

Ανάλυση επιδόσεων
συστημάτων πραγματικού
χρόνου

Τι είναι διεργασία;

- Διεργασία είναι ένα εκτελούμενο πρόγραμμα
- Η διεργασία είναι ενεργός οντότητα
 - έχει κατάσταση (τιμές μεταβλητών, τιμές καταχωρητών, δεδομένα στη στοίβα, κώδικα που έχει εκτελεστεί κ.λπ.), πόρους που της έχουν κατανεμηθεί κ.λπ.
 - το πρόγραμμα είναι πλήρως στατικό
- Μία διεργασία απεικονίζεται σε ακριβώς ένα αρχείο προγράμματος
 - ένα πρόγραμμα μπορεί να εκτελείται από πολλές διεργασίες ταυτόχρονα – η κάθε μία ακολουθεί διαφορετική πορεία εκτέλεσης
- Μία διεργασία εκτελείται σειριακά
 - Το λειτουργικό σύστημα πολυπλέκει την εκτέλεσή τους και έχουμε την αίσθηση της παραλληλίας

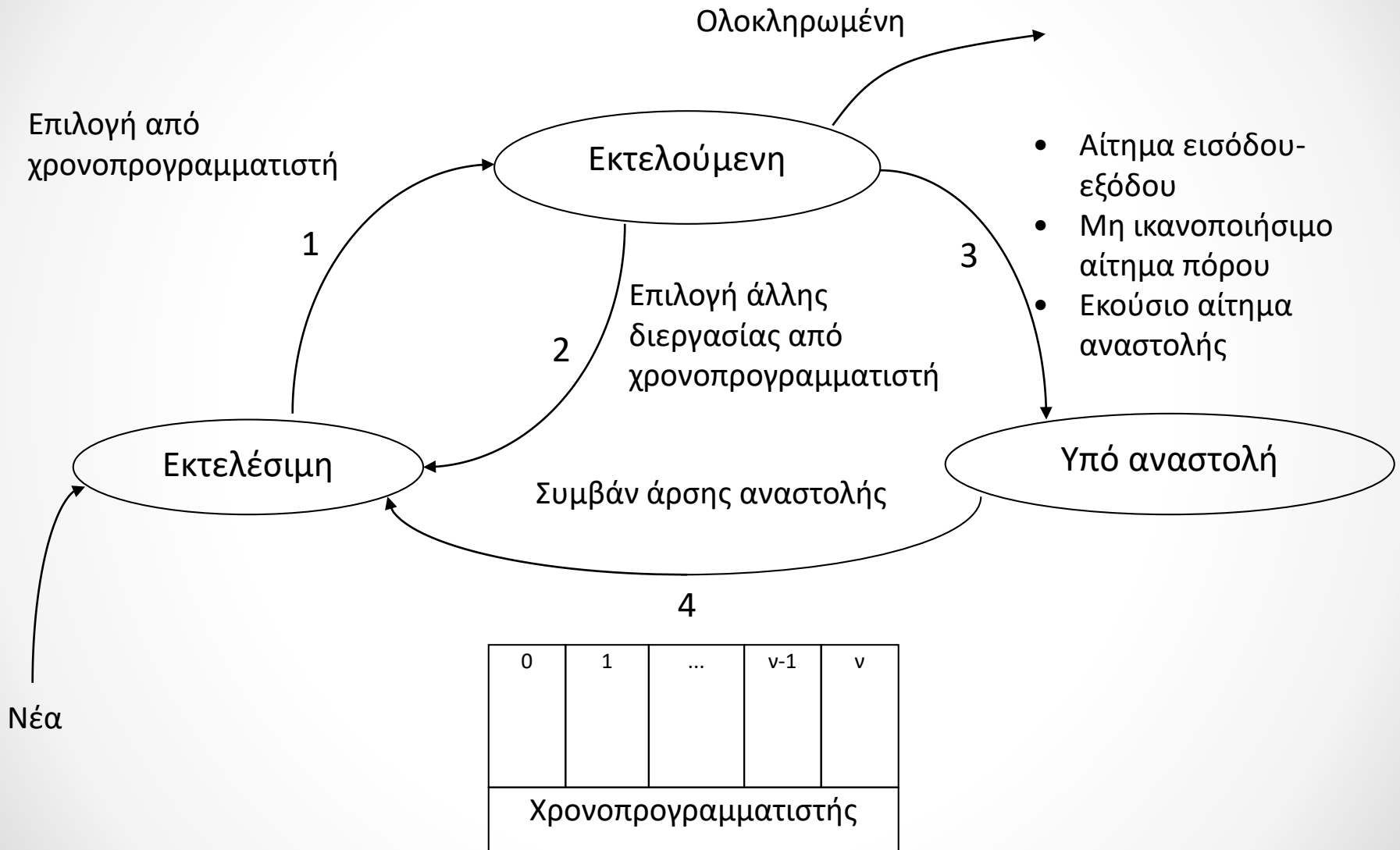


Καταστάσεις διεργασιών (1)

- Μία διεργασία μπορεί να είναι:
 - Νέα – μόλις έχει δημιουργηθεί
 - Εκτελούμενη – η κεντρική μονάδα επεξεργασίας εκτελεί δικές της εντολές
 - Εκτελέσιμη – θα μπορούσε να εκτελεί εντολές στην κεντρική μονάδα επεξεργασίας, αν αυτή της παραχωρούνταν
 - Υπό αναστολή – η διεργασία είναι εν αναμονή κάποιου συμβάντος
 - Τερματισμένη – έχει ολοκληρώσει την εκτέλεσή της



Καταστάσεις διεργασιών (2)



Χρονοπρογραμματισμός διεργασιών

- Σε πολυπρογραμματιζόμενα συστήματα είναι δυνατόν πάνω από μία διεργασίες να είναι εκτελέσιμες
- Το λειτουργικό σύστημα πρέπει να αποφασίσει ποια από αυτές θα καταλάβει την ΚΜΕ
- Το τμήμα του Λ.Σ. που είναι επιφορτισμένο με αυτό το καθήκον καλείται *χρονοπρογραμματιστής* και εφαρμόζει έναν *αλγόριθμο χρονοπρογραμματισμού*



Χρονοπρογραμματισμός σε συστήματα πραγματικού χρόνου

- Στα συστήματα πραγματικού χρόνου, ο χρονοπρογραμματιστής πρέπει να:
 - Υποστηρίζει πολυεπεξεργασία/πολυνηματισμό με δυνατότητα προεκτόπισης
 - Χειρίζεται πολλαπλά επίπεδα διακοπών (interrupts)
 - να μπορεί να διακόψει την εξυπηρέτηση ενός σήματος διακοπής για να μεταβεί στην εξυπηρέτηση ενός άλλου σήματος που θεωρείται πιο επείγον
 - Υποστηρίζει προτεραιότητες νημάτων
 - Παρέχει προβλέψιμους μηχανισμούς συγχρονισμού νημάτων
 - Να υπάρχει δυνατότητα ακριβούς εκτίμησης του χρόνου συγχρονισμού / αποκλεισμού
 - Μεριμνά ιδιαίτερα για την τήρηση των χρονικών προθεσμιών



Συμπεριφορά διεργασιών

- Σχεδόν όλες οι διεργασίες εναλλάσσονται μεταξύ επεξεργασίας στην ΚΜΕ και αιτήσεις εισόδου-εξόδου (από δίσκο, δίκτυο, πληκτρολόγιο κ.λπ.)
 - εκτέλεση μερικών εντολών, κατόπιν αναμονή για ανάγνωση/εγγραφή, εκτέλεση κι άλλων εντολών κ.λπ.
 - κάποιες μορφές εισόδου-εξόδου είναι ουσιαστικά χρόνος Κ.Μ.Ε. (π.χ. ενημέρωση μνήμης οθόνης)
 - Αναφορικά με τον χρονοπρογραμματισμό, ως είσοδο-έξοδο θεωρούμε τις περιπτώσεις που δεν απασχολείται η ΚΜΕ
- Ειδικότερα στα συστήματα πραγματικού χρόνου, οι κρίσιμες διεργασίες απαιτούν μικρό χρόνο κατάληψης της ΚΜΕ, αλλά τον απαιτούν *άμεσα*



Πότε χρονοπρογραμματίζουμε; (1)

- Υπάρχουν πολλές περιπτώσεις όπου μπορούμε να λάβουμε μία απόφαση χρονοπρογραμματισμού
 - όταν δημιουργείται μία νέα διεργασία – θα συνεχίσουμε να εκτελούμε τη γονική ή θα μεταχθούμε στη θυγατρική;
 - όταν μία διεργασία τερματίζει – πρέπει να επιλεχθεί κάποια από τις εκτελέσιμες ή η διεργασία αδράνειας
 - όταν μία διεργασία αναστέλλεται για είσοδο/έξοδο, σε σημαφόρο κ.λπ.
 - η αιτία αναστολής ίσως είναι σημαντική για την απόφαση χρονοπρογραμματισμού – π.χ. αν η A ανασταλεί περιμένοντας έναν πόρο που κατέχει η B, η B είναι καλός υποψήφιος για εκτέλεση.
 - Ο χρονοπρογραμματιστής δεν έχει όμως πάντα όλες τις πληροφορίες



Πότε χρονοπρογραμματίζουμε; (1)

- Όταν λάβει χώρα μία διακοπή εισόδου/εξόδου, καθώς πιθανότατα ολοκληρώνεται μία λειτουργία που είχε οδηγήσει μία διεργασία σε αναστολή
- Με την έλευση ενός σήματος διακοπής (χρονισμού ή άλλο)
 - Στα ΣΠΧ, η έλευση σήματος διακοπής μπορεί να οδηγεί στη δημιουργία νέας διεργασίας
 - Τα σήματα χρονισμού έχουν ιδιαίτερη αντιμετώπιση στον προεκχωρητικό (preemptive) χρονοπρογραμματισμό



Χρονοπρογραμματισμός σε συστήματα χωρίς χαρακτηριστικά πραγματικού χρόνου

- Πρώτη ερχόμενη-Πρώτη εξυπηρετούμενη (FCFS/FIFO)



- Πλεονεκτήματα

- πολύ εύκολος στην κατανόηση
- απλούστατος στην υλοποίηση
- δίκαιος, με την έννοια ότι όλοι εξυπηρετούνται, ο καθένας με τη σειρά του

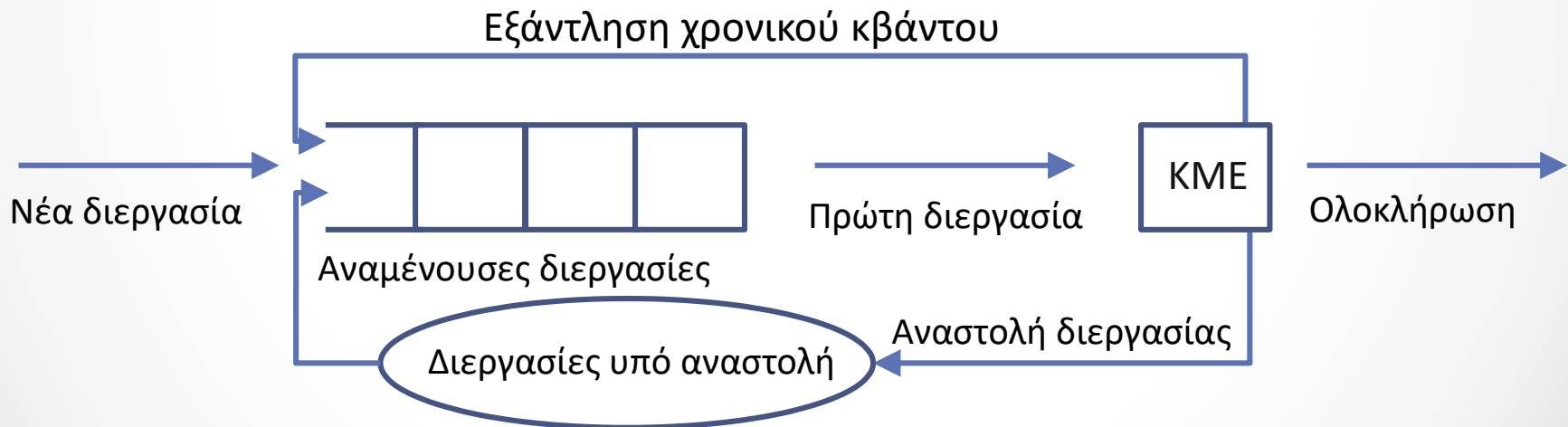
- Μειονεκτήματα

- Μεγάλη ατυχία αν μία μικρή διεργασία (ή περισσότερες) τοποθετηθεί πίσω από μία μεγάλη
 - π.χ. 100 διεργασίες εισόδου-εξόδου πίσω από μία μεγάλη διεργασία που τρέχει 3 ημέρες
- Όχι καλή αξιοποίηση των πόρων του συστήματος
- **Απολύτως ακατάλληλος για συστήματα πραγματικού χρόνου**



Χρονοπρογραμματισμός σε συστήματα χωρίς χαρακτηριστικά πραγματικού χρόνου

- *Εκ περιτροπής εξυπηρέτηση (round robin)*
 - απλός, δίκαιος και ιδιαίτερα διαδεδομένος
 - Σε κάθε διεργασία αποδίδεται ένα *χρονικό κβάντο* – μέγιστος χρόνος συνεχούς εκτέλεσής της
 - Αν η διεργασία εξαντλήσει το κβάντο ή ανασταλεί πριν το εξαντλήσει λαμβάνει χώρα μεταγωγή διεργασιών



Ακατάλληλος για συστήματα πραγματικού χρόνου

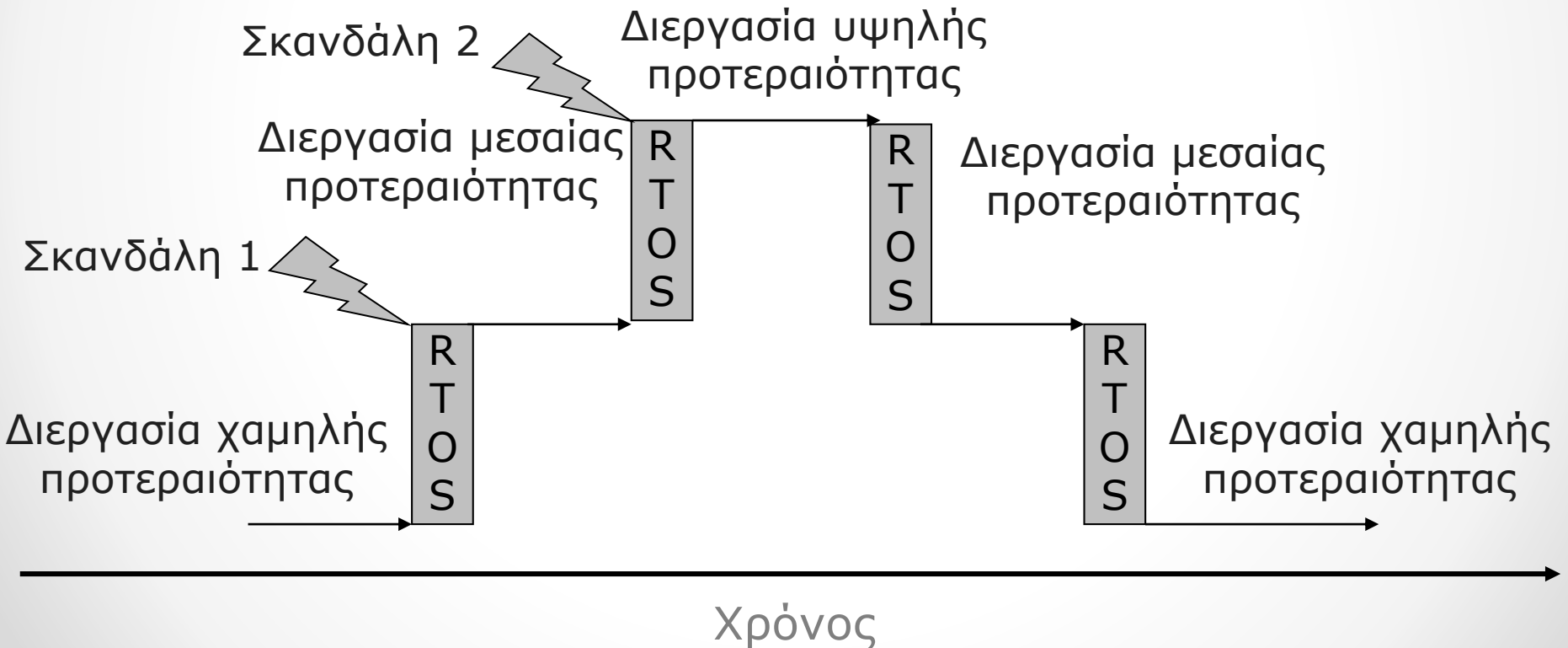


Χρονοπρογραμματισμός σε λειτουργικά συστήματα πραγματικού χρόνου

- Δύο κύριες κατηγορίες
 - Συστήματα καθοδηγούμενα από σήματα διακοπής
 - το κύριο πρόγραμμα είναι ένας άεργος βρόχος (idle loop)
 - Οι διάφορες διεργασίες στο σύστημα χρονοπρογραμματίζονται μέσω διακοπών είτε υλικού είτε λογισμικού
 - η αποστολή (dispatching) πραγματοποιείται από τις *ρουτίνες εξυπηρέτησης διακοπών* (interrupt service routine)
 - Όταν χρησιμοποιούνται σήματα διακοπής σε επίπεδο υλικού, ένα ρολόι ή άλλη εξωτερική συσκευή δίνει σήματα διακοπών που κατευθύνονται σε έναν ελεγκτή διακοπών. Ο ελεγκτής διακοπών παράγει σήματα διακοπών, ανάλογα με τη σειρά άφιξης και τη προτεραιότητα των εμπλεκόμενων σημάτων που φθάνουν από τις συσκευές

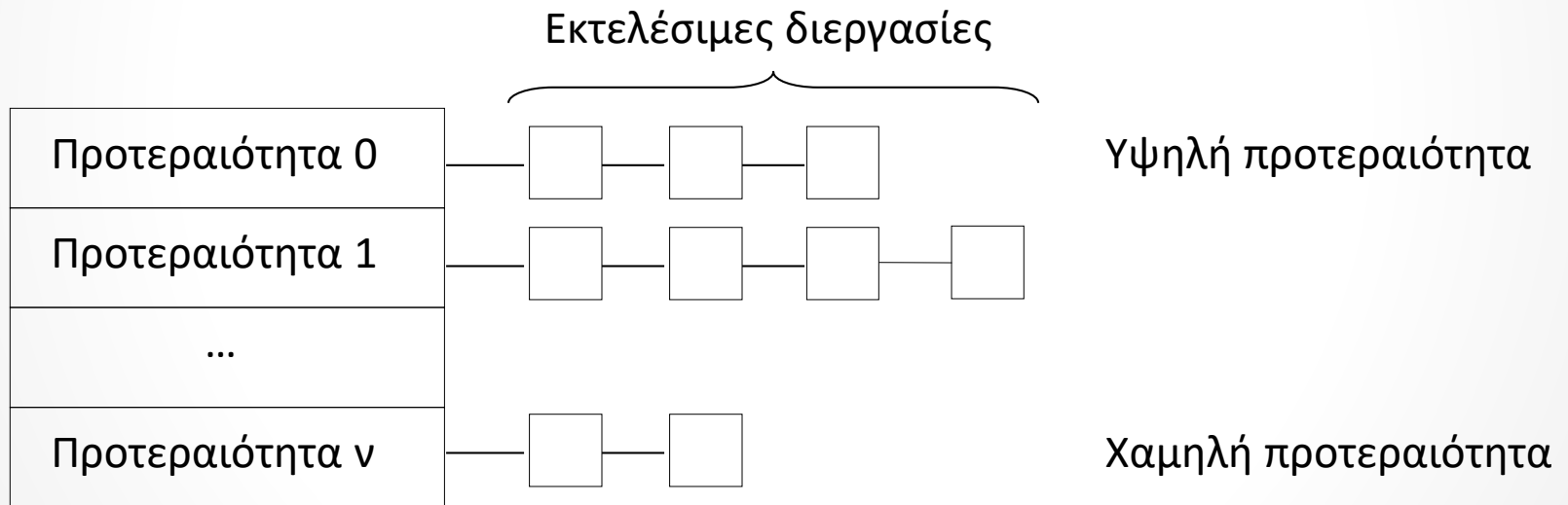


Σχηματική παράσταση



πολλαπλές ουρές

- Χρονοπρογραμματισμός προτεραιοτήτων
 - Διάταξη διεργασιών σε κλάσεις προτεραιοτήτων – εντός κάθε κλάσης εκ περιτροπής εξυπηρέτηση, απόλυτη προτεραιότητα της κλάσης i έναντι των επομένων της

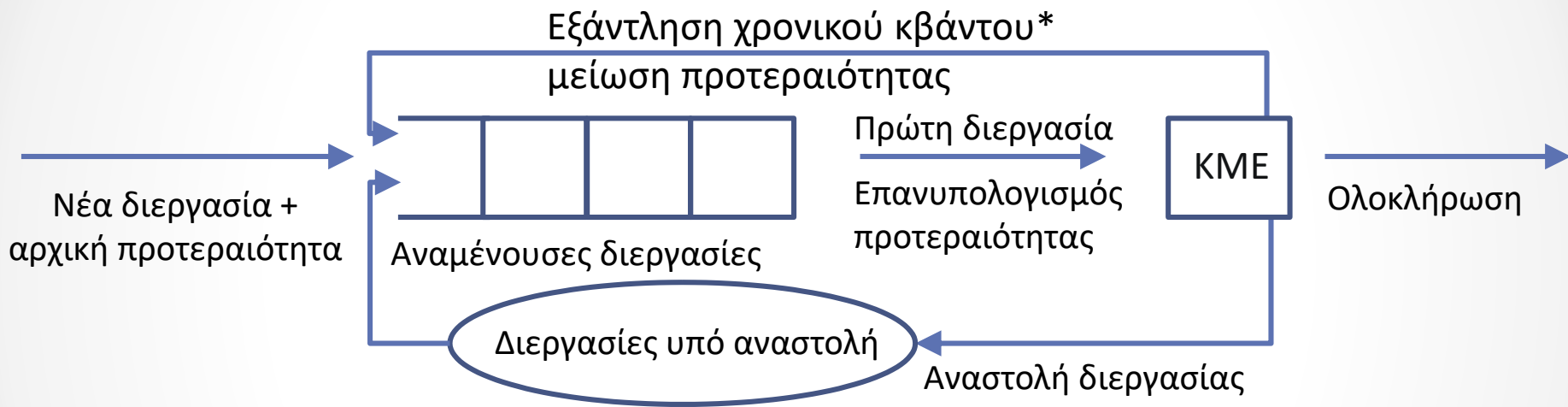


ενιαία ουρά

- Χρονοπρογραμματισμός προτεραιοτήτων
 - Οι προτεραιότητες μπορεί να είναι στατικές ή δυναμικές
 - π.χ.
 - στατική: στρατηγοί = 100, λοχαγοί = 40, στρατιώτες = 10
 - στατική: επιλογή χρήστη λόγω τιμολογιακών πολιτικών σε υπολογιστικά κέντρα (υψηλή προτεραιότητα = 10 ευρώ/ώρα, χαμηλή = 5 ευρώ/ώρα)
 - δυναμική: διεργασία που αξιοποίησε μόνο το μισό της κβάντο και κατόπιν ανεστάλη, λαμβάνει υψηλή δυναμική προτεραιότητα
 - δυναμική: διεργασία που συχνά αναστέλλεται, λαμβάνει υψηλή δυναμική προτεραιότητα
 - δυναμική: διεργασία με κοντινή προθεσμία λαμβάνει υψηλή προτεραιότητα



Χρονοπρογραμματισμός προτεραιοτήτων με ενιαία ουρά - Σχηματικά



Εξειδικεύσεις των κριτηρίων: Χρονοπρογραμματισμός μονοτονικού ρυθμού (1)

- Είναι ο πιο γνωστός αλγόριθμος με στατικές προτεραιότητες
- Χρησιμοποιείται υπό τις ακόλουθες συνθήκες:
 - όλες οι διεργασίες είναι περιοδικές,
 - όλες οι διεργασίες αποδεσμεύονται στην αρχή της περιόδου τους, ενώ πρέπει να ολοκληρώνονται μέσα στην περίοδό τους,
 - δεν υπάρχουν εξαρτήσεις μεταξύ των διεργασιών,
 - κάθε διεργασία δ_i απαιτεί την ίδια ποσότητα χρόνου CPU σε κάθε εκτέλεσή της ή έστω ποσότητα χρόνου κάτω από ένα συγκεκριμένο όριο. Η ποσότητα (ή το όριο) χρόνου είναι μικρότερο ή ίσο της περιόδου τους,
 - καμία διεργασία δεν μπορεί να αναστείλει τον εαυτό της (οικειοθελώς),
 - η χρονοδρομολόγηση των διεργασιών είναι προεκτοπιστική, με δυνατότητα προεκτόπισης οποιασδήποτε διεργασίας,
 - Οι επιβαρύνσεις από το χρονοπρογραμματισμό είναι μηδενικές
 - Το σύστημα διαθέτει έναν μόνον επεξεργαστή.



Εξειδικεύσεις των κριτηρίων: Χρονοπρογραμματισμός μονοτονικού ρυθμού (2)

- Στον αλγόριθμο μονοτονικού ρυθμού, κάθε διεργασία αντιστοιχίζεται με μία προτεραιότητα
 - Η προτεραιότητα είναι αύξουσα με τη συχνότητα της διεργασίας (και άρα φθίνουσα με την περίοδο)
 - Κατά τον χρόνο εκτέλεσης, ο χρονοπρογραμματιστής εκτελεί πάντα την έτοιμη διεργασία με την υψηλότερη προτεραιότητα, διακόπτοντας τη διεργασία που εκτελείται, αν είναι χαμηλότερης προτεραιότητας.
 - Αν δύο ή περισσότερες διεργασίες έχουν την ίδια περίοδο, ο RM επιλέγει τυχαία την επόμενη προς εκτέλεση διεργασία



Εξειδικεύσεις των κριτηρίων: Χρονοπρογραμματισμός μονοτονικού ρυθμού (3)

- Ο χρονοπρογραμματισμός μονοτονικού ρυθμού είναι βέλτιστος, υπό την έννοια ότι αν ένα σύνολο διεργασιών με στατικές προτεραιότητες είναι εφικτό να χρονοπρογραμματιστεί με οποιοδήποτε τρόπο ανάθεσης προτεραιοτήτων, θα είναι εφικτό να χρονοπρογραμματιστεί και με τον RM
 - ένα σύνολο n ανεξάρτητων και περιοδικών διεργασιών σε έναν επεξεργαστή, όπου οι προθεσμίες είναι μεγαλύτερες ή ίσες από τις περιόδους τους και όπου χρησιμοποιείται προεκτοπιστικός χρονοπρογραμματισμός είναι εφικτό αν και μόνο αν

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \leq 1$$



Εξειδικεύσεις των κριτηρίων: Χρονοπρογραμματισμός μονοτονικού ρυθμού (4)

- Επίσης έχει αποδειχθεί ότι, για κάθε σύστημα με περιοδικές διεργασίες, ο RM μπορεί να χρονοπρογραμματίσει τις διεργασίες, αν ισχύει η σχέση

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \leq n * \left(2^{\frac{1}{n}} - 1\right)$$

$$[\sigma\sigma. \lim_{n \rightarrow \infty} n * \left(2^{\frac{1}{n}} - 1\right) = \ln(2)]$$

- Αν η περίοδος κάθε διεργασίας είναι ακέραιο πολλαπλάσιο της επόμενης μεγαλύτερης περιόδου, ο χρονοπρογραμματισμός μπορεί να εγγυηθεί 100% αξιοποίηση της CPU



Εξειδικεύσεις των κριτηρίων: Αλγόριθμος μονοτονικής προθεσμίας

- Αλγόριθμος μονοτονικής προθεσμίας (deadline-monotonic priority assignment, DM)
 - Αντίστοιχος με τον RM, επιτρέπει όμως οι διεργασίες να έχουν συντομότερες προθεσμίες σε σχέση με την περίοδό τους, δηλ.
 $C_i \leq D_i \leq T_i$
 - Αν $D_i = T_i \forall i$, τότε $DM \equiv RM$



Εξειδικεύσεις των κριτηρίων: αλγόριθμοι EDF και LLF

- EDF (earliest deadline first, συντομότερη προθεσμία πρώτα)
 - Οι διεργασίες κατατάσσονται με βάση την προθεσμία που έχουν, και επιλέγεται αυτή με τη συντομότερη προθεσμία
- LLF (least laxity first, ελάχιστη σχετική προθεσμία πρώτα)
 - Οι διεργασίες κατατάσσονται με βάση τη σχετική προθεσμία τους $\sigma\pi_i = (\pi_i - \tau) - κμ\epsilon_i$ όπου:
 - π_i είναι η προθεσμία της διεργασίας i
 - τ είναι η τρέχουσα χρονική στιγμή
 - $κμ\epsilon_i$ είναι ο χρόνος ΚΜΕ χρειάζεται η διεργασία μέχρι να ολοκληρωθεί



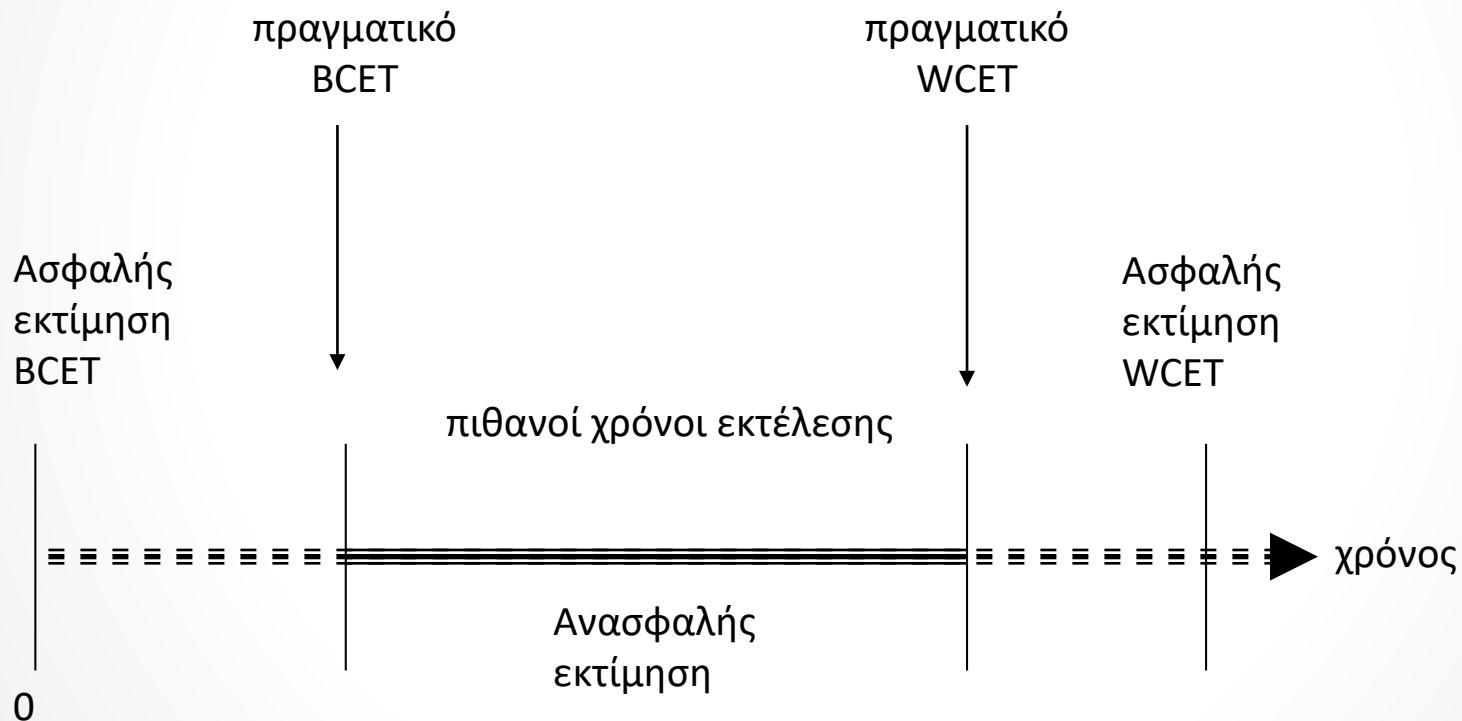
Υπολογισμός χρόνου εκτέλεσης διεργασιών (1)

- Στους περισσότερους αλγόριθμους, απαιτείται να γνωρίζουμε τον χρόνο εκτέλεσης των διεργασιών
 - Η πιο ευρέως χρησιμοποιούμενη σχετική μετρική είναι η WCET (worst case execution time).
 - Βασίζεται στη στατική ανάλυση του λογισμικού
 - Η εκτίμηση του χρόνου χειρότερης περίπτωσης υπολογίζεται δομώντας τις διεργασίες σε έναν γράφο και υπολογίζοντας τον χρόνο εκτέλεσης του μακρύτερου μονοπατιού (longest path) που μπορεί να ακολουθήσει η εκτέλεση του λογισμικού
 - Ο υπολογισμός του WCET περιπλέκεται από τις ιδιαιτερότητες της αρχιτεκτονικής (σωλήνωση, παραλληλία, κρυφή μνήμη, πρόβλεψη διακλαδώσεων κ.ο.κ.)
 - Κατ' αντιστοιχία ορίζεται και το BCET (best-case execution time)



Υπολογισμός χρόνου εκτέλεσης διεργασιών (2)

- Σχέση BCET, WCET



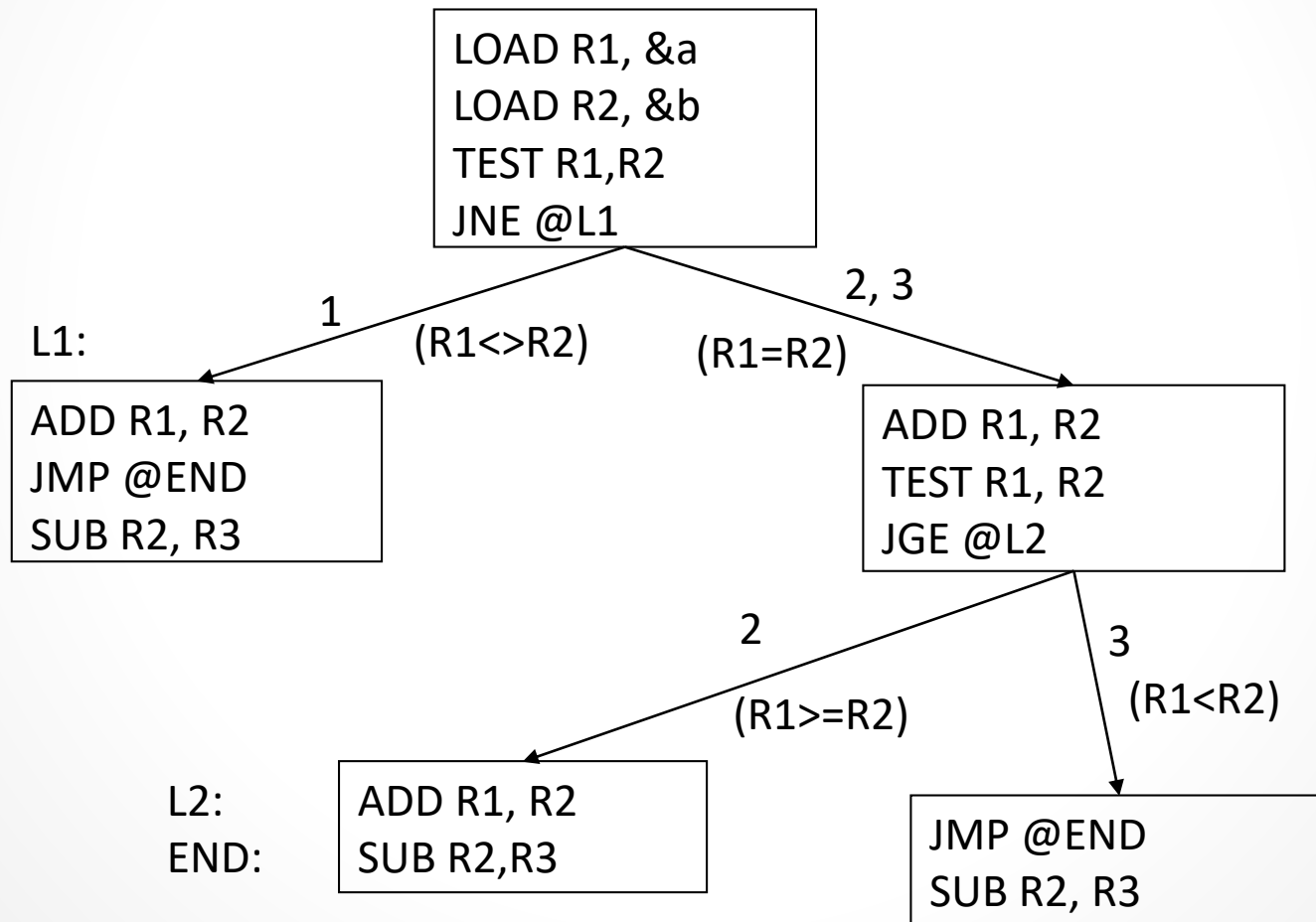
Υπολογισμός χρόνου εκτέλεσης διεργασιών (3)

- Ας θεωρήσουμε το ακόλουθο παράδειγμα κώδικα συμβολικής γλώσσας:

```
LOAD    R1,&a    ; R1 <-- περιεχόμενα της θέσης μνήμης "a"
LOAD    R2,&b    ; R2 <-- περιεχόμενα της θέσης μνήμης "b"
TEST    R1,R2    ; σύγκριση R1 και R2, τίθεται η σχετική ένδειξη
JNE     @L1      ; αν δεν προέκυψε ισότητα, άλμα στην ετικέτα L1
ADD     R1,R2    ; R1 <-- R1 + R2
TEST    R1,R2    ; σύγκριση R1 και R2, τίθεται η σχετική ένδειξη
JGE     @L2      ; αν R1 >= R2, άλμα στην ετικέτα L2
JMP     @END     ; άλμα στην ετικέτα END
L1:     ADD     R1, R2    ; R1 <-- R1 + R2
        JMP     @END     ; άλμα στην ετικέτα END
L2:     ADD     R1, R2    ; R1 <-- R1 + R2
END:    SUB     R2, R3    ; R2 <-- R2 - R3
```

Υπολογισμός χρόνου εκτέλεσης διεργασιών (4)

- Κατασκευάζουμε το δέντρο του προγράμματος στο οποίο απαριθμούνται όλα τα πιθανά μονοπάτια εκτέλεσης



Υπολογισμός χρόνου εκτέλεσης διεργασιών (5)

- Αν ο χρόνος εκτέλεσης μιας εντολής είναι 6μsec (ίδιος για κάθε εντολή), τότε:
 - για το 1^ο μονοπάτι ο χρόνος εκτέλεσης είναι 42μsec (περιλαμβάνει 7 εντολές)
 - και για το 2^ο και το 3^ο μονοπάτι είναι 54 μsec (περιλαμβάνουν 9 εντολές).
 - Επομένως υπολογίστηκε ότι ο WCET είναι 54 μsec

