

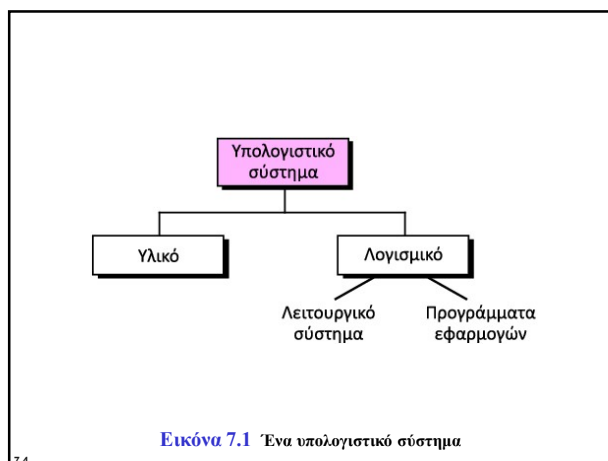


Στόχοι

Μετά την ολοκλήρωση αυτού του κεφαλαίου, ο σπουδαστής θα είναι σε θέση:

- Να κατανοεί τον ρόλο του λειτουργικού συστήματος.
- Να κατανοεί τη διαδικασία εκκίνησης για τη φόρτωση του λειτουργικού συστήματος στη μνήμη.
- Να απαριθμεί τα στοιχεία ενός λειτουργικού συστήματος.
- Να περιγράφει τον ρόλο του διαχειριστή μνήμης.
- Να περιγράφει τον ρόλο του διαχειριστή διεργασιών.
- Να περιγράφει τον ρόλο του διαχειριστή συσκευών.
- Να περιγράφει τον ρόλο του διαχειριστή αρχείων σε ένα λειτουργικό σύστημα.
- Να κατανοεί τα βασικά χαρακτηριστικά των τριών πιο συνηθισμένων λειτουργικών συστημάτων: των UNIX, Linux, και Windows NT.

Ο υπολογιστής είναι ένα σύστημα που αποτελείται από δύο βασικά στοιχεία: το υλικό (hardware) και το λογισμικό (software). Το υλικό του υπολογιστή είναι ο φυσικός (υλικός) εξοπλισμός. Το λογισμικό είναι μια συλλογή από προγράμματα που επιτρέπουν στο υλικό να λειτουργεί. Το λογισμικό των υπολογιστών διαιρείται σε δύο μεγάλες κατηγορίες: στα **λειτουργικά συστήματα** και στα **προγράμματα εφαρμογών** (Εικόνα 7.1). Τα προγράμματα εφαρμογών χρησιμοποιούν το υλικό του υπολογιστή για να λύνουν προβλήματα του χρήστη. Το λειτουργικό σύστημα, από την άλλη πλευρά, ελέγχει την πρόσβαση των χρηστών στο υλικό.



Εικόνα 7.1 Ένα υπολογιστικό σύστημα

7-1 ΕΙΣΑΓΩΓΗ

Ένα **λειτουργικό σύστημα** (operating system) έχει τέτοια πολυπλοκότητα που είναι δύσκολο να δώσουμε έναν μόνο γενικό ορισμό. Για τον λόγο αυτό παραθέτουμε μερικούς συνηθισμένους ορισμούς:

- ❑ Ένα λειτουργικό σύστημα αποτελεί τη διασύνδεση μεταξύ του υλικού ενός υπολογιστή και του χρήστη (προγράμματα ή άνθρωποι).
- ❑ Ένα λειτουργικό σύστημα είναι ένα πρόγραμμα (ή ένα σύνολο προγραμμάτων) που διευκολύνει την εκτέλεση άλλων προγραμμάτων.
- ❑ Ένα λειτουργικό σύστημα έχει τον ρόλο γενικού επιβλέποντος, συντονίζοντας τη δραστηριότητα κάθε στοιχείου στο υπολογιστικό σύστημα.

7.5



Ένα λειτουργικό σύστημα είναι μια διασύνδεση μεταξύ του υλικού ενός υπολογιστή και του χρήστη (προγράμματα ή άνθρωποι), η οποία διευκολύνει την εκτέλεση άλλων προγραμμάτων, καθώς και την πρόσβαση σε πόρους του υλικού και του λογισμικού.

Δύο μεγάλοι στόχοι του σχεδιασμού ενός λειτουργικού συστήματος είναι οι εξής:

- ❑ Η αποδοτική χρήση του υλικού.
- ❑ Η ευκολία στη χρήση των πόρων.

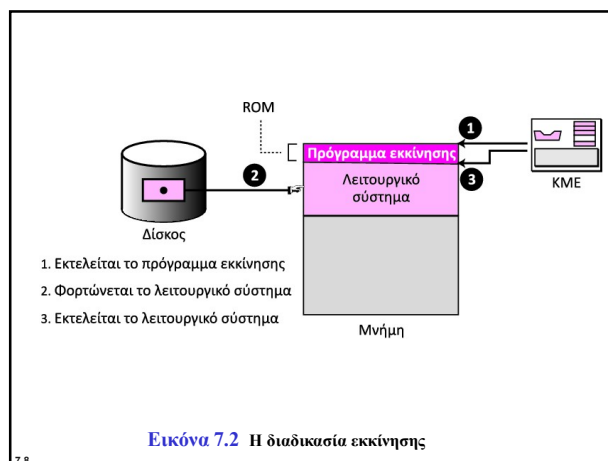
7.6

Διαδικασία εκκίνησης

Σύμφωνα με τους παραπάνω ορισμούς, το λειτουργικό σύστημα παρέχει υποστήριξη για άλλα προγράμματα. Για παράδειγμα, είναι υπεύθυνο για τη φόρτωση άλλων προγραμμάτων στη μνήμη προκειμένου να εκτελεστούν. Ωστόσο, και το ίδιο το λειτουργικό σύστημα είναι ένα πρόγραμμα που πρέπει να φορτώνεται στη μνήμη και να εκτελείται. Πώς λύνεται λοιπόν αυτό το δίλλημα;

Η λύση περιλαμβάνει μια διαδικασία με δύο στάδια. Ένα πολύ μικρό τμήμα μνήμης αποτελείται από ROM και περιέχει ένα μικρό πρόγραμμα που ονομάζεται **πρόγραμμα εκκίνησης** (bootstrap). Όταν ενεργοποιείται ο υπολογιστής, ο μετρητής της ΚΜΕ εκτελεί τις εντολές αυτού του προγράμματος εκκίνησης, ξεκινώντας από την πρώτη. Όταν ολοκληρωθεί η φόρτωση, ο μετρητής προγράμματος ξεκινά από την πρώτη εντολή του λειτουργικού συστήματος στη RAM.

7.7



Εικόνα 7.2 Η διαδικασία εκκίνησης

7.8

7-2 ΕΞΕΛΙΞΗ

Τα λειτουργικά συστήματα έχουν ένα μεγάλο ιστορικό εξέλιξης, το οποίο συνοψίζουμε εδώ.

Συστήματα δέσμης

Τα **λειτουργικά συστήματα δέσμης** (batch operating systems) σχεδιάστηκαν τη δεκαετία του 1950 για τον έλεγχο των μεγάλων υπολογιστικών συστημάτων (mainframes). Εκείνη την εποχή οι υπολογιστές ήταν μεγάλες μηχανές που χρησιμοποιούσαν διάτρητες κάρτες για είσοδο, εκτυπωτές γραμμής για έξοδο, και μονάδες ταινίας ως βοηθητικά αποθηκευτικά μέσα. Κάθε πρόγραμμα που επρόκειτο να εκτελεστεί ονομαζόταν εργασία (job). Ένας προγραμματιστής που επιθυμούσε να εκτελέσει μια εργασία έστειλε μια αίτηση στο λειτουργικό σύστημα.

7.9

Συστήματα χρονομερισμού

Για την αποδοτική χρήση των πόρων των υπολογιστικών συστημάτων δημιουργήθηκε ο **πολυπρογραμματισμός** (multiprogramming). Η ιδέα είναι η διατήρηση πολλών εργασιών στη μνήμη, και η αντιστοίχιση ενός πόρου σε μια εργασία που τον χρειάζεται μόνο υπό την προϋπόθεση ότι ο συγκεκριμένος πόρος είναι διαθέσιμος.

Ο πολυπρογραμματισμός εισήγαγε την ιδέα του **χρονομερισμού** (time sharing): δηλαδή, οι πόροι μπορούν να μοιράζονται μεταξύ διαφόρων εργασιών, και η κάθε εργασία έχει τη δυνατότητα να δεσμεύει ένα μέρος του χρόνου για να χρησιμοποιήσει έναν πόρο. Επειδή ο υπολογιστής είναι πολύ γρηγορότερος από τους ανθρώπους, ο χρονομερισμός δεν γίνεται αντιληπτός από τον χρήστη - κάθε χρήστης έχει την εντύπωση ότι όλο το σύστημα εξυπηρετεί μόνο εκείνον.

7.10

Προσωπικά συστήματα

Όταν εμφανίστηκαν οι προσωπικοί υπολογιστές, δημιουργήθηκε η ανάγκη για ένα λειτουργικό σύστημα που θα εξυπηρετούσε τον συγκεκριμένο τύπο υπολογιστή. Κατά τη διάρκεια αυτής της περιόδου εμφανίστηκαν τα λειτουργικά συστήματα ενός χρήστη (single-user operating systems), όπως το **DOS** (Disk Operating System — Λειτουργικό Σύστημα Δίσκου).

Παράλληλα συστήματα

Η ανάγκη για μεγαλύτερη ταχύτητα και αποδοτικότητα οδήγησε στη σχεδίαση των **παράλληλων συστημάτων** (parallel systems): πολλές ΚΜΕ στο ίδιο μηχάνημα. Κάθε ΚΜΕ μπορεί να χρησιμοποιηθεί για την εξυπηρέτηση ενός προγράμματος ή τμήματος προγράμματος, κάτι που σημαίνει ότι πολλές εργασίες διεκπεραιώνονται παράλληλα (ταυτόχρονα), και όχι σειριακά. Τα λειτουργικά συστήματα που απαιτούνται για τον σκοπό αυτό είναι πιο σύνθετα από τα άλλα των μηχανημάτων με μία μόνο ΚΜΕ.

7.11

Κατανεμημένα συστήματα

Όπως είδαμε στο Κεφάλαιο 6, η δικτύωση και η διαδικτύωση δημιούργησαν μια νέα διάσταση στα λειτουργικά συστήματα. Μια εργασία που προηγουμένως διεκπεραιωνόταν ολόκληρη σε έναν υπολογιστή τώρα μπορεί να μοιράζεται μεταξύ υπολογιστών οι οποίοι απέχουν χιλιάδες χιλιόμετρα μεταξύ τους. Τα κατανεμημένα συστήματα (distributed systems) συνδυάζουν λειτουργίες της προηγούμενης γενιάς με τις νέες απαιτήσεις, όπως ο έλεγχος της ασφάλειας.

Συστήματα πραγματικού χρόνου

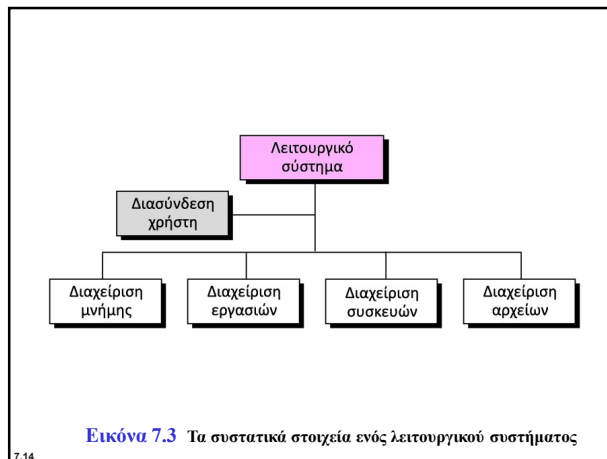
Ένα σύστημα πραγματικού χρόνου (real-time system) πρέπει να εκτελεί μια εργασία μέσα σε ένα συγκεκριμένο χρονικό πλαίσιο. Τα συστήματα αυτά χρησιμοποιούνται με εφαρμογές πραγματικού χρόνου, οι οποίες παρακολουθούν, αποκρίνονται, ή ελέγχουν εξωτερικές διεργασίες ή περιβάλλοντα.

7.12

7-3 ΣΥΣΤΑΤΙΚΑ ΣΤΟΙΧΕΙΑ

Τα σύγχρονα λειτουργικά συστήματα είναι πολύ σύνθετα. Ένα λειτουργικό σύστημα πρέπει να μπορεί να χειρίζεται πόρους διαφόρων ειδών σε ένα υπολογιστικό σύστημα. Μοιάζει με μια εταιρεία η οποία διαθέτει πολλούς διευθυντές σε ανώτατο επίπεδο. Κάθε διευθυντής είναι υπεύθυνος για το τμήμα του, αλλά πρέπει επίσης και να συνεργάζεται με τους υπόλοιπους διευθυντές και να συντονίζει δραστηριότητες. Ένα σύγχρονο λειτουργικό σύστημα έχει τουλάχιστον τέσσερα καθήκοντα: τη **διαχείριση μνήμης**, τη **διαχείριση διεργασιών**, τη **διαχείριση συσκευών**, και τη **διαχείριση αρχείων**.

7.13



Εικόνα 7.3 Τα συστατικά στοιχεία ενός λειτουργικού συστήματος

7.14

Διασύνδεση χρήστη

Κάθε λειτουργικό σύστημα έχει μια **διασύνδεση χρήστη**, η οποία είναι ένα πρόγραμμα που δέχεται αιτήσεις από τους χρήστες (διεργασίες) και τις διερμηνεύει για λογαριασμό του υπόλοιπου λειτουργικού συστήματος. Η διασύνδεση χρήστη σε ορισμένα λειτουργικά συστήματα, όπως το UNIX, ονομάζεται **κέλυφος** (shell). Σε άλλα λειτουργικά ονομάζεται **σύστημα παραθύρων** (windowing system), στο οποίο χρησιμοποιούνται μενού και υπάρχει ένα συστατικό στοιχείο διασύνδεσης γραφικών με το χρήστη (graphical user interface - GUI).

7.15

Διαχείριση μνήμης

Ένα από τα καθήκοντα των σύγχρονων υπολογιστικών συστημάτων είναι η **διαχείριση μνήμης**. Αν και το μέγεθος της μνήμης των υπολογιστών αυξήθηκε εντυπωσιακά τα τελευταία χρόνια, το ίδιο έχει συμβεί και με το μέγεθος των προγραμμάτων και των δεδομένων προς επεξεργασία. Η διαχείριση της κατανομής της μνήμης είναι απαραίτητη ώστε να αποτρέπεται η ανεπάρκεια μνήμης στις εφαρμογές. Τα λειτουργικά συστήματα χωρίζονται σε δύο μεγάλες κατηγορίες όσον αφορά τη διαχείριση μνήμης: τα **μονοπρογραμματικά** και τα **πολυπρογραμματικά**.

7.16

Μονοπρογραμματισμός

Στον **μονοπρογραμματισμό**, το μεγαλύτερο μέρος της μνήμης παραχωρείται σε ένα και μόνο πρόγραμμα: για το λειτουργικό σύστημα χρειάζεται μόνο ένα μικρό μέρος μνήμης. Με αυτή τη διεύθυνση, ολόκληρο το πρόγραμμα βρίσκεται στη μνήμη για εκτέλεση. Όταν ολοκληρωθεί η εκτέλεση του προγράμματος, η περιοχή μνήμης του προγράμματος καταλαμβάνεται από κάποιο άλλο πρόγραμμα.



Εικόνα 7.4 Μονοπρογραμματισμός

7.17

Πολυπρογραμματισμός

Στον **πολυπρογραμματισμό** υπάρχουν στη μνήμη πολλά προγράμματα ταυτόχρονα, τα οποία εκτελούνται παράλληλα, με την ΚΜΕ να περνά γρήγορα από το ένα στο άλλο.



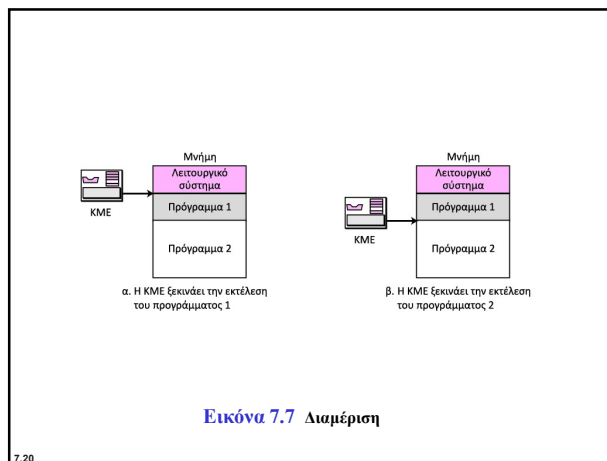
Εικόνα 7.5 Πολυπρογραμματισμός

7.18



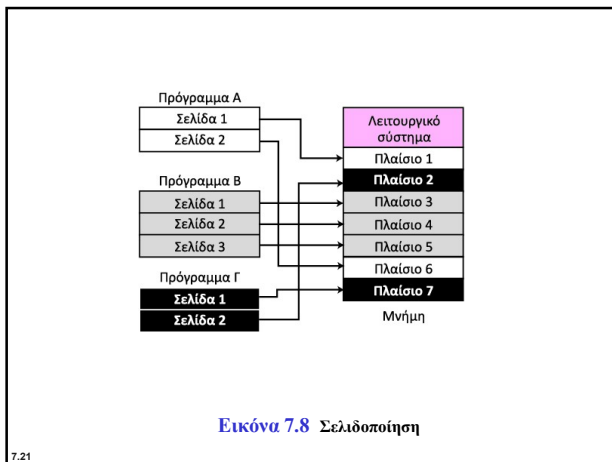
Εικόνα 7.6 Κατηγορίες πολυπρογραμματισμού

7.19

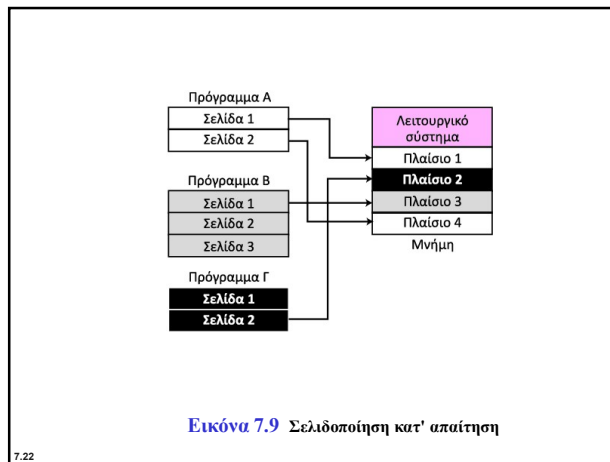


Εικόνα 7.7 Διαμέριση

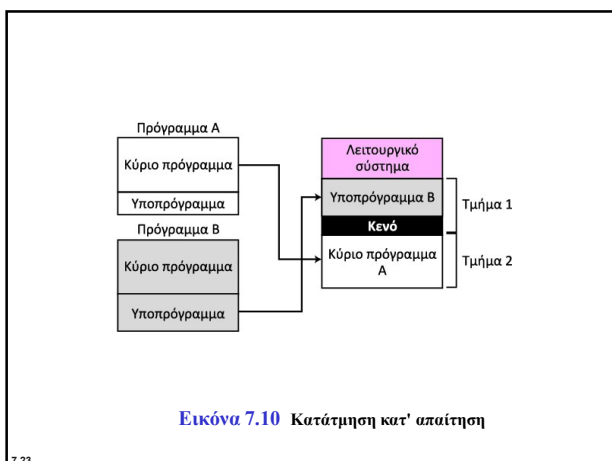
7.20



Εικόνα 7.8 Σελιδοποίηση



Εικόνα 7.9 Σελιδοποίηση κατ' απαίτηση

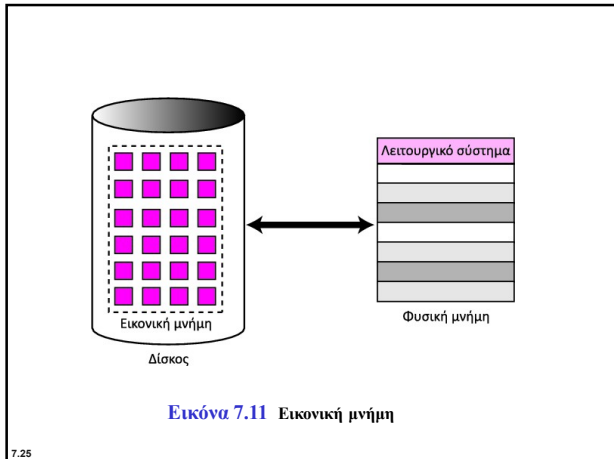


Εικόνα 7.10 Κατάτμηση κατ' απαίτηση

Εικονική μνήμη

Στη σελιδοποίηση κατ' απαίτηση και στην κατάτμηση κατ' απαίτηση ένα μέρος του προγράμματος που εκτελείται βρίσκεται στη μνήμη και ένα άλλο μέρος βρίσκεται στον δίσκο. Φανταστείτε, για παράδειγμα, μια μνήμη μεγέθους 10 MB στην οποία πρέπει να εκτελεστούν 10 προγράμματα καθένα από τα οποία είναι 3 MB — δηλαδή έχουμε ένα σύνολο 30 MB. Οποιαδήποτε στιγμή, 10 MB από τα 10 προγράμματα βρίσκονται στη μνήμη και 20 MB βρίσκονται στον δίσκο. Επομένως, υπάρχει μια πραγματική μνήμη μεγέθους 10 MB, και μια εικονική μνήμη (virtual memory) μεγέθους 30 MB. Η έννοια αυτή παρουσιάζεται στην Εικόνα 7.11. Η **εικονική μνήμη**, η οποία συνεπάγεται χρήση σελιδοποίησης κατ' απαίτηση, κατάτμησης κατ' απαίτηση, ή και των δύο τεχνικών, χρησιμοποιείται σχεδόν από όλα τα σύγχρονα λειτουργικά συστήματα.

7.24



7.25

Διαχείριση διεργασιών

Μια άλλη λειτουργία των λειτουργικών συστημάτων είναι η διαχείριση διεργασιών, όμως πριν συζητήσουμε αυτό το θέμα θα ορίσουμε κάποιους όρους.

Πρόγραμμα, εργασία, και διεργασία

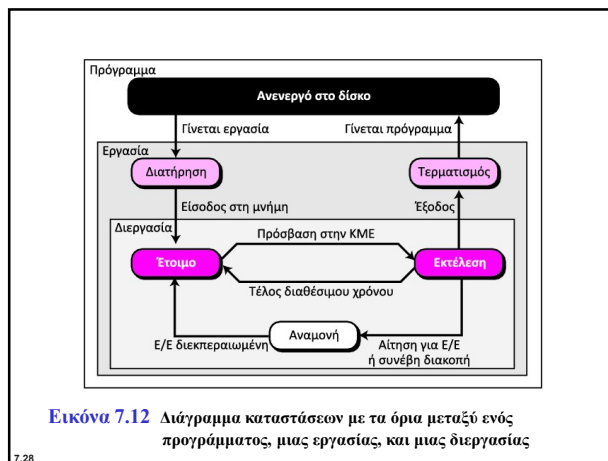
- **Πρόγραμμα** ονομάζεται ένα ανενεργό σύνολο εντολών οι οποίες έχουν αποθηκευτεί σε δίσκο.
- Ένα πρόγραμμα αποτελεί **εργασία** (job) από τη στιγμή που επιλέγεται για εκτέλεση μέχρι την ολοκλήρωση της εκτέλεσής του, οπότε μετατρέπεται και πάλι σε πρόγραμμα.
- **Διεργασία** (process) ονομάζεται ένα πρόγραμμα υπό εκτέλεση. Πρόκειται για ένα πρόγραμμα που έχει ξεκινήσει αλλά δεν έχει ολοκληρωθεί.

7.26

Διαγράμματα καταστάσεων

Η σχέση μεταξύ ενός προγράμματος, μιας εργασίας, και μιας διεργασίας γίνεται πιο ξεκάθαρη αν αναλογιστούμε το πώς ένα πρόγραμμα γίνεται εργασία και πώς μια εργασία γίνεται διεργασία. Αυτό μπορεί να απεικονιστεί σε ένα διάγραμμα καταστάσεων (state diagram), το οποίο παρουσιάζει τις διάφορες καταστάσεις κάθε μίας από αυτές τις έννοιες.

7.27

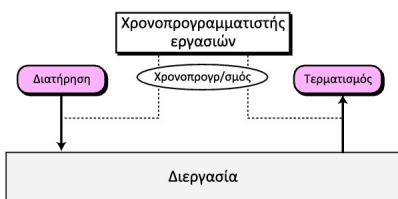


Εικόνα 7.12 Διάγραμμα καταστάσεων με τα όρια μεταξύ ενός προγράμματος, μιας εργασίας, και μιας διεργασίας

7.28

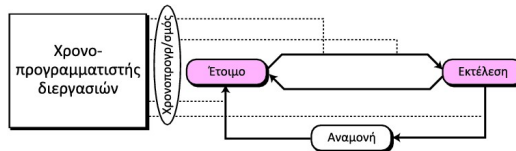
Χρονοπρογραμματιστές

Για τη μετάβαση μιας εργασίας ή μιας διεργασίας από τη μία κατάσταση στην άλλη, ο διαχειριστής διεργασιών χρειάζεται δύο χρονοπρογραμματιστές (schedulers): τον **χρονοπρογραμματιστή εργασιών** και τον **χρονοπρογραμματιστή διεργασιών**.



Εικόνα 7.13 Χρονοπρογραμματιστής εργασιών

7.29



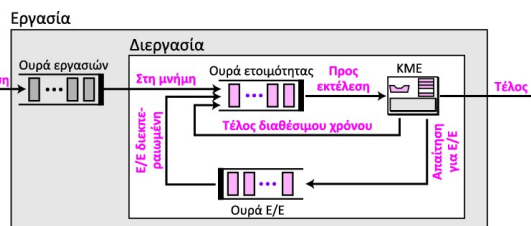
Εικόνα 7.14 Χρονοπρογραμματιστής διεργασιών

7.30

Χρήση ουράς

Το διάγραμμα καταστάσεων που χρησιμοποιήσαμε προηγουμένως δείχνει τον τρόπο μετάβασης από τη μία κατάσταση στην άλλη μόνο μίας εργασίας ή διεργασίας. Ωστόσο, στην πραγματικότητα υπάρχουν πολλές εργασίες και διεργασίες που ανταγωνίζονται η μία την άλλη για πόρους του υπολογιστή. Για τον ταυτόχρονο χειρισμό πολλών διεργασιών και εργασιών ο διαχειριστής διεργασιών χρησιμοποιεί ουρές (queues), δηλαδή λίστες αναμονής. Με κάθε εργασία ή διεργασία συσχετίζεται ένα τμήμα ελέγχου εργασίας ή τμήμα ελέγχου διεργασίας. Πρόκειται για ένα τμήμα μνήμης στο οποίο αποθηκεύονται πληροφορίες σχετικά με την εργασία ή τη διεργασία. Αντί ο διαχειριστής διεργασιών να αποθηκεύει στις ουρές τις ίδιες τις εργασίες ή τις διεργασίες, αποθηκεύει το τμήμα ελέγχου τους.

7.31



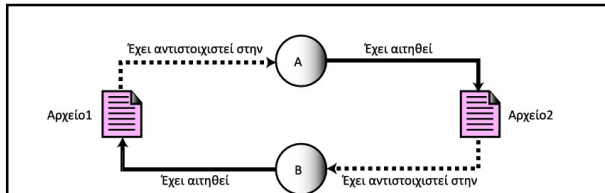
Εικόνα 7.15 Ουρές για διαχείριση διεργασιών

7.32

Συγχρονισμός διεργασιών

Η γενική ιδέα στην οποία βασίζεται η διαχείριση διεργασιών είναι ο συγχρονισμός των διαφόρων διεργασιών σε σχέση με τους διάφορους πόρους. Όταν οι πόροι χρησιμοποιούνται από περισσότερους από έναν χρήστες (διεργασίες, στην περίπτωση αυτή), είναι δυνατό να δημιουργηθούν δύο καταστάσεις: *αδιέξοδο* και *λιμοκτονία*.

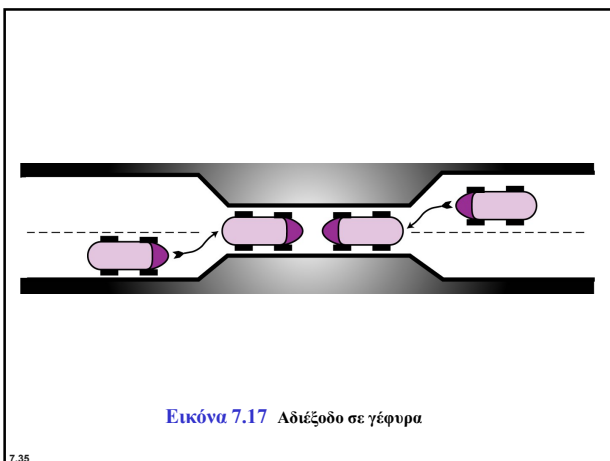
7.33



Εικόνα 7.16 Αδιέξοδο

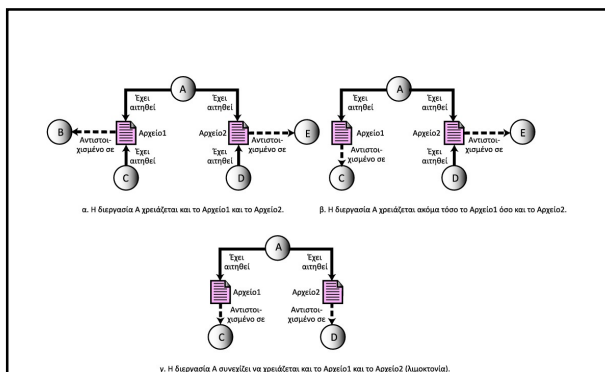
Αδιέξοδο προκαλείται όταν το λειτουργικό σύστημα δεν θέτει περιορισμούς στους πόρους που εκχωρούνται στις διεργασίες.

7.34



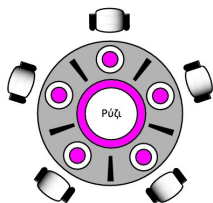
Εικόνα 7.17 Αδιέξοδο σε γέφυρα

7.35



Εικόνα 7.18 Λιμοκτονία

7.36



Εικόνα 7.19 Το πρόβλημα του δείπνου των φιλοσόφων



Η λιμοκτονία είναι το αντίθετο του αδιεξόδου. Μπορεί να παρουσιαστεί όταν το λειτουργικό σύστημα θέτει πάρα πολλούς περιορισμούς πόρων σε μια διεργασία.

7.37

Διαχειριστής συσκευών

Ο διαχειριστής συσκευών (device manager), ή διαχειριστής εισόδου/εξόδου, είναι υπεύθυνος για την προσπέλαση των συσκευών εισόδου/εξόδου. Σε ένα υπολογιστικό σύστημα υπάρχουν περιορισμοί στο πλήθος και την ταχύτητα των συσκευών εισόδου/εξόδου.

- Ο διαχειριστής συσκευών παρακολουθεί κάθε συσκευή εισόδου/εξόδου συνεχώς ώστε να εξασφαλίζει ότι λειτουργεί σωστά.
- Ο διαχειριστής διατηρεί μια ουρά για κάθε συσκευή εισόδου/εξόδου, ή μία ή περισσότερες ουρές για παρόμοιες συσκευές εισόδου/εξόδου.
- Ο διαχειριστής ελέγχει τις διάφορες πολιτικές για την προσπέλαση των συσκευών εισόδου/εξόδου.

7.38

Διαχείριση αρχείων

Για τον έλεγχο της προσπέλασης των αρχείων τα σύγχρονα λειτουργικά συστήματα χρησιμοποιούν **διαχειριστές αρχείων** (file managers). Η λεπτομερής περιγραφή του διαχειριστή αρχείων χρειάζεται επίσης εκτεταμένη γνώση των αρχών των λειτουργικών συστημάτων και είναι πέρα από τους στόχους αυτού του βιβλίου. Ο διαχειριστής αρχείων:

- ελέγχει την πρόσβαση σε αρχεία.
- επιβλέπει τη δημιουργία, τη διαγραφή, και την τροποποίηση των αρχείων.
- ελέγχει την ονομασία των αρχείων.
- επιβλέπει την αποθήκευση των αρχείων.
- είναι υπεύθυνος για την αρχειοθέτηση και τη λήψη εφεδρικών αντιγράφων.

7.39

7-4 ΜΙΑ ΑΝΑΣΚΟΠΗΣΗ ΤΩΝ ΛΕΙΤΟΥΡΓΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Σε αυτή την ενότητα θα παρουσιάσουμε μερικά δημοφιλή λειτουργικά συστήματα ως κίνητρο για περαιτέρω μελέτη. Επιλέξαμε τρία λειτουργικά συστήματα που είναι γνωστά στους περισσότερους χρήστες υπολογιστών: το UNIX, το Linux, και τα Windows.

7.40

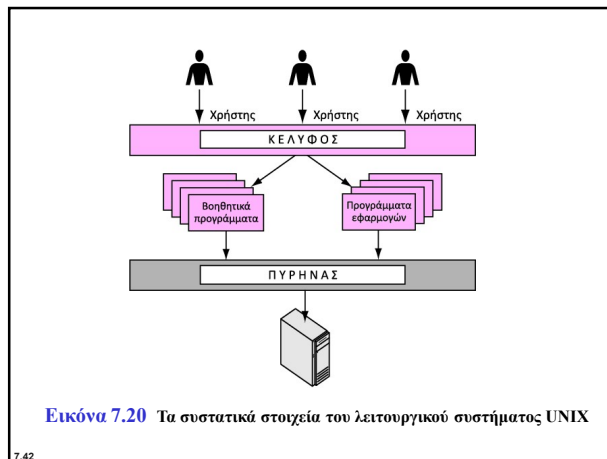
UNIX

Το UNIX σχεδιάστηκε το 1969 από τους Thomson και Ritchie της Ερευνητικής Ομάδας στην Επιστήμη των Υπολογιστών στα Εργαστήρια Bell. Από τότε έχουν κυκλοφορήσει πολλές εκδόσεις του UNIX. Πρόκειται για ένα δημοφιλές λειτουργικό σύστημα μεταξύ των προγραμματιστών υπολογιστών και των επιστημόνων της πληροφορικής.

I

Το UNIX είναι ένα πολυχρηστικό, πολυδιεργασιακό, φορητό λειτουργικό σύστημα. Έχει σχεδιαστεί για να διευκολύνει τον προγραμματισμό, την επεξεργασία κειμένου, και την επικοινωνία.

7.41



Εικόνα 7.20 Τα συστατικά στοιχεία του λειτουργικού συστήματος UNIX

7.42

Linux

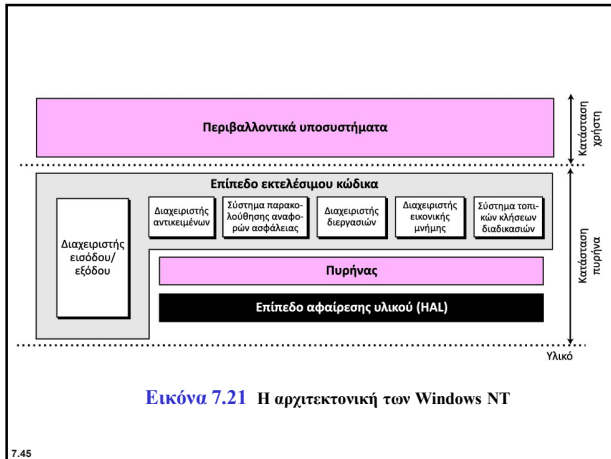
Το 1991, ο Linus Torvalds, ο οποίος φοιτούσε στο Πανεπιστήμιο του Ελσίνκι, ανέπτυξε ένα νέο λειτουργικό σύστημα που σήμερα είναι γνωστό ως Linux. Ο αρχικός πυρήνας, ο οποίος έμοιαζε με ένα μικρό υποσύνολο του UNIX, σήμερα έχει εξελιχθεί σε ένα πλήρες λειτουργικό σύστημα. Ο πυρήνας 2.0 του Linux, που κυκλοφόρησε το 1997, έγινε αποδεκτός ως εμπορικό λειτουργικό σύστημα, αφού διαθέτει όλα τα χαρακτηριστικά του UNIX.

7.43

Windows NT/2000/XP

Στο τέλος της δεκαετίας του 1980, η Microsoft υπό την ηγεσία του Dave Cutler ξεκίνησε την ανάπτυξη ενός νέου λειτουργικού συστήματος ενός χρήστη για την αντικατάσταση του MS-DOS (Microsoft Disk Operating System - Λειτουργικό Σύστημα Δίσκου της Microsoft). Το αποτέλεσμα αυτής της προσπάθειας ήταν τα Windows NT (όπου το NT προέρχεται από τα αρχικά των λέξεων New Technology, δηλαδή "Νέα Τεχνολογία"). Ακολούθησαν πολλές εκδόσεις των Windows NT, ενώ τελικά το όνομα άλλαξε σε Windows 2000. Το 2001 κυκλοφόρησαν τα Windows XP (όπου το XP προέρχεται από τη λέξη eXPerience, δηλαδή "εμπειρία"). Όλες αυτές οι εκδόσεις αναφέρονται ως Windows NT ή απλώς NT.

7.44



Εικόνα 7.21 Η αρχιτεκτονική των Windows NT

7.45