσι μπορούμε να χρησιμοποιούμε απομακρυσμένες, για παράδειγμα,

βάσεις δεδομένων και άλλες πηγές πληροφόρησης, για να αναζητήσουμε πληροφορίες σε βιβλιογραφικούς καταλόγους διαφόρων βιβλιοθηκών. Για παράδειγμα, μπορούμε να συνδεθούμε με τη βιβλιοθήκη του Παντείου Πανεπιστημίου στην διεύθυνση [www.panteion.gr](http://www.panteion.gr) και ακολουθώντας τις οδηγίες να μπούμε στη βιβλιοθήκη του. Μπορούμε ακόμη να συνδεθούμε με έναν υπερυπολογιστή (*supercomputer*) και να χρησιμοποιήσουμε την ισχύ του για την εκτέλεση πολύπλοκων αλγορίθμων. Όταν συνδεθούμε με τον απομακρυσμένο υπολογιστή, μας ζητείται *όνομα χρήστη (login name)* και *συνθηματικό (password)*. Επομένως, θα πρέπει να έχουμε λογαριασμό (δηλ. δικαίωμα πρόσβασης) στον υπολογιστή αυτό.

Μερικές φορές, για υπηρεσίες που διατίθενται δημόσια, μας υποδεικνύεται από τον απομακρυσμένο υπολογιστή κάποιο ειδικό login name (π.χ. guest), ώστε να μπορέσουμε να συνδεθούμε ακόμη κι αν δε διαθέτουμε λογαριασμό.

#### 6.4.5 Υπηρεσία συνομιλιών με άλλους χρήστες

Η *συνομιλία IRC (Internet Relay Chat)* είναι ένα μέσο γραπτής επικοινωνίας (σε πραγματικό χρόνο) με ανθρώπους από όλο τον κόσμο. Αποτελείται από διάφορα ξεχωριστά δίκτυα από *IRC servers*, μηχανήματα, δηλαδή, τα οποία χρησιμοποιούν οι χρήστες, για να συνδεθούν στο IRC.

Η *υπηρεσία τηλεδιάσκεψης (videoconference)*, επιτρέπει στους χρήστες, διαχειριζόμενοι κατάλληλο υλικό και λογισμικό, να έχουν τη δυνατότητα να συνομιλούν και να ανταλλάσσουν δεδομένα κειμένου, φωνής και εικόνας σε πραγματικό χρόνο (*real time*). Η ποιότητα αυτών των υπηρεσιών αρχικά δεν ήταν αυτή που θα επέτρεπε ευρεία χρήση τέτοιου είδους εφαρμογών. Όμως η αύξηση του εύρους των δικτύων και η αύξηση της ταχύτητας πρόσβασης των χρηστών δίνουν πλέον τη δυνατότητα να χρησιμοποιούμε

εφαρμογές *τηλεδιάσκεψης και τηλεκπαίδευσης*, οι οποίες απαιτούν οπτική και ηχητική επικοινωνία πραγματικού χρόνου μεταξύ των ατόμων που συμμετέχουν.

#### 6.4.6 Υπηρεσία συζητήσεων

Η υπηρεσία *συζητήσεων (Usenet news ή Newsgroups)* δίνει τη δυνατότητα σε ανθρώπους από όλο τον κόσμο, να συμμετέχουν σε ανοιχτές συζητήσεις πάνω σε θέματα που τους ενδιαφέρουν. Οι συζητήσεις αυτές πραγματοποιούνται σε χώρους, που λειτουργούν σαν πίνακες ανακοινώσεων. Κάθε χρήστης μπορεί να στείλει το μήνυμά του (άρθρο) και οι άλλοι χρήστες μπορούν να διαβάσουν το άρθρο του και, αν επιθυμούν, να απαντήσουν σε αυτό. Οι απαντήσεις στέλνονται και αυτές στον ίδιο χώρο, ώστε να μπορούν και αυτές με τη σειρά τους να διαβαστούν από όλους τους υπόλοιπους χρήστες.

Οι *λίστες ηλεκτρονικού ταχυδρομείου (mailing lists)* είναι ένας αυτοματοποιημένος μηχανισμός που διανέμει ένα μήνυμα (mail) σε πολλούς χρήστες ταυτόχρονα, χωρίς ο αποστολέας να γράφει τις διευθύνσεις των παραληπτών. Δηλαδή, ο χρήστης στέλνει το email σε μια «εικονική» διεύθυνση (π.χ. [gr-](mailto:gr-...)

[schools@pi-schools.gr](mailto:schools@pi-schools.gr)) και αυτό καταλήγει σε όλους τους εγγεγραμμένους στη συγκεκριμένη λίστα. Σε μια τέτοια λίστα, όπως είναι φυσικό, υπάρχουν άτομα με κοινά ενδιαφέροντα που επιθυμούν να ανταλλάσσουν e-mail για θέματα που τους απασχολούν.

#### 6.4.7 Υπηρεσία Αναζήτησης πληροφοριών

Η μηχανή αναζήτησης είναι μια υπηρεσία που διαθέτει μια βάση δεδομένων με καταγεγραμμένα στοιχεία για τις πληροφορίες που υπάρχουν στο Internet. Ο χρήστης αναζητεί αυτό που θέλει με βάση κάποια συγκεκριμένα κριτήρια - λέξεις κλειδιά (*keywords*) και η μηχανή αναζήτησης του παρουσιάζει τις διευθύνσεις εκείνες στις οποίες έχουν βρεθεί οι λέξεις κλειδιά. Μια πολύ διαδεδομένη μηχανή αναζήτησης είναι το [www.google.com](http://www.google.com), το [www.yahoo.com](http://www.yahoo.com) και άλλα.

#### 6.5 Internet και λειτουργία του

Το Internet λειτουργεί με το μοντέλο πελάτη – εξυπηρετητή (*client-server*), όπου ο πελάτης πρέπει να αντιλαμβάνεται τα πρωτόκολλα, δηλαδή τις συμφωνημένες παραδοχές επικοινωνίας, των υπηρεσιών που ζητά. Το πρόγραμμα πελάτη (*client*) στο δικό μας υπολογιστή είναι υπεύθυνο για τη μεταφορά των πληροφοριών στον υπολογιστή μας και την παρουσίασή τους με ένα γνωστό και προκαθορισμένο τρόπο. Έτσι απελευθερώνεται ο εξυπηρετητής (*server*) από αυτό το φορτίο και αφιερώνει περισσότερα μέσα για την επεξεργασία των πληροφοριών (εργασία που μόνο αυτός μπορεί να κάνει).

Η ιδέα του διαδικτύου είναι πολύ απλή. Ένας υπολογιστής που ονομάζεται πελάτης (*client*) παίρνει υπηρεσίες και πληροφορίες από έναν άλλο απομακρυσμένο υπολογιστή που ονομάζεται εξυπηρετητής (*server*). Στην καθημερινή μας ζωή, πρωτόκολλο είναι ένα σύνολο από συμβάσεις που καθορίζουν το πώς πρέπει να πραγματοποιηθεί κάποια διαδικασία. Σε ένα δίκτυο, πρωτόκολλο είναι ένα σύνολο από συμβάσεις που καθορίζουν πώς οι υπολογιστές του δικτύου ανταλλάσσουν μεταξύ τους δεδομένα, πώς γίνεται ο έλεγχος και ο χειρισμός λαθών. Το διαδίκτυο χρειάζεται ένα σύνολο από συμβάσεις, οι οποίες να καθορίζουν το πώς ανταλλάσσουν μεταξύ τους οι υπολογιστές δεδομένα, που μπορεί να είναι διαφορετικού τύπου και να ανήκουν σε διαφορετικής τεχνολογίας δίκτυα.

Το TCP/IP, είναι το πρωτόκολλο που επιτρέπει στους υπολογιστές που είναι συνδεδεμένοι στα χιλιάδες μικρότερα δίκτυα του Internet να μιλούν μια κοινή γλώσσα, για να συνεννοούνται παρά τις όποιες διαφορές τους. Το Internet χρησιμοποιεί την τεχνολογία μεταγωγής πακέτων για τη μεταφορά πληροφοριών. Τα

δεδομένα κόβονται σε κομμάτια που ονομάζονται πακέτα. Σε κάθε πακέτο μπαίνει μια «επικεφαλίδα» με τις διευθύνσεις του υπολογιστή - αποστολέα και του υπολογιστή - παραλήπτη, για παράδειγμα, 128.17.42.96 255.8.12.45 ΚΑΛΗΜΕΡΑ  
(Από: Προς: Δεδομένα:)

Κάθε πακέτο δεδομένων αριθμείται. Ο υπολογιστής - παραλήπτης και ο υπολογιστής - αποστολέας, αλλά όχι οι ενδιάμεσοι υπολογιστές, παρακολουθούν τους αριθμούς των πακέτων και ανταλλάσσουν μεταξύ τους πληροφορίες. Ο παραλήπτης λαμβάνει το πρώτο πακέτο, το δεύτερο, το τρίτο κτλ. Σε περίπτωση που παρουσιαστεί κάποιο πρόβλημα στο δίκτυο, είτε χαθεί κάποιο πακέτο κατά τη διάρκεια της μετάδοσης, το ξαναζητάει, και ο αποστολέας είναι υπεύθυνος για την αναμετάδοση του. Ο παραλήπτης ελέγχει, επίσης, αν το περιεχόμενο των πακέτων φτάνει σωστά. Η διαδρομή που ακολουθεί ένα πακέτο μέσα από το «σύννεφο» των συνδέσεων δεν είναι προκαθορισμένη.

Τα δίκτυα στο Internet συνδέονται μεταξύ τους με ειδικούς υπολογιστές που ονομάζονται *δρομολογητές (routers)* ή *πύλες (gateways)*. Ο δρομολογητής είναι λοιπόν ένας υπολογιστής που συνδέει δύο ή περισσότερα δίκτυα (ίδιου ή διαφορετικού τύπου). Η δουλειά τους είναι να δρομολογούν τα πακέτα των δεδομένων μέσα από τα διάφορα δίκτυα που αποτελούν το Internet, μέχρις ότου τα επιδώσουν στον προορισμό τους. Ο δρομολογητής ελέγχει την επικεφαλίδα του πακέτου και, αν ο παραλήπτης βρίσκεται στο ίδιο δίκτυο με τον αποστολέα, στέλνει κατευθείαν το πακέτο στον παραλήπτη, χωρίς να χρειαστεί να διαβεί τα όρια του δικτύου. Διαφορετικά, το προωθεί στον επόμενο δρομολογητή, που είναι συνδεδεμένος με το δίκτυο κ.ο.κ, μέχρις ότου το πακέτο προωθηθεί τελικά στο δρομολογητή που είναι συνδεδεμένος στο ίδιο δίκτυο με τον παραλήπτη.

Οι δρομολογητές διατηρούν πίνακες από τους οποίους προσδιορίζουν την κατεύθυνση που πρέπει να πάρει ένα πακέτο, προκειμένου να φτάσει στον προορισμό του. Κάθε φορά, το πακέτο μετακινείται όλο και πιο κοντά προς τον προορισμό του, έως ότου, τελικά, τον φτάσει. Ανάλογα με την κίνηση, ή σε περίπτωση που ένα τμήμα του δικτύου παρουσιάζει πρόβλημα και βρίσκεται προσωρινά σε αχρηστία, οι δρομολογητές επιλέγουν εναλλακτικούς δρόμους.

Κάθε υπολογιστής, κάθε τόπος, κάθε σελίδα με πληροφορίες, κάθε χρήστης που συνδέεται στο Internet έχει τη δική του διεύθυνση. Το σύστημα διευθύνσεων του Internet επιτρέπει σε κάποιον υπολογιστή να συνδέεται με κάποιον άλλον, λόγω του ότι κάθε υπολογιστής έχει το δικό του μοναδικό όνομα και διεύθυνση, όπως κάτι ανάλογο γίνεται και στο διεθνές τηλεφωνικό δίκτυο, όπου κάθε χρήστης έχει ένα μοναδικό αριθμό τηλεφώνου.

Η διεύθυνση ενός υπολογιστή στο Internet αναφέρεται και ως IP διεύθυνση (IP Address) και αποτελεί την «ταυτότητα» του στο διαδίκτυο. Κάθε διεύθυνση αποτελείται από τέσσερα νούμερα (αριθμός δικτύου και αριθμός υπολογιστή μέσα στο συγκεκριμένο δίκτυο). Εκτός της αριθμητικής διεύθυνσης μπορεί να προσδιορίζεται και με ένα όνομα. Τα ονόματα αυτά αποδίδονται με κάποιες συμβάσεις μέσω του συστήματος *DNS (Domain Name System)*.

Για παράδειγμα στην *IP Address 147.52.16.2* αντιστοιχεί το όνομα *crete.csd.uch.gr*.

Έχουμε, λοιπόν, ως δεδομένο ότι: Κάθε υπολογιστής είναι μοναδικός στο Internet.

Κάθε οργανισμός που θέλει να συνδέσει στο Internet τους υπολογιστές του, ζητά έναν αριθμό δικτύου από κάποιον επίσημο οργανισμό, που ασχολείται με την κατανομή των διευθύνσεων στο Internet, έτσι ώστε να εξασφαλίζεται η μοναδικότητα τους.



Σε έναν υπολογιστή του Internet, μπορούν να έχουν πρόσβαση περισσότεροι από ένας χρήστες. Κάθε χρήστης διαθέτει ένα λογαριασμό (account) που περιλαμβάνει το όνομα χρήστη και τον κωδικό πρόσβασης. Δηλαδή, κάθε χρήστης έχει τη δική του ταυτότητα, το δικό του όνομα χρήστη (user ID), το οποίο του επιτρέπει να ξεχωρίζει από τους άλλους χρήστες που χρησιμοποιούν τον ίδιο υπολογιστή. Για να πάρει άδεια να συνδεθεί, πρέπει να επιβεβαιώσει την ταυτότητα του, παρέχοντας στο σύστημα το προσωπικό του μυστικό κωδικό (password).

Συνδυάζοντας το γεγονός ότι κάθε υπολογιστής είναι μοναδικός στο Internet και κάθε χρήστης μοναδικός στον υπολογιστή που χρησιμοποιεί, καταλήγουμε στο ότι:

**Κάθε χρήστης είναι μοναδικός σε όλο το Internet!**

---

Η ηλεκτρονική ταχυδρομική διεύθυνση (το ηλεκτρονικό ταχυδρομικό κουτί) κάθε χρήστη προκύπτει από το όνομα που φέρει ο χρήστης στο σύστημά του, το σύμβολο "@" που ονομάζεται *at* (στην Ελλάδα μερικοί το αναφέρουν και με τον όρο *παπάκι*) και το όνομα του υπολογιστή που του εξασφαλίζει την πρόσβαση στο Internet ή εξυπηρετεί την αλληλογραφία του.

### 6.5.1 Εταιρείες Παροχής Υπηρεσιών Internet (ISP)

Σε κάθε χώρα υπάρχουν εταιρείες παροχής υπηρεσιών Internet (Internet Service Providers). Πρόκειται για εταιρείες που έχουν νοικιάσει τηλεπικοινωνιακές γραμμές από τηλεπικοινωνιακούς οργανισμούς και δημιουργούν έτσι ένα WAN δίκτυο το οποίο συνδέεται στο Internet. Οι γραμμές αυτές ονομάζονται «ραχοκοκαλιές» (*backbones*). Οι ταχύτερες μετάδοσης δεδομένων σε αυτές τις γραμμές είναι αρκετά μεγάλες (2 Mbps και άνω).

Οι εταιρείες αυτές διαθέτουν κόμβους σε διάφορα σημεία της χώρας, συνήθως στις μεγάλες πόλεις. Κόμβοι είναι τα σημεία εκείνα που είναι συνδεδεμένα στον κύριο κορμό (backbone) του δικτύου. Οι μεγάλες εταιρείες διαθέτουν στους πελάτες τους ΕΠΑΚ (Ενιαίο Πανελλαδικό Αριθμό Κλήσης) ή ΠΕΑΚ (Περιοχικός Αριθμός Κλήσης) για την κλήση και την οικονομικότερη χρέωση (ορισμένες ώρες και μέρες) σε όλη την Ελλάδα.

Το ΕΔΕΤ είναι το εθνικό δίκτυο κορμού (backbone) για τα δίκτυα των ακαδημαϊκών και ερευνητικών ιδρυμάτων της Ελλάδας.

Το DANTE είναι ένα πανευρωπαϊκό δίκτυο κορμού, πάνω στο οποίο συνδέονται τα εθνικά δίκτυα κορμού των ακαδημαϊκών και ερευνητικών ιδρυμάτων των ευρωπαϊκών χωρών. Μέρος του DANTE αποτελεί και το δίκτυο TEN-34 που διασυνδέει ευρωπαϊκά ακαδημαϊκά και ερευνητικά δίκτυα προσφέροντας υψηλές ταχύτητες μετάδοσης. Το DANTE παρέχει επίσης και διεθνείς συνδέσεις με τις υπόλοιπες ηπείρους.

Το ΕΔΕΤ συνδέεται επίσης με τα εθνικά δίκτυα κορμού των μεγαλύτερων εμπορικών παροχών υπηρεσιών Internet που υπάρχουν στην Ελλάδα (OTENET, FORTHNET, HOL, κ.α.) μέσω του κόμβου AIX (Athens Internet Exchange).

Οι εταιρίες αυτές διατηρούν και δικές τους απευθείας συνδέσεις με το εξωτερικό.

### 6.5.2 Φυλλομετρητής

Ο *φυλλομετρητής του ιστού (Web Browser)* είναι ένα πρόγραμμα περιήγησης στο Internet και παρουσίασης ηλεκτρονικών πολυμορφικών σελίδων (ιστοσελίδων). Τα προγράμματα που έχουν κυριαρχήσει στο μεγαλύτερο μέρος της αγοράς είναι: Ο Communicator ή Navigator της εταιρείας Netscape που κυριάρχησε στην αγορά τα πρώτα χρόνια του Internet και ο Internet Explorer της εταιρείας Microsoft που υπάρχει ενσωματωμένος στις τελευταίες εκδόσεις των Windows. Ωστόσο, αυτό το γεγονός πυροδότησε μια μεγάλη δικαστική αντιμονοπωλιακή διαμάχη στις ΗΠΑ.

### 6.5.3 Περιήγηση στο Web

Μερικές από τις βασικές δυνατότητες που προσφέρουν σήμερα τα προγράμματα περιήγησης στο Internet είναι και οι εξής:

Πύλη (portal) εισόδου στο Web είναι μια σελίδα που έχει σχεδιαστεί για να ξεκινάμε την περιήγηση στο Διαδίκτυο. Συνήθως περιέχει ειδήσεις και θεματικούς καταλόγους με διευθύνσεις άλλων σελίδων.

Όταν αρχίζουμε να πληκτρολογούμε στη γραμμή διευθύνσεων μια διεύθυνση Web που χρησιμοποιούμε συχνά, εμφανίζεται μια λίστα παρόμοιων διευθύνσεων από τις οποίες μπορούμε να επιλέξουμε.

Μπορούμε να αναζητήσουμε τοποθεσίες Web, κάνοντας κλικ στο κουμπί Αναζήτηση της γραμμής εργαλείων. Στη συνέχεια, στη γραμμή αναζήτησης, πληκτρολογούμε μια λέξη ή μια φράση που να περιγράφει αυτό που ζητάμε.

Όταν εμφανίζονται τα αποτελέσματα της αναζήτησής μας, μπορούμε να προβάσουμε κάθε ιστοσελίδα ξεχωριστά, χωρίς να χάσουμε τη λίστα με τα αποτελέσματα της αναζήτησης.

Μπορούμε να μεταβούμε σε ιστοσελίδες παρόμοιες με αυτή που έχει προβληθεί, χωρίς αναζήτηση.

Για να δούμε μια λίστα ιστοσελίδων που επισκεφθήκαμε πρόσφατα, μπορούμε να κάνουμε κλικ στο κουμπί Ιστορικό της γραμμής εργαλείων.

Για να εμφανισθεί μια ιστοσελίδα, αρκεί να έχουμε πληκτρολογήσει σωστά τη διεύθυνσή της.

Εάν γνωρίζουμε τη διεύθυνση μιας σελίδας, μπορούμε να την πληκτρολογήσουμε στο πεδίο Διεύθυνση: και να πατήσουμε το πλήκτρο Enter, για να εμφανισθεί το περιεχόμενό της.

## 6.6 Μηχανές Αναζήτησης

Οι *Μηχανές Αναζήτησης (Search Engines)* αποθηκεύουν πληροφορίες για εκατομμύρια σελίδες WEB σε μια τεράστια βάση δεδομένων. Από τα αρχεία που συγκεντρώνονται (με βάση τον τίτλο τους, το πλήρες κείμενο, το μέγεθος τους, τη διεύθυνσή τους κτλ.) δημιουργείται ένα ευρετήριο. Μπορούμε να κάνουμε αναζητήσεις σ' αυτές τις βάσεις δεδομένων, εισάγοντας λέξεις-κλειδιά (keywords).

Καποιες Μηχανές Αναζήτησης, είναι οι:

*www.yahoo.com*  
*www.altavista.com*  
*www.lycos.com*  
*www.google.com*

Μηχανή αναζήτησης είναι ένα λογισμικό που βρίσκει και ταξινομεί τα αποτελέσματα ανάλογα με το ποσοστό συνάφειας του περιεχομένου των ιστοσελίδων σε σχέση με τους όρους της έρευνας. Από τα αποτελέσματα μπορούμε να ανακαλούμε τα έγγραφα (τις σελίδες) που ικανοποιούν τις λέξεις κλειδιά που έχουμε εισάγει. Οι μηχανές αναζήτησης τελευταίας γενιάς επιπλέον, ομαδοποιούν τα αποτελέσματά τους σύμφωνα με το περιεχόμενο, τη δημοτικότητα και το είδος των τόπων, ενώ ορισμένες από αυτές μπορούν να δεχθούν ερωτήσεις και σε φυσική γλώσσα (π.χ. Αγγλικά).

**Μεταμηχανές** είναι μηχανές αναζήτησης οι οποίες ερευνούν τα ευρετήρια πολλών άλλων μηχανών ταυτόχρονα. Είναι ιδιαίτερα χρήσιμες όταν ψάχνουμε για ένα μοναδικό όρο ή φράση.

Δεν υπάρχει υπηρεσία ή μηχανή αναζήτησης που να είναι πλήρης. Γενικά, μια μηχανή αναζήτησης μπορεί να περιέχει διευθύνσεις από όλες τις υπηρεσίες του Internet, όπως FTP, World Wide Web, Usenet, Telnet κτλ.

Οι περισσότερες μηχανές αναζήτησης περιορίζονται στην «κατηγοριοποίηση σε καταλόγους» των πληροφοριών εκείνων, που μπορούν να προβληθούν μόνο μέσω του World Wide Web, δηλαδή με βάση το πρωτόκολλο HTTP κατά κύριο λόγο, ενώ ορισμένες υποστηρίζουν επιπλέον και FTP διευθύνσεις του δικτύου.

Τις περισσότερες φορές οι ιστοσελίδες που ανακτώνται ως αποτέλεσμα μιας αναζήτησης, είναι εν μέρει μόνο σχετικές με αυτό που ζητάμε. Πολλές φορές ακόμη στα αποτελέσματα περιλαμβάνονται πολλές ιστοσελίδες από τον ίδιο τόπο.

Παραδείγματα Ελληνικών Μηχανών Αναζήτησης – Ευρετηρίων - Πυλών

*www.forthnet.gr*  
*www.pathfinder.gr*

## 6.7 Περιγραφή συγκεκριμένων πακέτων

### 6.7.1 Word

#### Word

Το Word της Microsoft είναι ένα πακέτο επεξεργασίας κειμένου, χρησιμοποιείται για τη δημιουργία, διόρθωση, εκτύπωση, αποθήκευση, ανάκτηση και γενικά χειρισμό κειμένου, με τρόπο εύκολο και απαλλαγμένο από όλα τα μειονεκτήματα της κλασικής γραφομηχανής.

Η συγγραφή με τη βοήθεια ενός πακέτου του είδους είναι ιδιαίτερα απλή: με την εκκίνηση τους, εισάγεται ο χρήστης σε μια κενή οθόνη, την οποία μπορεί να θεωρήσει ως το χαρτί της γραφομηχανής και στην οποία μπορεί να αρχίσει αμέσως να γράφει. Η θέση που βρίσκεται στο 'ηλεκτρονικό χαρτί', φαίνεται από το δρομέα που κινείται, καθώς πληκτρολογεί ο χρήστης. Σε κάποιο άκρο της οθόνης, αναγράφεται η θέση του δρομέα καθώς και κάποιες συμπληρωματικές πληροφορίες.

Ο χρήστης καθορίζει το μήκος της γραμμής, έτσι η μεταπήδηση στην επομένη γραμμή γίνεται αυτόματα. Φυσικά και εδώ μπορούν να οριστούν στηλογνώμονες. Αν, ο χρήστης θέλει να δώσει έμφαση σε μια λέξη ή περιοχή του κειμένου του, μπορεί πολύ εύκολα να την ορίσει έτσι, ώστε να εκτυπωθεί με έντονα γράμματα ή πλάγια ή υπογραμμισμένη ή και σε συνδυασμό των πιο πάνω. Επιπλέον, μπορεί να χρησιμοποιήσει διαφορετικές γραμματοσειρές και σε διάφορα μεγέθη.

Τυχόν λάθη που θα γίνουν κατά τη συγγραφή, δεν απαιτούν γράψιμο από την αρχή. Το μόνο που έχει να κάνει ο χρήστης, είναι να τοποθετήσει το δρομέα πάνω ή δίπλα στο λάθος, να το σβήσει με το πλήκτρο Del ή Backspace και να γράψει το σωστό. Αν χρειαστεί να προσθέσει ένα γράμμα στο μέσον μιας λέξης, μπορεί να πάει στο επόμενο και απλά να το γράψει. Τα δεξιότερα από αυτό θα μεταφερθούν αυτόματα μια θέση δεξιά.

Με το πακέτο αυτό μπορούν να γίνουν ομαδικοί χειρισμοί σε ένα έγγραφο. Αν για παράδειγμα μια πρόταση ή παράγραφος (ή και λέξη) επαναλαμβάνεται, δεν χρειάζεται να γραφτεί πολλές φορές. Αρκεί μια και ένας απλούστατος χειρισμός για να αντιγραφεί στις επιθυμητές θέσεις.

Μια άλλη πολύ χρήσιμη και τυπικά διαθέσιμη δυνατότητα των επεξεργαστών κειμένου, είναι η αναζήτηση μιας λέξης μέσα στο κείμενο. Η αναζήτηση γίνεται με απλούς χειρισμούς. Όμοια με απλούς χειρισμούς μπορεί να γίνει αναζήτηση και αντικατάσταση με μια άλλη. Θα πρέπει να σημειωθεί ότι, σχεδόν όλα τα πακέτα επεξεργασίας κειμένου διαθέτουν πλέον και δυνατότητα ορθογραφικού ελέγχου των κειμένων.

### 6.7.2 Excel

#### Excel

Το Excel είναι ένα πακέτο φύλλων υπολογισμών, παρέχει στο χρήστη τη δυνατότητα ανάλυσης και πολύπλοκων υπολογισμών αριθμητικών δεδομένων, υπό την προϋπόθεση ότι αυτά μπορούν να οργανωθούν σε μορφή πίνακα. Τα σημερινά πακέτα περιέχουν επιπλέον τη δυνατότητα γραφικής παράστασης των δεδομένων που περιέχουν.

Οπτικά ένα φύλλο υπολογισμών, είναι μια απλή μεταφορά των κλασικών λογισμικών φύλλων που περιέχουν τεμνόμενες γραμμές και στήλες στην οθόνη του

υπολογιστή, δηλαδή στις γραμμές του φύλλου με αριθμούς, ενώ στις στήλες με γράμματα της λατινικής αλφαβήτου. Οι τομές των γραμμών με τις στήλες σχηματίζουν παραλληλόγραμμα που λέγονται κελιά.

Κάθε κελί παίρνει την ονομασία του από τη γραμμή και τη στήλη που το σχηματίζουν. Για παράδειγμα, το κελί που σχηματίζεται από την τομή της γραμμής 1 με τη στήλη A ονομάζεται A1 κ.λ.π. σε κάθε κελί μπορούν να τεθούν αριθμητικά δεδομένα, ημερομηνίες ή ακόμα και σχέσεις.

Για παράδειγμα, αν θέλουμε στο κελί A5 να εμφανίζεται το άθροισμα των κελιών A1, A2, A3, τοποθετούμε ως περιεχόμενο την παράσταση:  $=A1+A2+A3$

Το σημαντικό είναι, ότι οποιεσδήποτε αλλαγές στα κελιά που λαμβάνουν μέρος στους ορούς του αθροίσματος (A1, A2, A3), αντιστακτώνται αυτόματα και στο κελί A5. Με τον τρόπο αυτό, μπορεί κανείς να δοκιμάσει εύκολα και γρήγορα εναλλακτικές λύσεις, που φυσικά μπορεί να περιλαμβάνουν πολύ συνθετότερες εκφράσεις.

Τα πακέτα των φύλλων υπολογισμών έχουν εξελιχθεί σήμερα σε ισχυρά εργαλεία επίλυσης προβλημάτων. Με τη χρήση σημερινών πακέτων του είδους μπορεί κανείς να παραστήσει ένα μοντέλο του προβλήματος με βάση το οποίο να διατυπώσει ερωτήσεις του τύπου 'τι θα γίνει αν ελέγχοντας πρακτικά και άμεσα τι θα συμβεί σε αριθμούς, αν μεταβληθούν κάποιοι άλλοι. Θα πρέπει να σημειωθεί ότι ο όρος πρόβλημα εδώ χρησιμοποιείται με την ευρεία έννοια και αναφέρεται σε οποιοδήποτε τύπο πληροφορίας μπορεί να προσαρμοστεί σε ένα φύλλο, όπως οικονομικά μιας επιχείρησης ή ενός σπιτιού, ένα τιμολόγιο, προϋπολογισμός ενός έργου κ.λπ.

Τα σύγχρονα φύλλα υπολογισμών παρέχουν εκτεταμένες δυνατότητες αριθμητικών υπολογισμών, γραφικών και διαγραμμάτων και ανεπτυγμένα συστήματα μακροεντολών και τελευταία ενσωματωμένη γλώσσα προγραμματισμού π.χ. Visual Basic στο πακέτο Excel. Στα σημερινά φύλλα υπολογισμών έχει κανείς στη διάθεση του ισχυρές μεθόδους ανάλυσης, καθώς και μια βιβλιοθήκη από πολλές συναρτήσεις απλές αλλά και ιδιαίτερα πολύπλοκες. Τα φύλλα υπολογισμών συνδέονται εύκολα μεταξύ τους, ενώ περιλαμβάνουν πολλαπλές δυνατότητες μορφοποίησης του φύλλου και ανάπτυξης διαγραμμάτων.

### 6.7.3 Access

#### Access

Το Access είναι ένα πακέτο βάσης δεδομένων που επιτρέπει στο χρήστη του να αποθηκεύσει πληροφορίες με κάποια δομημένη μορφή, με τρόπο ώστε να μπορεί να τις ανάκτησή ή να τις χειριστεί (όλες ή ορισμένες) εύκολα και γρήγορα. Είναι μία συλλογή από εγγραφές και αρχεία, τα οποία είναι οργανωμένα με τέτοιο τρόπο ώστε να εξυπηρετούν ένα συγκεκριμένο σκοπό.

Το λογισμικό που χρησιμοποιείται για την οργάνωση και τη διαχείριση των δεδομένων μιας βάσης δεδομένων ονομάζεται *σύστημα διαχείρισης βάσεων δεδομένων* (Database Management System - DBMS). Τα συστήματα DBMS απαιτούν αρκετά μεγάλη υπολογιστική ισχύ, γι' αυτό αρχικά χρησιμοποιούνταν μόνο σε μεγάλα υπολογιστικά συστήματα (main-frames). Σήμερα, όμως, με την αύξηση των δυνατοτήτων των προσωπικών υπολογιστών, τα συστήματα διαχείρισης βάσεων δεδομένων μπορούν να εκτελούνται και να λειτουργούν ικανοποιητικά ακόμα και σε προσωπικούς υπολογιστές.

Η δουλειά με αυτό το πακέτο ξεκινάει κατά κανόνα με τον ορισμό της δομής του αρχείου ή των αρχείων τα οποία απαιτεί ο χρήστης. Όλο αυτό το πακέτο παρέχει κάποια ευκολία (λειτουργία) με την οποία γίνεται αυτό. Όταν δημιουργούμε ένα απλό

αρχείο πελάτη. Έστω ότι έχει αποφασιστεί να χρειάζεται ένας 6-ψήφιος κωδικός πελάτη, ο οποίος θα είναι αλφαριθμητικού τύπου. Η γραμμογράφηση περιλαμβάνει επίσης το ονοματεπώνυμο του πελάτη που δε θα ξεπερνάει τους 30 χαρακτήρες και θα είναι τύπου επίσης αλφαριθμητικού. Το ίδιο και για τα υπόλοιπα πεδία, εκτός από το υπόλοιπο που είναι αριθμητικό. Στη βασική αυτή φάση λοιπόν, ορίζεται το μήκος κάθε στοιχείου(πεδίου) των εγγράφων που πρόκειται να αποθηκευτούν, ανάλογα με τις προβλέψεις για τη μέγιστη τιμή του.

Ορίζεται επίσης ο τύπος κάθε πεδίου. Ο τύπος του πεδίου καθορίζει βασικά τις πράξεις που μπορούν να γίνουν σε αυτό. Για παράδειγμα δεν μπορούν να γίνουν αριθμητικές πράξεις (τουλάχιστον με την καθημερινή έννοια) σε πεδία που έχουν οριστεί ότι θα περιέχουν αλφαβητικούς χαρακτήρες. Σε δεύτερη φάση, εισάγονται στο αρχείο οι εγγραφές πεδίο-πεδίο. Αυτό μπορεί να γίνει άμεσα με ειδική λειτουργία που παρέχουν όλα τα γνωστά πακέτα του είδους. Εναλλακτικά μπορεί να γίνει με την κατασκευή μιας οθόνης εισαγωγής στοιχείων, σύμφωνα με τα προσωπικά γούστα του καθενός, υπόθεση γενικά εύκολη στα σύγχρονα πακέτα βάσεων δεδομένων.

Απ' τη στιγμή αυτή και έπειτα μπορεί κανείς να ανακτήσει όποτε το θελήσει, πληροφορίες από τη βάση του. Για παράδειγμα μπορεί πολύ εύκολα και με ταχύτητες παντελώς άγνωστες για συμβατικές μεθόδους, να βρεί όλους τους "ΠΑΠΑΔΟΠΟΥΛΟΣ" που υπάρχουν στο αρχείο του, ή όσους έχουν υπόλοιπο(χρωστούν) πάνω από ένα ποσό κ.λ.π.

Μπορεί μέσα σε ελάχιστο σχετικά χρόνο και με μικρό αριθμό απλών εντολών, να αλλάξει κάποια στοιχεία ενός πελάτη ή μιας ομάδας πελατών. Κατά κανόνα τα πακέτα βάσεων δεδομένων παρέχουν μια λειτουργία με την οποία μπορεί κανείς πολύ εύκολα να σχεδιάσει και να παράγει εκτυπώσεις όλων ή μέρους των εγγράφων της βάσης του ή να παράγει αυτοκόλλητα αλληλογραφίας, τυποποιημένων ή όχι διαστάσεων, απ' αυτήν.

#### 6.7.4 Power Point

##### Power Point

Το **PowerPoint** είναι ένα από τα πιο ευέλικτα και δυναμικά προγράμματα που κυκλοφορούν στην αγορά για την παρουσίαση δεδομένων. Είναι ένα πακέτο με το οποίο σχηματίζονται τα περιεχόμενα μιας παρουσίασης που θα αποτυπωθούν σε μια διαδοχική από στοιχείαφορείς παρουσίασης που τυπικά είναι slides ή διαφάνειες ή ακόμα και διαδοχικές οθόνες στον υπολογιστή.

Η διαδικασία γενικά αρχίζει με τη σύνταξη του πρώτου στοιχείου, στη συνέχεια προστίθεται με ειδική εντολή ένα δεύτερο κοκ μέχρι να εξαντληθούν τα στοιχεία που πρόκειται να παρουσιαστούν.

Βασικής σημασίας συνιστώσα για τον σχηματισμό κάθε στοιχείου είναι η γενική του διάταξη. Τυπικά σήμερα τα πακέτα παρουσιάσεων παρέχουν τη δυνατότητα έτοιμων γενικών διατάξεων από τις οποίες ο χρήστης διαλέγει όποια ταιριάζει καλύτερα στις ανάγκες του και στην οποία στη συνέχεια προσαρμόζει τα δεδομένα του.

Αφού ολοκληρωθεί η διαδικασία του σχηματισμού της όλης παρουσίασης, υπάρχει η δυνατότητα της διαδοχικής εμφάνισης των στοιχείων που την αποτελούν με βάση κάποιο χειρισμό π.χ. πίεση του ποντικιού. Επίσης, δίνεται η δυνατότητα να

τυπώσουμε τις διαφάνειες ή να τις εμφανίσουμε στην οθόνη και να οργανώσουμε το υλικό της παρουσίασης.

Ας σημειωθεί ότι οι επιτυχημένες παρουσιάσεις, δεν είναι κάτι απλό. Αποτελούν μια ολόκληρη διαδικασία με πολλά μυστικά, όπως για παράδειγμα επιλογές των κατάλληλων χρωμάτων, επιλογή του κατάλληλου τύπου διαγράμματος κ.λ.π.

### 6.7.5 Outlook

#### Outlook

Το Outlook εντάσσεται στη κατηγορία των προγραμμάτων που συμβάλουν στην οργάνωση των διάφορων εργασιών ενός γραφείου. Με τα πακέτα αυτά ο χρήστης μπορεί να χειριστεί το ημερολόγιο του, δηλαδή να καταγράψει τα ραντεβού του για μια ή περισσότερες ημέρες και να καθορίζει υπενθυμίσεις γι' αυτά. Επίσης να διαχειρίζεται την ηλεκτρονική αλληλογραφία με συνεργάτες στην ίδια επιχείρηση για τον καθορισμό των συναντήσεων αποφεύγοντας τα κουραστικά και επαναλαμβανόμενα τηλέφωνα.

Συνήθως τα προγράμματα αυτά έχουν και δυνατότητα χρονικού προγραμματισμού, με την οποία μπορούν να επιλέγονται τα κενά χρόνου των συναδέλφων και να προγραμματίζονται οι συνεδριάσεις στην καταλληλότερη στιγμή.

Μπορούν να κρατούν τα προσωπικά αρχεία του χρήστη ή άλλα αρχεία πελατών, συνεργατών, προμηθευτών, διαφημιζομένων, προσωπικού στους οποίους είναι δυνατή και εύκολη η αποστολή και λήψη μηνυμάτων ηλεκτρονικού ταχυδρομείου.

Τέλος διάφορες άλλες λειτουργίες όπως διαχείριση σημειώσεων, χρονικά διάφορων ενεργειών, αναζήτηση διάφορων πληροφοριών, εορτολόγιο, τήρηση πρωτοκόλλου κ.α. μπορούν να συμπεριλαμβάνονται στις δυνατότητες των πακέτων αυτών.

Όλες οι προηγούμενες δυνατότητες εντάσσονται στις προσπάθειες των κατασκευαστών λογισμικού προς την πορεία για το 'γραφείο χωρίς χαρτιά'.

### 6.7.6 Frontpage

#### Frontpage

Έχει ήδη γίνει αναφορά στο ρόλο και τη σημασία του διαδικτύου. Σήμερα ο κύριος όγκος πληροφοριών που μπορούν να αναζητηθούν περιέχονται στο λεγόμενο Παγκόσμιο Ιστό. Οι πληροφορίες στον ιστό είναι διασπασμένες σε τόπους, που στην ουσία αποτελούν ηλεκτρονικές σελίδες. Υπάρχει λοιπόν σοβαρό κίνητρο για κάθε επιχείρηση να διαθέτει σελίδα στον ιστό, η οποία πρέπει να είναι ελκυστική και να ενημερώνεται τακτικά. Η δημιουργία ιστοσελίδων αποτελεί πλέον μια ακόμη (νέα) εργασία γραφείου, που έχει ως κύριο εργαλείο το πακέτο του Frontpage.

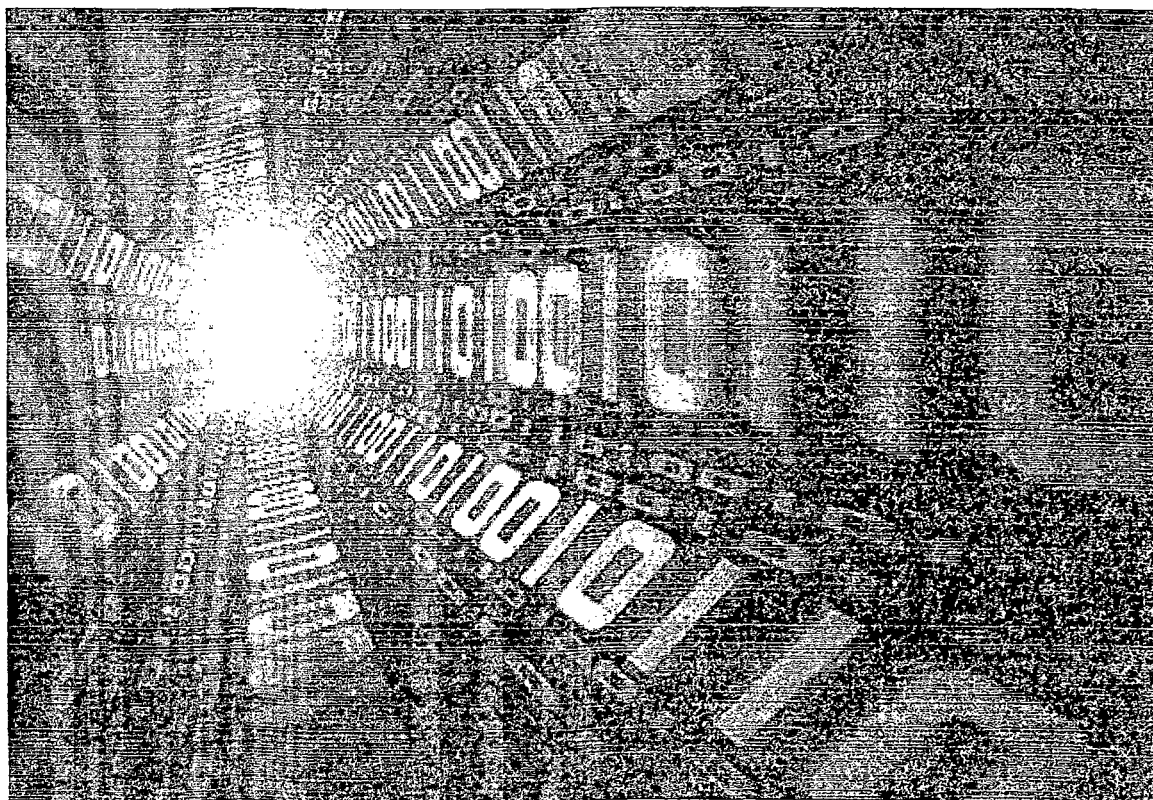
Για τη δημιουργία μιας ιστοσελίδας μπορεί να χρησιμοποιηθεί κάποια ειδική γλώσσα προγραμματισμού (π.χ. HTML, Java) ή κάποιο πακέτο.

Αυτά τα πακέτα αποτελούν ένα ολοκληρωμένο περιβάλλον ανάπτυξης και συντήρησης τόπων. Παρέχουν πολλές ευκολίες και εργαλεία για την ένθεση κειμένου, εικόνων, ήχου και βίντεο. Ο χρήστης μπορεί να δημιουργήσει εύκολα υπερσυνδέσεις που παραπέμπουν σε άλλες σελίδες ή τόπους. Η αποδοτική εργασία με τα πακέτα αυτά καθίσταται ευχερής μετά από μικρή εκπαίδευση και εξοικείωση.

### Επίλογος

Οι νέες τεχνολογίες πληροφορίας και επικοινωνίας είναι πια κομμάτι της καθημερινότητας. Τις χρησιμοποιούμε κάθε φορά που καλούμε κάποιον από ένα κινητό τηλέφωνο ή πληρωνουμε με μία πιστωτική κάρτα, όταν στο ταμείο του σουπερμάρκετ «σαρώνονται» οι κωδικοί των προϊόντων που αγοράσαμε, ή όταν χρησιμοποιούμε ένα από τα χιλιάδες προϊόντα και υπηρεσίες που ενσωματώνουν ή βασίζονται στις νέες τεχνολογίες. Με περισσότερο ή λιγότερο ορατό τρόπο, οι τεχνολογίες αυτές αλλάζουν το πώς ζούμε, δουλεύουμε, διασκεδάζουμε, μαθαίνουμε και συναναστρεφόμαστε.

Στο κατώφλι του 21<sup>ου</sup> αιώνα, η ραγδαία εξέλιξη αυτών των τεχνολογιών, η ευρεία τους διάχυση σε όλη την οικονομία και η ενσωμάτωσή τους σε όλες σχεδόν τις διαστάσεις της καθημερινής ζωής χτίζουν μία παγκόσμια **Κοινωνία της Πληροφορίας** με νέα δεδομένα και νέες ευκαιρίες για την ανάπτυξη, την απασχόληση, την ευημερία και την ποιότητα ζωής όλων των ανθρώπων.





Βιβλιογραφία:

- Βιβλίο ECDL, ΘΕΩΡΙΑ & ΠΡΑΞΗ – Χρήστος Γουλτίδης
- Βιβλίο ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ – Κ. Κλημόπουλος
- Βιβλίο ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ, Αρχές Σχεδίασης - Τζιόλη
  
- Πηγές INTERNET
- <http://el.wikipedia.org/wiki>
- <http://www.pcguides.com/ref/cpu/arch/int/compFPU-c.html>
- [http://utut.essortment.com/romramstorage\\_rio0.htm](http://utut.essortment.com/romramstorage_rio0.htm)
- <http://www.spinellis.gr/arch/memory/virtual.htm>
- <http://www.it.uom.gr/project/mycomputer/memory/>
- <http://en.wikipedia.org/wiki/Compiler>
- [http://en.wikipedia.org/wiki/Interpreter \(computing\)](http://en.wikipedia.org/wiki/Interpreter_(computing))
- [http://en.wikipedia.org/wiki/Blaise Pascal](http://en.wikipedia.org/wiki/Blaise_Pascal)
- <http://www.spinellis.gr/isdi/style/struct.htm>
- <http://www.spinellis.gr/c3/lisp/lisp.htm>
- <http://www.go-online.gr/>
- [http://en.wikipedia.org/wiki/World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web)
- [http://el.wikipedia.org/wiki/Web browser](http://el.wikipedia.org/wiki/Web_browser)