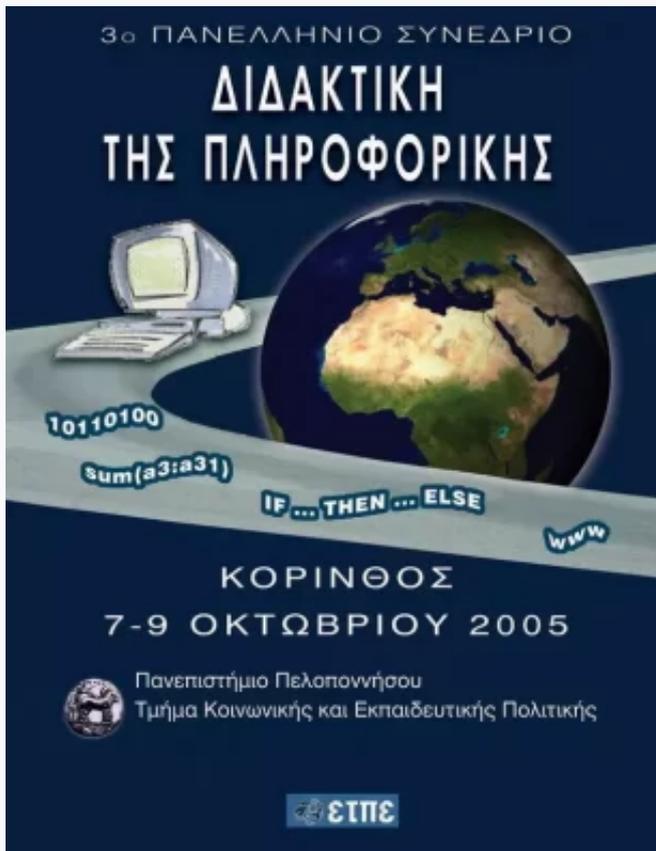


Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2005)

3ο Πανελλήνιο Συνέδριο «Διδακτική της Πληροφορικής»



Multi-Pascal για Windows: Ένα Προγραμματιστικό Περιβάλλον για την Διδασκαλία του Παράλληλου και Κατανεμημένου Προγραμματισμού

Αρετή Καπτάν, Κωνσταντίνος Μαργαρίτης

Βιβλιογραφική αναφορά:

Καπτάν Α., & Μαργαρίτης Κ. (2026). Multi-Pascal για Windows: Ένα Προγραμματιστικό Περιβάλλον για την Διδασκαλία του Παράλληλου και Κατανεμημένου Προγραμματισμού. *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, 1, 042-053. ανακτήθηκε από <https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/8694>

Multi-Pascal για Windows: Ένα Προγραμματιστικό Περιβάλλον για την Διδασκαλία του Παράλληλου και Κατανεμημένου Προγραμματισμού

Αρετή Καπτάν, Κωνσταντίνος Μαργαρίτης
Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας
areti@uom.gr, kmarg@uom.gr

ΠΕΡΙΛΗΨΗ

Η εργασία αυτή αποτελεί μία διδακτική προσέγγιση στις έννοιες της Παράλληλης και Κατανεμημένης (ΠκΚ) επεξεργασίας, με στόχο την πρακτική εφαρμογή των θεωρητικών εννοιών πάνω στον προγραμματισμό των ΠκΚ συστημάτων. Προτείνουμε μία σειρά μαθημάτων που ολοκληρώνονται με την επίλυση προγραμματιστικών ασκήσεων. Για την υλοποίηση των ασκήσεων αναπτύξαμε το ολοκληρωμένο προγραμματιστικό περιβάλλον Multi-Pascal για Windows (MPWin). Στην γραφική διεπαφή της MPWin ενσωματώσαμε εργαλεία για την εφαρμογή τεχνικών οπτικοποίησης, προκειμένου να ενισχύσουμε την διδασκαλία των θεωρητικών εννοιών του ΠκΚ προγραμματισμού, και να διευκολύνουμε την διαδικασία της σχεδίασης, υλοποίησης, αποσφαλμάτωσης, εκτίμησης και ρύθμισης της απόδοσης των ΠκΚ προγραμμάτων. Ακολουθεί μία συνοπτική αναφορά ορισμένων εργαλείων του περιβάλλοντος και παρουσιάζεται εν συντομία η χρήση τους και η συμβολή τους στην ενίσχυση της διδασκαλίας των πολύπλοκων ζητημάτων που προκύπτουν κατά τον προγραμματισμό των ΠκΚ συστημάτων.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Παράλληλος και κατανεμημένος προγραμματισμός, Προσομοιωτές, Οπτικοποίηση προγράμματος

ΕΙΣΑΓΩΓΗ

Τα Παράλληλα και Κατανεμημένα (ΠκΚ) υπολογιστικά συστήματα χρησιμοποιούνται ευρέως όχι μόνο για επιστημονικούς αλλά και για εμπορικούς σκοπούς. Η ευρεία αυτή χρήση των ΠκΚ συστημάτων καθιστά απαραίτητη την διδασκαλία του προγραμματισμού τους. Λογικό επακόλουθο αυτής της συνεχόμενης ανάπτυξης είναι το ότι δεν έχουν καθιερωθεί πρότυπα για την διδασκαλία του ΠκΚ προγραμματισμού. Παρά την διαρκώς αυξανόμενη χρήση των ΠκΚ συστημάτων, δεν έχει καθιερωθεί μία συγκεκριμένη αρχιτεκτονική ως ιδανική.

Κατά την διδασκαλία του ΠκΚ προγραμματισμού συνηθίζεται να δίνεται περισσότερη έμφαση είτε στους αλγορίθμους, είτε σε μία συλλογή παράλληλων προγραμματιστικών γλωσσών και συστημάτων, παρά στις έννοιες, τις δομές και τις βασικές αρχές που τον συνιστούν. Επιπλέον τα περισσότερα μαθήματα εστιάζουν σε μία μόνο από τις δύο

βασικές αρχιτεκτονικές προσεγγίσεις των ΠκΚ συστημάτων· είτε δηλαδή στον παράλληλο/ταντόχρονο προγραμματισμό συστημάτων πολυεπεξεργαστών, είτε στον προγραμματισμό κατανεμημένων συστημάτων. Τα τελευταία χρόνια έχουν αναπτυχθεί αρκετά περιβάλλοντα προσομοίωσης και οπτικοποίησης των ΠκΚ συστημάτων (Carr 2002, 2003, Koldehofe, Papatriantafidou & Tsigas 1999, Kurtz 1998, Strite & Null 2002). Τα περιβάλλοντα αυτά χρησιμοποιούνται για την ενίσχυση της διδασκαλίας των εννοιών του ΠκΚ προγραμματισμού. Παρόλα αυτά όμως, συνηθίζεται να εστιάζουν σε κάποια συγκεκριμένη αρχιτεκτονική και σε ζητήματα που αφορούν περισσότερο την υλοποίηση των ΠκΚ συστημάτων σε επίπεδο υλικού και λιγότερο τη συσχέτιση των θεωρητικών εννοιών με τις προγραμματιστικές δομές. Επιπλέον, τα περισσότερα περιβάλλοντα δεν παρέχουν επαρκή τεκμηρίωση και παραδείγματα για τη χρήση τους τόσο από τον καθηγητή όσο και από τον μαθητή.

Η παρούσα εργασία αποτελεί μια προσπάθεια για να προσεγγίσουμε διδακτικά της έννοιας του ΠκΚ προγραμματισμού μέσα από την χρήση ενός προσομοιωτή ΠκΚ εκτέλεσης και συστήνει μια σειρά από παραδείγματα που μπορούν να χρησιμοποιηθούν κατά την διδασκαλία. Ο αρχικός μας προσομοιωτής ήταν η Multi-Pascal (Lester 1993), την οποία τροποποιήσαμε κι ενισχύσαμε δημιουργώντας έτσι την Multi-Pascal για Windows (MPWin). Η MPWin (Kartan & Margaritis 2001) αποτελεί ένα ολοκληρωμένο προγραμματιστικό περιβάλλον που βασίζεται στον μεταγλωττιστή-διερμηνέα της Multi-Pascal κι εμπεριέχει ένα γραφικό περιβάλλον διεπαφής ενισχυμένο με πολλαπλά εργαλεία για την οπτικοποίηση των εκτελούμενων προγραμμάτων. Η οπτικοποίηση των προγραμμάτων αποσκοπεί στην κατανόηση και συσχέτιση των θεωρητικών εννοιών που διδάσκονται στην τάξη με τις κατάλληλες προγραμματιστικές δομές κατά την υλοποίηση, καθώς και στη διευκόλυνση της διαδικασίας αποσφαλμάτωσης, εκτίμησης και ρύθμισης της απόδοσής τους κατά την εκτέλεση.

Τα βασικά χαρακτηριστικά της διδακτικής μας προσέγγισης συνοψίζονται στις εξής συνιστώσες: i) στις θεωρητικές έννοιες του ΠκΚ προγραμματισμού που παρουσιάζονται με έναν ενιαίο τρόπο, ii) στη συσχέτιση των θεωρητικών εννοιών και των υλοποιήσεών τους αλγοριθμικά και iii) στην αποτελεσματική χρήση του ολοκληρωμένου προγραμματιστικού περιβάλλοντος της MPWin για την προσομοίωση κι εκτέλεση των υλοποιήσεων των θεωρητικών εννοιών προγραμματιστικά.

ΔΙΔΑΚΤΙΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΣΤΟΝ ΠΑΡΑΛΛΗΛΟ ΚΑΙ ΚΑΤΑΝΕΜΗΜΕΝΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

Σύμφωνα με τους (Fischer & Gross 1991) δύο είναι οι βασικές προϋποθέσεις που πρέπει να πληρεί ένα μάθημα ΠκΚ προγραμματισμού:

- Το μάθημα δεν πρέπει να περιορίζεται σε μία μοναδική αρχιτεκτονική, στις αρχιτεκτονικές ενός κατασκευαστή ή σε ένα συγκεκριμένο τύπο αρχιτεκτονικών. Αντίθετα, θα πρέπει να εστιάζει στα θεωρητικά μοντέλα των αρχιτεκτονικών που ενδείκνυνται για προγραμματισμό και στην μέτρηση κι εκτίμηση της απόδοσης των προγραμμάτων.

- Το μάθημα πρέπει να παρέχει επαρκή πρακτική εξάσκηση έτσι ώστε ο σπουδαστής να αποκτήσει την ικανότητα να χρησιμοποιεί ένα ΠκΚ σύστημα στην πράξη.
Από τα παραπάνω, είναι φανερό ότι η διάλεξη δεν μπορεί να αντικαταστήσει την εμπειρία που αποκομίζει ο σπουδαστής από την υλοποίηση και αποσφαλμάτωση ενός ΠκΚ προγράμματος ή από την παρατήρηση των αποδόσεων ενός προγράμματος κατά την εκτέλεσή του σε ένα πραγματικό ΠκΚ υπολογιστικό σύστημα. Επομένως μια ολοκληρωμένη διδακτική προσέγγιση στα ΠκΚ συστήματα πρέπει να προβλέπει την εξάσκηση των σπουδαστών στον προγραμματισμό τους. Κατά την άποψή μας το προγραμματιστικό περιβάλλον που θα επιλεγεί για να μυήσει τους σπουδαστές στην επίλυση παράλληλων προβλημάτων πρέπει:
- Να παρέχει μία απλή γλώσσα για την σύνταξη του κώδικα του ΠκΚ προγράμματος. Πιθανότατα μία γλώσσα που να περιλαμβάνει το συντακτικό –ή ένα υποσύνολό- μιας ήδη γνωστής γλώσσας προγραμματισμού με την προσθήκη δομών κι εντολών για την υλοποίηση των ΠκΚ εννοιών.
- Να συνοδεύεται από το ανάλογο βιβλίο μαθημάτων που θα ακολουθεί κοινή σύνταξη και κοινές δομές δεδομένων με την γλώσσα προγραμματισμού.
- Να υποστηρίζει τις βασικές αρχιτεκτονικές των ΠκΚ συστημάτων –παράλληλα συστήματα διαμοιραζόμενης μνήμης και παράλληλα συστήματα κατανεμημένης μνήμης.
- Να υποστηρίζει μεθόδους ανίχνευσης για την εκτίμηση και βελτιστοποίηση της απόδοσης των προγραμμάτων.
- Να περιλαμβάνει ή να μπορεί να επικοινωνεί με εργαλεία για την οπτικοποίηση του προγράμματος προκειμένου να διευκολύνεται η αποσφαλμάτωση και να αποσαφηνίζονται οι θεωρητικές έννοιες του ΠκΚ προγραμματισμού.
Λαμβάνοντας υπόψη τις παραπάνω προϋποθέσεις, στο τέλος κάθε διδακτικής ενότητας, οι σπουδαστές καλούνται να λύσουν μια σειρά από προγραμματιστικές ασκήσεις που σχετίζονται με τις θεωρητικές έννοιες της αντίστοιχης ενότητας.

ΠΡΟΣΟΜΟΙΩΤΕΣ ΠΑΡΑΛΛΗΛΩΝ ΚΑΙ ΚΑΤΑΝΕΜΗΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ

Προκειμένου να χρησιμοποιήσουμε στα μαθήματα μία γλώσσα προγραμματισμού που να ικανοποιεί τα κριτήρια που θέσαμε στην προηγούμενη παράγραφο αναπτύξαμε την MPWin, ένα ολοκληρωμένο προγραμματιστικό περιβάλλον προσομοίωσης ΠκΚ συστημάτων. Το προγραμματιστικό αυτό περιβάλλον χρησιμοποιείται για την υλοποίηση προγραμματιστικών ασκήσεων που αναθέτονται στους σπουδαστές κατά την διάρκεια των μαθημάτων. Ο όρος *προσομοιωτής παράλληλων και κατανεμημένων συστημάτων* αναφέρεται σε ένα προγραμματιστικό περιβάλλον στο οποίο δεν υφίσταται πραγματική παράλληλη εκτέλεση. Αντίθετα, ο προσομοιωτής παρέχει ένα συντακτικό για τον ορισμό ΠκΚ δομών κι εννοιών και αποτελείται από ένα σύστημα μεταγλωττιστή/διερμηνέα που έχει δημιουργηθεί σε μία γλώσσα υψηλού επιπέδου και το οποίο αναλαμβάνει την μεταγλώττιση, εκτέλεση και αποσφαλμάτωση του προγράμματος σε ένα υπολογιστικό σύστημα με έναν επεξεργαστή.

ΟΡΙΣΜΕΝΟΙ ΛΟΓΟΙ ΓΙΑ ΤΗΝ ΧΡΗΣΗ ΕΝΟΣ ΠΡΟΣΟΜΟΙΩΤΗ ΣΤΗ ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΠΑΡΑΛΛΗΛΟΥ ΚΑΙ ΚΑΤΑΝΕΜΗΜΕΝΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Για την διδασκαλία των βασικών εννοιών του ΠκΚ προγραμματισμού σε σπουδαστές του Τμήματός μας, οι οποίοι πρέπει να σημειωθεί ότι δεν έχουν προηγούμενη εμπειρία στην περιοχή της ΠκΚ επεξεργασίας, επιλέξαμε έναν προσομοιωτή παράλληλων και κατανεμημένων συστημάτων. Οι λόγοι που μας οδήγησαν στην επιλογή ενός προσομοιωτή έναντι της χρήσης ενός πραγματικού ΠκΚ συστήματος για την υλοποίηση των ασκήσεων είναι οι εξής:

- Ο προγραμματισμός των ΠκΚ συστημάτων απαιτεί εξειδικευμένη γνώση πάνω σε θέματα αρχιτεκτονικής και λειτουργικών συστημάτων, την οποία θα πρέπει να διαθέτει επιπρόσθετα ένας σπουδαστής χωρίς αυτή να ταυτίζεται απαραίτητα με τις έννοιες του ΠκΚ προγραμματισμού (Persky & Ben-Ari 1998).
- Τα ΠκΚ προγράμματα δεν έχουν πάντα επαναπαραγόμενη συμπεριφορά. Ακόμη κι όταν εκτελούνται με τα ίδια δεδομένα εισόδου η ροή εκτέλεσης των εντολών και των δεδομένων καθώς και τα αποτελέσματα τους μπορεί να διαφέρουν (μη επαναληψιμότητα, μη ντετερμινισμός).
- Η χρήση ενός πραγματικού ΠκΚ συστήματος απαιτεί από τον σπουδαστή την εκμάθηση ενός νέου και περίπλοκου προγραμματιστικού περιβάλλοντος, διαδικασία που αποδεικνύεται χρονοβόρα.
- Είναι συνηθισμένο φαινόμενο κατά την διδασκαλία του ΠκΚ προγραμματισμού οι ασκήσεις των μαθημάτων να υλοποιούνται σε διαφορετική προγραμματιστική γλώσσα από αυτή που χρησιμοποιεί το ΠκΚ σύστημα. Γεγονός που επιφορτίζει τον σπουδαστή με την μετάφραση μεταξύ διαφορετικών γλωσσών.
- Σε ένα πραγματικό ΠκΚ σύστημα οποιαδήποτε απόπειρα παρακολούθησης της εκτέλεσης του προγράμματος μπορεί να επηρεάσει την συμπεριφορά του. Για παράδειγμα, η χρήση εντολών αποσφαλμάτωσης μπορεί να μεταβάλλει δραστικά την συμπεριφορά και την απόδοση του προγράμματος (Tsai, Yang & Smith 1996).
- Τα ΠκΚ συστήματα έχουν υψηλό κόστος και η χρήση τους είναι συνήθως περιορισμένη. Επιπλέον, ακόμα κι όταν διατίθεται ένα τέτοιο σύστημα προς χρήση από τους προπτυχιακούς σπουδαστές ενός τμήματος, είναι σχεδόν βέβαιο ότι θα περιορίζεται σε μία συγκεκριμένη αρχιτεκτονική.
- Οι περισσότεροι προσομοιωτές είναι μεταφέρσιμοι και μπορούν να εγκατασταθούν σε οποιοδήποτε υπολογιστικό σύστημα γενικής χρήσης. Επομένως, παρέχεται στον σπουδαστή η δυνατότητα να πειραματιστεί πάνω στον ΠκΚ προγραμματισμό, χωρίς να απαιτείται η χρήση ενός εξειδικευμένου υπολογιστικού συστήματος.
- Ένας προσομοιωτής ΠκΚ συστημάτων, μπορεί να σχεδιαστεί έτσι ώστε να περιλαμβάνει όλες τις βασικές αρχιτεκτονικές των ΠκΚ συστημάτων –συστήματα διαμοιραζόμενης μνήμης και συστήματα κατανεμημένης μνήμης, παρέχοντας έτσι στον σπουδαστή την εμπειρία του προγραμματισμού σε ποικίλα συστήματα.

Στο σημείο αυτό πρέπει να αναφέρουμε ότι, κατά την άποψή μας, η χρήση ενός ΠκΚ προσομοιωτή αποτελεί μεν μια ιδανική εισαγωγική διδακτική προσέγγιση στην

διδασκαλία του ΠκΚ προγραμματισμού, δεν μπορεί όμως δε, να αντικαταστήσει την εμπειρία που προσφέρει ο προγραμματισμός ενός πραγματικού ΠκΚ συστήματος. Για τον λόγο αυτό προτείνουμε την συνδυασμένη χρήση των προσομοιωτών και των ΠκΚ συστημάτων.

ΠΑΡΑΛΛΗΛΗ ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΙ ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

Σε αυτή την ενότητα προτείνουμε μια σειρά μαθημάτων για την διδασκαλία του ΠκΚ προγραμματισμού σε προπτυχιακό επίπεδο. Κατά την άποψή μας, ο βασικός διδακτικός σκοπός των μαθημάτων είναι να εισάγει τους σπουδαστές στις βασικές έννοιες του ΠκΚ προγραμματισμού. Τα μαθήματα απευθύνονται σε τελειόφοιτους σπουδαστές, και προϋποθέτουν προηγούμενη εμπειρία των σπουδαστών πάνω στα κλασικά ζητήματα παραλληλισμού σε επίπεδο λειτουργικού συστήματος και βασικές γνώσεις στα δίκτυα υπολογιστών. Για το θεωρητικό υλικό των μαθημάτων προτείνουμε το βιβλίο του (Lester, 1993). Το πρώτο μισό του βιβλίου αφιερώνεται στα παράλληλα συστήματα διαμοιραζόμενης μνήμης, ενώ το δεύτερο μισό εστιάζει στα παράλληλα συστήματα κατανεμημένης μνήμης. Στο τέλος κάθε κεφαλαίου υπάρχουν θεωρητικές και πρακτικές ασκήσεις, ένα μέρος των οποίων αναθέτονται ως προγραμματιστικές εργασίες στους σπουδαστές. Τα μαθήματα στοχεύουν ταυτόχρονα στην κατανόηση των θεωρητικών εννοιών που διέπουν τα ΠκΚ συστήματα και στην πρακτική εξάσκηση πάνω στον προγραμματισμό των συστημάτων αυτών. Κατά αυτόν τον τρόπο, η επίλυση των προγραμματιστικών εργασιών προσφέρει στον σπουδαστή την δυνατότητα να εφαρμόσει τις θεωρητικές του γνώσεις σε σχετιζόμενα προγραμματιστικά προβλήματα. Το προγραμματιστικό περιβάλλον που προτείνουμε για την επίλυση των ασκήσεων είναι η MPWin.

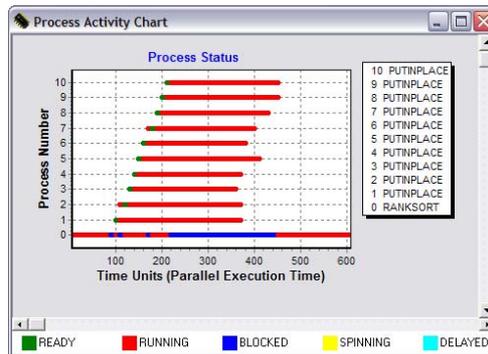
Multi-Pascal για Windows: ένα ολοκληρωμένο προγραμματιστικό περιβάλλον για την υλοποίηση των προγραμματιστικών ασκήσεων

Η Multi-Pascal για Windows αποτελεί την ενισχυμένη έκδοση της Multi-Pascal που προτείνεται ως γλώσσα προγραμματισμού στο (Lester 1993). Παρέχει ένα ολοκληρωμένο γραφικό περιβάλλον διεπαφής για την υλοποίηση, την προσομοίωση της εκτέλεσης σε διάφορες αρχιτεκτονικές, την αποσφαλμάτωση, την παρακολούθηση, την εκτίμηση και την ρύθμιση της απόδοσης ενός ΠκΚ προγράμματος. Οι λειτουργίες αυτές πραγματοποιούνται με την εφαρμογή μεθόδων οπτικοποίησης του προγράμματος μέσα από την χρήση διαφόρων εργαλείων που έχουν αναπτυχθεί για τον σκοπό αυτό κι έχουν ενσωματωθεί στην διεπαφή της MPWin.

Γράφημα Διεργασιακής Δραστηριότητας - Process Activity Chart

Το γράφημα στο Σχήμα 1 απεικονίζει τις καταστάσεις από τις οποίες διέρχεται κάθε διεργασία ως προς τον παράλληλο χρόνο εκτέλεσης του προγράμματος. Η εναλλαγή των καταστάσεων μιας διεργασίας σημειώνεται με την χρήση διαφορετικών χρωμάτων. Το γράφημα αυτό χρησιμοποιείται για την εξήγηση και κατανόηση βασικών εννοιών και

ζητημάτων που προκύπτουν κατά την εκτέλεση του προγράμματος και την υλοποίηση των ΠκΚ δομών:



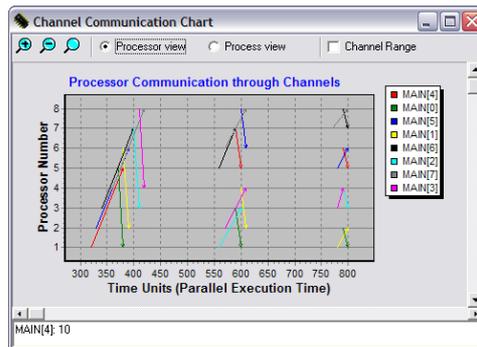
Σχήμα 1: Γράφημα διεργασιακής δραστηριότητας

- Για παράδειγμα, μπορούμε να επισημάνουμε την διαφορά μεταξύ των εντολών *forall* και *fork* που ενώ έχουν παρόμοια λειτουργία –την δημιουργία παράλληλων διεργασιών, η συμπεριφορά τους διαφέρει σημαντικά –η πρώτη αναστέλλει την εκτέλεση της διεργασίας-γονέα ενώ η δεύτερη όχι. Συγκρίνοντας τα γραφήματα δύο παρόμοιων προγραμμάτων θα παρατηρήσουμε ότι εκεί που χρησιμοποιείται η εντολή *forall*, η διεργασία-γονέας τίθεται σε κατάσταση αναμονής (*blocked status*) για όσο διάστημα εκτελούνται τα παιδιά της, ενώ στην εντολή *fork* η διεργασία-γονέας συνεχίζει την εκτέλεσή της παράλληλα με τα παιδιά της.
- Ένα άλλο θέμα γραφικής απεικόνισης προκύπτει κατά την υλοποίηση της έννοιας του αμοιβαίου αποκλεισμού των διεργασιών για την εκτέλεση των κρίσιμων τμημάτων ενός προγράμματος με τη χρήση μεταβλητών κλειδώματος και σηματοφορέων. Στην τεχνική του κλειδώματος οι αποκλειόμενες διεργασίες θέτονται σε ενεργό-αναμονή (*busy-waiting, spinning status*), ενώ κατά την χρήση των σηματοφορέων-καναλιών θέτονται σε κατάσταση αναμονής.
- Η διπλή λειτουργία των μεταβλητών καναλιών για τον συγχρονισμό των διεργασιών στα συστήματα διαμοιραζόμενης μνήμης και ως σύνδεσμοι επικοινωνίας στα συστήματα κατανεμημένης μνήμης απεικονίζεται με διαφορετικές καταστάσεις: στην πρώτη περίπτωση η διεργασία που αναστέλλεται λόγω συγχρονισμού, τίθεται σε κατάσταση αναμονής, ενώ στην δεύτερη περίπτωση η διεργασία που περιμένει να λάβει δεδομένα από έναν απομακρυσμένο επεξεργαστή μέσω ενός συνδέσμου επικοινωνίας τίθεται σε κατάσταση καθυστέρησης (*delayed status*).
- Το μέγεθος διεργασίας είναι άλλη μια έννοια που μπορεί να παρουσιαστεί μέσα από το γράφημα και να οριστεί ως το σύνολο των χρονικών στιγμών κατά τις οποίες μια διεργασία βρίσκεται σε κατάσταση εκτέλεσης (*running status*).

- Οι διαρκείς εναλλαγές των καταστάσεων των διεργασιών κατά την εκ περιδρομής εκτέλεσή τους, σε περίπτωση που οι επεξεργαστές του συστήματος είναι λιγότεροι από τις ταυτόχρονες διεργασίες που δημιουργούνται από την εκτέλεση του προγράμματος, γίνεται έκδηλη με μια απλή παρατήρηση του γραφήματος. Η χρήση του γραφήματος διευκολύνει σημαντικά την ανάδειξη και κατανόηση των παραπάνω ζητημάτων κατά την παρουσίαση τους μέσα στην τάξη.

Γράφημα Επικοινωνίας των Καναλιών - Channel Communication Chart

Οι επικοινωνίες μεταξύ των διεργασιών συντελούνται μέσω των καναλιών. Τα κανάλια -μεταβλητές τύπου *channel*- είναι ουσιαστικά ουρές μηνυμάτων FIFO. Το γράφημα στο Σχήμα 2 απεικονίζει τις λειτουργίες αποστολής και λήψης μηνυμάτων μεταξύ των διεργασιών ή των επεξεργαστών με τη μορφή τόξων, που αρχίζουν από τον αποστολέα και καταλήγουν στον παραλήπτη. Κάνοντας κλικ στην αρχή ενός τόξου εμφανίζεται το περιεχόμενο του μηνύματος που μεταδίδεται κατά την συγκεκριμένη επικοινωνία. Η απεικόνιση γίνεται σε συνάρτηση με τον παράλληλο χρόνο εκτέλεσης του προγράμματος. Η γραφική αναπαράσταση των επικοινωνιών αποκαλύπτει το επικοινωνιακό μοντέλο του προγράμματος το οποίο πολλές φορές παίζει σημαντικό ρόλο στην ανάλυση ενός αλγορίθμου. Τα τεμνόμενα τόξα φανερώνουν την ασύγχρονη επικοινωνία των διεργασιών, ενώ το μήκος τους προδίδει την επικοινωνιακή απόσταση και παρουσιάζει γραφικά την έννοια της επικοινωνιακής καθυστέρησης που υπάρχει κατά την μετάδοση μηνυμάτων μεταξύ των επεξεργαστών ενός καταναμημένου συστήματος.



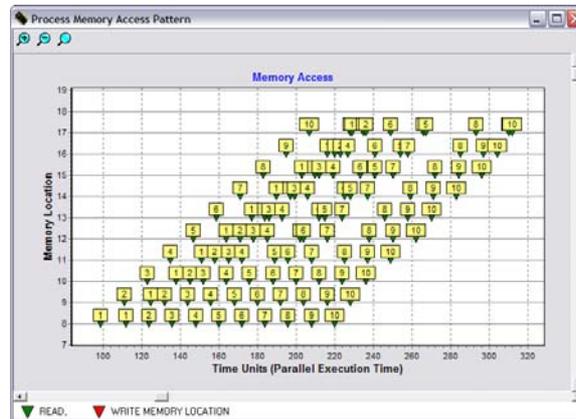
Σχήμα 2: Γράφημα επικοινωνίας καναλιών

Γράφημα Πρόσβασης των Διεργασιών στην Μνήμη - Memory Access Pattern

Η συνολική οργάνωση του συστήματος μνήμης είναι καθοριστική για την απόδοση ενός προγράμματος. Κι αυτό γιατί κατά την εκτέλεση μπορεί να προκύψει το φαινόμενο του ανταγωνισμού για πρόσβαση στη μνήμη. Το φαινόμενο αυτό μπορεί να εμφανιστεί τόσο στα παράλληλα συστήματα διαμοιραζόμενης μνήμης όσο και σε αυτά της

κατανεμημένης μνήμης. Η εμφάνιση αυτού του φαινομένου μπορεί να οφείλεται είτε στην αρχιτεκτονική της μνήμης του συστήματος, είτε στην οργάνωση του προγράμματος.

Μία μέθοδος που ακολουθείται για την αντιμετώπιση αυτού του προβλήματος είναι η προσπέλαση παρακείμενων διευθύνσεων. Σε πολλές περιπτώσεις προβλημάτων ζητείται η προσπέλαση διαδοχικών διευθύνσεων της μνήμης από διαφορετικούς επεξεργαστές ταυτόχρονα. Γεγονός που δημιουργεί προϋποθέσεις για την εμφάνιση ανταγωνισμού πρόσβασης στην μνήμη. Σε αυτές τις περιπτώσεις, για να μειωθεί ο ανταγωνισμός, παρακείμενες διευθύνσεις μνήμης αναθέτονται σε διαφορετικά τμήματα μνήμης. Αρχικά δημιουργείται ανταγωνισμός για την πρώτη διεύθυνση μνήμης της διάταξης, για παράδειγμα για το πρώτο στοιχείο ενός μεγάλου πίνακα δεδομένων. Αν υποθέσουμε ότι ο επεξεργαστής 1 αποκτά πρώτος πρόσβαση στο στοιχείο αυτό τότε καθυστερούνται όλοι οι υπόλοιποι επεξεργαστές. Όταν ο επεξεργαστής 1 προσπελάσει την επόμενη θέση μνήμης τότε ο επεξεργαστής 2 θα προσπελάσει την πρώτη θέση μνήμης κοκ. Με τη διαδικασία αυτή δημιουργείται ένα νέο μοντέλο πρόσβασης στην μνήμη όπως φαίνεται και στο Σχήμα 3 που δημιουργεί ένα κεκλιμένο επίπεδο. Μετά την αρχική καθυστέρηση που είναι ανάλογη του πλήθους των επεξεργαστών ο ανταγωνισμός απαλείφεται πλήρως. Το γράφημα μπορεί να βοηθήσει στην κατανόηση του μοντέλου πρόσβασης στην μνήμη και να διευκολύνει την σχεδίαση του προγράμματος έτσι ώστε να αυξηθεί η απόδοσή του.

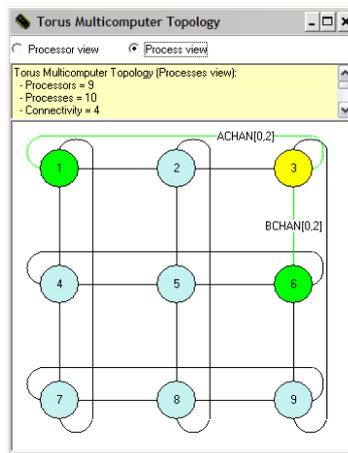


Σχήμα 3: Γράφημα πρόσβασης των διεργασιών στην μνήμη

Παράλληλα μπορεί να χρησιμοποιηθεί για την ανίχνευση των κρίσιμων τμημάτων ενός αλγορίθμου και να εντοπιστούν σημεία που μπορούν να προκαλέσουν πιθανά σφάλματα χρονισμού.

Γράφημα Κατανεμημένης Τοπολογίας - Multicomputer Topology

Η MPWin προσομοιώνει δύο ειδών συστήματα: τα παράλληλα συστήματα διαμοιραζόμενης μνήμης και τα παράλληλα συστήματα κατανεμημένης μνήμης. Για την δεύτερη κατηγορία συστημάτων παρέχει μια σειρά τοπολογιών: γραμμής (*line*), δακτυλίου (*ring*), πλέγματος δύο διαστάσεων (*mesh2*), τόρου (*torus*), πλέγματος τριών διαστάσεων (*mesh3*), υπερκύβου (*hypercube*) και πλήρους συνδεσμολογίας (*fullconnect*), οι οποίες –πλην της τελευταίας- οπτικοποιούνται σε μορφή γραφήματος. Στο γράφημα τοπολογίας στο Σχήμα 4 εμφανίζεται η διάταξη των επεξεργαστών και διεργασιών καθώς και οι μεταξύ τους συνδεσμολογία. Κάνοντας κλικ πάνω σε έναν επεξεργαστή εμφανίζονται οι ενεργοί γειτονικοί επεξεργαστές του και σημειώνονται τα ονόματα των μεταβλητών καναλιών που τον διασύνδεουν με αυτούς. Με τον όρο *ενεργός γειτονικός επεξεργαστής* αναφερόμαστε στον επεξεργαστή που συνδέεται με έναν άμεσο σύνδεσμο επικοινωνίας με τον επιλεγμένο επεξεργαστή και ο οποίος κατά την εκτέλεση του προγράμματος αποστέλλει μηνύματα στον δεύτερο. Ο επιλεγμένος επεξεργαστής (παραλήπτης) χρωματίζεται κίτρινος, ενώ οι ενεργοί γείτονες (αποστολείς) χρωματίζονται πράσινοι.



Σχήμα 4: Γράφημα κατανεμημένης τοπολογίας

Εδώ παρατηρούμε για πρώτη φορά τη φυσική διασύνδεση των επεξεργαστών και τον συσχετισμό της φυσικής τοπολογίας, δηλαδή του υλικού του κατανεμημένου συστήματος, με τις δομές του ΠκΚ προγράμματος, δηλαδή το λογισμικό. Σε κάθε επεξεργαστή της φυσικής τοπολογίας έχει ανατεθεί μία διεργασία προς εκτέλεση. Οι διεργασίες δημιουργούνται από το πρόγραμμα με την κλήση των εντολών δημιουργίας διεργασιών, *forall* και *fork*. Οι επεξεργαστές του κατανεμημένου συστήματος συνδέονται με τους συνδέσμους επικοινωνίας, οι οποίοι ορίζονται μέσα από το πρόγραμμα με την μορφή μεταβλητών καναλιών.

ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΕΚΤΑΣΕΙΣ

Μία αρχική έκδοση της MPwin, με ορισμένα μόνο από τα εργαλεία που αναφέρθηκαν στις προηγούμενες ενότητες, χρησιμοποιήθηκε για την διδασκαλία και την υλοποίηση των προγραμματιστικών ασκήσεων στα μαθήματα για τον ΠκΚ προγραμματισμό που διδάχθηκαν κατά τα προηγούμενα έτη στο Τμήμα της Εφαρμοσμένης Πληροφορικής του πανεπιστημίου μας. Τα μαθήματα παρακολούθησαν τελειόφοιτοι σπουδαστές οι οποίοι βρήκαν το περιβάλλον της MPWin αρκετά εύχρηστο και λειτουργικό. Η χρήση του προσομοιωτή βοήθησε σημαντικά το έργο του διδάσκοντα για την επίδειξη βασικών εννοιών και ζητημάτων που προκύπτουν από την παραλληλοποίηση ενός προγράμματος όπως, ο διαχωρισμός των εννοιών του επεξεργαστή και της διεργασίας, η λειτουργία των εντολών δημιουργίας διεργασιών, η απεικόνιση της επικοινωνίας των διεργασιών με τη χρήση των καναλιών κτλ.

Στα άμεσα σχέδιά μας περιλαμβάνεται η χρήση του προσομοιωτή και των νέων γραφημάτων κατά την διδασκαλία του ΠκΚ προγραμματισμού. Επίσης προβλέπεται η αξιολόγηση του προγραμματιστικού περιβάλλοντος και των εργαλείων του από τους σπουδαστές. Τέλος, θα μας ενδιέφερε και η μελέτη της μετάβασης από το περιβάλλον του προσομοιωτή σε ένα πραγματικό ΠκΚ σύστημα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Ben-Ari M. & Silverman S. (1999), DPLab: An environment for distributed programming, *Proceedings of the 4th annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE '99)*, 91-94, Cracow, Poland
- Ben-Ari M. (1990), *Principles of Concurrent and Distributed Programming*, Prentice Hall International Inc., NJ, USA
- Carr S., Fang C., Jozwowski T., Mayo J. & Shene C. K. (2003), ConcurrentMEntor: A Visualization System for Distributed Programming Education, *Proceedings of the 2003 International Conference on Parallel and Distributed Processing Techniques and Applications*, 1676-1682
- Carr S., Chen P., Jozwowski T. R., Mayo J. & Shene C.K. (2002), Channels, Visualization, and Topology Editor, *Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '02)*, 106-110, Aarhus, Denmark
- Fischer L. A. & Gross T. (1991), Teaching the programming of parallel computers, *Proceedings of the 22nd SIGCSE Technical Symposium on Computer Science Education (SIGCSE '91)*, 102-107, TX USA
- Kaptan A. & Margaritis K. (2001), Computer Simulators for Teaching Concurrent Programming, Conference on Computers and Education, *Proceedings of the 5th Hellenic-European Conference on Computer Mathematics and its Applications*, Vol. 1, 332-336, Athens
- Koldehofe B., Papatriantafilou M. & Tsigas P. (1999), Distributed algorithms visualization for educational purposes, *Proceedings of the 4th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education (ITiCSE '99)*, Vol.31.3 of SIGCSE Bulletin inroads, 103-106, NY USA
- Kurtz B. L., Cai H., Plock C. & Chen X. (1998), A concurrency simulator designed for sophomore-level instruction, *Proceedings of the 29th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '98)*, 237-241, Atlanta, GA, USA
- Lester B. P. (1993), *The art of parallel programming*, NJ: Prentice Hall International Inc.

- Persky Y. & Ben-Ari M. (1998), Re-engineering a concurrency simulator, *Proceedings of the 6th Annual Conference on the Teaching of Computing and the 3rd Annual Conference on Integrating Technology into Computer Science Education: Changing the Delivery of Computer Science Education (ITiSCE '98)*, 185-188, Dublin, Ireland
- Strite D. & Null L. (2002), BACI debugger: A GUI debugger for the BACI system, *Consortium for Computing in Small Colleges*, 112-122
- Tsai J. J. P., Bi Y., Yang S. J. H. & Smith R. A. W. (1996), *Distributed real-time systems: monitoring, visualization, debugging and analysis*, John Wiley & Sons Inc.