



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Αγρονόμων Τοπογράφων Μηχανικών

Εισαγωγή στην πληροφορική

Βασίλειος Βεσκούκης

Δρ. Ηλεκτρολόγος Μηχανικός &
Μηχανικός Υπολογιστών ΕΜΠ
v.vescoukis@cs.ntua.gr

Ρωμύλος Κορακίτης

Αστροφυσικός
Αναπλ. Καθηγητής ΕΜΠ
romylos@survey.ntua.gr

*Δομή των υπολογιστών, υλικό και λογισμικό
Προγραμματισμός και γλώσσες προγραμματισμού
Η έννοια του αλγορίθμου*

Πληροφοριακά συστήματα

Τρεις βασικές λειτουργίες

ΕΙΣΟΔΟΣ – ΕΠΕΞΕΡΓΑΣΙΑ – ΕΞΟΔΟΣ

Τρία ερωτήματα

- Ποια αποτελέσματα θα λαμβάνονται;
- Ποια είναι τα διαθέσιμα δεδομένα εισόδου;
- Ποιες πράξεις/μετασχηματισμοί θα μετατρέψουν τα δεδομένα εισόδου στα αιτούμενα αποτελέσματα;

Σε ακολουθία με τα παραπάνω, ένας Η/Υ πρέπει

- Να μπορεί να δέχεται δεδομένα (είσοδος)
- Να μπορεί να πραγματοποιεί υπολογισμούς (επεξεργασία)
- Να μπορεί να εμφανίζει αποτελέσματα (έξοδος)

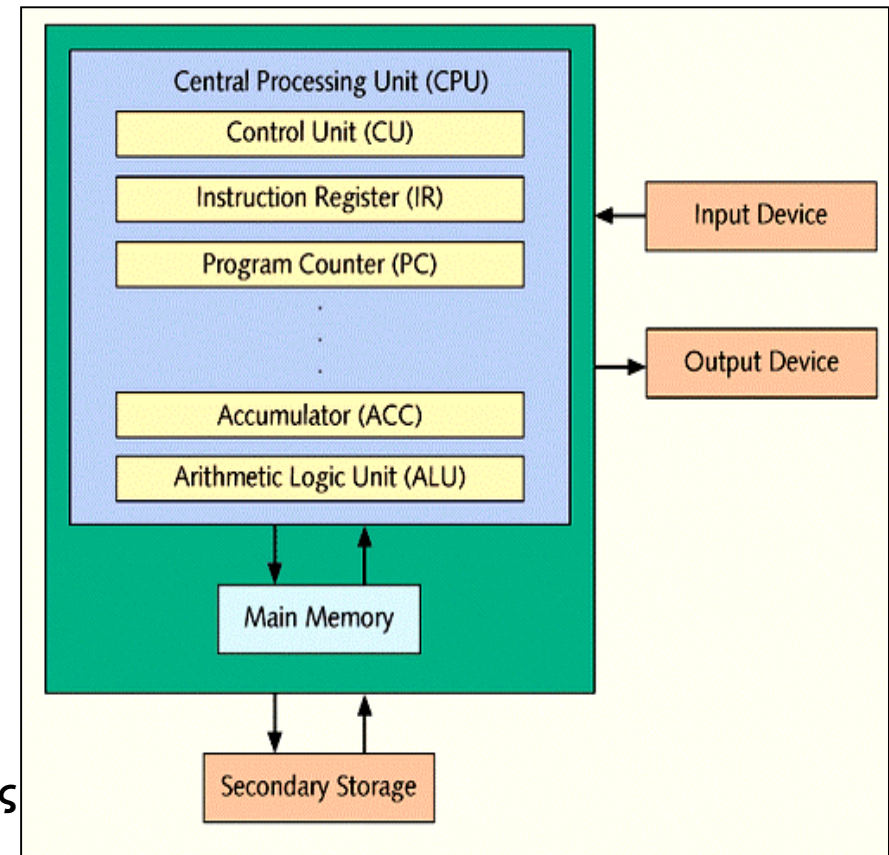
Επιπλέον, διαισθητικά απαιτητό, όχι όμως ρητά εκπεφρασμένο μέχρι τώρα:

- Να μπορεί να αποθηκεύει δεδομένα (με προσωρινό ή μόνιμο τρόπο)
- Να ανακτά αποθηκευμένα δεδομένα

Δομή ηλεκτρονικού υπολογιστή - Υλικό

Μια γενική διάκριση συστατικών που συνθέτουν έναν Η/Υ (πόροι [resources])

- Μονάδα εισόδου (Input unit)
 - Έλεγχος συσκευών εισόδου
- Μονάδα εξόδου (Output unit)
 - Έλεγχος συσκευών εξόδου
- Μονάδα προσωρινής μνήμη (Memory unit)
 - Σχετικά μικρή χωρητικότητα αλλά μεγάλη ταχύτητα, τα προγράμματα που «τρέχουν» βρίσκονται εδώ
- Μονάδα κεντρικής επεξεργασίας (CPU)
 - Εκτέλεση εντολών και αριθμητικών υπολογισμών
- Μονάδα ελέγχου μόνιμης αποθήκευσης (Secondary storage unit)
 - Διαχείριση μόνιμης, σχετικά φτηνής και μεγάλης αποθηκευτικής ικανότητας για τη μόνιμη αποθήκευση δεδομένων και μη-ενεργών προγραμμάτων



Συσκευές εισόδου και εξόδου

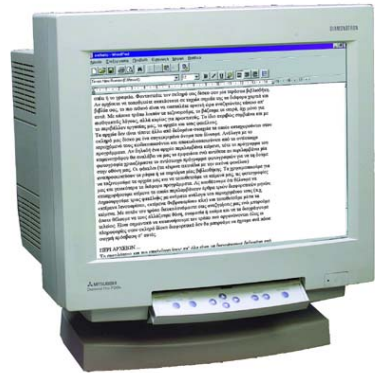
Σημερινές συσκευές εισόδου

- Το πληκτρολόγιο
- Το ποντίκι
- Το scanner
- Το μικρόφωνο
- Την κάμερα...
- Ένα ειδικό καταγραφικό μηχάνημα

Σημερινές συσκευές εξόδου

- Η οθόνη
- Ο εκτυπωτής
- Τα ηχεία
- Οι plotter

Μερικές συσκευές εισόδου και εξόδου



Μνήμη προσωρινή και μόνιμη

Προσωρινή μνήμη

- Είναι διαθέσιμη όσο λειτουργεί ο υπολογιστής
- Επικοινωνεί άμεσα με την κεντρική μονάδα επεξεργασίας

Είδη προσωρινής μνήμης

- RAM (Random Access Memory): αποθήκευση προγραμμάτων του χρήστη
- ROM (Read Only Memory): αποθήκευση μόνιμων προγραμμάτων που χρειάζεται ο ίδιος ο υπολογιστής για να λειτουργήσει
 - PROM (Programmable Read Only Memory)
 - EPROM (Erasable PROM)

Στην μνήμη ROM

- αποθηκεύονται από τον κατασκευαστή του υπολογιστή τα βασικά προγράμματα τα οποία δεν εξυπηρετούν το χρήστη, αλλά καθιστούν δυνατή την λειτουργία του υπολογιστή (BIOS: Basic Input-Output system)



Οργάνωση δεδομένων στο επίπεδο του υλικού της μνήμης

Μνήμη:

μια ακολουθία θέσεων στις οποίες αποθηκεύονται bytes

Διεύθυνση θέσης μνήμης:

η μοναδική διάκριση κάθε θέσης από τις υπόλοιπες

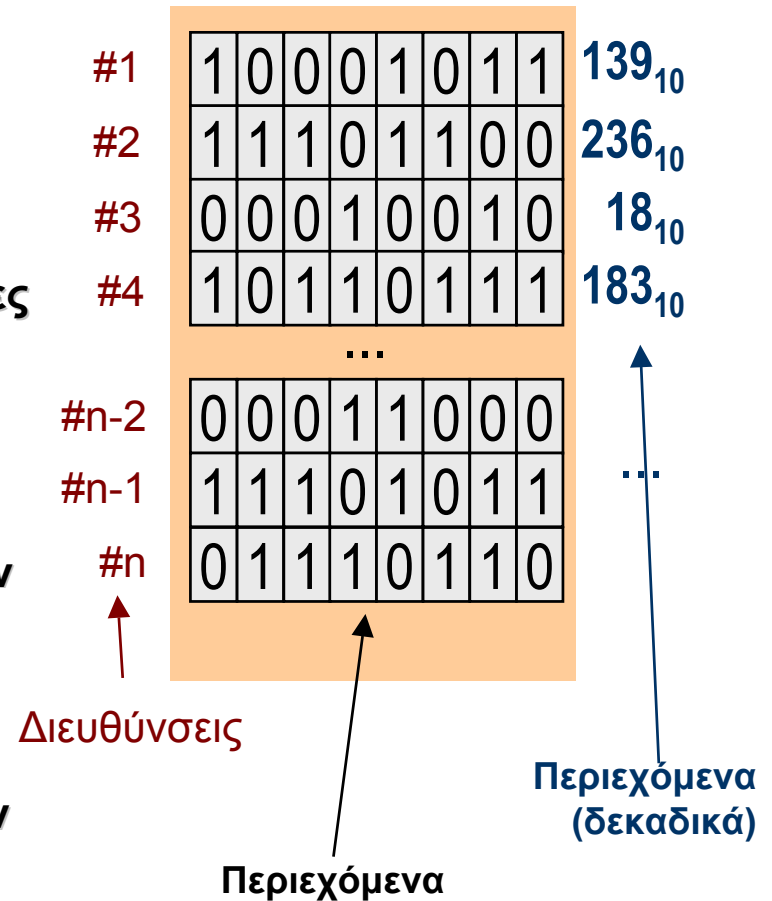
Περιεχόμενο θέσης μνήμης:

η χρήσιμη κατάσταση μιας θέσης μνήμης

Η **μέγιστη τιμή** που μπορεί να αποθηκευτεί σε μια θέση μνήμης εξαρτάται από το εύρος αυτής και την κωδικοποίηση που χρησιμοποιείται (βλ. προηγούμενα)

Το **πλήθος** των θέσεων μνήμης που μπορεί να δει ένας Η/Υ εξαρτάται από το εύρος των διευθύνσεων

- 8 bit : 256 θέσεις
- 16 bit : 65536 θέσεις ...



Κεντρική μονάδα επεξεργασίας

CPU: Central Processing Unit

- Έλεγχος λειτουργίας του υπολογιστή (Control Unit)
- Πραγματοποίηση υπολογισμών (ALU - Arithmetic Logic Unit)

Κύκλοι λειτουργίας

- Κύκλος εντολής
 - Ανάκληση εντολής από τη μνήμη
 - Αποκωδικοποίηση εντολής
- Κύκλος εκτέλεσης
 - Εκτέλεση εντολής
 - Διάθεση του αποτελέσματος

Οι σημερινοί επεξεργαστές: "Πανίσχυροι", πολύπλοκοι, "ευφυείς"



Μόνιμη αποθήκευση

Φύλαξη δεδομένων για μεγάλο χρονικό διάστημα

Τεχνολογίες αποθήκευσης

- Μαγνητική αποθήκευση
 - Σκληροί δίσκοι
 - Δισκέττες, φορητά μέσα (DAT, zip, κ.ά.)
- Οπτική
 - CDROM, DVD
- Μικτή



Οργάνωση αποθήκευσης

- Σειριακή (ταινίες, DAT)
- Τυχαίας προσπέλασης

Εκδοχές

- Εγγραφή και ανάγνωση
- Μία εγγραφή, πολλές αναγνώσεις
- Μόνο ανάγνωση



Μονάδες αποθήκευσης και ταχύτητας

Αίσθηση μεγεθών: τι παριστάνουμε με...

- **1 bit:** 0 ή 1, "ναι" ή "όχι", "αληθές" ή "ψευδές"
- **1 byte:** "a", "b", ..., "A", "B", ..., "α", "β", ..., "Α", "Β", ...,
- **1 byte:** 0, 1, 2, 3, ... 255 (256 θετικοί ακέραιοι αριθμοί από το 0 έως το 2^8-1)
- **16-bit word:** 0, 1, 2, ... 65.535 ($=2^{16}-1$) [τάξη μεγέθους: 10^4]
- **32-bit word:** 0, 1, 2, ..., 4.294.967.295 ($=2^{32}-1$) [τάξη μεγέθους: 10^9]
- **64-bit word:** 0, 1, 2, ..., 18.446.744.073.709.599.999 ($=2^{64}-1$) [τάξη μεγέθους: 10^{19}]

Σημείωση: μη προσημασμένοι θετικοί ακέραιοι

- 1000 ονοματεπώνυμα: 25-40 KB, 1000 πλήρεις διευθύνσεις: 40-100 KB
- 1 λεπτό μουσικής σε μορφή CD απαιτεί 10 MB
- 1 λεπτό MP3 απαιτεί περίπου 1 MB
- 1 ώρα ψηφιακού video (DV) απαιτεί 14 GB
- 1 ώρα video MPEG2 (DVD) απαιτεί περίπου 2 GB

Αίσθηση μεγεθών: πόσο γρήγορα τρέχει στα 2 GHz

- Ένας "κύκλος μηχανής" σε $0,5 \times 10^{-9}$ sec
- **Προσοχή:** "ένας κύκλος μηχανής" δεν σημαίνει υποχρεωτικά και "μία εντολή"

Το σύγχρονο computing

Σχετικά λίγα χρόνια πριν...

- Κεντρικοί υπολογιστές με τερματικά
- Δύσχρηστες μονάδες εισόδου – εξόδου
- Πρόσβαση ακριβή και δύσκολη

Σήμερα

- Προσωπικοί υπολογιστές (personal computers)
 - Αυτονομία και ανεξαρτησία
 - Φτηνοί, πανίσχυροι, πολύ μικροί
 - Υπάρχουν χωρίς να το ξέρουμε
- Δίκτυα, κατανεμημένα συστήματα (distributed computing)
 - Σε επίπεδο εταιρίας/οργανισμού
 - Σε παγκόσμιο επίπεδο (internet)
 - Πρόσβαση σε πληροφορία και κοινή χρήση υπολογιστικών πόρων
- Client / server, multi-tier computing
 - Διάταξη υλικού και λογισμικού ώστε να παρέχονται υπηρεσίες υπολογισμών και διάθεσης δεδομένων πάνω από ένα οποιασδήποτε εμβέλειας δίκτυο
 - Προβλήματα σχεδίασης και περιορισμοί ασφάλειας, ταχύτητας, κλπ είναι εκτός του πεδίου της δικής μας αναφοράς

Επικοινωνία ανθρώπου – υπολογιστή

Παρελθόν

- Ενδεικτικές λυχνίες και βύσματα
- Διάτρητες κάρτες
- Τερματικά και εκτυπωτές γραμμής

Παρόν

- Πληκτρολόγιο – ποντίκι
- Τερματικά οθόνης
- Περιβάλλοντα με γραφικά
 - Mac OS, X11, MS Windows
- Οθόνες αφής, έγχρωμοι εκτυπωτές
- Πολυμέσα, DVD, MP3, ...

Μέλλον

- "Εξυπνες" κάρτες
- Βιομετρικά
- Σύνθεση φωνής
- Αναγνώριση γραφής, φωνής, εικόνας και γλώσσας
- Αυτόματη μετάφραση

Λογισμικό

Διάφοροι ορισμοί:

- Το συστατικό των υπολογιστών που δεν είναι απτό
- Η περιγραφή της λύσης ενός προβλήματος και γενικά της εκτέλεσης μιας εργασίας, με χρήση εντολών Η/Υ
- Ένα σύνολο εντολών που φέρουν κάποιο αποτέλεσμα
- Προγράμματα

Διάκριση του λογισμικού

- **Λογισμικό συστήματος (systems software)**
Πρόγραμμα που φέρει τον υπολογιστή σε λειτουργία και τον καθιστά διαθέσιμο στον άνθρωπο
 - Λειτουργικά συστήματα (operating systems)
 - Προγράμματα μόνιμα αποθηκευμένα στη ROM του υπολογιστή και των συσκευών του (BIOS, firmware)
- **Λογισμικό εφαρμογών (application software)**
Προγράμματα που κάνουν χρήσιμες στον άνθρωπο εργασίες

Γενικά για λειτουργικά συστήματα

Ένα λειτουργικό σύστημα είναι λογισμικό το οποίο ελέγχει πλήρως τον υπολογιστή:

- Διαχείριση πόρων (CPU, μνήμη, I/O, ...)
- Χρονοπρογραμματισμός εκτέλεσης προγραμμάτων
- Προστασία μνήμης, έλεγχος πρόσβασης, απόδοση πόρων και προτεραιοτήτων
- Επικοινωνία με τον "έξω κόσμο"
 - "Παράθυρα", ποντίκι, πληκτρολόγιο
 - Τερματικό κειμένου με γραμμή εντολών
 - Διαχείριση άλλων περιφερειακών συσκευών, ανάλογα με το είδος του Η/Υ

Το λειτουργικό σύστημα φροντίζει για

- Την "ταυτόχρονη" εκτέλεση πολλών προγραμμάτων
- Τη "λογιστική" των χρηστών, τα δικαιώματα, την καταγραφή γεγονότων (logs)
- Την καλύτερη δυνατή αξιοποίηση των πόρων του υπολογιστή

Συνοπτικά ιστορικά στοιχεία

- Μέχρι το 1955: Δεν υπήρχε ανάγκη για λειτουργικό σύστημα (προγραμματισμός με καλωδιώσεις κλπ)
- Μέχρι το 1965: Προγραμματισμός με κάρτες και σειριακή εκτέλεση των εντολών και των διεργασιών
- 1966 : Το πρώτο πραγματικό ΛΣ: το OS/360 για τον IBM 360. Είχε αρκετά προβλήματα αλλά εισήγαγε τον πολυπρογραμματισμό (multitasking ή multiprogramming) και την ταυτόχρονη χρήση των μονάδων I/O (spooling)
- 1976 : Ο πρόγονος του UNIX: το MULTICS, που προσέφερε επιπλέον δρομολόγηση διεργασιών με καταμερισμό χρόνου (time sharing)
- 1978 : Ο πρόγονος των ΛΣ για μικροϋπολογιστές: το CP/M
- 1982 – 2002 : MS-DOS, MacOS, MS-Windows, Linux κλπ.

Οι κύριες ευθύνες του λειτουργικού συστήματος

- Διαχείριση διεργασιών (process management)
- Διαχείριση κύριας μνήμης (main memory management)
- Διαχείριση δευτερεύουσας μνήμης (secondary memory management)
- Διαχείριση συστήματος εισόδου / εξόδου (I/O system management)
- Διαχείριση συστήματος αρχείων (file system management)
- Διαχείριση πρόσβασης πόρων (resource access management)
- Διαχείριση επικοινωνιών και δικτύου (networking)
- Υπηρεσία πρόσβασης χρηστών (user access facility)

Οι παραπάνω εργασίες υλοποιούνται με την εκτέλεση των υπηρεσιών συστήματος (system services – system calls), που φροντίζουν, επιπλέον, την σταθερότητα του συστήματος (επαναφορά από συνθήκες σφάλματος), την ασφαλή και απομονωμένη εκτέλεση των διεργασιών και την «λογιστική» των διαφόρων διεργασιών και χρηστών.

Διαχείριση διεργασιών

Η Διαχείριση διεργασιών αναφέρεται στον έλεγχο της εκτέλεσης των διαφόρων προγραμμάτων, που είτε είναι τμήματα του λειτουργικού συστήματος είτε είναι εξωτερικά προγράμματα, ενεργοποιούμενα από τον χρήστη.

Ο όρος **διεργασία (process)** αναφέρεται σε ένα αυτόνομο πρόγραμμα που βρίσκεται στην κύρια μνήμη και εκτελείται, δηλαδή χρησιμοποιεί την CPU.

Το ΛΣ δημιουργεί μια διεργασία αποδίδοντας ένα χώρο στην κύρια μνήμη (για την τοποθέτηση του προγράμματος) και προσθέτοντάς την στην λίστα των διεργασιών προς εκτέλεση. Αντίστροφες ενέργειες συμβαίνουν κατά την διαγραφή μιας διεργασίας που ολοκληρώθηκε.

Προϋπόθεση για την συνύπαρξη ενός (μεγάλου) αριθμού διεργασιών είναι η δυνατότητα καταμερισμού της λειτουργίας της CPU. Συνήθως αυτό επιτυγχάνεται με **καταμερισμό χρόνου (time sharing)**.

Έτσι, κάθε διεργασία μπορεί να βρίσκεται σε μια από τις καταστάσεις: **τρέχουσα** , **έτοιμη** , **δεσμευμένη** (σε αναμονή για εκχώρηση πόρου ή διαδικασίας εισόδου / εξόδου)

Μια **τρέχουσα** διεργασία χρησιμοποιεί την CPU για ένα **χρονοτεμάχιο (time slice)** , οπότε μεταβαίνει στην κατάσταση “**έτοιμη**” και περιμένει να επιλεγεί ξανά προς εκτέλεση για ένα ακόμα χρονοτεμάχιο.

Διαχείριση μνήμης

- Τι βρίσκεται στην (προσωρινή) μνήμη του υπολογιστή;

Η διαχείριση της μνήμης έχει τρεις βασικές συνιστώσες:

- Εκχώρηση (allocation) και απελευθέρωση (deallocation) μιας περιοχής μνήμης για κάθε διεργασία
- Βέλτιστη χρησιμοποίηση της διαθέσιμης μνήμης, πραγματικής ή εικονικής. Για το σκοπό αυτό υπάρχουν τεχνικές όπως:
 - ⇒ Διάκριση διευθύνσεων μνήμης από διευθύνσεις προγράμματος (σε λίγο)
 - ⇒ Δυναμική (dynamic) εκχώρηση της μνήμης
 - ⇒ Ανταλλαγή (swapping) περιεχομένων κύριας – δευτερεύουσας μνήμης
 - ⇒ Σελιδοποίηση (paging) και χρήση εικονικής μνήμης (virtual memory)

Τελικός στόχος όλων των ανωτέρω είναι να εξασφαλισθεί ότι υπάρχουν πάντοτε αρκετές διεργασίες **έτοιμες** προς εκτέλεση από την CPU.

- προστασία (protection) της περιοχής μνήμης κάθε διεργασίας, ώστε να μην μπορεί να προσπελαστεί από άλλες διεργασίες

Διαχείριση συστήματος αρχείων

Τι είναι το αρχείο (file) ;

Το σύστημα αρχείων (file system) περιέχει τις απαραίτητες πληροφορίες για τον τρόπο που έχει οργανωθεί η αποθήκευση των αρχείων στο μόνιμο μέσο αποθήκευσης. Συνεπώς, υπάρχουν κατάλογοι (directories) που περιέχουν πληροφορίες για μια ομάδα αρχείων, όπως το όνομα, η θέση στο μέσο αποθήκευσης και το μέγεθος, η ημερομηνία δημιουργίας, ο ιδιοκτήτης του αρχείου κλπ.

Η πληροφορίες κάθε καταλόγου απαρτίζουν, πρακτικά, ένα αρχείο. Επομένως, ένας κατάλογος μπορεί να περιέχει υποκαταλόγους (sub-directories) σε πολλά επίπεδα (ιεραρχική δομή ή δομή δένδρου – tree structure)

Βασικές εργασίες της διαχείρισης αρχείων είναι:

- δημιουργία και διαγραφή των αρχείων (και καταλόγων)
- προσπέλαση στο περιεχόμενο των αρχείων
- προστασία των αρχείων από μη εξουσιοδοτημένη προσπέλαση
- προστασία των αρχείων από φθορά λόγω βλάβης του υλικού ή του λογισμικού.

Προγραμματισμός ηλεκτρονικών υπολογιστών

Ένας Η/Υ "καταλαβαίνει" μόνο ακολουθίες από ψηφία, που εκφράζουν

- Εντολές
- Δεδομένα

Πρακτικά αδύνατο για τον άνθρωπο να χρησιμοποιήσει αυτή την αναπαράσταση

Επίπεδα προγραμματισμού

- Γλώσσα μηχανής: άμεση αναφορά ακολουθιών από "0" και "1"
- Assembly: χρήση μνημονικών ονομάτων για τις εντολές γλώσσας μηχανής
- Γλώσσες υψηλού επιπέδου: εντολές και σύνταξη που ομοιάζουν στην αγγλική
- Γλώσσες τέταρτης γενιάς: περιγραφή των αιτούμενων από ένα πρόγραμμα σε "σχεδόν" φυσική γλώσσα

Παράδειγμα: ΜΙΣΘΟΣ = (ΩΡΟΜΙΣΘΙΟ) Χ (ΩΡΕΣ ΕΡΓΑΣΙΑΣ)

ΓΛΩΣΣΑ ΜΗΧΑΝΗΣ	ASSEMBLY	C, C++, JAVA
100100 0000 010001	LOAD rate	wages = rate * hours;
100110 0000 010010	MULT hour	
100010 0000 010011	STOR wages	

Προγραμματισμός

Κυριολεκτικά:

Η συγγραφή των εντολών ενός προγράμματος

Εντολές = πηγαίος κώδικας (program source code)

Συνήθης χρήση

- Όλη η διαδικασία ανάπτυξης λογισμικού
- Σχεδίαση και γράψιμο προγράμματος

«Τεχνοτροπίες» προγραμματισμού

- Spaghetti programming
- Δομημένος προγραμματισμός
- Αντικειμενοστρεφής προγραμματισμός
- Λογικός προγραμματισμός

Ανάλογα με τα εργαλεία προγραμματισμού και τον τρόπο χρήσης τους

Προγραμματισμός

Η συγγραφή του πηγαίου κώδικα για κάποιες μονάδες προγράμματος οι οποίες, όταν εκτελούνται, παράγουν ένα επιθυμητό αποτέλεσμα.

Κανόνες που ακολουθούνται

- Συντακτικοί (γλώσσα προγραμματισμού)
- Σημασιολογικοί (γλώσσα προγραμματισμού)
- Λογικοί (πρόβλημα)

Οι κανόνες της γλώσσας προγραμματισμού αφορούν...

- Τα σύμβολα, τις εντολές, τις δομές της γλώσσας
- Τον τρόπο χρήσης των ιδιαίτερων χαρακτηριστικών κάθε υλοποίησης της γλώσσας (Visual C++, Borland C++, gcpp, κλπ)
- Τους γενικούς κανόνες και παραδοχές που αφορούν όλα τα προγράμματα

Η «δυσκολία» εκμάθησης μιας γλώσσας προγραμματισμού εντοπίζεται

- Στην απόσταση από τη φυσική γλώσσα του ανθρώπου
- Στους κανόνες σύνταξης οι οποίοι συμβάλλουν στον εμπλουτισμό της σημασιολογίας διότι οι Η/Υ δεν διαθέτουν ευφυΐα

Γλώσσες προγραμματισμού

Υπάρχει πλήθος γλωσσών προγραμματισμού ανάλογα με

- Το επίπεδο της τεχνολογίας των υπολογιστών
- Την επικρατούσα φιλοσοφία ανάπτυξης λογισμικού
- Το πεδίο εφαρμογής

Παραδείγματα (περίπου με χρονολογική σειρά)

FORTRAN

COBOL

BASIC

ALGOL

PASCAL

PROLOG

LISP

C

C++

JAVA

Προγραμματισμός ηλεκτρονικών υπολογιστών

Η χρήση μιας γλώσσας προγραμματισμού για την περιγραφή της λύσης ενός προβλήματος με χρήση ηλεκτρονικού υπολογιστή

- **Προσοχή! Πρέπει να γνωρίζουμε, ή να είμαστε σε θέση να βρούμε τη λύση**

Τα διαθέσιμα "εργαλεία" μας καθοδηγούν στον τρόπο περιγραφής και ενίοτε και εύρεσης της λύσης

- Όπως ο πολιτικός μηχανικός σχεδιάζει ένα τεχνικό έργο γνωρίζοντας τα δομικά στοιχεία που έχει στη διάθεσή του για να το κατασκευάσει
- Πρόβλημα είναι οι ιδιαιτερότητες του λογισμικού και του Η/Υ γενικότερα

Δημιουργούμε προγράμματα χρησιμοποιώντας...

- Εντολές και δομές ελέγχου εκτέλεσης

...ώστε να κατασκευάσουμε...

- σύνθετες μονάδες προγράμματος

...οι οποίες "κάνουν τη δουλειά"...

- ΕΙΣΟΔΟΣ – ΕΠΕΞΕΡΓΑΣΙΑ – ΕΞΟΔΟΣ

...και τέλος να τις μετατρέψουμε σε μορφή εκτελέσιμη από έναν Η/Υ

Του προγραμματισμού προηγείται η ανάλυση

Από το «πρόβλημα» μέχρι την συγγραφή του προγράμματος διακρίνουμε δύο φάσεις

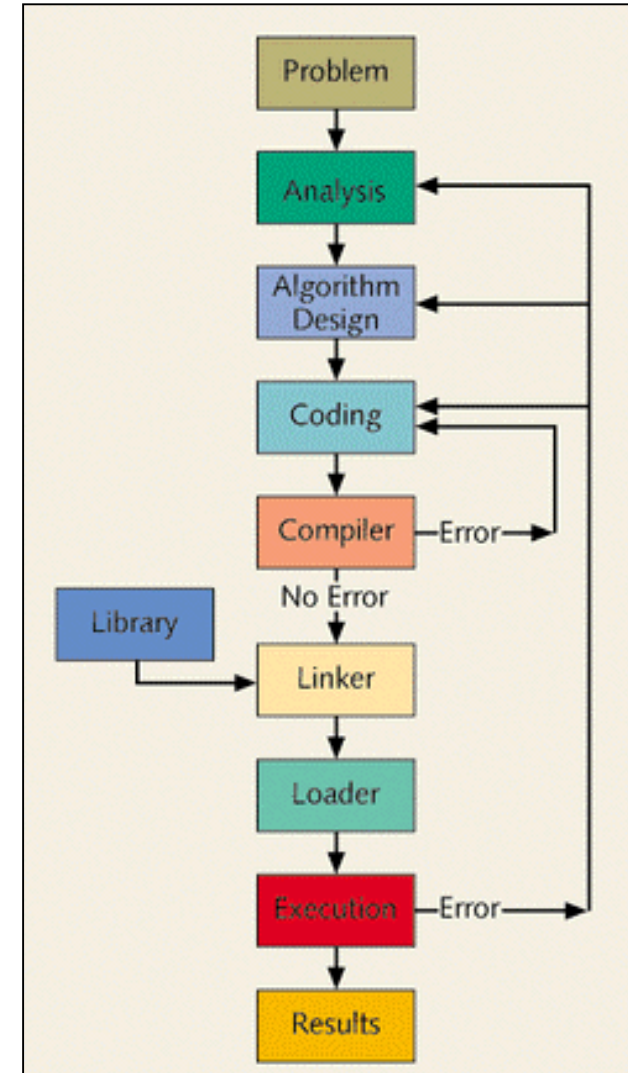
- Ανάλυση
- Σχεδίαση

Για μικρά έργα λογισμικού, η φάση της σχεδίασης ταυτίζεται με τη σχεδίαση του αλγορίθμου.

Τι είναι «ανάλυση»??

Ανάλυση: μια δημιουργική εργασία, όπου **καθορίζονται** και **καταγράφονται**:

- Οι **μετασχηματισμοί** από τους οποίους διέρχονται τα δεδομένα εισόδου, ώστε να δημιουργηθούν τα δεδομένα εξόδου
- Τα **δεδομένα** και οι **ροές** τους μεταξύ των μετασχηματισμών



Από την ανάλυση στον προγραμματισμό

Μερικές χρήσιμες παρατηρήσεις

- Η λύση ενός προβλήματος είναι στην καλύτερη περίπτωση τόσο «καλή» όσο «καλός» είναι ο ορισμός του προβλήματος
- «Καλός»: σωστός και ακριβής
- Στη φάση του προγραμματισμού (πρέπει να) είναι γνωστό το «τι θα κάνει το πρόγραμμα»...
- έτσι ώστε να μας απασχολεί πώς το πρόγραμμα θα κάνει με **σωστό τρόπο** τα **σωστά πράγματα**

Ενα πρόγραμμα... (1)

- είναι μια ακολουθία από εντολές μιας γλώσσας προγραμματισμού...
- οι οποίες οργανώνονται σε υποσύνολα που ονομάζονται «μονάδες προγράμματος»

Ενα πρόγραμμα... (2)

- είναι ένα σύνολο από **μονάδες προγράμματος**

Κάθε μονάδα προγράμματος

- είναι μια ακολουθία εντολών μιας γλώσσας προγραμματισμού

Προγραμματισμός ηλεκτρονικών υπολογιστών

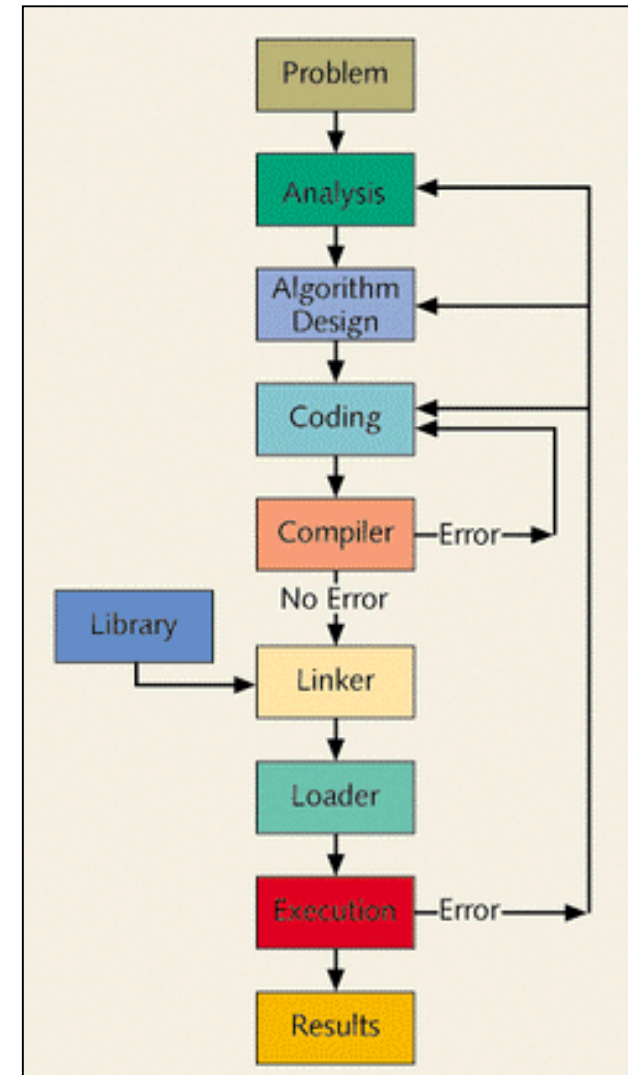
Μια ιδιαίτερα δημιουργική εργασία επίλυσης προβλημάτων (problem solving)

- Κατανόηση και ανάλυση του προβλήματος
- Περιγραφή των απαιτήσεων του προβλήματος
- Σχεδίαση των διαδοχικών βημάτων για την επίλυση

ΑΛΓΟΡΙΘΜΟΣ:

Η περιγραφή των διαδοχικών βημάτων μιας διαδικασίας επίλυσης προβλήματος η οποία καταλήγει σε λύση σε πεπερασμένο χρόνο

- Υλοποίηση του αλγορίθμου σε μια γλώσσα προγραμματισμού
- Επαλήθευση της λειτουργίας
- Εκτέλεση
- Συντήρηση (διορθώσεις, μεταβολές)



Προγραμματισμός ηλεκτρονικών υπολογιστών

Ανάλυση και σχεδίαση προγραμμάτων:

- Κατανόηση του πεδίου του προβλήματος (problem domain)
- Οριοθέτηση των απαιτήσεων του προβλήματος (problem requirements)
 - Χρειάζεται αλληλεπίδραση με τον χρήστη;
 - Τι είδους δεδομένα εμπλέκονται στο πρόβλημα;
 - Τι είδους αποτελέσματα πρέπει να υπολογίζονται;
- Κατασκευή ενός αλγορίθμου
- Εφαρμογή του "διαίρει και βασίλευε" αν το πρόβλημα είναι μεγάλο
 - Κατάτμηση σε μικρά προβλήματα και εφαρμογή των παραπάνω για καθένα από αυτά

Η παραπάνω διαδικασία λέγεται **"δομημένη ανάλυση και σχεδίαση"**

Οργανώνεται μακροσκοπικά με τεχνικές που αναφέρονται ως:

- Top-down design
- Stepwise refinement
- Modular programming

Περισσότερα στη γνωστική περιοχή **"Τεχνολογία Λογισμικού"**
(software engineering)

Η έννοια του αλγορίθμου (algorithm)

Τι είναι αλγόριθμος;

Η αναλυτική περιγραφή της μεθοδολογίας που οδηγεί στην ολοκλήρωση μιας συγκεκριμένης εργασίας, π.χ. την λύση ενός συγκεκριμένου προβλήματος.

Η λέξη αλγόριθμος προέρχεται από παραφθορά του ονόματος του μαθηματικού Abu Ja'Far Mohammed Ibn Musa **al-Khwarizmi** (9ος αιώνας μ.Χ.) που το βιβλίο του “**Al-Jabr wa-al-Muqabilah**” έβαλε τα θεμέλια της Άλγεβρας.

Παράδειγμα 1:

εργασία: ένα φλυτζάνι ελληνικού καφέ

αλγόριθμος: τι ακριβώς πρέπει να κάνει κανείς για να ετοιμάσει ένα φλυτζάνι καφέ

Παράδειγμα 2:

εργασία: να προσδιορισθεί ο μέγιστος κοινός διαιρέτης (ΜΚΔ) των ακεραίων αριθμών A και B

αλγόριθμος: τα διαδοχικά βήματα που οδηγούν στο ζητούμενο αποτέλεσμα (Αλγόριθμος του Ευκλείδη, τρίτος αιώνας π.Χ.)

Η έννοια του αλγορίθμου - Παράδειγμα 1

Τι είναι αλγόριθμος;

Η αναλυτική περιγραφή της μεθοδολογίας που οδηγεί στην ολοκλήρωση μιας συγκεκριμένης εργασίας, π.χ. την λύση ενός συγκεκριμένου προβλήματος.

Παράδειγμα:

εργασία: ένα φλυτζάνι ελληνικού καφέ

αλγόριθμος: τι ακριβώς πρέπει να κάνει κανείς για να ετοιμάσει ένα φλυτζάνι καφέ

- 1) να πάρει από το ντουλάπι το μπρίκι, το φλυτζάνι, την ζάχαρη, τον καφέ και το κουταλάκι
- 2) να γεμίσει το φλυτζάνι νερό και να το αδειάσει στο μπρίκι
- 3) να βάλει το μπρίκι στην φωτιά
- 4) να βάλει στο μπρίκι δυο κουταλάκια ζάχαρη
- 5) να βάλει στο μπρίκι ένα κουταλάκι καφέ
- 6) να ανακατέψει καλά το μίγμα στο μπρίκι
- 7) όταν ο καφές φουσκώσει, να βγάλει το μπρίκι από την φωτιά
- 8) να σερβίρει τον καφέ στο φλυτζάνι
- 9) να πλύνει το μπρίκι και το κουταλάκι

Η έννοια του αλγορίθμου

Τι είναι αλγόριθμος; Άλλη διατύπωση:

Αλγόριθμος είναι ένα πεπερασμένο και οργανωμένο σύνολο εντολών, που οδηγεί στην λύση ενός συγκεκριμένου προβλήματος, με τις ακόλουθες προϋποθέσεις:

- 1) Ο αλγόριθμος να μπορεί να περιγραφεί χρησιμοποιώντας ένα σύνολο λέξεων από συγκεκριμένο αλφάβητο και με συγκεκριμένους δομικούς και σημασιολογικούς κανόνες
- 2) Το πρόβλημα να μπορεί να περιγραφεί ακριβώς από ένα σύνολο δεδομένων (δεδομένα εισόδου), με τα οποία θα εκτελεστεί ο αλγόριθμος
- 3) Οι εντολές του αλγορίθμου εκτελούνται βήμα προς βήμα (ακολουθία εντολών). Κάποιες εντολές μπορεί να εκτελούνται μόνο αν αληθεύει μια συνθήκη (επιλογή). Επίσης, κάποιες εντολές μπορούν, υπό συνθήκη, να επαναλαμβάνονται (επανάληψη).
- 4) Το αποτέλεσμα κάθε εντολής προσδιορίζεται απόλυτα από τον αλγόριθμο, τα δεδομένα εισόδου και τα αποτελέσματα προηγούμενων εντολών
- 5) Η τελική απάντηση (δεδομένα εξόδου) πρέπει να είναι σαφώς καθορισμένη
- 6) Ο αλγόριθμος πρέπει να τερματίζεται μετά την εκτέλεση πεπερασμένου πλήθους βημάτων, ανεξάρτητα από τα δεδομένα εισόδου

Η έννοια του αλγορίθμου - Παράδειγμα 2

Να προσδιορισθεί ο μέγιστος κοινός διαιρέτης (ΜΚΔ) των ακεραίων αριθμών A και B

Δεδομένα εισόδου: οι αριθμοί A και B

Δεδομένα εξόδου : ο ΜΚΔ

Προϋποθέσεις : οι A και B να είναι θετικοί ακέραιοι

Αλγόριθμος

B1) Διάβασε τους αριθμούς A και B (είσοδος)

B2) Αν $A < B$ τότε : αντάλλαξε τις τιμές των A και B (ώστε $A > B$)

B3) Επανάλαβε τα ακόλουθα βήματα (B4 - B6) :

B4) εκτέλεσε την διαίρεση A / B και ονόμασε Y το υπόλοιπο

B5) δώσε στον A την τιμή του B

B6) δώσε στον B την τιμή του Y

B7) Αν $Y \neq 0$ τότε : επέστρεψε στο B3

B8) Τύπωσε τον ΜΚΔ (η τιμή του A) (έξοδος)

Επιλογή

Επανάληψη

Ακολουθία

Παράδειγμα εφαρμογής του αλγορίθμου του Ευκλείδη

Να προσδιορισθεί ο μέγιστος κοινός διαιρέτης (ΜΚΔ) των αριθμών 36 και 15

Εφαρμογή

B1: $A=36$, $B=15$

B2: (καμιά ενέργεια)

B4: $A/B = 2$, $Y = 6$

B5, B6: $A=15$, $B=6$

B7: $Y \neq 0$, επομένως επιστροφή στο B4

B4: $A/B = 2$, $Y = 3$

B5, B6: $A=6$, $B=3$

B7: $Y \neq 0$, επομένως επιστροφή στο B4

B4: $A/B = 2$, $Y = 0$

B5, B6: $A=3$, $B=0$

B7: $Y=0$, επομένως τέλος επανάληψης

B8: Ο ΜΚΔ είναι ο αριθμός 3

αλγόριθμος

B1) Διάβασε τους αριθμούς A και B (είσοδος)

B2) Αν $A < B$ αντάλλαξε τις τιμές των A και B (ώστε $A > B$)

B3) Επανάλαβε τα ακόλουθα βήματα (B4 - B6) :

B4) εκτέλεσε την διαίρεση A / B και ονόμασε Y το υπόλοιπο

B5) δώσε στον A την τιμή του B

B6) δώσε στον B την τιμή του Y

B7) Αν $Y \neq 0$ τότε : επέστρεψε στο B3

B8) Τύπωσε τον ΜΚΔ (η τιμή του A) (έξοδος)

Άσκηση: Βρείτε τον ΜΚΔ δύο ακεραίων της επιλογής σας, εφαρμόζοντας τον αλγόριθμο του Ευκλείδη.

Προϋποθέσεις ενός αλγορίθμου

- 1) Η εργασία, ή το πρόβλημα, να είναι σαφώς ορισμένο και διατυπωμένο με απλό και αναλυτικό τρόπο
- 2) Οι περιορισμοί, που τυχόν υπάρχουν κατά την εκτέλεση της εργασίας, να προσδιορίζονται με ακρίβεια
- 3) Οι πληροφορίες που χρειάζονται κατά την έναρξη της εργασίας (δεδομένα εισόδου), καθώς και τα αποτελέσματα (δεδομένα εξόδου), να προσδιορίζονται με σαφήνεια
- 4) Η εργασία να μπορεί να ολοκληρωθεί σε ένα πεπερασμένο αριθμό επί μέρους στοιχειωδών εργασιών (βημάτων)
- 5) Η μέθοδος να οδηγεί σε συγκεκριμένο αποτέλεσμα για οποιονδήποτε συνδυασμό δεδομένων εισόδου που είναι δυνατόν να εμφανισθεί.