



Διεργασίες

Θεωρητικό Υπόβαθρο

- Διεργασία: ένα πρόγραμμα σε εκτέλεση
- Σημαντικά θέματα
 - διαχείριση
 - χρονοπρογραμματισμός
- Στα Κατανεμημένα Συστήματα είναι βολικό να χρησιμοποιείται πολυνημάτωση (multithreading)
- Σημαντικά Ζητήματα:
 - Οργάνωση πελάτη – εξυπηρετητή
 - Μετανάστευση διεργασιών

Εισαγωγή στα Νήματα (1)

- Η κατασκευή κατανεμημένων εφαρμογών γίνεται ευκολότερη, καθώς και η επίτευξη καλύτερης απόδοσης, με τη χρήση multithreading
- Η διαφάνεια ταυτοχρονισμού των διεργασιών έχει υψηλό τίμημα (εκχώρηση μνήμης, εναλλαγή περιβάλλοντος λειτουργίας, κλπ)
- Στα νήματα δε γίνεται καμία προσπάθεια να επιτευχθεί υψηλός βαθμός διαφάνειας ταυτοχρονισμού εις βάρος της απόδοσης.
- Ένα περιβάλλον νήματος αποτελείται μόνο από το περιβάλλον CPU και μερικές πληροφορίες διαχείρισης

Εισαγωγή στα Νήματα (2)

- Παράδειγμα: ένα σύστημα νημάτων παρακολουθεί το γεγονός ότι ένα νήμα είναι μπλοκαρισμένο σε ένα σηματοφόρο και δεν το επιλέγει για εκτέλεση.
- Οι κατασκευαστές εφαρμογών είναι αρμόδιοι για την προστασία των δεδομένων από νήματα της ίδιας διεργασίας
- Η απόδοση των πολυνηματικών εφαρμογών είναι καλύτερη.

Μονονηματικές Διεργασίες

- Όταν εκτελείται μια κλήση συστήματος που οδηγεί σε μπλοκάρισμα, μπλοκάρεται ολόκληρη η διεργασία (π.χ. ενημέρωση κελιών στο Microsoft Excel, λόγω τροποποίησης άλλου κελιού)
- Δεν υπάρχει παραλληλισμός !! Απαιτείται πολυνημάτωση προκειμένου σε ένα πολυεπεξεργαστικό σύστημα να ανατεθεί από ένα νήμα σε διαφορετική CPU.
- Η επικοινωνία στα πολυεπεξεργαστικά συστήματα γίνεται μέσω διαδικεργασιακής επικοινωνίας (π.χ. pipes, shared memory)
 - Έχουμε καθυστερήσεις λόγω context switch

Πολυνηματικές Διεργασίες

- Αντί οι εφαρμογές να χτίζονται με πολλές διεργασίες, μπορούν να χρησιμοποιηθούν πολυνηματικές διεργασίες
- Η επικοινωνία γίνεται αποκλειστικά μέσω κοινόχρηστων δεδομένων
- Η εναλλαγή νημάτων μπορεί κάποιες φορές να γίνει εξ ολοκλήρου στο χώρο μνήμης χρήστη, πράγμα που οδηγεί σε βελτίωση της απόδοσης
- Για εφαρμογές που αποτελούνται από πολλές ανεξάρτητες μεταξύ τους λειτουργίες, είναι ευκολότερο να δομηθούν με πολυνημάτωση.

Υλοποίηση Νημάτων (1)

—Κατασκευή βιβλιοθήκης νημάτων που εκτελείται σε επίπεδο χρήστη

- Η δημιουργία και η κατάργηση νημάτων έχει μικρό κόστος, αφού έχει να κάνει με την εκχώρηση μνήμης χρήστη για την εγκατάσταση ή την απελευθέρωση στοίβας νημάτων.
- Η εναλλαγή γίνεται με λίγες μόνο εντολές. Μόνο οι καταχωρητές της CPU αλλάζουν τιμές και δεν υπάρχουν αλλαγές τιμών σε μνήμη, TLB, κλπ.
- Μια κλήση συστήματος όμως που οδηγεί σε μπλοκάρισμα, μπλοκάρει όλα τα νήματα

Υλοποίηση Νημάτων (2)

—Υλοποίηση νημάτων στον πυρήνα

- Κάθε λειτουργία νήματος εκτελείται μέσω κλήσης συστήματος στον πυρήνα (μεγάλο κόστος)
- Η εναλλαγή ισοδυναμεί με την εναλλαγή διεργασιών

—Υβριδική λύση: ελαφρές διεργασίες

- Μια διεργασία μπορεί να περιέχει πολλές «ελαφρές διεργασίες» τις οποίες διαχειρίζεται ο πυρήνας
- Το σύστημα παρέχει ένα πακέτο λειτουργιών νημάτων που εκτελείται σε επίπεδο χρήστη.
- Το πακέτο αυτό μπορεί να μοιραστεί σε πολλές ελαφρές διεργασίες

Υλοποίηση Ελαφρών Διεργασιών (1)

- Κάθε ελαφρά διεργασία εκτελεί το δικό της νήμα σε επίπεδο χρήστη
- Οι πολυνηματικές εφαρμογές κατασκευάζονται με τη δημιουργία νημάτων και την ανάθεση του κάθε νήματος σε μια ελαφρά διεργασία, με διάφανο για το χρήστη τρόπο
- Το πακέτο νημάτων διαθέτει μία μόνο ρουτίνα για το χρονοπρογραμματισμό του επόμενου νήματος
- Όταν μία κλήση συστήματος δημιουργεί μια ελαφρά διεργασία, τότε της παρέχεται μια στοίβα και της δίνεται εντολή εκτέλεσης της ρουτίνας χρονοπρογραμματισμού

Υλοποίηση Ελαφρών Διεργασιών (2)

- Κάθε ξεχωριστή ελαφρά διεργασία εκτελεί τη ρουτίνα χρονοπρογραμματισμού
- Υπάρχει πίνακας νημάτων που περιέχει το σύνολο των νημάτων και είναι κοινός για τις ελαφρές διεργασίες
- Η προστασία του πίνακα νημάτων γίνεται με μεταβλητές αμοιβαίου αποκλεισμού σε επίπεδο χρήστη.
- Όταν μια ελαφρά διεργασία βρει ένα εκτελέσιμο νήμα γίνεται εναλλαγή σε αυτό.

Υλοποίηση Ελαφρών Διεργασιών (3)

- Όταν ένα νήμα μπλοκαριστεί λόγω αμοιβαίου αποκλεισμού, κάνει τις απαιτούμενες διαχειριστικές εργασίες και καλεί τη ρουτίνα χρονοπρογραμματισμού
- Όταν βρεθεί άλλο νήμα γίνεται εναλλαγή σε αυτό, η οποία υλοποιείται σε επίπεδο χρήστη και δεν χρειάζεται η ελαφρά διεργασία να είναι ενήμερη.
- Όταν ένα νήμα μπλοκαριστεί λόγω κλήσης συστήματος, η εκτέλεση περνά σε κατάσταση πυρήνα. Αν η ελαφρά διεργασία δεν μπορεί να συνεχίσει, γίνεται εναλλαγή σε άλλη ελαφρά διεργασία σε κατάσταση χρήστη.

Πολυνηματικοί Πελάτες (1)

- Πρόβλημα: Οι μεγάλοι χρόνοι καθυστέρησης διαδικεργασιακής επικοινωνίας πάνω από WANs.
- Παράδειγμα: η προσκόμιση ενός HTML αρχείου από ένα web browser που περιέχει απλό κείμενο, εικόνες, φωτογραφίες, κλπ.
- Για την προσκόμιση ενός στοιχείου του εγγράφου ο web browser πρέπει να:
 - Εγκαταστήσει μια TCP/IP σύνδεση
 - Διαβάσει τα εισερχόμενα δεδομένα
 - Να τα μεταβιβάσει για παρουσίαση

Πολυνηματικοί Πελάτες (2)

- Οι λειτουργίες αυτές απαιτούν μπλοκάρισμα
- Για να λυθεί το πρόβλημα μερικοί browsers αρχίζουν να εμφανίζουν τα δεδομένα ενώ αυτά λαμβάνονται ακόμη.
- Τα στοιχεία εμφανίζονται με τη σειρά που λαμβάνονται.
- Η υλοποίηση γίνεται με ξεχωριστά νήματα που αναλαμβάνουν την προσκόμιση των διαφορετικών στοιχείων, μόλις ληφθεί το κύριο HTML αρχείο.
- Κάθε νήμα εγκαθιστά μια ξεχωριστή σύνδεση με το server και λαμβάνει τα δεδομένα.
- Τα νήματα του browser μπορούν να εγκαθιστούν συνδέσεις με ξεχωριστούς web servers (mirrors) προκειμένου να αποφευχθεί η υπερφόρτωση

Πολυνηματικοί Εξυπηρετητές (1)

- Σήμερα οι πολυεπεξεργαστικοί υπολογιστές έχουν αποκτήσει ευρεία διάδοση ως σταθμοί εργασίας γενικής χρήσης
- Η πολυνημάτωση είναι για το λόγο αυτό ακόμη πιο χρήσιμη
- Ένα νήμα διανομέα (dispatcher) διαβάζει τις εισερχόμενες αιτήσεις πελατών για λειτουργίες αρχείων
- Ο εξυπηρετητής διαβάζει την αίτηση και επιλέγει ένα αδρανές νήμα – εργάτη και του παραδίδει την αίτηση.

Πολυνηματικοί Εξυπηρετητές (2)

- Το νήμα – εργάτης εκτελεί ανάγνωση του αρχείου και μπλοκάρεται.
- Στη συνέχεια επιλέγεται άλλο νήμα (π.χ. νήμα – εργάτης) προς εκτέλεση.
- Αν ο εξυπηρετητής είναι μονονηματικός, τότε κάθε αίτηση πελάτη εκτελείται μέχρι τέλους, χωρίς παραλληλισμό
 - Οι λειτουργίες δίσκου μπλοκάρουν το νήμα και ο εξυπηρετητής μένει αδρανής μέχρι την ολοκλήρωση της λειτουργίας.
- Αν δεν υπάρχουν νήματα, τότε ο εξυπηρετητής μπορεί να υλοποιηθεί σαν μηχανή πεπερασμένων καταστάσεων
 - Αν φτάσει μια αίτηση, τότε αν μπορεί να εξυπηρετηθεί από την κρυφή μνήμη όλα είναι εντάξει, αλλιώς στέλνεται μήνυμα στο δίσκο

Πολυνηματικοί Εξυπηρετητές (3)

- Στο μεταξύ ο εξυπηρετητής δεν μπλοκάρεται, καταγράφει την κατάσταση της τρέχουσας αίτησης σε ένα πίνακα και πηγαίνει να εξυπηρετήσει το επόμενο μήνυμα.
- Το μήνυμα αυτό μπορεί να είναι νέα αίτηση ή μήνυμα από το δίσκο
 - Αν είναι μήνυμα νέας αίτησης, αυτή ξεκινά να εξυπηρετείται
 - Αλλιώς προσκομίζεται η κατάσταση της αίτησης αυτής από τον πίνακα και διεκπεραιώνεται
 - Ο εξυπηρετητής χρησιμοποιεί non blocking send και receive κλήσεις

Γενικά Ζητήματα Σχεδίασης Εξυπηρετητών (1)

- Οργάνωση εξυπηρετητή
 - Επαναληπτικός εξυπηρετητής
 - Ταυτόχρονος εξυπηρετητής
- Ύπαρξη ενός δαίμονα για παρακολούθηση των θυρών όπου ακούνε οι υπηρεσίες ενός εξυπηρετητή (π.χ. inetd στο UNIX). Οι πελάτες απευθύνονται πρώτα σε αυτούς
- Διακοπή Εξυπηρετητή
 - Ο πελάτης διακόπτει απότομα την επικοινωνία (ο εξυπηρετητής νομίζει ότι κατέρρευσε και κλείνει τη σύνδεση) ή
 - Ο πελάτης στέλνει δεδομένα out-of-bound σε άλλη θύρα, όπου ο εξυπηρετητής την παρακολουθεί με μεγαλύτερη προτεραιότητα

Γενικά Ζητήματα Σχεδίασης Εξυπηρετητών (2)

— Διατήρηση κατάστασης πελατών

- Stateless (π.χ. Web server)
- Stateful : διατηρεί ένα πίνακα πληροφοριών
 - Υπάρχει πρόβλημα σε περίπτωση κατάρρευσης του εξυπηρετητή (πρέπει να ανακτηθούν οι πληροφορίες του πίνακα ώστε να μην χαθούν ενημερώσεις αρχείων)
- Οι web servers μιμούνται τη λειτουργία των stateful εξυπηρετητών με τη χρήση αρχείων πληροφοριών (π.χ. ιστορικό πελάτη) που αποθηκεύονται στη μεριά του πελάτη (cookies)
 - Μετά την πρώτη επικοινωνία πελάτη – εξυπηρετητή, τα cookies στέλνονται μαζί με τις αιτήσεις
 - Υπάρχουν προβλήματα ασφαλείας

Εξυπηρετητές Αντικειμένων

- Είναι εξυπηρετητές για την υποστήριξη κατανεμημένων αντικειμένων
- Δεν παρέχει ο ίδιος κάποια υπηρεσία
- Παρέχει μόνο το μέσο για την κλήση των τοπικών αντικειμένων που παρέχουν τις υπηρεσίες
- Για το λόγο αυτό είναι εύκολη η προσθήκη και η κατάργηση αντικειμένων

Εναλλακτικοί Τρόποι Κλήσης Αντικειμένων (1)

- Για να κληθεί ένα αντικείμενο ο εξυπηρετητής πρέπει να γνωρίζει ποιος κώδικας θα εκτελεστεί, σε ποια δεδομένα, αν θα δημιουργηθεί νέο νήμα, κλπ.
- 1) Θεώρηση κλήσης στο DCE: όλα τα αντικείμενα είναι όμοια και υπάρχει μόνο ένας τρόπος κλήσης.
 - Όχι ευέλικτη λύση
- 2) Χρήση προσωρινών αντικειμένων (transient objects): ένα αντικείμενο δημιουργείται στη μνήμη με την πρώτη κλήση πελάτη και καταργείται όταν δεν υπάρχει πελάτης συνδεδεμένος με αυτό
- 3) Κάθε αντικείμενο τοποθετείται σε δικό του τμήμα μνήμης (όταν τα δεδομένα και ο κώδικας του αντικειμένου δεν ξεχωρίζουν)
 - Εναλλακτικά τα αντικείμενα μπορούν να διαμοιράζονται τον κώδικά τους

Εναλλακτικοί Τρόποι Κλήσης Αντικειμένων (2)

- Σε σχέση με τη χρήση νημάτων:
- 1) Υλοποίηση ενός εξυπηρετητή με ένα μόνο νήμα
- 2) Ο εξυπηρετητής έχει ένα νήμα για κάθε αντικείμενό του
 - Αν το νήμα για το συγκεκριμένο αντικείμενο είναι απασχολημένο τοποθετεί το αίτημα σε ουρά
- 3) Ένα νήμα για κάθε διαφορετική αίτηση
 - Απαιτείται προσοχή στην προσπέλαση κοινών πόρων

Προσαρμογέας Αντικειμένων (Object Adaptor)

- Έχει υπό τον έλεγχό του ένα ή περισσότερα αντικείμενα
- Μια αίτηση πελάτη που παραλαμβάνεται από τον εξυπηρετητή παραδίδεται στον προσαρμογέα
- Ο προσαρμογέας είναι υπεύθυνος για την ενεργοποίηση του αντικειμένου χρησιμοποιώντας μια αναφορά αντικειμένου
- Οι προσαρμογείς δεν είναι ενήμεροι για τα interfaces των αντικειμένων τους
- Γι' αυτό ο προσαρμογέας παραδίδει την αίτηση στο σκελετό αντικειμένου που αποκωδικοποιεί τις παραμέτρους της αίτησης και καλεί την κατάλληλη μέθοδο

Μετανάστευση Κώδικα

- Μετανάστευση κώδικα σημαίνει μετακίνηση ολόκληρων διεργασιών
 - Εργασία με μεγάλο κόστος
- Εκτελείται κυρίως για:
 - Εξισορρόπηση του φόρτου σε ένα καταναμημένο σύστημα ή
 - Απαιτείται μετανάστευση μεγάλου όγκου δεδομένων (π.χ. συμπλήρωση μιας φόρμας στη μεριά του πελάτη και όχι πάνω στο server) ή
 - Γίνεται αναζήτηση στον παγκόσμιο ιστό ή
 - Για λόγους ευελιξίας/δυναμικής διευθέτησης των συστημάτων

Μοντέλα Μετανάστευσης Κώδικα (1)

- Μια διεργασία αποτελείται από τρία τμήματα: κώδικας, πόροι, τμήμα εκτέλεσης
- Μοντέλο περιορισμένης μετακίνησης: μετακινείται μόνο κώδικας και ίσως κάποια λίγα δεδομένα (π.χ. Applets – η εφαρμογή ξεκινά από την αρχική της κατάσταση)
- Μοντέλο εκτενούς μετακίνησης: μεταφέρεται και το τμήμα εκτέλεσης (μπορεί να υποστηριχθεί και με κλωνοποίηση)
- Ένα μοντέλο μπορεί να είναι:
 - Sender-initiated (π.χ. μια λειτουργία upload)
 - Receiver-initiated (π.χ. Applet)

Μοντέλα Μετανάστευσης Κώδικα (2)

- Κατά το ανέβασμα κώδικα σε εξυπηρετητή με sender – initiated τεχνική είναι απαραίτητη η προστασία – απαιτείται πιστοποίηση του πελάτη
- Αντίθετα, το κατέβασμα κώδικα μπορεί να γίνει και ανώνυμα
- Ο εξυπηρετητής δεν ενδιαφέρεται για τους πόρους του πελάτη
- Κατά την περιορισμένη μετακίνηση έχει σημασία αν ο κώδικας εκτελείται στο χώρο μνήμης της διεργασίας προορισμού (π.χ. Applet) ή σε χώρο ξεχωριστής διεργασίας

Μοντέλα Μετανάστευσης Κώδικα (3)

- Η εκτεταμένη μετακίνηση μπορεί να γίνει και με απομακρυσμένη κλωνοποίηση
- Παράγεται ένα ακριβές αντίγραφο της αρχικής διεργασίας που εκτελείται σε άλλη μηχανή, παράλληλα με την αρχική διεργασία
- Παρέχει ένα μοντέλο παρόμοιο με αυτό που χρησιμοποιείται σε άλλες μηχανές

Μετανάστευση και Τοπικοί Πόροι (1)

- Σε αντίθεση με το τμήμα κώδικα και το τμήμα εκτέλεσης το τμήμα πόρων μιας διεργασίας απαιτεί ειδική αντιμετώπιση.
- Το τμήμα πόρων δεν μπορεί πάντα να μετακινηθεί χωρίς να τροποποιηθεί
 - Π.χ. μια διεργασία που ακούει σε μια συγκεκριμένη θύρα, όταν μετακινηθεί πρέπει να ζητήσει μια νέα θύρα στον προορισμό.
- Τύποι συνδυασμού διεργασιών με πόρους
 - Συνδυασμός με αναγνωριστικό
 - Συνδυασμός με τιμή
 - Συνδυασμός με τύπο

Μετανάστευση και Τοπικοί Πόροι (2)

- Συνδυασμοί πόρων με μηχανήματα
 - Μη προσαρτημένοι πόροι
 - Προσδεδεμένοι πόροι
 - Σταθεροί πόροι
- Όταν έχουμε συνδυασμό με μη προσαρτημένο πόρο μέσω αναγνωριστικού, τότε ο πόρος μπορεί να μετακινηθεί.
- Όταν όμως πολλές διεργασίες μοιράζονται τον πόρο τότε απαιτείται μια καθολική αναφορά (π.χ. URL). Αυτό όμως δεν είναι πάντα σωστό.

Μετανάστευση και Τοπικοί Πόροι (3)

- Συνδυασμοί πόρων με τιμή
 - Όταν ο πόρος είναι σταθερός (π.χ. μια διεργασία θεωρεί ότι η μνήμη πρέπει να μοιραστεί μεταξύ διαφορετικών διεργασιών) πρέπει να υλοποιηθούν μηχανισμοί κατανεμημένης κοινής μνήμης (μη βιώσιμη λύση)
 - Οι προσδεδεδεμένοι πόροι είναι συνήθως βιβλιοθήκες που φορτώνονται κατά το χρόνο εκτέλεσης. Η καθολική αναφορά είναι η καλύτερη λύση όταν πρόκειται να γίνει αντιγραφή τεράστιας ποσότητας δεδομένων
 - Οι μη προσαρτημένοι πόροι είτε μετακινούνται είτε δημιουργείται καθολική αναφορά όταν διαμοιράζονται
- Συνδυασμοί πόρων με τύπο
 - Επανασυνδυασμός της διεργασίας με ένα πόρο ίδιου τύπου

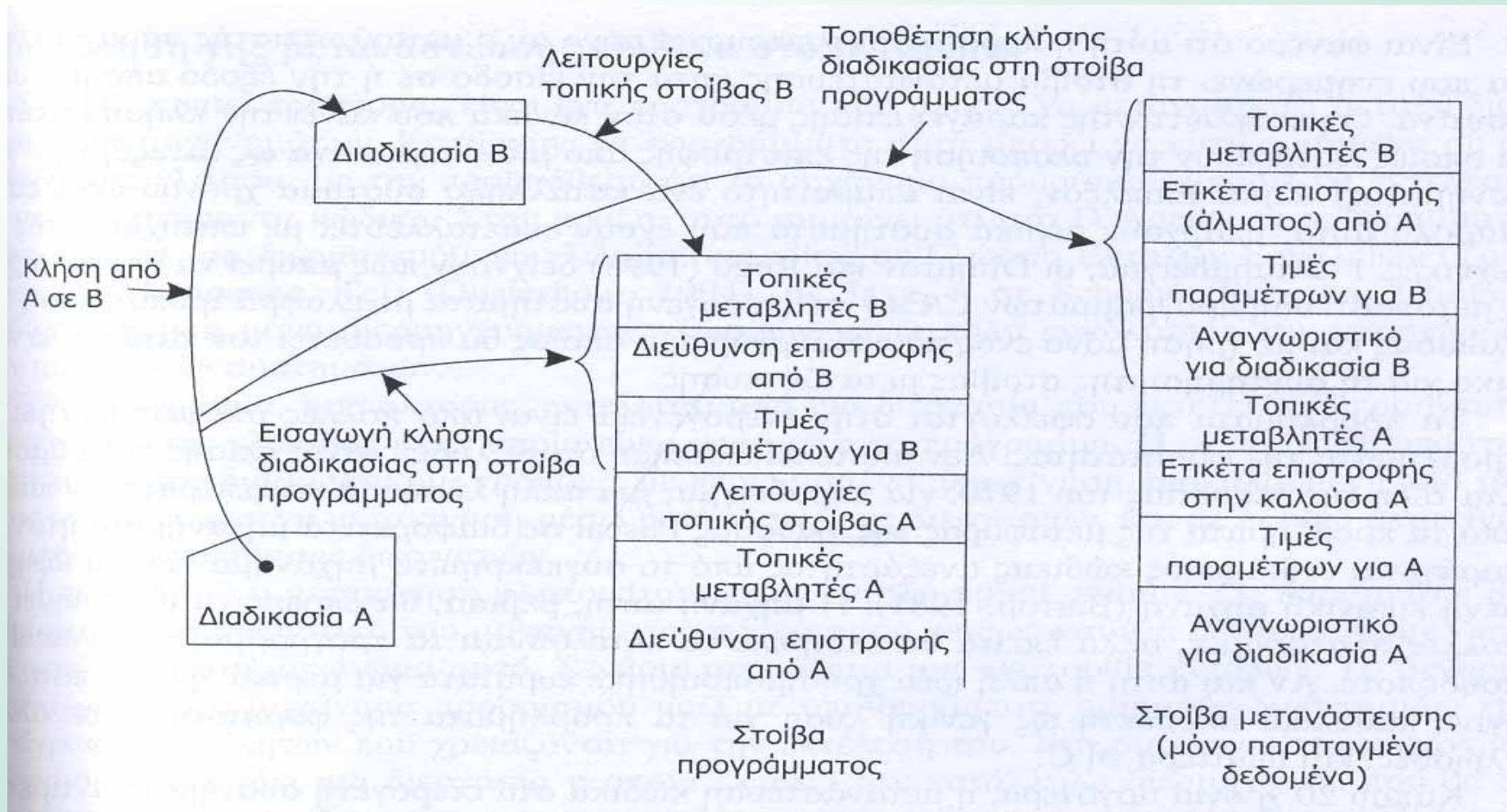
Μετανάστευση στα Ετερογενή Συστήματα (1)

- Στην περιορισμένη μετακίνηση δεν έχουμε μεγάλα προβλήματα
- Στην εκτεταμένη μετακίνηση το σημαντικότερο πρόβλημα είναι η μετακίνηση του τμήματος εκτέλεσης
 - Εξαρτάται από το περιβάλλον στο οποίο εκτελείται η διεργασία
 - Περιέχει ιδιωτικά δεδομένα της διεργασίας, το μετρητή προγράμματος και τη στοίβα
 - Η στοίβα εκτός από δεδομένα τοπικών μεταβλητών μπορεί να περιέχει και τιμές καταχωρητών που εξαρτώνται από την πλατφόρμα εκτέλεσης.
 - Γλώσσες όπως η C και η java περιορίζουν τη μετακίνηση μόνο σε συγκεκριμένα σημεία ενός προγράμματος (π.χ. κλήση υπορουτίνας)

Μετανάστευση στα Ετερογενή Συστήματα (2)

- Το σύστημα χρόνου εκτέλεσης διατηρεί ένα αντίγραφο της στοίβας προγράμματος, με τρόπο ανεξάρτητο πλατφόρμας, που καλείται στοίβα μετανάστευσης
 - Ενημερώνεται όταν κατά την κλήση μιας υπορουτίνας ή κατά την επιστροφή από μια υπορουτίνα
- Χρησιμοποιείται ένα αναγνωριστικό για την καλούμενη υπορουτίνα και μια ετικέτα άλματος για τη διεύθυνση επιστροφής κατά το πέρας της υπορουτίνας

Μετανάστευση στα Ετερογενή Συστήματα (3)



Μετανάστευση στα Ετερογενή Συστήματα (4)

- Κατά την εγγραφή στη στοίβα μετακίνησης αγνοούνται τα δεδομένα που αφορούν το συγκεκριμένο μηχάνημα, καθώς και η τρέχουσα στοίβα.
- Τα παραταγμένα δεδομένα και η στοίβα μετανάστευσης μεταφέρονται στον προορισμό
- Στον προορισμό φορτώνεται το κατάλληλο τμήμα κώδικα που περιέχει τα κατάλληλα δυαδικά δεδομένα για την αρχιτεκτονική του μηχανήματος και για το λειτουργικό σύστημα
- Τα δεδομένα αποπαρατάσσονται και δημιουργείται μια νέα στοίβα χρόνου εκτέλεσης

Μετανάστευση στα Ετερογενή Συστήματα (5)

- Σήμερα το πρόβλημα λύνεται με γλώσσες σεναρίων και γλώσσες υψηλής φορητότητας (π.χ. Java)
- Βασίζονται σε εικονικές μηχανές
- Στις γλώσσες σεναρίων η εικονική μηχανή διερμηνεύει άμεσα τον πηγαίο κώδικα
- Στις γλώσσες υψηλής φορητότητας διερμηνεύει ενδιάμεσο κώδικα που παράγεται από κάποιο μεταγλωττιστή
- Η μέθοδος της εικονικής μηχανής μας περιορίζει σε μια συγκεκριμένη γλώσσα, η οποία συχνά δεν έχει χρησιμοποιηθεί στο παρελθόν. Γι' αυτό πρέπει να παρέχουν διασυνδέσεις για υπάρχουσες γλώσσες.

Πράκτορες Λογισμικού

- Πράκτορας λογισμικού είναι μια αυτόνομη διεργασία ικανή να αντιδρά σε αλλαγές και να προκαλεί αλλαγές στο περιβάλλον της, ενδεχομένως σε συνεργασία με χρήστες και άλλους πράκτορες (Jennings & Woolridge, 1998)
- Ο πράκτορας ενεργεί αυτοβούλως, παίρνει πρωτοβουλίες.
- Συνεργατικός πράκτορας: έχει δυνατότητα συνεργασίας με άλλους πράκτορες στα πλαίσια ενός πολυπρακτορικού συστήματος
- Μετακινούμενοι πράκτορες: μετακινούνται μεταξύ διαφορετικών μηχανημάτων
- Πράκτορες διασύνδεσης: βοηθούν τον τελικό χρήστη στη χρήση μιας ή περισσότερων εφαρμογών – έχουν ικανότητες μάθησης
- Πράκτορες πληροφοριών: διαχειρίζονται πληροφορίες από πολλές διαφορετικές πηγές.