



Πανεπιστήμιο Δυτικής Μακεδονίας  
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

---

# Λειτουργικά Συστήματα

## Ενότητα 12: Ενσωματωμένα Συστήματα & RTOS

Δρ. Μηνάς Δασυγένης

[mdasyg@ieee.org](mailto:mdasyg@ieee.org)

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.icte.uowm.gr/mdasyg>

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

---



Πανεπιστήμιο Δυτικής Μακεδονίας



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ  
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

# 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.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
*επένδυση στην κοινωνία της γνώσης*  
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
Πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# Σκοπός της Ενότητας

---

- Η παρουσίαση των ιδιαίτερων χαρακτηριστικών των ΕΣ και η ικανοποίηση τους από ΛΣ ειδικής αρχιτεκτονικής.



# Λειτουργικά Συστήματα

---

- Μεγάλοι, γενικού σκοπού υπολογιστές είχαν ΛΣ για πολλά χρόνια - 1950's-1970's.
- Συστήματα βασισμένα σε μικροεπεξεργαστή – 1970:
  - Καθώς βελτιωνόταν η ισχύς (ταχύτητα/μνήμη).
  - DOS, CP/M, Windows, Linux.
- Πραγματικού χρόνου:
  - Πιο αργή εκκίνηση.
  - Πρόσφατα τα RTOS έγιναν επικρατούσα τάση.



# RTS: Γιατί να χρησιμοποιήσουμε ένα RTOS;

---

- Για πολλά χρόνια μικρά ενσωματωμένα RTS δεν χρησιμοποιούσαν RTOS.
- Ριζικές (αλλά όχι άμεσα εμφανείς) επιπτώσεις από τη χρήση ενός RTOS - σημαντικές επιπτώσεις για το λογισμικό:
  - Αξιοπιστία.
  - Παραγωγικότητα.
  - Συντηρησιμότητα.
- Απλοποιεί σημαντικά την ανάπτυξη του μεσαίου/μεγάλου RTS.



# Παράδειγμα: Απλό ενσωματωμένο σύστημα

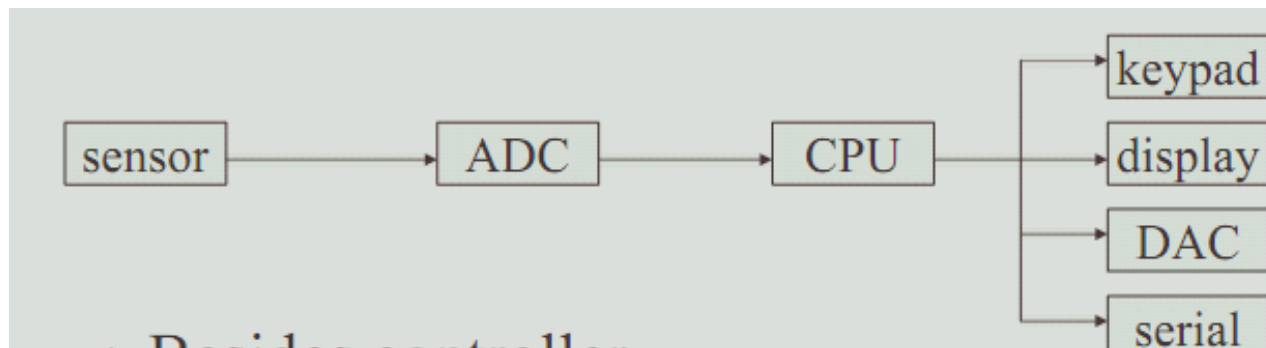
---

- Ελεγκτής φούρνου:
  - Μπορείτε να σκεφτείτε πολλούς τρόπους προγραμματισμού:
    - Ατέρμων βρόχο.
    - Χρονοδηγούμενο.
    - Σταθμοσκόπηση.
  - Όλα “μονού νήματος” προγράμματα.
- Αρκετά αποδεκτά.
- Σημειώστε ότι ο προγραμματιστής πρέπει να ασχοληθεί συστηματικά με συσκευές χαμηλού επιπέδου (αισθητήρες) – χρειάζεται καλή εξάσκηση σε υλικό/λογισμικό για καλά αποτελέσματα.



# Παράδειγμα: Ενσωματωμένο σύστημα μεσαίου μεγέθους

- Έλεγχος θερμοκρασίας σε ένα σωλήνα μηχανής Jet.
- Εκτός από τον ελεγκτή:
  - cf φούρνος + ακριβείς προθεσμίες.
- Δευτερεύουσες εργασίες:
  - Πρόσβασης πληροφοριών - πληκτρολόγιο/οθόνη.
  - Σειριακή σύνδεση με τον καταγραφέα πτήσης.





# Παράδειγμα: Ενσωματωμένο σύστημα μεσαίου μεγέθους (συνέχεια)

---

- Ένα απλό νηματικό πρόγραμμα δεν είναι ικανοποιητικό.
- Ξεχωριστές εργασίες:
  - Θα πρέπει να μπορούν να εξυπηρετούνται σε άγνωστες χρονικές στιγμές (async).
  - Θα πρέπει να μπορούν να επεξεργάζονται ταυτόχρονα.
- Προγραμματίσιμο με γεγονότα διακοπής (ISR=εργασία):
  - Προγραμματισμός χαμηλού επιπέδου σε βάθος.
  - Ασφάλεια, χρονισμοί και άλλα.
- Το RTOS αφαιρεί αυτό το βάρος από τον προγραμματιστή.



# Βασικά χαρακτηριστικά ΛΣ που απαιτούνται από ένα RTS

---

- Πρέπει να υποστηρίζει πολλαπλή δομή εργασιών από RT προγράμματα και ως εκ τούτου:
  - Χρονοπρογραμματισμό πολλαπλών εργασιών.
  - Συγχρονισμό πολλαπλών εργασιών.
  - (Διαχείριση μνήμης μεταξύ εργασιών).
- Ακόμα υποστήριξη προγραμματισμού με ελάχιστο λεξιλόγιο υλικού.



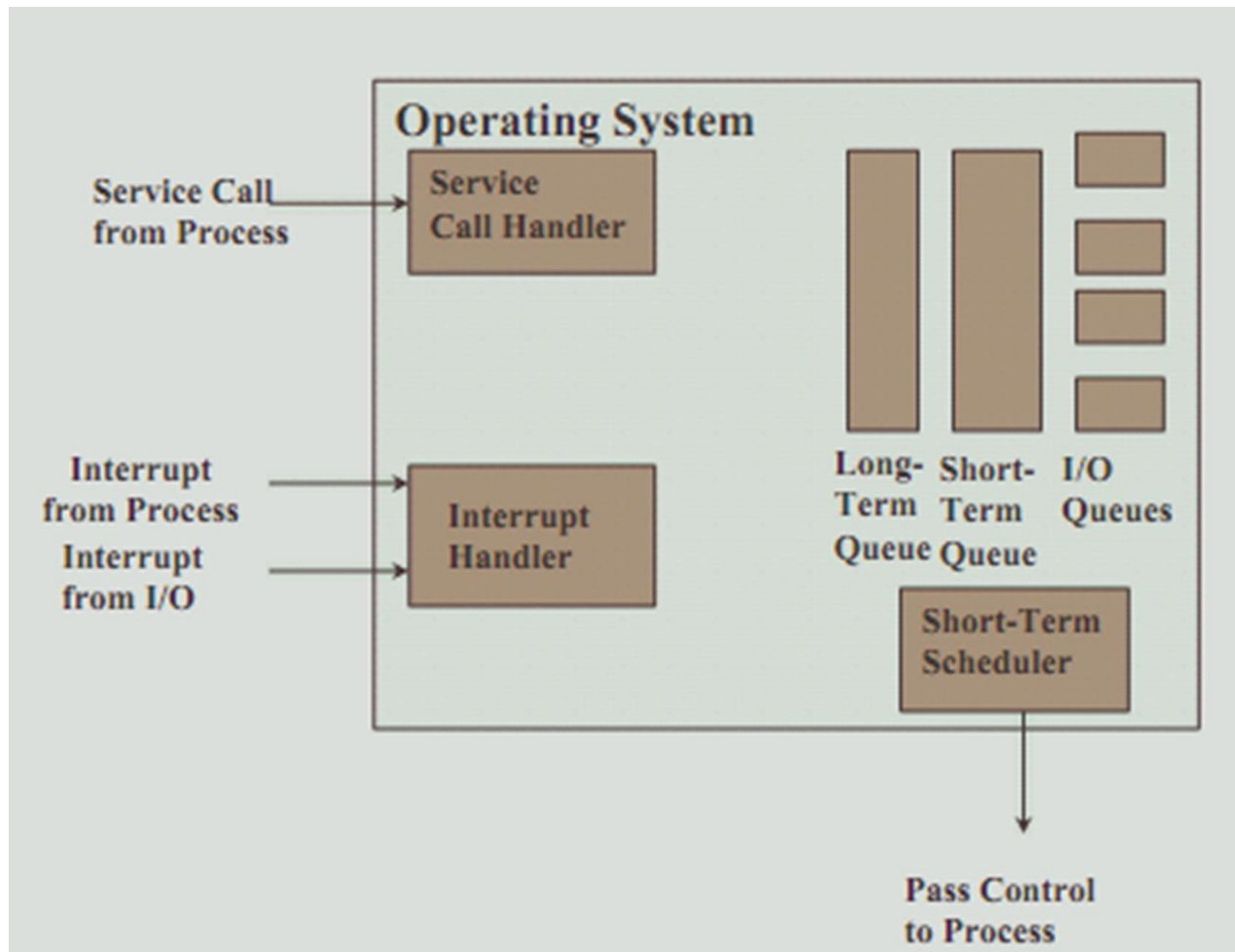
# Λειτουργικό σύστημα

---

- Μια διεπαφή μεταξύ χρήστη και υλικού:
  - Ένα πρόγραμμα που χειρίζεται το υλικό.
  - Ο πυρήνας έγκειται στη μνήμη.
  - Αποκρύπτει τις λεπτομέρειες του υλικού.
- Ελέγχει την πρόσβαση άλλων εφαρμογών στη CPU:
  - Διαλέγει και ρυθμίζει τον επεξεργαστή για το πρόγραμμα που θα εκτελεστεί.
  - Εγκαταλείπει την δικιά του χρήση από τον επεξεργαστή.



# Βασικά στοιχεία ενός ΛΣ



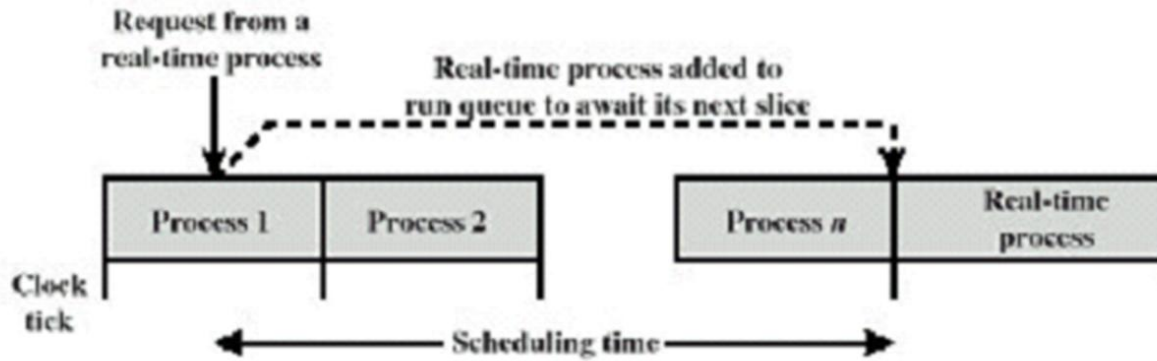
# Χρονοπρογραμματισμός στα RTOS

---

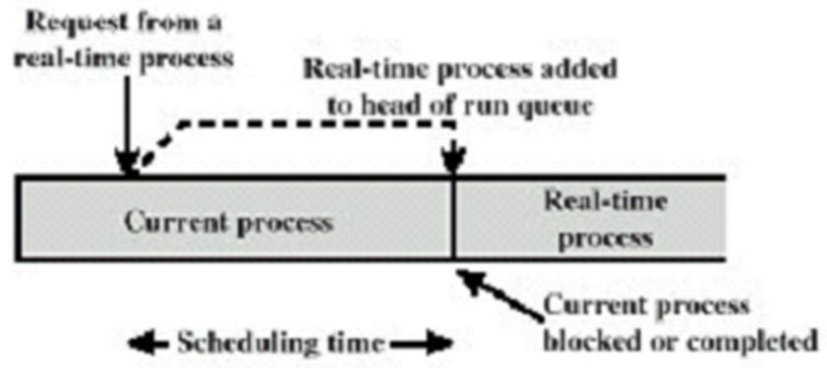
- Τα RTOS δεν πρέπει να είναι υποχρεωτικά γρήγορα, αλλά πρέπει να είναι:
  - Ευέλικτα στη πολιτική προγραμματισμού.
  - Ικανά να προγραμματίσουν νέες εργασίες.
- Οι μέθοδοι των ΛΣ γενικού σκοπού είναι ανεπαρκής.
- Τα RTOS συνήθως έχουν μια προτίμηση με γνώμονα την προτεραιότητα του χρονοπρογραμματιστή.



# Που έγκειται το πρόβλημα με το RR, προτεραιότητα FIFO;



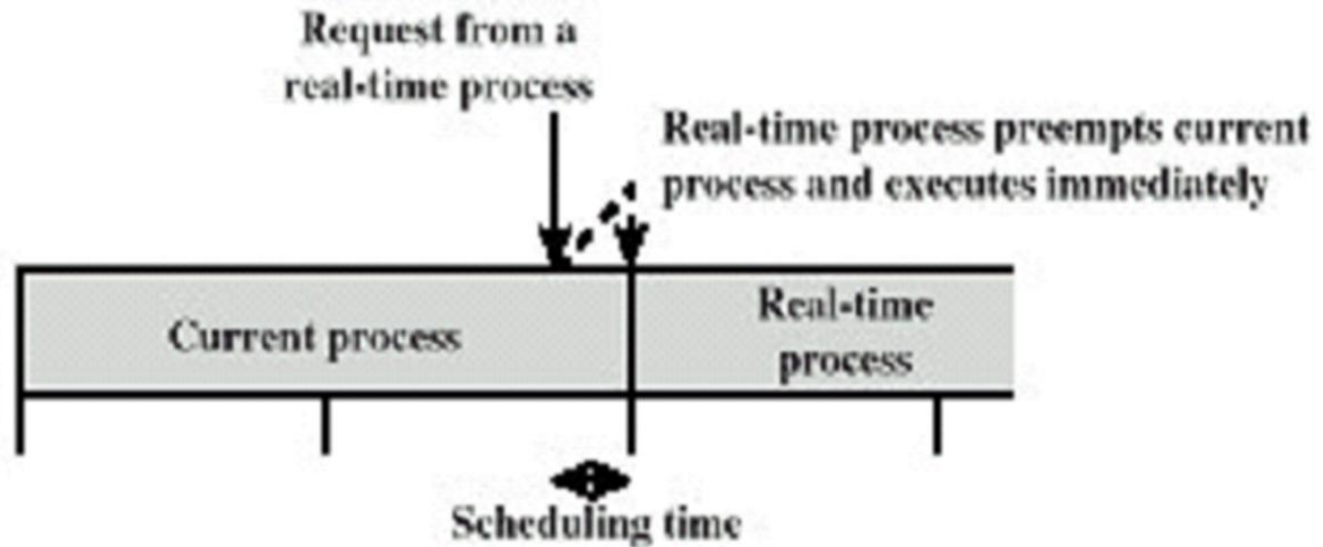
(a) Round-robin Preemptive Scheduler



(b) Priority-Driven Nonpreemptive Scheduler



# Προεκχωρούμενο, οδηγούμενο με προτεραιότητα



(d) Immediate Preemptive Scheduler

# RT χρονοπρογραμματιστής

---

- Προτίμηση.
- Όταν καλείται ο χρονοπρογραμματιστής;
  - Ολοκλήρωση διαδικασίας.
  - Απελευθέρωση διαδικασίας (μια διακοπή, είτε ρολογιού είτε συσκευής).
  - Διαδικασία μπλοκαρισμένη για E/E.
  - Όχι τεμαχισμός χρόνου (όπως το RR).
  - “Γεγονοδηγούμενος” (event driven) προγραμματισμός.
- Γεγονότα με προτεραιότητα.
- Πως αποφασίζει ποιός θα πάρει την CPU?
  - Υψηλότερη προτεραιότητα - όχι FIFO.
  - Η ευελιξία εξαρτάται από το πως εκχωρούνται οι προτεραιότητες.





# Προεκχώρηση (preempt) του πυρήνα

---

- Πολλά ΛΣ (σχεδόν όλα τα UNIX-like) επιτρέπουν τις διαδικασίες χρήστη να προεκχωρηθούν, αλλά όχι τις διαδικασίες επιπέδου πυρήνα όπως τις κλήσεις συστήματος.
- Αυτό γίνεται για να προστατευτούν τα δεδομένα πυρήνα κατά τη διάρκεια της πρόσβασης τους.
- Σε αυτή τη περίπτωση, η απόκριση μπορεί να είναι αργή.
- Κάποια ΛΣ χρησιμοποιούν μηχανισμούς συγχρονισμού για να προστατεύσουν τα δεδομένα πυρήνα (Solaris).



# Πραγματικός χρόνος και ΛΣ

---

- Τα RTS συχνά θέλουν ένα ΛΣ:
  - Για αξιοπιστία.
  - Πολυδιεργασίες και νήματα.
  - Φθηνότερο να αναπτυχθούν μεγάλα συστήματα πραγματικού χρόνου.
- Αλλά, δεν θέλουμε να χάσουμε την ικανότητα να τηρούμε τις προθεσμίες.



# Αυστηρά συστήματα πραγματικού χρόνου

---

- Πρέπει να τηρούν όλες τις προθεσμίες.
- Δεν είναι δυνατό να ενσωματώσει τα χαρακτηριστικά των μοντέρνων γενικού σκοπού ΛΣ που απομακρύνουν τα προγράμματα από το υλικό (το οποίο το χρησιμοποιούν μέσω κλήσεων συστήματος).
- Αυτά τα χαρακτηριστικά είναι περίπλοκα και δυνητικά μπορούν να εισάγουν απεριόριστες καθυστερήσεις.
- Σίγουρα όχι εικονική μνήμη.
- Χρησιμοποίηση ενός ειδικού σκοπού RTOS



# Ανεκτικά συστήματα πραγματικού χρόνου

---

- Πρέπει να τηρούν τις περισσότερες προθεσμίες.
- Για συγκεκριμένο είδος εφαρμογών:
  - απαιτούν προχωρημένα χαρακτηριστικά ΛΣ.
- Μπορούν να χρησιμοποιήσουν ένα γενικού σκοπού ΛΣ τροποποιημένο ώστε να:
  - Χρησιμοποιεί προτεραιότητες.
  - Μειώσει τις καθυστερήσεις των διακοπών.
  - Υποστηρίζει προεκχώρηση προτεραιοτήτων.
- Τα μοντέρνα γενικού σκοπού ΛΣ συχνά ενσωματώνουν αυτές τις τροποποιήσεις όπως τα Windows 2000 και νεότερα.



# RTOS απαιτήσεις

---

- Ντετερμινισμό.
- Ανταπόκριση (αναφέρεται από τους κατασκευαστές):
  - Γρήγορη εναλλαγή διαδικασίας/νήματος.
  - Γρήγορη απόκριση διακοπής.
- Έλεγχος χρήστη πάνω στις πολιτικές του ΛΣ:
  - Κυρίως τον προγραμματισμό, πολλά επίπεδα προτεραιότητας.
  - Υποστήριξη μνήμης (ειδικά ενσωματωμένη).
- Αξιοπιστία.



# Τι κάνει ένα ΛΣ να είναι ΛΣ πραγματικού χρόνου;

---

- Πολυνηματικά και με προτεραιότητες.
- Προτεραιότητες νημάτων:
  - (υπάρχουν συστήματα οδηγούμενα χωρίς προθεσμίες).
- Προβλέψιμοι μηχανισμοί συγχρονισμού νημάτων.
- Κληρονομικότητα προτεραιότητας.
- Γνωστή συμπεριφορά ΛΣ.



# Λειτουργικά συστήματα πραγματικού χρόνου (1/2)

---

- Περίπου 100 λειτουργικά συστήματα πραγματικού χρόνου.
- LynxOs, Blue Cat Linux, Ecos, ChorusOS, VRXT, AMX Kadak, JMI C executive, CMX-RTX, CORTEX , Intime, iRMX, IRX, Jbed, MQX,ON Time RTOS-32, OnCore OS, Linux for Real Time, OS-9, OSE, PDOS, QNX Neutrino, REDICE Linux, RTLinux, Prose, Katix, uCR, Maruti, Spring, Harmony, DeltaOS, embOS, eRTOS, ETS-RTX, EYRX, INTERGRITY, RTSecure, Nucleus Plus, RTMach, SHARCOS, Smx, SuperTask!, TxOS&TxSDK, Vx Works, RTEms, RTXC, Hurd, Hard Hat, TimeSys, Virtuoso, Embedix, ElinOS, RED-Linux, uClinux, RTAI-Linux, RTXC, pSOSsystem3, ADEOS.



# Λειτουργικά συστήματα πραγματικού χρόνου (2/2)

---

- Λειτουργικά συστήματα από τον ακαδημαϊκό χώρο.
- Λειτουργικά συστήματα από την βιομηχανία / εμπορικά λειτουργικά συστήματα.
- Λειτουργικά συστήματα ανοιχτού κώδικα / ελεύθερου λογισμικού.





# Λειτουργικά συστήματα πραγματικού χρόνου: Έννοιες προς σκέψη

---

- Η αρχιτεκτονική του συστήματος.
- Δυνατότητες πραγματικού χρόνου.
- Κόστος απόκτησης.
- Διαθεσιμότητα πηγαίου κώδικα.
- Υποστηριζόμενο υλικό.
- Εργαλεία, λογισμικό.
- Τεκμηρίωση.
- Υποστήριξη.



# Λειτουργικά συστήματα πραγματικού χρόνου από τον ακαδημαϊκό χώρο

---

- + Κόστος απόκτησης.
- + Διαθεσιμότητα πηγαίου κώδικα.
- + Τεκμηρίωση.
- Υποστηριζόμενο υλικό.
- Εργαλεία, λογισμικό.
- Υποστήριξη.



# Λειτουργικά συστήματα πραγματικού χρόνου από τη βιομηχανία

---

- + Υποστηριζόμενο υλικό.
- + Εργαλεία, λογισμικό.
- + Τεκμηρίωση.
- + Υποστήριξη.
- Κόστος απόκτησης.
- Διαθεσιμότητα πηγαίου κώδικα.



# Λειτουργικά συστήματα πραγματικού χρόνου ανοιχτού κώδικα

---

- + Κόστος απόκτησης.
- + Διαθεσιμότητα πηγαίου κώδικα.
- + Τεκμηρίωση.
- + Υποστήριξη.
- Υποστηριζόμενο υλικό.
- Υποστήριξη.



# RTOS: Virtuoso

---

| OS                                | Virtuoso   |
|-----------------------------------|--|
| Distributed<br>Central-controlled | Distributed  |
| Memory Support                    | Memory Maps<br>Memory Pools  |
| Communication                     | Link Drivers   |
| Synchronization                   | Semaphores, Events, Queues,<br>Mailboxes, Semaphore Channels,<br>Linked-List Channels. Stack Channels                            |
| Low power                         | Όχι  |
| Free                              | Όχι  |
| Source Code                       | Όχι  |
| Platforms Supported               | Analog Devices [SHARC ADSP-<br>2106, HammerHead SHARC<br>ADSP-2116X, ADSP-21020]<br>TI [TM320C6x,RMS320C4x]<br>Win NT simulation |



# RTOS: QNX

---

| OS                                | QNX   |
|-----------------------------------|---|
| Development Environment Support   | QNX + Third Party Tools   |
| Modularity Scalability            | Modular<br>Scalable<br>MicroKernel  |
| Distributed<br>Central-controlled | Central-controlled<br>SMP Supported   |
| Memory Support                    | Physical memory is divided in 4k pages.<br>Full protection model Private Virtual<br>Memory                                |
| Communication                     | Message Passing, POSIX Message Queues<br>, Pipes, FIFOs   |
| Synchronization                   | Mutexes, Condvars, Barriers, Sleep on<br>Locks, Read/Writer locks, Semaphores,<br>Atomic Operations, Send /Receive Policy |
| Low power                         | Όχι   |
| Free                              | Όχι   |
| Source Code                       | Όχι   |
| Platforms Supported               | PowerPC, MIPS, ARM, x86   |



# RTOS: ECos

---

| OS                                | ECos  |
|-----------------------------------|---|
| Development Environment Support   | GNU Tool Chain + RedHat Tools                                     |
| Modularity Scalability            | Scalable Modular  |
| Distributed<br>Central-controlled | Central-controlled  |
| Memory Support                    | Memory Pools  |
| Communication                     | Message Boxes   |
| Synchronization                   | Mutex, Conditions Variables, Event Flags, Messages Queueus        |
| Low power                         | Όχι   |
| Free                              | Ναι   |
| Source Code                       | Ναι   |
| Platforms Supported               | ARM, Hitachi SH3, Intel x86,MIPS, Matsushita AMBx, PowerPC, SPARC |



# RTOS: RTLinux

| OS                              | RTLinux  |
|---------------------------------|--|
| Development Environment Support | GNU Tool Chain + FSMLabs   |
| Modularity Scalability          | Modular Scalable   |
| Distributed Central-controlled  | Central-Controlled<br>Partial SMP Support  |
| Memory Support                  | Linux Memory Management<br>Memory Buffer Pool  |
| Communication                   | SYS V IPC, Shared Memory, RT-FIFOs   |
| Synchronization                 | Standard Linux Synchronization Methods   |
| Low power                       | Όχι  |
| Free                            | Ναι  |
| Source Code                     | Ναι  |
| Platforms Supported             | (IA32) ,PowerPC: 603, 604, 7400, MPC8260,MPC860, IBM 4056P, PowerMac G4, Synergy VGM5, MIPSAMD Elan NetSC520 , Alpha (ev6) |





---

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



---

# Τέλος Ενότητας



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

