

# ΚΕΦΑΛΑΙΟ 1

## ΕΙΣΑΓΩΓΗ - ΕΠΙΣΚΟΠΗΣΗ

### Σύνοψη

Στο κεφάλαιο 1, γίνεται μια επισκόπηση των Κατανεμημένων Συστημάτων (Κ.Σ.), το τι είναι, το πώς διαστασιολογούνται, ποια προβλήματα και ανάγκες θεραπεύουν, και ποια η σχέση τους με τα Πληροφοριακά Συστήματα (Π.Σ.) και τις λοιπές κατανεμημένες σύγχρονες εφαρμογές. Επίσης, εξετάζονται οι πρόγονοι των Κ.Σ. και η εξέλιξή τους, και γίνεται αναφορά στη σημαντικότητα της αποτελεσματικής διαχείρισης των Κ.Σ.. Επιπρόσθετα, δίνεται μια συνοπτική παρουσίαση των θεμάτων του βιβλίου προκειμένου ο αναγνώστης να έχει μια συνολική εικόνα του τι πραγματεύεται αυτό, καθώς και τη μεθοδολογική του προσέγγιση.

### Προαπαιτούμενη γνώση

- 1) Δουλγέρης, Χ., Μητρόπουλος, Σ., 2015. Πληροφοριακά συστήματα στο διαδίκτυο. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: <http://hdl.handle.net/11419/3969>
- 2) Andrew S. Tanenbaum, Herbert Bos (2018), Σύγχρονα Λειτουργικά Συστήματα, Έκδοση: 4η Αμερικανική, Εκδόσεις Κλειδάριθμος ΕΠΕ
- 3) Παπακωνσταντίνου Γ., Τσανάκας Π., Θεοχάρης Θ., Συστήματα Παράλληλης Επεξεργασίας, Εκδόσεις Σ. Αθανασόπουλος & ΣΙΑ Ο.Ε.
- 4) Κάβουρας Ι.Κ. (2009), Λειτουργικά Συστήματα (2ος τόμος), Έκδοση: 7η, Εκδόσεις Κλειδάριθμος ΕΠΕ.

### 1.1. Τι είναι τα Κατανεμημένα Συστήματα

Τα Κατανεμημένα Συστήματα (ΚΣ) είναι ένα αναπόσπαστο κομμάτι των σύγχρονων τεχνολογιών. Τα σύγχρονα συστήματα σχεδόν όλα είναι στη φύση τους κατανεμημένα και εξ αυτού καταλαβαίνουμε τη σπουδαιότητα των συγκεκριμένων συστημάτων. Παρά ταύτα, ενώ υπάρχει τόσο μεγάλη διάδοση στην υλοποίησή τους, αποτελούν σχετικά ένα δυσνόητο κομμάτι στην περιοχή της πληροφορικής. Ένα ΚΣ είναι ένα λογισμικό το οποίο διασφαλίζει ότι μια συλλογή από ανεξάρτητους υπολογιστές δίνεται στους χρήστες του με την εικόνα ενός και μόνο συστήματος, το οποίο θα πρέπει να λειτουργεί, όπως άλλωστε και όλοι οι υπολογιστές, με αξιοπιστία και διαθεσιμότητα. Σημειώνεται ότι στον όρο ΚΣ συμπεριλαμβάνονται όλες οι διαστάσεις ενός πληροφοριακού συστήματος, ήτοι, η κατανεμημένη πληροφορία, τα δεδομένα, το κατανεμημένο υλικό, τα υποκείμενα λειτουργικά συστήματα και υπηρεσίες και οι εφαρμογές του χρήστη.

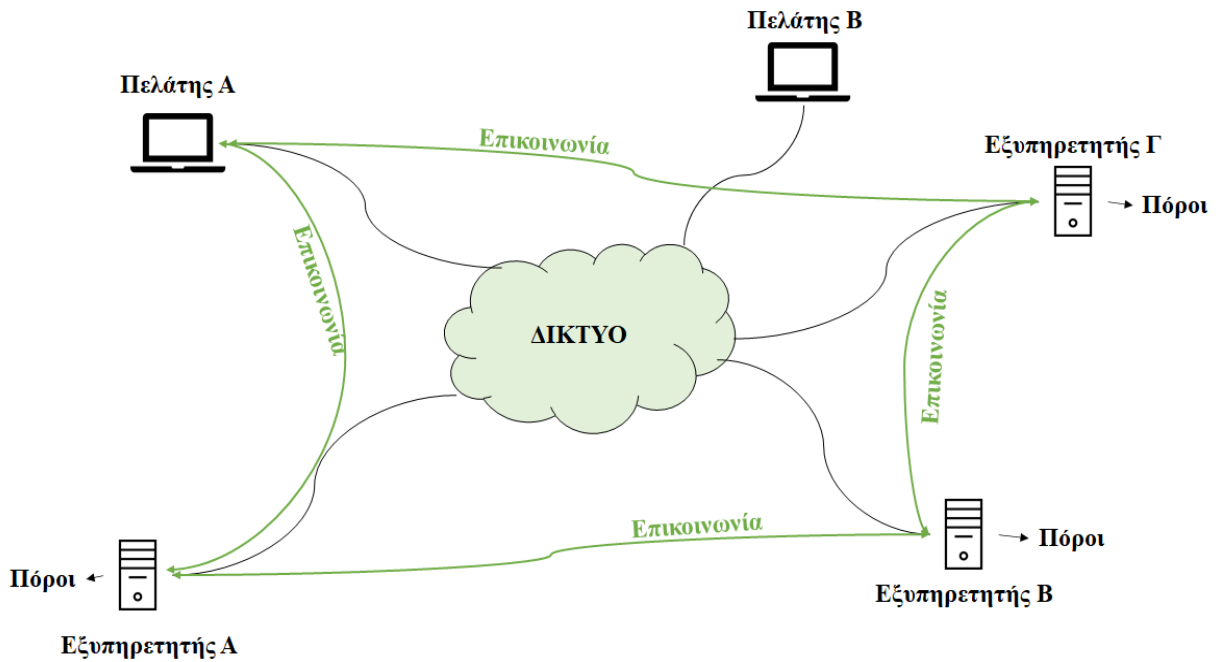
Από τα παραπάνω βλέπουμε ότι στα ΚΣ συνυπάρχουν δύο βασικές έννοιες: η διασύνδεση ανεξάρτητων υπολογιστικών μηχανών, οι οποίες συνεργάζονται προκειμένου να πετύχουν έναν κοινό στόχο και η έννοια της εικόνας του ενιαίου συστήματος η οποία επιτυγχάνεται μέσω ενός λογισμικού το οποίο ονομάζεται ενδιάμεσο λογισμικό (middleware). Η εισαγωγή του ενδιάμεσου αυτού λογισμικού δίνει τη δυνατότητα στους χρήστες να «βλέπουν» ένα ενιαίο σύστημα το οποίο είναι συνεχώς διαθέσιμο, ακόμα και αν ένας εκ των υποσυστημάτων/υπολογιστών είναι εκτός λειτουργίας. Επίσης, αυτή η ενιαία εικόνα δεν θα πρέπει να χαλάει εάν πρόκειται να επεκταθεί με περισσότερους υπολογιστές ή εάν αντικατασταθούν κάποιοι από τους παλαιότερους π.χ. για λόγους αναβάθμισης. Ο χρήστης του ΚΣ θα πρέπει να λαμβάνει τις υπηρεσίες του και να χρησιμοποιεί τους διαμοιραζόμενους πόρους του με ευκολία, φιλικότητα, ταχύτητα, ακεραιότητα και ασφάλεια.

Βασικός στόχος ενός ΚΣ είναι η αξιοποίηση των υπαρχόντων υπολογιστικών πόρων, μέσω της μεταξύ τους διασύνδεσης, καθώς και των χρηστών τους για την επίτευξη ενός κοινού στόχου με ταχύτητα, αξιοπιστία και ανοχή σε σφάλματα. Παρότι υπάρχει αυτή η διάσπαρτη κατανομή των υπολογιστικών πόρων σε μια μη κατ' ανάγκη εστιασμένη γεωγραφική περιοχή θα πρέπει αυτό να μην γίνεται αντιληπτό στους χρήστες διότι όπως είπαμε το ΚΣ θα πρέπει να δίνει την εικόνα ενός συστήματος - υπολογιστή. Για να επιτευχθεί αυτό, απαιτείται η ανάπτυξη κατάλληλων μηχανισμών οι οποίοι αποκρύπτουν την κατανομημένη φύση του συστήματος. Αυτοί οι μηχανισμοί παρέχουν τις λεγόμενες διαφάνειες κατανομής (distribution transparencies), που είναι το κύριο χαρακτηριστικό των κατανομημένων συστημάτων. Πέρα από το γεγονός ότι αποκρύπτεται ότι οι πόροι, λογισμικού ή υλικού, είναι κατανομημένοι σε πολλούς υπολογιστές, η έννοια της διαφάνειας επεκτείνεται και σε άλλες περιοχές της αρχιτεκτονικής των ΚΣ, όπως είναι αυτές του τρόπου προσπέλασης ενός πόρου, της μετακίνησης ενός πόρου, του παραλληλισμού στην εκτέλεση διεργασιών, της αστοχίας κάποιου πόρου, καθώς και άλλων που θα εξεταστούν λεπτομερώς στο επόμενο κεφάλαιο.

Από τη φύση τους τα ΚΣ θα πρέπει να μπορούν να αντιμετωπίζουν την ετερογένεια των συστημάτων προκειμένου να μπορούν να τα ολοκληρώσουν (integrate) σε ένα ενιαίο σύστημα, και να βοηθήσουν τη διαλειτουργικότητα (interoperability) μεταξύ αυτών. Αυτό οδηγεί στην απαίτηση ότι ένα ΚΣ θα πρέπει να είναι ανοικτό (open) - δηλαδή να παρέχει τις υπηρεσίες του με καθορισμένους ανοικτούς κανόνες οι οποίοι καθορίζουν την σύνταξη (προδιαγραφή) αυτών των υπηρεσιών, καθώς και τους τρόπους κλήσης τους.

Επίσης, ένα ΚΣ δεν πρέπει να είναι στατικό, αλλά να έχει την δυνατότητα δυναμικής προσαρμογής σύμφωνα με τις απαιτήσεις των χρηστών του. Έτσι μία άλλη σημαντική απαίτηση για ένα ΚΣ είναι η εύκολη κλιμάκωσή του, δηλαδή το να υπάρχει δυνατότητα να προσθέτουμε ή να αφαιρούμε πόρους από το δίκτυο διασυνδεδεμένων υπολογιστών κατ' απαίτηση (on demand). Η υπηρεσία της κλιμάκωσης ή αλλιώς η δυνατότητα επεκτασιμότητας είναι ένας από τους κύριους λόγους της επέκτασης του Διαδικτύου, με όλα τα πλεονεκτήματα που βιώνουμε τις τελευταίες δεκαετίες από αυτό. Η επεκτασιμότητα περιλαμβάνει τρεις διαστάσεις: Πρώτη διάσταση είναι η επεκτασιμότητα ως προς το μέγεθος (size) του ΚΣ, υπό την έννοια του πλήθους των πόρων που μπορεί να προστεθούν σε αυτό. Η δεύτερη διάσταση είναι η γεωγραφική διασπορά των πόρων, με όλες τις απαιτήσεις που κάτι τέτοιο συνεπάγεται. Τρίτη διάσταση είναι η διαχειρισιμότητά του ΚΣ, ανεξάρτητα του εάν οι πόροι είναι πολλοί ή λίγοι και του πού αυτοί βρίσκονται γεωγραφικά. Το παρόν βιβλίο δίνει ιδιαίτερη σημασία στην τελευταία διάσταση, γι' αυτόν το λόγο αφιερώνει ολόκληρο κεφάλαιο σε αυτό.

Η ανάπτυξη των ΚΣ συμβαδίζει με όλες τις σύγχρονες τεχνολογικές εξελίξεις, όπως είναι η κινητή υπολογιστική, το διαδίκτυο των πραγμάτων, το υπολογιστικό νέφος, η διάχυτη υπολογιστική, οι έξυπνες συσκευές και η ασύρματη δικτύωση. Στο σχήμα 1.1 βλέπουμε την κατανομημένη φύση των σύγχρονων συστημάτων.



Σχήμα 1.1: Η κατακεντρωμένη φύση των σύγχρονων συστημάτων.

## 1.2 Κατακεντρωμένα Συστήματα έναντι Κεντροκοιμημένων Συστημάτων

Η ανάπτυξη των ΚΣ δεν είναι μια απλή διαδικασία καθώς απαιτεί κατάλληλους μηχανισμούς υποστήριξης και δίκτυα διασύνδεσης. Το ερώτημα που τίθεται είναι γιατί αξίζει η επένδυση σε τέτοια συστήματα, έναντι των παραδοσιακών κεντροκοιμημένων συστημάτων. Οι απαντήσεις είναι ποικίλες και διάφορες χωρίς αυτό να σημαίνει ότι η υιοθέτηση ενός ΚΣ αποτελεί πανάκεια σε όλες τις περιπτώσεις. Στην συνέχεια θα αναφερθούν οι βασικότεροι λόγοι για την επικράτηση των κατακεντρωμένων συστημάτων έναντι των κεντροκοιμημένων.

Βασικότερος λόγος είναι η αξιοποίηση των ήδη υπαρχόντων υπολογιστικών πόρων. Έτσι μπορούμε να πούμε ότι τα υπολογιστικά συστήματα σε ένα ΚΣ έχουν καλύτερη σχέση κόστους προς απόδοση σε σχέση με τα κεντροκοιμημένα συστήματα, τα οποία πολλές φορές είναι ακριβά και δεν αξιοποιούνται σύμφωνα με το ύψος της επένδυσης σε αυτά.

Τα κατακεντρωμένα υπολογιστικά συστήματα, επίσης, μπορούν να λειτουργούν με παραλληλισμό πετυχαίνοντας καλύτερους χρόνους εκτέλεσης αξιοποιώντας τη μεγαλύτερη υπολογιστική ισχύ που πραγματοποιείται μέσα από τη συνεργασία των περισσότερων υπολογιστικών συστημάτων. Δεδομένου ότι πάρα πολλές σύγχρονες εφαρμογές είναι από τη φύση τους κατακεντρωμένες, απαιτείται η ύπαρξη έξτρα μηχανισμών για την επίτευξη ανοχής σε σφάλματα και διαθεσιμότητας, αξιοπιστίας και ασφάλειας.

Η ύπαρξη αυτών των μηχανισμών οδηγεί ένα ΚΣ σε μια ανθεκτικότητα μεγαλύτερη έναντι ενός κεντροκοιμημένου. Τα ΚΣ, καταθέτοντας το φόρτο εργασίας τους σε πολλές μηχανές, ανταποκρίνονται καλύτερα σε περιπτώσεις βλάβης των μηχανών αυτών. Παραδείγματος χάρι, παρατηρούμε ότι υπηρεσίες στο διαδίκτυο μπορούν να συνεχίσουν να λειτουργούν παρά την απώλεια επεξεργαστικής ισχύος, έστω και με μειωμένη απόδοση.

Η κλιμακούμενη επέκταση των ΚΣ είναι ένα άλλο χαρακτηριστικό υπεροχής τους διότι δίνεται έτσι η δυνατότητα για σταδιακή επένδυση στην ανάπτυξή τους, στον έλεγχο εισαγωγή τους στο επιχειρησιακό περιβάλλον και στην αξιολόγησή τους. Ανάλογα με τις απαιτήσεις της εκάστοτε εφαρμογής μπορεί να επεκταθεί η υπολογιστική ισχύς του ΚΣ με τρόπο ευέλικτο και αποτελεσματικό.

Τέλος, ο διαμοιρασμός των πόρων, υπηρεσιών, δεδομένων και συσκευών που βρίσκονται διάσπαρτοι σε ένα ΚΣ αποτελεί εφελκυστικό για την καλύτερη αξιοποίησή τους, αλλά και για την καλύτερη εξυπηρέτηση των χρηστών τους. Κι αυτό έχει να κάνει με όλες τις διαστάσεις ενός πληροφοριακού συστήματος, ανεξάρτητα του αν πρόκειται για εφαρμογές και υπηρεσίες, για δεδομένα και αρχεία, για μονάδες επεξεργασίας, για αποθηκευτικούς χώρους, ή για εκτυπωτές και άλλες συσκευές.

Από την άλλη, τα ΚΣ παρουσιάζουν μειονέκτηματα σε σχέση με τα κεντροποιημένα σε σχέση με τη σχεδίαση και την ανάπτυξη του λογισμικού τους. Δεν υπάρχει συγκριτικά η αντίστοιχη εμπειρία, ενώ πολλές φορές σχεδιαστικά λάθη μπορεί να οδηγήσουν σε κακά αποτελέσματα.

Επιπρόσθετα, ένα ΚΣ εξαρτάται απόλυτα από την αξιόπιστη λειτουργία του δικτύου. Αργό δίκτυο σημαίνει αργή εκτέλεση της κατανεμημένης επεξεργασίας. Διακοπτόμενο δίκτυο σημαίνει πιθανή μη λειτουργία του κατανεμημένου συστήματος. Άρα σε κάθε περίπτωση, ένα κατανεμημένο σύστημα απαιτεί αξιόπιστο δίκτυο ανάλογης ταχύτητας.

Η ύπαρξη της διασυνδεσιμότητας μεταξύ των υπολογιστικών κόμβων του κατανεμημένου συστήματος αυξάνει σοβαρά την πιθανότητα επιθέσεων από επιτιθέμενους και κακόβουλους χρήστες. Άρα, παρότι και στα κεντροποιημένα συστήματα υπάρχει απαίτηση για ασφάλεια, στα κατανεμημένα η απαίτηση αυτή είναι σαφώς μεγαλύτερη.

### **1.3 Εξέλιξη των Συστημάτων μέχρι τα Κατανεμημένα Συστήματα**

Οι υπολογιστές γνωρίζουν ραγδαία ανάπτυξη μετά το δεύτερο Παγκόσμιο Πόλεμο και θα λέγαμε ότι οι εξελίξεις είναι καταγιστικές. Οι πρώτοι υπολογιστές που εμφανίστηκαν ήταν πολύ μεγάλου μεγέθους και με πολύ υψηλό κόστος απόκτησης. Αυτό είχε ως αποτέλεσμα οι οργανισμοί και οι επιχειρήσεις να επενδύουν σε μικρό αριθμό υπολογιστών, οι οποίοι λειτουργούσαν συνήθως ανεξάρτητα και μεμονωμένα διότι δεν υπήρχε τρόπος διασυνδέσής τους. Όταν αργότερα αναπτύχθηκαν τα τοπικά δίκτυα, μεγάλοι κεντρικοί υπολογιστές διασυνδέονταν με τερματικά χρηστών προκειμένου να τους παρέχουν επεξεργαστική ισχύ, αποθηκευτικό χώρο, πλατφόρμες ανάπτυξης, καθώς και έτοιμες εφαρμογές και εργαλεία.

Στις αρχές της δεκαετίας του 1980, έχουμε, όμως, δύο σημαντικά τεχνολογικά επιτεύγματα τα οποία έδωσαν μεγάλη ώθηση στη χρήση των υπολογιστών. Το πρώτο ήταν η ανάπτυξη των μικροεπεξεργαστών, οι οποίοι ώθησαν την κατασκευή μικροϋπολογιστών και το δεύτερο η δημιουργία τοπικών δικτύων υψηλών ταχυτήτων.

Το τελικό αποτέλεσμα αυτών των δύο καινούριων τεχνολογιών και επιτευγμάτων με τη συνδυαστική τους λειτουργία ήταν η ανάπτυξη υπολογιστικών συστημάτων που αποτελούνταν από μεγάλο αριθμό κεντρικών μονάδων επεξεργασίας συνδεδεμένων μέσω δικτύων υψηλών ταχυτήτων (high-speed networks). Παραδοσιακά τα συστήματα αυτά λειτουργούσαν με έναν κεντροποιημένο τρόπο αποτελούμενα από μία ή μερικές Κεντρικές Μονάδες Επεξεργασίας (ΚΜΕ) με μία κεντρική μνήμη, καθώς και τα περιφερειακά τους. Μέσω τερματικών σταθμών οι χρήστες ελάμβαναν τις υπηρεσίες του συστήματος.

Αρχικοί πρόγονοι των ΚΣ αποτελούν τα συστήματα πολυεπεξεργασίας με την χρήση πολυεπεξεργαστών μεταγωγής καταρχάς, κατόπιν των πολύ-υπολογιστών και, τέλος, των κατανεμημένων λειτουργικών συστημάτων. Στη συνέχεια θα ασχοληθούμε με αυτούς τους τρεις τύπους συστημάτων.

### 1.3.1. Κατηγοριοποίηση των συστημάτων επεξεργασίας κατά Flynn

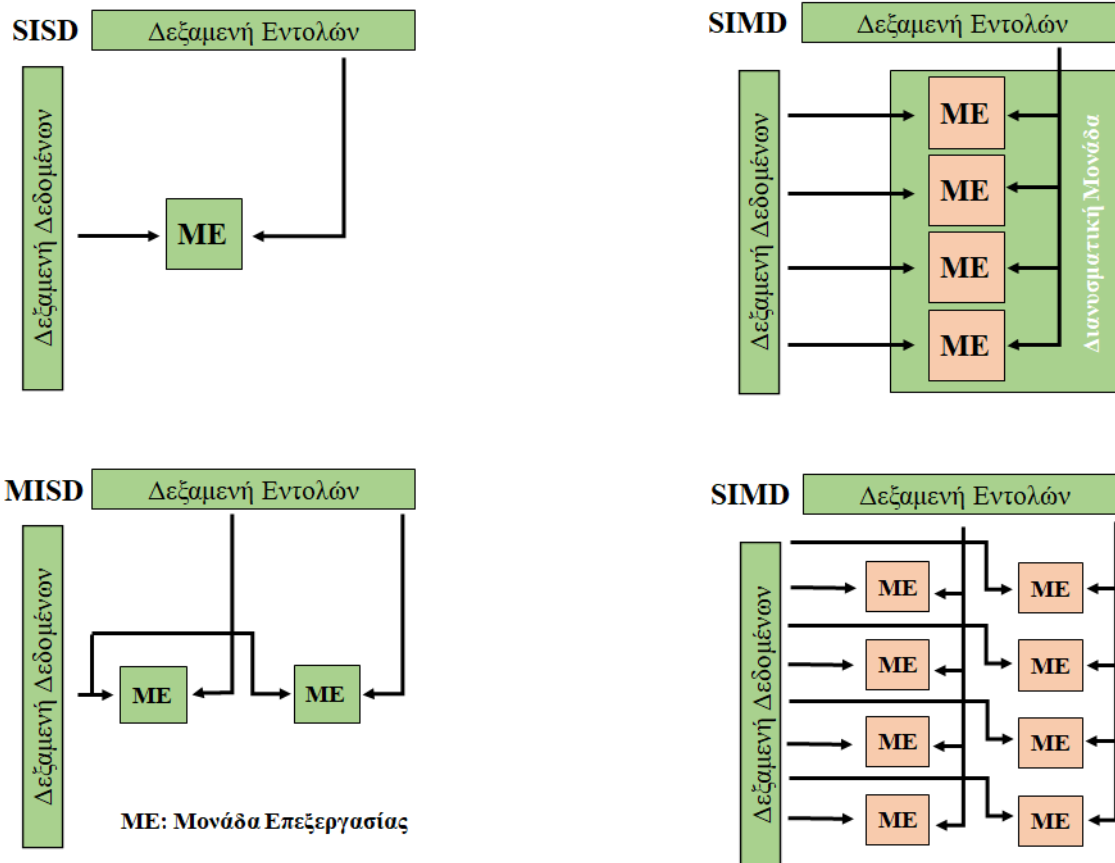
Η κατηγοριοποίηση των υπολογιστικών συστημάτων επεξεργασίας κατά Flynn έχει χρησιμοποιηθεί ως ένα αποτελεσματικό εργαλείο κατά τον σχεδιασμό των σύγχρονων επεξεργαστών και των λειτουργιών τους<sup>1</sup>. Η κατηγοριοποίηση αυτή λαμβάνει χώρα βάσει δύο κριτηρίων, (α) το πλήθος των ροών εντολών (instruction strings) και (β) το πλήθος των ροών δεδομένων (data streams). Βάσει αυτών των κριτηρίων διακρίνονται οι εξής κατηγορίες:

- Συστήματα τύπου SISD (Simple Instruction Simple data), όπως π.χ. είναι οι πρώτοι προσωπικοί υπολογιστές (PCs).
- Συστήματα τύπου SIMD (Simple Instruction Multiple Data), τα οποία αναφέρονται σε επεξεργαστές σε παράθεση (array processors), όπως π.χ. είναι οι λεγόμενοι υπερυπολογιστές (supercomputers)
- Συστήματα τύπου MISD (Multiple Instruction Simple Data), τα οποία δεν έχουν εφαρμοστεί επί της ουσίας σε κάποια γνωστά προϊόντα υπολογιστών.
- Συστήματα τύπου MIMD (Multiple Instruction Multiple Data), στα οποία ανήκουν και τα κατανεμημένα συστήματα

Στο σχήμα 1.2 παρουσιάζονται οι αρχιτεκτονικές αυτών των συστημάτων επεξεργασίας, όπου το PU είναι τα αρχικά του Processing Unit (Μονάδα Επεξεργασίας).

---

<sup>1</sup> <https://course.ece.cmu.edu/~ece447/s13/lib/exe/fetch.php?media=01447203.pdf> (πρόσβαση 10-09-2021)



Σχήμα 1.2: Κατηγοριοποίηση των συστημάτων επεξεργασίας κατά Flynn<sup>1</sup>.

Η παραπάνω κατηγοριοποίηση όμως δεν μας καλύπτει επαρκώς την περαιτέρω κατηγοριοποίηση των Συστημάτων τύπου MIMD, όπου μπορεί να υπάρχει μια ποικιλία τρόπων διασύνδεσης των μονάδων επεξεργασίας. Προς τούτο, υιοθετήθηκε η προσέγγιση του υλικού (hardware).

### 1.3.2 Προσέγγιση Υλικού

Η προσέγγιση του υλικού εστιάζει στον τρόπο με τον οποίο διασυνδέονται και επικοινωνούν οι πολλαπλές Κεντρικές Μονάδες Επεξεργασίας (ΚΜΕ) σε ένα σύστημα παράλληλης επεξεργασίας.

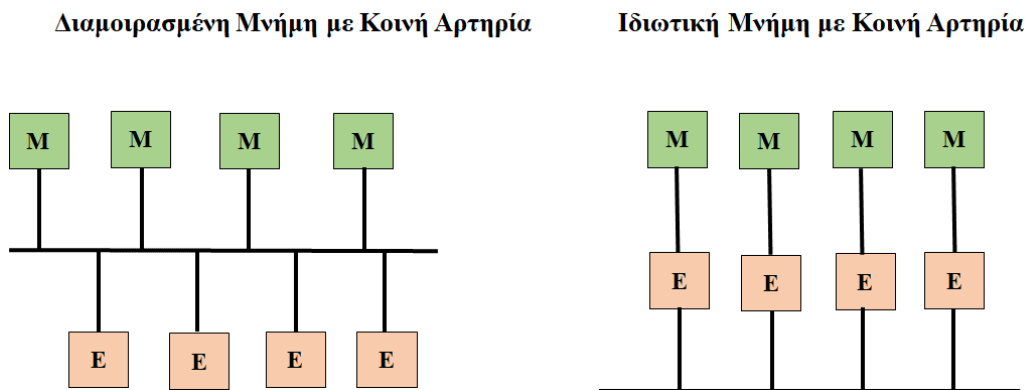
Όπως αναφέραμε ήδη, υπάρχουν οι πολυεπεξεργαστές (multiprocessors) με διαμοιραζόμενη κύρια μνήμη, καθώς και οι υπολογιστές (multi-computers), οι οποίοι απαιτούν τη χρήση ενός τοπικού δικτύου. Σε ένα πολυεπεξεργαστή δεδομένου του διαμοιρασμού της κύριας μνήμης από το σύνολο των επεξεργαστών, υπάρχει ένας και μοναδικός χώρος διευθύνσεων, ο οποίος καλείται και ιδεατός χώρος μνήμης. Σε αυτό το σημείο υπάρχουν δύο προσεγγίσεις στην αρχιτεκτονική διασύνδεσης μνημών και επεξεργαστών. Η πρώτη είναι η αρχιτεκτονική του δικτύου διασύνδεσης μέσω αρτηρίας (bus) και η δεύτερη μέσω μεταγωγής (switched).

Προς τούτο, δημιουργούνται τέσσερις κατηγορίες υλικού:

- οι πολυεπεξεργαστές αρτηρίας (bus-based multiprocessor),
- οι πολυεπεξεργαστές μεταγωγής (switched-based multiprocessor),
- οι υπολογιστές αρτηρίας (bus-based multicomputers) και, τέλος,
- οι υπολογιστές μεταγωγής (switched-based computers)

Οι πολυεπεξεργαστές είναι στενά συνδεδεμένα (tightly coupled) συστήματα, ενώ οι πολυπολογιστές χαλαρά συνδεδεμένα (loosely coupled) συστήματα. Τα στενά συνδεδεμένα συστήματα έχουν υψηλό ρυθμό μετάδοσης δεδομένων και μικρή καθυστέρηση μετάδοσης των μηνυμάτων. Από την άλλη, τα χαλαρά συνδεδεμένα συστήματα έχουν χαμηλό ρυθμό μετάδοσης δεδομένων και μεγάλη καθυστέρηση μετάδοσης των μηνυμάτων. Ας δούμε στη συνέχεια αυτές τις κατηγορίες υλικού.

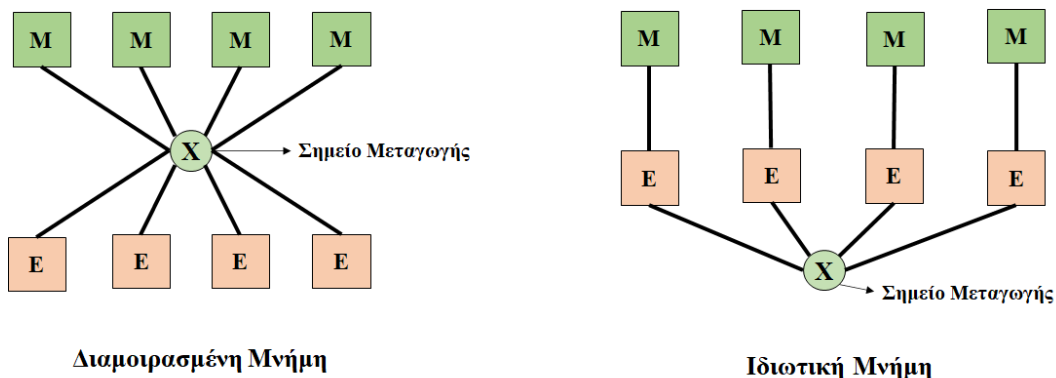
Οι Πολυεπεξεργαστές Αρτηρίας αποτελούνται από έναν αριθμό ΚΜΕ, όπου όλες συνδέονται με μία κοινή αρτηρία και ένα τμήμα μνήμης (βλ. σχήμα 1.3). Η αύξηση του πλήθους των ΚΜΕ που εξυπηρετούνται από την κοινή αρτηρία οδηγεί σε μείωση της απόδοσης του συστήματος. Μια προσεγγιστική λύση σε αυτό το πρόβλημα αποτελεί η χρήση *κρυφής μνήμης μεταξύ ΚΜΕ και αρτηρίας*. Στην κρυφή μνήμη, που θα εξετάσουμε και σε μεταγενέστερο κεφάλαιο, τηρείται αντίγραφο των πιο πρόσφατων προσπελάσεων στη μνήμη. Όλες οι αιτήσεις από μια ΚΜΕ προς τη μνήμη περνάνε πρώτα από την κρυφή μνήμη, προκειμένου να εξεταστεί εάν η κρυφή μνήμη διαθέτει τα στοιχεία που θέλει να ανακτήσει η ΚΜΕ. Όσο μεγαλύτερο είναι το ποσοστό επιτυχίας, τόσο μεγαλύτερη είναι η δυνατότητα διασύνδεσης περισσότερων ΚΜΕ στην αρτηρία. Εδώ βέβαια ανακύπτει το πρόβλημα συνέπειας της μνήμης. Αυτό σημαίνει ότι μπορεί τα δεδομένα που ανακτούμε από την κρυφή μνήμη να μην είναι ίδια με τα πραγματικά δεδομένα που θέλουμε να ανακτήσουμε από την κύρια μνήμη, διότι έχουν αλλάξει στο ενδιάμεσο από την αποθήκευσή τους στην κρυφή μνήμη. Το ερώτημα που εγείρεται είναι το τι συμβαίνει στην περίπτωση που κάποια ΚΜΕ αλλάξει την τιμή σε κάποιες θέσεις της κύριας μνήμης. Μια λύση είναι μία λέξη να εγγράφεται ταυτόχρονα στην κρυφή και στην κύρια μνήμη (write through cache technique). Μια άλλη λύση είναι οι κρυφές μνήμες να παρακολουθούν συνεχώς την αρτηρία (bus) και όταν εντοπιστεί εγγραφή σε θέση της κύριας μνήμης, τότε η κρυφή μνήμη ή διαγράφει την τιμή της δικής της θέσης ή την ενημερώνει με τη νέα τιμή (Snoopy cache).



Σχήμα 1.3: Αρχιτεκτονικές Κοινής Αρτηρίας

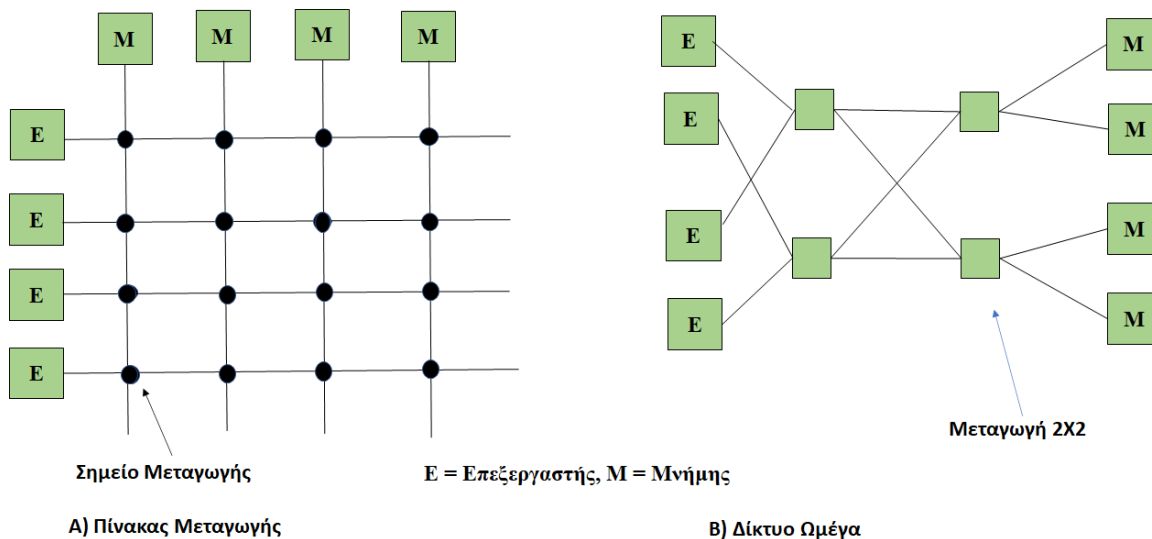
Οι πολυεπεξεργαστές μεταγωγής (σχήμα 1.4) είναι μια εναλλακτική αρχιτεκτονική επεξεργαστών, μνημών και αρτηριών. .

## ΜΕΤΑΓΩΓΗ



Σχήμα 1.4: Πολυεπεξεργαστές μεταγωγής

Στην περίπτωση που ο αριθμός των μονάδων επεξεργασίας αυξάνεται σημαντικά, π.χ. πλέον των 10 ή 15, τότε εισάγουμε τα δίκτυα μεταγωγής για την επικοινωνία επεξεργαστών – μνημών, όπου μεταξύ τους υπάρχει μια μοναδική διαδρομή επικοινωνίας. Και σε αυτή την περίπτωση υπάρχει το πρόβλημα της συνεκτικότητας, το οποίο επιλύεται με τα πρωτόκολλα καταλόγων. Οι κατάλογοι αυτοί είναι πίνακες που διατηρούνται στις μνήμες με πληροφορίες σχετικές με το ποια δεδομένα διατηρούνται σε ποια κρυφή μνήμη επεξεργαστή. Οπότε, σε μια τυχόν τροποποίηση των πρωτότυπων δεδομένων γίνονται οι αντίστοιχες ενημερώσεις στις κρυφές μνήμες. Αυτή η διαδικασία δημιουργεί πρόσθετο φόρτο εργασίας στα συστήματα και πρόσθετη κίνηση στα δίκτυα διασύνδεσης. Υπό αυτή την έννοια θα πρέπει να υπάρξει πρόβλεψη για τη μείωση του αριθμού τέτοιων και μάλιστα συνεχών ενημερώσεων. Στο σχήμα 1.5 βλέπουμε δύο περιπτώσεις δικτύων μεταγωγής.



Σχήμα 1.5: Δίκτυα Μεταγωγής: Α) Πίνακας Μεταγωγής και Β) Δίκτυο Ωμέγα

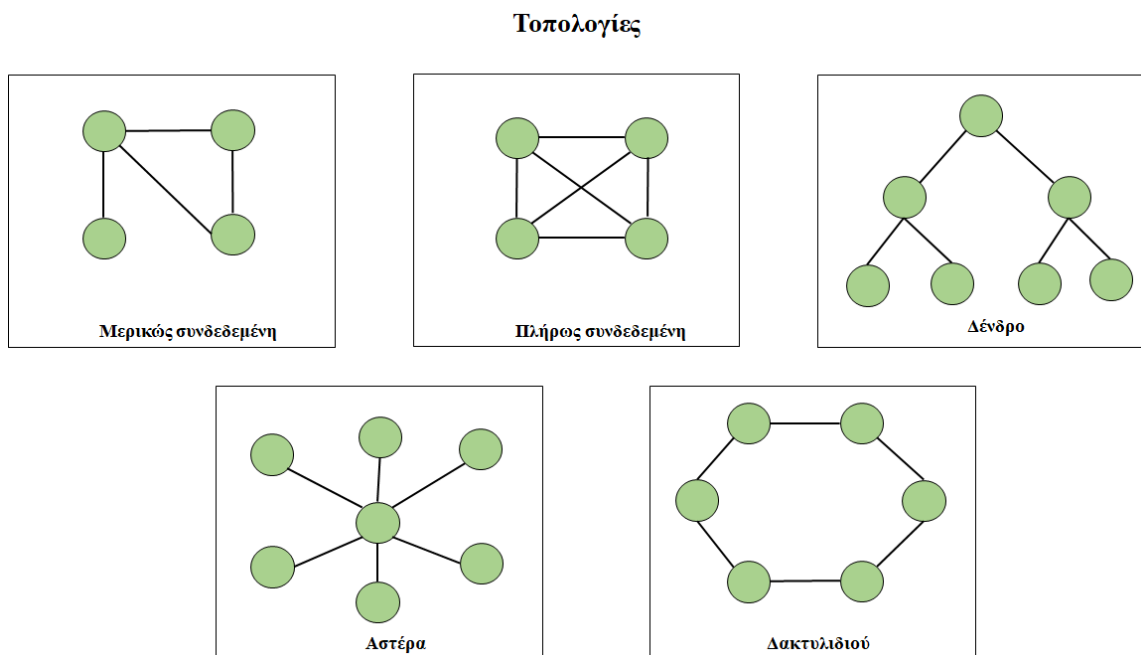
Βασικό μειονέκτημα με τον πίνακα μεταγωγής είναι ότι εάν έχουμε  $N$  επεξεργαστές και  $N$  μνήμες, τότε χρειάζονται  $N^2$  σημεία μεταγωγής, που είναι απαγορευτικού κόστους για μεγάλο αριθμό  $N$ . Επίσης και η χρήση δικτύων, όπως είναι το δίκτυο Ωμέγα, όπως φαίνεται στο σχήμα 1.5 παρουσιάζει μειονεκτήματα. Συγκεκριμένα, εάν έχουμε  $N$  επεξεργαστές και  $N$  μνήμες, τότε χρειάζονται  $N/2$  στάδια μεταγωγής, που το καθένα περιέχει  $\log_2 N$  μεταγωγούς. Η λύση είναι η κατασκευή μηχανών NUMA (NonUniform Memory Access-Μη Ομοιόμορφης Πρόσβασης Μνήμης). Στις μηχανές NUMA μία



μνήμη συσχετίζεται με κάθε επεξεργαστή. Οπότε με αυτό το σχήμα κάθε επεξεργαστής έχει γρήγορη πρόσβαση στη δική του μνήμη, αλλά αργή πρόσβαση σε όλες τις άλλες μνήμες. Οπότε εδώ εγείρεται το πρόβλημα το πώς μπορεί να γίνει η τοποθέτηση των διεργασιών και των δεδομένων τους, ώστε οι περισσότερες προσπελάσεις ενός συγκεκριμένου επεξεργαστή να γίνονται στη δική του μνήμη.

Οι επόμενες κατηγορίες εξεταζόμενων αρχιτεκτονικών είναι οι *ομογενείς πολύ-υπολογιστές* και οι *ετερογενείς πολύ-υπολογιστές*. Στους ομοιογενείς πολύ-υπολογιστές υπάρχει ένα μοναδικό δίκτυο διασύνδεσης που χρησιμοποιεί παντού την ίδια τεχνολογία, όλοι οι επεξεργαστές είναι ίδιοι, ενώ εν γένει έχουν πρόσβαση στην ίδια ποσότητα ιδιωτικής μνήμης. Εδώ έχουμε την τυπική περίπτωση ενός παράλληλου συστήματος. Από την άλλη μεριά, στους ετερογενείς πολύ-υπολογιστές υπάρχει ετερογένεια στο δίκτυο διασύνδεσης, καθώς και στους επεξεργαστές. Και στις δύο κατηγορίες υπάρχουν αιτιάσεις για το υψηλό κόστος κατασκευής τους, αλλά είναι έξω από τον σκοπό του βιβλίου η αναλυτική παρουσίαση των αιτιάσεων αυτών.

Οι διασυνδεόμενοι πολύ-υπολογιστές στο πλαίσιο της παράλληλης κατανομής επεξεργασίας μπορεί να χρησιμοποιούν διάφορες τοπολογίες. Η επιλογή της κατάλληλης ή/και εφικτής τοπολογίας εξαρτάται από διάφορα κριτήρια, όπως το κόστος διασύνδεσης, δηλαδή το πόσο κοστίζει να συνδέσουμε τους διάφορους υπολογιστές μεταξύ τους, το κόστος επικοινωνίας, δηλαδή το πόσος χρόνος απαιτείται για να στείλουμε ένα μήνυμα από τον υπολογιστή Α στον υπολογιστή Β και την αξιοπιστία, δηλαδή το θα συνεχίσουν να επικοινωνούν μεταξύ τους οι υπόλοιποι υπολογιστές αν αστοχήσει μια δικτυακή σύνδεση ή ένας υπολογιστής του πολυυπολογιστικού συστήματος. Οι διάφορες τοπολογίες αναπαρίστανται ως γράφοι, των οποίων οι κόμβοι αντιστοιχούν σε υπολογιστές. Μια ακμή από τον κόμβο Α στον κόμβο Β αντιστοιχεί σε μία άμεση σύνδεση των δύο αυτών κόμβων. Στο σχήμα 1.6 φαίνονται οι διάφορες τοπολογίες πολύ-υπολογιστών.



Σχήμα 1.6: Τοπολογίες πολυυπολογιστών

### 1.3.3 Δίκτυα Υπολογιστών

Τα δίκτυα υπολογιστών ήρθαν για να δημιουργήσουν ευελιξία στη διασύνδεση μεταξύ των υπολογιστών και να επιτρέψουν τη συνεργασία μεταξύ τους. Ενώ στα πολύ-υπολογιστικά συστήματα ο στόχος είναι η παράλληλη εκτέλεση των διεργασιών – εργασιών, στα δίκτυα υπολογιστών ο στόχος είναι η διασύνδεση ανεξάρτητων υπολογιστών μεταξύ τους.

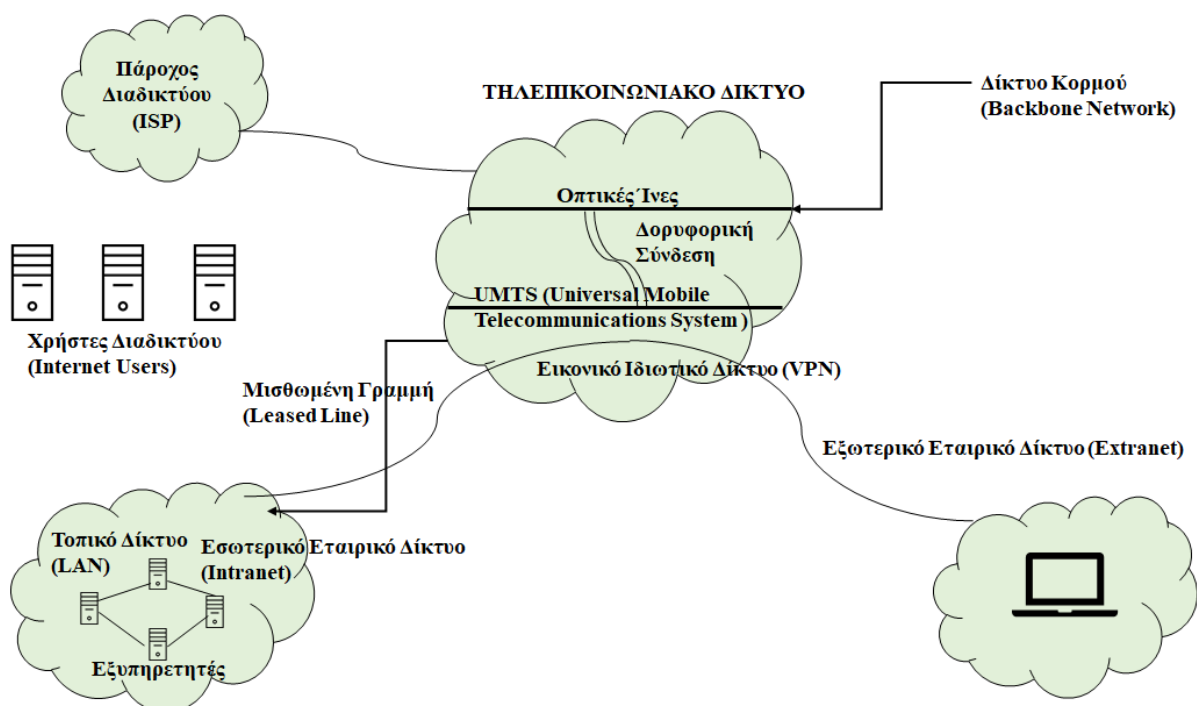
Στην πραγματικότητα υπάρχει ένας υψηλός βαθμός ετερογένειας των κόμβων. Οι κατηγορίες κόμβων (υπολογιστών) που έχουμε είναι:

- τα παράλληλα συστήματα υψηλής απόδοσης (πολλαπλοί επεξεργαστές, καθώς και πολυυπολογιστές)
- οι προσωπικοί υπολογιστές και οι εξυπηρετητές υψηλής τεχνολογίας
- οι απλοί υπολογιστές δικτύου (προσφέρουν στους χρήστες μόνο πρόσβαση στο δίκτυο)
- οι κινητοί-φορητοί υπολογιστές, και
- οι σταθμοί εργασίας πολυμέσων.

Όλα αυτά θα πρέπει να έχουν δυνατότητα διασύνδεσης μέσω των δικτύων, τα οποία όμως με τη σειρά τους παρουσιάζουν επίσης υψηλό βαθμό ετερογένειας. Συνοπτικά έχουμε:

- τα δίκτυα τοπικής περιοχής (Local Area Networks-LAN)
- τις ασύρματες συνδέσεις (wireless networks)
- τις συνδέσεις POTS (Plain Old Telephone Service) μεγάλης απόστασης και μεγάλης καθυστέρησης, και
- τις συνδέσεις μεταγωγής ευρείας περιοχής

Στο σχήμα 1.7 βλέπουμε ένα παράδειγμα τοπολογίας δικτύων και υπολογιστών.



Σχήμα 1.7: Παράδειγμα τοπολογίας δικτύου υπολογιστών ευρείας και τοπικής εμβέλειας.

### 1.3.4. Θέματα Λογισμικού (Software)

Το λογισμικό είναι εξαιρετικής σημασίας στα διασυνδεδεμένα συστήματα και θα λέγαμε και πιο σημαντικό από το υλικό διότι αυτό είναι που αναλαμβάνει να κρύψει και να διαχειριστεί όλες τις ετερογένειες. Πράγματι, η εικόνα που ένα υπολογιστικό σύστημα παρουσιάζει στους χρήστες του εξαρτάται από το λογισμικό του λειτουργικού συστήματος (operating system) και όχι από το υλικό. Ένα *λειτουργικό σύστημα στενής σύνδεσης* είναι ένα κατακεντρωμένο λειτουργικό σύστημα το οποίο διαχειρίζεται πολυεπεξεργαστικά και ομοιογενή πολυπολογιστικά περιβάλλοντα. Ενώ ένα *λειτουργικό σύστημα χαλαρής σύνδεσης* είναι ένα δικτυακό λειτουργικό σύστημα, το οποίο χρησιμοποιείται για τη διαχείριση ετερογενών πολυπολογιστών. Σημειώνεται ότι το χαλαρά ή στενά συνδεδεμένο λογισμικό χαρακτηρίζεται ανάλογα με τον βαθμό ανεξαρτησίας και αλληλεπίδρασης των χρηστών και του υπολογιστικού συστήματος.

Στη συνέχεια θα εξετάσουμε τρεις προσεγγίσεις λειτουργικών συστημάτων – λογισμικού, τα οποία κινούνται στην κατεύθυνση ανάπτυξης των κατάλληλων υποδομών για τη λειτουργία κατακεντρωμένων συστημάτων. Η κάθε προσέγγιση το πετυχαίνει σε διαφορετικό βαθμό και με διαφορετικό τρόπο. Οι κατηγορίες αυτές είναι τα Κατακεντρωμένα Λειτουργικά Συστήματα (Distributed Operating Systems), τα Δικτυακά Λειτουργικά Συστήματα (Network Operating Systems) και το Ενδιάμεσο Λογισμικό (Middleware).

Πριν, όμως, από την παρουσίασή τους να αναφέρουμε ότι εννοιολογικά αυτά που τίθενται ως απαίτηση από ένα λειτουργικό σύστημα με χαρακτηριστικά κατακεντρωμένου είναι:

- οι χρήστες να μην έχουν γνώση για την πολλαπλότητα των μηχανών σε ένα δίκτυο
- η πρόσβαση σε απομακρυσμένους πόρους να γίνεται όπως ακριβώς η πρόσβαση σε τοπικούς πόρους.
- να υπάρχει διαφανής μετανάστευση δεδομένων (data migration) με τη μεταφορά ολοκλήρου ή μέρους αρχείου σε τρίτο υπολογιστή
- να επιτρέπεται η μετανάστευση υπολογισμών και κώδικα σε τρίτο υπολογιστή
- να επιτρέπεται η μετανάστευση διεργασιών και εκτέλεση μιας ολόκληρης διεργασίας ή τμημάτων αυτής σε τρίτο υπολογιστή
- να υποστηρίζεται η δυνατότητα εξισορρόπησης του φόρτου εργασίας και η παράλληλη εκτέλεση διεργασιών.

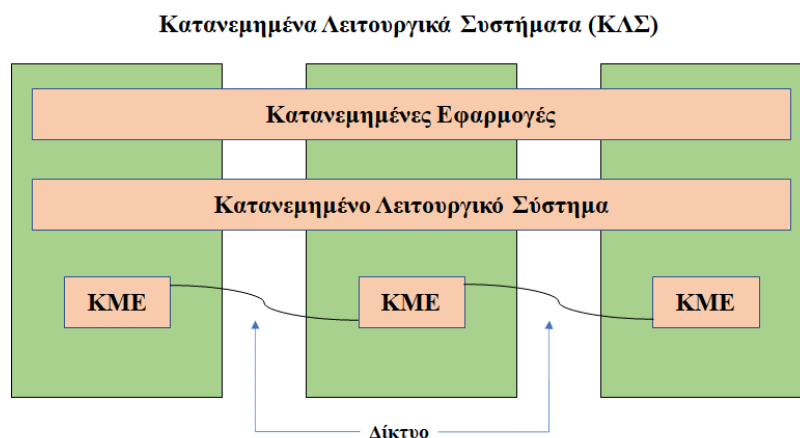
Δεδομένης της δικτυακής προσέγγισης μεταξύ των αλληλοεπιδρώντων υπολογιστών, υπήρξε η απαίτηση για την ανάπτυξη εξειδικευμένων λειτουργικών συστημάτων, όπως είναι τα λειτουργικά συστήματα πολυπολογιστών, καθώς και τα δικτυακά λειτουργικά συστήματα.

Τα *λειτουργικά συστήματα πολυπολογιστών* ή αλλιώς *κατακεντρωμένα λειτουργικά συστήματα* (distributed operating system) (ΚΛΣ) είναι δυσκολότερα στην κατασκευή τους από τα παραδοσιακά λειτουργικά συστήματα πολυεπεξεργαστών, δεδομένου ότι η μνήμη δεν είναι διαμοιραζόμενη, ενώ η έμφασή τους μετατοπίζεται στην επικοινωνία μεταξύ των επεξεργαστών με την ανταλλαγή μηνυμάτων.

Τα ΚΛΣ είναι ισχυρά συνδεδεμένα λειτουργικά συστήματα για ομογενοποιημένους πολυπολογιστές (Σχήμα 1.8) και προσφέρουν καθολική επικοινωνία καθώς και διαχείριση των κατακεντρωμένων πόρων υλικού. Μόνο οι πολυπολογιστές βασισμένοι στην αρχιτεκτονική αρτηρίας παρέχουν δυνατότητα

ανοικτής εκπομπής (broadcasting) βασισμένη στο υλικό. Ένα αποτελεσματικό broadcasting απαιτεί ειδικές προγραμματιστικές τεχνικές διεπαφής δικτύου (network interfaces).

Επίσης, οι μηχανισμοί συγχρονισμού μεταξύ όλων των υπολογιστικών συστημάτων δεν είναι απλοί. Η εικονική διαμοιραζόμενη μνήμη απαιτεί ειδικά λειτουργικά συστήματα προκειμένου να διατηρείται ένας πίνακας καθολικής μνήμης με τεχνικές λογισμικού. Η έμφυτη διαχείριση των διεσπαρμένων στο δίκτυο πόρων μας οδηγεί στο να μην υπάρχει ένα κεντρικό σημείο λήψης αποφάσεων σε σχέση με την τοποθεσία των χρησιμοποιούμενων πόρων. Μόνο λίγα πολυυπολογιστικά λειτουργικά συστήματα υπάρχουν, όπως είναι το αποεβα. Βασικός λόγος είναι ότι ειδικές αρχιτεκτονικές τους έχουν υψηλό κόστος κατασκευής και σημαντική δυσκολία στη διαχείριση.



Σχήμα 1.8: Διαστρωμάτωση των Καταναμημένων Λειτουργικών Συστημάτων

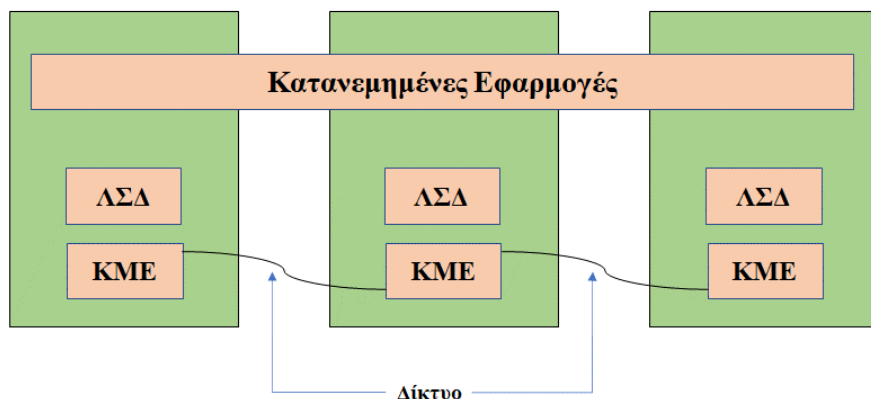
Γι' αυτόν ακριβώς το λόγο, υπήρξε η απαίτηση για την ανάπτυξη των Λειτουργικών Συστημάτων Δικτύου (ΛΣΔ) (*network operating systems*). Τα ΛΣΔ είναι χαλαρά συνδεδεμένα λειτουργικά συστήματα κυρίως για ετερογενείς υπολογιστές διασυνδεδεμένους μέσω ενός τοπικού δικτύου (LAN) ή μέσω ενός δικτύου ευρείας περιοχής (WAN), με κύριο στόχο να παρέχονται υπηρεσίες από αυτούς τους υπολογιστές σε απομακρυσμένους πελάτες (clients).

Στα ΛΣΔ κάθε υπολογιστής διαθέτει το δικό του λειτουργικό σύστημα με δυνατότητες δικτύωσης. Οι υπολογιστές λειτουργούν ανεξάρτητα (δηλαδή, μπορεί να έχουν διαφορετικά λειτουργικά συστήματα), ενώ οι υπηρεσίες αναφέρονται σε μεμονωμένους υπολογιστικούς κόμβους (π.χ. http, smtp, ftp, telnet, RPC, database, κτλ.). Άρα η σύνδεση των πελατών γίνεται δικτυακά σε υπολογιστές που τρέχουν το ΛΣΔ και οι οποίοι παρέχουν τις ζητούμενες υπηρεσίες καλώντας την κατάλληλη θύρα που ακούει η υπηρεσία με το κατάλληλο πρωτόκολλο εφαρμογής.

Τα ΛΣΔ έχουν ιδιαίτερο προσανατολισμό στον διαμοιρασμό αρχείων και είναι ένας από τους βασικούς λόγους που αυτά δημιουργήθηκαν. Με κατάλληλα δικαιώματα είναι εφικτό ένας πελάτης να έχει πρόσβαση στα αρχεία του απομακρυσμένου υπολογιστή (εξυπηρετητή) και αυτό ήταν μία από τις πρωτοπορίες των ΛΣΔ. Βέβαια τα αρχεία δεν είναι οι μόνοι υπολογιστικοί πόροι που διαμοιράζονται μέσω του δικτύου. Μπορεί, π.χ. μέσω απομακρυσμένης διασύνδεσης (remote logging), να γίνεται μεταφορά δεδομένων από απομακρυσμένες σε τοπικές μηχανές και αντίστροφα μέσω του πρωτοκόλλου μεταφοράς αρχείων (file transfer protocol - ftp), ή της απομακρυσμένης κλήσης μεθόδου (remote procedure call).

Ως παράδειγμα, θα εστιάσουμε στα Δικτυακά Συστήματα Αρχείων (Network File Systems - NFS) σε χαλαρά συνδεδεμένους υπολογιστές μέσω ενός ΛΣΔ. Ένα λειτουργικό σύστημα δικτύου προϋποθέτει γνώση εκ μέρους του χρήστη/πελάτη των πόρων που διαθέτει το δίκτυο. Σχετικά με τα αρχεία, μπορεί να υπάρχει ένα καθολικό και διαμοιραζόμενο σύστημα αρχείων (file system) προσβάσιμο από όλους τους υπολογιστικούς κόμβους. Οι εξυπηρετητές αρχείων μπορεί να μην περιορίζονται σε έναν και μόνο υπολογιστή (file server) για λόγους απόδοσης του συστήματος. Οι εξυπηρετητές διατηρούν ιεραρχικά συστήματα αρχείων (π.χ. Windows, Linux), ενώ οι πελάτες μπορούν να προσαρτήσουν (mount) μέρος των συστημάτων αρχείων πάνω στα δικά τους τοπικά συστήματα. Οπότε η πρόσβαση στα αρχεία γίνεται σαν να ήταν τα αρχεία αυτά τοπικά στο σύστημα αρχείων στον σταθμό εργασίας του πελάτη. Το ΛΣΔ μπορεί να είναι ή να μην είναι το ίδιο σε όλους τους σταθμούς εργασίας ή του εξυπηρετητές. Οπότε οι απαιτήσεις για το σύνολο του συστήματος είναι μινιμαλιστικές, όπως π.χ. στο NFS της Sun.

### Λειτουργικά Συστήματα Δικτύου (ΛΣΔ)



ΛΣΔ: Λ.Σ. Δικτύου (Windows / Linux / Unix)

Σχήμα 1.9: Διαστρωμάτωση των Λειτουργικών Συστημάτων Δικτύου

Σε ρεαλιστικές συνθήκες δεν υπάρχει σύστημα που να μπορεί να ικανοποιεί όλα τα κριτήρια ώστε να θεωρείται απολύτως καταναμημένο σύστημα, παρότι υπάρχουν αρκετά τα οποία προσεγγίζουν τις απαραίτητες συνθήκες.

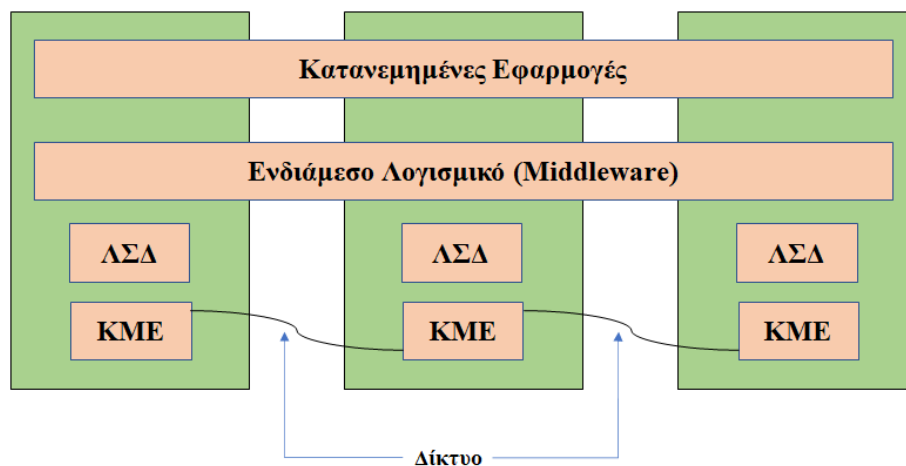
Τα ΛΣΔ έχουν να επιλύσουν προβλήματα υπηρεσιών διαφάνειας, ενώ τα ΚΛΣ είναι ιδιαίτερα ακριβά για να έχουν επαρκή διάδοση και καθιέρωση. Για να λυθεί το πρόβλημα αυτό, η λύση είναι η χρήση του ενδιάμεσου λογισμικού (middleware), το οποίο αποτελεί ένα επιπλέον επίπεδο λογισμικού, πάνω από το λειτουργικό σύστημα δικτύου, το οποίο προστίθεται ώστε να παράσχει κατάλληλες υπηρεσίες διαφανειών κατανομής, διαλειτουργικότητας και επεκτασιμότητας (Σχήμα 1.10). Το ενδιάμεσο λογισμικό τοποθετείται ενδιάμεσα από τις καταναμημένες εφαρμογές οι οποίες συνιστούν ένα Καταναμημένο Πληροφοριακό Σύστημα (ΚΠΣ) και το δικτυακό λειτουργικό σύστημα προσδίδοντας τα εξής πρόσθετα χαρακτηριστικά:

- τη γνώση του εκάστοτε υπολογιστή για τους άλλους υπολογιστές
- την ικανότητα να χρησιμοποιείται από υπολογιστές με διαφορετικό λειτουργικό σύστημα διαχειριζόμενο έτσι την ετερογένεια των συστημάτων
- την επικοινωνία και τον διαμοιρασμό κοινόχρηστων πόρων του
- ένα σύνολο καταναμημένων υπηρεσιών στους υπολογιστές του συστήματος παρεχόμενων διαφανώς (transparently)
- διευκόλυνση του προγραμματισμού των καταναμημένων συστημάτων.

Το κίνητρο δημιουργίας του ενδιάμεσου λογισμικού έχει να κάνει, συνεπώς, με τα παραπάνω προβλήματα. Χωρίς τη βοήθεια του ενδιάμεσου λογισμικού δυσχεραίνεται η δυνατότητα τμήματα εφαρμογών να τρέχουν σε διαφορετικά ΛΣΔ, καθώς και η δυνατότητα ενσωμάτωσης και διαλειτουργικότητας των υπηρεσιών των ΛΣΔ. Επίσης, το ενδιάμεσο λογισμικό ενισχύει την ενοποίηση εφαρμογών και πληροφοριακών συστημάτων. Είναι η ενοποιημένη εικόνα των εφαρμογών παρά τη χρήση κατανεμημένων βάσεων δεδομένων που δίνει προβάδισμα στη χρήση του ενδιάμεσου λογισμικού έναντι των ΛΣΔ, ενώ ταυτόχρονα σε αυτή την ενοποιημένη εικόνα δύνανται να προστίθενται νέες υπηρεσίες και τεχνολογίες, όπως πχ της κινητής υπολογιστικής, των υπηρεσιών νέφους, των διασυνεργατικών εφαρμογών και του ιστοχώρου.

Το ενδιάμεσο λογισμικό παρέχει αρκετές υπηρεσίες βάσει των κινήτρων δημιουργίας του. Προσφέρει υπηρεσίες επικοινωνίας, μέσω κλήσης μεθόδων και περάσματος μηνυμάτων, διαφανούς προσπέλασης, όπως π.χ. για έναν κινούμενο κόμβο, ονοματοδοσίας και ευρετηρίασης, π.χ. για υπηρεσίες στον παγκόσμιο ιστό, ασφάλειας, οι οποίες διασφαλίζουν τις διαδικασίες και την επικοινωνία μέσω της επαλήθευσης των υπηρεσιών και την κρυπτογράφησή τους, μηχανισμούς διατήρησης για αποθήκευση δεδομένων σε κατανεμημένα συστήματα αρχείων και σε βάσεις δεδομένων, καθώς και υπηρεσίες ελέγχου πρόσβασης στα δεδομένα ή σε κώδικα. Περισσότερη ανάλυση των υπηρεσιών θα δοθεί στο επόμενο κεφάλαιο.

### Κατανεμημένα Πληροφοριακά Συστήματα



ΛΣΔ: Λ.Σ. Δικτύου (Windows / Linux / Unix)

Σχήμα 1.10: Διαστρωμάτωση των Κατανεμημένων Πληροφοριακών Συστημάτων

Βάσει των παραπάνω περιγραφών για να συγκριθούν μεταξύ τους τα κατανεμημένα λειτουργικά συστήματα, τα δικτυακά λειτουργικά συστήματα και το ενδιάμεσο λογισμικό, πρέπει να τεθούν ερωτήματα όπως εκείνο του όμοιου λειτουργικού συστήματος σε κάθε κόμβο, της κλιμάκωσης και της ανοικτότητάς του συστήματος, της επικοινωνίας, του βαθμού διαφάνειας, της διαχειρισιμότητάς των πόρων, και του αριθμού αντιγράφων του λειτουργικού συστήματος.

Εν συνόψει, μπορούμε να πούμε ότι τα ΛΣΔ έχουν χαμηλό διαφάνειας σε σχέση με τα άλλα λειτουργικά συστήματα, ενώ τα ΛΣΔ και το ενδιάμεσο λογισμικό υπηρετούν τη διαχείριση της ετερογένειας. Ενώ τα ΛΣΔ έχουν την επικοινωνία τους βασισμένη στον διαμοιρασμό αρχείων, τα άλλα λειτουργικά συστήματα είναι πιο αποτελεσματικά μέσω του διαμοιρασμού μνήμης ή μέσω της ανταλλαγής μηνυμάτων. Τα ΛΣΔ και το ενδιάμεσο λογισμικό έχουν καλύτερη ανάλυση της λεπτομέρειας

(granularity) από την άποψη ότι ο κάθε κόμβος είναι διαχειριζόμενος εξατομικευμένα και αυτό συν τοις άλλοις τα βοηθάει να είναι εύκολα επεκτάσιμα και σαφώς πιο ανοικτά από τα ΚΛΣ. Εν κατακλείδει, βλέπουμε ότι η λύση του ενδιάμεσου λογισμικού προκρίνεται έναντι των υπολοίπων λόγω της ευελιξίας που παρουσιάζει, της ανοικτότητας και διαλειτουργικότητας, παρουσιάζοντας ταυτόχρονα μια ενιαία εικόνα των συστημάτων και των εφαρμογών.

#### **1.4 Κατανεμημένα Πληροφοριακά Συστήματα και Διαχείρισή τους.**

Όπως είδαμε και προηγουμένως, το ενδιάμεσο λογισμικό και τα ΚΣ έχουν βασικό στόχο την ενοποίηση των διάσπαρτων εφαρμογών σε ένα επιχειρησιακό περιβάλλον. Αυτό οδηγεί στον συγκερασμό των Πληροφοριακών Συστημάτων (ΠΣ) με τις τεχνολογίες των ΚΣ στα λεγόμενα *Κατανεμημένα Πληροφοριακά Συστήματα (ΚΠΣ)*. Να θυμίσουμε ότι ένα πληροφοριακό σύστημα, έχοντας διαθέσιμο ένα σύνολο δεδομένων, επεξεργάζεται αυτά, με τρόπο κεντρικοποιημένο ή κατανεμημένο, σύμφωνα με μια συγκεκριμένη επιχειρησιακή λογική και παράγει αποτελέσματα αξιοποιήσιμα από τους χρήστες του, προκειμένου να ολοκληρώσουν κάποια δοσοληψία ή επιχειρησιακή λειτουργία, ή να πάρουν κάποιες αποφάσεις.

Σε αυτή την κατεύθυνση, η προσέγγιση της υπηρεσιοστρεφούς αρχιτεκτονικής (Service-Oriented Architecture - SOA) φαίνεται να εξυπηρετεί τον στόχο υλοποίησης των ΚΠΣ. Με το ενδιάμεσο λογισμικό, οι διεργασίες ή αλλιώς οι υπηρεσίες μας είναι κατανεμημένες και επικοινωνούν διαφανώς με κατάλληλους μηχανισμούς. Οι υπηρεσίες μπορεί να θεωρηθούν αντικείμενα κατά το πρότυπο του αντικειμενοστρεφούς προγραμματισμού. Η σύνθεση των υπηρεσιών προκειμένου να επιτύχουν έναν επιχειρησιακό στόχο μπορεί να θεωρηθεί το ανάλογο της σύνθεσης των υπολογιστικών συστημάτων σε ένα ΚΣ προκειμένου να δώσουν την εικόνα ενός υπολογιστή ή μιας υπηρεσίας.

Στην Υπηρεσιοστρεφή Αρχιτεκτονική οι επιμέρους συνιστώσες-συστατικά (components) χρησιμοποιούνται ως υπηρεσίες (services) και είναι διαθέσιμες στο δίκτυο ή στον παγκόσμιο ιστό. Οι υπηρεσίες αυτές επικοινωνούν μέσω διεπαφών υλοποιημένων ανεξαρτήτως προγραμματιστικής γλώσσας, λειτουργικού συστήματος και υλικού. Άρα έχουμε πλήρη διαχείριση της ετερογένειας, ενώ οι υπηρεσίες ολοκληρώνονται χαλαρά. Άρα, έχουμε ένα ΚΣ σε επίπεδο εφαρμογής και αυτό το καλούμε Κατανεμημένο Πληροφοριακό Σύστημα. Εναλλακτικά έχουμε δει και τη χρήση του όρου Δικτυοκεντρικό Πληροφοριακό Σύστημα. Υπάρχει, εντούτοις, μια βασική διαφορά υπό την έννοια τα Κατανεμημένα Πληροφοριακά Συστήματα δεν περιορίζονται μόνο στο επίπεδο εφαρμογής, π.χ. τουριστικών, ή υγείας, αλλά περιλαμβάνουν και τις υπηρεσίες της υποκείμενης πλατφόρμας που δίνουν την απαιτούμενη λειτουργικότητα στις επιχειρησιακές εφαρμογές προκειμένου να λειτουργούν αυτές κατανεμημένα και αποτελεσματικά.

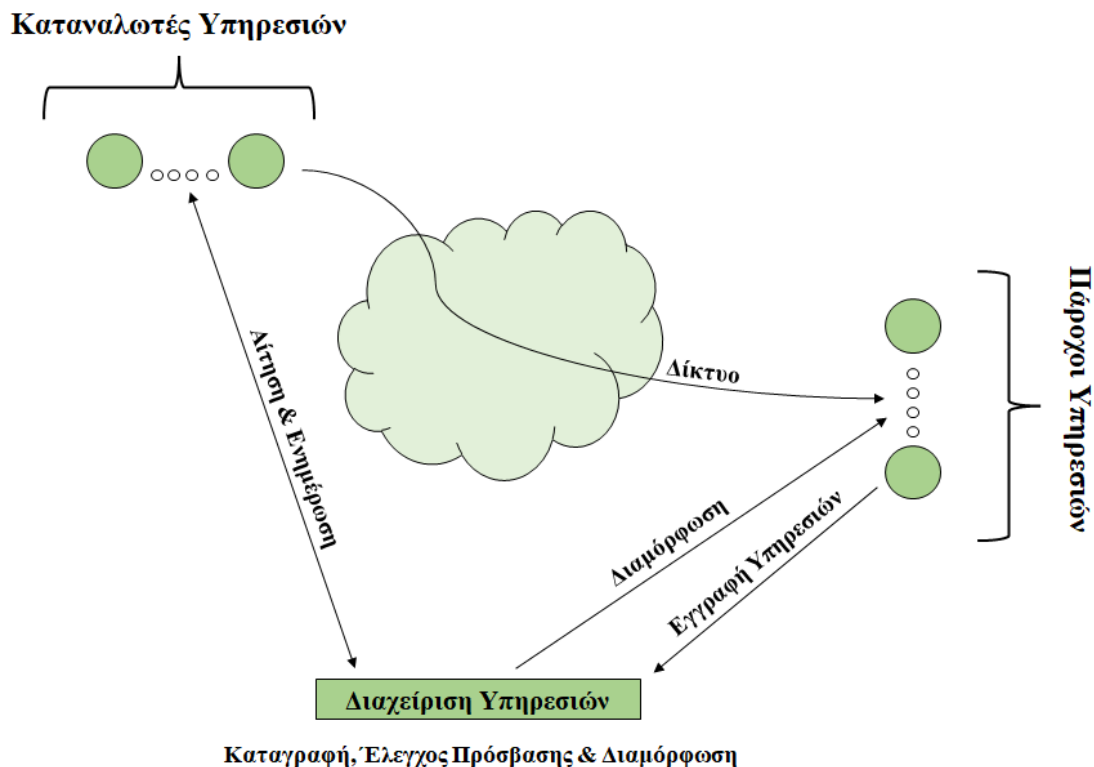
Σε κάθε περίπτωση, τα ΚΠΣ οδηγούνται από την ανάγκη για την ύπαρξη ευέλικτων επιχειρησιακών εφαρμογών και την ανταλλαγή πληροφοριών μεταξύ ετερογενών συστημάτων και πλατφορμών. Πλεονεκτήματα των χαλαρά ολοκληρωμένων συστημάτων είναι η ευελιξία τους, η ικανότητά τους να προσαρμόζονται στις δομικές αλλαγές για την υλοποίηση των υπηρεσιών, η ικανότητα προτυποποίησης στην ανταλλαγή μηνυμάτων, η δυνατότητα αξιοποίησης μέσω ολοκλήρωσης υπαρχόντων συστημάτων και η αύξηση της ταχύτητας στην ανταλλαγή πληροφοριών.

Με την έννοια υπηρεσία εννοούμε μια μεμονωμένη μονάδα μιας υπηρεσιοστραφούς αρχιτεκτονικής, η οποία συνίσταται από μια λογική ομάδα συνιστωσών (components), με κύριο σκοπό την εκπλήρωση των απαιτήσεων μιας επιχείρησης.

Η χρήση μιας υπηρεσίας περιλαμβάνει τα εξής: την εύρεση (Find), τη σύνδεση (Bind) και την εκτέλεση (Execute) της υπηρεσίας. Το σήμα κατατεθέν αυτών των υπηρεσιών είναι η Υπηρεσία Ιστού (Web Service – WS), η οποία περιλαμβάνει:

- τον *πρόχο υπηρεσιών (service provider)*, ο οποίος παρέχει ευμετάβλητες επιχειρησιακές υπηρεσίες
- το *μητρώο υπηρεσιών (service registry)*, το οποίο παρέχει στους καταναλωτές (consumers) τη δυνατότητα να αναζητήσουν τις υπηρεσίες που πληρούν τις απαιτούμενες προδιαγραφές, και
- τον *καταναλωτή υπηρεσιών (service consumer)*, ο οποίος χρησιμοποιεί τον πρόχο υπηρεσιών για την ολοκλήρωση της επιχειρησιακής διαδικασίας.

Στο σχήμα 1.11 βλέπουμε μια πιο εκτεταμένη προσέγγιση της υπηρεσιοστρεφούς αρχιτεκτονικής όπου υπάρχει και η πρόβλεψη της διαχείρισης των υπηρεσιών, υπό την έννοια όχι μόνο της εγγραφής τους αλλά και του ελέγχου πρόσβασης σε αυτές και της διαμόρφωσής τους. Σε αυτή την περίπτωση η διαχείριση των υπηρεσιών είναι ευθύνη των παρόχων.



Σχήμα 1.11: Η Υπηρεσιοστρεφής Προσέγγιση των Πληροφοριακών Συστημάτων

Αυτή η αρχιτεκτονική προσέγγιση των Κατακεμημένων Πληροφοριακών Συστημάτων έχει πολύ σημαντικά επιχειρησιακά πλεονεκτήματα, δεδομένου ότι δίνει την δυνατότητα στην επιχείρηση ή τον οργανισμό που θα την υιοθετήσει να εστιάζει στην επίλυση των προβλημάτων της επιχείρησης, προσθέτοντας υπηρεσίες στην υπάρχουσα υποδομή, την οποία ουσιαστικά διευρύνει με έναν ευέλικτο τρόπο και μειώνοντας το ρίσκο υλοποίησης, που ενδεχόμενα να σημαίνει μεγάλα επενδυόμενα ποσά.

Από τεχνική άποψη, η αρχιτεκτονική προσέγγιση των Κατακεμημένων Πληροφοριακών Συστημάτων έχει επίσης σημαντικά πλεονεκτήματα, όπως αυτά που σχετίζονται με την αυτοματοποίηση και την επικοινωνία. Η χαλαρή ολοκλήρωση των υπηρεσιών πέρα του ότι αυξάνει την ευελιξία και την



επεκτασιμότητα, βελτιώνει και την διαχειρισιμότητά και επαναχρησιμοποίησή τους. Αυτό είναι τεράστιας αξίας στα σύγχρονα συστήματα ανάπτυξης Πληροφοριακών Συστημάτων. Επίσης, προωθούν τη διαφάνεια τοποθεσίας των υπηρεσιών, την ανεξάρτητη από πρωτόκολλα επικοινωνία, κρύβοντας έτσι την ετερογένεια, ενώ εξυπηρετείται και η παράλληλη επεξεργασία με τη σύγχρονη μεταφορά μηνυμάτων.

Εστιάζοντας στην διαχειρισιμότητα, θα πρέπει να αναφερθεί ότι αυτή δεν στοχεύει μόνο στην επεκτασιμότητα, την επαναχρησιμοποίηση και τη σύνθεση, που κατά κύριο λόγο αφορούν στη διαμόρφωση των υπηρεσιών (service configuration management), αλλά περιλαμβάνει και άλλα θέματα τα οποία είναι επίσης σημαντικά στην αποτελεσματική διαχείριση των ΚΠΣ. Ως τέτοια, είναι η επίδοση ενός ΚΣΠ (performance management), η ανοχή σε σφάλματα του ΚΣΠ (fault tolerance), η ασφάλεια του ΚΠΣ (security), και η λογιστική του ΚΠΣ (accounting management). Η χαλαρή σύνθεση και η υπηρεσιοστρεφής προσέγγιση εξυπηρετούν την εφαρμογή πολιτικών διαχείρισης του ΚΠΣ, που είναι ένα μεγάλο κεφάλαιο για τα ΚΠΣ και το οποίο θα εξετασθεί διεξοδικά στο 12<sup>ο</sup> κεφάλαιο.

## Βιβλιογραφικές αναφορές

- [1] Andrew S. Tanenbaum, Maarten Van Steen (2006), Κατανεμημένα Συστήματα: Αρχές και Υποδείγματα, Έκδοση: 1η/2006, Εκδόσεις Κλειδάριθμος.
- [2] Coulouris, J. Dollimore, T. Kindberg, G. Blair (2020), Κατανεμημένα Συστήματα, Έκδοση: 2η, Da Vinci M.E.Π.Ε.
- [3] Κάβουρας Ι.Κ., Μήλης Ι.Ζ., Ρουκουνάκη Α.Α., Ξηλωμένος Γ.Β. (2011), Κατανεμημένα συστήματα σε Java, 3η έκδοση, Εκδόσεις Κλειδάριθμος ΕΠΕ.
- [4] David Culler, J.P. Singh, and Anoop Gupta. Parallel Computer Architecture: A Hardware/Software Approach. Morgan Kaufmann. ISBN-10: 1558603433, 1998.
- [5] Παπαδάκης Στ., Διαμαντάρας Κ., Προγραμματισμός και Αρχιτεκτονική Συστημάτων Παράλληλης Επεξεργασίας, Εκδόσεις Κλειδάριθμος, 2012.
- [6] Θεμιστοκλέους Μ. Μαντζάνα Β., 2010 ‘Υπηρεσίες Παγκόσμιου Ιστού και Υπηρεσιοστρεφείς Αρχιτεκτονικές’, αυτοέκδοση, Αθήνα, Ελλάδα.
- [7] Δημακόπουλος Β., 2015, ‘Παράλληλα Συστήματα και Προγραμματισμός’, [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: <https://repository.kallipos.gr/bitstream/11419/3209/4/book.pdf> .

## Κριτήρια αξιολόγησης

### Ερώτηση 1

Ποιο από τα παρακάτω εκφράζει τους στόχους των Κατανεμημένων Συστημάτων;

- α. Η σύνδεση των χρηστών και των πόρων
- β. Η διαφάνεια των κατανεμημένων υπηρεσιών
- γ. Η διαλειτουργικότητα
- δ. Όλα τα παραπάνω

### Ερώτηση 2

Ποιο από τα παρακάτω δεν είναι έννοια λογισμικού στα Κατανεμημένα Συστήματα;

- α. Κατανεμημένο Λειτουργικό Σύστημα
- β. Δικτυακό Λειτουργικό Σύστημα
- γ. Ενδιάμεσο Λογισμικό

## **δ. Stand Alone Εφαρμογή**

### **Ερώτηση 3**

Ποιο είναι βασικό μειονέκτημα των Κ.Σ. ;

- α. Ο παραλληλισμός
- β. Η κατανομή των πόρων
- γ. Η εξάρτηση από το δίκτυο**
- δ. Όλα τα παραπάνω

### **Ερώτηση 4**

Ποιο από τα παρακάτω αφορά ο διαμοιρασμός των πόρων;

- α. Εφαρμογές
- β. Αρχεία
- γ. Συσκευές
- δ. Όλα τα παραπάνω**

### **Ερώτηση 5**

Πως διαχειρίζονται την ετερογένεια τα Κατανεμημένα Συστήματα;

- α. μέσω μηνυμάτων
- β. μέσω χρήσης κοινής γλώσσας προγραμματισμού
- γ. μέσω Ενδιάμεσου Λογισμικού (Middleware)**
- δ. μέσω Πεπαλαιωμένων Συστημάτων (Legacy Systems)

### **Ερώτηση 6**

Ποιο από τα παρακάτω δεν είναι λειτουργική απαίτηση των ΚΣ;

- α. πολλαπλοί προμηθευτές**
- β. ανοχή στα σφάλματα
- γ. υπηρεσίες ονοματολογίας
- δ. διαλειτουργικότητα

### **Ερώτηση 7**

Ποιο από τα παρακάτω δεν είναι βασικό χαρακτηριστικό των ΚΣ;

- α. Διαφάνεια τοποθεσίας των υπηρεσιών
- β. Χρήση κοινόχρηστης μνήμης**
- γ. Έλλειψη καθολικής εικόνας του χρόνου
- δ. Η χαλαρή ολοκλήρωση με σκοπό την ευελιξία

### **Ερώτηση 8**

Ποιο είναι αληθές από τα παρακάτω;

- α. Ιδιωτική μνήμη των πολυεπεξεργαστών σε σχέση με διαμοιραζόμενη μνήμη των πολυπολογιστών**
- β. Πολυεπεξεργαστές και πολυπολογιστές τρέχουν πάνω από κοινή αρτηρία (bus).
- γ. Πολυεπεξεργαστές και πολυπολογιστές συνδυάζουν διαφορετικές Βάσεις Δεδομένων
- δ. Πολυεπεξεργαστές και πολυπολογιστές τρέχουν το ίδιο λειτουργικό

### **Ερώτηση 9**

Ποιο είναι αληθές από τα παρακάτω για τα δίκτυα υπολογιστών;

- α. Διασυνδέουν αρχεία

- β. Δεν αποτυγχάνουν ποτέ
- γ. Αφορούν πρωτόκολλα του επιπέδου 7 της στοίβας OSI
- δ. Διασυνδέουν υπολογιστικούς κόμβους**

#### **Ερώτηση 10**

Ποιο από τα παρακάτω αποτελεί βασικό κίνητρο για τα ΚΣ;

- α. Διασύνδεση με τις Βάσεις Δεδομένων
- β. Κατανεμημένη φύση των Εφαρμογών**
- γ. Αυτοματισμός γραφείου
- δ. Όλα τα παραπάνω

#### **Ερώτηση 11**

Ποια υπηρεσία από τις παρακάτω πρέπει να προσφέρει ένα κατανεμημένο σύστημα;

- α. Διαφανούς προσπέλασης**
- β. Αδιαφανούς προσπέλασης
- γ. Μη διατήρησης δεδομένων
- δ. Ανασφάλειας

#### **Ερώτηση 12**

Για ποιο λόγο από τους παρακάτω θα προτιμούσατε χαλαρή σύνδεση μεταξύ των κόμβων ενός ΚΣ;

- α. Διαφάνειας
- β. Διαλειτουργικότητας
- γ. Επεκτασιμότητας**
- δ. Ταχύτητας

#### **Ερώτηση 13**

Ποια είναι η διαφορά ενός κατανεμημένου λειτουργικού συστήματος και ενός δικτυακού λειτουργικού συστήματος;

- α. Το δεύτερο προσφέρει διαφάνεια υπηρεσιών ενώ το πρώτο όχι.
- β. Το πρώτο προσφέρει διαφάνεια υπηρεσιών ενώ το δεύτερο όχι.
- γ. Το δεύτερο διαχειρίζεται με διαφάνεια πόρους υλικού ενώ το πρώτο προσφέρει τοπικές υπηρεσίες σε απομακρυσμένους υπολογιστές πελάτες.
- δ. Το δεύτερο προσφέρει τοπικές υπηρεσίες σε απομακρυσμένους υπολογιστές πελάτες, ενώ το πρώτο διαχειρίζεται με διαφάνεια πόρους υλικού.**

#### **Ερώτηση 14**

Είναι απαραίτητο να υπάρχει το ίδιο λειτουργικό σύστημα σε κάθε υπολογιστή συνδεδεμένο σε ένα Κατανεμημένο Σύστημα.

- α. Σωστό
- β. Λάθος**
- γ. Ελλιπής Πληροφόρηση

#### **Ερώτηση 15**

Ποια από τα παρακάτω είναι τεχνικό πλεονέκτημα των Υπηρεσιοστρεφών Αρχιτεκτονικών (SOA);

- α. Αυτόνομες υπηρεσίες με σκοπό την επαναχρησιμοποίηση
- β. Διαφάνεια τοποθεσίας
- γ. Ασύγχρονη μεταφορά μηνυμάτων με σκοπό την παράλληλη επεξεργασία και την προσαρμοστικότητα

**δ. Όλα τα παραπάνω**