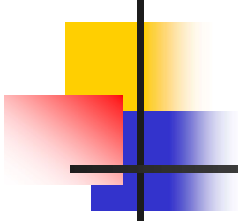


Τεχνολογία Υπολογιστικών Συστημάτων

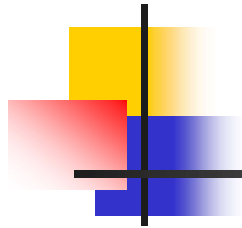
Πληροφοριακά Συστήματα Διοίκησης

Διδάσκων: Δημόκας Νίκος, Ph.D.



Εισαγωγή

- 1936, Ο Turing αποδεικνύει αυτό που είναι δυνατόν να γίνει σε μια αυτόματη μηχανή
- 1941, Η μηχανή Z3 του Konrad Zuse το 1941, ήταν η πρώτη που χρησιμοποιούσε το δυαδικό σύστημα αρίθμησης.
- 1943, Η ανακάλυψη των transistor από την BELL
- 1944, Η IBM σε συνεργασία με το Harvard κατασκευάζει το Mark I
- 1946, Οι Mauchly και Eckert κατασκευάζουν τον Eniac στο Πανεπιστήμιο της Πενσυλβάνια



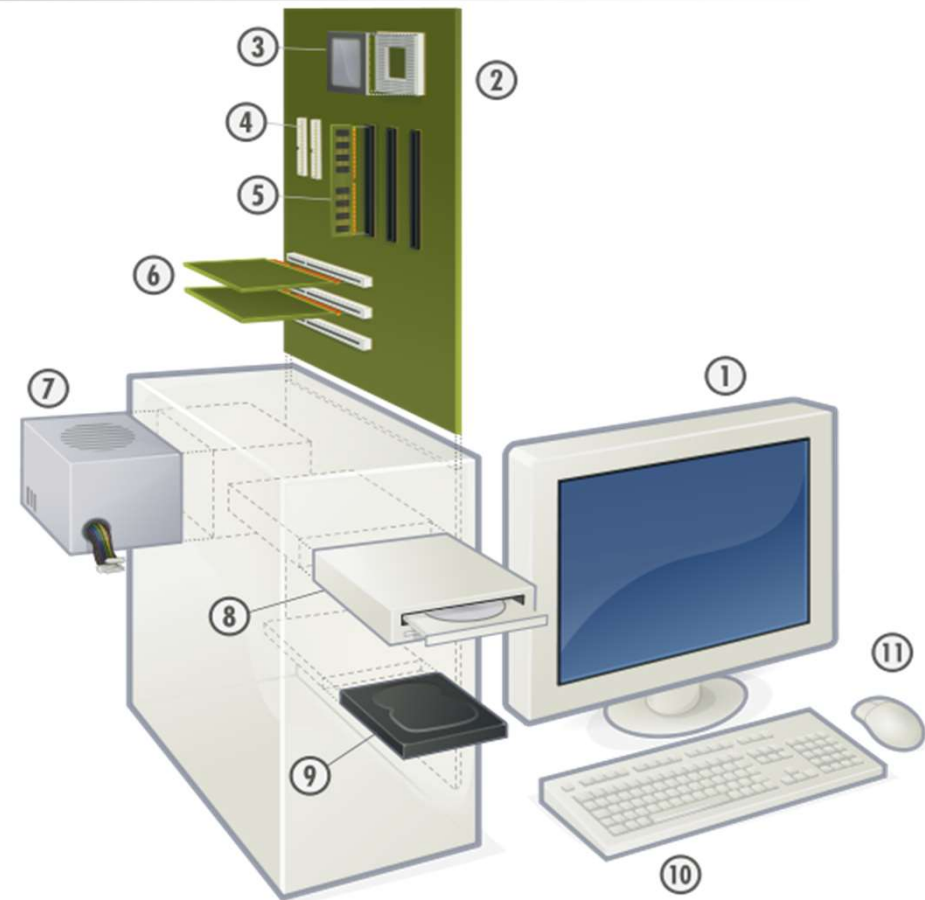
Εισαγωγή

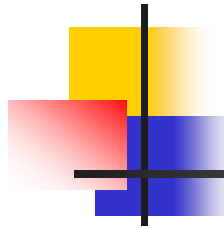
- 1947, Η χρήση καταχωρητών έγινε για πρώτη φορά στον υπολογιστή του Πανεπιστημίου του Manchester
- 1951, Univac
- 1953, Η κατασκευή της σειράς IBM 701
- 1956, Η κατασκευή της γλώσσας Fortran από τον Backus
- 1958, Η παρουσίαση του πρώτου ολοκληρωμένου κυκλώματος σε μορφή microchip
- 1959, Η παρουσίαση των γλωσσών Algol και Cobol

Αρχιτεκτονική ενός Η/Υ

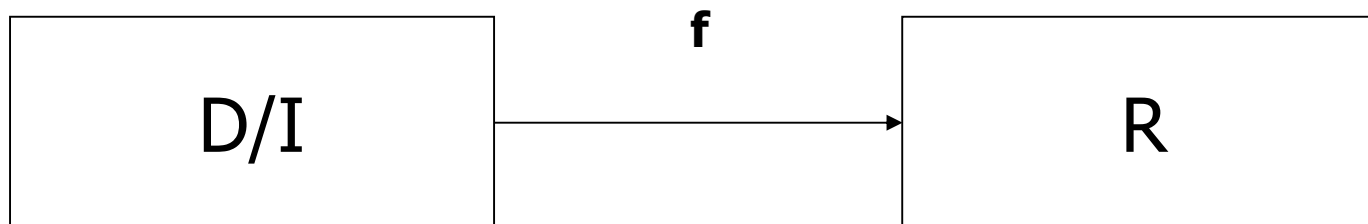
- Τα βασικά τμήματα ενός επιτραπέζιου υπολογιστή:

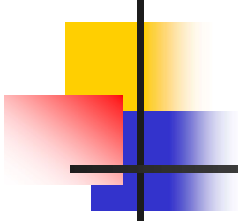
1. Οθόνη
2. Μητρική κάρτα
3. Κεντρικός επεξεργαστής
4. Θύρες επέκτασης
5. Μνήμη RAM
6. Κάρτες γραφικών και επέκτασης
7. Τροφοδοτικό
8. Οπτικό μέσο αποθήκευσης (CD-DVD)
9. Σκληρός δίσκος
10. Πληκτρολόγιο
11. Ποντίκι





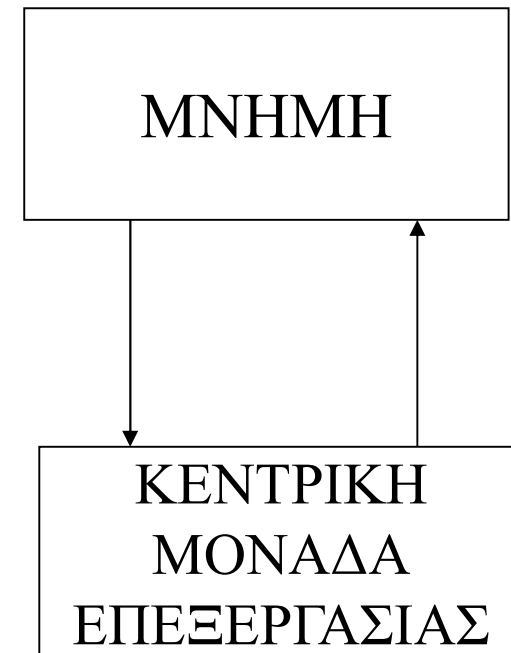
Αρχές λειτουργίας

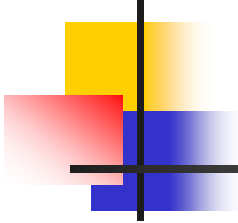




Συστατικά στοιχεία (1/5)

- Η κεντρική μονάδα αποτελείται:
 - Την Κεντρική Μονάδα Επεξεργασίας που είναι ένα σύνολο κυκλωμάτων που ήταν αρχικά ηλεκτρομηχανικά, αργότερα ηλεκτρονικά και σήμερα ολοκληρωμένα
 - Τη μνήμη





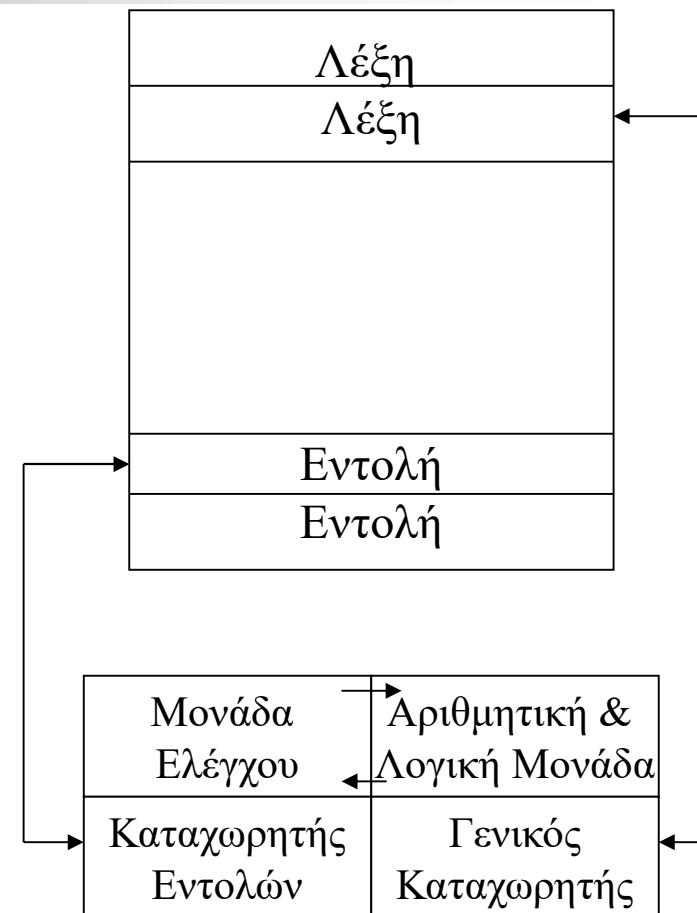
Συστατικά στοιχεία (2/5)

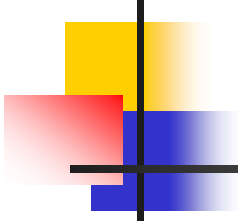
- Μια μνήμη αναπαρίσταται από N φυσικά συστήματα M διαφορετικών καταστάσεων
- Κάθε ένα από τα N συστήματα ονομάζεται λέξη
- Οι τιμές του M κυμαίνονται από $2^8=256$ έως 2^{60} ενώ για το N ξεκινώντας από μερικές χιλιάδες μπορεί να φτάσει μέχρι πολύ υψηλές τιμές

Λέξη N-1
Λέξη N-2
Λέξη 1
Λέξη 0

Συστατικά στοιχεία (3/5)

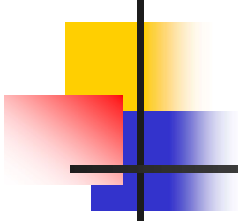
- Η κεντρική μονάδα συνίσταται από δύο μέρη
 - Τη μονάδα ελέγχου που καθορίζει τη χρονική διαδοχή των εντολών και ασχολείται με την αναζήτηση των προς εκτέλεση εντολών στη μνήμη
 - Τη μονάδα επεξεργασίας που εκτελεί τις εντολές που μεταφέρονται από τη μονάδα ελέγχου
- Οι καταχωρητές βοηθούν στην αποθήκευση των εντολών που εκτελούνται και των δεδομένων-αποτελεσμάτων που επεξεργάζονται





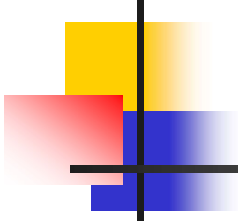
Συστατικά στοιχεία (4/5)

- Οι χρόνοι εκτέλεσης μπορούν να ομαδοποιηθούν σε πέντε επίπεδα:
 - Εκτέλεσης εντολής (της τάξης του nanosecond)
 - Οι μεταφορές καθαρά ηλεκτρονικές, μεταξύ κύριας μνήμης και κεντρικής μονάδας της τάξης του εκατομμυριοστού του δευτερολέπτου



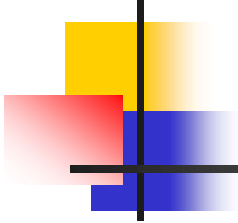
Συστατικά στοιχεία (5/5)

- Οι χρόνοι εκτέλεσης μπορούν να ομαδοποιηθούν σε πέντε επίπεδα:
 - Οι μεταφορές μεταξύ μαγνητικών μέσων και μνήμης στις οποίες χρησιμοποιούνται και μηχανικά μέσα όπως οι αναγνώστες ταινιών και δίσκων (κεφαλές ανάγνωσης) της τάξης των μερικών χιλιοστών του δευτερολέπτου
 - Ηλεκτρομηχανικές μεταφορές όπως η ανάγνωση καρτών ή η εκτύπωση χαρακτήρων της τάξης του χιλιοστού του δευτερολέπτου
 - Στις μεταφορές που παρεμβαίνει ο άνθρωπος όπως είναι η πληκτρολόγηση (πάρα πολύ αργή)



Γενικά στοιχεία για τις μνήμες (1/5)

- Ένα σύστημα που συγκροτείται από n στοιχεία δύο καταστάσεων περιλαμβάνει 2^n δυνατές καταστάσεις
- Η μνήμη μετράται σε Bytes ($2^8 = 256$ καταστάσεις)
- Συχνά το μέγεθος της εκφράζεται σε K-Bytes ή K-Λέξεις $2^{10} = 1024$ bytes ή λέξεις
- Ο αριθμός μιας λέξης ονομάζεται διεύθυνση.
- Όλες οι λέξεις είναι ισοδύναμες καθώς ο χρόνος προσπέλασης είναι ανεξάρτητος από τη θέση στη μνήμη
- Χρόνος προσπέλασης = χρόνος μεταφοράς από τη μνήμη σε κάποιο καταχωρητή



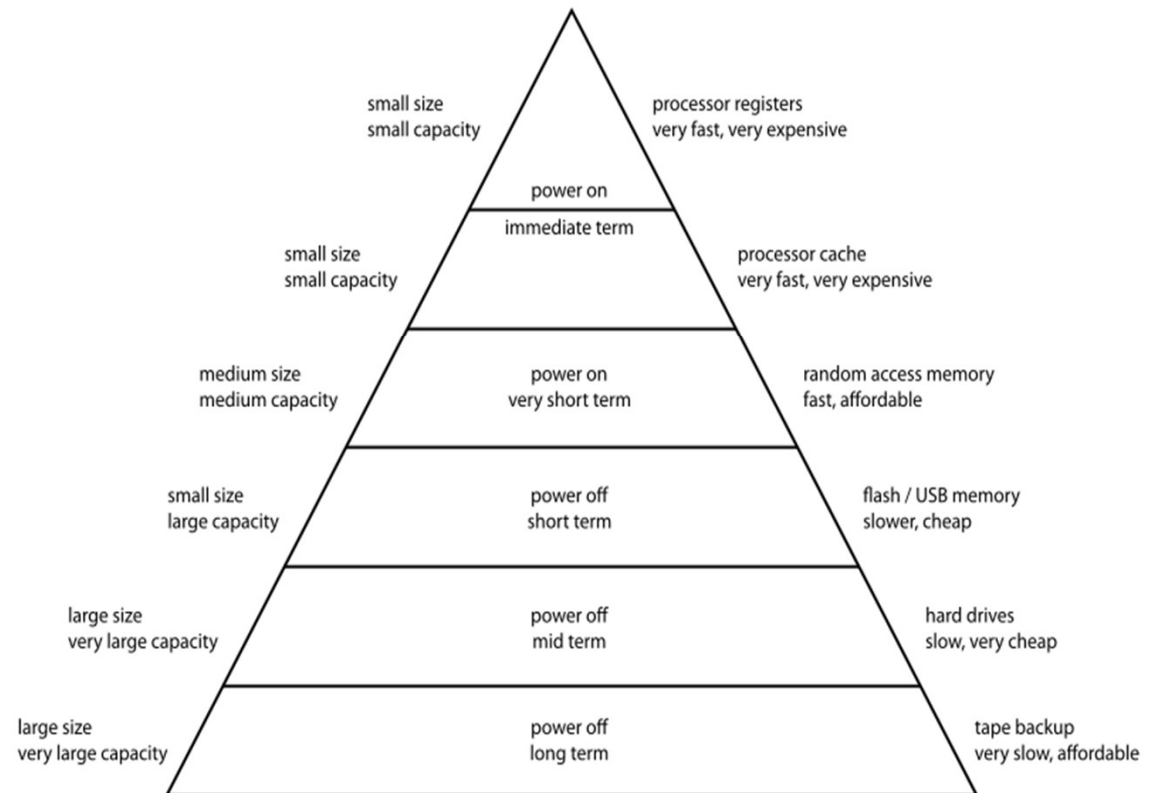
Γενικά στοιχεία για τις μνήμες (2/5)

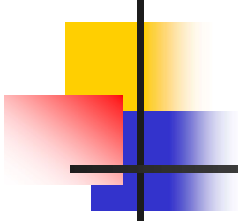
- Η μνήμη υπολογιστή έχει να κάνει με τις συσκευές και τα αποθηκευτικά μέσα ενός ηλεκτρονικού υπολογιστή, όπου αποθηκεύονται ψηφιακά δεδομένα τα οποία χρησιμοποιούνται από τον επεξεργαστή για κάποια χρονική περίοδο.
- Ένα σύστημα που συγκροτείται από n στοιχεία δύο καταστάσεων περιλαμβάνει 2^n δυνατές καταστάσεις
- Η μνήμη μετράται σε Bytes ($2^8 = 256$ καταστάσεις)
- Συχνά το μέγεθος της εκφράζεται σε K-Bytes ή K-Λέξεις $2^{10} = 1024$ bytes ή λέξεις

Γενικά στοιχεία για τις μνήμες (3/5)

- Ο αριθμός μιας λέξης ονομάζεται διεύθυνση.
- Όλες οι λέξεις είναι ισοδύναμες καθώς ο χρόνος προσπέλασης είναι ανεξάρτητος από τη θέση στη μνήμη
- Χρόνος προσπέλασης = χρόνος μεταφοράς από τη μνήμη σε κάποιο καταχωρητή

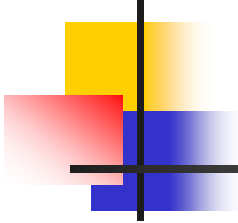
Computer Memory Hierarchy





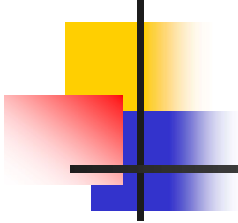
Γενικά στοιχεία για τις μνήμες (4/5)

- Τα κατώτερα επίπεδα της πυραμίδας προσφέρουν μεγαλύτερη αλλά πιο αργή μνήμη· αντιθέτως, τα ανώτερα προσφέρουν μικρότερη μνήμη αλλά πολύ πιο γρήγορη. Το κατώτατο επίπεδο της πυραμίδας είναι οι μαγνητικές ταινίες και ανεβαίνοντας προς τα πάνω συναντά κανείς:
 - τις μνήμες USB («φλασάκια»), τα CD-ROM ή DVD-ROM και τους σκληρούς δίσκους
 - την κύρια μνήμη RAM
 - την κρυφή μνήμη του επεξεργαστή
 - τους καταχωρητές του επεξεργαστή



Γενικά στοιχεία για τις μνήμες (5/5)

- Κύρια μνήμη ή κεντρική μνήμη ονομάζεται η μνήμη υπολογιστή που είναι προσβάσιμη από την Κεντρική Μονάδα Επεξεργασίας
- Συνήθως χρησιμοποιούνται μικροτσίπ μνήμης RAM, τα οποία είναι «πτητικές» μνήμες υπό την έννοια ότι διατηρούν τα δεδομένα τους μόνο όσο τροφοδοτούνται με ηλεκτρικό ρεύμα.

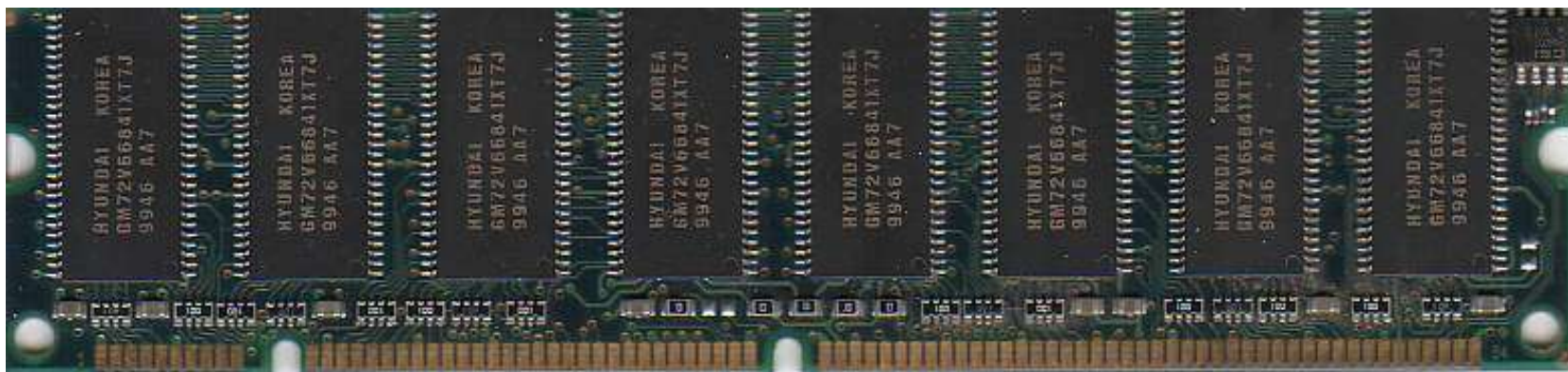


Μνήμη Τυχαίας Προσπέλασης (1/5)

- Μνήμη τυχαίας προσπέλασης (RAM, Random Access Memory) είναι όρος που χρησιμοποιούμε για ηλεκτρονικές διατάξεις προσωρινής αποθήκευσης ψηφιακών δεδομένων (μνήμης υπολογιστή), οι οποίες επιτρέπουν πρόσβαση στα αποθηκευμένα δεδομένα στον ίδιο χρόνο οπουδήποτε και αν βρίσκονται αυτά, δηλαδή με «τυχαία πρόσβαση».
 - Δυναμική RAM (DRAM): Η DRAM είναι η πιο κοινή μορφή αλλά πρέπει να ανανεώνεται χιλιάδες φορές ανά δευτερόλεπτο
 - Στατική RAM (SRAM): η SRAM δεν χρειάζεται κάτι τέτοιο.
- Η DRAM αποθηκεύει μπιτ σε έναν ξεχωριστό πυκνωτή. Επειδή το φορτίο των πυκνωτών εξασθενεί με το πέρασμα του χρόνου, πρέπει περιοδικά να επαναφορτίζεται ο πυκνωτής, εξ ου και ο όρος «δυναμική».

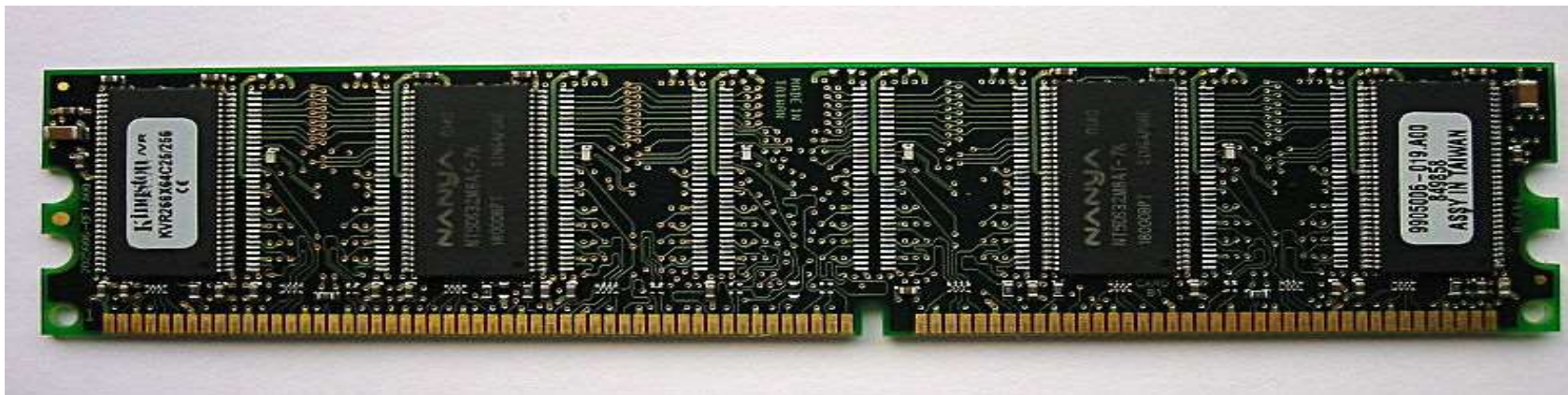
Μνήμη Τυχαίας Προσπέλασης (2/5)

- SDRAM: Synchronous Dynamic Random Access Memory
 - Είναι συγχρονισμένη με τον δίαυλο του επεξεργαστή. Η συχνότητα λειτουργίας είναι καθοριστικός παράγοντας για την ταχύτητά της.
 - Ο χρόνος προσπέλασης πρέπει να είναι όσο το δυνατόν χαμηλότερος για την ταχύτερη λειτουργία του υπολογιστή.
 - Δέχεται μία εντολή ανά κύκλο ρολογιού γράφοντας 64 bit δεδομένων.
 - Ταχύτητες: SDRAM PC-66, SDRAM PC-100, SDRAM PC-133.



Μνήμη Τυχαίας Προσπέλασης (3/5)

- DDR SDRAM: Double Data Rate Synchronous Dynamic Random Access Memory
 - Σε έναν κύκλο ρολογιού εκτελείται μία εντολή που διαβάζει και γράφει τα διπλάσια δεδομένα (128 bits)
 - Μεταφέρονται δεδομένα και κατά την ακμή ανόδου και την ακμή καθόδου του σήματος του ρολογιού του συστήματος
 - Ταχύτητες: DDR-200, DDR-233, DDR-333 έως DDR-400

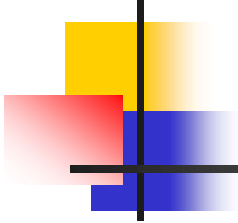


Μνήμη Τυχαίας Προσπέλασης (4/5)

- DDR2 SDRAM

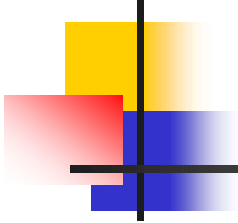
- Σε κάθε κύκλο λειτουργίας εγγράφονται 256 bit δεδομένων
- Έχει τάση λειτουργίας 1,8V
- Ταχύτητες: DDR2-400, DDR2-533, DDR2-667, DDR2-800





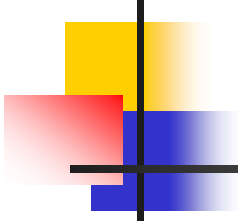
Μνήμη Τυχαίας Προσπέλασης (5/5)

- DDR3 SDRAM
 - Σε κάθε κύκλο λειτουργίας εγγράφονται 512 bit δεδομένων
 - Έχει τάση λειτουργίας 1,5V
 - Ταχύτητες: DDR3-800 έως DDR3-1600



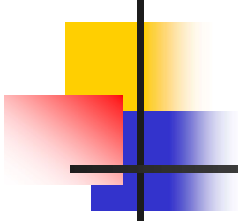
Βασικές Λειτουργίες H/Y (1/3)

- Οι λειτουργίες ονομάζονται και εντολές γλώσσας μηχανής και χωρίζονται σε τέσσερις κατηγορίες:
 - Εντολές υπολογισμού ή επεξεργασίας: πραγματοποιούνται από την μονάδα επεξεργασίας, π.χ. πρόσθεση, λογικό και, κ.λ.π.
 - Εντολές ανταλλαγής πληροφοριών μεταξύ μνήμης και μονάδας επεξεργασίας: π.χ. η αποθήκευση σε καταχωρητή του περιεχομένου μιας λέξης της μνήμης ή το αντίστροφο



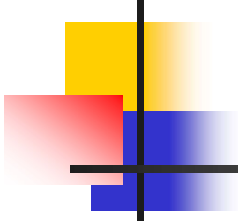
Βασικές Λειτουργίες H/Y (2/3)

- Οι λειτουργίες ονομάζονται και εντολές γλώσσας μηχανής και χωρίζονται σε τέσσερις κατηγορίες:
 - Εντολές ελέγχου: Οι εντολές κανονικά εκτελούνται σειριακά και σύμφωνα με τις διευθύνσεις που έχουν στο πρόγραμμα. Με τη χρήση εντολών ελέγχου είναι δυνατόν να καθορίσουμε όπως η επόμενη εντολή αναζητηθεί σε μια διαφορετική διεύθυνση
 - Εντολές εισόδου και εξόδου: καθορίζουν μια ανταλλαγή δεδομένων μεταξύ της κυρίας μνήμης και περιφερειακών



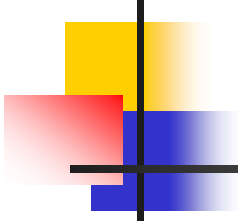
Βασικές Λειτουργίες H/Y (3/3)

- Μερικές εντολές σε IBM 370
 - 01011010 0001 000000000000010000001
πρόσθεση του καταχωρητή 1 στο περιεχόμενο της διεύθυνσης μνήμης 129 και το αποτέλεσμα στον καταχωρητή 1
 - 01011000 0011 00000000000000010010
μεταφορά στον καταχωρητή 3 του περιεχομένου της λέξης με διεύθυνση 37 στη μνήμη



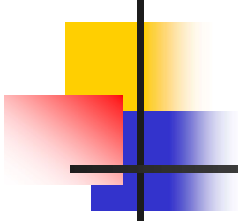
Μελέτη της οικογένειας επεξεργαστών 80x86

- Οι καταχωρητές της Κεντρικής Μονάδας Επεξεργασίας
- Το μέγεθος των καταχωρητών ποικίλει στους διάφορους επεξεργαστές της οικογένειας 80x86.
 - Οι 886, 8286, 8486 και 8686 CPU έχουν 4 καταχωρητές, όλοι μεγέθους 16 bits.
 - Επειδή ο x86 επεξεργαστής έχει τόσο λίγους καταχωρητές, θα δώσουμε σε κάθε καταχωρητή ένα όνομα και θα αναφερόμαστε σ' αυτόν με το όνομα του και όχι με τη διεύθυνση του.



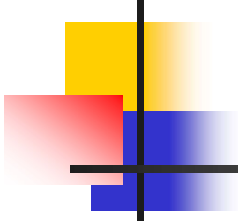
Οι καταχωρητές της Κεντρικής Μονάδας Επεξεργασίας (1/2)

- Για τον x86 επεξεργαστή τα ονόματα των καταχωρητών είναι :
 - AX: Αθροιστής (accumulator register)
 - BX: Καταχωρητής Βάσης (base address register)
 - CX: Μετρητής (counter register)
 - DX: Καταχωρητής δεδομένων (data register)
- Εκτός από τους παραπάνω καταχωρητές που είναι ορατοί στον προγραμματιστή, ο x86 επεξεργαστής έχει επίσης έναν καταχωρητή δείκτη εντολών, που περιέχει τη διεύθυνση της επόμενης εντολής για εκτέλεση, και ένα καταχωρητή σημαίας, που κρατάει το αποτέλεσμα από μια σύγκριση



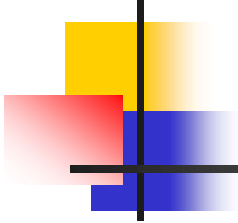
Οι καταχωρητές της Κεντρικής Μονάδας Επεξεργασίας (2/2)

- Επειδή η CPU χειρίζεται τους καταχωρητές ιδιαιτέρως και βρίσκονται on-chip, είναι πολύ γρηγορότεροι από τη μνήμη
- Η πρόσβαση σε μια θέση μνήμης απαιτεί έναν ή περισσότερους κύκλους ρολογιού. Η πρόσβαση δεδομένων σε καταχωρητή συνήθως χρειάζεται 0 κύκλους ρολογιού.
- Γι' αυτό θα πρέπει να προσπαθούμε να αποθηκεύουμε μεταβλητές στους καταχωρητές. Οι καταχωρητές αποτελούν το κατάλληλο μέρος για την αποθήκευση προσωρινών δεδομένων.



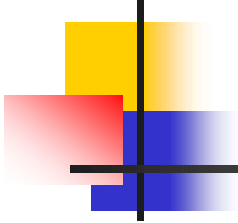
Αριθμητική και λογική μονάδα (ALU)

- Η αριθμητική και λογική μονάδα είναι εκεί όπου λαμβάνουν χώρα οι περισσότερες λειτουργίες μέσα στη CPU.
- Για παράδειγμα αν θέλουμε να προσθέσουμε την αξία του 5 στον καταχωρητή AX η CPU :
 - Αντιγράφει την αξία του AX στην ALU
 - Στέλνει την αξία 5 στην ALU
 - Δίνει εντολή στην ALU να προσθέσει τις δύο αξίες μεταξύ τους
 - Μετακινεί το αποτέλεσμα πίσω στον AX



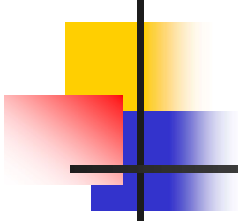
Μονάδα Διεπαφής Διάυλου (BIU)

- Η BIU είναι υπεύθυνη για τον έλεγχο των δίαυλων διευθύνσεων και δεδομένων, όταν γίνεται προσπέλαση στην κύρια μνήμη.
- Αν υπάρχει κρυφή μνήμη (cache) στο chip της CPU, τότε η BIU είναι επίσης υπεύθυνη για την προσπέλαση δεδομένων στην κρυφή μνήμη.



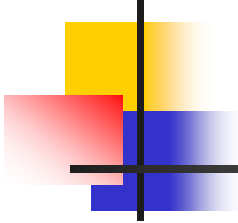
Το x86 σετ εντολών (1/7)

- Οι x86 CPU παρέχουν 20 βασικές τάξεις εντολών. Οι 7 από αυτές τις εντολές έχουν 2 τελεστές, 8 από αυτές έχουν έναν τελεστή και 5 εντολές δεν έχουν καθόλου τελεστές.
- Η εντολή mov είναι στην πραγματικότητα 2 τάξεις εντολών συγχωνευμένες στην ίδια εντολή. Οι δύο μορφές της εντολής mov παίρνουν τους ακόλουθους τύπους :
 - `mov reg, reg/memory/constant`
 - `mov memory, reg`



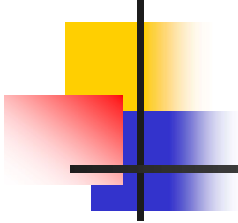
Το x86 σετ εντολών (2/7)

- Οι αριθμητικές και λογικές εντολές παίρνουν τις παρακάτω μορφές :
 - add reg, reg/memory/constant
 - sub reg, reg/memory/constant
 - cmp reg, reg/memory/constant
 - and reg, reg/memory/constant
 - or reg, reg/memory/constant
 - not reg/memory



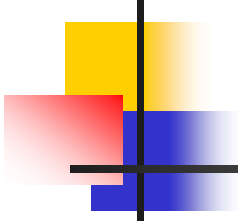
Το x86 σετ εντολών (3/7)

- Η εντολή `add` προσθέτει την αξία του δεύτερου τελεστή στον πρώτο (καταχωρητής), αφήνοντας το άθροισμα στον πρώτο τελεστή.
- Η `sub` εντολή αφαιρεί την αξία του δεύτερου τελεστή από τον πρώτο και αφήνει το αποτέλεσμα στον πρώτο.



Το x86 σετ εντολών (4/7)

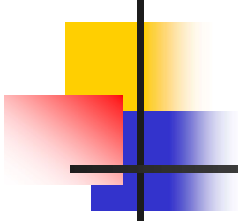
- Η `cmp` συγκρίνει τον πρώτο με τον δεύτερο τελεστή και σώζει το αποτέλεσμα της σύγκρισης, για χρήση με μια από τις υποθετικές `jump` εντολές (που θα περιγραφούν πιο κάτω).
- Οι `and` και `or` εντολές υπολογίζουν την αντίστοιχη λογική λειτουργία δύο τελεστών και αποθηκεύουν το αποτέλεσμα στον πρώτο.
- Η εντολή `not` αντιστρέφει τα bits στη μνήμη ή σε έναν καταχωρητή.



Το x86 σετ εντολών (5/7)

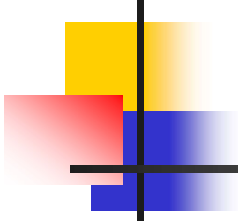
- Ο έλεγχος μεταφέρει εντολές, διακόπτει την ακολουθιακή εκτέλεση εντολών στη μνήμη, είτε άνευ όρων είτε μετά από εξέταση του αποτελέσματος της προηγούμενης cmp εντολής. Αυτές οι εντολές περιλαμβάνουν τα ακόλουθα:

■	Ja	dest	__ jump if above
■	Jae	dest	__ jump if above or equal
■	Jb	dest	__ jump if below
■	Jbe	dest	__ jump if below or equal
■	Je	dest	__ jump if equal
■	Jne	dest	__ jump if not equal
■	Jmp	dest	__ unconditional jump
■	Iret		__ return from an interrupt



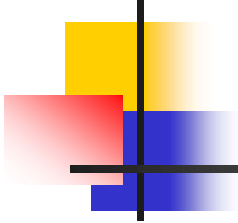
Το x86 σετ εντολών (6/7)

- Για παράδειγμα:
 - αν συγκρίνουμε τους ax και bx καταχωρητές με την cmp εντολή και εκτελέσουμε την ja εντολή, η x86 CPU θα πάει στην καθορισμένη θέση προορισμού αν ισχύει $ax > bx$. Αν το ax δεν είναι μεγαλύτερο του bx, τότε ο έλεγχος θα πάει στην επόμενη εντολή του προγράμματος
 - Η jmp εντολή άνευ όρων μεταφέρει τον έλεγχο στην εντολή της διεύθυνσης προορισμού
 - Η iret εντολή επιστρέφει τον έλεγχο από μια διακοπή



Το x86 σετ εντολών (7/7)

- Οι εντολές `get` και `put` μας επιτρέπουν να γράψουμε και να διαβάσουμε ακέραιες τιμές. Η `get` θα σταματήσει, θα ζητήσει από το χρήστη μια δεκαεξαδική τιμή και θα την αποθηκεύσει στον καταχωρητή `ax`. Η `put` εμφανίζει (σε δεκαεξαδικό σύστημα) την αξία του `ax` καταχωρητή.
- Η `Halt` τερματίζει την εκτέλεση του προγράμματος και η `brk` σταματά το πρόγραμμα σε μια κατάσταση, από την οποία μπορεί να ξαναξεκινήσει.
- Οι x86 επεξεργαστές χρειάζονται ένα μοναδικό opcode για κάθε διαφορετική εντολή, όχι απλά για κάθε τάξη εντολών. Παρόλο που η `"mov ax, bx"` και `"mov ax, cx"` ανήκουν στην ίδια τάξη, πρέπει να έχουν διαφορετικά opcodes για να τις ξεχωρίζει η CPU.



Παράσταση Εντολών (1/2)

- Μια τυπική CPU χρησιμοποιεί στον κωδικό εντολής ένα συγκεκριμένο αριθμό bits για να δηλώσει την τάξη της εντολής (π.χ mov, add κ.ο.κ) και έναν αριθμό bits για να κωδικοποιήσει τους τελεστές.
- Η οικογένεια Intel 80x86 έχει ένα από τα πιο πολύπλοκα σχήματα αποκωδικοποίησης κωδικών εντολών.

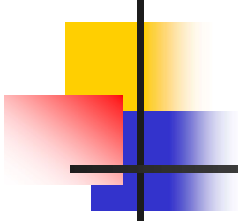


Παράσταση Εντολών (2/2)

- Για να καθορίσουμε ένα συγκεκριμένο κωδικό εντολής χρειάζεται μόνο να επιλέξουμε τα κατάλληλα bits για τα πεδία *iii*, *rr* και *mmm*. Για παράδειγμα για να κωδικοποιήσουμε την *mov ax,bx* εντολή θα διαλέγαμε *iii*=110, *rr*=00 και *mmm*=001 (*bx*) Αυτό παράγει την εντολή ενός byte 11000001

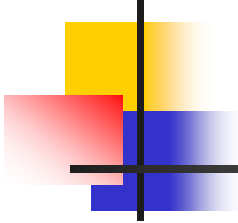
i	i	i	r	r	m	m	m	
---	---	---	---	---	---	---	---	--

iii	rr	mmm	This 16-bit field is present only if the instruction is a jump instruction or an operand is a memory addressing mode of the form [bx+xxxx], [xxxxx], or a constant.
000 = special	00 = AX	0 0 0 = AX	
001 = or	01 = BX	0 0 1 = BX	
010 = and	10 = CX	0 1 0 = CX	
011 = cmp	11 = DX	0 1 1 = DX	
100 = sub		1 0 0 = [BX]	
101 = add		1 0 1 = [xxxx+BX]	
110 = mov reg, mem/reg/const		1 1 0 = [xxxx]	
111 = mov mem, reg		1 1 1 = constant	



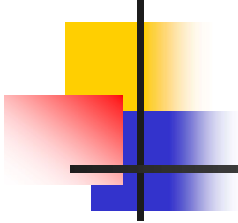
Βηματική εκτέλεση εντολών (1/3)

- Για την εντολή `mov memory, reg` η CPU κάνει τα ακόλουθα βήματα:
 - Καλεί το byte εντολής από τη μνήμη
 - Ανανεώνει τον IP για να δείχνει στο επόμενο byte
 - Αποκωδικοποιεί την εντολή
 - Αν απαιτείται καλεί τελεστή από τη μνήμη
 - Αν απαιτείται ανανεώνει τον IP για να δείχνει πέρα από τον τελεστή
 - Υπολογίζει τη διεύθυνση του τελεστή
 - Παίρνει την αξία του καταχωρητή για να την αποθηκεύσει (ένας κύκλος ρολογιού)
 - Αποθηκεύει την καλούμενη αξία στη θέση προορισμού (1 ή 2 ή 3 κύκλοι)
 - Συνολικά η εντολή `mov` χρειάζεται 5 με 11 κύκλους.



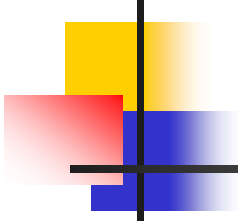
Βηματική εκτέλεση εντολών (2/3)

- Οι εντολές add, sub, cmp, and, or κάνουν τα ακόλουθα:
 - Καλούν το byte εντολής από τη μνήμη (1 κύκλος ρολογιού)
 - Ανανεώνουν τον IP (1 κύκλος ρολογιού)
 - Αποκωδικοποιούν την εντολή (1 κύκλος ρολογιού)
 - Αν απαιτείται καλούν μια σταθερά-τελεστή από τη μνήμη (0ή1ή2 κύκλους)
 - Αν απαιτείται ανανεώνουν τον IP να δείχνει πέραν της σταθεράς (0ή1κύκλος)



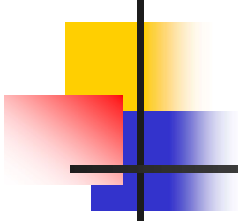
Βηματική εκτέλεση εντολών (3/3)

- Οι εντολές add, sub, cmp, and, or κάνουν τα ακόλουθα:
 - Υπολογίζουν τη διεύθυνση του τελεστή (1ή2 κύκλους)
 - Παίρνουν την αξία του τελεστή και τη στέλνουν στην ALU (0 κύκλοι αν είναι σταθερά, 1 αν είναι καταχωρητής, ή 2 κύκλους)
 - Καλούν την αξία του πρώτου τελεστή (καταχωρητή) και τη στέλνουν στην ALU (1 κύκλος ρολογιού)
 - Δίνουν εντολή στην ALU να προσθέσει, αφαιρέσει, συγκρίνει κ.ο.κ ανάλογα με την εντολή (1 κύκλος ρολογιού)
 - Αποθηκεύουν το αποτέλεσμα πίσω στον πρώτο τελεστή καταχωρητή (1 κύκλο ρολογιού)
 - Αυτές οι εντολές χρειάζονται μεταξύ 8 και 17 κύκλους ρολογιού για να εκτελεστούν.



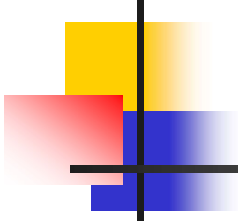
Εισαγωγή στον προγραμματισμό (1/5)

- Είναι πρακτικά αδύνατος χρησιμοποιώντας εντολές όπως οι προηγούμενες. Οι λόγοι είναι:
 - Η πρακτική δυσκολία κωδικοποίησης εντολών και δεδομένων
 - Η τεράστια απόσταση μεταξύ των ανθρωπίνων δυνατοτήτων που εμφανίζουν αυτές οι εντολές με τα όσα ζητούν οι άνθρωποι
 - Πολύ δύσκολη η εργασία εκσφαλμάτωσης
 - Η στενή εξάρτηση μεταξύ προγράμματος και μηχανής
- Γίνεται χρήση γλωσσών υψηλού επιπέδου και όχι Assemble

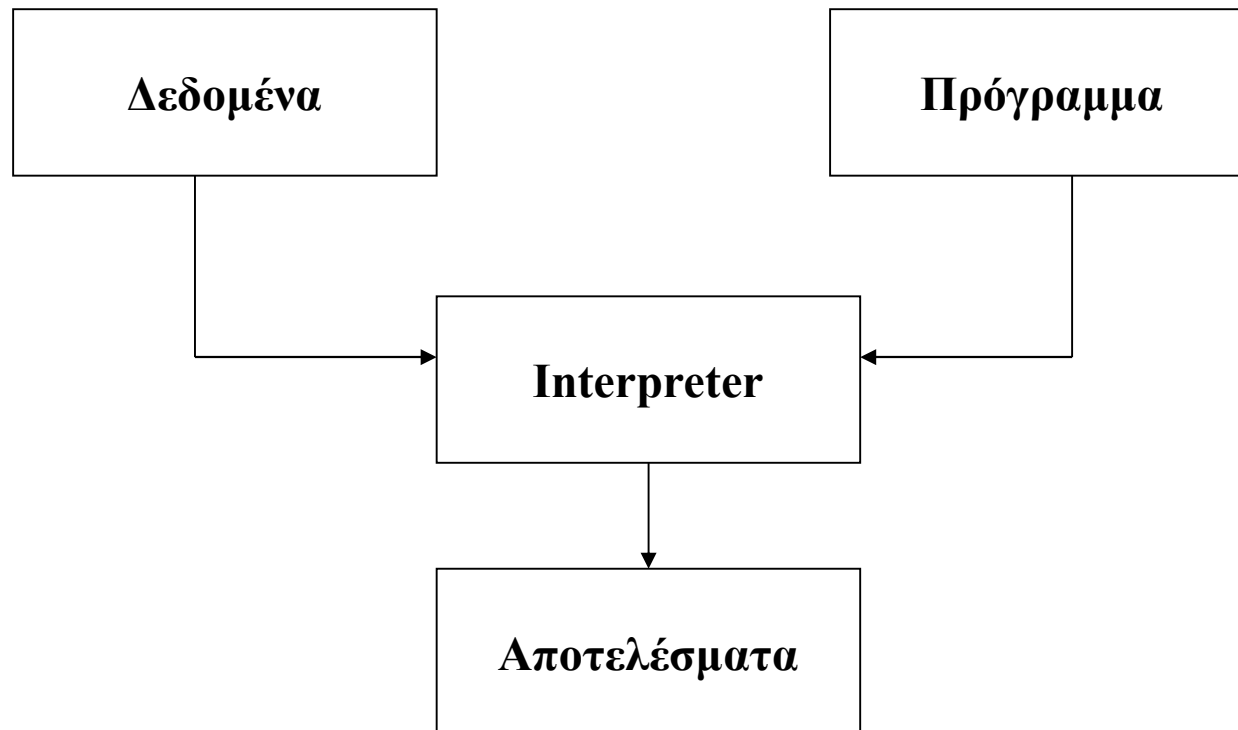


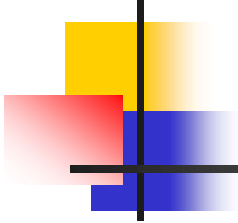
Εισαγωγή στον προγραμματισμό (2/5)

- Η εκτέλεση ενός προγράμματος που είναι γραμμένο σε μια γλώσσα προγραμματισμού προϋποθέτει την μετατροπή του σε γλώσσα μηχανής. Υπάρχουν δύο τρόποι.
- Ερμηνευτικός τρόπος
 - Με τη βοήθεια ενός προγράμματος που ονομάζεται ερμηνευτής (interpreter) η μετατροπή γίνεται ταυτόχρονα και βαθμηδόν τόσο στις εντολές του προγράμματος όσο και στα δεδομένα.



Εισαγωγή στον προγραμματισμό (3/5)

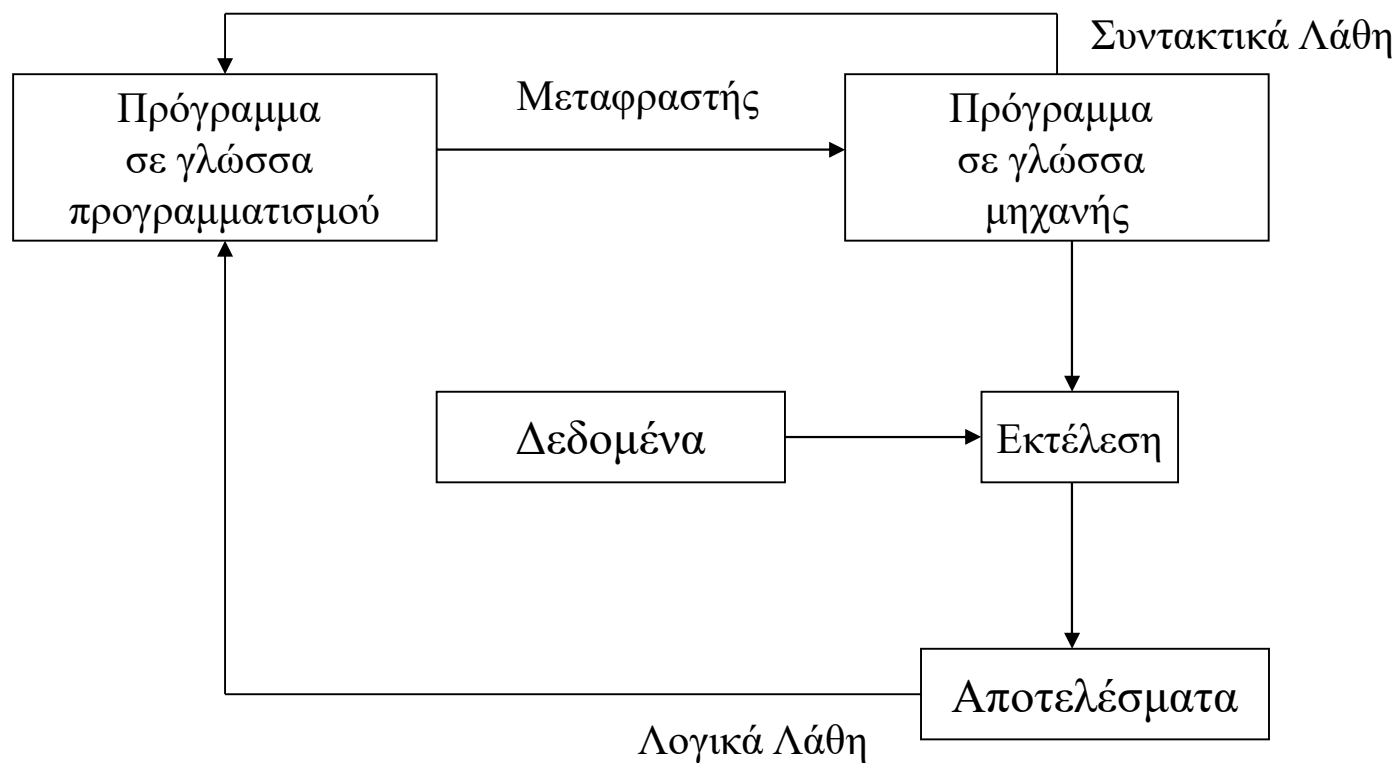


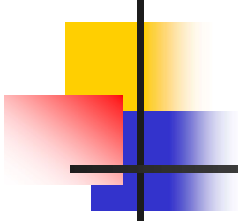


Εισαγωγή στον προγραμματισμό (4/5)

- Μεταφραστικός τρόπος. Η εκτέλεση ενός προγράμματος περιλαμβάνει δύο φάσεις
 - Στην πρώτη φάση γίνεται μετάφραση ενός προγράμματος σε ένα ισοδύναμο σε γλώσσα μηχανής. Αυτό γίνεται με το πρόγραμμα μεταφραστή (compiler)
 - Το δεύτερο στάδιο περιλαμβάνει την εκτέλεση του προγράμματος με τα δεδομένα

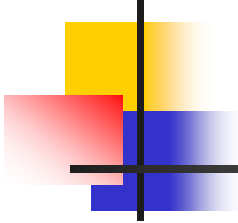
Εισαγωγή στον προγραμματισμό (5/5)





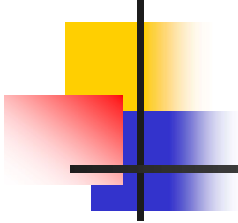
Τα σημερινά Υπολογιστικά Συστήματα (1/4)

- Προσωπικός Υπολογιστής
 - Εξυπηρετεί ένα μόνο χρήστη
 - Η ΚΜΕ, η κεντρική μνήμη, οι βοηθητικές μνήμες, η οθόνη και το πληκτρολόγιο βρίσκονται στον ίδιο χώρο
 - Προσιτό κόστος
 - Περιλαμβάνει μεταφραστικά προγράμματα όλων σχεδόν των γλωσσών υψηλού επιπέδου
 - Λειτουργικά συστήματα



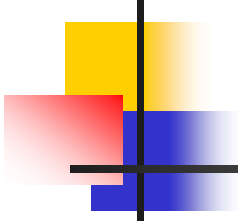
Τα σημερινά Υπολογιστικά Συστήματα (2/4)

- Συστήματα Πολλαπλών Χρηστών
 - Ένας κεντρικός, δυνατός υπολογιστής εξυπηρετεί πολλά τερματικά
 - Μεγάλοι υπολογιστές (mainframes) λέγονται οι Η/Υ που εξυπηρετούν πολλούς περιφερειακούς Η/Υ (π.χ. 1000 και πάνω). Μεγάλοι υπολογιστές υπάρχουν στις τράπεζες ή σαν web servers
 - Μεσαίοι υπολογιστές ονομάζονται Η/Υ μικρότεροι από τα mainframes. Εξυπηρετούν μέχρι 500 Η/Υ
 - Εξυπηρετητές (PC Servers) ονομάζονται οι Η/Υ, με αρχιτεκτονική ίδια με των προσωπικών υπολογιστών που παίζουν το ρόλο του εξυπηρετητή σε μικρά δίκτυα



Τα σημερινά Υπολογιστικά Συστήματα (3/4)

- Μέθοδος time sharing
 - Κάθε τερματικό είναι ανεξάρτητο από το άλλο και μια ορισμένη χρονική στιγμή μπορεί να κάνει χρήση των πόρων του κεντρικού Η/Υ
- Πολυπρογραμματισμός είναι η συνηθισμένη μέθοδος λειτουργίας των μεγάλων και μεσαίων υπολογιστών
 - Καταχωρούνται στη μνήμη και εκτελούνται ταυτόχρονα περισσότερα του ενός προγράμματα με επικάλυψη των ανεξάρτητων εργασιών τους
 - Όταν η ΚΜΕ παραμένει ανενεργή μπορεί να εκτελέσει κάποια εντολή ενός άλλου προγράμματος



Τα σημερινά Υπολογιστικά Συστήματα (4/4)

- Πολυδραστηριότητα (multitasking)
 - Με τη μέθοδο αυτή ένας Η/Υ μπορεί να εκτελεί ταυτόχρονα περισσότερες από μία ανεξάρτητες και διαφορετικές εντελώς δραστηριότητες
 - Π.χ. συνομιλία με ένα τερματικό ενώ συγχρόνως εκτελεί την εκτύπωση ενός άλλου προγράμματος
- Πολυεπεξεργαστής (multiprocessor)
 - Δύο ή περισσότερες ΚΜΕ μοιράζονται την ίδια μνήμη αλλά λειτουργούν ανεξάρτητα και αποτελούν έναν υπολογιστή