

Μάθημα: Εισαγωγή στην Πληροφορική και την Ανοικτότητα

Κεφάλαιο 1.5 - Υλικό και Λογισμικό υπολογιστή

Εισαγωγή

1.5.1 Υλικό (Hardware)

Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)

Κύρια μνήμη

Κρυφή μνήμη (cache)

Περιφερειακές μονάδες

Διάδρομοι επικοινωνίας

1.5.2 Λογισμικό (Software)

Λογισμικό συστήματος

Λειτουργικό σύστημα

Λογισμικό εφαρμογών

Λογισμικό γενικής χρήσης

Λογισμικό ειδικού σκοπού

Λογισμικό ανάπτυξης

Γλώσσες προγραμματισμού



Στόχοι

Μετά την ολοκλήρωση του κεφαλαίου οι σπουδαστές/τριες θα μπορούν να:

- ✓ αναγνωρίζουν τα μέρη του υπολογιστή
- ✓ κατανοούν τη λειτουργία των μονάδων του υλικού
- ✓ απαριθμούν τις μονάδες του υλικού υπολογιστή
- ✓ αναφέρουν τη χρησιμότητα των μονάδων εισόδου-εξόδου
- ✓ κατανοούν την έννοια του λογισμικού και τη σχέση του με το υλικό (hardware).
- ✓ γνωρίζουν τα είδη λογισμικού
- ✓ γνωρίζουν τον ρόλο του Λογισμικού Συστήματος
- ✓ γνωρίζουν τις κατηγορίες του λογισμικού εφαρμογών.
- ✓ κατανοούν το ιστορικό της ανάπτυξης και χρήσης των γλωσσών προγραμματισμού.

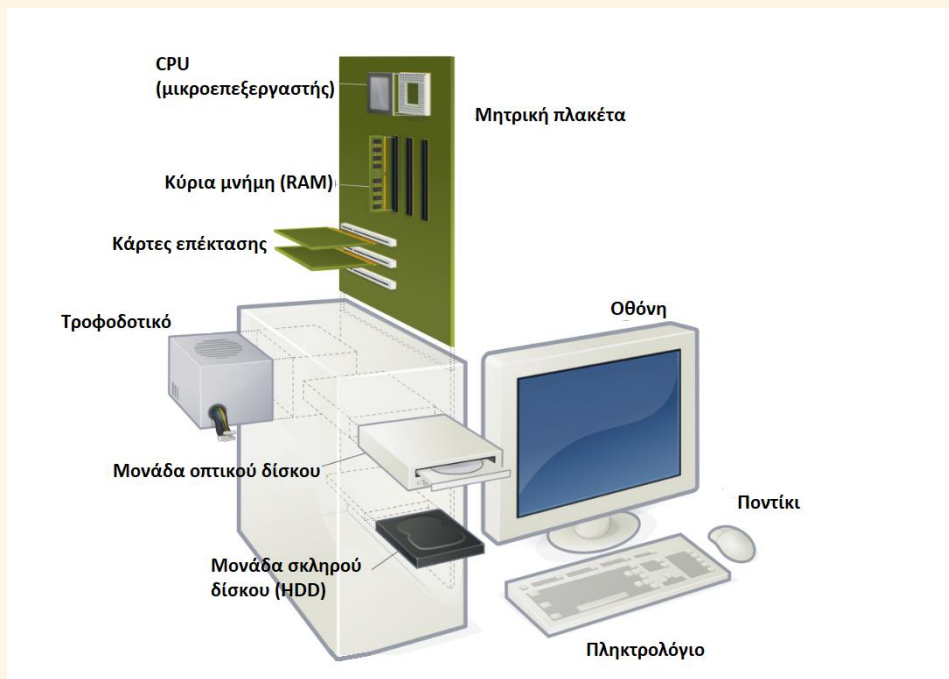
Εισαγωγή

Για να κατανοήσουμε ποια είναι τα βασικά μέρη από τα οποία αποτελείται ένα υπολογιστικό σύστημα και τον τρόπο που αυτά επικοινωνούν μεταξύ τους, θα αναφέρουμε τις βασικές λειτουργίες ενός υπολογιστή:

Τα δεδομένα που δέχεται μέσω των συσκευών εισόδου περνούν από διάφορα στάδια επεξεργασίας σύμφωνα με τις εντολές που δίνονται, οι οποίες προσδιορίζουν με απόλυτη ακρίβεια τον τρόπο επεξεργασίας των δεδομένων, ώστε να παραχθούν τα τελικά αποτελέσματα. Τα τελικά αποτελέσματα μπορούν να αποθηκευτούν προσωρινά ή μόνιμα για να είναι διαθέσιμα για μελλοντική χρήση. Επίσης, τα τελικά αποτελέσματα εξάγονται στις συσκευές εξόδου του υπολογιστικού συστήματος. Τα δεδομένα εισόδου καθώς και τα τελικά αποτελέσματα αναπαρίστανται στο εσωτερικό του υπολογιστή με ψηφιακό τρόπο (ακολουθίες 0 και 1). Συνεπώς, για την επεξεργασία τους χρησιμοποιούνται ψηφιακά κυκλώματα, οι βασικές αρχές λειτουργίας των οποίων είδαμε στο κεφάλαιο 1.4.

Κάθε υπολογιστικό σύστημα, όσο μεγάλο ή μικρό κι αν είναι, για να μπορέσει να εκτελέσει τις λειτουργίες που περιγράψαμε παραπάνω πρέπει να αποτελείται από δύο βασικά μέρη που αλληλοεπιδρούν μεταξύ τους, το υλικό μέρος (hardware) και το λογισμικό (software).

Ως υλικό (hardware) ορίζεται το σύνολο των φυσικών εξαρτημάτων ενός υπολογιστή, όπως π.χ. ηλεκτρικά και ηλεκτρονικά στοιχεία, μικροτσίπ, πληκτρολόγιο, ποντίκι κλπ.



Ως λογισμικό (software) ορίζεται το σύνολο από προγράμματα, διαδικασίες και οδηγίες χρήσης για την εκτέλεση συγκεκριμένων εργασιών σε ένα υπολογιστικό σύστημα.



Το υλικό και το λογισμικό είναι αλληλένδετα, χωρίς λογισμικό, το υλικό ενός υπολογιστή δεν θα λειτουργούσε. Ωστόσο, χωρίς το υλικό για την εκτέλεση των εντολών του προγράμματος λογισμικού, το λογισμικό θα ήταν άχρηστο.

1.5.1 Υλικό (Hardware)

Μπορούμε να χωρίσουμε τις μονάδες που αποτελούν το υλικό του υπολογιστή σε τρεις μεγάλες κατηγορίες ή υποσυστήματα με βάση τις διαφορετικές λειτουργίες που εκτελεί το κάθε υποσύστημα:

Την Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ / Central Processing Unit – CPU ή μικροεπεξεργαστής/ microprocessor), την κύρια μνήμη (Main Memory) και τις περιφερειακές μονάδες (I/O συσκευές). Όλα τα υποσυστήματα επικοινωνούν μεταξύ τους μέσω των διαδρόμων/διαύλων (buses) οι οποίοι περιέχουν γραμμές ελέγχου και γραμμές δεδομένων.



Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)

Η ΚΜΕ εκτελεί τις πιο σημαντικές λειτουργίες του υπολογιστή καθώς είναι υπεύθυνη για την επεξεργασία των δεδομένων, τον έλεγχο της λειτουργίας του υπολογιστή και την εκτέλεση λειτουργιών διασύνδεσης των μονάδων εισόδου και εξόδου.

Η ΚΜΕ κατασκευάζεται στη μορφή ενός ολοκληρωμένου κυκλώματος και αποτελείται από τα εξής μέρη :

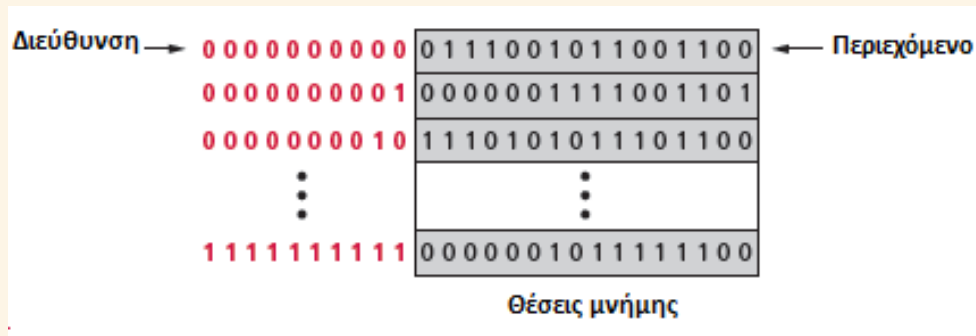
- την **Αριθμητική Λογική Μονάδα (ALU)** που εκτελεί αριθμητικές και λογικές πράξεις που είδαμε στα προηγούμενα κεφάλαια.
- Την **μονάδα ελέγχου (CU - Control Unit)** που ελέγχει τη μεταφορά πληροφοριών μεταξύ ΚΜΕ και κύριας μνήμης & αντίστροφα, την ενεργοποίηση και απενεργοποίηση των μονάδων εισόδου/εξόδου, την έναρξη και διακοπή της εκτέλεσης ενός προγράμματος, τον συντονισμό της εκτέλεσης των εντολών του προγράμματος, την εκτέλεση των πράξεων στην ALU
- Τους **καταχωρητές (registers)**, που χρησιμεύουν ως χώροι γρήγορης αποθήκευσης για να διατηρούν δεδομένα προσωρινά. Οι καταχωρητές είναι σύγχρονα ακολουθιακά κυκλώματα που αποτελούνται από έναν αριθμό κατάλληλα συνδεδεμένων flip-flop. Η ΚΜΕ περιέχει πολλούς καταχωρητές, από τους οποίους άλλοι είναι γενικής χρήσης, ενώ άλλοι επιτελούν μια συγκεκριμένη λειτουργία (καταχωρητές ειδικής χρήσης). Τους καταχωρητές τους χειρίζεται απευθείας η μονάδα ελέγχου της ΚΜΕ κατά την εκτέλεση των εντολών για να επιταχύνει την εργασία του επεξεργαστή καθώς η πρόσβαση σε αυτούς είναι πιο γρήγορη από την πρόσβαση σε θέσεις μνήμης. Ο αριθμός των καταχωρητών σε μια ΚΜΕ και το μέγεθος κάθε καταχωρητή επηρεάζουν την ισχύ και την ταχύτητα μιας ΚΜΕ. Όσο μεγαλύτερος είναι ο αριθμός των καταχωρητών (δέκα έως εκατοντάδες) και μεγαλύτερο το μέγεθος κάθε καταχωρητή (8 bit έως 64 bit), τόσο καλύτερος είναι ο επεξεργαστής.

Κύρια μνήμη

Η κύρια μνήμη είναι το δεύτερο σημαντικό υποσύστημα σε έναν υπολογιστή μετά την ΚΜΕ. Εκεί αποθηκεύονται τα προγράμματα που θα εκτελεστούν, αλλά και τα δεδομένα που θα υποστούν επεξεργασία καθώς και τα αποτελέσματα που θα παραχθούν.

Αποτελείται από μεγάλο αριθμό θέσεων αποθήκευσης (cells) που η καθεμία έχει ένα μοναδικό αναγνωριστικό που ονομάζεται διεύθυνση. Τα δεδομένα αποθηκεύονται σε κάθε θέση μνήμης σε ομάδες των 8 bit που ονομάζονται λέξεις (word). Μια λέξη μπορεί να είναι μια ομάδα από 8 bit, 16 bit, 32 bit ή 64 bit (και αυξάνεται).

Μία μνήμη των 2GB για παράδειγμα, αποτελείται από 2 δύο δισεκατομμύρια μοναδικά αναγνωρίσιμες θέσεις μνήμης καθεμία από τις οποίες περιέχει μια λέξη (word) πληροφοριών. Κάθε θέση της μνήμης μπορεί να περιέχει είτε δεδομένα, είτε μία εντολή. Επίσης σε κάθε θέση μνήμης χαρακτηρίζεται από δύο δυαδικούς αριθμούς, το περιεχόμενο της και τη διεύθυνση της. Στην πραγματικότητα, μόνο τα περιεχόμενα κάθε θέσης βρίσκονται αποθηκευμένα στη μνήμη. Οι διευθύνσεις υπονοούνται γνωρίζοντας ότι πρόκειται για αύξοντες δυαδικούς αριθμούς ξεκινώντας πάντα από την πρώτη θέση της μνήμης που έχει διεύθυνση 0.



Στην κύρια μνήμη ανήκουν δύο είδη ηλεκτρονικών κυκλωμάτων: η Μνήμη Τυχαίας Προσπέλασης (Random Access Memory, RAM) και η Μνήμη Μόνο για Ανάγνωση (Read Only Memory, ROM)

Η μνήμη **RAM (Random Access Memory)** είναι μνήμη ανάγνωσης/εγγραφής, δηλαδή η ΚΜΕ μπορεί να διαβάζει τα περιεχόμενα της όσο και να γράφει νέα περιεχόμενα αντικαθιστώντας τα προηγούμενα. Η RAM είναι μνήμη τυχαίας προσπέλασης, δηλαδή η πρόσβαση σε οποιαδήποτε θέση της απαιτεί τον ίδιο ακριβώς χρόνο. Τέλος, η RAM χαρακτηρίζεται ως ασταθής μνήμη, δηλαδή τα περιεχόμενα της χάνονται όταν δεν τροφοδοτείται με ηλεκτρικό ρεύμα ο υπολογιστής.

Οι σύγχρονες μνήμες RAM κατασκευάζονται είτε με flip-flop και ονομάζονται στατικές (SRAM), είτε από πυκνωτές και ονομάζονται δυναμικές (DRAM). Η διαφορά τους είναι ότι τα δεδομένα στην SRAM διατηρούνται στη μνήμη μέχρι ν' αντικατασταθούν από άλλα, ενώ στην DRAM τα δεδομένα για να διατηρηθούν στη μνήμη υπάρχει μια αυτόματη διαδικασία όπου αυτά ξαναγράφονται περιοδικά (ανανέωση). Η μνήμη SRAM είναι πιο γρήγορη, αλλά πιο ακριβή γι' αυτό και στην πλειοψηφία των υπολογιστών σαν κύρια μνήμη χρησιμοποιείται η DRAM.

Η μνήμη **ROM (Read Only Memory)**, είναι μια μνήμη που επιτρέπει μόνο την ανάγνωση των περιεχομένων της, σε αντίθεση με τη RAM που επιτρέπει τόσο ανάγνωση όσο και εγγραφή. Το πλεονέκτημα της είναι ότι τα περιεχόμενα της δεν χάνονται όταν δεν τροφοδοτείται με ηλεκτρικό ρεύμα ο υπολογιστής. Η ROM είναι επίσης μνήμη τυχαίας προσπέλασης και χρησιμοποιείται για τη μόνιμη αποθήκευση προγραμμάτων που είναι απαραίτητα για την εκτέλεση ορισμένων βασικών λειτουργιών του υπολογιστή, όπως το πρόγραμμα εκκίνησης (boot program) του συστήματος.

Κρυφή μνήμη (cache)

Μέχρι τη δεκαετία του '90 η ταχύτητα προσπέλασης της ΚΜΕ στην κύρια μνήμη ήταν ελάχιστα πιο αργή από την ταχύτητα προσπέλασης στους καταχωρητές. Με την εξέλιξη της τεχνολογίας όμως η ταχύτητα των επεξεργασιών αυξήθηκε σημαντικά ενώ η ταχύτητα της κύρια μνήμης δεν αυξήθηκε ανάλογα. Αυτό συνέβη γιατί όσο ταχύτερη είναι μια μνήμη τόσο περισσότερο ανεβαίνει το κόστος της και κατά συνέπεια γίνεται οικονομικά ασύμφορη η παραγωγή υπολογιστών. Προκειμένου να λυθεί αυτό το πρόβλημα υιοθετήθηκε η χρήση της **μνήμης cache ή κρυφής μνήμης**, δηλαδή η χρήση σε μικρή ποσότητα (δεκάδες ή εκατοντάδες KB) γρήγορης και ακριβής μνήμης με σκοπό να επιταχύνει τη λειτουργία του συστήματος και συνεπώς να βελτιώσει την ταχύτητα του υπολογιστή.

Η κρυφή μνήμη (cache) τοποθετείται ανάμεσα στον επεξεργαστή και την κύρια μνήμη και τη διαχειρίζεται αποκλειστικά ο επεξεργαστής. Σε αυτήν αποθηκεύονται προσωρινά αντίγραφα δεδομένων από την κύρια μνήμη που χρησιμοποιήθηκαν πρόσφατα και που υπάρχει αυξημένη πιθανότητα να επαναχρησιμοποιηθούν σύντομα. Παράδειγμα τέτοιων δεδομένων μπορεί να είναι οι εντολές μια δομής επανάληψης που εκτελούνται πολλές φορές.

Η διαφορά μεταξύ κρυφής μνήμης και καταχωρητών (registers), που είδαμε παραπάνω, είναι ότι οι καταχωρητές είναι ταχύτεροι από την κρυφή μνήμη όμως ο χώρος αποθήκευσης τους είναι πολύ μικρός. Αντίθετα, η κρυφή μνήμη έχει χαμηλότερη ταχύτητα από τους καταχωρητές, αλλά μεγαλύτερο χώρο αποθήκευσης.

Όπως φαίνεται από το παρακάτω σχήμα, η μνήμη ενός ηλεκτρονικού υπολογιστή μπορεί να διαταχθεί σε μορφή πυραμίδας. Τα κατώτερα επίπεδα της πυραμίδας προσφέρουν μεγαλύτερη αλλά πιο αργή μνήμη αντιθέτως, τα ανώτερα προσφέρουν μικρότερη μνήμη αλλά πολύ πιο γρήγορη.



Περιφερειακές μονάδες

Το τρίτο υποσύστημα σε ένα υπολογιστικό σύστημα είναι οι περιφερειακές μονάδες. Αυτό το υποσύστημα επιτρέπει στον υπολογιστή να επικοινωνεί με το εξωτερικό περιβάλλον και να αποθηκεύει μόνιμα δεδομένα και προγράμματα ακόμα κι όταν δεν τροφοδοτείται με ρεύμα.

Οι περιφερειακές μονάδες χωρίζονται σε εισόδου, εξόδου, εισόδου/εξόδου και βοηθητικής μνήμης.

Για την εισαγωγή δεδομένων για επεξεργασία στον υπολογιστή χρησιμοποιούμε διάφορες συσκευές που ονομάζονται **συσκευές εισόδου**. Χαρακτηριστικά παραδείγματα τέτοιων συσκευών είναι: το πληκτρολόγιο, το ποντίκι, η επιφάνεια αφής, η web κάμερα, ο σαρωτής, το μικρόφωνο, οι αναγνώστες barcode, τα χειριστήρια παιχνιδιών κ.α.

Για την εμφάνιση στο χρήστη των αποτελεσμάτων της επεξεργασίας των δεδομένων από τον υπολογιστή χρησιμοποιούνται οι **συσκευές εξόδου**. Τέτοιες συσκευές είναι: η οθόνη, ο εκτυπωτής, τα ηχεία, τα ακουστικά κ.α.

Για την επικοινωνία μεταξύ των υπολογιστών χρησιμοποιούνται οι **συσκευές εισόδου/εξόδου**. Οι συσκευές αυτές μπορούν τόσο να δίνουν είσοδο στον υπολογιστή, όσο και να παρουσιάζουν, να αποθηκεύουν ή να προωθούν πληροφορίες εξόδου. Τέτοιες συσκευές είναι η κάρτα δικτύου (λήψη και αποστολή δεδομένων) και η οθόνη αφής.

Τέλος, οι περιφερειακές μονάδες στις οποίες μπορούμε να αποθηκεύσουμε δεδομένα και προγράμματα και αυτά να διατηρούνται κι όταν ο υπολογιστής δεν τροφοδοτείται με ρεύμα, λέγονται **συσκευές βοηθητικής μνήμης**. Κάποιες από αυτές τις συσκευές εγκαθίστανται στον Η/Υ κατά την αρχική του συναρμολόγηση, άλλες μπορούν να εγκαθίστανται και να αποκαθίστανται αργότερα κατά βούληση.

Η βοηθητική μνήμη σε σύγκριση με την κύρια μνήμη του υπολογιστή είναι φθηνότερη σε κόστος, έχει μεγαλύτερη χωρητικότητα και ο χρόνος πρόσβασης σε αυτή είναι πιο αργός.

Τέτοιες συσκευές είναι: ο σκληρός δίσκος, η μνήμη flash (USB, SD), οι οπτικοί δίσκοι (CD, DVD), οι μαγνητικές ταινίες κ.α.

Στον παρακάτω πίνακα βλέπουμε τις κυριότερες διαφορές μεταξύ κύριας και βοηθητικής μνήμης.

Χαρακτηριστικό	Κύρια μνήμη	Βοηθητική μνήμη
Αποθήκευση δεδομένων	Όχι. Τα περιεχόμενα της χάνονται όταν σβήσει ο υπολογιστής	Τα περιεχόμενα παραμένουν μόνιμα

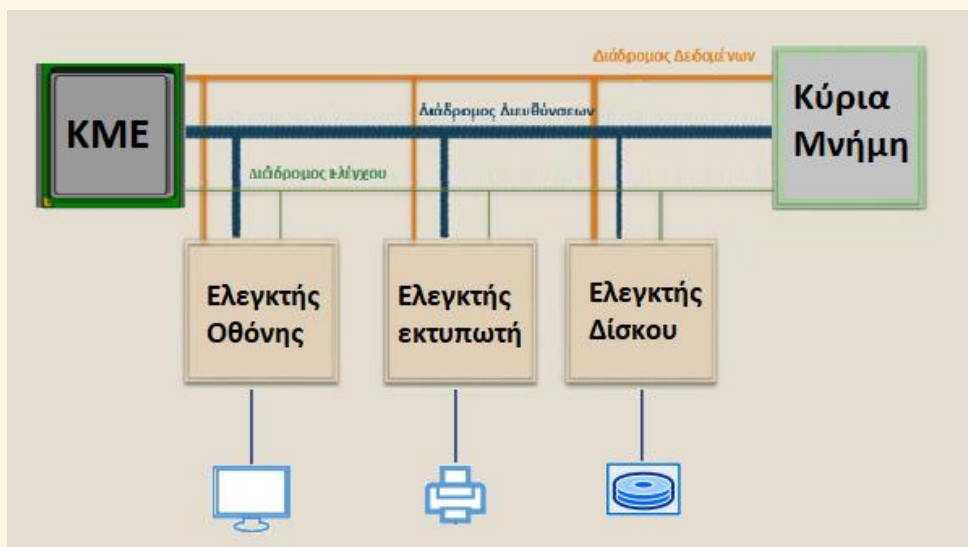
Χωρητικότητα	Μικρή. Ο περιορισμός οφείλεται στο κόστος	Άπειρη. Μπορούμε ν αγοράσουμε όση χωρητικότητα θέλουμε.
Ταχύτητα	Μεγάλη (nanoseconds / bit)	Μικρή (milliseconds / bit)

Διάδρομοι επικοινωνίας

Σε ένα υπολογιστικό σύστημα τα επιμέρους υποσυστήματα πρέπει να επικοινωνούν μεταξύ τους για να ανταλλάσσουν δεδομένα. Για παράδειγμα, η ΚΜΕ επικοινωνεί με την κύρια μνήμη, όπως και με τις περιφερειακές μονάδες. Αυτό γίνεται μέσω των **διαδρόμων επικοινωνίας (buses)**. Οι διάδρομοι επικοινωνίας είναι σύνολα καλωδίων στα οποία συνδέονται οι επιμέρους μονάδες με σκοπό τη μεταφορά δεδομένων, διευθύνσεων ή σημάτων ελέγχου από το ένα υποσύστημα στο άλλο με την μεγαλύτερη δυνατή ταχύτητα.

Ο κάθε διάδρομος αποτελείται από τις γραμμές ελέγχου (control), τις γραμμές διευθύνσεων (address) και τις γραμμές δεδομένων (data). Οι γραμμές ελέγχου καθορίζουν ποια μονάδα θα στείλει δεδομένα μέσα από τις γραμμές δεδομένων. Οι γραμμές δεδομένων μεταφέρουν τις δυαδικές πληροφορίες από τη συσκευή-αποστολέα στη συσκευή-παραλήπτη. Οι γραμμές διευθύνσεων μεταφέρουν τη διεύθυνση μιας θέσης μνήμης ή τη διεύθυνση μιας περιφερειακής συσκευής όπου θα γραφούν ή απ' όπου θα διαβαστούν τα δεδομένα.

Οι περιφερειακές μονάδες (I/O συσκευές) δεν μπορούν να συνδεθούν απευθείας στους διαδρόμους επικοινωνίας που συνδέουν την ΚΜΕ με την κύρια μνήμη, επειδή η τεχνολογία κατασκευής αυτών των συσκευών είναι διαφορετική. Οι περιφερειακές μονάδες είναι είτε ηλεκτρομηχανικές, είτε μαγνητικές ή οπτικές συσκευές και επομένως λειτουργούν με μικρότερες ταχύτητες από τη ΚΜΕ και την κύρια μνήμη που είναι ηλεκτρονικές συσκευές. Κάθε περιφερειακή μονάδα συνδέεται με τους διαδρόμους επικοινωνίας μέσω μιας συσκευής που ονομάζεται **ελεγκτής εισόδου/εξόδου (interface controller)**. Οι ελεγκτές διακρίνονται σε σειριακούς και παράλληλους. Οι σειριακοί χρησιμοποιούνται για να συνδέουν πιο αργές συσκευές (ποντίκι, πληκτρολόγιο), οι παράλληλοι χρησιμοποιούνται για να συνδέουν πιο γρήγορες συσκευές (εκτυπωτής, δίσκοι).



**Δραστηριότητες**

1. Ποια από τις παρακάτω μονάδες δεν περιλαμβάνεται στην Κεντρική Μονάδα Επεξεργασίας (CPU) ;

- A) Καταχωρητές
- B) Μονάδα Ελέγχου
- Γ) Αριθμητική και Λογική μονάδα
- Δ) Περιλαμβάνονται όλες οι προηγούμενες

2. Οι καταχωρητές είναι:

- A) Μικρές μνήμες μέσα στους επεξεργαστές
- B) Πολλοί στους σύγχρονους επεξεργαστές
- Γ) Το ταχύτερο επίπεδο μνήμης ενός υπολογιστικού συστήματος
- Δ) Όλα τα προηγούμενα

3. Ποια είναι η βασική διαφορά μεταξύ μίας μνήμης RAM και μίας μνήμης ROM;

- A) Καμία
- B) Η RAM είναι πιο γρήγορη από τη ROM
- Γ) Τα δεδομένα της RAM σβήνονται, όταν διακοπεί η τροφοδοσία, ενώ της ROM όχι
- Δ) Η ROM ενός συστήματος είναι μεγαλύτερη από τη RAM

4. Ποια από τα ακόλουθα είναι είδη βοηθητικής μνήμης;

- | | | |
|-------------------|--------------------|------------|
| A) Μνήμη RAM | Γ) Μνήμη ROM | Ε) DVD-ROM |
| B) Σκληρός δίσκος | Δ) USB Flash Drive | ΣΤ) BIOS |

5. Ποια λειτουργία επιτελεί η Αριθμητική και Λογική Μονάδα (Arithmetic and Logic Unit, ALU) σε μια Κεντρική Μονάδα Επεξεργασίας, ΚΜΕ (Central Processing Unit, CPU);

- A) Είσοδο δεδομένων και έξοδο πληροφοριών.
- B) Έλεγχο και συντονισμό.
- Γ) Αποθήκευση ενδιάμεσων αποτελεσμάτων.
- Δ) Μαθηματικές πράξεις και συγκρίσεις.

6. Πώς ονομάζονται τα κυκλώματα που παρεμβάλλονται μεταξύ επεξεργαστή και συσκευών εισόδου/εξόδου για να διευκολύνουν τη διασύνδεσή τους;

- A) Διερμηνευτές
- B) Ελεγκτές
- Γ) Μεταγλωττιστές
- Δ) Διάδρομοι επικοινωνίας

7. Αναζητήστε πληροφορίες σχετικά με τη λειτουργία των μνημών τύπου flash και τη σχέση τους με τις μνήμες ROM.

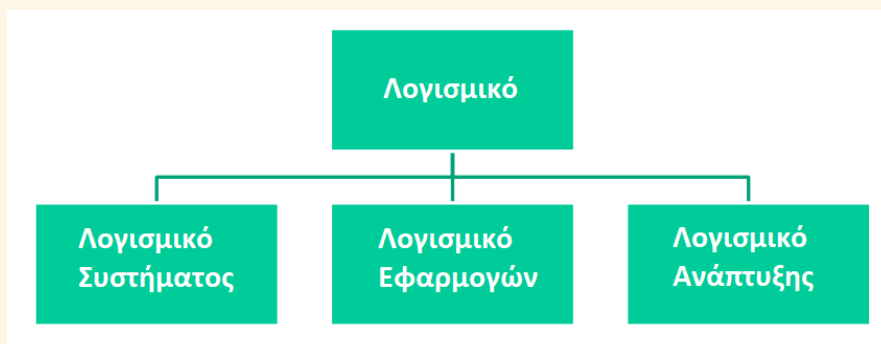
1.5.2 Λογισμικό (Software)

Για να μπορεί το υλικό μέρος του Υπολογιστή να εκτελεί και τη πιο απλή επεξεργασία δεδομένων χρειάζεται ένα σύνολο οδηγιών/εντολών που το καθοδηγούν βήμα προς βήμα, συντονίζουν τα διάφορα μέρη του, με σκοπό την πραγματοποίηση συγκεκριμένων λειτουργιών. Το σύνολο των οδηγιών αυτών που κατευθύνουν με κάθε λεπτομέρεια το υλικό για να εκτελέσει μια συγκεκριμένη εργασία ονομάζεται **πρόγραμμα**.

Το σύνολο των προγραμμάτων που χρησιμοποιούνται από ένα υπολογιστικό σύστημα ονομάζεται **Λογισμικό (Software)**. Το λογισμικό σε αντίθεση με το υλικό είναι ένα λογικό και όχι φυσικό συστατικό του υπολογιστικού συστήματος.

Το λογισμικό αναπτύσσεται χρησιμοποιώντας εντολές σε **γλώσσες προγραμματισμού υψηλού επιπέδου**, που είναι πιο κοντά στη φυσική γλώσσα του ανθρώπου. Τα προγράμματα αυτά στη συνέχεια μεταγλωττίζονται σε **γλώσσα μηχανής**, δηλαδή σε εντολές γραμμένες σε μορφή ακολουθιών bit που είναι άμεσα εκτελέσιμες από την ΚΜΕ.

Το λογισμικό διακρίνεται σε τρεις μεγάλες κατηγορίες: το λογισμικό συστήματος, το λογισμικό εφαρμογών και το λογισμικό ανάπτυξης (προγραμματισμού).



Λογισμικό συστήματος

Το λογισμικό συστήματος περιλαμβάνει τα προγράμματα τα οποία είναι απαραίτητα για τον έλεγχο της λειτουργίας και τη διαχείριση των πόρων του υπολογιστή. Επίσης, παρέχει το περιβάλλον για την ανάπτυξη και την εκτέλεση του λογισμικού εφαρμογών. Είναι ουσιαστικά ο ενδιάμεσος μεταξύ του υλικού και των εφαρμογών του χρήστη.

Το βασικότερο λογισμικό της κατηγορίας αυτής είναι το **Λειτουργικό Σύστημα (operating system, OS)**. Στην κατηγορία αυτή εντάσσονται επίσης και τα παρακάτω προγράμματα:

Μεταγλωττιστές (Compilers) προγράμματα που διαβάζουν κώδικα γραμμένο σε μια γλώσσα προγραμματισμού (την πηγαία γλώσσα) και τον μεταφράζουν σε ισοδύναμο κώδικα σε μια άλλη γλώσσα προγραμματισμού.

Διερμηνείς (Interpreters) προγράμματα που διαβάζουν, μεταφράζουν και εκτελούν εντολή προς εντολή κώδικα που έχει γραφτεί σε κάποια γλώσσα προγραμματισμού.

Αποσφαλματωτές (Debuggers) για την παρακολούθηση της εκτέλεσης ενός προγράμματος, βήμα προς βήμα, και τον εντοπισμό λαθών.

Συμβολομεταφραστές (Assemblers) προγράμματα τα οποία μεταφράζουν άλλα προγράμματα που είναι γραμμένα σε συμβολική γλώσσα (π.χ. assembly) στην αριθμητική γλώσσα μηχανής που είναι συμβατή με την μηχανή.

Βοηθητικά προγράμματα (utilities) όπως π.χ. προγράμματα ανασυγκρότησης και διαμόρφωσης δίσκων, η επαναφορά συστήματος κ.α.

Οδηγοί συσκευών (device drivers) μικρά προγράμματα που συνοδεύουν συνήθως μια περιφερειακή συσκευή ενός υπολογιστή, π.χ. ένα αποθηκευτικό μέσο, όπως μια συσκευή CD-ROM, και το οποίο πρέπει να εγκατασταθούν στον υπολογιστή προκειμένου αυτή να λειτουργήσει.

Το BIOS (Basic Input/Output System) είναι το πρόγραμμα που ξεκινά το υπολογιστικό σύστημα μόλις ενεργοποιηθεί. Το πρόγραμμα αυτό διαχειρίζεται τη ροή δεδομένων μεταξύ του λειτουργικού συστήματος και των συνδεδεμένων συσκευών όπως ο σκληρός δίσκος, ο προσαρμογέας βίντεο, το πληκτρολόγιο, το ποντίκι και ο εκτυπωτής. Το BIOS φορτώνει το λειτουργικό σύστημα στην κύρια μνήμη ή στη μνήμη τυχαίας προσπέλασης (RAM) του υπολογιστή.

Λειτουργικό σύστημα

Το λειτουργικό σύστημα είναι πολύ σημαντικό χαρακτηριστικό μέρος του υπολογιστικού συστήματος. Είναι το πρώτο πρόγραμμα που φορτώνεται αυτόματα με την εκκίνηση του υπολογιστή στην κύρια μνήμη του. Τα υπόλοιπα προγράμματα που εκτελούνται σε ένα υπολογιστή εξαρτώνται από το λειτουργικό σύστημα που αυτός διαθέτει. Στην πράξη πρόκειται για ένα επίπεδο λογισμικού που μεσολαβεί μεταξύ του υλικού και των εκτελούμενων προγραμμάτων σε έναν ηλεκτρονικό υπολογιστή.

Το λειτουργικό σύστημα είναι το λογισμικό που διαχειρίζεται τις λειτουργίες των μονάδων του υλικού και περιλαμβάνει όλες τις επιπρόσθετες υπηρεσίες που χρειάζονται ώστε ο υπολογιστής να μπορεί να εκτελεί τις εφαρμογές που επιθυμούν οι χρήστες. Με το μεσολαβητικό επίπεδο διασύνδεσης που προσφέρει το λειτουργικό σύστημα οι εκτελούμενες εφαρμογές δεν χρειάζεται να γνωρίζουν με ακρίβεια τη δομή του υποκείμενου υλικού και παρέχεται η δυνατότητα της ταυτόχρονης εκτέλεσης πολλών εφαρμογών χωρίς να έρχονται σε διένεξη μεταξύ τους ή με τον υπολογιστή.

Ποιο αναλυτικά το λειτουργικό σύστημα:

- Διαχειρίζεται και εκχωρεί του πόρους του υλικού (όπως είναι η ΚΜΕ, η κύρια μνήμη, οι περιφερειακές συσκευές) μεταξύ των διαφόρων προγραμμάτων που εκτελούνται στον υπολογιστή.
- Δημιουργεί ένα φιλικό περιβάλλον επικοινωνίας ανάμεσα στον χρήστη και τον υπολογιστή. Αυτό μπορεί να είναι μια γραφική διεπαφή χρήστη (GUI), μια διεπαφή γραμμής εντολών (CLI) ή ένας συνδυασμός και των δύο.
- Ενεργοποιεί, απενεργοποιεί και διαχειρίζεται διάφορα προγράμματα
- Οργανώνει και διαχειρίζεται τα αρχεία του συστήματος
- Εφαρμόζει μηχανισμούς ασφάλειας του υπολογιστή και των δεδομένων

Τα σύγχρονα Λ.Σ. είναι δομημένα σε **επίπεδα (layers)**. Κάθε επίπεδο εκτελεί μια συγκεκριμένη εργασία και συνεργάζεται με τα δύο γειτονικά του. Στα κατώτερα επίπεδα γίνεται η διαχείριση της μνήμης και της επικοινωνίας με τις περιφερειακές συσκευές του υπολογιστή, ενώ στα ανώτερα γίνεται η διαχείριση των προγραμμάτων που εκτελούν οι χρήστες. Σε ένα Λ.Σ. υπάρχουν τα ακόλουθα επίπεδα:

Ο Πυρήνας (Kernel), βρίσκεται πλησιέστερα προς το υλικό και αποτελεί τον ενδιάμεσο για να επιτευχθεί η επικοινωνία των προγραμμάτων με το υλικό. Ο πυρήνας «φορτώνεται» πρώτος στην κύρια μνήμη όταν ξεκινάει ο υπολογιστής. Για παράδειγμα, η διαχείριση συσκευών (device management) και η διαχείριση μνήμης (memory management) αποτελούν ευθύνες του πυρήνα.

Το Σύστημα Αρχείων (File System) διαχειρίζεται τα αρχεία (αποθήκευση, ονοματοδοσία, κτλ.) και φροντίζει για τη διάθεσή τους στους χρήστες.

Ο Φλοιός (Shell) είναι το σύνολο των προγραμμάτων, το οποίο επιτρέπει στο χρήστη και τις εφαρμογές του να επικοινωνεί με το Λ.Σ. Η επικοινωνία γίνεται είτε με απευθείας εντολές είτε μέσω ενός γραφικού περιβάλλοντος διεπαφής (GUI - Graphical User Interface).

Τα λειτουργικά συστήματα χωρίζονται σε τρεις μεγάλες κατηγορίες:

- Λειτουργικά Συστήματα για Προσωπικούς Υπολογιστές, όπως τα Windows, το LINUX, το MAC OS X

- Λειτουργικά Συστήματα για Μεγάλους Υπολογιστές, όπως το Unix, το Solaris
- Λειτουργικά Συστήματα για φορητές συσκευές, όπως το Android, το iOS, το Symbian κ.α.

Λογισμικό εφαρμογών

Παρόλο που το λογισμικό συστήματος είναι ζωτικής σημασίας για τη λειτουργία του υπολογιστή, το λογισμικό εφαρμογών είναι εκείνο που παρέχει στον τελικό χρήστη τις εφαρμογές που εκτελούν συγκεκριμένες λειτουργίες και εξυπηρετούν τις ανάγκες του, ώστε να μεταμορφώσει τον υπολογιστή του σε ένα εξειδικευμένο παραγωγικό εργαλείο.

Μέχρι σήμερα έχει αναπτυχθεί ένα μεγάλο πλήθος εφαρμογών που ταξινομούνται σε δύο μεγάλες κατηγορίες:

Λογισμικό γενικής χρήσης

Αυτές οι εφαρμογές δημιουργούνται για να καλύψουν τις κοινές ανάγκες ενός μεγάλου αριθμού χρηστών, συνήθως αναπτύσσονται από εταιρείες παραγωγής λογισμικού (όταν πρόκειται για εμπορικό λογισμικό) ή ομάδες προγραμματιστών (όταν πρόκειται για ελεύθερο λογισμικό) και έχουν τυποποιημένες δυνατότητες χρήσης. Μερικά παραδείγματα τέτοιων εφαρμογών είναι:

Οι επεξεργαστές κειμένου (Writer, WordPro, Microsoft Word)

Τα υπολογιστικά φύλλα (OpenOffice Calc, Microsoft Excel, Apple Numbers, Gnumeric)

Τα προγράμματα παρουσιάσεων (OpenOffice Impress, Microsoft PowerPoint, Prezi),

Τα συστήματα διαχείρισης βάσεων δεδομένων (MySQL, MariaDB, Oracle database, MongoDB)

Οι επεξεργαστές εικόνας, ήχου, βίντεο (Adobe photoshop, Gimp, PaintShop Pro, Inkscape, Audacity, MovieMaker)

Οι φυλλομετρητές ιστού (Opera, Safari, Internet explorer, Mozilla firefox, Google Chrome)

Το ηλεκτρονικό ταχυδρομείο (Outlook, Thunderbird)

Λογισμικό τηλεδιασκέψεων (Skype, Microsoft teams, Bigbluebutton, Zoom, Cisco webex)

Η χρήση των παραπάνω προγραμμάτων είναι τόσο διαδεδομένη, που συνήθως τα συναντάμε όχι μεμονωμένα αλλά ως «πακέτο λογισμικού» ή «σουίτα εφαρμογών». Ένα πακέτο λογισμικού είναι μια συλλογή προγραμμάτων υπολογιστών (λογισμικό εφαρμογών ή λογισμικό προγραμματισμού) που μοιράζονται κοινές λειτουργίες, έχουν παρόμοια διεπαφή χρήστη και έχουν δυνατότητα εύκολης ανταλλαγής δεδομένων μεταξύ τους. Τα πιο δημοφιλή πακέτα λογισμικού αυτοματισμού γραφείου είναι το LibreOffice και το Microsoft Office. Στην παρακάτω εικόνα βλέπουμε τις εφαρμογές που αποτελούν τη σουίτα λογισμικού LibreOffice. Η σουίτα είναι Ελεύθερο Λογισμικό, και ο πηγαίος κώδικας της εφαρμογής διατίθεται με την άδεια GNU Lesser General Public License (LGPL).

Συστατικό		Περιγραφή
	Writer	Επεξεργαστής κειμένου
	Calc	Υπολογιστικό φύλλο
	Impress	Παρουσιάσεις
	Base	Διαχείριση βάσεων δεδομένων
	Draw	Πρόγραμμα σχεδίασης
	Math	Μαθηματικοί τύποι

Πλεονεκτήματα και μειονεκτήματα της σουίτας εφαρμογών

Πλεονεκτήματα

- Λιγότερο δαπανηρή από την αγορά μεμονωμένων εφαρμογών
- Πανομοιότυπο ή πολύ παρόμοιο GUI
- Σχεδιασμένο για διασύνδεση μεταξύ τους
- Η εγκατάσταση όλων των εφαρμογών γίνεται ταυτόχρονα και όχι μία κάθε φορά
- Μπορεί να περιλαμβάνει πρόσθετες λειτουργίες ή πρόσθετα προγράμματα που δεν είναι διαθέσιμα για αγορά ξεχωριστά.

Μειονεκτήματα

- Δεν χρησιμοποιούνται πάντα όλες οι αγορασμένες λειτουργίες από τον χρήστη
- Καταλαμβάνει σημαντικό χώρο στο δίσκο (bloatware), σε σύγκριση με την αγορά μόνο των απαραίτητων εφαρμογών
- Απαιτεί προσπάθεια για να χρησιμοποιηθούν όλα τα προγράμματα μαζί
- Το κόστος εγκατάστασης και συντήρησης μιας εμπορικής σουίτας εφαρμογών, συμπεριλαμβανομένου του χρόνου που αφιερώνεται στην αναζήτηση των κατάλληλων αναβαθμίσεων λογισμικού, είναι υψηλό.

Λογισμικό ειδικού σκοπού

Στην κατηγορία αυτή ανήκουν εφαρμογές που στοχεύουν στην εξυπηρέτηση συγκεκριμένων αναγκών μιας επιχείρησης ή ενός ιδιώτη. Οι εφαρμογές αυτές αναπτύσσονται συνήθως από το τμήμα πληροφορικής της επιχείρησης ή του οργανισμού ή από κάποια εταιρεία παραγωγής λογισμικού ή από τον ίδιο τον χρήστη.

Μερικά παραδείγματα τέτοιων εφαρμογών είναι: τα εκπαιδευτικά λογισμικά για όλες τις βαθμίδες εκπαίδευσης, τα λογισμικά για επιχειρήσεις (εφαρμογές διαχείρισης αποθήκης, Λογιστηρίου, μισθοδοσίας, ERP κ.α), λογισμικά τραπεζών, εφαρμογές κράτησης θέσεων (ξενοδοχεία, αεροπλάνα κ.τ.λ), εφαρμογές προσομοίωσης (καιρικών φαινομένων, πτήσεων).

Λογισμικό ανάπτυξης

Το λογισμικό ανάπτυξης (ή προγραμματισμού) προορίζεται για προγραμματιστές που παράγουν κώδικα, σε αντίθεση με το λογισμικό εφαρμογών που απευθύνεται σε τελικούς χρήστες και το λογισμικό συστήματος που δημιουργείται για υπολογιστές ή κινητές συσκευές.

Η ανάπτυξη των σύγχρονων εφαρμογών είναι μια απαιτητική και δύσκολη διαδικασία. Οι προγραμματιστές χρησιμοποιούν για την ανάπτυξη, τη δοκιμή, τη βελτίωση, τον εντοπισμό σφαλμάτων και τη συντήρηση μιας

εφαρμογής ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment -IDE) για να διευκολύνεται και να επιταχύνεται η ανάπτυξη μιας εφαρμογής

Αυτά τα εργαλεία συνήθως περιλαμβάνουν κάποιον επεξεργαστή πηγαίου κώδικα (editor), έναν μεταγλωττιστή, εργαλεία αυτόματης παραγωγής κώδικα, αποσφαλματωτή (debugger), βιβλιοθήκες και εργαλεία κατασκευής γραφικών διεπαφής χρήστη (GUI) για τις υπό ανάπτυξη εφαρμογές.

Τα IDE διατίθενται σε διαφορετικούς τύπους και ανάλογα με τις γλώσσες προγραμματισμού που υποστηρίζουν. Έτσι έχουμε:

IDE πολλών γλωσσών

Αυτός ο τύπος ολοκληρωμένου περιβάλλοντος υποστηρίζει την ανάπτυξη εφαρμογών σε περισσότερες από μία γλώσσες προγραμματισμού. Σε αυτή την κατηγορία ανήκουν τα IDE όπως Netbeans, Eclipse, Aptana, Visual Studio.

IDE για ανάπτυξη εφαρμογών για κινητά

Καθώς η αγορά για την ανάπτυξη εφαρμογών για κινητά μεγαλώνει, πολλά εργαλεία προγραμματισμού γίνονται διαθέσιμα στους προγραμματιστές λογισμικού για να δημιουργήσουν αποτελεσματικές εφαρμογές για κινητά. Σε αυτή την κατηγορία ανήκουν τα Android studio, Xcode.

IDE για συγκεκριμένες γλώσσες προγραμματισμού

Αυτός ο τύπος IDE είναι ένα τυπικό αντίθετο του IDE πολλών γλωσσών. Έχουν κατασκευαστεί ειδικά για να υποστηρίζουν προγραμματιστές που εργάζονται σε μία μόνο γλώσσα προγραμματισμού. Σε αυτή την κατηγορία ανήκουν τα Jcreator για Java, Idle για Python και CodeLite για C++.

IDE που βασίζονται στο σύννεφο (cloud)

Αυτός ο τύπος IDE έχει ένα πρόσθετο πλεονέκτημα ότι οι προγραμματιστές μπορούν να έχουν πρόσβαση σε αυτό από οπουδήποτε στον κόσμο και όχι μόνο από τον υπολογιστή στην εργασία τους. Σε αυτή την κατηγορία ανήκουν τα Cloud 9 IDE που αναπτύχθηκε από την Amazon Web Services, Codeanywhere, Heroku.

Γλώσσες προγραμματισμού

Όπως γνωρίζουμε ήδη οι υπολογιστές δεν μπορούν να κατανοήσουν και να εκτελέσουν εντολές διατυπωμένες στις ανθρώπινες γλώσσες όπως τα αγγλικά, αραβικά ή κινέζικα, και παρόλο που οι άνθρωποι τεχνικά μπορούν να μάθουν τη βασική γλώσσα των υπολογιστών (δυναμική γλώσσα ή γλώσσα μηχανής), σχεδόν κανείς δεν το κάνει. Για αυτόν τον λόγο χρειαζόμαστε έναν ενδιάμεσο τρόπο επικοινωνίας, τις καλούμενες γλώσσες προγραμματισμού.

Δίνοντας έναν ορισμό θα μπορούσαμε να πούμε ότι γλώσσα προγραμματισμού είναι μια τεχνητή γλώσσα που μπορεί να χρησιμοποιηθεί για τον έλεγχο μιας μηχανής, συνήθως ενός υπολογιστή. Οι γλώσσες προγραμματισμού (όπως άλλωστε και οι ανθρώπινες γλώσσες) ορίζονται από ένα σύνολο συντακτικών και εννοιολογικών κανόνων, που ορίζουν τη δομή και το νόημα, αντίστοιχα, των προτάσεων της γλώσσας.

Ο προγραμματισμός ενός υπολογιστή μπορεί να γίνει σε πολλά επίπεδα. Στο χαμηλότερο επίπεδο βρίσκονται τα ηλεκτρονικά κυκλώματα του υπολογιστή που μπορούν να έχουν δύο μόνο καταστάσεις, 0 και 1. Έτσι, στο επίπεδο αυτό ο προγραμματισμός μπορεί να γίνει μόνο με τη χρήση μιας σειράς δυαδικών αριθμών. Η «γλώσσα» αυτή των δυαδικών αριθμών λέγεται **γλώσσα μηχανής** και είναι η μόνη γλώσσα που κατανοεί απευθείας το υλικό του υπολογιστή. Τα προγράμματα σε γλώσσα μηχανής προορίζονται μόνο για τον επεξεργαστή για τον οποίο δημιουργήθηκαν με αποτέλεσμα κάθε επεξεργαστής να έχει τη δική του γλώσσα μηχανής.

Επειδή η ανάπτυξη προγράμματος από τους προγραμματιστές σε γλώσσα μηχανής είναι ιδιαίτερα επίπονη και χρονοβόρα εργασία και απαιτεί βαθιά γνώση του υλικού και της αρχιτεκτονικής του υπολογιστή

αναζητήθηκαν τρόποι να γράφονται τα προγράμματα σε γλώσσα πιο προσιτή στον άνθρωπο. Οι πρώτες γλώσσες που αναπτύχθηκαν ήταν οι συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου,

Μια **συμβολική γλώσσα (assembly language)** είναι μια χαμηλού επιπέδου γλώσσα προγραμματισμού, δηλαδή μια γλώσσα πολύ κοντά στη γλώσσα μηχανής και στο υλικό του υπολογιστή. Ένα πρόγραμμα σε συμβολική γλώσσα αντιστοιχίζει κάθε εντολή της γλώσσας μηχανής με μια μνημονική λέξη η οποία θυμίζει το σκοπό της εντολής

Για παράδειγμα ένας επεξεργαστής της αρχιτεκτονικής x86 θα καταλάβει την εντολή σε γλώσσα μηχανής:

```
10110000 01100001
```

Ένας προγραμματιστής όμως είναι πιο εύκολο να θυμάται την ισοδύναμη συμβολική αναπαράσταση, για παράδειγμα μια τυπική εντολή σε συμβολική γλώσσα είναι η εξής:

```
mov al, 061h
```

που είναι συντομογραφία της αγγλικής λέξης move ("μετακίνησε"). Η εντολή αυτή μετακινεί τη δεκαεξαδική τιμή 61 στον καταχωρητή με το όνομα "al".

Η μετατροπή ενός προγράμματος από συμβολική γλώσσα σε γλώσσα μηχανής γίνεται από ένα ειδικό πρόγραμμα, τον συμβολομεταφραστή (assembler) και το αντίστροφο γίνεται από έναν αποσυμβολομεταφραστή (disassembler).

Οι συμβολικές γλώσσες διευκόλυναν σε ένα βαθμό τους προγραμματιστές στην ανάπτυξη προγραμμάτων όμως είχαν ένα σοβαρό μειονέκτημα που ήταν η πλήρης εξάρτησή τους από τον επεξεργαστή για τον οποίον είχαν δημιουργηθεί με αποτέλεσμα ένα πρόγραμμα σε συμβολική γλώσσα να μην μπορεί να εκτελεστεί σε άλλον επεξεργαστή.



Ένα από τα σημαντικά πλεονεκτήματα της assembly είναι ότι μπορούν να χρησιμοποιηθούν μικρές ποσότητες κώδικα γλώσσας assembly μέσα σε μια γλώσσα υψηλού επιπέδου όπως η C ή JAVA προκειμένου να επιτύχουμε καλύτερες επιδόσεις του προγράμματος. Για παράδειγμα σήμερα, λίγο λιγότερο από το 2% της έκδοσης 4.9 του πηγαίου κώδικα του πυρήνα Linux είναι γραμμένο σε γλώσσα assembly ενώ περισσότερο από το 97% είναι γραμμένο σε γλώσσα C.

Η ανάγκη για να υπάρχει η δυνατότητα ανάπτυξης προγραμμάτων με πιο «φιλικό» και οικείο για τον άνθρωπο τρόπο, οδήγησε στα τέλη της δεκαετίας του '50 στην εμφάνιση των πρώτων **γλωσσών προγραμματισμού υψηλού επιπέδου (high level languages)**. Ονομάστηκαν υψηλού επιπέδου επειδή οι γλώσσες αυτές είναι πιο φιλικές στον προγραμματιστή σε αντίθεση με τη γλώσσα μηχανής και την assembly που είναι πιο φιλικές στη μηχανή και ονομάζονται γλώσσες χαμηλού επιπέδου. Στις γλώσσες υψηλού επιπέδου οι εντολές είναι ένα σύνολο λέξεων, συμβόλων και προτάσεων της αγγλικής γλώσσας που ακολουθούν αυστηρούς γραμματικούς και συντακτικούς κανόνες που πρέπει να γνωρίζει ο προγραμματιστής. Κάθε εντολή σε μια γλώσσα υψηλού επιπέδου μπορεί να αντιστοιχεί σε ομάδα εντολών της γλώσσας μηχανής. Ένα άλλο πλεονέκτημα αυτών των γλωσσών είναι ότι, για πρώτη φορά, η γλώσσα είναι ανεξάρτητη από τον τύπο του υπολογιστή (επεξεργαστή) στον οποίο θα εκτελεστεί το πρόγραμμα. Έτσι, ένα πρόγραμμα γραμμένο σε μια γλώσσα υψηλού επιπέδου μπορεί να εκτελεστεί σε οποιοδήποτε υπολογιστή. Για να εκτελεστεί όμως από τον υπολογιστή ένα πρόγραμμα γραμμένο σε γλώσσα υψηλού επιπέδου, χρειάζεται να μετατραπεί στο αντίστοιχο πρόγραμμα σε γλώσσα μηχανής. Η μετατροπή αυτή γίνεται με τη βοήθεια ειδικών προγραμμάτων που συνοδεύουν τη γλώσσα προγραμματισμού και ονομάζονται μεταφραστές (compilers) ή διερμηνείς (interpreters).

Οι πρώτες γλώσσες υψηλού επιπέδου που αναπτύχθηκαν χρονολογούνται από τη δεκαετία του 1950. Οι σημαντικότερες γλώσσες αυτής της εποχής είναι η FORTRAN που αναπτύχθηκε για την επίλυση

επιστημονικών προβλημάτων, η COBOL που αναπτύχθηκε για εμπορικές εφαρμογές (π.χ. εφαρμογή μισθοδοσίας), η BASIC που αναπτύχθηκε σαν γλώσσα γενικού σκοπού για εκπαίδευση αρχαρίων και η ALGOL, μια γλώσσα που επηρέασε σημαντικά τις επόμενες γλώσσες προγραμματισμού, που χρησιμοποιήθηκε κυρίως από ερευνητές της επιστήμης υπολογιστών στις Ηνωμένες Πολιτείες και στην Ευρώπη. Άλλες γλώσσες που αναπτύχθηκαν κατά τη διάρκεια της δεκαετίας του 1970 και εισήγαγαν νέα χαρακτηριστικά που επηρέασαν την ανάπτυξη των μεταγενέστερων γλωσσών είναι:

- PASCAL που θεωρείται γλώσσα γενικής χρήσης. Αρχικά δεν διέθετε εξελιγμένο σύστημα χειρισμού αρχείων, (και γι' αυτό δεν εκτόπισε την COBOL στις εμπορικές εφαρμογές), αλλά, όπως οι περισσότερες γλώσσες, πέρασε από πολλές εκδόσεις που κάθε μια βελτίωνε την προηγούμενη. Διάφορες εκδόσεις της Pascal είναι σήμερα σε χρήση.
- C που αναπτύχθηκε πειραματικά στα εργαστήρια της εταιρείας BELL και γνώρισε μεγάλη επιτυχία λόγω της χρήσης της στην ανάπτυξη του λειτουργικού συστήματος UNIX. Η C συγκαταλέγεται πλέον στις πιο ευρέως χρησιμοποιούμενες γλώσσες προγραμματισμού όλων των εποχών και πολλές νεότερες γλώσσες έχουν επηρεαστεί άμεσα ή έμμεσα από αυτήν, συμπεριλαμβανομένων των C++, C#, D, Go, Java, JavaScript, Limbo, LPC, Perl, PHP, Python.
- Prolog ήταν το σημείο εκκίνησης μιας νέας οικογένειας γλωσσών: των γλωσσών λογικού προγραμματισμού. Χρησιμοποιείται κυρίως στον τομέα της τεχνητής νοημοσύνης.
- SQL αναπτύχθηκε στην IBM είναι μία γλώσσα προγραμματισμού που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων.

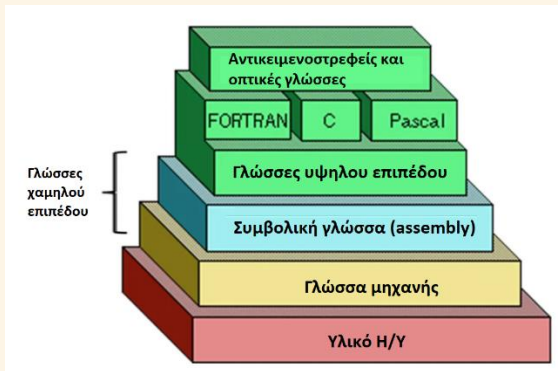
Στα τέλη της δεκαετίας του 1970 και στις αρχές της δεκαετίας του 1980, αναπτύχθηκε μια νέα μέθοδος προγραμματισμού που έγινε γνωστή ως Αντικειμενοστρεφής Προγραμματισμός ή OOP. Κεντρική ιδέα στον αντικειμενοστρεφή προγραμματισμό είναι η κλάση (class), μία αυτοτελής και αφαιρετική αναπαράσταση κάποιας κατηγορίας αντικειμένων, είτε φυσικών αντικειμένων του πραγματικού κόσμου είτε νοητών, εννοιολογικών αντικειμένων, σε ένα περιβάλλον προγραμματισμού. Τα αντικείμενα (objects) ομαδοποιούνται με βάση τα δεδομένα (ιδιότητες) που προσδιορίζουν τα χαρακτηριστικά τους και τις ενέργειες (μέθοδοι) που καθορίζουν τη συμπεριφορά τους. Οι πιο δημοφιλείς γλώσσες που αναπτύχθηκαν με αυτή τη μεθοδολογία είναι:

- C++ , ουσιαστικά είναι η γλώσσα C εφοδιασμένη με αντικειμενοστραφή χαρακτηριστικά.
- Python, ο κύριος στόχος της είναι η αναγνωσιμότητα του κώδικά της και η ευκολία χρήσης της. Το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα από ό,τι θα ήταν δυνατόν σε γλώσσες όπως η C++ ή η Java. Διακρίνεται λόγω του ότι έχει πολλές βιβλιοθήκες που διευκολύνουν ιδιαίτερα αρκετές συνηθισμένες εργασίες και για την ταχύτητα εκμάθησής της. Η Python αναπτύσσεται ως ανοιχτό λογισμικό (open source) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation.
- Java, σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems ως εργαλείο για την ανάπτυξη λογισμικού σε ψηφιακές κινητές συσκευές. 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 χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Η Java ακολουθεί μία ανοδική πορεία και είναι πλέον μία από τις πιο δημοφιλείς γλώσσες στον χώρο της πληροφορικής. Στις 13 Νοεμβρίου του 2006 η Java έγινε πλέον μια γλώσσα ανοιχτού κώδικα

Άλλες αντικειμενοστρεφείς γλώσσες προγραμματισμού είναι η JavaScript, η PHP, η Ruby, η C#.

Η εμφάνιση των γραφικών περιβαλλόντων εργασίας (στα τέλη της δεκαετίας του 1980 και τη δεκαετία του 1990) άλλαξε προς το καλύτερο τον τρόπο επικοινωνίας ανθρώπου και μηχανής, με αποτέλεσμα πολλές γλώσσες προγραμματισμού να επιτρέπουν στο χρήστη τη δημιουργία προγραμμάτων μέσω γραφικών αντικειμένων, αντί της πληκτρολόγησης του κειμένου που αντιστοιχεί σε εντολές. Οι γλώσσες αυτές ονομάστηκαν οπτικές γλώσσες προγραμματισμού (visual programming language, συντ. VPL). Έτσι γλώσσες όπως η BASIC, η C++, η PASCAL εξελίχθηκαν (Visual Basic, Visual C++, Delphi) ώστε να διαθέτουν και οπτικό περιβάλλον προγραμματισμού. Αναπτύχθηκαν όμως και γλώσσες όπως η SCRATCH του MIT σχεδιασμένη για παιδιά και σχολική εκπαίδευση, η Google App Inventor, εργαλείο δημιουργίας εφαρμογών για το Google Android, η LabVIEW, γραφική γλώσσα για μηχανικούς και επιστήμονες κ.α.

Τα επίπεδα των γλωσσών προγραμματισμού φαίνονται σχηματικά στην παρακάτω εικόνα.



Οι γλώσσες προγραμματισμού έχουν εξελιχθεί σημαντικά και θα συνεχίσουν έτσι για πολλά χρόνια ακόμα. Οι πρώτες γλώσσες χαρακτηρίζονταν από το γεγονός ότι προορίζονταν για ένα μόνο σκοπό, ενώ οι σημερινές γλώσσες μπορούν να χρησιμοποιηθούν για σχεδόν κάθε σκοπό. Καθώς η τεχνολογία εξελίσσεται και οι απαιτήσεις αλλάζουν, οι γλώσσες προγραμματισμού πρέπει να εξελίσσονται για να παραμένουν αποτελεσματικές.



Δραστηριότητες

1. Ποια είναι η διαφορά μεταξύ ενός λογισμικού εφαρμογών και ενός λειτουργικού συστήματος.
2. Αναφέρετε τα επίπεδα του λειτουργικού συστήματος.
3. Ποια είναι η διαφορά μεταξύ της γλώσσας μηχανής και της συμβολικής γλώσσας (assembly).
4. Ποια γλώσσα σχετίζεται άμεσα με το υλικό του υπολογιστή και είναι κατανοητή από αυτό.
5. Χρησιμοποιώντας το μαθησιακό αντικείμενο Φωτόδεντρο: ΤΟ ΛΟΓΙΣΜΙΚΟ ΤΟΥ ΥΠΟΛΟΓΙΣΤΗ (<http://photodentro.edu.gr/v/item/ugc/8525/1847>) εξερευνήστε το διαδραστικό βιβλίο για την εξοικείωσή σας με το λογισμικό του υπολογιστή.