

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός

Ενότητα # 8: Ταυτοχρονισμός και νήματα

Διδάσκων: Γεώργιος Ξυλωμένος

Τμήμα: Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Οικονομικό Πανεπιστήμιο Αθηνών**» 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



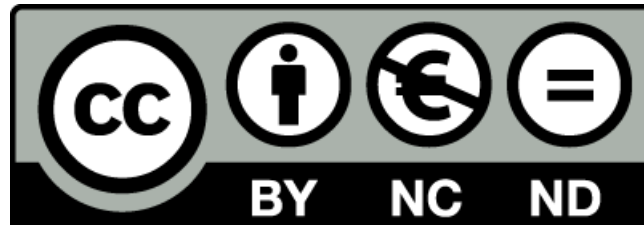
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Οι εικόνες προέρχονται από το βιβλίο «Κατανεμημένα Συστήματα με Java», Ι. Κάβουρας, Ι. Μήλης, Γ. Ξυλωμένος, Α. Ρουκουνάκη, 3<sup>η</sup> έκδοση, 2011, Εκδόσεις Κλειδάριθμος.



# Σκοποί ενότητας

- Εξοικείωση με την έννοια των νημάτων και εξήγηση της σχέσης τους με τις διεργασίες.
- Εισαγωγή στους μηχανισμούς διαχείρισης και συγχρονισμού νημάτων της Java.

# Περιεχόμενα ενότητας

- Διεργασίες και νήματα
- Υλοποίηση νημάτων
- Αξιοποίηση των νημάτων
- Νήματα στην Java
- Συγχρονισμός νημάτων

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Διεργασίες και νήματα

**Μάθημα:** Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,  
**Ενότητα # 8:** Ταυτοχρονισμός και νήματα

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο

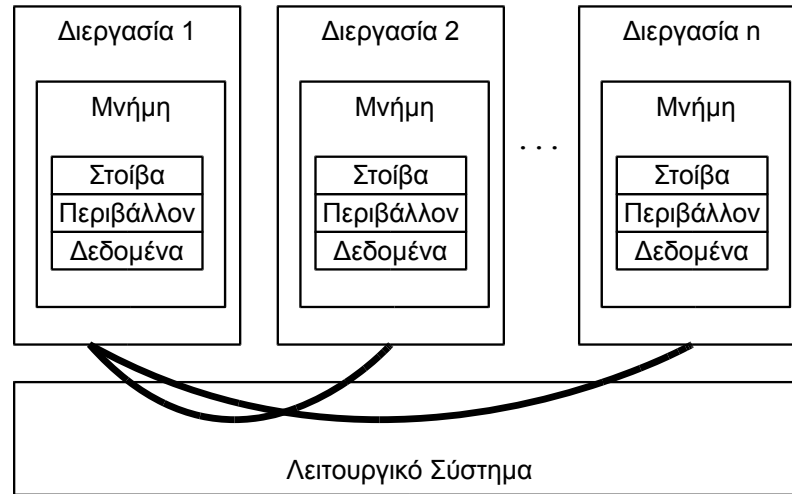


ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

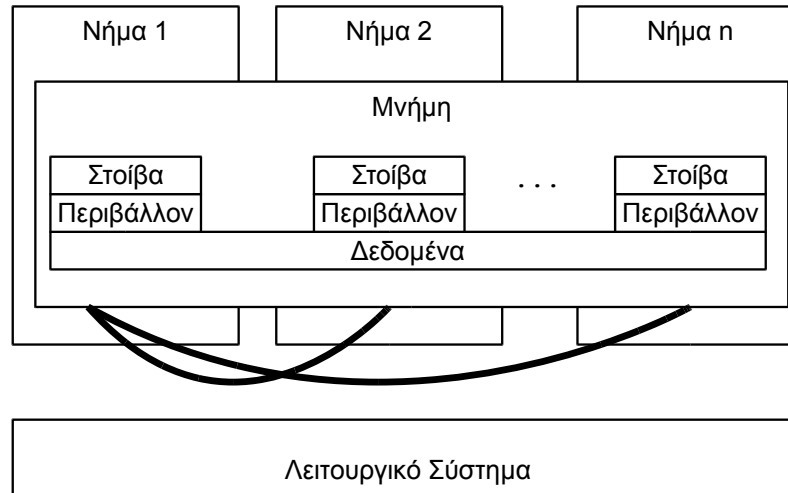


# Διεργασίες



- Διεργασία: εκτελούμενο πρόγραμμα
  - Κώδικας + κατάσταση (καταχωρητές + μνήμη)
  - Ψευδοταυτόχρονη/παράλληλη εκτέλεση διεργασιών
  - Απομόνωση διεργασιών για λόγους ασφάλειας
  - Κόστος δημιουργίας/μεταγωγής συμφραζομένων

# Νήματα



- Νήμα: μια ροή ελέγχου σε μια διεργασία
  - Επιτρέπονται πολλά νήματα ανά διεργασία
  - Όλα τα νήματα μοιράζονται την ίδια μνήμη
  - Χωριστή στοίβα, καταχωρητές, μετρητής
  - Ψευδοταυτόχρονη/παράλληλη εκτέλεση νημάτων



# Γιατί νήματα;

- Κοινός χώρος διευθύνσεων σε μία διεργασία
  - Δεν υπάρχει μηχανισμός προστασίας
- Συνεργασία νημάτων μέσω κοινής μνήμης
  - Πολύ γρήγορη επικοινωνία μέσω της μνήμης
- Χαμηλό κόστος μεταγωγής συμφραζομένων
  - Μόνο οι καταχωρητές αλλάζουν
  - Η εικονική μνήμη παραμένει ίδια

# Χρήσεις νημάτων

- Παράλληλη εκτέλεση πολλών βημάτων
  - Χωριστά νήματα για κλήσεις εισόδου/εξόδου
  - Επικοινωνία με χρήστη και επεξεργασία
- Αξιοποίηση παραλληλισμού εφαρμογής
  - Χρήση όλων των διαθέσιμων επεξεργαστών
    - Μικρό κόστος μεταγωγής από νήμα σε νήμα
  - Γρήγορη επικοινωνία μέσω κοινής μνήμης
    - Πολύ μικρότερη επιβάρυνση από τις διεργασίες

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Υλοποίηση νημάτων

**Μάθημα:** Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,  
**Ενότητα # 8:** Ταυτοχρονισμός και νήματα

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



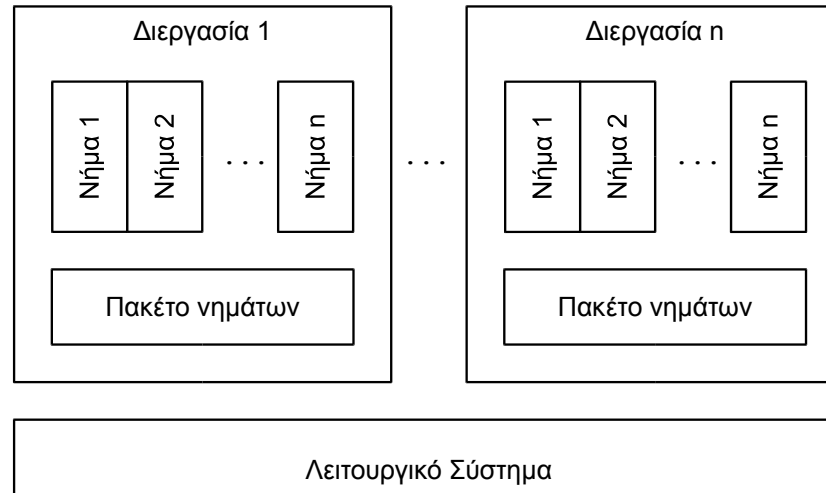
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



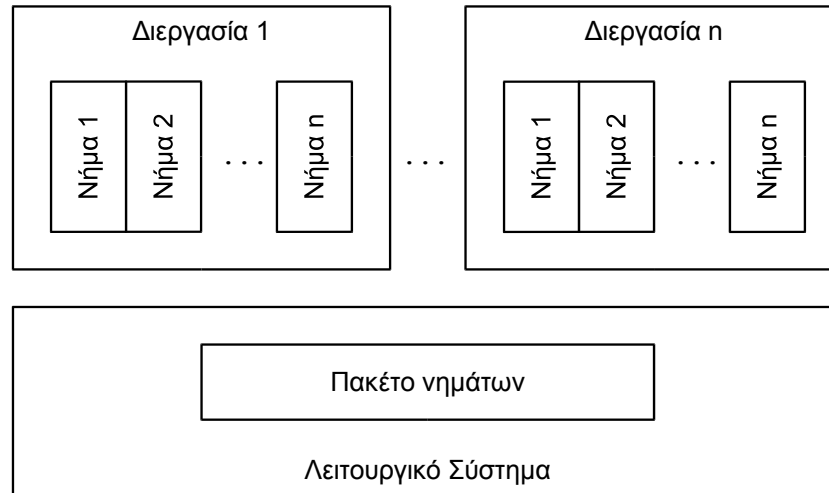
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Νήματα επιπέδου χρήστη



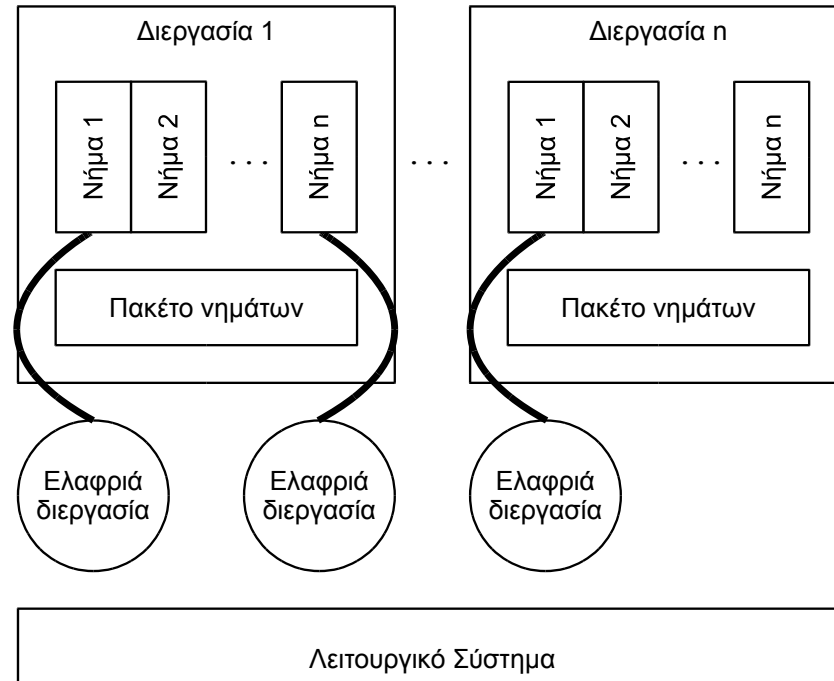
- Πακέτο νημάτων
  - Δημιουργία και καταστροφή νημάτων
  - Διαχείριση συνθηκών (conditions)
- Πακέτα νημάτων επιπέδου χρήστη (μοντέλο N:1)
  - Οικονομική υλοποίηση λειτουργιών και μεταγωγής
  - Ο εμποδισμός ισχύει για ολόκληρη τη διεργασία

# Νήματα επιπέδου πυρήνα



- Πακέτα νημάτων επιπέδου πυρήνα (μοντέλο 1:1)
  - Ακριβή υλοποίηση λειτουργιών και μεταγωγή
  - Ο εμποδισμός ισχύει μόνο για το νήμα
  - Απαιτεί τροποποίηση του λειτουργικού συστήματος
- Επίπεδο χρήστη ή επίπεδο πυρήνα;

# Ελαφριές διεργασίες (1 από 4)



- Υβριδική λύση (μοντέλο N:M)
  - Μεταβαλλόμενη απεικόνιση νημάτων σε διεργασίες
  - Βασική υποστήριξη νημάτων από πυρήνα
  - Πακέτο νημάτων επιπέδου χρήστη

# Ελαφριές διεργασίες (2 από 4)

- Επίπεδο πυρήνα
  - Πολλές ελαφριές διεργασίες σε μία κανονική
    - (Κατά)μερισμός του ενιαίου χώρου διευθύνσεων
- Επίπεδο χρήστη
  - Δημιουργία, διαγραφή, συγχρονισμός νημάτων
  - Ένα νήμα εκτελείται από μία ελαφριά διεργασία
    - Η συσχέτιση μεταξύ τους δεν είναι μόνιμη
    - Εξαρτάται από την πορεία εκτέλεσης

# Ελαφριές διεργασίες (3 από 4)

- Μεταγωγή συμφραζομένων: επίπεδο χρήστη
  - Εμποδισμός νήματος πάνω σε συνθήκη
  - Κλήση χρονοπρογραμματιστή επιπέδου χρήστη
  - Ανάθεση νέου νήματος από χρονοπρογραμματιστή
- Μεταγωγή συμφραζομένων: επίπεδο πυρήνα
  - Κλήση πυρήνα που προκαλεί εμποδισμό
  - Ο πυρήνας δίνει τον έλεγχο σε άλλη διεργασία
    - Δε χρειάζεται κάποια ειδική μεταχείριση
  - Αρκετές ελαφριές διεργασίες για παραλληλισμό



# Ελαφριές διεργασίες (4 από 4)

- Ιδιότητες ελαφριών διεργασιών
  - Δεν συνδέονται μόνιμα με ένα νήμα
    - Ο χρονοπρογραμματιστής τους αναθέτει νήματα δυναμικά
  - Εμποδισμός σε συνθήκες σε επίπεδο χρήστη
- Πλεονεκτήματα ελαφριών διεργασιών
  - Όλες οι λειτουργίες νημάτων σε επίπεδο χρήστη
    - Ο πυρήνας παρεμβαίνει μόνο στις κλήσεις του
  - Ο χρονοπρογραμματιστής μπορεί να αλλάξει

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Αξιοποίηση των νημάτων

**Μάθημα:** Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,  
**Ενότητα # 8:** Ταυτοχρονισμός και νήματα

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

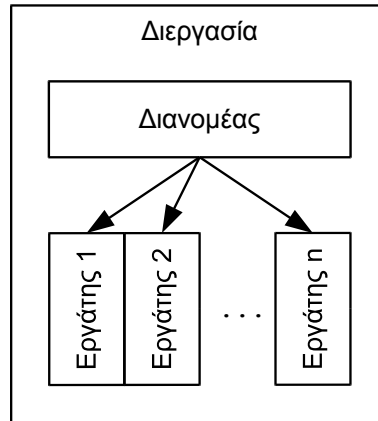
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



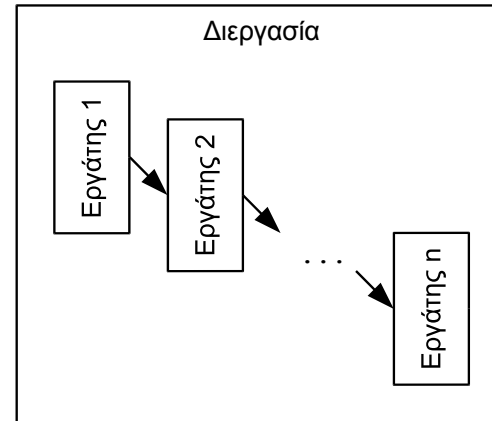
# Εφαρμογές νημάτων

- Αξιοποίηση στους πελάτες
  - Απόκρυψη καθυστερήσεων του δικτύου
  - Παράδειγμα: φυλλομετρητής ιστοσελίδων
- Αξιοποίηση στους εξυπηρετητές
  - Διαχείριση πολλών πελατών
  - Αξιοποίηση πολλών επεξεργαστών
  - Παράδειγμα: εξυπηρετητής αρχείων
    - Πολλά νήματα για εξυπηρέτηση αιτήσεων
    - (Κατά)μερισμός κρυφής μνήμης ενταμιευτών

# Μοντέλα οργάνωσης



(α)



(β)

- Μοντέλο διανομέα/εργατών: εξυπηρετητής αρχείων
  - Όλα τα νήματα-εργάτες είναι παρόμοια
  - Εξυπηρέτηση αιτήσεων από οποιοδήποτε νήμα
- Μοντέλο σωλήνωσης: εκτέλεση εργασίας σε στάδια
  - Κάθε νήμα είναι διαφορετικό
  - Ιδανικά, όλα τα νήματα απαιτούν τον ίδιο χρόνο

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**

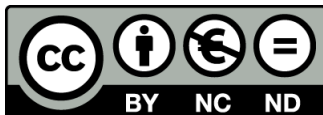


**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Νήματα στην Java

**Μάθημα:** Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,  
**Ενότητα # 8:** Ταυτοχρονισμός και νήματα

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

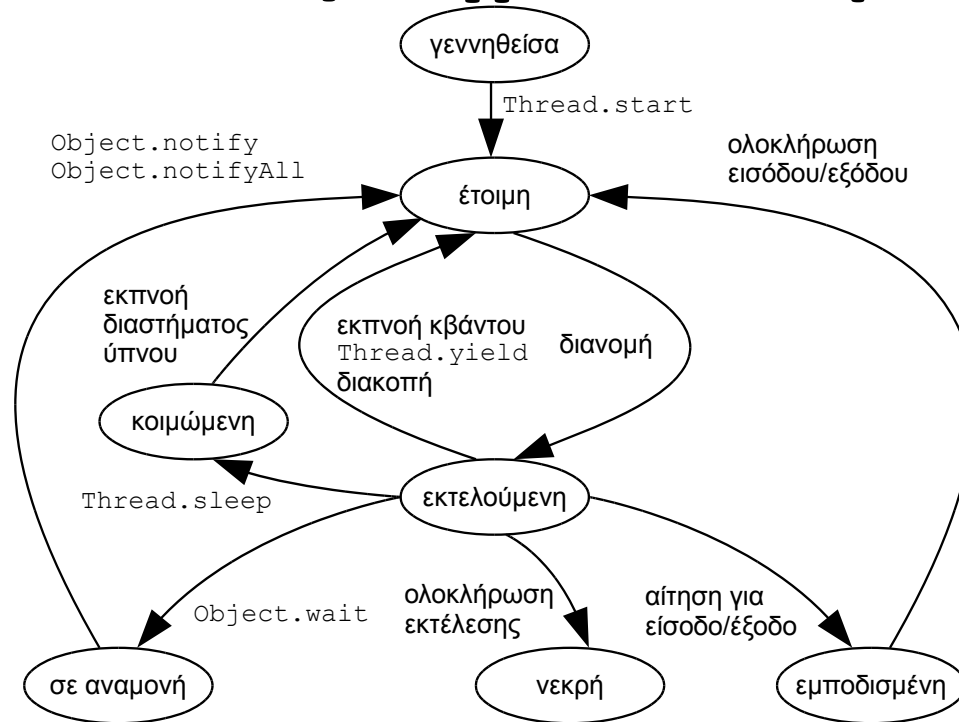


ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Οργάνωση νημάτων

- Τα νήματα είναι μονάδες εκτέλεσης
  - Κάθε διεργασία (πρόγραμμα) έχει νήματα
  - Κάθε νήμα εκτελείται ανεξάρτητα
  - Όλα τα νήματα μοιράζονται την ίδια μνήμη
    - Κώδικα και όλα τα δεδομένα
- Κάθε εφαρμογή ξεκινά με ένα νήμα
  - Εκτελεί τη μέθοδο `main` της εφαρμογής
  - Το νήμα αυτό μπορεί να δημιουργήσει άλλα

# Καταστάσεις νημάτων (1 από 2)



- Καταστάσεις νημάτων

- Έτοιμη/εκτελούμενη ανάλογα με χρονοπρογραμματιστή
- Κοιμώμενη/εμποδισμένη/σε αναμονή ανάλογα με εντολές

# Καταστάσεις νημάτων (2 από 2)

- Διάφοροι τρόποι διακοπής εκτέλεσης
  - Αναμονή εισόδου / εξόδου -> εμποδισμένη
  - Αναμονή χρονομέτρου (sleep) -> κοιμώμενη
  - Αναμονή γεγονότος (wait) -> σε αναμονή
- Προτεραιότητες νημάτων
  - Προτεραιότητα 1 (χαμηλή) έως 10 (υψηλή)
    - Η προτεραιότητα κληρονομείται από τον γονέα
  - Χρονοπρογραμματισμός με βάση προτεραιότητα
  - Διακοπτόμενος ή μη χρονοπρογραμματισμός
    - Εξαρτάται από την πλατφόρμα της Java



# Μέθοδοι νημάτων (1 από 6)

- Νήματα: Τάξη Thread
- Thread (String όνομα\_νήματος)
  - Κατασκευάζει νήμα με συγκεκριμένο όνομα
- Thread ()
  - Κατασκευάζει νήμα με όνομα "Thread-#"
- void start()
  - Ξεκινά την εκτέλεση νήματος (μέθοδος run())
    - Τα δύο νήματα εκτελούνται παράλληλα

# Μέθοδοι νημάτων (2 από 6)

- `void run ()`
  - Περιέχει τον κώδικα του νήματος
  - Κάθε υποτάξη ορίζει τη δική της `run()`
- `void setName (String όνομα)`
- `void String getName ()`
  - Θέτει/επιστρέφει το όνομα του νήματος
- `void sleep (int millis)`
  - Αποκοιμίζει το νήμα για διάστημα msec

# Μέθοδοι νημάτων (3 από 6)

- `void yield ()`
  - Παραχωρεί τον επεξεργαστή σε άλλα νήματα
- `void interrupt ()`
  - Στέλνει στο νήμα ένα σήμα διακοπής
  - Αν είναι κοιμισμένο ή σε αναμονή ξυπνάει
- `boolean isInterrupted ()`
  - Επιστρέφει `true` αν το νήμα έχει διακοπεί

# Μέθοδοι νημάτων (4 από 6)

- `static boolean interrupted ()`
  - True αν το εκτελούμενο έχει διακοπεί
  - Καθαρίζει (reset) την κατάσταση διακοπής
- `boolean isAlive ()`
  - Επιστρέφει true το νήμα εκτελείται
- `void setPriority (int προτεραιότητα)`
- `int getPriority ()`
  - Θέτει/επιστρέφει προτεραιότητα νήματος

# Μέθοδοι νημάτων (5 από 6)

- `void join (long millis)`
  - Περιμένει τερματισμό νήματος για `millis ms`
  - Αν το διάστημα είναι 0, αναμονή για πάντα
- `void join ()`: Ισοδύναμη με `join(0)`
- `void checkAccess ()`
  - Εξετάζει αν έχουμε πρόσβαση στο νήμα
    - Εξαίρεση αν δεν έχουμε
    - Επιστροφή αν έχουμε

# Μέθοδοι νημάτων (6 από 6)

- ThreadGroup getThreadGroup ()
  - Επιστρέφει την ομάδα του νήματος
- Thread.State getState ()
  - Επιστρέφει τρέχουσα κατάσταση νήματος:
    - NEW, RUNNABLE, BLOCKED
    - WAITING, TIMED\_WAITING, TERMINATED
- static Thread currentThread ()
  - Επιστρέφει αναφορά προς εκτελούμενο νήμα

# Αντικείμενα εκτέλεσης (1 από 2)

- Δημιουργία νημάτων
  - Επέκταση της τάξης Thread
  - Δεν γίνεται αν επεκτείνει άλλη τάξη
- Διεπαφή Runnable
  - Υλοποίηση από εκτελέσιμα αντικείμενα (νήματα)
    - Μπορεί να επεκτείνει οποιαδήποτε τάξη
  - Περιέχει μόνο την (αφηρημένη) μέθοδο run()
  - Περνάμε το νήμα σε κατασκευαστή της τάξης Thread
    - Εκκίνηση νήματος με τη μέθοδο start()

# Αντικείμενα εκτέλεσης (2 από 2)

- Thread (Runnable αντικείμενο, String όνομα\_νήματος)
  - Δημιουργεί επώνυμο νήμα για το αντικείμενο
- Thread (Runnable αντικείμενο)
  - Δημιουργεί νήμα "Thread-#" για το αντικείμενο
- Η τεχνική της διεπαφής είναι πιο γενική
  - Μπορούμε να υλοποιούμε πολλές διεπαφές
  - Υποκαθιστά την πολλαπλή κληρονομικότητα



# Νήματα δαίμονες (1 από 2)

- Νήματα χρήστη και νήματα δαίμονες
  - Κάθε νήμα είναι είτε χρήστη είτε δαίμονας
  - Η μέθοδος `main` εκτελείται από νήμα χρήστη
  - Οι δαίμονες εκτελούνται στο παρασκήνιο
  - Η εφαρμογή δημιουργεί όποια νήματα θέλει
  - Η εφαρμογή τερματίζεται με τα νήματα χρήστη
  - Τα νήματα δαίμονες θανατώνονται αυτόματα

# Νήματα δαίμονες (2 από 2)

- Ορισμός τύπου νήματος
  - Τα νέα νήματα κληρονομούν τύπο δημιουργού
- `void setDaemon (boolean είναι_δαίμων)`
  - Σημειώνει νήμα ως νήμα δαίμονα ή χρήστη
  - Επιτρέπεται μόνο από το δημιουργό του
    - Και μόνο πριν την εκκίνηση του νήματος
- `boolean isDaemon ()`
  - Επιστρέφει `true` αν το νήμα είναι δαίμων

# Ομάδες νημάτων (1 από 8)

- Ομάδες νημάτων: Τάξη ThreadGroup
  - Ενιαία διαχείριση και προστασία νημάτων
  - Ιεραρχική οργάνωση ομάδων νημάτων
  - Ρίζα η ομάδα main που δημιουργείται αυτόματα
- Ένταξη νημάτων σε ομάδες
  - Κάθε νήμα προστίθεται σε ομάδα δημιουργού
  - Ο δημιουργός μπορεί να ορίσει άλλη ομάδα
  - Ο διαχειριστής ασφαλείας μπορεί να ορίσει άλλη

# Ομάδες νημάτων (2 από 8)

- Ομάδες νημάτων δαίμονες
  - Καταστρέφονται αυτόματα
    - Όταν δεν περιέχουν άλλα ενεργά νήματα
  - Μπορεί να περιέχουν νήματα δαίμονες ή χρήστη
- ThreadGroup (String ομάδα)
  - Δημιουργεί ομάδα με γονέα την τρέχουσα
- ThreadGroup (ThreadGroup γονέας, String ομάδα)
  - Η ομάδα γονέας ορίζεται ρητά

# Ομάδες νημάτων (3 από 8)

- Σχετικοί κατασκευαστές νημάτων
- Thread (ThreadGroup ομάδα, String όνομα)
  - Δημιουργεί ένα νήμα και το εντάσσει σε ομάδα
- Thread (ThreadGroup ομάδα, Runnable αντικείμενο)
  - Χρησιμοποιεί επιπλέον ένα αντικείμενο Runnable
- Thread (ThreadGroup ομάδα, Runnable αντικείμενο, String όνομα)
  - Ορίζει επιπλέον και το όνομα του νήματος

# Ομάδες νημάτων (4 από 8)

- `String getName ()`
  - Επιστρέφει το όνομα αυτής της ομάδας νημάτων
- `ThreadGroup getParent ()`
  - Επιστρέφει την πατρική ομάδα νημάτων
- `void setDaemon(boolean δαίμονας)`
- `boolean isDaemon()`
  - Θέτει / επιστρέφει τον τύπο της ομάδας

# Ομάδες νημάτων (5 από 8)

- `void setMaxPriority (int μέγιστη)`
- `int getMaxPriority ()`
  - Θέτει / επιστρέφει τη μέγιστη προτεραιότητα
  - Δεν αλλάζει προτεραιότητα ενεργών νημάτων
- `boolean parentOf (ThreadGroup g)`
  - Επιστρέφει `true` αν η ομάδα είναι γονέας της `g`
- `void checkAccess()`
  - `True` αν το τρέχον νήμα έχει πρόσβαση στην ομάδα

# Ομάδες νημάτων (6 από 8)

- `int activeCount ()`
  - Πλήθος ενεργών νημάτων ομάδας και υποομάδων
- `int activeGroupCount ()`
  - Πλήθος ενεργών υποομάδων της ομάδας
- `int enumerate (Thread[] πίνακας, boolean αναδρ)`
  - Αντιγράφει κάθε ενεργό νήμα σε πίνακα
  - Αν `αναδρ == true` περιλαμβάνει τις ομάδες παιδιά



# Ομάδες νημάτων (7 από 8)

- `int enumerate (Thread[] πίνακας)`
  - Ισοδύναμη με `int enumerate (Thread[] πίνακας, true)`
- `int enumerate (ThreadGroup[] πίνακας, boolean αναδρ)`
  - Αντιγράφει ενεργές ομάδες της ομάδας σε πίνακα
- `int enumerate (ThreadGroup[] πίνακας)`
  - Ίδιο με `int enumerate (ThreadGroup[] πίνακας, true)`

# Ομάδες νημάτων (8 από 8)

- `void interrupt()`
  - Διακόπτει όλα τα νήματα ομάδας και υποομάδων
- `void destroy()`
  - Καταστρέφει την ομάδα και τις υποομάδες της
  - Η ομάδα πρέπει να είναι κενή!
- `uncaughtException` (Thread νήμα, Throwable εξαίρεση)
  - Καλείται όταν ένα νήμα πεθάνει
  - Λόγω άγνωστης εξαίρεσης

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Συγχρονισμός νημάτων

**Μάθημα:** Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,  
**Ενότητα # 8:** Ταυτοχρονισμός και νήματα

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Συγχρονισμός

- Συγχρονισμός και επικοινωνία νημάτων
  - Η Java παρέχει διάφορες σχετικές τεχνικές
  - Κάθε εφαρμογή επιλέγει τον πιο κατάλληλο
  - Αφορούν την κοινή μνήμη των νημάτων
    - Μοντέλο μνήμης της Java
  - Ο βασικός μηχανισμός είναι το monitor
    - Αρκετά περιορισμένο στη βασική του μορφή
    - Έχει επεκταθεί αρκετά στην πορεία

# Πτητικά πεδία

- Πτητικά πεδία
  - Παρακολουθούνται από όλα τα νήματα
  - Κάθε αλλαγή γίνεται άμεσα αντιληπτή
    - Σε όλα τα νήματα της εφαρμογής
  - Η αλλαγή γίνεται στην κοινή μνήμη
    - Όχι μόνο σε καταχωρητές
    - Όχι μόνο σε μία κρυφή μνήμη
  - Αρκεί να χρησιμοποιηθεί η λέξη κλειδί `volatile`

# Συγχρονισμένες μέθοδοι (1 από 5)

- Κάθε αντικείμενο έχει μια κλειδαριά (lock)
  - Μία κλειδαριά ανά υπόσταση (αντικείμενο)
    - Δεν ορίζεται ρητά, υπάρχει πάντα
  - Η κλειδαριά μπορεί να δεσμεύεται από ένα νήμα
    - Απόκτηση κλειδαριάς πριν από κρίσιμη περιοχή
    - Απελευθέρωση κλειδαριάς μετά από κρίσιμη περιοχή
  - Χρήση `synchronized` μεθόδων και εντολών
    - Αυτόματος έλεγχος κλειδαριάς

# Συγχρονισμένες μέθοδοι (2 από 5)

- Παρακολουθητής (monitor) αντικειμένου
  - Ελέγχει την κλειδαριά του αντικειμένου
  - Επιτρέπει σε ένα νήμα την απόκτηση της κλειδαριάς
    - Εμποδίζει όλα τα άλλα νήματα
  - Υλοποίηση με `synchronized` μεθόδους / εντολές
    - Απόκτηση κλειδαριάς για εκτέλεση `synchronized` μεθόδου
    - Επιτρέπονται πολλές `synchronized` μέθοδοι ανά αντικείμενο
    - Αμοιβαίος αποκλεισμός όλων των `synchronized` μεθόδων
    - Οι άλλες μέθοδοι δεν επηρεάζονται

# Συγχρονισμένες μέθοδοι (3 από 5)

- Προστασία δεδομένων αντικειμένου
  - Τα δημόσια δεδομένα δεν προστατεύονται
- Ιδιοκτησία κλειδαριάς αντικειμένου
  - Η κλειδαριά καλύπτει όλο το αντικείμενο
  - Το νήμα μπορεί να κάνει πολλές κλήσεις
    - Περιλαμβάνονται και άλλες κλήσεις synchronized
- Στατικές μέθοδοι synchronized
  - Το αντικείμενο τάξης έχει τη δική του κλειδαριά
    - Ανεξάρτητη από τις κλειδαριές των υποστάσεων



# Συγχρονισμένες μέθοδοι (4 από 5)

- Synchronized εντολές
  - Synchronized (παράσταση) { εντολές }
  - Αρχικά υπολογίζεται η παράσταση
    - Πρέπει να καταλήγει σε αναφορά αντικειμένου
  - Αποκτάται η κλειδαριά του αντικειμένου
  - Οι εντολές εκτελούνται με αμοιβαίο αποκλεισμό

# Συγχρονισμένες μέθοδοι (5 από 5)

- Γιατί *synchronized* εντολές;
  - Μικρότερες κρίσιμες περιοχές
  - Συγχρονισμός πάνω σε οποιοδήποτε αντικείμενο
  - Συγχρονισμός πάνω σε κλειδαριά αντικειμένου τάξης
- Γιατί *synchronized* μέθοδοι;
  - Συγχρονισμός από την πλευρά του εξυπηρετητή
  - Το αντικείμενο αποφασίζει για το συγχρονισμό του

# Αναμονή - ειδοποίηση (1 από 6)

- Συγχρονισμός με βάση συνθήκες
  - Παράδειγμα: πρόβλημα παραγωγού-καταναλωτή
  - Πεπερασμένο σύνολο ενταμιευτών
  - Ο παραγωγός περιμένει για κενές θέσεις
  - Ο καταναλωτής περιμένει για γεμάτες θέσεις
  - Αναμονή μέχρι να ισχύσει κάποια συνθήκη
  - Χειρισμός μέσα σε αντικείμενο `synchronized`

# Αναμονή - ειδοποίηση (2 από 6)

- `void wait (long tim)`: εμποδισμός νήματος
  - Απελευθέρωση κλειδαριάς αντικειμένου
  - Το νήμα εμποδίζεται μέχρι:
    1. Να κληθεί η μέθοδος `notify` και να επιλεγεί το νήμα
    2. Να κληθεί η μέθοδος `notifyAll`
    3. Να περάσουν `tim msec`
    4. Να κληθεί η `interrupt` για την αποστολή διακοπής
- `void wait ()`: ισοδύναμη με `wait (0)`
  - Αναμονή χωρίς εκπνοή χρονικού διαστήματος

# Αναμονή - ειδοποίηση (3 από 6)

- `void notifyAll ()`: απελευθέρωση όλων των νημάτων
  - Μπορεί να άλλαξε κάποια συνθήκη αναμονής
  - Ένα μόνο νήμα αποκτά την κλειδαριά
    - Τα υπόλοιπα εμποδίζονται ξανά στην κλειδαριά
- `void notify()`: απελευθέρωση ενός νήματος
  - Δεν επιλέγουμε νήμα προς απελευθέρωση
  - Χρήσιμη όταν δεν μας ενδιαφέρει το νήμα
  - Κατάλληλη όταν έχουμε μία συνθήκη ή ένα νήμα

# Αναμονή - ειδοποίηση (4 από 6)

- Σχέση wait και notify/notifyAll
  - Οι γνωστοποιήσεις δεν αποθηκεύονται
  - Το wait πρέπει να ακολουθείται από notify/notifyAll
  - Χρήση timeout για αποφυγή αδιεξόδων
- Χρήση wait και notify/notifyAll
  - Επικοινωνία μόνο μέσω synchronized μεθόδων
    - Ελευθερώνουν/δεσμεύουν αυτόματα την κλειδαριά
  - Δεν μπορούν να κληθούν μέσα σε άλλες μεθόδους

# Αναμονή - ειδοποίηση (5 από 6)

- Σημασιολογία εντολής wait
  - Εμποδισμός νήματος πάνω στο αντικείμενο
  - Απελευθέρωση κλειδαριάς αντικειμένου
  - Η απελευθέρωση θα γίνει μέσω notify/notifyAll
  - Όταν εκτελεστεί, το νήμα θα κατέχει την κλειδαριά
- Χρήση εντολής wait
  - Όταν εκτελεστεί το νήμα, ισχύει η συνθήκη του;
  - Κλήση wait μέσα σε βρόχο while (όχι εντολή if)
  - Ο βρόχος ελέγχει την επιθυμητή συνθήκη
  - Αν δεν ισχύει το νήμα εμποδίζεται

# Αναμονή - ειδοποίηση (6 από 6)

- Σημασιολογία εντολής `notifyAll`
  - Απελευθέρωση όλων των νημάτων που έκαναν `wait`
    - Κάθε νήμα αποκτά την κλειδαριά με τη σειρά
    - Τα νήματα ελέγχουν τη συνθήκη αναμονής
    - Αν δεν ικανοποιείται, εμποδίζονται ξανά
- Σημασιολογία εντολής `notify`
  - Απελευθερώνει ένα μόνο νήμα
- Πότε αξίζει η `notify`;
  - Όλα τα νήματα πρέπει να περιμένουν την ίδια συνθήκη
  - Μόνο ένα πρέπει να επωφελείται από τη συνθήκη



# Άλλοι μηχανισμοί (1 από 2)

- Παράλληλη εκτέλεση από δεξαμενή νημάτων
  - Τάξη ThreadPoolExecutor
- Ουρές που εμπεριέχουν το συγχρονισμό
  - Διεπαφή BlockingQueue
- Ταυτόχρονες εκδόσεις συλλογών
  - Τάξη ConcurrentHashMap

# Άλλοι μηχανισμοί (2 από 2)

- Κλασικά εργαλεία συγχρονισμού
  - Τάξη Semaphore
- Διάφορες υλοποιήσεις κλειδαριών
  - Διεπαφή ReadWriteLock
- Ατομικοί τύποι μεταβλητών
  - Τάξη AtomicBoolean και AtomicLong

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Τέλος Ενότητας # 8

**Μάθημα:** Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,  
**Ενότητα # 8:** Ταυτοχρονισμός και νήματα

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

