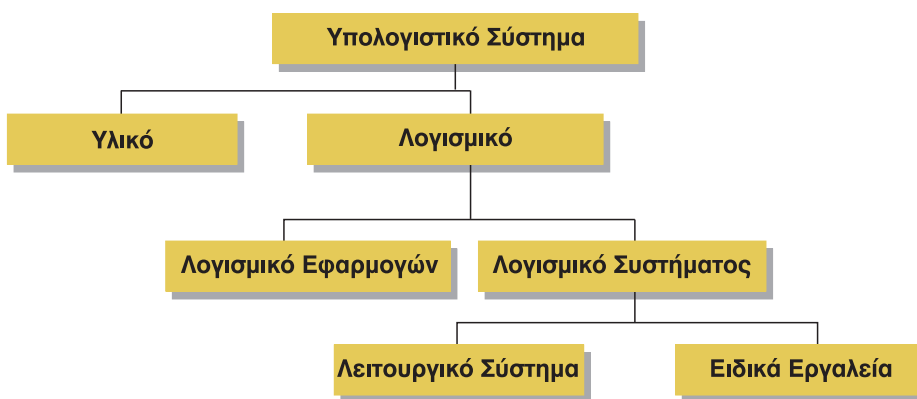


Στο προηγούμενο κεφάλαιο γνωρίσαμε το υλικό του υπολογιστή. Όμως μια άλλη βασική συνιστώσα είναι το λειτουργικό του σύστημα. Στο κεφάλαιο αυτό, λοιπόν, θα επιχειρήσουμε να κατανοήσουμε τις κυριότερες έννοιες των λειτουργικών συστημάτων, να αναλύσουμε το ρόλο και τη σημασία τους και να παρουσιάσουμε τα μέρη από τα οποία αποτελείται ένα λειτουργικό σύστημα. Τέλος, θα αναφερθούμε σε λειτουργικά συστήματα που χαρακτηρίζονται από τη μεγάλη τους διάδοση και χρήση.

4.1 Λογισμικό συστήματος

Για να λειτουργήσει ένα υπολογιστικό σύστημα, εκτός από το **υλικό** του, είναι απαραίτητο και το **λογισμικό**, το σύνολο δηλαδή των προγραμμάτων που συντονίζουν τις λειτουργίες του υλικού και αξιοποιούν τις δυνατότητές του για τη χρήση του υπολογιστή ως εργαλείου.



Το λογισμικό χωρίζεται σε δύο μεγάλες κατηγορίες:

- α) στο λογισμικό συστήματος
- β) στο λογισμικό εφαρμογών.

Στο **λογισμικό εφαρμογών** (application software), το οποίο θα εξετάσουμε σε επόμενο κεφάλαιο, συγκαταλέγονται τα διάφορα προγράμματα που έχουν αναπτυχθεί για την αντιμετώπιση εξειδικευμένων προβλημάτων των χρηστών. Όπως για παράδειγμα ο επεξεργαστής κειμένου, ειδικό λογισμικό για την έκδοση της μισθοδοσίας των υπαλλήλων μιας επιχείρησης. Η επιλογή του κατάλληλου λογισμικού εφαρμογών γίνεται από τον ίδιο το χρήστη και εξαρτάται από τις ανάγκες που θέλει να καλύψει με τη χρήση του υπολογιστή. Για να χρησιμοποιηθεί, όμως, ο υπολογιστής ως εργαλείο, είναι απαραίτητο να διαθέτει ήδη το λογισμικό συστήματος, το οποίο εξαρτάται από το είδος του συγκεκριμένου υλικού.

Στο **λογισμικό συστήματος** (system software), συγκαταλέγονται όλα τα προγράμματα που έχουν σχέση με το συντονισμό της λειτουργίας του συστήματος και με τη διαχείριση των μονάδων του. Τα προγράμματα αυτά χωρίζονται σε δύο μεγάλες κατηγορίες:

- α) τα ειδικά εργαλεία
- β) το λειτουργικό σύστημα.

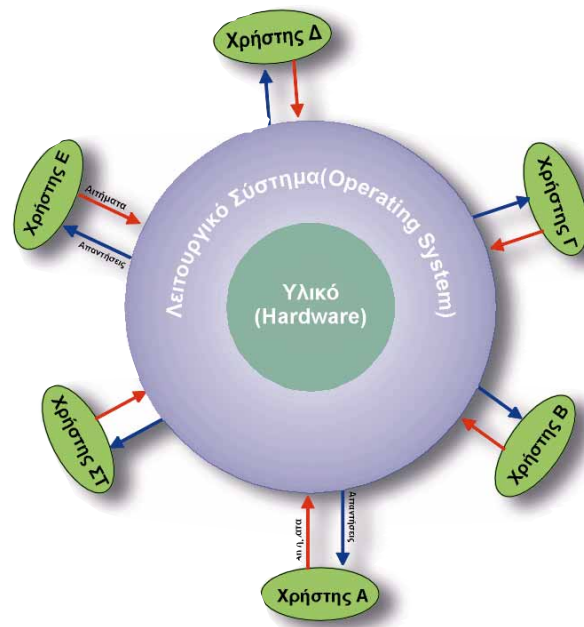
Τα **ειδικά εργαλεία** χρησιμοποιούνται για την ανάπτυξη λογισμικού εφαρμογών και την εκτέλεση βοηθητικών εργασιών.

Το **λειτουργικό σύστημα** (operating system) είναι ένα πρόγραμμα πολλών χιλιάδων εντολών που λειτουργεί ως ενδιάμεσος ανάμεσα στο χρήστη και το υλικό και ασχολείται με τον έλεγχο και το συντονισμό του υλικού, ώστε να εκτελούνται σωστά τα διάφορα προγράμματα εφαρμογών.

4.2 Λειτουργικό σύστημα

Το υλικό του υπολογιστικού συστήματος αποτελείται, όπως ήδη γνωρίζουμε, από μονάδες μνήμης, μονάδες εκτέλεσης αριθμητικών και λογικών πράξεων, μονάδες επικοινωνίας, μονάδες εκτύπωσης, μονάδες εισόδου και μονάδες εξόδου. Αυτές οι επιμέρους μονάδες του υλικού αποτελούν και τους **πόρους** (resources) -δηλαδή τον πλούτο- του υπολογιστικού συστήματος. Από την άλλη πλευρά του υπολογιστικού συστήματος βρίσκονται οι χρήστες, όπως φαίνεται στο Σχήμα 4.1, οι οποίοι διατυπώνουν αιτήματα προς το υπολογιστικό σύστημα και αναμένουν την ικανοποίησή τους. Τα αιτήματα αυτά απαιτούν τη χρήση των πόρων για να ικανοποιηθούν. Υπάρχει λοιπόν η ανάγκη ενός καλού «κυβερνήτη», που θα δέχεται τα αιτήματα του χρήστη ή των χρηστών, θα εκτιμά τη διαθεσιμότητα και τις δυνατότητες των πόρων και θα τους διαχειρίζεται με στόχο την ικανοποίηση των αιτημάτων των χρηστών. Το λειτουργικό σύστημα παίζει το ρόλο του «κυβερνήτη», ο οποίος:

- α) λαμβάνει τα αιτήματα των χρηστών
- β) τα αναλύει σε επιμέρους λειτουργίες
- γ) σχεδιάζει με ποιο τρόπο θα κάνει τη βέλτιστη χρήση του υλικού για την ικανοποίηση αυτών των αιτημάτων
- δ) παρακολουθεί την εκτέλεση των απαραίτητων λειτουργιών και
- ε) συνθέτει τις απαντήσεις και τις προωθεί στον χρήστη.



Σχήμα 4.1: Το λειτουργικό σύστημα

Το λειτουργικό σύστημα ασκεί το ρόλο του με στόχο:

- ◆ να εξασφαλίζει τη μέγιστη δυνατή ευκολία στο χρήστη, έτσι αυτός με την ελάχιστη δυνατή προσπάθεια να λαμβάνει τα επιθυμητά αποτελέσματα,
- ◆ να επιτυγχάνει τη μέγιστη δυνατή αξιοποίηση των πόρων του συστήματος.

Στην περίπτωση που οι παραπάνω στόχοι αλληλοσυγκρούονται, συνήθως υπερισχύει ο πρώτος.

4.2.1 Σταθμοί στην εξέλιξη των λειτουργικών συστημάτων

Όπως συνέβη με το υλικό, έτσι και στην εξέλιξη των λειτουργικών συστημάτων μπορούμε να διακρίνουμε γενιές. Οι πρώτοι υπολογιστές της δεκαετίας του 1940 δεν διέθεταν λειτουργικό σύστημα. Ο χρήστης έδινε εντολές στο υπολογιστικό σύστημα χρησιμοποιώντας τη γλώσσα μηχανής. Έτσι, παρά το οξύμωρο σχήμα, αναφερόμαστε στα λειτουργικά **μηδενικής γενιάς**.

Τα λειτουργικά συστήματα (Λ.Σ.) άρχισαν να αναπτύσσονται τη δεκαετία του 1950, οπότε συναντάμε τα ΛΣ **πρώτης γενιάς**, ενώ στην εποχή μας χρησιμοποιούμε ΛΣ **τέταρτης γενιάς**. Τα σημαντικά στάδια στην εξέλιξη των ΛΣ από την πρώτη μέχρι την τέταρτη γενιά είναι:

- α) Η **ομαδική επεξεργασία** (batch processing), κατά τη δεκαετία του 1950. Ο χρήστης δίνει στον υπολογιστή μια ομάδα αιτημάτων και ο υπολογιστής τα επεξεργάζεται και τα εκτελεί το ένα μετά το άλλο. Για να διατυπώσει ο χρήστης αναλυτικότερες απαιτήσεις, χρησιμοποιεί μια ειδική γλώσσα, τη **γλώσσα ελέγχου εργασιών** (job control language).
- β) Ο **πολυπρογραμματισμός** (multiprogramming), κατά τη δεκαετία του 1960. Ένας ή πολλοί χρήστες, διατυπώνουν ταυτόχρονα πολλά αιτήματα. Το Λ.Σ. επιλέγει τη σειρά εκτέλεσης, έτσι ώστε να αξιοποιείται αποδοτικά το σύστημα και διαφορετικοί χρήστες να χρησιμοποιούν διαφορετικές συσκευές του υπολογιστή «ταυτόχρονα». Ο πολυπρογραμματισμός είναι μια προσπάθεια να αυξηθεί η χρήση της ΚΜΕ, με στόχο να υπάρχει πάντα κάποια εργασία προς εκτέλεση.
- γ) Ο **καταμερισμός χρόνου** (time sharing), κατά τη δεκαετία του 1970. Στην περίπτωση που πολλοί χρήστες πρέπει να εξυπηρετηθούν ταυτόχρονα, ο υπολογιστής καταμερίζει το χρόνο ανά χρήστη -συνήθως 10 msec ανά χρήστη- και έτσι ικανοποιεί όλους τους χρήστες «συγχρόνως». Τα συστήματα αυτά καλούνται συστήματα καταμερισμού χρόνου.
- δ) Η **κατανεμημένη επεξεργασία** (distributed processing), κατά τη δεκαετία του 1990. Με την ανάπτυξη συστημάτων που διαθέτουν πολλούς επεξεργαστές και την ανάπτυξη των δικτύων, προέκυψε η ανάγκη της κατανεμημένης επεξεργασίας. Ένα αίτημα ή μια ομάδα αιτημάτων αναλύεται σε ένα σύνολο **επεξεργασιών** (process), εκ των οποίων κάθε επιμέρους επεξεργασία είναι δυνατόν να εκτελείται σε διαφορετικούς υπολογιστές ή και σε διαφορετικούς επεξεργαστές του ίδιου υπολογιστή που επικοινωνούν και συνεργάζονται μεταξύ τους.

- ΛΣ ομαδικής επεξεργασίας
- ΛΣ πολυπρογραμματισμού
- ΛΣ καταμερισμού χρόνου
- ΛΣ κατανεμημένης επεξεργασίας

4.2.2 Εργασίες λειτουργικών συστημάτων

Το λειτουργικό σύστημα εξασφαλίζει ένα περιβάλλον για την εκτέλεση προγραμμάτων. Στο περιβάλλον αυτό μπορούμε να διακρίνουμε τις παρακάτω ομάδες εργασιών που επιτελεί το λειτουργικό σύστημα:

- α) **Εκτέλεση προγραμμάτων.** Ο χρήστης «διατυπώνει» στο λειτουργικό σύστημα την επιθυμία του να εκτελεστεί ένα συγκεκριμένο πρόγραμμα. Στη συνέχεια το λειτουργικό σύστημα αναλαμβάνει όλες τις απαραίτητες εργασίες, ώστε το πρόγραμμα από την περιφερειακή μνήμη να φορτωθεί στην κεντρική μνήμη, να εκτελεστεί κάθε εντολή από την ΚΜΕ και να φθάσουν τα αποτελέσματα στο χρήστη.
- β) **Λειτουργίες εισόδου/εξόδου.** Ένα πρόγραμμα που εκτελείται είναι δυνατόν να προβάλλει απαίτηση επικοινωνίας με κάποιες περιφερειακές συσκευές εισόδου ή και εξόδου. Οι αναγκαίες ενέργειες για την υλοποίηση αυτής της επικοινωνίας γίνονται από το λειτουργικό σύστημα, π.χ. επικοινωνία με ποντίκι, εύρεση της αρχής ενός αρχείου σε μια συσκευή μαγνητικής ταινίας, απεικόνιση γραφικών στην οθόνη, κλπ.
- γ) **Διαχείριση του συστήματος αρχείων.** Μεγάλης σημασίας εργασίες, για τις οποίες υπεύθυνο είναι το λειτουργικό σύστημα, είναι αυτές που αφορούν ζητήματα διαχείρισης αρχείων. Τέτοιες λειτουργίες είναι η δημιουργία, η διαγραφή, η τροποποίηση, η μετονομασία, η μεταφορά ενός συγκεκριμένου αρχείου, κλπ.
- δ) **Ανίχνευση λαθών.** Το λειτουργικό σύστημα πρέπει να εντοπίζει σε κάθε λάθος πιθανές αιτίες και να μεταφέρει στο χρήστη το σωστό μήνυμα.
- ε) **Κατανομή πόρων.** Το λειτουργικό σύστημα θα πρέπει να αξιολογεί κάθε χρονική στιγμή τα διατυπωμένα από τους χρήστες αιτήματα και να κατανέμει τους διαθέσιμους πόρους κατάλληλα.
- στ) **Απολογιστικά στοιχεία.** Το λειτουργικό σύστημα θα πρέπει κάθε φορά να διατηρεί στοιχεία από τα οποία να προκύπτει η χρήση των πόρων ανά χρήστη και η εξαγωγή χρήσιμων στατιστικών για τη βελτίωση των επιδόσεων του συστήματος.
- ζ) **Προστασία - Ασφάλεια.** Το λειτουργικό σύστημα θα πρέπει να προσφέρει στο χρήστη προστασία, ώστε να είναι αδύνατο άλλος χρήστης ή το ίδιο το λειτουργικό (σε περίπτωση βλάβης) να αλλοιώσει τα προσωπικά του δεδομένα ή προγράμματα.

4.3 Μέρη λειτουργικού συστήματος

Το ΛΣ είναι ένα πρόγραμμα γραμμένο συχνά σε διάφορες γλώσσες προγραμματισμού. Πάντα όμως ένα μέρος του (συνήθως μικρό) είναι γραμμένο στη γλώσσα μηχανής που κατανοεί η ΚΜΕ του υπολογιστή στον οποίο θα εγκατασταθεί το λειτουργικό αυτό σύστημα.

Πυρήνας

Εάν το λειτουργικό σύστημα είναι ένας ενδιάμεσος μεταξύ του χρήστη και του υλικού, ο **πυρήνας** (Kernel) είναι το μέρος εκείνο που βρίσκεται πλησιέστερα στο υλικό. Επιτελεί δηλαδή τον ρόλο του ενδιάμεσου «κρίκου» μεταξύ του

υλικού και του υπόλοιπου λειτουργικού συστήματος. Επειδή εξαρτάται από το υλικό, ένα μέρος του έχει γραφεί στη γλώσσα μηχανής που κατανοεί η ΚΜΕ του υπολογιστή. Είναι το πρόγραμμα που φορτώνεται πρώτο, όταν ξεκινάει η λειτουργία του υπολογιστικού συστήματος. Επίσης, με ειδικούς μηχανισμούς (σηματοφορείς, ουρές μηνυμάτων) -που δεν είναι σκόπιμο να αναλυθούν στο παρόν κεφάλαιο- ρυθμίζει την επικοινωνία και το συγχρονισμό μεταξύ των διεργασιών, τις οποίες και διανέμει στον ή στους κατάλληλους επεξεργαστές. Άλλωστε ο πυρήνας ονομάζεται και «ελεγκτής της κυκλοφορίας».

Για παράδειγμα, αν δύο διεργασίες πρέπει να χρησιμοποιήσουν τον ίδιο πόρο -π.χ. να διαβάσουν ένα κοινό αρχείο- θα πρέπει να συντονίσουν την εκτέλεσή τους για να αποτραπεί εμπλοκή του συστήματος. Η διαχείριση και επικοινωνία των διεργασιών είναι απαραίτητη, ώστε όλα τα προγράμματα να εκτελούνται και να τρέχουν «δίκαια». Είναι ανεπιθύμητο το πρόγραμμα ενός χρήστη να εκτελείται με τρόπο που να παραμελούνται τα προγράμματα των άλλων χρηστών, οι οποίοι θα πρέπει να περιμένουν. Αντίθετα, κάποια προγράμματα -όπως π.χ. ο έλεγχος επικοινωνιών δικτύου- απαιτείται να εκτελούνται συχνότερα ή κατά προτεραιότητα από άλλα. Τα ζητήματα αυτά τα αντιμετωπίζει ο πυρήνας του λειτουργικού.

Στο σημείο αυτό αξίζει να αναφερθεί ότι η **διεργασία** (task) ενός συστήματος και η **επεξεργασία** (process) είναι έννοιες παρόμοιες. Και οι δύο έννοιες υποδηλώνουν μια ακολουθία από ενέργειες, που πρέπει να εκτελεστούν προκειμένου να ολοκληρωθεί μια λειτουργία. Όμως η διεργασία είναι ευρύτερη, καθώς αφορά ενέργειες του συστήματος που εμπλέκουν στοιχεία όπως περιβάλλον εκτέλεσης, επικοινωνία και άλλες συνθετότερες παραμέτρους, ενώ με την επεξεργασία αναφερόμαστε κυρίως σε τμήματα προγράμματος. Μια διεργασία μπορεί να είναι έτοιμη για εκτέλεση, να βρίσκεται σε φάση εκτέλεσης ή να έχει παρακωλυθεί η εκτέλεσή της.

Κατά την εκτέλεση των διαφόρων επεξεργασιών το λειτουργικό σύστημα θα πρέπει να μεριμνά συνεχώς ώστε να μην κινδυνεύει μια επεξεργασία να αλλοιώσει τη λειτουργία ή τα δεδομένα μιας άλλης. Γι' αυτό το σκοπό ένα μέρος του λειτουργικού συστήματος ασχολείται με την προστασία και ασφάλεια των επεξεργασιών. Πρόκειται για ένα μηχανισμό ελέγχου της πρόσβασης προγραμμάτων, επεξεργασιών και χρηστών στους πόρους του συστήματος. Για την υλοποίηση αυτού του μηχανισμού ελέγχου υπάρχουν πολλές πολιτικές. Κάθε λειτουργικό σύστημα υιοθετεί μια συγκεκριμένη πολιτική, την οποία υλοποιούν οι μηχανισμοί προστασίας - ασφάλειας που διαθέτει.

Το Λειτουργικό Σύστημα ανάλογα με το ρόλο που επιτελεί κάθε τμήμα του διακρίνεται στα παρακάτω μέρη.

4.3.1 Διαχείριση ΚΜΕ

Ένα σημαντικό χαρακτηριστικό των σύγχρονων λειτουργικών συστημάτων είναι η έννοια του **πολυπρογραμματισμού**. Ο πολυπρογραμματισμός συνίσταται στην ικανότητα του λειτουργικού συστήματος, την ίδια χρονική στιγμή που υπάρχουν πολλά προγράμματα «φορτωμένα» στην κεντρική μνήμη, να καθοδηγεί την ΚΜΕ να τα εκτελέσει όλα μαζί. Βέβαια, προβάλλει εύλογα το ερώτημα πώς είναι δυνατόν μια ΚΜΕ να εκτελέσει πολλά προγράμματα ταυτόχρονα;



**Προστασία -
Ασφάλεια**

Η ιδέα είναι απλή: όταν η ΚΜΕ εκτελεί ένα πρόγραμμα, υπάρχουν χρονικά διαστήματα κατά τα οποία θα πρέπει να μένει ανενεργή, αναμένοντας την ολοκλήρωση κάποιας επεξεργασίας που δεν απαιτεί δραστηριότητα από την ίδια -π.χ. είναι δυνατόν να αναμένει επικοινωνία με μια περιφερειακή συσκευή ή μέσω της μονάδας εισόδου/εξόδου, όπως είναι η μονάδα του σκληρού δίσκου. Κατά τη διάρκεια, λοιπόν, αυτών των διαστημάτων, που για τους χρονικούς ρυθμούς της ΚΜΕ είναι «δεκαετίες», είναι δυνατή η αξιοποίησή της στην εκτέλεση κάποιου άλλου προγράμματος που βρίσκεται στην κεντρική μνήμη και απαιτεί για την εκτέλεσή του την τρέχουσα χρονική στιγμή τη δραστηριοποίηση της ΚΜΕ.

Η υλοποίηση αυτής της ιδέας είναι ο πολυπρογραμματισμός, με τον οποίο επιτεύχθηκαν σημαντικά οφέλη όπως:

- α) μεγαλύτερη αξιοποίηση των δυνατοτήτων της ΚΜΕ και
- β) περισσότερες εργασίες στη μονάδα του χρόνου.

Αποσαφηνίζοντας το δεύτερο επίτευγμα, θα σκεφτούμε, βέβαια, ότι σε κάθε χρονική στιγμή η ΚΜΕ εκτελεί ένα μόνο πρόγραμμα, αλλά στη μονάδα του χρόνου -π.χ. σε 1 sec- η ΚΜΕ εκτελεί συνολικά πολλά προγράμματα.

Η υλοποίηση της έννοιας του πολυπρογραμματισμού επιτελείται από το ΛΣ μέσω της **διαχείρισης της ΚΜΕ** ή του χρονοπρογραμματισμού της. Η εκτέλεση ενός προγράμματος περιλαμβάνει συνήθως πολλές επιμέρους επεξεργασίες, που συνιστούν το σύνολο των επεξεργασιών του. Άρα, όταν έχουμε, για παράδειγμα, δύο προγράμματα «φορτωμένα» στην κεντρική μνήμη, έχουμε δύο σύνολα από επεξεργασίες.

Σύμφωνα με την ιδέα του πολυπρογραμματισμού, θα ήταν ουσιαστικό για λόγους οικονομίας χρόνου και πόρων να μην εκτελεστούν τα δύο προγράμματα ακολουθιακά, δηλαδή πρώτα οι επεξεργασίες του ενός και έπειτα του άλλου, αλλά να επιλεγεί για εκτέλεση μια σειρά επεξεργασιών και από τα δύο προγράμματα, ανάλογα με τα ιδιαίτερα χαρακτηριστικά καθεμιάς (π.χ. ανάλογα με το βαθμό σημαντικότητας). Έτσι, μέσω του χρονοπρογραμματισμού της ΚΜΕ, το ΛΣ με την εφαρμογή κατάλληλων αλγόριθμων επιτρέπει στην ΚΜΕ να αντιληφθεί ποια επεξεργασία τίνος προγράμματος θα εκτελέσει αυτή σε κάθε χρονική στιγμή, με σκοπό να επιτευχθούν τα θετικά αποτελέσματα του πολυπρογραμματισμού (Σχήμα 4.2). Η περιγραφή αυτών των αλγόριθμων δεν εμπίπτει στους στόχους αυτού του βιβλίου.

Στο σημείο αυτό θα ήταν σκόπιμο να αναφερθούμε στην έννοια της ουράς αναμονής. Υπάρχει συχνά το ενδεχόμενο περισσότερες από μια επεξεργασίες να ζητούν ταυτόχρονα την υλοποίησή τους από την ΚΜΕ. Για παράδειγμα, μπορεί σε ένα σύστημα πολλών χρηστών περισσότεροι από ένας χρήστες να επιθυμούν ταυτόχρονα την εκτύπωση κάποιου κειμένου. Όμως, επειδή η ΚΜΕ δεν μπορεί να εκτελέσει τις επεξεργασίες αυτές τη συγκεκριμένη χρονική στιγμή επειδή ήδη εξυπηρετείται μια αντίστοιχη εργασία κάποιου άλλου χρήστη, αυτές τοποθετούνται στην ουρά αναμονής του υπολογιστή. Όταν, μετά την πάροδο κάποιου χρονικού διαστήματος, η ΚΜΕ ολοκληρώσει τις προηγούμενες επεξεργασίες, τότε επιλέγονται από την ουρά αναμονής (με βάση ειδικά κριτήρια, όπως χρόνο παραμονής σε αυτήν ή προτεραιότητες) οι επόμενες επεξεργασίες για υλοποίηση.

Εστω $A_1...A_4$ επεξεργασίες προγράμματος A και $B_1,...B_5$ επεξεργασίες προγράμματος B και ότι τα δύο προγράμματα πρέπει να εκτελεστούν από την ΚΜΕ.

Χρόνος	
	0 1 2 3 4 5 6 7 8 9
Ακολουθιακή Εκτέλεση	$A_1 \Rightarrow A_2 \Rightarrow A_3 \Rightarrow A_4 \Rightarrow B_1 \Rightarrow B_2 \Rightarrow B_3 \Rightarrow B_4 \Rightarrow B_5$
Κατανομή χρόνου (time sharing)	$A_1 \Rightarrow B_1 \Rightarrow A_2 \Rightarrow B_2 \Rightarrow A_3 \Rightarrow B_3 \Rightarrow A_4 \Rightarrow B_4 \Rightarrow B_5$
Πολυπρογραμματισμός όπου η σειρά εκτέλεσης επιλέχθηκε με κατάλληλους αλγόριθμους	$A_1 \Rightarrow A_2 \Rightarrow B_1 \Rightarrow A_3 \Rightarrow B_2 \Rightarrow B_3 \Rightarrow A_4 \Rightarrow B_4 \Rightarrow B_5$

Ακολουθιακή εκτέλεση	A	A ₁	A ₂	A ₃	A ₄					
	B					B ₁	B ₂	B ₃	B ₄	B ₅
Κατανομή χρόνου	A	A ₁		A ₂		A ₃		A ₄		
	B		B ₁		B ₂		B ₃		B ₄	B ₅
Πολυπρογραμματισμός	A	A ₁	A ₂		A ₃			A ₄		
	B			B ₁		B ₂	B ₃		B ₄	B ₅
		1	2	3	4	5	6	7	8	9

Σχήμα 4.2: Δύο σχηματικές εκδοχές της χρονικής εξέλιξης της εκτέλεσης δύο προγραμμάτων

4.3.2 Διαχείριση κεντρικής μνήμης

Αποτέλεσμα του χρονοπρογραμματισμού της ΚΜΕ είναι η αύξηση της ταχύτητας και των επιδόσεων των υπολογιστών. Όμως ο **χρονοπρογραμματισμός** της ΚΜΕ προϋποθέτει ότι στην κεντρική μνήμη υπάρχουν ταυτόχρονα φορτωμένα περισσότερα του ενός προγράμματα. Επομένως, η μνήμη θα πρέπει να μοιράζεται μεταξύ διαφορετικών προγραμμάτων. Για το σκοπό αυτό ένα μέρος του λειτουργικού συστήματος ασχολείται με το μοίρασμα και την αποτελεσματική διαχείριση της κεντρικής μνήμης.

Η **διαχείριση μνήμης** (memory management) θα πρέπει να γίνεται με τρόπο που να εξυπηρετούνται οι συνεχώς τροποποιούμενες ανάγκες των επεξεργα-

σιών που εκτελούνται για κάθε πρόγραμμα. Για το λόγο αυτό η μνήμη διαμοιράζεται σε τμήματα ή σελίδες, το μέγεθος των οποίων είναι δυνατόν να ορίζεται στατικά, δηλαδή σε προκαθορισμένα σταθερά τμήματα, ή δυναμικά, δηλαδή σε τμήματα ανάλογα με τις ανάγκες της κάθε στιγμής. Με το σύστημα διαχείρισης μνήμης το ΛΣ επιτυγχάνει την υποστήριξη του πολυπρογραμματισμού και γι' αυτό το λόγο θεωρείται σημαντικό μέρος του. Κάθε λειτουργικό σύστημα διαθέτει τους δικούς του αλγόριθμους διαχείρισης μνήμης.

Ανταλλαγή Πολλές φορές ο αριθμός των **διεργασιών** που εξυπηρετούνται είναι μεγάλος και δεν επαρκεί η κεντρική μνήμη για τη φιλοξενία τους. Τότε από το λειτουργικό σύστημα μπορεί να κρατείται τμήμα της δευτερεύουσας μνήμης, όπου να αποθηκεύονται προσωρινά τα προγράμματα και τα δεδομένα κάποιων διεργασιών. Αυτά φορτώνονται στην κεντρική μνήμη όταν φθάσει η σειρά της αντίστοιχης διεργασίας για εκτέλεση, ενώ κάποια άλλα μεταφέρονται από την κεντρική μνήμη στο δίσκο. Η διαδικασία αυτή ονομάζεται **ανταλλαγή** (swapping). Δημιουργεί σοβαρές καθυστερήσεις από την εμπλοκή στην όλη διαδικασία της δευτερεύουσας μνήμης, η οποία είναι πολύ πιο αργή από την κεντρική μνήμη.

Εικονική μνήμη Όπως γνωρίζουμε η κεντρική μνήμη είναι οργανωμένη σε λέξεις -δηλαδή σε ομάδες byte- που καθεμιά έχει τη δική της διεύθυνση.

Το σύνολο αυτών των διευθύνσεων (για N λέξεις, οι δυνατές διευθύνσεις θα είναι $0, 1, 2, \dots, N-1$) ονομάζεται **χώρος φυσικών διευθύνσεων** (physical address space) ή **χώρος απόλυτων διευθύνσεων**.

Τα διάφορα προγράμματα χρησιμοποιούν ελεύθερα διευθύνσεις για να αναφέρονται στα δεδομένα τους. Για να μην υπάρχει αναφορά στην ίδια διεύθυνση από δυο διαφορετικά προγράμματα όταν εκτελούνται, χρησιμοποιείται η τεχνική των **εικονικών διευθύνσεων** -virtual addresses- (ο χώρος αυτός καλείται **εικονική μνήμη**). Με την τεχνική αυτή ένα πρόγραμμα μεταφράζεται σε γλώσσα μηχανής θεωρώντας ότι έχει στη διάθεσή του ολόκληρη τη μνήμη. Δηλαδή χρησιμοποιεί εικονικές διευθύνσεις. Όταν φορτώνεται στην μνήμη για να εκτελεστεί, το ΛΣ αντιστοιχίζει τις εικονικές αυτές διευθύνσεις με πραγματικές, που εκείνη την χρονική στιγμή είναι ελεύθερες, ενώ κατά την εκτέλεση γίνεται μια συνεχής αντιστοίχιση των εικονικών διευθύνσεων με φυσικές. Αυτό γίνεται με πολλές μεθόδους, από το τμήμα διαχείρισης της εικονικής μνήμης (π.χ. με σελιδοποίηση). Πολλές φορές ο **χώρος των εικονικών διευθύνσεων** είναι μεγαλύτερος του χώρου των φυσικών διευθύνσεων, οπότε λειτουργεί η λογική της χρήσης της βοηθητικής μνήμης για προσωρινή αποθήκευση και ανταλλαγή δεδομένων και προγραμμάτων.

4.3.3 Διαχείριση συστήματος αρχείων

Το μέρος εκείνο του λειτουργικού συστήματος που γίνεται περισσότερο αντιληπτό από το χρήστη είναι εκείνο που διαχειρίζεται το **σύστημα αρχείων** (file system).

Είναι γνωστό ότι τα δεδομένα και οι πληροφορίες μπορούν να αποθηκευθούν σε διάφορα μέσα αποθήκευσης καθένα από τα οποία έχει τις ιδιαιτερότητές του και το δικό του τρόπο φυσικής οργάνωσης. Το λειτουργικό σύστημα εξασφαλίζει έναν ομοιόμορφο τρόπο για να αντιλαμβανόμαστε την αποθήκευση

της πληροφορίας, με τη χρήση της έννοιας «αρχείο». **Αρχείο** (file) είναι μια νοητή μονάδα αποθήκευσης δεδομένων και προγραμμάτων. Γενικά, το **αρχείο** είναι μια σειρά από bit, το νόημα των οποίων ορίζεται από αυτόν που το δημιουργεί, ενώ για την αποθήκευσή τους σε μια συσκευή περιφερειακής μνήμης μεριμνά το λειτουργικό σύστημα. Κάθε **αρχείο** έχει συνήθως:

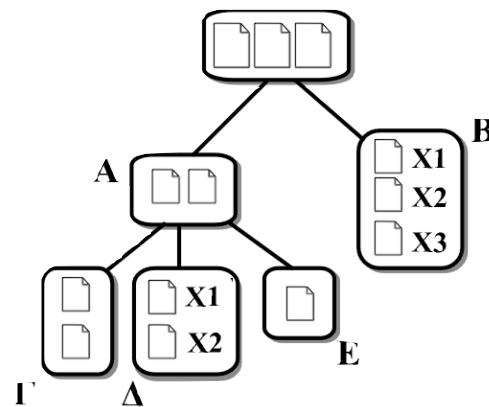
- ◆ όνομα
- ◆ τύπο
- ◆ χρόνο δημιουργίας ή/και χρόνο τροποποίησης ή/και χρόνο τελευταίας ανάγνωσης
- ◆ μέγεθος
- ◆ δημιουργό
- ◆ δικαιώματα πρόσβασης.

Το λειτουργικό σύστημα μας εξασφαλίζει τη δυνατότητα να εκτελούμε μια σειρά από πράξεις σε κάθε αρχείο. Οι πράξεις αυτές είναι οι εξής:

- α) δημιουργία αρχείου με προσδιορισμό ονόματος και τύπου
 - β) διαγραφή αρχείου
 - γ) τροποποίηση περιεχομένου αρχείου (edit)
 - δ) αντιγραφή του περιεχομένου ενός αρχείου σε μια άλλη περιφερειακή συσκευή (π.χ. εμφάνιση στην οθόνη, εκτύπωση σε εκτυπωτή)
 - ε) αντιγραφή αρχείου σε άλλο αρχείο
- στ) αλλαγή ονόματος αρχείου
- ζ) αντιγραφή ή μετακίνηση αρχείου σε άλλη περιφερειακή συσκευή
- η) αντιγραφή ή μετακίνηση αρχείου από μία ομάδα σε άλλη ομάδα αρχείων στην ίδια περιφερειακή συσκευή.

Όλα τα αρχεία που αποθηκεύονται σε μια περιφερειακή συσκευή καθώς και η οργάνωσή τους αποτελούν ένα **σύστημα αρχείων**. Πληροφορίες σχετικές με το σύστημα αρχείων καταχωρίζονται σε έναν ειδικό χώρο του μέσου αποθήκευσης και περιλαμβάνουν το όνομα κάθε αρχείου, τον τύπο του, το μέγεθός του και τις θέσεις που χρησιμοποιήθηκαν για την αποθήκευσή του πάνω στη συσκευή.

Τα αρχεία οργανώνονται σε ομάδες. Κάθε ομάδα αποτελεί ένα **κατάλογο** (directory) ή **φάκελο** (folder). Κάθε κατάλογος έχει όνομα και αποτελείται από αρχεία και ίσως κάποιους υποκαταλόγους (subdirectories) ή υποφάκελους. Το σύστημα αρχείων επομένως είναι οργανωμένο με ένα **δενδροειδή τρόπο** (tree), που φαίνεται στο σχήμα 4.3. Υπάρχει πάντα ένας αρχικός κατάλογος που ονομάζεται **ρίζικός κατάλογος** (root directory) και συνήθως συμβολίζεται με «\» ή «/», ανάλογα με το λειτουργικό σύστημα. Ο κατάλογος αυτός περιλαμβάνει αρχεία και υποκαταλόγους.



Σχήμα 4.3: Σύστημα αρχείων

Στο παράδειγμά μας οι υποκατάλογοι είναι ο κατάλογος A και ο κατάλογος B. Οι νέοι κατάλογοι μπορούν να περιέχουν και αυτοί αρχεία και υποκαταλόγους. Στο παράδειγμά μας ο κατάλογος A έχει υποκαταλόγους τους Γ, Δ, E. Με αυτόν τον τρόπο οργανώνεται κάθε σύστημα αρχείων, με τη θέση του αρ-

χείου να καθορίζεται από τη **διαδρομή** του (path).

Έτσι, για παράδειγμα, τα αρχεία του καταλόγου Δ έχουν διαδρομές:

\A\Δ\X1

\A\Δ\X2

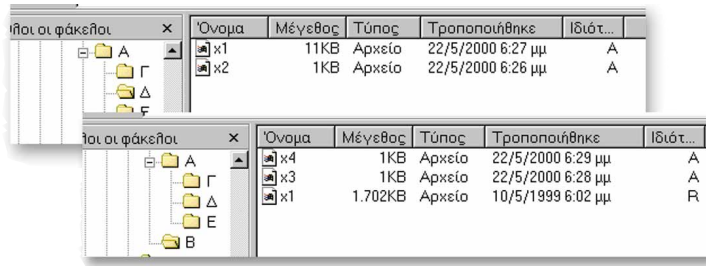
ενώ του καταλόγου Β έχουν διαδρομές:

\B\X1

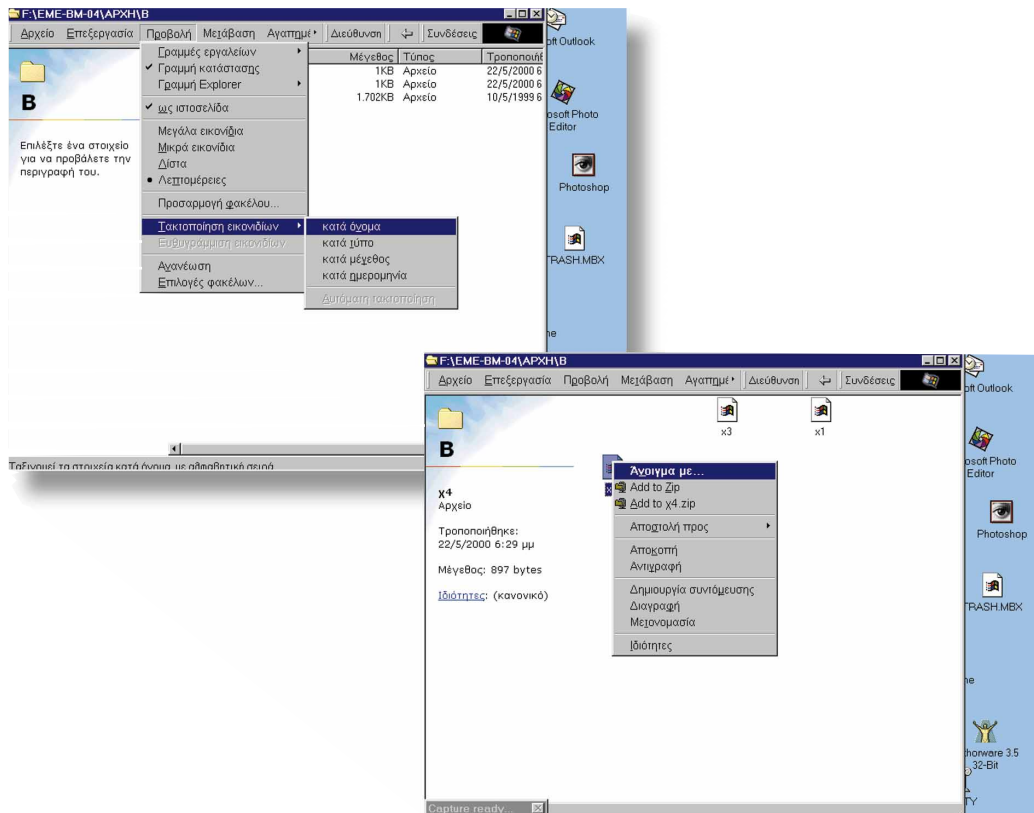
\B\X3

\B\X4.

Στον ίδιο υποκατάλογο το όνομα κάθε αρχείου είναι μοναδικό.



Όταν διαχειριζόμαστε ένα σύστημα αρχείων, κάθε χρονική στιγμή βρισκόμαστε σε έναν κατάλογο του συστήματος αρχείων. Ο κατάλογος αυτός ονομάζεται **τρέχων κατάλογος** (current directory). Το λειτουργικό σύστημα μας παρέχει την δυνατότητα να κινούμαστε πάνω στο δενδρικό σύστημα που συνθέτουν όλοι οι καταλόγοι του συστήματος αρχείων, όπως επίσης και να δημιουργούμε ή να διαγράφουμε καταλόγους.



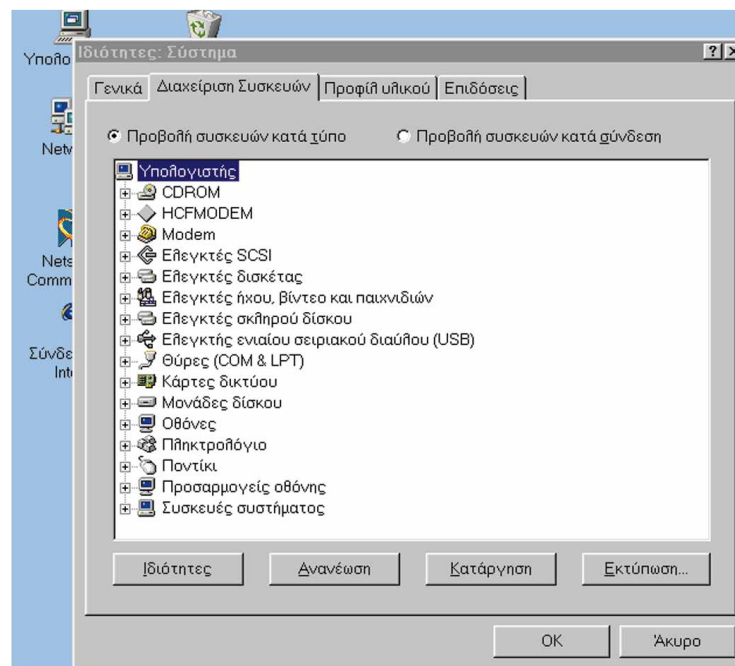
4.3.4 Διαχείριση περιφερειακών μονάδων μνήμης

Εκτός από την κεντρική μνήμη έχουμε και την περιφερειακή μνήμη, η σημασία της οποίας είναι μεγάλη. Κάθε πρόγραμμα, πριν φορτωθεί στην κεντρική μνήμη για να εκτελεσθεί, βρίσκεται στην περιφερειακή μνήμη, ενώ όταν εκτελείται θα πρέπει να διαβάσει δεδομένα από την περιφερειακή μνήμη ή να γράψει δεδομένα σε αυτήν. Βέβαια, με όσα είπαμε παραπάνω, είναι πιθανόν σε χρονικές στιγμές που απέχουν ελάχιστα μεταξύ τους να πρέπει δύο διαφορετικά προγράμματα να έχουν πρόσβαση στην ίδια συσκευή περιφερειακής μνήμης, π.χ. στο δίσκο. Υπάρχει λοιπόν η ανάγκη ενός υποσυστήματος, μέρους του λειτουργικού, που θα διαχειρίζεται το δίσκο, θα αποφασίζει ποιο αίτημα εγγραφής ή ανάγνωσης θα ικανοποιήσει, θα καθοδηγεί την περιφερειακή συσκευή για να βρει το φυσικό χώρο πάνω στο δίσκο όπου θα εγγραφεί ή θα αναζητήσει τα στοιχεία που επιθυμεί το αίτημα, θα δίνει εντολές για τη μετακίνηση των κεφαλών εγγραφής και ανάγνωσης, με δυο λόγια θα διαχειρίζεται αποδοτικά το δίσκο.

Εκτός της διαχείρισης των δίσκων, το λειτουργικό σύστημα διαθέτει διαχειριστές για κάθε περιφερειακή συσκευή. Διακρίνουμε τρεις μεθόδους για την υλοποίηση της διαχείρισης των περιφερειακών συσκευών.

- Η πρώτη ονομάζεται **μέθοδος αποκλειστικών συσκευών** (dedicated devices). Η περιφερειακή συσκευή εκτελεί μια ορισμένη επεξεργασία μέχρι αυτή να ολοκληρωθεί, π.χ. ο εκτυπωτής.
- Η δεύτερη μέθοδος ονομάζεται **μέθοδος διαμοιραζόμενων συσκευών** (shared devices). Η συσκευή μπορεί να χρησιμοποιείται ταυτόχρονα σε περισσότερες από μία επεξεργασίες, π.χ. ο δίσκος.
- Η τρίτη μέθοδος ονομάζεται **μέθοδος των εικονικών συσκευών** (virtual devices). Μια συσκευή εξομοιώνεται με μια άλλη, έτσι ώστε να μη διαταράσσεται η ομαλή λειτουργία εκτέλεσης των διαφόρων επεξεργασιών. Για παράδειγμα, όταν μια επεξεργασία επιχειρεί να εκτυπώσει ένα αποτέλεσμα και ο εκτυπωτής είναι ήδη απασχολημένος από μια άλλη επεξεργασία, τότε ένα ειδικό πρόγραμμα, που ονομάζεται **διαχειριστής παροχέτευσης** (spooler), στέλνει το αποτέλεσμα της εκτύπωσης στο δίσκο, από όπου στη συνέχεια την κατάλληλη χρονική στιγμή θα αποσταλεί στον εκτυπωτή.

Μέθοδοι υλοποίησης της διαχείρισης των περιφερειακών συσκευών



4.5 Γνωστά λειτουργικά συστήματα

Κάθε λειτουργικό σύστημα στοχεύει να καταστήσει τον υπολογιστή ένα αξιόπιστο, ισχυρό, αποτελεσματικό και εύχρηστο εργαλείο. Τα τελευταία σαράντα χρόνια που γράφεται η ιστορία των λειτουργικών συστημάτων έχουν αναπτυχθεί πολλά λειτουργικά συστήματα. Μερικά από αυτά έχουν κυριαρχήσει είτε τεχνολογικά είτε εμπορικά ή και στους δύο τομείς. Ένα λειτουργικό με λαμπρή πορεία είναι εκείνο με την ονομασία Unix. Άλλα με επίσης σημαντική πορεία είναι:

- ◆ το DOS
- ◆ τα Windows της Microsoft
- ◆ τα Windows NT της Microsoft
- ◆ το OS/2 της IBM
- ◆ το MacOS της Apple.

4.5.1 Unix

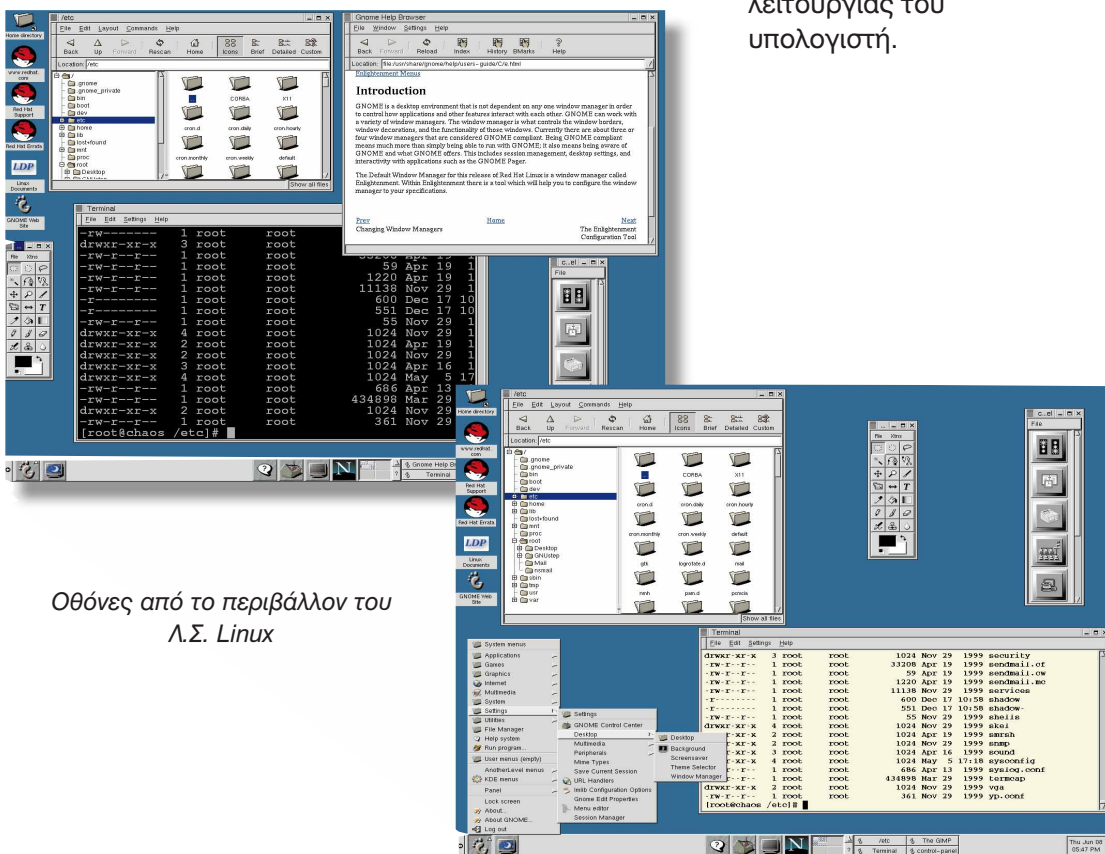
Το Unix αναπτύχθηκε στα εργαστήρια Bell Labs της AT&T το 1969. Από τότε μέχρι σήμερα συνεχώς εξελίσσεται. Διαθέτει μια σειρά από χαρακτηριστικά που του εξασφαλίζουν ξεχωριστή θέση μεταξύ των λειτουργικών συστημάτων, με κυριότερα τα εξής:

- ◆ Το Unix αναπτύχθηκε στο μεγαλύτερο μέρος του σε γλώσσα υψηλού επιπέδου (τη γλώσσα C).
- ◆ Από τεχνολογική άποψη διαθέτει σε όλα τα μέρη του προηγμένα τεχνολογικά χαρακτηριστικά, που το καθιστούν αξιόπιστο και αποδοτικό.
- ◆ Το Unix διατίθεται σε πηγαίο κώδικα και δεν ανήκει σε μια μόνο εταιρεία. Ο χρήστης της τεχνολογίας Unix δεν είναι υποχρεωμένος να προμηθευτεί υλικό (hardware) από ένα συγκεκριμένο κατασκευαστή. Υλικό διαφορετικών κατασκευαστών συνεργάζεται με το Unix.

Στο Unix διακρίνουμε δύο επίπεδα:

- α) Τον **πυρήνα** (kernel), που αλληλεπιδρά απευθείας με το υλικό του υπολογιστή. Είναι ένα πρόγραμμα από περίπου 13.000 γραμμές κώδικα γλώσσας C και 1.000 γραμμές κώδικα σε γλώσσα assembly του επεξεργαστή (ΚΜΕ) του υπολογιστή στον οποίο είναι εγκατεστημένο. Αντικείμενο του πυρήνα είναι η χρονοδρομολόγηση των διεργασιών και η διαχείριση των αρχείων, της μνήμης και του υλικού.
- β) Το **κέλυφος** (shell), το οποίο παρεμβάλλεται μεταξύ του χρήστη και του πυρήνα και ασχολείται με την αμφίδρομη επικοινωνία χρήστη-υπολογιστή. Επίσης διαθέτει μια δική του γλώσσα προγραμματισμού για τον έλεγχο της

λειτουργίας του υπολογιστή.



4.5.2 DOS

Το λειτουργικό σύστημα DOS αναπτύχθηκε από την εταιρεία Microsoft για λογαριασμό της εταιρείας IBM, στις αρχές της δεκαετίας του 1980. Από τότε, είχε μια λαμπρή πορεία διάδοσης και αποδοχής. Εκατομμύρια άνθρωποι παγκοσμίως το έχουν χρησιμοποιήσει και εξακολουθούν να το χρησιμοποιούν σε ένα βαθμό και σήμερα.

Από άποψη δομής στο DOS διακρίνουμε τρία επίπεδα:

- α) Τον **επεξεργαστή εντολών**, που αναγνωρίζει και εκτελεί τις εντολές του χρήστη.
- β) Τον **πυρήνα** (kernel), για τον χειρισμό αρχείων και περιφερειακών του υπολογιστή.
- γ) Το **BIOS**, που διαθέτει τα προγράμματα για τον χειρισμό των περιφερειακών που χρησιμοποιεί ο πυρήνας.

```

Microsoft(R) Windows 98
(C) Copyright Microsoft Corp 1981-1998.

C:\WINDOWS>dir c*.exe /p

Ο τόμος στη μονάδα δίσκου C είναι MS-DOS_6
Ο αριθμός σειράς του τόμου είναι 2443-7602
Κατάλογος C:\WINDOWS

CALC      EXE           94,208   02/09/98   8:17p   CALC.EXE
CONTROL  EXE            2,171   15/05/98   8:01p   CONTROL.EXE
CHARMAP  EXE           17,467   15/05/98   8:01p   CHARMAP.EXE
CDPLAYER EXE          106,496   02/09/98   8:17p   CDPLAYER.EXE
CDRUN$   EXE           43,520   15/05/97   1:32p   CDRUN$.EXE
CLEANMGR EXE          131,072   02/09/98   8:17p   CLEANMGR.EXE
CLSPACK  EXE           53,248   02/09/98   8:17p   CLSPACK.EXE
CVTAPLOG EXE           77,824   02/09/98   8:17p   CVTAPLOG.EXE
CVT1     EXE          118,784   02/09/98   8:17p   CVT1.EXE
CARDFILE EXE           93,184   15/05/98   8:01p   CARDFILE.EXE
CD32406  EXE          634,726   14/08/98   1:20a   cd32406.exe
11 αρχεία(ο)                1,372,700 byte
0 κατάλογοι/ος             2,270,363,648 ελεύθερα byte

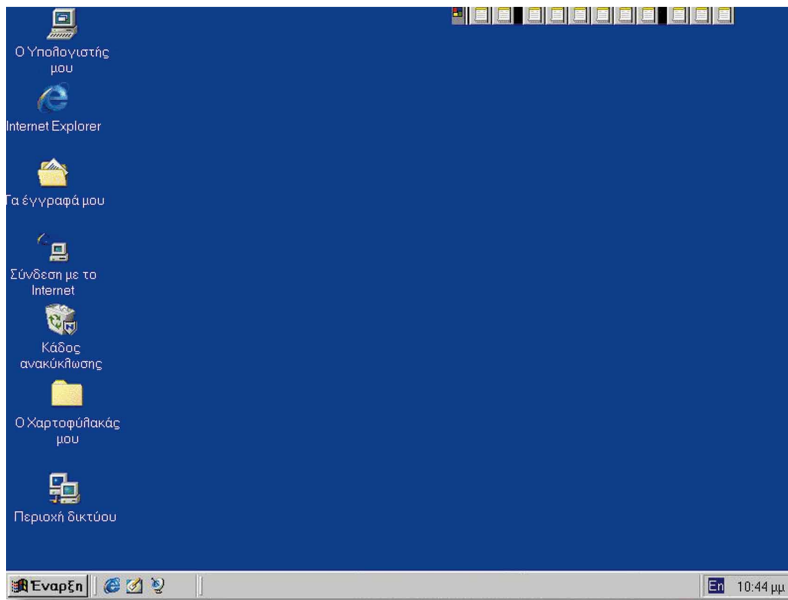
C:\WINDOWS>

```

Θόνη από το MS-DOS των Windows 98. Έχει ζητηθεί στο φάκελο windows η λίστα με αρχεία ή καταλόγους που το όνομά τους αρχίζει με C και έχουν επέκταση exe

4.5.3 Windows και Windows NT

Με την πάροδο του χρόνου αναπτύχθηκαν **γραφικά περιβάλλοντα διεπαφής (GUI-Graphical User Interface)** του χρήστη με το λειτουργικό σύστημα, με στόχο να διευκολύνουν την επικοινωνία του χρήστη με τον υπολογιστή. Επειδή το DOS είχε μεγάλο αριθμό χρηστών, αναπτύχθηκε από τη Microsoft το παραθυρικό περιβάλλον Windows (μέχρι την έκδοσή του 3.11), το οποίο εξασφαλίζει για λογαριασμό του DOS γραφικό περιβάλλον διεπαφής με το χρήστη, χωρίς όμως να αποτελεί λειτουργικό σύστημα και χωρίς να αντικαθιστά το DOS. Επίσης η ίδια εταιρεία έχει αναπτύξει άλλα λειτουργικά συστήματα με γραφικό περιβάλλον, με την ονομασία **Windows 95/98, Windows Me, Windows NT, Windows 2000**.



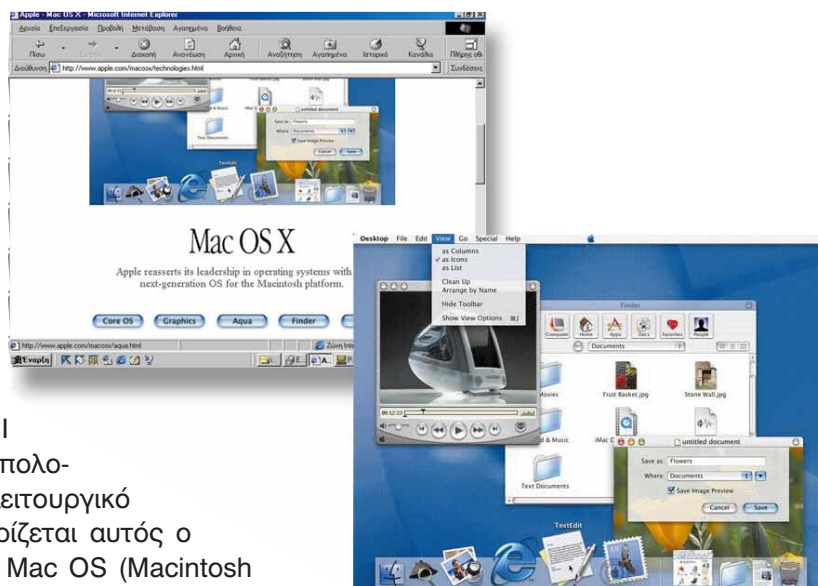
Οθόνη από την επιφάνεια εργασίας των Windows 98

4.5.4 Άλλα λειτουργικά συστήματα

Άλλα λειτουργικά συστήματα με σημαντική παρουσία στην αγορά της Πληροφορικής είναι:

- α) Το λειτουργικό **OS/2** της εταιρείας IBM.
- β) Το λειτουργικό **VMS** της εταιρείας Digital.
- γ) Το λειτουργικό **MVS** της εταιρείας IBM.
- δ) Το λειτουργικό **Mac** της

εταιρείας Apple. Το πιο σημαντικό και το πιο γνωστό μοντέλο υπολογιστή που κατασκεύασε η εταιρεία αυτή είναι το Macintosh, με βασικό του χαρακτηριστικό ότι ήταν το πρώτο που χρησιμοποιούσε γραφικό περιβάλλον χρήσης. Η διεπαφή των Windows έχει πολλά κοινά χαρακτηριστικά με το GUI που χρησιμοποιούσε ο υπολογιστής Macintosh. Το λειτουργικό σύστημα στο οποίο στηρίζεται αυτός ο υπολογιστής ονομάζεται Mac OS (Macintosh Operating System).



Οθόνες από Λ.Σ Mac της Apple



Ανακεφαλαίωση

Βασική και αυστηρά αναγκαία συνιστώσα κάθε υπολογιστικού συστήματος είναι το λειτουργικό του σύστημα.

Το ΛΣ είναι ένα μεγάλο πρόγραμμα που επικοινωνεί με το χρήστη, δέχεται τα αιτήματά του και στη συνέχεια ελέγχει και συντονίζει το υλικό του υπολογιστή, με στόχο την ικανοποίηση αυτών των αιτημάτων.

Για το σκοπό αυτό το λειτουργικό εκτελεί προγράμματα, διεκπεραιώνει λειτουργίες Εισόδου / Εξόδου, διαχειρίζεται συστήματα αρχείων, ανιχνεύει λάθη, κατανέμει τους πόρους του υλικού, διατηρεί απολογιστικά στοιχεία και μεριμνά για την ασφάλεια και προστασία των δεδομένων και προγραμμάτων κάθε χρήστη.

Το ΛΣ είναι υπεύθυνο για τη διαχείριση:

- ◆ της ΚΜΕ
- ◆ της κεντρικής μνήμης
- ◆ του συστήματος αρχείων
- ◆ των περιφερειακών μονάδων μνήμης,

ενώ διαθέτει μηχανισμούς για την προστασία και ασφάλεια των προγραμμάτων και δεδομένων.

Τα πλέον διαδεδομένα σε χρήση λειτουργικά συστήματα είναι το Unix, το DOS, τα Windows 9x, τα Windows Me, τα Windows NT, τα Windows 2000, το Mac και το OS/2.