

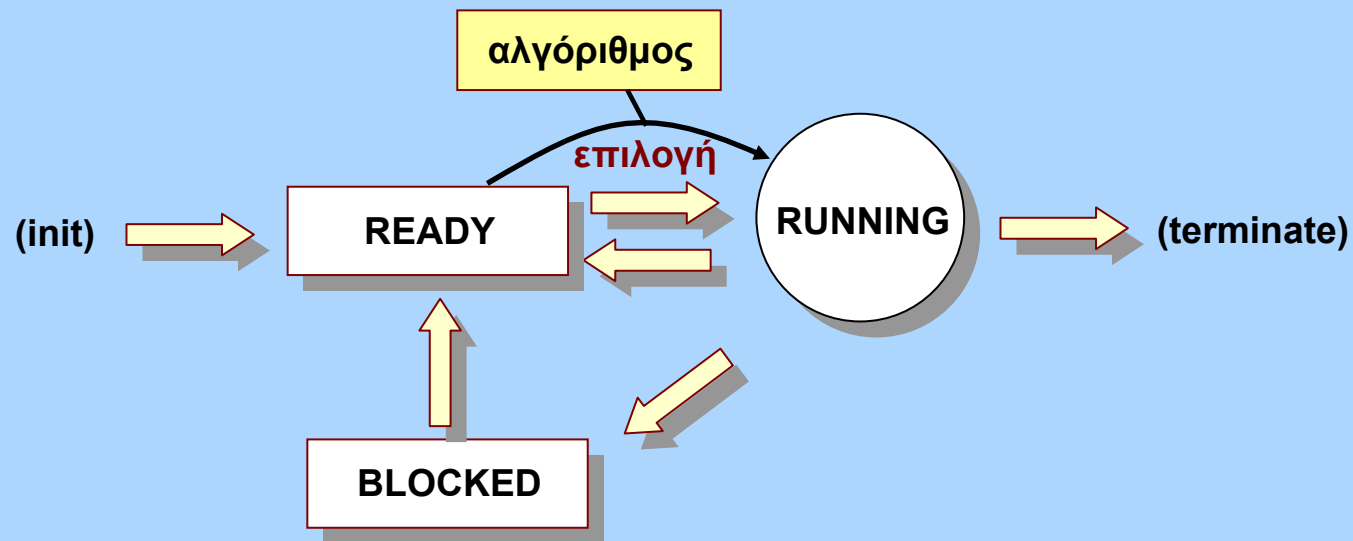
**Λειτουργικά Συστήματα Πραγματικού Χρόνου
2006-07**

Χρονοδρομολόγηση I

Μοντέλα διεργασιών, Προθεσμίες και Αλγόριθμοι

Μ.Στεφανιδάκης

Χρονοδρομολόγηση (scheduling)



- Αλγόριθμος χρονοδρομολόγησης
 - Επιλογή διεργασίας προς εκτέλεση
 - Τήρηση περιορισμών διεργασίας

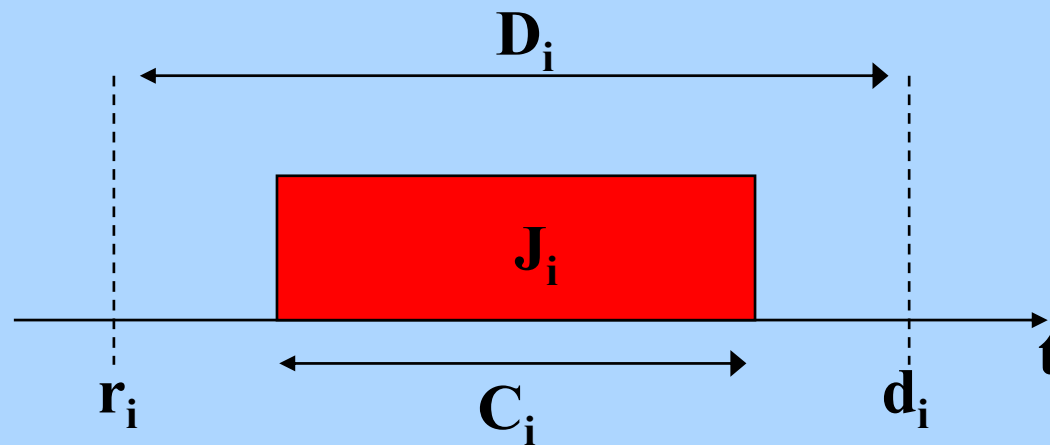
Χρονοδρομολόγηση

- Χρονοδρομολόγηση σε πραγματικό χρόνο:
 - Κατανομή “υπολογιστικής” ισχύος
 - Χρήση CPU
 - με εξασφάλιση περιορισμών διεργασιών
 - Χρονικές προθεσμίες (deadlines)
 - Σειρά εκτέλεσης
 - Χρήση κοινών πόρων

Δυνατότητα χρονοδρομολόγησης (schedulability)

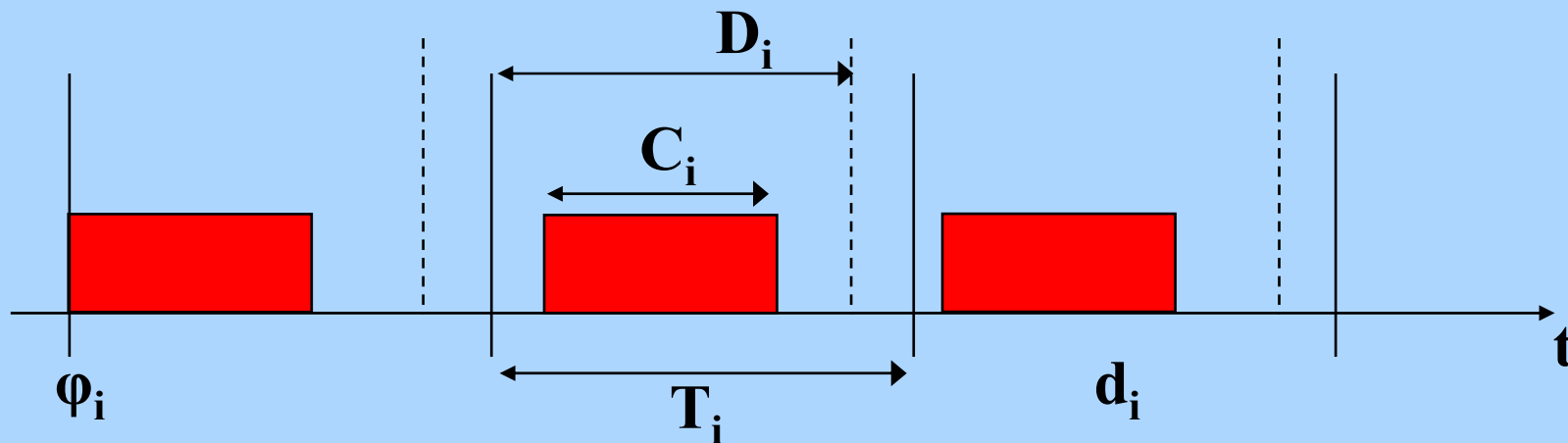
- δυνατότητα χρονοδρομολόγησης ομάδας διεργασιών
 - Με συγκεκριμένους περιορισμούς
 - Μπορούν να εξασφαλιστούν οι περιορισμοί αυτοί;
 - Υπολογισμός on-line
 - τεστ αποδοχής (*acceptance test*)
 - τι συμβαίνει σε περίπτωση απόρριψης;

Μοντέλο εκτέλεσης διεργασίας J_i



- r_i : χρόνος έκδοσης (*release* ή *arrival time*)
 - J_i γίνεται READY
- C_i : χρόνος εκτέλεσης (*execution time*)
- D_i : σχετική προθεσμία (*relative deadline*)
 - διαθέσιμος χρόνος ολοκλήρωσης

Περιοδικές διεργασίες



- φ_i : έκδοση πρώτου τμήματος διεργασίας (*phase*)
- T_i : χρόνος μεταξύ διαδοχικών εκδόσεων (min)
 - *Περίοδος*
 - *Και προθεσμία αν δεν ορίζεται το D_i*
- U_i : χρησιμοποίηση (*utilization*) = C_i/T_i

Μη περιοδικές διεργασίες

- Άγνωστος χρόνος έκδοσης (άφιξης)
 - Εξωτερικά γεγονότα
- Συνύπαρξη με περιοδικές διεργασίες
- Προθεσμίες
 - Soft ή καθόλου: **μη περιοδικές** διεργασίες
 - Hard: **σποραδικές** διεργασίες

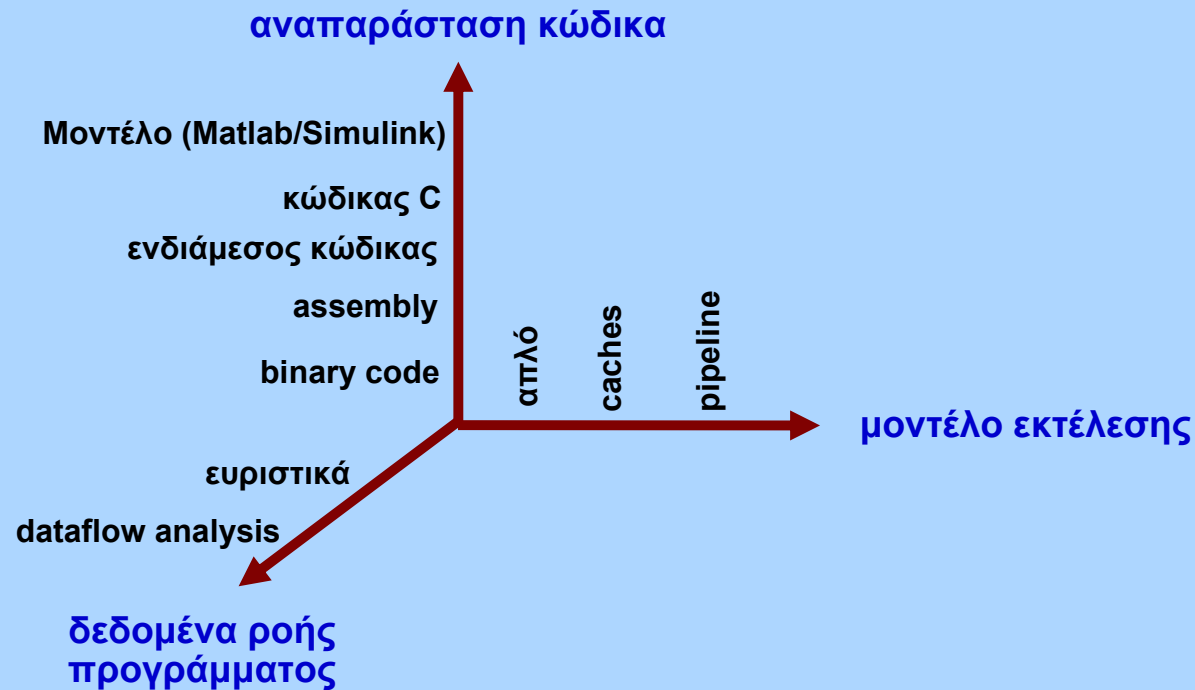
Χρόνος εκτέλεσης

- Πώς υπολογίζεται;
 - Κρίσιμο μέγεθος
 - Είσοδος σε όλους τους αλγορίθμους χρονοδρομολόγησης
- Χρόνος εκτέλεσης χειρότερης περίπτωσης
 - Worst Case Execution Time (**WCET**)
 - Η **μόνη** προσέγγιση του πραγματικού χρόνου εκτέλεσης!
 - Ακρίβεια προσέγγισης
 - Ασφάλεια λειτουργιών πραγματικού χρόνου
 - Αποδοτική χρησιμοποίηση CPU

Υπολογισμός WCET

- Στατικά
 - Υψηλή περιγραφή αλγορίθμου:
 - Εξαγωγή μονοπατιών (ροής) εκτέλεσης
 - Χαμηλό επίπεδο (κώδικας μηχανής)
 - Υπολογισμός χρονικών καθυστερήσεων
 - *Εξαρτάται από σύστημα εφαρμογής!*
- Μέσω μετρήσεων
 - Ευριστικές μέθοδοι
 - Επιλεγμένα σετ δεδομένων εισόδου
- Στοχαστικά

Υπολογισμός WCET



- **Ανοικτό ερευνητικό πεδίο!**
 - και στο εφαρμοσμένο επίπεδο της βιομηχανίας...

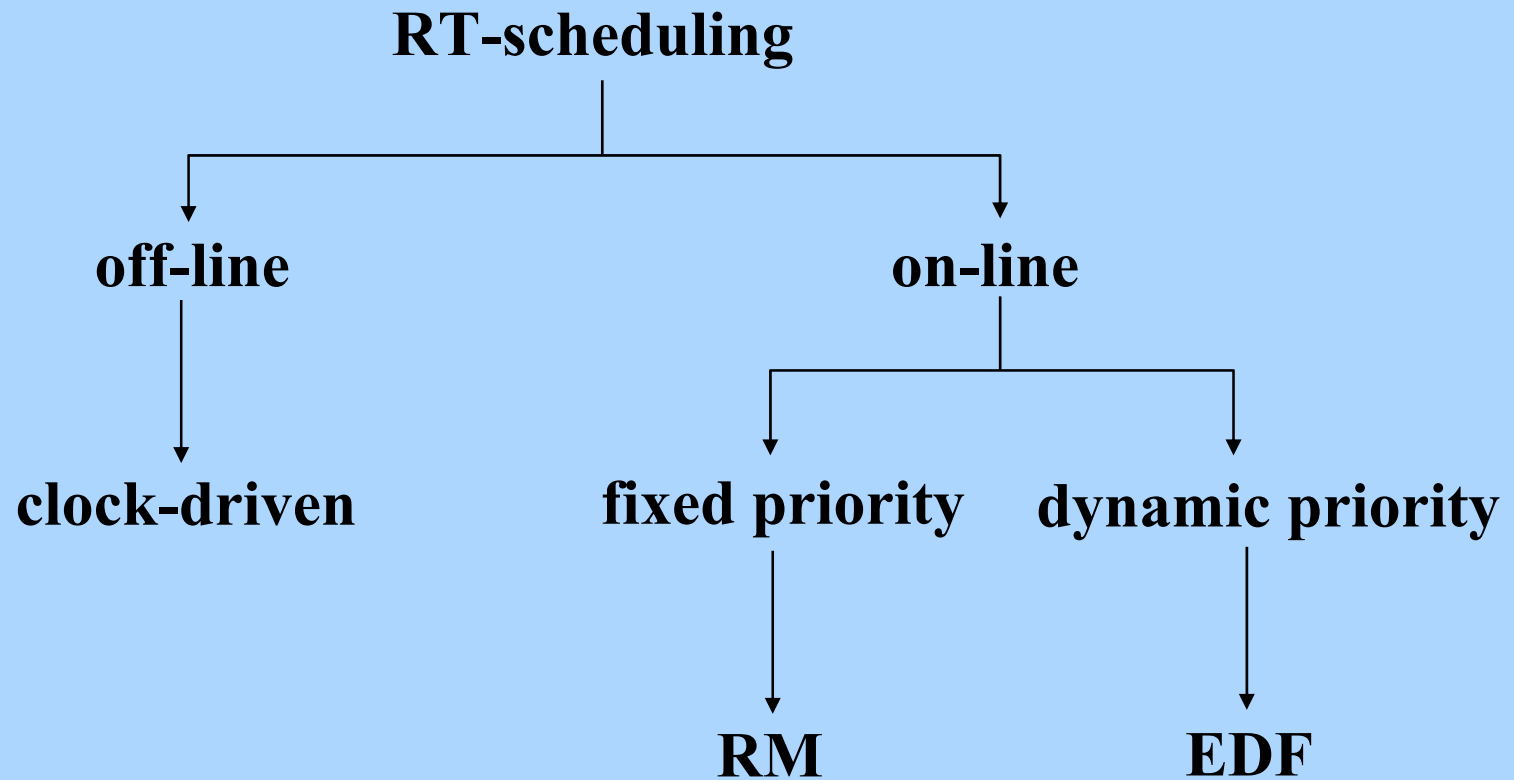
Υλικό: παράγοντες απροσδιοριστίας

- Επίδραση στον χρόνο εκτέλεσης
 - Μνήμες cache
 - Pipeline επεξεργαστή
 - Διακοπές (interrupts)
 - DMA, ανανέωση DRAM κ.λ.π.

Μοντέλο εκτέλεσης διεργασίας: παραδοχές και διορθώσεις

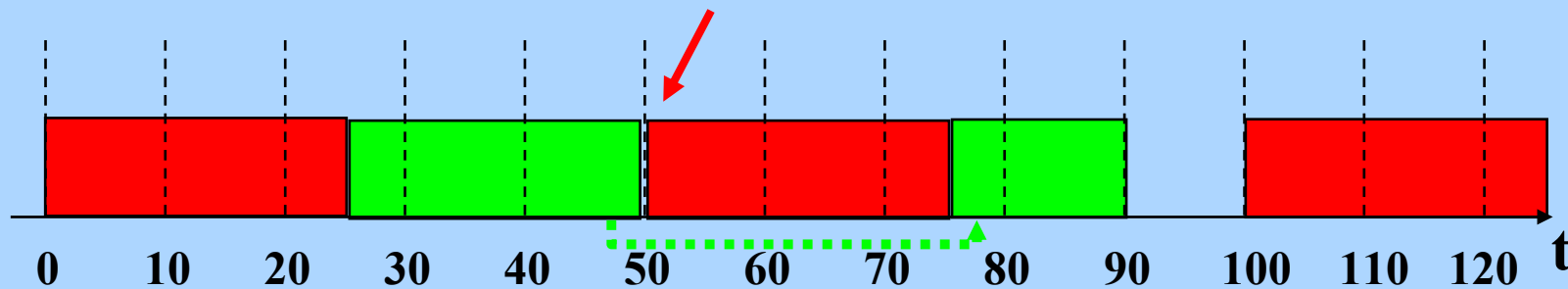
- Περιοδικές διεργασίες
 - μοντέλο server για μη περιοδικές
- Προσθήκη καθυστερήσεων για:
 - context-switching
 - διαμοιρασμό πόρων
 - αυτο-αδρανοποίηση (self-suspending)
- Προεκτοπιστικό (preemptive) σύστημα
- Εκτέλεση χρονοδρομολογητή
 - σε κάθε scheduling event (δημιουργία, τερματισμός διεργασίας)

Μέθοδοι χρονοδρομολόγησης



Προτεραιότητα διεργασιών

$(C_1=25, T_1=50)$ $(C_2=40, T_2=100)$

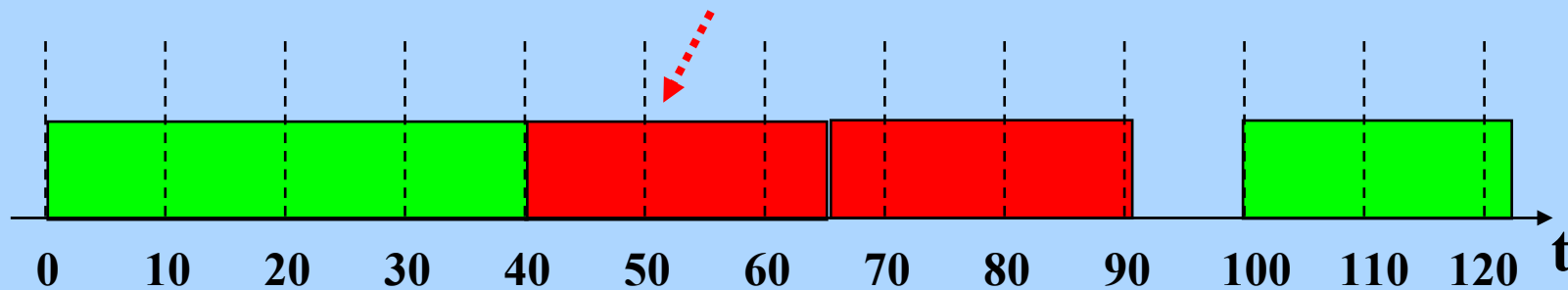


προτεραιότητα 1 > προτεραιότητα 2

- Σχήμα fixed-priority:
 - σταθερή προτεραιότητα σε όλη τη διάρκεια εκτέλεσης.
- Χρονοδρομολόγηση με βάση προτεραιότητες.
 - *Μέθοδος επιλογής προτεραιοτήτων;*

Προτεραιότητα διεργασιών

$(C_1=25, T_1=50)$ $(C_2=40, T_2=100)$



προτεραιότητα 2 > προτεραιότητα 1

- **Rate Monotonic scheduling**
 - ανάθεση προτεραιοτήτων πριν την εκτέλεση
 - “όσο μικρότερη η περίοδος, τόσο μεγαλύτερη η προτεραιότητα”
 - θα διάλεγε μεγαλύτερη προτεραιότητα για διεργασία-1

Αλγόριθμος RM

- Ο αλγόριθμος RM είναι βέλτιστος (optimal) μεταξύ fixed-priority αλγορίθμων:
 - “εάν μία ομάδα διεργασιών δεν μπορεί να χρονοδρομολογηθεί με τον RM, δεν μπορεί με κανέναν άλλον fixed-priority αλγόριθμο”
- Αλλά (ίσως) δεν χρησιμοποιεί πλήρως CPU

$$U = \sum_i \frac{C_i}{T_i}$$

Ικανότητα χρονοδρομολόγησης RM

- Όριο U με χρήση του RM για n διεργασίες:

$$U_{RM} = n(2^{1/n} - 1)$$

- όταν περίοδος ίση με προθεσμία!
- τείνει στο 69% (0.693)
- κάθε σύνολο διεργασιών με U μικρότερο του U_{RM} μπορεί να χρονοδρομολογηθεί επιτυχώς
 - *ικανή αλλά όχι αναγκαία συνθήκη!*
 - λεπτομερέστερος υπολογισμός;
 - αρμονικές περίοδοι διεργασιών $\Rightarrow U_{RM}=1$

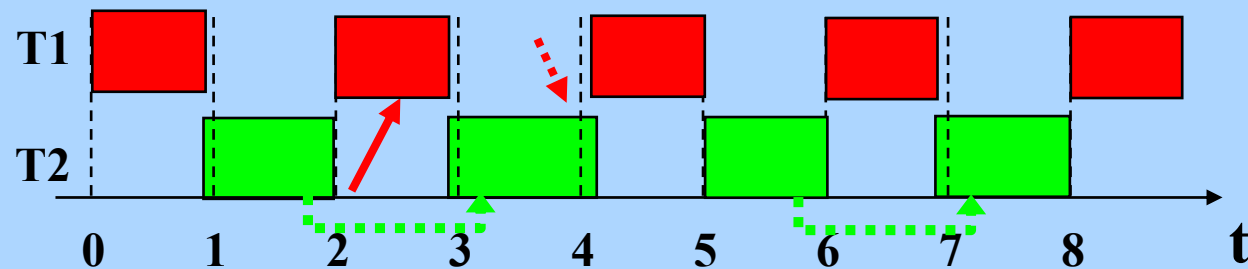
Δυναμική ανάθεση προτεραιοτήτων

- Όταν οι διεργασίες αλλάζουν “προτεραιότητα” κατά τη διάρκεια εκτέλεσης
 - καλύτερη χρησιμοποίηση επεξεργαστή
 - πολυπλοκότερη χρονοδρομολόγηση
 - μεγαλύτερος κώδικας
 - “ευριστικοί” αλγόριθμοι
 - λιγότερο προβλέψιμη συμπεριφορά σε υπερφόρτωση
- τρέχουσα προτεραιότητα υπολογίζεται από διάφορες παραμέτρους
 - προθεσμίες

Αλγόριθμος EDF

$U = 0.91$

$(C_1=0.9, T_1=2)$ $(C_2=2.3, T_2=5)$



- ***Earliest Deadline First (EDF)***
- “εκτέλεση της διεργασίας με την αμεσότερη (απόλυτη) προθεσμία”

Ικανότητα χρονοδρομολόγησης EDF

- “εάν υπάρχει τρόπος ορθής χρονοδρομολόγησης ομάδας διεργασιών, ο τρόπος που υπολογίζει ο EDF είναι ορθός”
- Κριτήριο δυνατότητας χρονοδρομολόγησης ομάδας n διεργασιών (**ικανή μόνο συνθήκη**) :

$$\sum_i \frac{C_i}{\min(T_i, D_i)} \leq 1$$

*πρόβλεψη και της περίπτωσης $T_i \neq D$
(αν $D_i = T_i$ τότε **ικανή και αναγκαία συνθήκη**)*

Σύγκριση μεθόδων χρονοδρομολόγησης

- **Fixed Priority**
 - μεγαλύτερη προβλεψιμότητα
 - μικρότερη χρησιμοποίηση CPU
 - γενικευμένη χρήση σε εμπορικά ΛΣ
- **Dynamic Priority**
 - καλύτερη χρησιμοποίηση CPU
 - ασταθής συμπεριφορά σε κατάσταση υπερφόρτωσης
 - υποστηρίζεται από νεώτερα ΛΣ