



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

Αρχιτεκτονική Υπολογιστών

Ενότητα 13: (Μέρος Β')

Λειτουργικό Σύστημα

Δρ. Μηνάς Δασυγένης
mdasyg@ieee.org

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών
<http://arch.icte.uowm.gr/mdasyg>



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «**Εκπαίδευση και Δια Βίου Μάθηση**» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σκοπός ενότητας

- Η κατανόηση της ανάγκης ύπαρξης λειτουργικού συστήματος.
- Η κατανόηση των λειτουργιών και των υπηρεσιών που προσφέρει το λειτουργικό σύστημα.
- Η παρουσίαση των βασικών υποσυστημάτων των λειτουργικών συστημάτων.



Η συμβολική γλώσσα είναι δύσχρηστη για σύνθετες ενέργειες

- Η ISA (assembly) είναι το επίπεδο που βρίσκεται ανάμεσα στο λογισμικό και το υλικό.
- Μπορούμε να κατασκευάσουμε ένα πρόγραμμα γραμμένο σε assembly για μια συγκεκριμένη αρχιτεκτονική, το οποίο θα εκτελεί μια συγκεκριμένη λειτουργία.
- Όμως αυτό απαιτεί ο προγραμματιστής να προγραμματίσει συναρτήσεις πρόσβασης και χρήσης πλήθος λειτουργιών χαμηλού επιπέδου, όπως πρόσβαση στο δίσκο, συσκευές I/O (οθόνη, πληκτρολόγιο).
- Ακόμη και η εγγραφή ενός χαρακτήρα στην οθόνη απαιτεί δεκάδες εντολές (η εντολή `int` είναι υπηρεσία του λειτουργικού συστήματος).



Το ΛΣ βοηθάει τον προγραμματιστή

- Αν ο προγραμματιστής θέλει η μηχανή του να εκτελεί εκτός από ένα πρόγραμμα, και επιπρόσθετα προγράμματα, τότε απαιτείται ένα μεγάλο ποσό κώδικα για την προστασία των εφαρμογών, την αλληλεπίδραση και πολλά άλλα.
- Όλα αυτά τα προβλήματα (και πλήθος άλλων) οδήγησαν στη δημιουργία και ευρεία υιοθέτηση των λειτουργικών συστημάτων.
- Το λειτουργικό σύστημα μπορεί να χαρακτηριστεί ως ένα πολύ μεγάλο και σύνθετο πρόγραμμα (λογισμικό) που κατευθύνει τον επεξεργαστή (υλικό).



Τι είναι το λειτουργικό σύστημα;

- Λειτουργικό σύστημα είναι ένα πρόγραμμα το οποίο, από την άποψη του προγραμματιστή, προσθέτει μια ποικιλία νέων εντολών και δυνατοτήτων πάνω και πέρα από εκείνες που παρέχει το επίπεδο ISA.
- Οι βασικές του λειτουργίες είναι δύο:
 - Παρουσιάζει ένα μηχανισμό αφαίρεσης, ο οποίος αφαιρεί όλες τις λεπτομέρειες που καθιστούν περίπλοκη τη χρήση του υλικού.
 - Παρουσιάζει μια εικονική μηχανή, η οποία είναι ευκολότερο να προγραμματιστεί και να διαχειριστεί (διαχειριστής πόρων).



Το λειτουργικό σύστημα ως μηχανισμός αφαίρεσης

- Κρύβει από τον προγραμματιστή τις λεπτομέρειες του υλικού.
- Ο χρήστης μπορεί να χρησιμοποιήσει έναν υπολογιστή χωρίς να ενδιαφέρεται για το πως λειτουργεί το υλικό.
- Ο προγραμματιστής δε χρειάζεται να δημιουργήσει πολύπλοκο κώδικα χαμηλού επιπέδου. Προγραμματίζει σε υψηλό επίπεδο.
- ΠΑΡΑΔΕΙΓΜΑ: Αν δεν υπήρχε λειτουργικό σύστημα για να γράψουμε ένα αρχείο θα έπρεπε να ρυθμιστεί ο ελεγκτής του δίσκου, να ρυθμιστεί ο bus master, να βρεθεί η ακριβής τοποθεσία στο δίσκο, να μετακινηθεί ο βραχίονας των κεφαλών κατάλληλα, να δοθεί η εντολή εγγραφής..... (και πολλά άλλα).
- Με το λειτουργικό σύστημα, ο χρήστης κάνει “SAVE”.
- Με το λειτουργικό σύστημα, ο προγραμματιστής κάνει fwrite (...).



Το λειτουργικό σύστημα ως μια εικονική μηχανή

- Το λειτουργικό σύστημα είναι δομημένο με τέτοιο τρόπο, ώστε να δημιουργεί την ψευδαίσθηση στις εφαρμογές ότι αυτές είναι οι μόνες που υπάρχουν και εκτελούνται στο υλικό.
- Δε μπορεί να παρεμβάλει η μια διεργασία την άλλη και να προκαλέσει πρόβλημα.
- Η κάθε διεργασία δε μπορεί να καταλάβει την ύπαρξη άλλων διεργασιών.
- Η κάθε διεργασία μπορεί αν θελήσει να χρησιμοποιήσει όλους τους πόρους (π.χ. μνήμη) του συστήματος (ή τουλάχιστον να έχει την ψευδαίσθηση) . Έτσι π.χ. είναι δυνατό να εκτελούμε 10 εργασίες που η κάθε μια απαιτεί 2GB RAM σε ένα μηχάνημα που έχει συνολικά φυσική μνήμη 2GB RAM.
- Μπορούμε να εκτελούμε πολλές περισσότερες διεργασίες από ότι αν δεν είχαμε λειτουργικό σύστημα.



Ποιοι είναι οι δυο θεμελιώδεις στόχοι;

- **Αποδοτικότητα:** Όσο περισσότερο καλύτερη εκμετάλλευση του υλικού.
- **Άνεση και εξυπηρέτηση:** Φιλική και απλή χρήση του υλικού.

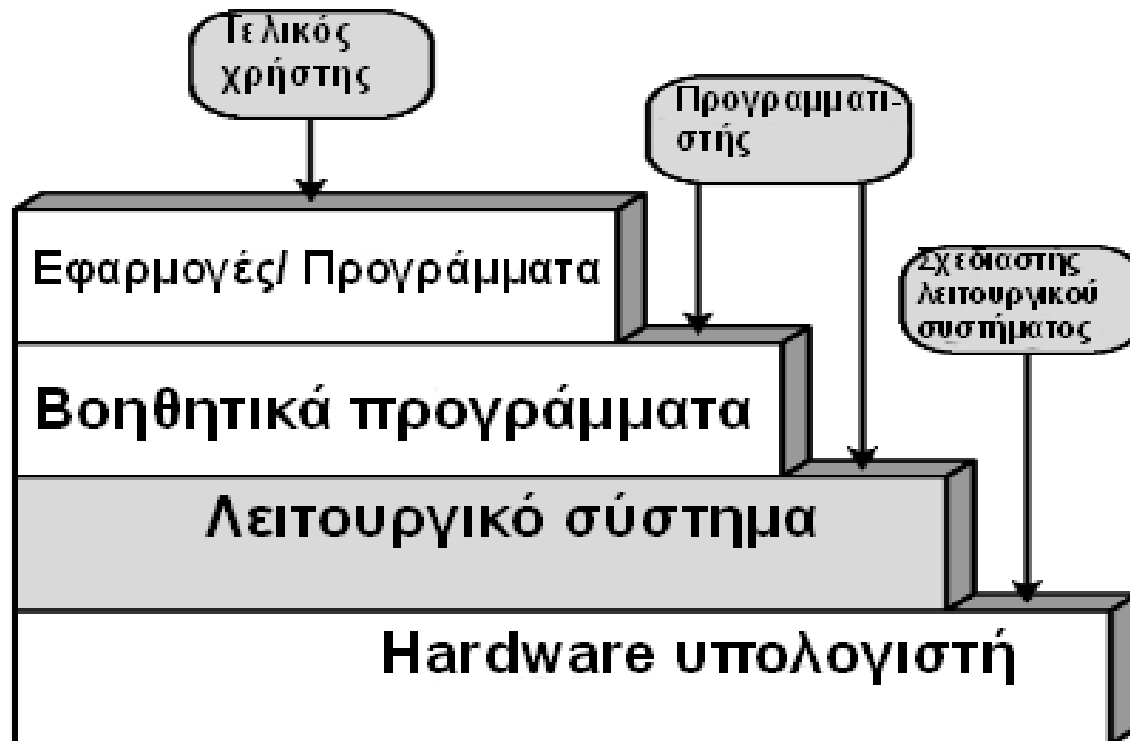


Το λειτουργικό σύστημα βρίσκεται πάνω από την ISA



Το επίπεδο αυτό, που βρίσκεται πάνω από την ISA, ονομάζεται “Επίπεδο Λειτουργικού Συστήματος”.

Ιεραρχία Software & Hardware



Πως το ΛΣ κρύβει την ανομοιομορφία του υλικού;

- Η ιεράρχηση του ΛΣ κρύβει την ανομοιομορφία του διαφορετικού υλικού (π.χ. επεξεργαστές, κάρτες γραφικών κ.ο.κ.).
- Με αυτόν τον τρόπο, προγράμματα μπορούν πολύ εύκολα να μεταφερθούν και να εκτελούνται σε πολλαπλές αρχιτεκτονικές με διαφορετικά υλικά.
- Χρησιμοποιούνται λοιπόν οι διεπαφές ή κλήσεις συστήματος (interfaces, system calls) του ΛΣ οι οποίες αυτές φροντίζουν να επικοινωνούν με το αντίστοιχο υλικό.
- Για παράδειγμα, μια εφαρμογή ενός παιχνιδιού μπορεί να εκτελείται σε κάθε κάρτα γραφικών που πληρεί τις ελάχιστες προδιαγραφές.



Το ΛΣ είναι μια μορφή προτυποποίησης

Το ΛΣ είναι μια **μορφή προτυποποίησης**, αφού ο προγραμματιστής χρησιμοποιεί τις **σαφώς ορισμένες διεπαφές** του Λ.Σ. (π.χ. για δημιουργία αρχείου, για εγγραφή αρχείου κ.ο.κ.) για να δημιουργεί εύκολα και γρήγορα, εφαρμογές που μπορούν να εκτελεστούν σε πλήθος αρχιτεκτονικών.



Πόσα ΛΣ υπάρχουν;

- Τα πρώτα ολοκληρωμένα λειτουργικά συστήματα παρουσιάστηκαν το 1960.
- Από τότε εκατοντάδες λειτουργικά συστήματα έχουν παρουσιασθεί, κάποια από τα οποία εγκαταλείφθηκαν τελείως, κάποια που ύστερα από μερικά χρόνια εγκατάλειψης τα ανέλαβαν άλλοι, και κάποια που συνεχίζουν να υπάρχουν για δεκαετίες.
- Τα λειτουργικά συστήματα έχουν διαφορετικούς στόχους.



Γιατί υπάρχουν τόσα πολλά ΛΣ;

- Υπάρχουν διαφορετικές ανάγκες.
 - γενικού σκοπού, διακομιστές, ενσωματωμένα συστήματα, παιχνιδιομηχανές, βιομηχανικά συστήματα, πραγματικού χρόνου.....
- Υπάρχουν (υπήρχαν) διαφορετικές εταιρίες.
 - Microsoft, IBM, SCO, Nokia, Google...
- Υπάρχουν εθελοντές.
 - Linux, FreeBSD,..
- Υπάρχει διαφορετικό hardware (διαφορετικό ISA).
 - Intel x86, IBM Cell, ARM, AVR, TI,...



Ποια είναι τα βασικά υποσυστήματα των ΛΣ;

- **Διαχείριση μνήμης.**
 - Παρέχει το ποσό της μνήμης που απαιτεί η κάθε διεργασία. Σελιδοποιεί τη μνήμη. Προστατεύει τη μνήμη.
- **Διαχείριση Αρχείων.**
 - Δημιουργία, Μετονομομασία, εγγραφή, διαγραφή και άλλες λειτουργίες. Προστασία.
- **Χρονοπρογραμματισμός Διεργασιών.**
 - Πότε για πόσο και σε ποιον επεξεργαστή θα εκτελεστεί μια διεργασία.
- **Διαχείριση και προστασία πόρων.**
 - Πότε θα δοθεί πρόσβαση και με τι δικαιώματα πρόσβασης;
- **Διαχείριση εισόδου εξόδου.**



Η εικονική μνήμη των ΛΣ (1/2)

- Διαχωρίζει την έννοια της φυσικής μνήμης και της μνήμης που βλέπει μια διεργασία.
- Χωρίζεται η φυσική μνήμη σε ισόποσα μεγέθη (συνήθως 4096 Bytes), που ονομάζονται πλαίσια (frames).
- Χωρίζεται η εικονική μνήμη, δηλαδή η μνήμη που δίνεται στις διεργασίες σε ισόποσα μεγέθη (ίδιου μεγέθους με τα πλαίσια), που ονομάζονται σελίδες (pages).
- Όταν η διεργασία ζητάει ένα ποσό μνήμης, το ΛΣ της δίνει έναν αριθμό από σελίδες (εικονική μνήμη).



Η εικονική μνήμη των ΛΣ (2/2)

- Στη συνέχεια το ΛΣ αντιστοιχίζει τις (εικονικές) σελίδες με τα (φυσικά) πλαίσια.
- Η εικονική μνήμη είναι πολύ μεγαλύτερη της φυσικής μνήμης, δεν έχουν όλες οι σελίδες αντιστοίχιση.
- Οι σελίδες που δεν έχουν αντιστοίχιση (επειδή δε χωράνε στη μνήμη) βρίσκονται σε μια περιοχή του σκληρού δίσκου που ονομάζεται swap file.
- Με αυτόν τον τρόπο μπορούμε να εκτελούνται πολλαπλές διεργασίες που διαφορετικά δε θα μπορούσαν να εκτελεστούν.



Ποια είναι τα κυριότερα προβλήματα στα ΛΣ;

- Μη επίτευξη 100% προστασίας: Υπάρχουν διεργασίες που ξεπερνούν τα μέτρα προστασίας του ΛΣ και μπορεί να προκαλέσουν κατάρρευση του.
- Αδιεξόδων: Υπάρχουν περιπτώσεις που λόγω εσφαλμένης ανάθεσης πόρων δημιουργούνται αδιέξοδα (παράδειγμα “πεινασμένοι φιλόσοφοι”).
- Χρονοδρομολόγησης: Υπάρχουν περιπτώσεις που διεργασίες χρησιμοποιούν σχεδόν αποκλειστικά τον επεξεργαστή και δε μπορεί να εξυπηρετήσει άλλες διεργασίες.
- Ανάκαμψη ύστερα από σφάλμα: Το υλικό θα καταρρεύσει κάποια στιγμή. Το ΛΣ θα πρέπει να φροντίσει όταν γίνει αυτό να μη καταρρεύσει.



Για περισσότερες πληροφορίες
υπάρχει το μάθημα
“Λειτουργικά Συστήματα”,
υποχρεωτικό σε ανώτερο εξάμηνο.



Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

