



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

Λειτουργικά Συστήματα

Ενότητα 3: Διεργασίες. Καταστάσεις Διεργασιών.
PCB. Νήματα.

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.ict.e.uowm.gr/mdasyg>

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών



Πανεπιστήμιο Δυτικής Μακεδονίας



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 ενότητας

- Η κατανόηση της έννοιας της διεργασίας.
- Η παρουσίαση των καταστάσεων των διεργασιών.
- Η σημαντικότητα των νημάτων.



Τι είναι η διεργασία; (1/2)

- Διεργασία (Process) είναι ο μηχανισμός εκτέλεσης ενός προγράμματος.
- Η διαδικασία είναι μια ενεργή οντότητα (περιέχει ένα σύνολο από συσχετισμένους με αυτή πόρους προκειμένου να εκτελεστεί). Το πρόγραμμα είναι παθητική οντότητα.
- Σημαντικό: ένα πρόγραμμα από μόνο του δεν αποτελεί διεργασία.
- Κάθε διεργασία εκτελεί ένα μοναδικό πρόγραμμα. Το ίδιο πρόγραμμα μπορεί να εκτελείται από πολλές διεργασίες (με διαφορετικές ταχύτητες και ακολουθίες εκτέλεσης των εντολών).
- Η εκτέλεση μιας διεργασίας γίνεται σειριακά. Το λειτουργικό μπορεί όμως να εκτελεί πολλές διεργασίες «παράλληλα» μεταξύ τους.



Τι είναι η διεργασία; (2/2)

- Ο πυρήνας του ΛΣ.
- Ο φλοιός του ΛΣ.
- Ο μεταφραστής (compiler).
- Βοηθητικά προγράμματα.
- ...



Λειτουργίες του ΛΣ για τις διεργασίες

- Παραχώρηση πόρων.
- Χρονοδρομολόγηση.
- Διαδιεργασιακή επικοινωνία (συγχρονισμός, αδιέξοδα,..).

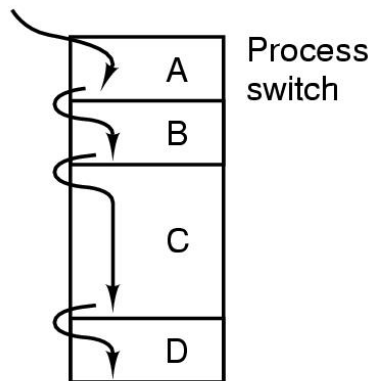


Το μοντέλο της διεργασίας.

Η έννοια του ψευδοπαράλληλισμού

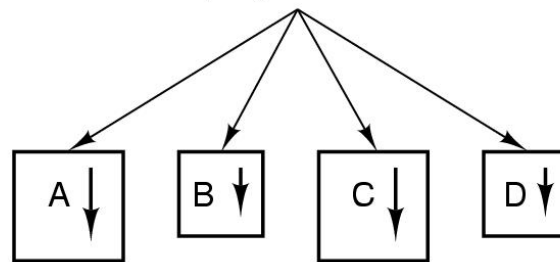
(a) Πολυπρογραμματισμός 4 προγραμμάτων, (b) Εννοιολογικό Μοντέλο 4 ανεξάρτητων σειριακών Διεργασιών, (c) Ένα πρόγραμμα ενεργό κάθε στιγμή!

One program counter

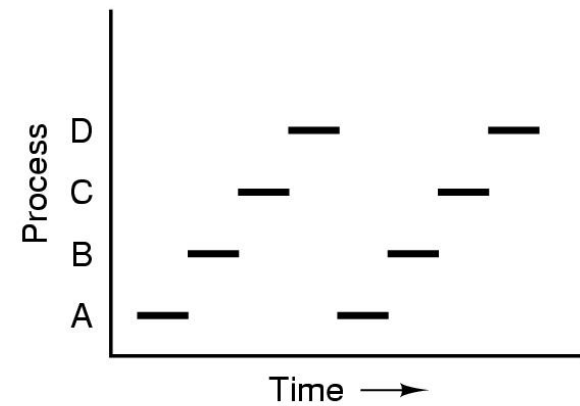


(a)

Four program counters



(b)



(c)



Δημιουργία της διεργασίας

- Αρχικοποίηση Συστήματος.
- Κλήση συστήματος από μια άλλη διεργασία.
- Αίτηση χρήστη.
- Ξεκίνημα μιας εργασίας δέσμης (a batch job).



Τερματισμός μιας διεργασίας

- Κανονική Έξοδος (οικειοθελής).
- «Κανονικό» σφάλμα (οικειοθελής).
- «Μοιραίο» σφάλμα (ανοικειοθελώς).
- «Σκότωμα» (Kill) από μια άλλη διεργασία (ανοικειοθελώς).

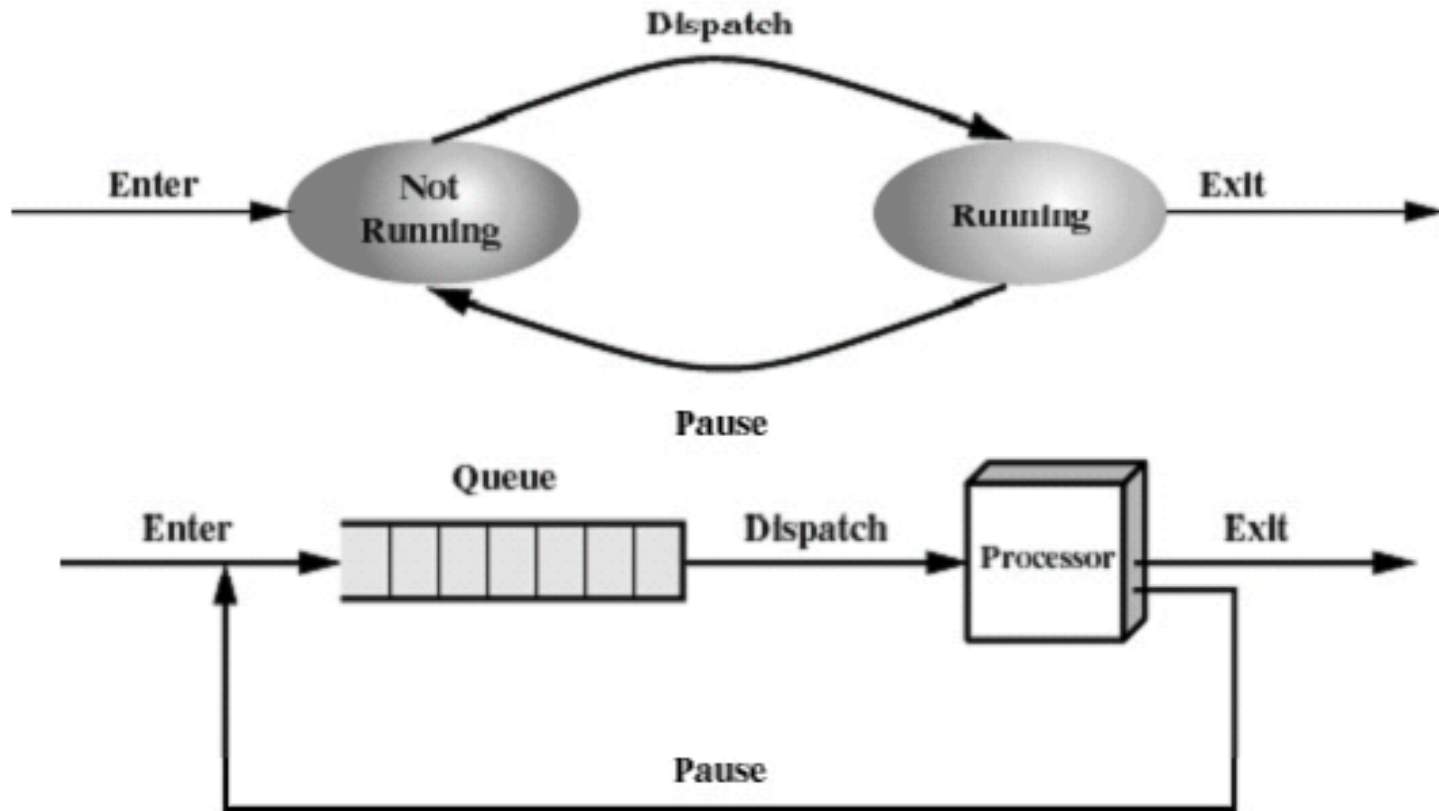


Ιεραρχίες Διεργασιών

- Η μητρική δημιουργεί θυγατρικές, οι οποίες μπορούν να δημιουργήσουν δικές τους θυγατρικές.
- Δημιουργία Ιεραρχίας.
 - Στο UNIX αποκαλείται «ομάδα διεργασιών» (process group).
 - Στα Windows δεν υπάρχει τέτοια έννοια: Όλες οι διεργασίες «γεννιούνται ίσες».



Οι 2 καταστάσεις διεργασιών

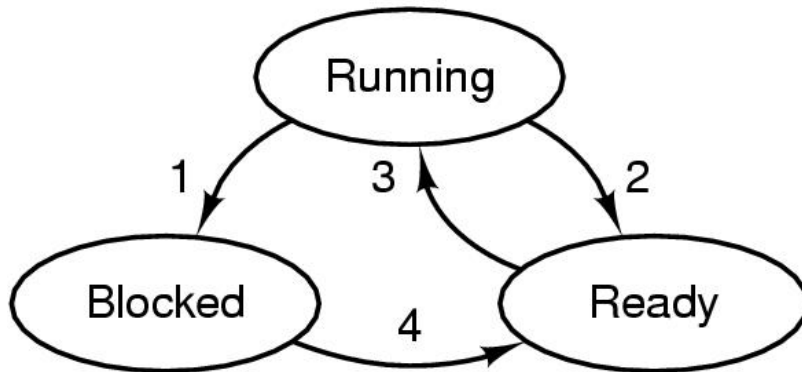


(b) Queuing diagram

3



Οι 3 καταστάσεις των διεργασιών



1. Process blocks for input
2. Scheduler picks another process
3. Scheduler picks this process
4. Input becomes available

Προσοχή στις έγκυρες μεταβάσεις!



Οι 5 καταστάσεις διεργασιών

- **new**: Η διεργασία δημιουργείται.
- **running**: Η διεργασία εκτελείται σε κάποιον επεξεργαστή.
- **waiting**: Η διεργασία αναμένει (απενεργοποιημένη) κάποιο συμβάν.
- **ready**: Η διεργασία αναμένει να της δοθεί (από το ΛΣ) σε κάποιον επεξεργαστή για την συνέχιση της εκτέλεσής της.
- **terminated**: Η διεργασία έχει ολοκληρώσει την εκτέλεσή της.

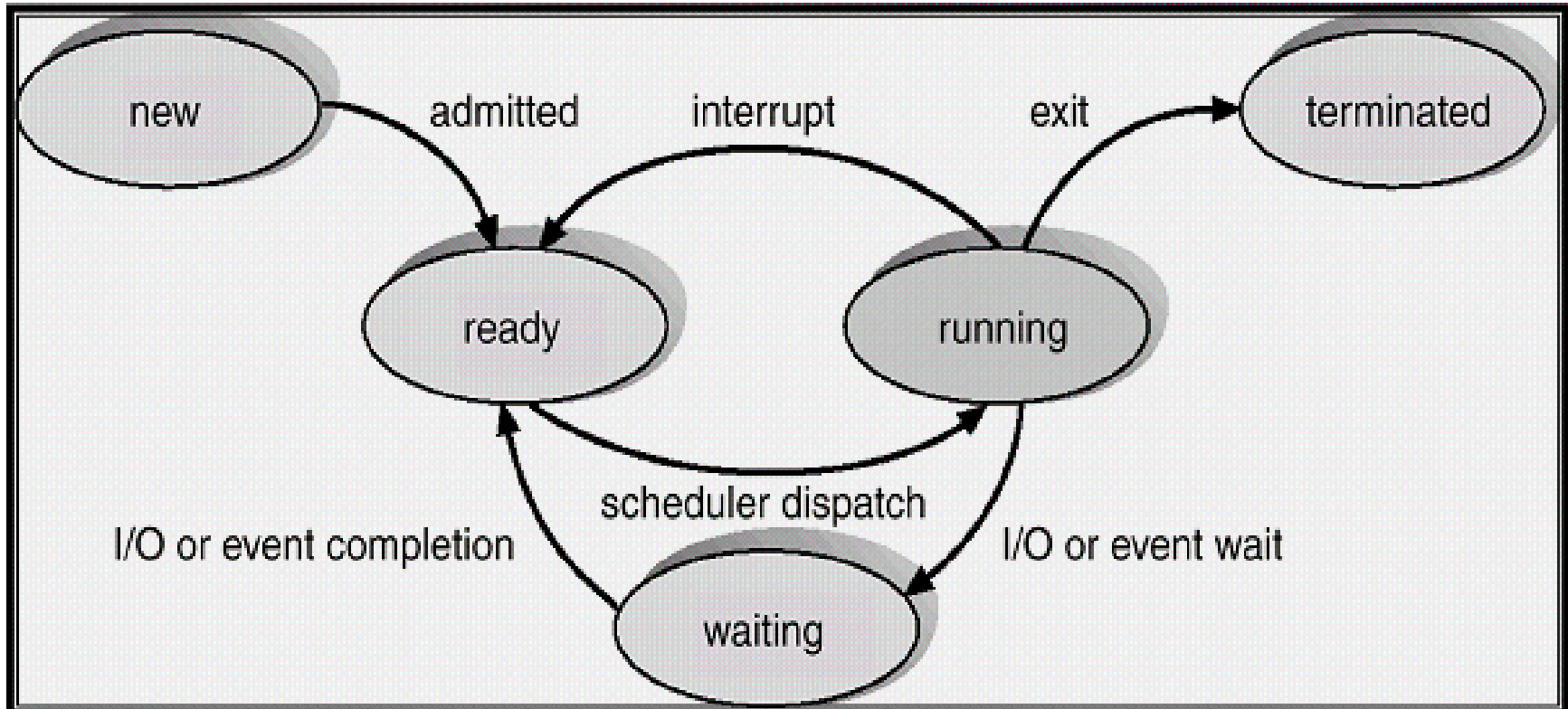


Αλλαγές Κατάστασης Διεργασίας

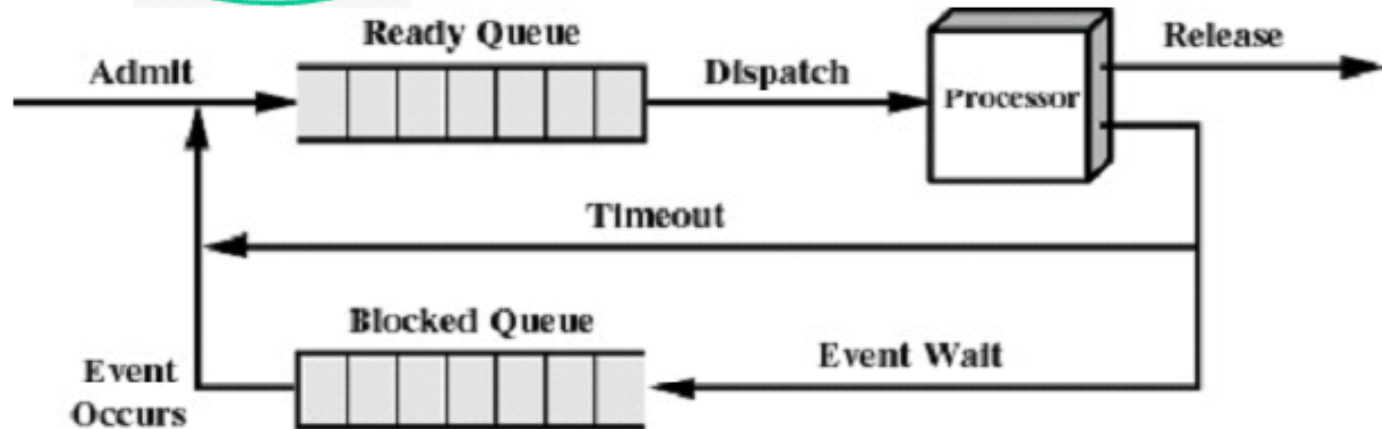
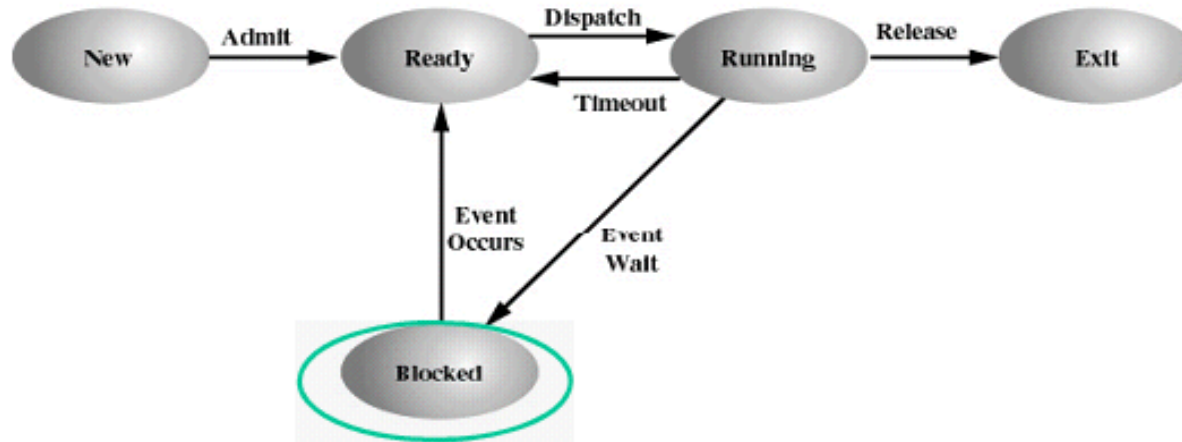
- **running -> waiting -> ready:** Μια διεργασία αλλάζει την κατάσταση της από running σε waiting συνήθως όταν περιμένει να της δοθεί κάποιος πόρος ή επιθυμεί να χρησιμοποιήσει κάποιον πόρο που δεν είναι διαθέσιμος και πρέπει να ελευθερωθεί. Εν συνεχεία, η διεργασία αλλάζει σε κατάσταση ready όταν της δοθεί ο πόρος που ζητήθηκε.
- **running -> ready:** Μια διεργασία αλλάζει την κατάσταση της από running σε ready συνήθως όταν η εκτέλεση της διακόπτεται από διακοπή υλικού (hardware interrupt) ή κλήση συστήματος (software trap).
- **ready -> running:** Μια διεργασία αλλάζει την κατάσταση της από ready σε running όταν της δοθεί επεξεργαστής για να συνεχίσει την εκτέλεση της.



Οι έγκυρες μεταβάσεις στις 5 καταστάσεις διεργασιών



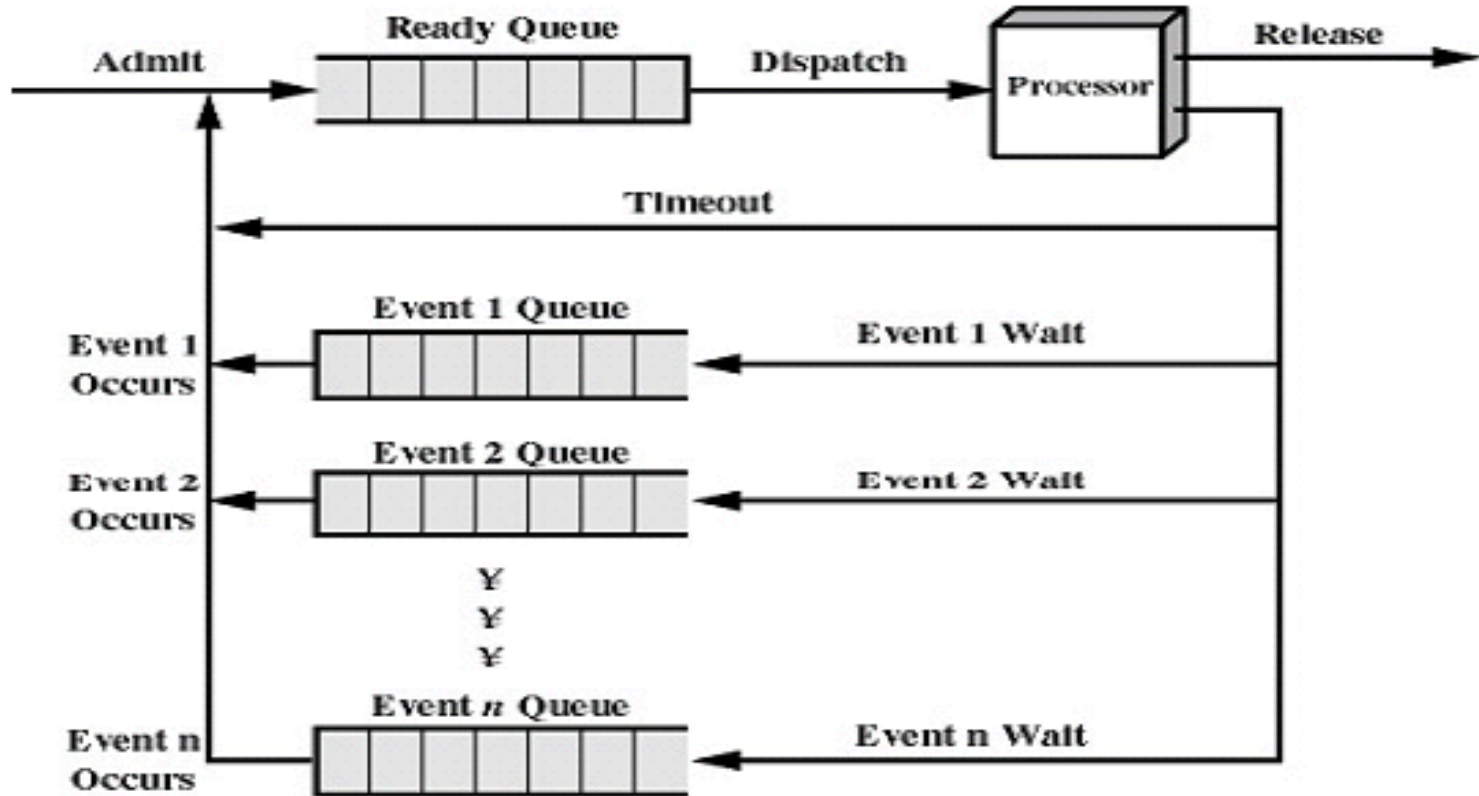
Οι διεργασίες blocked τοποθετούνται σε μια ουρά



(a) Single blocked queue

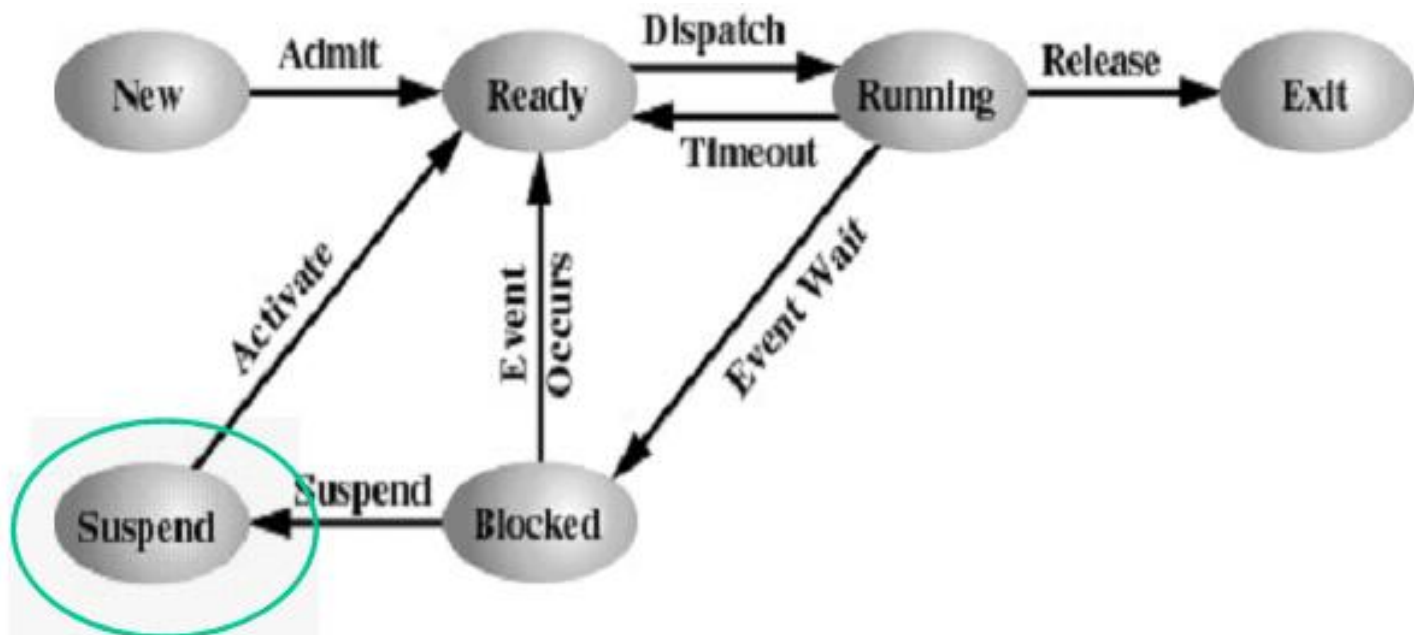


Πολλαπλές Ουρές

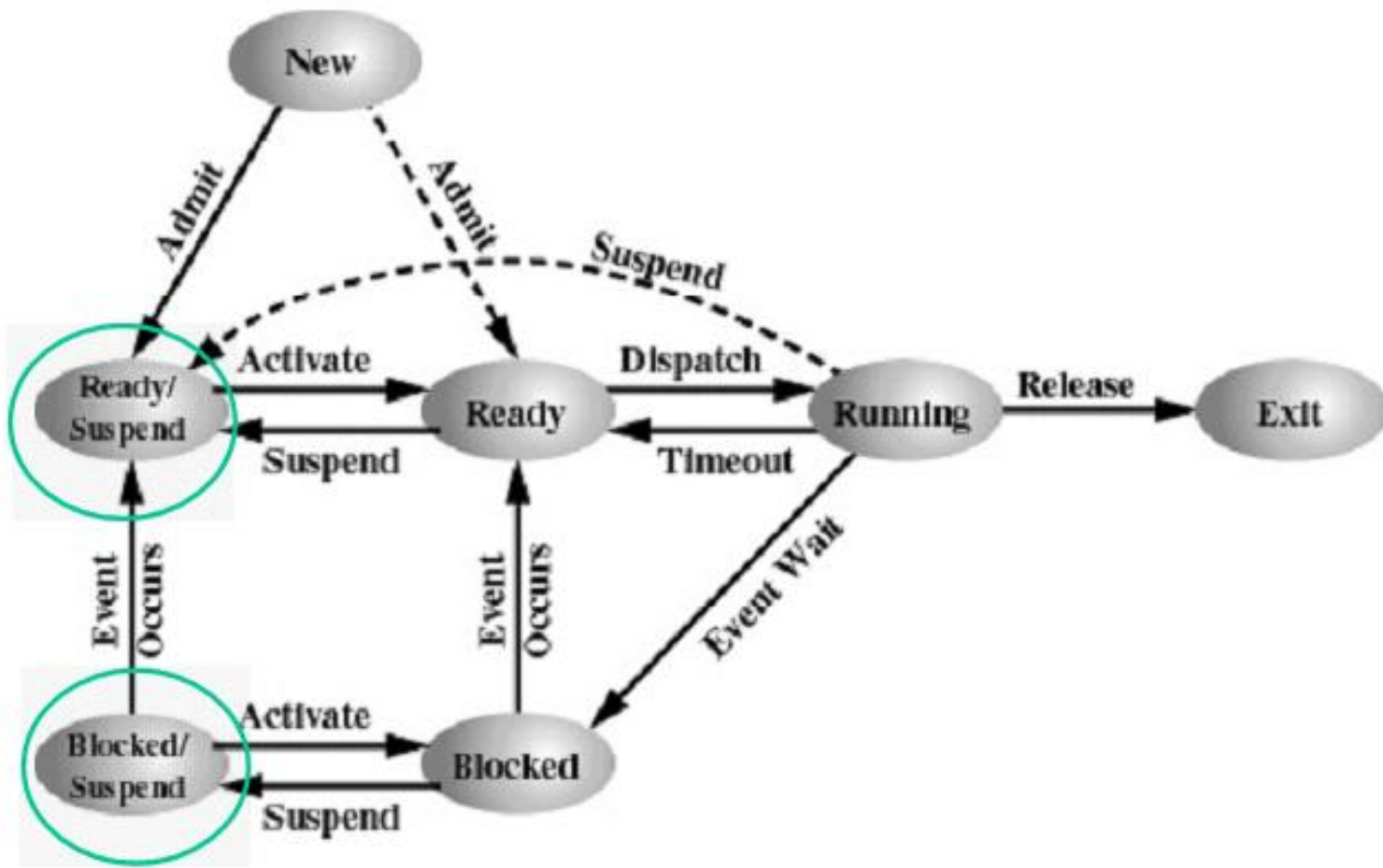


Το μοντέλο των 6 καταστάσεων

- Αν η διεργασία τοποθετηθεί στο swap-file (δηλαδή γίνει swapped) τότε υπάρχει η 6 κατάσταση.



Το μοντέλο των 7 καταστάσεων

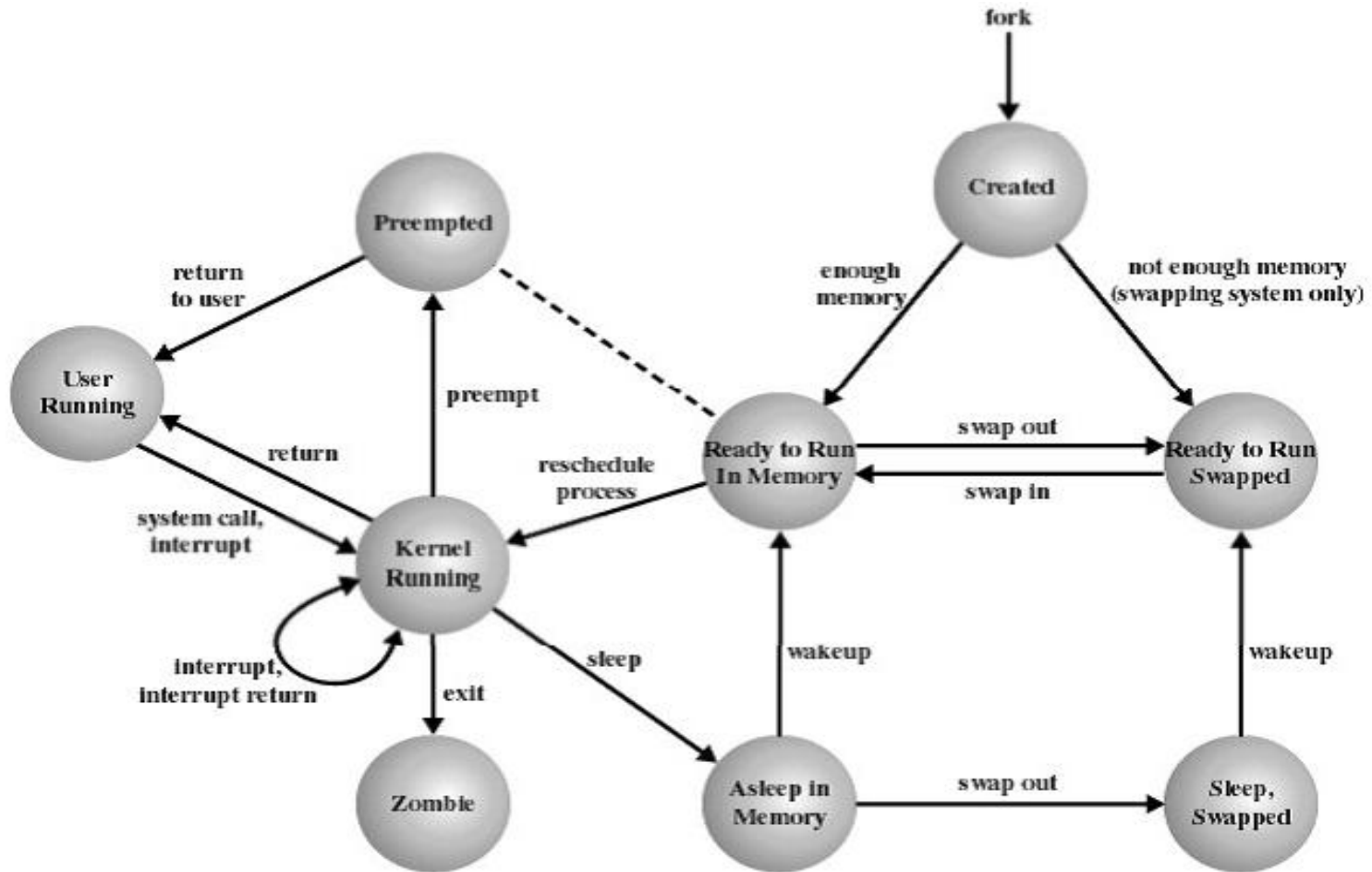


Πότε μπορεί να γίνει suspend;

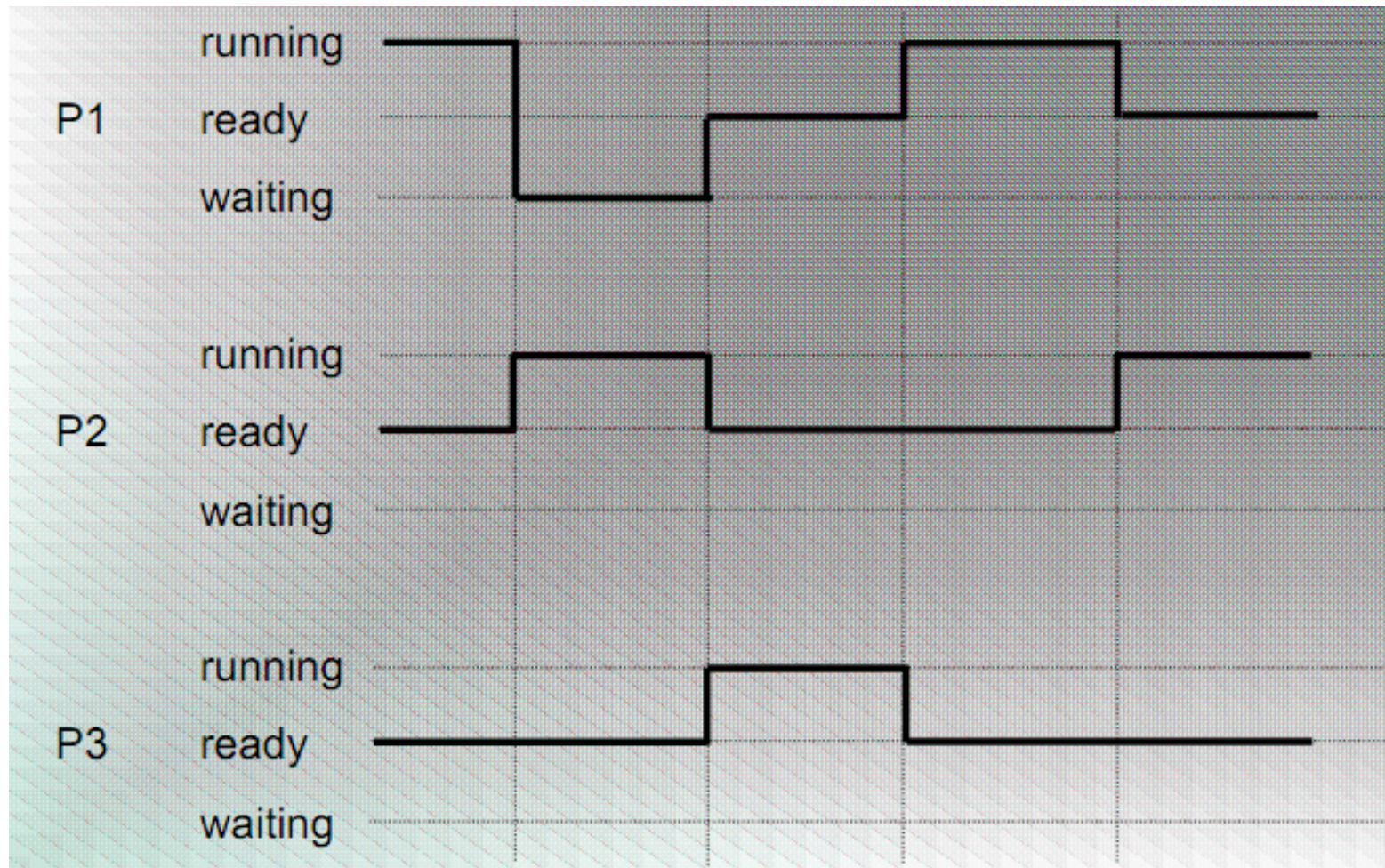
Ανταλλαγή	Το λειτουργικό σύστημα πρέπει να απελευθερώσει αρκετή μνήμη για να διαχειριστεί μια διεργασία που είναι έτοιμη να τρέξει
Άλλοι λόγοι ΛΣ	Το ΛΣ μπορεί να αναστείλει την εκτέλεση μιας βοηθητικής διεργασίας ή διεργασίας που τρέχει στο παρασκήνιο ή διεργασίας που ενδεχομένως προκαλεί προβλήματα πχ αδιέξοδο
Διαδραστική αίτηση χρήστη	Ένας χρήστης μπορεί να αναστείλει την εκτέλεση ενός προγράμματος για λόγους αποσφλαμάτωσης ή σε συνδυασμό με τον πόρο που χρησιμοποιείται
Χρονισμός	Μια διεργασία μπορεί να εκτελείται περιοδικά και μπορεί να ανασταλεί καθώς περιμένει την άφιξη της επόμενης χρονοθυρίδας
Αίτηση διεργασίας	Μια διεργασία μπορεί να αναστείλει μια διεργασία παιδιού για να την εξετάσει και να την τροποποιήσει ή για να συντονίσει την δραστηριότητα της με τις άλλες διεργασίες παιδιών



Το μοντέλο των 9 καταστάσεων (UNIX)



Αναπαράσταση Εκτέλεσης Διεργασιών (μια ΚΜΕ)



Λογικά Τμήματα Μνήμης Διεργασιών

- **Τμήμα κώδικα** (code section): περιέχει τις εντολές του προγράμματος που εκτελείται.
- **Τμήμα καθολικών δεδομένων** (static / global data section): περιέχει τα μόνιμα δεδομένα του προγράμματος που εκτελείται.
- **Τμήμα δυναμικών δεδομένων** (dynamic data section ή heap): περιέχει τα δυναμικά δεδομένα της εκτέλεσης .
- **Στοίβα** (stack): περιέχει τις προσωρινές τοπικές μεταβλητές της εκτέλεσης που δημιουργούνται λόγω κλήσεων συναρτήσεων.



Δημιουργία Νέας Διεργασίας

- Οι πατρικές διεργασίες (Parent process) δημιουργούν διεργασίες παιδιά (children processes), οι οποίες με τη σειρά τους, δημιουργούν άλλες διεργασίες, δημιουργώντας έτσι ένα δένδρο διεργασιών.
- Διαμοιρασμός πόρων.
 - Οι «γονείς και τα παιδιά» μοιράζονται όλους τους πόρους.
 - Τα παιδιά παίρνουν υποσύνολο των πόρων.
 - Οι πατρικές και οι θυγατρικές δεν μοιράζονται κανέναν πόρο.
- Εκτέλεση.
 - Οι πατρικές και οι θυγατρικές διεργασίες εκτελούνται ταυτόχρονα.
 - Οι πατρικές διεργασίες περιμένουν τον τερματισμό των παιδιών.

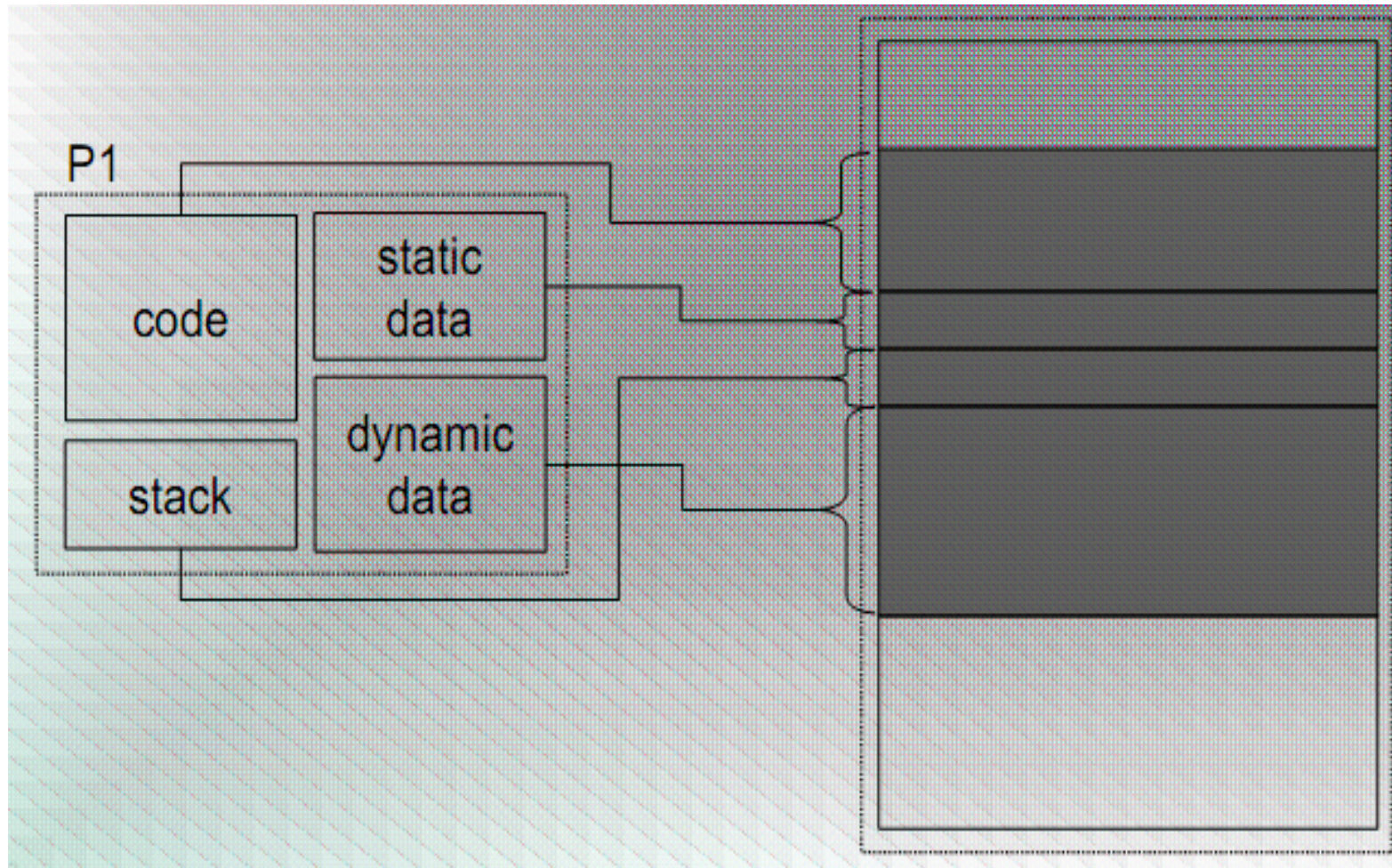


Τερματισμός Διεργασίας

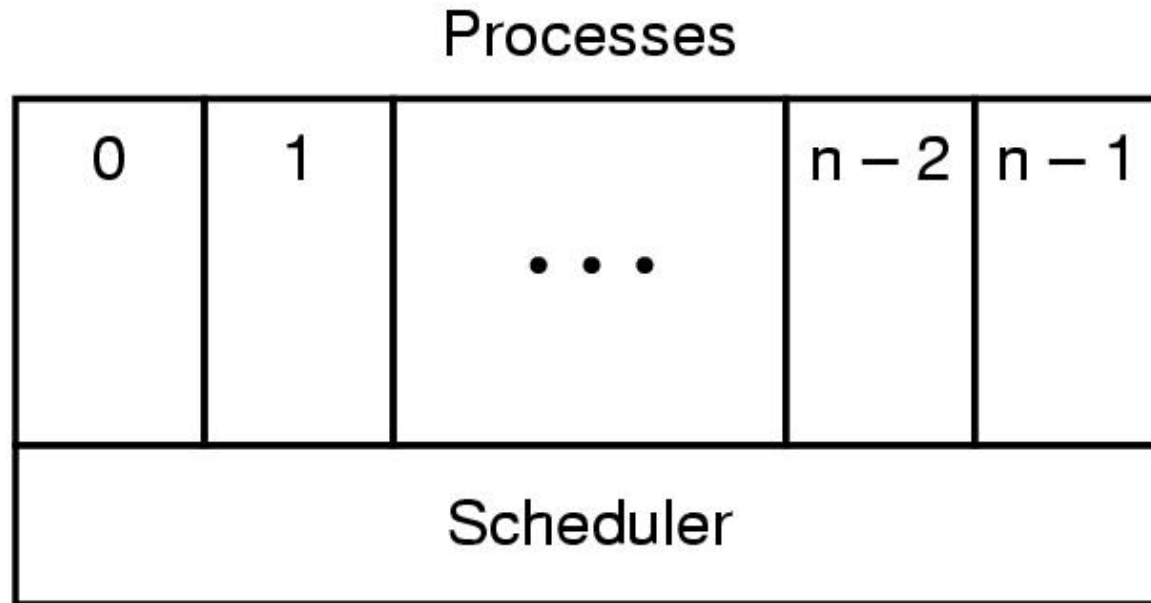
- Η διεργασία εκτελεί την τελευταία εντολή και καλεί το ΛΣ (exit).
- Μεταφορά δεδομένων προς τον πατέρα (μέσω wait).
- Οι πόροι της διεργασίας επανέρχονται στη διάθεση του ΛΣ.
- Η πατρική διεργασία μπορεί να διακόψει τη θυγατρική (abort).
- Η θυγατρική έχει υπερβεί τους πόρους που της είχαν ανατεθεί.
- Η εργασία που ανατέθηκε στο παιδί δεν χρειάζεται.
- Η διεργασία πατέρας τερματίζει την εκτέλεση της.
- Το ΛΣ δεν επιτρέπει στο παιδί να συνεχίσει εφόσον ο πατέρας τερματίσει.
- Διαδοχικοί Τερματισμοί.



Οι 4 περιοχές μνήμης μιας διεργασίας



Υλοποίηση των διεργασιών (1/2)

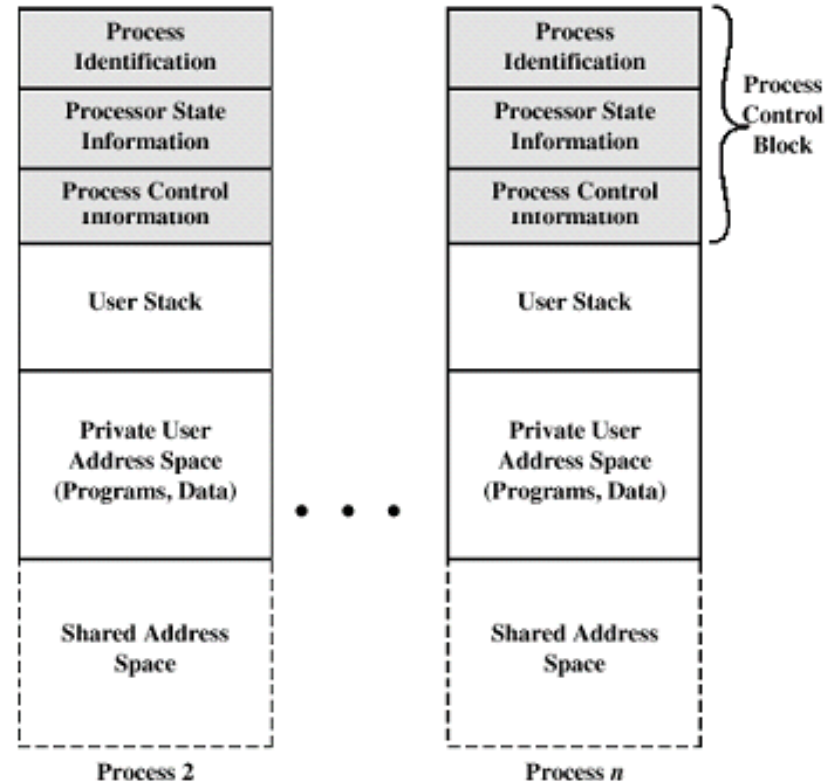


- Ο χρονοπρογραμματιστής βρίσκεται στο κατώτατο επίπεδο ενός «διεργασιακά» δομημένου (process-structured) ΛΣ {διακοπές, χρονοπρογραμματισμός}.



Μαζί με κάθε διεργασία συνδέεται ένα μπλοκ πληροφορίας

- Τι χρειάζεται μια διεργασία για να εκτελεστεί (εικόνα διεργασίας):
 - Πρόγραμμα.
 - Δεδομένα.
 - Στοίβα.
 - Μπλοκ διαδικασίας ελέγχου (πλαίσιο, για πολυπρογραμματισμού).
- Το ΛΣ πρέπει να κρατήσει:
 - Πληροφορίες σχετικά με τη τρέχουσα κατάσταση της κάθε διεργασίας και των πόρων.
 - Οι πίνακες έχουν κατασκευαστεί για κάθε οντότητα που διαχειρίζεται το ΛΣ.



Πίνακας Ελέγχου Διεργασίας

- Process Control Block (PCB)

- Βασικές πληροφορίες που συσχετίζονται με την κάθε διεργασία:
 - Κατάσταση διεργασίας (process state).
 - Μετρητής εντολών προγράμματος (program counter).
 - Καταχωρητές της ΚΜΕ (CPU registers).
 - Πληροφορίες για το χρονοπρογραμματισμό της ΚΜΕ (CPU scheduling information).
 - Πληροφορίες διαχείρισης μνήμης (memory-management information).
 - Συλλογή στοιχείων διαχείρισης (accounting information).
 - Πληροφορίες κατάστασης εισόδου/εξόδου (I/O status information).



Πίνακας Ελέγχου Διεργασίας

pointer	process state
process number	
program counter	
registers	
memory limits	
list of open files	
⋮	



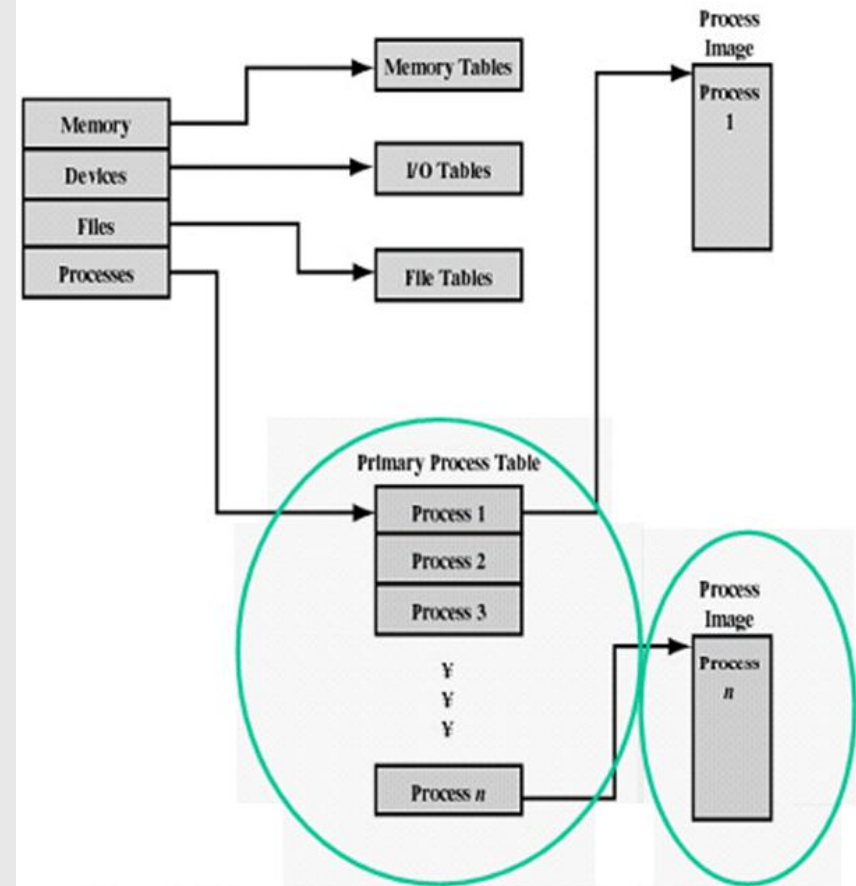
To PCB (Process Control Block)

Διαχείριση διεργασιών	Διαχείριση μνήμης	Διαχείριση αρχείων
<ul style="list-style-type: none">✓ Καταχωρητές✓ Μετρητής προγράμματος✓ Λέξη κατάστασης προγράμματος (PSW)✓ Δείκτης στοίβας✓ Κατάσταση διεργασίας✓ Προτεραιότητα✓ Παράμετροι χρονοδιαγράμματος✓ ID διεργασίας✓ Διεργασία πατέρα✓ Ομάδα διεργασίας✓ Σήματα✓ Ώρα εκκίνησης διεργασίας✓ Χρόνος χρήσης CPU✓ Χρόνος χρήσης CPU από την διεργασία παιδί✓ Χρόνος επόμενου συναγερμού	<ul style="list-style-type: none">✓ Δείκτης στις πληροφορίες τμήματος κειμένου✓ Δείκτης στις πληροφορίες του τμήματος δεδομένων✓ Δείκτης στις πληροφορίες του τμήματος στοίβας	<ul style="list-style-type: none">✓ Κατάλογος ρίζας✓ Κατάλογος εργασίας✓ Περιγραφείς αρχείων✓ ID χρηστών✓ ID ομάδας



Ο πίνακας διεργασιών και άλλοι πίνακες του ΛΣ

- Απαραίτητα χαρακτηριστικά για την διαχείριση της οντότητας:
 - ID διεργασίας.
 - Κατάσταση διεργασίας.
 - Περιοχή στη μνήμη.
 - Άλλα: μπλοκ ελέγχου διεργασίας (PCB).
- Άλλοι πίνακες διατηρούν πληροφορίες συγκεκριμένες με πόρους:
 - Πως είναι διατεταγμένη η κύρια/δευτερεύουσα μνήμη.
 - Κατάσταση συσκευών Ε/Ε, μνήμη buffer.
 - Κατάσταση αρχείων.

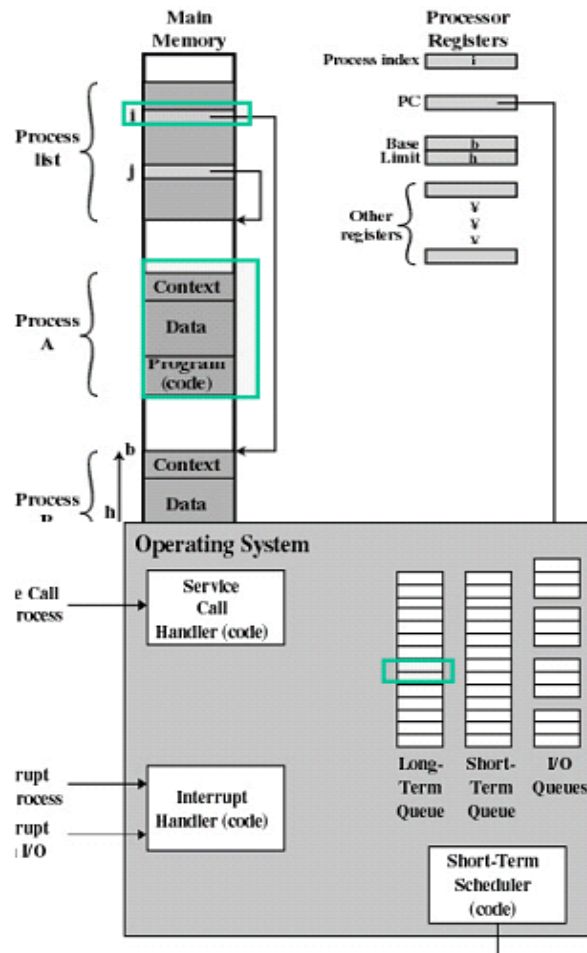


Τι γίνεται κατά τη δημιουργία μιας διεργασίας; (1/2)

- Εκχωρείται ένας μοναδικός αριθμός αναγνωριστικού (process id).
- Παραχωρείται χώρος στη μνήμη.
- Αρχικοποιείται το PCB.
- Τοποθετείται η διεργασία σε διάφορες ουρές (επόμενο slide).



Τι γίνεται κατά τη δημιουργία μιας διεργασίας; (2/2)



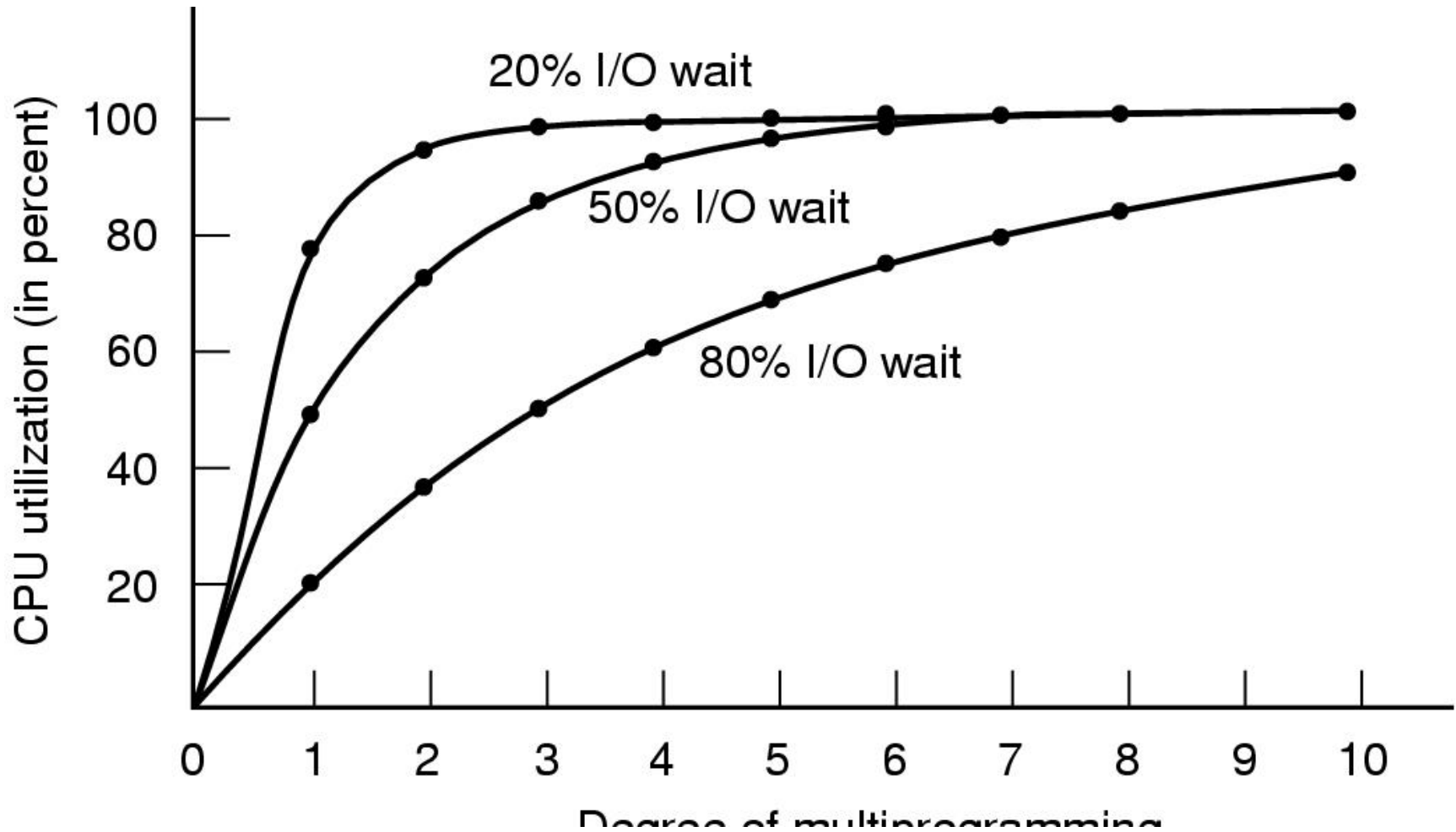
Υλοποίηση των διεργασιών (2/2)

Βήματα που εκτελούνται, όταν συμβαίνει κάποια Διακοπή (interrupt):

1. Το υλικό τοποθετεί στη στοίβα το μετρητή προγράμματος.
2. Το υλικό φορτώνει τον νέο μετρητή προγράμματος από το σήμα διακοπής.
3. Η διεργασία της γλώσσας Assembly αποθηκεύει τους καταχωρητές
4. Η διεργασία της γλώσσας Assembly δημιουργεί νέα στοίβα.
5. Τρέχει η διαδικασία διακοπής της C.
6. Ο χρονοπρογραμματιστής αποφασίζει μια διεργασία είναι η επόμενη προς εκτέλεση.
7. Η διαδικασία της C επιστρέφει στον κώδικα Assembly.
8. Η διεργασία της γλώσσας Assembly εκκινεί τη νέα τρέχουσα διεργασία.



Μοντελοποίηση Πολυπρογραμματισμού



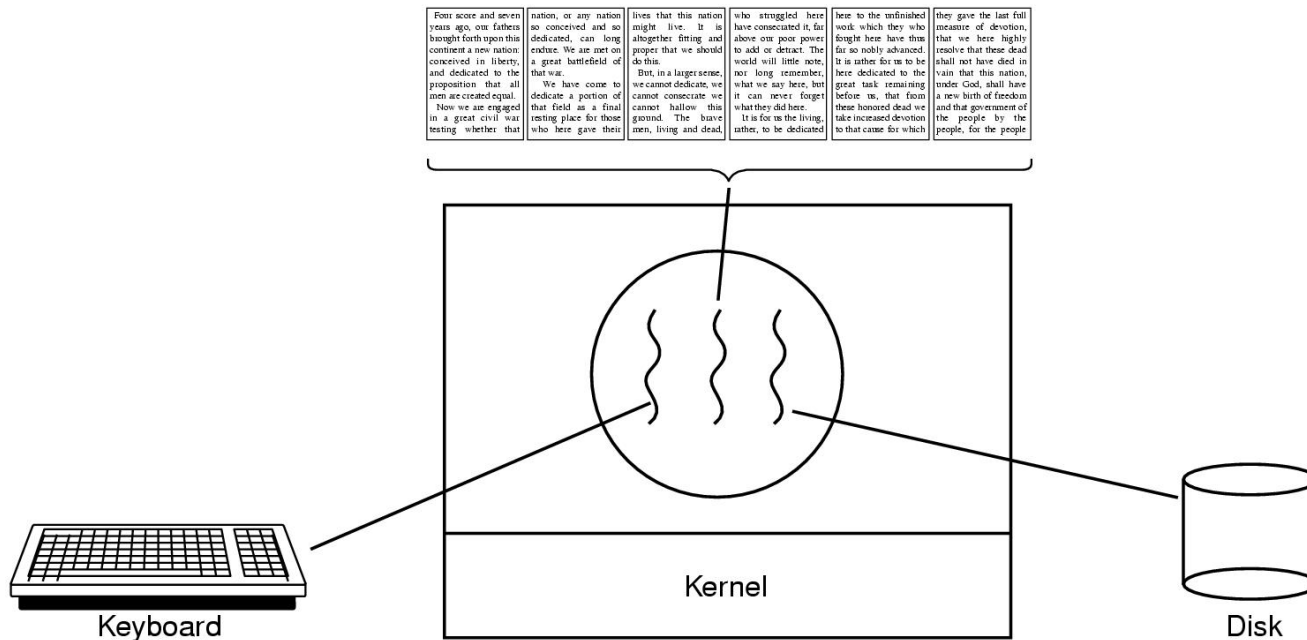
Η Λειτουργία της Στοίβας

- Η στοίβα χρησιμοποιείται για την προσωρινή αποθήκευση τοπικών μεταβλητών κατά την κλήση υποπρογραμμάτων (συναρτήσεων) καθώς και για την μετάδοση παραμέτρων και αποτελεσμάτων ανάμεσα στα υποπρογράμματα.
- Κάθε νέα κλήση δεσμεύει χώρο στην στοίβα για την αποθήκευση των αντίστοιχων τοπικών μεταβλητών του υποπρογράμματος. Αυτός ο χώρος αποδεσμεύεται όταν επιστρέψει το υποπρόγραμμα.
- Αλυσιδωτές κλήσεις υποπρογραμμάτων, η μια μέσα από την άλλη, έχουν σαν αποτέλεσμα την επέκταση της στοίβας.
- Αν γίνουν πάρα πολλές αλυσιδωτές κλήσεις, η στοίβα μπορεί να γεμίσει (stack overflow) με αποτέλεσμα να τερματιστεί η διεργασία.
- Κλασικό παράδειγμα είναι η ατέρμονη αναδρομή.



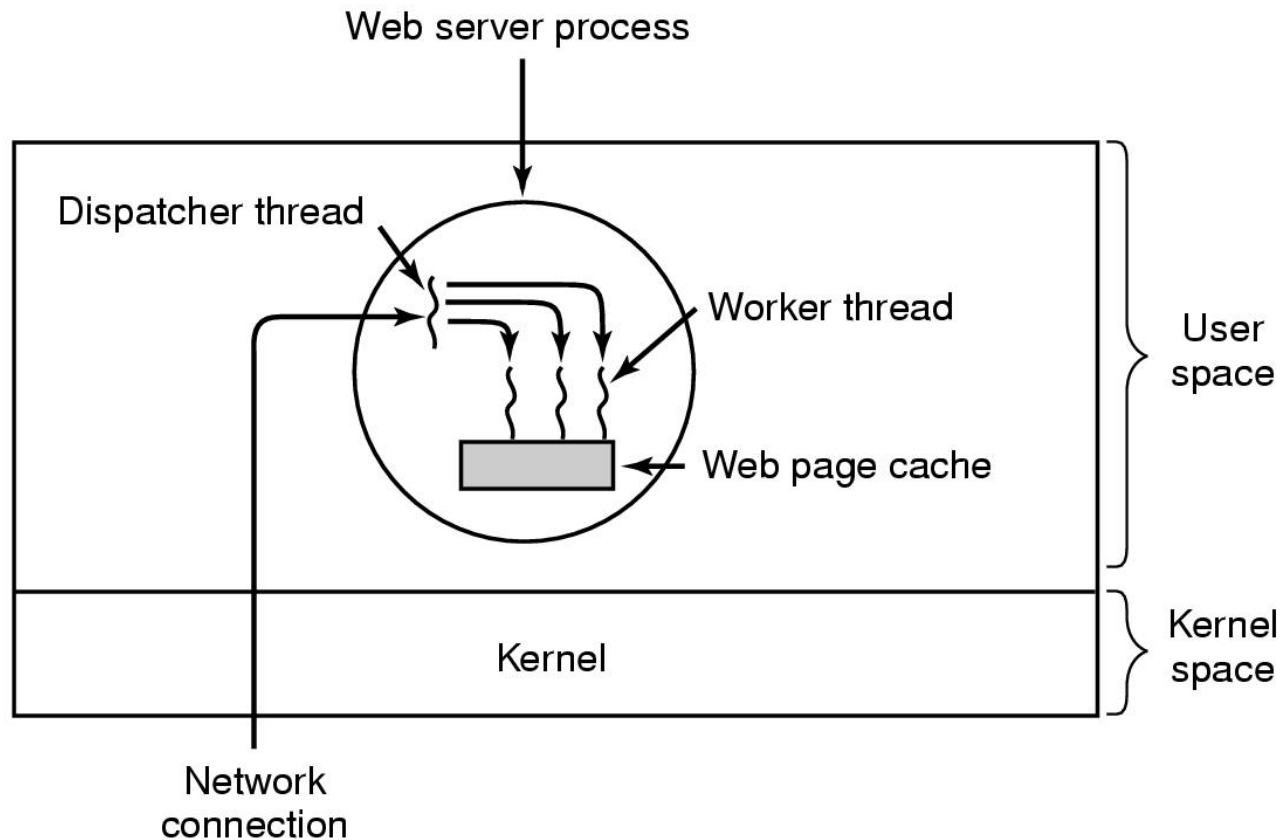
Νήματα (Threads) (1/2)

- Παράδειγμα: Επεξεργαστής κειμένου με 3 νήματα.



Νήματα (Threads) (2/2)

- Παράδειγμα : web browser.



Η έννοια της νημάτωσης (threading)

- Ένα νήμα (thread) είναι **μια απλή ροή δεδομένων διαμέσου του επεξεργαστή του συστήματος**. Κάθε εφαρμογή παράγει το δικό της ή τα δικά της νήματα εξαρτώμενα από τον τρόπο που αυτή εκτελείται. Στα πολυδιεργασιακά συστήματα ένας επεξεργαστής μπορεί να διαχειριστεί ένα νήμα κάθε στιγμή. Έτσι, το σύστημα εναλλάσσεται ταχύτατα μεταξύ των νημάτων για να επεξεργαστεί τα δεδομένα δίνοντας την αίσθηση της συνεξέλιξης των νημάτων.
- Νήμα: μια μονάδα εργασίας που διεκπεραιώνεται και μπορεί να διακόπτεται. Περιλαμβάνει:
 - το μετρητή προγράμματος,
 - το δείκτη στοίβας,
 - τη δική της περιοχή δεδομένων.



Τα νήματα ενισχύουν τον πολυπρογραμματισμό

- Ένα νήμα εκτελείται **σειριακά** και είναι **διακοπτόμενο**, ώστε ο επεξεργαστής να μπορεί να επιστρέφει σε άλλο νήμα.
- Ένα πρόγραμμα μπορεί να περιέχει αρκετά στοιχεία που διαμοιράζονται δεδομένα και μπορούν να εκτελούνται συγχρόνως. Π.χ. ένας Web browser μπορεί να περιέχει διαφορετικά συστατικά για την ανάγνωση ιστοσελίδων σε μορφή HTML, την ανάκτηση των συστατικών τους (εικόνες, video κλπ) και την εμφάνιση των σελίδων στο παράθυρο του browser. **Αυτά τα συστατικά του προγράμματος που εκτελούνται ανεξάρτητα, αλλά υλοποιούνται ως λειτουργίες σε μια κοινή περιοχή μνήμης, ονομάζονται νήματα (threads).**

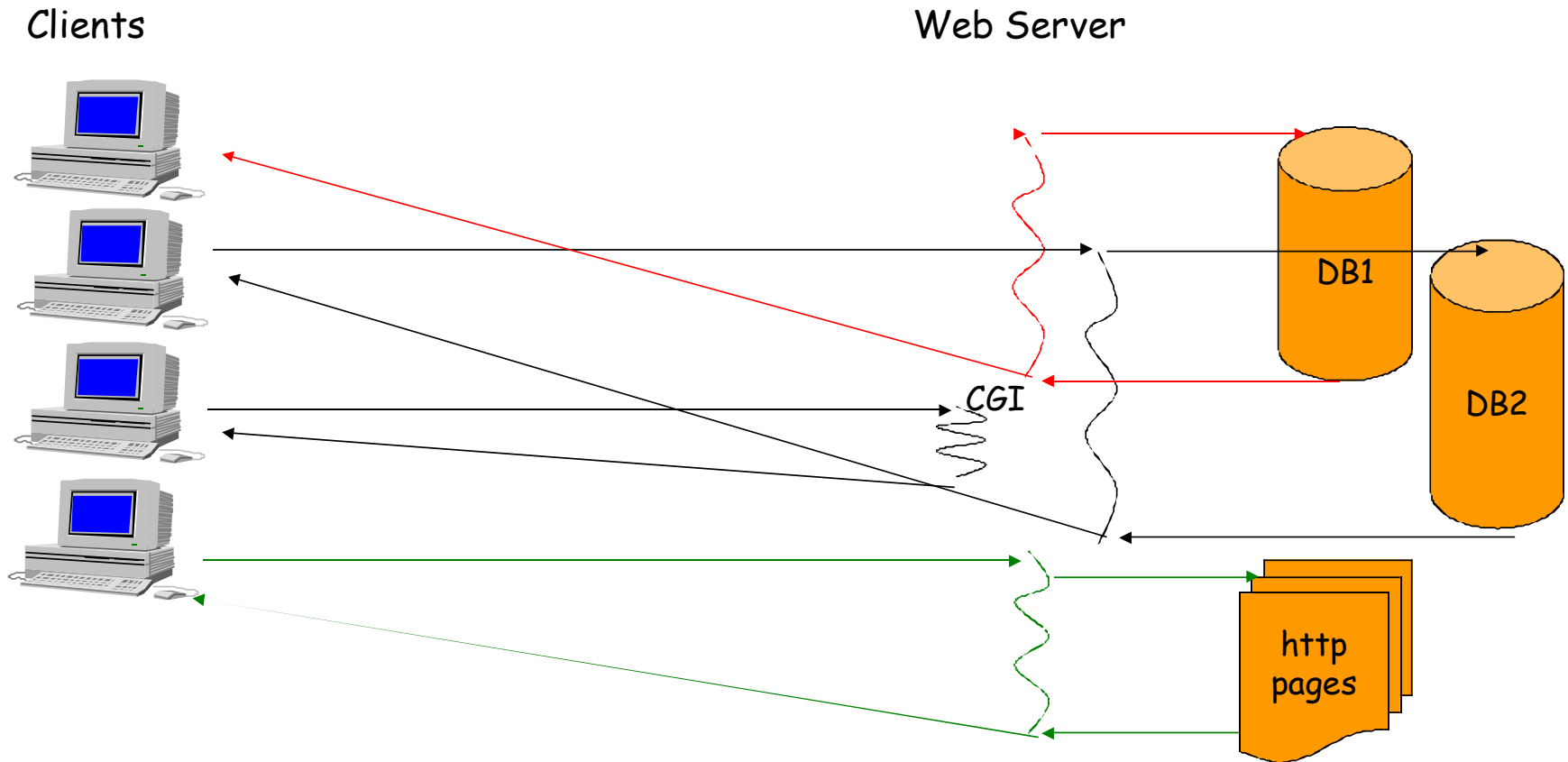


Τι είναι ποληνυμάτωση;

- Διεργασία: μια συλλογή νημάτων μαζί με συσχετιζόμενους πόρους του συστήματος.
- Η διάσπαση μιας εφαρμογής σε πολλά νήματα έχει ως αποτέλεσμα το **χρονισμό** των γεγονότων της εφαρμογής.
- Τα νήματα προσφέρουν έναν εναλλακτικό τρόπο λειτουργίας σε σχέση με το μοντέλο της διεργασίας.
- Υπάρχει δυσκολία και πολυπλοκότητα όταν χρησιμοποιείται το μοντέλο διεργασίας, που επιπλέον δεν είναι αρκετά αποτελεσματικό.
- Η πολυνημάτωση (multi-threading) είναι χρήσιμη σε εφαρμογές που εκτελούν πολλές ανεξάρτητες μεταξύ τους διεργασίες που δεν χρειάζεται να είναι σε σειρά, π.χ. ένας web server.



Τα νήματα χρησιμοποιούνται ευρέως σε διακομιστές



Παράδειγμα κώδικα νημάτων

```
while (TRUE) {  
    get_next_request(&buf);  
    handoff_work(&buf);  
}
```

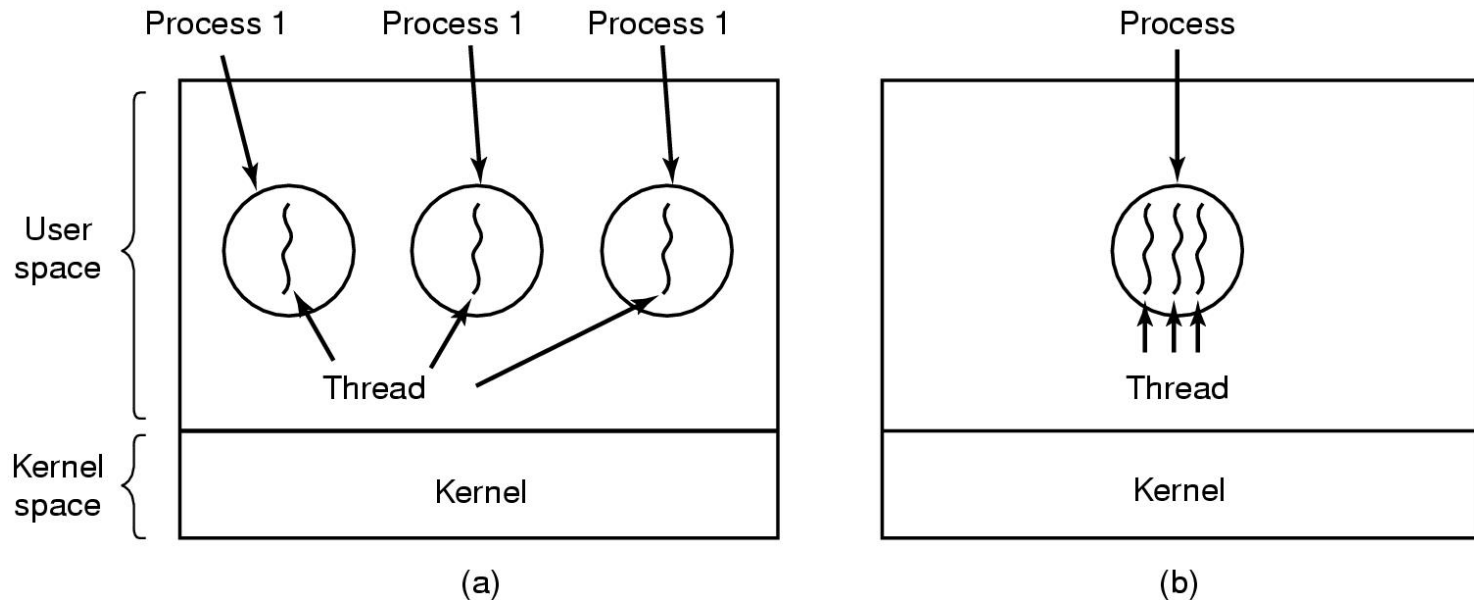
(a)

```
while (TRUE) {  
    wait_for_work(&buf)  
    look_for_page_in_cache(&buf, &page);  
    if (page_not_in_cache(&page))  
        read_page_from_disk(&buf, &page);  
    return_page(&page);  
}
```

(b)



Το μοντέλο των νημάτων (1/2)



- (a) Τρεις διεργασίες η κάθε μία με ένα νήμα.
- (b) Μια διεργασία με τρία νήματα.



Το μοντέλο νήματος

Το μοντέλο αυτό διαιρεί τη διεργασία σε δύο τμήματα:

Η διεργασία: περιέχει τους πόρους που χρησιμοποιούνται σε όλη τη διάρκεια του προγράμματος όπως:

- Καθολικές (global) εντολές.
- Καθολικά (global) δεδομένα.

Το νήμα: περιέχει πληροφορίες που σχετίζονται με την κατάσταση εκτέλεσης όπως:

- Program counter.
- Stack.



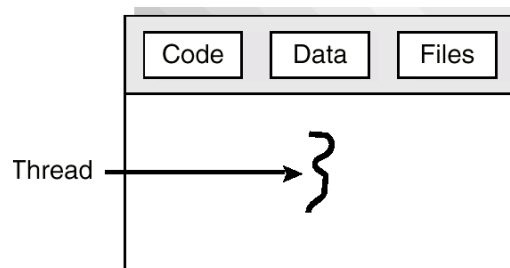
Διαφορές Διεργασίες και Νήματα

Διεργασίες:

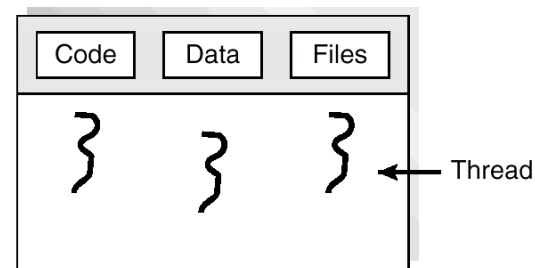
- Είναι οντότητες που δεσμεύουν οποιονδήποτε πόρο τις αφορά όπως τον χώρο διευθύνσεων, τα δεδομένα, τον κώδικα και τα αρχεία.
- Θεωρούνται ως heavyweight σε σχέση με τη δημιουργία, τον τερματισμό και την εναλλαγή πλαισίου.

Νήματα:

- Διαμοιράζονται τον χώρο διευθύνσεων, τα δεδομένα, τον κώδικα και τα αρχεία εφόσον ανήκουν στην ίδια διεργασία.
- Θεωρούνται ως lightweight σε σχέση με τη δημιουργία, τον τερματισμό και την εναλλαγή πλαισίου.



Single-threaded



Multi-threaded

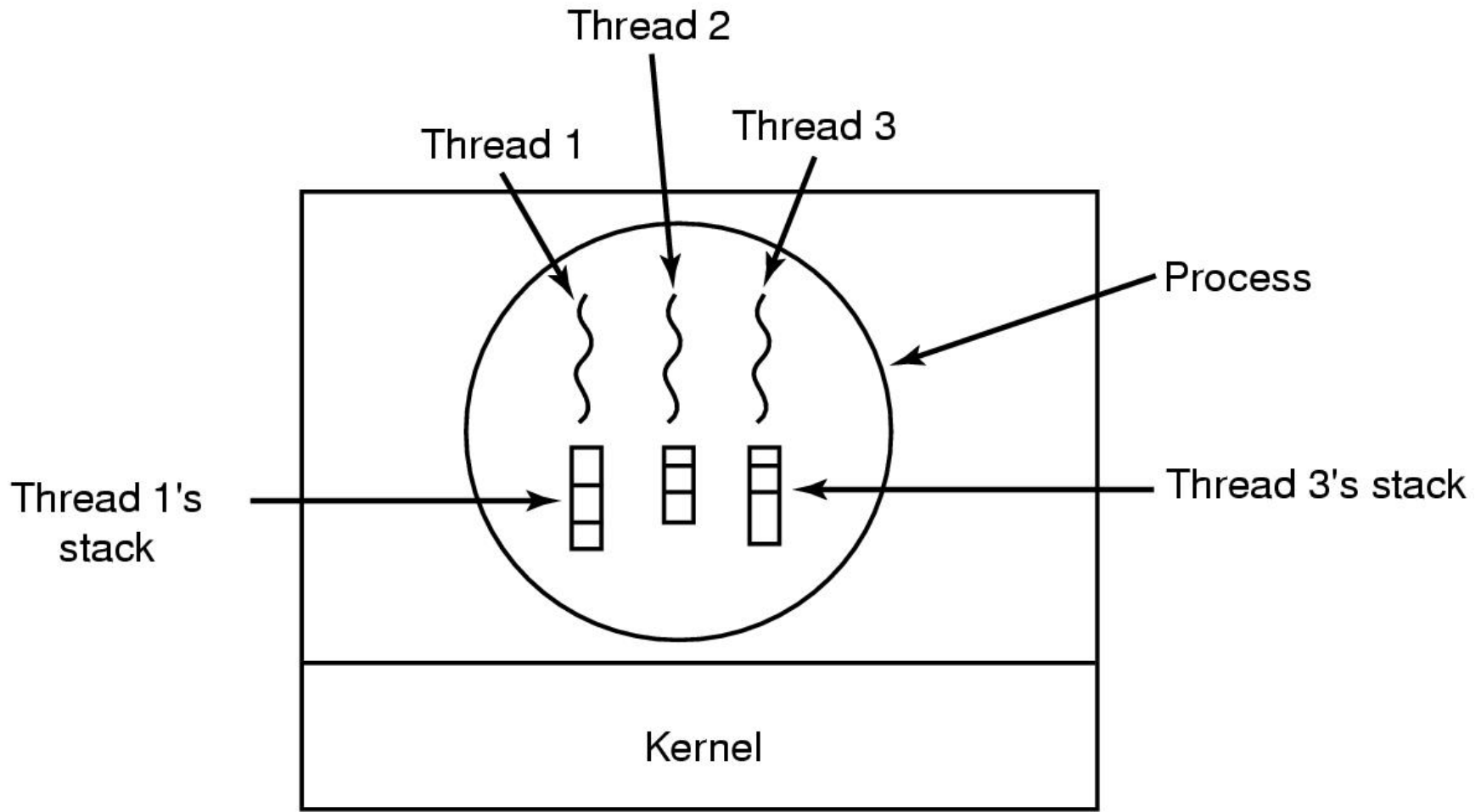
Το μοντέλο των νημάτων (2/2)

- Η πρώτη στήλη περιέχει κάποια αντικείμενα που είναι κοινά από όλα τα νήματα σε μια διεργασία ενώ η δεύτερη στήλη περιέχει κάποια αντικείμενα που είναι ιδιωτικά σε κάθε νήμα.

Ανα διεργασία	Ανα νήμα
Χώρος διεύθυνσης	Μετρητής προγράμματος (PC)
Καθολικές μεταβλητές	Καταχωρητές
Άνοιγμα αρχείων	Στοιβά
Διεργασίες παιδιών	Κατάσταση
Εκκρεμής συναγερμοί	
Σήματα και διαχειριστές σημάτων	
Πληροφορίες	



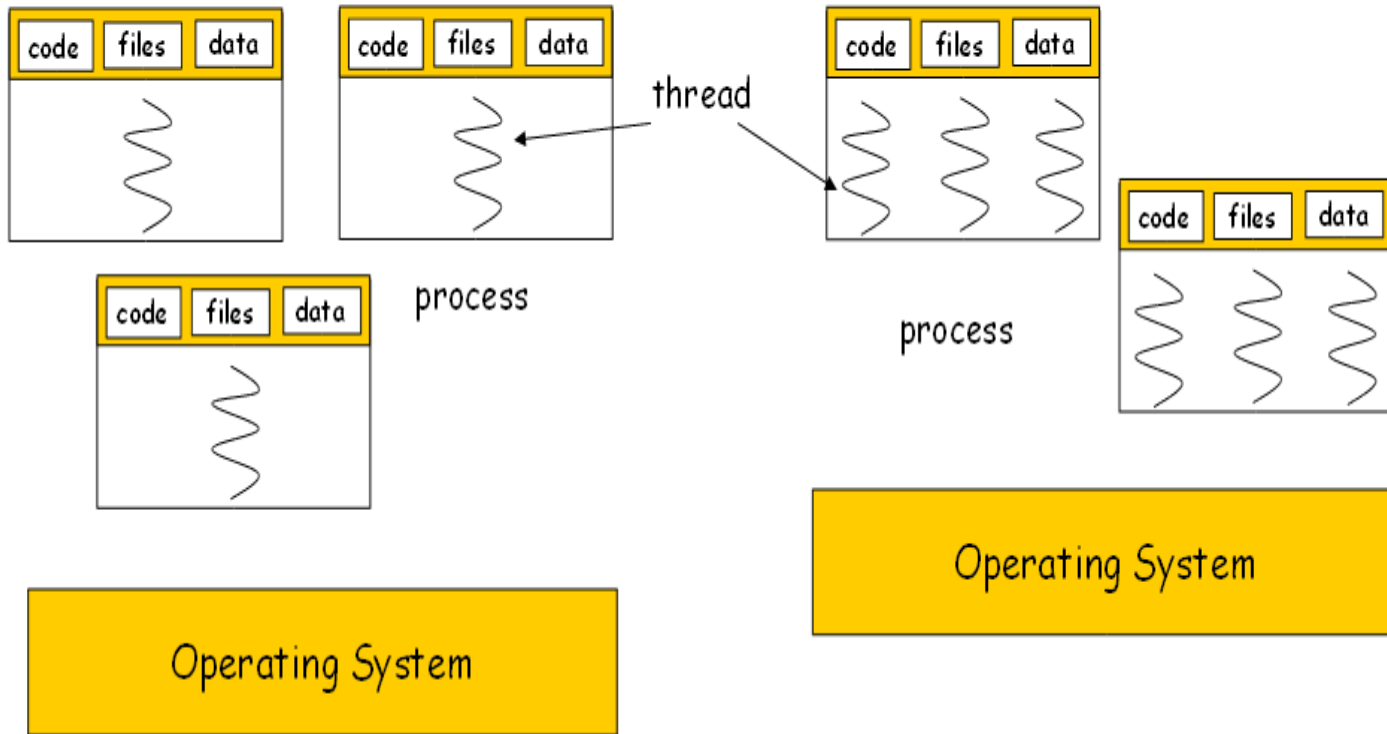
Κάθε νήμα έχει το δικό του σωρό



Μοντέλα ΛΣ ως προς τα νήματα

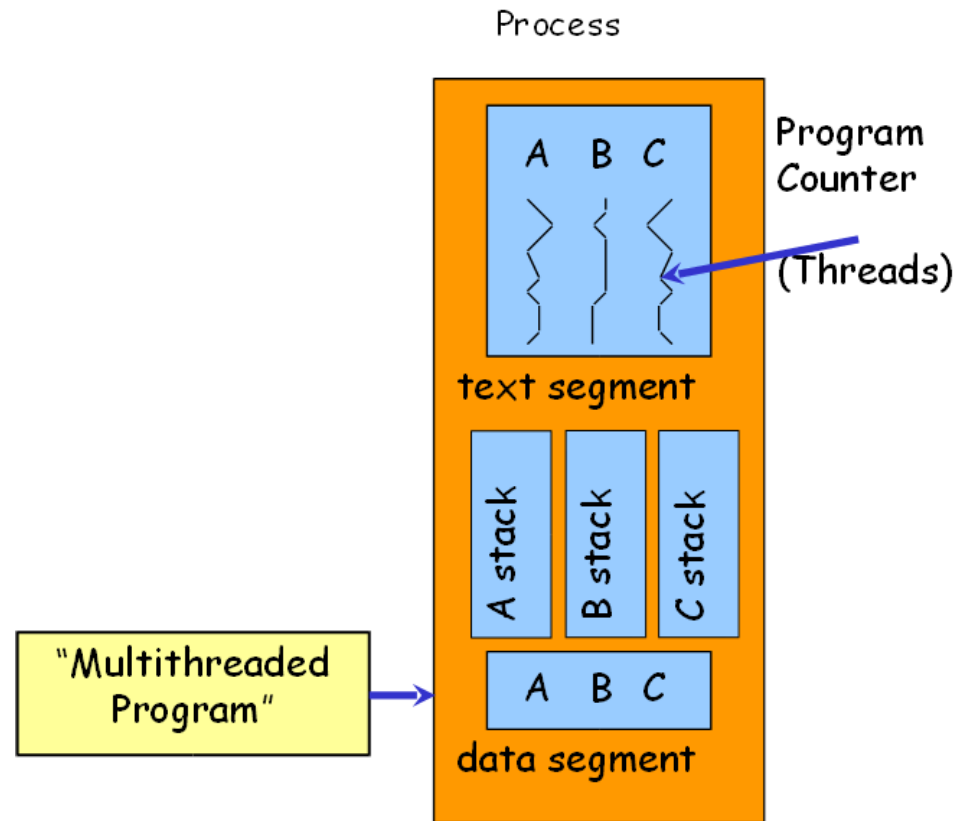
Process-oriented computing

Multithreaded computing

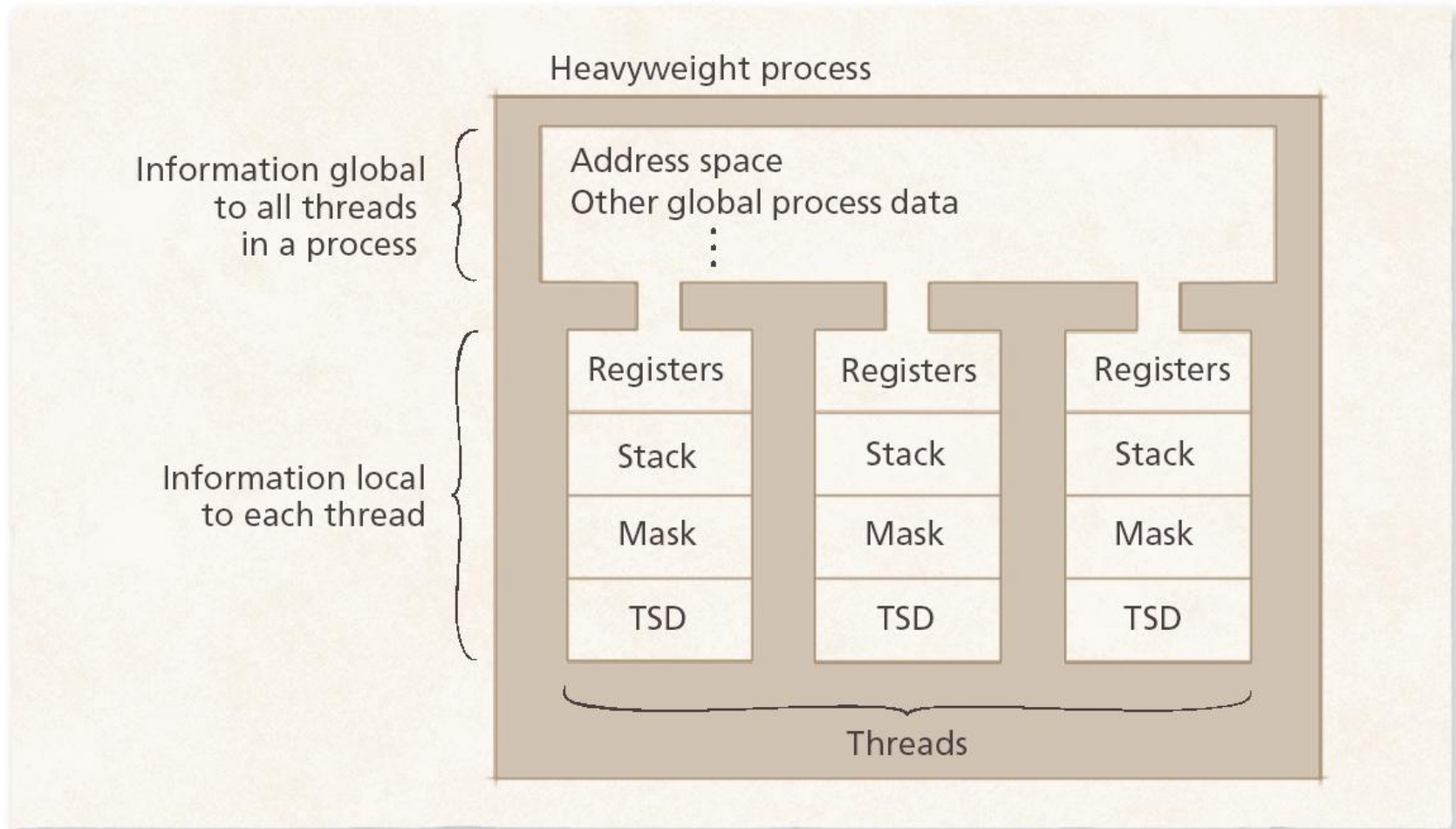


Threads (Lightweight Processes)

- Βασική μονάδα χρήσης της CPU:
- Κατέχει:
 - program counter.
 - register set.
 - stack space.
- Διαμοιράζεται:
 - code section.
 - data section.
 - OS resources.



Κοινά και Διαφορετικά Στοιχεία σε νήματα



Παράδειγμα προγράμματος σε C

```
void do_one_thing(int *);
void do_another_thing(int *);
void do_wrap_up(int, int);
int r1=0, r2=0;

main()
{
    do_one_thing(&r1);
    do_another_thing(&r2);
    do_wrap_up(r1, r2);
}

void do_one_thing(int *pnum_times)
{
    int i, j, x;

    for(i = 0; i < 4; i++)
    {
        printf("doing one thing\n");
        for (j = 0; j < 10000; j++)
        {
            x = x + i;
        }
        (*pnum_times)++;
    }
}
```

```
void do_another_thing(int *pnum_times)
{
    int i, j, x;

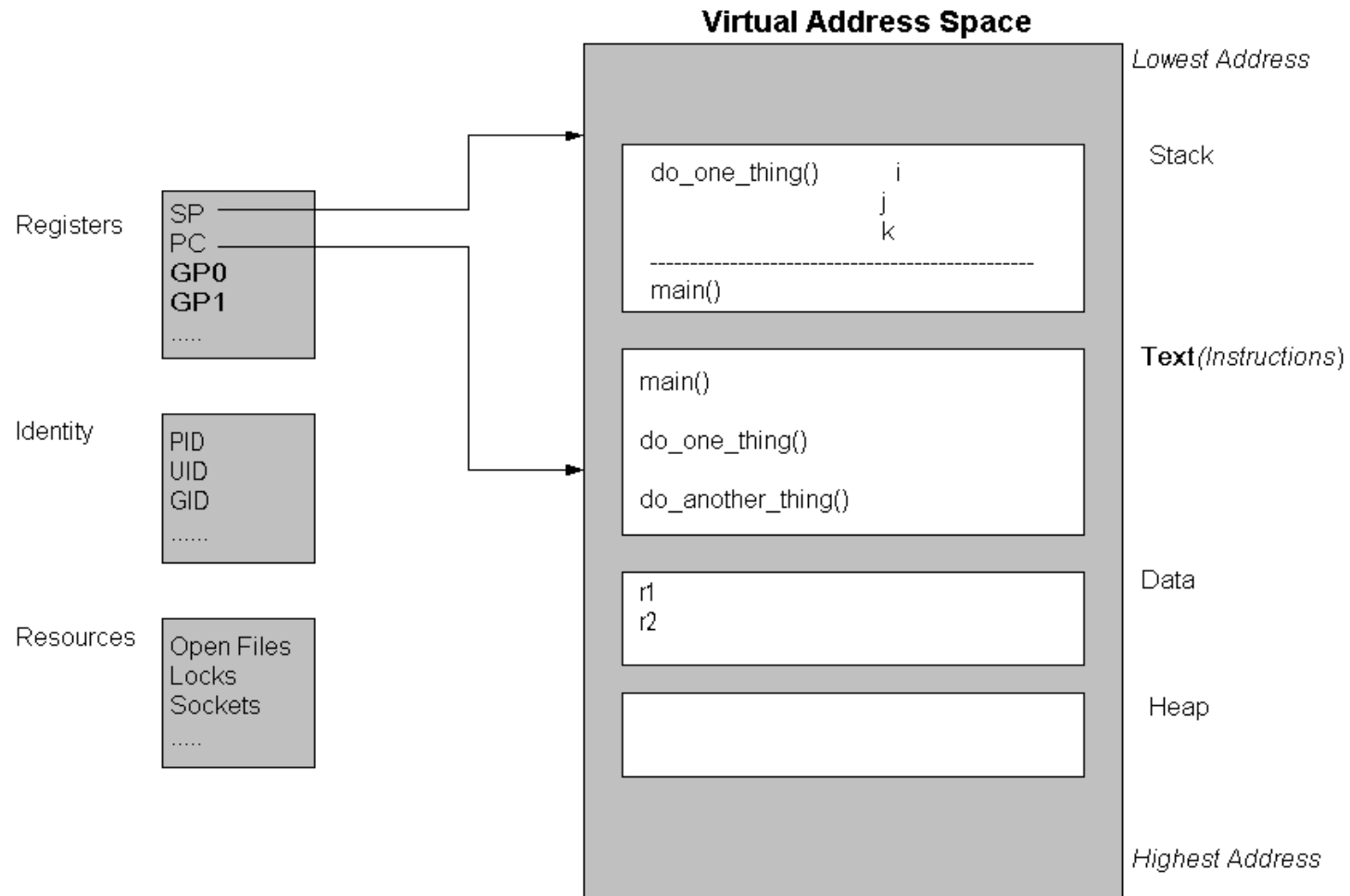
    for (i = 0; i < 4; i++)
    {
        printf("doing another thing \n");
        for(j = 0; j < 10000; j++)
        {
            x = x + i;
        }
        (*pnum_times)++;
    }
}

void do_wrap_up(int one_times, int another_times)
{
    int total;

    total = one_times + another_times;
    printf("wrap up: one thing %d, another %d, total %d\n",
           one_times, another_times, total);
}
```



Το πρόγραμμα ως μια διεργασία



Το πρόγραμμα ως διεργασία με ένα νήμα (1/2)

```
void do_one_thing(int *);
void do_another_thing(int *);
void do_wrap_up(int, int);

int r1=0, r2=0;

main()
{
    /* create and execute thread */
    CreateThread( do_one_thing );

    /* execute do_another_thing in main process */
    do_another_thing(&r2);

    /* wait for thread to complete */
    wait_for_thread();

    do_wrap_up(r1, r2);
}
```

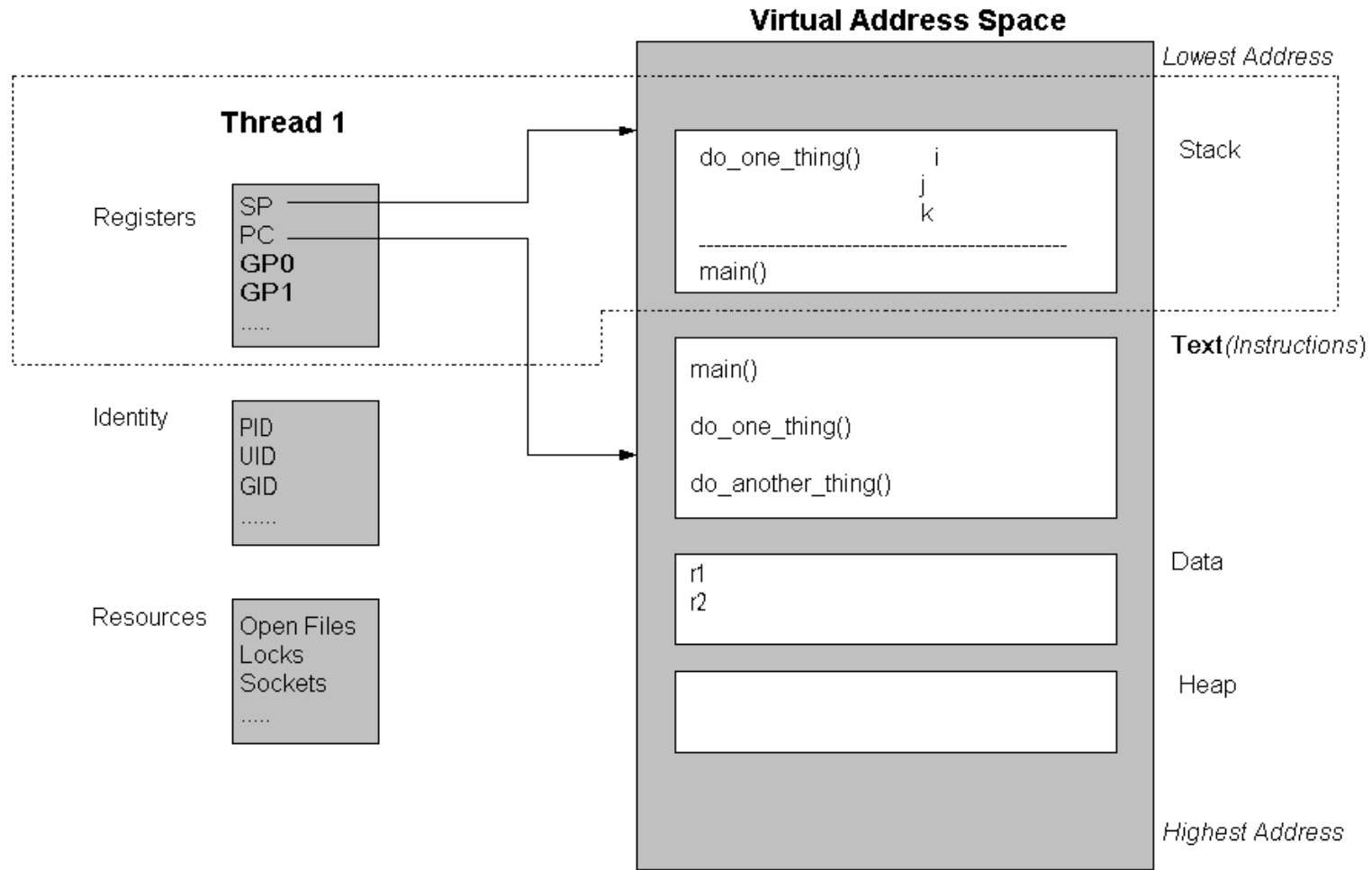


Το πρόγραμμα ως διεργασία με ένα νήμα (2/2)

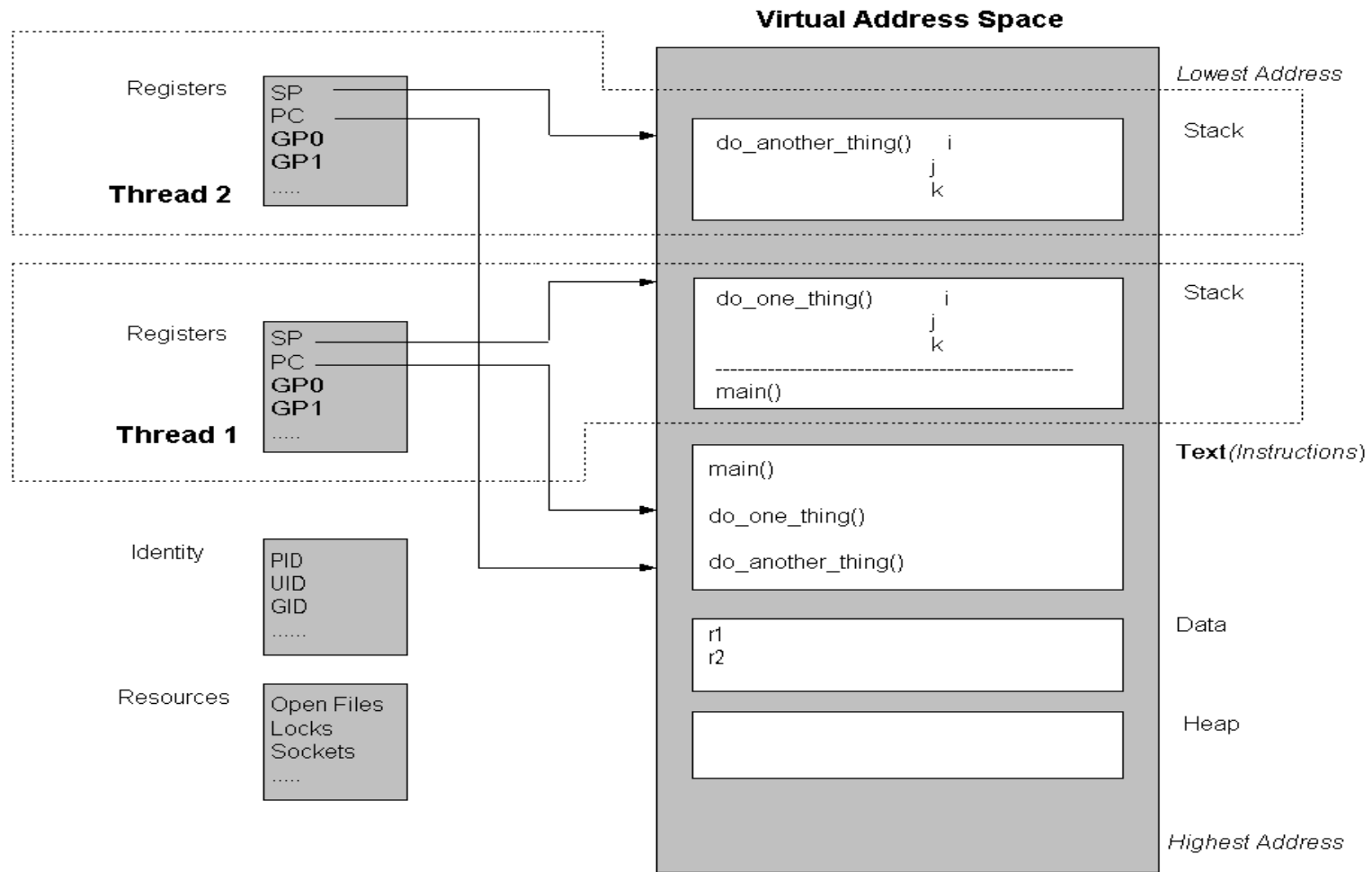
- Οι καταχωρητές είναι τμήμα του νήματος:
 - Program counter.
 - Stack pointer.
 - General purpose registers.
- Οτιδήποτε άλλο ανήκει στη διεργασία:
 - Πόροι του Λ.Σ.
 - Open files.
 - Locks.
 - Sockets, κλπ.
 - Πληροφορίες για τη διεργασία.
 - PID, UID, GID.



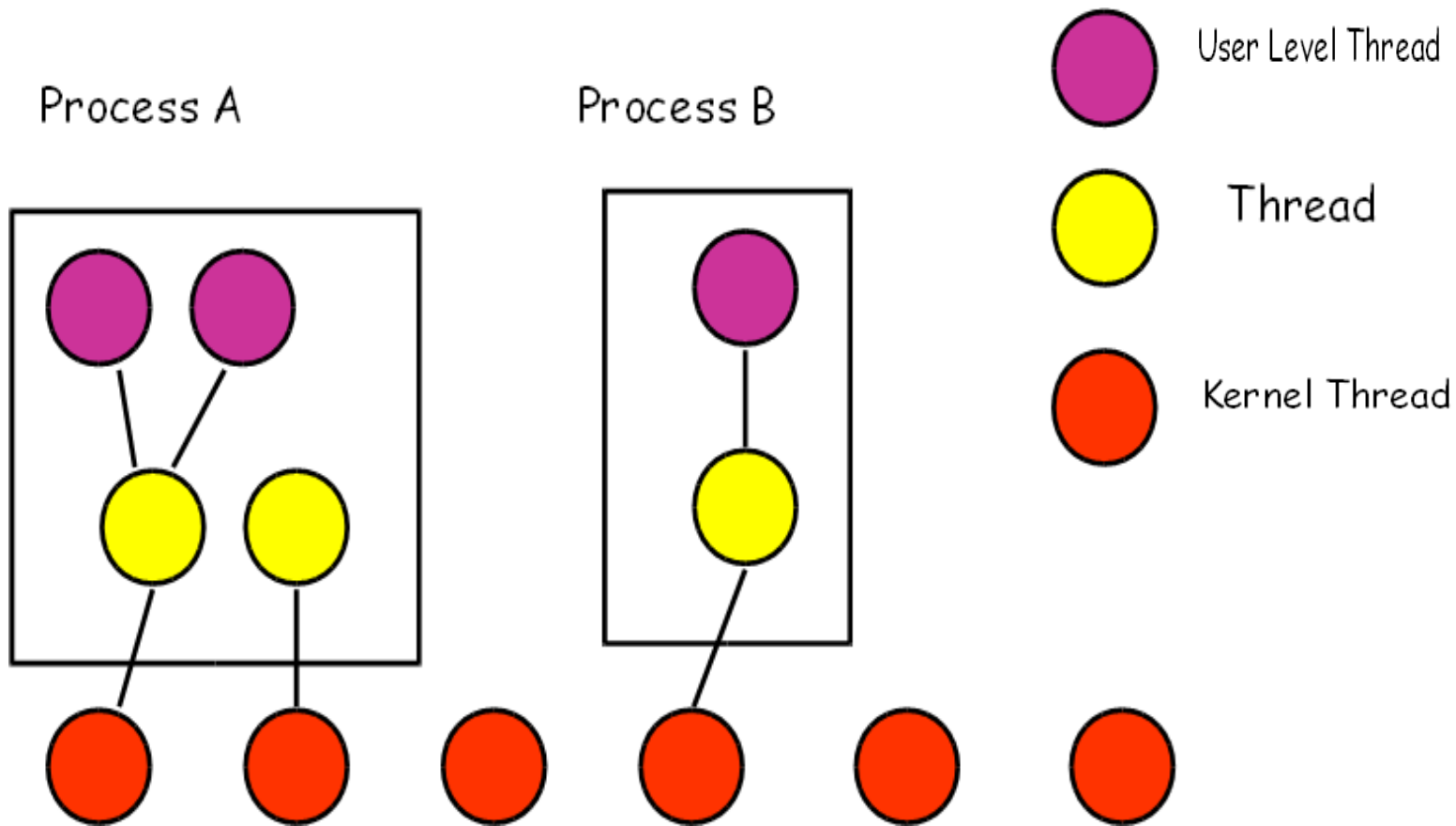
Μετατροπή σε νήματα (1/2)



Μετατροπή σε νήματα (2/2)



Μοντέλα (επίπεδα) νημμάτωσης



Υλοποίηση των POSIX νημάτων

Thread call	Description
Pthread_create	Create a new thread
Pthread_exit	Terminate the calling thread
Pthread_join	Wait for a specific thread to exit
Pthread_yield	Release the CPU to let another thread run
Pthread_attr_init	Create and initialize a thread's attribute structure
Pthread_attr_destroy	Remove a thread's attribute structure



Παράδειγμα pthreads

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>

#define NUMBER_OF_THREADS 10

void *print_hello_world(void *tid)
{
    /* This function prints the thread's identifier and then exits. */
    printf("Hello World. Greetings from thread %d0, tid);
    pthread_exit(NULL);
}

int main(int argc, char *argv[])
{
    /* The main program creates 10 threads and then exits. */
    pthread_t threads[NUMBER_OF_THREADS];
    int status, i;

    for(i=0; i < NUMBER_OF_THREADS; i++) {
        printf("Main here. Creating thread %d0, i);
        status = pthread_create(&threads[i], NULL, print_hello_world, (void *)i);

        if (status != 0) {
            printf("Oops. pthread_create returned error code %d0, status);
            exit(-1);
        }
    }
    exit(NULL);
}
```



Νήματα χρήστη και πυρήνα

Νήματα χρήστη

Η διαχείριση των νημάτων γίνεται από τη βιβλιοθήκη των νημάτων χρήστη.

Παραδείγματα:

- POSIX Pthreads.
- Win32 threads.
- Solaris threads.

Ο πυρήνας δεν ενδιαφέρεται για το πλήθος των νημάτων χρήστη.

Νήματα πυρήνα:

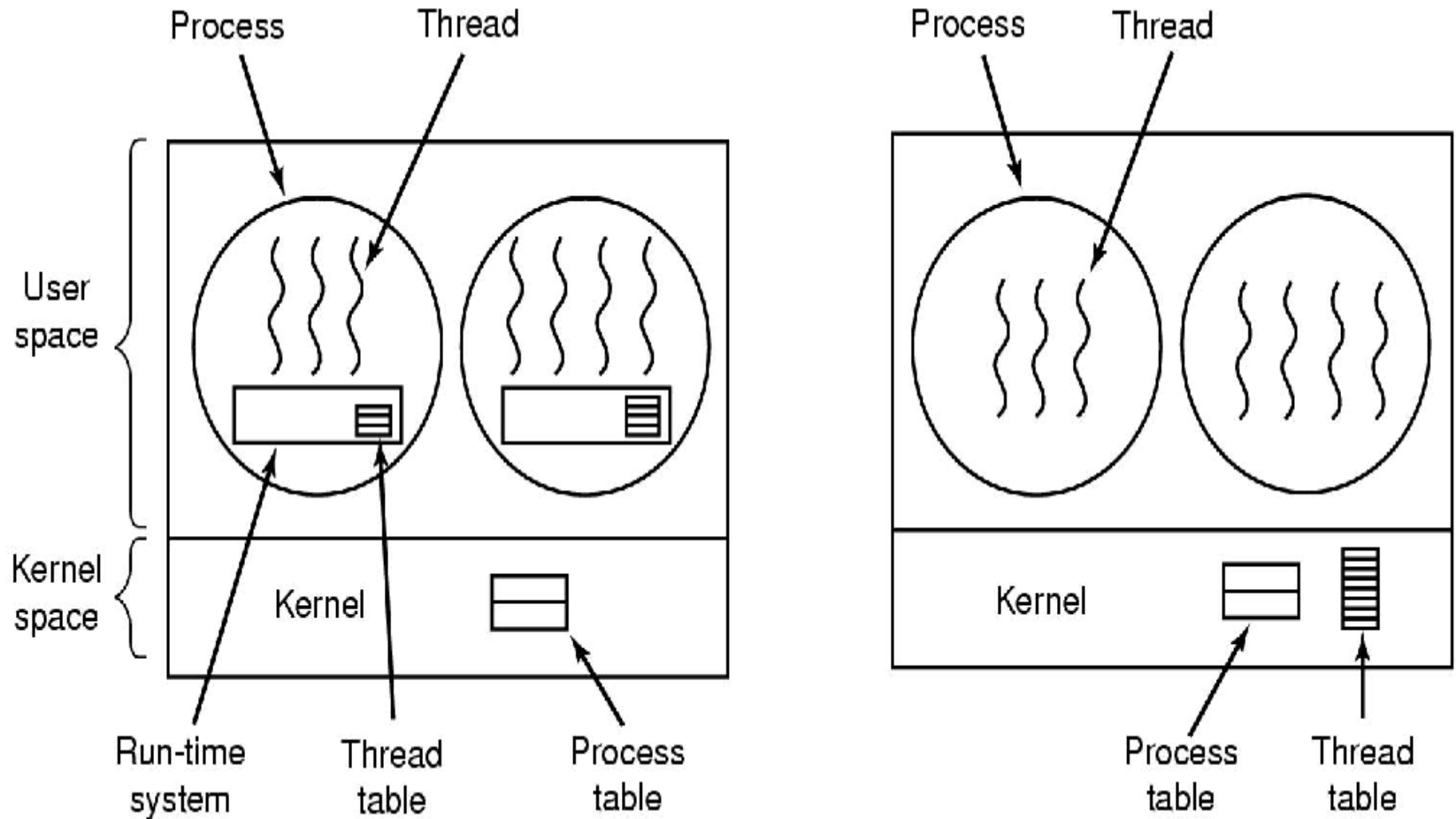
Υποστηρίζονται από τον πυρήνα.

Παραδείγματα:

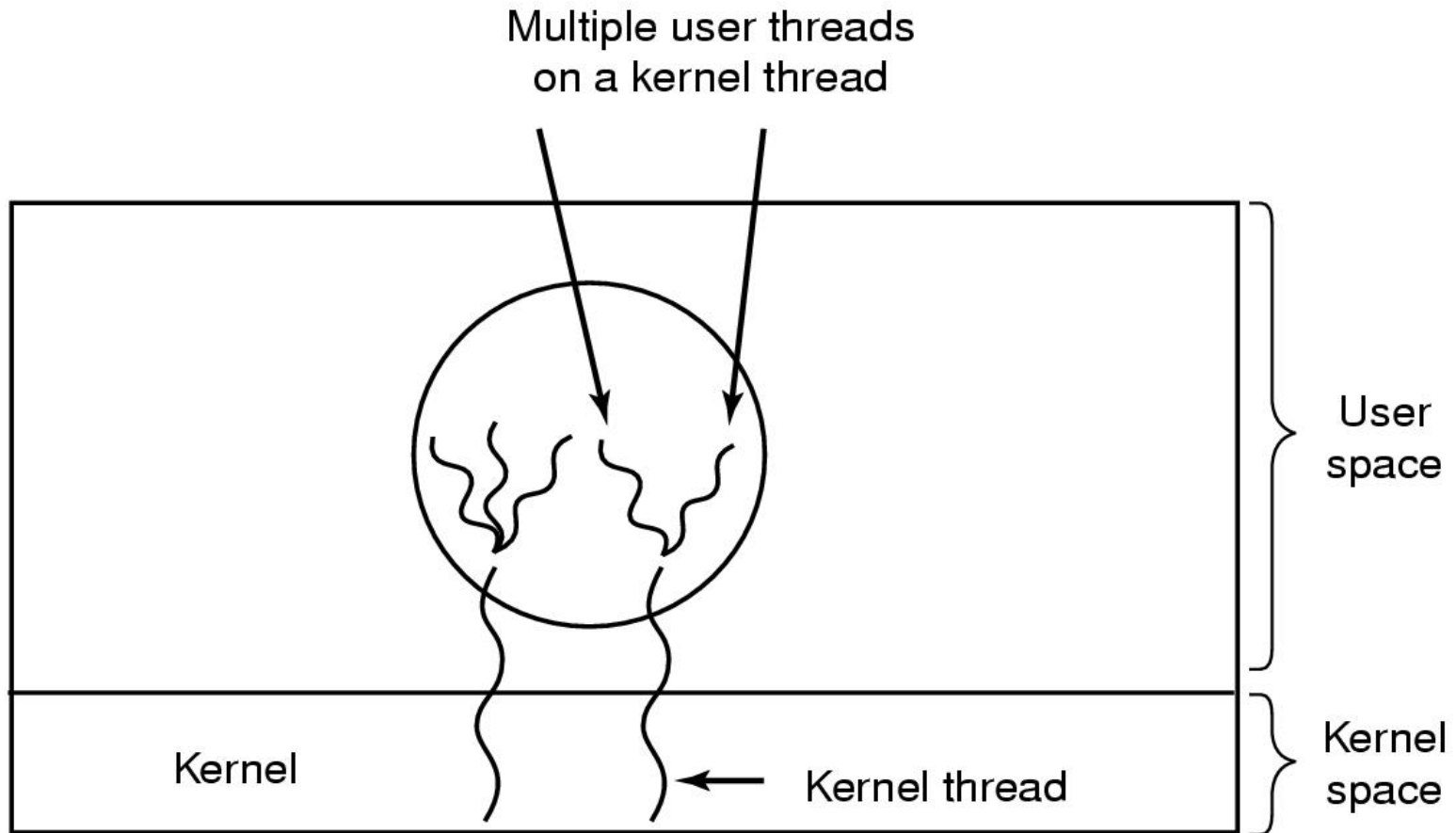
- Linux.
- Windows XP/2000.
- Solaris lightweight processes.



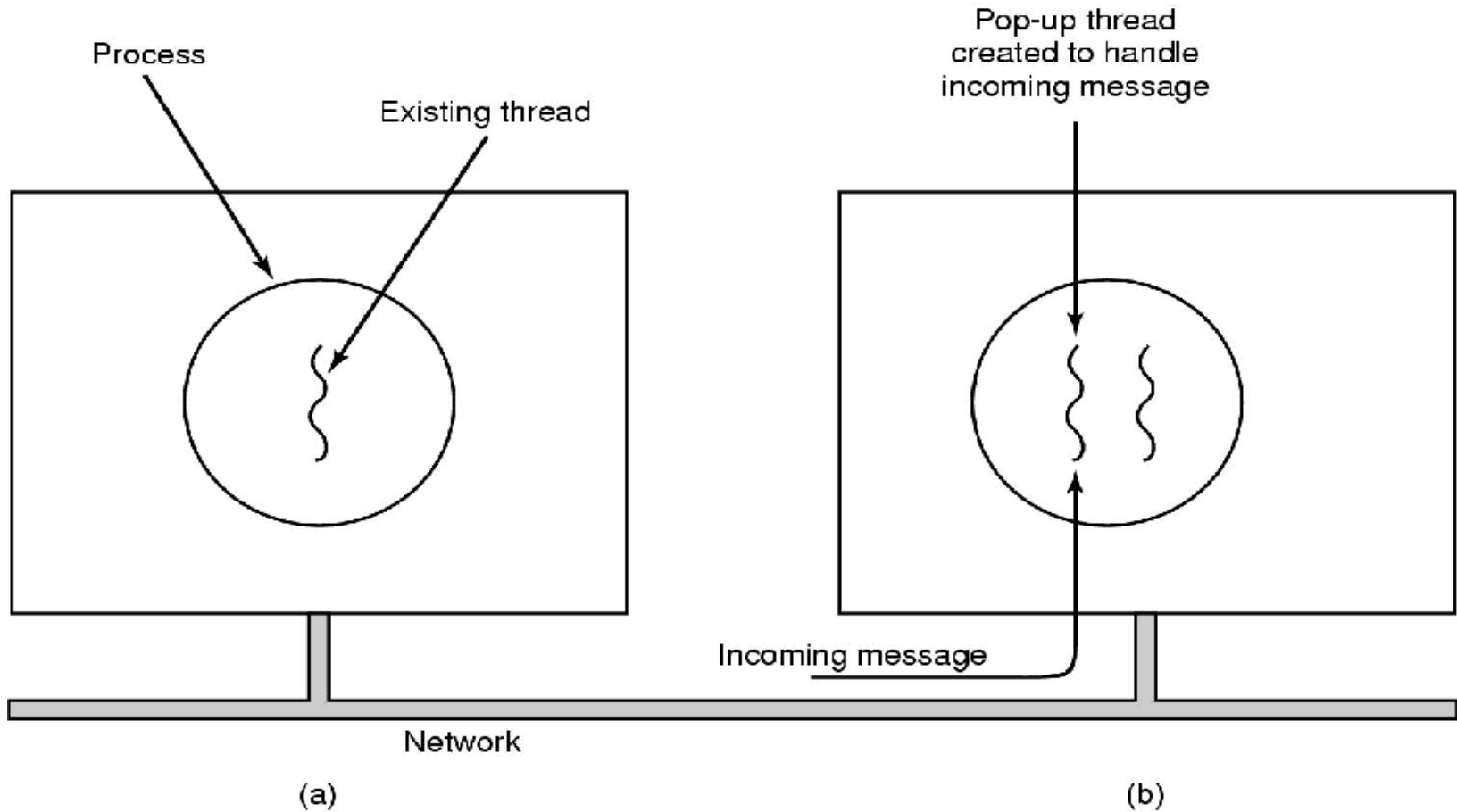
Υλοποίηση νημάτων στο (α) χώρο του χρήστη (β) πυρήνα



Υβριδική Υλοποίηση



Νήματα pop-up



Νήματα χρήστη και πυρήνα: σύγκριση

Νήματα πυρήνα:

- Είναι γνωστά στο Λ.Σ.
- Η εναλλαγή μεταξύ νημάτων πυρήνα στην ίδια διεργασία δεν είναι δαπανηρή.
- Οι τιμές των καταχωρητών, του PC, και των stack pointers μεταβάλλονται.
- Οι πληροφορίες που αφορούν τη διαχείριση μνήμης δεν αλλάζουν.
- Ο πυρήνας χρησιμοποιεί αλγορίθμους δρομολόγησης διεργασιών για τη διαχείριση των νημάτων.

Νήματα χρήστη:

- Το Λ.Σ. δεν γνωρίζει για τα νήματα χρήστη, είναι ενημερωμένο μόνο για τη διεργασία όπου περιέχονται.
- Το Λ.Σ. δρομολογεί τις διεργασίες και όχι τα νήματα.
- Ο προγραμματιστής χρησιμοποιεί τη βιβλιοθήκη νημάτων για να τα διαχειριστεί (δημιουργία, διαγραφή, συγχρονισμός και δρομολόγηση).



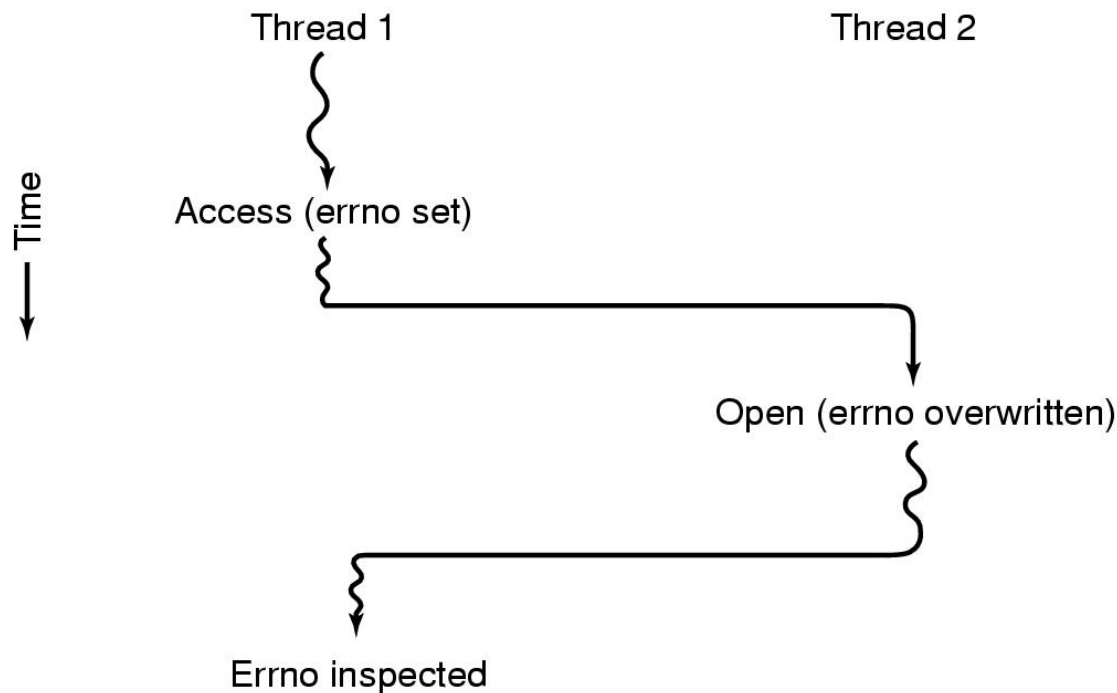
Ομάδες νημάτων

- Δημιουργία ενός αριθμού νημάτων σε μια πισίνα (pool) που αναμένουν να εκτελεστούν.
 - Πλεονεκτήματα:
 - Συνήθως ελαφρώς γρηγορότερη εξυπηρέτηση αίτησης από ένα υπάρχων νήμα παρά με την δημιουργία νέου.
 - Επιτρέπει ο αριθμός των νημάτων κατά την εφαρμογή(ές) να δεσμεύεται με το μέγεθος της πισίνας.



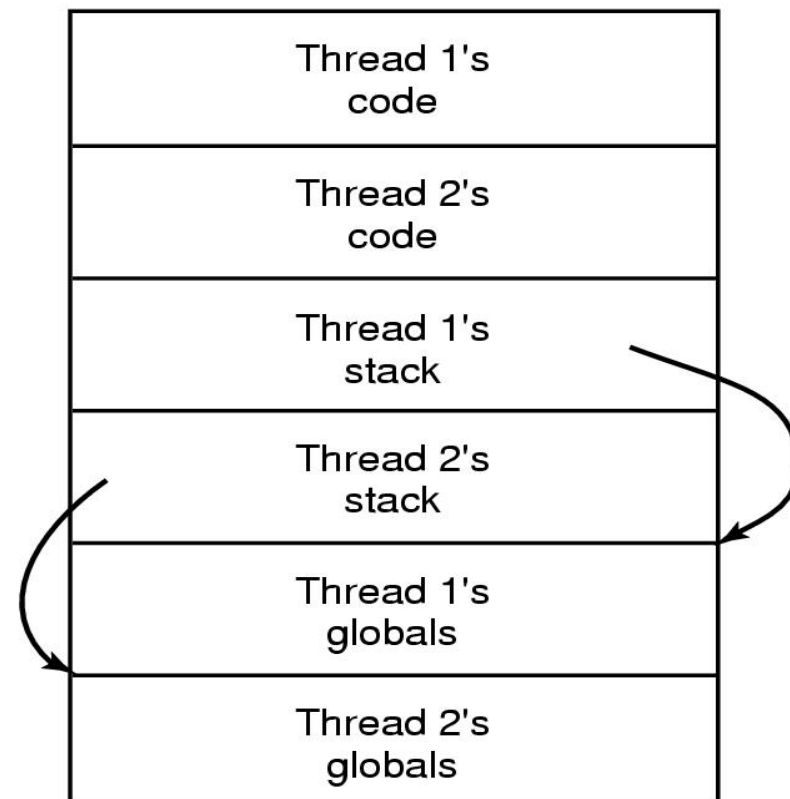
Προβλήματα με τη δημιουργία πολυνηματικών διεργασιών

- Συγκρούσεις μεταξύ νημάτων που χρησιμοποιούν την ίδια καθολική μεταβλητή.



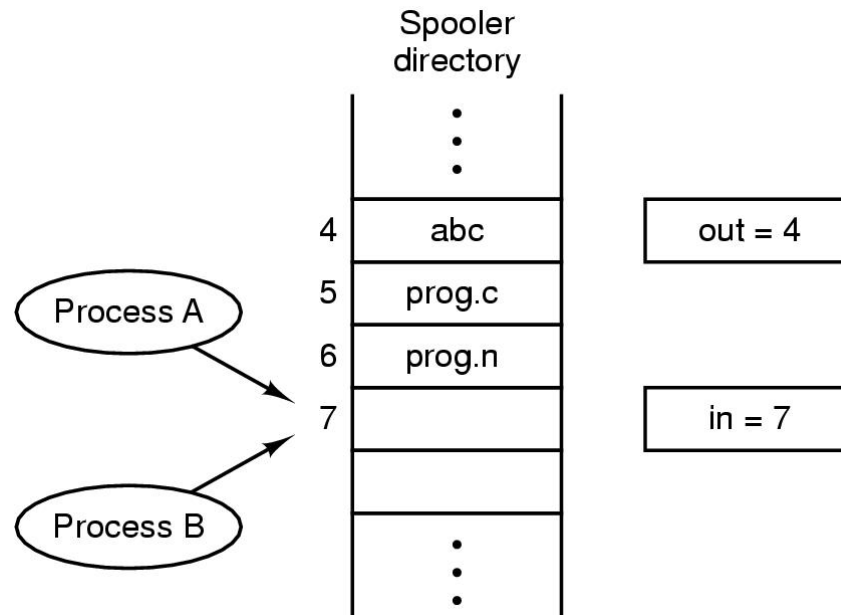
Υπάρχει ενδεχομένως πρόβλημα στις μεταβλητές των νημάτων

- Τα νήματα μπορούν να έχουν ιδιωτικές μεταβλητές.



Προβλήματα για πρόσβαση σε κοινή μνήμη

- Δύο διεργασίες επιθυμούν να προσπελάσουν κοινή μνήμη ταυτόχρονα!



Παραδείγματα νημάτων

- **Posix Pthreads:** (IEEE) standard:
 - Έχουν καθορισμένη διεπαφή.
 - Η υλοποίηση (χρησιμοποιώντας user/kernel level threads) εξαρτάται από όσους τα αναπτύσσουν.
 - Είναι πιο συνηθισμένα σε συστήματα UNIX.
- **Java threads:**
 - Υποστηρίζονται από την JVM.
 - Η υλοποίηση εξαρτάται από όσους τα αναπτύσσουν.



Hyper-threading

- Η Intel υλοποιεί την τεχνολογία **hyper-threading** που μοιάζει με την πολυνημάτωση, υλοποιείται σε συστήματα με έναν πυρήνα και τα κάνει να εμφανίζονται σαν να διαθέτουν πολλαπλούς επεξεργαστές. Εκείνο το οποίο επιτυγχάνεται είναι η αύξηση του ρυθμού με τον οποίο το σύστημα μπορεί να εναλλάσσεται μεταξύ πολλών νημάτων. Έτσι ενισχύεται ο πολυδιεργασιακός χαρακτήρας των προσωπικών υπολογιστών.



Πλεονεκτήματα των νημάτων ως προς τις διεργασίες

- Η δημιουργία μιας νέας διεργασίας μπορεί να είναι περισσότερο δαπανηρή.
- Απαιτεί χρόνο: χρειάζεται κλήση στον πυρήνα του Λ.Σ.
- Μπορεί να προκαλέσει επαναδρομολόγηση οπότε θα προκύψει εναλλαγή πλαισίου με αντικατάσταση ολόκληρης της διεργασίας
- Η επικοινωνία και ο συγχρονισμός κοστίζουν διότι απαιτείται κλήση στον πυρήνα.
- Τα νήματα μπορούν να δημιουργηθούν χωρίς να αντικατασταθεί ολόκληρη η διεργασία.
- Το περισσότερο έργο για τη δημιουργία του νήματος γίνεται στο χώρο διευθύνσεων του χρήστη παρά στον πυρήνα του Λ.Σ.
- Τα νήματα συγχρονίζονται με την παρακολούθηση μιας μεταβλητής, σε αντίθεση με τις διεργασίες που απαιτούν κλήση πυρήνα.



Νήματα & Διεργασίες: σύγκριση

Νήματα

- Ένα νήμα δεν διαθέτει data segment ή heap.
- Ένα νήμα δεν μπορεί να υπάρξει μόνο του, πρέπει να υπάρχει στα πλαίσια μιας διεργασίας.
- Υπάρχουν περισσότερα από ένα νήματα σε μια διεργασία. Το πρώτο νήμα είναι το κύριο και κατέχει τη στοίβα της διεργασίας.
- Η δημιουργία του καθώς και η εναλλαγή πλαισίου δεν είναι δαπανηρή.
- Αν ένα νήμα εκλείψει η στοίβα του επιστρέφεται στους πόρους του συστήματος.

Διεργασίες

- Μια διεργασία έχει code/data/heap καθώς και άλλα segments.
- Υπάρχει ένα τουλάχιστον νήμα σε μια διεργασία.
- Τα νήματα σε μια διεργασία διαμοιράζονται code/data/heap και I/O, αλλά το καθένα έχει τη δική του στοίβα και τους δικούς του καταχωρητές.
- Η δημιουργία τους καθώς και οι εναλλαγές πλαισίου είναι δαπανηρές.
- Αν μια διεργασία εκλείψει οι πόροι της επιστρέφονται στο σύστημα και όλα της τα νήματα εκλείπουν.



Νήματα και ΛΣ

Τα ΛΣ χαρακτηρίζονται ως προς τη σχέση που έχουν τα νήματα με τις ανεξάρτητες οντότητες χρονοδρομολόγησης ως εξής:

- 1:1 (νήματα πυρήνα) Κάθε νήμα αντιστοιχεί σε μια οντότητα.
- N:1 (νήματα χρήστη) Πολλαπλά νήματα που ανήκουν σε μια διεργασία αντιστοιχούν μόνο σε μια οντότητα (τη συγκεκριμένη διεργασία).
- N:M (υβριδική υλοποίηση) Συνδυασμός των παραπάνω.



Πιο προχωρημένα θέματα (1/2)

Εκτός από τα νήματα υπάρχουν:

- **Ίνες (fibers)**

πιο ελαφριά οντότητα από νήματα. Η κάθε ίνα παραχωρεί τον επεξεργαστή σε άλλη ίνα. Δεν γίνεται χρονοδρομολόγηση άλλης ίνας, αν δεν γίνει οικειοθελής παράδοση από την αρχική ίνα.

- **Πράσινα νήματα (green threads)**

νήματα που δρομολογούνται όχι από το ΛΣ αλλά από μια VM (όπως python ή java) {user level}.



Πιο προχωρημένα θέματα (2/2)

- **Παραδείγματα υλοποίησης σε επίπεδο πυρήνα:**
 - Light Weight Kernel Threads (LWKT) σε διάφορα BSDs.
 - M:N νήματα.
 - Native POSIX Thread Library (NPTL) για Linux, (μια υλοποίηση του προτύπου Posix Threads).
 - Apple Multiprocessing Services έκδοση 2.0.
 - Microsoft Windows από Windows 95 και νεότερα.
- **Παραδείγματα υλοποίησης σε επίπεδο χρήστη:**
 - GNU Portable Threads.
 - FSU Pthreads.
 - Apple inc's Thread Manager.
 - REALbasic.
 - Netscape Portable Runtime.
 - State threads.
- **Παραδείγματα υβριδικής υλοποίησης:**
 - Scheduler activations.
 - Marcel from the PM2 project.
 - The OS from the Tera/Cray MTA.
 - Microsoft Windows 7.
- **Παραδείγματα υλοποίησης ίνας:**
 - Οι ίνες μπορούν να υλοποιηθούν χωρίς την υποστήριξη του ΛΣ αν και κάποια .
 - ΛΣ ή βιβλιοθήκες παρέχουν σαφή υποστήριξη.
 - Win32 παρέχουν ένα API ίνας.
 - Ruby as Green Threads.



Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

