

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2007-08

Εισαγωγή στους Υπολογιστές

(Θεμελιώδεις αρχές λειτουργίας των υπολογιστών)

<http://www.ionio.gr/~mistral/tp/csintro/>

Μ.Στεφανιδάκης

Σχετικά με το μάθημα

- Εισαγωγή

- Ενότητες μαθήματος
 - Αρχές λειτουργίας υπολογιστών
 - Υλικό (hardware) – Αρχιτεκτονική Η/Υ
 - Αλγόριθμοι – Χειρισμός δεδομένων
 - Δομές δεδομένων – Λογισμικό (software)
 - Εφαρμογές
- Το βιβλίο του μαθήματος
 - Behrouz A. Forouzan, “Εισαγωγή στην Επιστήμη των Υπολογιστών”, ΚΛΕΙΔΑΡΙΘΜΟΣ, 2003.

Ο “Υπολογιστής” – Η γενική εικόνα

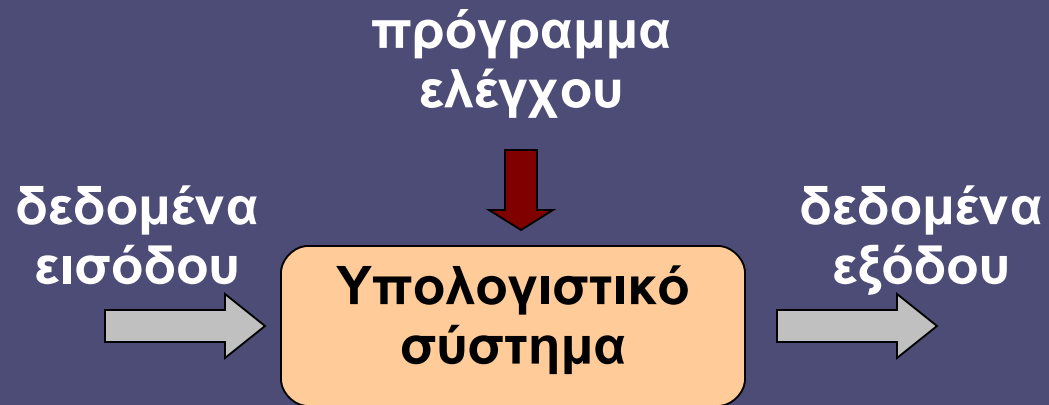
- Εισαγωγή

;

Από πού
προέρχονται τα
δεδομένα εισόδου;

Πού
αποθηκεύονται τα
δεδομένα εξόδου;

- Μετασχηματισμός δεδομένων εισόδου σε δεδομένα εξόδου, βάσει ενός προγράμματος ελέγχου

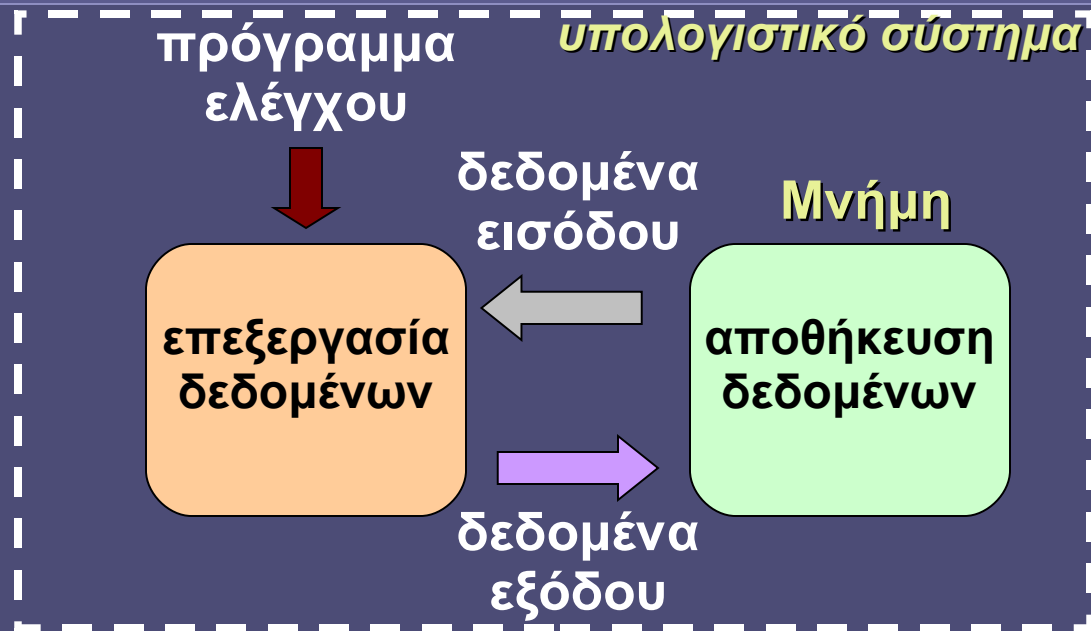


Η “μνήμη”

- Εισαγωγή

;

Και το πρόγραμμα ελέγχου;
Πώς αναπαρίσταται;
Πού βρίσκεται αποθηκευμένο;



- Τα δεδομένα εισόδου και εξόδου αποθηκεύονται στη μνήμη
 - Μπορούν να χρησιμοποιηθούν στην επόμενη φάση επεξεργασίας
 - Στην πραγματικότητα το σύστημα μνήμης είναι πολύπλοκότερο!

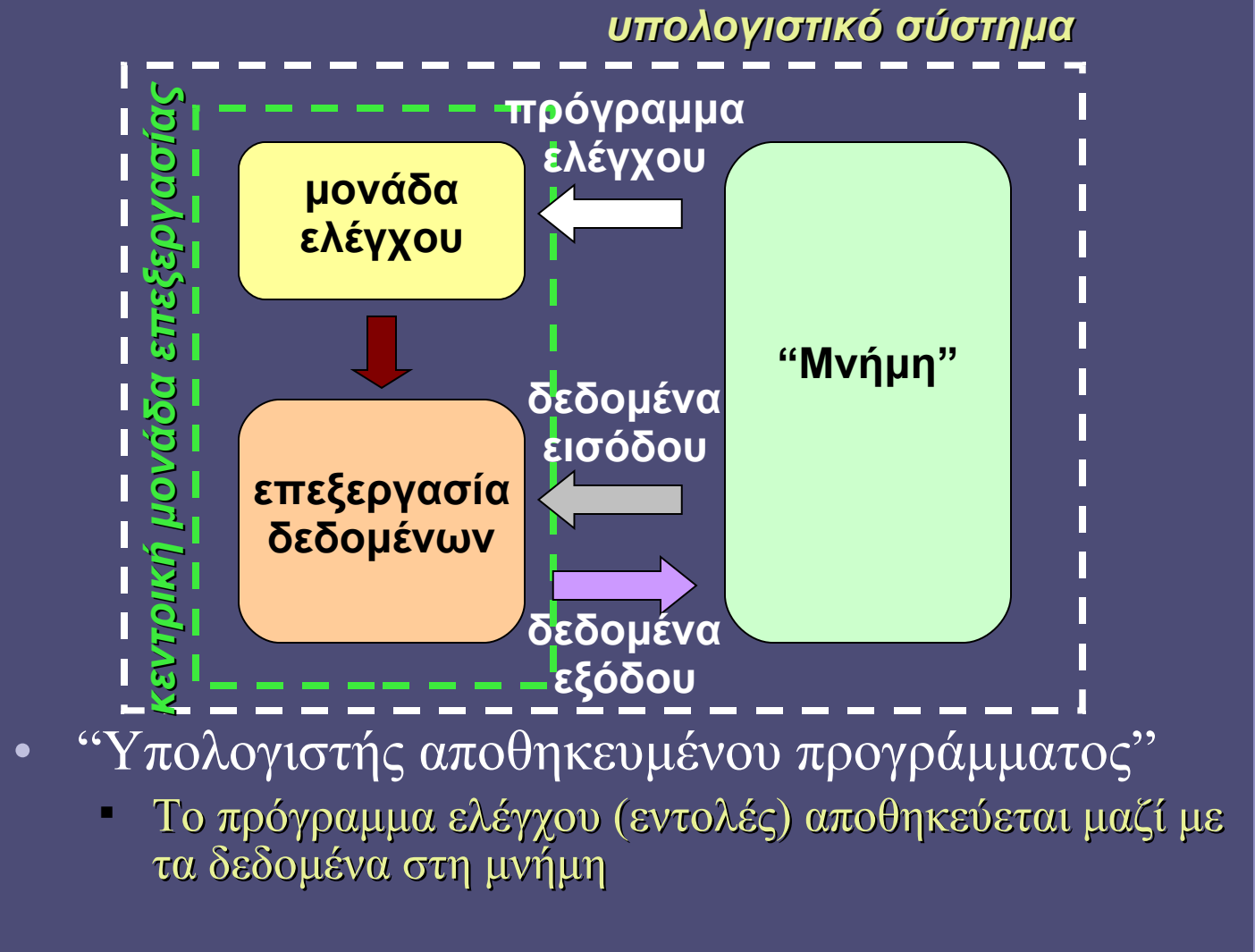
Το μοντέλο von Neumann

- Εισαγωγή

;

Πώς κατασκευάζεται ένα πρόγραμμα ελέγχου;

Ποιος δημιουργεί τις εντολές;



Το πρόγραμμα ελέγχου

- Εισαγωγή

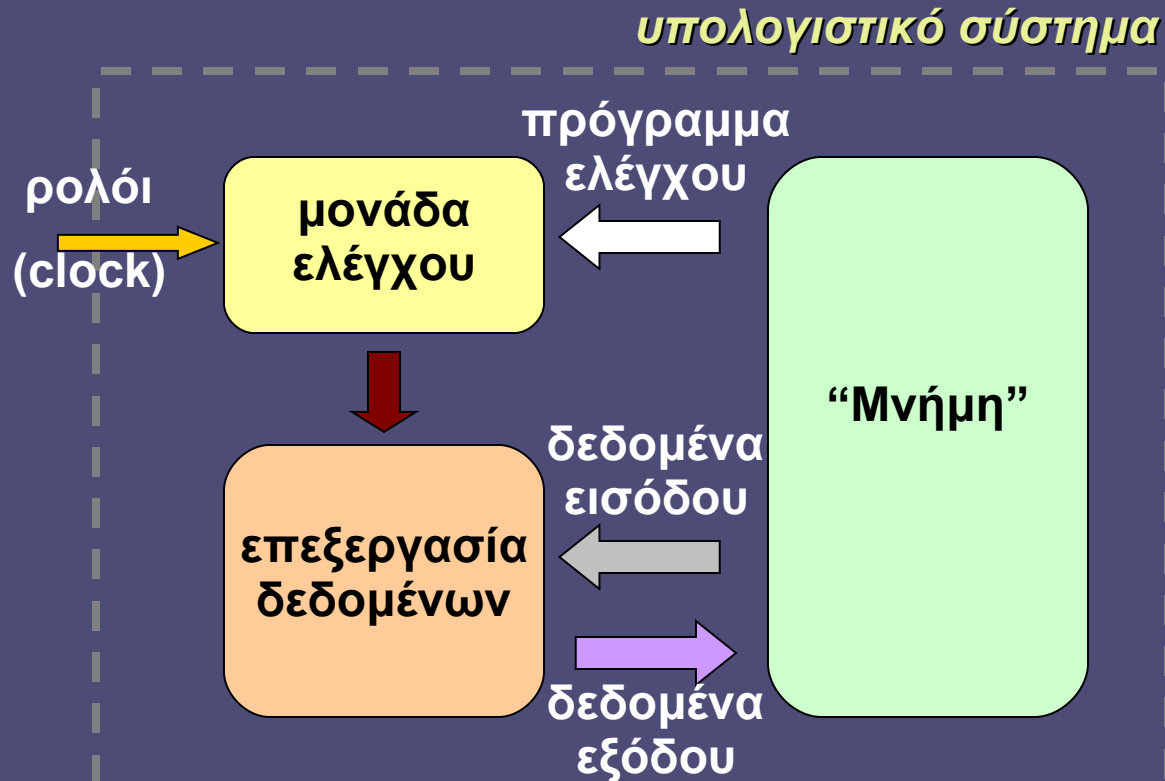
- **Σύνολο εντολών (οδηγιών)**
 - Περιγράφει το είδος της επεξεργασίας των δεδομένων
 - Τα δεδομένα εξόδου εξαρτώνται από
 - τα δεδομένα εισόδου
 - και το πρόγραμμα (λογισμικό)
 - **Γλώσσες προγραμματισμού**
 - Κάθε εντολή χειρίζεται κάποια δεδομένα
 - Οι εντολές εκτελούνται σειριακά
 - Η μια μετά την άλλη
 - **Εξαίρεση: εντολές διακλάδωσης**
 - Εάν μια συνθήκη αληθής, τότε μεταφορά εκτέλεσης σε διαφορετικό σημείο προγράμματος

Εκτέλεση ακολουθίας λειτουργιών

- Εισαγωγή



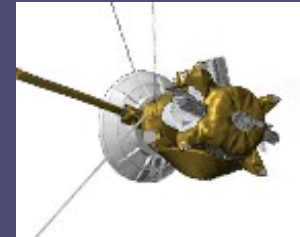
Το σχήμα συμπληρώνεται με τις μονάδες εισόδου-εξόδου, για επικοινωνία με τον έξω κόσμο



- Χρονισμός μέσω ενός σήματος ρολογιού
 - Ο “παλμός” του υπολογιστικού συστήματος
 - Έναρξη επόμενης λειτουργίας

Υπολογιστικά συστήματα(;

- Εισαγωγή



Πριν τους σύγχρονους υπολογιστές

- Εισαγωγή

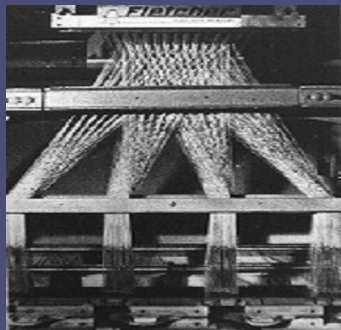
- Μηχανικά Συστήματα

- Εκτέλεση αριθμητικών πράξεων

- Για το εμπόριο, τη στατιστική...
- Την επιστήμη και τις εφαρμογές μηχανικού...
- Η κατασκευή τους απαιτούσε ακρίβεια μη διαθέσιμη τεχνολογικά πολλές φορές!

- Η μηχανή του Jacquard (1804)

- Αυτόματος αργαλειός που προγραμματιζόταν για την ύφανση πολύπλοκων σχεδίων μέσω διάτρητων χαρτοταινιών ή καρτελών
- Η γέννηση του προγράμματος ελέγχου;



Οι πρώτοι υπολογιστές

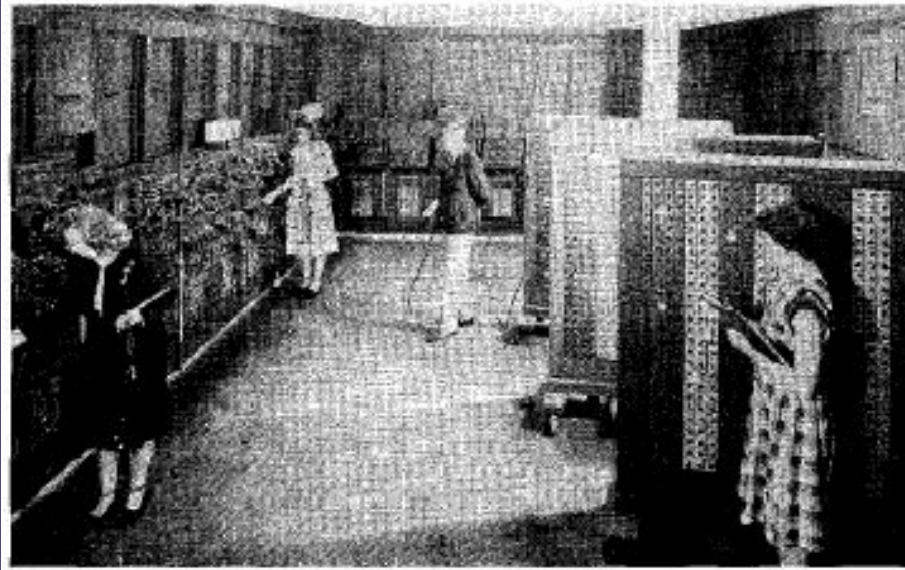
- Εισαγωγή

- Κατά και μετά τον 2ο παγκόσμιο πόλεμο
 - Εκτέλεση αριθμητικών πράξεων
 - Για στρατιωτικές εφαρμογές...
 - Υπολογισμός τροχιών πυροβολικού
 - Αποκρυπτογράφηση μηνυμάτων
 - Επιστημονικές πληροφορίες για τα συστήματα αυτά “δημοσιεύτηκαν” πολύ αργότερα!
 - Η εξέλιξη της τεχνολογίας
 - Αρχικά: ηλεκτρομηχανικά μέρη (ρελέ)
 - Στη συνέχεια: λυχνίες κενού ως διακόπτες (ON-OFF)
 - Η απαρχή της ηλεκτρονικής εποχής των υπολογιστών



The Electronic Numerical Integrator and Computer (ENIAC) 1946

- Εισαγωγή



- Κατασκευάστηκε για στρατιωτικούς σκοπούς στις ΗΠΑ
- Είχε 33 m μήκος και 2,5 m ύψος σε χώρο 270 m²
- Χρησιμοποιούσε 18.000 λυχνίες, 70.000 ηλεκτρικές αντιστάσεις, 5.000.000 κολλήσεις και το κόστος του υπερέβαινε τα 3.000.000\$
- Κατανάλωνε 160.000 Watts και ελεγχόταν με διακόπτες με το χέρι (δεν υπήρχε πρόγραμμα σε μορφή αποθηκευμένων εντολών)
- Παλμός συστήματος ανά 1/5000 sec

Γενιές υπολογιστών

- Εισαγωγή

- **1η γενιά (1950-1959)**
 - Λυχνίες ως διακόπτες
 - Μόνο σε μεγάλους οργανισμούς
- **2η γενιά (1959-1965)**
 - Τρανζίστορ ως διακόπτες
 - Προσιτοί σε μικρομεσαίες επιχειρήσεις
- **3η γενιά (1965-1975)**
 - Ολοκληρωμένα κυκλώματα
 - Εμπορικά πακέτα λογισμικού
- **4η γενιά (1975-1985)**
 - Μικροϋπολογιστές, δίκτυα..
- **5η γενιά (1985-)**
 - Φορητοί υπολογιστές, πολυμέσα, CDROM/DVD...

Η τρέχουσα τεχνολογία των υπολογιστών

- Εισαγωγή
- Τεχνολογία

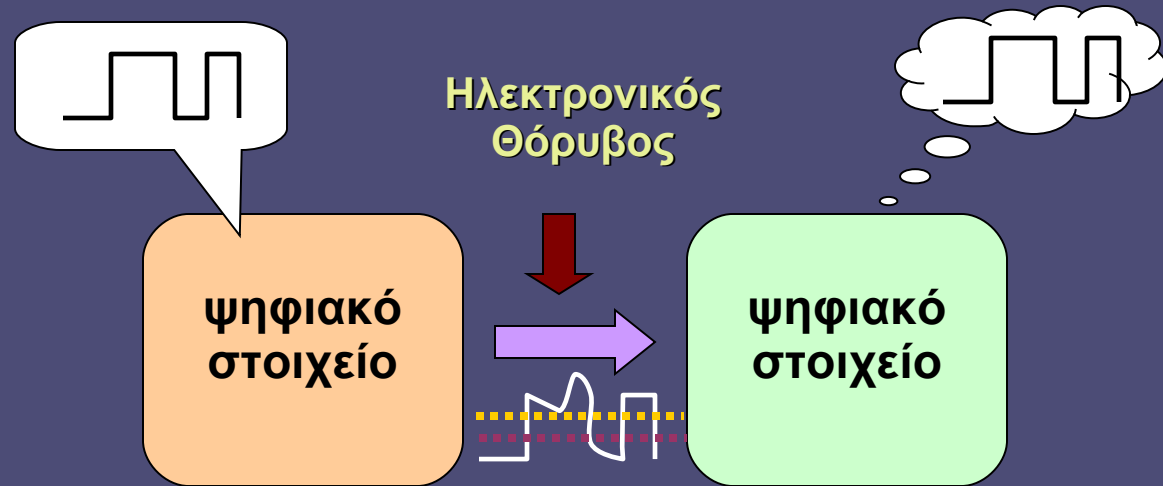
- Ηλεκτρονική
 - Τα στοιχεία που συγκροτούν τους υπολογιστές λειτουργούν με στάθμες τάσης ή φορά ρεύματος
 - Υλοποίηση με ημιαγωγούς
- Ψηφιακή
 - Οι επιτρεπόμενες τιμές πρέπει να ανήκουν σε διακριτές στάθμες
- Δυαδική
 - Τα στοιχεία που συγκροτούν τους υπολογιστές υλοποιούν διακοπτικές λειτουργίες
 - Δύο στάθμες, ON ή OFF, ‘0’ ή ‘1’

Γιατί ψηφιακή τεχνολογία;

- Εισαγωγή
- Τεχνολογία

;

Τι θα συνέβαινε μεταξύ δύο αναλογικών τμημάτων (με συνεχές πεδίο τιμών);



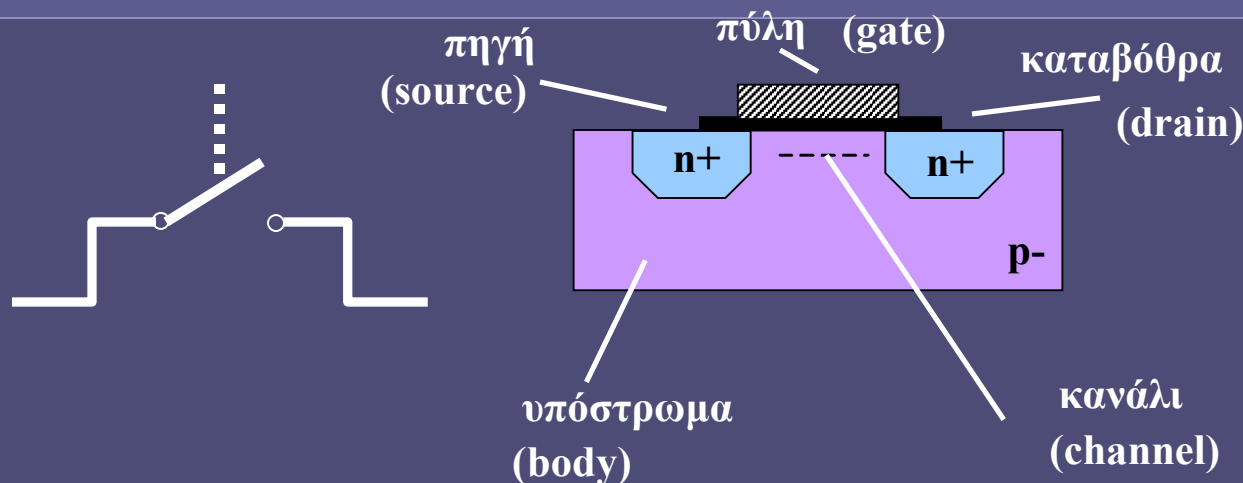
- Δυνατότητα αναγέννησης του αρχικού σήματος
 - Όσο η παραμόρφωση δεν υπερβαίνει κάποια όρια
 - Όσο λιγότερες στάθμες επιτρέπονται, τόσο τα επιτρεπτά όρια διακύμανσης είναι μεγαλύτερα
 - Πλεονέκτημα της δυαδικής τεχνολογίας

Γιατί δυαδική τεχνολογία;

- Εισαγωγή
- Τεχνολογία

i

Το τρανζίστορ αυτό είναι τύπου NMOS. Υπάρχει και το συμπληρωματικό PMOS.



- Το τρανζίστορ MOSFET
 - Ο μικροσκοπικός διακόπτης των σύγχρονων κυκλωμάτων
 - Η θεωρία λειτουργίας του είναι γνωστή από το 1925...
 - ...αλλά τα πρώτα λειτουργικά κυκλώματα κατασκευάστηκαν τη δεκαετία του 60

Η συρρίκνωση του τρανζίστορ

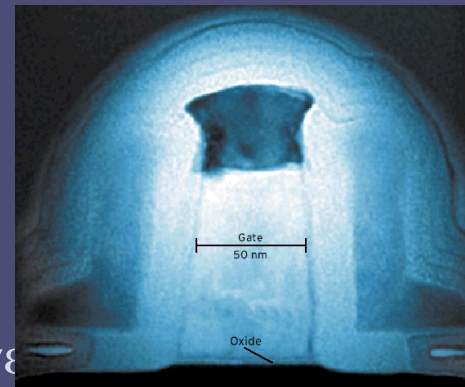
- Εισαγωγή
- Τεχνολογία

i

Ολοκληρωμένο κύκλωμα: όλα τα στοιχεία συνδυάζονται στο ίδιο υπόστρωμα πυριτίου.

Νόμος του Moore: ο αριθμός των τρανζίστορ ανά ολοκληρωμένο κύκλωμα διπλασιάζεται κάθε 1,5-2 χρόνια

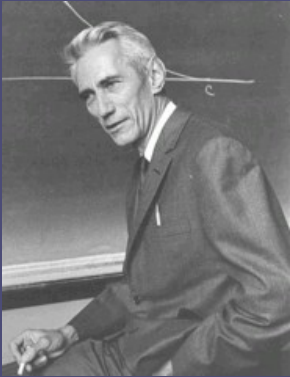
- Ένας σύγχρονος επεξεργαστής μπορεί να περιέχει πάνω από 1 δις τρανζίστορ σε μία επιφάνεια 250mm^2
- Πλεονεκτήματα
 - Ταχύτερη λειτουργία
 - Πιο γρήγοροι χρόνοι ON-OFF
 - Μικρότερη κατανάλωση ενέργειας
 - Για τον ίδιο αριθμό τρανζίστορ!
 - Μεγαλύτερη ολοκλήρωση
 - Μείωση κόστους παραγωγής και αύξηση λειτουργικότητας
- Τρέχουσα εμπορική τεχνολογία:
 - Μέγεθος (καναλιού) τρανζίστορ = 65nm
- Το μέλλον; 45 και 25 nm



Ψηφιακά Ηλεκτρονικά και Δυαδική Λογική

- Εισαγωγή
- Τεχνολογία
- Δυαδική λογική

- Η δυαδική λογική ταιριάζει με την τεχνολογία του τρανζίστορ
 - 2 καταστάσεις: ON-OFF, 1-0
 - Ψηφιακά ηλεκτρονικά (2 στάθμες)
- Δυαδική άλγεβρα Boole
 - Λογική άλγεβρα
 - Συσχέτιση με διακοπτικά κυκλώματα
 - Η εργασία του Shannon (1938)



C.E.Shannon

Ποσότητες Δυαδικής Λογικής

- Εισαγωγή
- Τεχνολογία
- Δυαδική λογική

- **Στη δυαδική λογική άλγεβρα**
 - Υπάρχουν 2 “ποσότητες” (σύμβολα):
 - Αληθές ή 1 ή ΝΑΙ
 - Ψευδές ή 0 ή ΟΧΙ
 - Ένα δυαδικό ψηφίο (**bit**) έχει τιμή 0 ή 1
- **Στα ψηφιακά ηλεκτρονικά κυκλώματα:**
 - 0 ή “χαμηλή τάση” ή “η μια φορά ρεύματος”
 - 1 ή “υψηλή τάση” ή “η άλλη φορά ρεύματος”

 - Ανάλογα με την τεχνολογία, ένα bit αναπαρίσταται με αντίστοιχη κατάσταση ενός **σημείου** σε ένα ηλεκτρονικό κύκλωμα

Πράξεις Δυαδικής Λογικής

- Εισαγωγή
- Τεχνολογία
- Δυαδική λογική

- **Στη δυαδική λογική άλγεβρα**
 - Καθορίζονται λογικές πράξεις μεταξύ λογικών ποσοτήτων (bits)
- **Στα ψηφιακά ηλεκτρονικά κυκλώματα:**
 - Το αντίστοιχο κύκλωμα δέχεται ως **είσοδο** την αναπαράσταση των λογικών ποσοτήτων και παράγει στην **έξοδό** του την αναπαράσταση του αποτελέσματος της συγκεκριμένης λογικής πράξης
 - Το κύκλωμα υλοποίησης μιας λογικής πράξης ονομάζεται **πύλη** (gate).

Βασικές Πράξεις

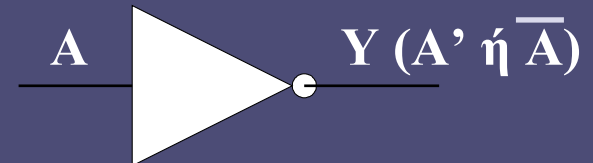
- Εισαγωγή
- Τεχνολογία
- Δυαδική λογική

- Αντιστροφή (NOT)
 - Αντιστροφή των bits

Πίνακας Αλήθειας

A	Y
0	1
1	0

Σύμβολο πύλης NOT



Βασικές Πράξεις

- Εισαγωγή
- Τεχνολογία
- Δυαδική λογική

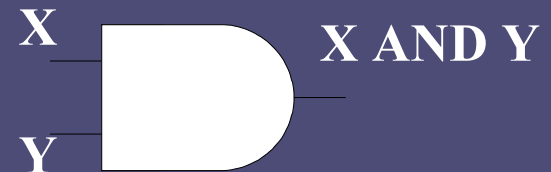
- Λογικό ΚΑΙ (AND)

- αποτέλεσμα 1, μόνο όταν και το X και το Y είναι 1
- $0 \text{ AND } X = X \text{ AND } 0 = 0$
- $1 \text{ AND } X = X \text{ AND } 1 = X$

Πίνακας Αλήθειας

X	Y	AND
0	0	0
0	1	0
1	0	0
1	1	1

Σύμβολο πύλης AND



Βασικές Πράξεις

- Εισαγωγή
- Τεχνολογία
- Δυαδική λογική

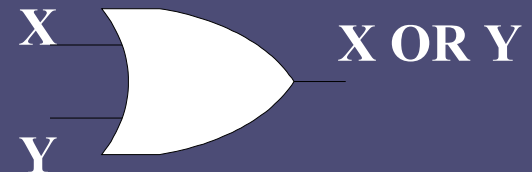
- Λογικό Ή (OR)

- αποτέλεσμα 1, όταν X ή Y ή και τα δύο είναι 1
- $1 \text{ OR } X = X \text{ OR } 1 = 1$
- $0 \text{ OR } X = X \text{ OR } 0 = X$

Πίνακας Αλήθειας

X	Y	OR
0	0	0
0	1	1
1	0	1
1	1	1

Σύμβολο πύλης OR



Βασικές Πράξεις

- Εισαγωγή
- Τεχνολογία
- Δυαδική λογική

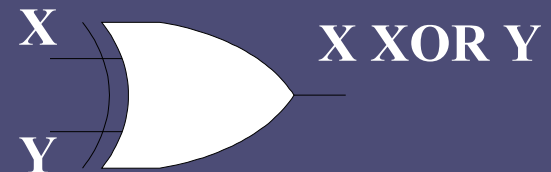
- Αποκλειστικό Ή (XOR)

- αποτέλεσμα 1, όταν **μόνο το X** ή **μόνο το Y** είναι 1
- $1 \text{ XOR } X = X \text{ XOR } 1 = \text{NOT } X$
- $0 \text{ XOR } X = X \text{ XOR } 0 = X$

Πίνακας Αλήθειας

X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Σύμβολο πύλης XOR



Στο επόμενο μάθημα...

- Εισαγωγή
- Τεχνολογία
- Δυαδική λογική
- Στο επόμενο...

- **Περισσότερα για την τεχνολογία των υπολογιστών**
 - Φυσική Ημιαγωγών
 - Το τρανζίστορ και η λειτουργία του
 - Πώς κατασκευάζονται τα ολοκληρωμένα κυκλώματα των υπολογιστών;

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2007-08

Η Τεχνολογία των Υπολογιστών

(Κατασκευή και Λειτουργία Ψηφιακών Κυκλωμάτων)

<http://www.ionio.gr/~mistral/tp/csintro/>

Μ.Στεφανιδάκης

Η τρέχουσα τεχνολογία των υπολογιστών

- Τεχνολογία

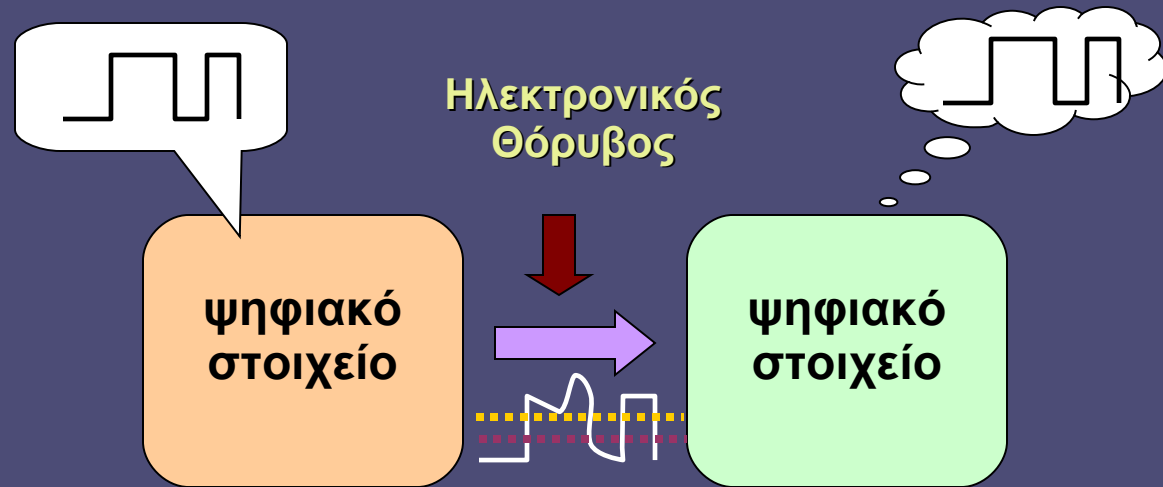
- Ηλεκτρονική
 - Τα στοιχεία που συγκροτούν τους υπολογιστές λειτουργούν με στάθμες τάσης ή φορά ρεύματος
 - Υλοποίηση με ημιαγωγούς (semiconductors)
- Ψηφιακή
 - Οι επιτρεπόμενες τιμές πρέπει να ανήκουν σε διακριτές στάθμες
- Δυαδική
 - Τα στοιχεία που συγκροτούν τους υπολογιστές υλοποιούν διακοπτικές λειτουργίες
 - Δύο στάθμες, ON ή OFF, ‘0’ ή ‘1’

Γιατί ψηφιακή τεχνολογία;

- Τεχνολογία

;

Τι θα συνέβαινε μεταξύ δύο αναλογικών τμημάτων (με συνεχές πεδίο τιμών);

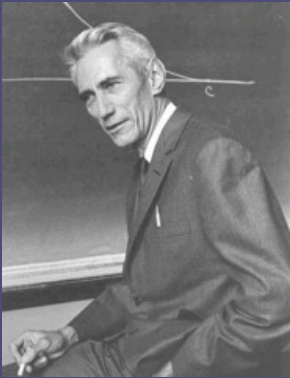


- Δυνατότητα αναγέννησης του αρχικού σήματος
 - Όσο η παραμόρφωση δεν υπερβαίνει κάποια όρια
 - Όσο λιγότερες στάθμες επιτρέπονται, τόσο τα επιτρεπτά όρια διακύμανσης είναι μεγαλύτερα
 - Πλεονέκτημα της δυαδικής τεχνολογίας

Ψηφιακά Ηλεκτρονικά και Δυαδική Λογική

- Τεχνολογία

- Η δυαδική λογική ταιριάζει με την τεχνολογία του τρανζίστορ
 - 2 καταστάσεις: ON-OFF, 1-0
 - Ψηφιακά ηλεκτρονικά (2 στάθμες)
- Δυαδική άλγεβρα Boole
 - Λογική άλγεβρα
 - Συσχέτιση με **διακοπτικά** κυκλώματα
 - Η εργασία του Shannon (1938)



C.E.Shannon

Στάθμες ψηφιακού σήματος

- Τεχνολογία

i

Στα πρώτα ολοκληρωμένα λογικά κυκλώματα η τάση τροφοδοσίας ήταν $>15V$ ενώ σήμερα βρίσκεται γύρω στο $1V$

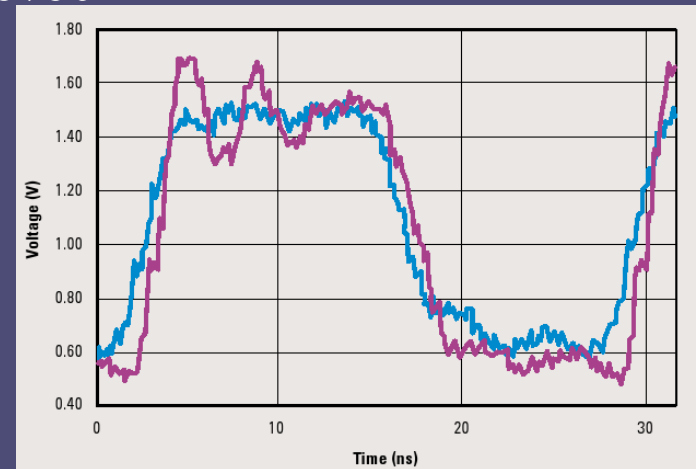
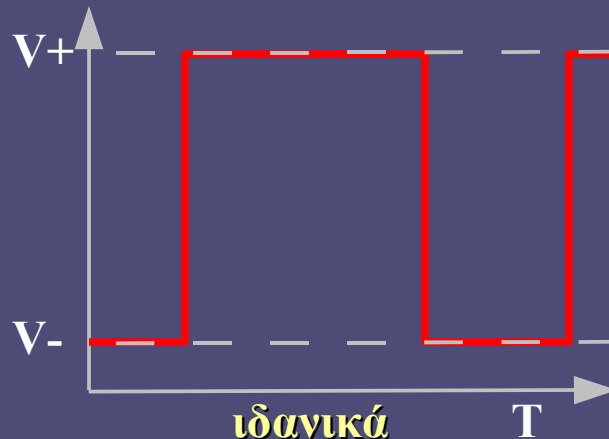
- Δυαδική λογική

- 2 στάθμες

- $V+$ και $V-$ (π.χ. $5V$ και $0V$)
 - “Τροφοδοσία” και “γείωση”
 - Αναπαριστούν το λογικό 1 και 0 αντίστοιχα

- Κυματομορφές

- Η στάθμη τάσης σε ένα σημείο του κυκλώματος στην εξέλιξη του χρόνου



στην πραγματικότητα...

Φυσική Ημιαγωγών – σε δύο διαφάνειες!

- Τεχνολογία

- Ημιαγωγοί
 - Στοιχεία με κρυσταλλική δομή (πυρίτιο, γερμάνιο...)
 - Μεταξύ αγωγών και μονωτών
 - Σε θερμοκρασία δωματίου και καθαρή μορφή: όχι καλοί αγωγοί του ηλεκτρισμού
 - Προσμίξεις με “ακάθαρτα” υλικά (impurities)
 - “Doping”
 - Προσθήκη ελεύθερων ηλεκτρονίων (donors)
 - n-type
 - Προσθήκη “οπών” (acceptors)
 - απουσία ηλεκτρονίων
 - p-type
 - Τα χαρακτηριστικά αγωγιμότητας αλλάζουν!

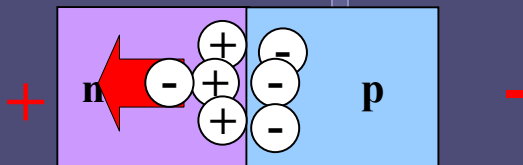
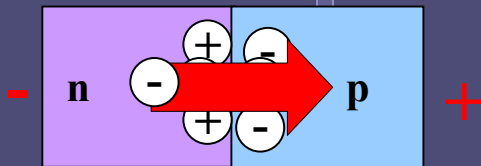
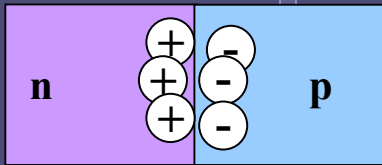
Φυσική Ημιαγωγών – σε δύο διαφάνειες!

• Τεχνολογία

• Το ενδιαφέρον σημείο (επαφής)

▪ Όταν ενώνονται ημιαγωγοί n-type και p-type

- Στο σημείο επαφής: κατάσταση ισορροπίας – μετά από την αρχική μετακίνηση ηλεκτρονίων στο p-type, περαιτέρω ηλεκτρόνια απωθούνται



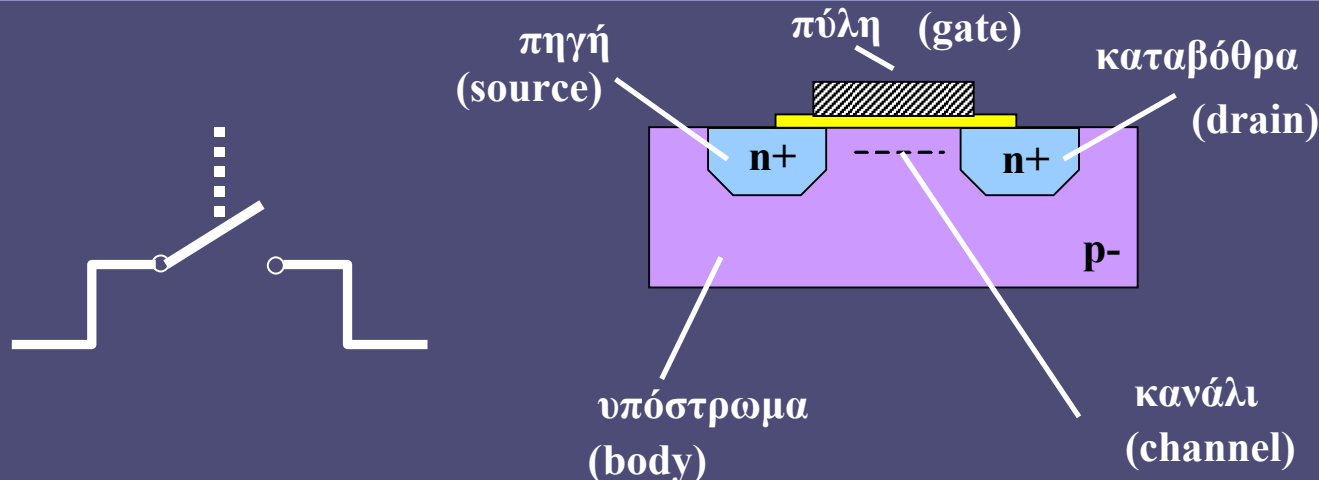
- Με ορθή πόλωση, και άλλα ηλεκτρόνια μπορούν να υπερπηδήσουν το εμπόδιο και να φτάσουν στο p-type, οπότε παρατηρείται ροή ρεύματος
-
- Με ανάστροφη πόλωση, νέα ηλεκτρόνια δεν έχουν την ευκαιρία να φτάσουν στο p-type, οπότε δεν ρέει ρεύμα μέσω της επαφής
 - την ανάστροφη πόλωση εκμεταλλεύονται τα ηλεκτρονική ψηφιακά κυκλώματα των υπολογιστών

Γιατί δυαδική τεχνολογία;

- Τεχνολογία

i

Το τρανζίστορ αυτό είναι τύπου NMOS. Υπάρχει και το συμπληρωματικό PMOS.



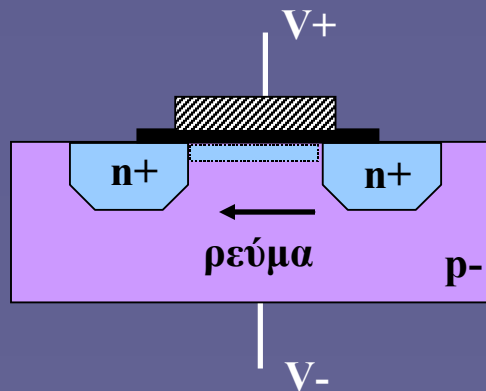
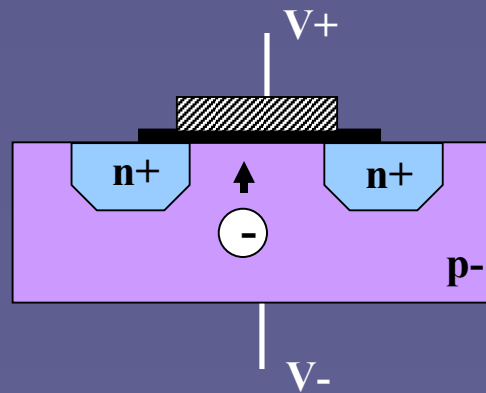
- Το τρανζίστορ MOSFET
 - Ο μικροσκοπικός διακόπτης των σύγχρονων κυκλωμάτων
 - Η θεωρία λειτουργίας του είναι γνωστή από το 1925...
 - ...αλλά τα πρώτα λειτουργικά κυκλώματα κατασκευάστηκαν τη δεκαετία του 60

Λειτουργία του τρανζίστορ MOS(FET)

- Τεχνολογία

;

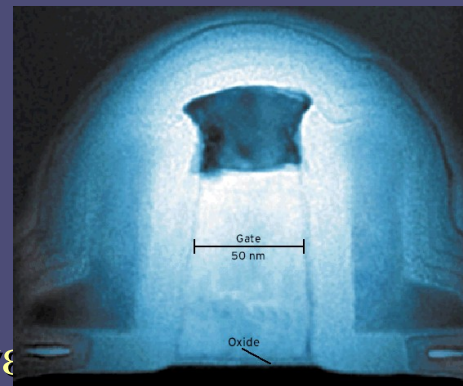
Τι συμβαίνει στο τρανζίστορ PMOS;



Η συρρίκνωση του τρανζίστορ

- Τεχνολογία

- Ένας σύγχρονος επεξεργαστής μπορεί να περιέχει πάνω από 1 δις τρανζίστορ σε μία επιφάνεια 250mm^2
- Πλεονεκτήματα
 - Ταχύτερη λειτουργία
 - Πιο γρήγοροι χρόνοι ON-OFF
 - Μικρότερη κατανάλωση ενέργειας
 - Για τον ίδιο αριθμό τρανζίστορ!
 - Μεγαλύτερη ολοκλήρωση
 - Μείωση κόστους παραγωγής και αύξηση λειτουργικότητας
- Τρέχουσα εμπορική τεχνολογία:
 - Μέγεθος (καναλιού) τρανζίστορ = 65nm
- Το μέλλον; 45 και 25 nm

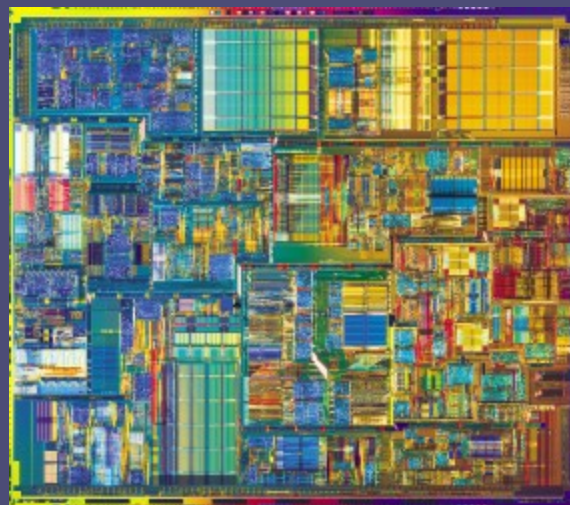


Το (μικρο)τσιπ

- Τεχνολογία

- Ολοκληρωμένο κύκλωμα
 - **micro(chip)**
 - όλα τα στοιχεία συνδυάζονται στο ίδιο υπόστρωμα πυριτίου.
 - Τρανζίστορ , αγωγοί
 - Πυκνωτές, αντιστάσεις

**Intel P4 processor
microchip (2005)
169 εκ. τρανζίστορ**



- Νόμος του Moore (εμπειρικός)
 - ο αριθμός των τρανζίστορ ανά ολοκληρωμένο κύκλωμα διπλασιάζεται κάθε 1,5-2 χρόνια

Κατασκευή ολοκληρωμένων κυκλωμάτων

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα

i

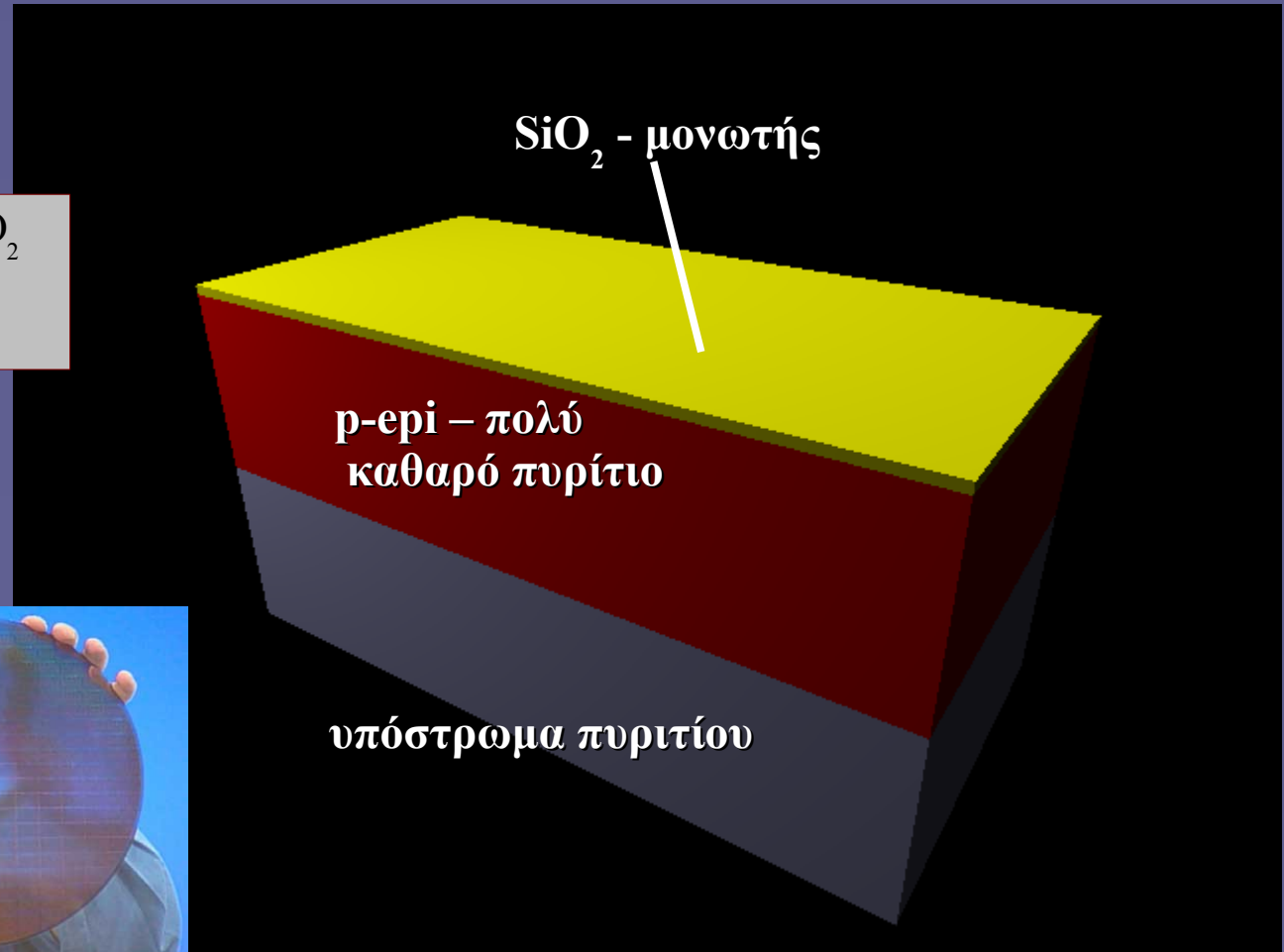
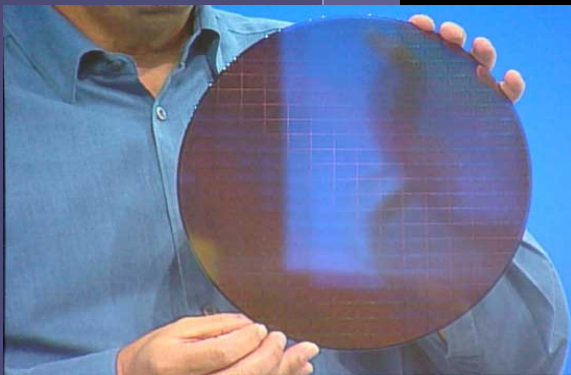
Λόγω της απαιτούμενης ακρίβειας, μια γραμμή παραγωγής κοστίζει δις. \$

- Γραμμές παραγωγής
 - Φωτολιθογραφία με μάσκες
 - Επικάλυψη με ειδικό φωτοανθεκτικό υλικό (photoresist)
 - Έκθεση σε υπεριώδες φως (με το ανάλογο μήκος κύματος)
 - Μέσω μιας μάσκας που επιλέγει τις περιοχές επεξεργασίας
 - Απομάκρυνση photresist από επιλεγμένες περιοχές, αφήνοντας εκτεθειμένα τα μέρη προς επεξεργασία
 - Διεργασίες στα εκτεθειμένα μέρη
 - Οξείδωση, απόξεση, απόθεση μετάλλου, εμφύτευση ιόντων...
 - Ταυτόχρονα σε εκατομμύρια τρανζίστορ!
 - Επανάληψη
 - Από το βήμα της μάσκας

Η αρχική επιφάνεια

- Τεχνολογία

Ένα λεπτό επίπεδο SiO_2
σχηματίζεται με
οξείδωση σε 1000°C

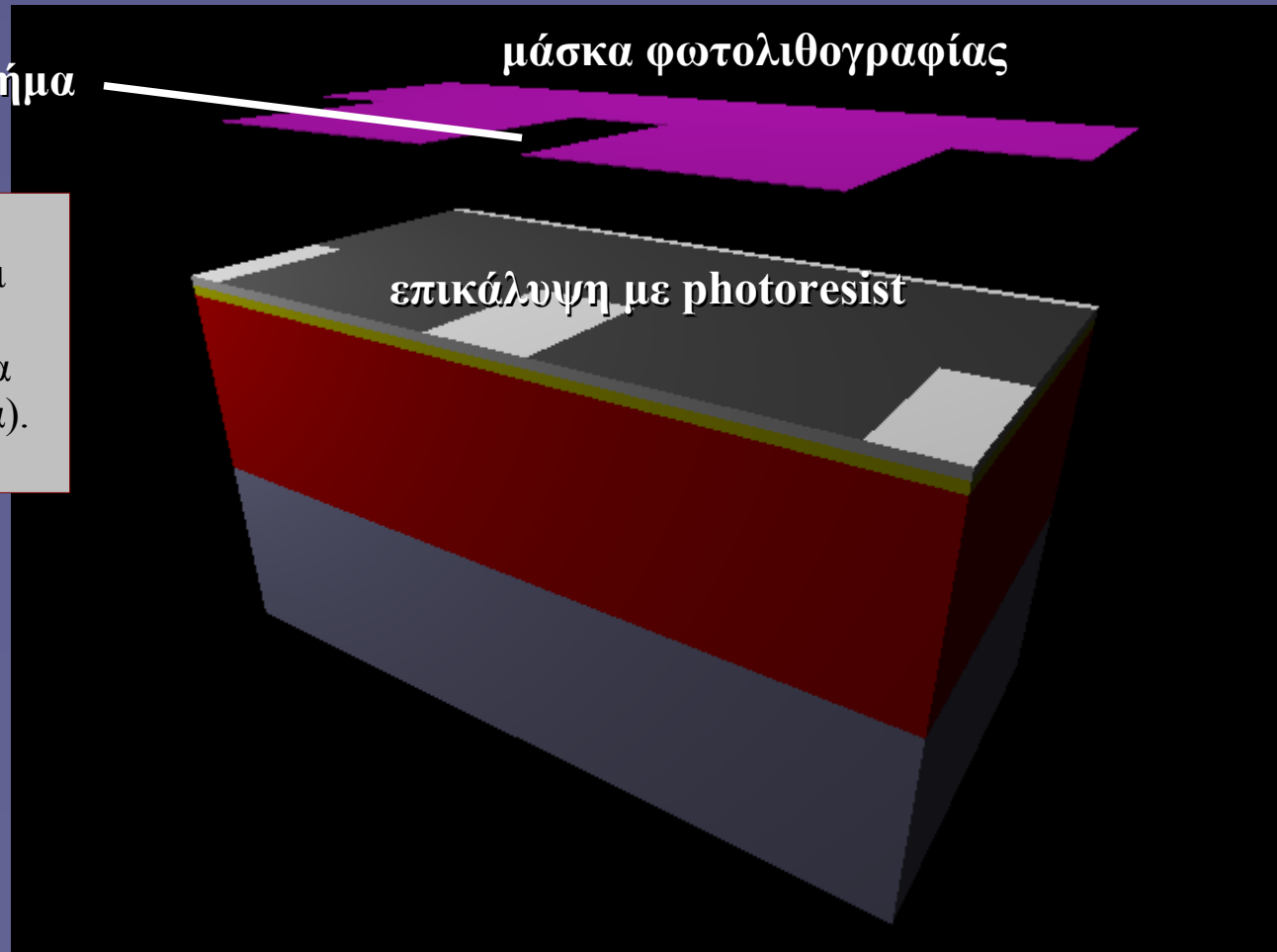


Εφαρμογή photoresist και μάσκας

- Τεχνολογία

εκτεθειμένο τμήμα

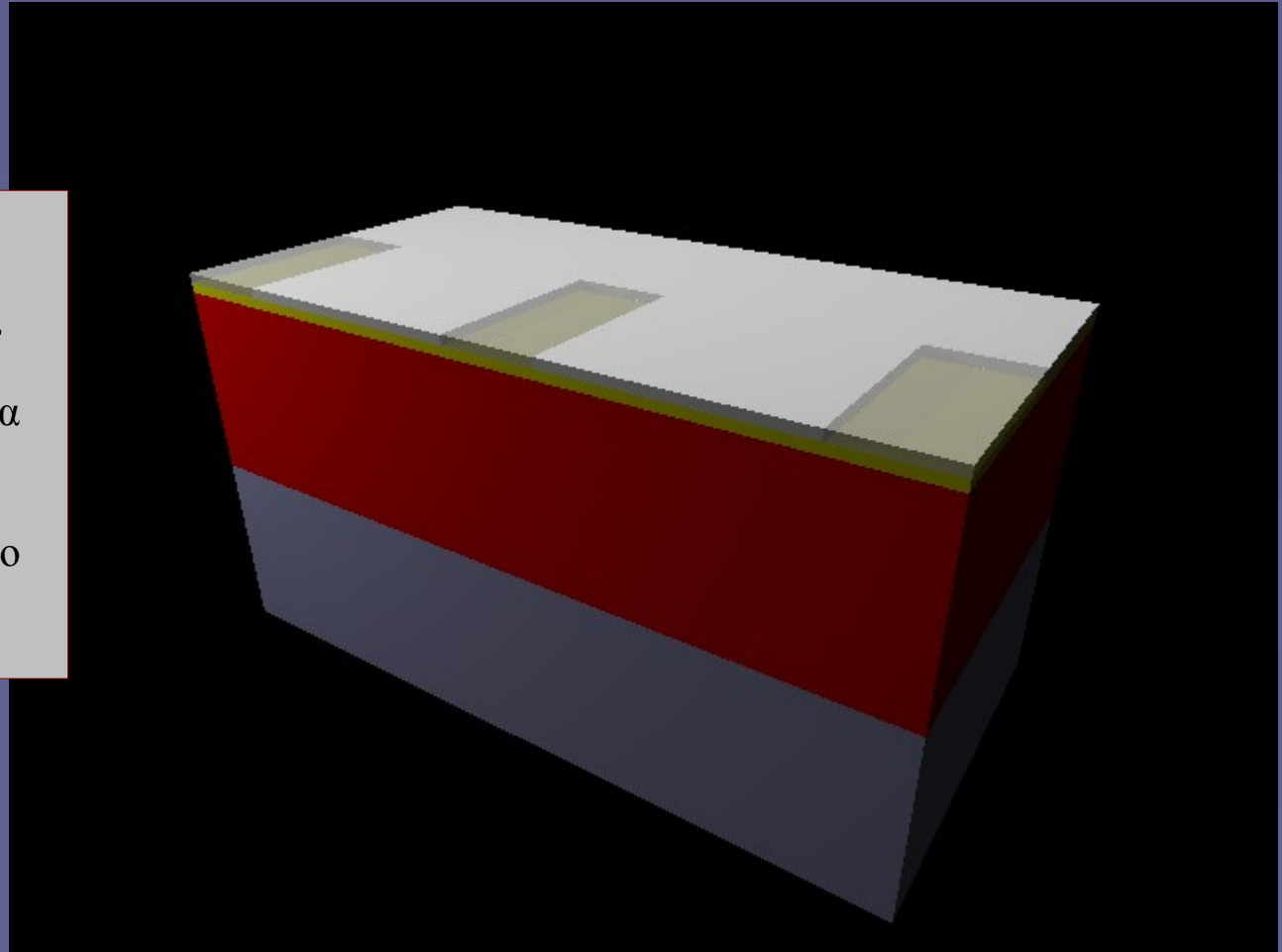
Η μάσκα της φωτολιθογραφίας είναι διαφορετική ανά βήμα επεξεργασίας (ανάλογα με το επιθυμητό σχήμα).



Εφαρμογή υπεριώδους ακτινοβολίας

- Τεχνολογία

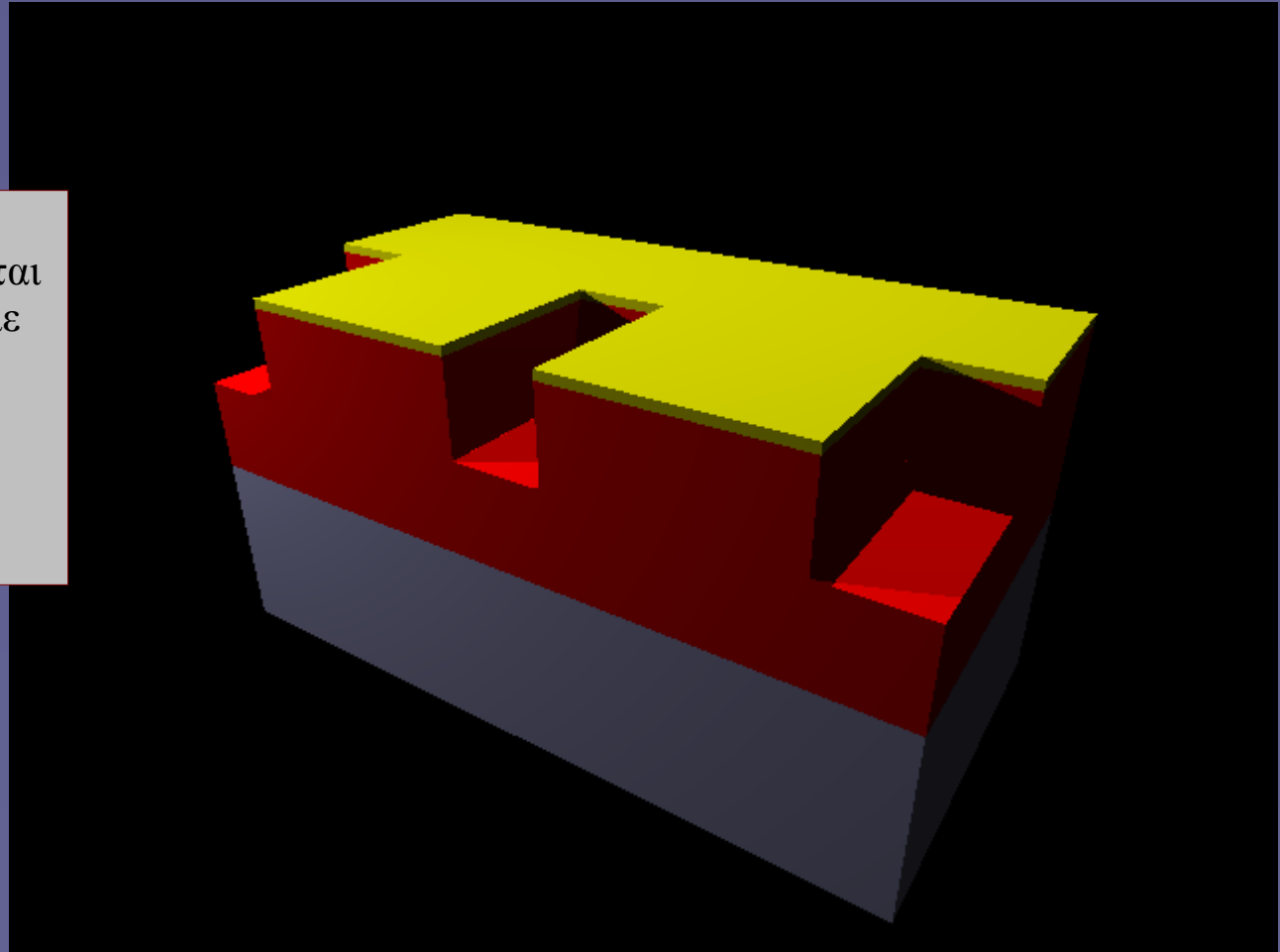
Στα μέρη που μένουν εκτεθειμένα μετά την υπεριώδη ακτινοβολία, το photoresist γίνεται εύπλαστο. Στη συνέχεια αφαιρείται με χημικό τρόπο, αφήνοντας εκτεθειμένα μέρη για το επόμενο βήμα κατεργασίας.



Μετά την απόξεση

- Τεχνολογία

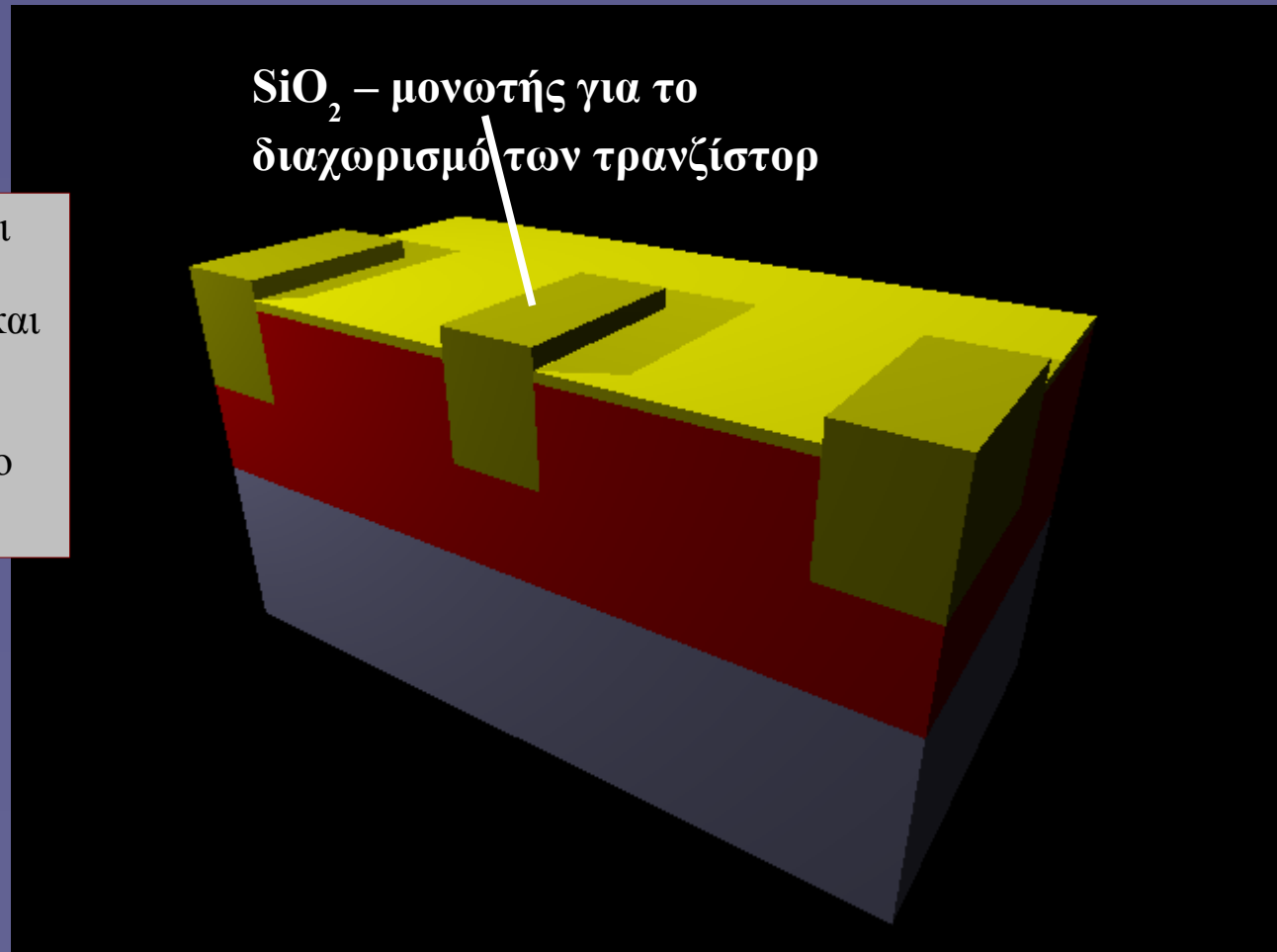
Στα μέρη που μένουν εκτεθειμένα εφαρμόζεται διαδικασία απόξεσης με τη βοήθεια οξέων. Στη συνέχεια η επιφάνεια καθαρίζεται με απιονισμένο νερό και στεγνώνει με άζωτο.



Απόθεση νέων στρωμάτων SiO_2

- Τεχνολογία

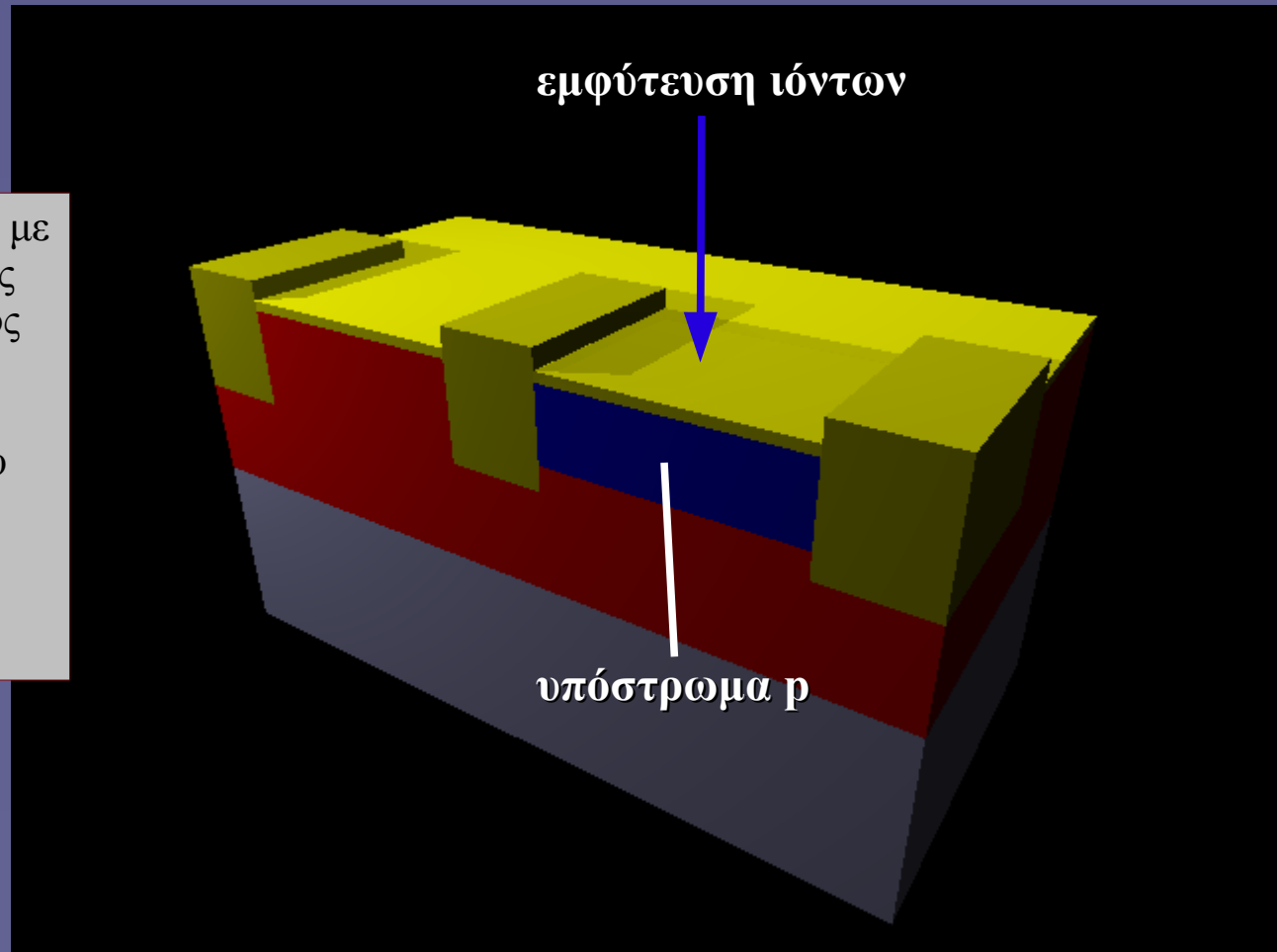
Πριν το βήμα αυτό έχει προηγηθεί πάλι η εφαρμογή photoresist και μάσκας! Στο εξής η εφαρμογή μάσκας θα εννοείται πριν κάθε νέο βήμα.



Doping – Υπόστρωμα τρανζίστορ

- Τεχνολογία

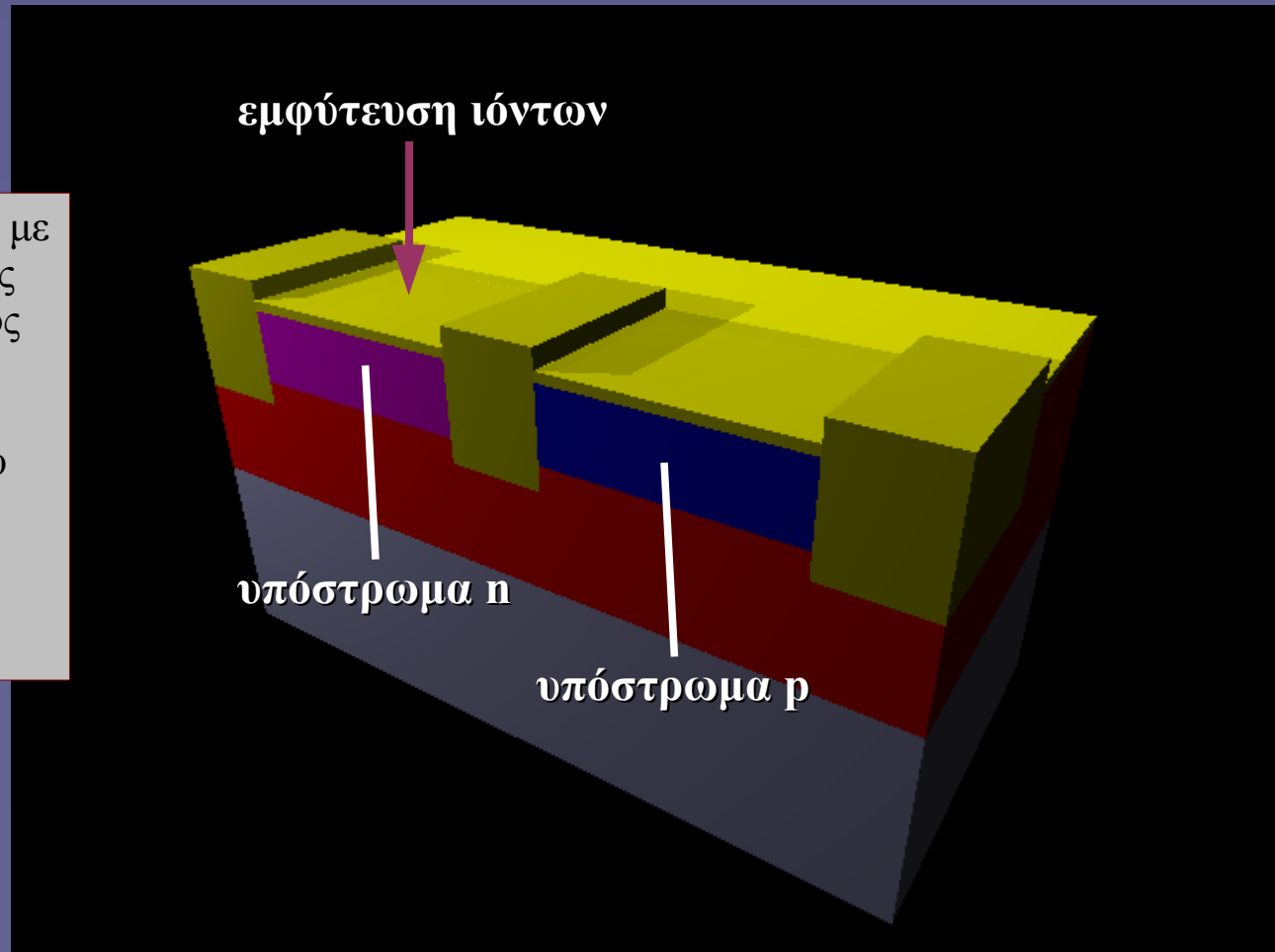
Μέσω της εμφύτευσης με τη βοήθεια μιας δέσμης ιόντων, αλλάζει ο τύπος του ημιαγωγού της περιοχής κάτω από το λεπτό στρώμα οξειδίου σχηματίζοντας το υπόστρωμα των τρανζίστορ NMOS (doping)



Doping – Υπόστρωμα τρανζίστορ

- Τεχνολογία

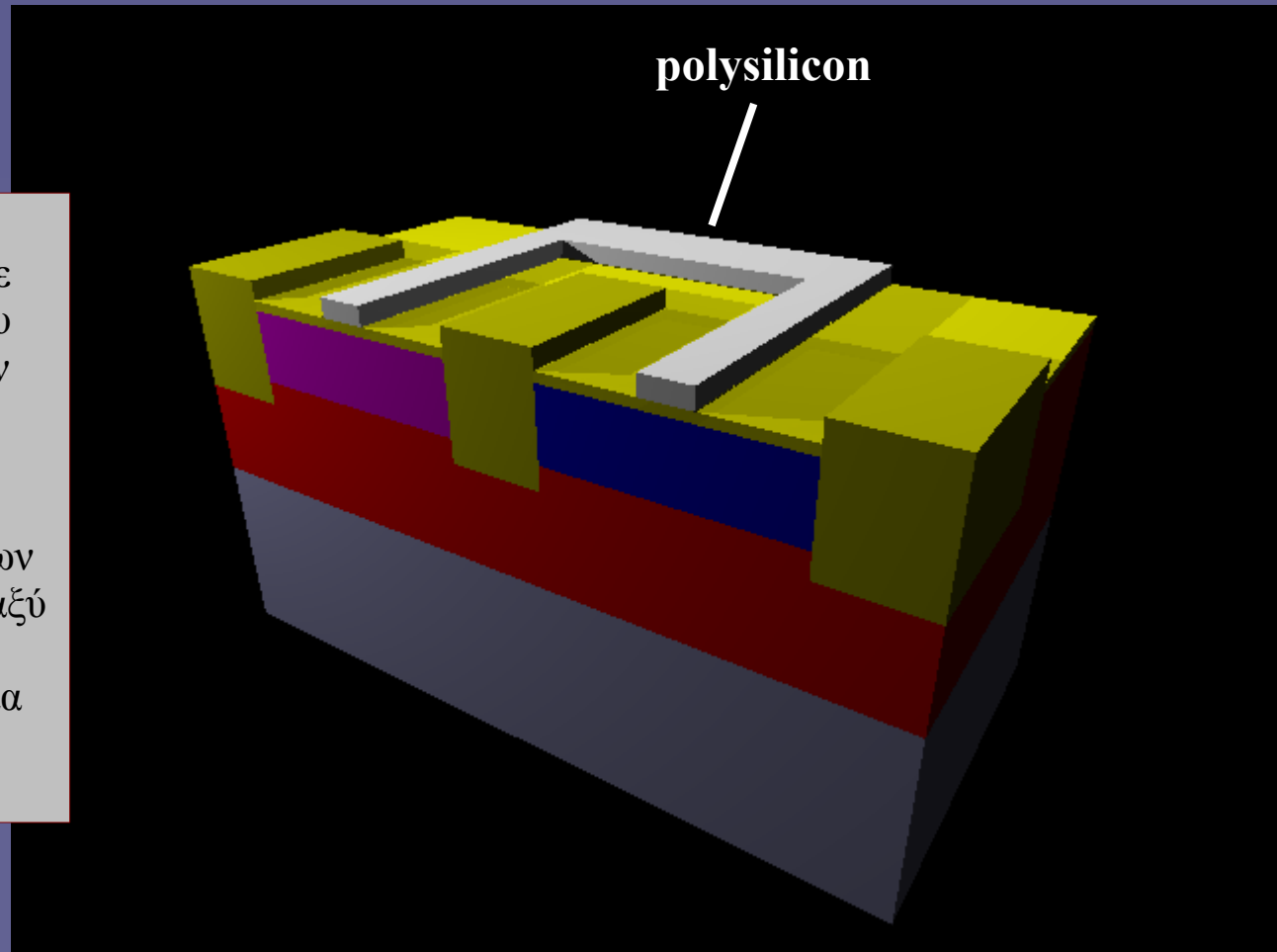
Μέσω της εμφύτευσης με τη βοήθεια μιας δέσμης ιόντων, αλλάζει ο τύπος του ημιαγωγού της περιοχής κάτω από το λεπτό στρώμα οξειδίου σχηματίζοντας το υπόστρωμα των τρανζίστορ PMOS (doping)



Εναπόθεση polysilicon

- Τεχνολογία

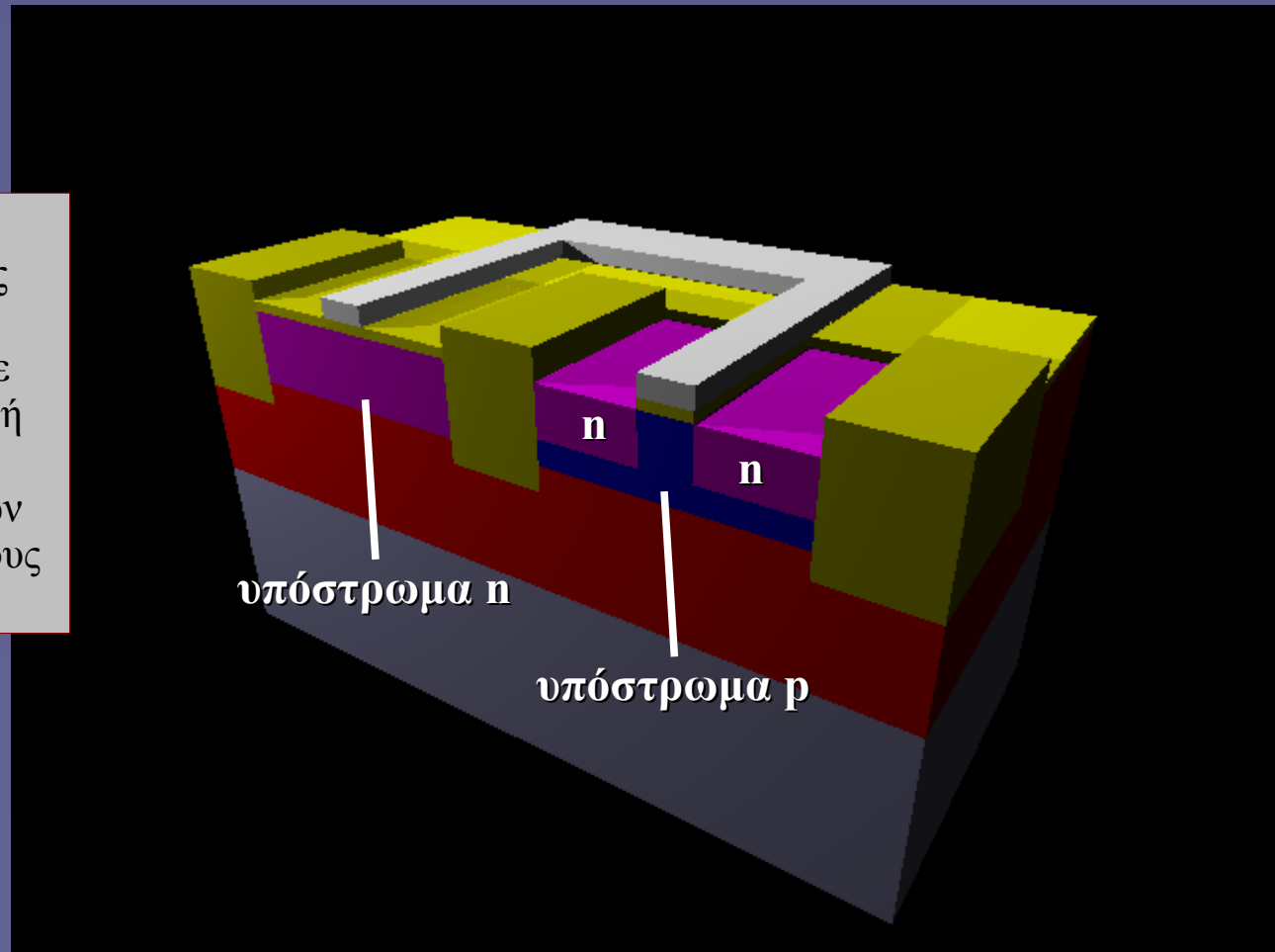
Ένα σχήμα polysilicon εναποτίθεται χημικά με την κυκλοφορία αερίου μίγματος πάνω από την επιφάνεια πυριτίου θερμαινόμενη στους 650°C. Το σχήμα σχηματίζει τις πύλες των τρανζίστορ και τη μεταξύ τους διασύνδεση. Θα ακολουθήσει doping για να αυξηθεί η αγωγιμότητά του.



Doping πηγής και καταβόθρας τρανζίστορ

- Τεχνολογία

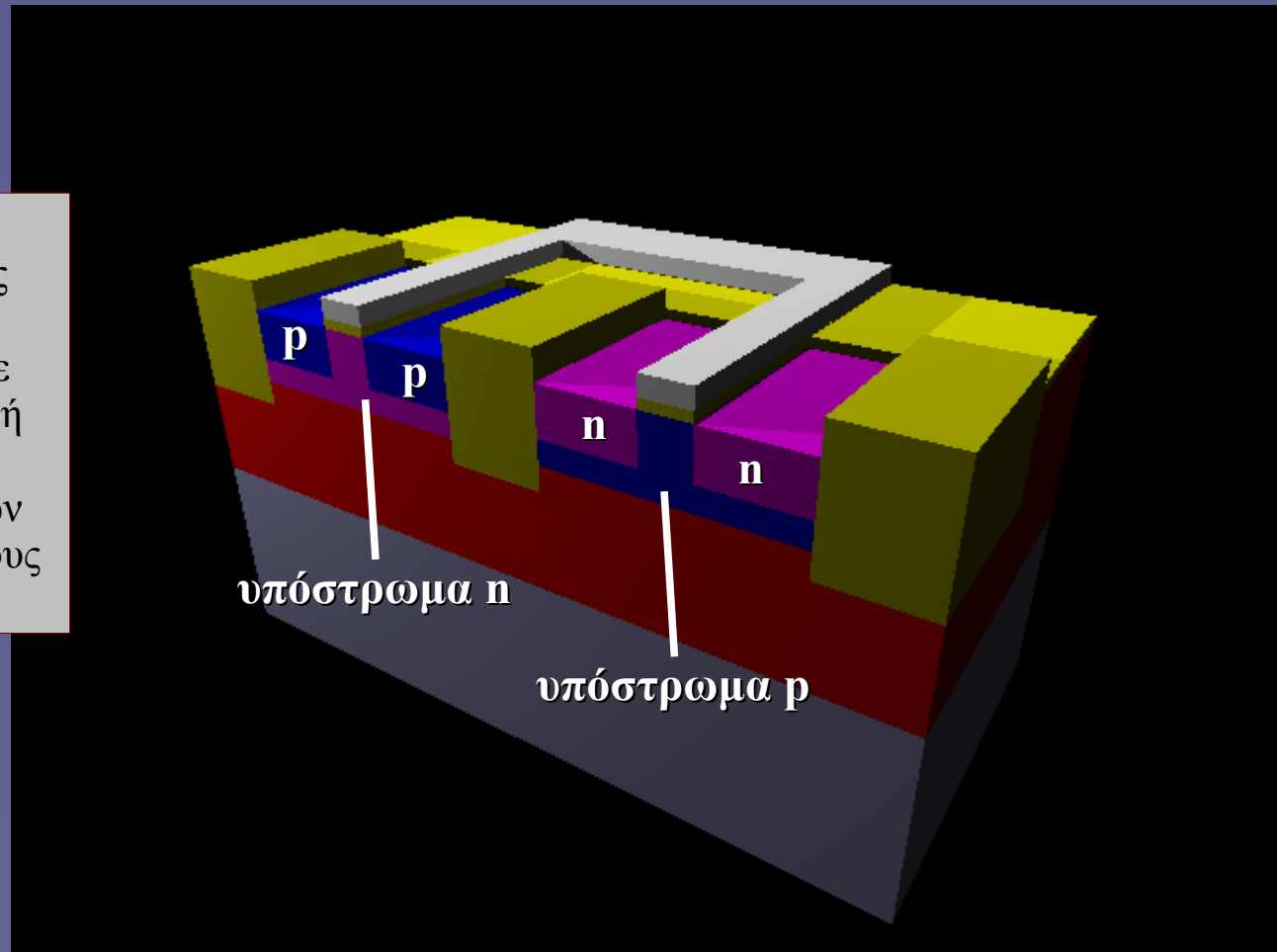
Στη συνέχεια σχηματίζονται οι πηγές και καταβόθρες των τρανζίστορ ανάλογα με τον τύπο τους (NMOS ή PMOS) με εμφύτευση ιόντων. Δημιουργία των σημείων επαφής για τους αγωγούς διασύνδεσης.



Doping πηγής και καταβόθρας τρανζίστορ

- Τεχνολογία

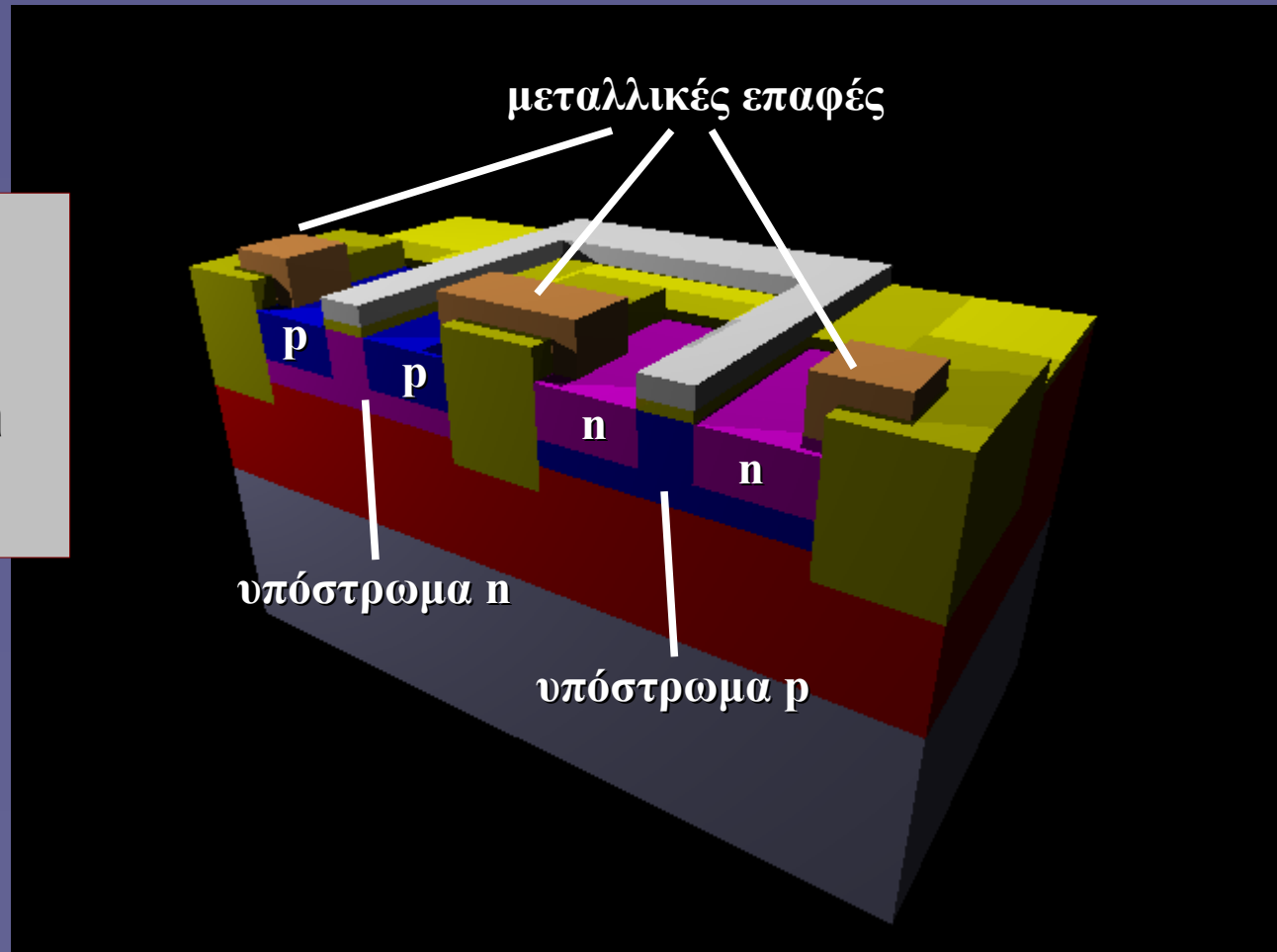
Στη συνέχεια σχηματίζονται οι πηγές και καταβόθρες των τρανζίστορ ανάλογα με τον τύπο τους (NMOS ή PMOS) με εμφύτευση ιόντων. Δημιουργία των σημείων επαφής για τους αγωγούς διασύνδεσης.



Πρώτο επίπεδο μεταλικών συνδέσεων

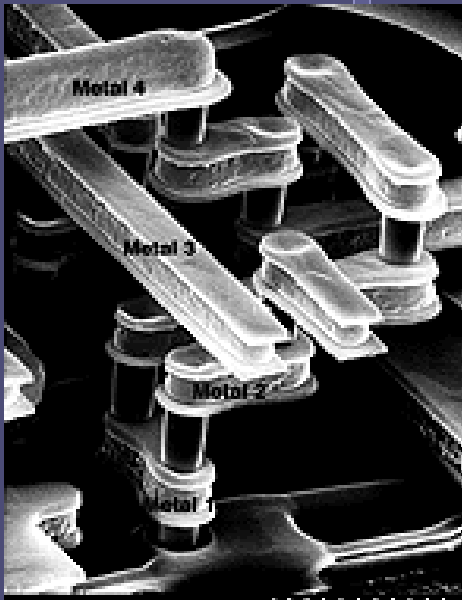
- Τεχνολογία

Τα επίπεδα μετάλλου δημιουργούνται με εξάχνωση του μεταλλικού υλικού σε κενό υπό την επίδραση δέσμης ηλεκτρονίων.



Διαδικασία παραγωγής

- Τεχνολογία

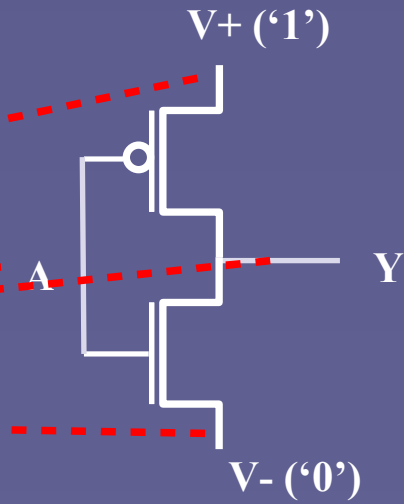
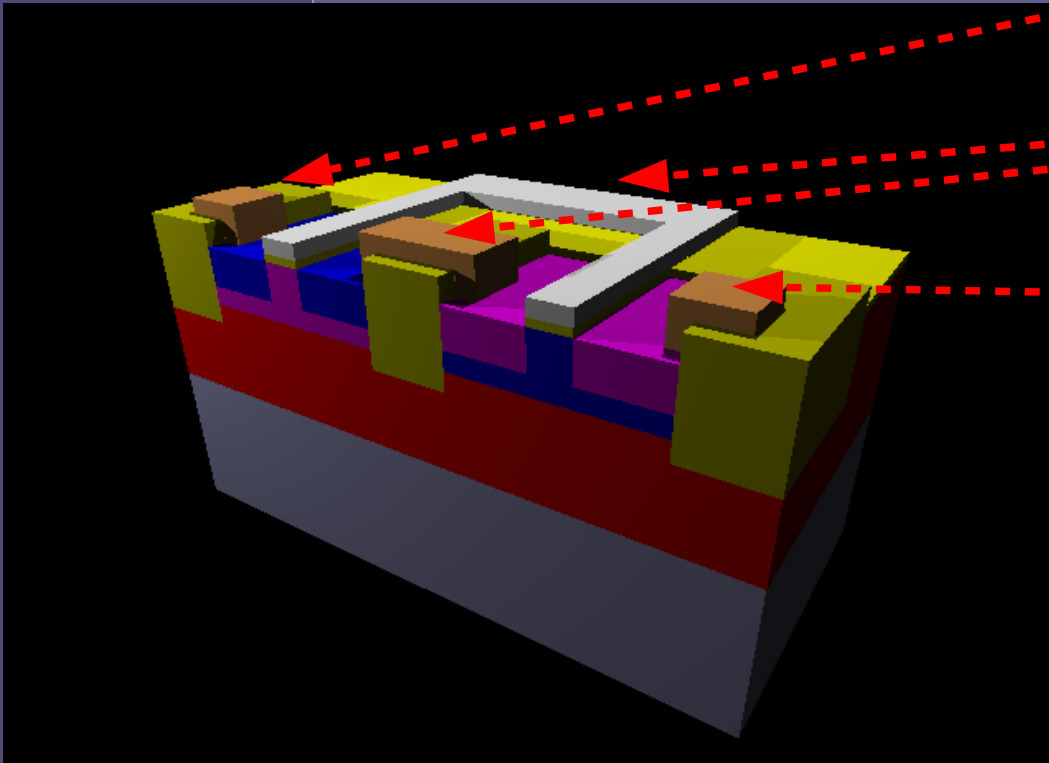


[Rabaey 2003]

- Στην πραγματικότητα
 - Οι δημιουργούμενες επιφάνειες δεν είναι απόλυτα επίπεδες – ούτε έχουν κάθετες γωνίες
 - τα χαρακτηριστικά είναι πιο ακανόνιστα
 - Οι αναλογίες διαστάσεων είναι διαφορετικές
 - ειδικά μελετημένες για την επιθυμητή ροή ηλεκτρονίων
 - Τα επίπεδα μετάλλου είναι πολύ περισσότερα
 - καταλαμβάνουν μεγάλο μέρος του ολοκληρωμένου κυκλώματος
 - Τα βήματα κατασκευής είναι πολύ περισσότερα
 - από την απλουστευμένη εικόνα που είδαμε
 - οι δομές που κατασκευάζονται είναι πολυπλοκότερες

Ποια η λειτουργία του;

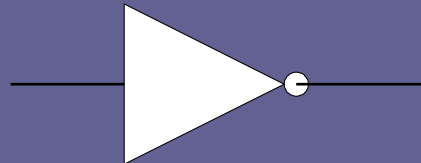
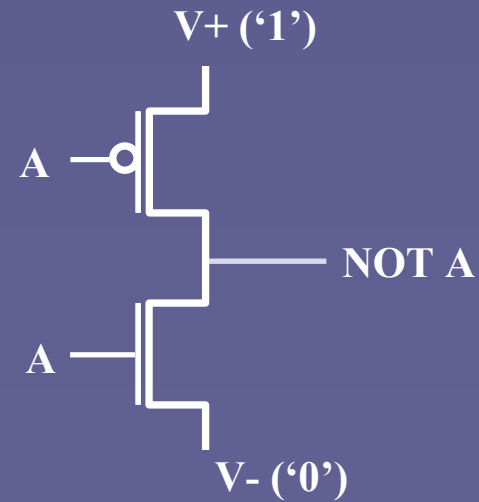
- Τεχνολογία



Η πύλη NOT

- Τεχνολογία

A	Y
0	1
1	0



Στο επόμενο μάθημα...

- Τεχνολογία
- Στο επόμενο...

- Ψηφιακά κυκλώματα για λογικές και αριθμητικές πράξεις

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2007-08

Πράξεις με δυαδικούς αριθμούς

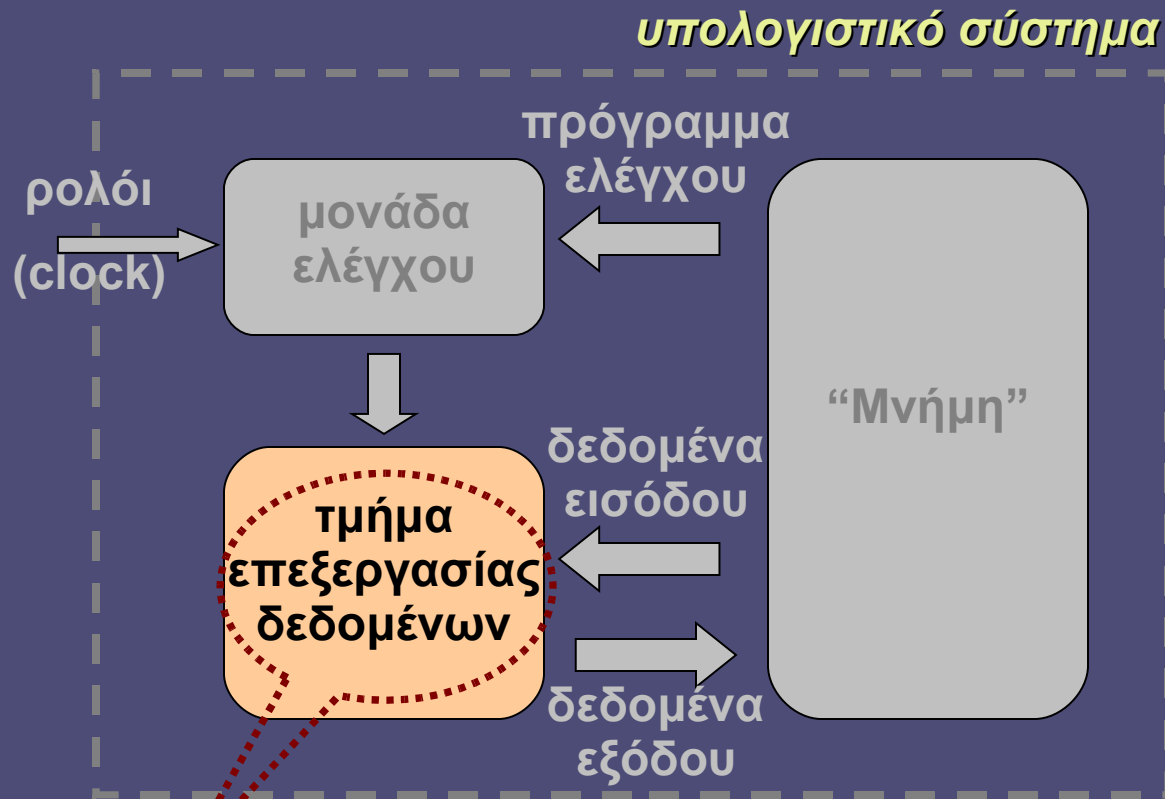
(λογικές πράξεις)

<http://www.ionio.gr/~mistral/tp/csintro/>

Μ.Στεφανιδάκης

Εκτέλεση πράξεων

- Εισαγωγή



;

Ποιες κατηγορίες πράξεων;

- Συγκροτείται από ψηφιακά δυαδικά κυκλώματα
 - Εκτελούν λογικές πράξεις μεταξύ σειρών bits...
 - ...οι οποίες αναπαριστούν δυαδικούς αριθμούς

Πράξεις με δυαδικούς αριθμούς

- Εισαγωγή

- Ο υπολογιστής μπορεί να εκτελέσει
 - Λογικές πράξεις
 - Αριθμητικές πράξεις
- Οι πράξεις εκτελούνται
 - Σε ομάδες bits: “δυαδικούς αριθμούς”
 - bytes ή πολλαπλάσιά τους

Bits & Bytes

- Εισαγωγή

- **Bit**

- Η **μικρότερη** λογική ποσότητα - η μικρότερη μονάδα δεδομένων - 0 ή 1.

- **Byte**

- Ομάδα 8 bits
- Η **ελάχιστη** ποσότητα που μπορεί να χειριστεί ο υπολογιστής κατά την εκτέλεση μιας πράξης
- Μια σειρά από bytes αναπαριστά έναν δυαδικό “αριθμό”

Η ανατομία ενός byte

- Εισαγωγή

το περισσότερο
σημαντικό bit

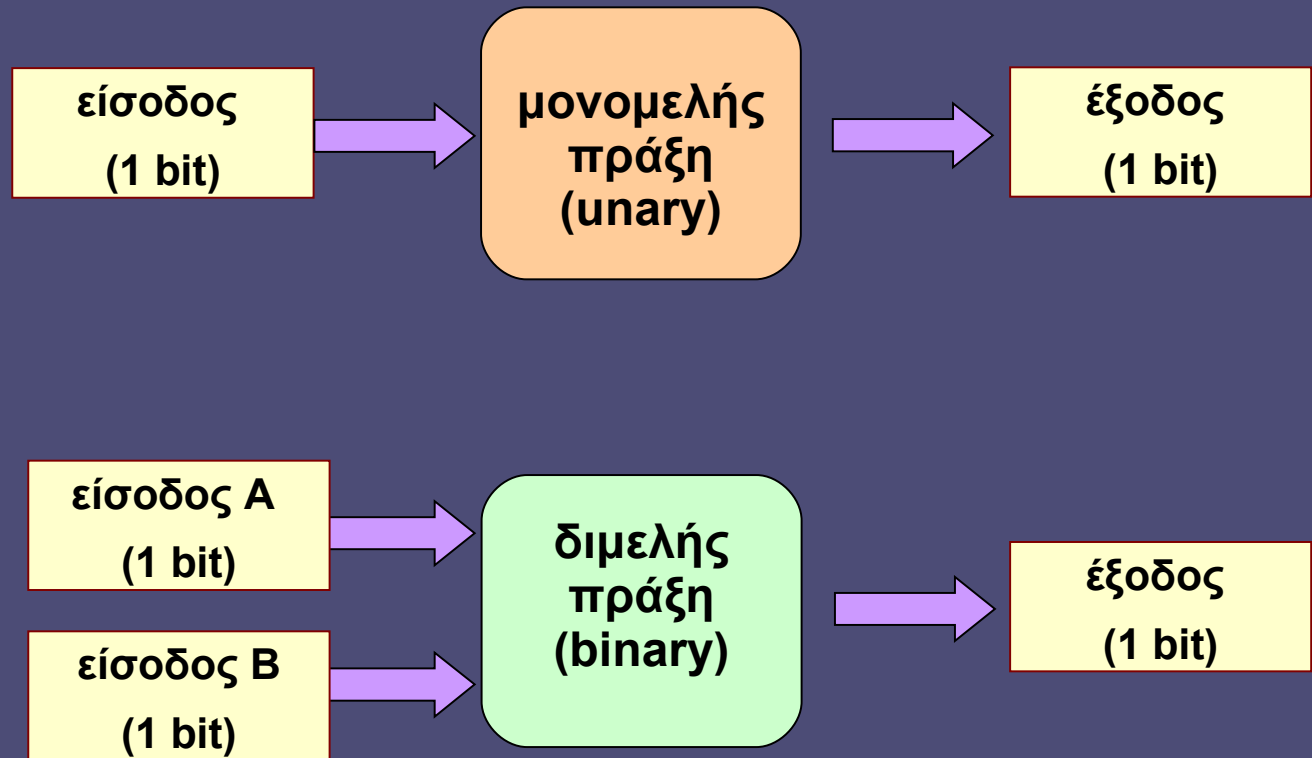
το λιγότερο
σημαντικό
bit



- Γιατί είναι αυτή η σειρά των bits;
 - Γιατί το λιγότερο σημαντικό bit είναι δεξιά και το περισσότερο σημαντικό αριστερά;
 - Θα φανεί όταν μιλήσουμε για **αριθμητικές πράξεις**
 - Προς το παρόν, το byte είναι απλώς μια οκτάδα bits

Λογικές πράξεις με bits

- Εισαγωγή
- Λογικές πράξεις



- Πώς εκτελείται η πράξη σε ομάδες bits;

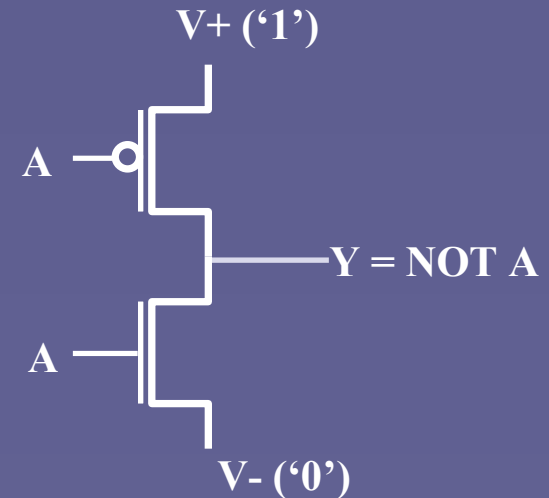
Λογικές πράξεις με bits

- Εισαγωγή
- Λογικές πράξεις

- **Μονομελής λογική πράξη**
 - NOT (αντιστροφή)
- **Διμελείς λογικές πράξεις**
 - AND (λογικό-ΚΑΙ)
 - OR (λογικό-Η)
 - XOR (αποκλειστικό-Η)
- **Πράξεις σε ομάδες bits**
 - Εφαρμογή της πράξης στα αντίστοιχα bits
 - Διμελείς πράξεις
 - Μεταξύ των αντίστοιχων bits των δύο αριθμών
 - Σε όλα τα bytes των αριθμών

Από το προηγούμενο: η πύλη NOT

- Εισαγωγή
- Λογικές πράξεις



Δυαδική λογική: Αντιστροφή bits

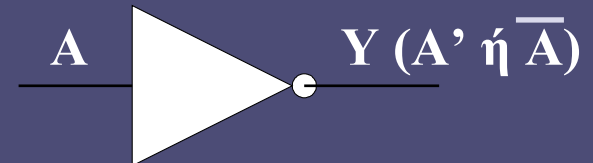
- Εισαγωγή
- Λογικές πράξεις

- Αντιστροφή (NOT)
 - Αντιστροφή των bits

Πίνακας Αλήθειας

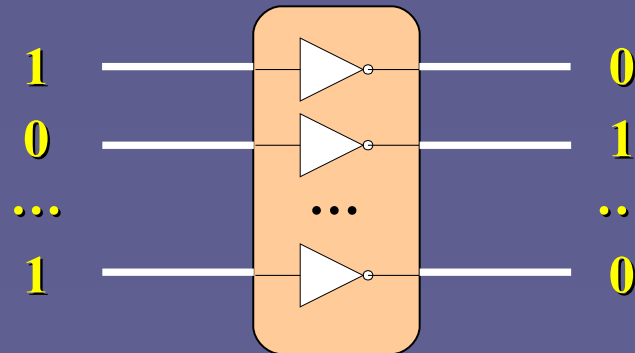
A	Y
0	1
1	0

Σύμβολο πύλης NOT



Ο τελεστής NOT σε δυαδικούς αριθμούς

- Εισαγωγή
- Λογικές πράξεις



!

Η “μέθοδος” του υλικού (hardware): πολλαπλές ίδιες μονάδες εκτελούν την ίδια λειτουργία παράλληλα

1	0	0	1	1	0	0	0	NOT
<hr/>								
0	1	1	0	0	1	1	1	

- Η έξοδος Y_i εξαρτάται μόνο από την είσοδο A_i

Δυαδική λογική: AND

- Εισαγωγή
- Λογικές πράξεις

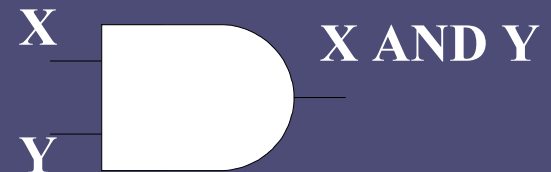
- Λογικό ΚΑΙ (AND)

- αποτέλεσμα 1, μόνο όταν και το X και το Y είναι 1
- $0 \text{ AND } X = X \text{ AND } 0 = 0$
- $1 \text{ AND } X = X \text{ AND } 1 = X$

Πίνακας Αλήθειας

X	Y	AND
0	0	0
0	1	0
1	0	0
1	1	1

Σύμβολο πύλης AND



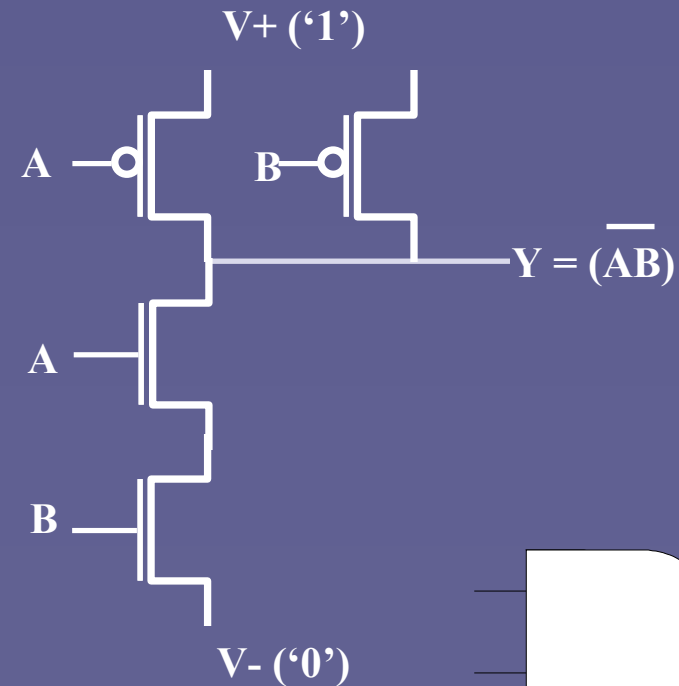
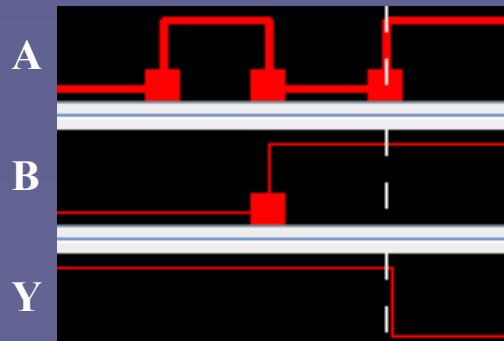
Παράδειγμα υλοποίησης: η πύλη NAND

- Εισαγωγή
- Λογικές πράξεις

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

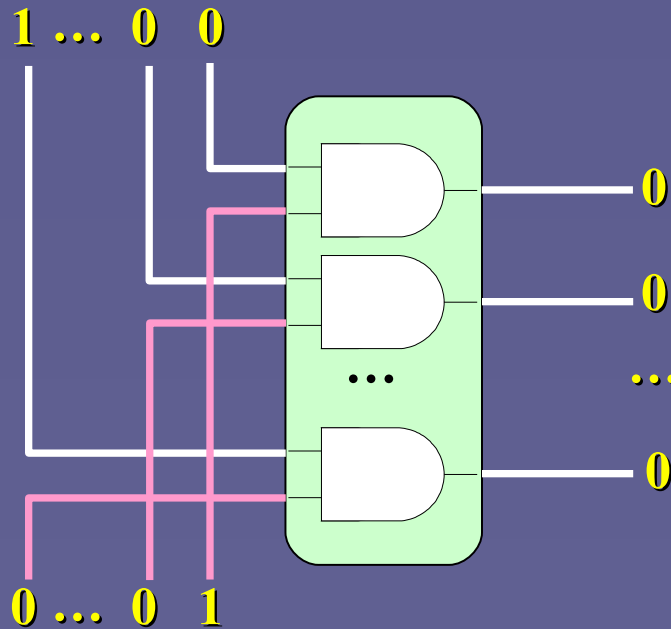
!

Υλοποίηση πύλης AND: χρησιμοποιώντας μια πύλη NAND και μια πύλη NOT



Ο τελεστής AND σε δυαδικούς αριθμούς

- Εισαγωγή
- Λογικές πράξεις



```
1 0 0 1 1 0 0 0   AND
0 0 1 1 0 1 0 1
-----
0 0 0 1 0 0 0 0
```


Μάσκα AND: για να θέσουμε bits στο 0

- Εισαγωγή
- Λογικές πράξεις

- Ζητούμενο: σε λέξη των 8 bits να τεθούν σε 0 τα 3 λιγότερο σημαντικά bits.

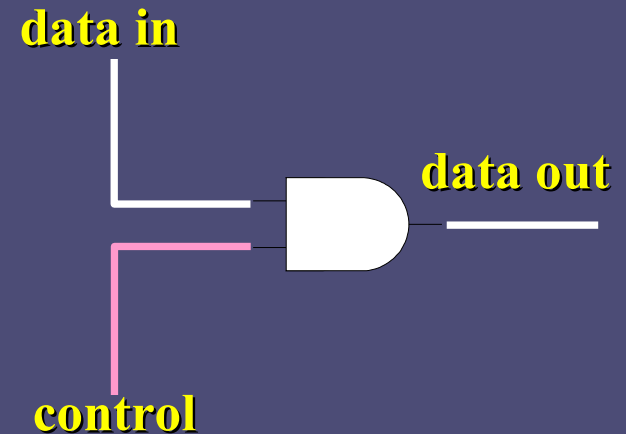
Στόχος:	1 0 0 1 1 0 0 0	AND
Μάσκα:	1 1 1 1 1 0 0 0	
Έξοδος:	1 0 0 1 1 0 0 0	

- Η AND μάσκα περιέχει:
 - 0 στα bits που θα γίνουν 0
 - 1 στα bits που θα παραμείνουν ως έχουν

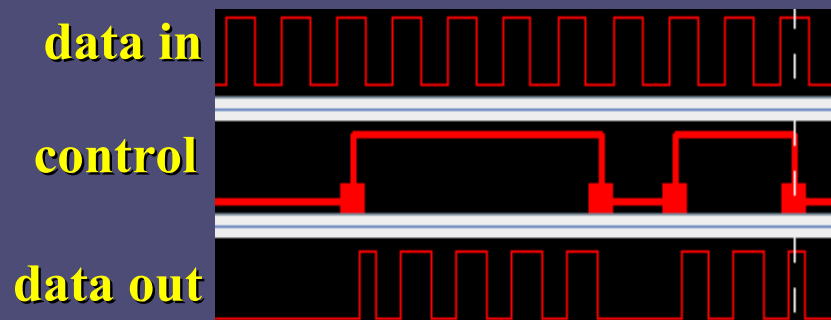
0 AND X = 0
1 AND X = X

Φραγή AND: για να θέσουμε σήμα στο 0

- Εισαγωγή
- Λογικές πράξεις



$$0 \text{ AND } X = 0$$
$$1 \text{ AND } X = X$$



Δυαδική λογική: OR

- Εισαγωγή
- Λογικές πράξεις

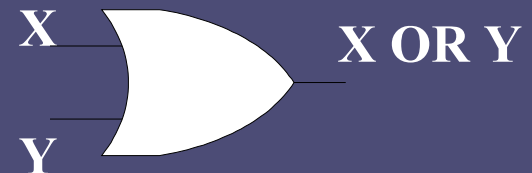
- Λογικό Ή (OR)

- αποτέλεσμα 1, όταν X ή Y ή και τα δύο είναι 1
- $1 \text{ OR } X = X \text{ OR } 1 = 1$
- $0 \text{ OR } X = X \text{ OR } 0 = X$

Πίνακας Αλήθειας

X	Y	OR
0	0	0
0	1	1
1	0	1
1	1	1

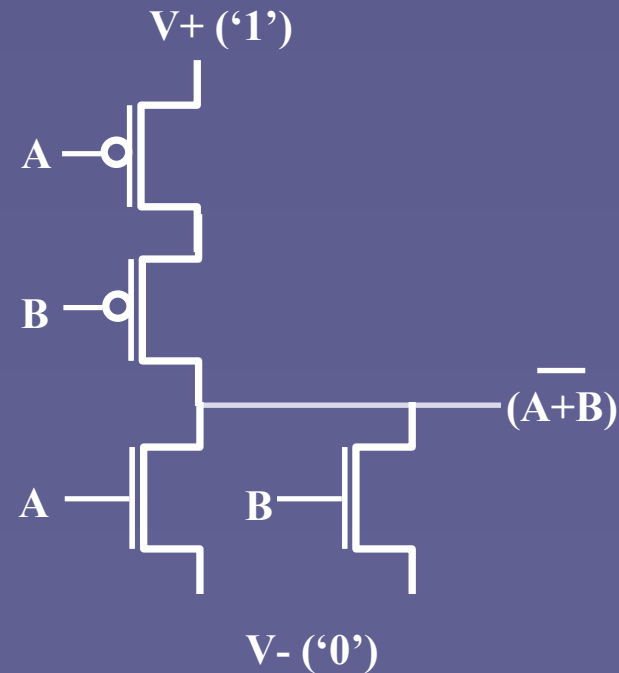
Σύμβολο πύλης OR



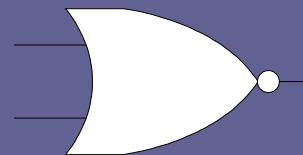
Παράδειγμα υλοποίησης: η πύλη NOR

- Εισαγωγή
- Λογικές πράξεις

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

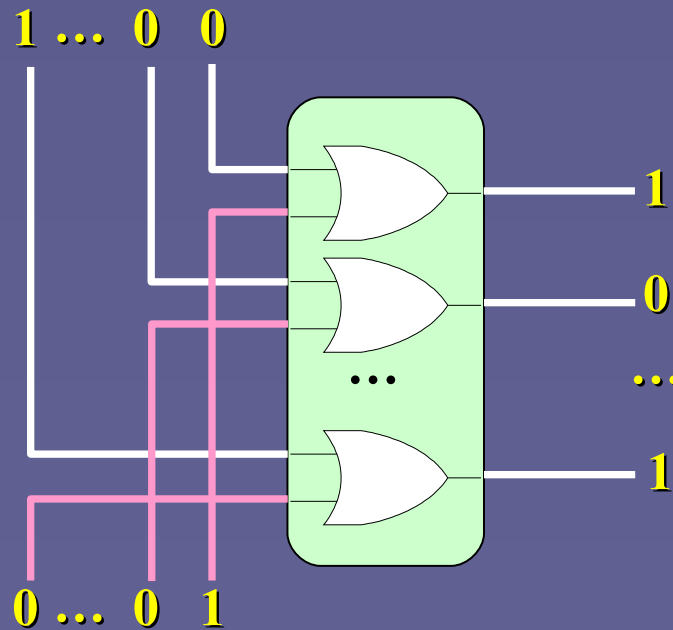


Υλοποίηση πύλης
OR: χρησιμοποι-
ώντας μια πύλη
NOR και μια πύλη
NOT



Ο τελεστής OR σε δυαδικούς αριθμούς

- Εισαγωγή
- Λογικές πράξεις



```
1 0 0 1 1 0 0 0   OR
0 0 1 1 0 1 0 1
-----
1 0 1 1 1 1 0 1
```

Μάσκα OR: για να θέσουμε bits στο 1

- Εισαγωγή
- Λογικές πράξεις

- Ζητούμενο: σε λέξη των 8 bits να τεθούν σε 1 τα bits 0,4 και 5.

Στόχος:	1 0 0 1 1 0 0 0	OR
Μάσκα:	0 0 1 1 0 0 0 1	
Έξοδος:	1 0 1 1 1 0 0 1	

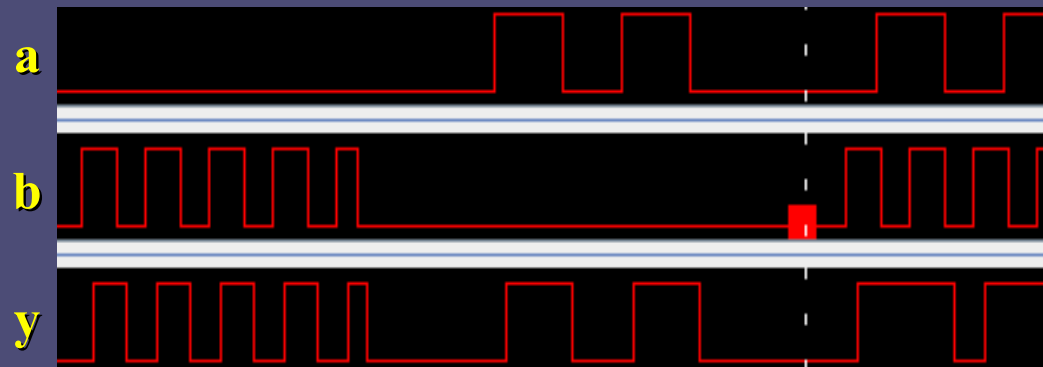
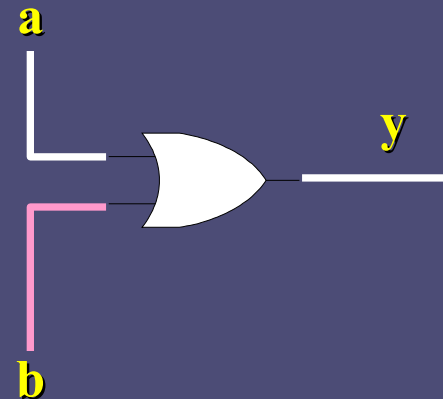
- Η OR μάσκα περιέχει:
 - 1 στα bits που θα γίνουν 1
 - 0 στα bits που θα παραμείνουν ως έχουν

$$0 \text{ OR } X = X$$

$$1 \text{ OR } X = 1$$

Συγκέντρωση σημάτων με OR

- Εισαγωγή
- Λογικές πράξεις



$$0 \text{ OR } X = X$$
$$1 \text{ OR } X = 1$$

- Θα πρέπει ανά πάσα στιγμή όλα τα σήματα πλην ενός να είναι 0!

Δυαδική λογική: XOR

- Εισαγωγή
- Λογικές πράξεις

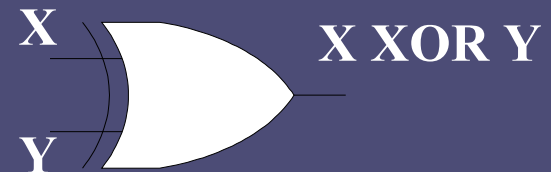
- Αποκλειστικό Ή (XOR)

- αποτέλεσμα 1, όταν **μόνο το X** ή **μόνο το Y** είναι 1
- $1 \text{ XOR } X = X \text{ XOR } 1 = \text{NOT } X$
- $0 \text{ XOR } X = X \text{ XOR } 0 = X$

Πίνακας Αλήθειας

X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Σύμβολο πύλης XOR



Υλοποίηση πύλης XOR: χρησιμοποιώντας συνδυασμούς άλλων πυλών
 $X \text{ xor } Y = XY' + X'Y$

Μάσκα XOR: για να αντιστρέψουμε bits

- Εισαγωγή
- Λογικές πράξεις

- Ζητούμενο: σε λέξη των 8 bits να αντιστραφούν τα bits 3,6 και 7.

Στόχος:	1 0 0 1 1 0 0 0	XOR
Μάσκα:	1 1 0 0 1 0 0 0	
Έξοδος:	0 1 0 1 0 0 0 0	

- Η XOR μάσκα περιέχει:
 - 1 στα bits που θα αντιστραφούν
 - 0 στα bits που θα παραμείνουν ως έχουν

Δυαδική λογική: XNOR

- Εισαγωγή
- Λογικές πράξεις

- Η συμπληρωματική συνάρτηση της XOR
 - το αποτέλεσμα είναι 1, όταν τα X και Y είναι όμοια
 - συνάρτηση “ισοδυναμίας”



Υλοποίηση πύλης XNOR:

χρησιμοποιώντας συνδυασμούς άλλων πυλών

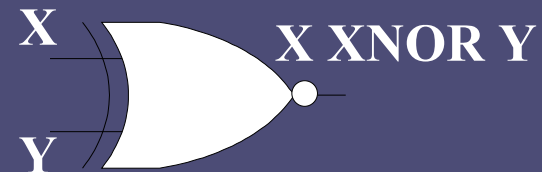
$X \text{ xnor } Y =$

$XY + X'Y'$

Πίνακας Αλήθειας

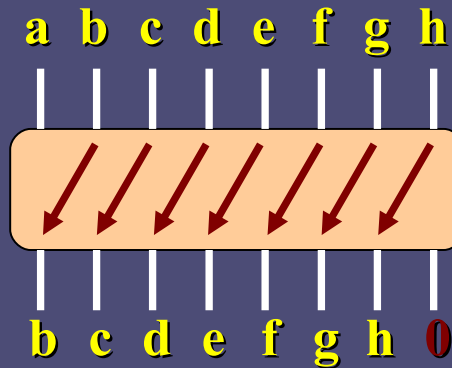
X	Y	XNOR
0	0	1
0	1	0
1	0	0
1	1	1

Σύμβολο πύλης XNOR

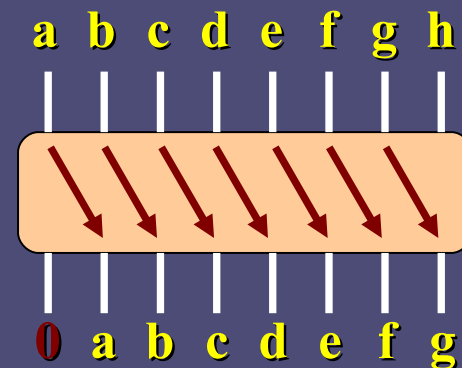


Ολίσθηση (Shift)

- Εισαγωγή
- Λογικές πράξεις
- Ολίσθηση



αριστερή ολίσθηση



δεξιά ολίσθηση

Το Byte ξανά: βάρη δυαδικών ψηφίων

- Δυαδικοί αριθμοί

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

ΤΟ ΠΕΡΙΣΣΟΤΕΡΟ
σημαντικό bit

ΤΟ ΛΙΓΟΤΕΡΟ
σημαντικό
bit

1	1	1	1	0	0	1	1								
1×128	1×64	1×32	1×16	0×8	0×4	1×2	1×1								
128	+	64	+	32	+	16	+	0	+	0	+	2	+	1	=
243 (δεκαδικό)															

- Μετατροπή από το δυαδικό στο δεκαδικό σύστημα

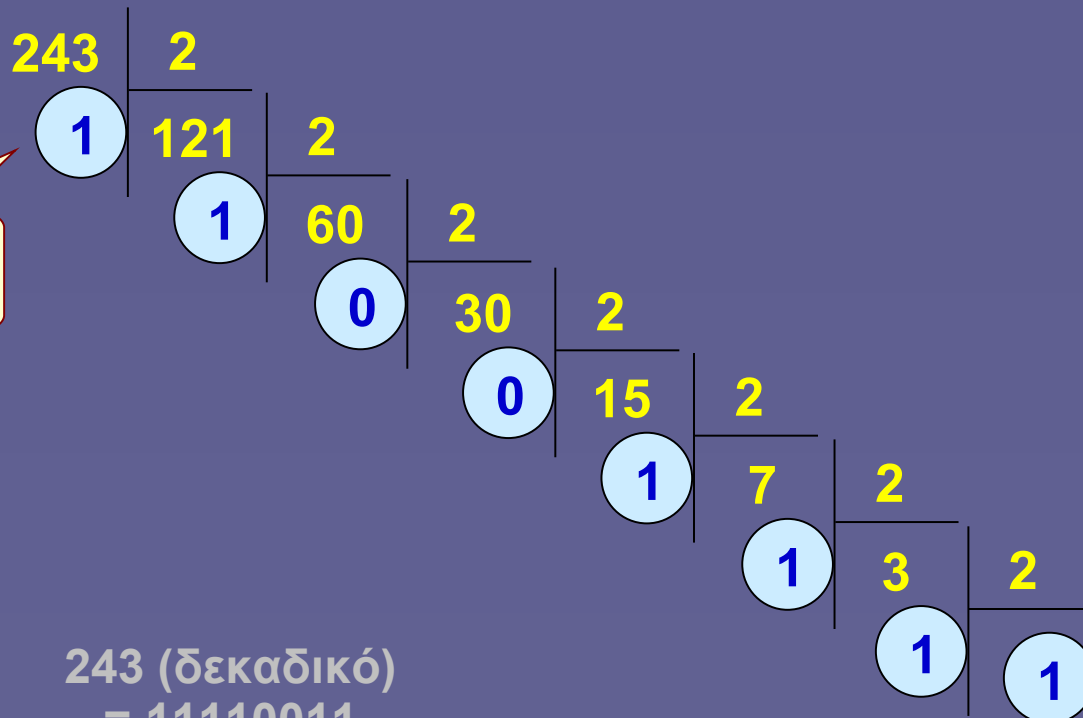
!

Εάν ο αριθμός διαθέτει περισσότερα bits, χρησιμοποιούμε μεγαλύτερες δυνάμεις του 2

Μετατροπή δεκαδικού σε δυαδικό

- Δυαδικοί αριθμοί

το λιγότερο σημαντικό bit



!

Προσοχή στη θέση του περισσότερο σημαντικού bit!

το περισσότερο σημαντικό bit

“Συντομογραφίες” δυαδικών αριθμών

- Δυαδικοί αριθμοί

ἰ

Το δεκαεξαδικό σύστημα (hex) διευκολύνει τη χρήση των δυαδικών αριθμών από τον άνθρωπο.

Παράδειγμα: στον ορισμό χρωμάτων για ιστοσελίδες.

- Δεκαεξαδικό σύστημα
 - Ομαδοποίηση ανά 4 bits

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Παράδειγμα στο δεκαεξαδικό σύστημα

- Δυαδικοί αριθμοί

▪ Παράδειγμα: 1100100110010100

1100 1001 1001 0100

C 9 9 4 = C994(hex)

▪ Παράδειγμα: 10000101011110

0010 0001 0101 1110

2 1 5 E = 215E (hex)

- Συμπλήρωση με 0 στα αριστερά
- Δεν αλλάζει τον αριθμό, όπως ακριβώς και στο δεκαδικό σύστημα

“Συντομογραφίες” δυαδικών αριθμών

- Δυαδικοί αριθμοί

i

Το οκταδικό σύστημα (oct) διευκολύνει τη χρήση των δυαδικών αριθμών από τον άνθρωπο.

Παράδειγμα: στον ορισμό δικαιωμάτων χρήσης αρχείων σε συστήματα τύπου Unix.

- **Οκταδικό σύστημα**
 - Ομαδοποίηση ανά 3 bits

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Στο επόμενο μάθημα...

- Δυαδικοί αριθμοί
- Στο επόμενο...

- **Αριθμητικές πράξεις**
 - Φυσικών αριθμών (χωρίς πρόσημο)
 - Ακεραίων (με πρόσημο)
 - Κλασματικών (κινητής υποδιαστολής)

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2007-08

Πράξεις με δυαδικούς αριθμούς

(αριθμητικές πράξεις)

<http://www.ionio.gr/~mistral/tp/csintro/>

Μ.Στεφανιδάκης

Πράξεις με δυαδικούς αριθμούς

- Δυαδικοί Αριθμοί

- Ο υπολογιστής μπορεί να εκτελέσει
 - Λογικές πράξεις
 - Αριθμητικές πράξεις
- Οι πράξεις εκτελούνται
 - Σε ομάδες bits (bytes ή πολλαπλάσιά τους)

Το Byte ως δυαδικός αριθμός

- Δυαδικοί αριθμοί

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

ΤΟ ΠΕΡΙΣΣΟΤΕΡΟ
σημαντικό bit

ΤΟ ΛΙΓΟΤΕΡΟ
σημαντικό
bit

1	1	1	1	0	0	1	1
---	---	---	---	---	---	---	---

1x128 1x64 1x32 1x16 0x8 0x4 1x2 1x1

128 + 64 + 32 + 16 + 0 + 0 + 2 + 1 =

243 (δεκαδικό)

- Μετατροπή από το δυαδικό στο δεκαδικό σύστημα

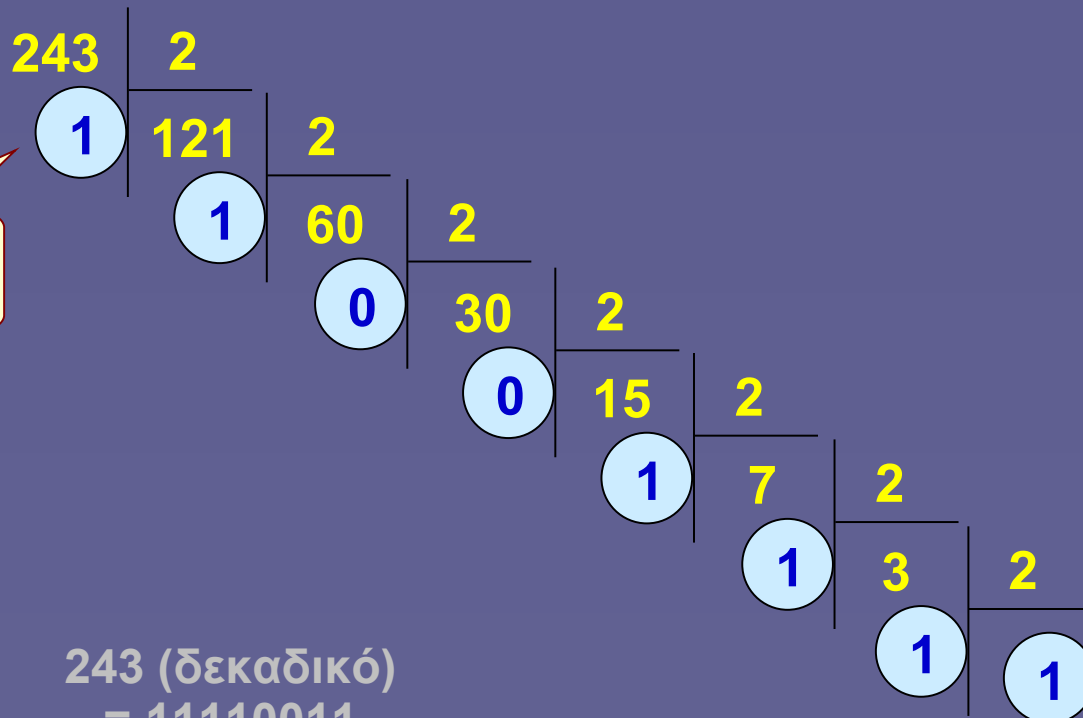
!

Εάν ο αριθμός διαθέτει περισσότερα bits, χρησιμοποιούμε μεγαλύτερες δυνάμεις του 2

Μετατροπή δεκαδικού σε δυαδικό

- Δυαδικοί αριθμοί

το λιγότερο σημαντικό bit



243 (δεκαδικό)
= 11110011

το περισσότερο σημαντικό bit

!

Προσοχή στη θέση του περισσότερο σημαντικού bit!

“Συντομογραφίες” δυαδικών αριθμών

- Δυαδικοί αριθμοί

- Δεκαεξαδικό σύστημα
 - Ομαδοποίηση ανά 4 bits

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Παράδειγμα στο δεκαεξαδικό σύστημα

- Δυαδικοί αριθμοί

▪ Παράδειγμα: 1100100110010100

1100 1001 1001 0100

C 9 9 4 = C994(hex)

▪ Παράδειγμα: 10000101011110

0010 0001 0101 1110

2 1 5 E = 215E (hex)

- Συμπλήρωση με 0 στα αριστερά
- Δεν αλλάζει τον αριθμό, όπως ακριβώς και στο δεκαδικό σύστημα

Φυσικοί αριθμοί (χωρίς πρόσημο)

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί

- Άμεση αντιστοιχία

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
.....	...

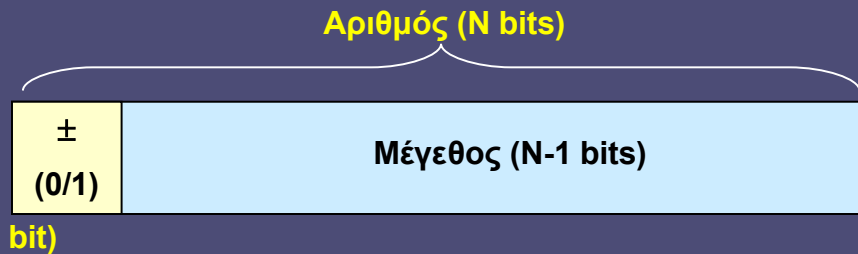
- Με n bits περιγράφονται

- Οι φυσικοί αριθμοί από 0 έως και $2^n - 1$

Ακέραιοι αριθμοί (με πρόσημο)

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί
- **Ακέραιοι**

- Πώς θα αναπαρασταθούν οι **αρνητικοί**;
 - Και θα γίνονται εύκολα οι πράξεις!
- Όχι καλή ιδέα:
 - Ξεχωριστό bit πρόσημου



- Διάστημα τιμών για αριθμούς με n bits
 $-(2^{n-1}-1)$ έως $+(2^{n-1}-1)$ (για $n=8$, $-127 \dots +127$)
 - ένα χρήσιμο bit λιγότερο
 - δυσκολία στις πράξεις
 - 2 αναπαραστάσεις του 0;

Ακέραιοι αριθμοί

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί
- **Ακέραιοι**

- Επίσης όχι καλή ιδέα:
 - **Συμπλήρωμα ως προς 1**
 - αντιστροφή όλων των bits του αριθμού
 - Πιο σημαντικό bit: 0 για θετικούς, 1 για αρνητικούς
 - Διάστημα τιμών για αριθμούς με n bits
 $-(2^{n-1}-1)$ έως $+(2^{n-1}-1)$ (γιατί;)
 - Τα ίδια προβλήματα με την χρήση ξεχωριστού bit πρόσημου!
- Καλή ιδέα!
 - **Συμπλήρωμα ως προς 2**
 - Πώς υπολογίζεται;

Συμπλήρωμα ως προς 2

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί
- **Ακέραιοι**

- Ισούνται με το συμπλήρωμα ως προς $1 + 1$
 - εμπειρικός κανόνας
 - “αντιστροφή όλων των bits εκτός από τα δεξιότερα συνεχόμενα 0 και το πρώτο 1 αριστερά από αυτά”
 - Δεν ισχύει για το 0 (και το 10000....0)
- Συμπλήρωμα ως προς 2: παραδείγματα
- 001011100 \Rightarrow 110100**100**
- 011111111 \Rightarrow 10000000**1**
- Αλλά:
- 000000000 \Rightarrow 000000000

Ακέραιοι με συμπλήρωμα ως προς 2

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί
- Ακέραιοι

- Διάστημα τιμών για αριθμούς με n bits
 $-(2^{n-1})$ έως $+(2^{n-1}-1)$ (για $n=8$, $-128 \dots +127$)
 - Μόνο το $+(2^{n-1})$ δεν μπορεί να αναπαρασταθεί
- Ευκολία στις πράξεις
 - αφαίρεση = πρόσθεση του συμπληρώματος ως προς 2
 - Μία και μοναδική αναπαράσταση του 0
- Πιο σημαντικό bit: 0 για θετικούς, 1 για αρνητικούς
 - Δεν είναι όμως bit προσήμου!!!

Κλασματικοί αριθμοί

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί
- Ακέραιοι
- Κλασματικοί

- Θεωρητικά:
 - Θα μπορούσαμε να επεξεργαζόμαστε ξεχωριστά το ακέραιο και το κλασματικό μέρος
- Αλλά:
 - Δυσκολία στις πράξεις – απώλεια ακρίβειας κατά τις διαιρέσεις
 - Αδυναμία αναπαράστασης πολύ μεγάλων και πολύ μικρών αριθμών
- Η λύση:
 - Αριθμοί κινητής υποδιαστολής (**floating point**)
 - Εύκολη αναπαράσταση τόσο του **1.000.000.000.000** όσο και του **0,000000000000000001**

Αριθμοί κινητής υποδιαστολής

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί
- Ακέραιοι
- Κλασματικοί

- 3 μέρη
 - Πρόσημο (Π) (1 bit)
 - $0 = +$ $1 = -$
 - Εκθέτης (E) (8 ή 11 bits)
 - Η βάση είναι το 2 (εννοείται)
 - Θετικοί και αρνητικοί εκθέτες με πλεόνασμα 127 ή 1023 (π.χ. αντί -55, $E = -55 + 127 = 72!$)
 - Σημαινόμενο τμήμα (Σ) (23 ή 52 bits)
 - Κανονικοποίηση: μορφή $1,xxxxxxxxxxxxx\dots$
 - Το '1,' εννοείται και δεν αποθηκεύεται
- Τελικός αριθμός: $-1^{\Pi} \times 1.\Sigma \times 2^{E-127}$ (ή $E-1023$)
 - Ειδικοί αριθμοί: 0, ∞ , NaN (Not a Number)

Αριθμητικές πράξεις

- Αριθμητικές πράξεις

- Οι βασικές πράξεις

- Πρόσθεση
- Αφαίρεση

- Άλλες πράξεις

- Πολλαπλασιασμός
- Διαίρεση
- Επίσης:
 - Τετραγωνική ρίζα, τριγωνομετρικές συναρτήσεις, εκθετικά, λογάριθμοι κλπ..
 - Υλοποίηση σε υλικό με διάφορες τεχνικές
 - Π.χ με πολυώνυμα

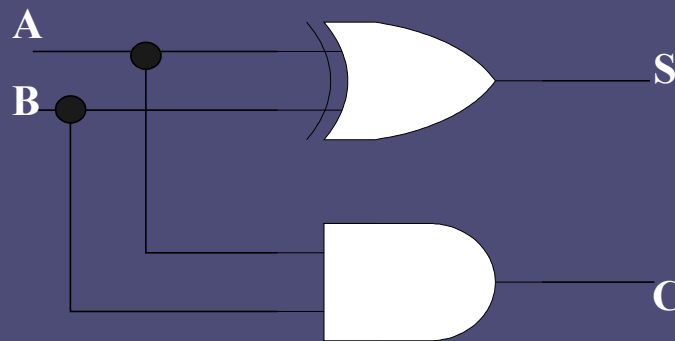
Προσθέτοντας 2 bits

- Αριθμητικές πράξεις

bits	άθροισμα	κρατούμενο
0 + 0	0	0
0 + 1	1	0
1 + 0	1	0
1 + 1	0	1

Ημιαθροιστής (half-adder)

- Αριθμητικές πράξεις



A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

;

Αν απαιτείται πρόσθεση αριθμών με περισσότερα bits;

Προσθέτοντας δυαδικούς αριθμούς (μη προσημασμένους)

Κρατούμενο


A' Αριθμός (119)	0	1	1	1	0	1	1	1
B' Αριθμός (88)	0	1	0	1	1	0	0	0
Άθροισμα (207)	1	1	0	0	1	1	1	1

1. Αριθμοί με ίδιο μήκος (ίσος αριθμός bits)
2. Αρχίζοντας από το λιγότερο σημαντικό bit (το δεξιότερο)
3. Προσθέτουμε ζεύγη bits και μεταφέρουμε το κρατούμενο (αν υπάρχει) προς τα αριστερά
 - Το προσθέτουμε στο επόμενο ζεύγος bits

Προσθέτοντας δυαδικούς αριθμούς (μη προσημασμένους)

- Υπερχείλιση
 - Στον υπολογιστή το πλήθος των bits ανά αριθμό είναι προκαθορισμένο
 - Το αποτέλεσμα της πρόσθεσης θα πρέπει να χωρά στα διαθέσιμα bits
 - Μη προσημασμένοι αριθμοί:
 - αριθμός με N bits \Rightarrow πεδίο τιμών $[0 \dots 2^N - 1]$
 - π.χ. για αριθμούς με 8 bits, από 0 έως 255

Κρατούμενο		1	1	1	1	1	1			
A' Αριθμός (180)		1	0	1	1	0	1	0	0	
B' Αριθμός (78)		0	1	0	0	1	1	1	0	
Άθροισμα (258)		1	0	0	0	0	0	0	1	0

υπερχείλιση  διαθέσιμος χώρος

Προσθέτοντας δυαδικούς αριθμούς (προσημασμένους)

- Προσημασμένοι ακέραιοι
 - Συμπλήρωμα ως προς 2
 - Το περισσότερο σημαντικό bit **υποδηλώνει** το πρόσημο
 - 0=θετικός, 1=αρνητικός
 - **αριθμός με N bits \Rightarrow πεδίο τιμών $[-2^{N-1} \dots 0 \dots +2^{N-1} - 1]$**
 - π.χ. για αριθμούς με 8 bits, από -128 έως +127
- Πρόσθεση
 - Όπως σε μη προσημασμένους
 - Τελικό κρατούμενο αγνοείται
 - Έλεγχος υπερχείλισης;
 - Αφαίρεση = πρόσθεση του συμπληρώματος ως προς 2 του αφαιρετέου
 - $A - B = A + (-B)$

Προσθέτοντας δυαδικούς αριθμούς (προσημασμένους)

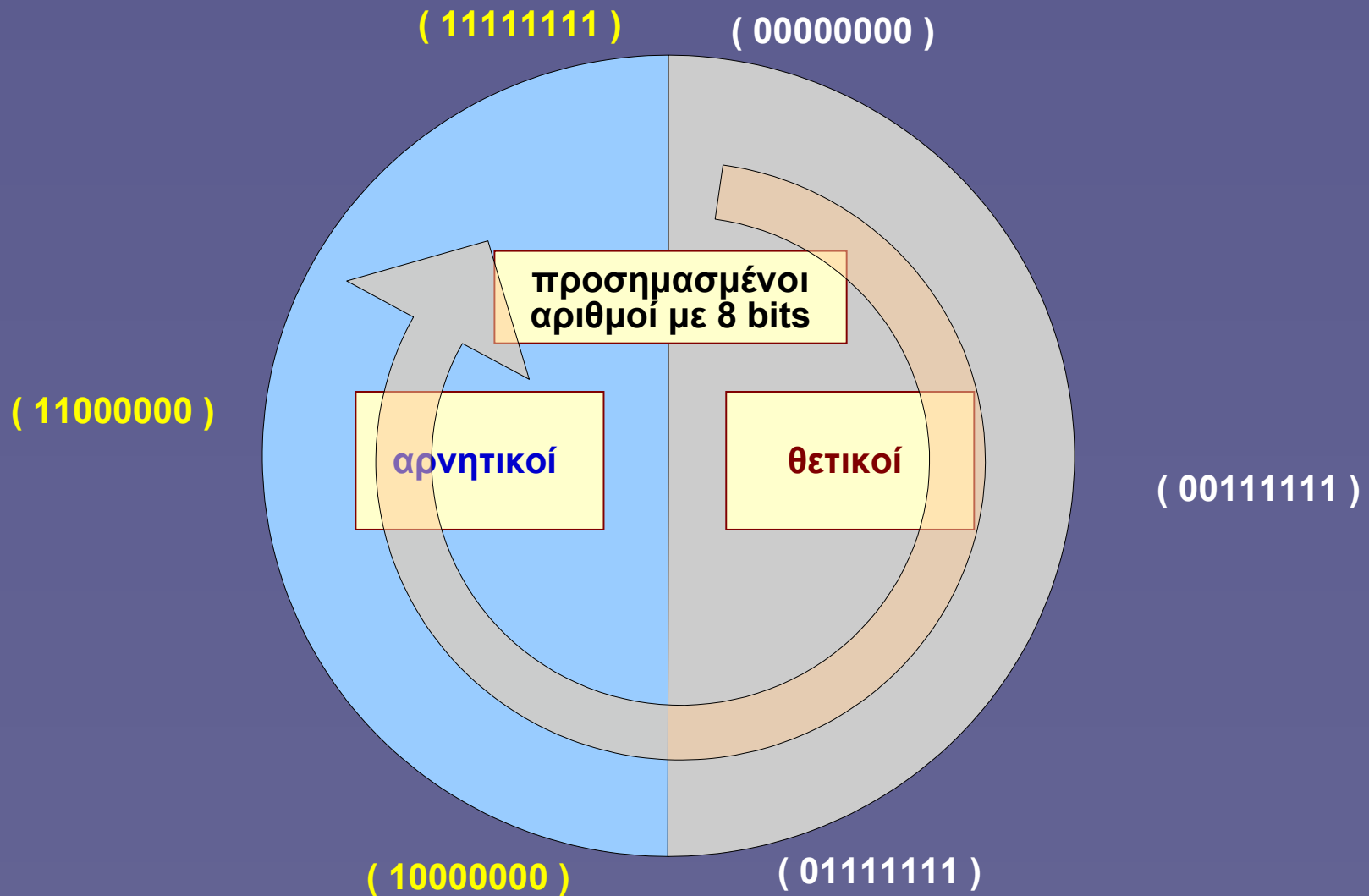
Κρατούμενο								
A' Αριθμός (+17)	0	0	0	1	0	0	0	1
B' Αριθμός (+22)	0	0	0	1	0	1	1	0
Άθροισμα (+39)	0	0	1	0	0	1	1	1

Προσθέτοντας δυαδικούς αριθμούς (προσημασμένους)

Κρατούμενο								
A' Αριθμός (+24)	0	0	0	1	1	0	0	0
B' Αριθμός (-17)	1	1	1	0	1	1	1	1
Άθροισμα (+7)	0	0	0	0	0	1	1	1

- το κρατούμενο αγνοείται

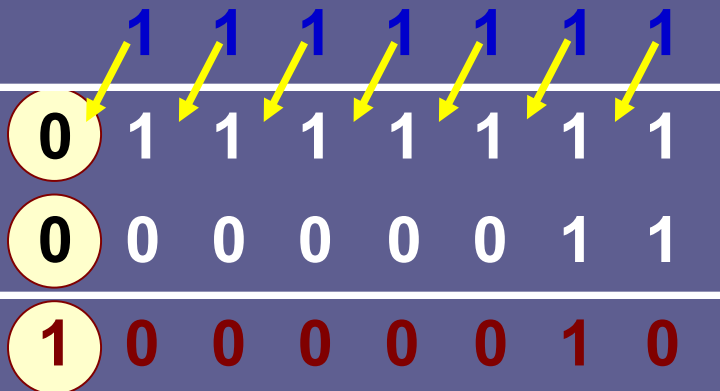
Υπερχείλιση σε προσημασμένους αριθμούς



Υπερχείλιση σε προσημασμένους αριθμούς

Κρατούμενο

A' Αριθμός (+127)	0	1	1	1	1	1	1	1
B' Αριθμός (+3)	0	0	0	0	0	0	1	1
Άθροισμα (-126;)	1	0	0	0	0	0	1	0



- Το άθροισμα αριθμών με ίδιο πρόσημο θα πρέπει να έχει επίσης το ίδιο πρόσημο!
 - στην αντίθετη περίπτωση: **υπερχείλιση**

Υπερχείλιση σε προσημασμένους αριθμούς

Κρατούμενο								
A' Αριθμός (-126)	1	0	0	0	0	0	1	0
B' Αριθμός (-5)	1	1	1	1	1	0	1	0
Άθροισμα (+124;)	0	1	1	1	1	1	0	0

- Το άθροισμα αριθμών με ίδιο πρόσημο θα πρέπει να έχει επίσης το ίδιο πρόσημο!
 - στην αντίθετη περίπτωση: **υπερχείλιση**

Πράξεις με αριθμούς κινητής υποδιαστολής

- Αριθμητικές πράξεις

- Σύνθετη διαδικασία
- Η γενική μορφή της πρόσθεσης:
 1. Σύγκριση προσήμων
 - αν είναι ίδια \Rightarrow πρόσθεση
 - αλλιώς \Rightarrow αφαίρεση
 2. Εξίσωση εκθετών
 - μετακίνηση υποδιαστολής
 3. Πρόσθεση ή αφαίρεση σημαινόμενων τμημάτων
 - ακέραιο και κλασματικό μέρος
 4. Κανονικοποίηση αποτελέσματος
 5. Έλεγχος για υπερχείλιση

Πράξεις με αριθμούς κινητής υποδιαστολής

A' αριθμός: 0 **10000100** **1011**000000000000000000000000

+ $2^{132-127} \times 1,1011$ ($+2^5 \times 1,1011$)

B' αριθμός: 0 **10000010** **0110**000000000000000000000000

+ $2^{130-127} \times 1,011$ ($+2^3 \times 1,011$)

A	$+2^5$	x	1,10110
+ B	$+2^5$	x	0,01011
<hr/>			
=	$+2^5$	x	10,00001
κανονικοποίηση	$+2^6$	x	1,000001

αποτέλεσμα: 0 **10000101** **000001**000000000000000000000000

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2007-08

Αναπαράσταση Μη Αριθμητικών Δεδομένων

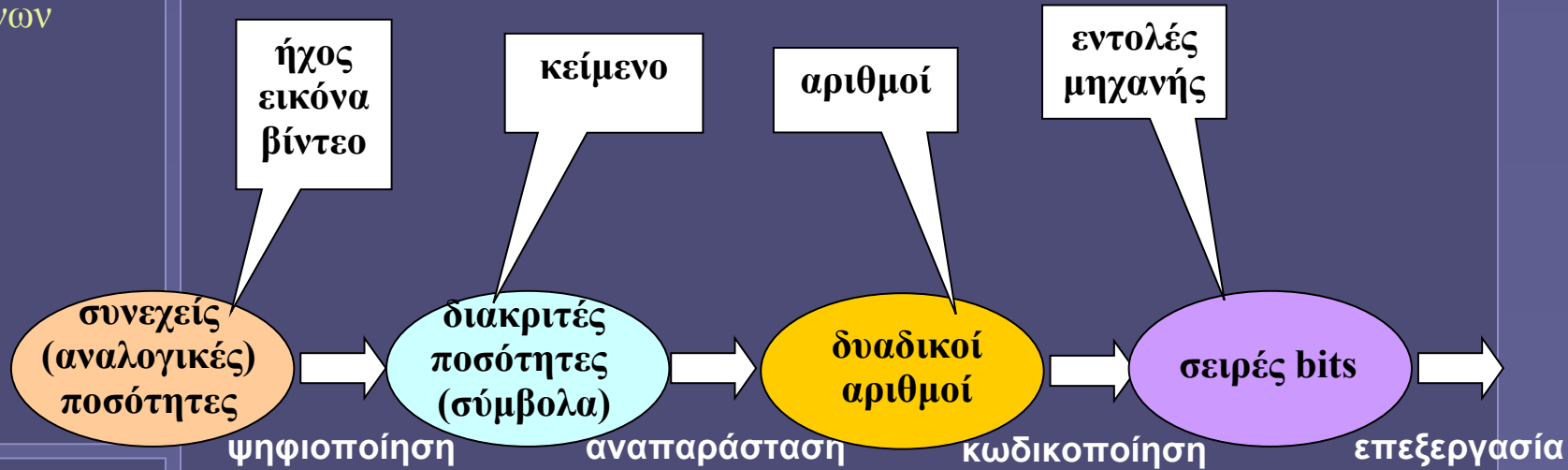
(κείμενο, ήχος και εικόνα στον υπολογιστή)

<http://www.ionio.gr/~mistral/tp/csintro/>

Μ.Στεφανιδάκης

Αναπαράσταση δεδομένων

- Αναπαράσταση δεδομένων



Δεδομένα:
ανεξάρτητα από
τύπο και
προέλευση, στον
υπολογιστή
υπάρχουν σε μία
μορφή: 0 και 1

- Ψηφιοποίηση
 - Διαδικασία μετατροπής συνεχών τιμών σε διακριτά σύμβολα
- Αναπαράσταση
 - Διαδικασία αντιστοίχισης συμβόλων σε δυαδικούς αριθμούς
- Κωδικοποίηση
 - Αποθήκευση δυαδικών αριθμών σε σειρές bits

Η ερμηνεία της αναπαράστασης

- Αναπαράσταση δεδομένων



Στα ερωτήματα αυτά μπορεί να απαντήσει μόνο ο προγραμματιστής της εφαρμογής που χειρίζεται τα δεδομένα!

- **Κάπου στη μνήμη του υπολογιστή...**
 - Βρίσκεται αποθηκευμένη η σειρά bits **0100110111010001**
- **Πόσα σύμβολα αναπαριστά;**
 - Πόσα bits ανά σύμβολο;
- **Ποιος ο τύπος των δεδομένων;**
- **Ποια συγκεκριμένη ποσότητα συμβολίζει;**
- **Πώς θα το χειριστεί ο υπολογιστής;**

Αναπαράσταση με δυαδικούς αριθμούς

- Αναπαράσταση δεδομένων

- **Σειρά n bits**

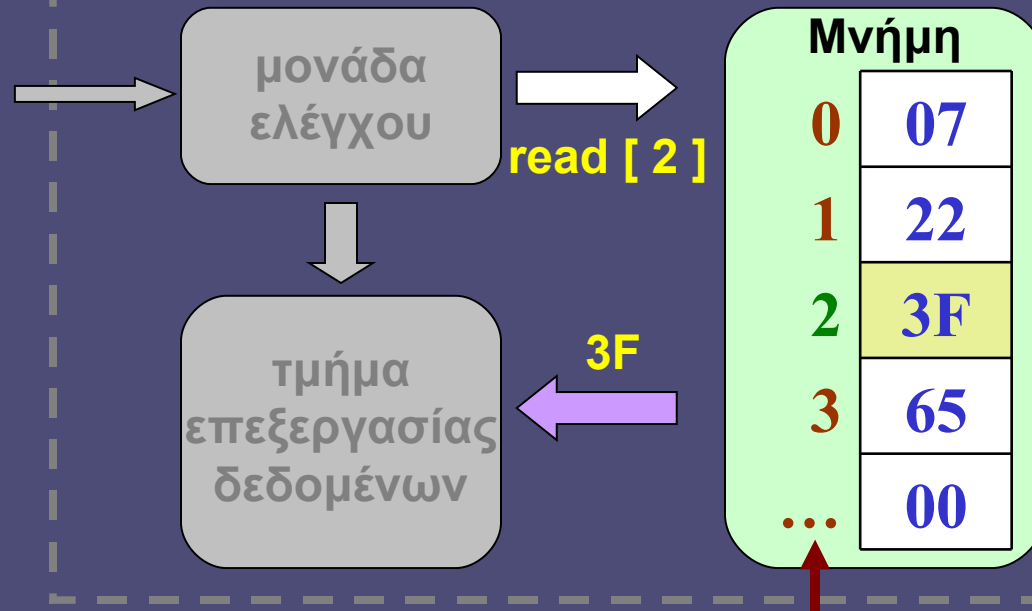
- Δυαδικός αριθμός με **n bits** ($n \geq 1$) μπορεί να αναπαραστήσει 2^n διαφορετικά σύμβολα

- **Μη αριθμητικά δεδομένα**

- Κείμενο, εντολές μηχανής, ήχος, εικόνα...
 - Σύνολο διαφορετικών αντικειμένων (**συμβόλων**)
- Αντιστοίχιση κάθε συμβόλου σε μοναδικό δυαδικό αριθμό (code point)
 - “Αναπαράσταση”
 - Η ακριβής αντιστοίχιση συνήθως ορίζεται σε ένα **πρότυπο** (standard)

Η αποθήκευση στη μνήμη

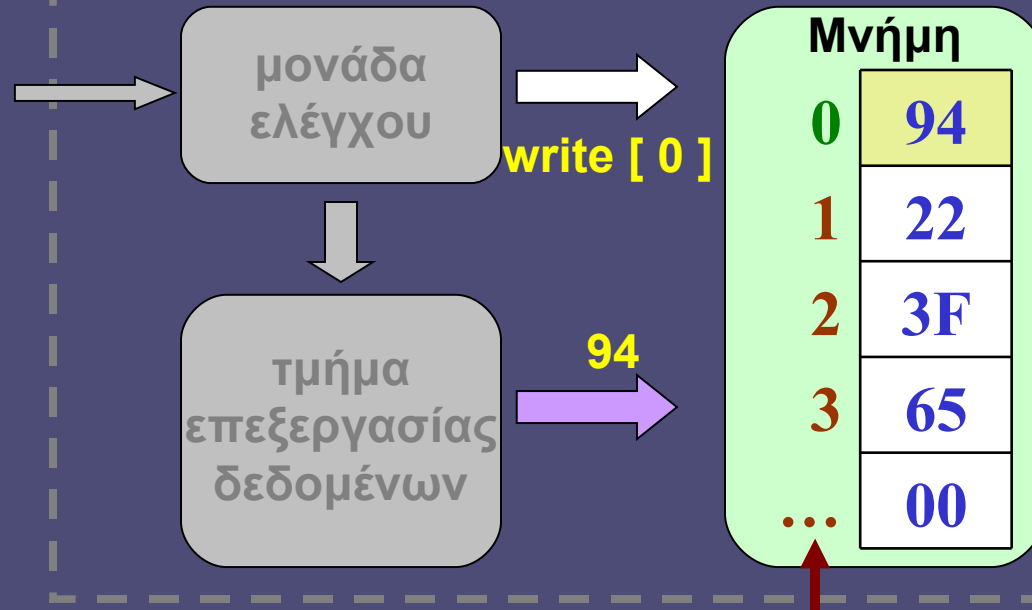
- Αναπαράσταση δεδομένων



- Μοντέλο μνήμης
 - Συστοιχία αποθηκευτικών θέσεων
 - Σε κάθε θέση αποθηκεύεται (συνήθως) 1 byte
 - Κάθε θέση διαθέτει μοναδική διεύθυνση
 - Επιλογή θέσης κατά την προσπέλαση (ανάγνωση-εγγραφή)

Η αποθήκευση στη μνήμη

- Αναπαράσταση δεδομένων



- Χωρητικότητα μνήμης

- Εκφράζεται σε πολλαπλάσια του byte
- 1 KByte (KB) = 1024 Bytes (2^{10})
- 1 MByte (MB) = 1024 KBytes (2^{10})
 - κλπ

Διευθύνσεις μνήμης

;

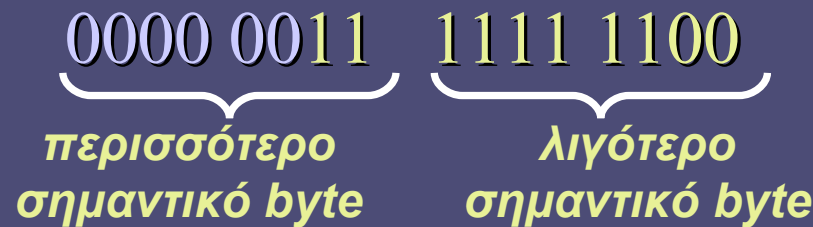
Με διεύθυνση των n bits, πόσες διαφορετικές θέσεις μνήμης μπορούμε να προσπελάσουμε;

Θέματα αποθήκευσης δυαδικών αριθμών

- Αναπαράσταση δεδομένων

- Όταν
 - Ένας δυαδικός αριθμός χρειάζεται περισσότερα από ένα byte για να αποθηκεύσει τα ψηφία του
- Παράδειγμα: 3FC (hex) = 11 1111 1100

Σε 2 bytes!



- Προφανώς σε συνεχόμενες θέσεις μνήμης
Αλλά: ποιο byte αποθηκεύεται πρώτο;

;

Πώς σχετίζεται η σειρά αποθήκευσης των bytes με τα “Ταξίδια του Γκιούλιβερ”;

Θέματα αποθήκευσης δυαδικών αριθμών

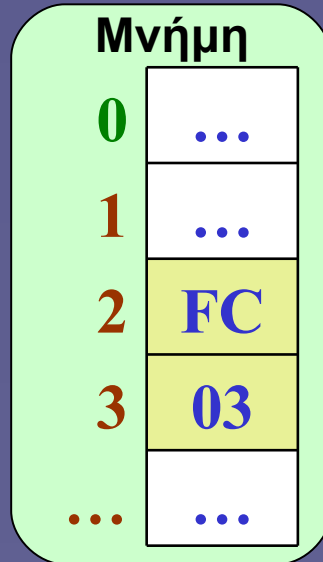
- Αναπαράσταση δεδομένων

αποθηκεύοντας το
03FC

00000011 11111100

i

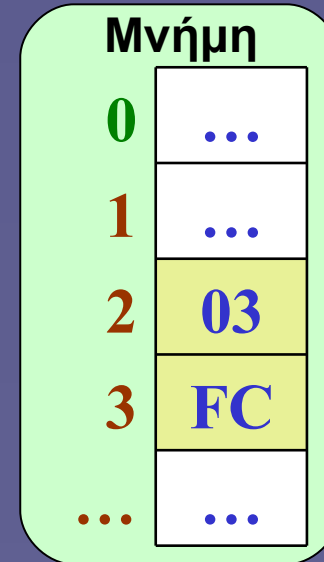
Στο Διαδίκτυο τα
δεδομένα
διακινούνται σε
big-endian μορφή



“little-endian”

(Intel)

*Το λιγότερο
σημαντικό byte στη
θέση μνήμης με
μικρότερη διεύθυνση*



“big-endian”

*(Motorola, Sun,
PowerPC)*

*Το περισσότερο
σημαντικό byte στη
θέση μνήμης με
μικρότερη διεύθυνση*

Αρχικές αναπαραστάσεις κειμένου

- Αναπαράσταση δεδομένων
- Κείμενο

- **Οι πρώτες αναπαραστάσεις κειμένου**
 - Στον υπολογιστή
 - 6-7 bits ανά χαρακτήρα
 - Πόσοι διαφορετικοί χαρακτήρες;
- **Μη εκτυπώσιμοι χαρακτήρες**
 - Χαρακτήρες ελέγχου
 - Ιδιαίτερα χρήσιμοι για τις συσκευές εξόδου της εποχής (εκτυπωτές, τηλέτυπα...)
 - Νέα γραμμή (LINE FEED – LF)
 - Επιστροφή κεφαλής εκτύπωσης (CARRIAGE RETURN – CR)
 - Καμπανάκι (BELL) κλπ

Κώδικας ASCII

- Αναπαράσταση δεδομένων
- Κείμενο

- Βασικό αρχικό πρότυπο αναπαράστασης κειμένου
 - 7 bits ανά χαρακτήρα

STANDARD ASCII ΚΩΔΙΚΑΣ

hex	char	hex	char	hex	char
20		40	@	60	`
21	!	41	A	61	a
22	"	42	B	62	b
23	#	43	C	63	c
24	\$	44	D	64	d
25	%	45	E	65	e
26	&	46	F	66	f
27	'	47	G	67	g
28	(48	H	68	h
29)	49	I	69	i
2A	*	4A	J	6A	j
2B	+	4B	K	6B	k
2C	,	4C	L	6C	l
2D	-	4D	M	6D	m
2E	.	4E	N	6E	n
2F	/	4F	O	6F	o

i

ASCII: American Standard Code for Information Exchange

Κείμενο σε κώδικα ASCII

- Αναπαράσταση δεδομένων
- Κείμενο

;

Με 7 bits ανά χαρακτήρα και χρήση bytes, 1 bit μένει αχρησιμοποίητο. Πόσοι επιπλέον χαρακτήρες με το bit αυτό;

- **7 bits ανά χαρακτήρα**
 - 128 χαρακτήρες
 - Αναπαράσταση με τους αριθμούς 0...127
- **Κανονικοί χαρακτήρες (εκτυπώσιμοι)**
 - 32...64, 91...96, 123...126 = σημεία στίξης κ.ά. (32 = SPACE!)
 - 65...90 = κεφαλαία λατινικά (A-Z)
 - 97...122 = πεζά λατινικά (a-z)
- **Χαρακτήρες ελέγχου (μη εκτυπώσιμοι)**
 - 0...31, 127 – επιζούν τα: 9 (TAB), 13/10 (CR/LF, σήμανση “νέας γραμμής”)

Κείμενο σε κώδικα ASCII

- Αναπαράσταση δεδομένων
- Κείμενο

- **Παράδειγμα**

H	a	v	e		a		n	i	c	e		d	a	y	!
72	97	118	101	32	97	32	110	105	99	101	32	100	97	121	33

- **Γλώσσες προγραμματισμού**

- Συμβολοσειρά (string)
- Σε γλώσσες όπως η C, το 0 (αριθμητικό) συμβολίζει το τέλος της συμβολοσειράς
- Ο υπολογιστής μπορεί να κάνει πράξεις (π.χ. σύγκριση) με τη συμβολοσειρά

!

Εφόσον η κωδικοποίηση είναι με 1 byte ανά χαρακτήρα, δεν τίθεται θέμα “little-” ή “big-endian”

Επεκτάσεις κώδικα ASCII

- Αναπαράσταση δεδομένων
- Κείμενο



Χρησιμοποιώντας τον ISO-8859-1 δεν είναι δυνατή η αναπαράσταση των ελληνικών!

- **Χρήση του 1 επιπλέον bit του byte**
 - 128 + 128 χαρακτήρες, αριθμοί 0...255
 - 0...127 αντιστοιχούν στον αρχικό ASCII
 - 127...255: επεκταμένα αλφάβητα
- **Επέκταση αλφαβήτων (πρότυπα)**
 - Χαρακτήρες που δεν υπάρχουν στον ASCII
 - Διαφορετικά ανά γλώσσα! Π.χ.:
 - ISO-8859-1: Δυτική Ευρώπη (Å, Ñ, Æ, ä, ø κλπ)
 - ISO-8859-7: Νέα Ελληνικά
 - ...και πολλά άλλα πρότυπα για τις υπόλοιπες γλώσσες
 - Επίσης: μη πρότυπες λύσεις
 - Για Windows, Mac ..

Κώδικας ISO-8859-7

- Αναπαράσταση δεδομένων
- Κείμενο

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	<i>unused</i>															
1x																
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8x	<i>unused</i>															
9x																
Ax	NBSP	·	´	£	€	Ɔp		§	¨	©	,	*	¬	SHY		—
Bx	°	±	²	³	´	ˆ	À	·	È	Ĥ	ı	*	Ï	½	Υ	Ω
Cx	İ	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
Dx	Π	Ρ		Σ	Τ	Υ	Φ	Χ	Ψ	Ω	İ	ÿ	ά	έ	ή	ί
Ex	ú	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
Fx	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ı	ü	ó	ú	ώ	

[Wikipedia]

Κείμενο σε κώδικα ISO-8859-7

- Αναπαράσταση δεδομένων
- Κείμενο

- **Παράδειγμα**

Γ	ε	ι	α		σ	ο	υ	!
195	229	233	225	32	243	239	245	33

- **Επέκταση κώδικα ASCII**

- 0...127 όπως στον ASCII
- 128...159 πρόσθετοι χαρακτήρες ελέγχου
- 160...255 ελληνικά και σχετικά σύμβολα

!

Οι αναπαραστάσεις αλφαβήτων με 1 byte ανά χαρακτήρα τείνουν να καταργηθούν!

Κώδικας Unicode

- Αναπαράσταση δεδομένων
- Κείμενο



Με περισσότερα από 1 bytes ανά χαρακτήρα τίθεται θέμα σειρά αποθήκευσης των bytes!

- Για την αναπαράσταση όλων των αλφαβήτων!
 - Έχουν οριστεί σχεδόν 100.000 χαρακτήρες
 - Καλύπτει ιδεογράμματα, φωνητικές αναπαραστάσεις κλπ
 - Θα μπορούσε να καλύψει πάνω από 1 εκ. χαρακτήρες!
 - Κάθε χαρακτήρας αναπαρίσταται με περισσότερα από ένα bytes
 - Συνήθεις κωδικοποιήσεις: UCS-2 και UTF-8

Ελληνικά και Unicode

- Αναπαράσταση δεδομένων
- Κείμενο

Greek and Coptic **03FF**

	037	038	039	03A	03B	03C	03D	03E	03F
0			ὀ 0390	Π 03A0	ὐ 03B0	π 03C0	β 03D0	ϗ 03E0	Ϡ 03F0
1			Α 0391	Ρ 03A1	α 03B1	ρ 03C1	θ 03D1	ϣ 03E1	ϙ 03F1
2			Β 0392		β 03B2	ς 03C2	Υ 03D2	Ϡ 03E2	Ϙ 03F2
3			Γ 0393	Σ 03A3	γ 03B3	σ 03C3	Υ 03D3	Ϡ 03E3	ϙ 03F3
4	΄ 0374	΄ 0384	Δ 0394	Τ 03A4	δ 03B4	τ 03C4	Ϛ 03D4	ϣ 03E4	Θ 03F4
5	΄ 0375	΄ 0385	Ε 0395	Υ 03A5	ε 03B5	υ 03C5	φ 03D5	ϣ 03E5	Ε 03F5
6		Α 0386	Ζ 0396	Φ 03A6	ζ 03B6	φ 03C6	ω 03D6	ϣ 03E6	ε 03F6

Κείμενο σε Unicode

- Αναπαράσταση δεδομένων
- Κείμενο

• Παράδειγμα

Γ	ε	ι	α		σ	ο	υ	!
915	949	953	945	32	963	959	965	33
0393	03B5	03B9	03B1	0032	03C3	03BF	03C5	0033

Κωδικοποίηση UCS-2 (big-endian)

03	93	03	B5	03	B9	03	B1	00	32	03	C3	03	BF	03	C5	00	33
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Κωδικοποίηση UCS-2 (little-endian)

93	03	B5	03	B9	03	B1	03	32	00	C3	03	BF	03	C5	03	33	00
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Unicode σε κωδικοποίηση UTF-8

- Αναπαράσταση δεδομένων
- Κείμενο



Η κωδικοποίηση UTF-8 τείνει να επικρατήσει σε όλα τα προγράμματα που χειρίζονται κείμενα Unicode!

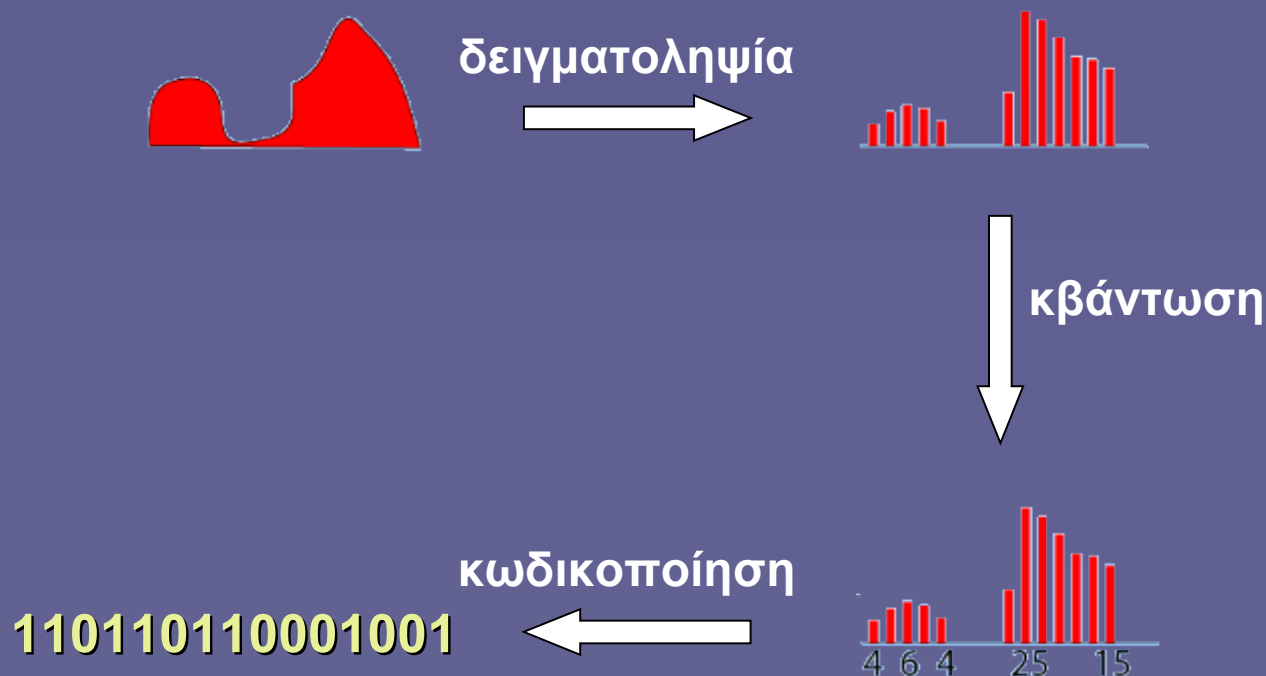
- Αναπαράσταση μεταβλητού μήκους

Unicode	Κωδικοποίηση UTF-8
00...7F	0xxxxxxx
80...7FF	110xxxxx 10xxxxxx
800...FFFF	1110xxxx 10xxxxxx 10xxxxxx
10000...10FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

- Το βασικό λατινικό αλφάβητο (ASCII) χρησιμοποιεί 1 byte ανά χαρακτήρα
 - Προς τα πίσω συμβατότητα
- Τα ελληνικά, 2 bytes
 - Ποια η κωδικοποίηση κατά UTF-8 του τελευταίου παραδείγματος;

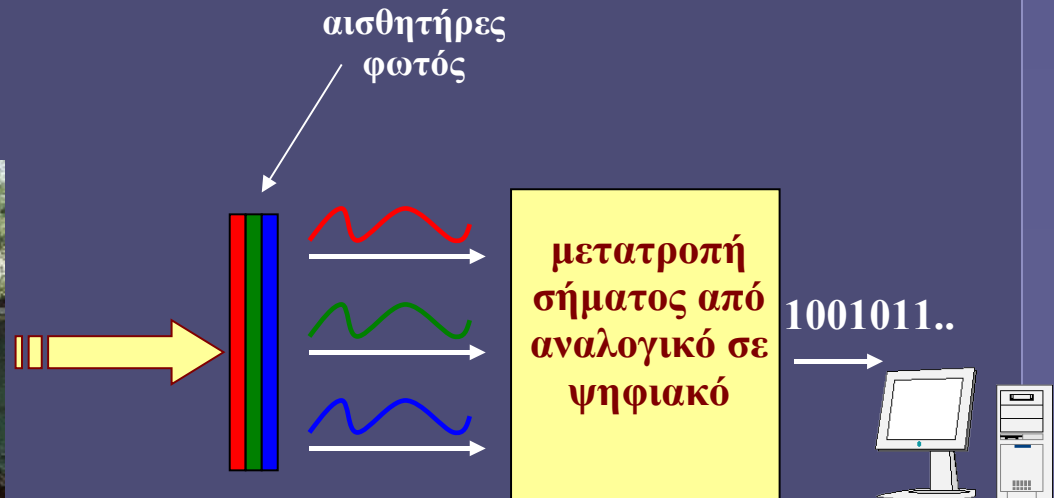
Ήχος: Ψηφιοποίηση και Αποθήκευση

- Αναπαράσταση δεδομένων
- Κείμενο
- Ήχος



Εικόνα: από τον αναλογικό στον ψηφιακό κόσμο

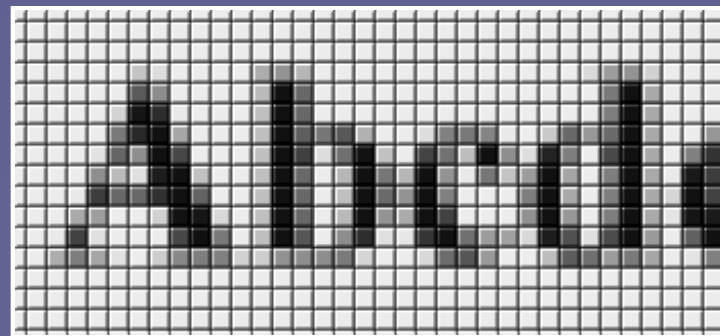
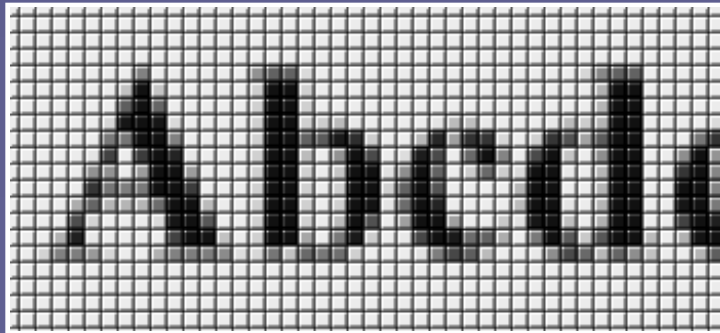
- Αναπαράσταση δεδομένων
- Κείμενο
- Ήχος
- Εικόνα



- Φωτοευαίσθητα κύτταρα
 - για τρία χρώματα (κόκκινο-πράσινο-μπλε)
- Μετατροπή σήματος σε ψηφιακή πληροφορία

Από τον αναλογικό στον ψηφιακό κόσμο: ανάλυση εικόνας

- Αναπαράσταση δεδομένων
- Κείμενο
- Ήχος
- **Εικόνα**



- Μεταφορά στον υπολογιστή:
 - Χωρισμός σε pixels
 - (εικονοστοιχεία)
- Διαστάσεις εικόνας
 - $X \times Y$ pixels (πλάτος \times ύψος)
- Ανάλυση εικόνας
 - dpi (dots per inch)
 - pixels/ίντσα
 - *μεγαλύτερη ανάλυση = περισσότερη λεπτομέρεια*
 - αλλά και μέγεθος αρχείων

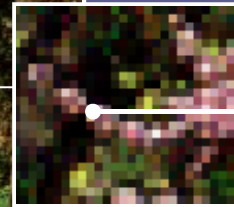
Από τον αναλογικό στον ψηφιακό κόσμο: βάθος χρώματος

- Αναπαράσταση δεδομένων
- Κείμενο
- Ήχος
- **Εικόνα**



- Στον φυσικό κόσμο
 - Συνεχές (αναλογικό) πεδίο τιμών
- Στον υπολογιστή:
 - Διακριτές (ψηφιακές) στάθμες
 - Βάθος Χρώματος
 - bits/pixel
 - **μεγαλύτερο βάθος = περισσότερες αποχρώσεις**
 - αλλά και μέγεθος αρχείων

Αναπαράσταση pixels: 16,7 εκ. χρώματα (true color)

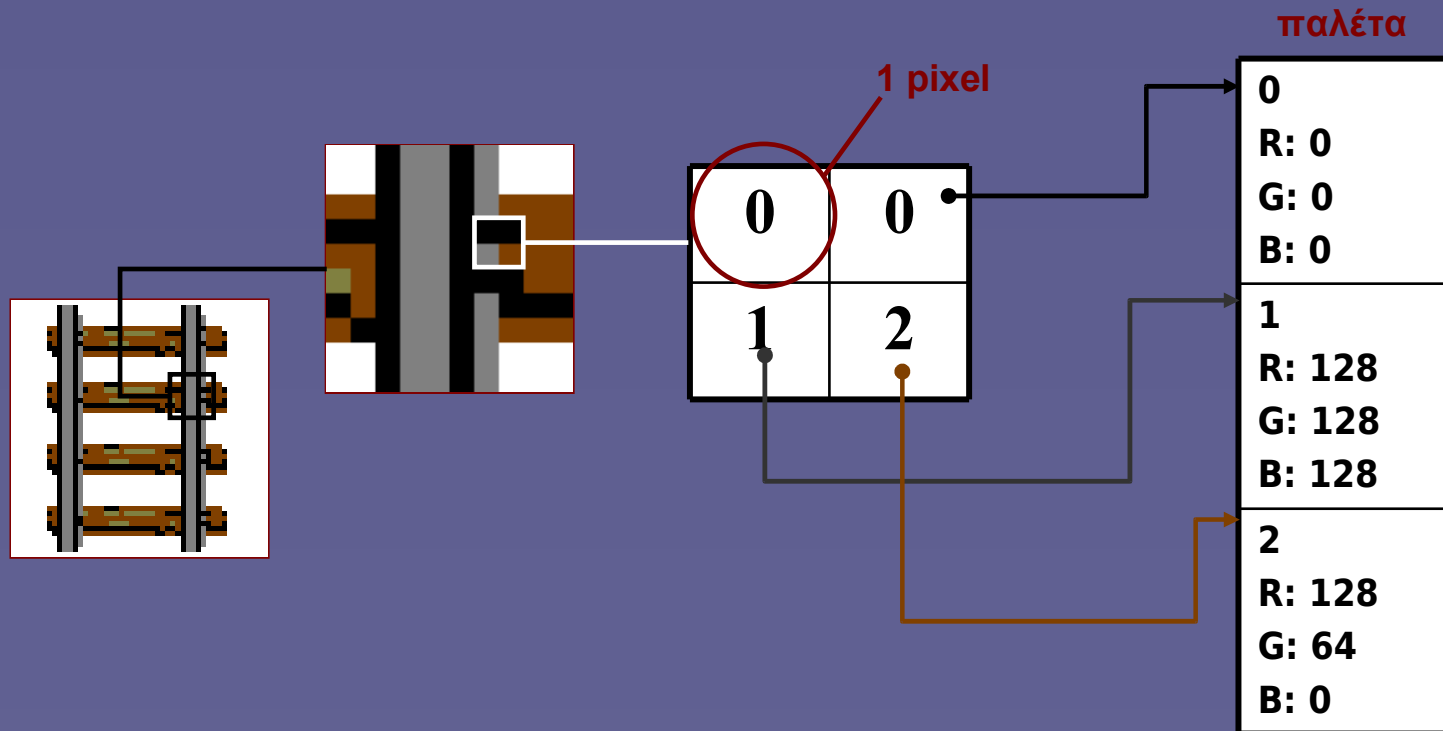


1 pixel

R:144	R:193	
G:128	G:164	...
B:118	B:179	
R:201		
G:174
B:134		
...

- 3 bytes/pixel (24bits): **R**(ed) **G**(reen) **B**(lue)
 - 256 στάθμες ανά συνιστώσα χρώματος
 - $256 \times 256 \times 256 = 16.777.216$ χρώματα
 - εικόνες με μεγαλύτερο βάθος χρώματος
 - 32 έως 48 bits

Αναπαράσταση pixels: Παλέτες



- Παλέτα: πίνακας αντιστοίχισης χρωμάτων
 - Μέγεθος παλέτας
 - καθορίζει bits/pixel
 - και μέγιστο αριθμό χρωμάτων

Διανυσματικά γραφικά

- Αναπαράσταση δεδομένων
- Κείμενο
- Ήχος
- **Εικόνα**

- **Περιγραφή σχημάτων**
 - Ως σύνολο ευθύγραμμων και καμπύλων τμημάτων
 - Με συντεταγμένες
 - Εύρεση σημείων μέσω μαθηματικού τύπου
- **Εύκολη αλλαγή μεγέθους γραφικών**
 - Χωρίς παραμόρφωση των σχημάτων

Αναπαράσταση βίντεο

- Αναπαράσταση δεδομένων
- Κείμενο
- Ήχος
- Εικόνα
- Βίντεο

- “Κινούμενη εικόνα” (καρέ)
 - όπως αναπαριστούμε τις απλές εικόνες
 - αλλά: με χρήση συμπίεσης
 - Για μείωση όγκου δεδομένων
 - Γειτονικά καρέ έχουν πολλές ομοιότητες

Κωδικοποίηση εντολών μηχανής

- Αναπαράσταση δεδομένων
- Κείμενο
- Ήχος
- Εικόνα
- Βίντεο
- Εντολές Μηχανής



Περιγράφει το είδος της πράξης που θα εκτελεστεί

Περιγράφουν την **προέλευση** των δεδομένων εισόδου (αριθμό καταχωρητή, διεύθυνση μνήμης κλπ) και τον **προορισμό** των δεδομένων εξόδου (αποτέλεσμα πράξης)

Το είδος της πράξης προσδιορίζει τον τύπο, την προέλευση και τον αριθμό των δεδομένων που συμμετέχουν στην πράξη !

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2007-08

Οργάνωση Υπολογιστών (I)

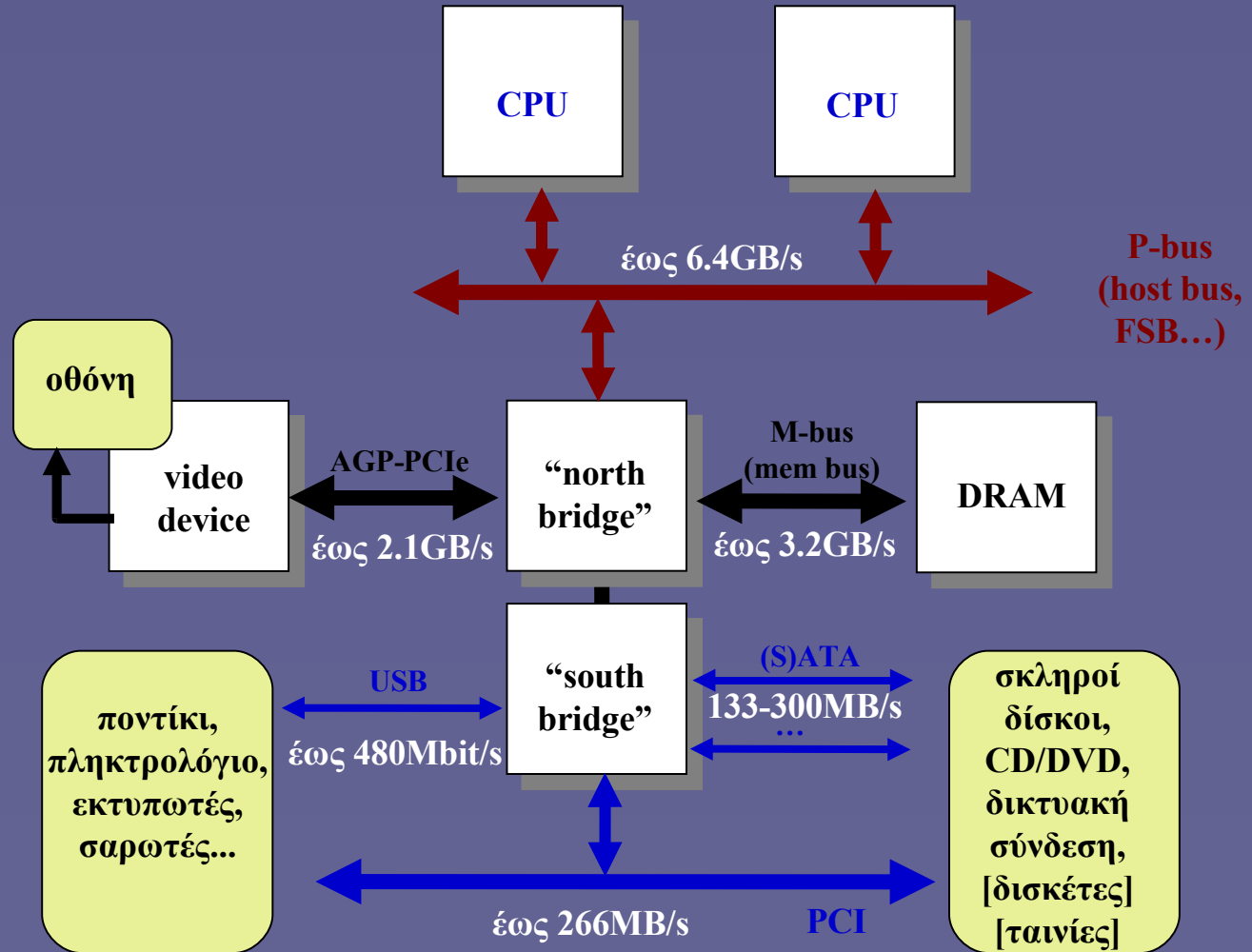
(η κεντρική μονάδα επεξεργασίας και η κύρια μνήμη)

<http://www.ionio.gr/~mistral/tp/csintro/>

Μ.Στεφανιδάκης

Ένα τυπικό υπολογιστικό σύστημα σήμερα

- Εισαγωγή

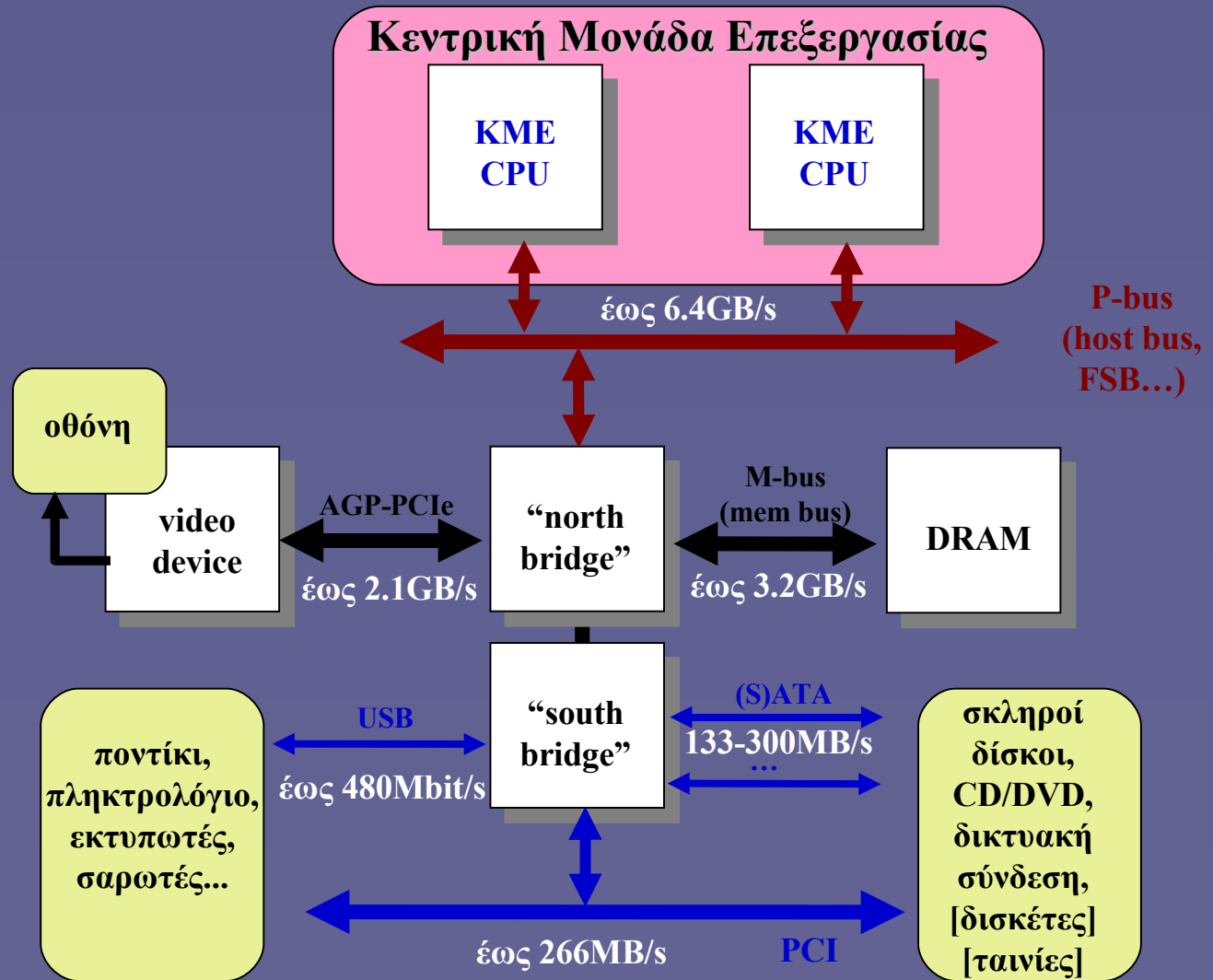


Τα βασικά υποσυστήματα

- Εισαγωγή



Τα σημερινά συστήματα συνήθως διαθέτουν περισσότερες από μία ΚΜΕ: είτε σε διαφορετικά μικροτσιπ είτε ως πολλαπλούς πυρήνες (cores) στο ίδιο μικροτσιπ!



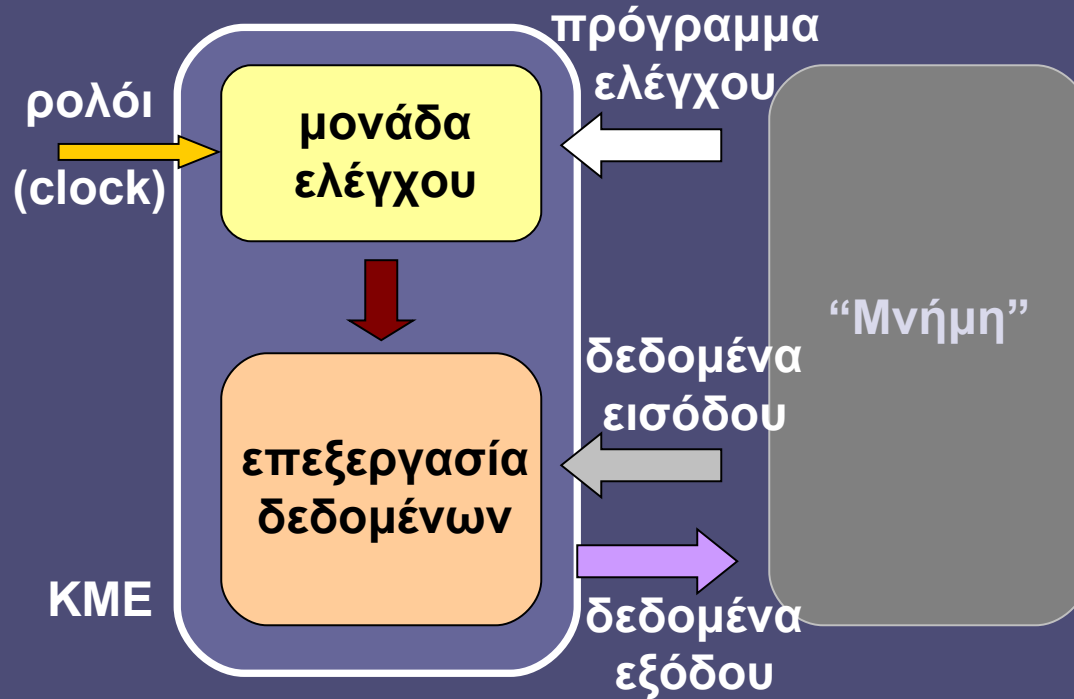
Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)

- Εισαγωγή
- ΚΜΕ

- **Central Processing Unit (CPU)**
- **Εκτέλεση πράξεων επί των δεδομένων**
 - Αριθμητικές-λογικές πράξεις
 - Μεταφορές δεδομένων από-προς τη μνήμη
 - Συγκρίσεις και διακλάδωση υπό συνθήκη
- **Επιλογή επιθυμητής πράξης**
 - Εντολές
 - Πρόγραμμα: ακολουθία εντολών

Τμήματα ΚΜΕ (CPU)

- Εισαγωγή
- ΚΜΕ



- Επεξεργασία δεδομένων
 - Καθοδήγηση από τη μονάδα ελέγχου
 - Βάσει του προγράμματος εκτέλεσης (εντολών)

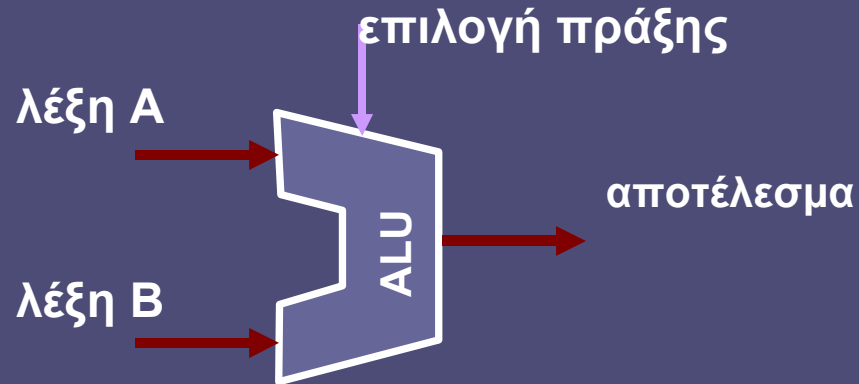
Επεξεργασία δεδομένων

- Εισαγωγή
- ΚΜΕ

- “Μονοπάτι δεδομένων” (Datapath)
- Ποιος εκτελεί τις πράξεις μεταξύ δεδομένων;
 - Αριθμητική-λογική μονάδα (ΑΛΜ)
- Από πού προέρχονται τα δεδομένα εισόδου στην ΑΛΜ και πού αποθηκεύεται το αποτέλεσμα της πράξης;
 - Καταχωρητές

Αριθμητική-Λογική Μονάδα (ΑΛΜ)

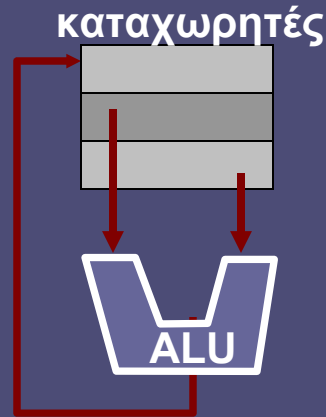
- Εισαγωγή
- ΚΜΕ



- Arithmetic-Logic Unit (ALU)
 - Εκτέλεση αριθμητικών και λογικών πράξεων
 - Πρόσθεση, αφαίρεση, AND, OR, XOR, NOT κλπ
 - Επιλογή πράξης από τη μονάδα ελέγχου
 - Μεταξύ “λέξεων” δεδομένων
 - Πολλαπλάσια του byte

Καταχωρητές (registers)

- Εισαγωγή
- ΚΜΕ

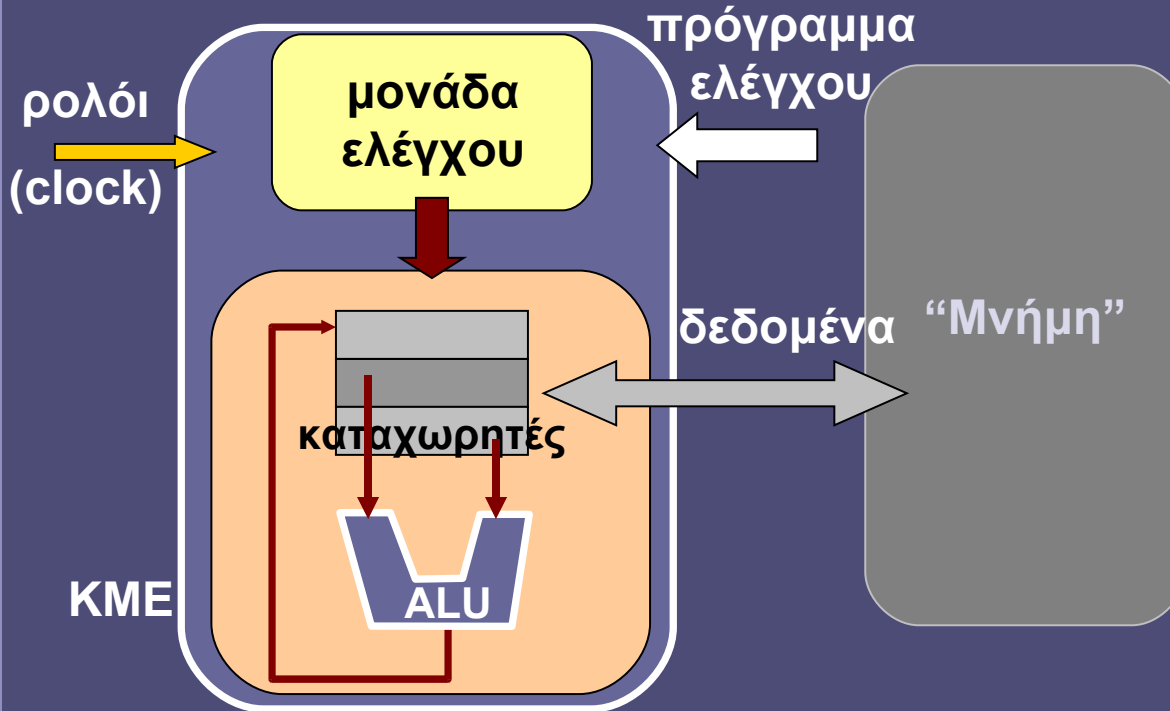


add R1, R2,R3
(R1 = R2 + R3)

- Καταχωρητές
 - Αυτόνομες θέσεις αποθήκευσης μέσα στην ΚΜΕ
 - Μία “λέξη” δεδομένων
 - Γρήγορη προσπέλαση
 - Προσωρινή αποθήκευση δεδομένων
 - Δεδομένα εισόδου και εξόδου ΑΛΜ (ALU)
 - Γενικοί καταχωρητές δεδομένων: π.χ. R1, R2, R3...

Τμήματα ΚΜΕ (CPU)

- Εισαγωγή
- ΚΜΕ



- Εντολές φόρτωσης καταχωρητών από/προς μνήμη (load – store)
 - Ώστε να χρησιμοποιηθούν στη συνέχεια στην ΑΛΜ

Έλεγχος εκτέλεσης εντολών

- Εισαγωγή
- ΚΜΕ

- **Μονάδα ελέγχου (control unit)**
 - Εκτέλεση εντολών μηχανής
- **Ποια είναι η επόμενη προς εκτέλεση εντολή;**
 - Μετρητής προγράμματος (Program Counter – PC)
- **Πώς επιλέγεται η πράξη της ΑΛΜ και οι καταχωρητές για τα δεδομένα εισόδου εξόδου;**
 - Αποκωδικοποίηση εντολών
 - Με βάση τα bits της εντολής δημιουργούνται σήματα ελέγχου λειτουργίας όλης της ΚΜΕ
 - Τήρηση χρονισμών εκτέλεσης – “κύκλος μηχανής”

Program Counter (PC)

- Εισαγωγή
- ΚΜΕ

- **Καταχωρητής διεύθυνσης**
 - Της επόμενης εντολής στη μνήμη
 - Ανάκληση εντολής από μνήμη
- **Αυτόματη αύξηση**
 - Δείχνει στην επόμενη θέση μνήμης (επόμενη εντολή) μετά την εκτέλεση της τρέχουσας εντολής
 - $PC = PC + d$
- **Ειδική περίπτωση: διακλάδωση**
 - Μετά από σύγκριση
 - Φόρτωση του PC με διαφορετική τιμή
 - Μεταφορά της ροής εκτέλεσης σε άλλο σημείο
 - Υλοποίηση δομών if ... then ... else

Τι περιγράφουν τα bits μιας εντολής μηχανής;

- Εισαγωγή
- ΚΜΕ



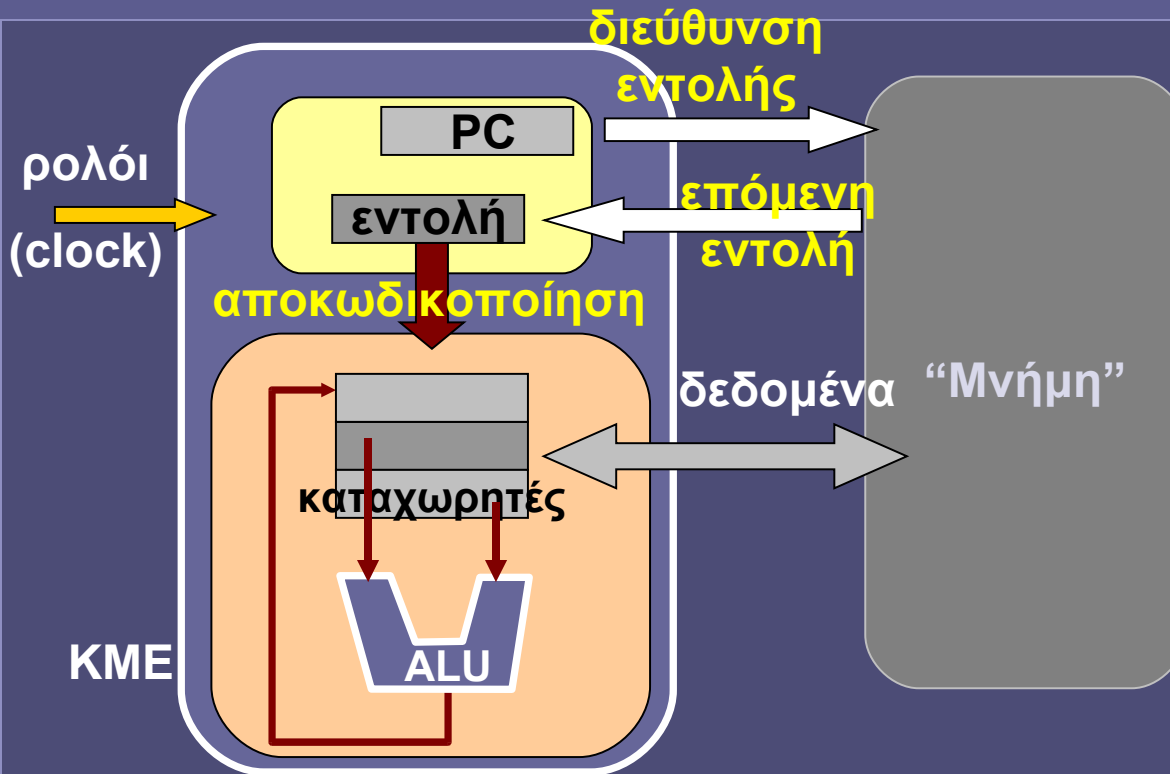
Περιγράφει το είδος της πράξης που θα εκτελεστεί

Περιγράφουν την **προέλευση** των δεδομένων εισόδου (αριθμό καταχωρητή, διεύθυνση μνήμης κλπ) και τον **προορισμό** των δεδομένων εξόδου (αποτέλεσμα πράξης)

Το είδος της πράξης προσδιορίζει τον τύπο, την προέλευση και τον αριθμό των δεδομένων που συμμετέχουν στην πράξη !

Τμήματα ΚΜΕ (CPU)

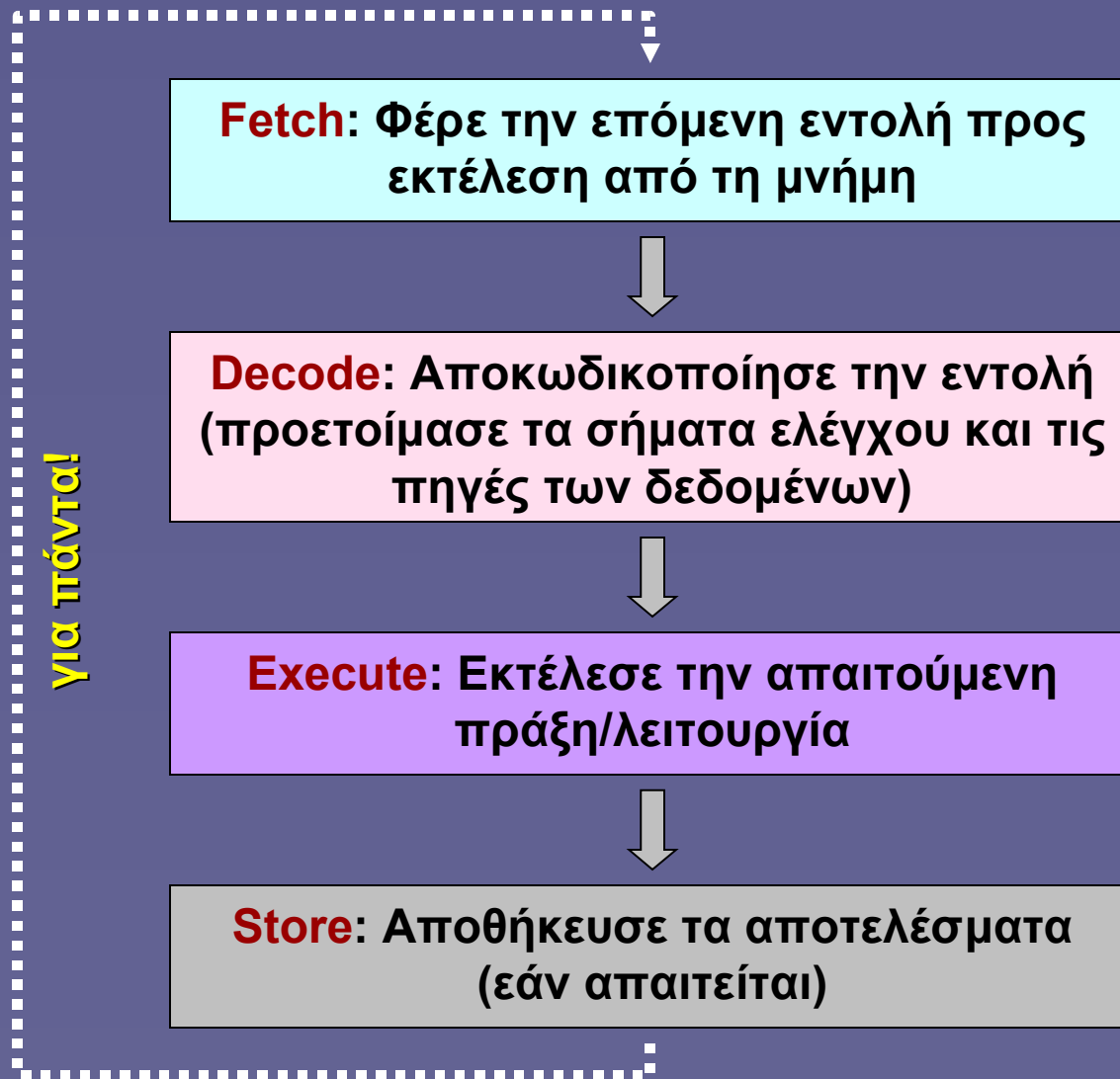
- Εισαγωγή
- ΚΜΕ



- **Αποκωδικοποίηση εντολής**
 - Δημιουργία σημάτων ελέγχου ανάλογα με bits εντολής

Εκτέλεση εντολών: ο κύκλος μηχανής

- Εισαγωγή
- ΚΜΕ



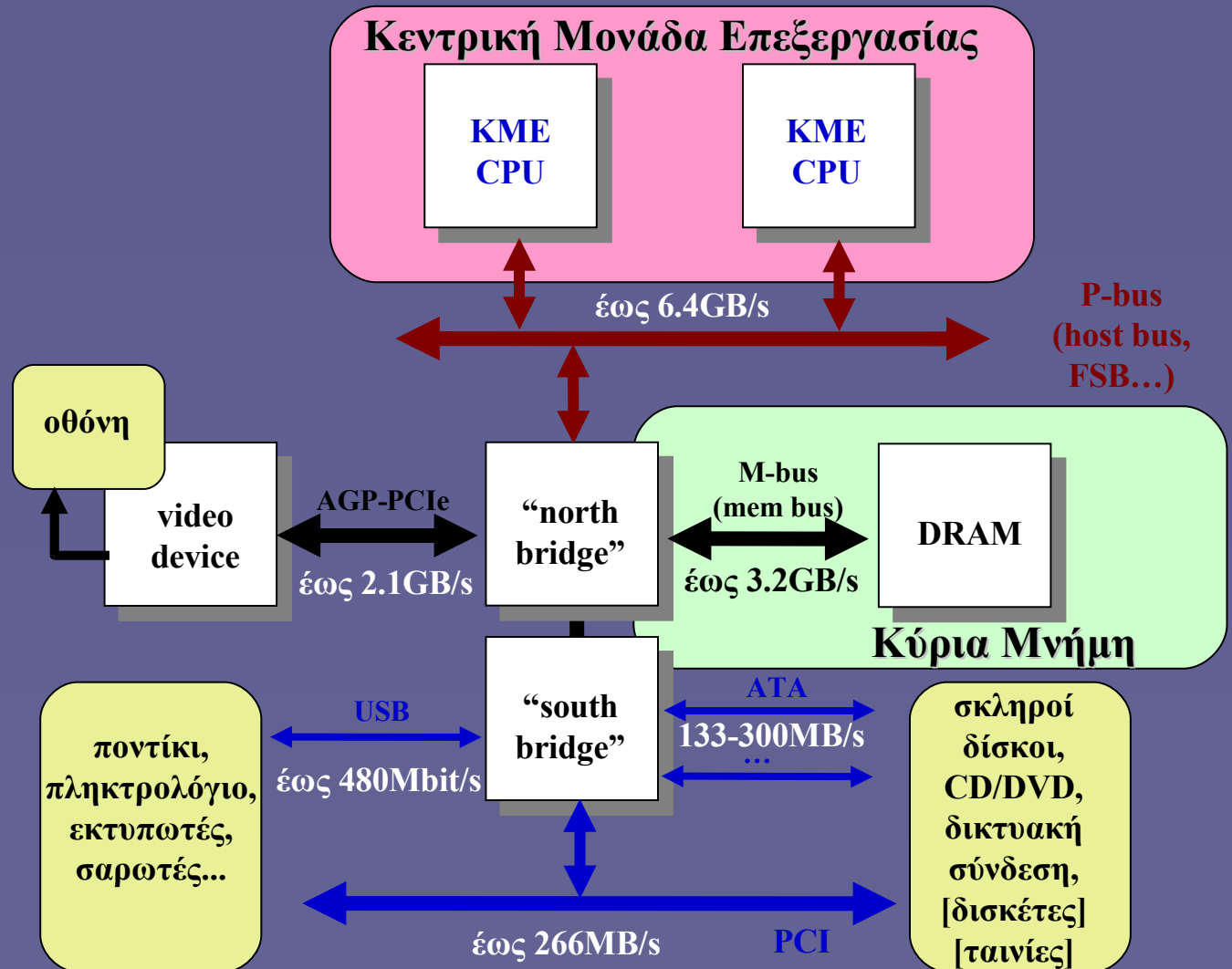
Αρχιτεκτονικές συνόλου εντολών

- Εισαγωγή
- ΚΜΕ

- **Υπολογιστές σύνθετου συνόλου εντολών**
 - Complex Instruction Set Computer (**CISC**)
 - Μεγάλο σετ εντολών (σύνθετες λειτουργίες)
 - Εύκολη συγγραφή προγραμμάτων
 - Πολύπλοκη ΚΜΕ
- **Υπολογιστές περιορισμένου σετ εντολών**
 - Reduced Instruction Set Computer (**RISC**)
 - Μικρό σύνολο απλών εντολών (απλές λειτουργίες)
 - Μεγαλύτερα προγράμματα
 - Απλούστερη ΚΜΕ
 - Μεγαλύτερη χρησιμοποίηση (παραλληλία)

Τα βασικά υποσυστήματα

- Εισαγωγή
- ΚΜΕ
- Μνήμη



Κύρια μνήμη

- Εισαγωγή
- ΚΜΕ
- Μνήμη

- **Βασικό υποσύστημα του υπολογιστή**
 - Αποθήκευση δεδομένων και προγραμμάτων
- **Συλλογή από θέσεις αποθήκευσης**
 - Σε κάθε θέση αποθηκεύεται μία λέξη των n bits
 - n = εύρος λέξης
 - π.χ. $n = 8, 16, 32$ bits (ή 1, 2, 4 bytes)
 - Σε κάθε θέση αντιστοιχεί μία μοναδική διεύθυνση
 - μη προσημασμένος δυαδικός αριθμός
 - με m bits επιλέγουμε μεταξύ 2^m διευθύνσεων
 - Χώρος διευθύνσεων μνήμης: $0 \dots 2^m - 1$
 - Συνολική χωρητικότητα μνήμης:
 $2^m \times n$ bits

Μονάδες μέτρησης χωρητικότητας

- Εισαγωγή
- ΚΜΕ
- Μνήμη

- 1 Byte = 8 bits
- 1 KiloByte (KB) = 2^{10} Bytes
 - 1.024 Bytes
- 1 MegaByte (MB) = 2^{10} KB = 2^{20} Bytes
 - 1.048.576 Bytes
- 1 GigaByte (GB) = 2^{10} MB = 2^{20} KB = 2^{30} bytes
 - 1.073.741.824 Bytes
- Κλπ...

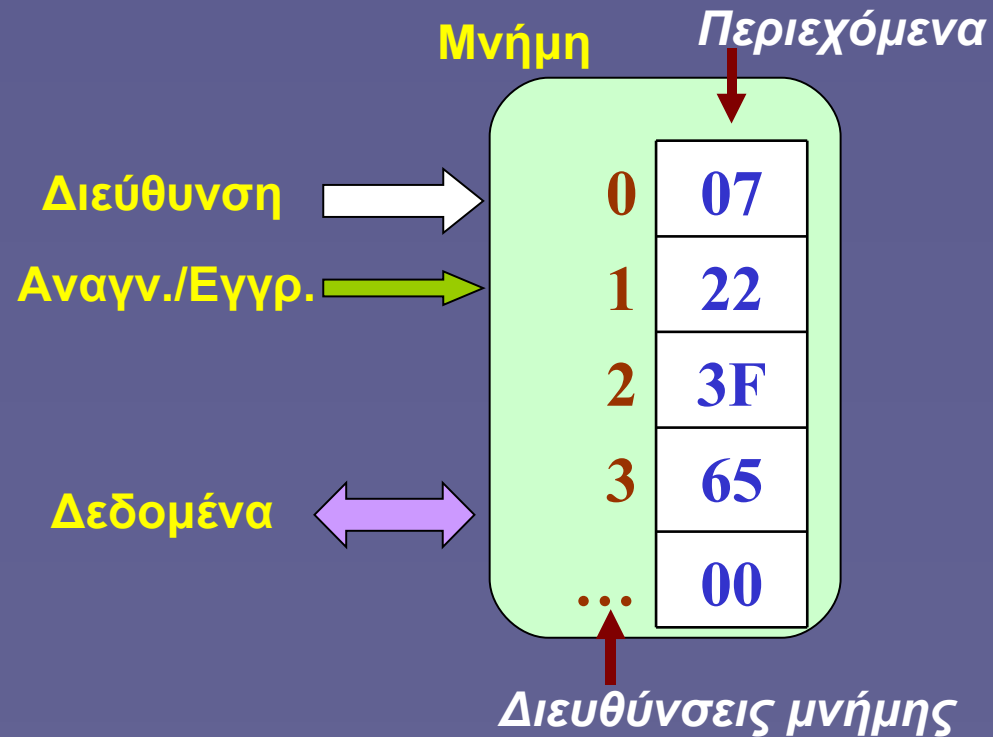
Τεχνολογίες μνημών

- Εισαγωγή
- ΚΜΕ
- Μνήμη

- **Μνήμη τυχαίας προσπέλασης (RAM)**
 - Random Access Memory
 - Ανάγνωση-Εγγραφή
 - Στατική (SRAM) και δυναμική (DRAM)
 - Μέθοδος υλοποίησης “κελιών” μνήμης
 - Και στις δύο χάνονται τα δεδομένα με τη διακοπή της τροφοδοσίας
- **Μνήμες μόνιμης αποθήκευσης**
 - Διατήρηση δεδομένων χωρίς τροφοδοσία
 - Μόνο για ανάγνωση
 - Read Only Memory(ROM), PROM, EPROM, EEPROM..
 - Αργή ανάγνωση-εγγραφή αλλά μαζική αποθήκευση
 - FLASH

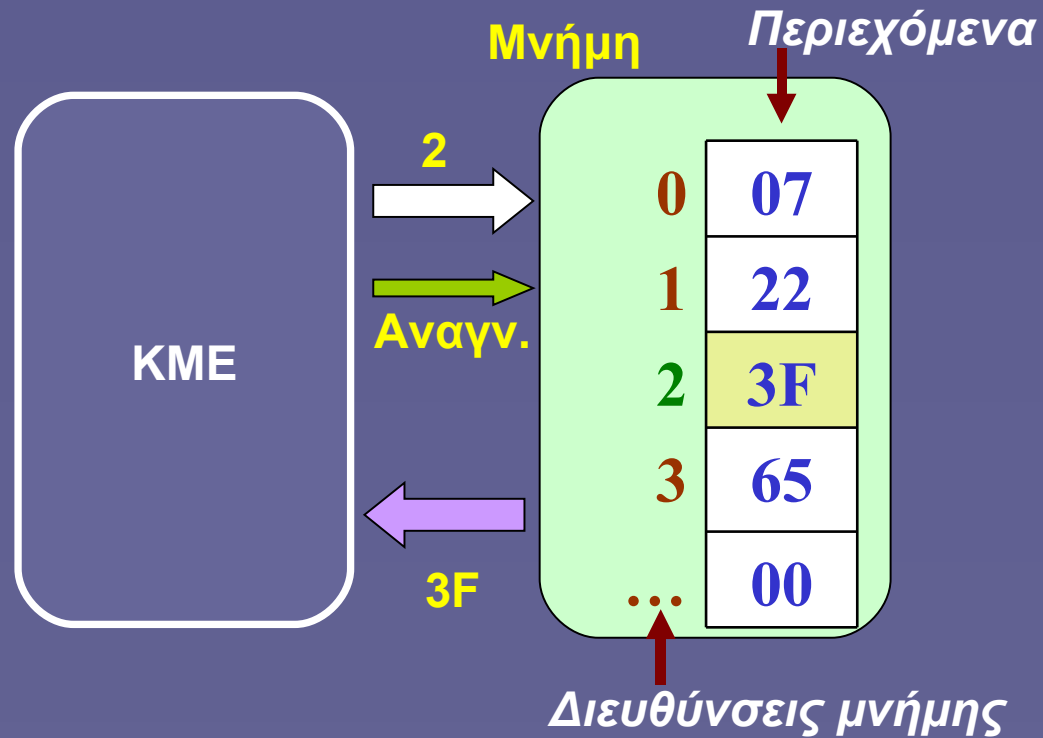
Μοντέλο λειτουργίας μνήμης

- Εισαγωγή
- ΚΜΕ
- Μνήμη



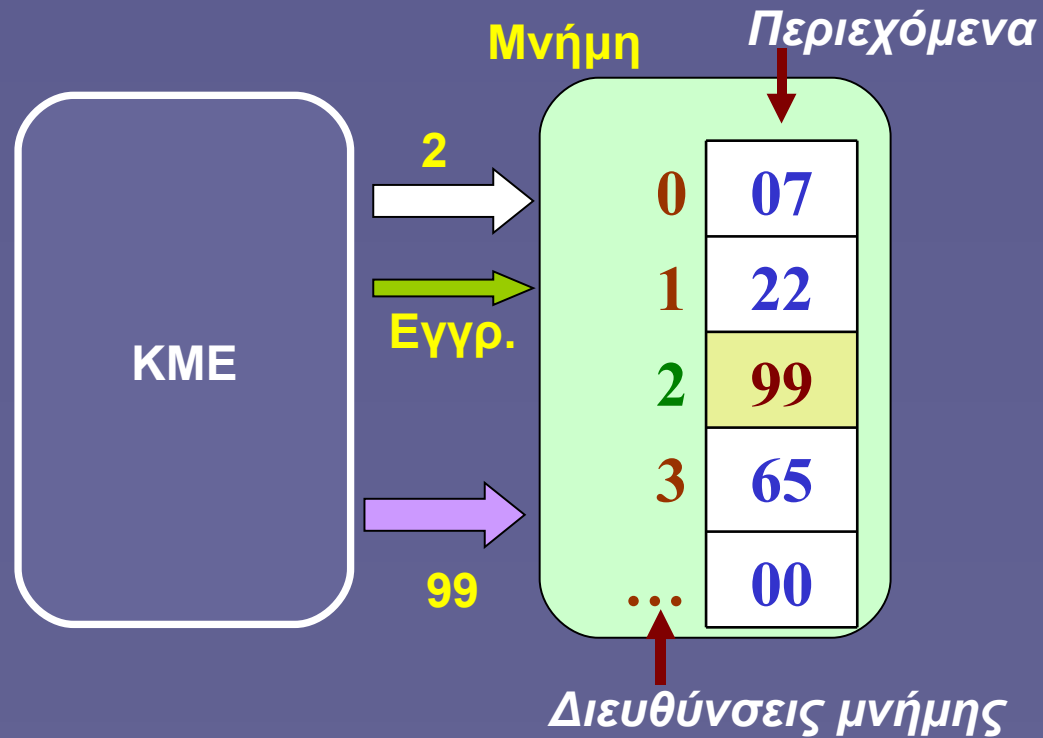
Ανάγνωση από μνήμη

- Εισαγωγή
- ΚΜΕ
- Μνήμη



Εγγραφή στη μνήμη

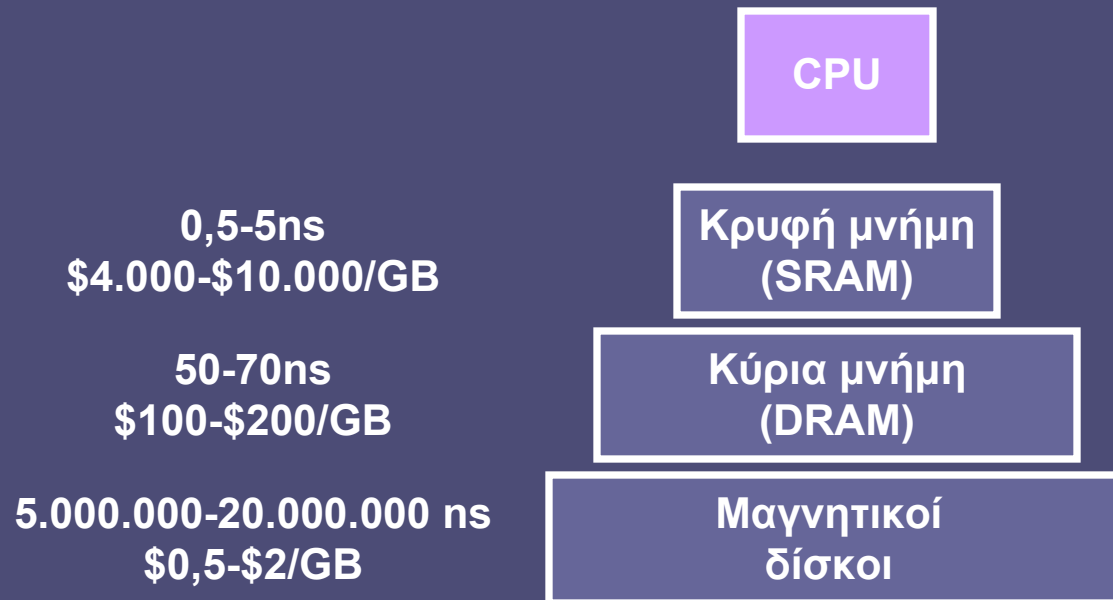
- Εισαγωγή
- ΚΜΕ
- Μνήμη



Ιεραρχία Μνήμης

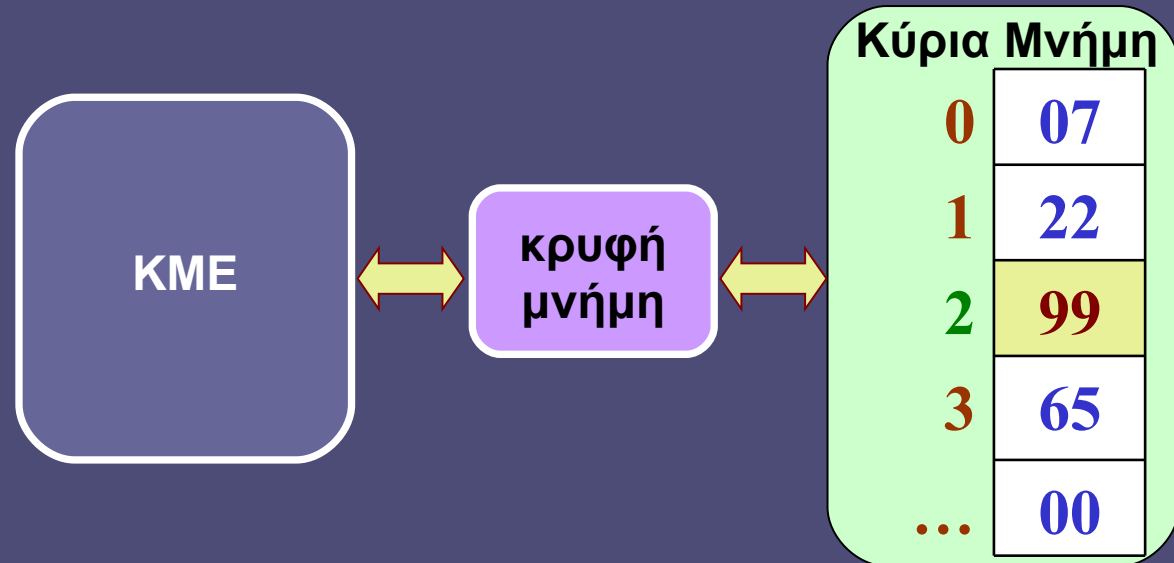
- Εισαγωγή
- ΚΜΕ
- Μνήμη

- Προσέγγιση της ιδανικής μνήμης
 - Ο επεξεργαστής να βλέπει “μνήμη”
 - Με την ταχύτητα του υψηλότερου επιπέδου
 - Και το μέγεθος του χαμηλότερου επιπέδου



Κρυφή μνήμη

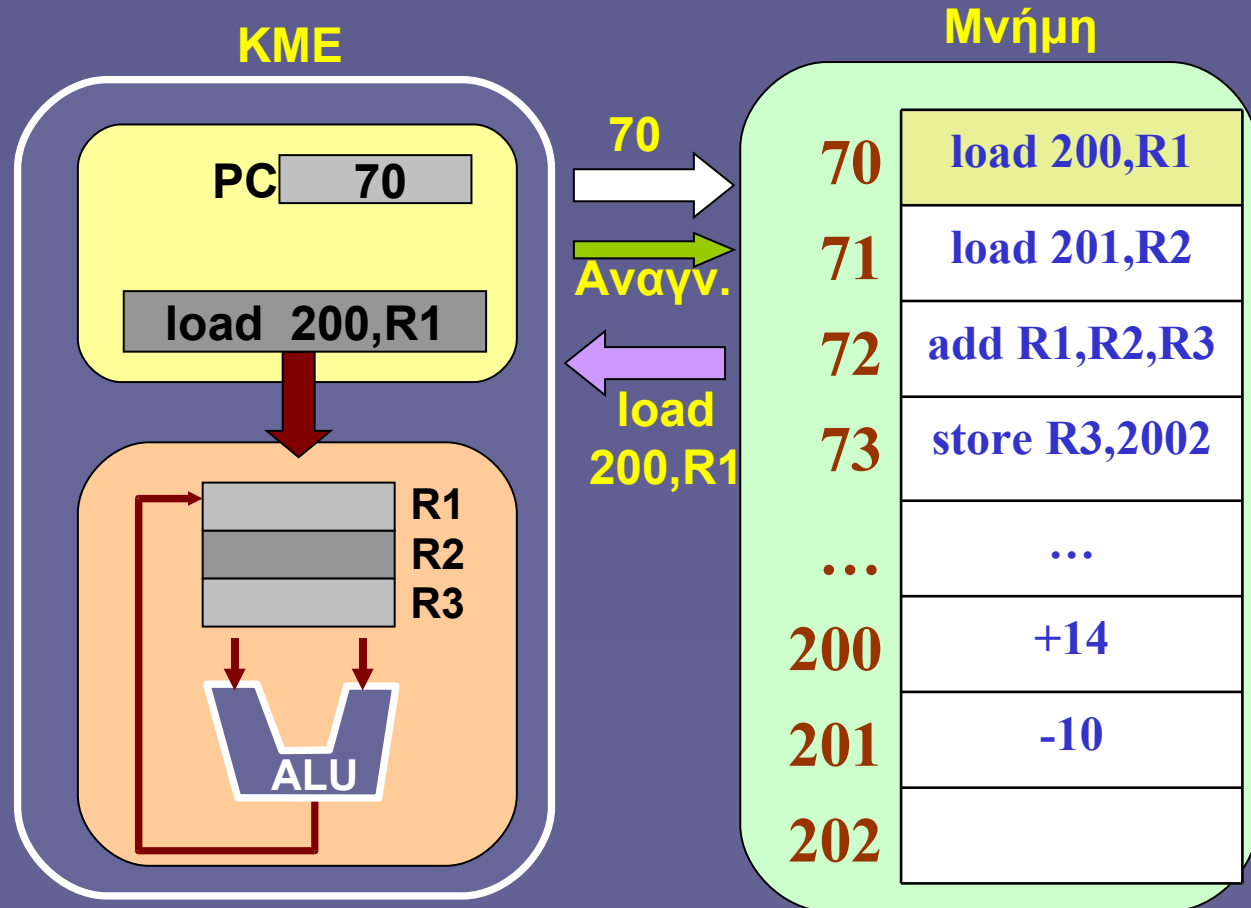
- Εισαγωγή
- ΚΜΕ
- Μνήμη



- Μεταξύ ΚΜΕ και κύριας μνήμης
 - Περιέχει **ένα μέρος μόνο** των περιεχομένων της κύριας μνήμης
 - Γρηγορότερη από κύρια μνήμη
 - Εκμετάλλευση της **τοπικότητας** των προσπελάσεων
 - Χωρική και χρονική

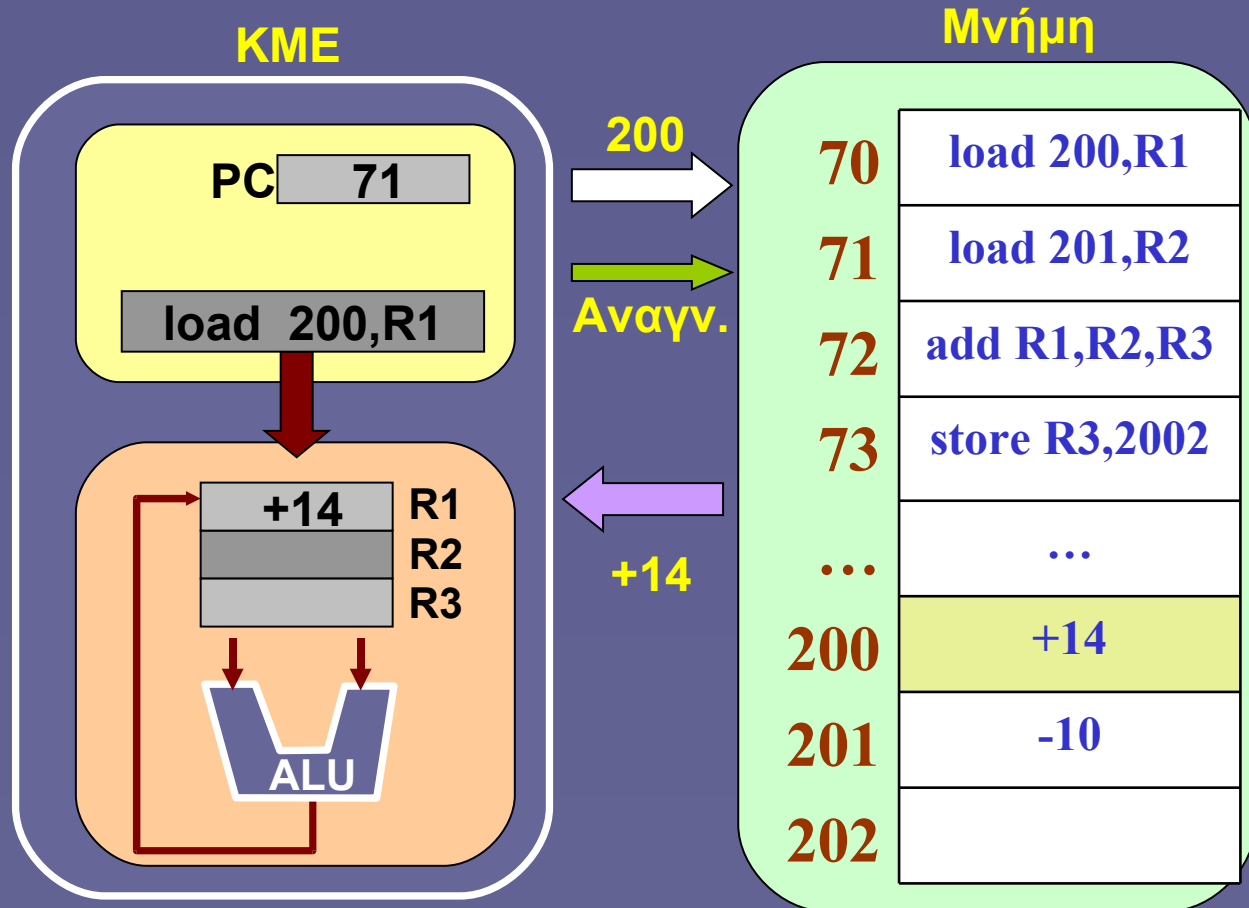
Παράδειγμα εκτέλεσης εντολών

- Εισαγωγή
- ΚΜΕ
- Μνήμη
- Παράδειγμα εκτέλεσης εντολών



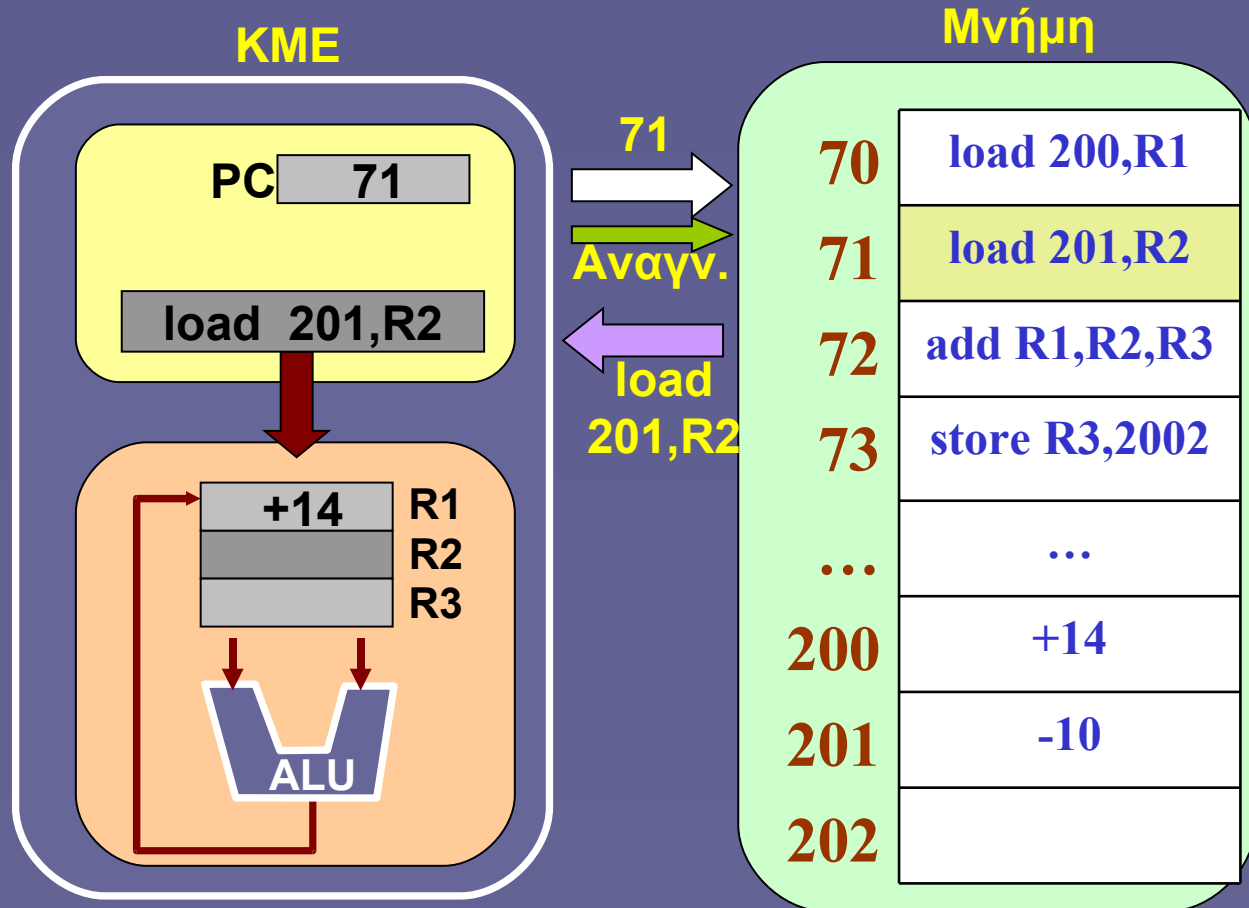
Παράδειγμα εκτέλεσης εντολών

- Εισαγωγή
- ΚΜΕ
- Μνήμη
- Παράδειγμα εκτέλεσης εντολών



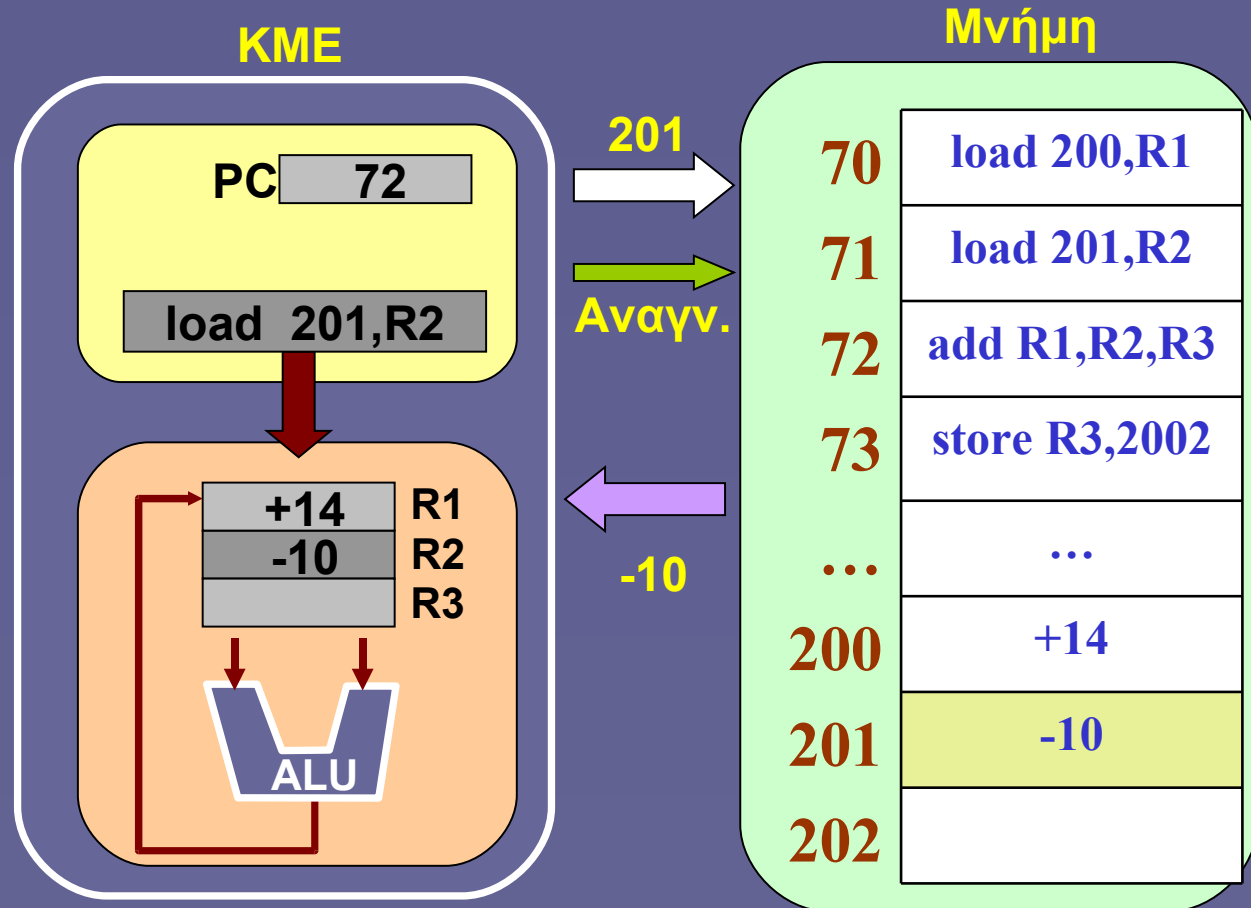
Παράδειγμα εκτέλεσης εντολών

- Εισαγωγή
- ΚΜΕ
- Μνήμη
- Παράδειγμα εκτέλεσης εντολών



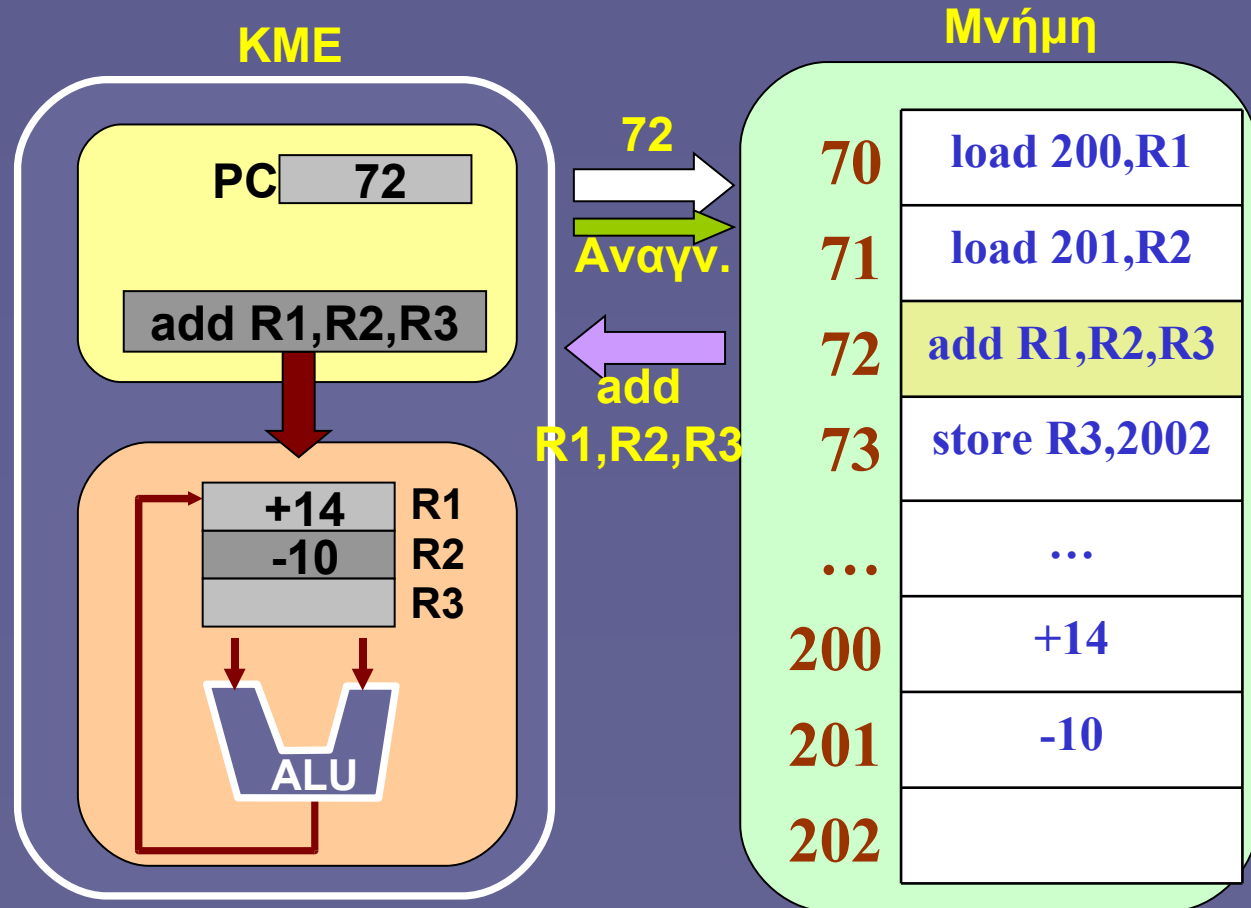
Παράδειγμα εκτέλεσης εντολών

- Εισαγωγή
- ΚΜΕ
- Μνήμη
- Παράδειγμα εκτέλεσης εντολών



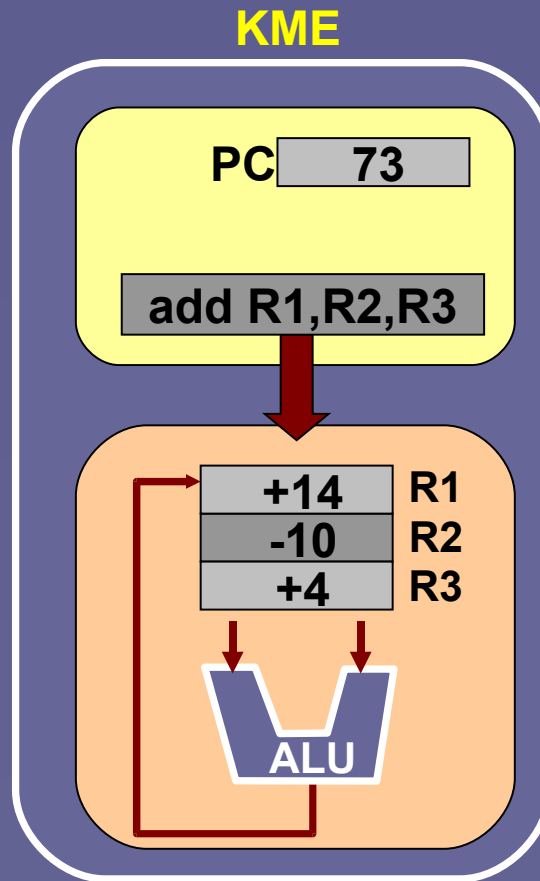
Παράδειγμα εκτέλεσης εντολών

- Εισαγωγή
- ΚΜΕ
- Μνήμη
- Παράδειγμα εκτέλεσης εντολών



Παράδειγμα εκτέλεσης εντολών

- Εισαγωγή
- ΚΜΕ
- Μνήμη
- Παράδειγμα εκτέλεσης εντολών

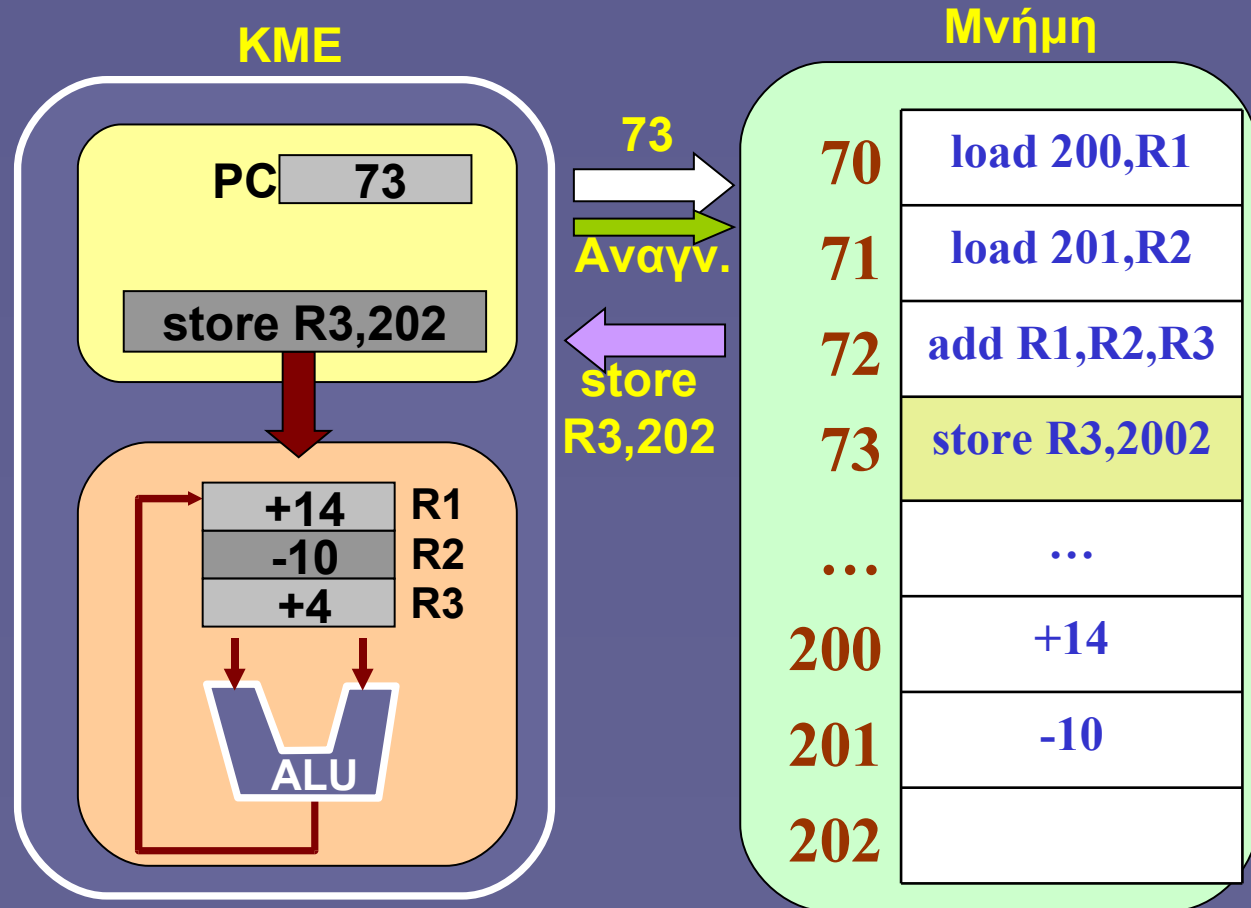


Μνήμη

70	load 200,R1
71	load 201,R2
72	add R1,R2,R3
73	store R3,2002
...	...
200	+14
201	-10
202	

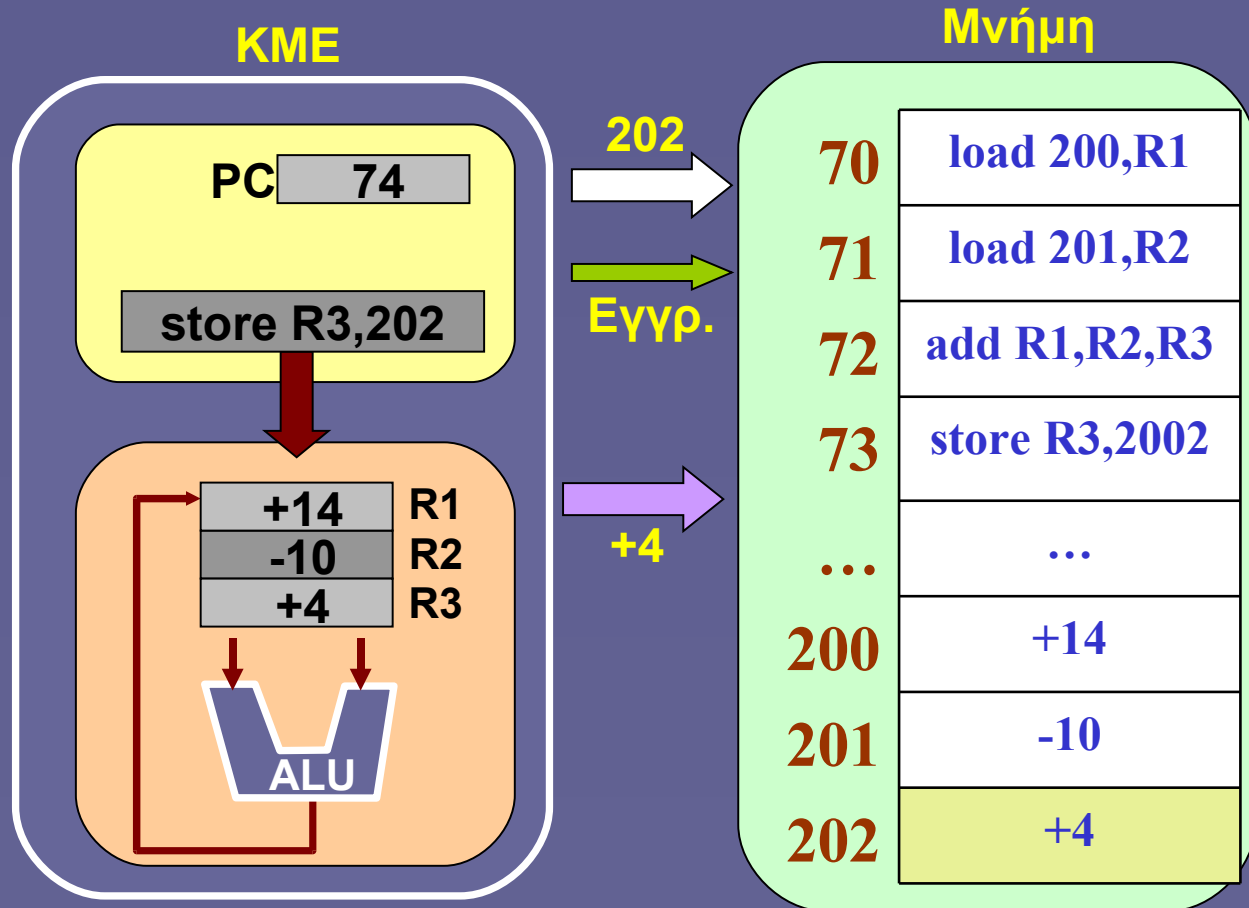
Παράδειγμα εκτέλεσης εντολών

- Εισαγωγή
- ΚΜΕ
- Μνήμη
- Παράδειγμα εκτέλεσης εντολών



Παράδειγμα εκτέλεσης εντολών

- Εισαγωγή
- ΚΜΕ
- Μνήμη
- Παράδειγμα εκτέλεσης εντολών



Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2007-08

Οργάνωση Υπολογιστών (II)

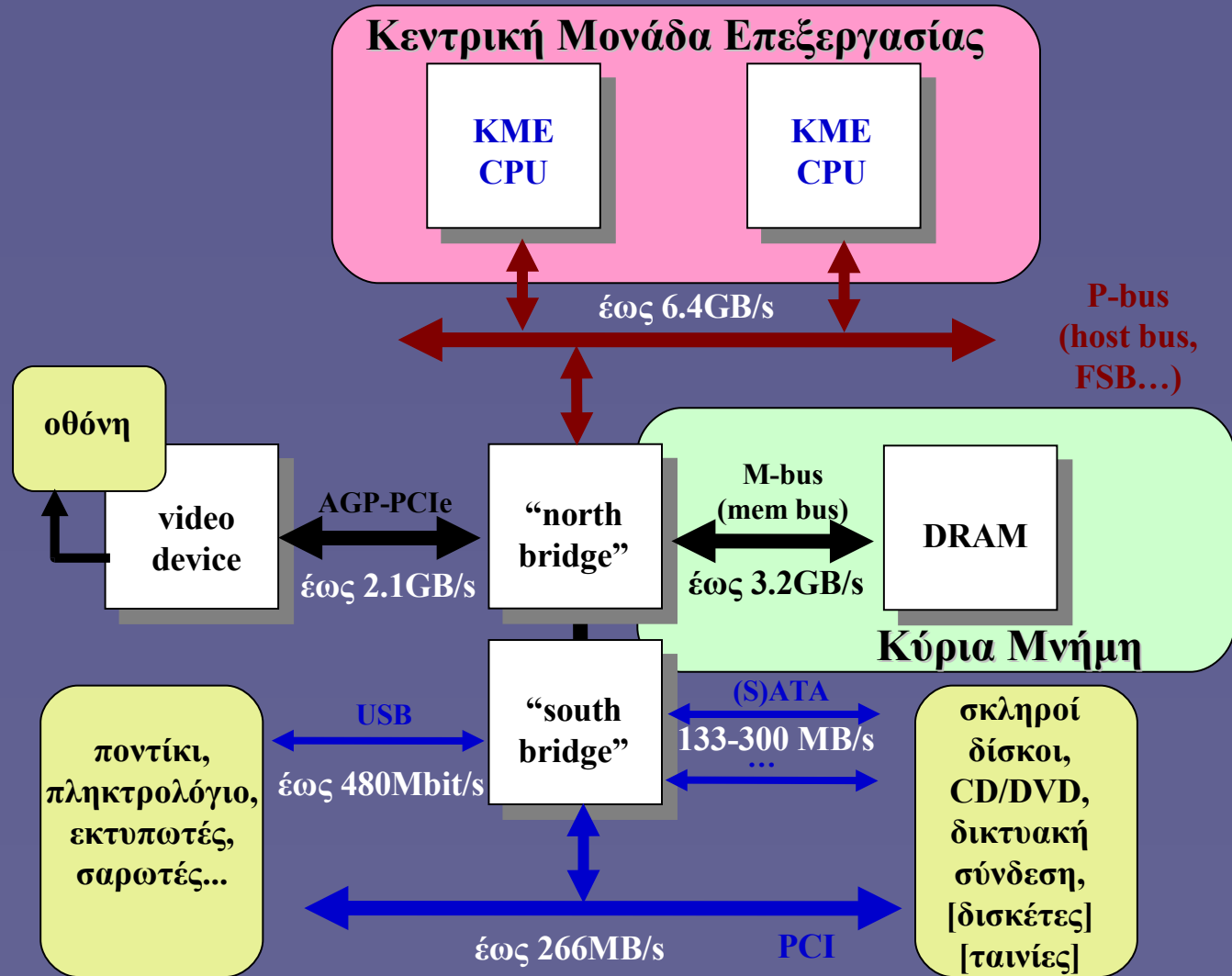
(δίαυλοι και συσκευές εισόδου-εξόδου)

<http://www.ionio.gr/~mistral/tp/csintro/>

Μ.Στεφανιδάκης

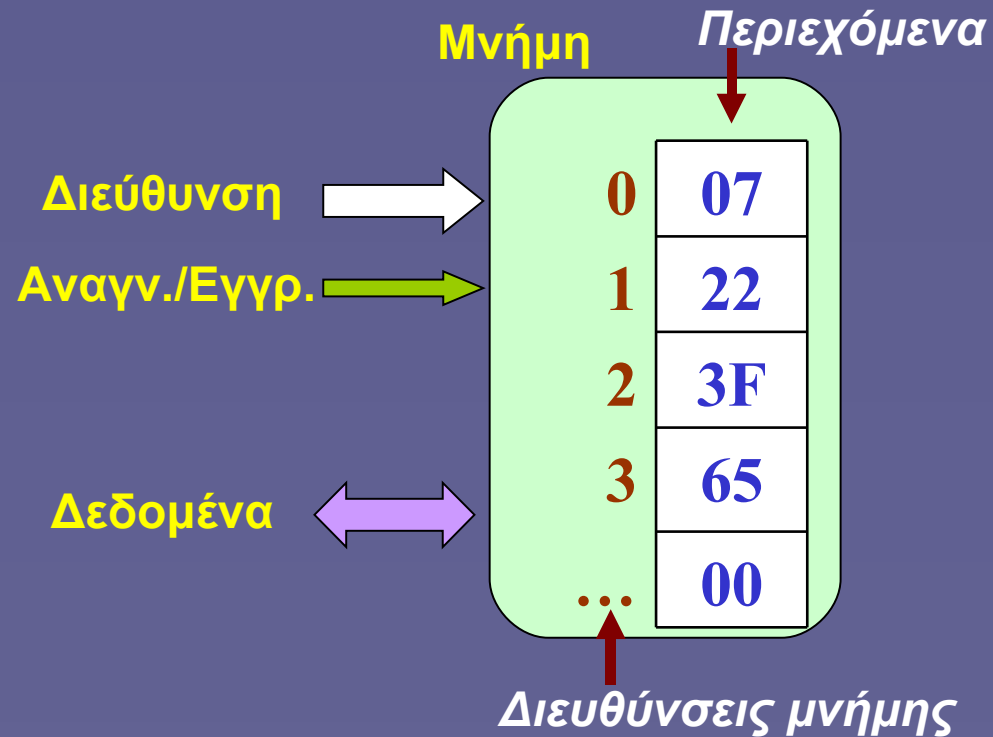
Στο προηγούμενο μάθημα

- Εισαγωγή



Μοντέλο λειτουργίας μνήμης

- Εισαγωγή

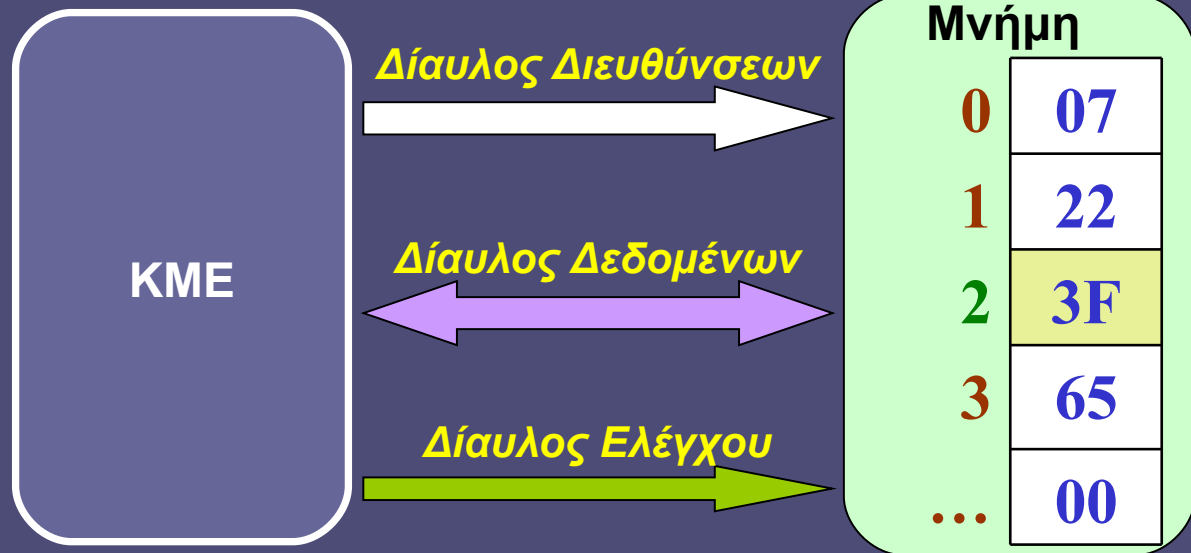


Διασύνδεση επεξεργαστή-μνήμης

- Εισαγωγή
- Δίαυλοι



Δίαυλοι: ομάδες αγωγών – διασύνδεση για τη μεταφορά πληροφορίας. Ο ρυθμός μεταφοράς στους διαύλους επηρεάζει τη συνολική απόδοση του υπολογιστή!



- Διεύθυνση
 - Προς/από πού γίνεται η προσπέλαση;
- Δεδομένα
 - Τα δεδομένα ανάγνωσης/εγγραφής
- Έλεγχος
 - Ανάγνωση ή εγγραφή; - συγχρονισμός μεταφοράς

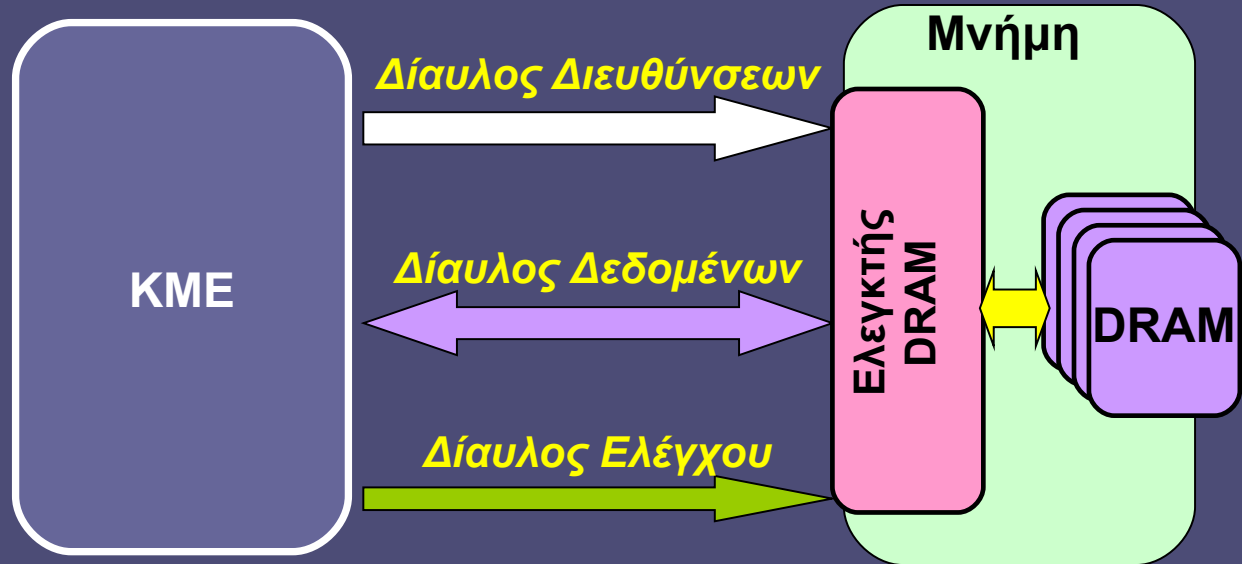
Η κύρια μνήμη σήμερα

- Εισαγωγή
- Δίαυλοι

- **Υποσύστημα κύριας μνήμης**
 - Μεγάλες χωρητικότητες (GBs)
 - Μεγάλο εύρος (bits) διαύλου μεταφοράς
 - Για την ικανοποίηση των αναγκών της ΚΜΕ
 - 64 και πλέον bits ανά μεταφορά
 - Βελτιστοποιήσεις απόδοσης
 - Για τον ελάχιστο χρόνο προσπέλασης δεδομένων
 - ≥ 400 MTransfers/sec, ≥ 3.2 GB/s
- **Ελεγκτής κύριας μνήμης**
 - Λόγω πολυπλοκότητας διασύνδεσης
 - ΚΜΕ δεν συνδέεται απευθείας στη μνήμη
 - Αλλά: παρεμβολή **ελεγκτή μνήμης**
 - **Το μοντέλο προσπέλασης δεν αλλάζει**

Διασύνδεση επεξεργαστή-μνήμης

- Εισαγωγή
- Δίαυλοι



- Σε ένα μοντέρνο υπολογιστικό σύστημα
 - Κύρια μνήμη: σύνθετο υποσύστημα

Είσοδος – Έξοδος στον υπολογιστή

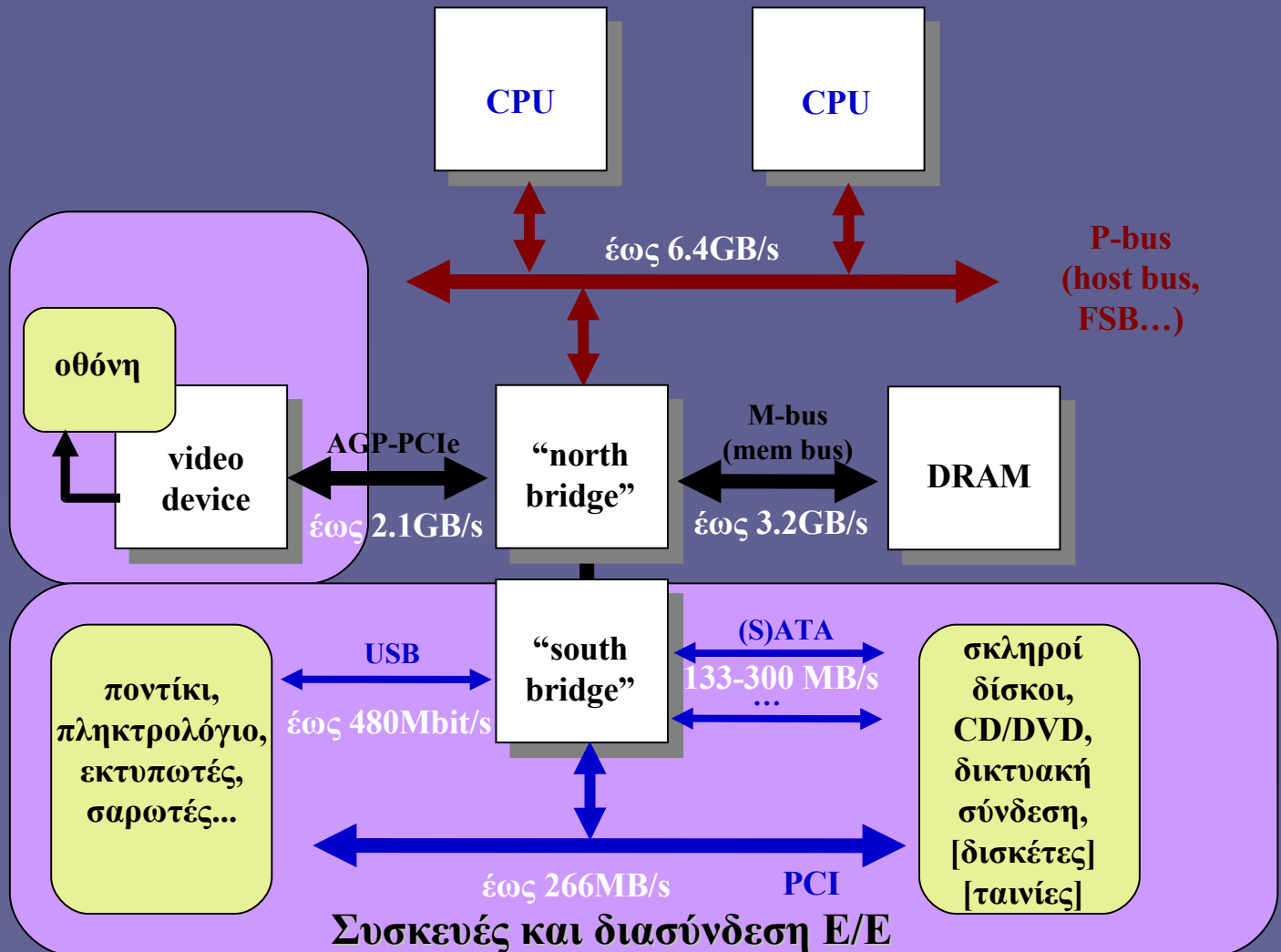
- Εισαγωγή
- Δίαυλοι
- **Συσκευές E/E**

- **Συσκευές εισόδου-εξόδου (E/E)**
 - Συμβατικά: οτιδήποτε εκτός από ΚΜΕ και μνήμη
 - Συσκευές με **τελείως διαφορετικούς** μεταξύ τους ρόλους και χαρακτηριστικά
 - **Λειτουργία**: είσοδος, έξοδος ή και τα δύο (αποθήκευση)
 - **Συνεργάτης** (στην άλλη άκρη): άνθρωπος ή μηχανή
 - **Ρυθμός μεταφοράς** δεδομένων

Συσκευή	Λειτουργία	Ρυθμός (Mbps)
πληκτρολόγιο	είσοδος	0,0001
ποντίκι	είσοδος	0,0038
εκτυπωτής	έξοδος	3,2
σύνδεση δικτύου	είσοδος/έξοδος	100-1000
μαγν. δίσκος	αποθήκευση	240-2565
οθόνη	έξοδος	800-8000

Ένα τυπικό υπολογιστικό σύστημα σήμερα

- Εισαγωγή
- Δίαυλοι
- Συσκευές E/E



Βασικές συσκευές Ε/Ε

- Εισαγωγή
- Δίαυλοι
- Συσκευές Ε/Ε

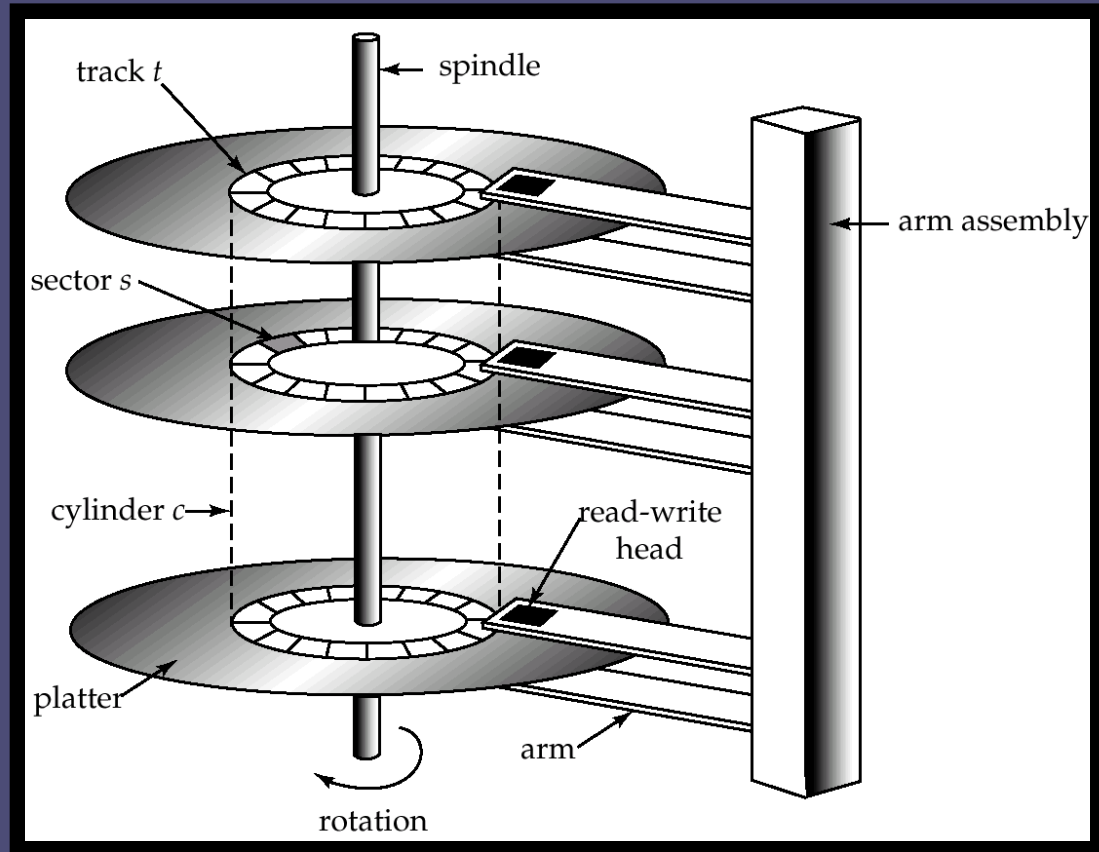


Η πληροφορία σε οθόνη διαστάσεων 1280x1024 pixels ανανεώνεται 60 φορές/sec. Αγνοώντας πρόσθετες επιβαρύνσεις, με τι ρυθμό πρέπει να αποστέλλεται η πληροφορία απεικόνισης στην οθόνη;

- **Πληκτρολόγιο**
 - Είσοδος από χρήστη – κωδικοί πλήκτρων
- **Οθόνη**
 - Έξοδος προς χρήστη – κωδικοί πλήκτρων
- **Αποθηκευτικά μέσα**
 - Μαγνητικοί και οπτικοί δίσκοι
 - Θεωρούνται ως **δευτερεύουσα μνήμη**
 - Εκατοντάδες χιλιάδες φορές αργότερα από κύρια μνήμη
 - Τα δεδομένα πρέπει πρώτα να περάσουν στην κύρια μνήμη και μετά στην ΚΜΕ
 - Μόνιμη αποθήκευση
 - Και εκτός τροφοδοσίας

Μαγνητικοί δίσκοι

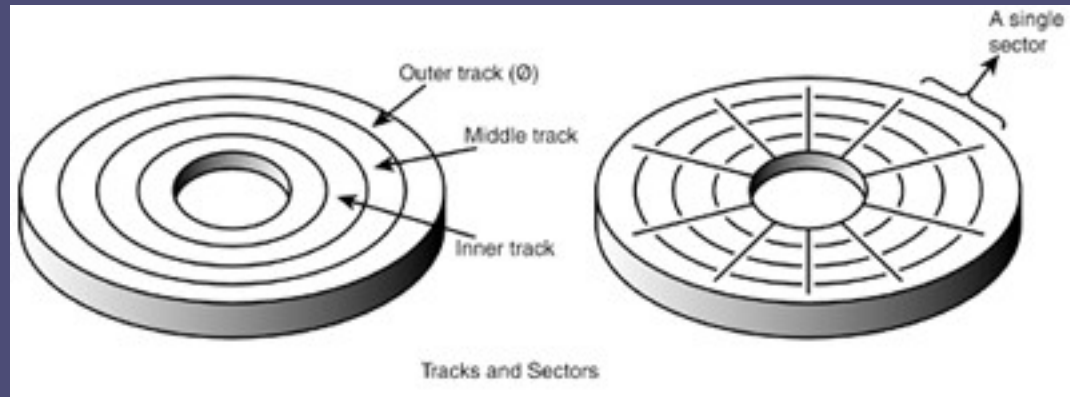
- Εισαγωγή
- Δίσκοι
- Συσκευές E/E



- Αποθήκευση πληροφορίας μαγνητικά στην επιφάνεια των δίσκων
- Κεφαλές ανάγνωσης-εγγραφής

Δομή μαγνητικών δίσκων

- Εισαγωγή
- Δίσκοι
- Συσκευές E/E



- Οργάνωση σε **tracks** και **sectors**
 - Sector = 512 – 4096 bytes
 - Πληροφορία αναγνώρισης sector
- Ταχύτητα περιστροφής
 - 7.200 – 10.000 RPM

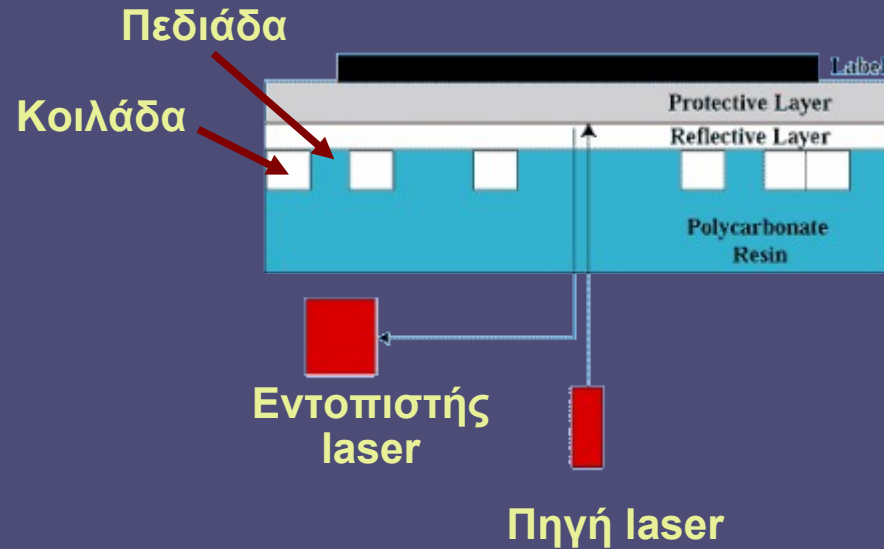
Προσπέλαση πληροφορίας στον δίσκο

- Εισαγωγή
- Δίσκοι
- Συσκευές E/E

- **Προσπέλαση πληροφορίας**
 - Σε τυχαία θέση (sector) του δίσκου
 - Μετάβαση κεφαλής στο επιθυμητό track
 - **Seek time**
 - Αναμονή για εμφάνιση του επιθυμητού sector κάτω από την κεφαλή
 - **Rotational latency**
 - Ανάγνωση και μεταφορά των δεδομένων του sector
 - **(Block) transfer time**
 - Όλες οι κεφαλές κινούνται μαζί – όχι ανεξάρτητα!

Οπτικοί δίσκοι (CD-ROM)

- Εισαγωγή
- Δίσκοι
- Συσκευές Ε/Ε



- Ανάγνωση:
 - Διαφορετική ανάκλαση σε πεδιάδες και κοιλάδες
 - Ανάκτηση του bit



Ταχύτητα CD-ROM:

1x = 150KB/sec,
2x = 300KB/sec,
κλπ

Κωδικοποίηση δεδομένων σε CD-ROM

- Εισαγωγή
- Δίαυλοι
- Συσκευές E/E

- Κάθε byte μετατρέπεται σε ένα σύμβολο των 14 bits
 - Πρόσθετη πληροφορία για διόρθωση λαθών
 - 6 bits – κώδικας Hamming
 - Ένα πλαίσιο (frame) αποτελείται από 24 σύμβολα
 - Ένας τομέας (sector) περιέχει 98 πλαίσια

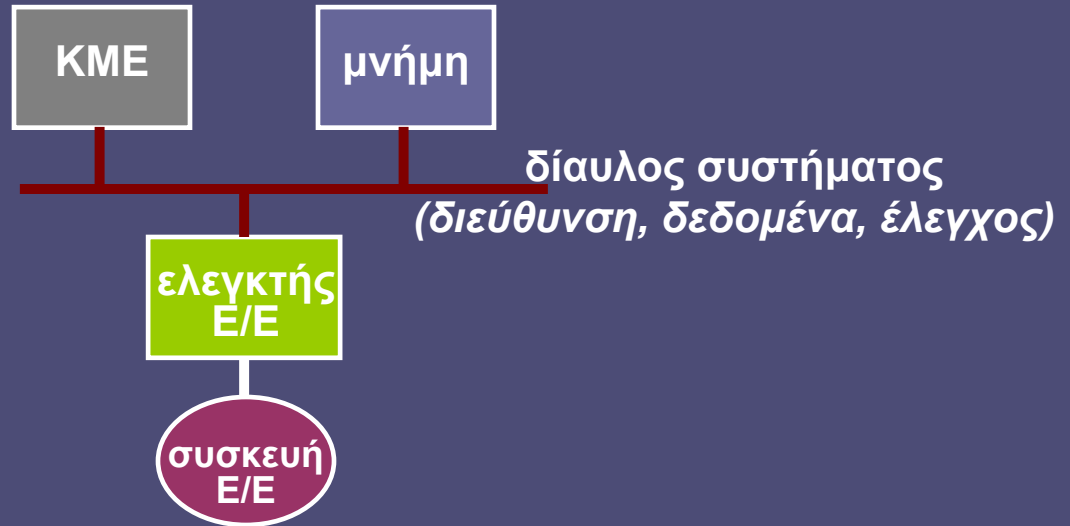
CD-R και CD-RW

- Εισαγωγή
- Δίσκοι
- Συσκευές E/E

- **CD-R**
 - Γράφονται μία φορά
 - Προσομοίωση κοιλάδων και πεδιάδων
 - Με την αλλοίωση επιφάνειας χρωστικής ουσίας
 - Με laser υψηλής ισχύος
- **CD-RW**
 - Επανεγγράψιμοι δίσκοι
 - Κράμα 2 καταστάσεων: κρυσταλλική και άμορφη
 - Προσομοίωση κοιλάδων και πεδιάδων
 - Μεταπήδηση μεταξύ των δύο καταστάσεων με ακτίνα laser διαφορετικής ισχύος
- **DVD-...**
 - Διαφορετικές διαστάσεις κοιλάδων και tracks, laser διαφορετικού μήκους κύματος

Διασύνδεση συσκευών E/E

- Εισαγωγή
- Δίαυλοι
- Συσκευές E/E
- Διασύνδεση E/E



- Διασύνδεση στον δίαυλο του συστήματος μέσω ενός **ελεγκτή E/E** (I/O controller ή adapter)
- Ο ελεγκτής αναλαμβάνει την επικοινωνία με την **πολύ αργότερη** συσκευή E/E

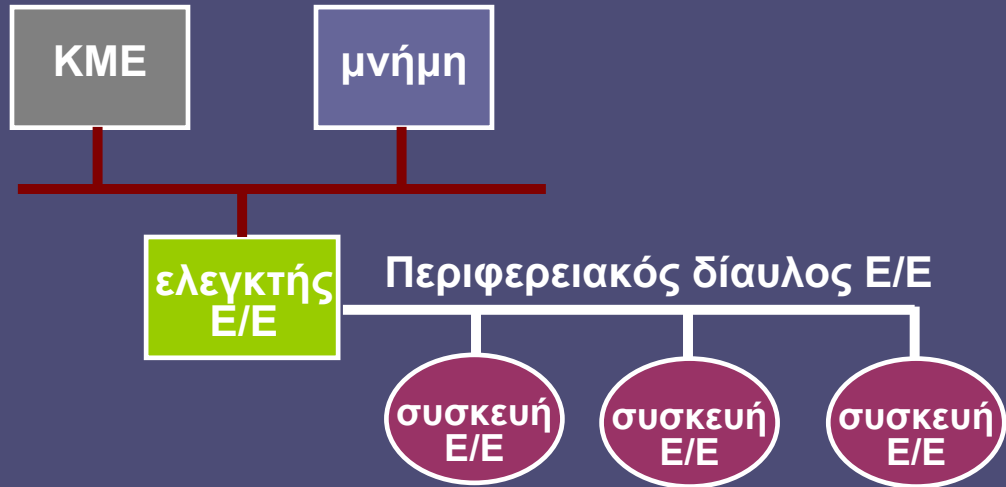
Επικοινωνία συσκευών-ελεγκτή E/E

- Εισαγωγή
- Δίαυλοι
- Συσκευές E/E
- Διασύνδεση E/E

- Πρωτόκολλο επικοινωνίας ανάλογο της φύσης της συσκευής
 - Χαρακτήρες για πληκτρολόγιο-εκτυπωτή
 - Πληροφορία χρώματος για οθόνη
 - Πακέτα δεδομένων για το δίκτυο
 - Μεταφορά μπλοκ δεδομένων για δίσκους
- Ο ελεγκτής E/E αναλαμβάνει να μεταφράσει από/προς αιτήσεις ανάγνωσης-εγγραφής της ΚΜΕ

Περιφερειακοί δίαυλοι E/E

- Εισαγωγή
- Δίαυλοι
- Συσκευές E/E
- Διασύνδεση E/E



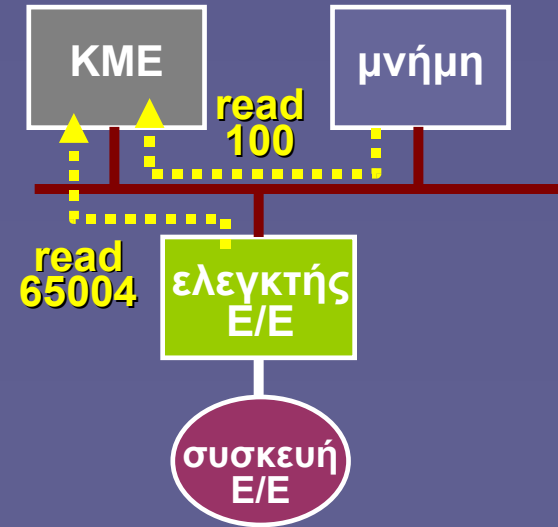
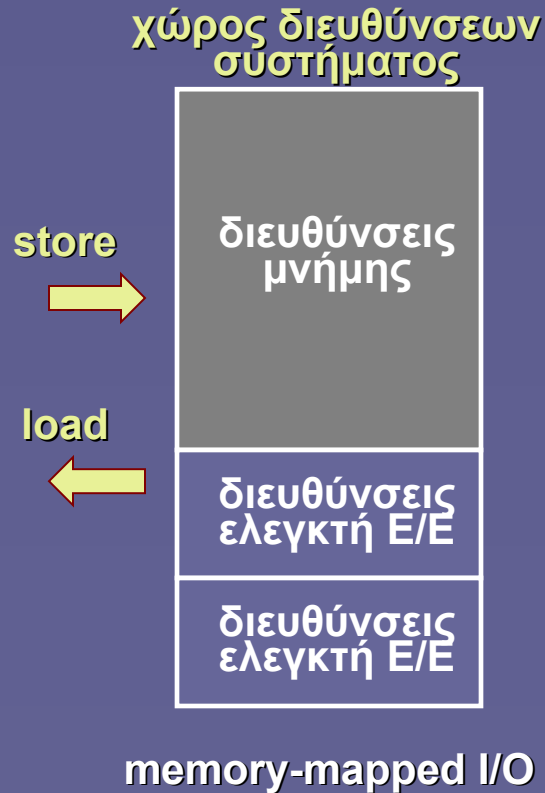
- Όταν η σύνδεση ελεγκτή και συσκευών E/E γενικεύεται
 - Δίσκοι, CD-ROM:
 - (S)ATA (AT-Attachment), SCSI (Small Computer System Interface)
 - Εξωτερικές συσκευές:
 - USB (Universal Serial Bus), Firewire



Η τοπολογία κάθε περιφερειακού διαύλου μπορεί να είναι τελείως διαφορετική

Χώροι διευθύνσεων για E/E

- Εισαγωγή
- Δίαυλοι
- Συσκευές E/E
- Διασύνδεση E/E

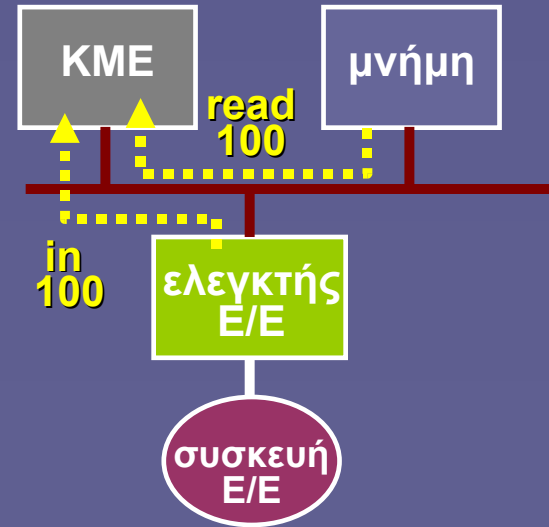
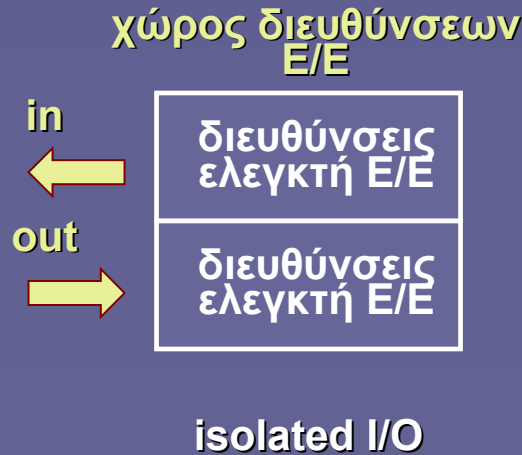
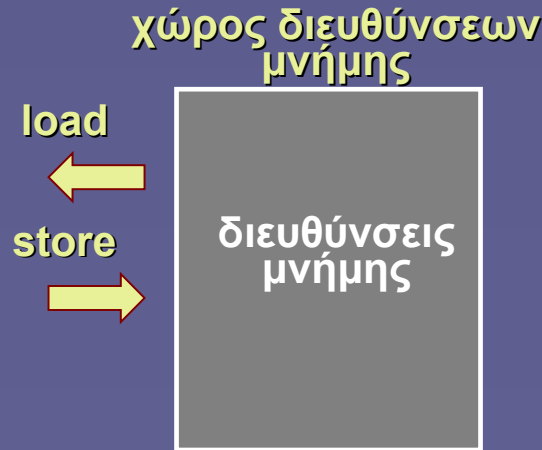


Αντιστοιχισμένη σε μνήμη E/E (memory mapped I/O access)

- Διευθύνσεις E/E στον χώρο μνήμης
- Κοινές εντολές προσπέλασης μνήμης (π.χ. read, write)

Χώροι διευθύνσεων για E/E

- Εισαγωγή
- Δίαυλοι
- Συσκευές E/E
- Διασύνδεση E/E



Απομονωμένη E/E (isolated I/O access)

- Διαφορετικές εντολές για προσπέλαση μνήμης και E/E
- Ίδια διεύθυνση = διαφορετική θέση σε μνήμη και E/E

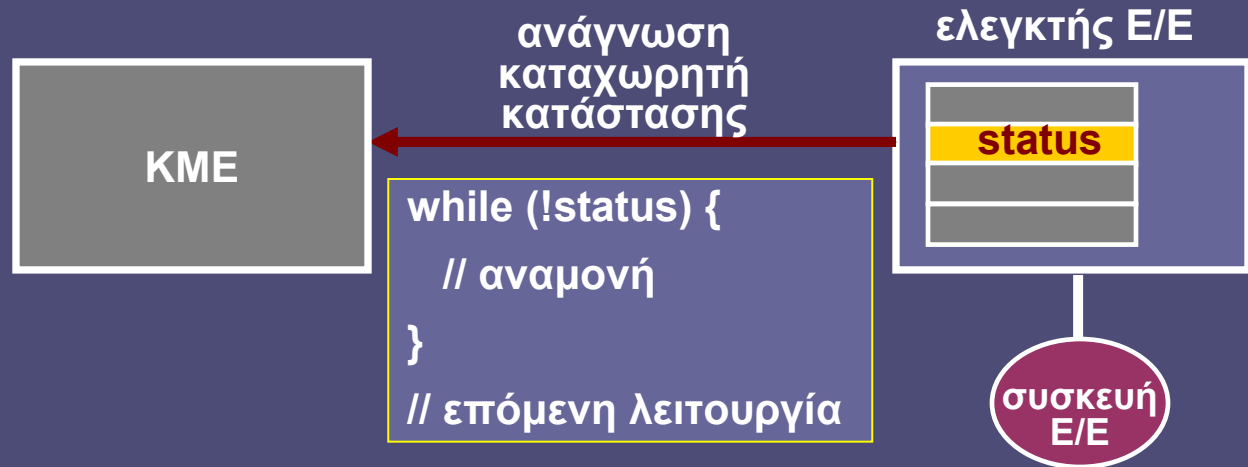
Εξυπηρέτηση αιτήσεων E/E

- Εισαγωγή
- Δίαυλοι
- Συσκευές E/E
- Διασύνδεση E/E
- Αιτήσεις E/E

- Ολοκλήρωση αιτήσεων E/E
 - Υπερβολικά αργή λειτουργία...
 - ...σε σχέση με την ταχύτητα της ΚΜΕ
- Η ΚΜΕ πρέπει να περιμένει!
 - Πώς γίνεται αντιληπτή η ολοκλήρωση της εξυπηρέτησης E/E;
- Μεταφορά δεδομένων από/προς συσκευές E/E
 - Ποιος αναλαμβάνει τη μεταφορά των δεδομένων προς/από την κύρια μνήμη;

Ανίχνευση ολοκλήρωσης αίτησης E/E

- Εισαγωγή
- Δίαυλοι
- Συσκευές E/E
- Διασύνδεση E/E
- Αιτήσεις E/E



- **Περιοδική ανάγνωση καταχωρητή κατάστασης**
 - Ολοκληρώθηκε η τρέχουσα αίτηση;
 - Εμφανίστηκε κάποια αλλαγή κατάστασης στη συσκευή E/E;
 - Επαναληπτική διαδικασία (**polling**)
 - Επιβάρυνση KME με άσκοπους κύκλους εκτέλεσης – για μεγάλο διάστημα η κατάσταση θα παραμένει σταθερή περιμένοντας τη συσκευή E/E

Ανίχνευση ολοκλήρωσης αίτησης E/E

- Εισαγωγή
- Δίαυλοι
- Συσκευές E/E
- Διασύνδεση E/E
- Αιτήσεις E/E

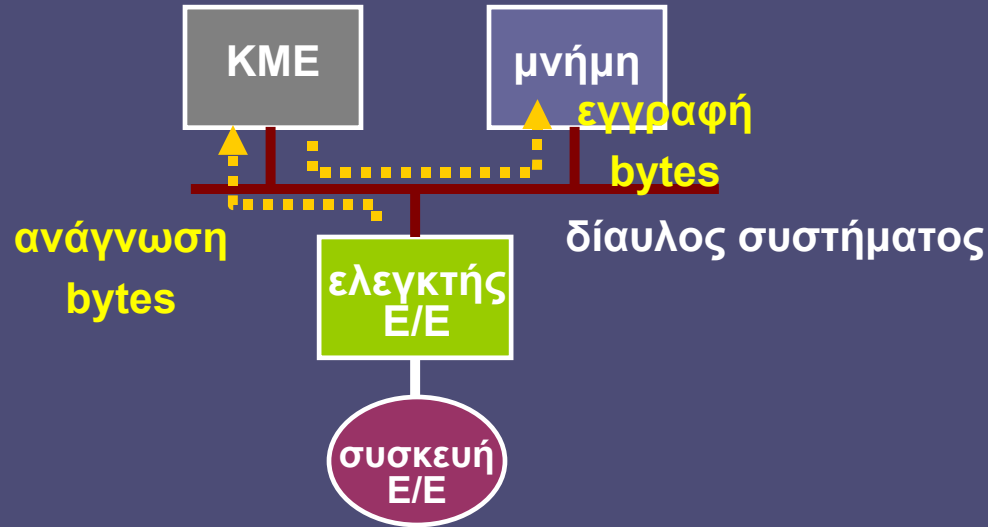


```
JobA() {  
  ...  
  ΑίτησηE/E();  
  <εκτέλεση άλλης  
  διεργασίας>  
  ...  
}  
  
ISR() {  
  <επιστροφή στο  
  jobA>  
}
```

- **Ασύγχρονη ειδοποίηση KME (interrupts)**
 - Ο ελεγκτής ειδοποιεί την KME για αλλαγή κατάστασης της συσκευής E/E
 - Στο μεταξύ η KME μπορεί να εκτελεί άλλη διεργασία
 - Η διακοπή θα προκαλέσει την εκτέλεση ρουτίνας εξυπηρέτησης διακοπής (interrupt service routine – ISR)

Μετακίνηση δεδομένων από/προς μνήμη

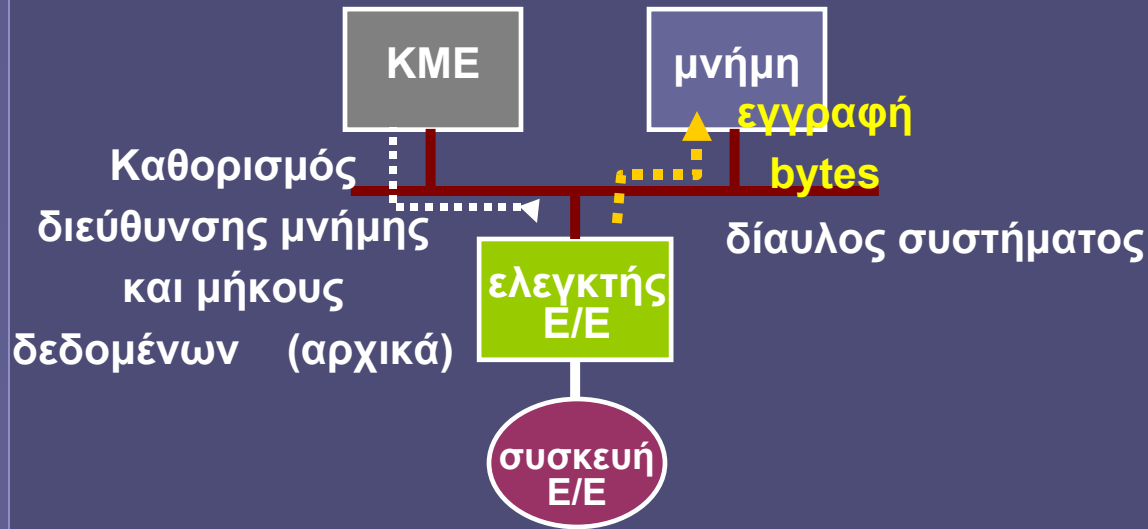
- Εισαγωγή
- Δίαυλοι
- Συσκευές E/E
- Διασύνδεση E/E
- Αιτήσεις E/E



- Μετά από διακοπή ή rolling ελεγκτή E/E
- Η ΚΜΕ *επαναληπτικά* διαβάζει δεδομένα από τον ελεγκτή E/E και τα γράφει στη μνήμη – και το αντίστροφο
- Σημαντική επιβάρυνση της ΚΜΕ!

Direct Memory Access (DMA)

- Εισαγωγή
- Δίαυλοι
- Συσκευές E/E
- Διασύνδεση E/E
- Αιτήσεις E/E



- Η ΚΜΕ απλά θέτει τις παραμέτρους της μεταφοράς
 - Διευθύνσεις πηγής-προορισμού, μέγεθος δεδομένων
- Ο ελεγκτής διεκδικεί τον δίαυλο του συστήματος και μεταφέρει δεδομένα απευθείας προς/από τη μνήμη
 - Η ΚΜΕ μπορεί να εκτελεί άλλη διεργασία
 - Σύγκρουση στη χρήση μνήμης, αλλά ή ΚΜΕ έχει και την κρυφή μνήμη!

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2007-08

Δίκτυα υπολογιστών

(και το Διαδίκτυο)

<http://www.ionio.gr/~mistral/tp/csintro/>

Μ.Στεφανιδάκης

Τι είναι ένα δίκτυο υπολογιστών;

- Εισαγωγή

- Διασύνδεση υπολογιστικών συστημάτων
- Μέσο διασύνδεσης
 - Ενσύρματο ή ασύρματο
- Ανταλλαγή δεδομένων
 - Και μετάδοση εντολών
- Γεφύρωση γεωγραφικών περιοχών
 - Μικρού (τοπικού), μεσαίου και μεγάλου μεγέθους
- Καθορισμένοι κανόνες επικοινωνίας
 - Πρωτόκολλα
 - Ανεξάρτητα από το είδος (αρχιτεκτονική) του υπολογιστή

Δικτυακά πρωτόκολλα

- Εισαγωγή
- Πρωτόκολλα

- **Κανόνες επικοινωνίας**
 - για την ανταλλαγή μηνυμάτων δεδομένων μεταξύ δύο (δια)δικτυακών εφαρμογών
- **Ένα δικτυακό πρωτόκολλο καθορίζει:**
 - Το μορφότυπο (format) των ανταλλασσόμενων μηνυμάτων
 - Τη σειρά (αλληλουχία) των μηνυμάτων
 - Τις απαιτούμενες ενέργειες κατά την αποστολή ή παραλαβή των μηνυμάτων
- **Υλοποίηση πρωτοκόλλων**
 - Σε υλικό ή/και λογισμικό
- **Το Διαδίκτυο**
 - στηρίζεται σε μεγάλο βαθμό σε πρωτόκολλα!

Αρχιτεκτονική επιπέδων πρωτοκόλλων

- Εισαγωγή
- Πρωτόκολλα



παρέχει υπηρεσίες προς το
ανώτερο επίπεδο

χρησιμοποιεί τις υπηρεσίες των
κατώτερων επιπέδων

- **Δομημένη αρχιτεκτονική**
 - Διευκολύνει τον σχεδιασμό, μειώνει πολυπλοκότητα
 - Επιτρέπει την απρόσκοπτη εξέλιξη των πρωτοκόλλων
 - Επιτρέπει τη χρήση εναλλακτικών τεχνολογιών

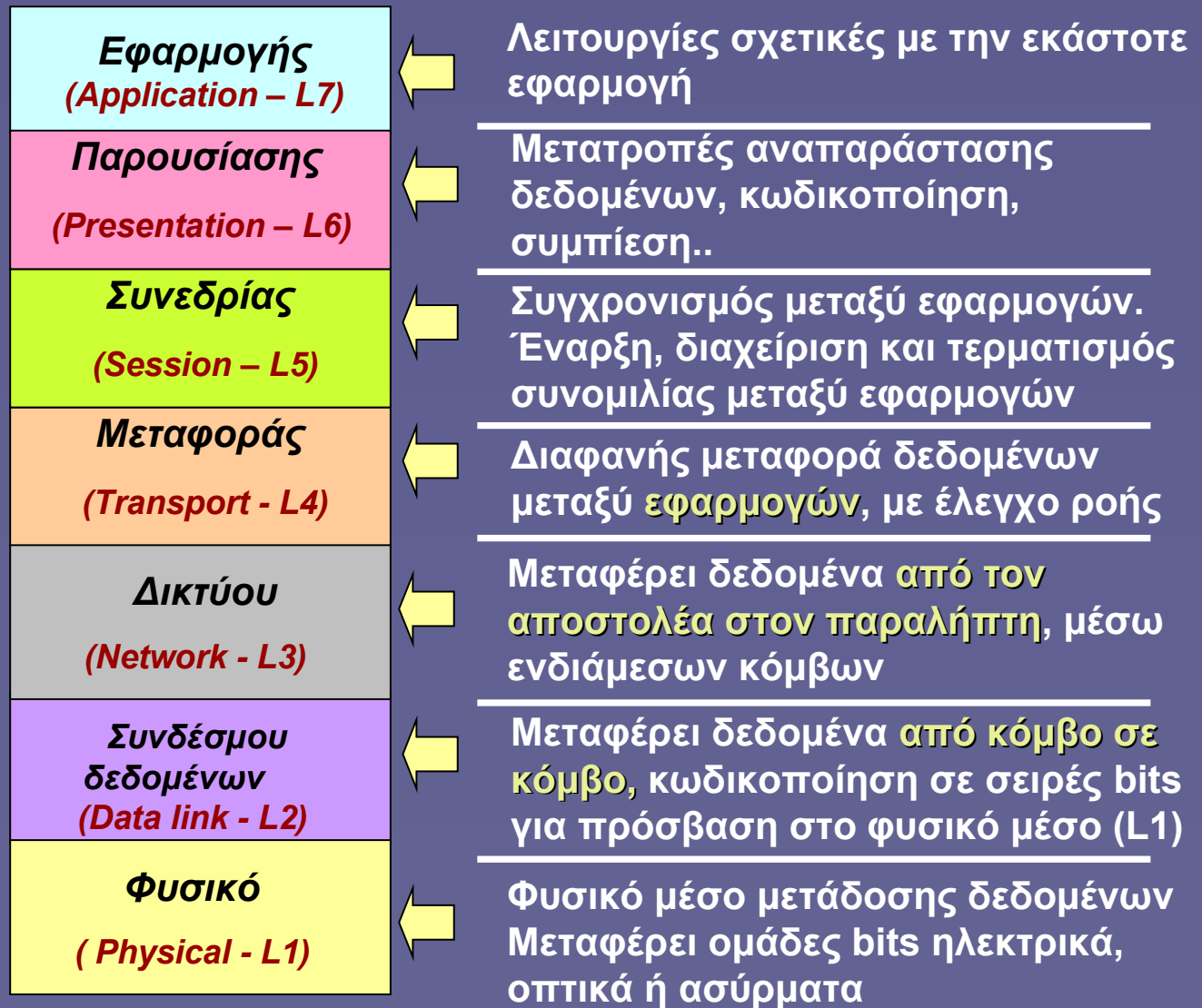
Το μοντέλο OSI

- Εισαγωγή
- Πρωτόκολλα

- **Open System Interconnection**
- **Θεωρητικό μοντέλο**
 - Παρέχει το πλαίσιο υλοποίησης των **πρωτοκόλλων** κάθε δικτύου
 - Κάποια δίκτυα δεν το υλοποιούν πιστά
- **Μετάδοση δεδομένων**
 - Μέσω των 7 επιπέδων του μοντέλου
- **Εφαρμογές**
 - Επίπεδα 5 έως 7

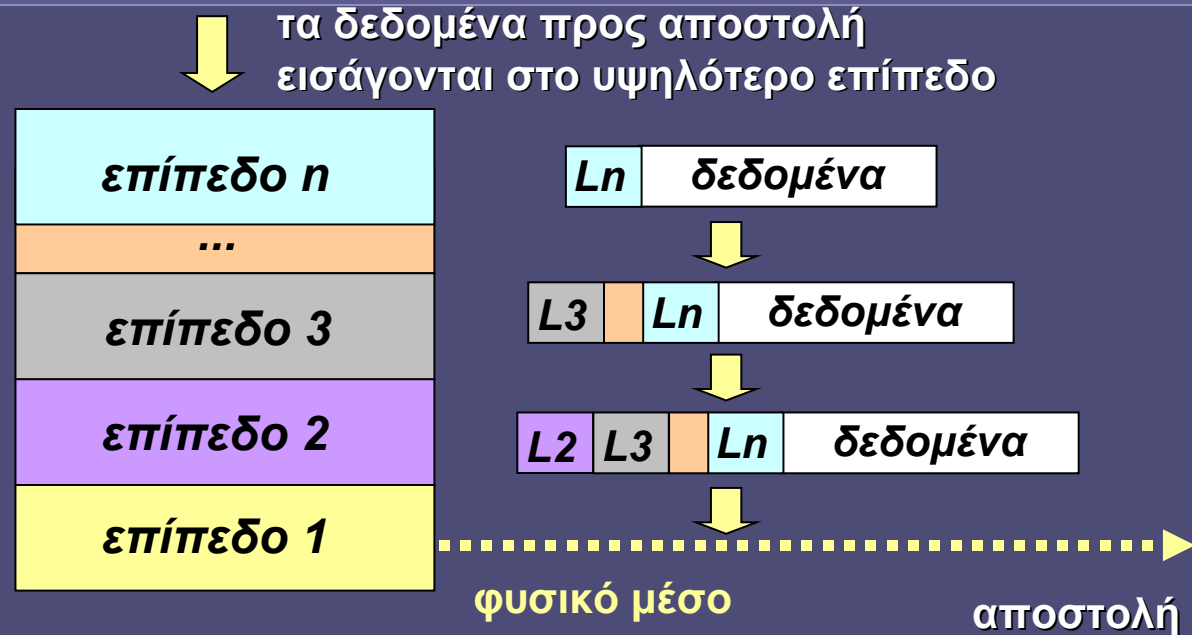
Τα επίπεδα του μοντέλου OSI

- Εισαγωγή
- Πρωτόκολλα



Διάσχιση επιπέδων – αποστολή

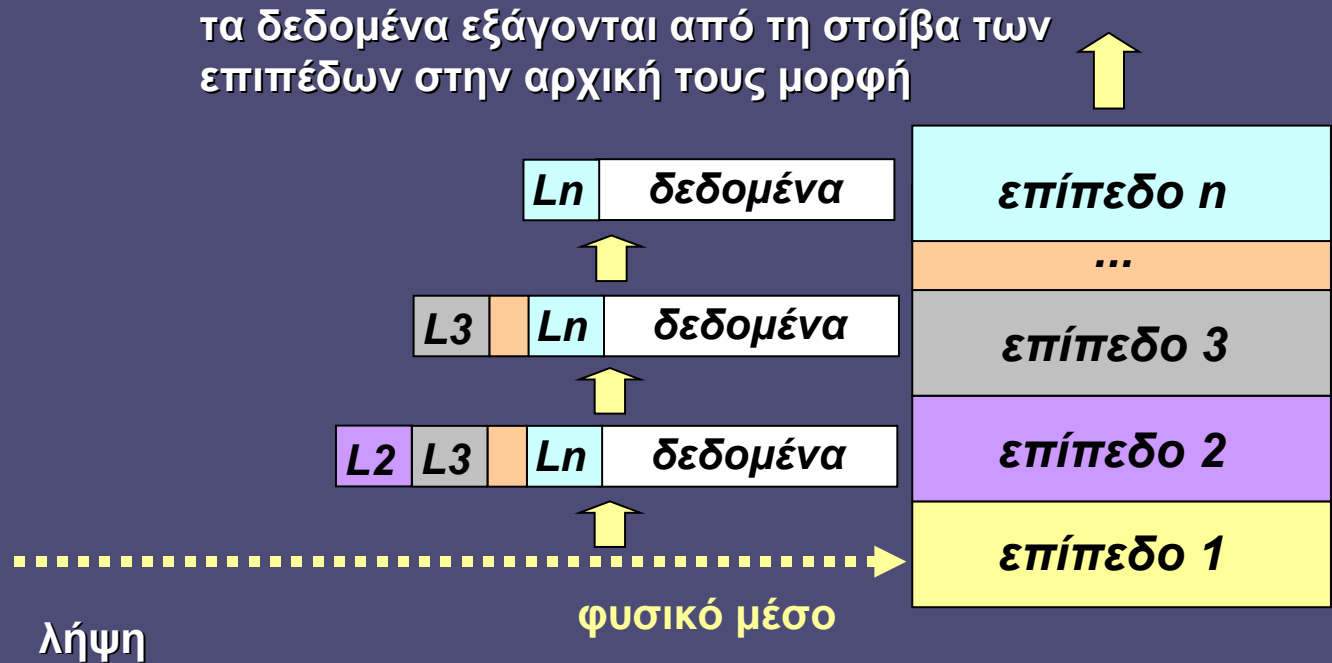
- Εισαγωγή
- Πρωτόκολλα



- Προσθήκη επικεφαλίδων (headers)
 - Σε κάθε επίπεδο – πρόσθετη πληροφορία
 - Η πληροφορία του προηγούμενου επιπέδου “ενθυλακώνεται” στο πακέτο του επόμενου (encapsulation)

Διάσχιση επιπέδων – λήψη

- Εισαγωγή
- Πρωτόκολλα



- Ανασυναρμολόγηση αρχικής πληροφορίας
 - Κατά τη διάσχιση των επιπέδων πρωτοκόλλων στη λήψη
 - από το χαμηλότερο προς το υψηλότερο επίπεδο

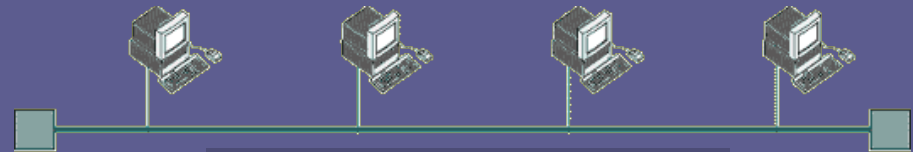
Κατηγορίες δικτύων

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων

- **Τοπικά Δίκτυα**
 - Local Area Networks (LANs)
- **Μητροπολιτικά Δίκτυα**
 - Metropolitan Area Networks (MANs)
- **Δίκτυα Ευρείας Περιοχής**
 - Wide Area Networks (WANs)

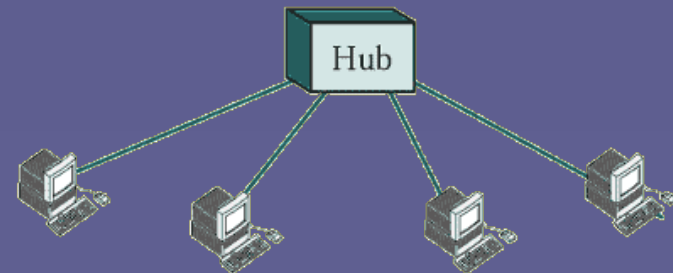
Τοπικά Δίκτυα (LANs)

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων



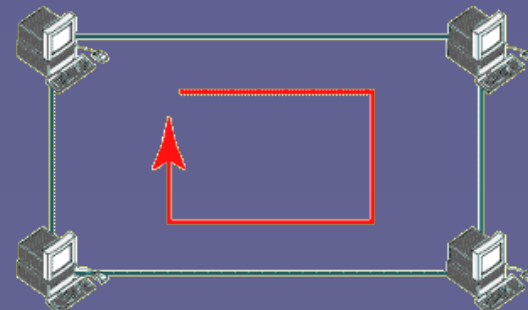
Τοπολογία Διαύλου

- Διασύνδεση υπολογιστών στον ίδιο χώρο
 - γραφείο, κτήριο..



Τοπολογία Αστέρα

- Η τοπολογία αστέρα έχει επικρατήσει



Τοπολογία Δακτυλίου

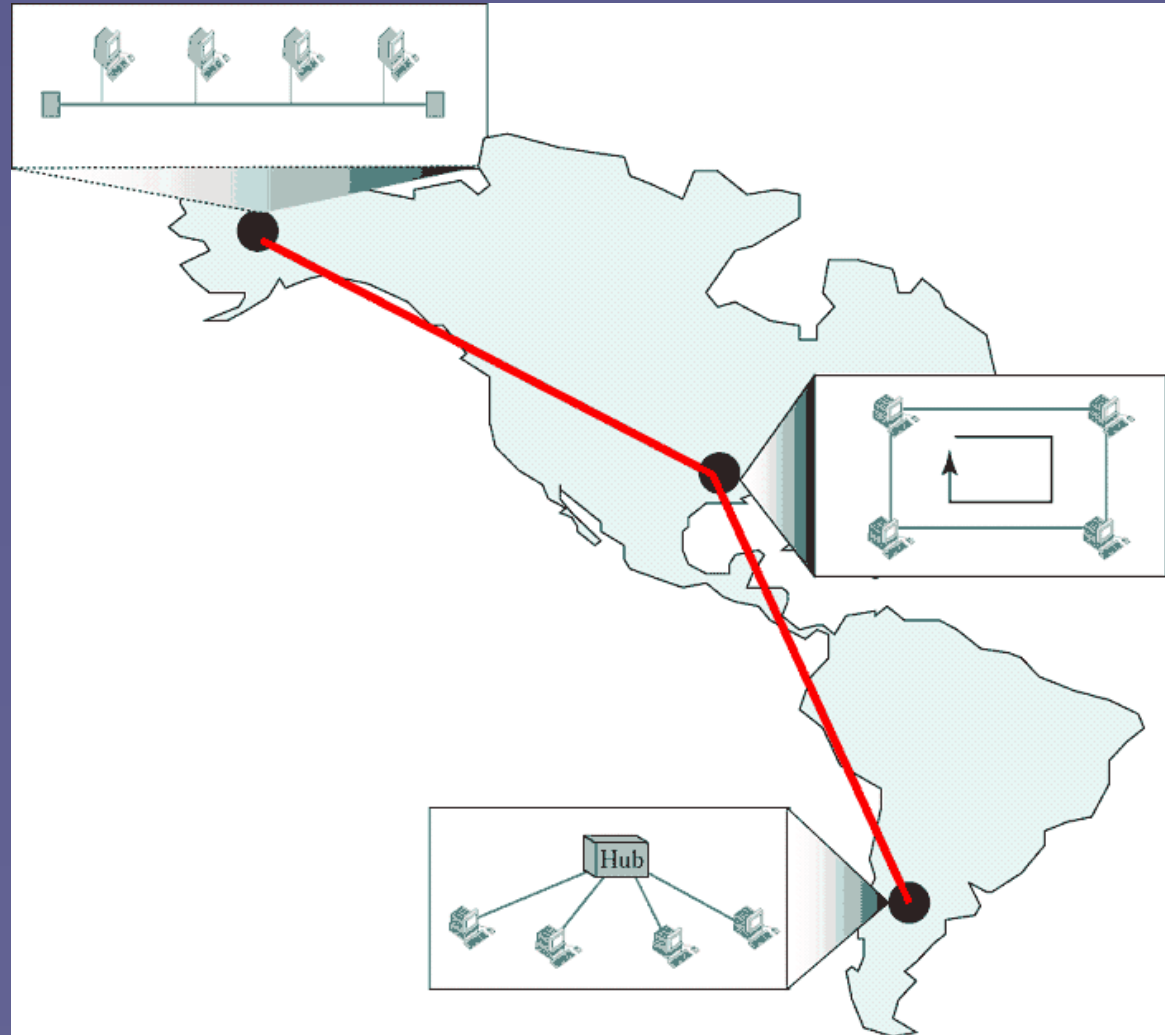
Μητροπολιτικά Δίκτυα (MANs)

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων



Δίκτυα Ευρείας Περιοχής (WANs)

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων



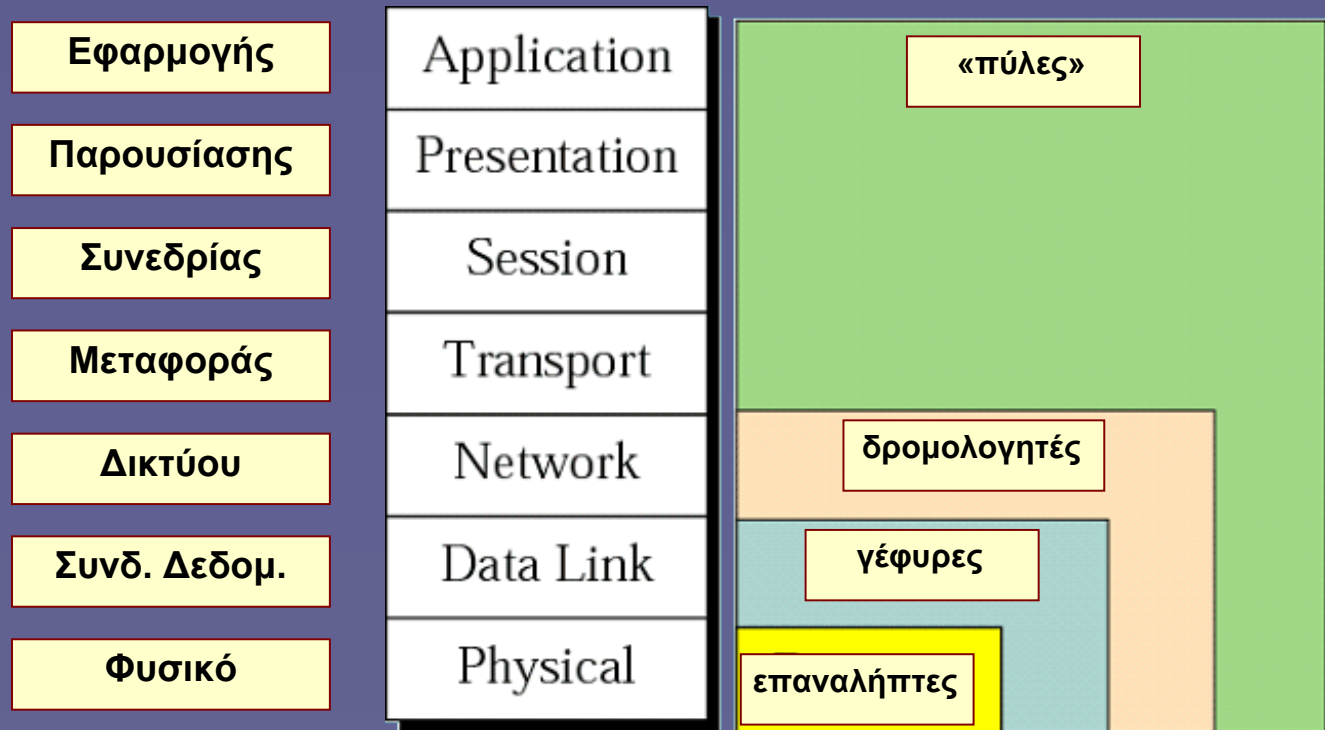
Συσκευές Διασύνδεσης

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων
- Συσκευές

- **Συσκευές δικτύωσης**
 - Επαναλήπτες (repeaters)
 - Γέφυρες (bridges)
- **Συσκευές διαδικτύωσης**
 - Δρομολογητές (routers)
 - “Πύλες” (gates)

Συσκευές διασύνδεσης και το μοντέλο OSI

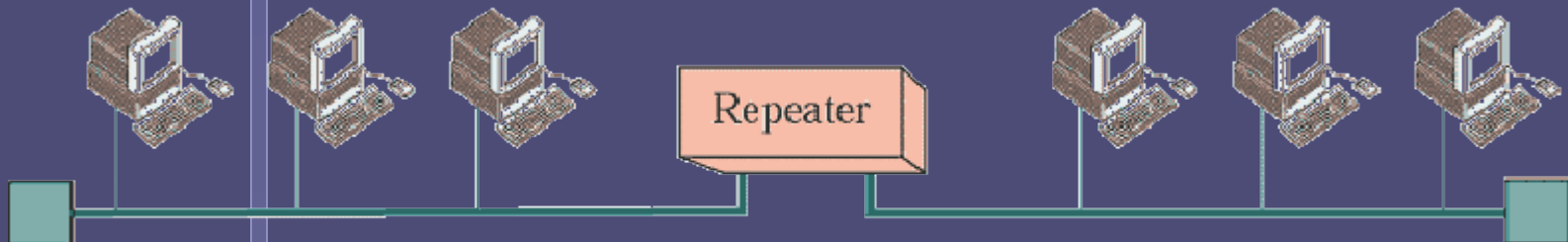
- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων
- Συσκευές



Επαναλήπτες (repeaters)

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων
- Συσκευές

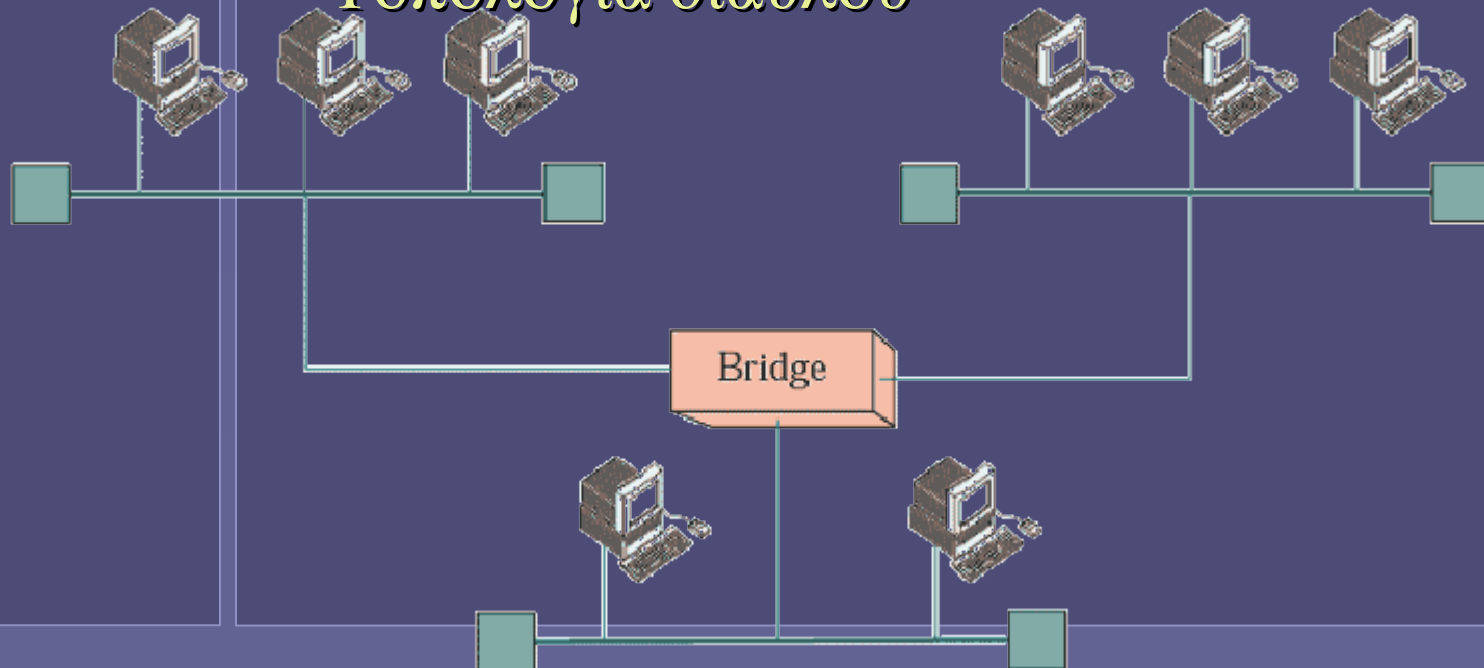
- Στο 1ο επίπεδο OSI
 - Ηλεκτρική αναδημιουργία σήματος
- Τοπολογία διαύλου



Επαναλήπτες (repeaters)

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων
- Συσκευές

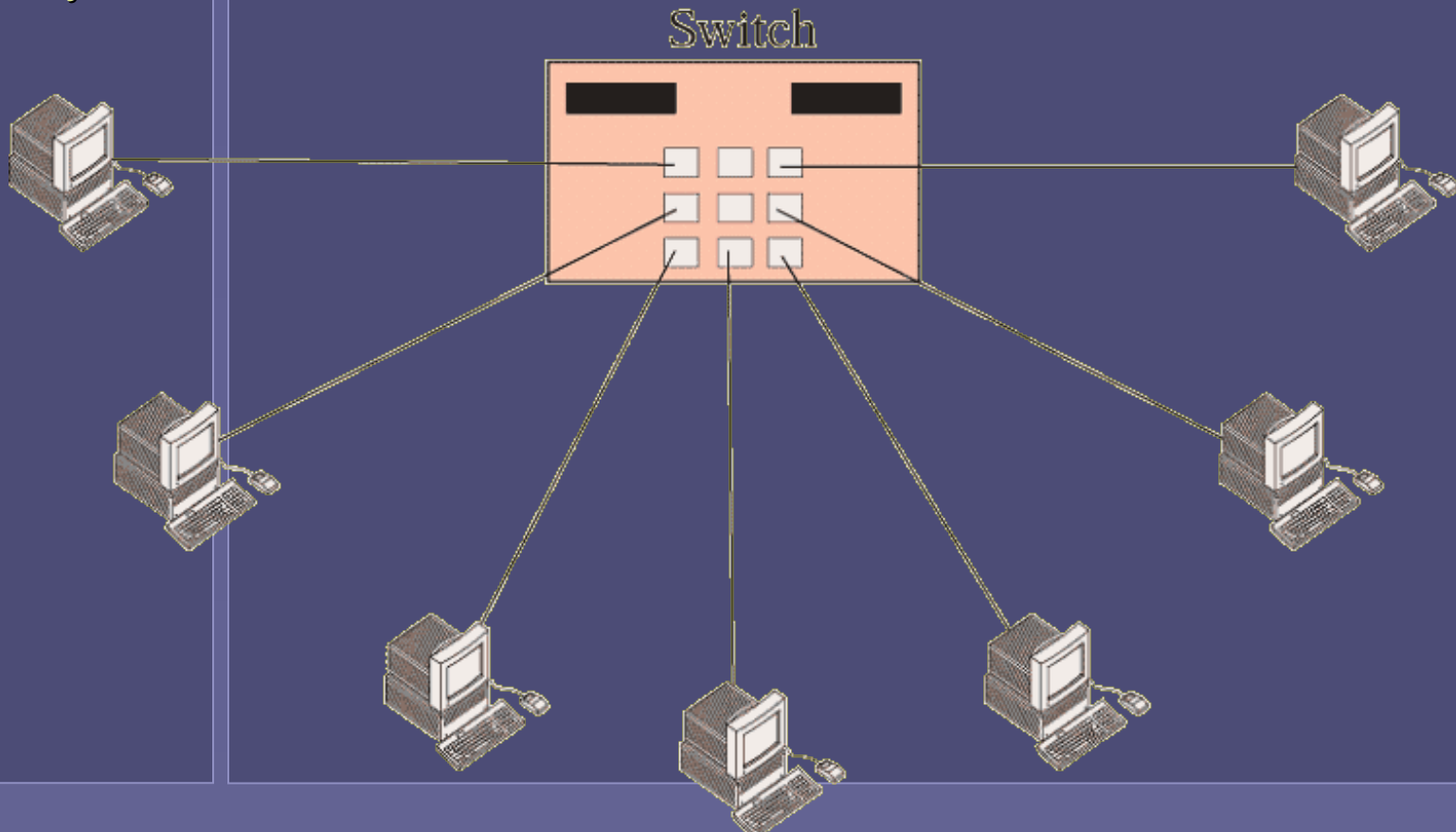
- Στο 1ο και 2ο επίπεδο OSI
 - Ρύθμιση κυκλοφορίας ανάλογα με φυσική διεύθυνση προορισμού
 - Ηλεκτρική αναδημιουργία σήματος
- Τοπολογία διαύλου



Μεταγωγείς (switches)

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων
- Συσκευές

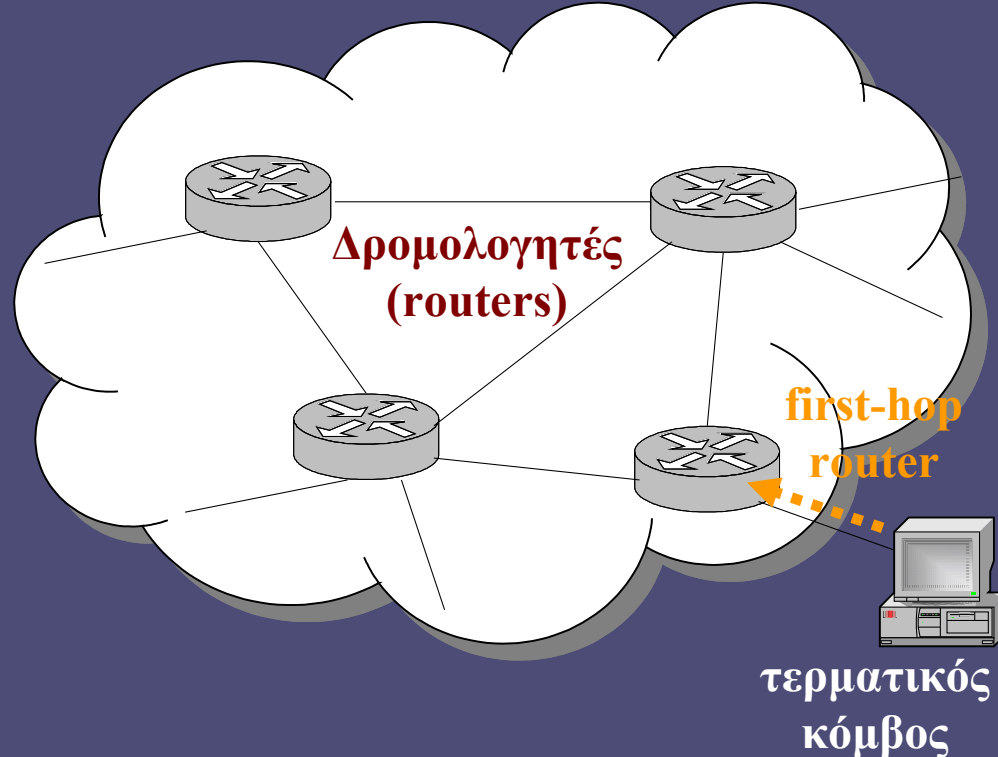
- Έξυπνη παραλλαγή της γέφυρας
 - Τοπολογία αστέρα



Δρομολογητές (routers)

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων
- Συσκευές

- **Διαδικτύωση**
 - Στα επίπεδα 1, 2 και 3 του OSI
 - Δρομολόγηση μέσω **λογικής** διεύθυνσης



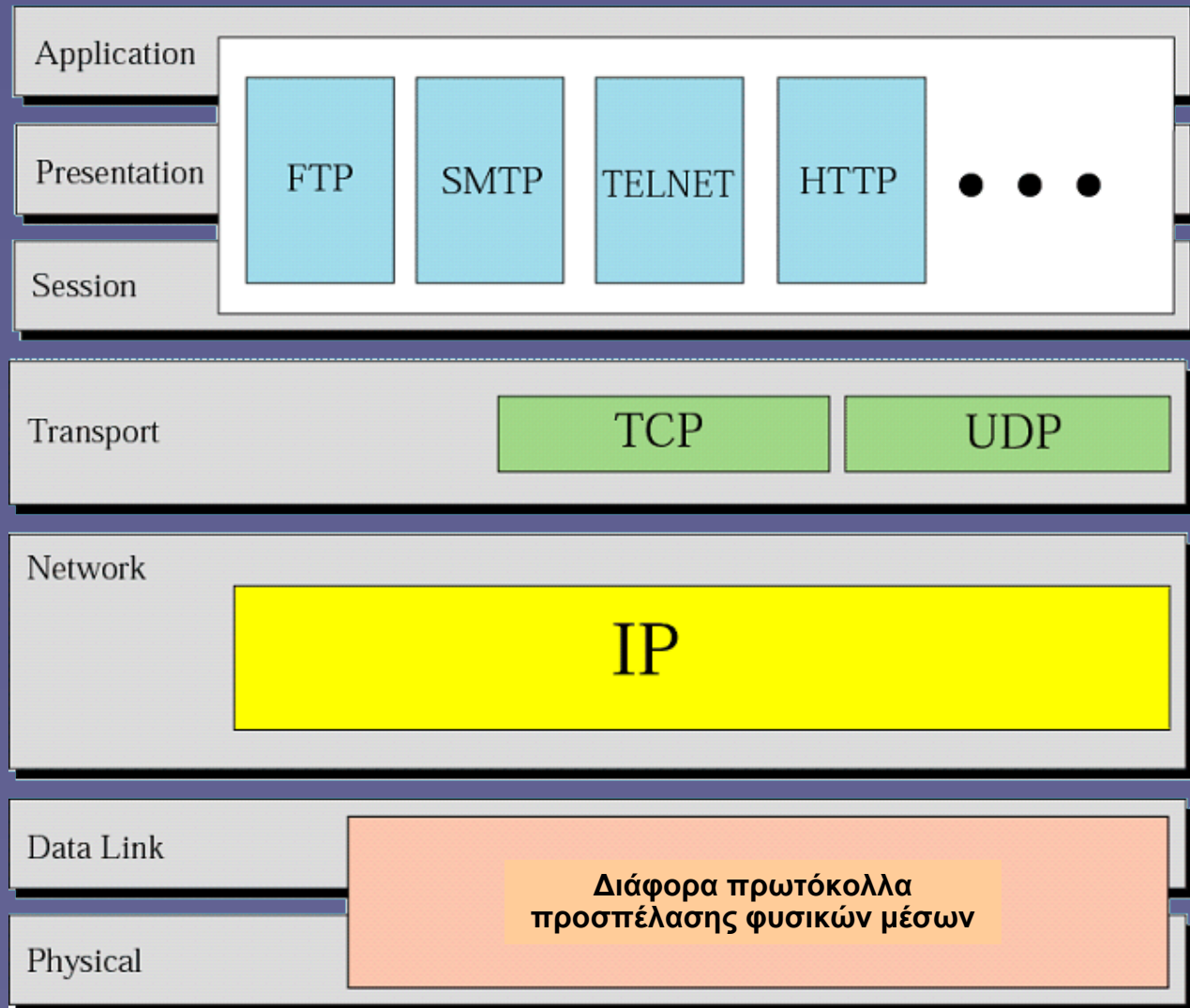
Το Διαδίκτυο

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων
- Συσκευές
- Το Διαδίκτυο

- **Διασύνδεση εκατομμυρίων υπολογιστών**
 - Μόνιμη ή προσωρινή
 - Μέσω διαφόρων φυσικών μέσων
- **Πρωτόκολλα TCP/IP**
 - Επίπεδα OSI: **Δικτύου** (3) και **Μεταφοράς** (4)
 - Πληροφορία χωρίζεται σε πολλαπλά πακέτα
 - Δρομολόγηση μέσω πολλαπλών κόμβων
- **Ethernet**
 - Κύριο φυσικό μέσο σε τοπικά δίκτυα
 - 10Mbps, 100Mbps, 1000Mbps ...
 - Επίπεδα OSI **Φυσικό** (1) και **Συνδέσμου Δεδομένων** (2)

Το Διαδίκτυο και το μοντέλο OSI

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων
- Συσκευές
- Το Διαδίκτυο



Internet Protocol (IP)

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων
- Συσκευές
- Το Διαδίκτυο

- Στο επίπεδο 3 (Δικτύου) του OSI
- Αυτοδύναμα πακέτα δεδομένων
 - IP datagrams
- Δρομολόγηση μέσω διαφορετικών κόμβων
- Διεύθυνση IP
 - 4 bytes
 - Συμβολισμός: A.B.C.D
 - Π.χ. 129.10.7.30
 - Πρέπει να είναι μοναδική στο Διαδίκτυο
 - Ανάθεση από διεθνείς οργανισμούς
 - Δυναμική ή στατική ανάθεση

Transmission Control Protocol (TCP)

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων
- Συσκευές
- Το Διαδίκτυο

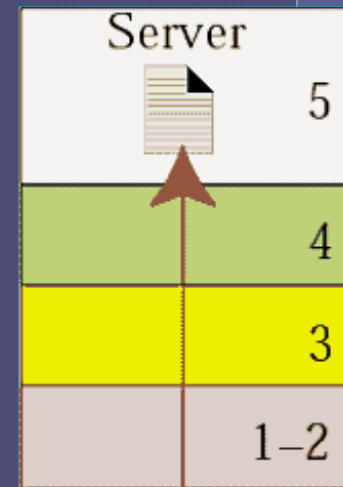
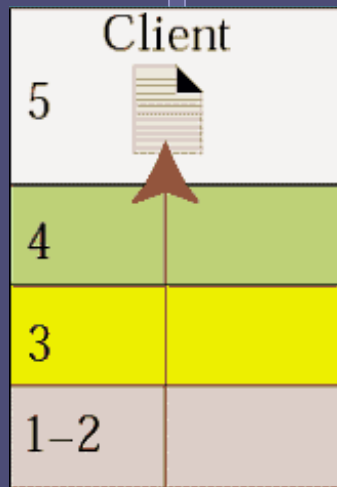
- Στο επίπεδο 4 (Μεταφοράς) του OSI
- Αξιόπιστη μεταφορά
 - Καταμερισμός πληροφορίας σε IP πακέτα
 - Αρίθμηση πακέτων
 - Έλεγχος μετάδοσης-παραλαβής πακέτων
 - Ταξινόμηση λαμβανόμενων πακέτων
 - Παρακολούθηση χρόνων άφιξης πακέτων
 - Αίτηση αναμετάδοσης σε περίπτωση απώλειας
 - Συναρμολόγηση τελικής πληροφορίας

Το μοντέλο επικοινωνίας client-server

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων
- Συσκευές
- Το Διαδίκτυο



- Π.χ. **client** είναι ο προσωπικός μας υπολογιστής και **server** ένα απομακρυσμένο σύστημα που παρέχει διάφορες υπηρεσίες (όπως το e-mail)



Διακίνηση ηλεκτρονικού ταχυδρομείου

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων
- Συσκευές
- Το Διαδίκτυο



Πρόγραμμα διαχείρισης e-mail

SMTP Client



POP Client



Διακομιστής e-mail



POP Server



SMTP Server



Διευθύνσεις e-mail:
user@domain
(π.χ. nikos@ionio.gr)

Διαδίκτυο

- Post Office Protocol (POP)
 - για παραλαβή e-mail
- Simple Mail Transfer Protocol (SMTP)
 - για αποστολή/διακίνηση e-mail

HyperText Transfer Protocol (HTTP)

- Εισαγωγή
- Πρωτόκολλα
- Είδη Δικτύων
- Συσκευές
- Το Διαδίκτυο

- Για την προσπέλαση και μεταφορά εγγράφων web μεταξύ client-server
 - μεταφορά κειμένου, εικόνας, ήχου, βίντεο κλπ
 - Τα επιστρεφόμενα αντικείμενα
 - ή είναι **στατικά** αποθηκευμένα στον server
 - ή δημιουργούνται **δυναμικά** με την εκτέλεση κάποιου προγράμματος (στον server)
 - Αίτηση (client) \Rightarrow Απόκριση (server)
 - URL (Uniform Resource Locator)
 - Πού βρίσκεται το αντικείμενο που θέλουμε;
 - **πρωτόκολλο** `:// server [:port] / μονοπάτι`
 - `http://www.ionio.gr/cs/index.html`

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2007-08

Εφαρμογές στο Διαδίκτυο

<http://www.ionio.gr/~mistral/tp/csintro/>

Μ.Στεφανιδάκης

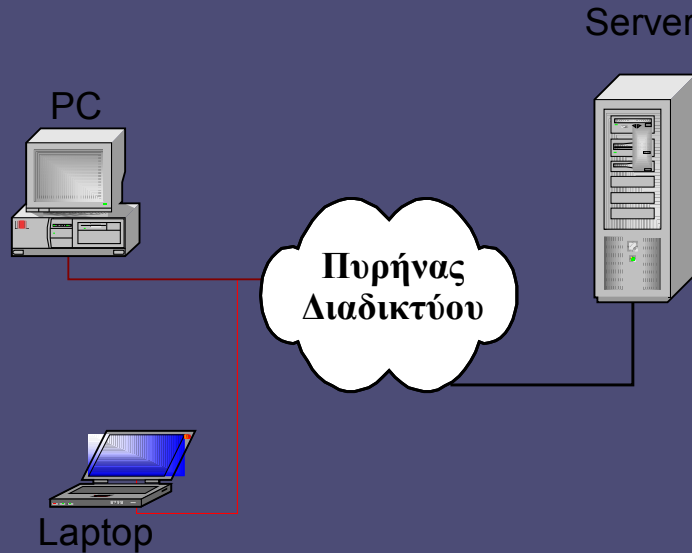
Το Διαδίκτυο

- Το Διαδίκτυο

- **Διασύνδεση εκατομμυρίων υπολογιστών**
 - Μόνιμη ή προσωρινή
 - Μέσω διαφόρων φυσικών μέσων
- **Πρωτόκολλα TCP/IP**
 - Επίπεδο **Δικτύου** (L3)
 - Επίπεδο **Μεταφοράς** (L4)
 - Πληροφορία χωρίζεται σε αυτοδύναμα πακέτα
 - Δρομολόγηση μέσω πολλαπλών κόμβων

Στις παρυφές του διαδικτύου

- Το Διαδίκτυο

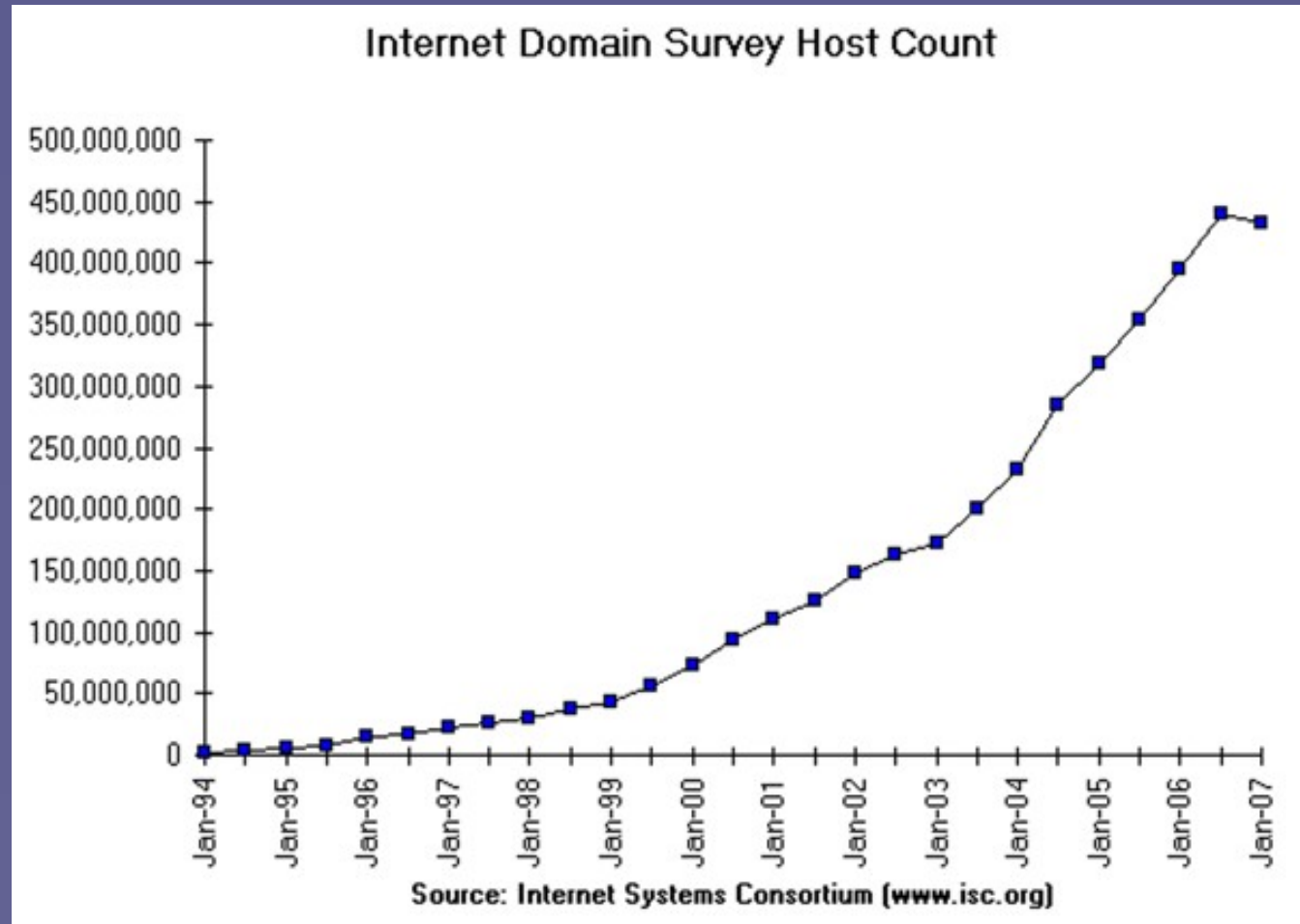


Τερματικοί κόμβοι (endnodes ή hosts)

- Στα άκρα του Διαδικτύου
- Διασύνδεση διαφόρων συσκευών
- Όχι μόνο κλασσικοί υπολογιστές
 - PCs, servers, ενσωματωμένες συσκευές

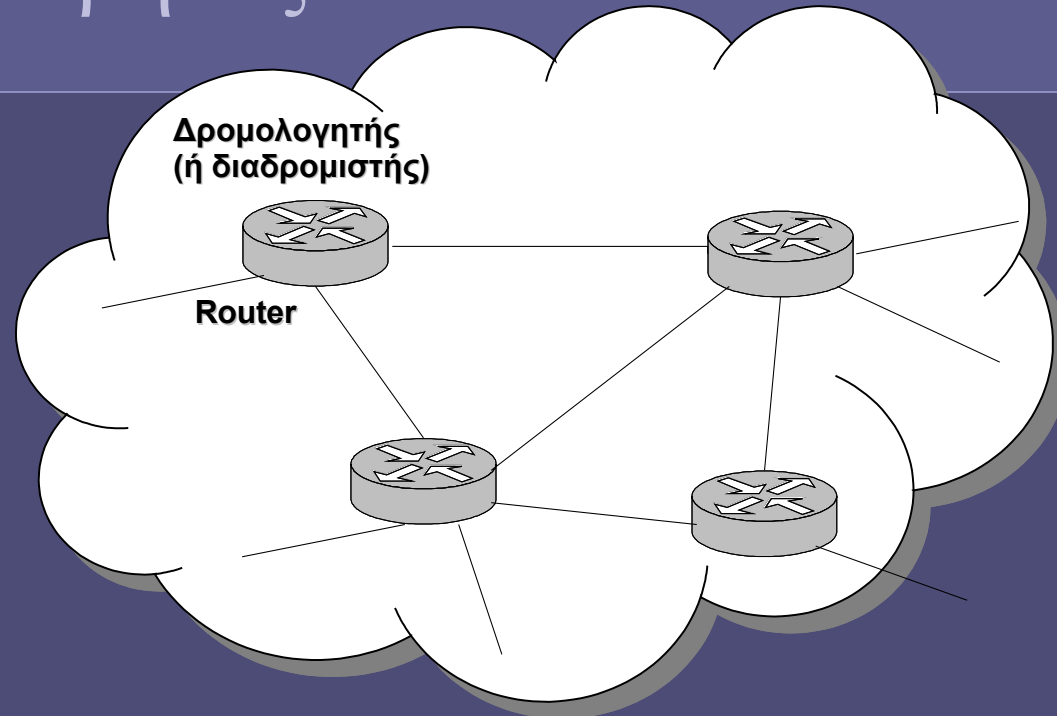
Ο αριθμός των τερματικών κόμβων σήμερα

- Το Διαδίκτυο



Ο πυρήνας του Διαδικτύου

- Το Διαδίκτυο



- Ιεραρχική διασύνδεση δικτύων
 - Όχι υποχρεωτικά με δενδρική δομή
 - Σε επίπεδο εταιρίας, χώρας, και παγκοσμίως
- Επιλογή επόμενου κόμβου και προώθηση πακέτων

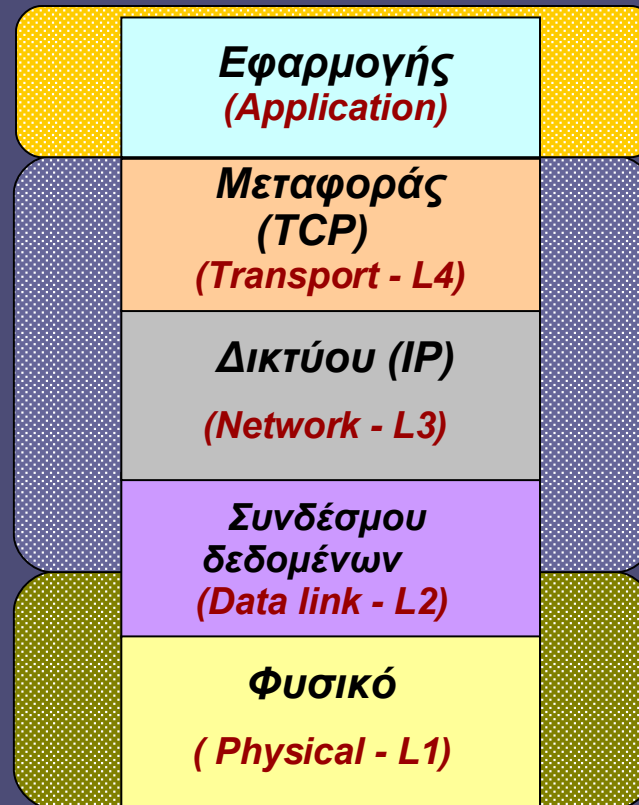
Διαδίκτυο και επίπεδα πρωτοκόλλων

- Το Διαδίκτυο

διαδικτυακή
εφαρμογή

λειτουργικό
σύστημα

υλικό



- Το πλήρες σύνολο της δικτυακής στοίβας (netstack) υλοποιείται στους τερματικούς κόμβους

Internet Protocol (IP)

- Το Διαδίκτυο

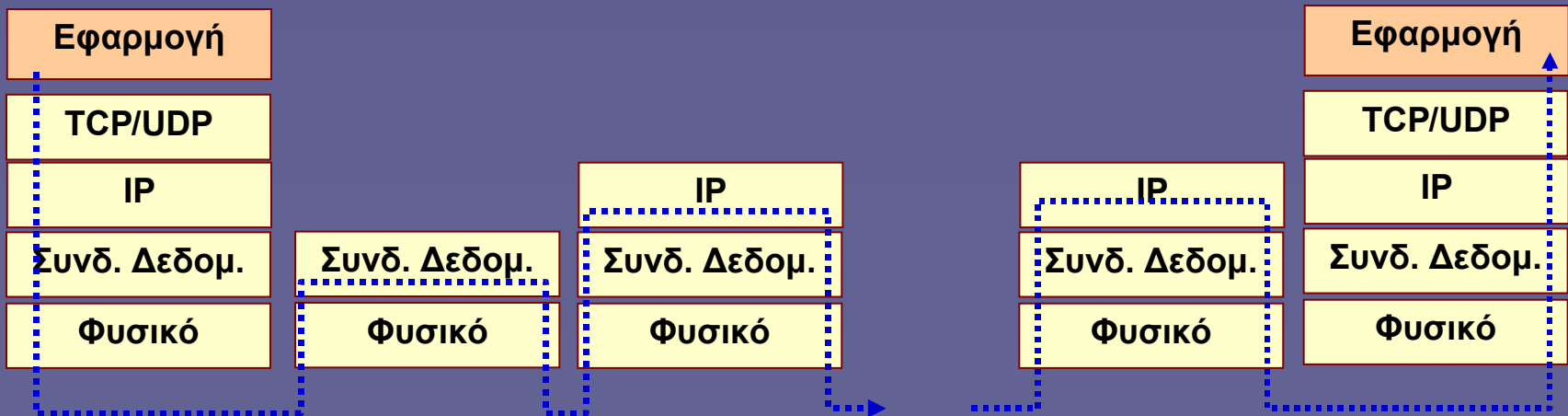
- Στο επίπεδο 3 (Δικτύου) του OSI
- Αυτοδύναμα πακέτα δεδομένων
 - IP datagrams
- Δρομολόγηση μέσω διαφορετικών κόμβων
- Διεύθυνση IP
 - 4 bytes
 - Συμβολισμός: A.B.C.D
 - Π.χ. 129.10.7.30
 - Πρέπει να είναι μοναδική στο Διαδίκτυο
 - Ανάθεση από διεθνείς οργανισμούς
 - Δυναμική ή στατική ανάθεση

Transmission Control Protocol (TCP)

- Το Διαδίκτυο

- Στο επίπεδο 4 (Μεταφοράς) του OSI
- Αξιόπιστη μεταφορά
 - Καταμερισμός πληροφορίας σε IP πακέτα
 - Αρίθμηση πακέτων
 - Έλεγχος μετάδοσης-παραλαβής πακέτων
 - Ταξινόμηση λαμβανόμενων πακέτων
 - Παρακολούθηση χρόνων άφιξης πακέτων
 - Αίτηση αναμετάδοσης σε περίπτωση απώλειας
 - Συναρμολόγηση τελικής πληροφορίας

Διάσχιση επιπέδων Διαδικτύου



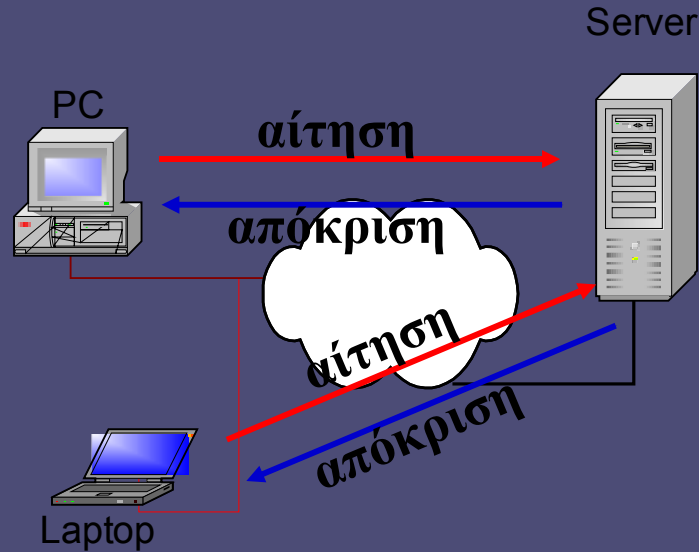
Οι διαδικτυακές εφαρμογές

- Το Διαδίκτυο
- Εφαρμογές

- **Κατανεμημένες εφαρμογές**
 - Σε διασυνδεδεμένους υπολογιστές
 - Επικοινωνία μέσω Διαδικτύου
- Για την παροχή υπηρεσιών (services) προς
 - Τον άνθρωπο
 - Web, e-mail κλπ
 - Άλλους υπολογιστές
 - Web services
- Εκμετάλλευση κατανεμημένων πόρων
 - Υπολογιστικοί πόροι
 - Πόροι μνήμης και δίσκων
 - Αποθηκευμένης πληροφορίας (digital libraries)

Το μοντέλο client-server

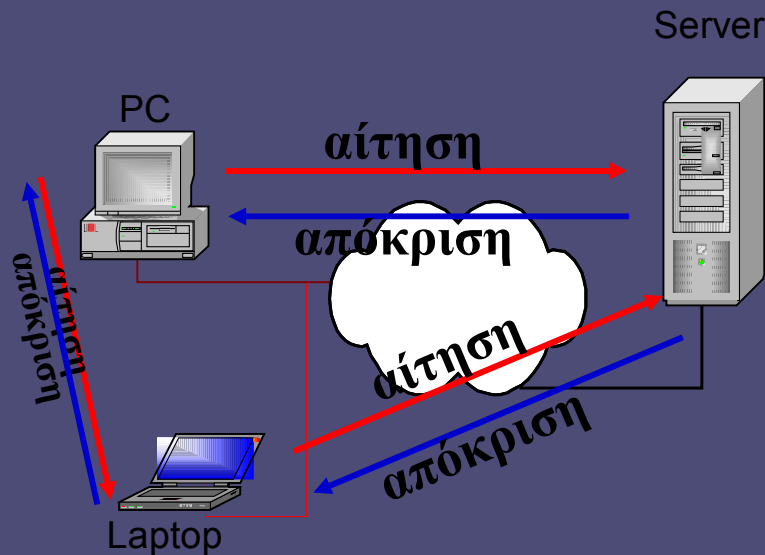
- Το Διαδίκτυο
- Εφαρμογές



- Το παραδοσιακό μοντέλο πελάτη-εξυπηρετητή
 - Οι κλασικές εφαρμογές του Διαδικτύου
 - Web, e-mail, μεταφορές αρχείων, απομακρυσμένη πρόσβαση, κλπ
 - Ο πελάτης (client) στέλνει μια αίτηση (request) για εξυπηρέτηση στον εξυπηρετητή (server). Ο τελευταίος εκτελεί τη ζητούμενη λειτουργία και στέλνει μια απόκριση (response) δεδομένων.

Εναλλακτικό μοντέλο P2P

- Το Διαδίκτυο
- Εφαρμογές



- Το νεώτερο μοντέλο peer-to-peer (P2P)
 - Όλοι οι κόμβοι μπορούν να δράσουν ως clients ή/και ως servers
 - Ανάλογα με τις δυνατότητές τους

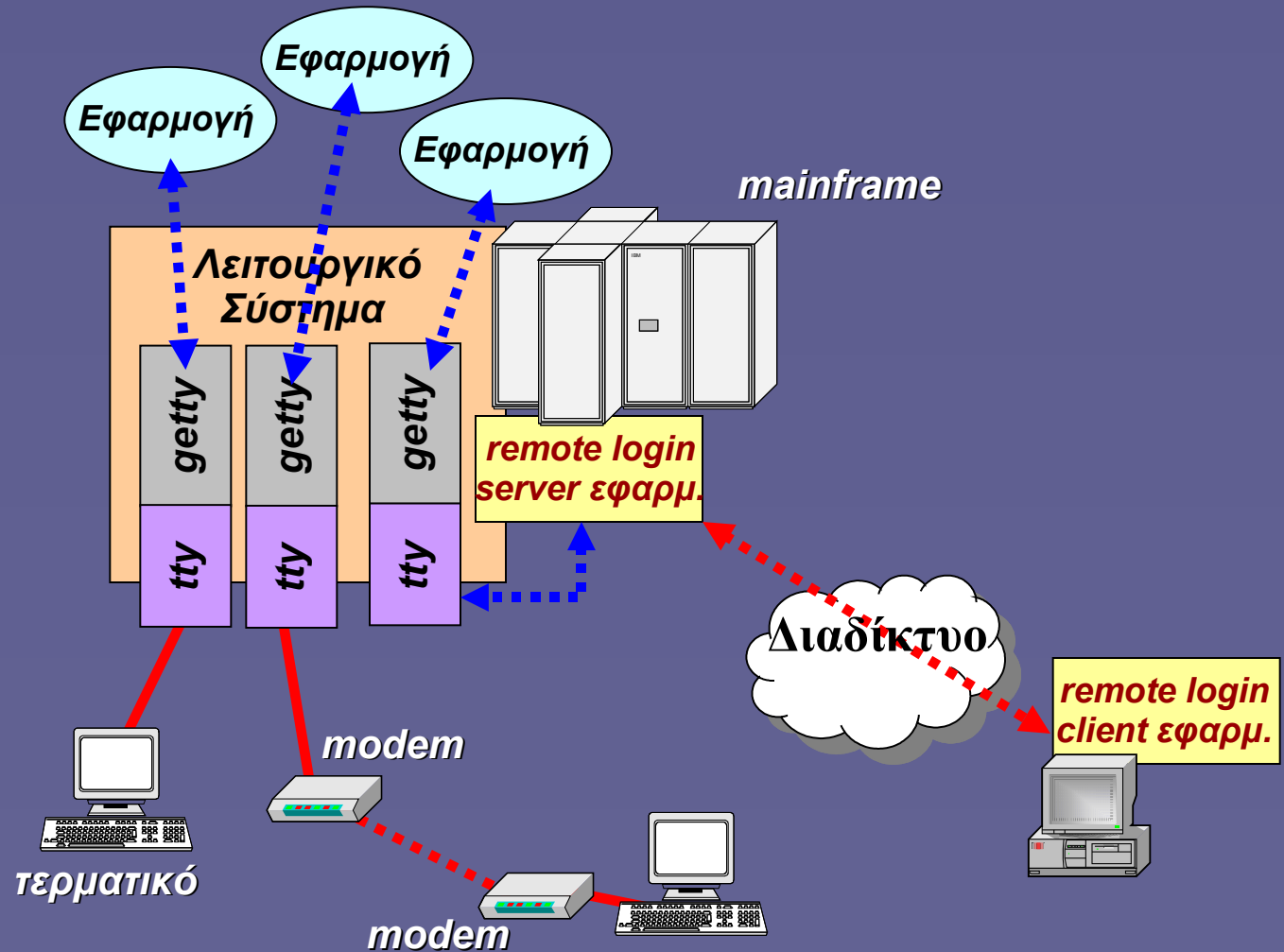
Κατηγορίες εφαρμογών client-server

- Το Διαδίκτυο
- Εφαρμογές

- **Γενικός στόχος client-server εφαρμογών: η προσπέλαση απομακρυσμένων πόρων**
 - Μεταφορά αντικειμένων (αρχείων)
 - Εκτέλεση λειτουργιών
- **Οι πρώτες (δια)δικτυακές εφαρμογές**
 - Απομακρυσμένη πρόσβαση χρήστη (remote login) σε υπολογιστικά συστήματα..
 - ..και άμεση εκτέλεση εντολών
- **Η νεώτερη κατηγορία εφαρμογών**
 - Αίτηση και λήψη αντικειμένων (αρχείων)
 - Αίτηση εκτέλεσης λειτουργιών
 - και λήψη αποτελεσμάτων

Απομακρυσμένη πρόσβαση

- Το Διαδίκτυο
- Εφαρμογές



Εφαρμογές για αίτηση και απόκτηση πόρων

- Το Διαδίκτυο
- Εφαρμογές

- **Διαφορετική κατηγορία client-server εφαρμογών**
 - Αίτηση και λήψη αντικειμένων (αρχείων)
 - Αίτηση εκτέλεσης λειτουργιών
 - και λήψη αποτελεσμάτων
 - Χωρίς να συνδεθεί ο χρήστης σε ένα απομακρυσμένο “εικονικό τερματικό”
 - Δεν είναι απαραίτητο πάντοτε ο χρήστης να έχει λογαριασμό στο σύστημα server
- **Client εφαρμογή**
 - Στέλνει αιτήσεις (εντολές) – συνήθως σε 7-bit ASCII
 - Παρέχει πληροφορία του χρήστη προς τον server
- **Server εφαρμογή**
 - Απαντά σε αιτήσεις επιστρέφοντας δεδομένα
 - Παρέχει πληροφορία ολοκλήρωσης της αίτησης

Ο παγκόσμιος ιστός (Web)

- Το Διαδίκτυο
- Εφαρμογές
- Web

- **World Wide Web (www)**
 - Διακίνηση ιστοσελίδων
 - Και πολλών άλλων μορφών δεδομένων
 - Η εφαρμογή που ανέδειξε το Διαδίκτυο
 - Σήμερα βάση για πολλές άλλες εφαρμογές (εφαρμογές web)
 - Τα βασικά τμήματα (εφαρμογές, γλώσσες και πρωτόκολλα) που συγκροτούν το Web σχεδιάστηκαν στις αρχές του '90
- Πριν την εμφάνιση του Web
 - E-mail
 - Ftp
 - News
 - Telnet

Τα τμήματα που συγκροτούν το Web

- Το Διαδίκτυο
- Εφαρμογές
- Web

- **Εφαρμογή web client (browser)**
 - Παρουσιάζει στον χρήστη τα δεδομένα που λαμβάνει από τον web server
- **Εφαρμογή web server**
 - Εξυπηρετεί τις αιτήσεις των web clients, επιστρέφοντας τα αντικείμενα (ιστοσελίδες και άλλα αρχεία) που ζητούν
- **Γλώσσα HTML**
 - Περιεχόμενο και μορφή ιστοσελίδων
- **Πρωτόκολλο HTTP**
 - Για τη μεταφορά δεδομένων μεταξύ web server και web client

HyperText Markup Language (HTML)

- Το Διαδίκτυο
- Εφαρμογές
- Web

- **Τι είναι η γλώσσα HTML;**
 - Μια γλώσσα για τη συγκρότηση εγγράφων web με πρόσθετες οδηγίες (markup) σχετικά με τη δομή και την παρουσίασή τους
- **Ποιες οι δυνατότητες της HTML; [W3C]**
 - Εμφάνιση κειμένου, επικεφαλίδων, τίτλων, εικόνων, λιστών, πινάκων κλπ.
 - Σύνδεση με άλλους πόρους μέσω υπερσυνδέσμων
 - Χρήση φορμών για τη μετάδοση των δεδομένων χρήστη
 - Ενσωμάτωση ήχου, εικόνας και άλλων εφαρμογών μέσα στα έγγραφα web
- **Επίσης**
 - Ξεχωριστές πληροφορίες παρουσίασης (stylesheets)
 - Ενσωμάτωση προγραμμάτων (scripts)

HyperText Transfer Protocol (HTTP)

- Το Διαδίκτυο
- Εφαρμογές
- Web

- Για την προσπέλαση και μεταφορά εγγράφων web μεταξύ client-server
 - μεταφορά κειμένου, εικόνας, ήχου, βίντεο κλπ
 - Τα επιστρεφόμενα αντικείμενα
 - ή είναι **στατικά** αποθηκευμένα στον server
 - ή δημιουργούνται **δυναμικά** με την εκτέλεση κάποιου προγράμματος (στον server)
 - Αίτηση (client) \Rightarrow Απόκριση (server)
 - URL (Uniform Resource Locator)
 - Πού βρίσκεται το αντικείμενο που θέλουμε;
 - **πρωτόκολλο** `:// server [:port] / μονοπάτι`
 - `http://www.ionio.gr/cs/index.html`

Ηλεκτρονικό Ταχυδρομείο (e-mail)

- Το Διαδίκτυο
- Εφαρμογές
- Web
- E-mail

- **Η πρώτη επιτυχημένη δικτυακή εφαρμογή**
 - Πολύ πριν την ανάπτυξη του διαδικτύου!
 - Ασύγχρονη εφαρμογή
 - Δεν απαιτείται ο χρήστης να είναι συνεχώς on-line για να λάβει e-mail
 - Όμως κάποιος υπολογιστής πρέπει να είναι on-line συνεχώς
 - Παλαιά πρωτόκολλα
 - Αρχικά για τη μεταφορά μηνυμάτων σε 7-bit ASCII
 - Αναγκαία ειδική κωδικοποίηση για μεταφορά δυαδικών δεδομένων (π.χ. multimedia)
 - Το μήνυμα και όλα τα συνημμένα αντικείμενα μεταφέρονται ως ένα ενιαίο πακέτο δεδομένων

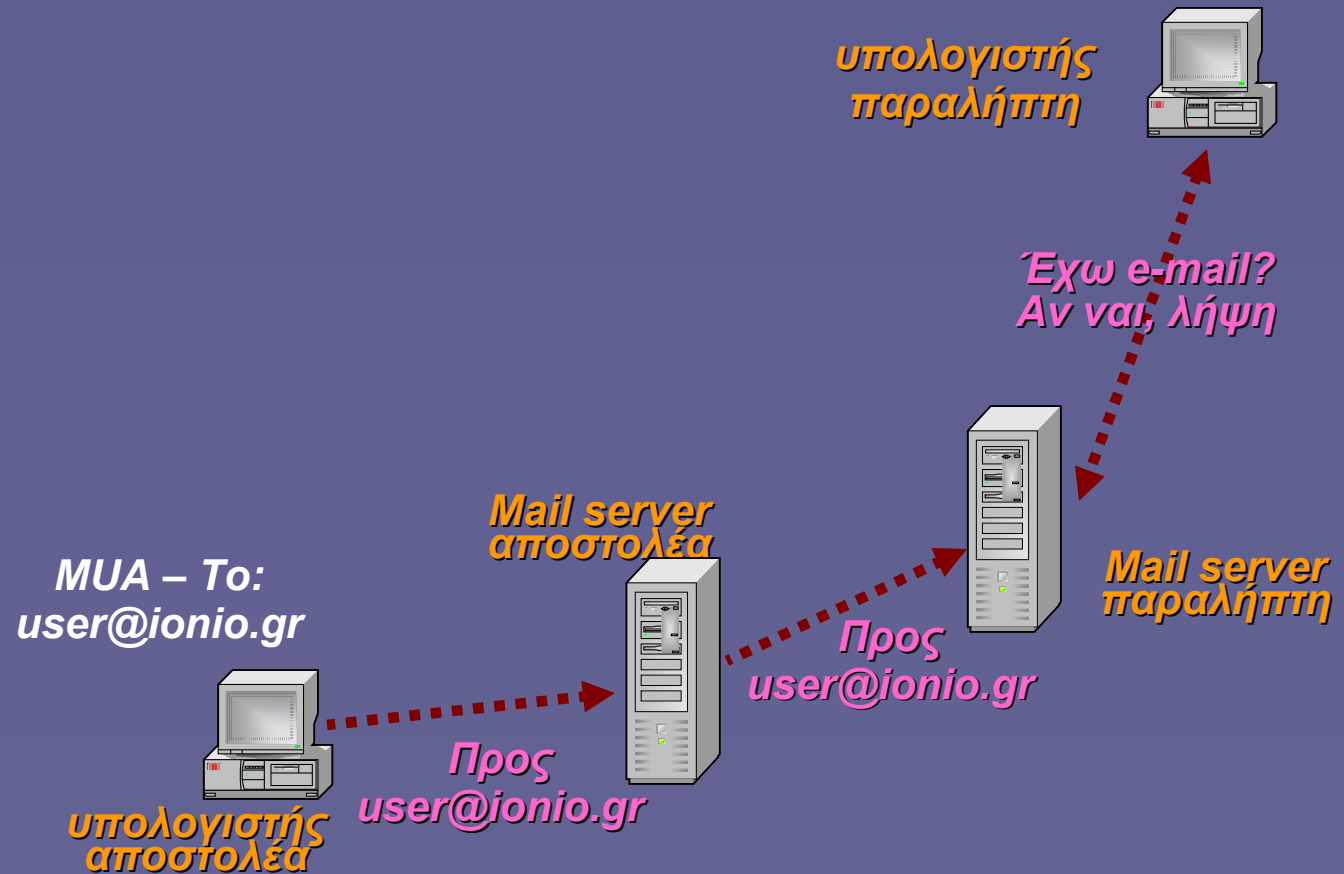
Τμήματα της εφαρμογής e-mail

- Το Διαδίκτυο
- Εφαρμογές
- Web
- E-mail

- **Εφαρμογή χρήστη (mail user agent – MUA)**
 - Για τη σύνθεση-αποστολή και λήψη-απεικόνιση e-mail
- **Εφαρμογή mail server (συνεχώς on-line)**
 - Για την προώθηση των εξερχόμενων μηνυμάτων
 - Και την παραλαβή-αποθήκευση των εισερχόμενων
- **Πρωτόκολλο προώθησης μηνυμάτων**
 - Από τον mail server του αποστολέα στον mail server του παραλήπτη
- **Πρωτόκολλο λήψης εισερχόμενων μηνυμάτων**
 - Από την εφαρμογή MUA του χρήστη
- **Πρωτόκολλο μορφής μηνυμάτων e-mail**
 - Το μορφότυπο (format) των μεταδιδόμενων μηνυμάτων

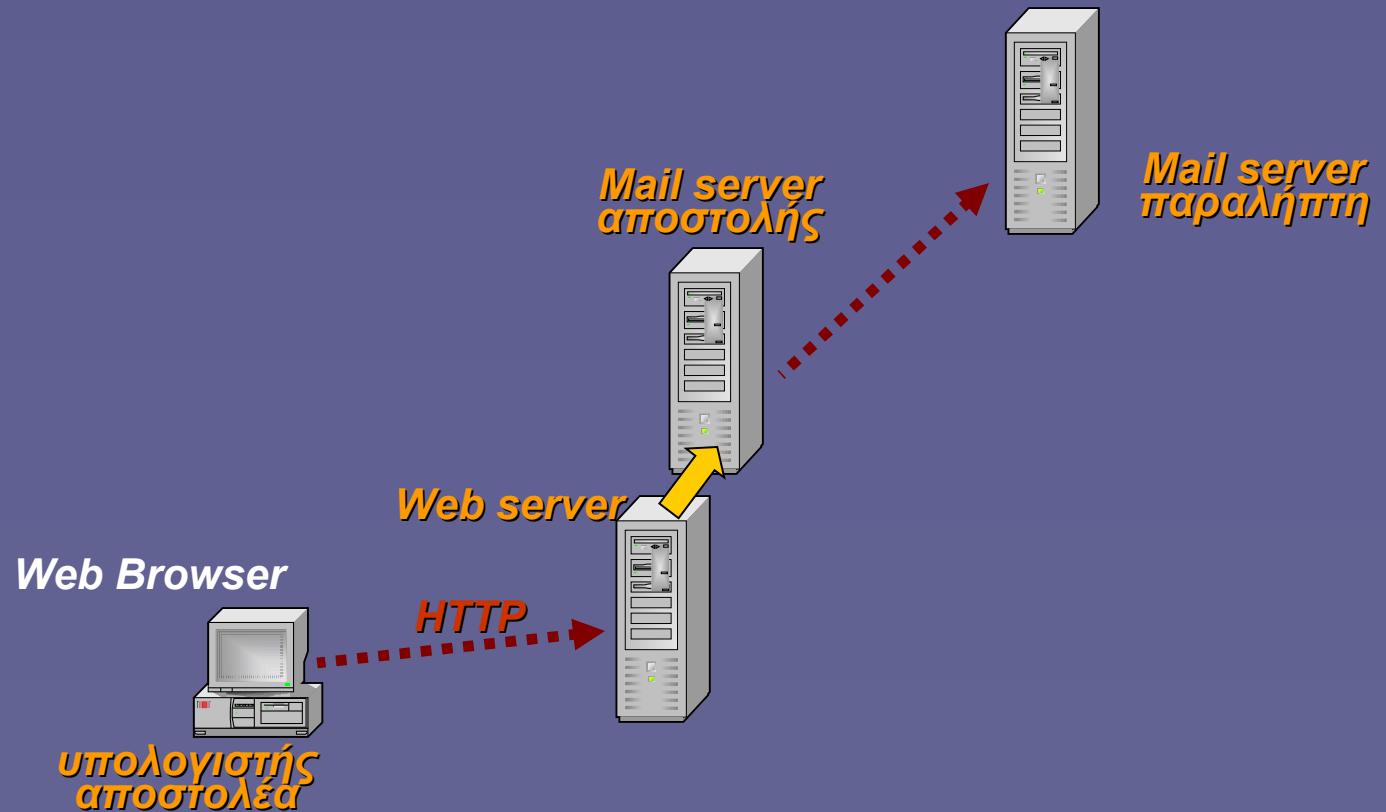
Μεταφορά e-mail

- Το Διαδίκτυο
- Εφαρμογές
- Web
- E-mail



Εφαρμογές Webmail

- Το Διαδίκτυο
- Εφαρμογές
- Web
- E-mail



Domain Name System (DNS)

- Το Διαδίκτυο
- Εφαρμογές
- Web
- E-mail
- **DNS**

- Η “υπηρεσία καταλόγου” του Διαδικτύου
 - Μετάφραση μνημονικών ονομάτων (domain names) σε διευθύνσεις IP
 - Και το αντίστροφο
 - Π.χ. **www.ionio.gr** \Rightarrow **195.130.124.68**
 - Και **...@ionio.gr** \Rightarrow **195.130.124.68**
 - Ένας μηχανισμός μετάφρασης – εύρεσης της αντιστοιχίας μεταξύ των δύο μορφών διεύθυνσης

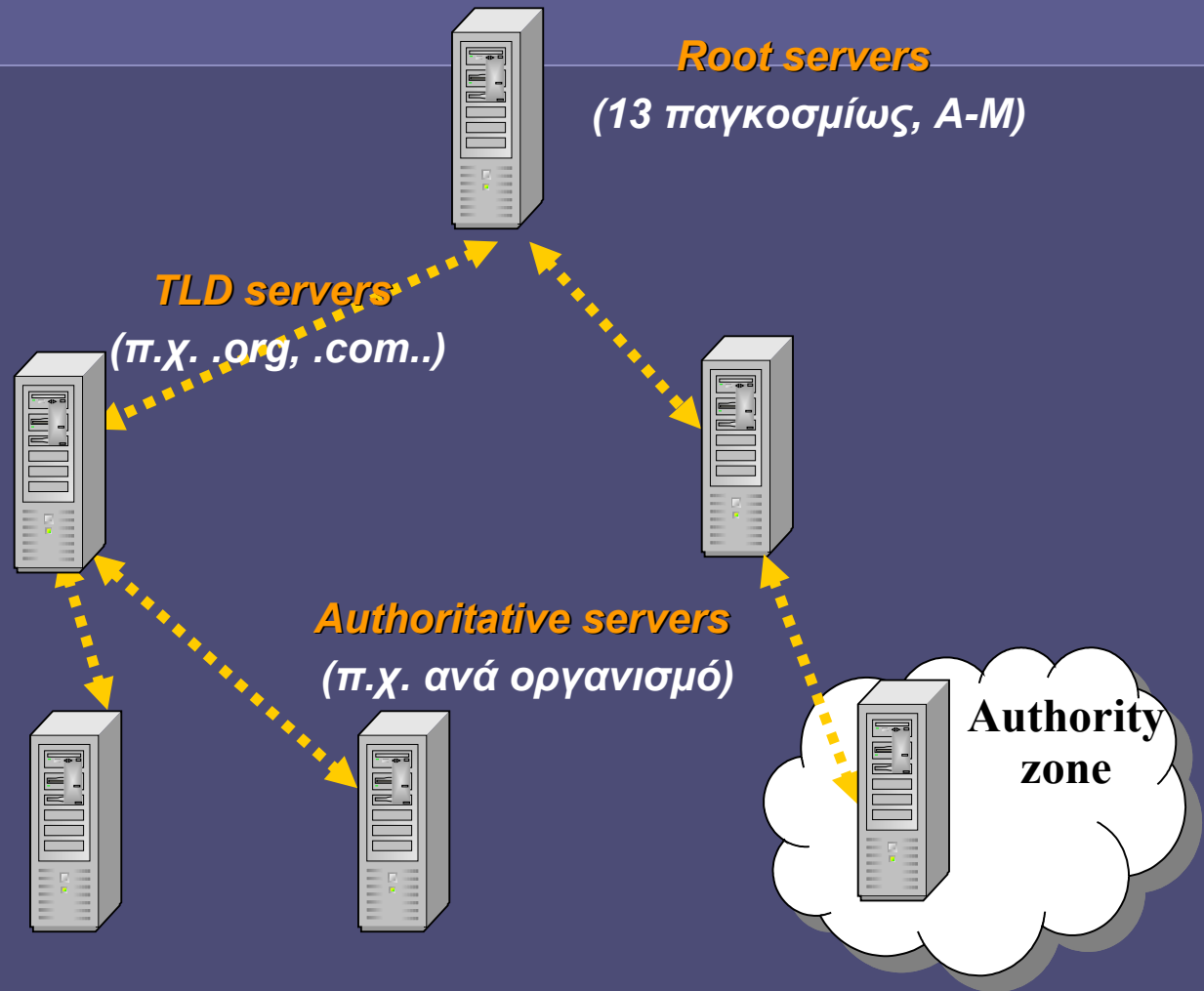
Τμήματα της εφαρμογής DNS

- Το Διαδίκτυο
- Εφαρμογές
- Web
- E-mail
- DNS

- **Βάση (ή βάσεις) δεδομένων**
 - Με τις αντιστοιχίες domain names – διευθύνσεων IP
- **DNS servers**
 - Δέχονται ερωτήσεις και απαντούν με αντιστοιχίες ονομάτων – διευθύνσεων IP
- **Πρωτόκολλο DNS**
 - Καθορίζει τη μορφή των αιτήσεων και απαντήσεων

Η ιεραρχία των DNS servers

- Το Διαδίκτυο
- Εφαρμογές
- Web
- E-mail
- DNS



- Επίσης: τοπικοί DNS servers

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2007-08

Λειτουργικά Συστήματα (Operating Systems)

<http://www.ionio.gr/~mistral/tp/csintro/>

Μ.Στεφανιδάκης

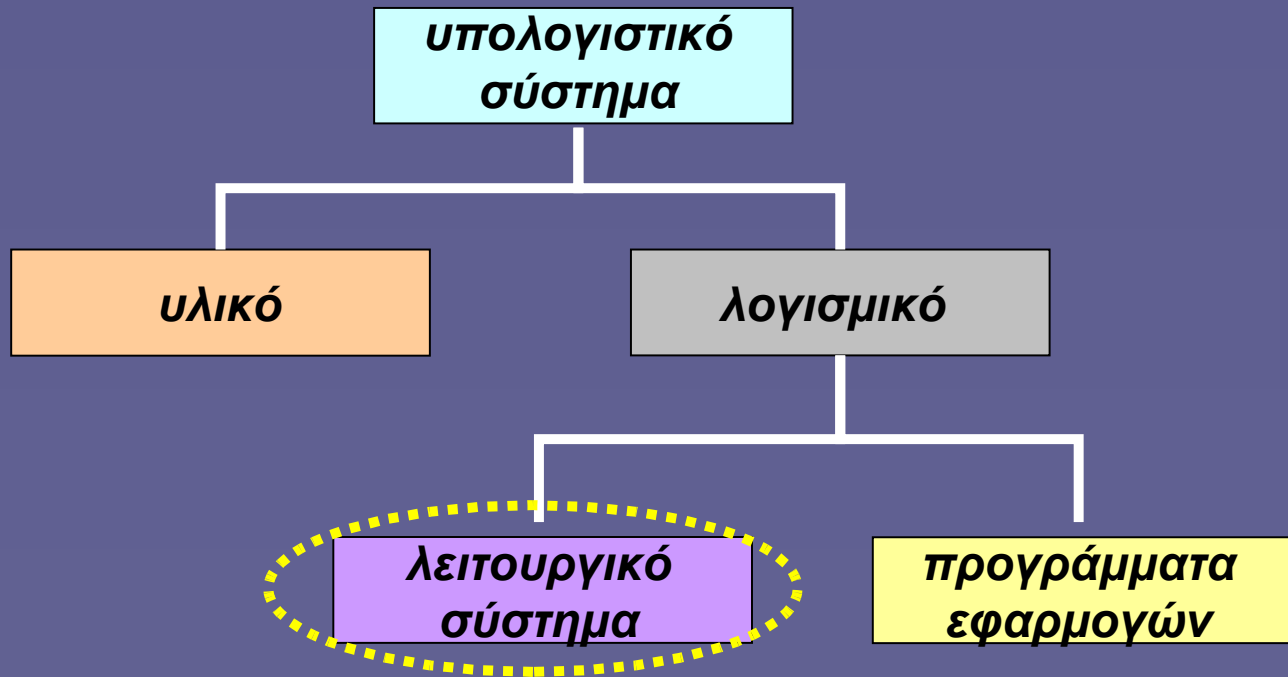
Τι είναι ένα λειτουργικό σύστημα;

- Εισαγωγή

- Λογισμικό – ο “γενικός επιβλέπων” ενός υπολογιστικού συστήματος
 - Εκτέλεση προγραμμάτων εφαρμογών
 - Διαχείριση υπολογιστικών πόρων
 - ΚΜΕ
 - Μνήμη
 - Συσκευές E/E
 - Ενδιάμεσο μεταξύ χρήστη και υπολογιστή
 - Κοινές-συχνά χρησιμοποιούμενες λειτουργίες εφαρμογών
 - “Βιβλιοθήκες” λειτουργικού συστήματος

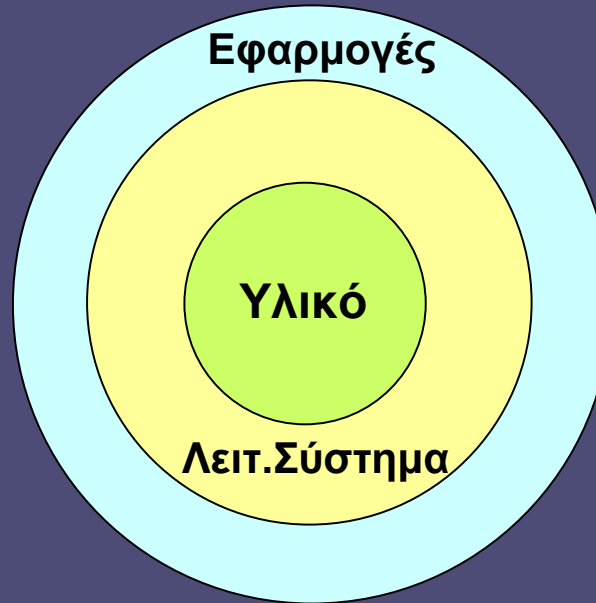
Το Λειτουργικό Σύστημα ως μέρος του υπολογιστή

- Εισαγωγή



Μια άλλη απεικόνιση

- Εισαγωγή



- Κέλυφος (shell)

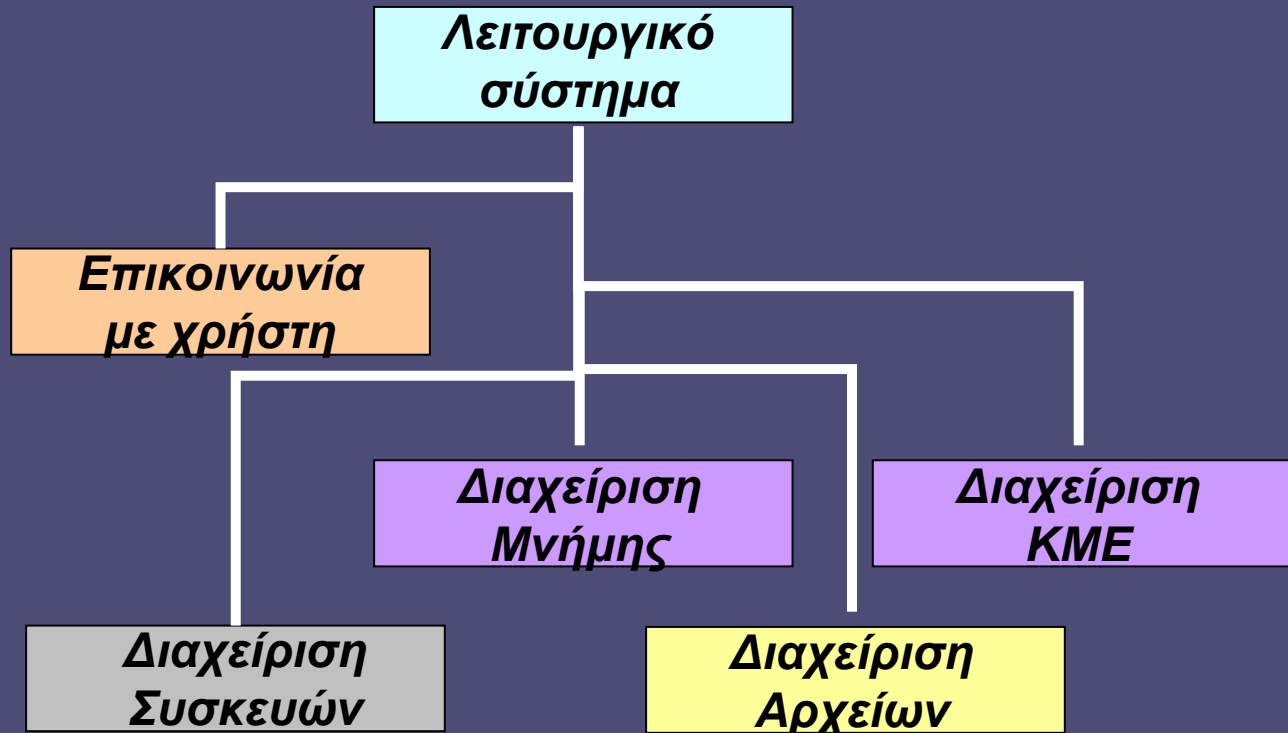
Η εξέλιξη των Λειτουργικών Συστημάτων

- Εισαγωγή

- **Ανάλογα με την εξέλιξη των υπολογιστών:**
- **Τα πρώτα μεγάλα συστήματα (mainframes)**
 - Λ.Σ. δέσμης (batch operating systems)
- **Συστήματα με πολυπρογραμματισμό (multiprogramming)**
 - Λ.Σ. με δυνατότητες χρονοπρογραμματισμού
- **Προσωπικοί Υπολογιστές**
 - Λ.Σ. ενός χρήστη (single user operating system)
- **Παράλληλα συστήματα (πολλές ΚΜΕ)**
 - Σύνθετα Λ.Σ. για κατανομή εργασίας
- **Κατανεμημένα συστήματα (μέσω δικτύου-internet)**
 - Λ.Σ. με πρόσθετες ικανότητες (συντονισμός, μηνύματα, ασφάλεια δεδομένων..)

Διαχείριση πόρων

- Εισαγωγή



- Η επικοινωνία με τον χρήστη είναι μέρος του Λ.Σ;

Μονοπρογραμματισμός

- Εισαγωγή
- Διαχείριση μνήμης

- Τα πρώτα χρόνια των υπολογιστών...
- Στην μνήμη βρίσκονται
 - Ένα πρόγραμμα
 - Το λειτουργικό σύστημα
- Απλή διαχείριση μνήμης
- Αν το μέγεθος του προγράμματος είναι μεγάλο;
- Πότε μπορεί να εκτελεστεί άλλο πρόγραμμα;

**Λειτουργικό
Σύστημα**

**Πρόγραμμα:
κώδικας και
δεδομένα**

μνήμη

Πολυπρογραμματισμός

- Εισαγωγή
- Διαχείριση μνήμης

- Πολλά προγράμματα στη μνήμη
- “Ταυτόχρονη” εκτέλεση
 - Στην πραγματικότητα διαδοχική εκτέλεση εντολών από κάθε πρόγραμμα
 - Φαινομενικά τα προγράμματα εκτελούνται “παράλληλα”
- Οι υπολογιστικοί πόροι πρέπει να μοιράζονται μεταξύ των προγραμμάτων
 - Διαιτησία
 - Ρόλος του Λειτουργικού Συστήματος!



μνήμη

Πολυπρογραμματισμός (συνέχεια)

- Εισαγωγή
- Διαχείριση μνήμης

- Το Λειτουργικό Σύστημα διατηρεί πληροφορία για κάθε εκτελούμενο πρόγραμμα (διεργασία)
 - Process Control Block (PCB) ή Task Control Block (TCB)
- Όταν διακόπτεται η εκτέλεση:
 - Αποθήκευση τιμής program counter (PC)
 - Τρέχουσα διεύθυνση εκτελούμενης εντολής
 - Αποθήκευση τιμών καταχωρητών
- Όταν ξεκινά πάλι η εκτέλεση:
 - Αποκατάσταση αποθηκευμένων τιμών
 - Το πρόγραμμα συνεχίζει την εκτέλεση από το σημείο που διακόπηκε

Τεχνικές διαχείρισης μνήμης

- Εισαγωγή
- Διαχείριση μνήμης

- Πώς θα εξασφαλιστεί η διαθεσιμότητα της μνήμης για τα προγράμματα που εκτελούνται “ταυτόχρονα”
- Χωρίς εναλλαγή (swapping)
 - Το πρόγραμμα παραμένει στη μνήμη
 - Διαμέριση - Σελιδοποίηση
- Με εναλλαγή
 - Μέρος του προγράμματος μπορεί να εναλλάσσει θέση μεταξύ μνήμης και δίσκου κατά την εκτέλεση
 - Σελιδοποίηση κατ’απαίτηση
 - Κατάτμηση κατ’απαίτηση

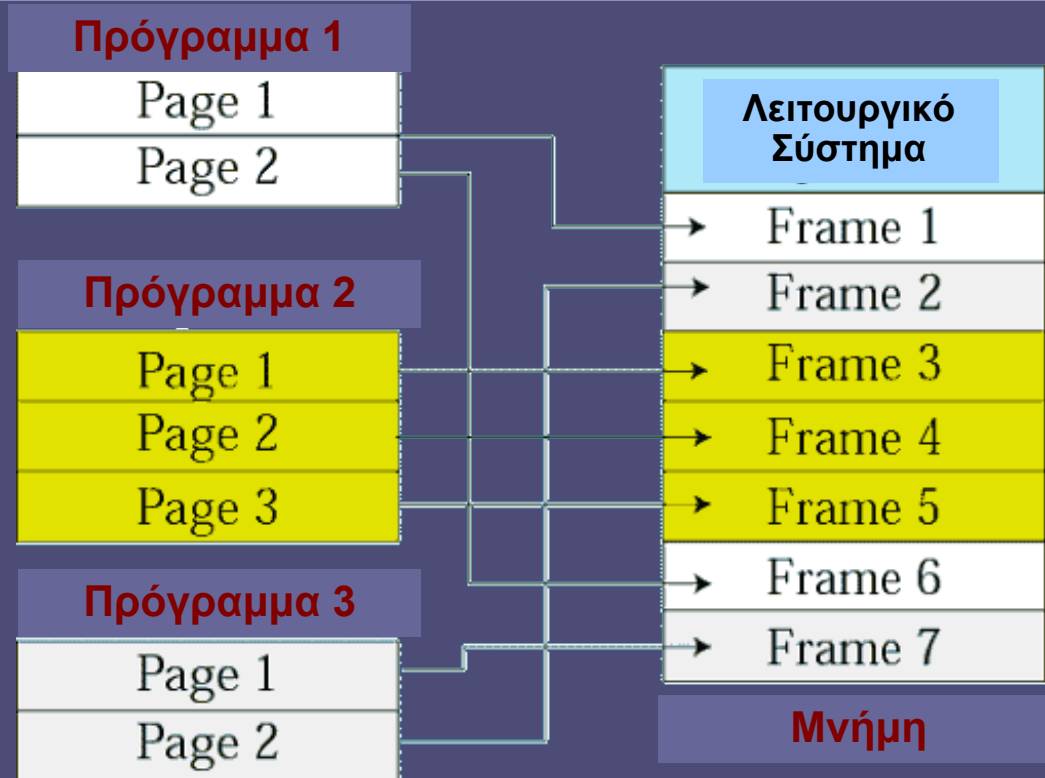
Διαμέριση (partitioning)

- Εισαγωγή
- Διαχείριση μνήμης

- Κάθε πρόγραμμα έχει το δικό του τμήμα (διαμέριση) στη μνήμη
 - Όλο το πρόγραμμα σε συνεχόμενες θέσεις μνήμης
 - Αντικατάσταση προγραμμάτων μετά τον τερματισμό
 - Εισαγωγή νέων
 - Πιθανή δημιουργία κενών τμημάτων
 - Συμπύκνωση (compaction)
 - Αργή διαδικασία!
 - Το Λ.Σ. διατηρεί λίστα χρησιμοποιούμενων τμημάτων μνήμης

Σελιδοποίηση

- Εισαγωγή
- Διαχείριση μνήμης



- Κάθε πρόγραμμα χωρίζεται σε σελίδες (**pages**)
- Η μνήμη χωρίζεται σε πλαίσια (**frames**)
 - Σελίδες και πλαίσια: ίδιο μέγεθος
 - Αποδοτικότερη χρήση μνήμης (λιγότερα κενά)

Σελιδοποίηση (συνέχεια)

- Εισαγωγή
- Διαχείριση μνήμης

- Οι σελίδες έχουν προκαθορισμένο μέγεθος
 - π.χ. 4KBytes
- Μπορούν να τοποθετηθούν σε οποιοδήποτε πλαίσιο μνήμης
- Ενδεχομένως να μην βρίσκονται όλες στη μνήμη
 - Ανάκληση από δίσκο όταν τις χρειαστεί το πρόγραμμα
 - Σελιδοποίηση κατ' απαίτηση (demand paging)
- Πρακτικά...
 - Το πρόγραμμα βλέπει διαφορετική διεύθυνση μνήμης από την πραγματική (φυσική) του πλαισίου
 - Εικονική μνήμη (virtual memory)
 - Απαιτείται η τήρηση πινάκων αντιστοίχισης διευθύνσεων
 - Απαραίτητη η υποστήριξη από το υλικό (ΚΜΕ)

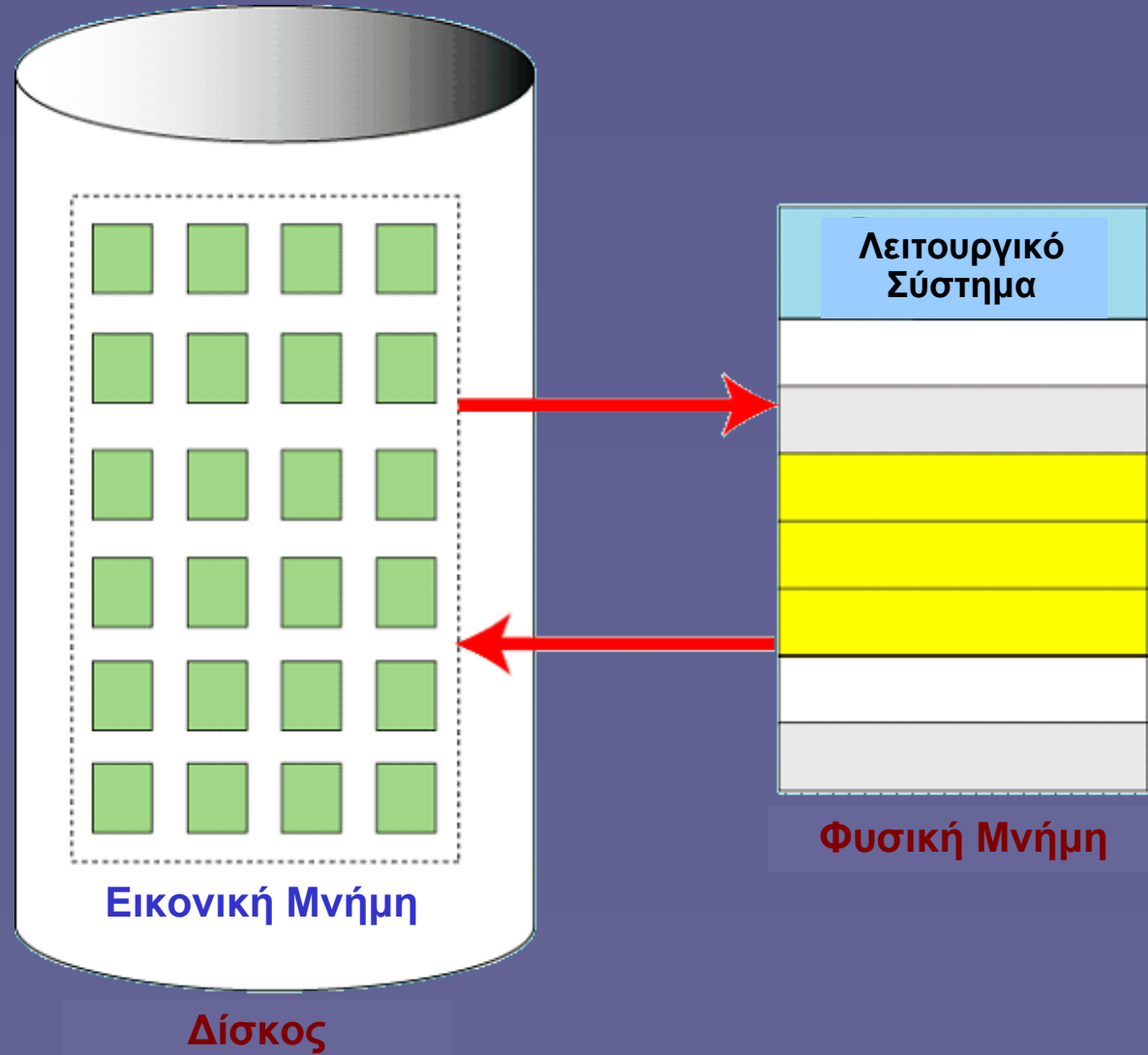
Κατάτμηση (segmentation)

- Εισαγωγή
- Διαχείριση μνήμης

- Παρόμοια με σελιδοποίηση
- Διαίρεση σε τμήματα μεταβλητού μεγέθους
 - “Για να ταιριάζει με τα χαρακτηριστικά του προγράμματος” (;)
 - Πρέπει να φυλάσσεται και η πληροφορία μεγέθους
- Κατ’απαίτηση – εικονική μνήμη
- Μπορεί να συνδυαστεί με σελιδοποίηση
 - Σε δύο επίπεδα

Εικονική Μνήμη

- Εισαγωγή
- Διαχείριση μνήμης



Προγράμματα, εργασίες και διεργασίες

- Εισαγωγή
- Διαχείριση μνήμης
- Διεργασίες

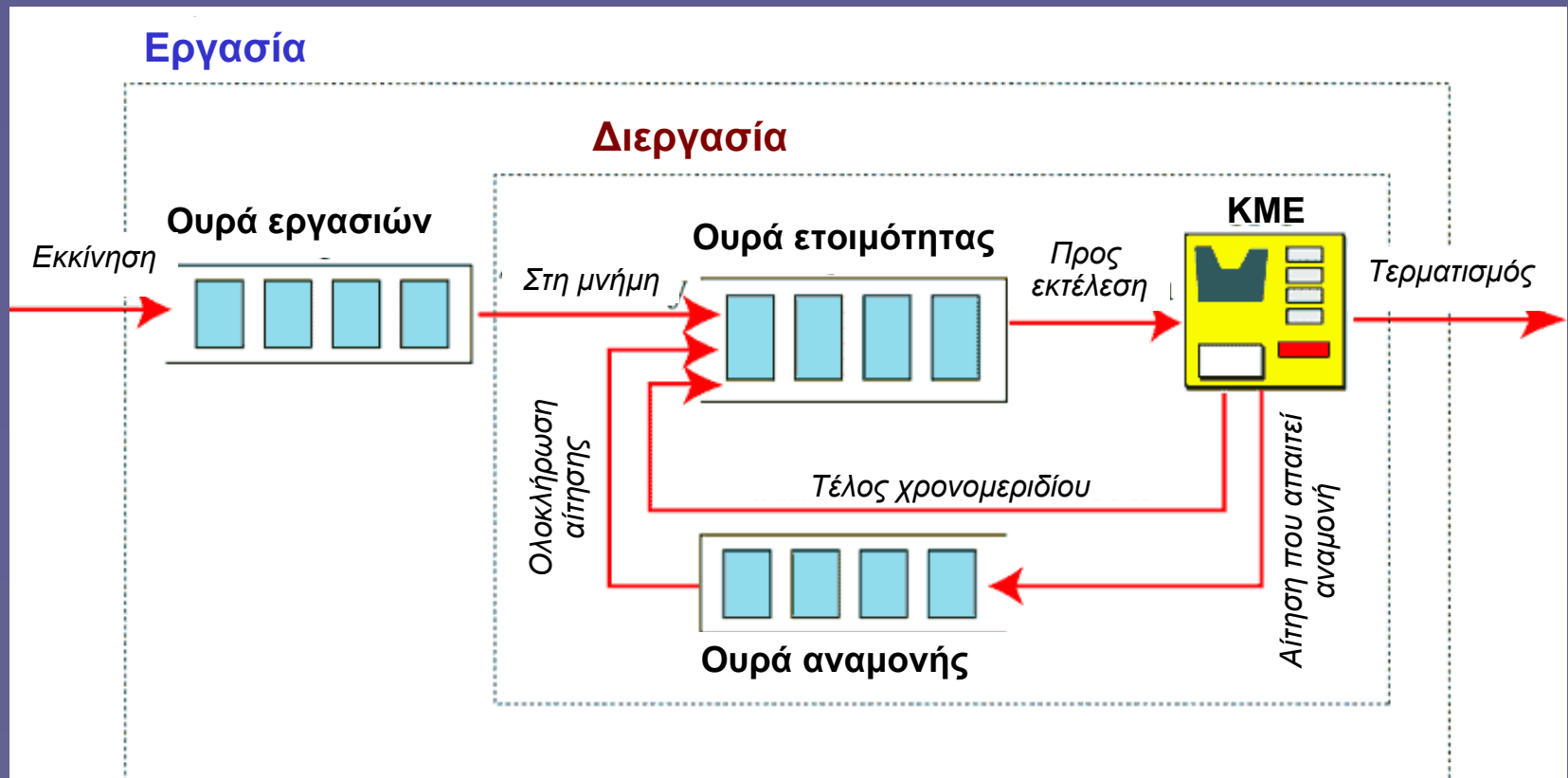
- **Πρόγραμμα (program)**
 - Ανενεργό σύνολο εντολών (στον δίσκο)
- **Εργασία (job)**
 - Πρόγραμμα που έχει επιλεγεί για εκτέλεση
 - Στον δίσκο ή στη μνήμη
- **Διεργασία (process)**
 - Πρόγραμμα υπό εκτέλεση
 - Στη μνήμη
 - Εκτελείται ή αναμένει για χρήση πόρων

Χρονοδρομολόγηση (scheduling)

- Εισαγωγή
- Διαχείριση μνήμης
- Διεργασίες

- **Καταστάσεις διεργασίας**
 - Έτοιμη προς εκτέλεση (ready)
 - Εκτελούμενη (running)
 - Σε αναμονή (waiting/blocked)
 - Για τη χρήση υπολογιστικών πόρων
- **Όταν πολλές διεργασίες είναι έτοιμες, ποια θα εκτελεστεί;**
 - Απόφαση χρονοδρομολογητή (scheduler)
 - είναι μέρος του Λειτουργικού Συστήματος
 - Κριτήριο Επιλογής
 - “Όποιος ήρθε πρώτος” (first-in first-out – FIFO)
 - Χρησιμοποιώντας προτεραιότητες
 - Ανάλογα με προθεσμίες

Ουρές διαχείρισης διεργασιών

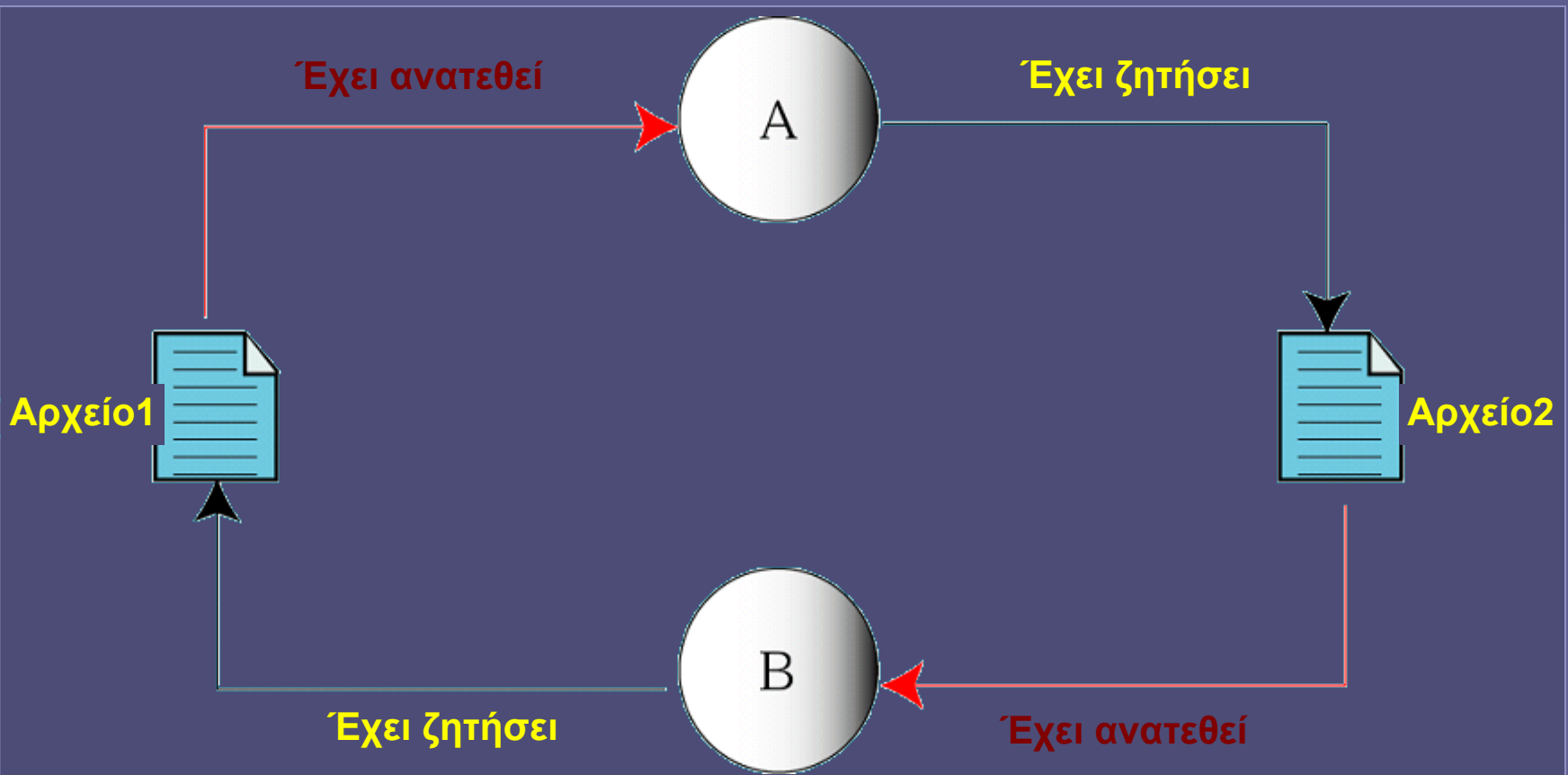


Συγχρονισμός διεργασιών

- Εισαγωγή
- Διαχείριση μνήμης
- Διεργασίες

- Στη χρήση διαμοιραζόμενων πόρων
- **Οι κοινοί πόροι μπορούν να χρησιμοποιούνται μόνο από μία διεργασία κάθε στιγμή**
- Αναμονή υπολοίπων διεργασιών
- Εμφάνιση καταστάσεων σύγκρουσης
 - Αδιέξοδο
 - Λιμοκτονία

Αδιέξοδο



- Αρχεία: κοινόχρηστοι πόροι

Αδιέξοδο (συνέχεια)

- Εισαγωγή
- Διαχείριση μνήμης
- Διεργασίες

- **Συνθήκες εμφάνισης**
 - Αμοιβαίος αποκλεισμός στη χρήση των πόρων
 - Παρακράτηση περισσότερων του ενός πόρων
 - Μη προεκτοπισμός κατά τη χρήση των πόρων
 - Κυκλική αναμονή
- **Αν επιτρέψουμε την εκτέλεση διεργασίας μόνο εάν όλοι οι πόροι που ζητά είναι ελεύθεροι;**
 - Κίνδυνος λιμοκτονίας
 - Η διεργασία μπορεί να μην εκτελεστεί “ποτέ”

Διαχείριση συσκευών και αρχείων

- Εισαγωγή
- Διαχείριση μνήμης
- Διεργασίες
- Διαχείριση συσκευών και αρχείων

- **Συσκευές εισόδου-εξόδου (E/E)**
 - Διαχείριση
 - Εξασφάλιση ορθής λειτουργίας
 - Ρύθμιση προσπέλασης από διεργασίες
- **Διαχείριση αρχείων**
 - Λειτουργίες δημιουργίας, διαγραφής, τροποποίησης
 - Έλεγχος προσπέλασης
 - Επίβλεψη αποθήκευσης

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2007-08

Αρχεία και Συστήματα Αρχείων

<http://www.ionio.gr/~mistral/tp/csintro/>

Μ.Στεφανιδάκης

Το Λειτουργικό Σύστημα (ξανά)

- Εισαγωγή

- Ο “γενικός επιβλέπων” ενός υπολογιστικού συστήματος
 - Εκτέλεση προγραμμάτων εφαρμογών
 - Διαχείριση υπολογιστικών πόρων
 - ΚΜΕ
 - Μνήμη
 - Συσκευές E/E
 - Δίσκοι
 - Αρχεία
 - Ενδιάμεσο μεταξύ χρήστη και υπολογιστή
 - Κοινές-συχνά χρησιμοποιούμενες λειτουργίες εφαρμογών
 - “Βιβλιοθήκες” λειτουργικού συστήματος

Τι είναι ένα “αρχείο”;

- Εισαγωγή

- **Μια αφαιρετική δομή (abstraction)**
 - Για την αποθήκευση δεδομένων στους δίσκους του συστήματος
 - Προσφέρεται από το **λειτουργικό σύστημα**
- **Αρχεία δεδομένων**
 - Ως ακολουθία (stream) από bytes
 - “binary mode”
 - Ως διαδοχικές γραμμές κειμένου
 - “text mode”
- **Ειδικά αρχεία**
 - Συσκευές Ε/Ε, κατάλογοι, πληροφορία συστήματος...
 - **Ανάλογα με το λειτουργικό σύστημα!**

Δυαδικά αρχεία

- Εισαγωγή

- Περιέχουν οποιαδήποτε ακολουθία bytes
 - Ανάγνωση (read) n bytes

read(4) : 2D, 98, 44, 59

..0F 35 2D 98 44 59 FD 77 47 23 89 12 ..



τρέχουσα θέση



τρέχουσα θέση
(μετά)

- Εγγραφή (write) n bytes

write(AA, 91, 11, 15)

..0F 35 AA 91 11 15 FD 77 47 23 89 12 ..



τρέχουσα θέση



τρέχουσα θέση
(μετά)

Επίσης:

Μετακίνηση
τρέχουσας θέσης
οποιαδήποτε μέσα
στο αρχείο (seek)

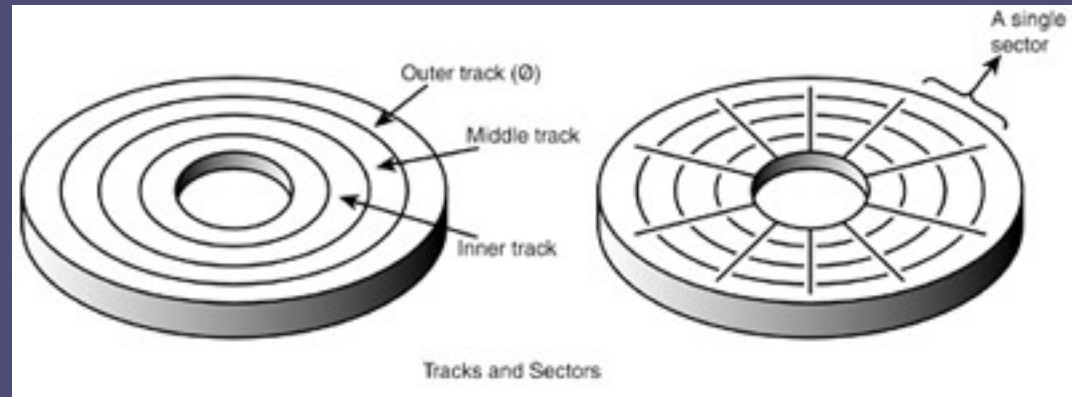
Αρχεία Κειμένου

- Εισαγωγή

- Διαδοχικές γραμμές από χαρακτήρες
 - Σε κάποια γνωστή κωδικοποίηση
 - 7/8 bit ASCII, Unicode, Unicode (UTF-8)...
 - Τερματισμός γραμμής με '\n' (newline)
 - Windows: 0D 0A – Unix: 0A
- Λειτουργίες
 - Ανάγνωση επόμενης γραμμής
 - Εγγραφή νέας γραμμής
 - στο τέλος του αρχείου
- Προσοχή!
 - Μπορούμε να χειριστούμε ένα αρχείο κειμένου ως δυαδικό αρχείο
 - Όχι όμως και το αντίστροφο!

Αποθηκευτικοί δίσκοι (ξανά)

- Εισαγωγή



- Οργάνωση σε **tracks** και **sectors**
 - Sector = 512 – 4096 bytes
 - Πληροφορία αναγνώρισης sector
- Ταχύτητα περιστροφής
 - 7.200 – 10.000 RPM

Οργάνωση δίσκων σε λογικά μπλοκ

- Εισαγωγή

- Για την προσπέλαση ενός συγκεκριμένου φυσικού τμήματος δεδομένων στον δίσκο
 - Απαιτείται σύνθετη πληροφορία
 - track, sector, head...
- Οργάνωση σε ακολουθία λογικών μπλοκ
 - logical block addressing (LBA)
 - $0...n$ λογικά μπλοκ



Διαμέριση δίσκου (partitioning)

- Εισαγωγή

- **Partitions**

- Ο συνολικός δίσκος χωρίζεται σε μικρότερα **λογικά** τμήματα

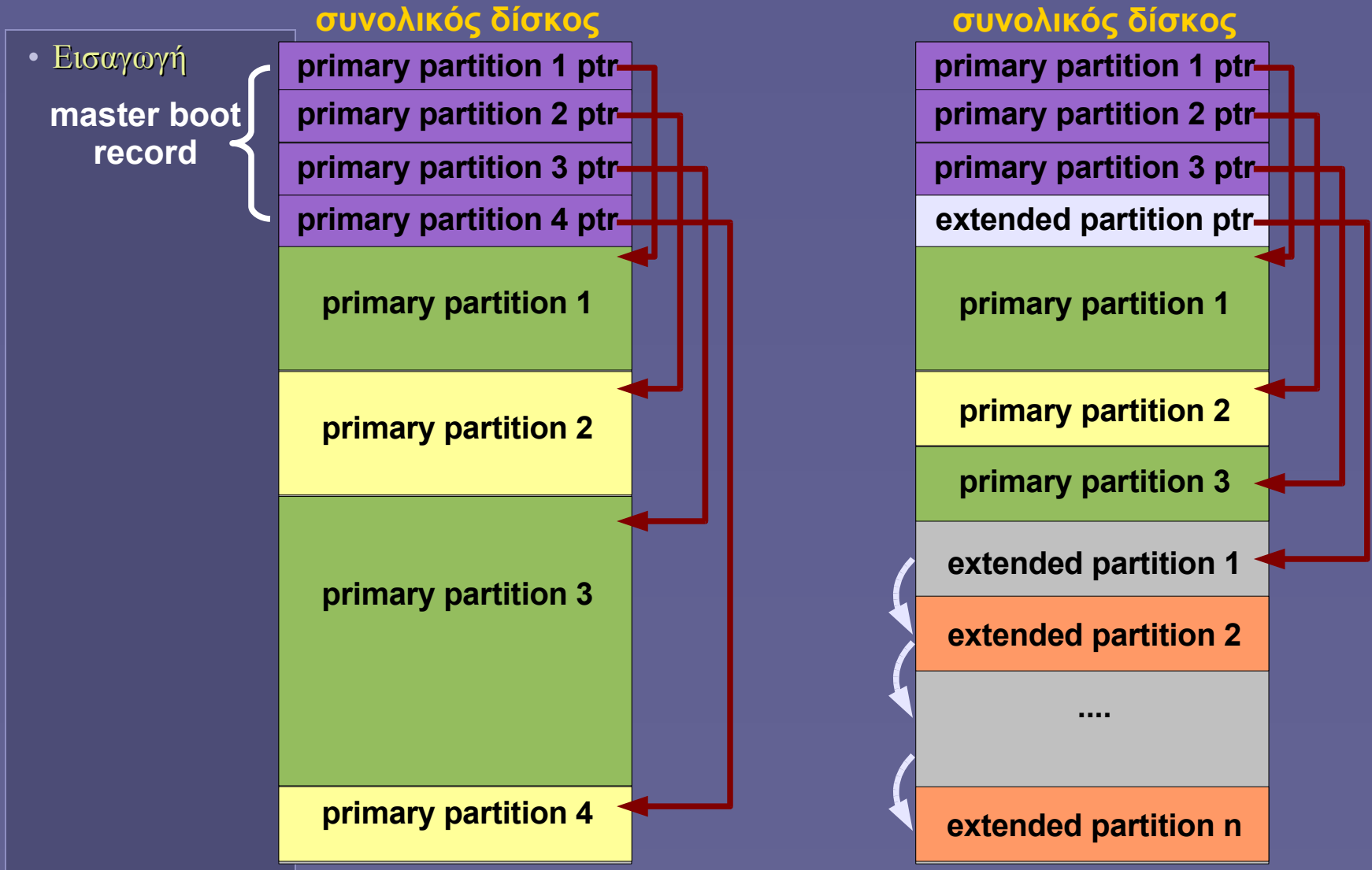
- **Γιατί;**

- Διατήρηση περισσότερων του ενός ΛΣ στο σύστημα
 - dual (multi)-booting
- Διατήρηση ΛΣ σε ξεχωριστό τμήμα από τα δεδομένα του χρήστη
 - για ευκολότερες επανεγκαταστάσεις

- **Μειονέκτημα**

- Μη ευέλικτο σχήμα
 - Το μέγεθος κάθε partition δεν αλλάζει δυναμικά!

Partitions σε συστήματα PC



Συστήματα Αρχείων

- Εισαγωγή
- Συστήματα Αρχείων

- **File systems**
 - Οργάνωση αρχείων
 - Ονόματα αρχείων
 - Σε καταλόγους (directories) ή φακέλους (folders)
 - Μέθοδοι για τη δημιουργία, τροποποίηση και διαγραφή αρχείων
 - Διαμοιρασμός κοινών αρχείων
 - Έλεγχος πρόσβασης
- **Αποθήκευση δομών ενός συστήματος αρχείων σε ένα τμήμα (partition) του δίσκου**
 - Και στην κεντρική μνήμη για γρήγορη πρόσβαση

Το περιεχόμενο ενός partition

- Εισαγωγή
- Συστήματα Αρχείων

- **Boot Block**
 - Εκτελέσιμος κώδικας
 - Όταν ο υπολογιστής μπορεί να ξεκινήσει τη λειτουργία του εκτελώντας κώδικα στο partition αυτό
- **Super Block**
 - Κεντρικές πληροφορίες για το σύστημα αρχείων που περιέχεται στο partition
- Πληροφορία ελεύθερου χώρου
- Πληροφορία θέσης αρχείων
 - Ποιο αρχείο βρίσκεται πού στον δίσκο
- Περιεχόμενα Αρχείων και Καταλόγων

Πώς αποθηκεύονται τα αρχεία;

- Εισαγωγή
- Συστήματα Αρχείων

- **Αποθήκευση σε μπλοκ (ή “cluster”)**
 - π.χ. 4KB ανά μπλοκ
 - Προσοχή! Διαφορετικό από το “λογικό μπλοκ” του δίσκου!
 - Προφανώς ένα αρχείο μπορεί να καταλαμβάνει περισσότερα από ένα μπλοκ
 - και ένας κατάλογος επίσης (είναι αρχείο κι αυτός)
- **Μέθοδος δέσμευσης μπλοκ #1: συνεχόμενα μπλοκ**
 - Όλο το αρχείο σε συνεχόμενα μπλοκ
 - Γρήγορη προσπέλαση – αρκεί να ξέρουμε αρχή και τέλος
 - Τι συμβαίνει όταν το αρχείο αλλάζει μέγεθος;
 - Fragmentation

Πώς αποθηκεύονται τα αρχεία;

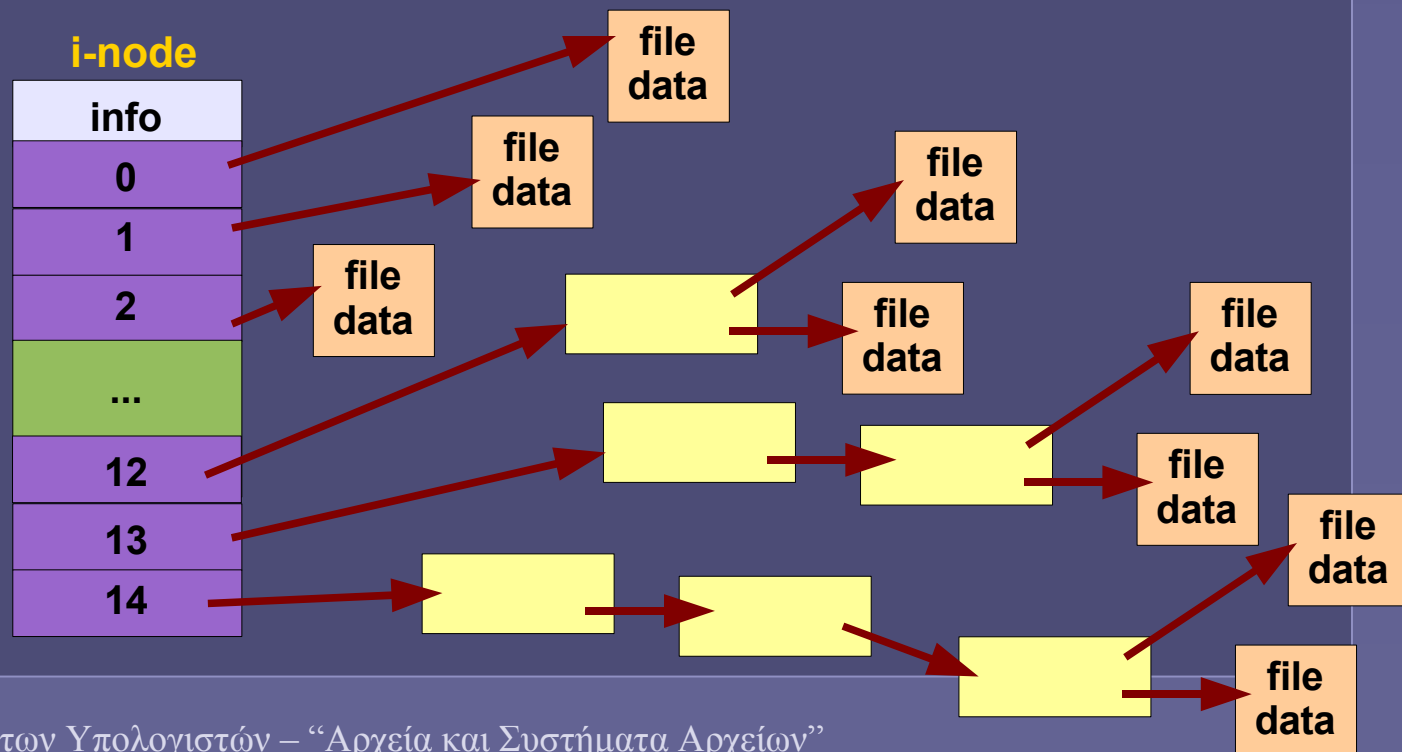
- Εισαγωγή
- Συστήματα Αρχείων

- Μέθοδος δέσμευσης μπλοκ #2: διασυνδεδεμένη λίστα
 - Σε κάθε μπλοκ υπάρχει δείκτης για το επόμενο μπλοκ
 - Αρκεί να ξέρουμε το πρώτο μπλοκ
 - Πώς θα προσπελάσω ένα σημείο προς το τέλος του αρχείου;
- Μέθοδος δέσμευσης μπλοκ #2β: διασυνδεδεμένη λίστα (παραλλαγή)
 - Πίνακας πληροφορίας για κάθε μπλοκ
 - Αν είναι δεσμευμένο και ποιο είναι το επόμενο
 - Γρήγορη προσπέλαση τυχαίου σημείου αρχείου
 - Ο πίνακας καταλαμβάνει μεγάλο χώρο

Πώς αποθηκεύονται τα αρχεία;

- Εισαγωγή
- Συστήματα Αρχείων

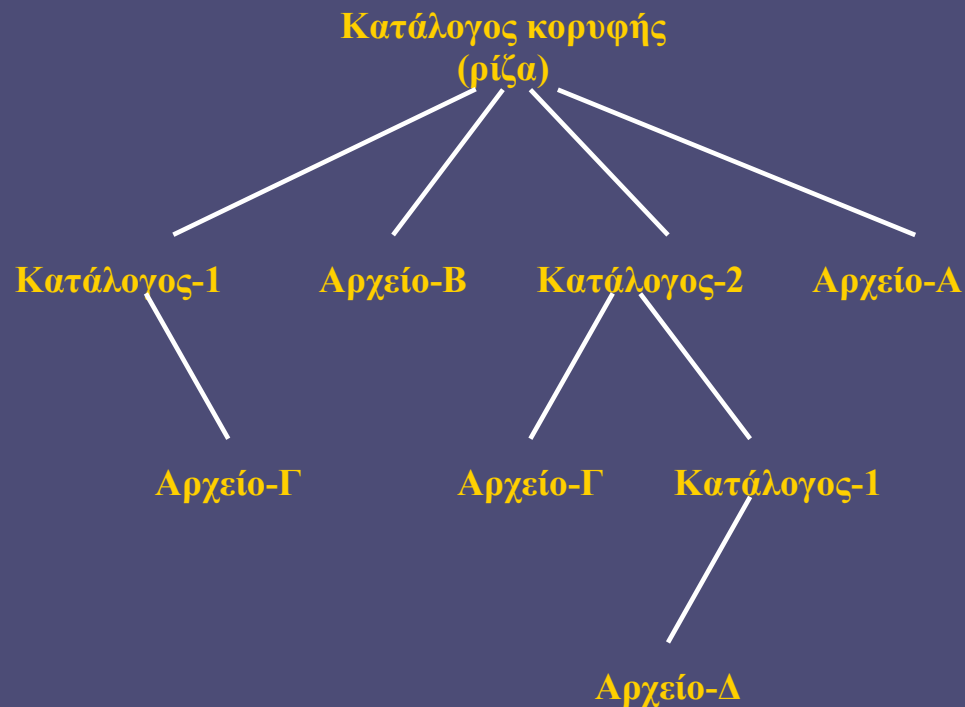
- Μέθοδος δέσμευσης μπλοκ #3: έμμεση δεικτοδότηση
 - Ειδικά μπλοκ (i-nodes) περιέχουν δείκτες σε άλλα μπλοκ
 - Συστήματα αρχείων στο ΛΣ Unix



Ιεραρχική δομή καταλόγων

- Εισαγωγή
- Συστήματα Αρχείων
- Κατάλογοι

- Η ρίζα (/) είναι στην κορυφή
 - Οι κατάλογοι (ή φάκελοι) είναι ειδικά αρχεία που περιέχουν ζεύγη (όνομα-αρχείου, θέση)



Πού βρίσκεται η ρίζα;

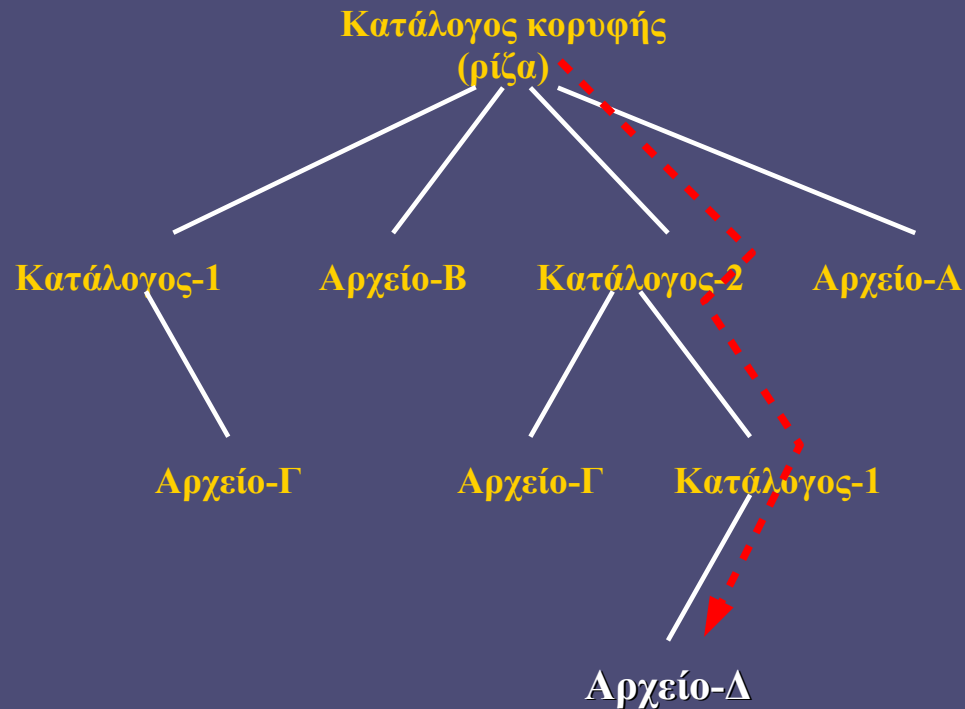
- Εισαγωγή
- Συστήματα Αρχείων
- Κατάλογοι

- **Εξαρτάται από το Λειτουργικό Σύστημα!**
 - Windows: κάθε partition που αναγνωρίζεται προστίθεται ως ένα λογικό drive (π.χ. C: D: κλπ), το οποίο αποτελεί τη ρίζα για όλους τους φακέλους που περιέχει
 - άρα έχουμε ένα δάσος από δέντρα!
 - τύπου Unix: υπάρχει μια μοναδική ρίζα και κάθε partition μπορεί να τοποθετηθεί (mount) σε οποιοδήποτε σημείο του δέντρου
 - δεν υπάρχουν λογικά drives

Μονοπάτι Αρχείου

- Εισαγωγή
- Συστήματα Αρχείων
- **Κατάλογοι**

- Από τη ρίζα προς το αρχείο (απόλυτο μονοπάτι)
 - Ως αναγνωριστικό του αρχείου που επιλέγουμε
 - Για ευκολία: . είναι ο τρέχων κατάλογος, .. είναι ο κατάλογος πάνω από τον τρέχοντα



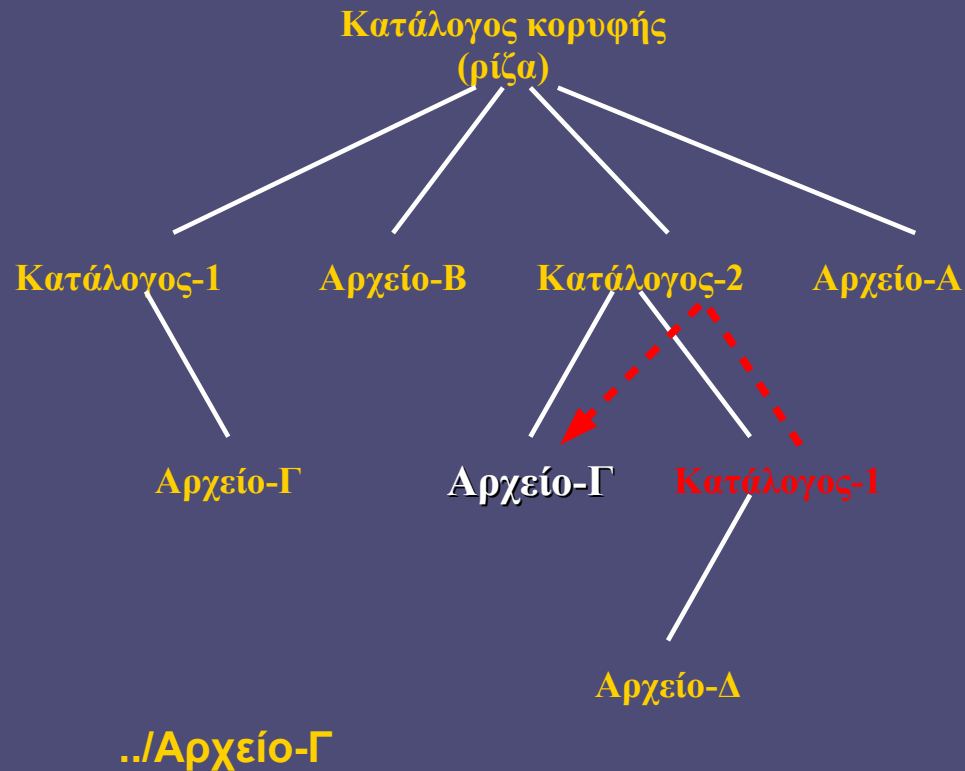
/Κατάλογος-2/Κατάλογος-1/Αρχείο-Δ

Μονοπάτι Αρχείου

- Εισαγωγή
- Συστήματα Αρχείων
- Κατάλογοι

- Σχετικό μονοπάτι

- Έστω ότι ο τρέχων κατάλογος είναι ο **Κατάλογος-1**
- Για ευκολία: **.** είναι ο τρέχων κατάλογος, **..** είναι ο κατάλογος πάνω από τον τρέχοντα



Ονόματα Αρχείων

- Εισαγωγή
- Συστήματα Αρχείων
- Κατάλογοι
- Αρχεία

- **Παλαιότερα υπήρχαν περιορισμοί**
 - π.χ η μορφή 8.3
 - 8 χαρακτήρες (αλφαριθμητικοί και ορισμένα σημεία στίξης)
 - τελεία
 - 3 χαρακτήρες επέκταση
 - Η επέκταση και σήμερα δηλώνει τον τύπο του αρχείου
 - χωρίς να ισχύει κατ'ανάγκη!
 - .doc .txt .html .pdf .exe κλπ
- **Σήμερα υπάρχει μεγαλύτερη ελευθερία στα ονόματα των αρχείων**
 - αρκεί στον ίδιο κατάλογο το όνομα να είναι μοναδικό

“Ανοίγοντας” ένα αρχείο

- Εισαγωγή
- Συστήματα Αρχείων
- Κατάλογοι
- **Αρχεία**


- **Άνοιγμα αρχείου**
 - `open(filename,mode)`
 - `filename` = το όνομα του αρχείου (και το μονοπάτι αν χρειάζεται)
 - `mode` = το είδος της ζητούμενης επεξεργασίας
 - `read` (ανάγνωση μόνο), `write` (εγγραφή μόνο, τυχόν παλιό αρχείο διαγράφεται), `read-write` (ανάγνωση-εγγραφή), `append` (προσθήκη δεδομένων στο τέλος)...
 - `binary` ή `text mode` (όχι σε όλα τα ΛΣ)
 - Προετοιμασία δομών ΛΣ για επεξεργασία του αρχείου
 - **Υπάρχει και το αντίστοιχο `close`**

Δικαιώματα (Permissions)

- Εισαγωγή
- Συστήματα Αρχείων
- Κατάλογοι
- Αρχεία

- Ποιος χρήστης (ή ομάδα χρηστών) μπορεί να κάνει τι σε ένα αρχείο

- Παράδειγμα: Unix permissions

permissions		user	group					
	drwxr-xr-x	2	mistral	mistral	4096	2007-10-07	21:23	.
	drwxr-xr-x	5	mistral	mistral	4096	2007-10-07	21:05	..
	-rw----r--	1	mistral	mistral	5136	2007-10-07	21:08	banner.gif
	-rw----r--	1	mistral	mistral	4237	2007-10-07	21:23	index.html

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2007-08

Εκτέλεση Προγραμμάτων
(μεταγλωττιστές και περιβάλλον εκτέλεσης)

<http://www.ionio.gr/~mistral/tp/csintro/>

Μ.Στεφανιδάκης

Το Λειτουργικό Σύστημα (ξανά)

- Εισαγωγή

- Ο “γενικός επιβλέπων” ενός υπολογιστικού συστήματος
 - Εκτέλεση προγραμμάτων εφαρμογών
 - Διαχείριση υπολογιστικών πόρων
 - ΚΜΕ
 - Μνήμη
 - Συσκευές E/E
 - Ενδιάμεσο μεταξύ χρήστη και υπολογιστή
 - Κοινές-συχνά χρησιμοποιούμενες λειτουργίες εφαρμογών
 - “Βιβλιοθήκες” λειτουργικού συστήματος

Προγραμματισμός: Πηγαίος Κώδικας

- Εισαγωγή

- Source Code

- Αρχεία Κειμένου

- Γλώσσες υψηλού επιπέδου

- Αποσύνδεση από την αρχιτεκτονική της ΚΜΕ
 - Μεταβλητές - Αντικείμενα
 - Σύνθετες δομές εκτέλεσης
 - Συναρτήσεις – Μέθοδοι
 - Χρήση βιβλιοθηκών

- Απαιτείται μετάφραση (“μεταγλώττιση”) στη γλώσσα που καταλαβαίνει ο υπολογιστής!

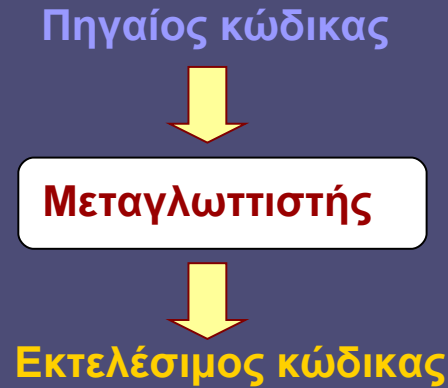
Ο τελικός στόχος

- Εισαγωγή

- **Γλώσσα Μηχανής:** η μοναδική γλώσσα που καταλαβαίνει ο υπολογιστής!
 - Λέξεις δυαδικών ψηφίων
 - 1011 1100 1000 0001 1011 0011 1110 0111
 - ...
 - Προγράμματα σε γλώσσες υψηλού επιπέδου
 - όπως η C, η Java και όχι μόνο αυτές...
 - Πρέπει να μεταγλωττιστούν (μεταφραστούν) σε γλώσσα μηχανής
 - Με **ισοδύναμη λειτουργικότητα (;)**
 - Η μεταγλώττιση δεν είναι τετριμμένη μετατροπή
 - Το παραγόμενο πρόγραμμα έχει ελάχιστη σχέση με το αρχικό!

Η διαδικασία της μεταγλώττισης

- Εισαγωγή
- Μεταγλωττιστές



- Η κλασική εικόνα
 - Ο μεταγλωττιστής (**compiler**) μεταφράζει το πρόγραμμα γλώσσας υψηλού επιπέδου (τυπικά) σε εκτελέσιμη γλώσσα μηχανής
 - Εδώ σταματά ο ρόλος του μεταγλωττιστή
 - Αργότερα, το λειτουργικό σύστημα θα “τρέξει” το εκτελέσιμο πρόγραμμα

Η εξάρτηση από τον επεξεργαστή

- Εισαγωγή
- Μεταγλωττιστές

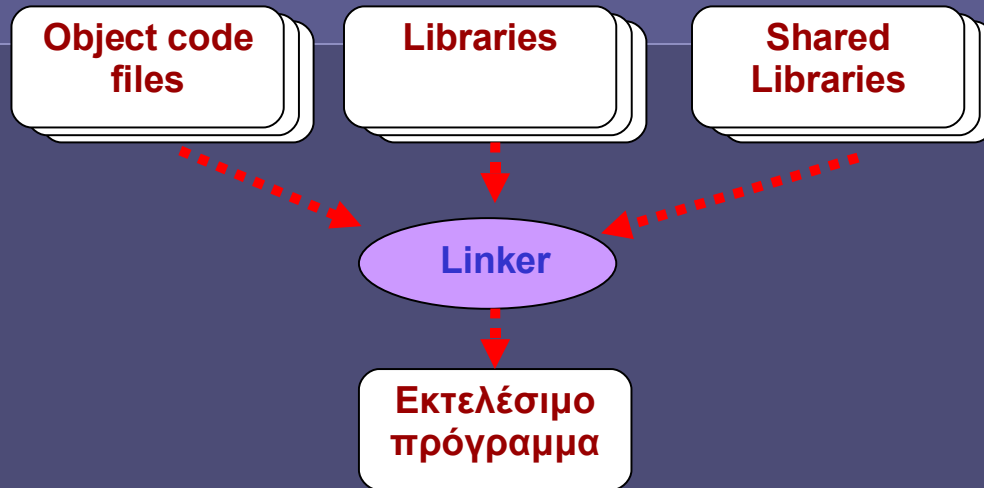
- Ο μεταγλωττιστής πρέπει να γνωρίζει το σετ εντολών
 - Του επεξεργαστή που θα εκτελέσει το τελικό πρόγραμμα
 - Διαθέσιμες εντολές – μέθοδοι προσπέλασης μνήμης
 - Και η κωδικοποίησή τους
 - Διαθέσιμοι καταχωρητές
 - Και οι πιθανές ειδικές λειτουργίες του καθενός
 - Χρήσιμες “συμβάσεις”
 - Πώς καλούνται οι συναρτήσεις;
 - Πώς θα περαστούν οι παράμετροι;
 - Πού αποθηκεύεται η διεύθυνση επιστροφής

Ο Μεταγλωττιστής

- Εισαγωγή
- Μεταγλωττιστές

- Ο μεταγλωττιστής
 - Είναι επίσης ένα εκτελέσιμο πρόγραμμα
 - Προφανώς γραμμένο σε κάποια γλώσσα προγραμματισμού υψηλού επιπέδου!
- Μεταγλώττιση
 - Πρέπει να κατανοηθεί πλήρως (χωρίς ασάφειες) το νόημα του προγράμματος γλώσσας υψηλού επιπέδου
 - Μέσω μιας σειράς φάσεων ανάλυσης και επεξεργασίας του πηγαίου κώδικα

Linkers & Loaders

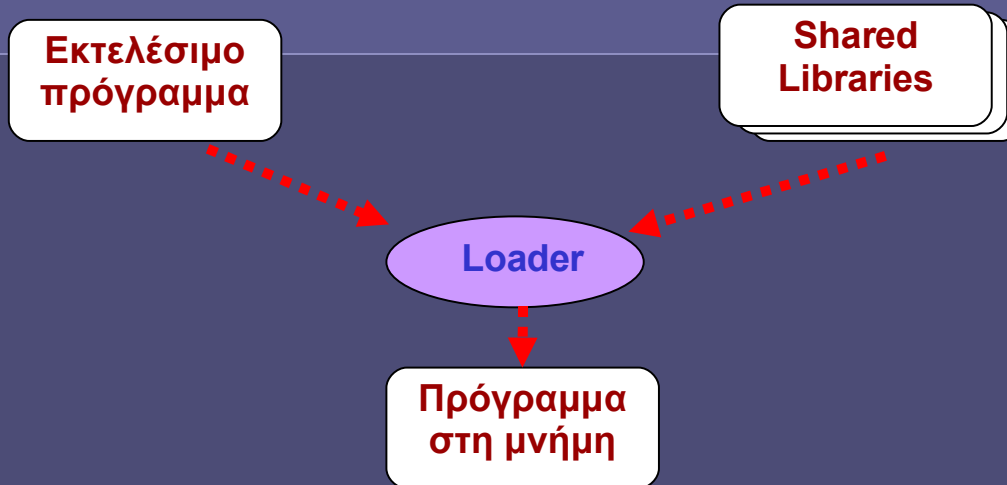


- Εισαγωγή
- Μεταγλωττιστές
- **Linkers & Loaders**

- Η έξοδος από μεταγλωττιστή (object code) περιέχει κενά
 - **Εξωτερικές αναφορές** σε μεταβλητές και συναρτήσεις
- Ο **linker** (συνδέτης;) τροποποιεί object code
 - Σαρώνει όλα τα αρχεία εισόδου (object code)
 - Τοποθετεί τα τμήματα κώδικα-δεδομένων στη σειρά
 - Υπολογίζει τις διευθύνσεις που λείπουν
 - Δημιουργεί πληροφορία για τον **loader**

Linkers & Loaders

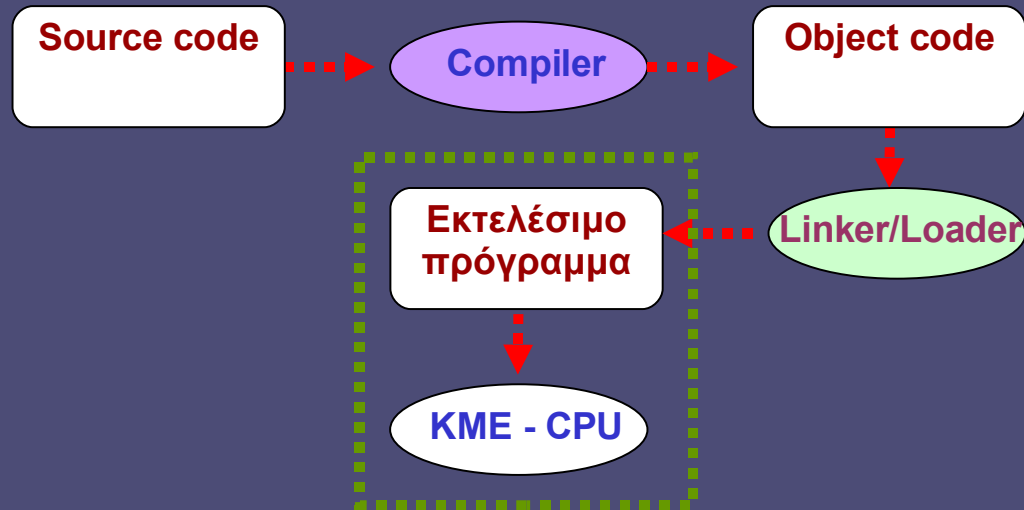
- Εισαγωγή
- Μεταγλωττιστές
- **Linkers & Loaders**



- **Loader** (φορτωτής;)
 - Προετοιμασία για την εκτέλεση του προγράμματος
 - Φόρτωση σελίδων κώδικα-δεδομένων στη μνήμη
 - Συνήθως: απεικόνιση στις σελίδες της εικονικής μνήμης
 - Δυναμική σύνδεση κοινών βιβλιοθηκών
 - Όπου αυτό υποστηρίζεται

Μεταγλωττιζόμενες γλώσσες

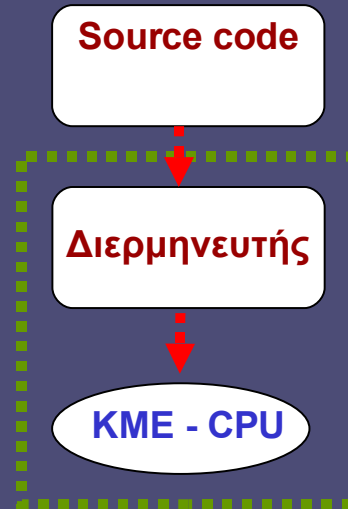
- Εισαγωγή
- Μεταγλωττιστές
- Linkers & Loaders
- Είδη γλωσσών



- **Κατά την εκτέλεση**
 - Το εκτελέσιμο πρόγραμμα έχει τον έλεγχο της ΚΜΕ
 - Οι εντολές του σε γλώσσα μηχανής καθορίζουν τι ακριβώς θα εκτελέσει η ΚΜΕ
 - Ο μεταγλωττιστής δεν παίζει πλέον κανέναν ρόλο

Διερμηνευόμενες γλώσσες

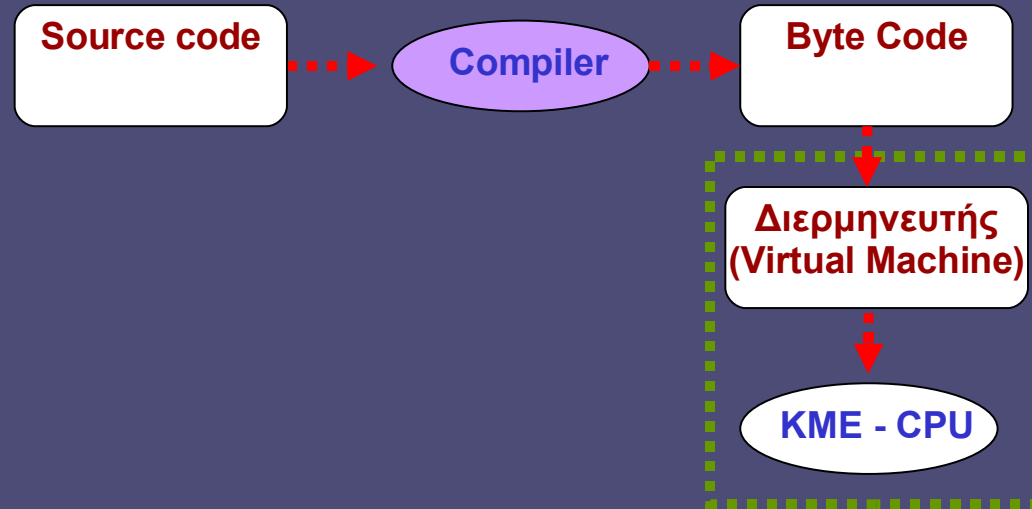
- Εισαγωγή
- Μεταγλωττιστές
- Linkers & Loaders
- Είδη γλωσσών



- Ο “καθαρός” διερμηνευτής (interpreter)
 - Είναι ένα εκτελέσιμο πρόγραμμα
 - Έχει ο ίδιος τον έλεγχο της ΚΜΕ
 - Διαβάζει τον πηγαίο κώδικα (π.χ. γραμμή-προς-γραμμή)
 - Και εκτελεί την ανάλογη λειτουργία
 - Επαναληπτικά μέχρι το τέλος του πηγαίου κώδικα

Διερμηνευόμενες γλώσσες

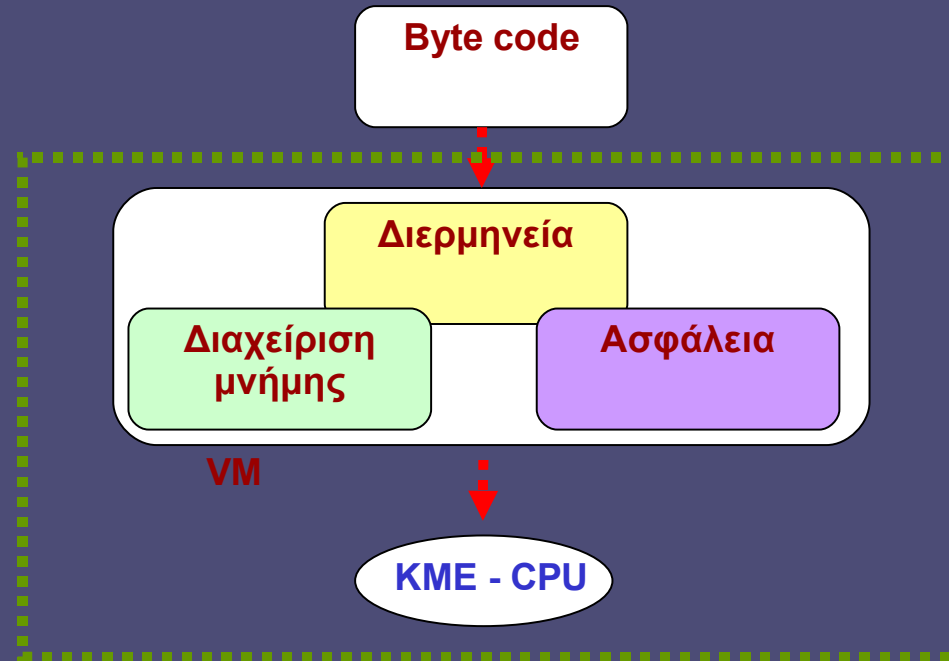
- Εισαγωγή
- Μεταγλωττιστές
- Linkers & Loaders
- Είδη γλωσσών



- Υβριδικός διερμηνευτής
 - Προηγείται μεταγλώττιση σε ενδιάμεση μορφή κώδικα
 - Byte Code
 - Στη συνέχεια η ενδιάμεση μορφή κώδικα διερμηνεύεται από μια **εικονική μηχανή** (VM)
 - Συνήθως σε άλλο υπολογιστικό σύστημα
 - Το σχήμα όλων των σύγχρονων διερμηνευόμενων γλωσσών

Εικονική Μηχανή (Virtual Machine)

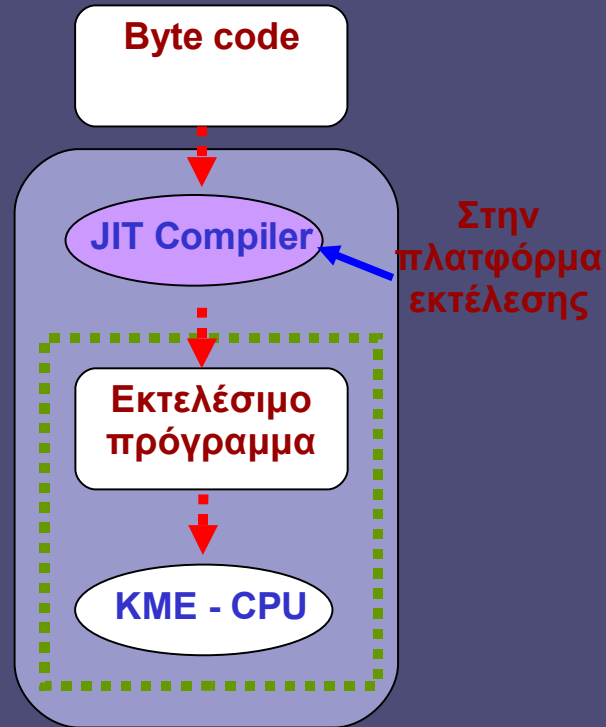
- Εισαγωγή
- Μεταγλωττιστές
- Linkers & Loaders
- Είδη γλωσσών



- Εικονική μηχανή
 - Σύνθετο περιβάλλον εκτέλεσης
 - Ανεξάρτητο αρχιτεκτονικής εκτέλεσης
 - Αλλά: διερμηνεία **αργότερη** της άμεσης εκτέλεσης

Μεταγλωττιστής Just-In-Time (JIT)

- Εισαγωγή
- Μεταγλωττιστές
- Linkers & Loaders
- Είδη γλωσσών



- Για αύξηση της ταχύτητας εκτέλεσης

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2007-08

Ασφάλεια Δεδομένων

<http://www.ionio.gr/~mistral/tp/csintro/>

Μ.Στεφανιδάκης

Οι απειλές

- Εισαγωγή

- Ένας κακόβουλος χρήστης
 - Καταγράφει μηνύματα που ανταλλάσσονται σε ένα “κανάλι” επικοινωνίας και εισάγει νέα μηνύματα
 - Τροποποιεί ξένα δεδομένα
- Και μπορεί
 - Να υποκλέψει κωδικούς πρόσβασης και δεδομένα
 - Να προσποιηθεί ότι είναι κάποιος άλλος
 - Να θέσει υπό έλεγχο επικοινωνία μεταξύ τρίτων
 - Να αποκλείσει τρίτους από τις υπηρεσίες Διαδικτύου
 - κλπ...

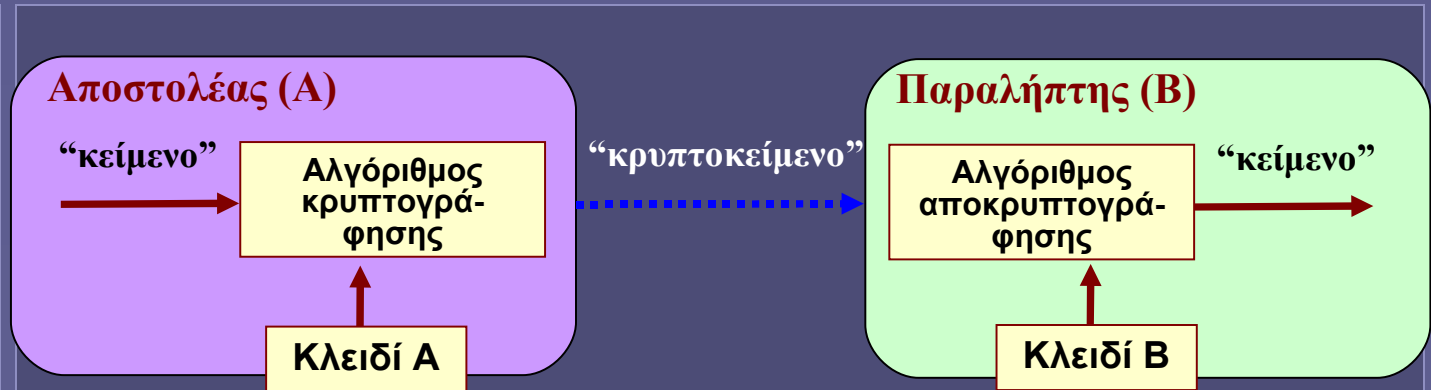
Αρχές ασφάλειας δεδομένων

- Εισαγωγή

- **Απόρρητο (privacy)**
 - Μέσω κρυπτογράφησης
- **Πιστοποίηση αυθεντικότητας**
 - Ταυτότητα χρήστη
 - authentication
 - Δικαιώματα χρήσης δεδομένων
 - authorization
- **Ακεραιότητα δεδομένων (integrity)**
 - Και μη απάρνηση από αποστολέα
 - non-repudiation

Απόρρητο επικοινωνίας

- Εισαγωγή
- Μυστικό κλειδί



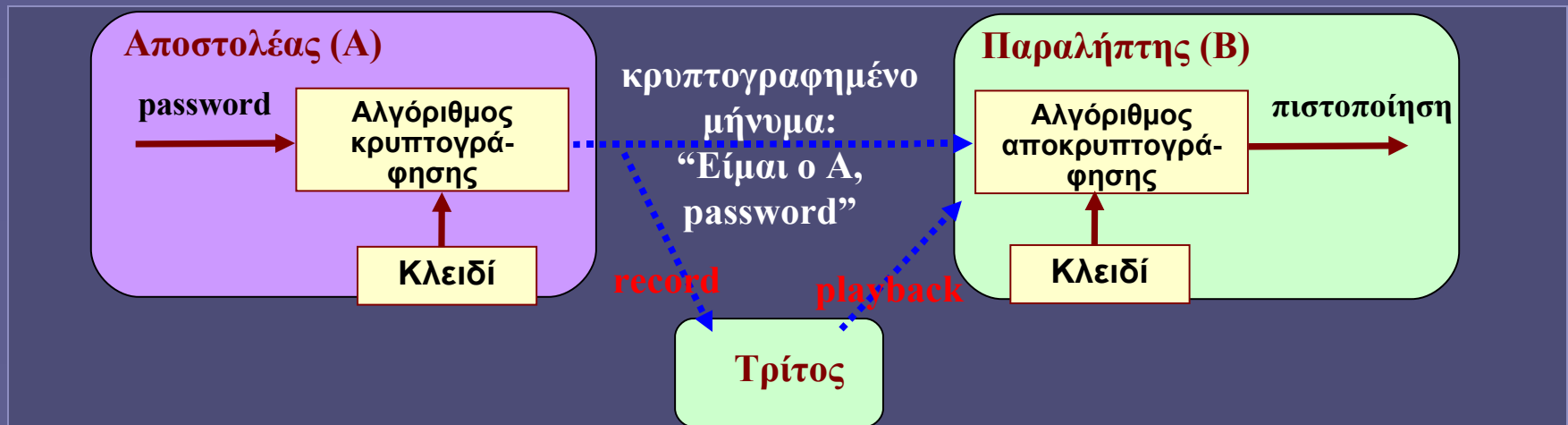
- Εξασφάλιση απορρήτου μέσω κρυπτογράφησης
- Αποστολέας-παραλήπτης
 - Άτομα αλλά και υπολογιστές (π.χ. εξυπηρετητές)
- Αλγόριθμος κρυπτογράφησης
 - Παλαιότερα: κρυφός
 - Σήμερα: ευρέως γνωστός (πρότυπο!) – χρήση κλειδιών

Κρυπτογράφηση με μυστικό (συμμετρικό) κλειδί

- Εισαγωγή
- Μυστικό κλειδί

- **Ίδιο κλειδί αποστολέα (A) – παραλήπτη (B)**
 - Κλειδί A = Κλειδί B
 - Το κλειδί πρέπει να είναι μυστικό!
 - Και μόνο μεταξύ των A και B
 - Συμμετρικοί αλγόριθμοι κρυπτογράφησης και αποκρυπτογράφησης
- **Data Encryption Standard (DES)**
 - 56-bit κλειδί
 - Επαναληπτικά βήματα αντιμετάθεσης bits και πράξεων XOR
 - 3DES, AES (128, 192 ή 256-bit κλειδιά)
- **Το πρόβλημα**
 - η επιλογή και ανταλλαγή του μυστικού κλειδιού μεταξύ A, B
 - Με ποιον τρόπο θα υπάρξει ασφαλής επικοινωνία;

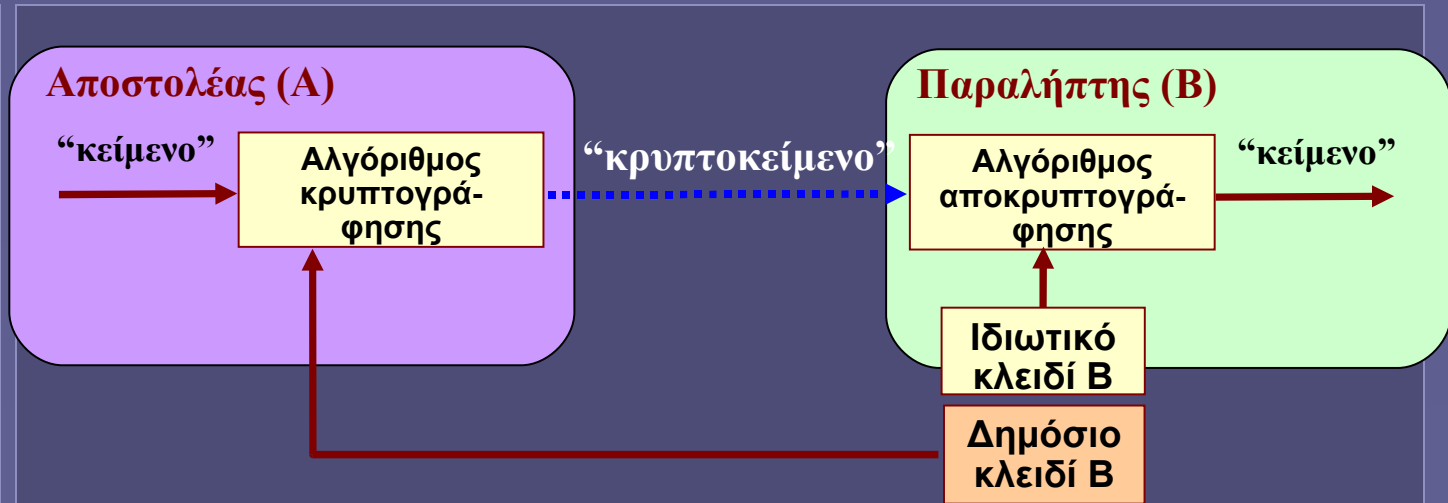
Πιστοποίηση ταυτότητας με μυστικό κλειδί



- Απόδειξη της ταυτότητας του “συνομιλητή”
 - Μέσω κάποιου πρωτοκόλλου πιστοποίησης
 - Π.χ. με αποστολή μέσω δικτύου ενός password
- Η χρήση του μυστικού κλειδιού
 - Κωδικοποίηση password
 - Ο κάτοχος του κλειδιού είναι αυτός που ισχυρίζεται
 - Με μεγάλη βεβαιότητα
 - Όχι απόλυτη όμως!

Κρυπτογράφηση με δημόσιο κλειδί

- Εισαγωγή
- Μυστικό κλειδί
- Δημόσιο κλειδί



- Ζεύγη κλειδιών παραλήπτη (B)
 - Δημόσιο κλειδί: για κρυπτογράφηση (αποστολέας A)
 - Διαθέσιμο σε όλους
 - Ιδιωτικό κλειδί: για αποκρυπτογράφηση (παραλήπτης B)
 - Γνωστό μόνο στον κάτοχό του (B)
- Ό,τι κρυπτογραφείται με το ένα κλειδί, αποκρυπτογραφείται με το άλλο
 - και αντίστροφα

Κρυπτογράφηση με δημόσιο κλειδί

- Εισαγωγή
- Μυστικό κλειδί
- Δημόσιο κλειδί

- **Αλγόριθμος Rivest-Shamir-Adleman (RSA)**
 - Δημιουργία κλειδιών
 - Επιλογή μεγάλων πρώτων αριθμών p, q ($n = pq \geq 768$ bits)
 - Υπολογισμός αριθμών e, d από $pq, (p-1)(q-1)$
 - Δημόσιο κλειδί: (n, e)
 - Ιδιωτικό κλειδί: (n, d)
- **Δεν υπάρχει γνωστός γρήγορος αλγόριθμος για εύρεση p, q από n**
 - **Εδώ στηρίζεται η ασφάλεια του RSA**
- **Κρυπτογράφηση**
 - Έστω “κείμενο” m
 - “κρυπτοκείμενο” $c = m^e \bmod n$ (κρυπτογράφηση)
 - “κείμενο” $m = c^d \bmod n$ (αποκρυπτογράφηση)
 - Η διαδικασία επιτυγχάνεται και με αντιστροφή κλειδιών!

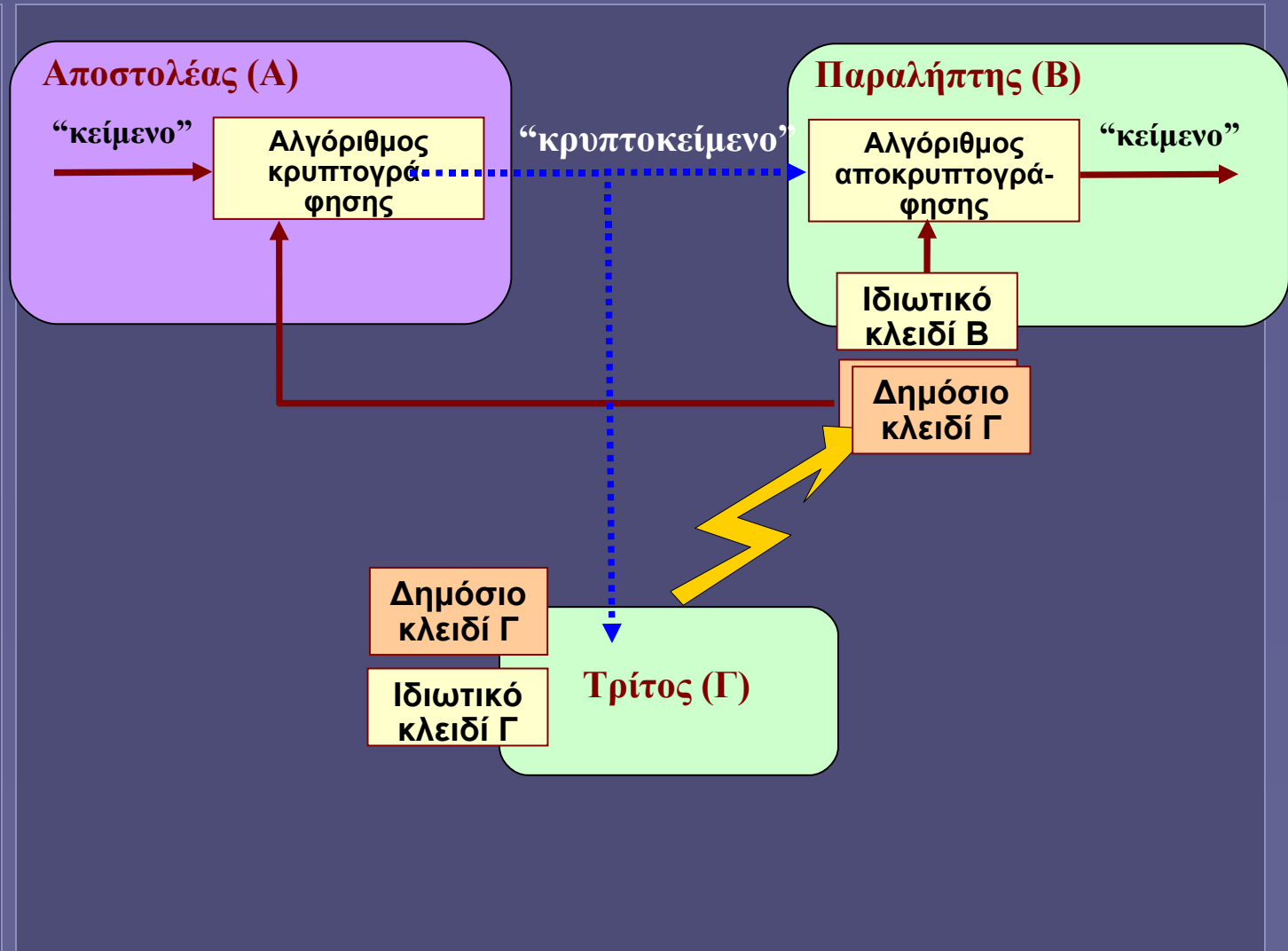
Κρυπτογράφηση με δημόσιο κλειδί

- Εισαγωγή
- Μυστικό κλειδί
- Δημόσιο κλειδί

- **Χρονοβόρα διαδικασία κρυπτογράφησης – αποκρυπτογράφησης με δημόσιο κλειδί**
 - Με μυστικό (συμμετρικό) κλειδί: τουλάχιστον 100 φορές γρηγορότερα!
- **Συνήθως**
 - Πρώτα ανταλλαγή μυστικού κλειδιού με RSA
 - Στη συνέχεια επικοινωνία με (από)κρυπτογράφηση με το μυστικό κλειδί
- **Ποιος εγγυάται τη διανομή του δημόσιου κλειδιού;**
 - Σε αντίθεση με το μυστικό κλειδί: εδώ ο καθένας μπορεί να ισχυριστεί ότι είναι ο οποιοσδήποτε!
 - Και να στείλει το δημόσιο κλειδί του αντί του σωστού... !

Πιστοποίηση δημόσιου κλειδιού;

- Εισαγωγή
- Μυστικό κλειδί
- Δημόσιο κλειδί



Ψηφιακή υπογραφή

- Εισαγωγή
- Μυστικό κλειδί
- Δημόσιο κλειδί
- Ψηφιακή υπογραφή

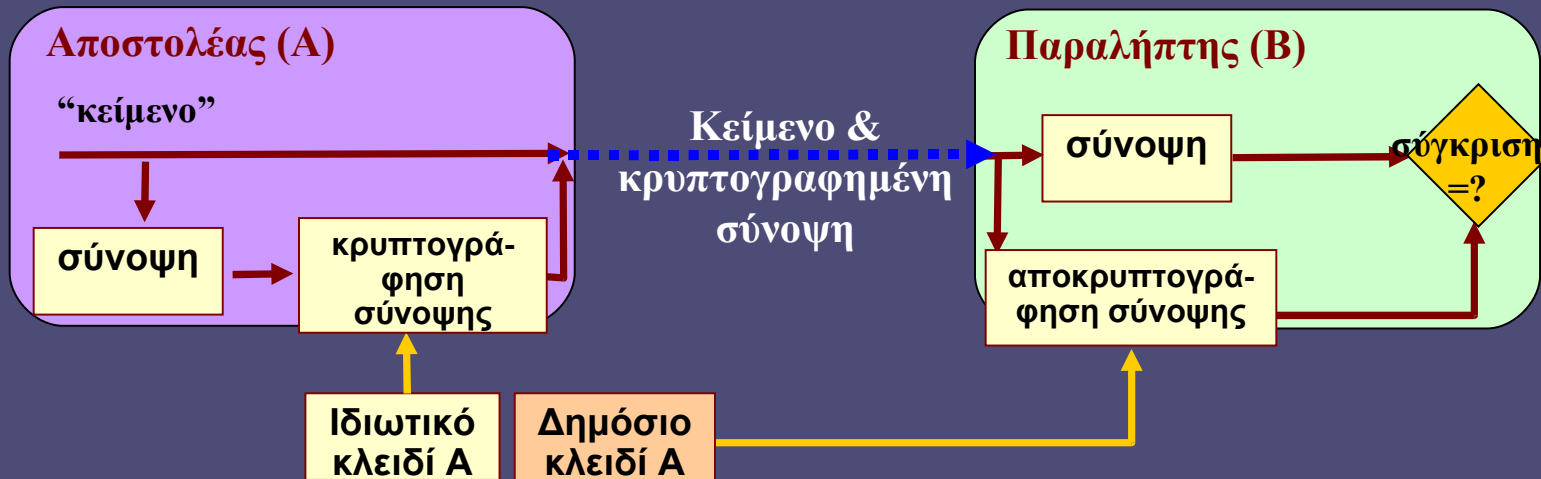
- **Κρυπτογραφική τεχνική για εξασφάλιση:**
- **Ακεραιότητας**
 - Αποφυγή αλλοίωσης δεδομένων
- **Αυθεντικότητας**
 - Ποιος είναι ο δημιουργός-αποστολέας;
- **Μη απάρνησης**
 - Από την πλευρά του αποστολέα
- **Προσοχή: η ψηφιακή υπογραφή δεν εξασφαλίζει το απόρρητο!**

Τεχνική ψηφιακής υπογραφής

- Εισαγωγή
- Μυστικό κλειδί
- Δημόσιο κλειδί
- Ψηφιακή υπογραφή

- Χρήση μεθόδου κρυπτογράφησης δημόσιου κλειδιού
- **Αντίστροφα** με διαδικασία τήρησης απορρήτου:
 - Ο αποστολέας κρυπτογραφεί το κείμενο με το ιδιωτικό του κλειδί
 - Παραγωγή ψηφιακής υπογραφής, αποστολή **μαζί με αρχικό κείμενο**
 - Ο παραλήπτης αποκρυπτογραφεί την ψηφιακή υπογραφή με το δημόσιο κλειδί του αποστολέα
 - Μόνο με το κλειδί αυτό λαμβάνεται το αρχικό κείμενο
 - Και πιστοποιείται ο αποστολέας

Συνοψεις (digests)



- Για την αποφυγή κρυπτογράφησης όλων των δεδομένων
 - Δημιουργία σύνοψης
 - Αλγόριθμοι κατακερματισμού
 - (Σχεδόν) αδύνατο διαφορετικό μήνυμα να δώσει την ίδια σύνοψη
 - MD5 (128-bit), SHA-1 (160-bit)
 - Κρυπτογράφηση της σύνοψης μόνο με ιδιωτικό κλειδί αποστολέα

Υποδομή Δημόσιου Κλειδιού (Public Key Infrastructure – PKI)

- Εισαγωγή
- Μυστικό κλειδί
- Δημόσιο κλειδί
- Ψηφιακή υπογραφή
- Πιστοποίηση δημόσιων κλειδιών

- **Έμπιστοι ενδιάμεσοι: αρχές πιστοποίησης**
 - Certification Authorities (CAs)
- **Πιστοποίηση ταυτότητας ενός “συνομιλητή”**
 - Πιστοποίηση εταιριών ηλεκτρονικού εμπορίου
 - Σύνδεση ενός δημόσιου κλειδιού με μια “οντότητα”
 - π.χ. διεύθυνση IP, εταιρία...
 - **Η αξιοπιστία της πιστοποίησης εξαρτάται από την αξιοπιστία της CA!**
- **Υποδομή Δημόσιου Κλειδιού**
 - Αλυσίδα CAs
 - Root CA

Πιστοποιητικά (Certificates)

- Εισαγωγή
- Μυστικό κλειδί
- Δημόσιο κλειδί
- Ψηφιακή υπογραφή
- Πιστοποίηση δημόσιων κλειδιών

- **Πιστοποιητικά (certificates)**
 - Περιέχουν πληροφορία για έναν χρήστη
 - Και το δημόσιο κλειδί του
 - για τα οποία εγγυάται κάποια CA
 - υπογράφοντας ψηφιακά το πιστοποιητικό με το **ιδιωτικό κλειδί** της

Η CA: XYZ
βεβαιώνει ότι ο: A
έχει δημόσιο κλειδί: ΔK(A)
<ψηφιακή υπογραφή από IK(CA)>

Αλυσίδα CA

- Εισαγωγή
- Μυστικό κλειδί
- Δημόσιο κλειδί
- Ψηφιακή υπογραφή
- Πιστοποίηση δημόσιων κλειδιών

Η CA1: XYZ
βεβαιώνει ότι ο: XYZ
έχει δημόσιο κλειδί: $\Delta K(CA1)$
<ψηφιακή υπογραφή από $IK(CA1)$ >

root CA

Η CA1: XYZ
βεβαιώνει ότι ο: MNO
έχει δημόσιο κλειδί: $\Delta K(CA2)$
<ψηφιακή υπογραφή από $IK(CA1)$ >

Η CA2: MNO
βεβαιώνει ότι ο: A
έχει δημόσιο κλειδί: $\Delta K(A)$
<ψηφιακή υπογραφή από $IK(CA2)$ >

Παράδειγμα χρήσης Certificates

- Εισαγωγή
- Μυστικό κλειδί
- Δημόσιο κλειδί
- Ψηφιακή υπογραφή
- Πιστοποίηση δημόσιων κλειδιών



This certificate has been verified for the following uses:

SSL Server Certificate

Issued To

Common Name (CN) www.amazon.com
Organization (O) Amazon.com Inc.
Organizational Unit (OU) <Not Part Of Certificate>
Serial Number 48:1B:72:10:BC:80:55:E4:76:EF:28:AB:7F:9B:24:B4

Issued By

Common Name (CN) VeriSign Class 3 Secure Server CA
Organization (O) VeriSign, Inc.
Organizational Unit (OU) VeriSign Trust Network

Validity

Issued On 09/18/2007
Expires On 09/18/2008

Fingerprints

SHA1 Fingerprint A6:3A:B8:E8:BA:74:CA:F9:34:66:E2:34:08:31:36:5A:AC:4E:1A:D6
MD5 Fingerprint CE:0C:D0:84:75:74:0C:0C:C1:06:98:A6:C5:3C:EB:40

www.amazon.com



Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2006-07

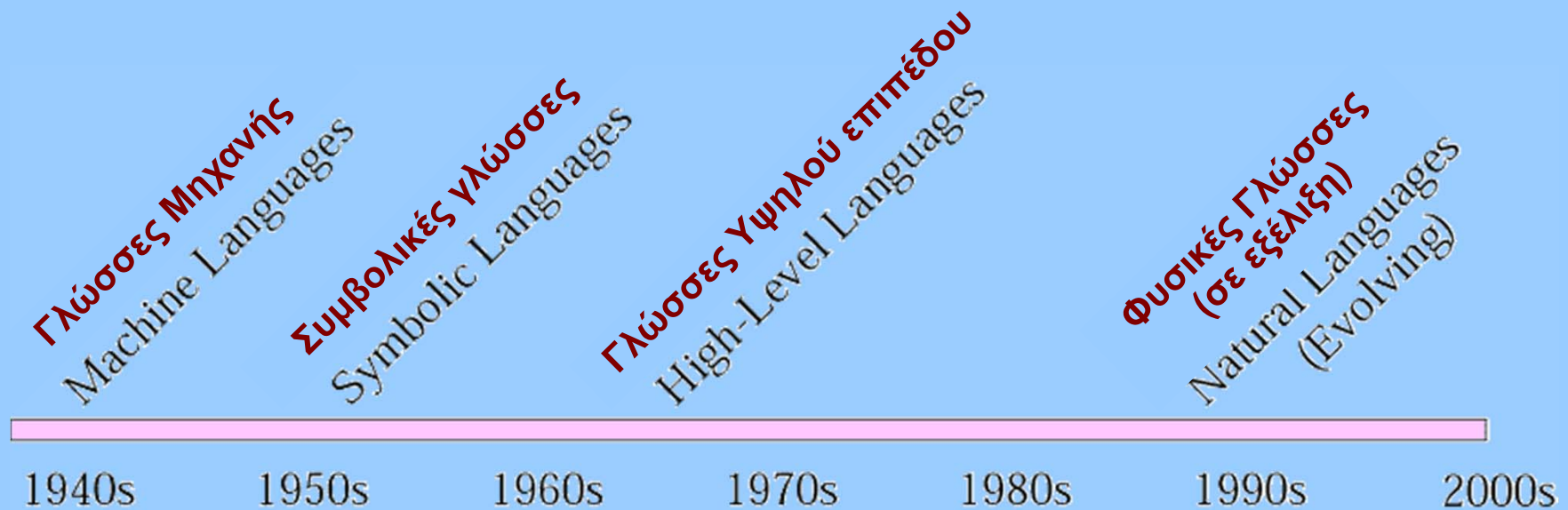
Γλώσσες προγραμματισμού

Μ.Στεφανιδάκης

Στόχοι του μαθήματος

- Η εξέλιξη των γλωσσών προγραμματισμού
- Η γλώσσα μηχανής, η γλώσσα assembly, γλώσσες υψηλότερου επιπέδου
- Τα βήματα δημιουργίας και εκτέλεσης ενός προγράμματος
- Οι κατηγορίες γλωσσών προγραμματισμού
 - Διαδικασιακές (procedural)
 - Αντικειμενοστραφείς (object oriented)
 - Συναρτησιακές (functional)
 - Δηλωτικές (declarative)
 - Άλλες γλώσσες
- Βασικά χαρακτηριστικά γλωσσών προγραμματισμού

Εξέλιξη των γλωσσών προγραμματισμού



- **Γλώσσα προγραμματισμού**
 - Για τη δημιουργία **εκτελέσιμων** προγραμμάτων
 - Προκαθορισμένες λέξεις και κανόνες σύνταξης

Γλώσσα μηχανής

1	00000000	00000100	000000000000000000
2	01011110	00001100	11000010000000000000010
3		11101111	000101100000000000000101
4		11101111	10011110 00000000000001011
5	11111000	10101101	11011111 0000000000010010
6		01100010	11011111 0000000000010101
7	11101111	00000010	11111011 0000000000010111
8	11110100	10101101	11011111 0000000000011110
9	00000011	10100010	11011111 0000000000100001
10	11101111	00000010	11111011 0000000000100100
11	01111110	11110100	10101101
12	11111000	10101110	110001010000000000101011
13	00000110	10100010	11111011 0000000000110001
14	11101111	00000010	11111011 0000000000110100
15			00000100 0000000000111101
16			00000100 0000000000111101

- Η μόνη “γλώσσα” που καταλαβαίνει ο υπολογιστής
 - Οδήγηση εσωτερικών κυκλωμάτων (τρανζίστορ) με 0 και 1
 - **Κάθε διαφορετική ΚΜΕ (CPU) έχει τη δική της γλώσσα μηχανής!**

Συμβολικές γλώσσες

1	Entry	main,<r2>
2	subl2	#12,sp
3	jsb	C\$MAIN_ARGS
4	movab	\$CHAR_STRING_CON
5		
6	pushal	-8(fp)
7	pushal	(r2)
8	calls	#2,read
9	pushal	-12(fp)
10	pushal	3(r2)
11	calls	#2,read
12	mull3	-8(fp),-12(fp),-
13	pushal	6(r2)
14	calls	#2,print
15	clrl	r0
16	ret	

- Τα πρώτα προγράμματα
 - Σύμβολα για την αναπαράσταση των εντολών μηχανής
 - Ποιος κάνει τη μετάφραση;

Ο συμβολομεταφραστής (assembler)

- Μετατροπή της συμβολικής γλώσσας σε γλώσσα μηχανής
- Η διαδικασία μετατροπής: **assembly**
 - Γλώσσες “assembly”
- Ένα επίπεδο πάνω από τη γλώσσα μηχανής
- Αντιστοιχία γλώσσας assembly και αρχιτεκτονικής ΚΜΕ
 - *Κάθε (διαφορετικής αρχιτεκτονικής) ΚΜΕ προγραμματίζεται στη δική της γλώσσα assembly*

Γλώσσες υψηλού επιπέδου

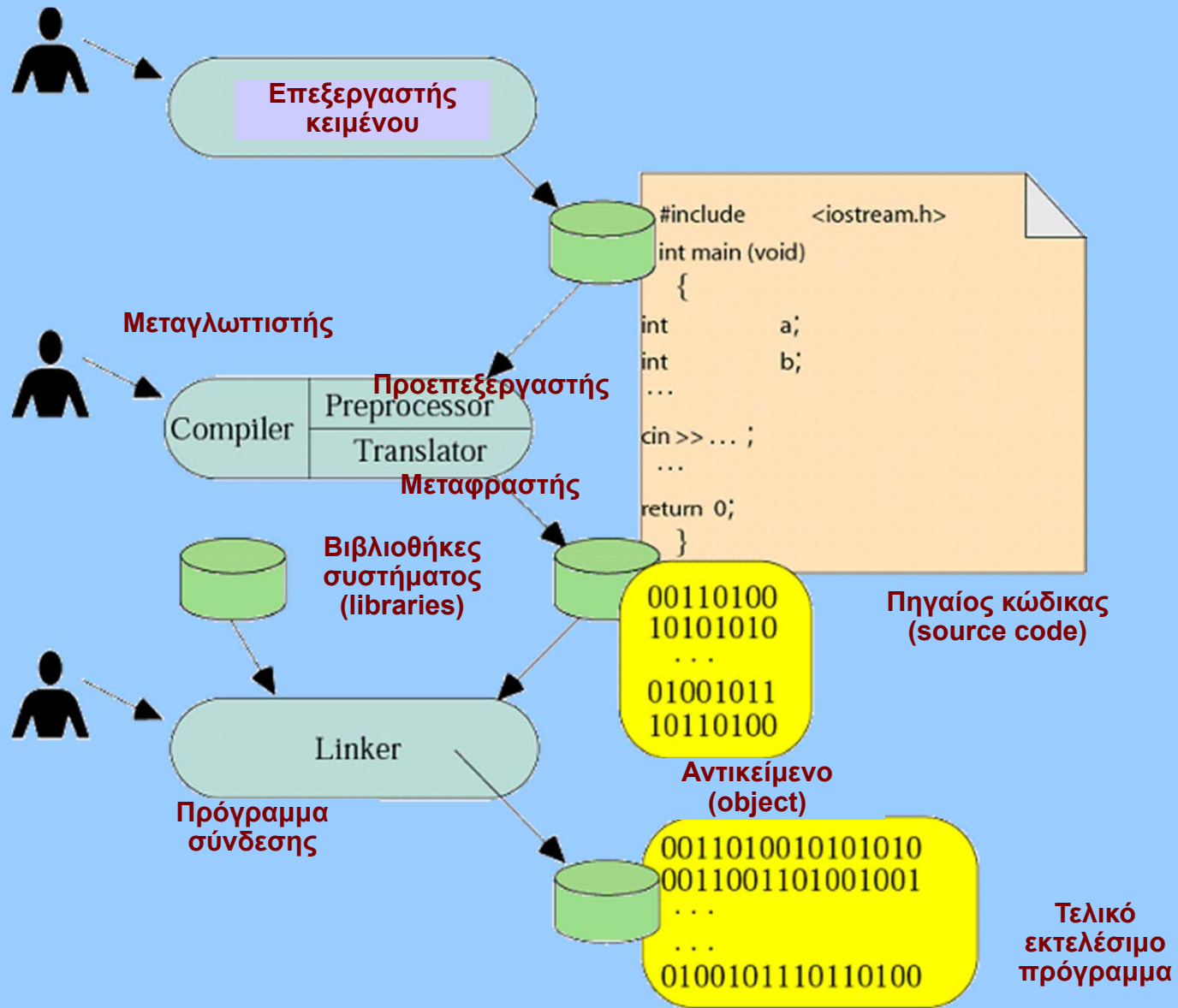
- Συγγραφή προγραμμάτων σε assembly
 - Μη αποδοτική
 - Μακριά από τον υλοποιούμενο αλγόριθμο
 - **Εξαρτημένη από την αρχιτεκτονική του υπολογιστή**
- Γλώσσες υψηλότερου επιπέδου
 - Προγράμματα **ανεξάρτητα** αρχιτεκτονικής ΚΜΕ
 - Απαλλαγή από λεπτομέρειες συμβολικής γλώσσας
 - Δεν απαιτείται η πλήρης γνώση της αρχιτεκτονικής
 - **Τι συμβαίνει στα προγράμματα υψηλών απαιτήσεων;**
 - Πρέπει επίσης να μεταφραστούν σε γλώσσα μηχανής
 - **Μεταγλώττιση (compilation)**

Πρόγραμμα σε C++

```
1  /* This program reads two integer numbers from the
2     keyboard and prints their product.
3  */
4  #include <iostream.h>
5
6  int main (void)
7  {
8     // Local Declarations
9     int number1;
10    int number2;
11    int result;
12    // Statements
13    cin >> number1;
14    cin >> number2;
15    result = number1 * number2;
16    cout << result;
17    return 0;
18 } // main
```

- Η ίδια λειτουργία με τα προηγούμενα προγράμματα

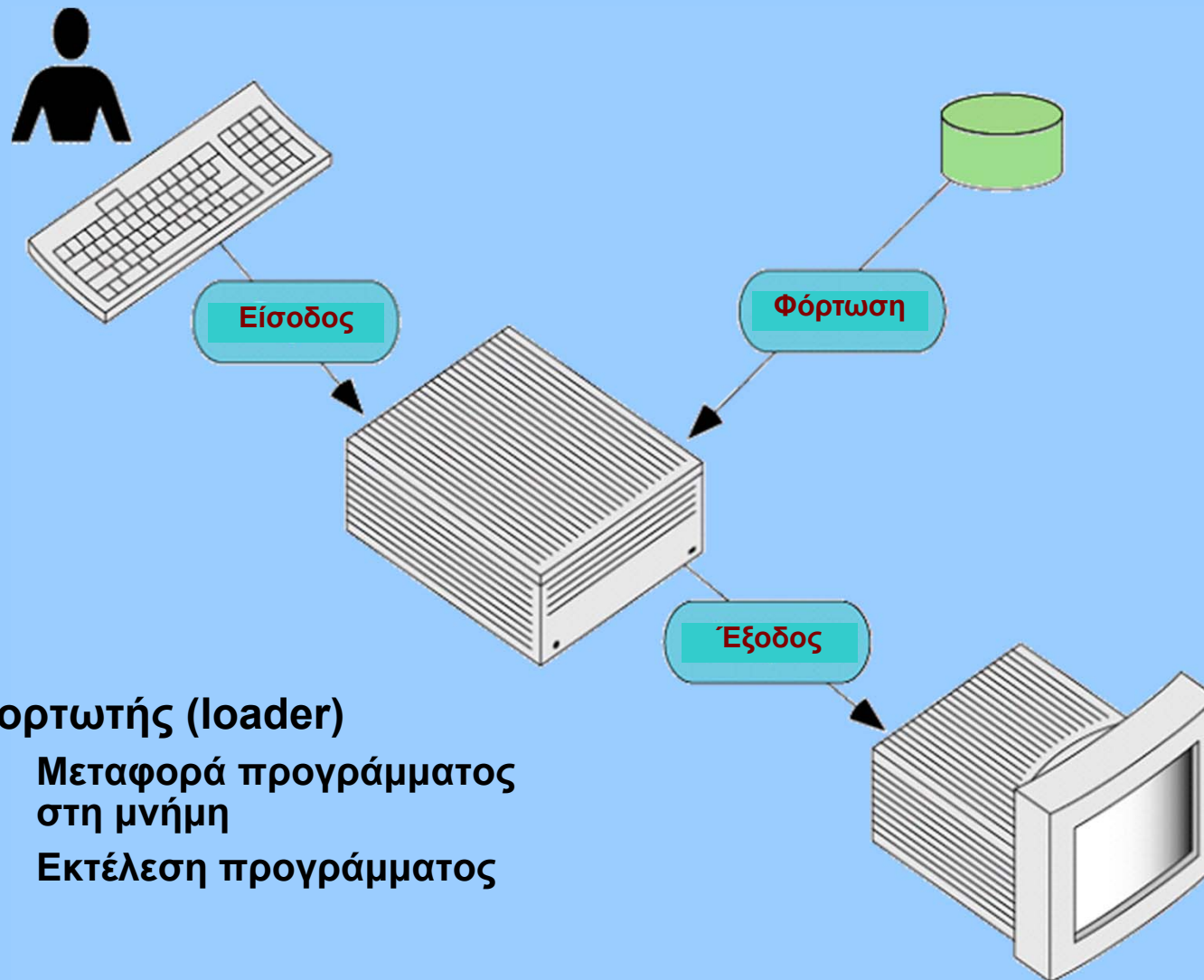
Δημιουργώντας ένα εκτελέσιμο πρόγραμμα



Προγράμματα σύνδεσης (linkers)

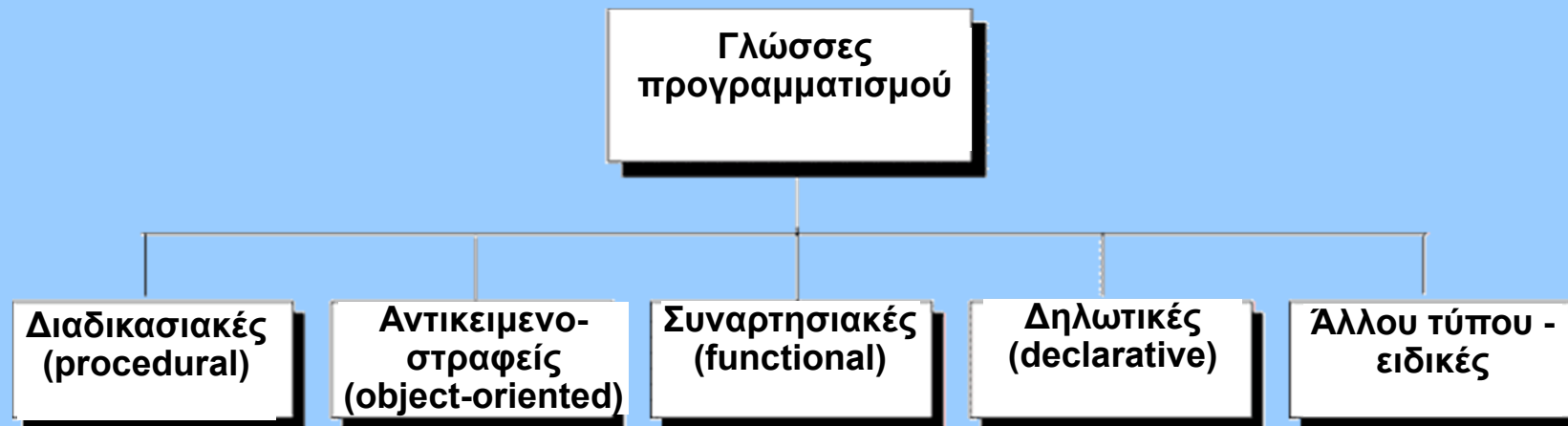
- Υποπρογράμματα
 - Σε διαφορετικά αρχεία – αντικείμενα
 - Προσάρτηση στο κυρίως πρόγραμμα
- Βιβλιοθήκες συστήματος
 - Κλήση συναρτήσεων
 - Εύρεση διευθύνσεων
- Εξωτερικές μεταβλητές
- Τελικό εκτελέσιμο αρχείο
 - Ένα βήμα πριν την εκτέλεση του προγράμματος

Εκτελώντας ένα πρόγραμμα



- **Φορτωτής (loader)**
 - Μεταφορά προγράμματος στη μνήμη
 - Εκτέλεση προγράμματος

Κατηγορίες γλωσσών προγραμματισμού



- Ανάλογα με την προσέγγιση του προβλήματος προς επίλυση
- Στο τέλος όλες καταλήγουν σε γλώσσα μηχανής!
- Επίσης: **διερμηνευόμενες** γλώσσες

Διαδικασιακές γλώσσες (procedural)

- Η παραδοσιακή μέθοδος προγραμματισμού
 - Σύνολο εντολών
 - Διαδικασία επίλυσης
 - Κάθε εντολή = “προσταγή” προς το σύστημα
 - Προστακτικές (imperative) γλώσσες
- Διαδικασιακές γλώσσες:
 - FORTRAN (1957)
 - για μαθηματικές πράξεις
 - COBOL (1960)
 - για εμπορικές επιχειρήσεις – βάσεις δεδομένων
 - Pascal (1971)
 - Δομημένος προγραμματισμός
 - C (1971-1973)
 - Δημοφιλής γλώσσα – αρχικά για προγραμματισμό UNIX
 - Ada (1984)
 - Για ασφαλή συστήματα – επεξεργασία σε πραγματικό χρόνο

Πρόγραμμα σε FORTRAN

```
PROGRAM area  
REAL base, height, area  
PRINT *, 'Enter the values for the base and height of a triangle.'  
READ *, base, height  
area = (1.0/2.0) * base * height  
PRINT *, 'The area of a triangle with base ', base  
PRINT *, 'and height ', height, ' is ', area  
STOP  
END
```

- Υπολογισμός εμβαδού τριγώνου

Πρόγραμμα σε COBOL

```
$ SET SOURCEFORMAT"FREE"
```

```
IDENTIFICATION DIVISION.
```

```
PROGRAM-ID. Multiplier.
```

```
AUTHOR. Michael Coughlan.
```

```
* Example program using ACCEPT, DISPLAY and MULTIPLY to
```

```
* get two single digit numbers from the user and multiply them together
```

```
DATA DIVISION.
```

```
WORKING-STORAGE SECTION.
```

```
01 Num1 PIC 9 VALUE ZEROS.
```

```
01 Num2 PIC 9 VALUE ZEROS.
```

```
01 Result PIC 99 VALUE ZEROS.
```

```
PROCEDURE DIVISION.
```

```
DISPLAY "Enter first number (1 digit) : " WITH NO ADVANCING.
```

```
ACCEPT Num1.
```

```
DISPLAY "Enter second number (1 digit) : " WITH NO ADVANCING.
```

```
ACCEPT Num2.
```

```
MULTIPLY Num1 BY Num2 GIVING Result.
```

```
DISPLAY "Result is = ", Result.
```

```
STOP RUN.
```

Πρόγραμμα σε Pascal

```
program hello(output);

  procedure WriteResponseHeader;
  begin
    writeln('content-type: text/html');
    writeln
  end;

begin
  WriteResponseHeader;
  writeln('<HTML>');
  writeln('<HEAD>');
  writeln('<TITLE>IriePascal Hello World Program</TITLE>');
  writeln('</HEAD>');
  writeln('<BODY>');
  writeln('<BIG> Hello world!!! </BIG>');
  writeln('</BODY>');
  writeln('</HTML>')
end.
```

Πρόγραμμα σε Ada95

```
-----  
-- (C) Michael A Smith 1993-2000          --  
-- Taken from the book Object-oriented software in Ada 95 --  
-- See http://www.brighton.ac.uk/ada95/home.htm --  
-----
```

```
with Ada.Text_Io, Ada.Command_Line, Class_Line;  
use  Ada.Text_Io, Ada.Command_Line, Class_Line;  
procedure Main is  
  I_Fd : Ada.Text_Io.File_Type;    -- File descriptor  
  O_Fd : Ada.Text_Io.File_Type;    -- File descriptor  
  A_Line : Class_Line.Line;  
begin  
  if Argument_Count >= 1 then  
    for I in 1 .. Argument_Count loop -- Repeat for each file  
      begin  
        Open( File=>I_Fd, Mode=>In_File, -- Open file  
              Name=>Argument(I) );  
        while not End_Of_File(I_Fd) loop -- For each Line  
          Get_Line( A_Line, I_Fd );  
          case Status(A_Line) is  
            when Text_Line =>      -- Write to file  
              Put_Line( A_Line, O_Fd );  
            when File_Name =>      -- Get file name  
              Get_Fd( A_Line, O_Fd );  
            when Unknown =>        -- Ignore  
              null;
```

Αντικειμενοστραφείς γλώσσες (object-oriented)

- Αντικείμενα συνδεδεμένα με τις λειτουργίες τους
 - **Ενθυλάκωση (encapsulation)**
 - Απόκρυψη δεδομένων και ιδιοτήτων αντικειμένων
 - **Κληρονομικότητα (inheritance)**
 - Μεταβίβαση ιδιοτήτων από μητρικές κλάσεις αντικειμένων
 - **Πολυμορφισμός**
 - Λειτουργίες με το ίδιο όνομα εκτελούν διαφορετικά πράγματα στις κλάσεις τους
- Αντικειμενοστραφείς γλώσσες:
 - C++
 - Java (διερμηνευόμενη)

Συναρτησιακές Γλώσσες (functional)

- Το πρόγραμμα ως μαθηματική συνάρτηση
- Συναρτήσεις: αντιστοίχιση λίστας δεδομένων εισόδου σε λίστα δεδομένων εξόδου
- LISP

```
;;; returns a list of the actual viewport corners in WCS
(defun ZOOMPTS (/ ctr h screen ratio size size_2)
  (setq ctr (xy-of (getvar "VIEWCTR"))) ;3D -> 2D
        h (getvar "VIEWSIZE") ;real
        screen (getvar "SCREENSIZE") ;2D: Pixel x,y
        ratio (/ (float (car screen)) ;aspect ratio
                 (cadr screen))
        size (list (* h ratio) h) ;screensize in coords
        size_2 (mapcar '/ size '(2.0 2.0)))
  (list (mapcar '- ctr size_2)
        (mapcar '+ ctr size_2)))
(defun XY-OF (pt) (list (car pt)(cadr pt))) ;assure 2D coords
```


Δηλωτικές γλώσσες (declarative)

- Αρχές λογικού συλλογισμού
 - Προτάσεις (αληθείς ή ψευδείς)
 - Κανόνες λογικής
 - Επαγωγή νέων προτάσεων
- Prolog

```
parent(adam,peter). % means adam is parent of peter
parent(eve,peter).
parent(adam,paul).
parent(marry,paul).
```

```
father(F,C):-man(F),parent(F,C).
mother(M,C):-woman(M),parent(M,C).
```

```
is_father(F):-father(F,_).
is_mother(M):-mother(M,_).
```

```
?-father(X,paul)
```

Άλλες γλώσσες: HTML

```
<HTML>
  <HEAD>
    <TITLE> Sample Document </TITLE>
  </HEAD>
  <BODY>
    This is the picture of a book:
    <IMG SRC="Pictures/book1.gif" ALIGN="MIDDLE" >
  </BODY>
</HTML>
```

- Για περιγραφή ιστοσελίδων
 - Εντολές μορφοποίησης και απεικόνισης σε browser

Βασικά χαρακτηριστικά (διαδικασιακών) γλωσσών προγραμματισμού

- **Αναγνωριστικό (identifier)**
 - Για την ονομασία δεδομένων και άλλων αντικειμένων
 - **Συμβολική αναφορά σε θέσεις μνήμης**
 - Τοποθέτηση κατά τη μεταγλώττιση
 - Κάθε γλώσσα έχει κανόνες συγγραφής των αναγνωριστικών
 - Απαγορεύεται η χρήση των δεσμευμένων λέξεων της γλώσσας
 - **Reserved words - keywords**

Βασικά χαρακτηριστικά (διαδικασιακών) γλωσσών προγραμματισμού

- **Τύποι δεδομένων**
 - **Ανάλογα με γλώσσα**
 - **Βασικοί τύποι**
 - Ακέραιοι, κινητής υποδιαστολής, χαρακτήρες..
 - **Ακολουθίες χαρακτήρων (strings)**
 - Ιδιαίτερη αντιμετώπιση σε όλες τις γλώσσες
 - **Σύνθετοι τύποι**
 - Δείκτες, δομές...

Βασικά χαρακτηριστικά (διαδικασιακών) γλωσσών προγραμματισμού

- **Μεταβλητές**
 - Βασικές οντότητες γλώσσας
 - Αναγνωριστικό
 - Αναφορά σε θέση (θέσεις) μνήμης
 - Κατά τη μεταγλώττιση
 - Δήλωση – Αρχικοποίηση
 - Στη συγγραφή – στη μεταγλώττιση ή κατά την εκτέλεση
 - **Σφαιρικές (global) – τοπικές (local)**
 - Περιβάλλον εκτέλεσης (context)

Βασικά χαρακτηριστικά (διαδικασιακών) γλωσσών προγραμματισμού

- **Τελεστές (operators)**
 - Αριθμητικοί (π.χ. +, -, *, / ...)
 - Σχισιακοί (σύγκρισης) (π.χ. <, >, ==, != ...)
 - Λογικοί (π.χ. &&, || ...)
 - Ανάθεσης (π.χ. = ...)
- Δρουν σε έναν ή περισσότερους τελεστέους (operands)
- Κάθε γλώσσα ορίζει την προτεραιότητα των τελεστών
 - $x+y*z/8&&!a$ (;)

Βασικά χαρακτηριστικά (διαδικασιακών) γλωσσών προγραμματισμού

- **Εντολές**
 - Παραστάσεις
 - Σύνθετες εντολές
 - Εντολές επιλογής (συνθήκες)
 - Επαναληπτικές εντολές
 - Εντολές μετάβασης

Βασικά χαρακτηριστικά (διαδικασιακών) γλωσσών προγραμματισμού

- **Συναρτήσεις (υποπρογράμματα)**
 - Δήλωση συνάρτησης
 - Παράμετροι εισόδου και εξόδου
 - Κλήση συνάρτησης
 - Μεταβίβαση παραμέτρων
 - Κατ'αξία (by value)
 - Κατ'αναφορά (by reference)
 - Side-effects
 - Αλλαγή κατάστασης αντικειμένων εκτός συνάρτησης

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2006-07

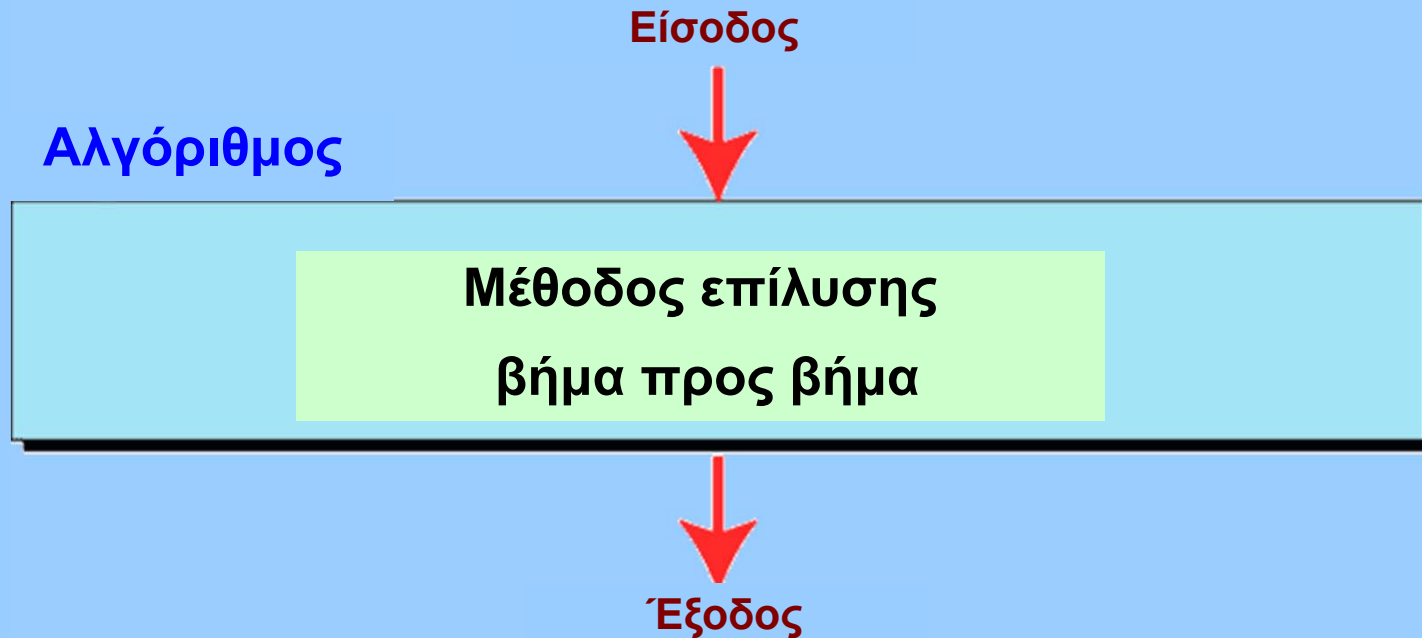
Εισαγωγή στους Αλγορίθμους

Μ.Στεφανιδάκης

Στόχοι του μαθήματος

- Η έννοια του **αλγορίθμου**
- 3 βασικές αλγοριθμικές δομές
 - **Ακολουθία** (sequence)
 - **Απόφαση** (decision)
 - **Επανάληψη** (repetition)
- **Αναπαραστάσεις αλγορίθμων**
 - **Ψευδοκώδικας**
 - **Διαγράμματα ροής**
- **Υποπρογράμματα**
- **Βασικοί αλγόριθμοι**
 - **Ταξινόμηση** (sorting)
 - **Αναζήτηση** (searching)

Τι είναι αλγόριθμος;

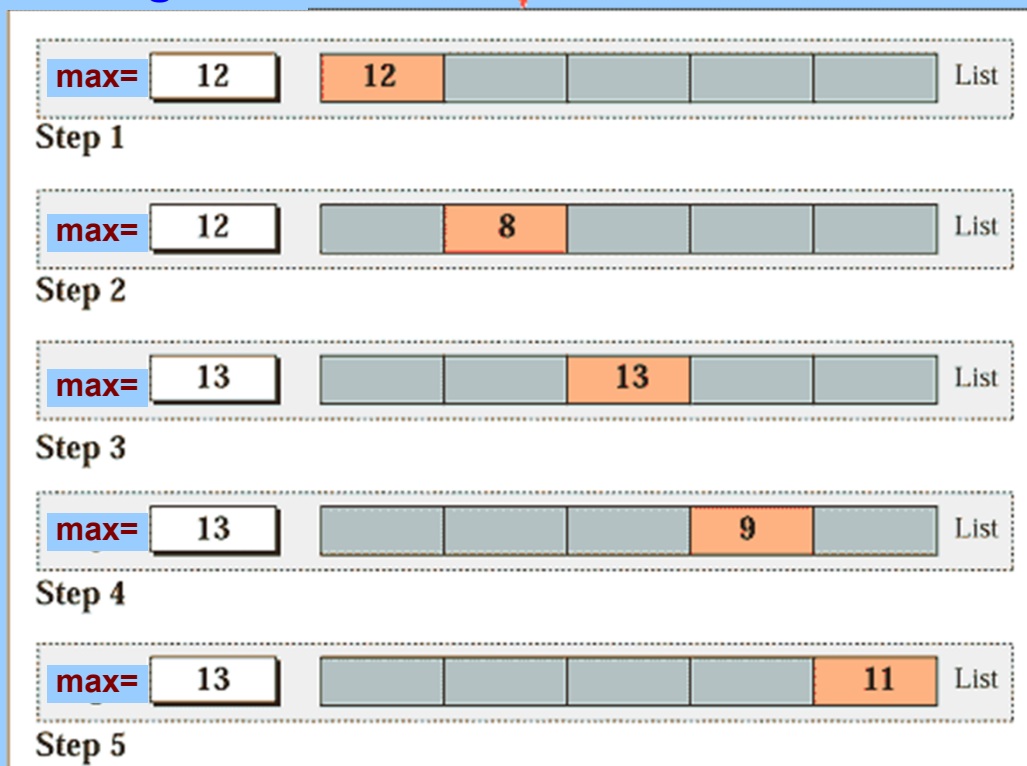


- **Ανεπίσημος ορισμός:** βήμα προς βήμα μέθοδος για την επίλυση ενός προβλήματος
- **Ανεξάρτητος από το υπολογιστικό σύστημα!**

Εύρεση μέγιστου θετικού αριθμού

12 8 13 9 11 Είσοδος

FindLargest



13 Έξοδος

- Διαισθητική λύση με μικρό σύνολο δεδομένων εισόδου

Περισσότερη λεπτομέρεια...

12 8 13 9 11 Είσοδος

FindLargest

max = ο πρώτος αριθμός

Βήμα 1

αν 2ος αριθμός > max, τότε max = 2ος αριθμός

Βήμα 2

αν 3ος αριθμός > max, τότε max = 3ος αριθμός

Βήμα 3

αν 4ος αριθμός > max, τότε max = 4ος αριθμός

Βήμα 4

αν 5ος αριθμός > max, τότε max = 5ος αριθμός

Βήμα 5

13

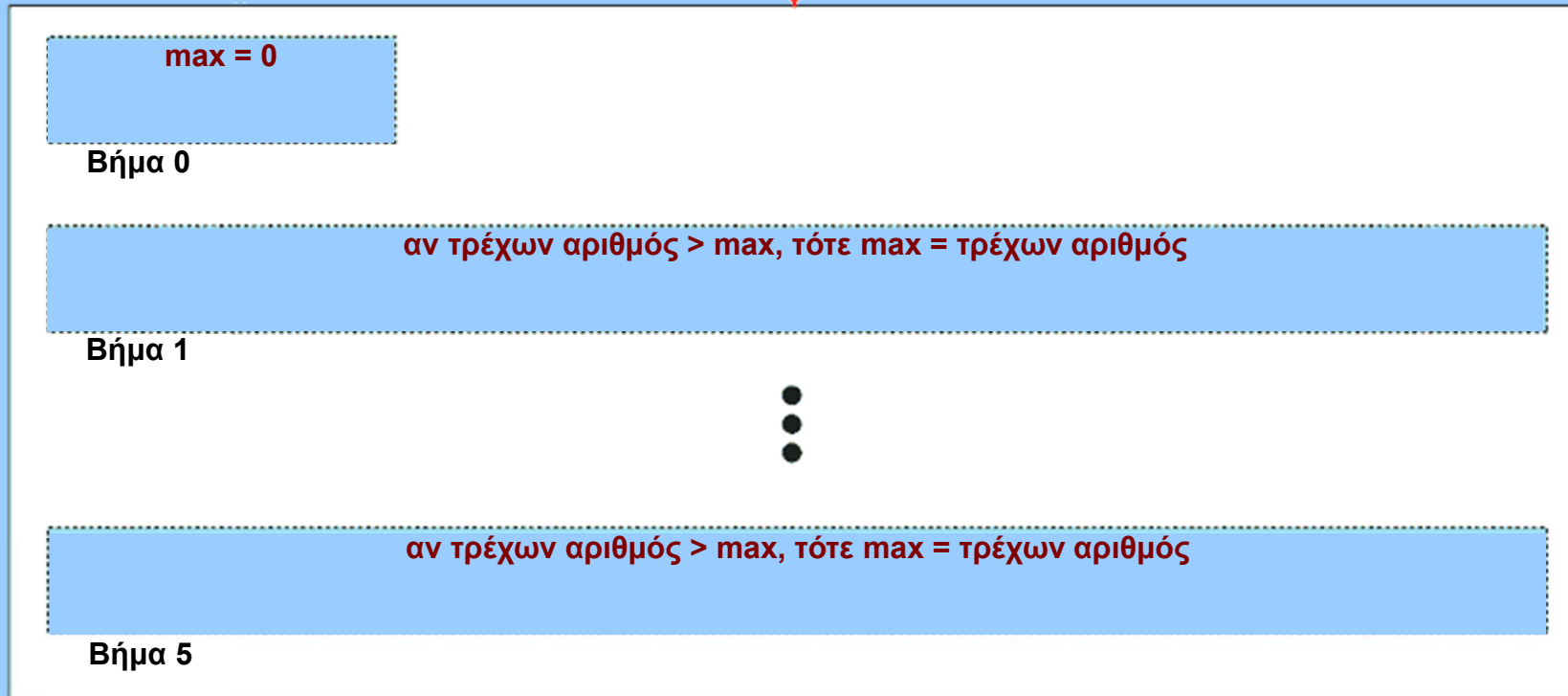
Έξοδος

- Το πρώτο σχήμα του αλγορίθμου
- Τι γίνεται με το 1^ο βήμα;

Μία βελτίωση...

12 8 13 9 11 Είσοδος

FindLargest



- Κανένας θετικός < 0!
- Μπορεί να γενικευτεί για N αριθμούς;

13 Έξοδος

Ο αλγόριθμος στη γενική του μορφή

Είσοδος: Λίστα N θετικών

FindLargest

max = 0

Επανάληψη για N φορές

αν τρέχων αριθμός > max, τότε max = τρέχων αριθμός

max

- Δομή επανάληψης
- Πώς ορίζεται μέγεθος λίστας;

Οι τρεις βασικές αλγοριθμικές δομές

Ενέργεια 1
Ενέργεια 2
...
...
Ενέργεια n

Ακολουθία

Εάν **συνθήκη** αληθής

τότε

Σειρά ενεργειών 1

αλλιώς

Σειρά ενεργειών 2

Απόφαση

Όσο **συνθήκη** αληθής

Ενέργεια 1

Ενέργεια 2

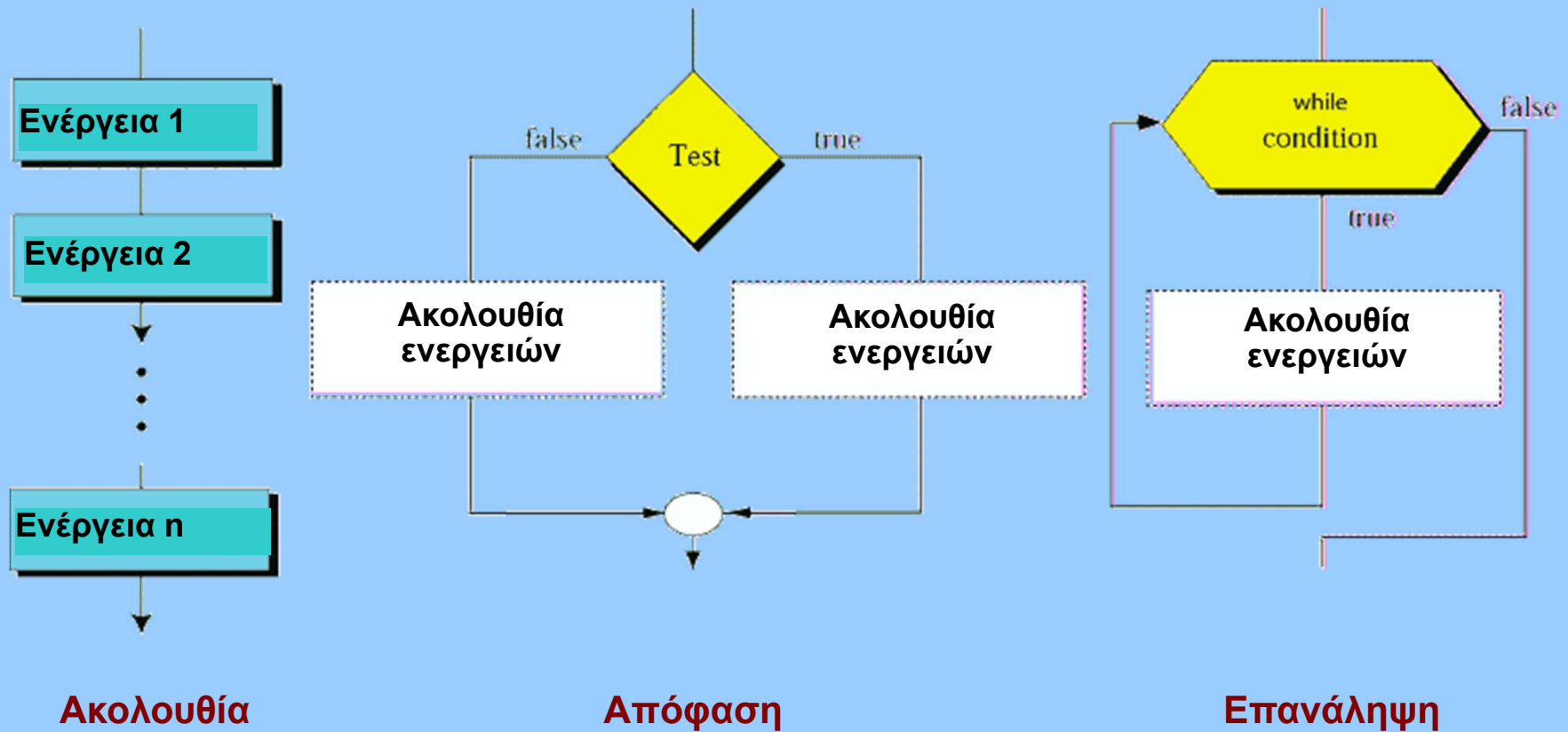
...

...

Ενέργεια n

Επανάληψη

Διαγράμματα ροής



Ψευδοκώδικας

```
action 1  
action 2  
⋮  
action n
```

Ακολουθία

```
if (condition)  
  then  
    action  
    action  
    ...  
  else  
    action  
    action  
    ...  
End if
```

Απόφαση

```
while (condition)  
  action  
  action  
  ...  
End while
```

Επανάληψη

- Σε ελεύθερη μορφή (π.χ. κοντά σε μια γλώσσα προγραμματισμού)
- Δεν υπάρχουν συγκεκριμένοι κανόνες

Η εύρεση του μεγαλύτερου θετικού (σε ψευδοκώδικα)

ΕύρεσηΜεγίστου

Είσοδος: Λίστα θετικών αριθμών

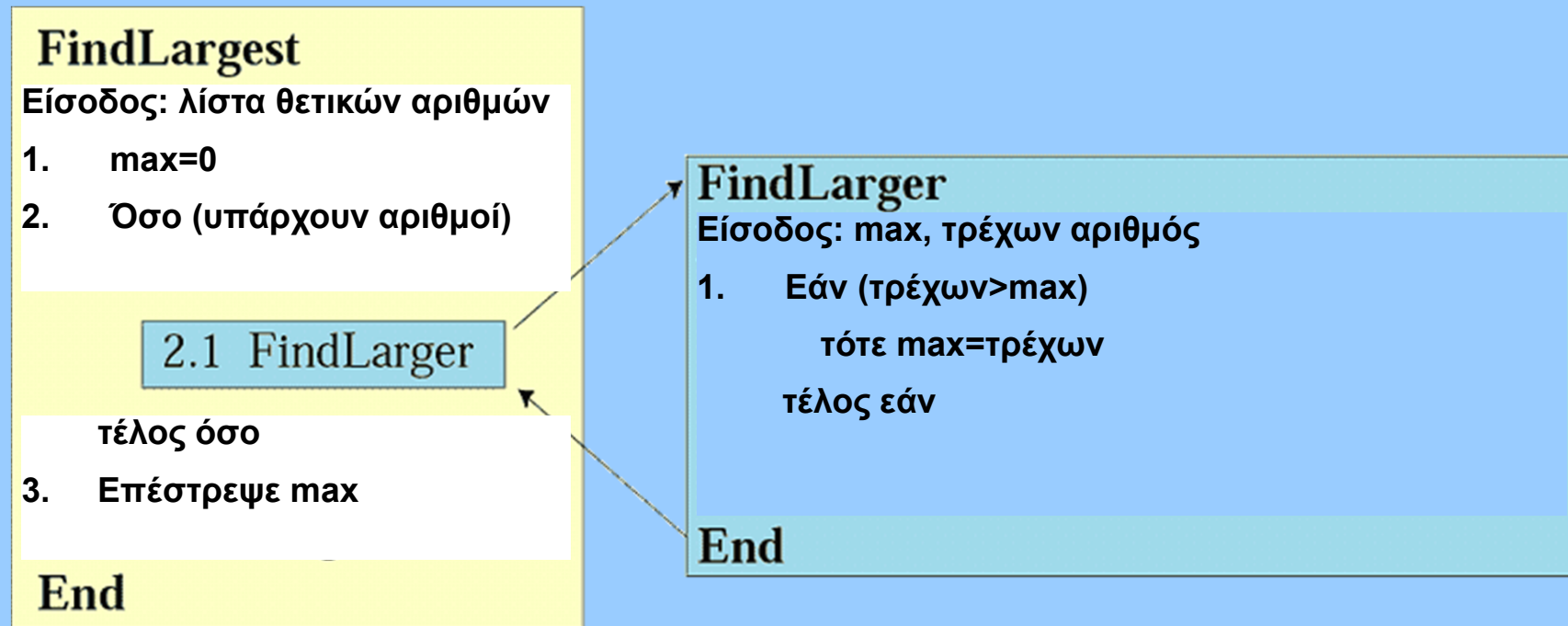
1. $max=0$, τρέχων αριθμός = αρχή λίστας
2. Όσο (υπάρχουν άλλοι αριθμοί) {
3. αν (τρέχων αριθμός $> max$) { $max=$ τρέχων αριθμός } }
4. προώθηση στον επόμενο αριθμό
5. } // τέλος όσο
6. Επιστροφή max
7. Τέλος

- Για τον προγραμματισμό χρειάζεται να οριστούν κάποιες λεπτομέρειες!
 - Προσπέλαση λίστας, Έλεγχος μήκους λίστας...

Ο αλγόριθμος (έναν πιο αυστηρός ορισμός)

- Ένα διατεταγμένο σύνολο
- σαφών βημάτων
- το οποίο παράγει κάποιο αποτέλεσμα
- και τερματίζεται σε πεπερασμένο χρόνο

Υποπρογράμματα (υποαλγόριθμοι)



- Επαναχρησιμοποίηση
 - Κλήση από οποιοδήποτε σημείο
- Ευκολότερη κατανόηση προγράμματος

Ο βασικός αλγόριθμος άθροισης

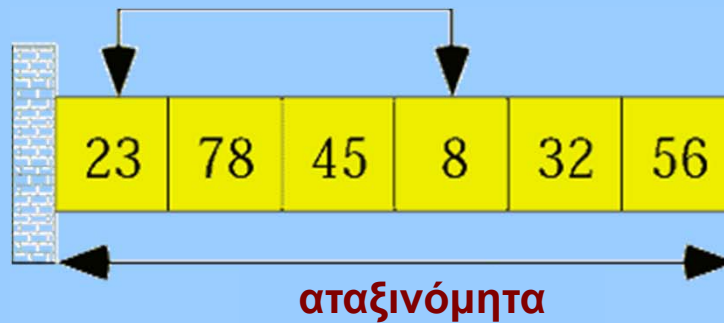
- Επαναληπτική (iterative) διαδικασία
- Αρχική τιμή sum
- Με τον ίδιο τρόπο
 - γινόμενο
 - x^n
 - παραγοντικό



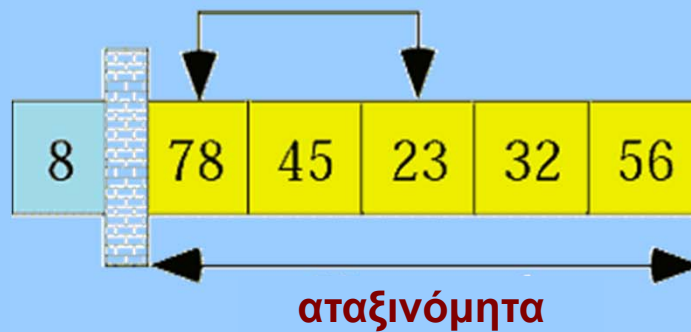
Αλγόριθμοι ταξινόμησης και αναζήτησης

- Βασικοί αλγόριθμοι
 - **Ταξινόμηση**: διάταξη δεδομένων με βάση κάποιο “κλειδί”
 - **Αναζήτηση**: εντοπισμός θέσης δεδομένων σε μια λίστα
- Πολυπλοκότητα αλγορίθμων
 - **Big-O notation**
- Επιλογή κατάλληλου αλγόριθμου
 - Ανάλογα με φύση και διάταξη δεδομένων

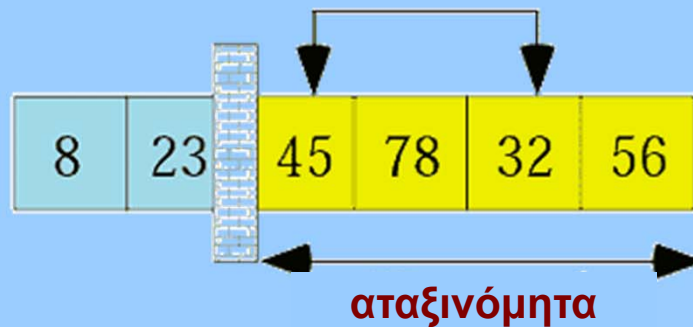
Ταξινόμηση επιλογής (selection sort)



Αρχική Λίστα



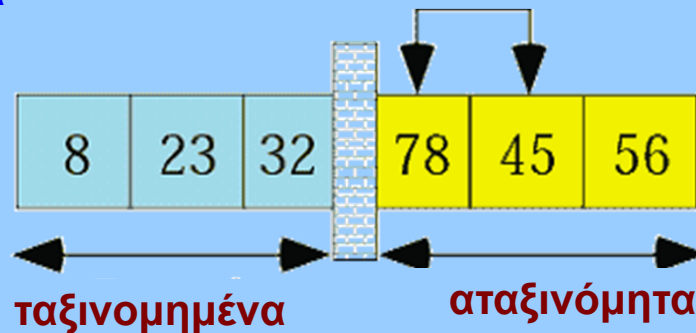
Μετά το 1^ο πέρασμα



Μετά το 2^ο πέρασμα

Ταξινόμηση επιλογής (selection sort)

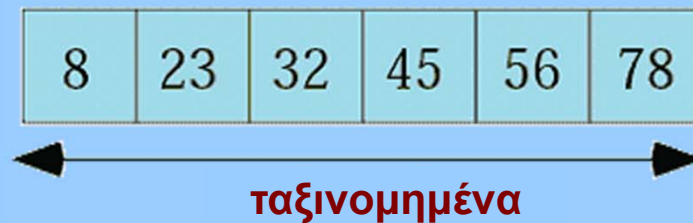
N-1 πέρασματα



Μετά το 3^ο πέρασμα



Μετά το 4^ο πέρασμα



Μετά το 5^ο πέρασμα

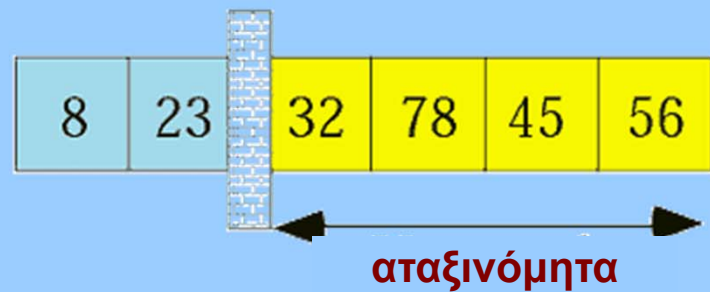
Ταξινόμηση φουσαλίδας (bubble sort)



Αρχική Λίστα



Μετά το 1^ο πέρασμα



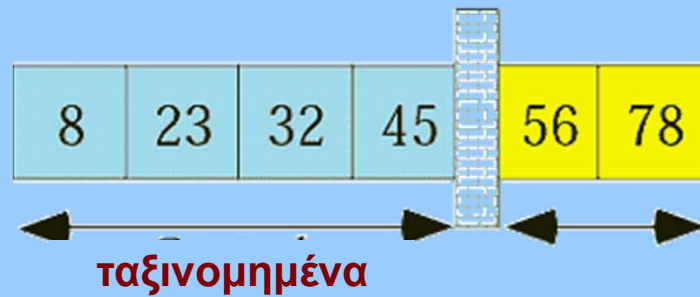
Μετά το 2^ο πέρασμα

Ταξινόμηση φυσαλίδας (bubble sort)

Το πολύ $N-1$ περάσματα



Μετά το 3^ο πέρασμα

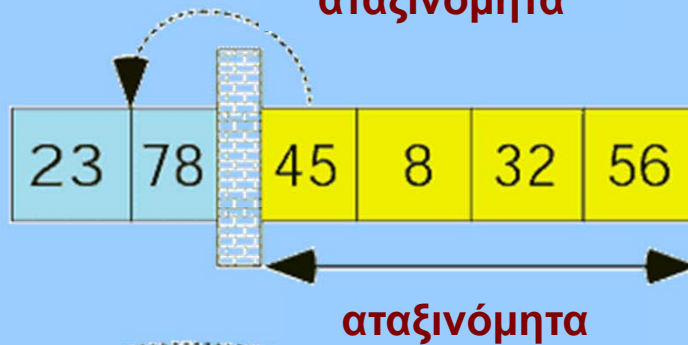


Μετά το 4^ο πέρασμα

Ταξινόμηση παρεμβολής (insertion sort)



Αρχική Λίστα



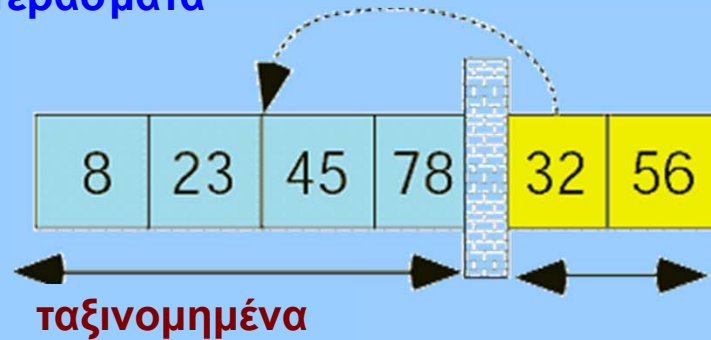
Μετά το 1^ο πέρασμα



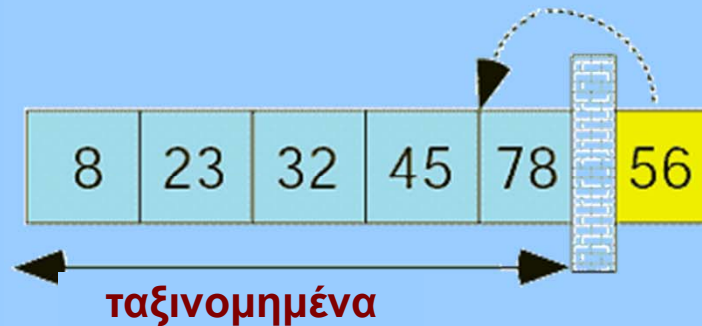
Μετά το 2^ο πέρασμα

Ταξινόμηση παρεμβολής (insertion sort)

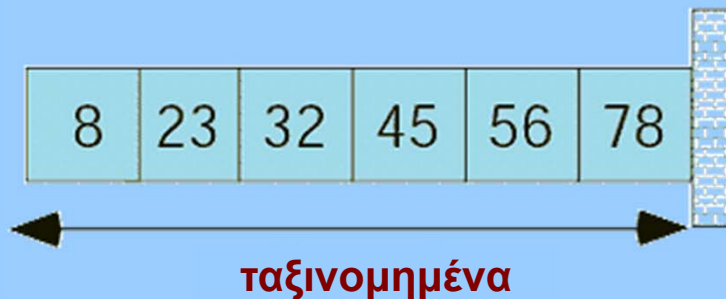
Το πολύ N-1 περάσματα



Μετά το 3^ο πέρασμα



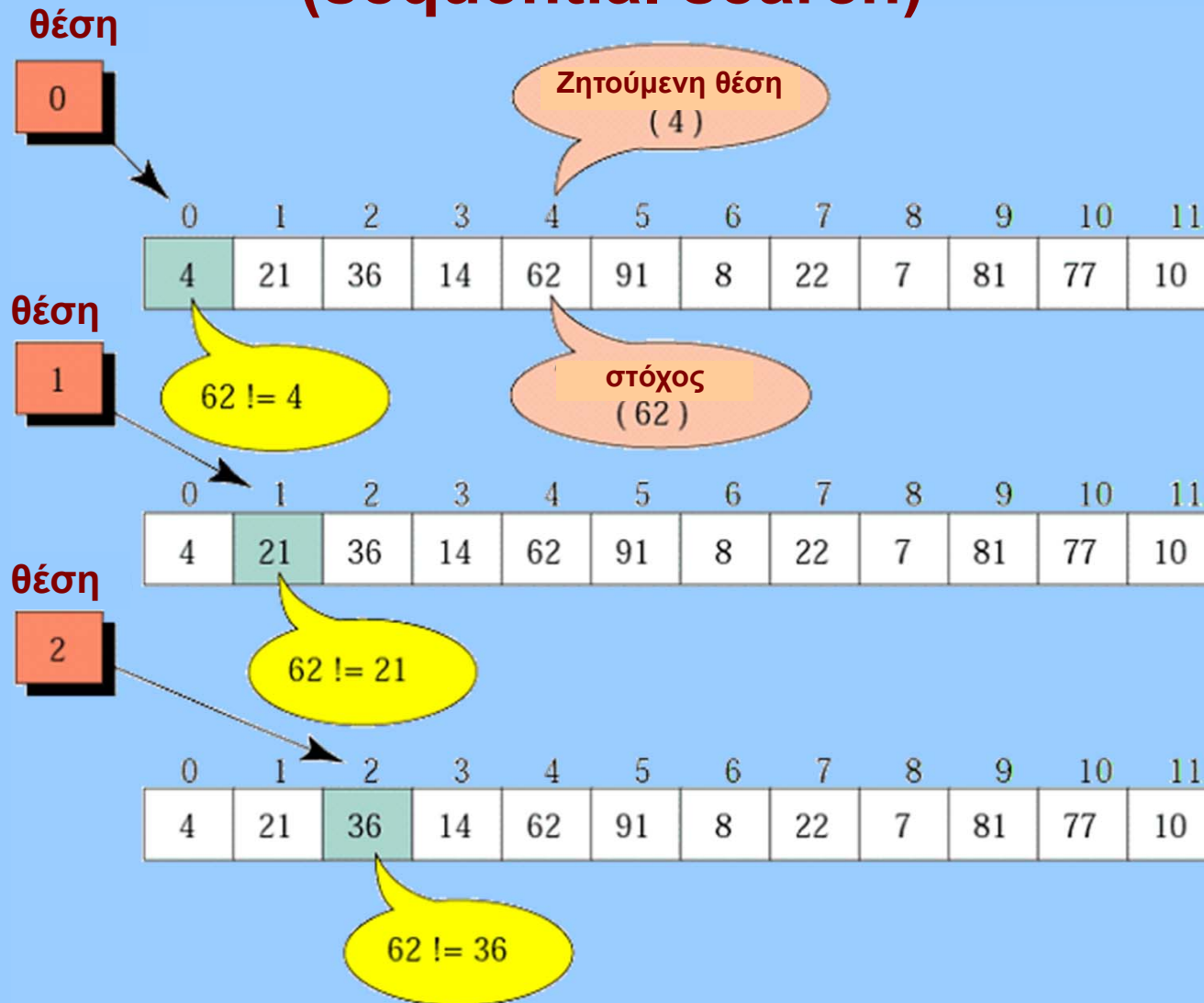
Μετά το 4^ο πέρασμα



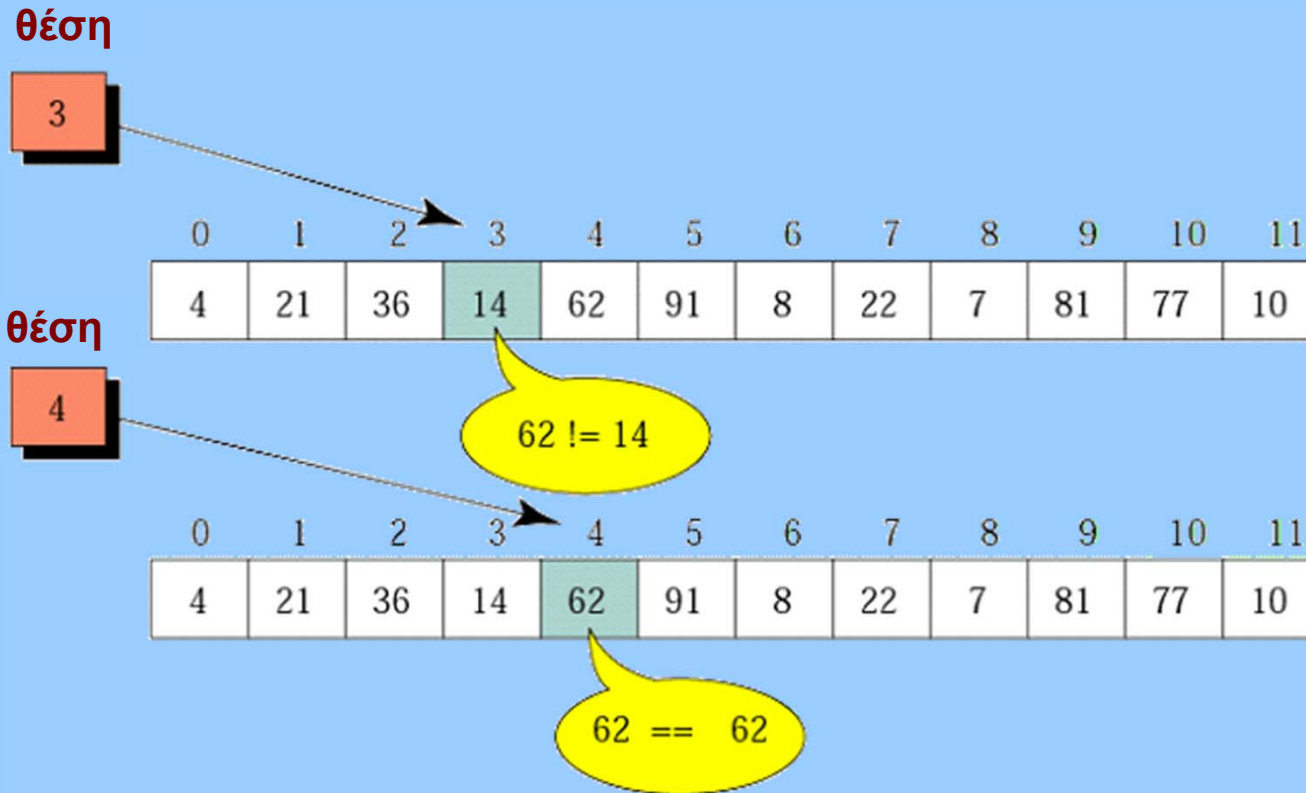
Μετά το 5^ο πέρασμα

Υπάρχουν και πιο αποδοτικοί
αλγόριθμοι ταξινόμησης!

Σειριακή αναζήτηση (sequential search)



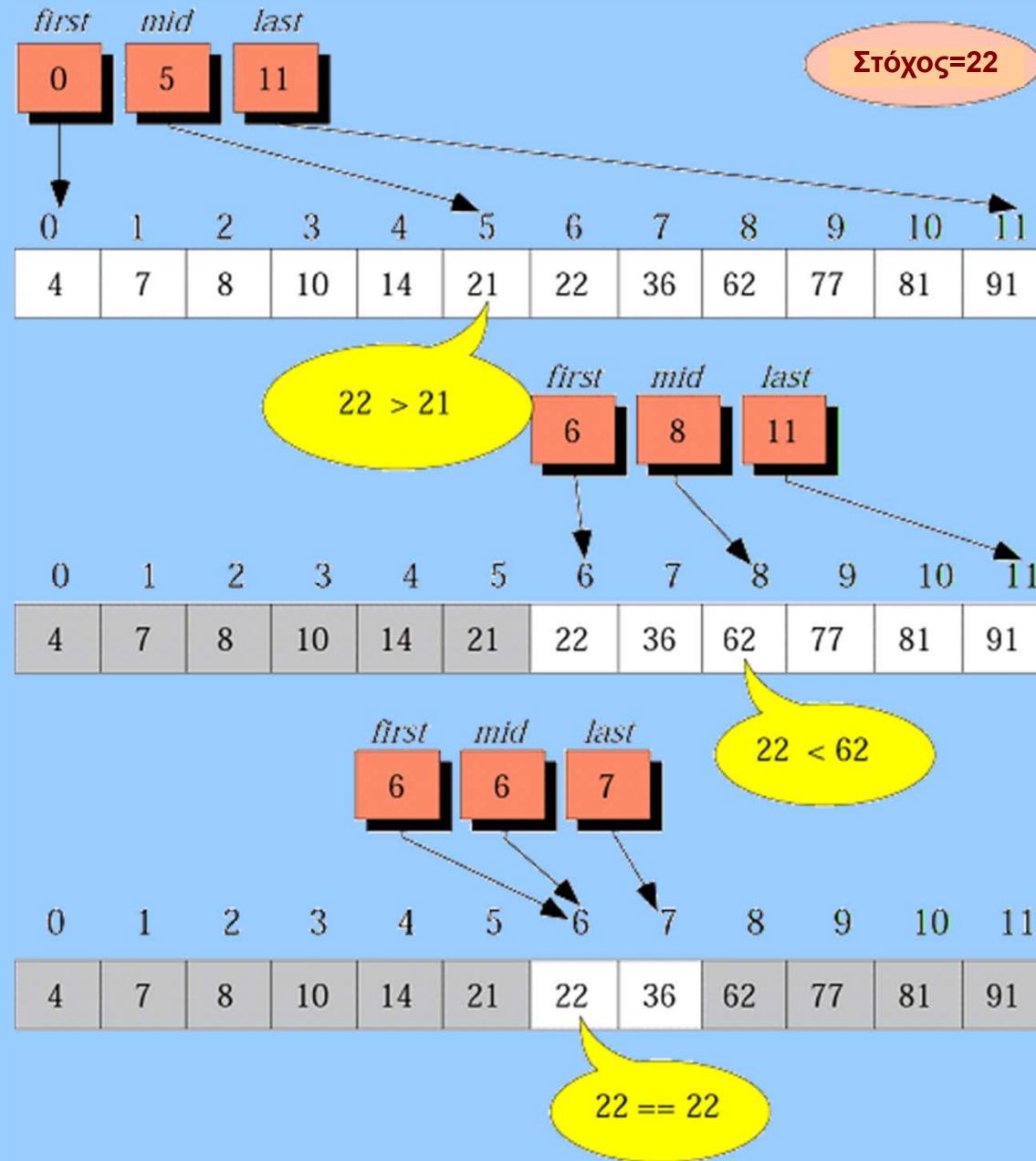
Σειριακή αναζήτηση (sequential search)



- Υπάρχουν και πιο αποδοτικοί αλγόριθμοι εύρεσης!
 - Στη χειρότερη περίπτωση: έως και N συγκρίσεις
 - Χρήση μόνο σε μικρές ή αταξινόμητες λίστες

Διαδική αναζήτηση (binary search)

- Η λίστα πρέπει να είναι ταξινομημένη!
- Έως $\log_2 N$ συγκρίσεις



Αναδρομή (recursion)

- Παράδειγμα: παραγοντικό
 - $n! = \text{factorial}(n) = 1 \times 2 \times 3 \times \dots \times (n-1) \times n$
 - Επαναληπτικός ορισμός
- Αναδρομικός ορισμός:
 - $\text{factorial}(n) = n \times \text{factorial}(n-1)$
- Αναδρομή
 - “Κομψή” προγραμματιστική λύση
 - Απαιτεί υποστήριξη από τη γλώσσα προγραμματισμού
 - Μπορεί να προκαλέσει υπερχείλιση μνήμης

Το παραγοντικό επαναληπτικά

Παραγοντικό(n)

1. **Factorial=1, i=1**
2. **Όσο (i<=n) {**
3. **factorial=factorial x i**
4. **i=i+1**
5. **} // τέλος όσο**
6. **Επιστροφή factorial**
7. **Τέλος**

Το παραγοντικό αναδρομικά

Παραγοντικό(n)

1. Αν $(n=0)$ τότε {
2. επέστρεψε 1
3. }
4. αλλιώς {
5. επέστρεψε $n \times$ Παραγοντικό(n-1)
6. }
7. Τέλος

- Το (υπο)πρόγραμμα καλεί τον εαυτό του!

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2006-07

Δομές Δεδομένων (I)

Μ.Στεφανιδάκης

Στόχοι του μαθήματος

- Αφηρημένες Δομές (Τύποι) δεδομένων
 - Γραμμικές Λίστες
 - Στοίβες και Ουρές
 - Δένδρα
 - Γράφοι

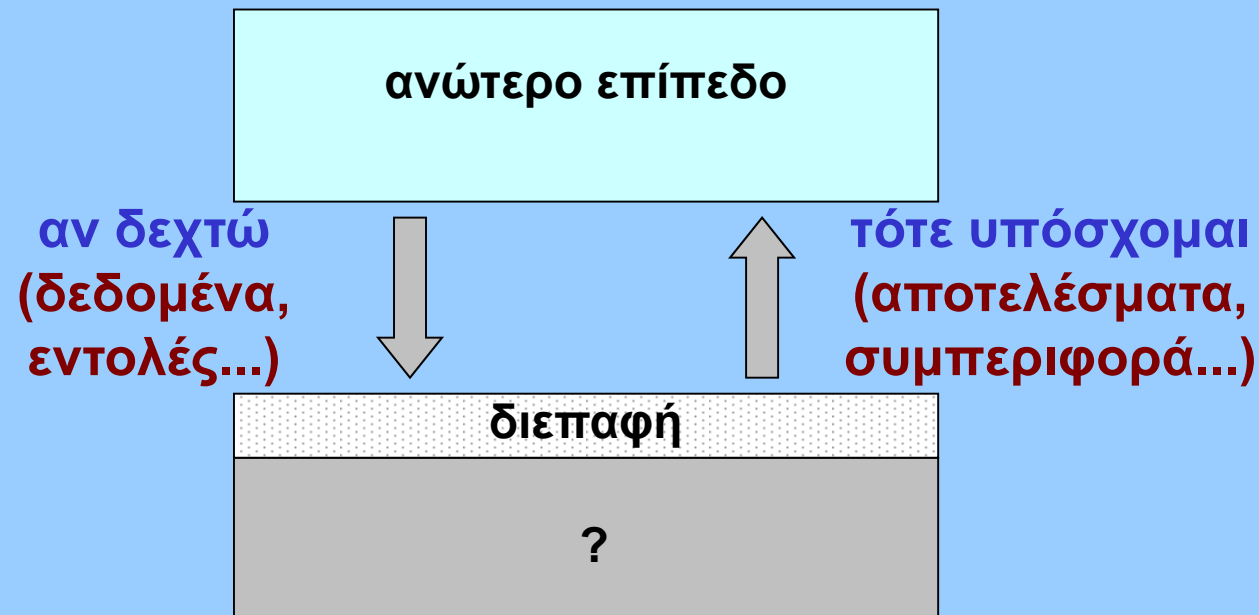
Δομές Δεδομένων

- **Ορισμός δομής δεδομένων**
 - Σύνολο συσχετιζόμενων δεδομένων
 - με καθορισμένη διάταξη
 - και λειτουργίες διαχείρισης
 - Προσπέλαση
 - Αναζήτηση
 - Τροποποίηση
 - ...
- **Βασικές δομές (τύποι) δεδομένων**
 - Παρέχονται από τη γλώσσα προγραμματισμού
 - Παράδειγμα:
 - πίνακες (arrays)
 - εγγραφές (records, structures)

Αφηρημένοι Τύποι Δεδομένων

- **Abstract data types (ADTs)**
- **Προγραμματιστικά οριζόμενοι τύποι**
 - **Δήλωση δεδομένων**
 - **Δήλωση λειτουργιών**
 - **Ενθυλάκωση δεδομένων και λειτουργιών**
 - Προσπέλαση μέσω διεπαφής (interface)
 - Απόκρυψη λεπτομερειών από χρήστη
 - **Παραδείγματα αφηρημένων τύπων δεδομένων**
 - **Λίστες (lists)**
 - **Στοιίβες (stacks)**
 - **Ουρές (queues)**
 - **Δένδρα (trees)**
 - **Γράφοι (graphs)**

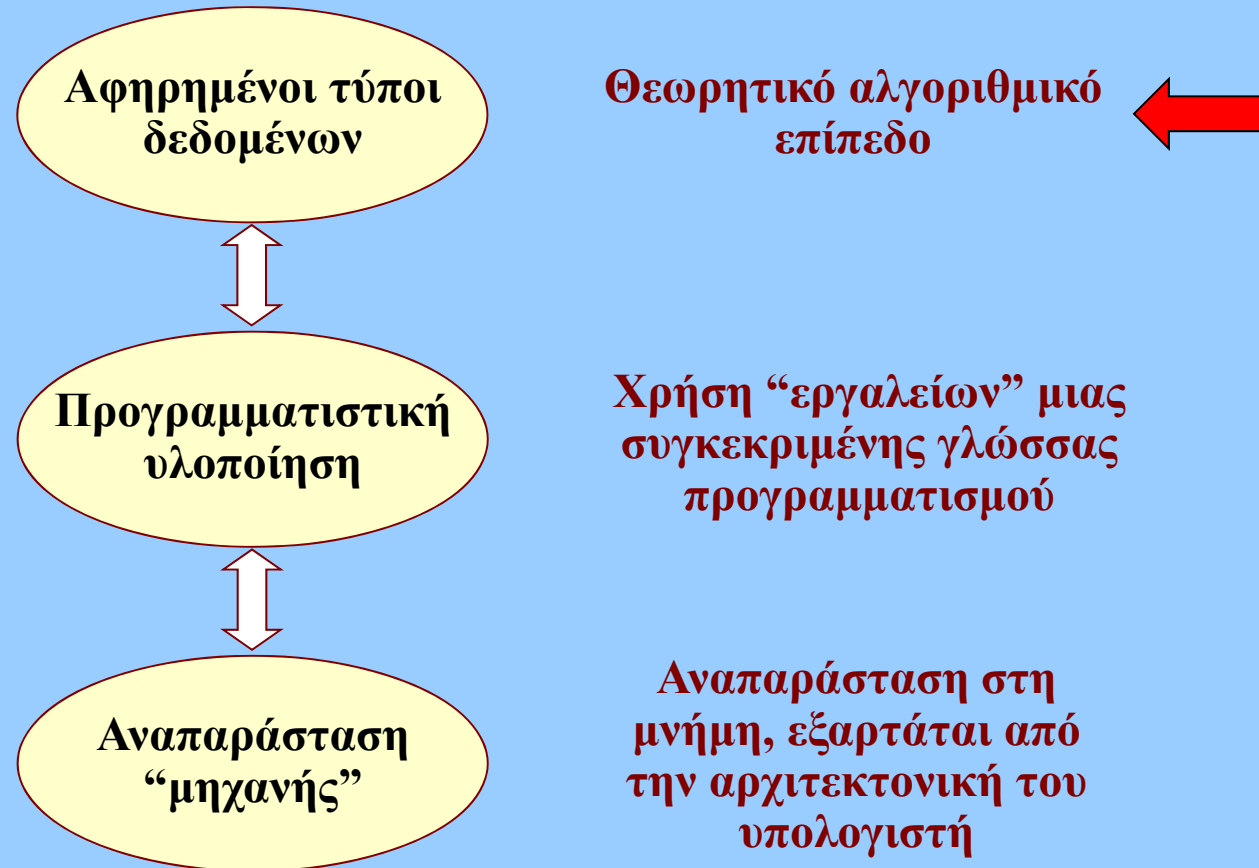
Η σπουδαιότητα των διεπαφών



- Αντιμετώπιση πολυπλοκότητας σχεδιασμού
- Επαναχρησιμοποίηση τμημάτων
- Αξιόπιστος σχεδιασμός

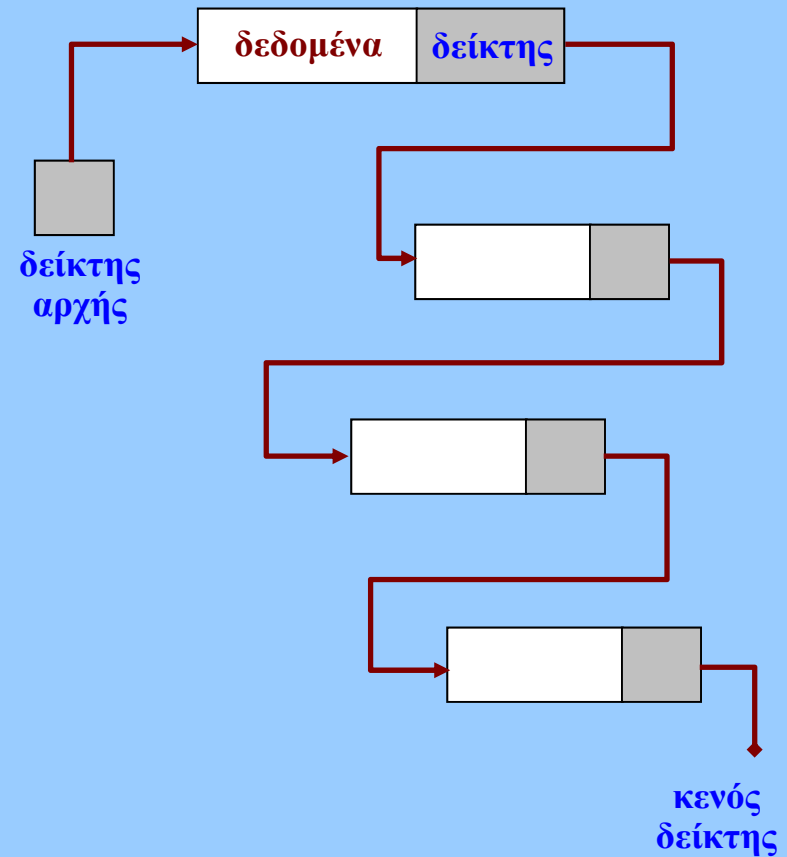
Αφηρημένοι Τύποι Δεδομένων

Η γενική εικόνα

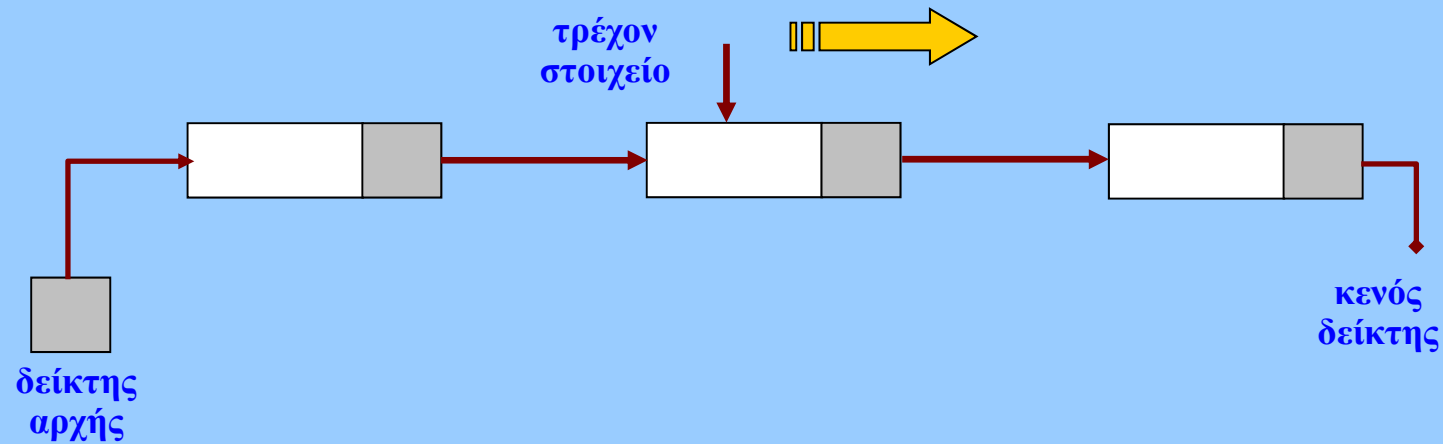


Γραμμικές Λίστες

- Κάθε στοιχείο έχει έναν μόνο **διάδοχο** και **προκάτοχο**
 - Ειδική περίπτωση: πρώτο και τελευταίο στοιχείο
 - Σειριακή διάταξη
 - Πεπερασμένος ή μη αριθμός στοιχείων
 - Τυχαίες λίστες και διατεταγμένες (με κλειδί)
- **Εφαρμογές**
 - Αποθήκευση δεδομένων
 - με ή χωρίς διάταξη

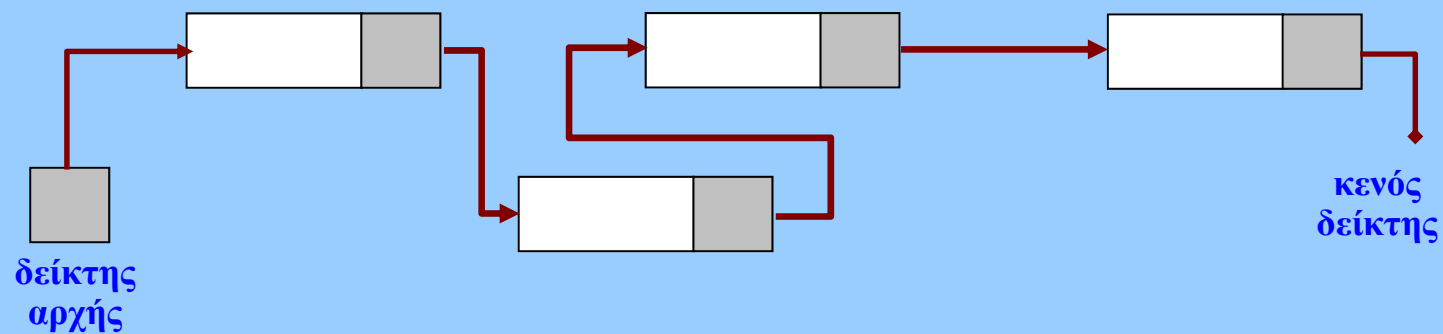
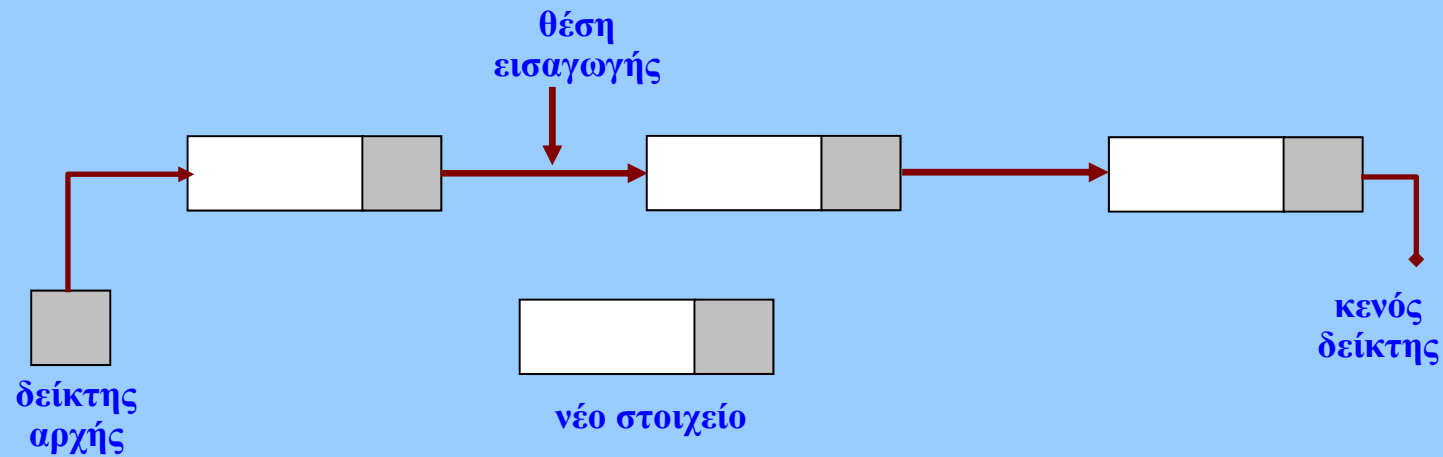


Γραμμικές Λίστες: διάσχιση



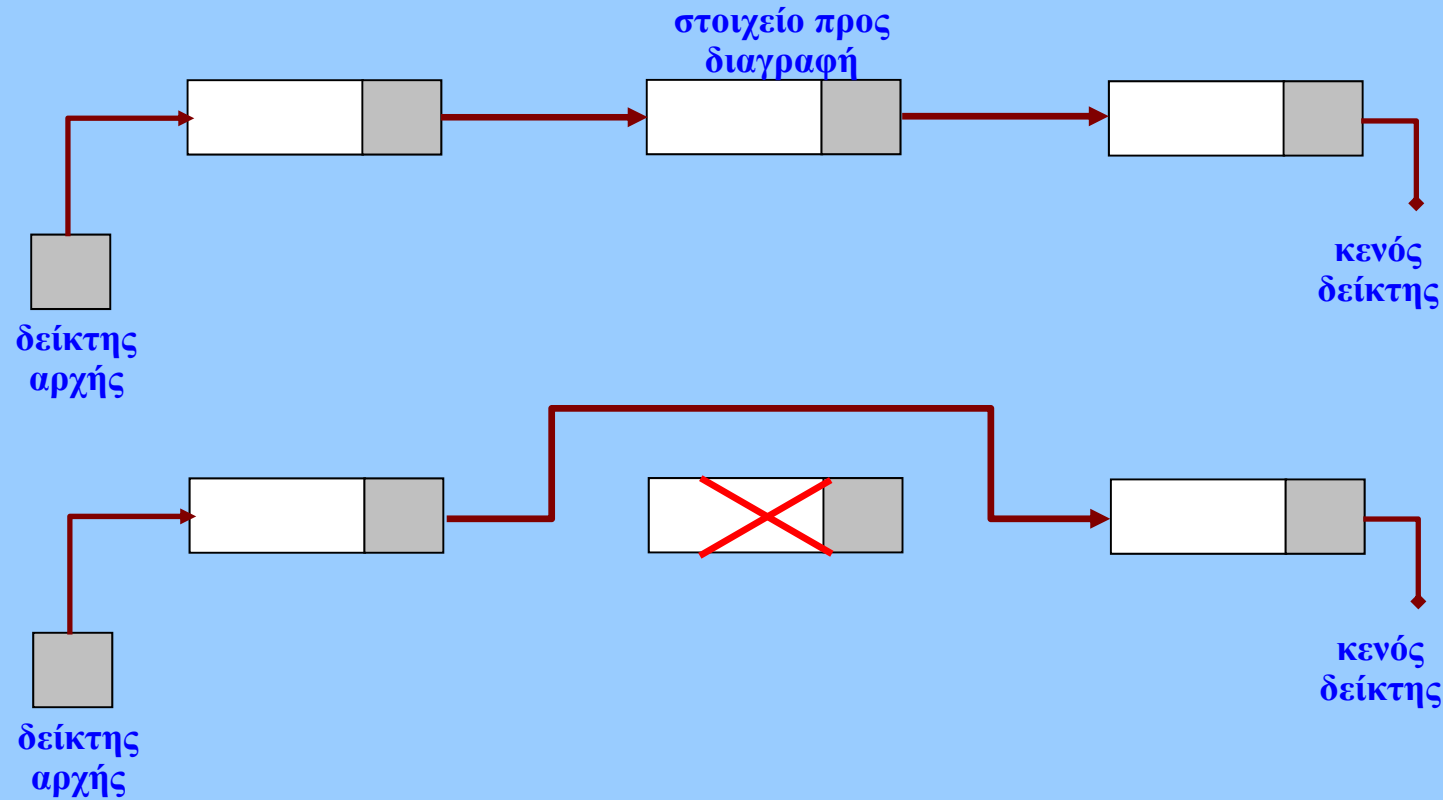
- Σειριακή λειτουργία
 - Για αναζήτηση στοιχείων
 - **Κλειδί**: μέρος δεδομένων κάθε στοιχείου
 - Για επεξεργασία όλων των στοιχείων

Γραμμικές Λίστες: εισαγωγή



- **Θέση εισαγωγής**
 - **Μεταξύ στοιχείων της λίστας**
 - Προσδιορισμός θέσης μετά από αναζήτηση
 - **Εάν η λίστα είναι πεπερασμένου μεγέθους: πιθανότητα υπερχείλισης**

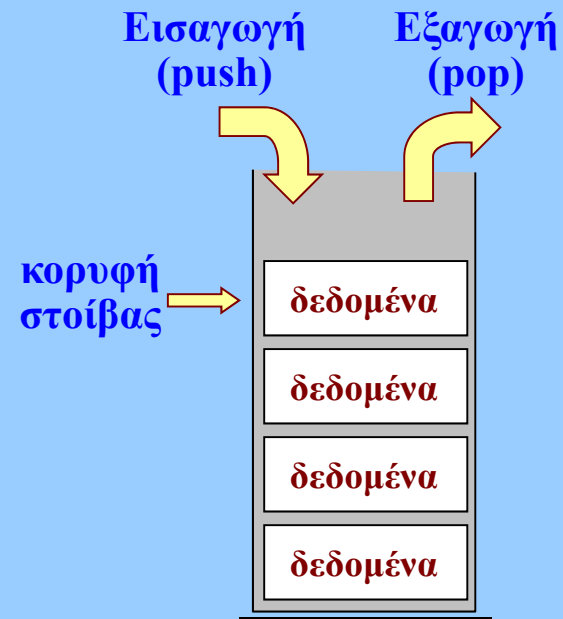
Γραμμικές Λίστες: διαγραφή



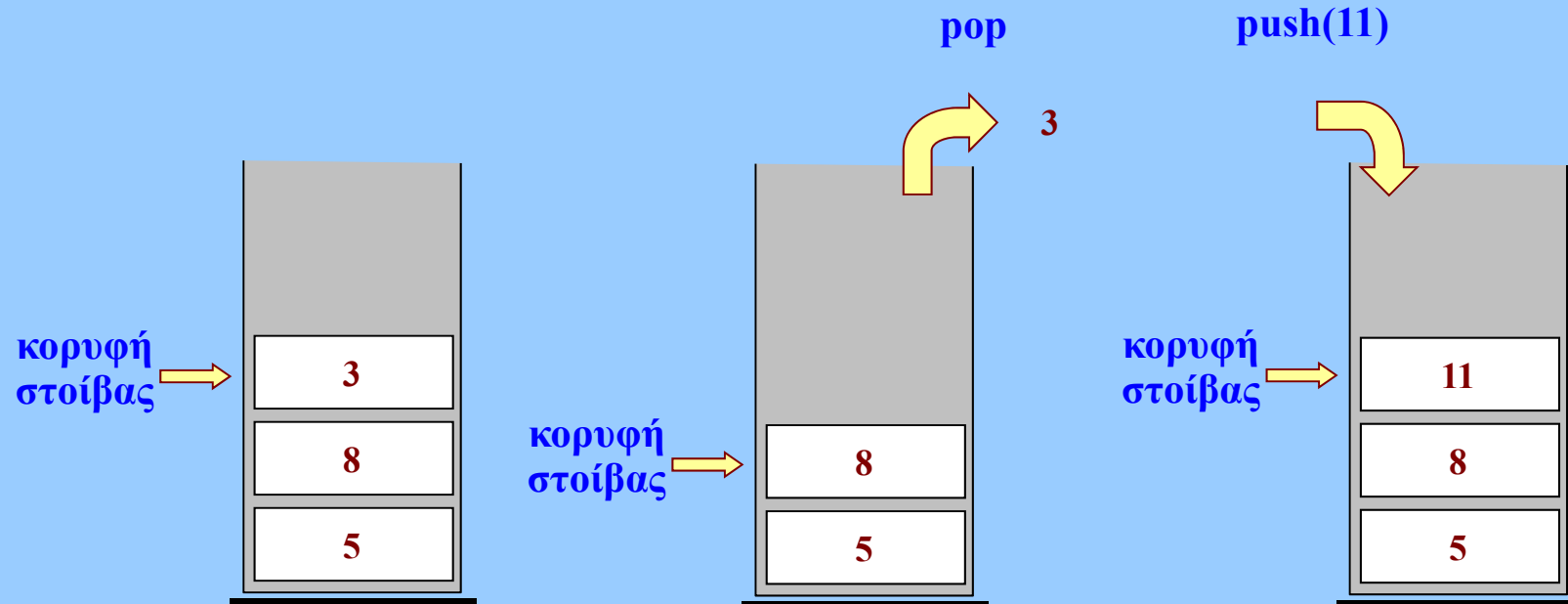
- **Θέση διαγραφής**
 - Προσδιορισμός στοιχείου μετά από αναζήτηση
 - Εάν η λίστα είναι κενή: πιθανότητα υποχείλισης

Στοιίβες (stacks)

- Είδος γραμμικής λίστας
 - Περιορισμένη (restricted)
 - εισαγωγή-διαγραφή στοιχείων μόνο στο ένα άκρο
- LIFO (last-in first-out)
- Λειτουργίες
 - Εισαγωγή (ώθηση-push)
 - Εξαγωγή (απώθηση-pop)
 - Έλεγχος υπερ(υπο)χειρίσισης
- Εφαρμογές
 - Συντακτική ανάλυση
 - Αλγεβρικές παραστάσεις
 - “Backtracking”
 - Τεχνητή νοημοσύνη
 - Παιχνίδια υπολογιστών
 - Runtime stacks

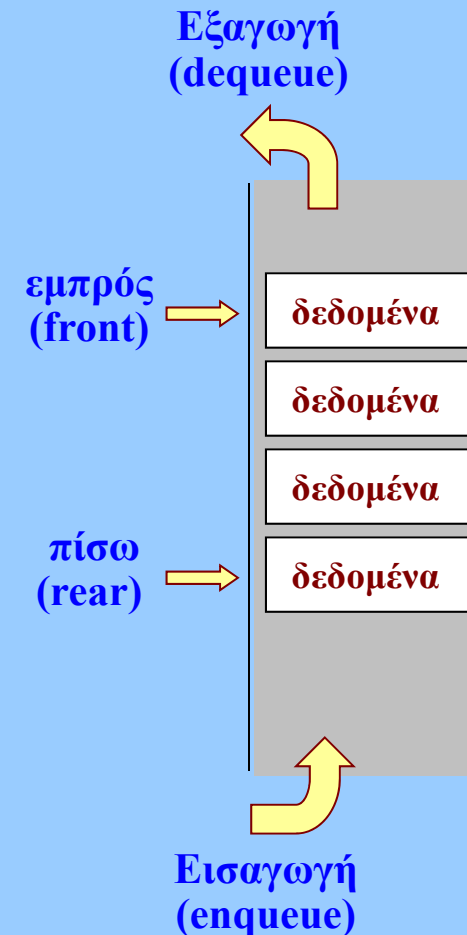


Στοίβες: εισαγωγή και εξαγωγή



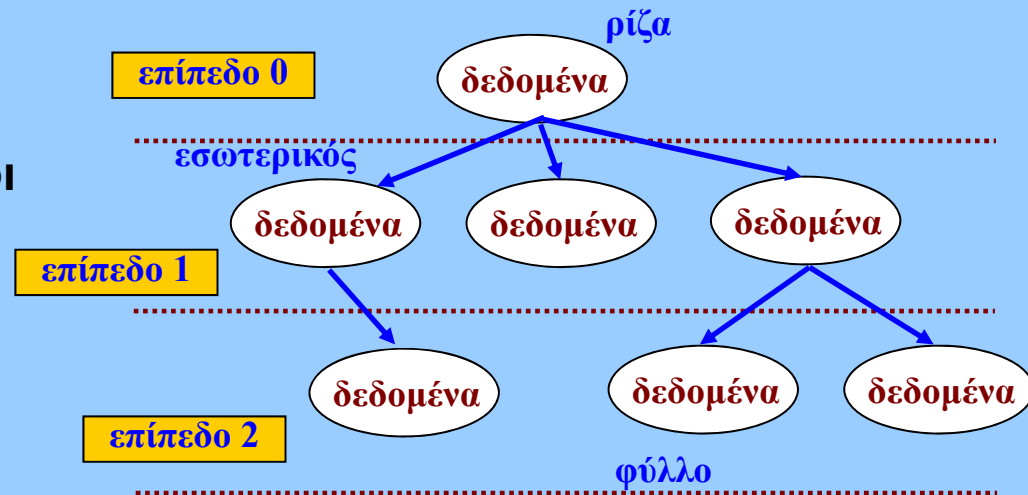
Ουρές (queues)

- Είδος γραμμικής λίστας
 - Περιορισμένη (restricted)
 - Εισαγωγή στο πίσω μέρος
 - Εξαγωγή στο εμπρός μέρος
- FIFO (first-in first-out)
- Λειτουργίες
 - Εισαγωγή (enqueue)
 - Εξαγωγή (dequeue)
 - Έλεγχος υπερ(υπο)χείλισης
- Εφαρμογές
 - Ουρές αναμονής
 - π.χ. σε εκτυπωτές
 - Μοντέλο παραγωγού-καταναλωτή
 - (producer-consumer)
 - με διαφορετικούς ρυθμούς



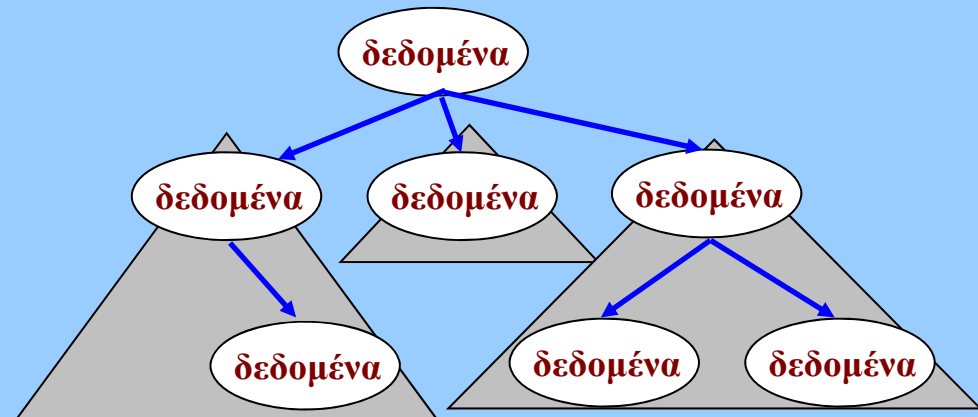
Δένδρα (trees)

- **Κόμβοι (nodes)**
 - Εσωτερικοί
 - Φύλλα
 - Πρόγονος – απόγονοι
 - **Ρίζα (root)**
- **Επίπεδο κόμβου**
 - Απόσταση από ρίζα
 - Ύψος (βάθος)
 - Μέγιστο επίπεδο +1
- **Κλάδοι**
 - Βαθμός εισόδου (1)
 - Εκτός από ρίζα
 - Βαθμός εξόδου



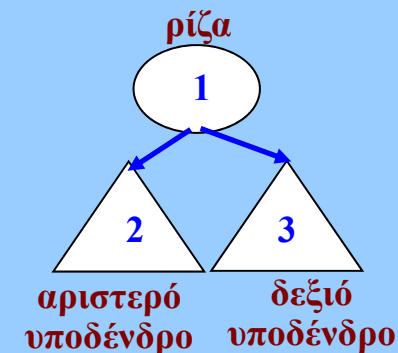
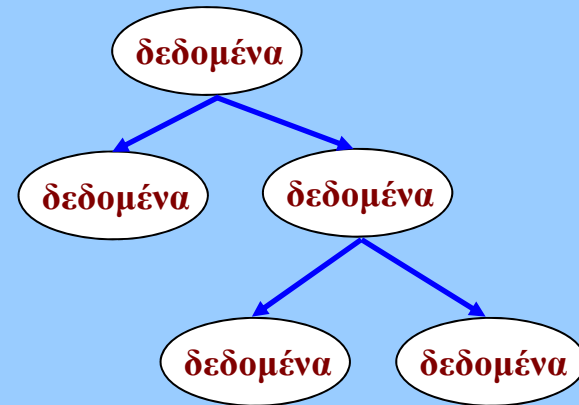
Δένδρα (trees)

- Υποδένδρα
- Διαδρομή (path)
 - Ακολουθία γειτονικών κόμβων
- **Λειτουργίες**
 - Κατασκευή δένδρου
 - Διάσχιση
 - Αναζήτηση και επεξεργασία
- **Εφαρμογές**
 - πολλές επιστημονικές αλγοριθμικές εφαρμογές!
 - Αναζήτηση, κωδικοποίηση...



Δυαδικά Δένδρα (binary trees)

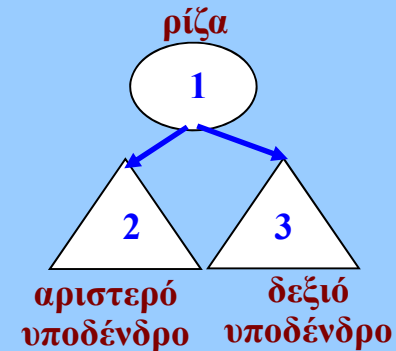
- **Δυαδικά δένδρα**
 - **Ύψος με N κόμβους**
 - $\min = \lceil \log_2(N + 1) \rceil$
 - $\max = N$
 - **Κόμβοι με ύψος H**
 - $\max = 2^H - 1$
 - $\min = H$
 -



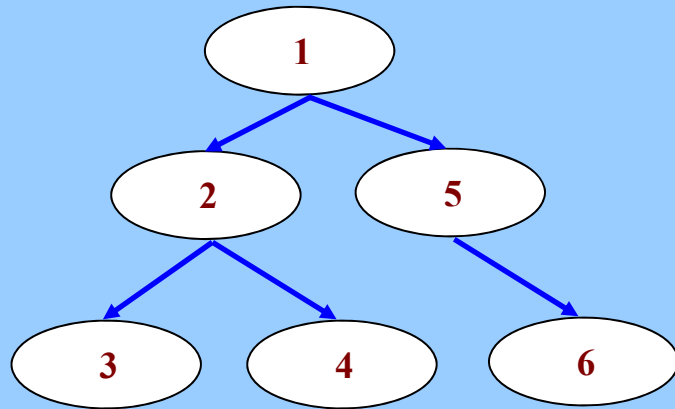
Κάθε υποδένδρο είναι δυαδικό δένδρο

Δυαδικά Δένδρα: διάσχιση

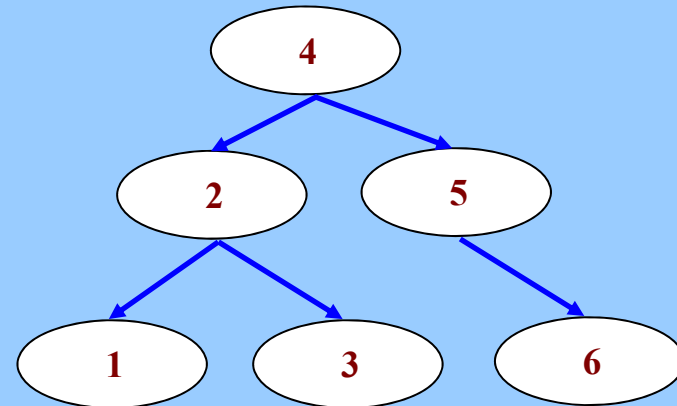
- **Διάσχιση (traversal)**
 - **Διάσχιση με προτεραιότητα βάθους**
 - προδιατεταγμένη (preorder)
 - 1 – 2 – 3
 - ενδοδιατεταγμένη (inorder)
 - 2 – 1 – 3
 - μεταδιατεταγμένη (postorder)
 - 2 – 3 – 1
 - Και άλλες 3 ανώνυμες...
 - Χρήση αναδρομής
 - **Διάσχιση προτεραιότητας εύρους**
 - κατά επίπεδα



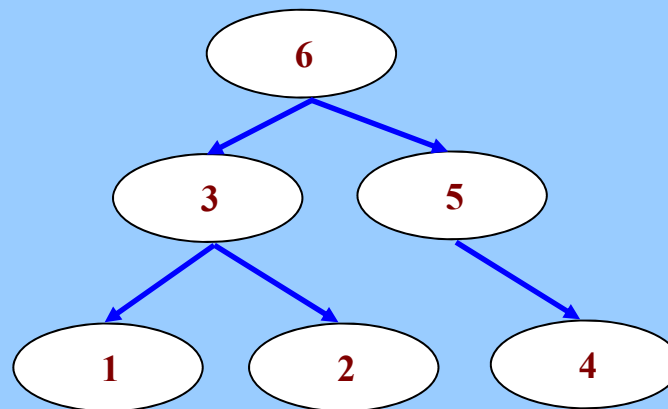
Παραδείγματα διάσχισης



προδιατεταγμένη



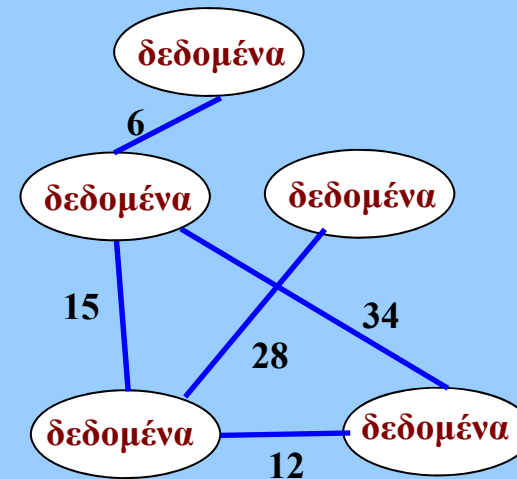
ενδοδιατεταγμένη



μεταδιατεταγμένη

Γράφοι (graphs)

- Συλλογή κόμβων (**κορυφών**) και γραμμών (**ακμών** ή **τόξων**)
 - Προσανατολισμένοι ή μη
 - Βάρη ακμών
- Γειτονικές κορυφές (adjacent)
 - όταν συνδέονται με ακμή
- Διαδρομή
 - Ακολουθία γειτονικών κορυφών
- Κύκλος
 - Διαδρομή τουλάχιστον 3 κορυφών με ίδια αρχή-τέλος
- Βρόχος
 - Τόξο με ίδια αρχή-τέλος
- Βαθμός εισόδου-εξόδου κορυφής

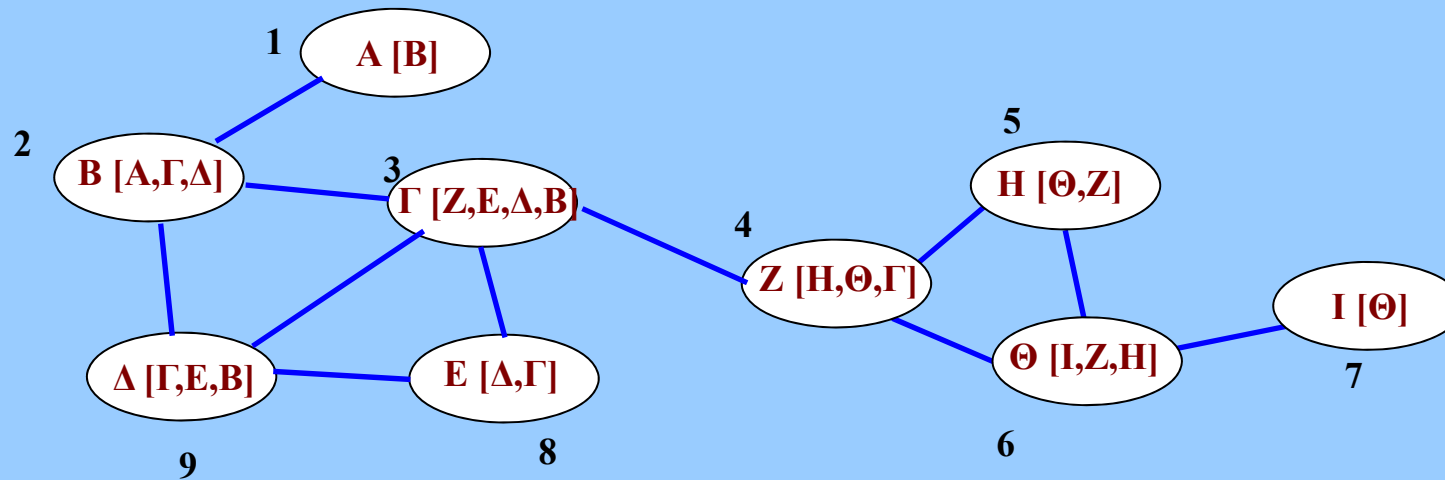


- Συνδεδεμένος γράφος
 - Υπάρχει διαδρομή μεταξύ οποιωνδήποτε κορυφών

Γράφοι (graphs)

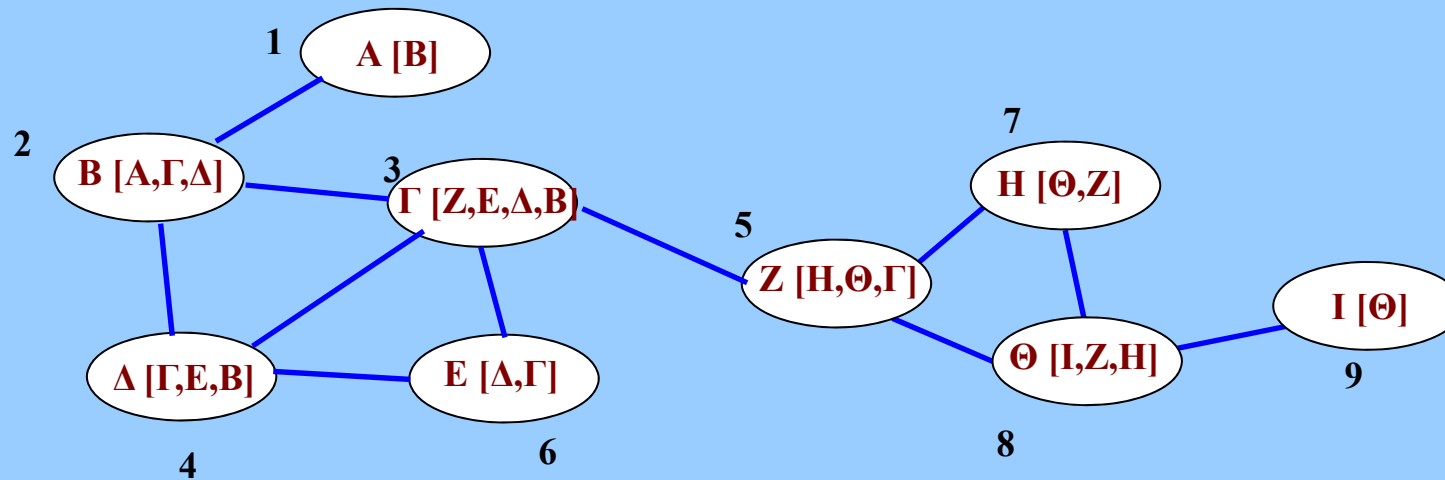
- **Λειτουργίες**
 - Προσθήκη-διαγραφή κορυφών και ακμών
 - Διάσχιση-αναζήτηση
 - Πιθανότητα επίσκεψης ίδιας κορυφής περισσότερες από μία φορές
 - Σημαία (flag) επίσκεψης
- **Εφαρμογές**
 - Εύρεση συντομότερων διαδρομών
 - Ανάθεση πόρων
 - Χρονοδρομολόγηση
 - ...
 - *πάρα πολλά προβλήματα μπορούν να επιλυθούν με την απεικόνισή τους σε έναν γράφο*

Διασχίσεις γράφων



- Διάσχιση κατά βάθος πρώτα (depth-first traversal)
 - Μετακίνηση σε γειτονική κορυφή
 - Εξάντληση απογόνων
 - Επιστροφή (αναδρομή ή στοίβα)
 - Άλλες γειτονικές κορυφές

Διασχίσεις γράφων



- Διάσχιση κατά εύρος πρώτα (breadth-first traversal)
 - Μετακίνηση σε γειτονική κορυφή
 - Εξάντληση όλων γειτονικών κορυφών
 - Μετακίνηση στους απογόνους (χρήση ουράς)

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2006-07

Δομές Δεδομένων (II)

Μ.Στεφανιδάκης

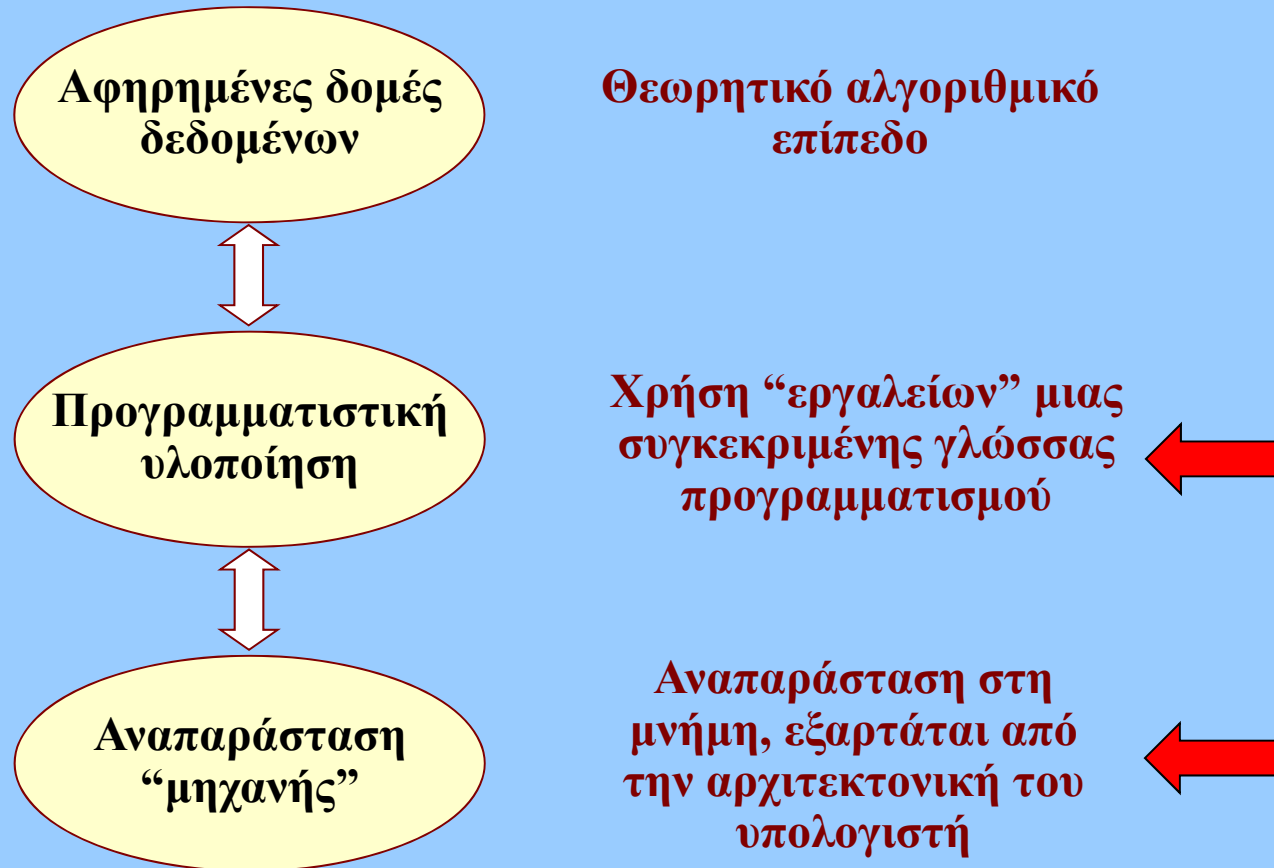
Στόχοι του μαθήματος

- Προγραμματιστικά εργαλεία
 - Απλές μεταβλητές
 - Σύνθετες μεταβλητές
 - Πίνακες
 - Δείκτες
- Προγραμματιστικά εργαλεία
 - Κατηγορίες μεταβλητών
 - Εγγραφές και Δείκτες
 - Πίνακες και Διασυνδεδεμένοι Κόμβοι
- Μέθοδοι υλοποίησης αφηρημένων δομών δεδομένων

Δομές Δεδομένων

- **Ορισμός δομής δεδομένων**
 - Σύνολο συσχετιζόμενων δεδομένων
 - με καθορισμένη διάταξη
 - και λειτουργίες διαχείρισης
 - προσπέλαση, αναζήτηση, τροποποίηση...
- **Βασικές δομές (τύποι) δεδομένων**
 - παρέχονται από τη γλώσσα προγραμματισμού
 - πίνακες (arrays), εγγραφές (records, structures)
- **αφηρημένες δομές (τύποι) δεδομένων**
 - abstract data types (ADTs)
 - προγραμματιστικά οριζόμενες
 - λίστες, σωροί (stacks), ουρές (queues), δένδρα, γράφοι

Η γενική εικόνα



Εργαλεία γλώσσας προγραμματισμού

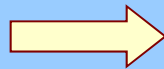
- **Μεταβλητές**
 - Απλές δομές δεδομένων
- **Υποστηριζόμενοι τύποι μεταβλητών**
 - **Ανάλογα με γλώσσα**
 - **Βασικοί τύποι**
 - Ακέραιοι, κινητής υποδιαστολής, χαρακτήρες..
 - **Ακολουθίες χαρακτήρων (strings)**
 - Ιδιαίτερη αντιμετώπιση σε όλες τις γλώσσες
 - **Σύνθετοι τύποι**
 - Δείκτες, πίνακες, δομές...

Εργαλεία γλώσσας προγραμματισμού

- Βασικοί Τύποι Μεταβλητών

- απλοί
 - ακέραιοι
 - χαρακτήρες
 - float, double
 - ...
- 1 ή περισσότερα bytes
 - `sizeof(..)`

από τα “Ταξίδια του Γκιούλιβερ” !



(short) int x = 4825;

ή μήπως... 12 D9 ;

4825 (dec) = 12D9 (hex)

2 bytes – ποιο αποθηκεύεται πρώτο στη μνήμη;

big-endian: το **περισσότερο** σημαντικό byte στη **χαμηλότερη** θέση μνήμης!

– Motorola, Sun, PowerPC...

little-endian: το **λιγότερο** σημαντικό byte στη **χαμηλότερη** θέση μνήμης!

– Intel (Pentium)...

...

500		
502	D9	12
504		
506		

...

Εργαλεία γλώσσας προγραμματισμού

- Βασικοί Τύποι Μεταβλητών

- σύνθετοι

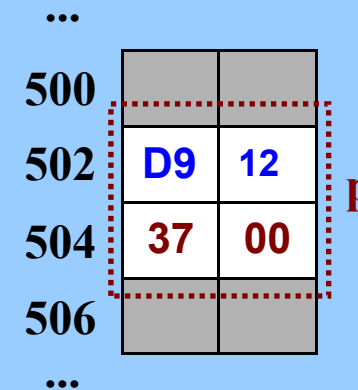
- εγγραφές (**records**)
- ή “δομές” (**structures**)

```
struct point {  
    int x;  
    int y;  
};
```

όνομα	τύπος
x	int
y	int

δομή: **Σημείο**

```
struct point p;  
p.x = 4825;  
p.y = 55;
```



Εργαλεία γλώσσας προγραμματισμού

- Πίνακες (arrays)

- **συνεχόμενες θέσεις**
- **όμοιου τύπου μεταβλητές**
 - απλές
 - σύνθετες
- **πολλαπλές διαστάσεις**
 - Αποθήκευση ανά γραμμές ή ανά στήλες
 - ανάλογα με τη γλώσσα

```
char t[5];  
t[3] = 'A';
```

600		
602	??	??
604	??	41
606	??	
...		
...		

Εργαλεία γλώσσας προγραμματισμού

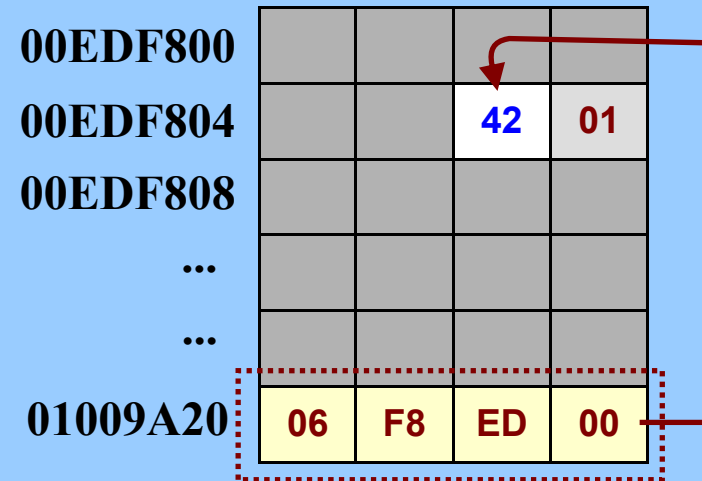
- Συμβολοσειρές στη γλώσσα C
 - **strings**
 - Πίνακες χαρακτήρων
 - Σε συνεχόμενες θέσεις
 - Σύμβολο τερματισμού
- Άλλες γλώσσες προσφέρουν πιο ολοκληρωμένη υποστήριξη σε strings

```
char str[] =  
{ "ABCD" };
```

600		
602	41	42
604	43	44
606	0	
...		
...		

Εργαλεία γλώσσας προγραμματισμού

- Δείκτης (pointer)
 - περιέχει διεύθυνση μνήμης
 - “δείχνει” σε κάποια θέση μνήμης
 - τι δείχνει ο δείκτης;
 - ό,τι ορίζει ο προγραμματιστής!
 - αν δείχνει σε χαρακτήρα (ASCII) ...
 - ...τότε δείχνει στον χαρακτήρα ‘B’
 - αν δείχνει σε ακέραιο (short int) ...
 - ... τότε δείχνει στον ακέραιο 322
 - 0142 hex - little-endian



Εργαλεία γλώσσας προγραμματισμού

- **Μεταβλητές τύπου δείκτη**
 - περιέχουν τη **διεύθυνση**
 - οποιαδήποτε διεύθυνση στον χώρο μνήμης!
 - μιας άλλης μεταβλητής
 - οποιουδήποτε τύπου!
- Έμμεση προσπέλαση μέσω μεταβλητής δείκτη
“δώσε μου την τιμή που βρίσκεται στη διεύθυνση μνήμης...
...η οποία (η διεύθυνση!) περιέχεται ως τιμή στον χώρο μνήμης...
...που έχει δεσμευθεί για τη μεταβλητή-δείκτη”

```
... = *ptr;
```

Κατηγορίες μεταβλητών σε δομημένες γλώσσες

•Σφαιρικές

- στατική θέση
- γνωστή κατά τη μεταγλώττιση

mem[address]

•Τοπικές

- πλαίσιο (frame)
- μετατόπιση (offset)
- Ορισμός θέσης κατά την εκτέλεση

mem[framebase+offset]

•Δυναμικής δέσμευσης

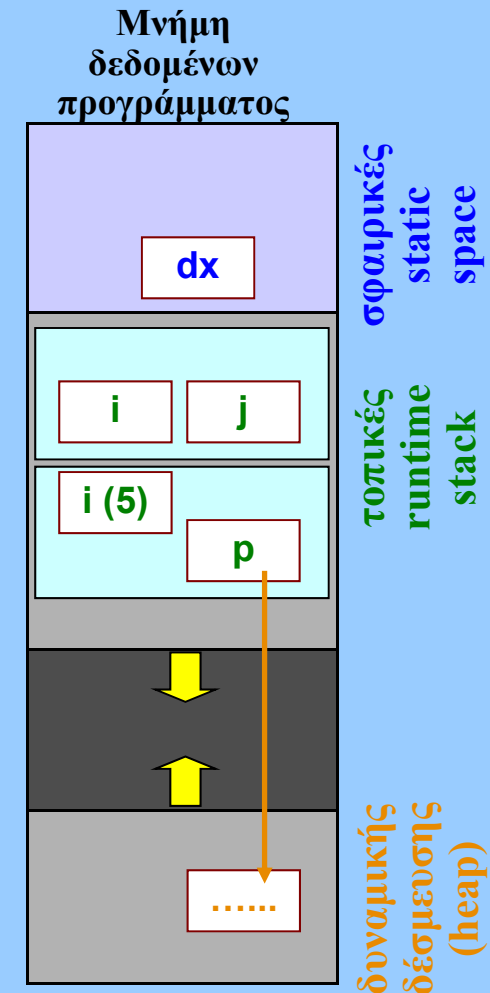
- Χώρος heap
- Δέσμευση και απελευθέρωση
- Προσπέλαση μέσω δεικτών

mem[mem[ptr address]]

```
#include <stdio.h>
void functA(int);
double dx;

main() {
    int i,j;
    functA(5);
}

void functA(int i) {
    char *p;
    p = malloc(i);
}
```



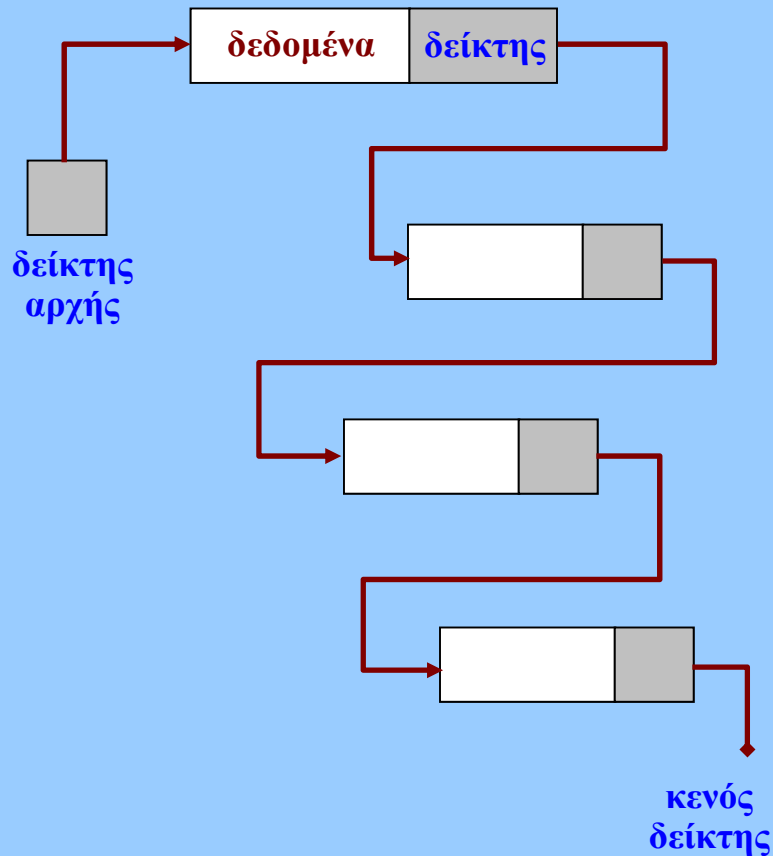
Υλοποίηση αφηρημένων τύπων δεδομένων

- **Με πίνακες (arrays)**
 - Πιο συμπαγής αποθήκευση στη μνήμη
 - Συνεχόμενες θέσεις μνήμης
 - Αποδοτικότερη διαχείριση μνήμης συστήματος
 - Στατικού μεγέθους
 - (μέγιστο) πλήθος στοιχείων πρέπει να είναι γνωστό
 - ή να υπάρχει εύκολος τρόπος αλλαγής μεγέθους πίνακα
 - Εξαρτάται από γλώσσα προγραμματισμού

Υλοποίηση αφηρημένων τύπων δεδομένων

- **Με διασυνδεδεμένους κόμβους**
 - **Κόμβος = δεδομένα + δείκτες**
 - Δυναμική δέσμευση-απελευθέρωση κόμβων
 - Ευκολότερη διαχείριση δομών προγραμματιστικά
 - **Δυναμικού μεγέθους**
 - Πλήθος στοιχείων αλλάζει δυναμικά
 - Απαιτείται αυξημένη φροντίδα στη διαχείριση μνήμης

Γραμμικές λίστες

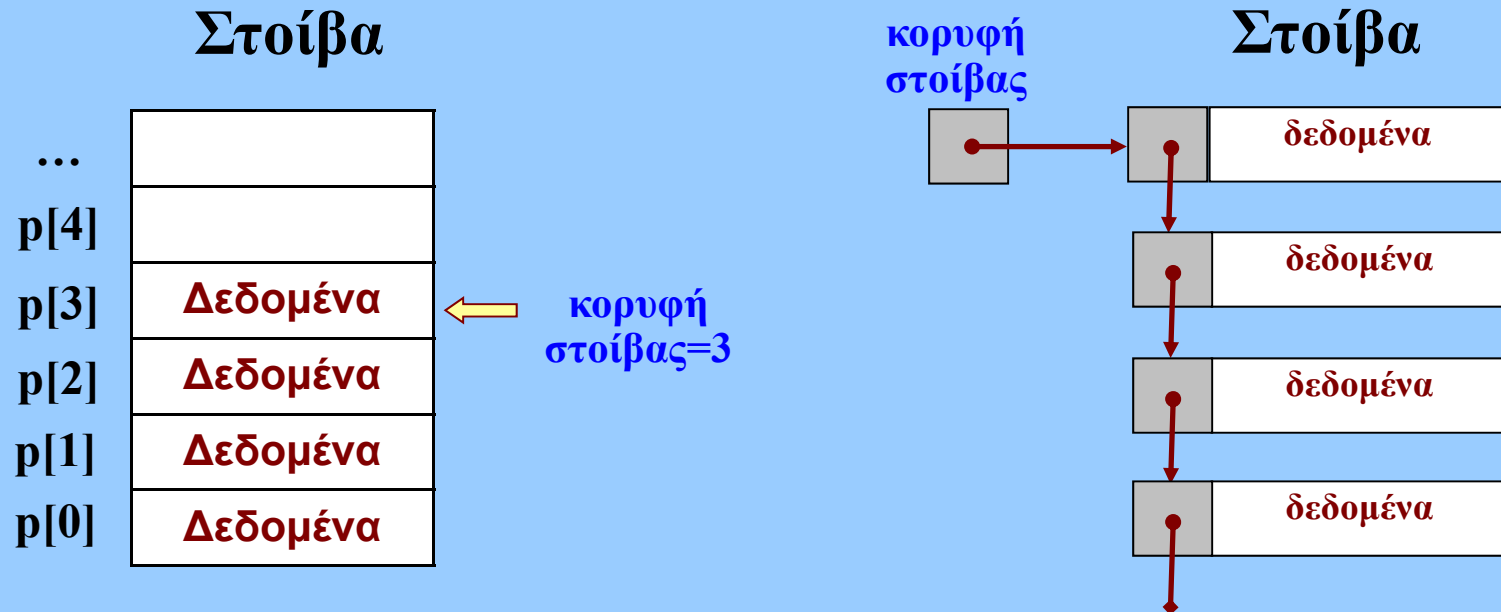


p[0]	δεδομένα	επόμενος
p[1]		
p[2]		
p[3]	..δεδομένα..	1
...		
...		

- Υλοποίηση με διασυνδεδεμένους κόμβους

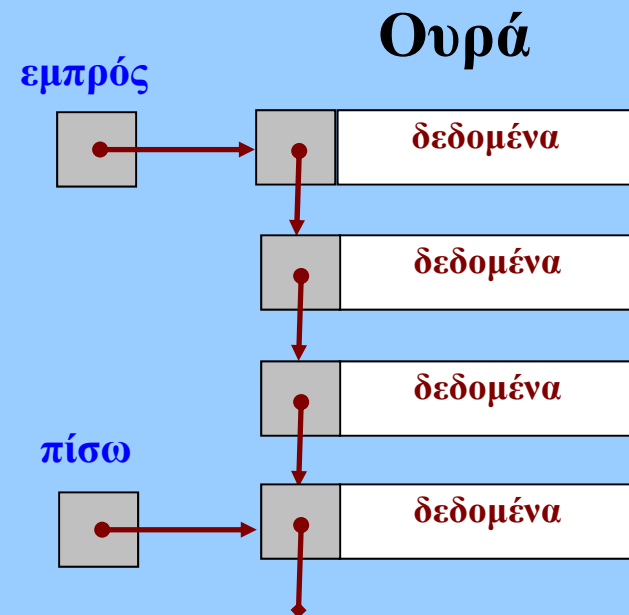
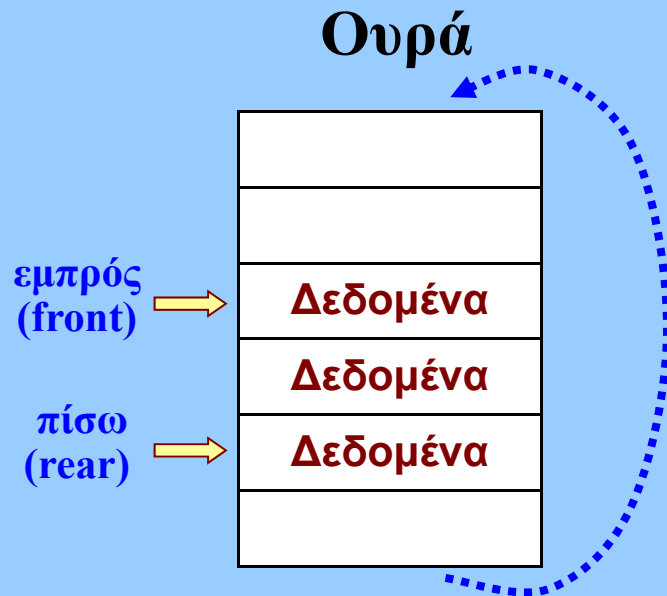
- Υλοποίηση με πίνακες

Περιορισμένες γραμμικές λίστες



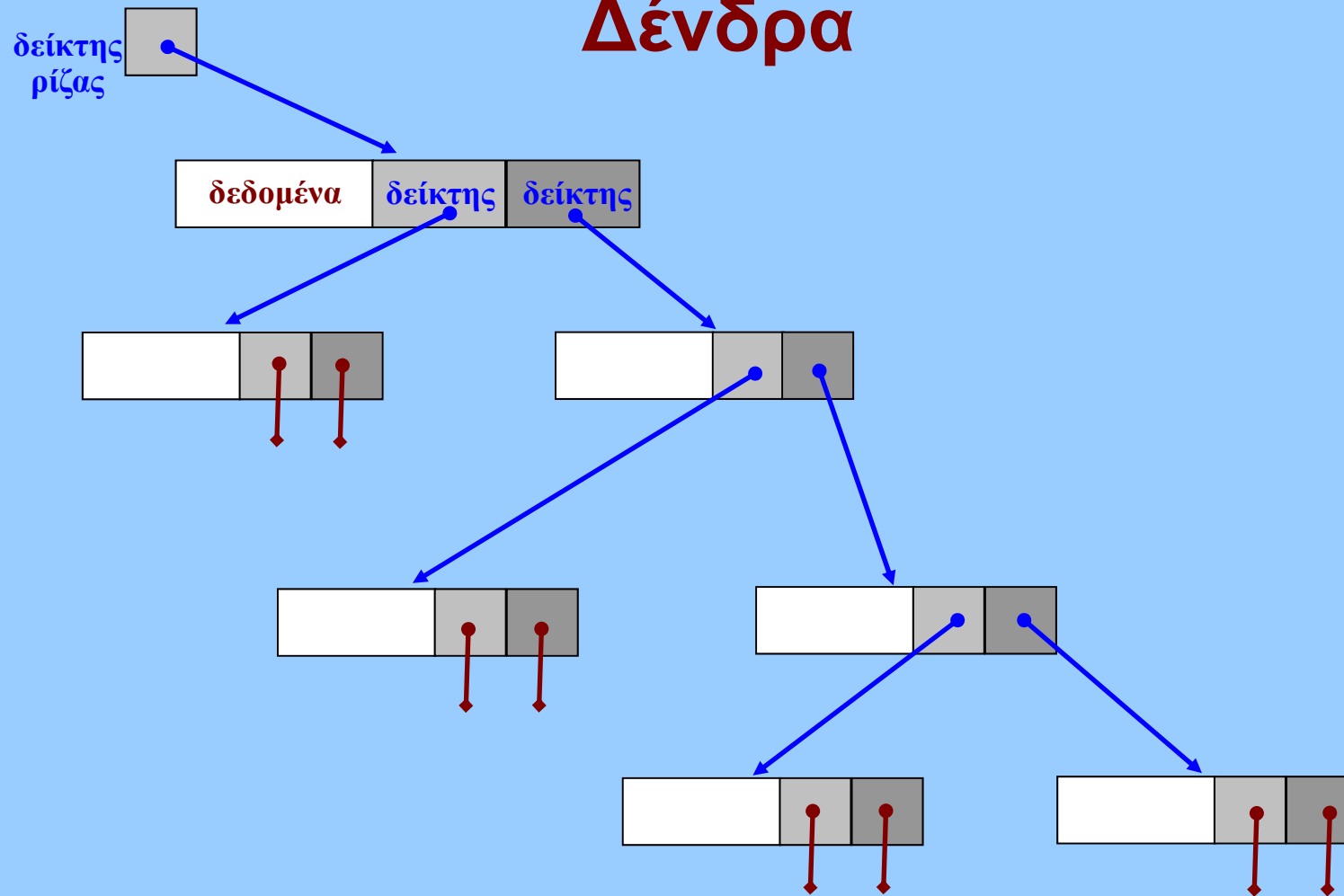
- Υλοποίηση με πίνακες
 - Μεταβλητή “κορυφή στοίβας”
- Εναλλακτικά με διασυνδεδεμένους κόμβους
 - Δείκτες από κορυφή προς βάση

Περιορισμένες γραμμικές λίστες



- Υλοποίηση με πίνακες
 - Μεταβλητές “εμπρός” “πίσω”
 - Κυκλική ουρά
 - Επαναφορά μεταβλητών
- Εναλλακτικά με διασυνδεδεμένους κόμβους
 - Δείκτες από εμπρός προς πίσω

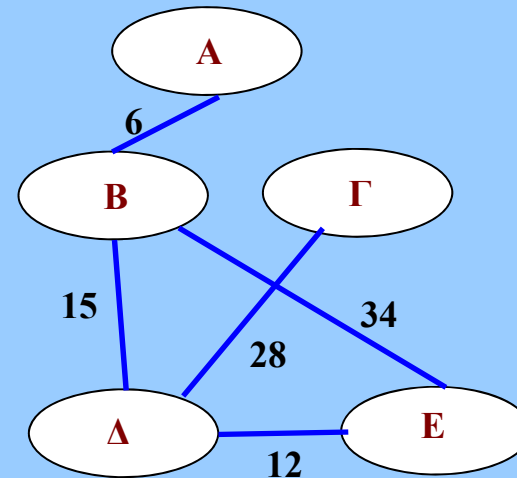
Δένδρα



- Υλοποίηση με διασυνδεδεμένους κόμβους
 - Εναλλακτικά με πίνακες;

Γράφοι

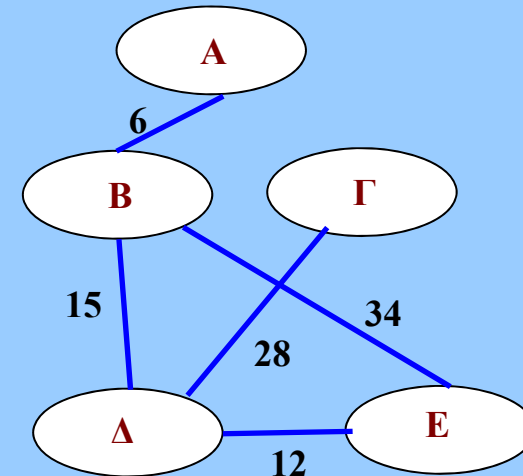
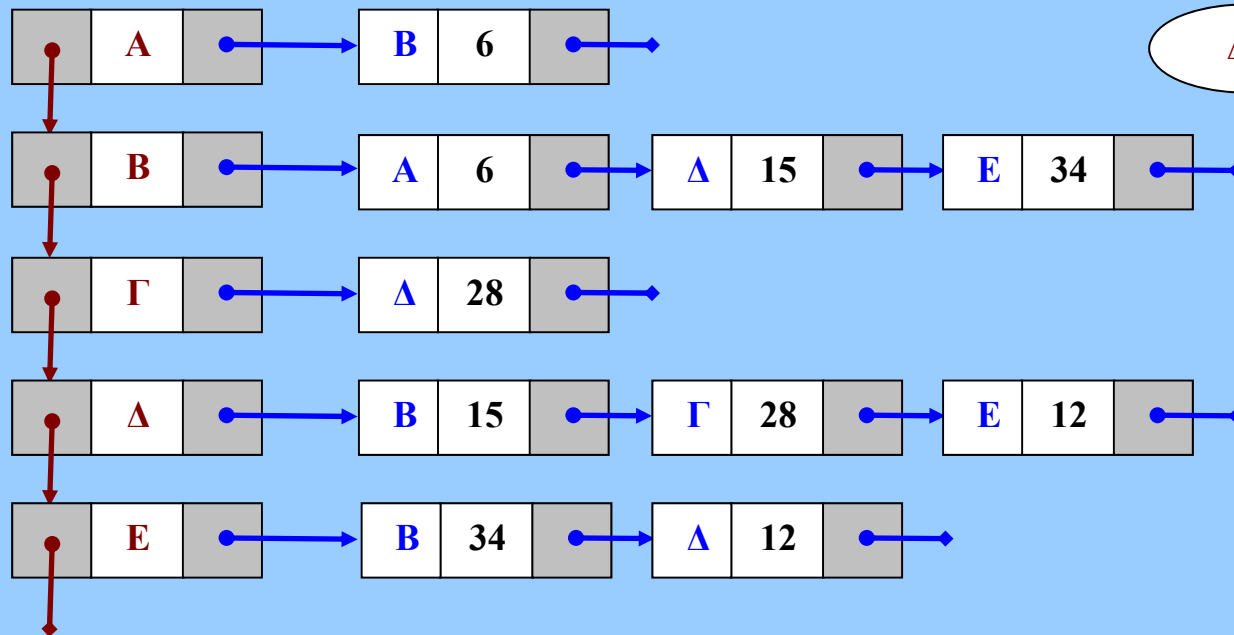
	A	B	Γ	Δ	E
A	0	6	0	0	0
B	6	0	0	15	34
Γ	0	0	0	28	0
Δ	0	15	28	0	12
E	0	34	0	12	0



- Πίνακας γειτνίασης (adjacency matrix)

Γράφοι

- **Λίστα γειτνίασης (adjacency list)**



Χρήση βιβλιοθηκών υλοποίησης αφηρημένων τύπων δεδομένων

Πρόγραμμα χρήστη

```
#include <stdio.h>
#include <list.h>

main() {

    lp = createList();

    insertList(lp, 50);

}
```

Include files

Δηλώσεις μεταβλητών,
τύπων και συναρτήσεων
που μπορεί να
χρησιμοποιήσει ο χρήστης

object files

Εκτελέσιμος κώδικας που
υλοποιεί τις συναρτήσεις
της βιβλιοθήκης

Π.χ. βιβλιοθήκη
υλοποίησης λιστών
ακεραίων

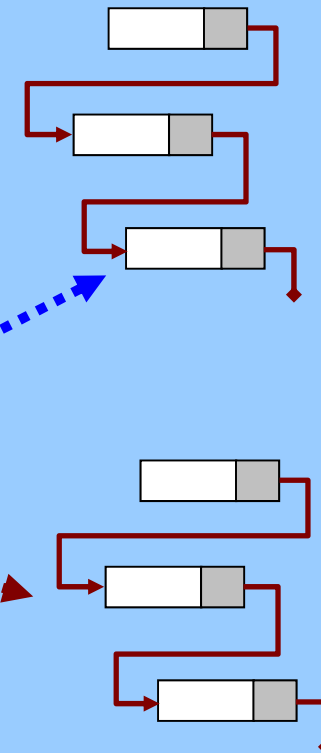
Επιλογή στιγμιότυπου δομής σε γλώσσες όπως η C

Πρόγραμμα χρήστη

```
#include <stdio.h>
#include <list.h>

main() {
    "handle"
    lp1 = createList();
    lp2 = createList();

    insertList(lp1, 50);
}
```



Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2006-07

Δομές Αρχείων

Μ.Στεφανιδάκης

Στόχοι του μαθήματος

- **Αποθήκευση πληροφορίας**
 - **Οργανώσεις αρχείων**
 - Σε επίπεδο υλικού και λογισμικού
 - **Η άποψη της εφαρμογής για τα αρχεία**
 - Σειριακή και τυχαία προσπέλαση
 - Αρχεία κειμένου και δυαδικά αρχεία

Αποθήκευση δεδομένων

- **Δεδομένα επεξεργασίας**
 - Είσοδος – έξοδος σε προγράμματα εφαρμογών
 - (τα ίδια τα εκτελέσιμα προγράμματα)
 - Ολοένα αυξανόμενος όγκος δεδομένων
- **Ανάγκες αποθήκευσης**
 - Δευτερεύων αποθηκευτικός χώρος – μετά την κύρια μνήμη
 - Κύρια αποθηκευτικά μέσα
 - Δίσκοι, CDs, DVDs
 - Ταινίες (παλαιότερα)
 - Επίσης: αποθήκευση μέσω δικτύου

Τι είναι ένα “αρχείο”;

- Αρχεία
 - Αποθηκευτικές συλλογές δεδομένων
 - Για την εφαρμογή:
 - Λογικές μονάδες δεδομένων
 - Αφαιρετική εικόνα
 - Στην πραγματικότητα:
 - Φυσικά αποθηκευτικά μέσα
 - Ελεγκτές συσκευών
 - Συστήματα αρχείων - δομές καταλόγων
 - Διαχείριση από λειτουργικό σύστημα
 - Βιβλιοθήκες Ε/Ε

Η πολυεπίπεδη υλοποίηση ενός αρχείου

Βλέπει το αρχείο ως μια λογική δομή ανάκτησης-αποθήκευσης δεδομένων

Παρέχουν τις συναρτήσεις υψηλού επιπέδου για τον χειρισμό των αρχείων

Παρέχει τις βασικές συναρτήσεις για την προσπέλαση των αρχείων

Αναλαμβάνουν τις λεπτομέρειες επικοινωνίας με τις συσκευές E/E

Αποθηκεύουν την πληροφορία σε σειρές δυαδικών αριθμών

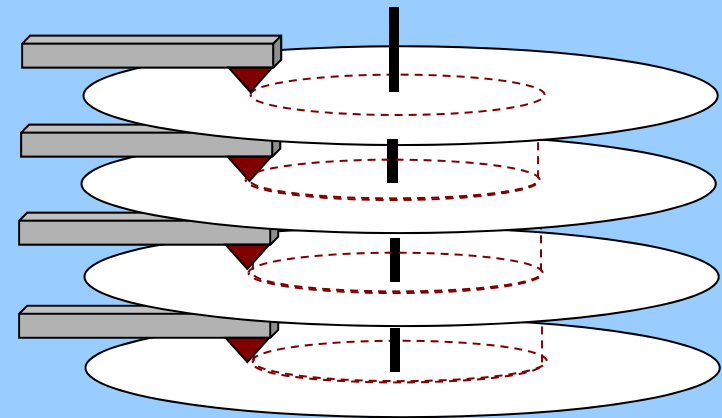


Αποθηκευτικές συσκευές

- **Μαγνητικοί Δίσκοι**
 - Οι κύριες συσκευές αποθήκευσης δεδομένων
 - **2004: έως 200GB, 0,5\$/GB**
- **Παλαιότερα:**
 - Μαγνητικές ταινίες
 - Για φύλαξη δεδομένων
- **CDs, DVDs..**
- **Μεταφορά τμημάτων δεδομένων**
 - **Block devices**

Οργάνωση ενός δίσκου

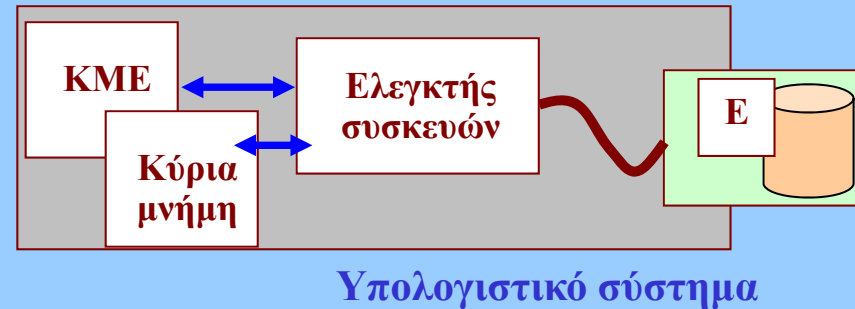
- 1-4 επιφάνειες διπλής όψης με διάμετρο έως 3.5 ίντσες
- Έως 50.000 tracks / επιφάνεια
- Έως 500 τομείς (sectors) / track
- 512+ bytes / sector
- Σταθερή απόσταση ανάμεσα σε bits
- Περιστροφή 7.200 – 15.000 rpm
- Τομέας
 - Αριθμός sector
 - Πληροφορία
 - Κώδικας διόρθωσης λαθών



Ανάγνωση- εγγραφή τομέα

- **Τομέας:** η μικρότερη ποσότητα πληροφορίας που μεταφέρεται
- **Μεταφορά σε 3 στάδια**
 1. Μετακίνηση κεφαλών στο κατάλληλο track (**seek**)
 2. Αναμονή προσέγγισης επιλεγμένου τομέα
 3. Μεταφορά δεδομένων
 - Εξαρτάται από μέγεθος τομέα, ταχύτητα περιστροφής και πυκνότητα εγγραφής
 - **2004: 30-80MB/sec**

Ο ελεγκτής δίσκου



- **Ελεγκτής ενσωματωμένος στον δίσκο**
 - Έλεγχος μηχανικών και ηλεκτρονικών μερών δίσκου
 - Πρωτόκολλο επικοινωνίας με υπολογιστή
 - Διαχείριση κρυφής μνήμης δίσκου
- **Ελεγκτής συσκευών αποθήκευσης**
 - PATA, SATA, SCSI κλπ
 - Εντολές προς τις συσκευές αποθήκευσης
 - Μεταφορές δεδομένων από-προς την κύρια μνήμη
 - Επικοινωνία με ΚΜΕ

Έλεγχος συσκευών

- Οι συσκευές αποθήκευσης είναι πολύ αργές σε σχέση με την ΚΜΕ
- **Υποχρεωτική αναμονή για ολοκλήρωση αίτησης**
 - Polling: έλεγχος κατάστασης συσκευής σε τακτά διαστήματα
 - Interrupts: ο ελεγκτής ειδοποιεί ΚΜΕ
 - Δυνατότητα εκτέλεσης άλλων λειτουργιών
- **Μεταφορές δεδομένων από ελεγκτή συσκευής στην κύρια μνήμη**
 - Σήμερα: χωρίς τη μεσολάβηση ΚΜΕ
 - Direct Memory Access (DMA)

Οδηγοί συσκευών (device drivers)

- **Μέρος του Λειτουργικού Συστήματος**
 - **Συνδέεται στενά με ελεγχόμενη συσκευή**
- **Παροχή γενικής διεπαφής (interface)**
 - **Στο χαμηλότερο επίπεδο επικοινωνίας λογισμικού-υλικού**
- **Ένας οδηγός συσκευής αναλαμβάνει τις λεπτομέρειες επικοινωνίας με τη συσκευή**
 - **Δημιουργία εντολών προς τη συσκευή**
 - **Έλεγχος κατάστασης των αιτήσεων**

Διαχείριση αρχείων από ΛΣ

- **Συναρτήσεις χαμηλού επιπέδου**
 - **Open:** προετοιμασία για εκτέλεση λειτουργιών
 - **Create:** δημιουργία νέας δομής αρχείου
 - **Read/write:** αιτήσεις ανάγνωσης/εγγραφής δεδομένων
 - **Seek:** μετακίνηση μέσα στο αρχείο
 - **Close:** ολοκλήρωση λειτουργιών σε αρχείο
 - **Unlink:** διαγραφή αρχείου
- **Δυνατότητα απευθείας κλήσης συναρτήσεων από εφαρμογές**

Διαχείριση αρχείων από ΛΣ

- Έλεγχος επιπέδων προνομίων
 - Εάν η εφαρμογή έχει δικαίωμα να προσπελάσει ένα αρχείο
 - Και τι είδους λειτουργίες επιτρέπεται να εκτελέσει
- Μεταφορά δεδομένων εφαρμογής από/προς προσωρινές θέσεις αποθήκευσης του λειτουργικού συστήματος κατά την ανάγνωση/εγγραφή από/προς συσκευές
 - Χρονοβόρα διαδικασία
 - Πρέπει να αποφεύγεται όσο είναι δυνατόν

Βιβλιοθήκες Ε/Ε

- Υψηλού επιπέδου διαχείριση αρχείων
 - Π.χ. η **stdio** (standard I/O) βιβλιοθήκη της C
- Σύνδεση με εκτελέσιμο πρόγραμμα εφαρμογής
 - **Στατικά ή δυναμικά**
- Σε ποιο αρχείο απευθυνόμαστε;
 - File Handles ή File Pointers
- Ειδικές συσκευές Ε/Ε
 - Πληκτρολόγιο, οθόνη...

Συστήματα αρχείων

- **Συστήματα αρχείων (file systems)**
 - Οργάνωση των λογικών δομών (αρχείων) στο φυσικό μέσο αποθήκευσης (δίσκοι)
 - Επηρεάζει την απόδοση του συστήματος αποθήκευσης
- **Πληροφορίες ανά αρχείο**
 - Όνομα
 - Χαρακτηριστικά
 - Ημ/νίες δημιουργίας και τροποποίησης
 - Τύπος αρχείου
 - Δικαιώματα προσπέλασης
 - Λίστα τμημάτων του αρχείου και κατανομή τους στην φυσική επιφάνεια του δίσκου

Η άποψη της εφαρμογής για τα αρχεία

- **Μέθοδοι προσπέλασης των αρχείων**
 - **Σειριακά** (ακολουθιακά): τα δεδομένα του αρχείου προσπελούνται το ένα μετά το άλλο
 - **Τυχαία**: δυνατότητα μετακίνησης σε οποιοδήποτε σημείο του αρχείου
 - **Ακόμα και προς τα πίσω**
- **Είδη αρχείων**
 - **Δεδομένων**, κειμένου-δυναμικά
 - **Εκτελέσιμα**, δυναμικά

Σειριακή προσπέλαση

- **Δείκτης επόμενης θέσης**
 - Από ποιο σημείο του αρχείου θα επιστραφούν τα επόμενα δεδομένα
- **Σειριακή προσπέλαση**
 - Άνοιγμα αρχείου
 - Ανάγνωση δεδομένων
 - Αύξηση δείκτη επόμενης θέσης
 - Επανάληψη διαδικασίας μέχρι το τέλος του αρχείου
- **Ταιριάζει σε συσκευές μαγνητικής ταινίας**
 - Αδυναμία μετακίνησης σε τυχαίες θέσεις μέσα στο αρχείο
- **Όταν δεν υπάρχει κάποια μορφή οργάνωσης των δεδομένων μέσα στο αρχείο**
 - Όπως τα αρχεία απλού κειμένου

Τυχαία προσπέλαση

- Ο **δείκτης επόμενης θέσης** μπορεί να μετακινηθεί οπουδήποτε μέσα στο αρχείο
 - Κατά τη διάρκεια της ανάγνωσης ή εγγραφής
 - Όσες φορές απαιτείται
- Μετακίνηση **δείκτη επόμενης θέσης**
 - Ελευθερία μετακίνησης n bytes μπροστά ή πίσω
 - Από τρέχουσα θέση
 - Από αρχή ή τέλος
- Χρήσιμο, όταν η πληροφορία είναι π.χ. οργανωμένη σε εγγραφές
 - Μετακίνηση μεταξύ εγγραφών και προς τις δύο κατευθύνσεις

Αρχεία κειμένου

- Περιέχουν απλούς χαρακτήρες
 - Συνήθως 1 byte ανά χαρακτήρα
 - Χωρίς ιδιαίτερη οργάνωση των δεδομένων
- Ειδικοί Χαρακτήρες
 - Τερματισμού (end of file - EOF)
 - Νέας γραμμής (\n – 13,10 ή μόνο 10)
- Ανάγνωση – Εγγραφή
 - Ανά χαρακτήρα
 - Ανά γραμμή κειμένου
- Τροποποίηση
 - Μόνο προσθήκη κειμένου στο τέλος
- Σειριακή επεξεργασία
 - Δεν έχει νόημα η τυχαία μετακίνηση μέσα στο αρχείο

Παράδειγμα: απλό αρχείο κειμένου

```
sample1.txt
00000000 50 61 72 65 74 6F 20 65 66 66 69 63 69 65 6E 63 Pareto efficienc
00000010 79 2C 20 6F 72 20 50 61 72 65 74 6F 20 6F 70 74 y, or Pareto opt
00000020 69 6D 61 6C 69 74 79 20 69 73 20 61 20 63 65 6E imality is a cen
00000030 74 72 61 6C 20 63 6F 6E 63 65 70 74 20 69 6E 20 tral concept in
00000040 67 61 6D 65 20 74 68 65 6F 72 79 0D 0A 77 69 74 game theory..wit
00000050 68 20 62 72 6F 61 64 20 61 70 70 6C 69 63 61 74 h broad applicat
00000060 69 6F 6E 73 20 69 6E 20 65 63 6F 6E 6F 6D 69 63 ions in economic
00000070 73 2C 20 65 6E 67 69 6E 65 65 72 69 6E 67 20 61 s, engineering a
00000080 6E 64 20 74 68 65 20 73 6F 63 69 61 6C 20 73 63 nd the social sc
00000090 69 65 6E 63 65 73 2E iences.
```

Κείμενο ANSI

Διαδικά αρχεία

- Η αναπαράσταση των δεδομένων καθορίζεται από την εφαρμογή
 - Μορφή οργάνωσης (format) των δεδομένων σε πεδία
 - Μπορεί να περιγράψει δεδομένα οποιουδήποτε είδους
 - Όλες οι τιμές μπορούν να αποθηκευθούν
 - όχι μόνο χαρακτήρες
- Ανάγνωση –εγγραφή
 - Κάθε φορά, όσα bytes επιθυμεί η εφαρμογή
 - Τροποποίηση σε οποιοδήποτε σημείο
- Τυχαία προσπέλαση
 - Η οργάνωση των δεδομένων ευνοεί τη μετακίνηση από πεδίο σε πεδίο
 - Υπάρχει και η δυνατότητα σειριακής προσπέλασης

Παράδειγμα: αρχείο εικόνας GIF

```
banner.gif
00000000 47 49 46 38 39 61 AC 00 64 00 F7 00 00 31 29 29 GIF09a..d...l))
00000010 39 31 31 39 31 39 39 39 39 42 31 31 42 39 39 42 911919999B11B99B
00000020 39 42 42 42 42 42 42 4A 4A 31 31 4A 39 31 4A 39 9BBBBBBJJ11J91J9
00000030 39 4A 42 31 4A 42 39 4A 42 42 4A 42 4A 4A 4A 39 9JB1JB9JBBJBJJJ9
00000040 4A 4A 42 4A 4A 4A 4A 4A 52 52 39 31 52 39 42 52 JJBJJJJRR91R9BR
00000050 42 39 52 42 42 52 42 4A 52 42 52 52 4A 39 52 4A B9RBBRBJRBRJ9RJ
00000060 42 52 4A 4A 52 4A 52 52 52 42 52 52 4A 52 52 52 BRJJRJRBRBRJRBR
00000070 52 52 5A 52 5A 52 5A 4A 42 5A 4A 4A 5A 4A 52 5A RRZRZRZJBZJJZJRZ
00000080 52 4A 5A 52 52 5A 52 5A 5A 5A 52 5A 5A 5A 5A 5A RJZRRZRZ2Z2RZ2Z2Z
00000090 63 5A 63 63 63 52 52 63 52 5A 63 5A 52 63 5A 5A eZcccRRcRZcZRcZ2
000000A0 63 5A 63 63 63 5A 63 63 63 63 63 6B 63 6B 63 63 eZcccZccccckckcc
000000B0 6B 6B 6B 52 5A 6B 5A 5A 6B 5A 63 6B 63 5A 6B 63 kkkR2k22k2ckc2kc
000000C0 63 6B 63 6B 6B 63 73 6B 6B 63 6B 6B 6B 6B 6B 73 ckkckckckckckckck
000000D0 73 5A 63 73 63 63 73 63 6B 73 6B 63 73 6B 6B 73 sZccccckckckckck
  3 73 6B 7B 73 73 6B 73 73 73 73 73 73 7B 73 7B kskk(sskssss(s(s(
  B 6B 6B 7B 6B 73 7B 73 6B 7B 73 73 7B 73 7B ((kk(kk(ss(ss(ss(
  3 84 7B 7B 7B 7B 7B 84 7B 7B 8C 7B 84 84 7B (s.(((((.(((..(
  C 84 73 6B 84 73 73 84 73 7B 84 7B 6B 84 7B ...sk.ss.s(.k.(
00000120 73 84 7B 7B 84 7B 84 84 7B 8C 84 84 84 84 84 8C s.(((..(.....
00000130 84 84 94 84 8C 94 8C 7B 73 8C 7B 7B 8C 7B 84 8C .....(s.(((..(.
```



Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2006-07

Βάσεις Δεδομένων

Μ.Στεφανιδάκης

Στόχοι του μαθήματος

- **Αποθήκευση πληροφορίας**
 - Οργανώσεις αρχείων
 - Ευρετήρια
 - Κατακερματισμένα αρχεία
- **Βάσεις δεδομένων**
 - Αρχιτεκτονική βάσεων δεδομένων
 - Το μοντέλο οντοτήτων-σχέσεων
 - Μοντέλα βάσεων δεδομένων

Αρχεία (επανάληψη)

- Αρχεία
 - Λογικές μονάδες
 - Στα φυσικά αποθηκευτικά μέσα
 - Συστήματα αρχείων - δομές καταλόγων
 - Διαχείριση από λειτουργικό σύστημα
 - Συνεργασία υλικού-λογισμικού
 - Η άποψη της εφαρμογής
 - Συλλογές δεδομένων
 - Μέθοδος προσπέλασης
 - Σειριακή – Τυχαία
 - Είδη αρχείων
 - Κειμένου - Διαδικά

Οργάνωση πληροφορίας

Εργαζόμενος

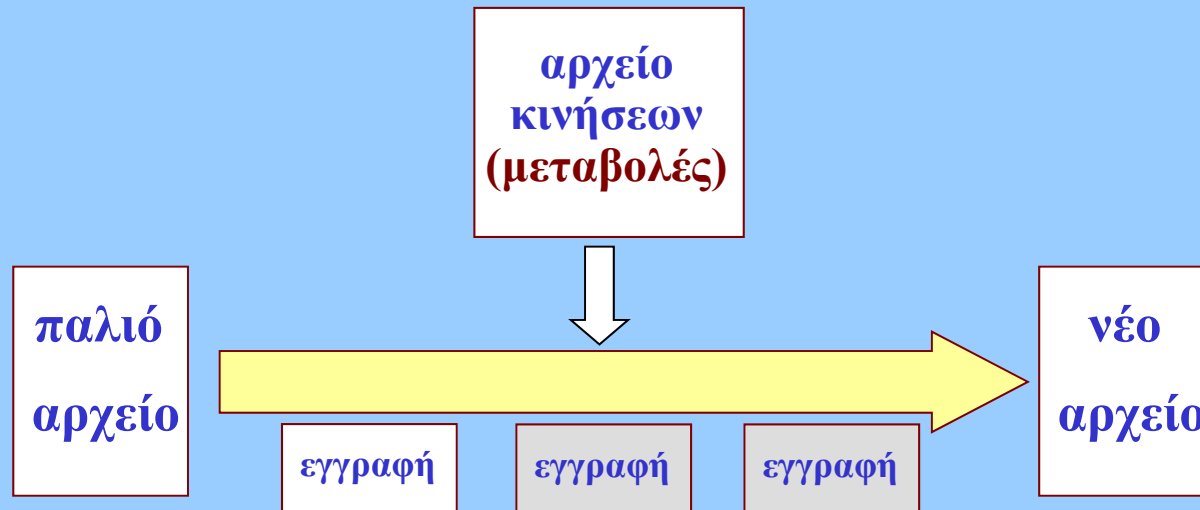
Όνομα	Επώνυμο
ΑΦΜ	Μισθός
...άλλη πληροφορία...	

- **Εγγραφή** (record)
 - Πεδία δεδομένων
- Μετεξέλιξη της χειρόγραφης οργάνωσης
 - καρτέλες...
- Αποθηκευμένη πληροφορία
 - σύνολο εγγραφών

Προσπέλαση πληροφορίας

- Προσπέλαση εγγραφών σε αρχεία
 - Αναζήτηση δεδομένων
 - Ανάκτηση - Ενημέρωση δεδομένων
- **Κλειδί (key)**
 - ένα ή περισσότερα πεδία
 - μοναδικός προσδιορισμός μιας εγγραφής
 - εύρεση στόχου για ανάκτηση – ενημέρωση πληροφορίας
- Μέθοδος προσπέλασης
 - με ποιον τρόπο προσπελούνται οι εγγραφές
 - καθορίζει τον τρόπο αποθήκευσης στα αρχεία
 - εσωτερική οργάνωση αρχείων

Σειριακή προσπέλαση

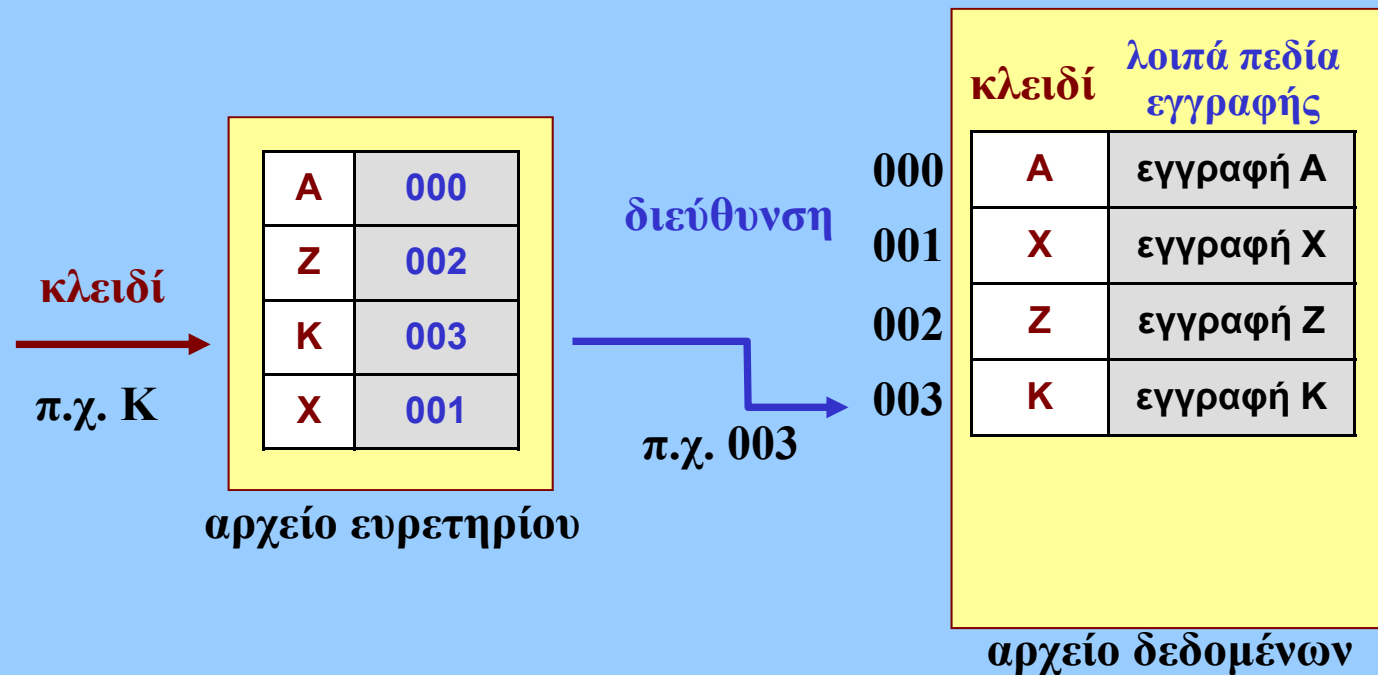


- Διαδοχική προσπέλαση εγγραφών
 - η μία μετά την άλλη
 - **σειριακή αναζήτηση**
- Ταιριάζει όταν χρησιμοποιούνται ταινίες
 - ακολουθιακή προσπέλαση δεδομένων

Τυχαία προσπέλαση

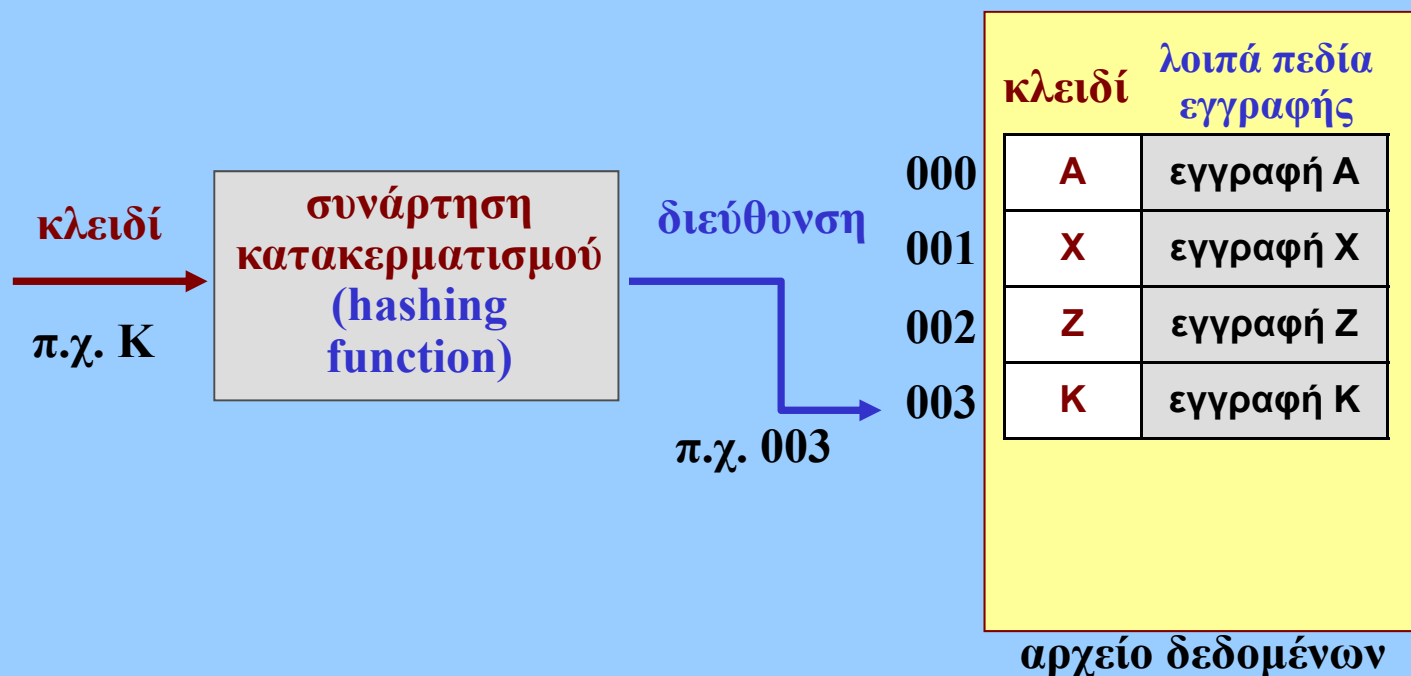
- Απευθείας προσπέλαση επιλεγμένης εγγραφής
 - απαιτείται η **διεύθυνση** της εγγραφής μέσα στο αρχείο
 - όταν η διεύθυνση δεν είναι γνωστή \Rightarrow **αναζήτηση**
 - χρησιμοποιώντας ένα **κλειδί**
- Αναζήτηση εγγραφών
 - σειριακά
 - με κάποιον αποδοτικό αλγόριθμο αναζήτησης
- Αναζήτηση σε **μικρό μέρος** κάθε εγγραφής
 - πεδίο – κλειδί
 - αν χρησιμοποιηθεί δεύτερο **μικρότερο** αρχείο;

Χρήση αρχείων ευρετηρίου



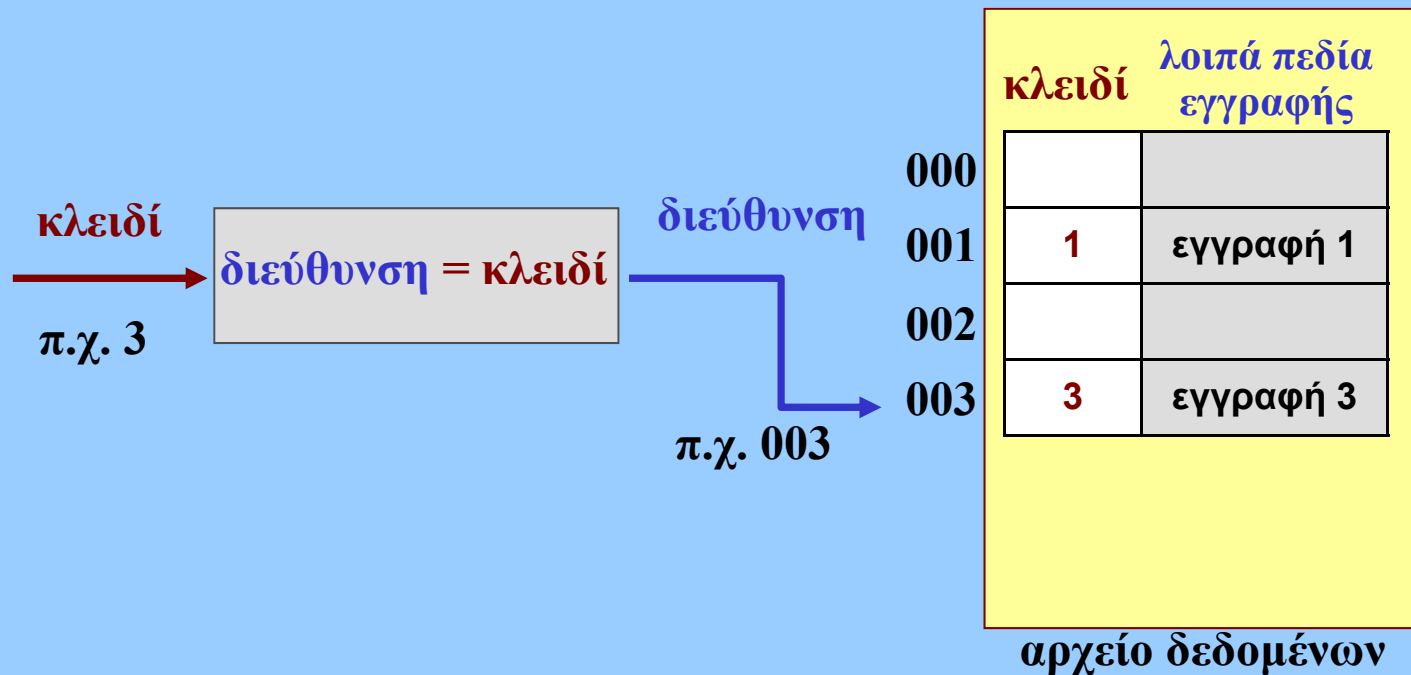
- Πρόσθετο αρχείο **ευρετηρίου** (index)
 - περιέχει τα κλειδιά αναζήτησης
 - **οργάνωση ανάλογη του αλγορίθμου αναζήτησης**
 - μικρότερο μέγεθος \Rightarrow μπορεί να φορτωθεί στη μνήμη
 - τα κυρίως δεδομένα περιέχονται σε δεύτερο (πολύ μεγαλύτερο) αρχείο
 - η αναζήτηση στο ευρετήριο επιστρέφει τη διεύθυνση στο μεγάλο αρχείο δεδομένων

Κατακερματισμένα αρχεία



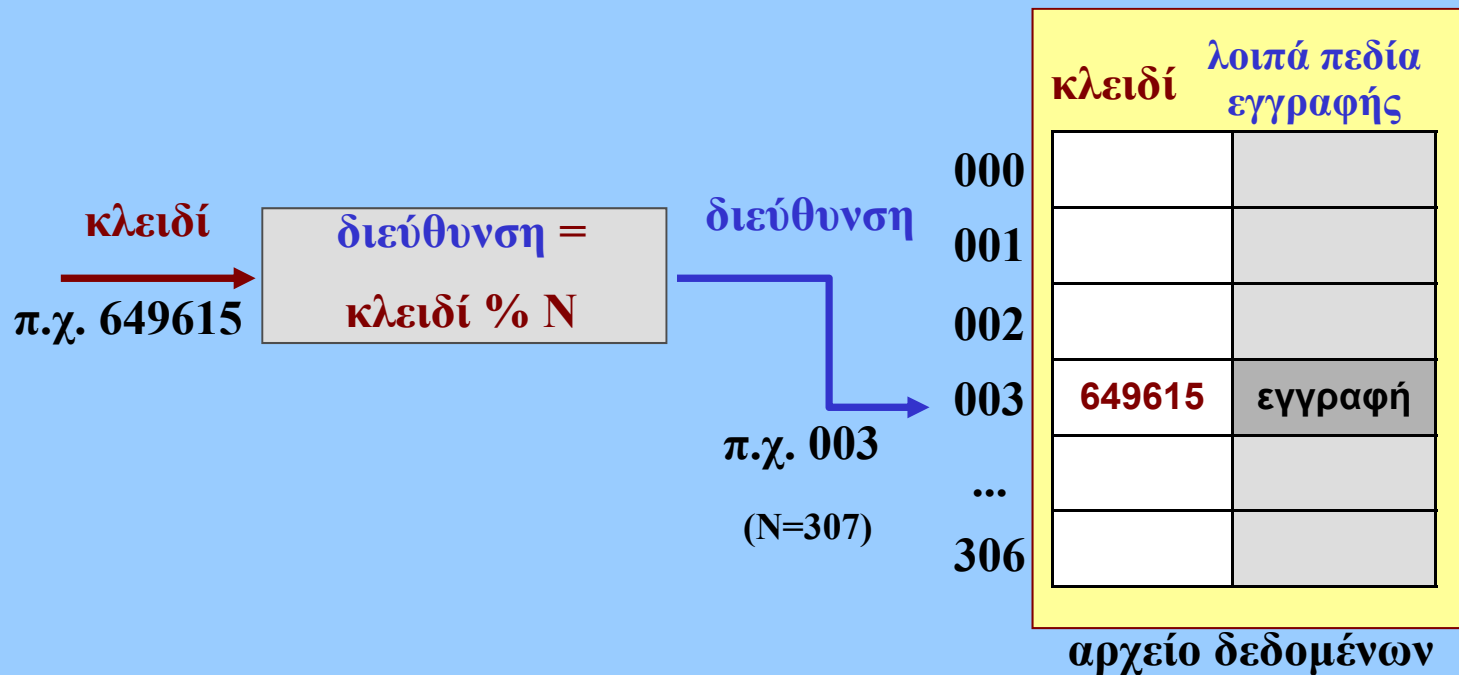
- Κατακερματισμός (hashing)
 - πολύ αποδοτική μέθοδος οργάνωσης και αναζήτησης εγγραφών ($O(1)$)
 - Αντιστοίχιση κλειδιού σε διεύθυνση εγγραφής μέσω μαθηματικής συνάρτησης
 - ίδια αντιστοίχιση τόσο για εισαγωγή όσο και για αναζήτηση

Συναρτήσεις κατακερματισμού



- Άμεσος κατακερματισμός (direct hashing)
 - διεύθυνση = κλειδί
 - μοναδική αντιστοίχιση (ιδανική περίπτωση)
 - αλλά σπάνια μπορεί να χρησιμοποιηθεί στην πράξη
 - θα πρέπει οι διαθέσιμες διευθύνσεις να είναι όσες τα πιθανά κλειδιά!
 - κι αν το κλειδί είναι π.χ. ο ΑΦΜ;

Συναρτήσεις κατακερματισμού (2)



- Κατακερματισμός υπολοίπου διαίρεσης (division remainder hashing)
 - $\text{διεύθυνση} = \text{κλειδί} \bmod \text{μέγιστο_πλήθος_εγγραφών_αρχείου (N)}$
 - όχι πάντοτε μοναδική αντιστοίχιση \Rightarrow σύγκρουση!
 - συχνά ως **N** επιλέγεται **πρώτος** αριθμός

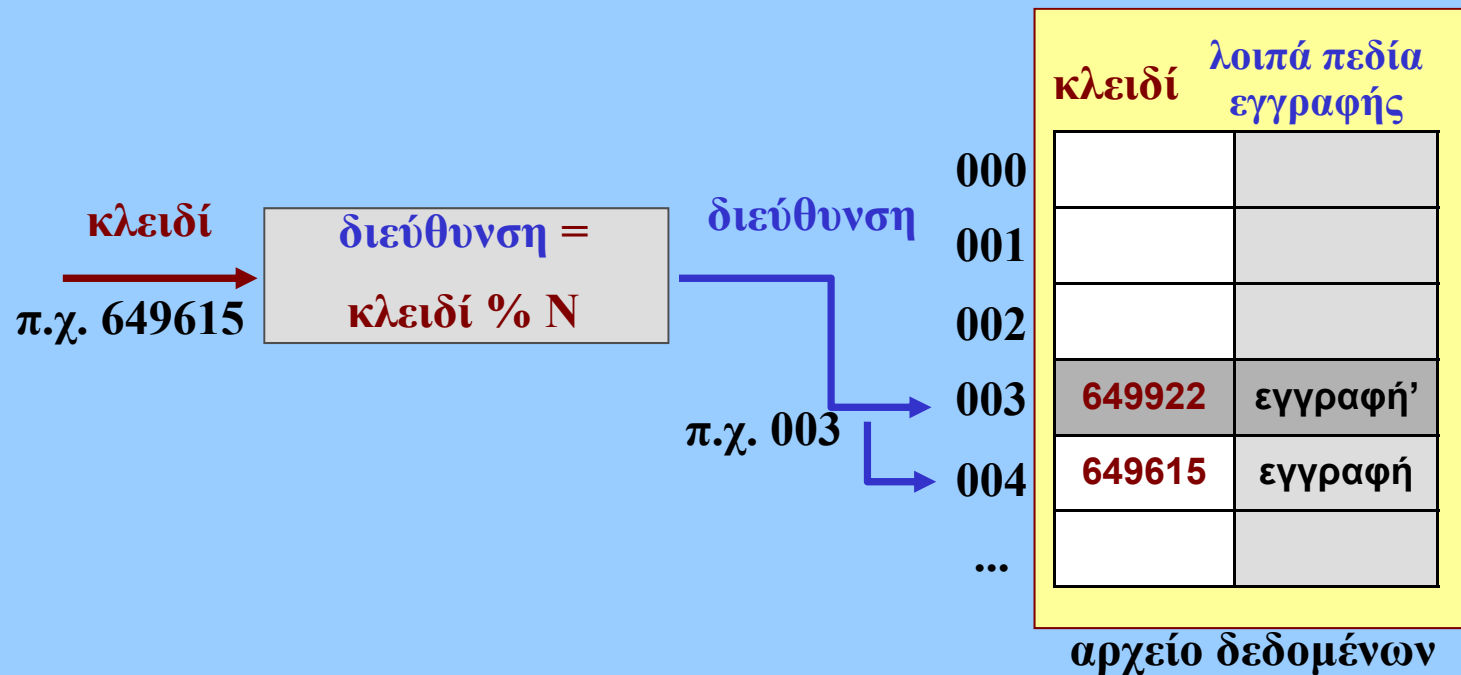
Άλλες συναρτήσεις κατακερματισμού

- **Εξαγωγή ψηφίων**
 - επιλογή ορισμένων ψηφίων κλειδιού
- **Μέσου τετραγώνου**
 - μεσαία ψηφία του τετραγώνου του κλειδιού
- **Αναδιπλώσεις και περιστροφές ψηφίων**
- **Ψευδοτυχαία μέθοδος**
 - όχι τυχαία!
- **Μετατροπή κλειδιών**
 - όταν τα κλειδιά δεν είναι αριθμητικά

Συγκρούσεις

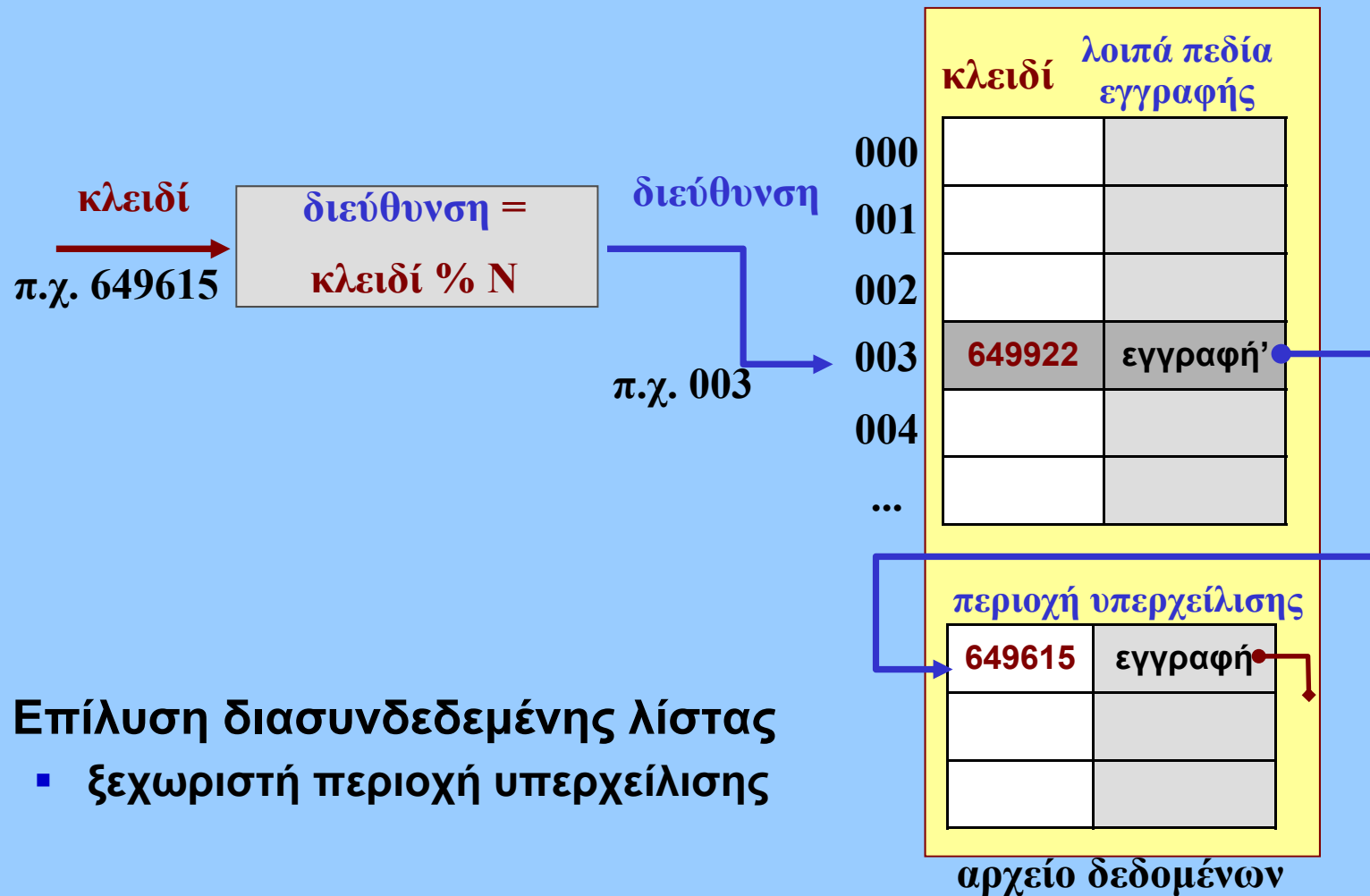
- Στην ιδανική περίπτωση:
 - η συνάρτηση κατακερματισμού θα έπρεπε να αντιστοιχίζει κάθε διαφορετικό κλειδί σε διαφορετική διεύθυνση
- **Αυτό όμως δεν είναι ο σκοπός του κατακερματισμού!**
 - όπου μεγάλος αριθμός πιθανών κλειδιών αντιστοιχίζονται σε περιορισμένο σύνολο θέσεων
- Σύγκρουση
 - όταν δύο διαφορετικά κλειδιά αντιστοιχίζονται στην ίδια διεύθυνση
 - μια καλή συνάρτηση κατακερματισμού ελαχιστοποιεί τις συγκρούσεις αλλά...
 - ...δεν μπορούν να αποφευχθούν τελείως!
- Επίλυση συγκρούσεων (collision resolution)
 - Εξεύρεση μεθόδων για την αποθήκευση των εγγραφών σε εναλλακτικές θέσεις

Επίλυση συγκρούσεων



- Επίλυση ανοικτής διευθυνσιοδότησης
 - χρήση γειτονικών διευθύνσεων
 - τάση συσσώρευσης εγγραφών σε ομάδες (clustering)

Επίλυση συγκρούσεων (2)



- Επίλυση διασυνδεδεμένης λίστας
 - ξεχωριστή περιοχή υπερχείλισης

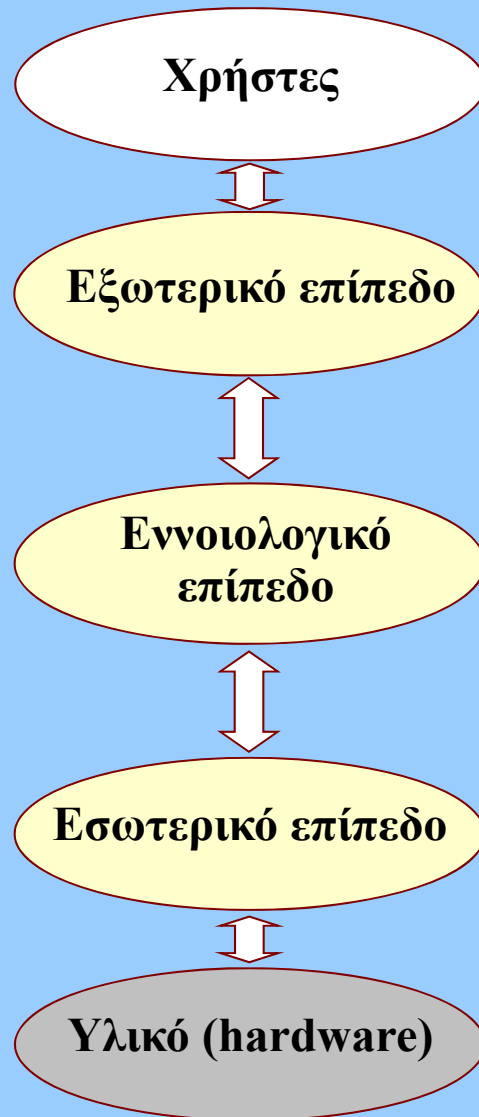
Πολυπλοκότητα οργάνωσης πληροφορίας

- **Στις μοντέρνες εφαρμογές**
 - ο όγκος της πληροφορίας αυξάνεται
 - ο τύπος της πληροφορίας έχει πολλαπλές μορφές
 - όχι μόνο “καρτέλες” εργαζομένων
 - π.χ. πολυμέσα
 - οι εφαρμογές είναι πολυπλοκότερες
 - οικονομικές
 - σχεδιαστικές
 - εμπορικές (βλ. e-)
 - η πληροφορία είναι κατανεμημένη
 - δικτυακές εφαρμογές
 - πώς επιτυγχάνεται η οργάνωση-διαχείριση πληροφορίας;
 - δεν αρκεί μια απλή συλλογή αρχείων

Βάσεις δεδομένων

- **Βάσεις Δεδομένων:** λογική συλλογή από συσχετιζόμενα δεδομένα
 - Η πληροφορία αποθηκεύεται σε φυσικό επίπεδο με διάφορους τρόπους
- **Χρήστες – Εφαρμογές**
 - Άντληση πληροφορίας από βάσεις δεδομένων
- **Συστήματα Διαχείρισης Βάσεων Δεδομένων**
 - Δημιουργούν και συντηρούν βάσεις δεδομένων
 - Ορίζουν τους κανόνες λειτουργίας των βάσεων

Αρχιτεκτονική βάσεων δεδομένων

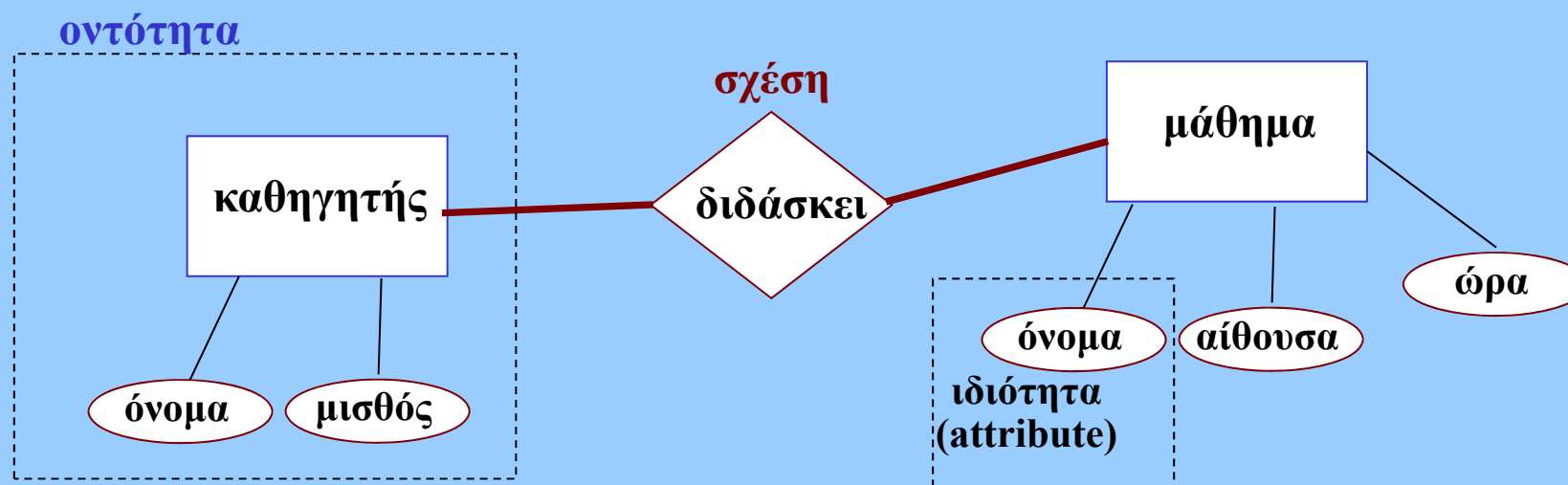


αλληλεπίδραση με τον
χρήστη

ορίζεται το μοντέλο
δεδομένων (που βλέπει ο
χρήστης) και παρέχονται
οι κύριες λειτουργίες ΒΔ

αλληλεπιδρά με το υλικό
και καθορίζει τον
φυσικό τρόπο
αποθήκευσης των
δεδομένων

Αναπαράσταση σχέσεων δεδομένων

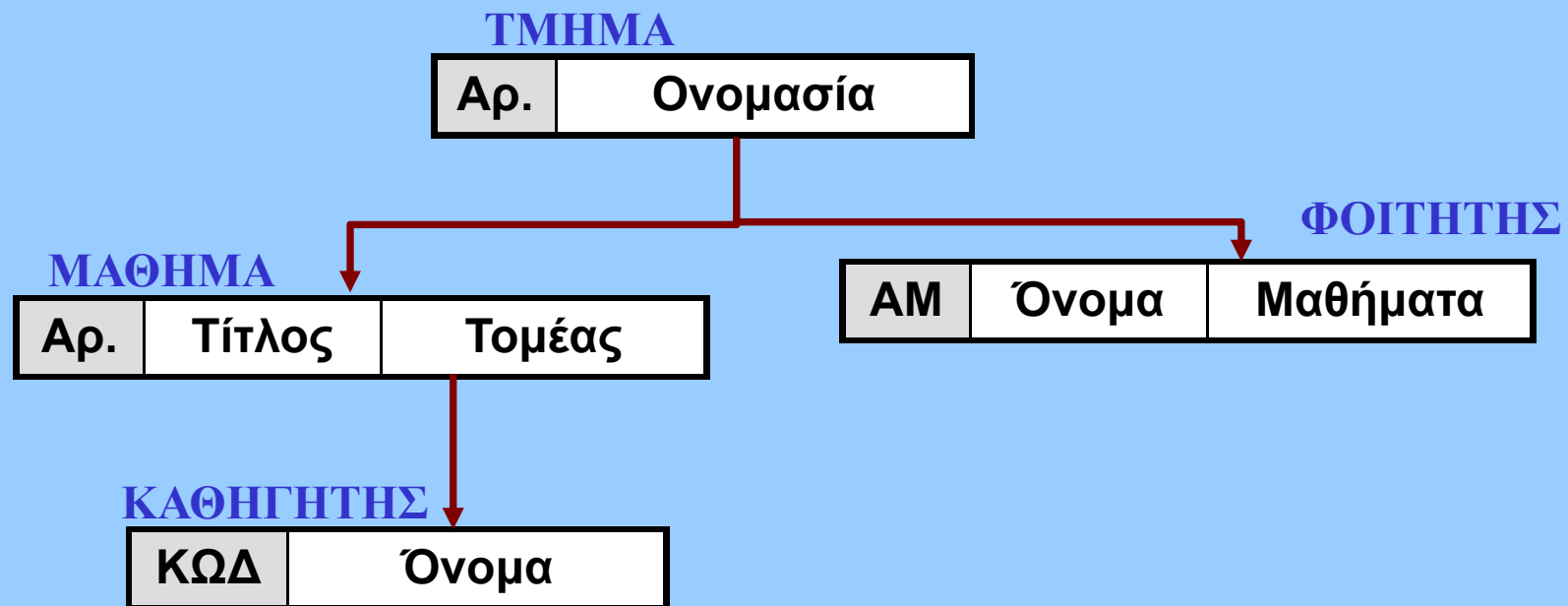


- **Μοντέλο Οντοτήτων-Σχέσεων (Entity-Relationship Model)**
 - περιγράφει τις σχέσεις μεταξύ λογικών οντοτήτων δεδομένων
 - μετατρέπεται σε κάποιο μοντέλο βάσης δεδομένων
 - όπως ο ψευδοκώδικας και ο πηγαίος κώδικας

Τύποι σχέσεων δεδομένων

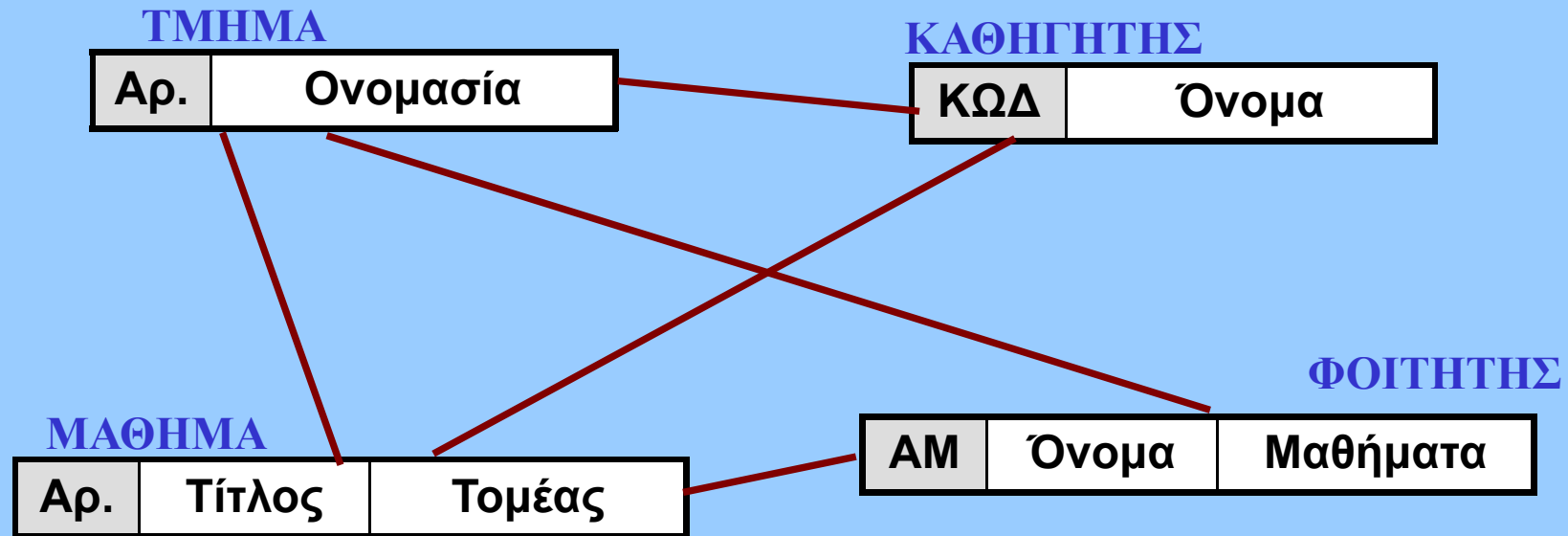
- **1:1**
 - χώρα – (έχει πρωτεύουσα) – πόλη
- **1:N**
 - πόλη – (γεννήθηκε στην) – άνθρωπος
- **N:M**
 - μάθημα επιλογής – (έχει επιλέξει) – φοιτητής

Ιεραρχικές βάσεις δεδομένων



- Μόνο 1:1 και 1:N σχέσεις
- τέλη δεκαετίας '60 (δεν χρησιμοποιείται πλέον)

Δικτυακό μοντέλο βάσης δεδομένων



- υποστηρίζει και M:N σχέσεις
- αρχές δεκαετίας '70 (δεν χρησιμοποιείται πλέον)

Σχεσιακό μοντέλο βάσης δεδομένων

ΤΜΗΜΑ

Αρ.	Όνομασία
...	...
...	...

ΚΑΘΗΓΗΤΗΣ

ΚΩΔ	Όνομα	Αρ.Τμήματος	Αρ.Μαθήματος
...
...

ΜΑΘΗΜΑ

Αρ.	Τίτλος	Κωδ.Καθηγητή	Τομέας
...
...

ΦΟΙΤΗΤΗΣ

ΑΜ	Όνομα	Αρ.Μαθήματος
...
...

- το πλέον χρησιμοποιούμενο μοντέλο σήμερα
- λογικοί πίνακες αντιπροσωπεύουν τις **οντότητες**
- Γραμμή (**tuple**): ορίζει μια συλλογή τιμών **ιδιοτήτων**
- **και οι σχέσεις ορίζονται δυναμικά από τις ιδιότητες κάθε γραμμής!**

Σχεσιακές βάσεις δεδομένων

- Λογική οργάνωση σε **πίνακες** (σχέσεις)
- Κάθε **γραμμή** του πίνακα είναι μία **εγγραφή** ή **πλειάδα (tuple)** και κάθε **στήλη** ένα **πεδίο** (ιδιότητα)
- κάθε γραμμή (πρέπει να) είναι μοναδική (χρήση **κλειδιών**)
- οι σχέσεις ορίζονται δυναμικά από τις τιμές των ιδιοτήτων
 - η πληροφορία των σχέσεων δεν φυλάσσεται ξεχωριστά
- υποστηρίζονται όλα τα είδη των σχέσεων
 - 1:1, 1:N, N:M
- **Structured Query Language**
 - εκτέλεση λειτουργιών στη βάση δεδομένων

Άλλα μοντέλα βάσεων δεδομένων

- **Αντικειμενοστραφείς βάσεις δεδομένων**
 - Ορισμός αντικειμένων και σχέσεων
 - Ως επέκταση μιας γλώσσας προγραμματισμού
 - Δεδομένα ως “μεταβλητές” με τιμές που διατηρούνται και μετά την εκτέλεση του προγράμματος
- **Κατανεμημένες βάσεις δεδομένων**
 - Κατανομή των δεδομένων μιας βάσης σε πολλαπλές τοποθεσίες, συστήματα ή δίσκους
 - Διασύνδεση μέσω διαδικτύου
 - Δυνατότητα πολλαπλών αντιγράφων

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών
2006-07

Συμπύεση & Ασφάλεια Δεδομένων

Μ.Στεφανιδάκης

Στόχοι του μαθήματος

- **Συμπύεση Δεδομένων**
 - Απωλεστικές- Μη απωλεστικές μέθοδοι
 - Κωδικοποιήσεις Huffman, LZ
 - Κωδικοποίηση JPEG, MPEG
- **Αρχές ασφάλειας δεδομένων**
 - Απόρρητο, μέθοδοι μυστικού και δημόσιου κλειδιού
 - Ψηφιακή υπογραφή
 - Συνόψεις

Συμπίεση δεδομένων

- **Ανάγκη συμπίεσης**
 - Μείωση χώρου αποθήκευσης
 - Μείωση όγκου μεταδιδόμενης πληροφορίας
- **Μέθοδοι συμπίεσης**
 - **Μη απωλεστικές (lossless)**
 - Αποσυμπίεση δεδομένων στην ακριβή αρχική μορφή
 - Για προγράμματα, αρχεία κειμένου...
 - **Απωλεστικές (lossy)**
 - Απώλεια πληροφορίας κατά τη συμπίεση
 - Ανεκτή για ήχο, εικόνα και βίντεο

Κωδικοποίηση τρέχοντος μήκους (run-length encoding)

BBBBBBBBBAAAAAAAAAAAAAAAAANMMMMMMMMMM

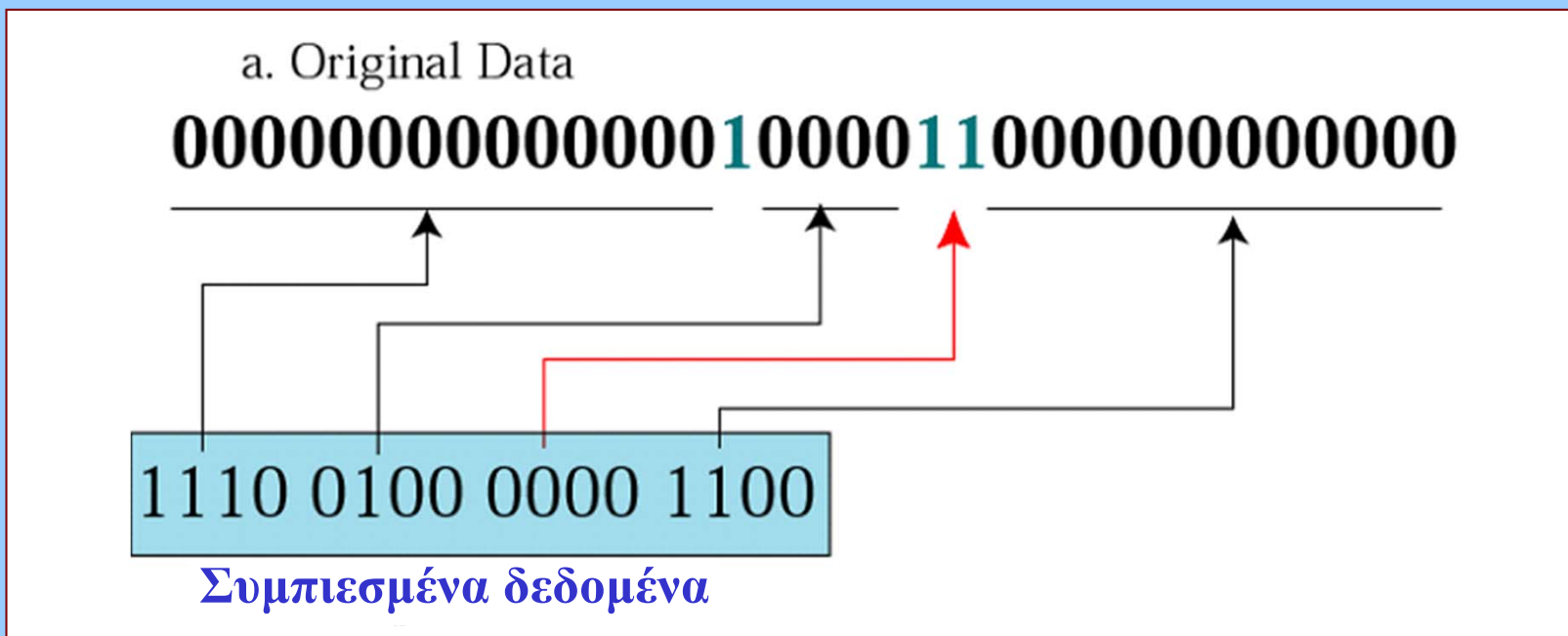
Αρχικά δεδομένα

B09A16N01M10

Συμπιεσμένα δεδομένα

- Απλή μέθοδος συμπίεσης
 - Αντικατάσταση συνεχόμενων ίδιων συμβόλων από **<σύμβολο:αριθμό εμφανίσεων>**
 - Δεν απαιτείται η γνώση της συχνότητας εμφάνισης

Κωδικοποίηση τρέχοντος μήκους (run-length encoding)



- Παραλλαγή για 2 σύμβολα
 - π.χ. 0 και 1

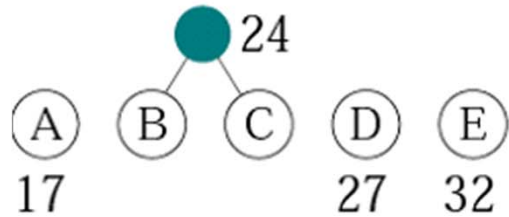
Κωδικοποίηση Huffman

- “Ανάθεση συντομότερων κωδικών στα σύμβολα που εμφανίζονται συχνότερα”
- Διαφορετικοί κώδικες για κάθε διαφορετικό αντικείμενο προς συμπίεση
- Συμπίεση
 - Εύρεση συχνότητας εμφάνισης
 - Κατασκευή δένδρου ανάθεσης κωδικών
 - Αντικατάσταση συμβόλων
- Αποσυμπίεση
 - Λήψη δένδρου ανάθεσης κωδικών
 - Αντικατάσταση κωδικών από σύμβολα

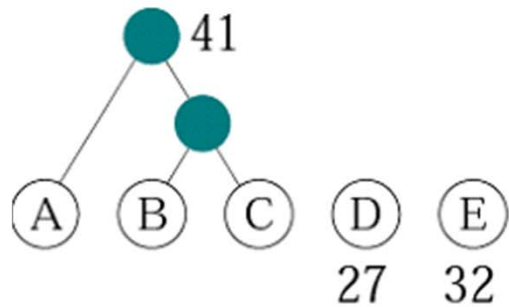
Παράδειγμα κώδικα Huffman

(A) (B) (C) (D) (E)
17 12 12 27 32

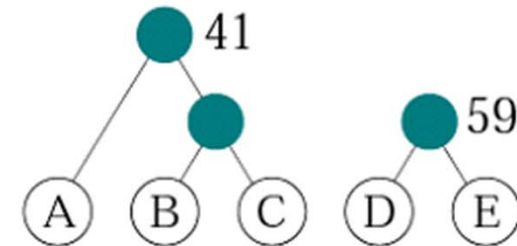
a.



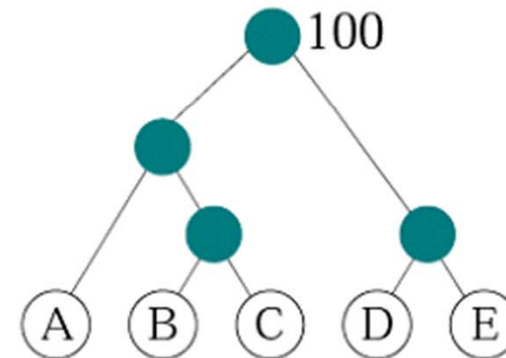
b.



c.

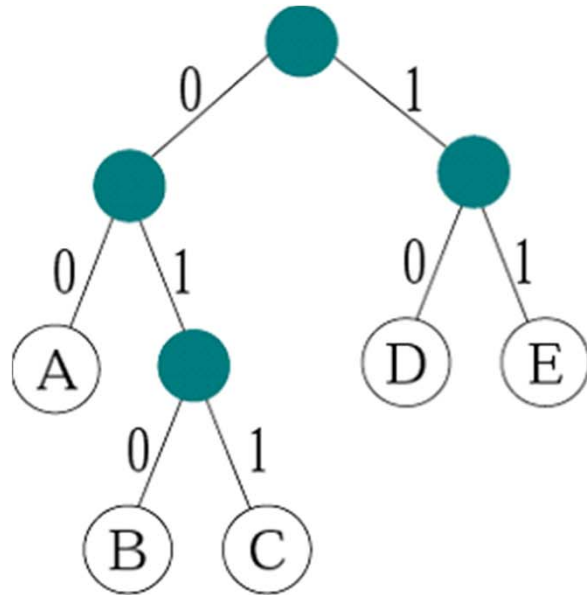


d.



e.

Παράδειγμα κώδικα Huffman (2)

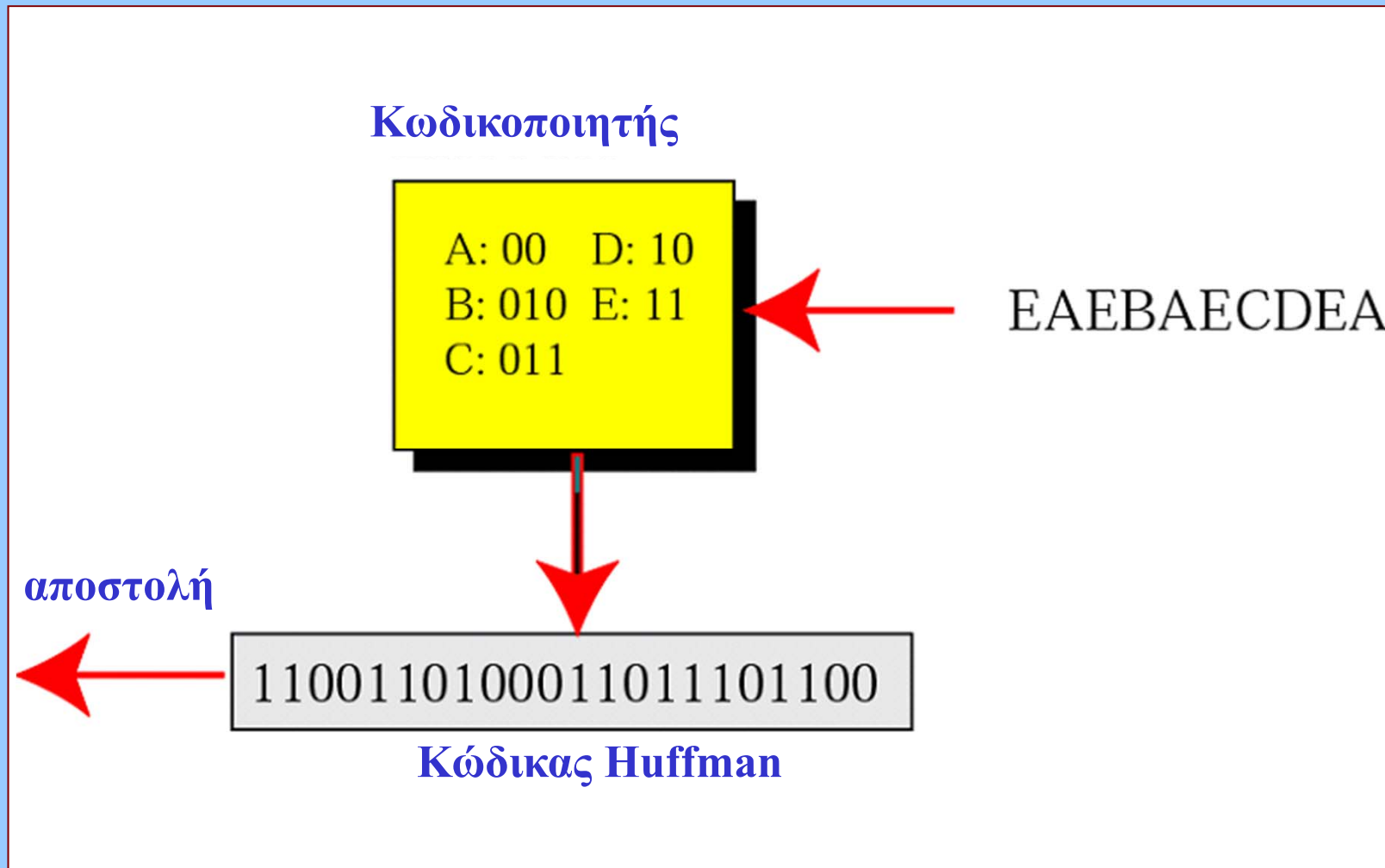


A: 00 D: 10
B: 010 E: 11
C: 011

κώδικες

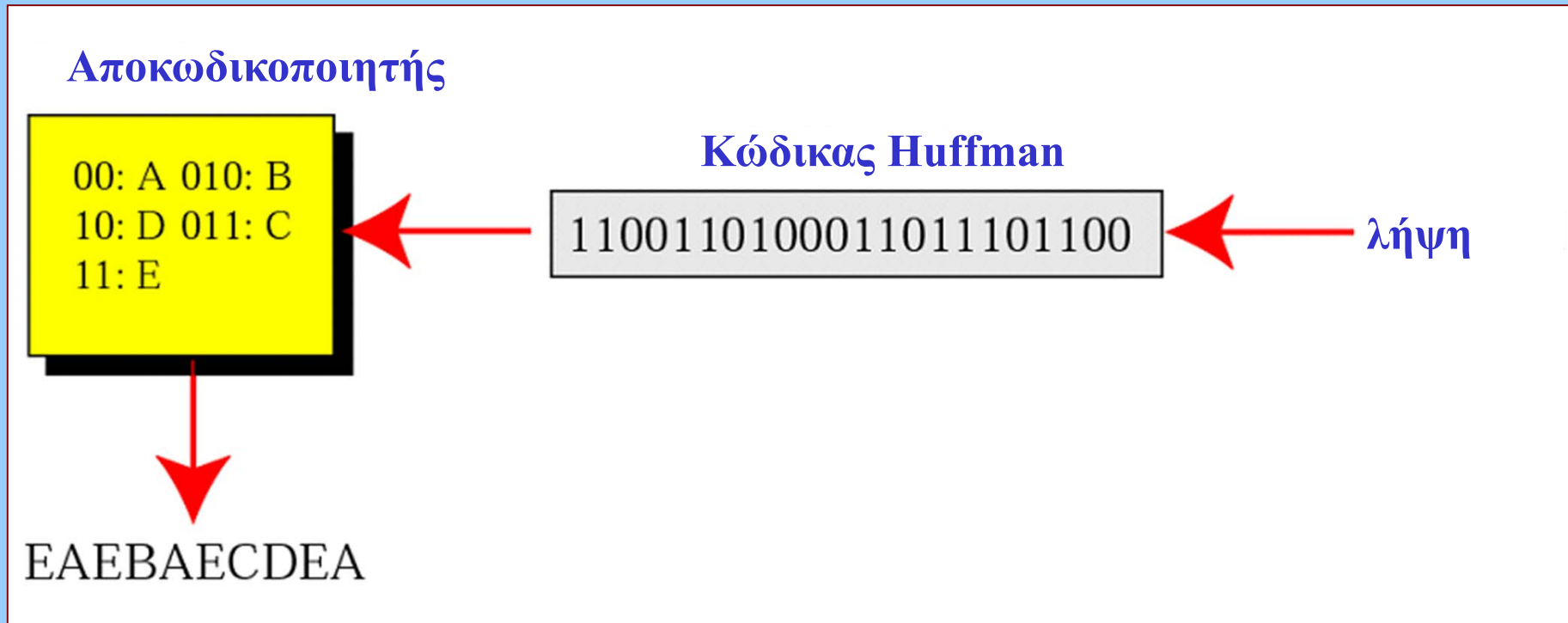
- Πιθανή ύπαρξη περισσότερων κωδίκων
 - Όταν υπάρχουν ίδιες συχνότητες εμφάνισης
 - Επιλογή βελτιστοποίησης: π.χ. με μικρότερο μέσο μήκος

Παράδειγμα κώδικα Huffman



- **Κανένας κωδικός δεν είναι πρόθεμα άλλου!**

Παράδειγμα κώδικα Huffman

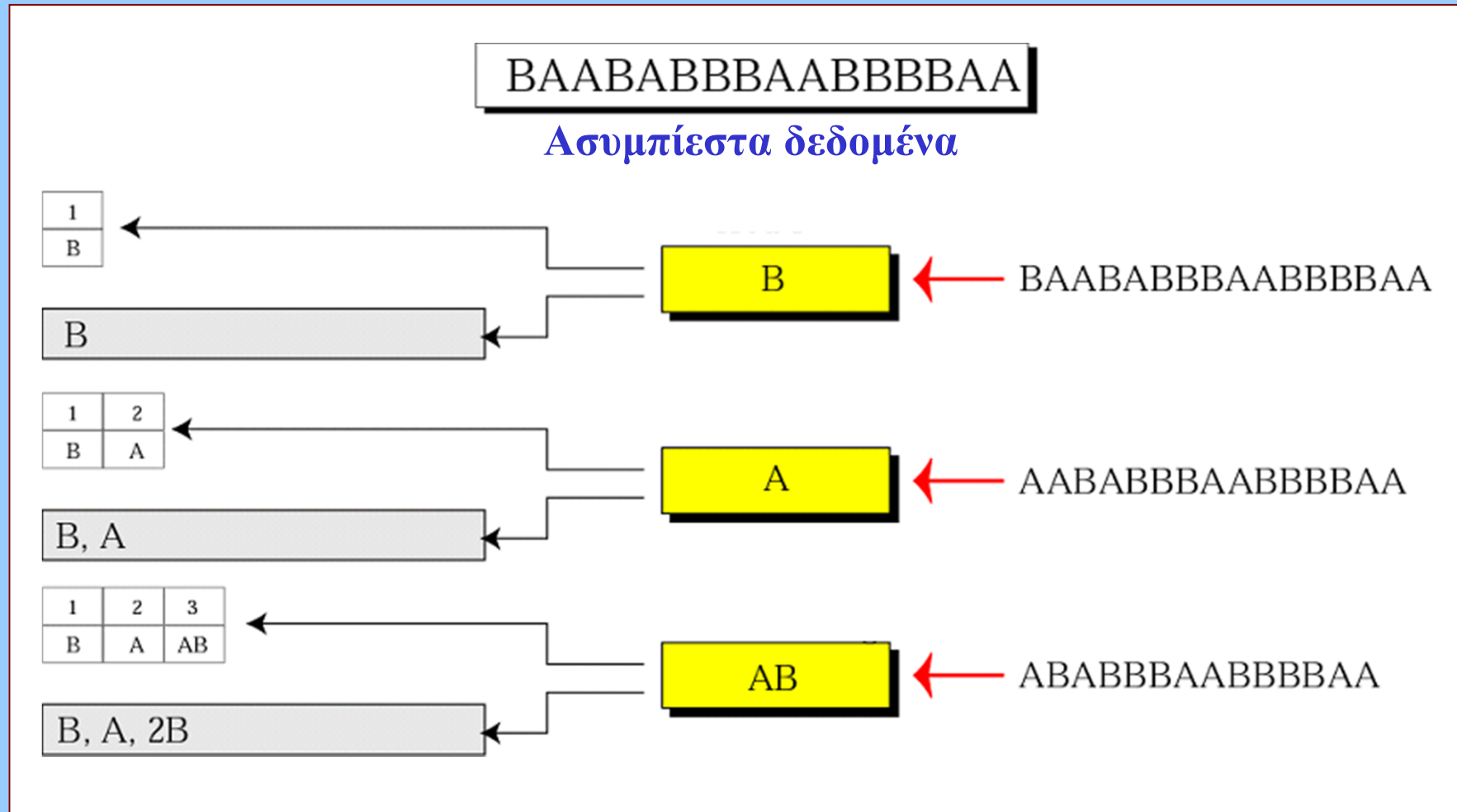


- Ο αποκωδικοποιητής πρέπει να λάβει και την αντιστοιχία συμβόλων-κωδίκων
 - Αποστολή μαζί με τα κωδικοποιημένα δεδομένα
- Προσαρμοστική κωδικοποίηση Huffman
 - Δεν απαιτείται εξαρχής γνώση όλων των συχνοτήτων εμφάνισης

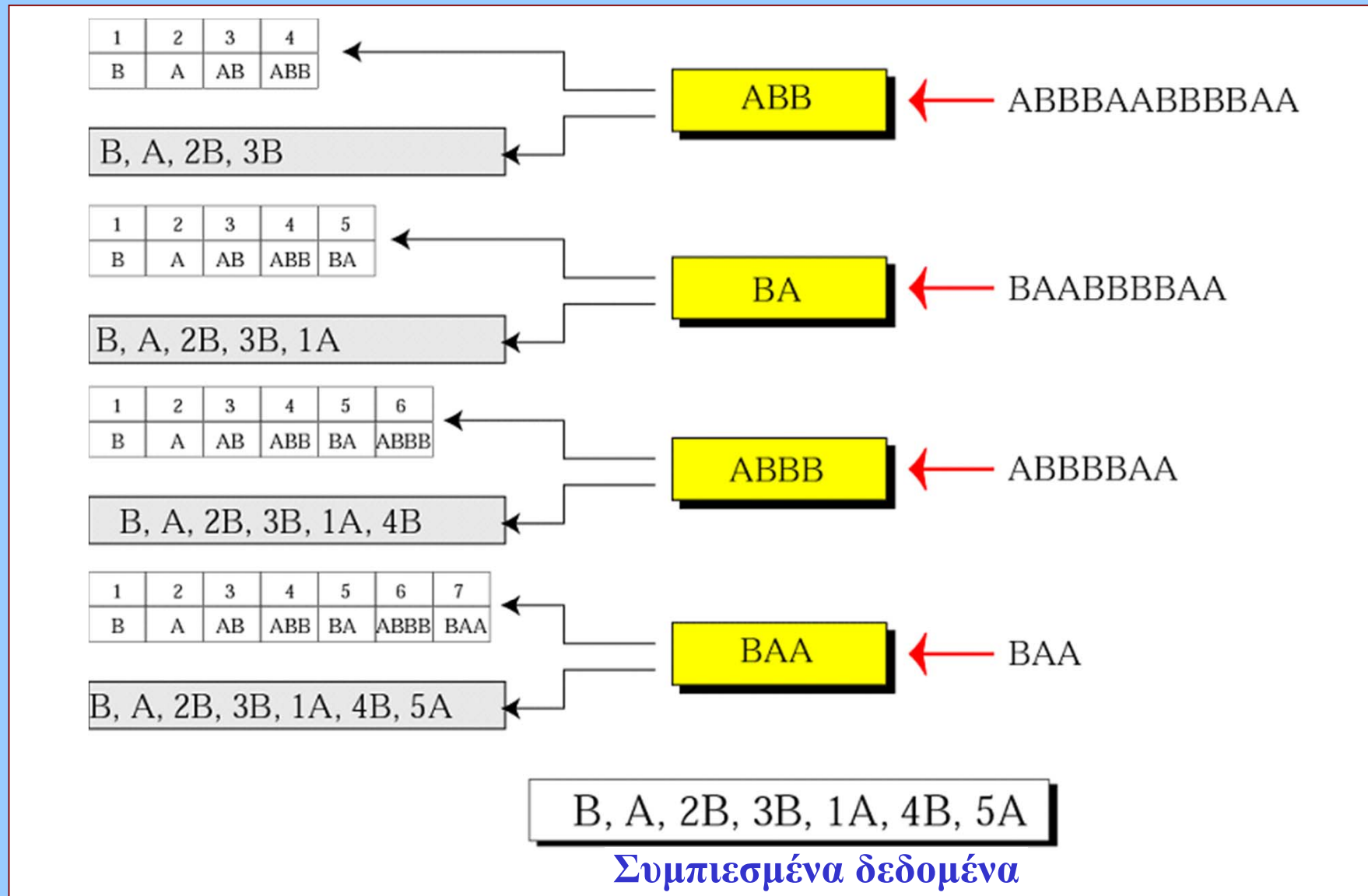
Συμπίεση με μεθόδους λεξικού

- **Συμπίεση**
 - Κατασκευή λεξικού αντιστοίχισης συμβόλων
- **Αποσυμπίεση**
 - Όμοια κατασκευή λεξικού
 - Δεν απαιτείται αποστολή του λεξικού
- **Αλγόριθμος Lempel-Ziv (LZ)**
 - Πολλές παραλλαγές
 - Χρησιμοποιούνται και στα αρχεία gif

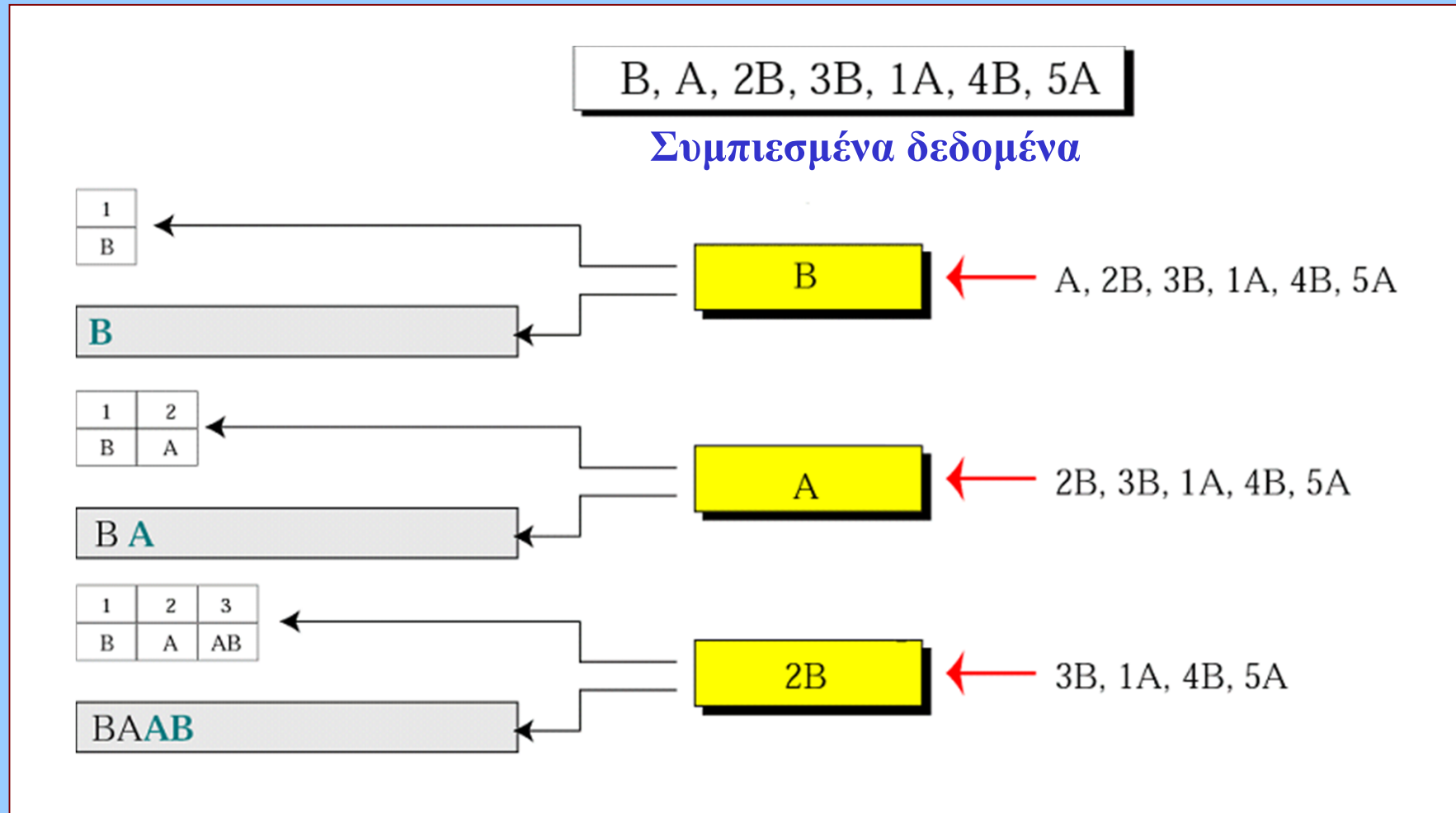
Παράδειγμα κωδικοποίησης LZ (1)



Παράδειγμα κωδικοποίησης LZ (2)



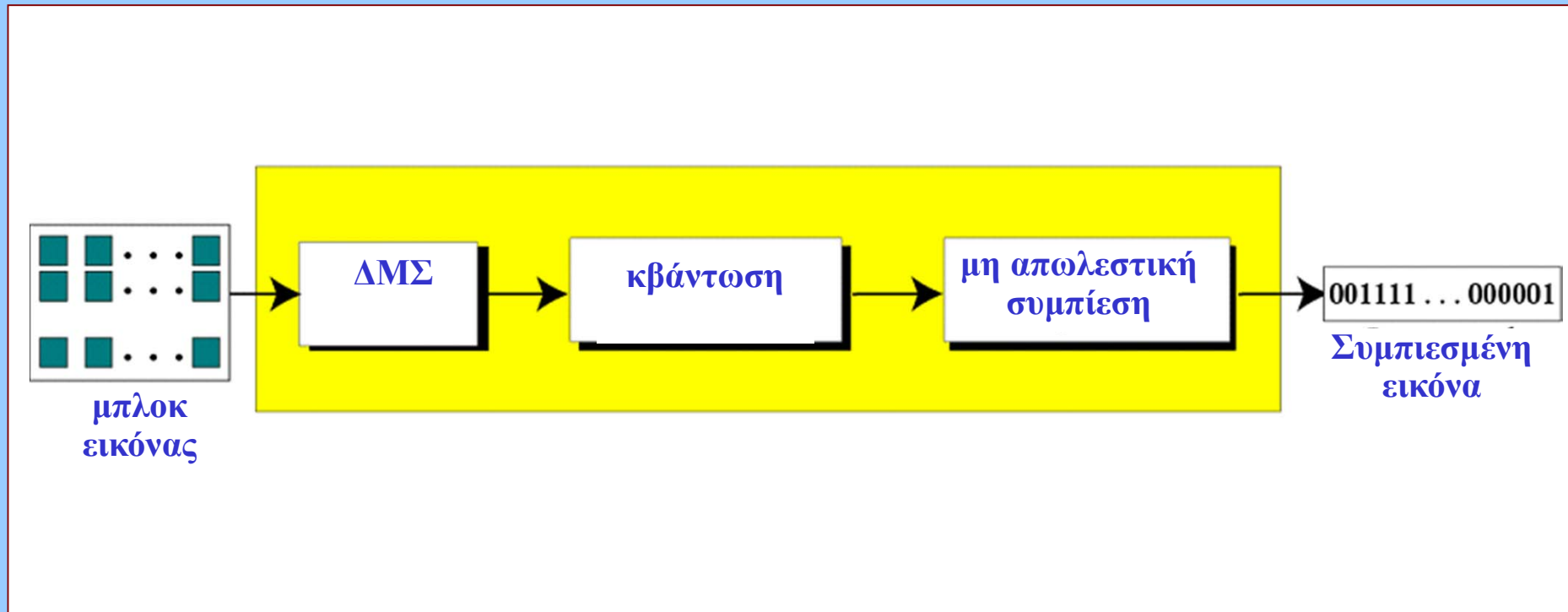
Παράδειγμα αποκωδικοποίησης LZ (1)



Απωλεστικές (lossy) μέθοδοι συμπίεσης

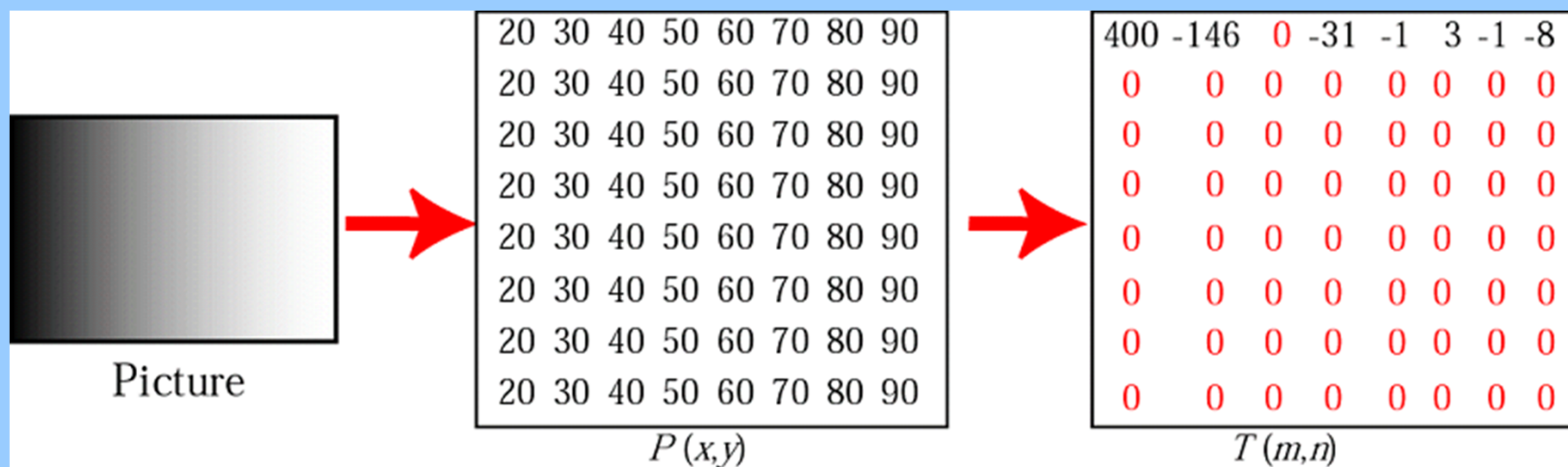
- Απώλεια μέρους δεδομένων κατά τη διαδικασία συμπίεσης-αποσυμπίεσης
- Χρήση: ψηφιακός ήχος – εικόνα –βίντεο
 - Το ανθρώπινο μάτι/αυτί δεν αντιλαμβάνεται ορισμένου τύπου αποκλίσεις από τα πρωτότυπα δεδομένα
- Παραδείγματα
 - JPEG (εικόνες)
 - MPEG (βίντεο)

Κωδικοποίηση JPEG



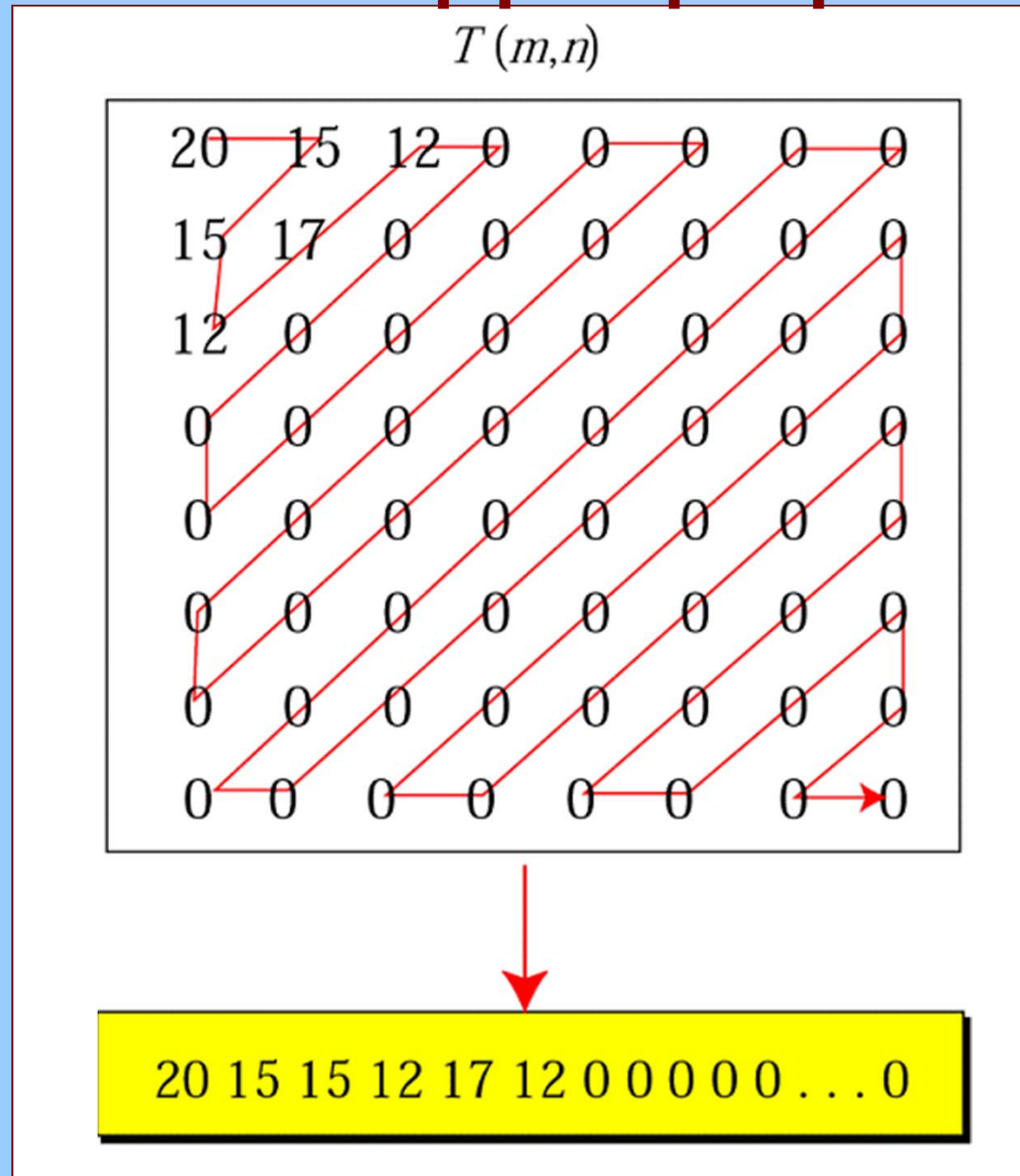
- **ΔΜΣ:** διακριτός μετασχηματισμός συνημιτόνου (discrete cosine transform)
 - Συσχέτιση-συμπύκνωση της “ενέργειας” πληροφορίας του μπλοκ
- Κβάντωση (απωλεστική)
- Συμπίεση πλεονάζουσας πληροφορίας (μη απωλεστική)

Διακριτός μετασχηματισμός συνημιτόνου

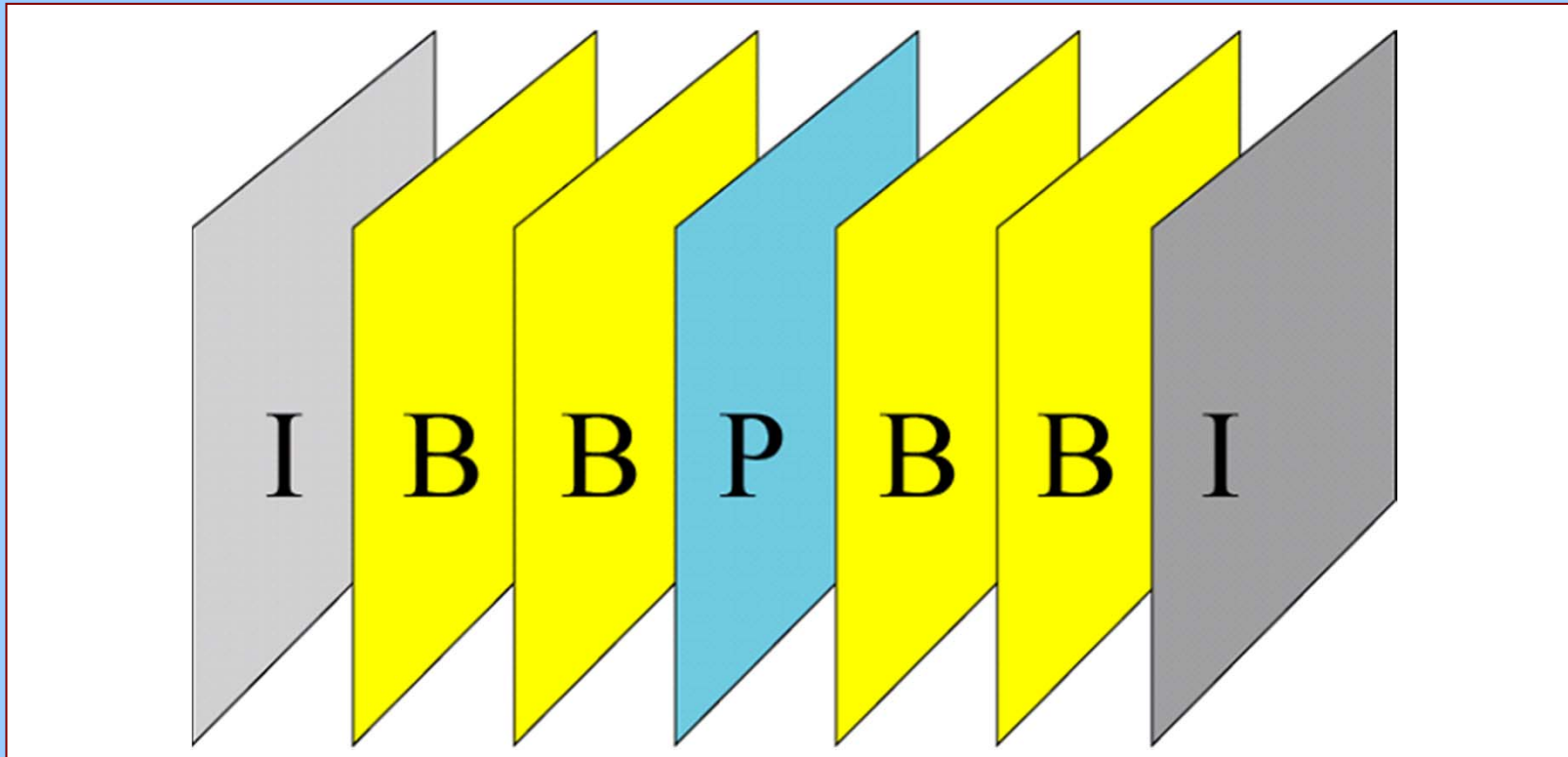


- Ρητό αποτέλεσμα (ακολουθεί κβάντωση)
- Η σημαντική πληροφορία συγκεντρώνεται στην πάνω αριστερή γωνία του πίνακα
 - Όσο πιο ομοιόμορφη η εικόνα τόσο πιο μεγάλη συγκέντρωση

JPEG: Τελική φάση συμπίεσης



Κωδικοποίηση MPEG



- Χωρική συμπίεση (σε κάθε καρέ, παρόμοια με JPEG)
- Χρονική συμπίεση (διαφορές μεταξύ γειτονικών καρέ)

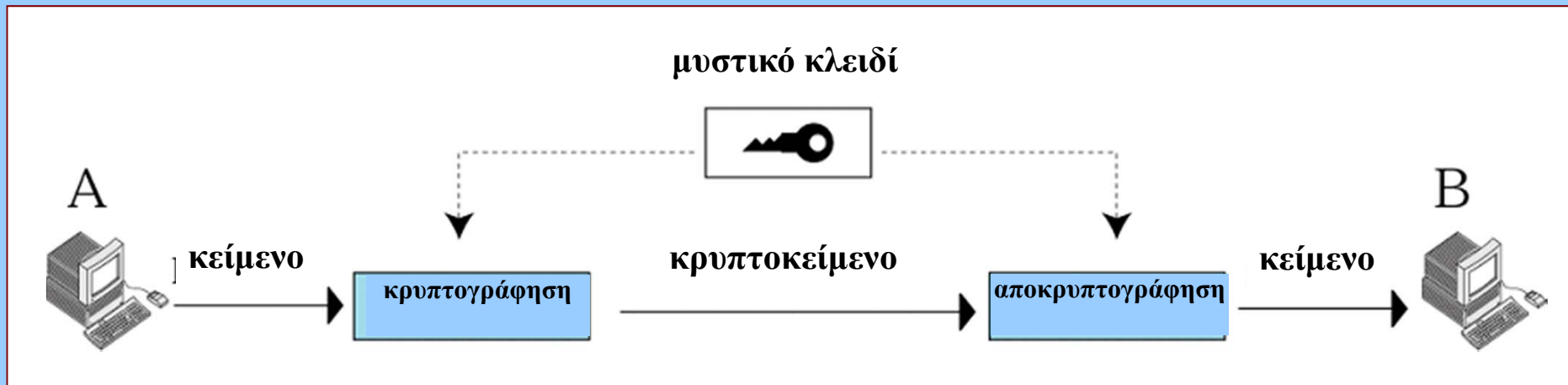
Καρέ (frames) MPEG

- **I-frames (intracoded)**
 - Ανεξάρτητα καρέ, ως σημεία αναφοράς στον χρόνο
- **P-frames (predicted)**
 - Αλλαγές από προηγούμενα I-frames ή P-frames
- **B-frames (bidirectional)**
 - Συσχέτιση προηγούμενων και επόμενων I-frame ή P-frames

Αρχές ασφάλειας δεδομένων

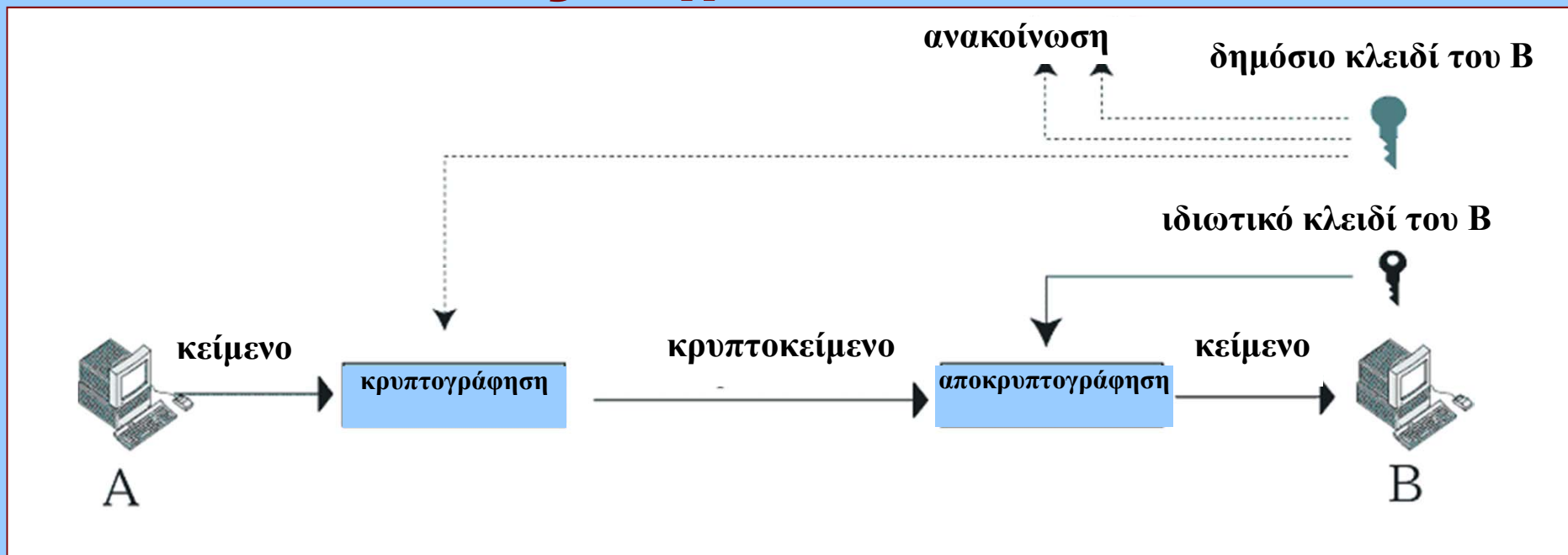
- **Ιδιωτικό Απόρρητο**
 - κρυπτογράφηση
- **Πιστοποίηση αυθεντικότητας**
- **Ακεραιότητα**
- **Μη απάρνηση**
 - ψηφιακή υπογραφή

Κρυπτογράφηση



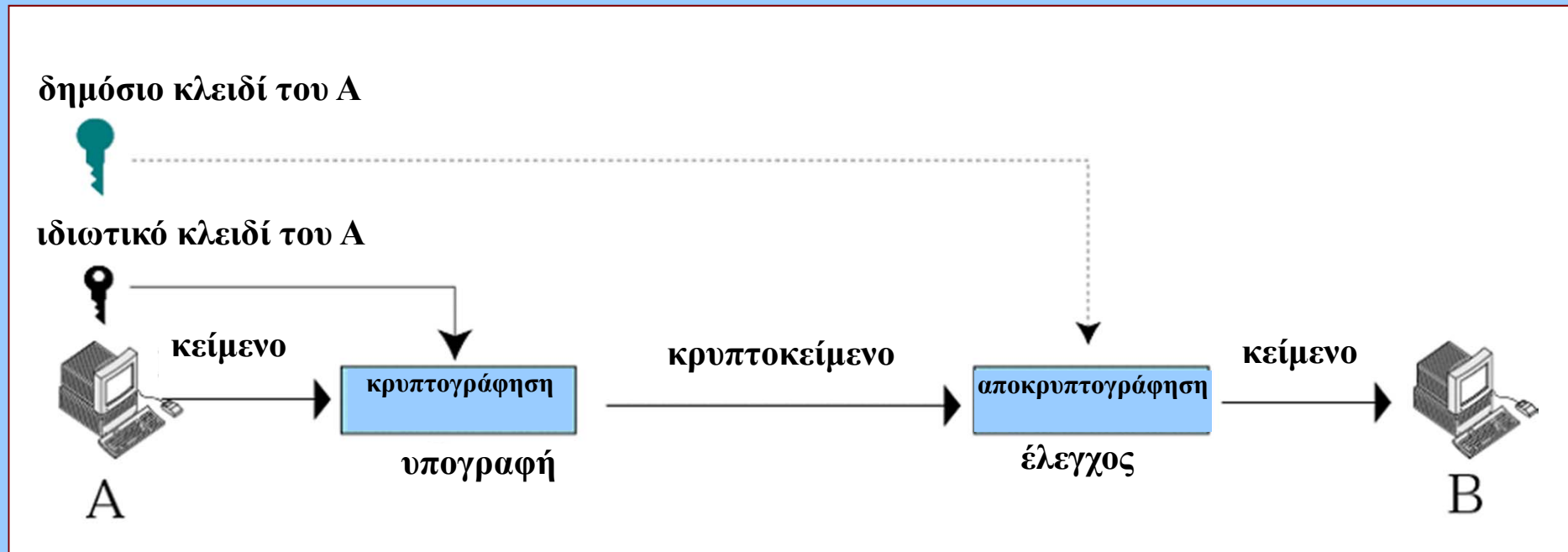
- **Μυστικού κλειδιού**
 - Αλγόριθμοι κρυπτογράφησης-αποκρυπτογράφησης
 - Συμμετρικοί
 - π.χ. DES (κλειδί 56 bits)
 - Αμφίδρομη επικοινωνία με το ίδιο κλειδί
 - Γρήγοροι αλγόριθμοι αλλά απαιτούνται πολλά κλειδιά

Μέθοδος δημόσιου κλειδιού



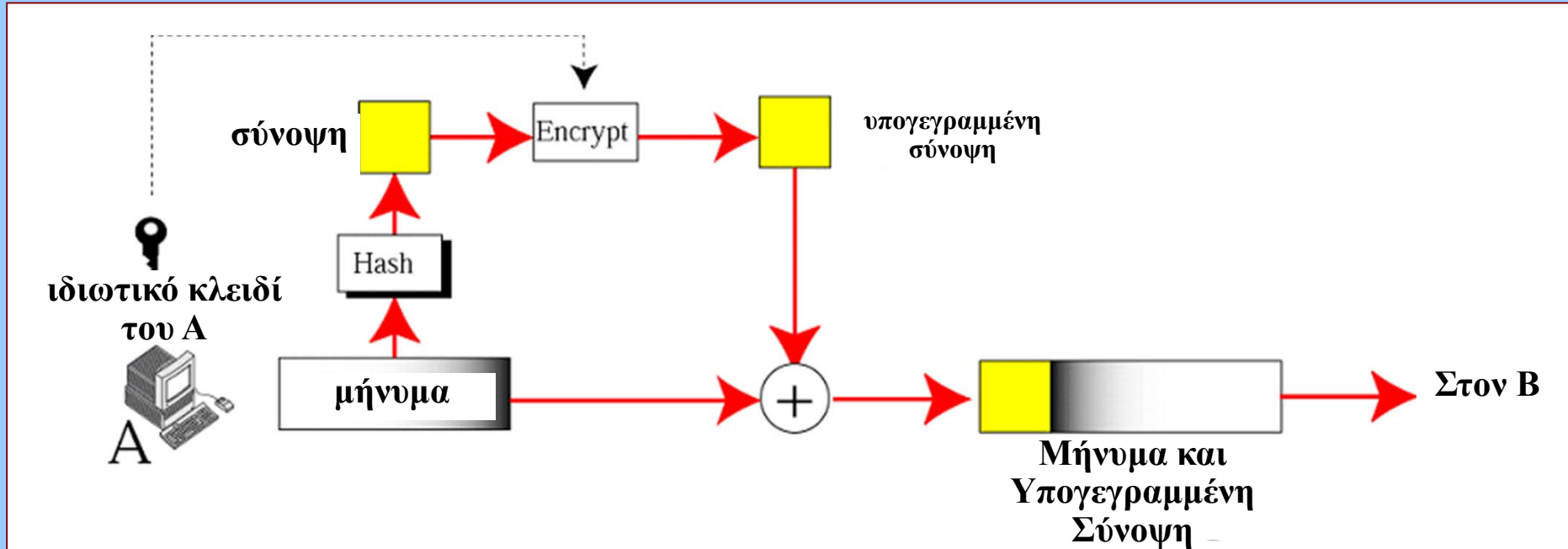
- Ζεύγος δημόσιου και ιδιωτικού κλειδιού
 - π.χ. RSA: ιδιωτικό κλειδί=(N,d) δημόσιο κλειδί=(N,e)
 - $C = P^e \text{ mod } N$ ($P = \text{κείμενο}$, $C = \text{κρυπτοκείμενο}$)
 - $P = C^d \text{ mod } N$
 - Δημιουργία κλειδιών
 - Αργοί αλγόριθμοι, πολύ μεγάλοι αριθμοί d,e
- Επίσης: συνδυασμοί μεθόδων μυστικού και δημόσιου κλειδιού

Ψηφιακή υπογραφή



- Εξασφάλιση αυθεντικότητας, ακεραιότητας και μη απάρνησης
 - Σε όλο το μεταδιδόμενο κείμενο (παραλλαγή της μεθόδου δημόσιου κλειδιού)
 - Δεν εξασφαλίζει απόρρητο!!
 - Δημιουργία σύνοψης
 - Συναρτήσεις κατακερματισμού: MD5 (128 bits), SHA-1 (160 bits)

Δημιουργία και αποστολή σύνοψης



Λήψη και έλεγχος σύνοψης

