

# ΜΥΥ601 Λειτουργικά Συστήματα

Εαρινό 2024

## Μάθημα 3

### Νήματα

---

Τμήμα Μηχανικών Η/Υ & Πληροφορικής  
Πανεπιστήμιο Ιωαννίνων

1

## Περίγραμμα

---

- Εισαγωγή
- Πολυνηματισμός
- Υλοποίηση
- Πολυεπεξεργαστές
- Solaris
- Linux
- Windows
- Pthreads

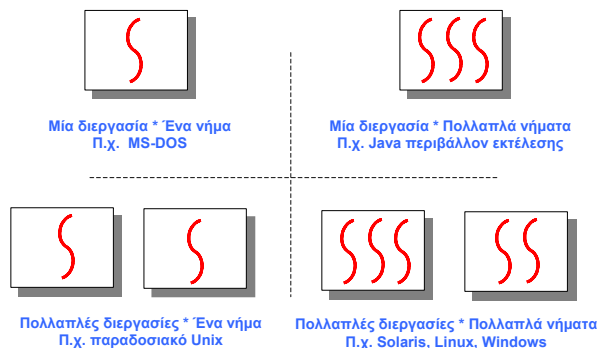
2

## Χαρακτηριστικά Διεργασιών

- **Κατοχή πόρων**
  - Ο χώρος εικονικών διευθύνσεων περιέχει την εικόνα διεργασίας
  - Κύρια μνήμη, κανάλια και συσκευές Ε/Ε, αρχεία
  - Ονομάζεται *διεργασία* (process) ή *στοιχειώδης εργασία* (task)
- **Δρομολόγηση εκτέλεσης**
  - Μονοπάτι εκτέλεσης μέσα από ένα ή περισσότερα προγ/τα
  - Δρομολογήσιμη οντότητα με προτεραιότητα διεκπεραίωσης
  - Ονομάζεται *νήμα* (thread) ή *ελαφριά διεργασία* (lightweight process)

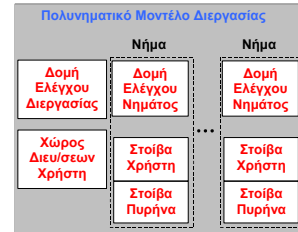
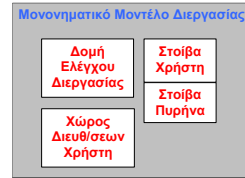
## Πολυνηματισμός

- **Ικανότητα του λειτουργικού συστήματος να**
  - Υποστηρίζει πολλαπλά νήματα εκτέλεσης σε μία διεργασία



## Πολυνηματικό Περιβάλλον

- **Διεργασία:** μονάδα καταχώρησης και προστασίας
  - Χώρος εικονικής μνήμης με εικόνα της διεργασίας
  - Προστατευμένη πρόσβαση στους επεξεργαστές, άλλες διεργασίες, αρχεία, Ε/Ε, κλπ.
- **Νήμα:** ανεξάρτητος μετρητής προγράμματος μέσα σε μια διεργασία
  - Ένα ή περισσότερα νήματα μέσα σε μια διεργασία
  - Κατάσταση εκτέλεσης νήματος (Εκτελούμενο, κλπ)
  - Περιβάλλον εκτέλεσης νήματος
  - Στοιβα εκτέλεσης
  - Στατική αποθήκευση για τοπικές μεταβλητές
  - Πρόσβαση στους πόρους της διεργασίας



## Πλεονεκτήματα

- **Ταχύτητα διαχείρισης**
  - Χρόνος δημιουργίας
  - Χρόνος τερματισμού
  - Χρόνος αλλαγής νήματος μέσα σε μία διεργασία
  - Χρόνος επικοινωνίας μεταξύ νημάτων μέσα σε μία διεργασία
- **Παραδείγματα**
  - Προσκήνιο/παρασκήνιο, π.χ. εκτέλεση εντολών και απεικόνιση
  - Ασύγχρονη εκτέλεση, π.χ. περιοδική αποθήκευση δεδομένων
  - Ταχεία εκτέλεση, π.χ. ταυτόχρονη ανάγνωση/επεξεργασία εργασιών
  - Αρθρωτή δομή προγράμματος, π.χ. ταυτόχρονη πρόσβαση στο δίσκο και μεταφορά δεδομένων στο δίκτυο

## Λειτουργικότητα

- **Καταστάσεις**
  - Εκτελούμενη, Έτοιμη, Αποκλεισμένη: παρόμοιες με των διεργασιών
- **Πράξεις**
  - Έναρξη
    - Όταν δημιουργείται μια διεργασία ή ένα νήμα δημιουργεί ένα άλλο
    - Καταχώρηση μνήμης καταχωρητών και στοίβας σε νέο νήμα
  - Αποκλεισμός
    - Αποθήκευση καταχωρητών και δεικτών στοίβας όταν το νήμα περιμένει
  - Άρση αποκλεισμού
    - Μετακίνηση νήματος από την ουρά Αποκλεισμένων στην ουρά Έτοιμων, όταν συμβεί το γεγονός
  - Τερματισμός
    - Αποδέσμευση χώρου καταχωρητών και στοίβας

Εαρινό 2024

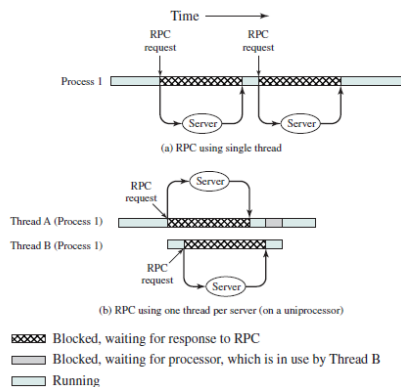
©Σ. Β. Αναστασιάδης

7

7

## Παράδειγμα

- **Θεωρούμε**
  - Έναν επεξεργαστή, μία διεργασία, ένα ή δύο νήματα
- **Remote Procedure Call (RPC)**
  - Απομακρυσμένη εξυπηρέτηση τύπου πελάτη/διακομιστή με κλήση διαδικασιών
- **Αλλαγή νήματος**
  - Μετά τον αποκλεισμό ή την ολοκλήρωση του εκτελούμενου νήματος



Εαρινό 2024

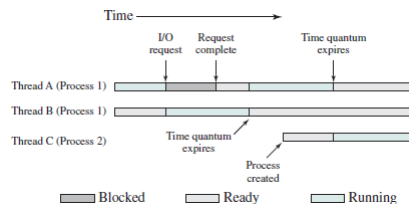
©Σ. Β. Αναστασιάδης

8

8

## Δεύτερο Παράδειγμα

- **Θεωρούμε**
  - Έναν επεξεργαστή, δύο διεργασίες, τρία νήματα
- **Αλλαγή νήματος**
  - Αποκλεισμός εκτελούμενου νήματος ή εξάντληση κβάντου χρόνου



Εαρινό 2024

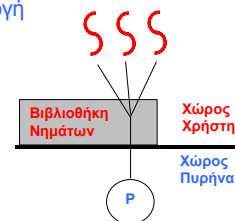
©Σ. Β. Αναστασιάδης

9

9

## Νήματα Επιπέδου Χρήστη

- **Διαχείριση νημάτων**
  - Από βιβλιοθήκη νημάτων συνδεμένη με την εφαρμογή
  - Ο πυρήνας αγνοεί την ύπαρξη νημάτων
- **Βιβλιοθήκη νημάτων**
  - Δημιουργεί και καταστρέφει νήματα
  - Περνάει μηνύματα και δεδομένα μεταξύ νημάτων
  - Δρομολογεί εκτέλεση νημάτων
  - Αποθηκεύει και επαναφέρει περιεχόμενα νημάτων
- **Πλεονεκτήματα**
  - Αλλαγή νημάτων σε επίπεδο χρήστη, εξειδικευμένη χρονοδρομολόγηση, μεταφερισιμότητα κώδικα εφαρμογών μεταξύ λειτουργικών συστημάτων
- **Μειονεκτήματα**
  - Κλήση συστήματος με αποκλεισμό αποκλείει όλα τα νήματα μιας διεργασίας
  - Μία διεργασία δεν αξιοποιεί πολλαπλούς διαθέσιμους επεξεργαστές



Εαρινό 2024

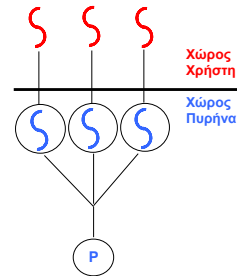
©Σ. Β. Αναστασιάδης

10

10

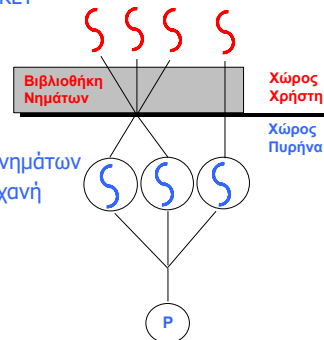
## Νήματα Επιπέδου Πυρήνα

- **Διαχείριση νημάτων**
  - Διεπαφή προγραμματισμού εφαρμογών με τον πυρήνα
  - Καθόλου κώδικας διαχείρισης νημάτων στην εφαρμογή
  - Π.χ. Linux, Windows
- **Πλεονεκτήματα**
  - Νήματα από μία διεργασία σε πολλαπλούς επεξεργαστές
  - Αποκλεισμένο νήμα δεν αποκλείει την διεργασία του
  - Οι ρουτίνες πυρήνα μπορούν να είναι πολυνηματικές
- **Μειονεκτήματα**
  - Αλλαγή νημάτων απαιτεί μεταφορά του επιπέδου εκτέλεσης στον πυρήνα



## Συνδυασμένη Προσέγγιση

- **Υπηρεσία**
  - Συνδυασμός νημάτων επιπέδου χρήστη (ULT) και επιπέδου πυρήνα (KLT)
  - Απεικόνιση των ULT σε ίσο ή μικρότερο πλήθος KLT
  - Π.χ. Solaris
- **Βιβλιοθήκη επιπέδου χρήστη**
  - Δημιουργία, χρονοδρομολόγηση, συγχρονισμός νημάτων
  - Προσαρμογή του πλήθους KLT σε εφαρμογή/μηχανή
- **Πυρήνας**
  - Τρέχει KLT σε πολλαπλούς επεξεργαστές
  - Αλλάζει το KLT όταν μπαίνει σε αποκλεισμό



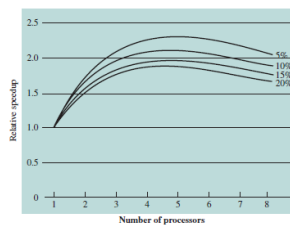
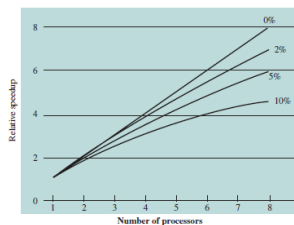
## Απόδοση Παράλληλου Λογισμικού

- **Νόμος του Amdahl**

$$Speedup = \frac{\text{χρόνος εκτέλεσης προγράμματος σε έναν επεξεργαστή}}{\text{χρόνος εκτέλεσης προγράμματος σε } N \text{ παράλληλους επεξεργαστές}} = \frac{1}{(1-f) + \frac{f}{N}}$$

όπου

- $f$  είναι το ποσοστό παραλληλοποιήσιμου κώδικα και
  - $(1-f)$  το ποσοστό του σειριακού κώδικα ενός προγράμματος
- **Παραδείγματα για διάφορες τιμές του  $(1-f)$**



Εαρινό 2024

©Σ. Β. Αναστασιάδης

13

13

## Κατηγορίες Υπολογιστικών Συστημάτων

- **Απλής Εντολής Απλών Δεδομένων (SISD)**
  - Ένας επεξεργαστής εκτελεί μια ροή εντολών
  - Λειτουργεί με δεδομένα αποθηκευμένα σε απλή μνήμη
- **Απλής Εντολής Πολλαπλών Δεδομένων (SIMD)**
  - Μια απλή εντολή ελέγχει την εκτέλεση πολλαπλών στοιχείων επεξεργασίας
  - Κάθε στοιχείο επεξεργασίας έχει τη δική του μνήμη δεδομένων
  - Π.χ. επεξεργαστές διανύσματος ή πίνακα, κάρτα γραφικών
- **Πολλαπλών Εντολών Απλών Δεδομένων (MISD)**
  - Ακολουθία από δεδομένα μεταφέρεται σε πολλαπλούς επεξεργαστές
  - Κάθε επεξεργαστής εκτελεί διαφορετική ακολουθία εντολών
  - Δεν υλοποιήθηκε ποτέ
- **Πολλαπλών Εντολών Πολλαπλών Δεδομένων (MIMD)**
  - Σύνολο από επεξεργαστές εκτελούν διαφορετικές ακολουθίες εντολών σε διαφορετικά δεδομένα

Εαρινό 2024

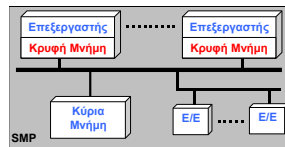
©Σ. Β. Αναστασιάδης

14

14

## Πολλαπλών Εντολών Πολλαπλών Δεδομένων

- **Συστοιχία υπολογιστών (computer cluster)**
  - Κάθε επεξεργαστής ένας αυτοτελής υπολογιστής
  - Επικοινωνία μεταξύ επεξεργαστών μέσω δικτύου
- **Πολυεπεξεργαστής κοινόχρηστης μνήμης (shared-memory multiprocessor)**
  - Επεξεργαστές
    - Έχουν πρόσβαση σε κοινόχρηστη μνήμη
    - Προσπελίζουν προγράμματα και δεδομένα από την κοινόχρηστη μνήμη
    - Επικοινωνούν μέσω της κοινόχρηστης μνήμης
  - Εκτέλεση πυρήνα λειτουργικού συστήματος μπορεί να γίνεται
    1. Μόνο σε συγκεκριμένο επεξεργαστή (αρχιτεκτονική master/slave)
    2. Σε οποιοσδήποτε επεξεργαστές (συμμετρικός πολυεπεξεργαστής - SMP)



Εαρινό 2024

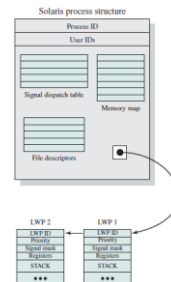
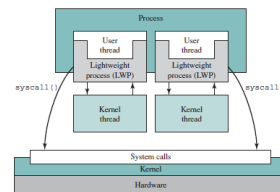
©Σ. Β. Αναστασιάδης

15

15

## Solaris

- **Διεργασία**
  - Κανονική διεργασία Unix
  - Περιλαμβάνει χώρο διευθύνσεων χρήστη, στοίβα και δομή ελέγχου διεργασιών
- **Νήμα χρήστη (user thread)**
  - Υλοποίηση από βιβλιοθήκη νημάτων στο χώρο διευθύνσεων της διεργασίας
  - Αόρατα στο λειτουργικό σύστημα
- **Ελαφριά διεργασία (lightweight process, LWP)**
  - Αντιστοιχεί σε ένα νήμα πυρήνα
  - Υποστηρίζει ένα ή πολλαπλά νήματα χρήστη
- **Νήμα πυρήνα (kernel thread)**
  - Δρομολογούνται από τον πυρήνα
  - Εκτελούνται σε έναν από τους επεξεργαστές



Εαρινό 2024

©Σ. Β. Αναστασιάδης

16

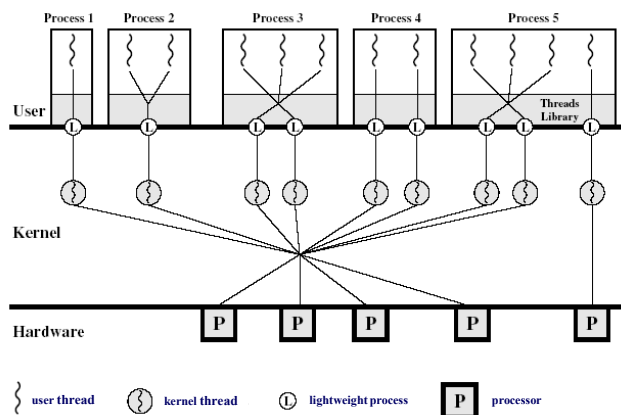
16



## Παράδειγμα

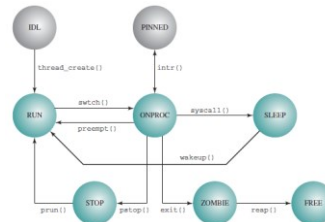
- **Διεργασία 1**
  - Ένα νήμα χρήστη αντιστοιχεί σε μία LWP, π.χ. πολύ απλή εφαρμογή
- **Διεργασία 2**
  - Πολλαπλά νήματα χρήστη αντιστοιχούν σε μία LWP, π.χ. πολλαπλά παράθυρα με ένα ενεργό
- **Διεργασία 3**
  - Πολλαπλά νήματα χρήστη πολυπλέκονται σε λιγότερες LWP
  - Π.χ. πολλά νήματα μερικά από τα οποία μπαίνουν σε αποκλεισμό E/E
- **Διεργασία 4**
  - Κάθε νήμα χρήστη αντιστοιχεί σε μία LWP
  - Π.χ. πολλαπλασιασμό πινάκων με σταθερό πλήθος γραμμών/νήμα
- **Διεργασία 5**
  - Πολλαπλά νήματα αντιστοιχούν σε λιγότερες LWP
  - Μία LWP αντιστοιχεί σε έναν επεξεργαστή π.χ εφαρμογή πραγματικού χρόνου

## Σχήμα



## Εκτέλεση Νήματος

- **Ενεργή κατάσταση**
  - Αντιστοιχεί σε μία LWP και τρέχει όταν τρέχει το νήμα πυρήνα
- **Τι απενεργοποιεί ένα νήμα επιπέδου χρήστη ;**
  - Άλλο νήμα χρήστη με υψηλότερη προτεραιότητα
  - Αίτηση παραχώρησης από το ίδιο το νήμα
  - Συγχρονισμός που επιτρέπει μόνο ένα νήμα να προχωρήσει
  - Αίτηση αναστολής από άλλο νήμα χρήστη
- **Καταστάσεις**
  - RUN: Έτοιμο για εκτέλεση
  - ONPROC: Εκτελείται
  - STOP: Σταματημένο (π.χ., για debugging)
  - SLEEP: Σε αποκλεισμό
  - ZOMBIE: Τερματισμένο
  - FREE: Πριν την πλήρη διαγραφή του



Εαρινό 2024

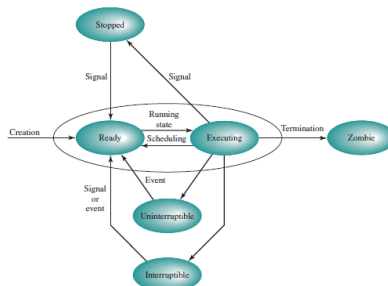
©Σ. Β. Αναστασιάδης

19

19

## Καταστάσεις Διεργασίας Linux

- **Το Linux δεν κάνει διάκριση μεταξύ διεργασιών και νημάτων**
  - Running: Η διεργασία είναι είτε Εκτελούμενη (Executing) ή Έτοιμη (Ready)
  - Interruptible: Αναμένει κάποιο γεγονός, μπορεί να διακοπεί από σήμα
  - Uninterruptible: Αναμένει κάποιο γεγονός, αλλά αγνοεί τα σήματα
  - Stopped: Μπορεί να ξεκινήσει από άλλη διεργασία π.χ. έναν debugger
  - Zombie: Τερματίζεται αλλά παραμένει στον πίνακα διεργασιών



Εαρινό 2024

©Σ. Β. Αναστασιάδης

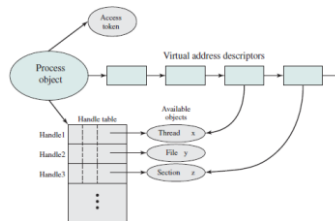
20

20

# Windows

- **Διεργασία**

- Υλοποιείται σαν αντικείμενο που κατέχει πόρους
- Περιέχει ένα ή περισσότερα νήματα, καθένα εκτελείται ακολουθιακά
- Το κουπόνι ασφάλειας πρόσβασης ελέγχει την πρόσβαση σε άλλα αντικείμενα
- Ο χώρος εικονικής μνήμης αποτελείται από κομμάτια μνήμης
- Το τμήμα (section) δίνει πρόσβαση στην κοινόχρηστη μνήμη



Εαρινό 2024

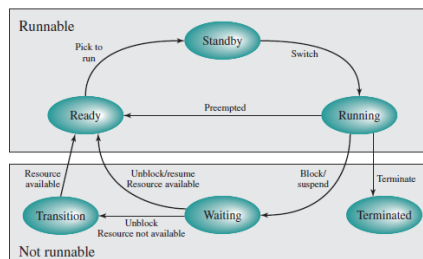
©Σ. Β. Αναστασιάδης

21

21

# Καταστάσεις Νημάτων

- **Ready:** Μπορεί να δρομολογηθεί για εκτέλεση
- **Standby:** Περιμένει μέχρι να γίνει διαθέσιμος ο επεξεργαστής
- **Running:** Εκτελείται σε έναν επεξεργαστή
- **Waiting:** Περιμένει να συμβεί ένα γεγονός π.χ. E/E
- **Transition:** Περιμένει να γίνουν διαθέσιμοι κάποιοι πόροι π.χ. μνήμη
- **Terminated:** Απομακρύνεται από το σύστημα



Εαρινό 2024

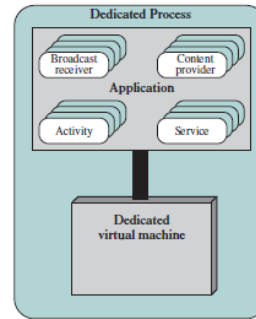
©Σ. Β. Αναστασιάδης

22

22

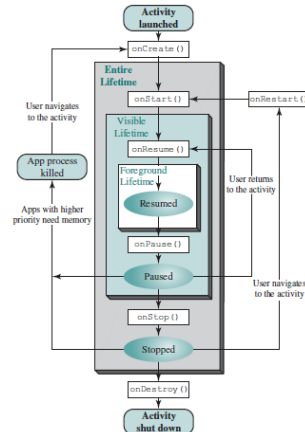
## Εφαρμογή Android

- Υλοποιεί μια εφαρμογή με συστατικά ένα ή περισσότερα στιγμιότυπα από τα παρακάτω
  - Activities** μοναδική οθόνη ως διεπαφή χρήστη, π.χ. λίστα με email, σύνθεση ή ανάγνωση email
  - Services** εκτελούν λειτουργίες στο υπόβαθρο, π.χ. παίζουν μουσική, φέρνουν δεδομένα από το δίκτυο
  - Content providers** διεπαφές σε δεδομένα εφαρμογών, π.χ., αποθηκεύουν δεδομένα σε σύστημα αρχείων, SQLite βάση δεδομένων, Ιστό
  - Broadcast receivers** αποκρίνονται σε ανακοινώσεις του συστήματος, π.χ. κατέβασμα αρχείου, ένδειξη χαμηλής μπαταρίας
- Κάθε εφαρμογή τρέχει στην δική της διεργασία με ένα ή περισσότερα νήματα



## Δραστηριότητα (Activity)

- Συστατικό εφαρμογής που παρέχει μια οθόνη
  - Επιτρέπει τον χρήστη να αλληλεπιδράσει με την εφαρμογή
  - Εμφανίζει μια διεπαφή χρήστη σε παράθυρο
- Πολλαπλές δραστηριότητες ανά εφαρμογή
  - Οργανώνονται σε στοιβά (last-in-first-out)
  - Η δραστηριότητα προσκηνίου αλληλεπιδρά με τον χρήστη
- Καταστάσεις δραστηριότητας
  - onCreate** αρχικοποιείται η δραστηριότητα
  - onStart** γίνεται ορατή η δραστηριότητα
  - onResume** επιτρέπει εισαγωγή εισόδου χρήστη
  - onPause** μετάβαση σε άλλη δραστηριότητα
  - onStop** σταματά την δραστηριότητα



## Νήματα POSIX

- Διεπαφή προγραμματισμού εφαρμογών
  - Ορίστηκε από την επιτροπή IEEE POSIX (Portable Operating System Interface) τον Ιούνιο 1995
- Κίνητρο
  - Καλύτερη απόδοση λόγω χαμηλότερης επιβάρυνσης σε σχέση με διεργασίες
- Πως γράφουμε πολυνηματικά προγράμματα

```
#include <pthread.h>
...
```
- Πως μεταγλωττίζουμε πολυνηματικά προγράμματα

```
gcc -lpthread ...
```

## Δημιουργία

- `int pthread_create(  
pthread_t *tid,  
const pthread_attr_t *attr,  
void *(*start) (void *),  
void *arg );`
  - Δημιουργεί νήμα που εκτελεί τη συνάρτηση *start* με παράμετρο *arg*
  - Καθορίζει κάποια χαρακτηριστικά στην παράμετρο *attr* π.χ. NULL
  - Ταυτοποιεί το νέο νήμα στο γονικό μέσω της επιστρεφόμενης τιμής *tid*
  - Επιστρέφει 0 αν επιτύχει, EAGAIN σε ανεπάρκεια πόρων, EINVAL σε άκυρο *attr*

## Αναμονή και Παραχώρηση

- `int pthread_join(pthread_t thread, void **value_ptr);`
  - Αναμονή για τον τερματισμό του νήματος με ταυτότητα *thread*
  - Η τιμή εξόδου του νήματος επιστρέφεται μέσω της *value\_ptr*
  - Το νήμα που ορίζει η *thread* πρέπει να δημιουργηθεί με χαρακτηριστικό `PTHREAD_CREATE_JOINABLE`
- `int pthread_yield();`
  - Το νήμα που την καλεί παραχωρεί τον επεξεργαστή σε άλλα που περιμένουν
  - Θα εκτελεστεί ξανά αφού τρέξουν όλες οι διεργασίες ίδιας προτεραιότητας

## Τερματισμός

- `int pthread_exit(void *value_ptr);`
  - Τερματίζει αμέσως το νήμα που την κάλεσε
  - Επιστρέφει *value\_ptr* στο νήμα που κάλεσε `pthread_join` για το τερματισμένο
  - Όταν η συνάρτηση `main()` τερματίζει
    - ΜΕ `pthread_exit()`, τα εκτελούμενα νήματα που δημιούργησε συνεχίζουν
    - ΧΩΡΙΣ `pthread_exit()`, τα εκτελούμενα νήματα που δημιούργησε τερματίζουν
- `int pthread_detach(pthread_t thread);`
  - Το νήμα δεν τερματίζει αμέσως
  - Όταν το νήμα τερματίσει, το σύστημα αποδεσμεύει τους πόρους του
  - Χρησιμοποιείται για να μην περιμένουμε τον τερματισμό ενός νήματος (*pthread\_join*)
  - Δε χρειαζόμαστε την κλήση, αν δημιουργήσουμε το νήμα με χαρακτηριστικό `PTHREAD_CREATE_DETACHED`

## Χαρακτηριστικά

- `pthread_attr_t attr;`
  - Δηλώνουμε τη μεταβλητή χαρακτηριστικών *attr*
  - Π.χ. `pthread_attr_t thread_attr;`
- `int pthread_attr_init (pthread_attr_t *attr);`
  - Αρχικοποιούμε τη μεταβλητή χαρακτηριστικών *attr*
  - Π.χ. `pthread_attr_init(&thread_attr);`
- `int pthread_attr_setdetachstate(pthread_attr_t *attr, int detachstate);`
  - Θέτουμε το κατάλληλο χαρακτηριστικό στη μεταβλητή *attr*
  - Π.χ. `pthread_attr_setdetachstate(&thread_attr, PTHREAD_CREATE_DETACHED);`