



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

2011

Ανάπτυξη διαδικτυακής πύλης για τη χρήση των πακέτων MATLAB και MPI

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

των

Βαλιαντή Βασιλική

Πολυχρονιάδου Βασιλική



Επιβλέπων : Κωνσταντίνος Ι. Διαμαντάρας

Καθηγητής



ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη διαδικτυακής πύλης για τη χρήση των πακέτων MATLAB και MPI

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

των

Βαλιαντή Βασιλική
Πολυχρονιάδου Βασιλική

Επιβλέπων : Κωνσταντίνος Ι. Διαμαντάρας
Καθηγητής

Θεσσαλονίκη, Μάιος 2011

Περίληψη

Ο σκοπός της πτυχιακής εργασίας ήταν ο σχεδιασμός και η ανάπτυξη διαδικτυακής πύλης για τη χρήση των πακέτων MATLAB και MPI. Για να υλοποιηθεί αυτό χρησιμοποιήσαμε την τεχνολογία mediawiki. Φύσικα θα μπορούσε να χρησιμοποιηθεί οποιοδήποτε άλλο wiki άφου η βασική ιδέα είναι ίδια.

Το wiki είναι μια διαδικτυακή τεχνολογία που επιτρέπει να δημιουργήσουμε μια συλλογική ιστοσελίδα πληροφόρησης στο διαδίκτυο. Η υλοποίησή της είναι γρήγορη και απλή. Το περιεχόμενο μπορεί να γραφτεί και να τροποποιηθεί από οποιονδήποτε. Κάθε χρήστης έχει την δυνατότητα να αλλάξει και να εμπλουτίσει το ήδη υπάρχον περιεχόμενο.

Στην παρούσα εργασία δημιουργήθηκαν δύο τέτοια wiki, ένα για το **Matlab** και ένα για το **Mpi** αντίστοιχα.

Το Matlab (MATrix LABoratory) είναι ένα interactive σύστημα για:

- ✓ αριθμητικούς υπολογισμούς, δίχως προγραμματισμό σε συμβατικές γλώσσες (Fortran, C)·
- ✓ γρήγορη ανάπτυξη και έλεγχο αλγορίθμων, (πλήθος έτοιμων συναρτήσεων και απλουστευμένη αλγοριθμική γλώσσα)·
- ✓ ανάλυση δεδομένων και γραφική παρουσίαση τους·
- ✓ εφαρμογές από διάφορες θεματικές περιοχές μέσω κατάλληλων toolboxes

Το MPI – **M**essage **P**assing **I**nterface

- ✓ Είναι ένα επεκταμένο μοντέλο μεταβίβασης μηνυμάτων
- ✓ Είναι η πρώτη τυποποιημένη, βιβλιοθήκη μεταβίβασης μηνυμάτων που στηρίζεται στην συναίνεση του MPI forum
- ✓ Είναι ένα πρωτόκολλο επικοινωνίας το οποίο χρησιμοποιείται για τον προγραμματισμό παράλληλων υπολογιστικών συστημάτων
- ✓ Δεν είναι γλώσσα προγραμματισμού
- ✓ Δεν αποτελεί μια συγκεκριμένη υλοποίηση

Βασική ιδέα: επικοινωνία μεταξύ ομάδων διεργασιών

Με το πέρας αυτής της εργασίας οι δύο διαδικτυακές πύλες για Matlab και Mpi είναι διαθέσιμες στον παγκόσμιο ιστό για μελέτη και επιπλέον εμπλουτισμό του περιεχομένου τους από νέους χρήστες.

Abstract

The purpose of this thesis was the design and development portal for the use of packages MATLAB and MPI. To achieve this we used the technology mediawiki. Of course you could use any other wiki since the basic idea is the same.

A wiki is a web technology that allows you to create a collaborative website on the Internet. The implementation is fast and simple. Content can be written and modified by anyone. Each user has the ability to change and enrich the existing content.

In this study created two such wiki, one for Matlab and one for Mpi, respectively.

To Matlab (MATrix LABoratory) is an interactive system:

- ✓ numerical computation without conventional programming languages (Fortran, C);
- ✓ fast development and control algorithms (number ready functions, and simplified algorithmic language);
- ✓ data analysis and graphical display them;
- ✓ applications from various subject areas through appropriate

The MPI - **M**essage **P**assing **I**nterface

- ✓ It is an expanded model transfer messages
- ✓ It is the first standardized, library transfer messages based on the consensus of the MPI forum
- ✓ It is a communication protocol that is used for programming parallel computing systems
- ✓ It is not is a programming language
- ✓ It is not a specific implementation

Basic idea: communication processes between teams

Upon completion of this work are two portals in Matlab and Mpi are available on the web for study and further enrich the content by new users.

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε κάποια πρόσωπα η συμβολή των οποίων ήταν κάτι παραπάνω από πολύτιμη. Πρώτα από όλους τον επίβλεποντα καθηγητή Κ.Διαμαντάρα η αρωγή του οποίου ήταν καταλυτική για την περάτωση αυτής της πτυχιακής εργασίας. Επίσης ευχαριστούμε τον καθ. Σ.Χαρχαλάκη καθώς και τον φίλο και συνάδελφο Θ.Ζησόπουλο για τη σημαντική βοήθεια τους στο ξεκίνημα αυτής της εργασίας.

Πίνακας περιεχομένων

Περίληψη	4
Abstract	5
Ευχαριστίες	6
Εισαγωγή	10
1 Wiki	11
1.1 WIKI: ΤΙ ΕΙΝΑΙ ΑΥΤΟ;	11
1.2 ΣΥΝΤΟΜΗ ΙΣΤΟΡΙΑ ΤΟΥ WIKI	12
1.3 ΒΑΝΔΑΛΙΣΜΟΙ ΚΑΙ ΠΡΟΣΤΑΣΙΑ	14
1.4 ΕΦΑΡΜΟΓΕΣ ΠΟΥ ΒΑΣΙΖΟΝΤΑΙ ΣΤΟ WIKI	14
1.4.1 WikiWikiWeb	14
1.4.2 Wikimedia Foundation.....	15
1.5 ΕΦΑΡΜΟΓΕΣ ΜΑΘΗΣΗΣ & ΔΙΔΑΣΚΑΛΙΑΣ	16
1.5.1 Εφαρμογές Wiki στη μάθηση και τη διδασκαλία	16
1.6 WIKIPEDIA	19
1.7 ΛΟΓΙΣΜΙΚΟ WIKI & WIKI ENGINES	21
1.7.1 Λογισμικό Wiki.....	21
1.8 ΣΥΓΚΡΙΣΗ ΤΩΝ ΚΑΛΥΤΕΡΩΝ ΜΗΧΑΝΩΝ WIKI	23
1.9 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ WIKI	26
1.10 ΣΥΓΓΡΑΦΗ ΣΤΟ WIKI	27
1.10.1 Αναπαραστάσεις Σελίδας	27
1.10.2 Πρότυπα.....	29
1.10.3 Σύνδεση και δημιουργία των σελίδων	30
1.10.4 Αναζήτηση.....	30
1.10.5 Wiki από την πλευρά του εξυπηρετητή είτε από την πλευρά του πελάτη.....	31
1.10.6 Ελέγχοντας Αλλαγές	31
1.11 WIKI ΚΟΙΝΟΤΗΤΕΣ	32
1.12 ΣΥΓΚΡΙΣΗ ΤΩΝ ΣΕΛΙΔΩΝ WIKI ΜΕ ΤΑ BLOGS	33
1.13 ΣΥΓΚΡΙΣΗ WIKI SITES & ΚΟΙΝΩΝ WEBSITES	33
1.14 ΥΠΕΡ ΚΑΙ ΚΑΤΑ ΤΗΣ ΧΡΗΣΗΣ ΤΩΝ WIKIS	34
1.14.1 Σχετικά με την ποιότητα του περιεχομένου.....	35
1.15 ΚΡΙΤΗΡΙΑ ΕΠΙΛΟΓΗΣ ΤΟΥ ΣΩΣΤΟΥ WIKI	35
1.16 ΕΛΕΓΧΟΣ ΤΩΝ ΔΗΜΟΣΙΕΥΜΑΤΩΝ	37
1.16.1 Έλεγχος δημοσίευσης (Publishing controls)	37
1.17 ΟΦΕΛΗ ΑΠΟ ΤΑ WIKI ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ	39
1.17.1 Οφέλη που προκύπτουν από τη χρήση των wiki ως εργαλεία διδασκαλίας και μάθησης:	39

1.18 ΘΕΩΡΙΕΣ ΕΦΑΡΜΟΣΜΕΝΕΣ ΣΤΑ WIKI.....	40
1.18.1 Εκπαιδευτικές θεωρίες που εφαρμόζονται στα wiki.....	40
1.19 ΑΠΟΦΑΣΕΙΣ ΓΙΑ WIKI ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ	41
1.20 ΤΡΟΠΟΙ ΧΡΗΣΗΣ ΣΤΙΣ ΕΠΙΧΕΙΡΗΣΕΙΣ	42
2 ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΣΤΟ LOCALHOST.....	44
3 ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΣΤΟ DIAS.....	49
4 ΕΠΕΞΕΡΓΑΣΙΑ WIKI.....	54
4.1 ΑΛΛΑΓΗ ΚΕΝΤΡΙΚΗΣ ΕΙΚΟΝΑΣ-LOGO	54
4.2 ΑΛΛΑΓΗ ΧΡΩΜΑΤΟΣ.....	54
4.3 ΕΠΕΞΕΡΓΑΣΙΑ ΚΕΙΜΕΝΟΥ ΣΕΛΙΔΑΣ	55
4.3.1 Στοιχηση.....	56
4.3.2 Συνδέσεις, URLs, Εικόνες.....	58
4.3.3 Μορφοποίηση Χαρακτήρων.....	65
4.4 ΠΡΟΣΘΗΚΗ ΑΝΑΔΥΟΜΕΝΟΥ MENU ΣΤΟ WIKI	70
5 Matlab-Wiki	74
5.1 ΕΙΣΑΓΩΓΗ	74
5.2 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ.....	75
5.3 ΕΡΓΑΛΕΙΟΘΗΚΕΣ(TOOLBOXES).....	76
5.4 ΠΕΡΙΒΑΛΛΟΝ ΕΡΓΑΣΙΑΣ.....	76
5.5 Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ MATLAB	78
5.5.1 Βασικές πράξεις	78
5.5.2 Η εντολή καταχώρησης.....	79
5.5.3 Ο χώρος εργασίας(workspace).....	79
5.5.4 Matlab Search Path.....	79
5.5.5 Διανύσματα, πίνακες, cells.....	80
5.5.6 Δομές ελέγχου ροής.....	80
5.5.7 Δομές (Structures).....	81
5.5.8 Συναρτήσεις (Functions)	81
5.5.9 Επεξεργασία αρχείων	82
5.5.10 Interpreter και Compiler	83
5.5.11 Ο Editor/Debugger	83
5.6 ΓΡΑΦΙΚΕΣ ΠΑΡΑΣΤΑΣΕΙΣ.....	85
5.6.1 Γραφικές Παραστάσεις 2 Διαστάσεων	85
5.6.2 Γραφικές Παραστάσεις 3 Διαστάσεων	92
5.7 ΑΝΑΠΤΥΞΗ ΓΡΑΦΙΚΩΝ ΕΦΑΡΜΟΓΩΝ	99
5.7.1 Παρουσίαση του GUIDE (GUI Layout Editor)	99
5.7.2 Παρουσίαση και δημιουργία gui αντικειμένων	100
5.7.3 Αποθήκευση και έναρξη εφαρμογής	101
5.7.4 Fig και M-File του figure	103
5.7.5 Ιδιότητες των gui αντικειμένων	104
5.7.6 Δέντρο αντικειμένων(handles, findall, guihandles)	105

5.7.7 Οι εντολές set, get.....	106
5.7.8 Callbacks.....	106
5.7.9 Flexarrays (ActiveX control).....	107
5.8 NEURAL NETWORK TOOLKIT	108
5.8.1 ΔΟΜΕΣ ΚΑΙ ΣΥΝΑΡΤΗΣΕΙΣ ΤΟΥ NEURAL NETWORK TOOLKIT.....	108
5.8.2 NEURAL NETWORK TOOLKIT PLOTS.....	113
5.8.3 INTERFACE ΤΟΥ NEURAL NETWORK TOOLKIT.....	117
6 Mpi-Wiki.....	118
6.1 ΠΑΡΑΛΛΗΛΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ.....	118
6.1.1 Εισαγωγή	118
6.1.2 Ταξινομήσεις Παράλληλων Αρχιτεκτονικών	119
6.1.3 Μοντέλα Παράλληλης Επεξεργασίας	122
6.1.4 Πέρασμα μηνυμάτων	125
6.2 MPI.....	126
6.2.1 Ιστορική Εξέλιξη	126
6.2.2 Οι στόχοι του MPI	127
6.2.3 Βασικές έννοιες	129
6.2.4 Η Διεργασία.....	130
6.2.5 Μέσο Επικοινωνίας (Communicator).....	132
6.2.6 Είδη επικοινωνιών ανάμεσα σε διεργασίες	133
6.2.7 Η σύνταξη του MPI μηνύματος	134
6.2.8 Οι Τύποι Δεδομένων του MPI	135
6.3 ΟΙ ΒΑΣΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ MPI	136
6.3.1 Αρχικοποίηση του MPI	137
6.3.2 Τερματισμός του MPI	138
6.3.3 Communicator.....	139
6.3.4 Point-to-Point	140
6.4 ΚΟΜΒΟ - ΜΕ – ΚΟΜΒΟ ΕΠΙΚΟΙΝΩΝΙΑ (Point-to-Point)	143
6.4.1 Είδη διαχωρισμού για τις διαδικασίες επικοινωνίας Point-to-Point.....	145
6.4.2 Παρεμποδιστικές Διαδικασίες (blocking).....	147
6.4.3 Μη Παρεμποδιστικές Επικοινωνίες (non blocking)	155
6.5 ΣΥΛΛΟΓΙΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ	157
6.5.1 Οι θεμελιώδεις συναρτήσεις της συλλογικής επικοινωνίας.....	161
6.6 MPI ΣΥΝΑΡΤΗΣΕΙΣ ΚΑΙ ΣΤΑΘΕΡΕΣ	166
ΠΑΡΑΡΤΗΜΑ Α.....	173
ΠΑΡΑΡΤΗΜΑ Β.....	191

Εισαγωγή

Η εργασία με θέμα «σχεδιασμός και η ανάπτυξη διαδικτυακής πύλης για τη χρήση των πακέτων MATLAB και MPI» εκπονήθηκε στα πλαίσια πτυχιακής εργασίας στο τμήμα πληροφορικής του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Θεσσαλονίκης. Η εργασία χωρίζεται σε έξι κεφάλαια και δύο παραρτήματα.

Στο πρώτο κεφάλαιο παρουσιάζεται συνοπτικά η ιστορία του wiki και τα βασικά χαρακτηριστικά του. Περιγράφονται επίσης τα πλεονεκτήματα και τα μειονεκτήματα του καθώς και οι σημαντικότερες εφαρμογές του.

Στο δεύτερο κεφάλαιο περιγράφονται αναλυτικά τα βήματα που ακολουθήθηκαν για τη δημιουργία της τοπικής βάσης δεδομένων στους προσωπικούς μας υπολογιστές ώστε να κατασκευάσουμε μια τοπική πλατφόρμα mediawiki που θα είναι ορατή μόνο σε μας.

Στο τρίτο κεφάλαιο παρουσιάζονται ένα ένα τα βήματα που ακολουθήθηκαν για τη δημιουργία της βάσης και της πλατφόρμας του wiki στον παγκόσμιο ιστό.

Το τέταρτο κεφάλαιο αναφέρεται στη γλώσσα που χρησιμοποιεί το wiki για την επεξεργασία του περιεχομένου του. Περιγράφονται αναλυτικά ολόι οι πιθανοί τρόποι επεξεργασίας και δίνονται σχετικά παραδείγματα ενώ ιδιαίτερη έμφαση δίνεται και στα τμήματα κώδικα rhr που χρησιμοποιήθηκαν για τη μορφοποίηση της σελίδας των wikis.

Στο πέμπτο και έκτο κεφάλαιο περιγράφεται το περιεχόμενο των wikis, δηλαδή το matlab και το mpi αντίστοιχα.

Τέλος, στο πρώτο παράρτημα δίνεται ο πηγαίος κώδικας της εφαρμογής και στο δεύτερο η βιβλιογραφία.

1 Wiki

Σε αυτό το κεφάλαιο παρουσιάζονται βασικές πτυχές μιας από τις πλέον αξιόλογες διαδραστικές υπηρεσίες του διαδικτύου, του wiki. Σκοπός του κεφαλαίου είναι η παρουσίαση του wiki από το ξεκίνημα του μέχρι σήμερα, τα βασικά χαρακτηριστικά του καθώς επίσης και τα πλεονεκτήματα και τη χρησιμότητα του ως ένα από τα καταλληλότερα εργαλεία για τη μετάδοση της πληροφορίας, τη διευκόλυνση της συζήτησης και την ανταλλαγή απόψεων μέσα από τον ελεύθερο σχολιασμό των δημοσιευμάτων.

1.1 WIKI: ΤΙ ΕΙΝΑΙ ΑΥΤΟ;

Ένα Wiki είναι συνήθως μία ιστοσελίδα που επιτρέπει στους χρήστες της να προσθέσουν, να αφαιρέσουν, ή να επεξεργαστούν το περιεχόμενό της, πολύ γρήγορα και εύκολα, χωρίς να έχουν κάνει υποχρεωτικά εγγραφή. Έτσι, διευκολύνεται η συνεργασία πολλών ατόμων για τη συγγραφή ενός έργου. Ο όρος Wiki όμως, μπορεί να αναφέρεται και στο λογισμικό που χρησιμοποιείται για να κατασκευαστούν Wiki σελίδες.

Αυτό που στην ουσία κάνει ένα σύστημα wiki είναι να απλοποιεί τη διαδικασία δημιουργίας σελίδων HTML και να καταγράφει κάθε μεμονωμένη αλλαγή που εμφανίζεται κατά τη διάρκεια του χρόνου, έτσι ώστε σε οποιαδήποτε στιγμή μια σελίδα να μπορεί να επανέλθει σε κάποια από τις προηγούμενες καταστάσεις της. Πολλές φορές το wiki περιλαμβάνει εργαλεία που επιτρέπουν στους χρήστες να παρακολουθούν την κατάστασή του. Μπορεί ακόμη να παρέχει στους χρήστες κάποιο χώρο για να συζητούν διάφορα θέματα όπως για παράδειγμα το περιεχόμενο που προστίθεται στο site.

Τα περισσότερα wikis επιτρέπουν την πρόσβαση των χρηστών χωρίς κανέναν απολύτως περιορισμό. Έτσι όλοι έχουν το δικαίωμα να συμβάλουν στη συγγραφή του περιεχομένου της ιστοσελίδας χωρίς να υποβληθούν σε διαδικασία "εγγραφής" όπως συνήθως επιβάλλεται σε σελίδες συζητήσεων

π.χ. στα περισσότερα forum. Αυτό σημαίνει ότι σε πολλές περιπτώσεις δεν είναι δυνατό να ελεγχθεί η εγκυρότητα των πληροφοριών των wiki σελίδων.

Η λέξη Wiki ερμηνεύεται μερικές φορές ως ακρώνυμο για το "What I know is" δηλαδή "Αυτό που εγώ ξέρω είναι". Είναι μια χαρακτηριστική φράση για τον τρόπο λειτουργίας του Wiki: ο κάθε χρήστης που συμμετέχει στη συγγραφή κάποιου έργου προσθέτει την προσωπική του γνώση, έτσι ώστε όλοι να μπορούν να τη μοιράζονται.



1.2 ΣΥΝΤΟΜΗ ΙΣΤΟΡΙΑ ΤΟΥ WIKI

- Το πρώτο wiki, δημιουργήθηκε το 1994 και εγκαταστάθηκε στον Ιστό στις 25 Μαρτίου του 1995 από τον Ward Cunningham. Αυτός εφηύρε το όνομα και την έννοια wiki και κατασκεύασε την πρώτη μηχανή wiki. Το Wiki αυτό είναι ευρύτερα γνωστό ως Wiki Wiki Web και φιλοξενείται στην ιστοσελίδα: <http://c2.com/cgi/wiki?WelcomeVisitors>. Δημιουργήθηκε στην κοινότητα των design pattern σαν ένα συμπλήρωμα του Portland Pattern Repository για να διευκολύνει τη συνεργασία πολλών προγραμματιστών. Ορισμένοι υποστηρίζουν ότι μόνο το αρχικό wiki πρέπει να λέγεται Wiki (με κεφαλαίο) ή WikiWikiWeb.
- Ο Cunningham εμπνεύστηκε τον όρο wiki από τα "wiki wiki" δηλαδή τα "γρήγορα" λεωφορεία πυκνών δρομολογίων στον αερολιμένα της

Χονολουλού. Το wiki Wiki ήταν η πρώτη χαβανέζικη λέξη που έμαθε στην πρώτη επίσκεψή του στα νησιά, όταν τον κατεύθυνε ο πράκτορας αερολιμένων για να πάρει το λεωφορείο wikiwiki μεταξύ των τερματικών. Ο Cunningham δήλωσε: "επέλεξα το wiki-wiki ως παρηχητικό υποκατάστατο για το "γρήγορα" και με αυτόν τον τρόπο απέφυγα την ονομασία quick-web".

- Προς το τέλος της δεκαετίας του '90, η ιδέα της χρήσης των wikis σε ιδιωτικές και δημόσιες βάσεις γνώσεων άρχισε να βρίσκει μεγάλη αποδοχή. Αυτό το γεγονός ενέπνευσε τους ιδρυτές του έργου Nupedia Jimmy Wales και Larry Sanger να χρησιμοποιήσουν την τεχνολογία wiki σαν βάση για μια ηλεκτρονική εγκυκλοπαίδεια. Η Wikipedia ξεκίνησε τον Ιανουάριο του 2001 και αρχικά βασίστηκε στο λογισμικό UseMod. Αργότερα μεταπήδησε στον δικό της ανοικτό κώδικα codebase, που υιοθετήθηκε στη συνέχεια από πολλά άλλα wikis.
- Στις αρχές του 2000, τα wikis άρχισαν να υιοθετούνται όλο και περισσότερο στις επιχειρήσεις σαν λογισμικό που επιτρέπει τη συνεργασία πολλών χρηστών. Οι πιο συνηθισμένες χρήσεις του ήταν στις επικοινωνίες, σε intranets και στη συλλογική τεκμηρίωση προγραμμάτων, αρχικά από τεχνικούς χρήστες.
- Το Δεκέμβριο του 2002, η Socialtext προώθησε την πρώτη εμπορική ανοικτή λύση wiki: Wikis ανοικτού κώδικα όπως τα MediaWiki, Kwiki και TWiki που ξεπέρασαν το 1εκ downloads στα repos του Sourceforge μέχρι και το 2004. Σήμερα ορισμένες επιχειρήσεις χρησιμοποιούν τα wikis σαν το μοναδικό συλλογικό λογισμικό και σαν υποκατάστατο των στατικών intranets. Κατά μια άποψη, γίνεται μεγαλύτερη χρήση των wikis πίσω από τα firewalls παρά στο δημόσιο διαδίκτυο.
- Το 2005 οι Los Angeles Times πειραματίστηκαν με τη χρήση wiki στο εκδοτικό τμήμα της ιστοσελίδας τους. Το έργο Wikitorial σύντομα έκλεισε καθώς βάνδαλοι το κατέστρεψαν.

1.3 ΒΑΝΔΑΛΙΣΜΟΙ ΚΑΙ ΠΡΟΣΤΑΣΙΑ

Η ανοικτή φιλοσοφία των περισσότερων wikis — το να επιτρέπεται στον καθένα να τροποποιεί το περιεχόμενο — δεν εξασφαλίζει ότι όλοι οι συντάκτες έχουν καλές προθέσεις. Ο βανδαλισμός είναι ένα σταθερό πρόβλημα για τα wikis, αν και μερικές πηγές με λίγη σχετική εμπειρία με wikis τείνουν να υπερβάλλουν τον κίνδυνο του βανδαλισμού. Μελέτες της IBM ισχυρίζονται ότι το μεγαλύτερο μέρος του βανδαλισμού στη Wikipedia αφαιρείται σε 5 λεπτά ή λιγότερο. Εντούτοις, σε μερικές περιπτώσεις παίρνει περισσότερο, όπως στην περίπτωση της βιογραφίας του John Seigenthaler. Σε μερικές περιπτώσεις, προκειμένου να αποφευχθεί ο παραπέρα βανδαλισμός αποκλείονται από τη δυνατότητα συγγραφής λογαριασμοί χρηστών ή διευθύνσεις IP.

1.4 ΕΦΑΡΜΟΓΕΣ ΠΟΥ ΒΑΣΙΖΟΝΤΑΙ ΣΤΟ WIKI

1.4.1 WikiWikiWeb

Το WikiWikiWeb είναι η πρώτη εφαρμογή wiki που έχει πραγματοποιηθεί και περιελάμβανε διάφορα θέματα σχετικά με την τεχνολογία λογισμικού . Ο όρος wiki που αναφέρεται σε sites με ιστοσελίδες στις οποίες μπορούν να γράψουν οι χρήστες , πχ Wikipedia , προήλθε από το αρχικό wiki. Αναπτύχθηκε από τον Ward Cunningham το 1994 και η ονομασία του προήλθε από την χαβανέζικη λέξη “wiki” που σημαίνει γρήγορα.

Η ως τώρα δημοτικότητα των wikis ,οφείλεται στην ιδέα των editors και των επισκεπτών του WikiWikiWeb, να δημιουργήσουν δικές τους ιστοσελίδες, που να μπορούν να διαμορφωθούν από τους χρήστες . Κατά αυτόν τον τρόπο προέκυψαν νέες μηχανές wiki και νέα wikis . Οι νέες κοινότητες wikis σχετιζόταν με θέματα διαφορετικά από τον προγραμματισμό.

1.4.2 Wikimedia Foundation

Η Wikimedia Foundation είναι ένας μη κερδοσκοπικός οργανισμός που ιδρύθηκε στις 20 Ιουνίου του 2003 από τον Jimmy Wales, και έχει ως έδρα την Florida των ΗΠΑ. Στόχος αυτού του οργανισμού είναι η ανάπτυξη wiki-based projects και η δωρεάν παροχή τους στο κοινό. Η ανάπτυξη αυτών των projects οφείλεται κυρίως σε δωρεές.

Προσωπικό της Wikimedia

Το προσωπικό της απαρτίζεται κυρίως από εθελοντές : προγραμματιστές , server administrators , τεχνικούς και communications managers . Επιπλέον το Advisory Board (ένα παγκόσμιο δίκτυο ειδικών) προσφέρει αφιλοκερδώς τη βοήθειά του σε νομικού , τεχνολογικού και αναπτυξιακού προσανατολισμού θέματα.

Projects

Παρακάτω βλέπουμε μερικά από τα πιο σημαντικά wiki – projects που έχει αναλάβει η Wikimedia :

Πίνακας 1 Wiki Projects

Wikipedia	15-1-2001	Εγκυκλοπαίδεια που περιλαμβάνει περισσότερα από 7 εκατομμύρια άρθρα σε 250 γλώσσες
Wictionary	12-12-2002	Λεξικό , συνώνυμα ,ετυμολογίες , μεταφράσεις
Wikibooks	7-10-2003	Συλλογή εκπαιδευτικού υλικού και βιβλίων
Wikiquote	7-10-2003	Συλλογή quotations
Wikisource	24-11-2003	Project που παρέχει και μεταφράζει έγγραφα όπως δημόσια έγγραφα
Wikimedia Commons	7-9-2004	Αποθήκη εικόνων, ήχου και video με πάνω από 1,000,000 αρχεία
Wikimedia Incubator		Ελέγχει νέα Wikimedia projects και εκδόσεις παλαιών projects σε νέες γλώσσες
Wikispecies	13-9-2004	Κατάλογος ειδών : ζώα βακτήρια πλαγκτόν και άλλα είδη ζωής
Wikinews	3-12-2004	Νέα από διάφορες χώρες του κόσμου
Wikiversity	15-8-2006	Μαθήματα, υλικό και tests



1.4 ΕΦΑΡΜΟΓΕΣ ΜΑΘΗΣΗΣ & ΔΙΔΑΣΚΑΛΙΑΣ

1.5.1 Εφαρμογές Wiki στη μάθηση και τη διδασκαλία

Πολλές από τις εφαρμογές που περιγράφονται παρακάτω προσεγγίζουν τα παραδείγματα που αναφέρονται από το Stewart Mader του Brown university και ενός άρθρου από τον Bruce Lamb (2004). Υπάρχουν τέσσερις ομάδες εφαρμογών που μπορούμε να θεωρήσουμε:

1. Το Wikis που δημιουργήθηκε μόνο από το προσωπικό (staff).
2. Το Wikis που δημιουργήθηκε αρχικά από τους σπουδαστές.
3. Το Wikis που δημιουργήθηκε και χρησιμοποιείται εξ ίσου από το προσωπικό και τους σπουδαστές.
4. Το Wikis που είναι δημόσια διαθέσιμο και δημιουργήθηκε από άλλους.

1. Το προσωπικό (staff authored)

- Η δημιουργία και η διατήρηση ενός συνόλου που υποστηρίζει την υλικοτεχνική υποδομή με μια ομάδα ακαδημαϊκών (πρόγραμμα σπουδών, βιβλία κειμένων, έγγραφα διαγωνισμάτων κ.λπ.). Η χρήση ενός wiki επιτρέπει στα υλικά να υποστούν εκκαθάριση με την πάροδο του χρόνου παρά να ξαναγράφονται κάθε φορά που ένα νέο μέλος του προσωπικού παραδίδει τη σειρά μαθημάτων (course). (Lamb 2004, Mader 2006, Gibson 2006).
- Δημιουργία και διατήρηση των καταλόγων αναφοράς σειράς μαθημάτων (course reference lists) (Lamb 2004).

2. Ο σπουδαστής (student authored)

- Ομαδικές εργασίες (group assignments)- Το ιστορικό αναθεώρησης

σελίδων (page revision history) επιτρέπει στο δάσκαλο (tutor) να ελέγχει τις συνεισφορές και να παρατηρεί την ανάπτυξη των εργασιών και των προσωπικών συμβολών. Αυτό μπορεί να γίνει σε όλη τη διαδικασία σύνταξης εγγράφου (writing process) έτσι ώστε τα θέματα (issues) να μπορούν να εντοπιστούν και να δοθεί ανατροφοδότηση πριν από την υποβολή της εργασίας (π.χ. ένας αμέτοχος σπουδαστής μπορεί να έρθει σε επαφή, ένα λάθος στην κατεύθυνση διορθώνεται).

- Συζητήσεις ομάδας (group debates)- Οι αντίθετες θέσεις συζητούνται και παρουσιάζονται τα στοιχεία.
- Όμοιες σελίδες αυτοβοήθειας (peer self-help pages)- Ένα wiki κατευθυνόμενο από το σπουδαστή (student directed wiki) αφιερώνεται στους σπουδαστές που βοηθούν άλλους σπουδαστές οι οποίοι αντιμετωπίζουν προβλήματα.
- Ανατροφοδότηση σπουδαστή στο προσωπικό (student feedback to staff)- Ένα wiki επιτρέπει την ανατροφοδότηση για να ελεγχθεί από τους σπουδαστές σε αντιδιαστολή με τη κρυμμένη και κατευθυνόμενη ανατροφοδότηση που συλλέγεται μέσω ερωτηματολογίου (Lamb 2004).
- Θεματικό γλωσσάριο (Subject glossary) - Άτομα/ ομάδες σπουδαστών που έχουν την ευθύνη να δημιουργήσουν ορισμούς για καθορισμένη ορολογία ώστε να κατασκευάσουν ένα διευρυμένο θεματικό γλωσσάριο (extending subject glossary). Η λειτουργία Wiki σημαίνει ότι αυτό το γλωσσάριο μπορεί να υπόκειται σε συνεχή όμοια αναθεώρηση.
- Όμοια αναθεώρηση των εργασιών κατά τη διάρκεια της δημιουργίας τους (peer review of assignments during their creation)- Σπουδαστές προσδιορίζονται να καθοδηγήσουν άλλους σπουδαστές κατά τη διάρκεια της διαδικασίας δημιουργίας των μεμονωμένων εργασιών τους. Οι βαθμοί (marks) μπορούν έπειτα να οριστούν για τις συνεισφορές που κάνουν στους peers τους και καταγράφονται στο ιστορικό αναθεώρησης σελίδων εργασιών wiki (assignment wiki page revision history).
- Μεμονωμένα χαρτοφυλάκια (individual portfolios)- Η ευέλικτη φύση του wiki επιτρέπει σε ένα άτομο να είναι πολύ δημιουργικό στην προσωπική δημιουργία χαρτοφυλακίων του και επιτρέπει επίσης πολύ ευέλικτη διαχείριση χαρτοφυλακίων.

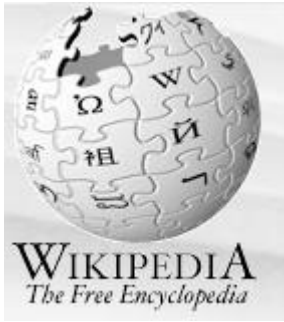
3. Το προσωπικό και ο σπουδαστής (staff and student authored)

- Βαθμολογικό σύστημα για τις εργασίες (marking schemes for assignments) - Ιδανικό για να επιτρέπει τους σπουδαστές να εμπλακούν πραγματικά στον καθορισμό του βαθμολογικού συστήματος από το οποίο οι εργασίες τους θα αξιολογηθούν.
- Θεματικά γλωσσάρια (subject glossaries) - Το προσωπικό μπορεί να προσδιορίσει τους όρους για το συνυπολογισμό (inclusion) και επίσης τις όμοιες προσθήκες αναθεώρησης (peer review additions).
- Συχνές ερωτήσεις (frequently asked questions)- Οι σπουδαστές/ προσωπικό μπορούν να θέσουν ερωτήσεις και το αρμόδιο προσωπικό (ή σπουδαστές) μπορεί να απαντήσει σε αυτές τις ερωτήσεις.
- Δεδομένου ότι ένα φόρουμ συζήτησης (discussion forum) χρησιμοποιεί τις σελίδες συζήτησης (discussion pages) και παρέχει τη δυνατότητα σχολίου.
- Δημιουργία περιπτώσιολογικών μελετών (case studies), αναφορές τομέων (field reports) κ.λπ.
- Γράφοντας μια agenda για συνάντηση (meeting) και δημοσιεύοντας έπειτα και τροποποιώντας τα πρακτικά της συνάντησης (meeting minutes) (από το Lamb 2004).
- Ο Stewart Mader (2006) υποστηρίζει τη χρήση των wikis ως η πιο ευέλικτη, φιλική προς το χρήστη και φτηνότερη αντικατάσταση για ολόκληρα συστήματα διαχείρισης σειράς μαθημάτων (εικονικά μαθησιακά περιβάλλοντα) [(course management systems) (virtual learning environments)].

4. Δημιουργημένο από άλλους (created by others)

- Το υπάρχον Wiki χρησιμοποιείται ως πηγή πληροφοριών (π.χ. Wikipedia).
- Η μελέτη μιας υπάρχουσας σελίδας wiki, το ιστορικό αναθεώρησης της και οι σελίδες συζήτησης ως πρότυπο για το πώς κατασκευάζεται η γνώση.
- Η ανάπτυξη της έρευνας και των δεξιοτήτων σύνταξης εγγράφου μέσω της συμβολής σε ένα υπάρχον wiki. Το Wikipedia ενθαρρύνει ενεργά τις μαθησιακές δραστηριότητες που βασίζονται γύρω από τις συνεισφορές στα άρθρα του. (σχολικά και πανεπιστημιακά προγράμματα Wikipedia).

1.6 WIKIPEDIA



Εικόνα 1: Λογότυπο Wikipedia

Είναι η μεγαλύτερη αυτή τη στιγμή εγκυκλοπαίδεια στο internet, με έναν αριθμό λημμάτων που συναγωνίζονται επάξια αυτόν πολλών "παραδοσιακών" εγκυκλοπαιδειών, όπως της Britannica. Στα πέντε χρόνια λειτουργίας της, η Wikipedia (ή Βικιπαίδεια, κατά την... εμπνευσμένη ελληνική μετάφρασή του ονόματος) κατάφερε να ξεπεράσει τα 1.000.000 άρθρα στην αγγλική γλώσσα.

Ο μεγάλος αριθμός λημμάτων οφείλεται, κατά κύριο λόγο, στον αριθμό των... συγγραφέων της εγκυκλοπαίδειας, που είναι, εν δυνάμει, όλοι οι χρήστες της. Αυτό σημαίνει ότι οποιοσδήποτε το επιθυμεί, μπορεί να διορθώσει ένα άρθρο, να προσθέσει πληροφορίες ή να δημιουργήσει εξ αρχής ένα λήμμα που δεν υπάρχει στην εγκυκλοπαίδεια.

Η δυνατότητα αυτή πολλές φορές δημιουργεί προβλήματα αξιοπιστίας των πληροφοριών και παραποίησης ορισμένων άρθρων, προκειμένου να αντανakλούν τις όχι και τόσο αντικειμενικές απόψεις του εκάστοτε συγγραφέα. Ένα τέτοιο πρόβλημα είχε προκύψει το 2005, όταν κάποιος παραποίησε τη βιογραφία αμερικανού δημοσιογράφου, εμφανίζοντάς τον να σχετίζεται με τη δολοφονία του Kennedy.

Η παραποιημένη σελίδα διορθώθηκε γρήγορα, από τους ειδικούς διορθωτές της Wikipedia, η είδηση όμως έκανε το γύρο του κόσμου. Λίγο καιρό αργότερα, ο "δράστης" ομολόγησε πως απλά έκανε πλάκα σε ένα συνάδελφό του, και πως δε γνώριζε ότι η Wikipedia χρησιμοποιείται ως εργαλείο αναζήτησης πληροφοριών.

Για τον λόγο αυτό υπάρχουν οι λεγόμενοι "διαχειριστές". Πρόκειται για ενεργότατους χρήστες- συγγραφείς της εγκυκλοπαίδειας που πλέον βασική τους αρμοδιότητα είναι η κατά το δυνατόν πιστοποίηση της αλήθειας των γραφομένων και το "καθάρισμα" των κειμένων από δυσφημιστικές ή διαφημιστικές παρεμβάσεις.

Γεγονός είναι ότι τέτοιου είδους προβλήματα μπορούν να εμφανιστούν σε μια ελεύθερη πηγή πληροφόρησης, την οποία ο καθένας μπορεί να τροποποιήσει. Στις περισσότερες, πάντως, περιπτώσεις, οι

διαστρεβλωμένες πληροφορίες εντοπίζονται και διορθώνονται σχετικά γρήγορα.

Η ιδιόμορφη αυτή φύση του μέσου, πέρα από τα προφανή προβλήματα, προσδίδει στη Wikipedia και αρκετά πλεονεκτήματα, όπως η παντελής έλλειψη λογοκρισίας, η ολοκληρωμένη κάλυψη των θεμάτων που προκύπτει από τη συμβολή πολλών χρηστών που προέρχονται από διαφορετικά πολιτιστικά backgrounds στη συγγραφή των θεμάτων, η ανανέωση των θεμάτων σε πραγματικό χρόνο ανάλογα με τις παγκόσμιες εξελίξεις και η έλλειψη copyright –πράγμα που σημαίνει ότι οι χρήστες της εγκυκλοπαίδειας μπορούν να χρησιμοποιήσουν και να αναδημοσιεύσουν το περιεχόμενό της.

Σε αρκετά άρθρα θα βρείτε τη σημείωση "αυτό το άρθρο χρειάζεται συμπλήρωση". Είναι η ευκαιρία σας! Είτε γίνετε πρώτα μέλος της wikipedia είτε και χωρίς αυτή τη δέσμευση μπορείτε να καταθέσετε τις γνώσεις σας στο λήμμα και να πλουτίσετε το διασημότερο ηλεκτρονικό κανάλι ελεύθερης διακίνησης πληροφοριών. Η επιμέλεια και η διασταύρωση των στοιχείων που ο κάθε χρήστης θα προσθέσει είναι θέμα... φιλότιμου.

Όλα τα λήμματα της Wikipedia συνδέονται μεταξύ τους, γεγονός που μπορεί να σας "κολλήσει" για ώρες στην οθόνη του υπολογιστή σας, χωρίς να το καταλάβετε. Τα πολυάριθμα links μεταξύ των άρθρων είναι, επίσης, και ο λόγος για τον οποίο οι περισσότερες λέξεις του κειμένου ενός λήμματος είναι μπλε –χρώμα που αντιπροσωπεύει το σύνδεσμο της συγκεκριμένης λέξης με κάποιο άλλο λήμμα της εγκυκλοπαίδειας ή με τον ορισμό της στο λεξικό.

Η Wikipedia είναι μεταφρασμένη σε 229 συνολικά γλώσσες, συμπεριλαμβανομένων κάποιων που δύσκολα θα φανταζόσασταν, όπως τα Λατινικά, ορισμένες διάλεκτοι των αμερικανών ιθαγενών (Cherokee, Navajo) ή η γλώσσα των Κλίγκον του Star Trek(!). Υπάρχει, επίσης, η "simple English" εκδοχή, για όσους δεν κατέχουν επαρκώς την αγγλική γλώσσα. Η ελληνική έκδοση της εγκυκλοπαίδειας πρόσφατα ξεπέρασε τα 10.000 λήμματα.

Άλλα projects του Wikimedia Foundation, του μη κερδοσκοπικού οργανισμού που ευθύνεται για τη δημιουργία της Wikipedia περιλαμβάνουν λεξικά σε πολλές γλώσσες (Wiktionaries), ειδησεογραφικές σελίδες

(WikiNews), ηλεκτρονικά βιβλία (WikiBooks) και παιδική εγκυκλοπαίδεια (Wikijunior).

Η φιλοσοφία της Wikipedia είναι στην πραγματικότητα αυτή του Internet. Η διαφορά τους είναι ότι η Wikipedia προήλθε από συνειδητή προσπάθεια να κατασκευαστεί μια παγκόσμια και πολυγλωσσική εγκυκλοπαίδεια, ενώ ο παγκόσμιος ιστός είναι περισσότερο αποτέλεσμα της ανάγκης των καιρών.

Ο ιδρυτής της Τζίμι Γουίλς την περιέγραψε ως "προσπάθεια να δημιουργηθεί και να διανεμηθεί μια ελεύθερη εγκυκλοπαίδεια της υψηλότερης πιθανής ποιότητας σε κάθε μεμονωμένο πρόσωπο στον πλανήτη, στη γλώσσα του".

Από το 2001, αναπτύσσεται στον ιστοχώρο wikipedia.org, που χρησιμοποιεί έναν τύπο λογισμικού που ονομάζεται "wiki"- ένας όρος που χρησιμοποιήθηκε αρχικά για το WikiWikiWeb και που προέρχεται από το Χαβάνέζικο Wiki Wiki, το όνομα της γραμμής λεωφορείων που κάνουν το δρομολόγιο προς το διεθνή αερολιμένα της Χονολουλού. Αυτό με τη σειρά του προήλθε από ένα αναδιπλασιασμό της λέξης wiki, που στην τοπική διάλεκτο σημαίνει "γρήγορα".

Ο Γουίλς πιστεύει ότι η wikipedia πρέπει να επιτύχει μία "ίση ή καλύτερη" ποιότητα σε σχέση με τη Britannica και να εκτυπωθεί.

1.7 ΛΟΓΙΣΜΙΚΟ WIKI & WIKI ENGINES

1.7.1 Λογισμικό Wiki

Το λογισμικό Wiki είναι ένας τύπος λογισμικού συνεργασίας που χρησιμοποιείται για την εκτέλεση ενός συστήματος Wiki. Επιτρέπει στις ιστοσελίδες να δημιουργηθούν και να εκδοθούν μέσα από ένα κοινό web browser. Εφαρμόζεται συνήθως ως server-side script που τρέχει σε έναν ή περισσότερους web servers, με το περιεχόμενο να αποθηκεύεται γενικά σε κάποιο σύστημα διαχείρισης βάσεων δεδομένων, αν και μερικές εφαρμογές χρησιμοποιείται αντί για αυτό το σύστημα αρχείων (file system) του server.

Κατηγορίες:

Web-based Wiki

Το πρώτο τέτοιο σύστημα δημιουργήθηκε από τον Ward Cunningham το 1995. Δεδομένης της απλότητας του wiki, ένας μεγάλος αριθμός εφαρμογών που υπάρχουν σήμερα, κυμαίνεται από την πολύ απλή εφαρμογή λειτουργιών πυρήνα (core functionality) μέχρι και τα πιο σύνθετα «συστήματα διαχείρισης περιεχομένου». Η αρχική διαφορά μεταξύ των wikis και των πιο σύνθετων συστημάτων διαχείρισης περιεχομένου (content management systems) είναι ότι το λογισμικό wiki τείνει να επικεντρωθεί στο περιεχόμενο, εις βάρος του ισχυρότερου ελέγχου στο layout που έχουν λογισμικά CMS όπως Drupal, WebGUI, ή εις βάρος των μη-wiki χαρακτηριστικών (άρθρα, blogs.) όπως σε Tikiwiki.

Το λογισμικό Wiki θα μπορούσαμε να πούμε ότι περικλείει όλο το λογισμικό που απαιτείται για να τρέξει ένα wiki, το οποίο περιλαμβάνει έναν web server όπως Apache, σε αντίθεση με τη «μηχανή Wiki» (Wiki machine), η οποία εφαρμόζει την τεχνολογία wiki (wiki technology). Σε μερικές περιπτώσεις, όπως το ProjectForum, ή κάποιους WikiServers, ο web server και η μηχανή wiki συσσωρεύονται μαζί ως ένα self-contained σύστημα, το οποίο μπορεί συχνά να τους καταστήσει ευκολότερους να εγκαταστήθουν.

Η πλειοψηφία των μηχανών είναι open source, συχνά διαθέσιμη με GNU General Public License (GPL). Τα μεγάλα προγράμματα όπως το TWiki, η Wikipedia μηχανή, το MediaWiki, αναπτύσσονται σε συνεργασία (collaboratively). Πολλά wikis είναι ιδιαίτερα μορφοματικά (modular), παρέχοντας APIs που επιτρέπουν στους προγραμματιστές να αναπτύξουν νέα χαρακτηριστικά χωρίς να απαιτείται αυτά να είναι συγγενικά (familiar) με ολόκληρη τη βάση κώδικα (codebase).

Είναι δύσκολο να καθοριστεί ποιες μηχανές wiki είναι οι δημοφιλέστερες, αν και ένας κατάλογος υποψηφίων περιλαμβάνει τα: TWiki, MoinMoin, PmWiki, DokuWiki και MediaWiki (Google trend history comparison).

Personal Wiki

Κάποιο λογισμικό wiki δεν προορίζεται για τη συλλογική εργασία (collaborative work), αλλά είτε για διαχείριση περιεχομένου είτε για την προσωπική οργάνωση πληροφοριών (Personal information organizing). Αυτό καλείται μερικές φορές Desktop Wiki ή Personal (προσωπικό) Wiki. Ένα παράδειγμα αυτού είναι το Tiddlywiki.

Peer-to-Peer Wiki

Μία ιστοσελίδα wiki (wikisite) μπορεί επίσης να μοιραστεί μεταξύ των peers χρησιμοποιώντας ένα P2P λογισμικό. Ένα τέτοιο P2P wiki σύστημα είναι ενσωματωμένο στο Code Co-op.

1.8 ΣΥΓΚΡΙΣΗ ΤΩΝ ΚΑΛΥΤΕΡΩΝ ΜΗΧΑΝΩΝ WIKI

TikiWiki, MediaWiki ή DokuWiki

Οι μηχανές Wiki είναι ένα μεγάλο κομμάτι του λογισμικού - επιτρέπουν στο χρήστη να χτίσει ένα διαδραστικό (interactive) ιστοχώρο ή μια βάση γνώσεων (knowledge base) όπου κάθε μέλος μπορεί να προσθέσει και να εκδώσει το περιεχόμενο. Τα εργαλεία Wiki μπορούν να χρησιμοποιηθούν για τα προσωπικά ή συλλογικά web συστήματα. Η δύναμη του λογισμικού wiki είναι στην απλότητά της χρήσης του. Η ταχυδρόμηση ενός εγγράφου με Wiki είναι μία εύκολη δουλειά, που ο καθένας μπορεί να κάνει.

Η δημιουργία και η διατήρηση ενός wiki site δεν είναι ιδιαίτερος δύσκολος και απαιτούν μόνο βασικές ικανότητες προγραμματισμού. Πιθανώς το καλύτερο μέρος ενός wiki project είναι ότι μπορείτε να κατεβάσετε ένα έτοιμο εργαλείο και να το χρησιμοποιήσετε για να δημιουργήσετε έναν ιστοχώρο εντελώς δωρεάν!

Κύρια πλεονεκτήματα Wiki:

Υπάρχουν πολλές μηχανές wiki που είναι δωρεάν για κατέβασμα και χρήση. -Το λογισμικό Wiki μπορεί να χρησιμοποιηθεί από τις εταιρίες ή και από μεμονωμένα άτομα. -Η εγκατάσταση ενός λογισμικού wiki είναι πολύ εύκολη. -Το Wiki δημιουργεί έναν συνεργάσιμο ιστοχώρο όπου κάθε μέλος

μπορεί να προσθέσει και να εκδώσει το περιεχόμενο. -Το Wikis είναι όπως τα φόρουμ όπου τα posts μπορούν να εκδοθούν. -Η έκδοση ενός άρθρου σε ένα wiki site είναι εύκολη ακόμη και για τους μη έμπειρους χρήστες του διαδικτύου.

Μόλις αποφασίσετε να χρησιμοποιήσετε ένα από τα wikis, έρχεται το δίλημμα: Ποιο εργαλείο wiki να επιλέξετε για τον ιστοχώρο σας; Σε αυτό το άρθρο θα συγκρίνουμε τρία από τα καλύτερα δωρεάν εργαλεία wiki: το MediaWiki, το TikiWiki και το DokuWiki. Αλλά καταρχήν, πρέπει να εστιάσουμε σε μερικά από τα βασικά κριτήρια για τη σύγκριση.

Κριτήρια επιλογής μιας μηχανής wiki

Όταν πρόκειται να επιλέξετε μία μηχανή wiki, εξέτασε τα εξής: Ποιος είναι ο δημιουργός του εργαλείου wiki, Είναι ένα μεμονωμένο πρόσωπο ή μια ομάδα; Είναι το wiki ένα δωρεάν λογισμικό και open source ή πρέπει να αγοράσετε μια άδεια για αυτό; Ποιος χρησιμοποιεί αυτό το εργαλείο; Μια καλή μηχανή wiki είναι πιθανό να έχει μια μεγάλη ομάδα υπαρχόντων χρηστών. Αυτό είναι χρήσιμο στην περίπτωση όπου χρειασθείτε υποστήριξη ή έχετε ερωτήσεις για τη λειτουργία του.



MediaWiki - το λογισμικό πίσω από Wikipedia!

Το MediaWiki είναι το δημοφιλέστερο λογισμικό wiki στο internet. Χρησιμοποιείται για να τρέξει Wikipedia (τη μεγαλύτερη on line εγκυκλοπαίδεια του κόσμου) και άλλα projects του μη κερδοσκοπικού

ιδρύματος Wikimedia, καθώς επίσης και πολλά άλλα wikis. Το MediaWiki είναι ένα ελεύθερο πακέτο λογισμικού που χορηγούν άδεια με το GNU General Public License.

Το MediaWiki περιλαμβάνει τα χαρακτηριστικά γνωρίσματα όπως:

- Εύκολο ανέβασμα (upload) αρχείων
- Υποστηρίζει πολλές γλώσσες
- Επιλογές αναζήτησης
- Δείκτης των στοιχείων περιεχομένων (content items)
- Συζητήσεις
- και πολλά άλλα.



TikiWiki

Το TikiWiki είναι ένα ισχυρό, open source Groupware και σύστημα διαχείρισης περιεχομένου, γραμμένο σε PHP. Το TikiWiki μπορεί να χρησιμοποιηθεί για να δημιουργήσει web sites και intranets. Το TikiWiki προσφέρει μεγάλους πόρους εάν χρησιμοποιηθεί ως εργαλείο συνεργασίας. Το TikiWiki παρέχει όλα τα χαρακτηριστικά γνωρίσματα που αναμένετε από τα παρόντα συστήματα wiki. Μπορείτε να χρησιμοποιήσετε το TikiWiki για φόρουμ, chatrooms, τη λήψη ψηφοφορίας (poll taking), blog, gallery αρχείων και εικόνας, FAQ, ημερολόγιο και ακόμη περισσότερα.

DokuWiki

Το DokuWiki είναι ένα εύχρηστο σύστημα wiki. Το Doku-Wiki θα ήταν η καλύτερη επιλογή σας εάν πρέπει να γράψετε μια τεκμηρίωση (documentation) για ένα λογισμικό, επειδή διευκολύνει τη δημιουργία του

δομημένου περιεχομένου (structured content). Έχει μια ισχυρή σύνταξη και τα αρχεία δεδομένων του μπορούν να διαβαστούν και έξω από το Wiki. Αυτό βοηθάει τις ομάδες των developers που δουλεύουν ένα project να αλληλεπιδράσουν πολύ ευκολότερα. Όλο τα δεδομένα αποθηκεύονται στα plain text αρχεία και έτσι δεν απαιτείται καμία βάση δεδομένων. Το DokuWiki (καθώς επίσης και το MediaWiki και το TikiWiki) γράφονται σε PHP. Μερικά από τα χαρακτηριστικά γνωρίσματα του DokuWiki είναι:

- Ανέβασμα και ενσωμάτωση εικόνων
- Πολυγλωσσική υποστήριξη
- Συνδέσεις Interwiki
- και άλλα

1.9 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ WIKI

Ένα wiki επιτρέπει τη συλλογική συγγραφή κειμένων σε μια πολύ απλή γλώσσα σήμανσης χρησιμοποιώντας έναν φυλλομετρητή Ιστού. Μια σελίδα μόνη της σε ένα wiki αναφέρεται ως "σελίδα wiki", ενώ ολόκληρο το σώμα των σελίδων, οι οποίες είναι συνήθως ιδιαίτερα πυκνά διασυνδεδεμένες με υπερσυνδέσμους λέγεται "το wiki". Στην πραγματικότητα, ένα wiki είναι μια πολύ απλή, εύχρηστη διατηρούμενη από το χρήστη βάση δεδομένων για αναζήτηση πληροφορίας.

Ένα καθοριστικό χαρακτηριστικό της τεχνολογίας wiki είναι η ευκολία με την οποία οι σελίδες μπορούν να δημιουργηθούν και να ενημερωθούν. Γενικά, οι τροποποιήσεις γίνονται αποδεκτές χωρίς κανένα προηγούμενο έλεγχο. Τα περισσότερα wikis είναι ανοικτά στο ευρύ κοινό χωρίς την ανάγκη να καταχωρηθεί οποιοσδήποτε λογαριασμός χρήστη. Μερικές φορές χρειάζεται μια διαδικασία εισαγωγής προκειμένου για να ενεργοποιηθεί ένα cookie "wiki υπογραφής" έτσι ώστε να είναι ενυπόγραφες οι τροποποιήσεις. Οι ιδιωτικοί εξυπηρετητές wiki απαιτούν την ταυτοποίηση των χρηστών. Βέβαια, πολλές επεξεργασίες του κειμένου μπορούν να γίνουν σε πραγματικό χρόνο, και να εμφανιστούν σχεδόν στιγμιαία σε απευθείας σύνδεση. Αυτό μπορεί συχνά να οδηγήσει στην κατάχρηση του συστήματος.

1.10 ΣΥΓΓΡΑΦΗ ΣΤΟ WIKI

1.10.1 Αναπαραστάσεις Σελίδας

Σε ένα παραδοσιακό wiki, κάθε σελίδα έχει τρεις αναπαραστάσεις:

- Ο "πηγαίος κώδικας" (που μπορεί να τροποποιηθεί), ο οποίος αποτελεί τη μορφή με την οποία αποθηκεύεται τοπικά στον εξυπηρετητή. Είναι συνήθως σκέτο κείμενο, που γίνεται ορατό στο χρήστη μόνο όταν η λειτουργία επεξεργασίας το παρουσιάζει σε μια φόρμα φυλλομετρητή.
- Ένα σχεδιαστικό πρότυπο (που ενδεχόμενα παράγεται εσωτερικά) και καθορίζει τη διάταξη και τα κοινά για όλες τις σελίδες στοιχεία.
- Ο αποδιδόμενος κώδικας HTML που παράγεται από τον εξυπηρετητή επί τόπου από το κείμενο πηγή όταν ζητείται μια ιδιαίτερη σελίδα.

Η πηγαία μορφή, γνωστή μερικές φορές σαν «κείμενο wiki» βελτιώνεται με μια απλουστευμένη γλώσσα σήμανσης ώστε να δείξει τις διάφορες δομικές και οπτικές συμβάσεις. Ένα συχνά χρησιμοποιημένο παράδειγμα μιας τέτοιας σύμβασης είναι να αρχίσει μια γραμμή κειμένου με αστερίσκο ("*") προκειμένου να το χαρακτηρίσει ως στοιχείο σε ένα κατάλογο με κουκίδες. Το στίλ και η σύνταξη μπορούν να ποικίλουν πολύ μεταξύ των εφαρμογών, μερικές από τις οποίες επιτρέπουν επίσης τις ετικέτες HTML.

Ο συλλογισμός πίσω από αυτό το σχέδιο είναι ότι το HTML, με τις πολλές αινιγματικές ετικέτες του, δεν είναι κυρίως κατανοήσιμο από άνθρωπο. Το να γίνει ορατός στο χρήστη ο πηγαίος κώδικας HTML κάνει το κείμενο εξαιρετικά δύσκολο στο διάβασμα και την επεξεργασία του για τους περισσότερους χρήστες. Είναι επομένως καλύτερα να προωθείται η επεξεργασία σε μορφή κειμένου με μερικές απλές συμβάσεις για τη δομή και το στίλ.

Είναι επίσης κάπως ευεργετικό το ότι οι χρήστες δεν μπορούν άμεσα να χρησιμοποιήσουν όλη τη λειτουργικότητα που επιτρέπει το HTML, όπως

JavaScript. Έτσι επιτυγχάνεται ομοιομορφία στην εμφάνιση και την αίσθηση, μαζί με κάποια πρόσθετη ασφάλεια για το χρήστη. Σε πολλές εφαρμογές wiki, ένας ενεργός υπερασύνδεσμος είναι ακριβώς όπως παρουσιάζεται.

Η χρήση του κειμένου wiki (wikitext), και όχι άμεσα του HTML, είναι σημαντική και για λόγους ασφάλειας. Επιτρέποντας στους χρήστες να γράφουν σε HTML θα επέτρεπε πιθανά επιβλαβή ή ενοχλητικό κώδικα (πχ κώδικα JavaScript που δεν επιτρέπει στον αναγνώστη να μαρκάρει μέρος του κειμένου).

Παράδειγμα με ένα απόσπασμα κειμένου από το βιβλίο Foundation του

Ισαάκ Ασίμωφ:

κείμενο σε σύνταξη για μηχανή media wiki:

'''γιατρός''; Κανένας άλλος τίτλος; Ένας '''ακαδημαϊκός''';
Και μετρά παραπάνω από την πολιτική αρχή;"
"Γιατί, βεβαίως," απάντησε ο Χαρντίν, φιλικά. "Είμαστε όλοι
ακαδημαϊκοί λίγο ως πολύ. Τελικά, δεν είμαστε τόσο πολύ ένας
κόσμος όσο ένα επιστημονικό ίδρυμα —κάτω από τον άμεσο
έλεγχο του αυτοκράτορα."

κώδικας σε HTML:

```
<p>  
&quot;<em>Doctor</em>; Κανένας άλλος τίτλος; Ένας <  
em>ακαδημαϊκός</em >; Και μετρά παραπάνω από την πολιτική  
αρχή;&quot;;  
</p>  
<p>  
&quot;Γιατί, βεβαίως, &quot;; Απάντησε ο Χαρντίν, φιλικά.  
&quot;Είμαστε όλοι ακαδημαϊκοί λίγο ως πολύ. Τελικά, δεν  
είμαστε τόσο πολύ ένας κόσμος όσο ένα επιστημονικό  
ίδρυμα&mdash;κάτω από τον άμεσο έλεγχο του αυτοκράτορα.&quot;  
</p>
```

παραγόμενο αποτέλεσμα:

"Γιατρός; Κανένας άλλος τίτλος; Ένας *ακαδημαϊκός*; Και μετρά παραπάνω από την πολιτική αρχή;"

"Γιατί, βεβαίως," απάντησε ο Χαρντίν, φιλικά. "Είμαστε όλοι ακαδημαϊκοί λίγο ως πολύ. Τελικά, δεν είμαστε τόσο πολύ ένας κόσμος όσο ένα επιστημονικό ίδρυμα —κάτω από τον άμεσο έλεγχο του αυτοκράτορα."

Μερικές νέες μηχανές wiki χρησιμοποιούν μια διαφορετική μέθοδο: επιτρέπουν WYSIWYG (What You See Is What You Get) που εκδίδει, συνήθως με τη βοήθεια JavaScript ή ενός αντικειμένου ελέγχου ActiveX που μεταφράζει γραφικά τις εισηγμένες οδηγίες μορφοποίησης, όπως το «έντονα» και «πλάγια» στις αντίστοιχες ετικέτες HTML. Στις εφαρμογές αυτές, η σήμανση μια έκδοσης HTML που έχει πρόσφατα υποστεί επεξεργασία παράγεται και υποβάλλεται στον εξυπηρετητή με διαφάνεια, και ο χρήστης προστατεύεται από αυτήν την τεχνική λεπτομέρεια. Οι χρήστες που δεν έχουν το απαραίτητο plugin μπορούν γενικά να εκδώσουν τη σελίδα, συνήθως με άμεση έκδοση του ακατέργαστου κώδικα HTML. Πιο πρόσφατα, οι μηχανές wiki παράγουν σύνταξη wiki αντί HTML. Κατ' αυτό τον τρόπο, χρήστες που αισθάνονται άνετα με τη σύνταξη wiki μπορούν να συνεχίσουν.

1.10.2 Πρότυπα

Ενώ για χρόνια το de facto πρότυπο ήταν η σύνταξη του αρχικού WikiWikiWeb, αυτήν την περίοδο οι οδηγίες μορφοποίησης ποικίλλουν ανάλογα με τη μηχανή wiki. Τα απλά wikis επιτρέπουν μόνο βασική μορφοποίηση κειμένων, ενώ τα πιο σύνθετα υποστηρίζουν πίνακες, εικόνες, τύπους, ή ακόμα και αλληλεπιδραστικά στοιχεία όπως ψηφοφορίες και παιχνίδια. Πολλοί άνθρωποι μεταπηδούν μεταξύ των μηχανών wiki, από τη μια στην άλλη. Λόγω της δυσκολίας χρησιμοποίησης διαφορετικών συντάξεων, πολλοί άνθρωποι καταβάλλουν σημαντική προσπάθεια για τον προσδιορισμό ενός πρότυπου σήμανσης wiki (βλ. τις προσπάθειες Meatball και TikiWiki).

1.10.3 Σύνδεση και δημιουργία των σελίδων

Το Wikis είναι ένα αληθινό μέσο υπερκειμένων, με δομές μη γραμμικής πλοήγησης. Κάθε σελίδα περιέχει χαρακτηριστικά έναν μεγάλο αριθμό συνδέσεων με άλλες σελίδες. Στα μεγαλύτερα wikis υπάρχουν συχνά ιεραρχικές σελίδες πλοήγησης, συνέπεια συχνά της αρχικής διαδικασίας δημιουργίας σελίδας, αλλά δεν είναι απαραίτητο να χρησιμοποιηθούν. Οι συνδέσεις δημιουργούνται χρησιμοποιώντας μια συγκεκριμένη σύνταξη, το αποκαλούμενο «σχέδιο συνδέσεων».

Αρχικά, τα περισσότερα wikis χρησιμοποιούσαν Camel Case ως σχέδιο συνδέσμου που παράγεται από τις λέξεις μιας φράσης αν ξεκινάνε με κεφαλαίο γράμμα και αφαιρεθούν τα μεταξύ τους διαστήματα (η λέξη "CamelCase" είναι ο ίδιος ένα παράδειγμα CamelCase). Ενώ το CamelCase καθιστά τη σύνδεση πολύ εύκολη, οδηγεί επίσης σε συνδέσμους γραμμένους σε μια μορφή που παρεκκλίνει από την τυποποιημένη ορθογραφία. Τα βασισμένα σε CamelCase wikis είναι αμέσως αναγνωρίσιμα επειδή έχουν πολλές συνδέσεις με τα ονόματα όπως "TableOfContents" and "BeginnerQuestions".

1.10.4 Αναζήτηση

Τα περισσότερα wikis προσφέρουν τουλάχιστον αναζήτηση τίτλου και μερικές φορές αναζήτηση πλήρους κειμένου. Η εξέλιξη της αναζήτησης εξαρτάται από το εάν η μηχανή wiki χρησιμοποιεί μια βάση δεδομένων. Η πρόσβαση σε βάση δεδομένων είναι απαραίτητη για αναζητήσεις υψηλής ταχύτητας στα μεγάλα wikis. Στη Wikipedia μια εγκυκλοπαίδεια βασισμένη στα wiki, το αποκαλούμενο «Go πλήκτρο» επιτρέπει στους αναγνώστες να δουν μια σελίδα που ταιριάζει με τα κριτήρια αναζήτησης όσο το δυνατόν περισσότερο. Η μηχανή αναζήτησης MetaWiki δημιουργήθηκε για να επιτρέψει τις αναζητήσεις σε περισσότερα wikis. Η αναζήτηση βασίζεται σε λέξεις-κλειδιά..

1.10.5 Wiki από την πλευρά του εξυπηρετητή είτε από την πλευρά του πελάτη

Σε μεγάλο βαθμό, τα πιο κοινά συστήματα wiki είναι από την πλευρά του εξυπηρετητή (η Wikipedia είναι ένα wiki από την πλευρά του εξυπηρετητή). Στην ουσία, οι λειτουργίες επεξεργασίας, παρουσίασης και ελέγχου παρέχονται από τον εξυπηρετητή μέσω της μηχανής wiki που δίνει το περιεχόμενο σε μια σελίδα βασισμένη σε HTML για παρουσίαση σε έναν φυλλομετρητή Ιστού.

Ένα σύστημα wiki από την πλευρά του client απαιτεί μόνο τον εξυπηρετητή για "να εξυπηρετήσει" τα αρχεία wiki με τον ίδιο σχεδόν τρόπο όπως ένας κεντρικός υπολογιστής δικτύου επιτρέπει στα αρχεία HTML να ανακτώνται χρησιμοποιώντας το HTTP.

Τα συστήματα wiki από την πλευρά του client μπορούν να είναι ελάχιστα περισσότερο από έναν κώδικα plugin σε έναν παραδοσιακό φυλλομετρητή Ιστού.

1.10.6 Ελέγχοντας Αλλαγές

Οι εκθέσεις σύγκρισης ιστορικού δίνουν έμφαση στις αλλαγές μεταξύ δύο αναθεωρήσεων μιας σελίδας.

Τα Wiki γενικά σχεδιάζονται με τη φιλοσοφία μάλλον να γίνει εύκολη η διόρθωση των λαθών, παρά να καταστήσει δύσκολο το να γίνουν λάθη. Κατά συνέπεια ενώ τα wiki είναι πολύ ανοικτά, παρέχουν τα μέσα για να ελεγχθεί η εγκυρότητα των πρόσφατων προσθηκών στο σώμα των σελίδων. Η πιο προεξέχουσα, σχεδόν σε κάθε wiki, είναι η σελίδα "πρόσφατων αλλαγών", όπου ένας συγκεκριμένος κατάλογος απαριθμεί τις πρόσφατες τροποποιήσεις που έγιναν μέσα σε ένα δεδομένο χρονικό διάστημα. Μερικά wikis μπορούν να φιλτράρουν τον κατάλογο προκειμένου να απομακρύνουν ασήμαντες τροποποιήσεις και τροποποιήσεις από αυτόματη εισαγωγή scripts ("bots").

Από το κατάστιχο αλλαγών, διάφορες λειτουργίες είναι προσιτές στα περισσότερα wikis: η ιστορία αναθεώρησης που παρουσιάζει προηγούμενες εκδόσεις σελίδων και το χαρακτηριστικό γνώρισμα diff, που δίνει έμφαση στις αλλαγές μεταξύ δύο αναθεωρήσεων. Χρησιμοποιώντας την ιστορία αναθεώρησης, ένας συντάκτης μπορεί να δει και να αποκαταστήσει μια

προηγούμενη έκδοση του άρθρου. Το χαρακτηριστικό γνώρισμα diff μπορεί να χρησιμοποιηθεί για να αποφασίσει εάν αυτό είναι ή όχι απαραίτητο. Ένας κανονικός χρήστης wiki μπορεί να δει το diff μιας τροποποίησης στη σελίδα "πρόσφατων αλλαγών" και, εάν είναι μια απαράδεκτη τροποποίηση, συμβουλεύεται το ιστορικό και αποκαθιστά μια προηγούμενη αναθεώρηση. Η διαδικασία αυτή βελτιώνεται λίγο πολύ, ανάλογα με το χρησιμοποιούμενο λογισμικό wiki.

Σε περίπτωση που απαράδεκτες τροποποιήσεις έχουν παραληφθεί στη σελίδα "πρόσφατων αλλαγών", μερικές μηχανές wiki παρέχουν έναν πρόσθετο έλεγχο περιεχομένου. Μια σελίδα ή ένα σύνολο σελίδων μπορεί να παρακολουθούνται προκειμένου να εξασφαλιστεί ότι διατηρεί την ποιότητά της. Ένα πρόσωπο πρόθυμο να διατηρήσει τις σελίδες θα ειδοποιηθεί για τις τροποποιήσεις στις σελίδες, επιτρέποντάς του να επαληθεύσει γρήγορα αν ισχύουν οι νέες τροποποιήσεις.

1.11 WIKI ΚΟΙΝΟΤΗΤΕΣ

Πολλές wiki κοινότητες είναι ιδιωτικές ιδιαίτερα μέσα στις επιχειρήσεις σαν συλλογικό λογισμικό. Χρησιμοποιούνται συχνά για εσωτερική τεκμηρίωση για εσωτερικά συστήματα και εφαρμογές. Η δημοκρατική, περιλαμβάνουσα τα πάντα φύση της Wikipedia είναι ένας σημαντικός παράγοντας στην ανάπτυξη της, ενώ πολλά άλλα wikis είναι υψηλά ειδικευμένα. Η Wikipedia έχει σχεδιαστεί έτσι ώστε να φροντίζει για όλες τις ομάδες και είναι προσηλωμένη στην παροχή μετριοπαθών γνώμων και άρθρων με μοναδικό στόχο την ενημέρωση.

1.12 ΣΥΓΚΡΙΣΗ ΤΩΝ ΣΕΛΙΔΩΝ WIKI ΜΕ ΤΑ BLOGS

Ποια είναι η διαφορά ανάμεσα σε ένα wiki και ένα blog;

Σε ένα *blog* ή *web log* η συγγραφή και το πολυμεσικό περιεχόμενο της ιστοσελίδας έχουν τη μορφή των "*post*" είναι δηλαδή καταχωρίσεις που αποτελούν σημεία έναρξης. Οι απαντήσεις έχουν τη μορφή σχολίων και λέγονται "*comments*".

Ενώ η δυνατότητα εισαγωγής *comments*, ακόμα και *posts*, είναι ανοιχτή στα μέλη στα μέλη του blog ή στο ευρύ κοινό, κανένας δεν μπορεί να αλλάξει το *comment* ή το *post* το οποίο αναρτήθηκε από κάποιον άλλον. Η συνηθισμένη διάταξη είναι *post* – *comment* – *comment* – *comment*, κ.ο.κ. Για αυτόν το λόγο, τα blogs συνήθως είναι το μέσο το οποίο επιλέγεται για την έκφραση προσωπικών απόψεων.

Ένα wiki έχει πολύ πιο ανοιχτή δομή και επιτρέπει σε άλλους να τροποποιήσουν το περιεχόμενο του κειμένου που κάποιος έχει γράψει. Αυτή η ελευθερία μπορεί να προκαλέσει την υπερίσχυση της γνώμης της πλειοψηφίας μιας ομάδας έναντι της προσωπικής άποψης.

1.13 ΣΥΓΚΡΙΣΗ WIKI SITES & ΚΟΙΝΩΝ WEBSITES

Πίνακας 2: Wiki vs Ιστοσελίδα

Wiki	Standard ιστοσελίδες
Ελεύθερη-ανοιχτή σύνταξη	Περιορισμένη σύνταξη
Γλώσσα απλής μορφής κειμένου	Standard HTML, Javascript, PHP, κλπ
Online αποθήκευση προηγούμενων εκδόσεων	Μη αποθήκευση προηγούμενων εκδόσεων online
Εύκολη δημιουργία νέων σελίδων	Δυσκολότερη δημιουργία νέων σελίδων
Χαμηλότερος βαθμός ασφάλειας	Υψηλότερος βαθμός ασφάλειας
Ύπαρξη πολλαπλών συγγραφέων	Περιορισμένος αριθμός συγγραφέων
Κοινοτικό, Συνεργατικό	Προσωπικό
Διαρκής εξέλιξη	Θεωρητικά ολοκληρωμένο

1.14 ΥΠΕΡ ΚΑΙ ΚΑΤΑ ΤΗΣ ΧΡΗΣΗΣ ΤΩΝ WIKIS

Πλεονεκτήματα:

- Καθένας μπορεί να συμμετέχει στη σύνταξη-διόρθωση του περιεχομένου
- Ευκολία στην εκμάθηση και στη χρήση
- Εξοικονόμηση χρόνου στην έκδοση και την ανανέωση του περιεχομένου
- Δυνατότητα συνεργασίας στο ίδιο έργο ανθρώπων που βρίσκονται σε διαφορετικά μέρη του πλανήτη
- Το λογισμικό κρατάει στοιχεία για κάθε τροποποίηση που γίνεται και η επαναφορά μιας προηγούμενης έκδοσης κάποιου άρθρου αποτελεί μια απλή διαδικασία
- Ευρύνει την πρόσβαση στις διαδικτυακές εκδόσεις και για χρήστες με περιορισμένες τεχνολογικές γνώσεις
- Το wiki δεν έχει προκαθορισμένη δομή – συνεπώς είναι ένα ευέλικτο εργαλείο που μπορεί να χρησιμοποιηθεί για μεγάλο εύρος εφαρμογών
- Υπάρχει μεγάλη ποικιλία από ανοικτού κώδικα λογισμικό wiki, συνεπώς το κόστος του δικαιώματος χρήσης δεν πρέπει να αποτελεί εμπόδιο στην εγκατάσταση ενός wiki από ένα ίδρυμα

Μειονεκτήματα:

Τα χαρακτηριστικά που υπό κάποιες συνθήκες θεωρούνται πλεονεκτήματα, υπό άλλες συνθήκες μπορούν να αποτελέσουν μειονεκτήματα.

- Ο καθένας μπορεί να τροποποιήσει το περιεχόμενο και μπορεί αυτό να αποτελεί υπερβολική ελευθερία για κάποιες εφαρμογές, για παράδειγμα όταν πρόκειται για εμπιστευτικά έγγραφα. Ωστόσο υπάρχει η δυνατότητα ρύθμισης της πρόσβασης που έχουν οι χρήστες στο περιεχόμενο.
- Το wiki είναι ευάλωτο στο spam και τον βανδαλισμό αν δεν γίνουν οι απαραίτητες ρυθμίσεις και χειρισμοί προστασίας. Όμως υπάρχουν εύκολοι τρόποι επαναφοράς μιας σελίδας. Επίσης, ο περιορισμός της πρόσβασης μόνο σε εγγεγραμμένους στη σελίδα χρήστες

ελαχιστοποιεί τον βανδαλισμό που προκύπτει από προγράμματα που παράγουν αυτόματα spam (spam-bots).

- Είναι απαραίτητη η σύνδεση στο internet για την προαγωγή της συνεργασίας, αλλά σημειώνονται εξελίξεις στις τεχνολογίες παραγωγής εκτυπώσιμων εκδόσεων των άρθρων.
- Η ευελιξία της δομής ενός wiki μπορεί να σημαίνει ότι η πληροφορία γίνεται ανοργάνωτη. Όσο μεγαλώνει το μέγεθος ενός wiki, η κοινότητα σχεδιάζει και διαχειρίζεται τη δομή του συλλογικά.

1.14.1 Σχετικά με την ποιότητα του περιεχομένου

Υπάρχουν αντικρουόμενες απόψεις και μία υγιής διαμάχη μεταξύ των εκπαιδευτικών που αφορά την ποιότητα του περιεχομένου που αναπτύσσεται σε ένα wiki περιβάλλον. Δεδομένης της πρόσφατης εμφάνισης του φαινομένου του wiki, η έρευνα σχετικά με αυτό το θέμα είναι περιορισμένη.

Για παράδειγμα, εφόσον ο καθένας μπορεί να τροποποιήσει τις εγγραφές της wikipedia, πώς μπορούν οι χρήστες να ξέρουν ότι η wikipedia είναι εξίσου ακριβής με άλλες καθιερωμένες πηγές όπως π.χ. η εγκυκλοπαίδεια Britannica; Αν ένας πυρηνικός φυσικός γράψει ένα άρθρο σχετικό με το αντικείμενο του στην wikipedia θα έχει το ίδιο κύρος με κάποιο συνεργάτη του χωρίς ακαδημαϊκό υπόβαθρο. Οι διαφωνίες σχετικά με το περιεχόμενο συνήθως λύνονται με συζήτηση μεταξύ των χρηστών.

1.15 ΚΡΙΤΗΡΙΑ ΕΠΙΛΟΓΗΣ ΤΟΥ ΣΩΣΤΟΥ WIKI

Προτού επιλέξουμε το εργαλείο Wiki για μια συγκεκριμένη εργασία, πρέπει να μελετήσουμε ορισμένα χαρακτηριστικά του, που θα διευκολύνουν τη χρήση του στη συνέχεια. Μερικά από αυτά είναι τα επόμενα:

- Ευκολία Χρήσης

Η ιδανική περίπτωση είναι όταν ο χρήστης μπορεί να μάθει να χρησιμοποιεί το wiki εργαλείο μέσα σε μερικά λεπτά, ακόμα και όταν

πρόκειται για μαθητές μικρής ηλικίας, εφόσον μπορούν να εντοπίσουν τα γράμματα σε ένα πληκτρολόγιο.

- Εμφάνιση

Είναι επιθυμητή η δυνατότητα εύκολης αναζήτησης μέσα στο wiki και πιθανώς η προσαρμογή της συνολικής εμφάνισης του wiki. Επίσης μπορεί ο χρήστης να μην θέλει να περιέχονται διαφημίσεις μέσα στο wiki του.

- Κόστος

Υπάρχει πλήθος από ελεύθερα-δωρεάν wiki εργαλεία στο διαδίκτυο. Τα περισσότερα χρηματοδοτούνται βάζοντας τις διαφημίσεις που εμφανίζονται στο wiki σας. Αυτό μπορεί να αποδειχθεί πολύ αποπροσανατολιστικό ή ακόμα και ανάρμοστο για μαθητές που μπορούν αναπόφευκτα να πάνε στα προκλητικά links. Και φυσικά πάντα τυχαίνει κάποιο link να οδηγήσει σε κάτι φρικτό τη στιγμή ακριβώς που κάποιος γονέας ή προϊστάμενος εμφανιστεί ξαφνικά για επίσκεψη ;-)

- Ασφάλεια

Δυνατότητα προστασίας συγκεκριμένων σελίδων από αλλαγές. Δυνατότητα ρύθμισης του wiki για ιδιωτική χρήση (μόνο τα μέλη του wiki να έχουν πρόσβαση και δυνατότητα τροποποίησης του wiki). Δυνατότητα ρύθμισης του wiki για προστατευμένη χρήση (ανοιχτό στο κοινό για ανάγνωση αλλά μόνο τα μέλη του wiki έχουν δυνατότητα τροποποίησης του).

Σχετικά με τα πνευματικά δικαιώματα:

Τα wiki μπορούν να περιλαμβάνουν κείμενο, εικόνες, και αρχεία ήχου ή βίντεο. Τα περισσότερα wiki υπόκεινται σε μία συγκεκριμένη συμφωνία πνευματικών δικαιωμάτων που αποκαλείται Creative Commons. Αυτό ουσιαστικά σημαίνει ότι όλο το υλικό που τοποθετούν οι χρήστες στο wiki μπορεί να χρησιμοποιηθεί από άλλους σύμφωνα με έναν διακανονισμό «κοινής χρήσης» (“share and share alike” arrangement). Προσοχή! Όλες οι αναφορές σε υλικό που εμπίπτει στα Creative Commons πρέπει να παρατίθενται καταλλήλως. Επίσης, είναι απαραίτητη η επιβεβαίωση ότι οποιοδήποτε υλικό από άλλες πηγές χρησιμοποιείται να συμφωνεί και αυτό με την αρχή Creative Commons. Για παράδειγμα, οι περισσότερες φωτογραφίες που βρίσκονται στο Flickr συμφωνούν με αυτή την αρχή.

Σημείωση: Εκτός και αν το wiki είναι εξ ολοκλήρου προστατευμένο με passwords και έχουν πρόσβαση σ' αυτό μόνο οι εγγεγραμμένοι χρήστες π.χ. τα μέλη μιας τάξης, ΔΕΝ επιτρέπεται η χρήση υλικού που προστατεύεται από την αρχή "Fair Use". Αυτή η αρχή δεν επιτρέπει τη χρήση του υλικού στο διαδίκτυο.

1.16 ΕΛΕΓΧΟΣ ΤΩΝ ΔΗΜΟΣΙΕΥΜΑΤΩΝ

1.16.1 Έλεγχος δημοσίευσης (Publishing controls)

Το μέγεθος του ελέγχου δημοσίευσης (publishing control) που έχει ο επισκέπτης του wiki εξαρτάται από τις ρυθμίσεις που θα κάνει ο διαχειριστής του wiki στο λογισμικό. Μπορεί να περιλαμβάνονται διαφορετικά περιβάλλοντα του wiki, πλοήγηση, και άλλα χαρακτηριστικά ρυθμίσεων. Αυτά συνήθως αφορούν τα στοιχεία που βρίσκονται εκτός ή γύρω από το κυρίως κείμενο του wiki. Η εσωτερική περιοχή, ή η περιοχή στην οποία βρίσκεται το κυρίως περιεχόμενο συνήθως ρυθμίζεται έτσι ώστε οι άλλοι να μπορούν να το τροποποιήσουν και να προσθέσουν περιεχόμενο στο wiki.

Ο έλεγχος δημοσίευσης μπορεί να πάρει διάφορες μορφές, μερικά παραδείγματα των οποίων είναι τα παρακάτω:

- *Προσωπική δημοσίευση:*

Ένα wiki μπορεί να χρησιμοποιηθεί σαν ένας προσωπικός χώρος, όπου ένας χρήστης (ο διαχειριστής του wiki) έχει πρόσβαση στο wiki και όλοι οι υπόλοιποι γίνονται απλοί αναγνώστες του περιεχομένου του wiki.

- *Ομαδική δημοσίευση:*

Μία ομάδα ανθρώπων έχει πρόσβαση στο wiki – συνήθως χρησιμοποιώντας users-names και/ή passwords που δίνονται από τον διαχειριστή του wiki. Ο αριθμός των ατόμων που μπορούν να γράψουν, διορθώσουν και να προσθέσουν σελίδες στο wiki είναι περιορισμένος, αλλά όλοι μπορούν να το διαβάσουν.

- *Δημόσια wiki:*

Τροποποιήσεις μπορούν να γίνουν από οποιονδήποτε επισκεφτεί το wiki, ενώ οι αλλαγές και οι προσθήκες παρακολουθούνται από την ομάδα

σαν σύνολο. Ένα καλό παράδειγμα είναι η wikipedia. Αυτού του είδους οι άδειες επιτρέπουν τη συνεργασία με το κοινό.

- *Ιδιωτικά wiki:*

Τα wiki μπορούν να χρησιμοποιηθούν για έναν αριθμό από διαφορετικούς σκοπούς οι οποίοι περιορίζουν την πρόσβαση στο wiki. Για παράδειγμα, αν το wiki χρησιμοποιείται σε ένα τοπικό δίκτυο αυτόματα περιορίζεται η πρόσβαση σ' αυτό το wiki. Η πρόσβαση μπορεί επίσης να εξασφαλίζεται μόνο με usernames/passwords για όλο το περιεχόμενο του wiki. Ή μπορεί να μην είναι επιθυμητή η συνεισφορά από κανέναν άλλο εκτός από μια επιλεγμένη ομάδα ανθρώπων.

Πιθανά εμπόδια όταν χρησιμοποιείται στην εκπαίδευση:

Τα wiki συγκρούονται με τις παραδοσιακές αντιλήψεις σχετικά με την συγγραφή και την πνευματική ιδιοκτησία. – Οι μαθητές κάποιες φορές είναι απρόθυμοι να συνεισφέρουν στα wiki επειδή δεν έχουν αρκετή αυτοπεποίθηση για τα γραπτά τους, ανησυχούν ότι δεν θα αναγνωρισθεί η συνεισφορά τους, ή δεν τους αρέσει να διορθώνονται ή διαγράφονται οι ιδέες, οι λέξεις και η συμμετοχή τους χωρίς τη συγκατάθεση τους. - Κάποιοι δάσκαλοι ή μαθητές έχουν αμφιβολίες σχετικά με τα πλεονεκτήματα και τα μειονεκτήματα της δημοσιοποίησης των κειμένων. - Κάποιοι μαθητές μπορεί να μην είναι ευχαριστημένοι από την υποχρέωση να μάθουν να χρησιμοποιούν ένα wiki ή/και να βρουν ακόμα και τα σχετικά απλά βήματα τροποποίησης ή ανάρτησης υλικού αποθαρρυντικά. - Επειδή τα wiki δεν είναι λογισμικά παρουσίασης, η χρήση γραφικών και οι επιλογές σχεδιασμού είναι περιορισμένες. - Αν και είναι εύκολη η επαναφορά περιεχομένου το οποίο αναπόφευκτα έχει διαγραφεί ή σκοπίμως αλλοιωθεί, η διαδικασία της σύνταξης παρόλα αυτά είναι μία πρόκληση.

1.17 ΟΦΕΛΗ ΑΠΟ ΤΑ WIKI ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ

"Ποια είναι τα οφέλη από τη συμμετοχή των μαθητών-σπουδαστών σε ένα wiki;"



1.17.1 Οφέλη που προκύπτουν από τη χρήση των wiki ως εργαλεία διδασκαλίας και μάθησης:

ΣΥΝΔΕΣΕΙΣ-ΔΙΑΣΥΝΔΕΣΕΙΣ

- Ανάπτυξη μεγαλύτερων συνδέσεων μεταξύ της νέας και της παλιότερης γνώσης επιτρέποντας στους εκπαιδευόμενους τη δημιουργία δομών για τις πληροφορίες και τις ιδέες.
- Οι εκπαιδευόμενοι χρησιμοποιούν τις λειτουργίες της σύνθεσης και της αξιολόγησης σταθερά και συνεπώς όταν εργάζονται σε ένα wiki.

ΔΗΜΙΟΥΡΓΙΚΟΤΗΤΑ

- Ανάπτυξη δημιουργικών δεξιοτήτων, ειδικά τις δεξιότητες της επεξεργασίας και της ευφράδειας. Ανάπτυξη της δημιουργικής ευελιξίας με την αποδοχή των τροποποιήσεων των άλλων.
- Ενθάρρυνση της «εκμαίευσης» ιδεών (ένας τύπος δημιουργικής επεξεργασίας και αναλυτικής σκέψης : αν ισχύει το X, τι ισχύει τότε για το Y;).
- Εισαγωγή και ενίσχυση της ιδέας ότι ένα δημιουργικό έργο δεν είναι ποτέ «ολοκληρωμένο».

ΔΕΣΜΕΥΣΗ

- Αυξάνεται η δέσμευση όλων των εκπαιδευόμενων.
- Οι εκπαιδευόμενοι δεν κατέχουν πλέον τον παθητικό ρόλο του «καταναλωτή», που δέχεται την πληροφορία που του παρουσιάζεται (με αποτέλεσμα να χάνουν το ενδιαφέρον τους και να την αγνοούν), ως συντελεστές του wiki απαντούν, κάνουν αλλαγές και κάνουν βελτιώσεις.
- Έργα τα οποία βρίσκονται σε πολύ προχωρημένο στάδιο δεν είναι απαραίτητο να ολοκληρωθούν.

ΔΙΑΠΡΟΣΩΠΙΚΕΣ ΣΧΕΣΕΙΣ

- Ανάπτυξη διαπροσωπικών και επικοινωνιακών δεξιοτήτων, ιδιαίτερα την επίτευξη ομοφωνίας και του συμβιβασμού, μέσα σε ένα περιβάλλον όπου το αποτέλεσμα κινητοποιεί την διαπροσωπική λύση προβλημάτων.
- Ανάπτυξη ουσιαστικών δεξιοτήτων ομαδικής εργασίας.

ΣΥΓΓΡΑΦΗ

- Βελτίωση της ικανότητας αναθεώρησης και επανεξέτασης.
- Αύξηση της ευελιξίας στην εύρεση εναλλακτικών τρόπων έκφρασης του ιδίου νοήματος.
- Προσαρμογή σε ένα περιβάλλον μεγαλύτερου και πιο αυθεντικού κοινού.

1.18 ΘΕΩΡΙΕΣ ΕΦΑΡΜΟΣΜΕΝΕΣ ΣΤΑ WIKI

1.18.1 Εκπαιδευτικές θεωρίες που εφαρμόζονται στα wiki

- Εποικοδομισμός / Κονστρουκτιβισμός (Constructivism) :

Αντιμετωπίζει τη μάθηση ως μια δυναμική διαδικασία στην οποία οι μαθητές οικοδομούν νέες ιδέες ή αντιλήψεις σχετικές με την παρούσα/παλαιότερη γνώση και σε ανταπόκριση με τις εκπαιδευτικές συνθήκες. Ο εποικοδομισμός υποδηλώνει την ιδέα ότι οι εκπαιδευόμενοι δεν απορροφούν τις πληροφορίες παθητικά αλλά τις συνθέτουν μόνοι τους.

- Κοινότητες πρακτικής :

Πρόκειται για διαδικτυακές κοινότητες ανθρώπων που έχουν κοινό επάγγελμα, θέση, ή έργο. Αυτές οι κοινότητες διευκολύνουν την επαγγελματική συναλλαγή, επιτρέπουν στα μέλη να δημιουργήσουν ένα δεσμό από κοινές εμπειρίες και προκλήσεις.

- Συνεργατική Μάθηση :

Η διαδικασία ομαδικής εργασίας με σκοπό τη συνεισφορά ιδεών.

1.19 ΑΠΟΦΑΣΕΙΣ ΓΙΑ WIKI ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ

Αποφάσεις που πρέπει να ληφθούν πριν την δημιουργία ενός

εκπαιδευτικού wiki:

- Πώς οραματίζεστε ότι θα χρησιμοποιείται το wiki; (Πώς θα το περιγράψετε στους γονείς – αν οι εκπαιδευόμενοι είναι ανήλικοι – και στη διεύθυνση;)
- Ποιοι θα μπορούν να δουν το wiki; (οποιοσδήποτε; Μόνο τα μέλη του wiki;)
- Ποιοι θα μπορούν να τροποποιήσουν το wiki; (οποιοσδήποτε; Μόνο τα μέλη του wiki; Διαφορετικά άτομα ανάλογα με το τμήμα του wiki;)
- Ποιοι θα μπορούν να συμμετέχουν στο wiki; (μόνο οι μαθητές; Οι γονείς; Άτομα που δεχτεί πρόσκληση; Οποιοσδήποτε;)
- Ποια τμήματα του wiki θα «προστατέψετε» (απαγορεύοντας την τροποποίηση τους);
- Ποιος θα διαχειρίζεται - ελέγχει το wiki για την καταλληλότητα του, κλπ. ;
- Ποιος θα έχει τη δυνατότητα να επαναφέρει το wiki μετά από κάποια αλλαγή;
- Εσείς, ως ο υπεύθυνος καθηγητής, θα ενημερώνεστε για όλες τις αλλαγές;

Έγκριση από τη διεύθυνση:

Αφού αποφασίσετε το πώς θα λειτουργεί και πως θα χρησιμοποιείται το wiki, καλύτερα να εξασφαλίσετε την έγκριση του διευθυντή ή του προϊσταμένου σας αν είστε ένας από τους πρώτους δασκάλους που χρησιμοποιούν ένα τέτοιο «νέο» διαδικτυακό εργαλείο. Η πολιτική που ακολουθεί το σχολείο σας μπορεί να έχει κάποιους συγκεκριμένους κανόνες σχετικά με τη χρήση εργαλείων όπως το wiki. Θα χρειαστεί να πληροφορηθείτε τα παρακάτω:

- Επιτρέπεται η ανάρτηση των εργασιών των μαθητών στο διαδίκτυο;
- Ποια είναι η τακτική σχετικά με τη δημοσιοποίηση των ονομάτων των μαθητών (γίνεται χρήση των αρχικών; Ψευδωνύμων;)
- Ποια είναι η τακτική σχετικά με την ανάρτηση εικόνων των μαθητών ή στιγμιότυπων από την αίθουσα;

- Ποια είναι η τακτική σχετικά με την δημοσιοποίηση οποιασδήποτε πληροφορίας από την οποία θα μπορούσε να αναγνωριστεί η τάξη του wiki;
- Υπάρχει η δυνατότητα οι παραπάνω τακτικές να ακολουθηθούν υλοποιώντας κάποιες ρυθμίσεις ασφαλείας, και συμφωνίες με τους μαθητές και τους γονείς τους;
- Υπάρχει κάποιος περιορισμός στην πρόσβαση στα εργαλεία wiki από το σχολικό δίκτυο; Εάν ναι, υπάρχει η δυνατότητα ο διαχειριστής του εργαστηρίου να επιτρέψει την πρόσβαση στο συγκεκριμένο url του wiki;

Επιλογή του σωστού εργαλείου wiki:

Ο σχεδιασμός που θα κάνετε και η ανταπόκριση που θα έχετε από τη διεύθυνση του σχολείου – ιδρύματος σας θα σας βοηθήσουν να αποφασίσετε αν ένα εργαλείο wiki παρέχει τις δυνατότητες και την ασφάλεια που χρειάζεστε για να δημιουργήσετε ένα επιτυχημένο wiki. Έχετε υπόψη σας τις απαντήσεις που έχετε για το βασικό σχεδιασμό και τις διαχειριστικές απαιτήσεις καθώς θα διαλέγετε το εργαλείο που θα χρησιμοποιήσετε.

1.20 ΤΡΟΠΟΙ ΧΡΗΣΗΣ ΣΤΙΣ ΕΠΙΧΕΙΡΗΣΕΙΣ

1. Υποστήριξη πωλήσεων (sales support) - Οι πωλήσεις έχουν μια τεράστια ανάγκη πληροφοριών και χρόνο που περιορίζει. Μια επιχείρηση Wiki είναι ένας καλός τρόπος για έναν οργανισμό να υποστηρίξει τις πωλήσεις του και για τις πωλήσεις να μοιραστούν πληροφορίες.
2. Επικοινωνία project (project communication)- Μια επιχείρηση Wiki δεν πρόκειται να παρέχει Gantt charts αλλά είναι ένας πολύ αποτελεσματικός τρόπος να ρυθμιστεί η επικοινωνία που είναι ζωτικής σημασίας στην επιτυχία του project.
3. Ατυπη εκμάθηση (informal learning)- Το wiki βοηθάει τους ανθρώπους να ψάξουν και να μοιραστούν τη γνώση που ενισχύει την οργανωτική απόδοση.

4. Υποστήριξη εφαρμογής (application support)- Το Wikis είναι ένας ιδανικός τρόπος να φέρει σε επαφή ανθρώπους γρήγορα και να κρατηθούν ενήμεροι για τις σύνθετες εφαρμογές.
5. Επικοινωνίες καναλιών (channel communication)- Η καλή επικοινωνία είναι ουσιαστική για την αποτελεσματικότητα του καναλιού. Το Wikis καθιστά εύκολη τη δημιουργία δυναμικών επικοινωνιών διπλής κατεύθυνσης (two way communications) μεταξύ ενός οργανισμού και του δικτύου του.
6. Μη-κέρδη (non profits) - Επειδή τα μη-κέρδη έχουν συχνά διαφορετικούς συμμετόχους (stakeholders), το wikis είναι ένας ιδανικός τρόπος να ευνοηθούν οι επικοινωνίες και η ανάδραση (feedback).
7. Intranets - Σε πολλές περιπτώσεις η επιχείρηση Wikis αναλαμβάνει ως πρώτηιστη ανάγκη τις εσωτερικές επικοινωνίες και τη διακίνηση της γνώσης.
8. Έρευνα αγοράς (market research) - Οι επιχειρήσεις Wikis, ειδικά εκείνες που ενσωματώνουν εκτεταμένες ικανότητες blog (extensive blog capabilities), αποτελούν μια ισχυρή πλατφόρμα για τη σύλληψη και τη διανομή των σχετικών πληροφοριών για τα projects, τις αγορές, τις τάσεις των ανταγωνιστών και της βιομηχανίας.
9. Προγραμματισμός προϊόντων (product planning)- Μια επιχείρηση Wiki είναι ένας άριστος τρόπος για τη διαμοίραση ιδεών, τη συμμετοχή στη διαδικασία σχεδιασμού προϊόντων καθώς επίσης και την παροχή μιας ισχυρής πλατφόρμας επικοινωνίας καθώς εφαρμόζεται το concept των προϊόντων.
10. Εξυπηρέτηση πελατών (customer service)- Πολλοί μεγάλοι οργανισμοί αρχίζουν να χρησιμοποιούν ένα wiki σαν ένα τρόπο για την εξυπηρέτηση πελατών και τη διαμοίραση των πληροφοριών. Κατόπιν, με την πάροδο του χρόνου οι επιχειρήσεις παίρνουν όλο και περισσότερο τους πελάτες που εμπλέκονται στη συνεργασία και στη διαδικασία διαμοίρασης γνώσης.

2 ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΣΤΟ LOCALHOST

Σκοπός της παρακάτω διαδικασίας είναι να εγκαταστήσουμε στους προσωπικούς μας υπολογιστές το phpMyAdmin wiki. Με αυτό τον τρόπο μπορούμε να επεξεργαστούμε ελεύθερα το wiki που θα δημιουργήσουμε, να ανεβάσουμε αρχεία και να πειραματιστούμε με τη μορφή του wiki χωρίς το αποτέλεσμα να είναι ορατό από κανέναν άλλο πέρα από εμάς. Αυτό βοηθάει τους αρχάριους κυρίως χρήστες να πειραματιστούν ελεύθερα με τις δυνατότητες και τη γλώσσα του wiki αφού λειτουργεί ως «δοκιμαστικό» wiki πριν τη δημιουργία του κανονικού(δηλ. αυτού που είναι ορατό στο internet).

Βήματα:


1. <http://www.wampserver.com/en/download.php>

Επιλογή της έκδοσης wampserver που μας ενδιαφέρει(**DOWNLOAD WampServer 2.0i.**)

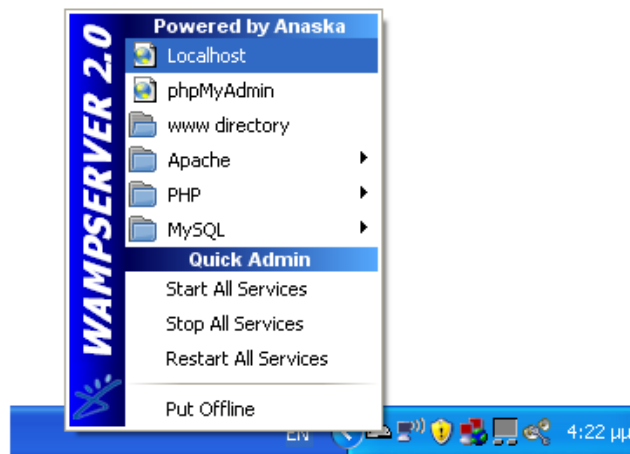
2. Για να μπορέσουμε να μετατρέψουμε το **PC** μας σε **server**, πρέπει να εγκαταστήσουμε το **Wamp** για να υποστηρίξει **Server Apache**, **PHP** (γλώσσα προγραμματισμού) και **MySQL** (βάση δεδομένων). Εμφανίζεται η παρακάτω οθόνη. Κάνουμε κλικ στο σύνδεσμο **direct link**.



Εικόνα 2: Λήψη Wamp Server

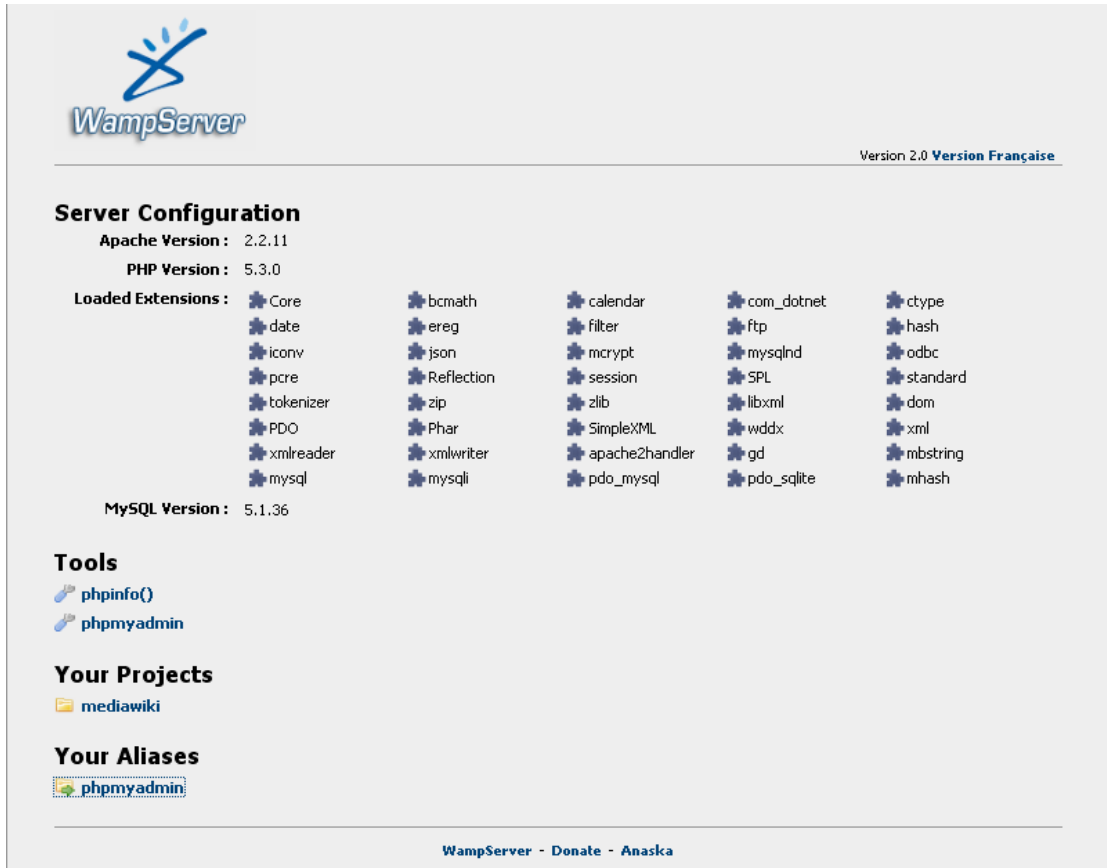
3. Μετά την εγκατάσταση του Wamp εμφανίζεται στην κάτω δεξιά γωνία της γραμμής εργασιών της επιφάνειας εργασίας, το εικονίδιο του **Wamp** .

4. Στο φάκελο [www](#) του WampServer κάνουμε extract το συμπιεσμένο αρχείο [mediawiki.tar.gz](#) και μετονομάζουμε τον πλέον αποσυμπιεσμένο φάκελο σε [mediawiki](#).
5. Ανοίγουμε το localhost για να δημιουργήσουμε τη βάση δεδομένων μας.

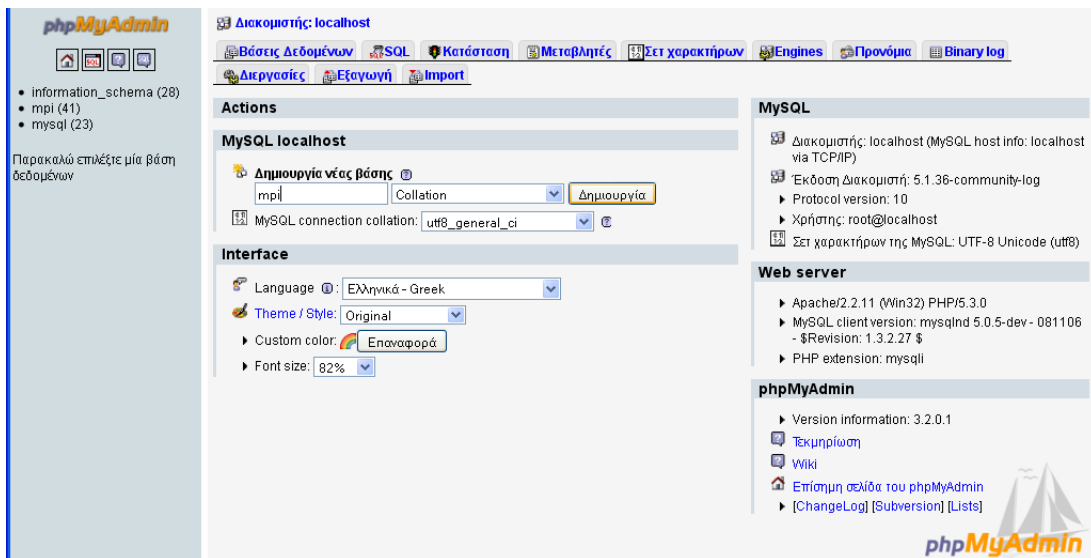


Εικόνα 3: Menu Wamp Server

6. Δημιουργούμε τη νέα μας βάση και της δίνουμε το όνομα που επιθυμούμε (π.χ. mri). Στην επιλογή *language* επιλέγουμε τη γλώσσα που επιθυμούμε. Εδώ επιλέξαμε «Ελληνικά-Greek» γιατί θέλουμε η βάση μας να εμφανίζεται στα ελληνικά και στο πλαίσιο [Theme / Style](#) επιλέξαμε «Original».

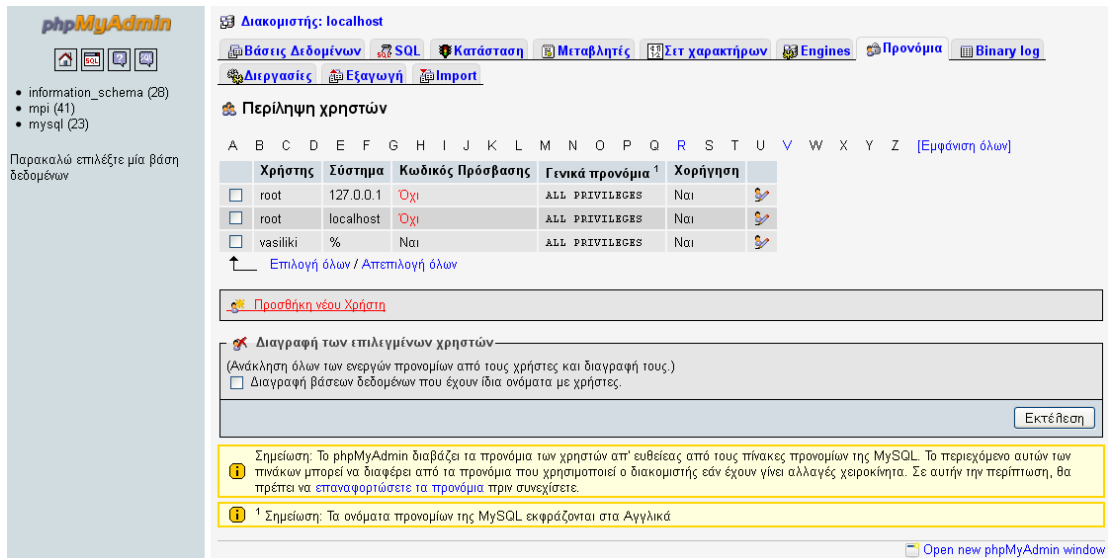


Εικόνα 4: Wamp Server-Άνοιγμα localhost



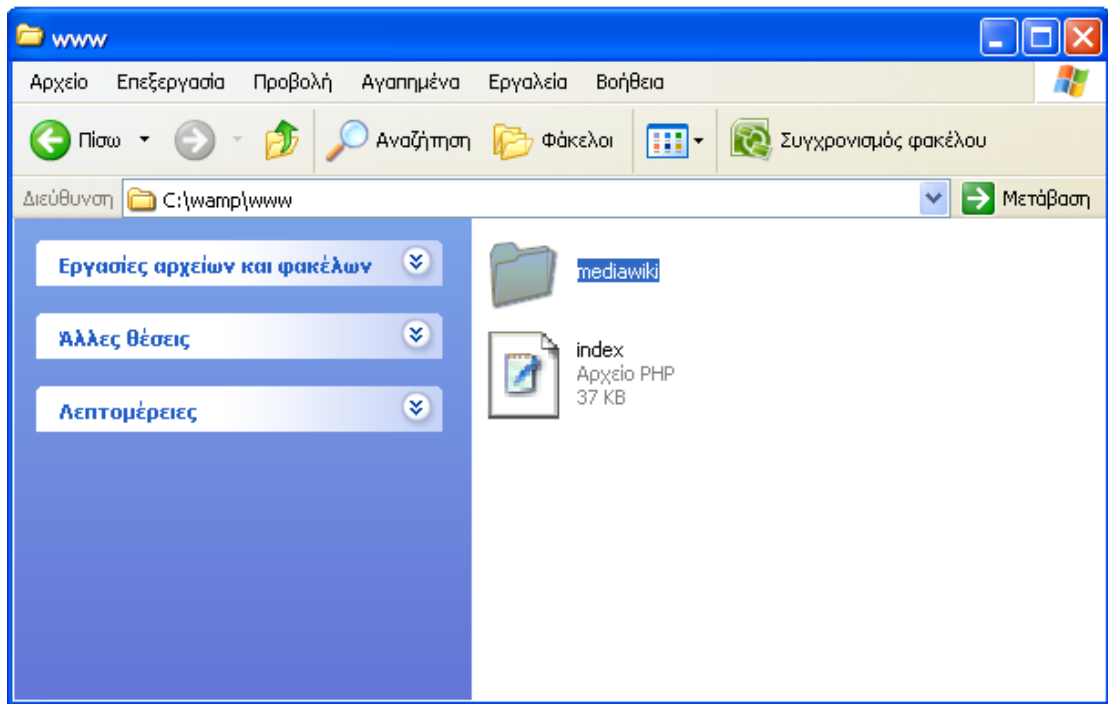
Εικόνα 5: Επιλογές localhost

7. Στη καρτέλα «Πρόνομα» επιλέγουμε «Προσθήκη νέου Χρήστη» και θέτουμε τα γενικά πρόνομα ως *ALL PRIVILEGES*.



Εικόνα 6: Επιλογές localhost

8. Ανοίγουμε το site <http://localhost/mediawiki> (το όνομα mediawiki προέκυψε από το όνομα του φακέλου στον οποίο περιέχονται όλα τα σχετικά αρχεία).



Εικόνα 7: Φάκελος στον οποίο έχει αποθηκευτεί η τοπική βάση

9. Αφού δημιουργήθηκε πλέον η βάση μπορούμε να κάνουμε setup στο wiki – (Setup wiki first) με τα εξής στοιχεία:

Wikiname: π.χ. mpi

e-mail: προαιρετικό

language: π.χ. ελληνικά

Admin username: π.χ WikiSysop

Password: οποιοδήποτε αλφαριθμητικό

10. Αφού ολοκληρωθεί ολοκληρωθεί το install του mediawiki μεταφέρουμε το αρχείο [LocalSettings.php](#) από το φάκελο C:\wamp\www\mpri\config στο γενικό φάκελο C:\wamp\www\mpri

Μετά την παραπάνω διαδικασία το wiki στο localhost είναι έτοιμο για χρήση. Το αρχείο LocalSettings.php είναι πολύ σημαντικό αφού μέσω αυτού ρυθμίζονται πολλές σημαντικές παράμετροι του wiki.

3 ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΣΤΟ DIAS

Βήματα:

1. <http://hydra.it.teithe.gr/index.php>
2. Επιλογή της υπηρεσίας OpenVPN



Εικόνα 8: Υπηρεσίες που παρέχει το ΑΤΕΙΘ


3. Αποδοχή policy χρησιμοποιώντας username και password του τμήματος πληροφορικής.

Policy

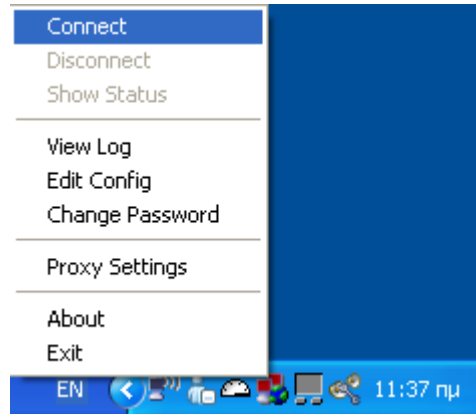
ΠΡΟΣΟΧΗ! Πριν μπορέσετε να κάνετε χρήση της υπηρεσίας θα πρέπει να αποδεχτείτε το σχετικό [policy](#)!

Διαφορετικά δεν λειτουργεί η αυθεντικοποίηση!

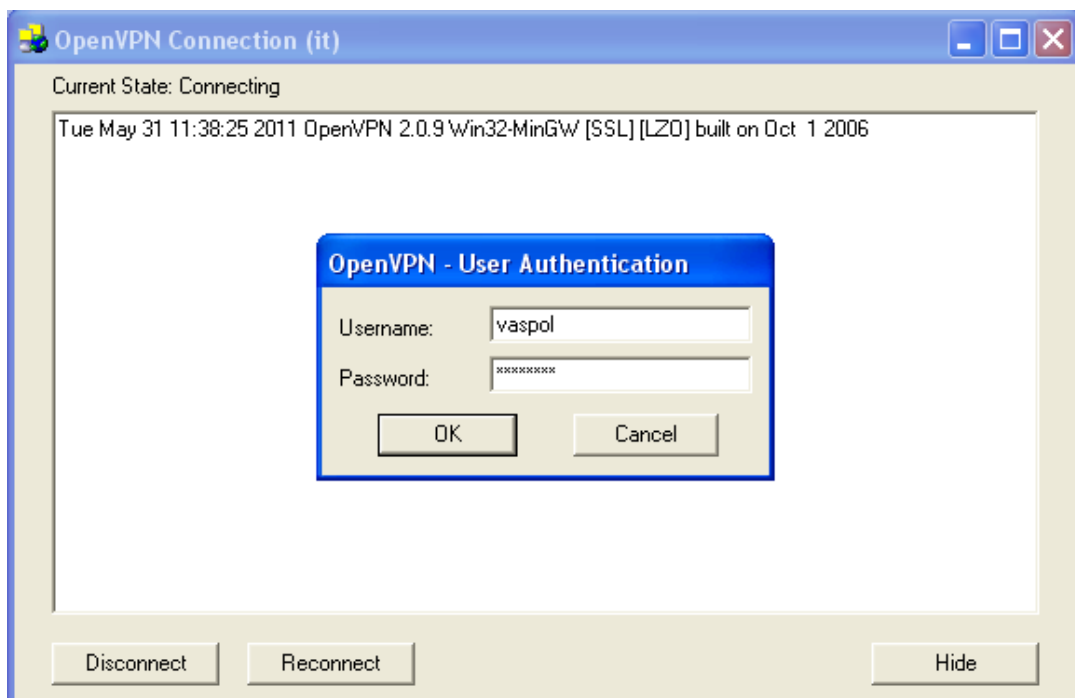
Εικόνα 9: Εμφάνιση μηνύματος πριν τη λήψη της VPN

4. Κάναμε εγκατάσταση της υπηρεσίας για Windows και εμφανίζεται στη γραμμή εργασιών το εικονίδιο της υπηρεσίας .

5. Στο σημείο αυτό ενεργοποιήσαμε την υπηρεσία VPN κάνοντας connect με username και password του aetos.



Εικόνα 10: Menu της υπηρεσίας VPN



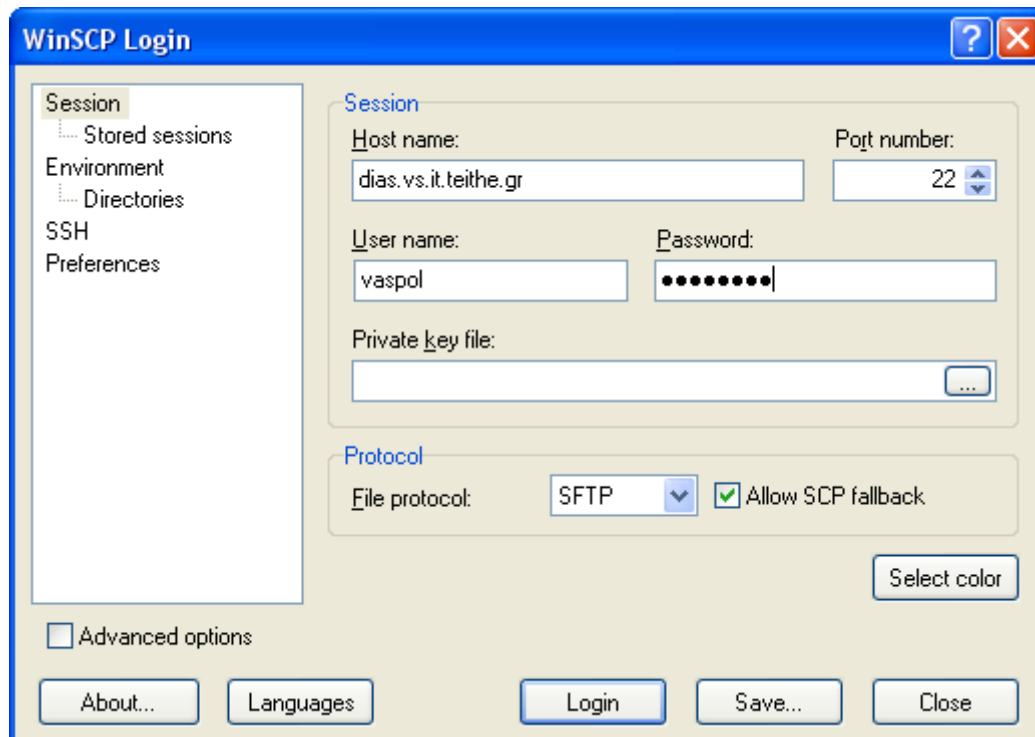
Εικόνα 11: Είσοδος στην υπηρεσία VPN

!!! Ένδειξη ότι το VPN είναι ενεργοποιημένο .

6. Κατεβάσαμε και κάναμε εγκατάσταση το το πρόγραμμα winscp 4.1.9.

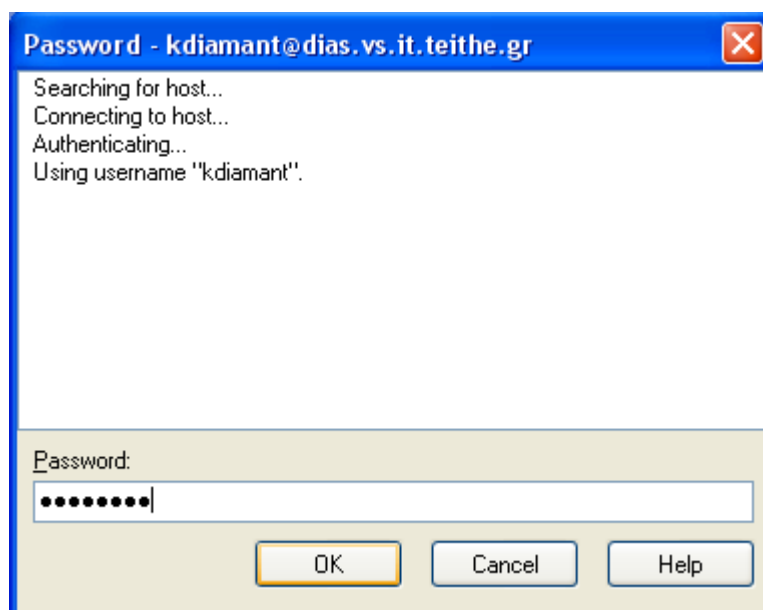
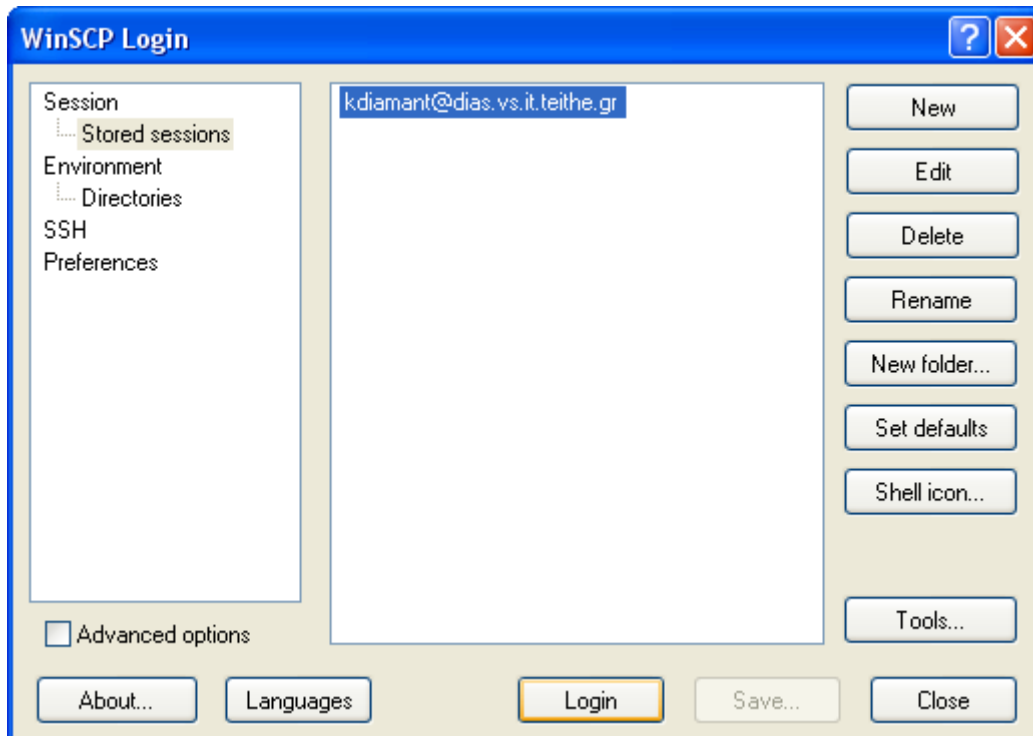
7. Μέσω του winscp συνδεθήκαμε στο dias με τα στοιχεία του aetos του τμήματος πληροφορικής (Επιλογή New).

!!! Με το dias έχουμε πρόσβαση στο server του ΤΕΙ απ' τον υπολογιστή μας.



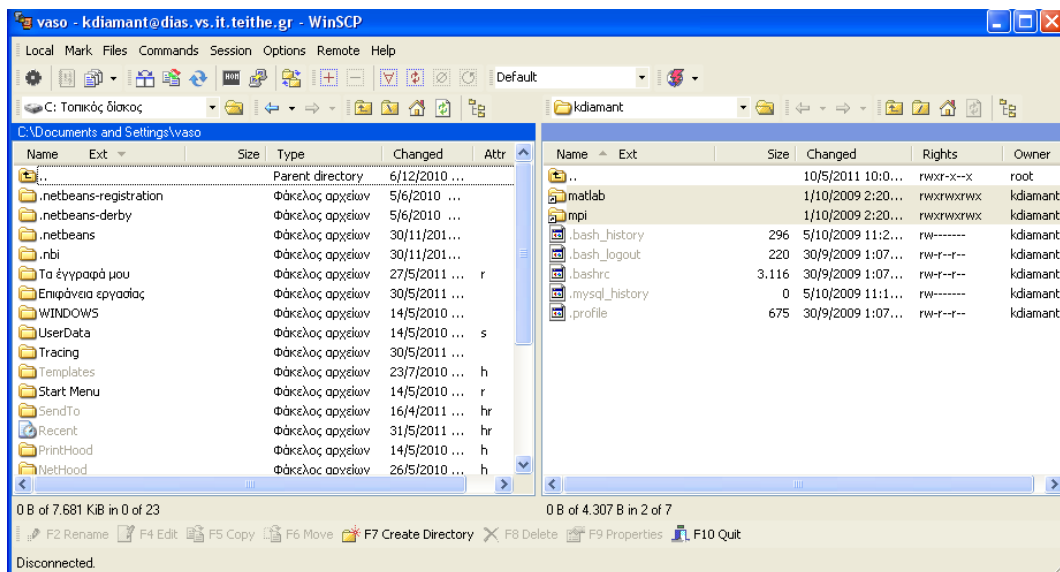
Εικόνα 12: Πρόσβαση στο dias

8. Στο παράθυρο του winscplogin εμφανίζεται πλέον το Hostname που δημιουργήσαμε και κάναμε login δίνοντας τα κατάλληλα username και password όπως φαίνεται στην εικόνα.



Εικόνα 13: Εισαγωγή στο WinScp

9. Μέσα στο dias υπήρχαν δύο φάκελοι με ονόματα matlab και mpi αντίστοιχα. Σε αυτούς τους δύο φακέλους μεταφέραμε τα περιεχόμενα του mediawiki αφού πρώτα το κάναμε extract, ενώ το αρχείο LocalSettings.php το μεταφέραμε στο γενικό φάκελο του καθενός όπως ακριβώς κάναμε και στο localhost.

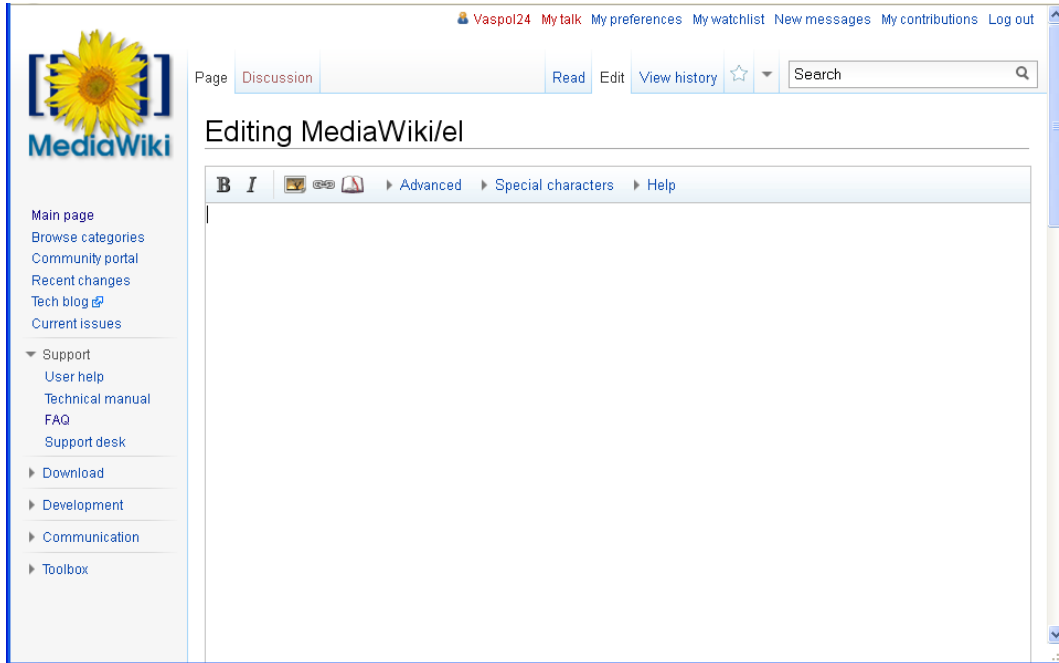


Εικόνα 14: Πρόσβαση στο dias- παράθυρο βάσης

Με την εκτέλεση των παραπάνω βημάτων το mediawiki είναι έτοιμο και ορατό στο διαδίκτυο για μορφοποίηση και επεξεργασία.

4 ΕΠΕΞΕΡΓΑΣΙΑ WIKI

Η γενική μορφή του mediawiki παρουσιάζεται στην επόμενη εικόνα.



Εικόνα 15: Αρχική μορφή του wikimedia

Ωστόσο προσφέρει στους δημιουργούς του τη δυνατότητα να προσαρμόσουν τόσο το μενου του όσο και τη γενικότερη παρουσίαση του ανάλογα με τις επιθυμίες τους.

4.1 ΑΛΛΑΓΗ ΚΕΝΤΡΙΚΗΣ ΕΙΚΟΝΑΣ-LOGO

Για να αλλάξουμε την κεντρική εικόνα του wiki μας, δηλαδή την εικόνα που θα εμφανίζεται σε κάθε σελίδα του wiki πάνω αριστερά, αντικαταστήσαμε την εικόνα-logo στον φάκελο </srv/kdiamant/matlab/skins/common/images> και όνομα [wiki.png](#) με την εικόνα της επιλογής μας.

4.2 ΑΛΛΑΓΗ ΧΡΩΜΑΤΟΣ

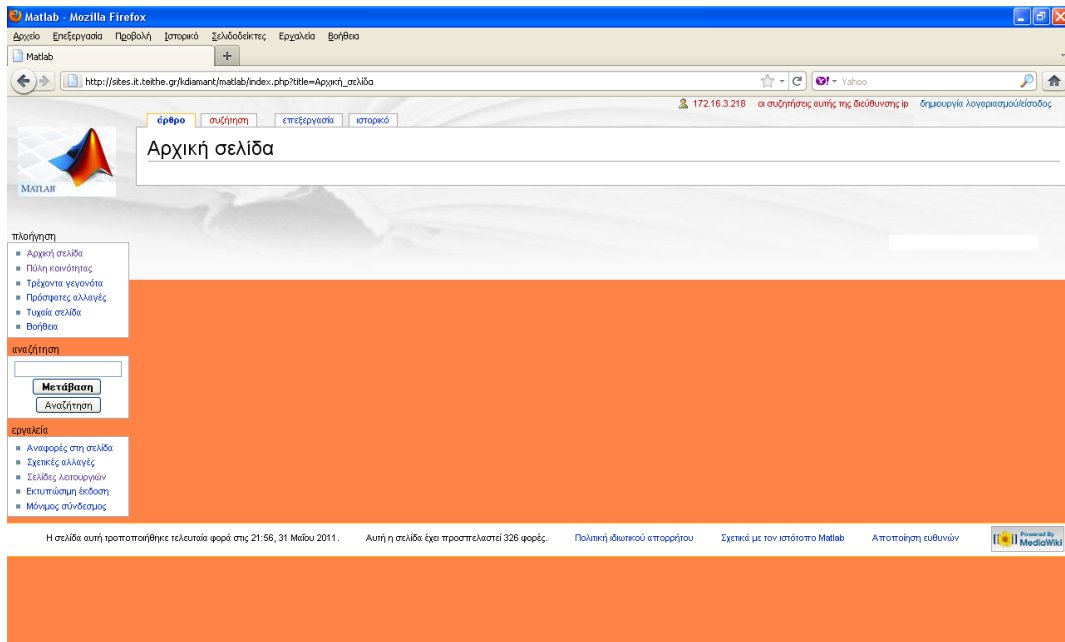
Αλλάξαμε το χρώμα του wiki μας γράφοντας τον κωδικό του χρώματος που επιθυμούμε στον κώδικα του αρχείου [main.css](#) που βρίσκεται στο

φάκελο `/srv/kdiamant/matlab/skins/monobook`. Το τμήμα κώδικα που αλλάζει το χρώμα φαίνεται παρακάτω.

Ο φάκελος `monobook` δείχνει πως από τις όψεις που παρέχει το mediawiki για το wiki μας, εμείς επιλέξαμε την κλασική `monobook`. Άλλες βασικές όψεις που είναι διαθέσιμες είναι οι `modern`, `myskin` και `simple`.

```
body {  
    font: x-small sans-serif;  
    background:#FF8247 url(headbg.jpg) 0 0 no-repeat;  
    color: black;  
    margin: 0;  
    padding: 0;  
}
```

Σε αυτό το σημείο το wiki μας είχε την εξής μορφή



Εικόνα 16: Μορφή wiki του μετά την αλλαγή χρώματος και κεντρικής εικόνας

4.3 ΕΠΕΞΕΡΓΑΣΙΑ ΚΕΙΜΕΝΟΥ ΣΕΛΙΔΑΣ

Για να επεξεργαστούμε ένα άρθρο απλά ενεργοποιήσαμε τον σύνδεσμο (πατώντας τον) «**Επεξεργασία**» που υπάρχει στο μενού κάθε άρθρου.

Κατόπιν χρησιμοποιώντας τη γλώσσα σήμανσης **markup** των σελίδων WikiWiki γράψαμε τα άρθρα μας σχετικά με το matlab και το mpi. Μερικά

στοιχεία της γλώσσα σήμανσης **markup** που χρησιμοποιήθηκαν για τη συγγραφή των άρθρων μας παρουσιάζονται παρακάτω.

4.3.1 Στοιχίση

- ✓ Μία μόνο νέα γραμμή (Enter) δεν επηρεάζει το κείμενο. Ωστόσο δύο νέες γραμμές (δηλαδή μια κενή γραμμή) αρχίζουν μια νέα παράγραφο.

```
Οι γραμμές μπορούν να χωριστούν<br>χωρίς να αρχίσετε νέα παράγραφο.
```

Οι γραμμές μπορούν να χωριστούν
χωρίς να αρχίσετε νέα παράγραφο.

- ✓ Λίστες

```
* Οι λίστες υλοποιούνται πολύ εύκολα:  
** αρχίστε κάθε γραμμή με έναν αστερίσκο  
*** έναν για κάθε επίπεδο βάθους της λίστας
```

- Οι λίστες υλοποιούνται πολύ εύκολα:
 - κάθε γραμμή αρχίζει με έναν αστερίσκο
 - έναν για κάθε επίπεδο βάθους της λίστας

```
# Μπορείτε επίσης να κάνετε αριθμημένες λίστες  
## είναι χρήσιμες  
## και εύκολες
```

1. Επίσης οι αριθμημένες λίστες
 1. είναι χρήσιμες
 2. και εύκολες

```
* Μπορείτε επίσης να κάνετε μεικτές λίστες  
*# και να τις βάλετε τη μία μέσα στην άλλη  
**# εναλλάσσοντας αστερίσκους και διέσεις
```

- Μπορείτε επίσης να κάνετε μεικτές λίστες

1. και να τις βάλετε τη μία μέσα στην άλλη
 - εναλλάσσοντας αστερίσκους και διέσεις

; Λίστα ορισμών: ορισμοί
; αντικείμενο : ορισμός του αντικειμένου

Λίστα ορισμών

ορισμοί

αντικείμενο

ορισμός του αντικειμένου

✓ : Μια άνω-κάτω τελεία βάζει μια εσοχή πριν την πρόταση.

Μια άνω-κάτω τελεία βάζει μια εσοχή πριν την πρόταση.

✓ <center>Κείμενο στοιχισμένο στη μέση.</center>

Κείμενο στοιχισμένο στη μέση.

✓ Μια οριζόντια διαχωριστική γραμμή:
αυτό είναι από πάνω

και αυτό είναι από κάτω

Μια οριζόντια διαχωριστική γραμμή:

αυτό είναι από πάνω

και αυτό είναι από κάτω

4.3.2 Συνδέσεις, URLs, Εικόνες

✓ Σύνδεσμοι

Το `[[MATLAB]]` είναι ένα σύγχρονο ολοκληρωμένο μαθηματικό λογισμικό πακέτο.

Το [MATLAB](#) είναι ένα σύγχρονο ολοκληρωμένο μαθηματικό λογισμικό πακέτο.

- Το πρώτο γράμμα του συνδέσμου γίνεται κεφαλαίο
- Εσωτερικά, κάθε κενό μετατρέπεται στο χαρακτήρα `'_'` (μπορεί να γράφετε κατευθείαν το σύμβολο `_` αλλά δεν προτείνεται)

Έτσι, ο παραπάνω σύνδεσμος οδηγεί στο <http://sites.it.teithe.gr/kdiamant/matlab/MATLAB> του wiki μας, που είναι ένα άρθρο σχετικό με το MATLAB.

`[[Γρηγοριανό ημερολόγιο#Ακρίβεια]]`

Σύνδεσμος προς μια συγκεκριμένη ενότητα ενός άρθρου, πχ [Γρηγοριανό ημερολόγιο#Ακρίβεια](#) (σύνδεσμοι σε ενότητες που δεν υπάρχουν δεν αποτελούν πρόβλημα, στέλνουν το χρήστη στην αρχή της σελίδας).

Όταν ο στόχος έχει διαφορετικό όνομα από τη λέξη που θέλετε να στέλνει το χρήστη εκεί, το γράφετε έτσι:
`[[στόχος|λέξη]]`.
Για παράδειγμα: `[[Wikipedia:Sandbox|Μπούρδες]]`

Όταν ο στόχος έχει διαφορετικό όνομα από τη λέξη που θέλετε να στέλνει το χρήστη εκεί, το γράφετε έτσι: [λέξη](#). Για παράδειγμα:

[Μπούρδες](#)

Ό,τι είναι σε παρένθεση κρύβεται αυτόματα `[[Σύννεφα με παντελόνια (συγκρότημα)|]]`.

όπως και τα namespaces:
`[[Βικιπαίδεια:Αγορά|]]`.

Ό,τι είναι σε παρένθεση κρύβεται αυτόματα αν τοποθετήσετε μια κάθετο πριν κλείσετε τον σύνδεσμο: [Σύννεφα με παντελόνια](#).

όπως και τα namespaces: [Αγορά](#).

Ο διακομιστής αυτόματα γράφει το κομμάτι μετά το | όταν αποθηκεύεται η σελίδα. Την επόμενη φορά που θα επεξεργαστείτε τη σελίδα θα δείτε ένα σύνδεσμο με διαφορετικό όνομα . Αυτό δε συμβαίνει όταν προεπισκοπείτε μια μια σελίδα, αν και ο διερμηνέας καταλαβαίνει σωστά το σύνδεσμο. Το ίδιο συμβαίνει και την επόμενη λειτουργία.

```
Όταν προσθέτετε ένα σχόλιο στις σελίδες
συνομιλίας
πρέπει να το υπογράφετε. Αυτό το κάνετε
γράφοντας:
~~~
που εμφανίζει το όνομα χρήστη σας, ή
~~~~ που εμφανίζει το όνομα χρήστη
και την ημερομηνία και την ώρα.
```

Ο

ταν προσθέτετε ένα σχόλιο στις σελίδες συνομιλίας , πρέπει να το υπογράφετε. Αυτό το κάνετε γράφοντας: [Όνομα](#) που εμφανίζει το όνομα χρήστη σας, ή [Όνομα](#) 22:01, 22 Μαρτ. 2011 (UTC) που εμφανίζει το όνομα χρήστη και την ημερομηνία και την ώρα.

```
Η [[υπογεννητικότητα των Βεδουίνων]]
είναι μια σελίδα που δεν υπάρχει ακόμα.
```

Η [υπογεννητικότητα των Βεδουίνων](#)

είναι μια σελίδα που δεν υπάρχει ακόμα. Για αυτό είναι σε κόκκινο.

- Μπορεί να τη δημιουργηθεί πατώντας πάνω στο σύνδεσμο.
- Για να δημιουργηθεί μια νέα σελίδα:
 1. Προσθέτουμε ένα σύνδεσμο προς αυτήν σε μια άλλη σελίδα.
 2. Αποθηκεύουμε.

3. Πατάμε πάνω στο σύνδεσμο που δημιουργήσαμε. Η νέα σελίδα θα ανοίξει έτοιμη να γραφτεί.

```
#REDIRECT [[MATLAB]]
```

Για να κατευθύνουμε μια σελίδα σε μια άλλη, γράφουμε το παραπάνω κείμενο σε αυτή.

```
Εξωτερικός σύνδεσμος:  
[http://www.nupedia.com Nupedia]
```

Εξωτερικός σύνδεσμος: [Nupedia](http://www.nupedia.com)

```
Ή απλά δώστε το URL:  
http://www.nupedia.com.
```

Ή απλά δίνουμε το URL: <http://www.nupedia.com>.

- Στο **URL** μόνο τα σύμβολα που ακολουθούν αναγνωρίζονται: A-Z a-z 0-9 ._~%~+&#?!=()@ \x80-\xFF. Αν ένα URL περιέχει άλλους χαρακτήρες, πρέπει να μετατραπούν στο ASCII αντίστοιχό τους. Για παράδειγμα, ο χαρακτήρας ^ γράφεται %5E.

Για να ανοίγουν οι εξωτερικοί σύνδεσμοι σε νέο παράθυρο ή καρτέλα πρέπει να δώσουμε αυτή την επιλογή στο wiki μας προσθέτωντας την παρακάτω γραμμή κώδικα στο αρχείο LocalSettings.php.

```
$wgExternalLinkTarget = "_blank";
```

```
ISBN 0123456789X
```

Για να συνδεθούμε με βιβλία μπορούμε να χρησιμοποιήσουμε συνδέσμους [ISBN](#).

ISBN 0123456789X

✓ Προσθήκη Εικόνων

Για να ανεβάσουμε κάποια εικόνα ή αρχείο πολυμέσων γενικότερα στο wiki μας, πρέπει να δηλώσουμε πως επιτρέπουμε το ανέβασμα τέτοιων αρχείων.

Για να είναι εφικτό αυτό γράψαμε το παρακάτω απόσπασμα κώδικα στο αρχείο LocalSettings.php.

```
$wgEnableUploads = true;  
$wgFileExtensions = array('png', 'gif', 'jpg',  
'jpeg', 'doc', 'xls', 'mpp', 'pdf', 'ppt',  
'tiff', 'bmp', 'docx', 'xlsx', 'pptx', 'ps');
```

Με την πρώτη γραμμή ενεργοποιείται η φόρτωση αρχείων πολυμέσων στο wiki ενώ η επόμενη γραμμή περιλαμβάνει τους τύπους των αρχείων που επιτρέπεται να φορτώσουμε.

Στη συνέχεια κάθε σύνδεσμος μέσα σε σελίδα του wiki που οδηγεί σε εικόνα όταν πατηθεί για πρώτη φορά ανοίγει έναν browser για αναζήτηση του αρχείου που θέλουμε να φορτώσουμε.

```
[[Αρχείο:Name_of_image.ext]]
```

```
[[Αρχείο:Name_of_image.ext|left]]
```

Η απλούστερη τροποποίηση στην θέση μιας εικόνας που μπορείτε να κάνετε είναι να την στοιχίσετε δεξιά ή αριστερά σε σχέση με το κείμενο, μετατρέποντάς την σε δεξί ή αριστερό περιθώριο του κειμένου αντίστοιχα. Το κείμενο που θα γράψετε μετά την εικόνα σε αυτήν την περίπτωση, θα αναδιπλωθεί δίπλα στην εικόνα.

```
[[Αρχείο:Paradeigma.jpg|frame]]
```

Για να τοποθετήσετε μια εικόνα σε πλαίσιο (όσο για το σε τι χρησιμεύει ένα πλαίσιο, δείτε παρακάτω), απλά προσθέτετε τη λέξη *frame* μετά από μία κάθετο οπουδήποτε μετά το όνομα της εικόνας στην αναφορά σας σε αυτήν.



Π.χ.

Εικόνα 17: Εικόνα σε πλαίσιο

ΠΡΟΣΟΧΗ: όταν τοποθετείτε μία εικόνα σε πλαίσιο, αυτομάτως στοιχίζεται δεξιά. Για να αποφύγετε τη στοίχιση, μπορείτε να γράψετε τη λέξη *none* μετά από μία κάθετο στην αναφορά στην εικόνα, όπως ακριβώς θα κάνατε για να τη στοιχίσετε.

Λεζάντα

Συχνά μία εικόνα χρειάζεται μια σύντομη επεξήγηση για να γίνει κατανοητό τι δείχνει ή πως συνδέεται αυτό που δείχνει με το άρθρο. Στην περίπτωση αυτή γράφουμε μία λεζάντα η οποία θα εμφανιστεί κάτω από την εικόνα.

Για να εμφανίζεται η λεζάντα κάτω από μια εικόνα, πρέπει αυτή να βρίσκεται σε πλαίσιο. Έτσι, αυτό που γράφουμε τελικά είναι της παραπάνω μορφής.

`[[Αρχείο:Name_of_Image.ext |frame |λεζάντα]]`



Π.χ.

Εικόνα 18: Εικόνα με λεζάντα

Αλλαγή μεγέθους

Άλλη μια εύκολη μορφοποίηση που μπορείτε να εφαρμόσετε σε μια εικόνα είναι να αλλάξετε το μέγεθός της. Αυτό το κάνετε προσθέτοντας μετά από μια κάθετο *XXXXpx* όπου *XXXX* το πλάτος σε εικονοστοιχεία (*pixels*) που θέλετε να έχει η τελική εικόνα (οι αναλογίες διατηρούνται κατά την αλλαγή μεγέθους). Για παράδειγμα:



Π.χ. Αυτή είναι η εικόνα μας, μειωμένη στα 75 εικονοστοιχεία πλάτος.



...και εδώ μεγεθυμένη στα 200

Κείμενο αντικατάστασης

Μπορούμε επίσης να προσθέσετε κείμενο αντικατάστασης σε μία εικόνα, δηλαδή ένα κείμενο που θα εμφανίζεται σε περιηγητές που έχουν απενεργοποιημένες τις εικόνες, ή δεν μπορούν να τις εμφανίσουν, όπως οι περιηγητές κειμένου. Αυτό το κάνουμε προσθέτοντας το κείμενο που θέλουμε μετά από μία κάθετο, **μετά από οποιαδήποτε άλλη παράμετρο** στην αναφορά στην εικόνα.

Σε περίπτωση που έχουμε τοποθετήσει την εικόνα σας σε πλαίσιο ή την έχουμε κάνει μικρογραφία, το κείμενο αντικατάστασης θα εμφανιστεί επίσης ως λεζάντα στο κάτω μέρος του κειμένου.

Μικρογραφίες

Μια πολύ χρήσιμη δυνατότητα που δίνει το mediawiki είναι η εμφάνιση μικρογραφίας μιας εικόνας αντί για την ίδια την εικόνα. Εάν αποφασίσουμε να προσθέσουμε στη σελίδα μας τη μικρογραφία μιας εικόνας, αυτή θα σμικρυνθεί, θα στοιχηθεί στα δεξιά και η μικρογραφία θα γίνει αυτόματα σύνδεσμος προς την πλήρη εικόνα.

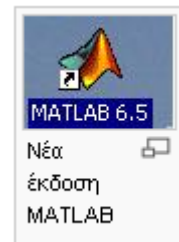
Αυτό είναι ιδιαίτερα χρήσιμο με μεγάλες εικόνες που δεν μπορούν να προστεθούν σε μια σελίδα αυτούσιες λόγω μεγέθους. Επίσης, όταν η εικόνα συνοδεύεται από κάποιο κείμενο που συμπληρώνει το νόημά της, αυτό μπορεί να γραφτεί στη σελίδα της εικόνας, δίνοντας στους χρήστες τη δυνατότητα να το διαβάσουν ακολουθώντας το σύνδεσμο της μικρογραφίας (σε διαφορετική περίπτωση, η εικόνα θα εμφανιζόταν μόνη της στη σελίδα και το συνοδευτικό κείμενο δε θα φαινόταν πουθενά). Επίσης, οι μικρογραφίες εικόνων μπορούν να είναι μεγαλύτερες από τις εικόνες (οπότε

αυτός είναι ένας τρόπος να μεγεθύνετε την εικόνα σας - αν και αυτό δεν συνιστάται γιατί η ποιότητα της εικόνας υποβαθμίζεται).

Για να εισάγουμε τη μικρογραφία μιας εικόνας αντί για την ίδια την εικόνα, προσθέτουμε απλώς στην αναφορά της τη λέξη *thumb* μετά από μία κάθετο.

[[Αρχείο:Name_of_Image.ext|thumb|Εδώ γράφετε τη λεζάντα της μικρογραφίας]]

Σημείωση: για να ξεχωρίσουμε τις μικρογραφίες από τις εικόνες σε πλαίσιο, παρατηρούμε τα δύο μικρά παραλληλόγραμμα που βρίσκονται στην κάτω δεξιά γωνία κάθε μικρογραφίας. Αυτό υποδηλώνει πως το πλαίσιο είναι σύνδεσμος σε μια μεγαλύτερη εκδοχή της εικόνας.



Εικόνα 19: Εικόνα-Μικρογραφία

Συνδυασμός μεθόδων

Τα παραπάνω εργαλεία είναι εύχρηστα και απλά στη χρήση. Πρακτικά όμως ποτέ δε θα χρησιμοποιήσουμε μόνο ένα, αλλά θα χρειαστεί να τα συνδυάσουμε. Αυτό γίνεται πολύ απλά: προσθέτουμε όλα τα χαρακτηριστικά που θέλουμε να έχει η εικόνα μας, χωρισμένα με καθέτους, με το μόνο περιορισμό ότι **η πρώτη παράμετρος πρέπει να είναι πάντα το όνομα της εικόνας και η τελευταία η λεζάντα/κείμενο αντικατάστασης**. Φυσικά πρέπει επίσης να προσέξουμε να μην γράψουμε παραπάνω από μία παραμέτρους για το ίδιο χαρακτηριστικό. Στην περίπτωση αυτή, θα εφαρμοστεί η παράμετρος που βρίσκεται τελευταία.

Μη αναδίπλωση

Εάν θέλουμε το κείμενο να μην αναδιπλώνεται δίπλα σε μια εικόνα αλλά να ξεκινά μετά από αυτήν, προσθέτουμε την εξής γραμμή HTML στο σημείο όπου θέλουμε το κείμενο να πάψει να αναδιπλώνεται:

```
<br style="clear:both"/>
```


Εμφάνιση συνδέσμου αντί για εικόνα

Εάν θέλουμε αντί για την εικόνα στη σελίδα μας να υπάρχει σύνδεσμος προς αυτή, αντί για *Εικόνα*, γράφουμε *Μέσον* στην αναφορά της, όπως ακριβώς θα κάναμε με κάποιο άλλο πολυμέσο, όπως ένα αρχείο ήχου. Αυτός ο σύνδεσμος οδηγεί στην εικόνα αυτή καθεαυτή, όχι στη σελίδα της.

Εάν θέλουμε ο σύνδεσμος να οδηγεί στη σελίδα της εικόνας, απλά προσθέτουμε μία άνω κάτω τελεία (":") πριν το *Εικόνα* στην αναφορά της εικόνας.

Μπορούμε στις παραπάνω αναφορές να προσδιορίσουμε το κείμενο του συνδέσμου όπως ακριβώς θα γράφαμε το κείμενο αντικατάστασης της εικόνας.

```
<br />[[Μέσον:Demo chorus.ogg|Ήχος]]  
<br />  
<br />[[Μέσον:Tornado.jpg|Εικόνα τυφώνα]]
```

Γκαλερί εικόνων

Οι εικόνες μπορούν να οριοθετούνται κατά πλάτος, καθ' ύψος ή και συνδυασμένα, συμπεριλαμβάνοντας ή παραλείποντας τις αντίστοιχες παραμέτρους (στο παρακάτω παράδειγμα γίνεται στοίχιση της σειράς των εικόνων στο κέντρο της σελίδας).

```
<center><gallery widths=160px heights=180px>  
Αρχείο:εικόνα1.jpg|λεζάντια1  
Αρχείο:εικόνα2.jpg|λεζάντια2  
Αρχείο:εικόνα3.jpg|λεζάντια3  
</gallery></center>
```

4.3.3 Μορφοποίηση Χαρακτήρων

✓ **Δώστε έμφαση, δυνατά, πολύ δυνατά.**

- Αυτά είναι διπλά κι τριπλά απόστροφα.

```
' Δώστε έμφαση', 'δυνατά',  
'πολύ δυνατά'.
```

- ✓ Γράψτε επίσης με *πλάγια* και **έντονα** τυπογραφικά στοιχεία εάν το επιθυμητό αποτέλεσμα είναι ένας τύπος:

$$\mathbf{V} = R\mathbf{I}$$

```
Γράψτε επίσης με <i>πλάγια</i> και <b>έντονα</b>  
τυπογραφικά στοιχεία εάν το επιθυμητό αποτέλεσμα  
είναι ένας μαθηματικός τύπος:  
<b>V</b> = <i>R</i><b>I</b>
```

- Παρόλα αυτά η διαφορά δεν είναι πολύ σημαντική και οι περισσότεροι την επιλέγουν να την αγνοήσουν,
- ✓ Γραμματοσειρά με εμφάνιση γραφομηχανής για τους τεχνικούς όρους.

```
A typewriter font for <tt>technical terms</tt>.
```

- ✓ Μπορείτε να χρησιμοποιήσετε μικρά γράμματα για λεζάντες.

```
Μπορείτε να χρησιμοποιήσετε <small>μικρά γράμματα</small> για λεζάντες
```

- ✓ Μπορείτε να ~~σβήσετε~~
και να υπογραμμίσετε.

```
Μπορείτε <strike> να σβήσετε</strike> <u>και να υπογραμμίσετε</u>.
```

- ✓ Διαλυτικά και τόνοι:

À Á Â Ã Ä Å

Æ Ç È É Ê Ë

Ì Í Î Ñ Ò

Ó Ô Õ Ö Ø Ù

Ú Û Ü β à á

â ã ä å æ ç

è é ê ë ì í
î ï ñ ò ó ô
œ õ ö ø ù ú
û ü ÿ

```
&Agrave; &Aacute; &Acirc; &Atilde; &Auml; &Aring;  
&AElig; &Ccedil; &Egrave; &Eacute; &Ecirc; &Euml;  
&Igrave; &Iacute; &Icirc; &Iuml; &Ntilde; &Ograve;  
&Oacute; &Ocirc; &Otilde; &Ouml; &Oslash; &Ugrave;  
&Uacute; &Ucirc; &Uuml; &szlig; &agrave; &aacute;  
&acirc; &atilde; &auml; &aring; &aelig; &ccedil;  
&egrave; &eacute; &ecirc; &euml; &igrave; &iacute;  
&icirc; &iuml; &ntilde; &ograve; &oacute; &ocirc;  
&oelig; &otilde; &ouml; &oslash; &ugrave; &uacute;  
&ucirc; &uuml; &yuml;
```

✓ Στίξη:

¿ ¡ « » § ¶
† ‡ • —

```
&iquest; &iexcl; &laquo; &raquo; &sect; &para;  
&dagger; &Dagger; &bull; &mdash;
```

✓ Εμπορικοί χαρακτήρες:

™ © ® ¢ € ¥
£ ¤

```
&trade; &copy; &reg; &cent; &euro; &yen;  
&pound; &curren;
```

✓ Υπογεγραμμένοι όροι: x_2

Εκθέτες: x^2 or x^2

- Η δεύτερη μέθοδος για την εμφάνιση του εκθέτη δεν μπορεί να χρησιμοποιηθεί σε γενική χρήση, αλλά προτιμάται όπου είναι δυνατόν (πχ. στις μονάδες μέτρησης) επειδή οι περισσότεροι φυλλομετρητές την μορφοποιούν πιο εύκολα.

$$\epsilon_0 = 8.85 \times 10^{-12}$$

C² / J m.

```
Subscript: x<sub>2</sub>
Superscript: x<sup>2</sup> or x&sup2;

&epsilon;<sub>0</sub> =
8.85 &times; 10<sup>&minus;12</sup>
C&sup2; / J m.
```

✓ Ελληνικοί χαρακτήρες:

α β γ δ ε ζ

η θ ι κ λ μ ν

ξ ο π ρ σ ς

τ υ φ χ ψ ω

Γ Δ Θ Λ Ξ Π

Σ Φ Ψ Ω

```
&alpha; &beta; &gamma; &delta; &epsilon; &zeta;
&eta; &theta; &iota; &kappa; &lambda; &mu; &nu;

&xi; &omicron; &pi; &rho; &sigma; &sigmaf;
&tau; &upsilon; &phi; &chi; &psi; &omega;
&Gamma; &Delta; &Theta; &Lambda; &Xi; &Pi;
&Sigma; &Phi; &Psi; &Omega;
```

✓ Μαθηματικοί χαρακτήρες:

∫ ∑ ∏ √ − ± ∞

≈ α ≡ ≠ ≤ ≥ →

x · ÷ ∂ ' "

∇ ‰ ∘ ∴ ∞ ∅

∈ ∉ ∩ ∪ ⊂ ⊃ ⊆ ⊇

¬ ∧ ∨ ∃ ∀ ⇒ ⇔

→ ↔

```
&int; &sum; &prod; &radic; &minus; &plusmn; &infin;  
&asymp; &prop; &equiv; &ne; &le; &ge; &rarr;  
&times; &middot; &divide; &part; &prime; &Prime;  
&nabla; &permil; &deg; &there4; &alefsym; &oslash;  
&isin; &notin; &cap; &cup; &sub; &sup; &sube; &supe;  
&not; &and; &or; &exist; &forall; &rArr; &hArr;  
&rarr; &harr;
```

- ✓ $x^2 \geq 0$ αληθεύει.
- Για την εισαγωγή κενών διαστημάτων χρησιμοποιείτε non-breaking spaces - ` `.
- Το ` ` αποτρέπει επίσης τις αλλαγές γραμμών στο μέσο του κειμένου κάνοντας το χρήσιμο για τους μαθηματικούς τύπους

```
<i>x</i><sup>2</sup>&nbsp;&nbsp;&ge;&nbsp;&nbsp;&nbsp;0 true
```

- ✓ Πολύπλοκοι τύποι:

$$\sum_{n=0}^{\infty} \frac{x^n}{n!}$$

```
<math>\sum_{n=0}^{\infty} \frac{x^n}{n!}</math>
```

Για να είναι εφικτή η σύνταξη μαθηματικών τύπων θα πρέπει να ενεργοποιηθεί η δυνατότητα `textvc` του wiki. Για να συμβεί αυτό πρέπει στο αρχείο `LocalSettings.php` να θέσουμε την ιδιότητα `usetex` από `false` σε `true` όπως φαίνεται παρακάτω.

```
$wgUseTeX = true;
```

- ✓ Τοποθέτηση σχολίων στην πηγαία σελίδα:

δεν φαίνεται στην σελίδα

- Χρησιμοποιείται για την τοποθέτηση σχολίων σε μια σελίδα για τους μελλοντικούς συντάκτες.

```
<!-- τοποθετήστε το σχόλιο σας εδώ -->
```

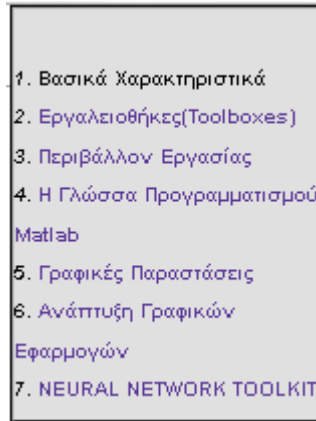
4.4 ΠΡΟΣΘΗΚΗ ΑΝΑΔΥΟΜΕΝΟΥ MENU ΣΤΟ WIKI

Σε κάθε σελίδα του wiki μας επιλέξαμε να εμφανίζεται ένα επιπλέον menu με τα περιεχόμενα του wiki ώστε να διευκολύνεται ο αναγνώστης τόσο στην περιήγηση όσο και στην αναζήτηση του υλικού που τον ενδιαφέρει.

Έτσι σε κάθε σελίδα προσθέσαμε ένα κομμάτι (div) με τους συνδέσμους /περιεχόμενα που μας ενδιαφέρουν, όπως φαίνεται παρακάτω:

```
<div id="hideshow" style="font-size:100%; border:2px groove;
background-color:#e0e0e0; font-weight:bold; right:10px;
width: 152px; position: fixed; top:37px; ">
<div style="font-size:100; border:0px solid; background-
color:#D3D3D3 ; ">
<small><br>'1.' [[Βασικά Χαρακτηριστικά]] <br>
'2.' [[Εργαλειοθήκες (Toolboxes)]] <br>
'3.' [[Περιβάλλον Εργασίας]] <br>
'4.' [[Η Γλώσσα Προγραμματισμού Matlab]] <br>
'5.' [[Γραφικές Παραστάσεις]] <br>
'6.' [[Ανάπτυξη Γραφικών Εφαρμογών]] <br>
'7.' [[NEURAL NETWORK TOOLKIT]]<br>
</small>
</div>
</div>
```

Οι πρώτες γραμμές προσδιορίζουν το μέγεθος, τη μορφή και τη θέση του div μέσα στη σελίδα. Ακολουθεί το περιεχόμενο του, το οποίο περικλείεται στην καρτέλα <small></small> του wiki με σκοπό το menu να εμφανίζεται με μικρότερα γράμματα απ' αυτά της εκάστοτε σελίδας.



Εικόνα 20: menu περιεχομένων μέσα στο wiki 1

Για μεγαλύτερη ευελιξία στη χρήση του wiki μας, δώσαμε τη δυνατότητα στο menu (div που προσθέσαμε παραπάνω) να εμφανίζεται και να εξαφανίζεται όταν επιθυμεί ο χρήστης. Αυτό το πετύχαμε προσθέτοντας τον κατάλληλο κώδικα στο αρχείο MonoBook.php που βρίσκεται στο φάκελο skins της βάσης μας.

Οι μέθοδοι που χρησιμοποιήθηκαν φαίνονται παρακάτω:

```
<script type="text/javascript">
function hideDiv(divid) {
    if (document.getElementById) {
document.getElementById(divid).style.visibility = 'hidden';
        }else{
            if (document.layers) {
                document.divid.visibility = 'hidden';
            }else{
                document.all.divid.style.visibility =
'hidden';
            }
        }
    }
}
```

```
function showDiv(divid) {
    if (document.getElementById) {
document.getElementById(divid).style.visibility = 'visible';
        }else{
```

```
        if (document.layers){
            document.divid.visibility =
'visible';
        }else{
            document.all.divid.style.visibility =
'visible';
        }
    }
}
</script>
```

Κατόπιν δημιουργήσαμε δύο κουμπιά στο wiki, ένα για εμφάνιση κι ένα για εξαφάνιση του menu μας. Κάθε φορά που ο χρήστης πατάει σε ένα απ' αυτά τα κουμπιά καλείται και η ανάλογη μέθοδος για εμφάνιση ή εξαφάνιση του menu.

Ο κώδικας για τη δημιουργία των κουμπιών καθώς και η λειτουργία τους παρουσιάζεται παρακάτω.

```
<div id="hideshow" style="font-size:100%; border:2px groove;
background-color:#e0e0e0; font-weight:bold; position: fixed;
right:10px; width: 152px; top:18px; ">
<div style="font-size:100; border:0px solid; background-
color:#D3D3D3 ; ">
```

```
&Pi;&Epsilon;&Rho;&Iota;&Epsilon;&Chi;&Omicron;&Mu;&Epsilon;&
Nu;&Alpha;
```

```
&nbsp;
```

```
<a href="#" onclick="showDiv('hideshow');">&dArr;</a>
```

```
&nbsp;
```

```
<a href="#" onclick="hideDiv('hideshow');">&uArr;</a>
```

```
</div>
```

```
</div>
```

Κι εδώ δημιουργήσαμε ένα div το οποίο περιέχει τα δύο κουμπιά (εμφάνιση/ εξαφάνιση) καθώς και μια ετικέτα που προσδιορίζει τη λειτουργία τους. Για την ετικέτα χρησιμοποιήσαμε ειδικούς συμβολισμούς για κάθε

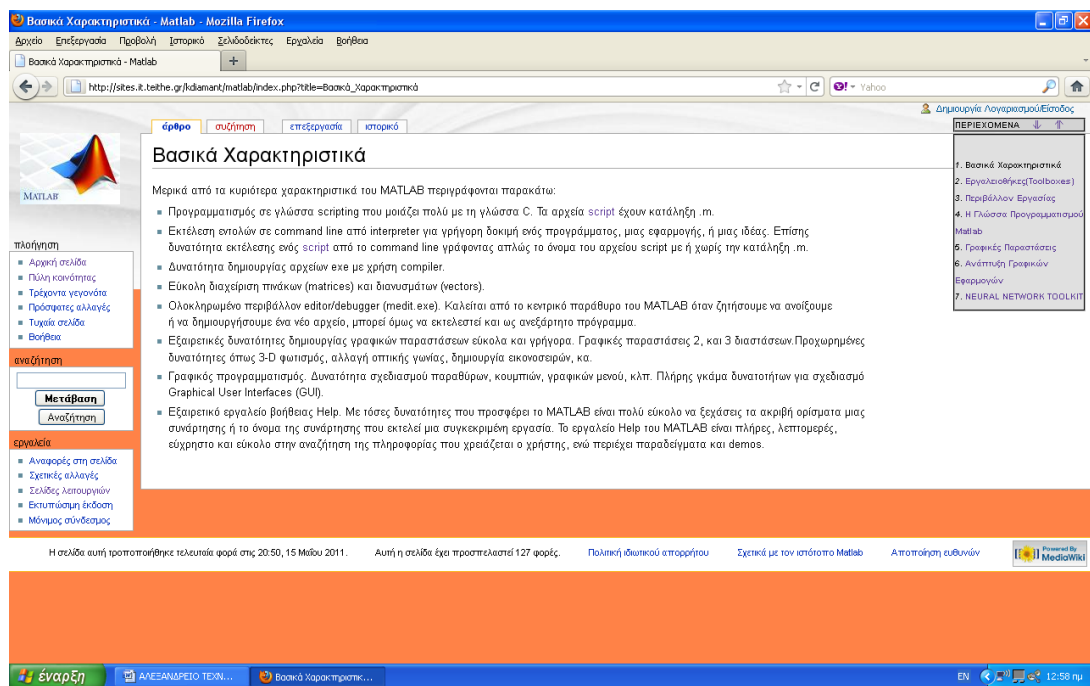
γράμμα ώστε το αποτέλεσμα να είναι η εμφάνιση της ετικέτας γραμμένης στα ελληνικά. Αυτό συμβαίνει γιατί ένα αρχείο γραμμένο σε ρηρ δεν υποστηρίζει τη χρήση ελληνικών χαρακτήρων. Έτσι για να γράψουμε τη λέξη «ΠΕΡΙΕΧΟΜΕΝΑ» χρησιμοποιήσαμε την παρακάτω γραμμή.

`Π Ε Ρ Ι Ε Χ Ο Μ Ε Ν Α`

Επίσης και αυτό το κομμάτι(div) το προσδιορίζουμε με το ίδιο id που προσδιορίζουμε το κομμάτι(div) του menu μας. Αυτό συμβαίνει για να μπορεί η αλληλεπίδραση του ενός να έχει αντίκρισμα στο άλλο.



Εικόνα 21: κουμπία εμφάνισης/εξαφάνισης του menu



Εικόνα 22 Μορφή του wiki μαζί με το προστιθέμενο menu

5 Matlab-Wiki

5.1 ΕΙΣΑΓΩΓΗ

Το MATLAB είναι ένα σύγχρονο ολοκληρωμένο μαθηματικό λογισμικό πακέτο που χρησιμοποιείται σε πανεπιστημιακά μαθήματα αλλά και ερευνητικές και άλλες εφαρμογές με επιστημονικούς υπολογισμούς καθώς περιέχει ένα σύνολο συναρτήσεων και εξωτερικών βιβλιοθηκών ([Εργαλειοθήκες\(Toolboxes\)](#)) για εφαρμογές όπως η στατιστική ανάλυση δεδομένων κ.α. Το όνομά του προέρχεται από τα αρχικά γράμματα των λέξεων **MAT**rix **LAB**oratory καθώς αποθηκεύει και κάνει τις πράξεις με βάση την **άλγεβρα μήτρων**. Το MATLAB είναι ένα διαδραστικό πρόγραμμα για αριθμητικούς υπολογισμούς και οπτικοποίηση δεδομένων με δυνατότητες προγραμματισμού που το καθιστούν ένα ισχυρό και χρήσιμο εργαλείο στις μαθηματικές και φυσικές επιστήμες.

Όπως υποδηλώνεται και από το όνομά του, το MATLAB είναι ειδικά σχεδιασμένο για υπολογισμούς με πίνακες, όπως η επίλυση γραμμικών συστημάτων, η εύρεση ιδιοτιμών και ιδιοδιανυσμάτων, η αντιστροφή τετραγωνικών πινάκων κλπ. Επιπλέον το πακέτο αυτό είναι εφοδιασμένο με πολλές επιλογές για γραφικά (δηλ. την κατασκευή γραφικών παραστάσεων) και προγράμματα γραμμένα στη δική του γλώσσα προγραμματισμού για την επίλυση άλλων προβλημάτων όπως η εύρεση των ριζών μη γραμμικής εξίσωσης, η επίλυση μη γραμμικών συστημάτων, η επίλυση προβλημάτων αρχικών τιμών με συνήθεις διαφορικές εξισώσεις κα.

Η γλώσσα προγραμματισμού του MATLAB δίνει την ευχέρεια στον χρήστη να το επεκτείνει με δικά του προγράμματα. Το MATLAB είναι σχεδιασμένο για την αριθμητική επίλυση προβλημάτων σε αριθμητική πεπερασμένης ακρίβειας (finite-precision arithmetic), δηλαδή δεν βρίσκει την ακριβή αλλά μια προσεγγιστική λύση ενός προβλήματος. Αυτή είναι και η βασική του διαφορά από τα συστήματα συμβολικών υπολογισμών όπως η Maple και το Mathematica.

5.2 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Μερικά από τα κυριότερα χαρακτηριστικά του MATLAB περιγράφονται παρακάτω:

Προγραμματισμός σε γλώσσα scripting που μοιάζει πολύ με τη γλώσσα C.

Τα αρχεία [script](#) έχουν κατάληξη .m.

- Εκτέλεση εντολών σε command line από interpreter για γρήγορη δοκιμή ενός προγράμματος, μιας εφαρμογής, ή μιας ιδέας. Επίσης δυνατότητα εκτέλεσης ενός [script](#) από το command line γράφοντας απλώς το όνομα του αρχείου script με ή χωρίς την κατάληξη .m.
- Δυνατότητα δημιουργίας αρχείων exe με χρήση compiler.
- Εύκολη διαχείριση πινάκων (matrices) και διανυσμάτων (vectors).
- Ολοκληρωμένο περιβάλλον editor/debugger (medit.exe). Καλείται από το κεντρικό παράθυρο του MATLAB όταν ζητήσουμε να ανοίξουμε ή να δημιουργήσουμε ένα νέο αρχείο, μπορεί όμως να εκτελεστεί και ως ανεξάρτητο πρόγραμμα.
- Εξαιρετικές δυνατότητες δημιουργίας γραφικών παραστάσεων εύκολα και γρήγορα. Γραφικές παραστάσεις 2, και 3 διαστάσεων. Προχωρημένες δυνατότητες όπως 3-D φωτισμός, αλλαγή οπτικής γωνίας, δημιουργία εικονοσειρών, κα.
- Γραφικός προγραμματισμός. Δυνατότητα σχεδιασμού παραθύρων, κουμπιών, γραφικών μενού, κλπ. Πλήρης γκάμα δυνατοτήτων για σχεδιασμό Graphical User Interfaces (GUI).
- Εξαιρετικό εργαλείο βοήθειας Help. Με τόσες δυνατότητες που προσφέρει το MATLAB είναι πολύ εύκολο να ξεχάσεις τα ακριβή ορίσματα μιας συνάρτησης ή το όνομα της συνάρτησης που εκτελεί μια συγκεκριμένη εργασία. Το εργαλείο Help του MATLAB είναι πλήρες, λεπτομερές, εύχρηστο και εύκολο στην αναζήτηση της πληροφορίας που χρειάζεται ο χρήστης, ενώ περιέχει παραδείγματα και demos.

5.3 ΕΡΓΑΛΕΙΟΘΗΚΕΣ(TOOLBOXES)

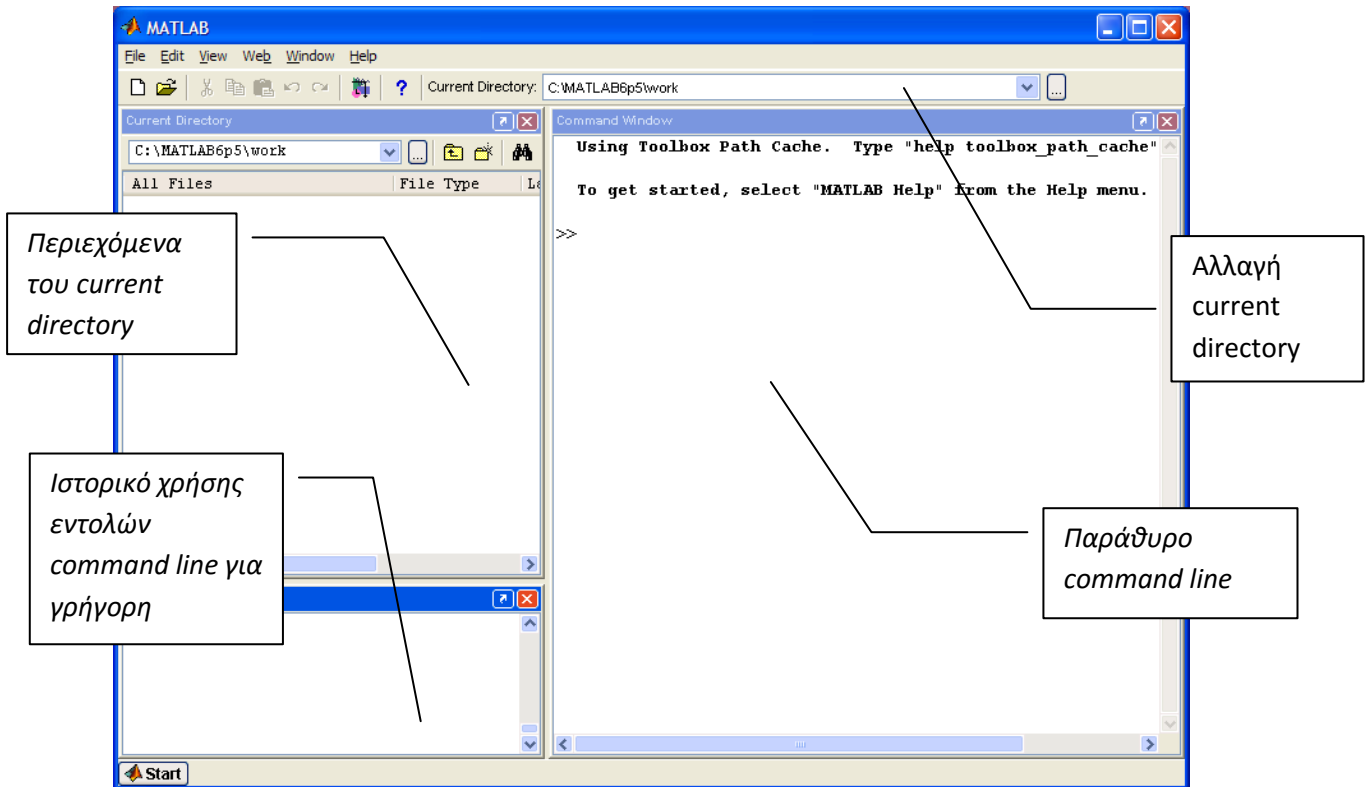
Το MATLAB περιέχει μεγάλο πλήθος έτοιμων βιβλιοθηκών. Οι standard βιβλιοθήκες προσφέρουν έτοιμες συναρτήσεις για επεξεργασία αριθμών, διανυσμάτων, πινάκων, δημιουργία plots, κλπ. Υπάρχει επίσης μεγάλη σειρά ειδικών βιβλιοθηκών που λέγονται εργαλειοθήκες (toolboxes) και ειδικεύονται σε μια επιστημονική περιοχή. Χαρακτηριστικά τέτοια toolboxes είναι:

- Signal processing toolbox
- Image processing toolbox
- Neural networks toolbox
- Fuzzy Logic toolbox
- Statistics toolbox
- Optimization toolbox
- Communications toolbox
- Virtual Reality toolbox
- Database toolbox
- Control toolbox
- Symbolic math toolbox
- Financial toolbox
- Mapping toolbox
- Wavelet toolbox

και πολλά άλλα.

5.4 ΠΕΡΙΒΑΛΛΟΝ ΕΡΓΑΣΙΑΣ

Η παρακάτω δείχνει το βασικό παράθυρο εργασίας του MATLAB. Εκτός από τα εικονιζόμενα παράθυρα μπορεί κανείς να ανοίξει και άλλα χρησιμοποιώντας το μενού *View*. Άλλα χρήσιμα παράθυρα είναι τα *Workspace*, *Help*, και *Profiler*. Στο παράθυρο *Workspace* βλέπουμε τις μεταβλητές που έχουμε ορίσει και το χώρο που καταλαμβάνουν στη μνήμη.



Εικόνα 23: παράθυρο matlab

Ο *Profiler* είναι ένα χρήσιμο εργαλείο για τη βελτιστοποίηση του κώδικά μας. Για ένα συγκεκριμένο πρόγραμμα (script) μας δείχνει πόσος χρόνος CPU σπαταλήθηκε από κάθε γραμμή του κώδικα πόσες φορές εκτελέστηκε κάθε loop, κλπ. Το παράθυρο *Help* είναι ένα ολοκληρωμένο περιβάλλον αναζήτησης πληροφορίας μέσα στο πλήθος των εργαλειοθηκών, των εντολών, και των συναρτήσεων του MATLAB. Μέσα από το παράθυρο αυτό μπορεί κανείς να αναζητήσει πληροφορία

- κάνοντας πλοήγηση στον πίνακα περιεχομένων (*Contents*)
- μέσα από τον ονομαστικό κατάλογο των εντολών/συναρτήσεων (*Index*),
- μέσα από αναζήτηση με λέξεις κλειδιά (*Search*)
- μέσα από προγράμματα επίδειξης (*Demos*)

5.5 Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ MATLAB

Το MATLAB έρχεται εξοπλισμένο με μια γλώσσα προγραμματισμού που μοιάζει πολύ με τη γλώσσα C. Η γλώσσα αυτή μπορεί να χρησιμοποιηθεί είτε στο command line για να γράφουμε μια-μια τις εντολές που θέλουμε να εκτελέσουμε είτε σε ένα αρχείο script με την κατάληξη .m. Με το που ξεκινάμε το MATLAB στο παράθυρο command line εμφανίζεται το prompt

>>

έτοιμο να δεχτεί τις εντολές που θα δακτυλογραφήσουμε σε αυτό.

5.5.1 Βασικές πράξεις

Το MATLAB μπορεί να χρησιμοποιηθεί σαν μια απλή αριθμομηχανή χρησιμοποιώντας τις τέσσερις πράξεις της αριθμητικής(πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση) και την πράξη της ύψωσης σε δύναμη όπως φαίνονται παρακάτω, μαζί με ενδεικτικά παραδείγματα.

Σύμβολο	Αριθμητική πράξη	Παραδείγματα
+	Πρόσθεση	3+4=7
-	Αφαίρεση	1-10=-9
*	Πολλαπλασιασμός	3*4=12
\ ή /	Διαίρεση	3\15=3 ή 15/3=3
^	Ύψωση σε δύναμη	3^2=9

```

>> 3+4
ans =
     7
>>
>> 1-10, 3*4
ans =
    -9
     12
>> 3\15, 15/3
ans =
     5
     5
>> 3^2
ans =
     9
    
```

Εικόνα 24: Βασικές πράξεις matlab μέσα στο wiki

Το MATLAB μπορεί να υπολογίσει οποιαδήποτε αριθμητική παράσταση με μια μόνο εντολή. Η σειρά των αριθμητικών πράξεων μπορεί να καθοριστεί με παρενθέσεις όπως ακριβώς είναι γνωστό από τα Μαθηματικά. Η προτεραιότητα των πράξεων στο MATLAB είναι ανάλογη με τους κανόνες προτεραιότητας των πράξεων της άλγεβρα.

5.5.2 Η εντολή καταχώρησης

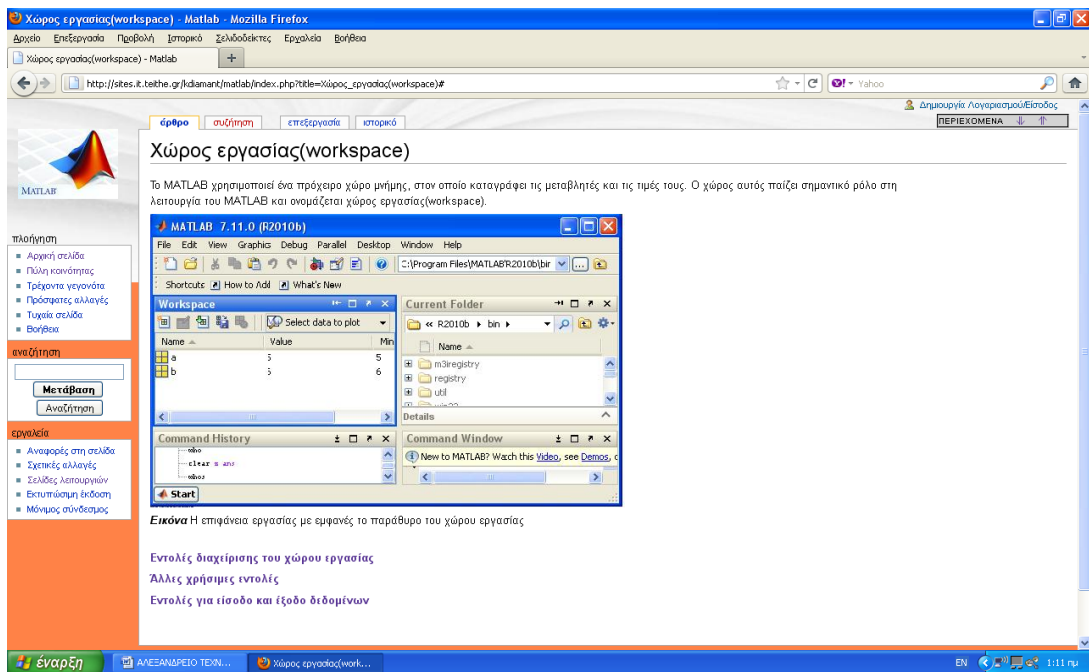
Όσο αφορά στους υπολογισμούς το MATLAB έχει στην ουσία μια μόνο εντολή, την εντολή καταχώρησης. Η γενική μορφή της εντολής είναι

`όνομα_μεταβλητης = τιμή ή μεταβλητή ή αποτέλεσμα πράξεων`

Το όνομα μεταβλητής δεν είναι τίποτε άλλο από μια θέση μνήμης στην οποία καταγράφεται η τιμή. Όπως και στην C, οι μεταβλητές μπορούν να έχουν διάφορους τύπους, για παράδειγμα double, char, string, logical, κλπ.

5.5.3 Ο χώρος εργασίας(workspace)

Το MATLAB χρησιμοποιεί ένα πρόχειρο χώρο μνήμης, στον οποίο καταγράφει τις μεταβλητές και τις τιμές τους. Ο χώρος αυτός παίζει σημαντικό ρόλο στη λειτουργία του MATLAB και ονομάζεται χώρος εργασίας(workspace).



Εικόνα 25: Χώρος εργασίας matlab μέσα στο wiki

5.5.4 Matlab Search Path

Το Matlab search path είναι τα directories στα οποία ψάχνει το MATLAB για υπάρχουσες μεταβλητές, functions, scripts, M-Files, αρχεία και άλλες δομές του MATLAB. Η σειρά κατά την οποία ψάχνει είναι η παρακάτω.

1. Μεταβλητές
2. Subfunctions
3. Private Functions
4. Class Constructors
5. Overloaded Methods
6. M-File που βρίσκεται στον παρόν κατάλογο
7. Τέλος, M-File που βρίσκεται στο path ή build-in functions του MATLAB

5.5.5 Διανύσματα, πίνακες, cells

Στις αρχικές εκδοχές της MATLAB (μέχρι και την εκδοχή 3) κάθε μεταβλητή ήταν ένας διδιάστατος, δηλ. $m \times n$, πίνακας με (μιγαδικούς) αριθμούς διπλής ακρίβειας. Τα διανύσματα και οι αριθμοί ήταν απλώς ειδικές περιπτώσεις ($1 \times n$ ή $n \times 1$ και 1×1 πίνακες αντίστοιχα). Από την εκδοχή 4, η MATLAB δέχεται επιπλέον n -διάστατους πίνακες καθώς και άλλους τύπους δεδομένων. Οι νέοι αυτοί τύποι δεδομένων περιλαμβάνουν τις **δομές (structures)**, τις **τάξεις (classes)** και τους **πίνακες κελίων (cell arrays)**, οι οποίοι είναι πίνακες με στοιχεία όχι αναγκαστικά του ίδιου τύπου. Για παράδειγμα σ' ένα μονοδιάστατο πίνακα, το πρώτο στοιχείο μπορεί να είναι ένας αριθμός, το δεύτερο ένα αλφαριθμητικό (string) , το τρίτο ένα διάνυσμα κοκ.

5.5.6 Δομές ελέγχου ροής

Οι δομές ελέγχου ροής του προγράμματος είναι σύνολα εντολών που έχουν ειδικό τρόπο σύνταξης. Όπως και σε άλλες γλώσσες έτσι και η MATLAB διαθέτει τέσσερις βασικές **δομές ελέγχου ροής**:

- **εντολή if** (if statement)
- **εντολή switch** (switch statement)
- **βρόχοι for** (for loops)
- **βρόχοι while** (while loops)

Για να μπορούν να λειτουργήσουν αυτές οι δομές πρέπει να ικανοποιούνται μερικές συνθήκες. Για παράδειγμα είναι εφικτή η υλοποίηση της πρότασης "αν ο αριθμός x ικανοποιεί τη σχέση $x > 3$, τότε προστίθενται οι αριθμοί 8 και 9". Για την περιγραφή των συνθηκών, που ικανοποιούνται ή

όχι, συνδυάζονται **λογικοί τελεστές** (logical operators) και **σχεσιακοί τελεστές** (relational operators).

5.5.7 Δομές (Structures)

Το Matlab δίνει τη δυνατότητα στον χρήστη να δημιουργήσει δικούς του τύπους δεδομένων προσαρμοσμένους στις ανάγκες του εκάστοτε προγράμματος. Οι δομές δημιουργούνται με μεγάλη ευκολία στο Matlab με τη χρήση της εντολής `struct()` η οποία παίρνει ως ορίσματα το όνομα του πεδίου και δίπλα την τιμή του πεδίου.

Σύνταξη εντολής: `A = struct('Πεδίο_1', {}, 'Πεδίο_2', {})`

Τα ορίσματα αυτά μπορεί να είναι πολλαπλά ανάλογα με τα πεδία που έχουμε σε κάθε δομή. Τα πεδία είναι απλά ονόματα για τις τιμές. Οι τιμές μπορεί να είναι και πίνακες.

5.5.8 Συναρτήσεις (Functions)

Οι συναρτήσεις περιέχουν μια ολοκληρωμένη ακολουθία εντολών της MATLAB με μεταβλητές εισόδου,

`input1, input2, ...`

με την οποία υπολογίζονται νέες μεταβλητές εξόδου

`output1, output2, ...`

Η δομή τους είναι η εξής:

[1. Επικεφαλίδα \(header\)](#) Μια γραμμή της μορφής:

`function [output1,output2,...]= filename (input1,input2,...)`

Το filename είναι το όνομα της function το οποίο αποθηκεύεται (υποχρεωτικά) στο m-file με όνομα filename.m. Προσέξτε ότι οι μεταβλητές εισόδου είναι σε παρενθέσεις ενώ οι μεταβλητές εξόδου βρίσκονται σε αγκύλες. Οι τελευταίες δεν είναι απαραίτητες αν έχουμε μόνο μια μεταβλητή εξόδου.

Για παράδειγμα αντί

```
function [out1] = parad(x, y)
```

μπορούμε να γράψουμε

```
function out1 = parad(x, y)
```

!!! Οι συναρτήσεις βιβλιοθήκης της MATLAB, όπως οι `sin`, `max`, `exp` και `size`, είναι `function m-files`. **Το όνομα μιας νέας `function` δεν πρέπει να συμπίπτει με όνομα συνάρτησης βιβλιοθήκης.**

2. Σχόλια (comments)

Αυτά ξεκινούν αναγκαστικά με το σύμβολο `%` και είναι προαιρετικά. Τα σχόλια μετά την επικεφαλίδα αποτελούν και το κείμενο βοήθειας για το `function`. Με διαφορετικά λόγια, αυτά εμφανίζονται αν γράψουμε:

```
>> help filename
```

3. Εντολές (statements)

Ακολουθία εντολών της MATLAB με την οποία υπολογίζονται οι μεταβλητές εξόδου

Προκαταρκτικές εντολές

```
output1 = ...
```

```
output2 = ...
```

κλπ

4. Υποσυναρτήσεις (subfunctions)

Αυτές είναι εσωτερικές συναρτήσεις `functions` που περιέχονται στο αρχείο `filename.m`. Αναγνωρίζονται μόνο από τα `functions` που περιέχονται στο αρχείο αυτό. Ορίζονται με τον ίδιο ακριβώς τρόπο, π.χ.

```
function [out1, out2,] = subfname (x, y, z)
```

5.5.9 Επεξεργασία αρχείων

Πολλές φορές είναι αναγκαίο τα εξαγόμενα από μια πράξη ή από ένα σύνολο πράξεων να τα κρατάμε, ώστε να μπορούμε να τα επεξεργαστούμε. Το Matlab δίνει τη δυνατότητα να επεξεργαστούμε τόσο απλά **δυσάδικα αρχεία** όσο και **αρχεία κειμένων**.

5.5.10 Interpreter και Compiler

Η γραμμή εντολών (*command line*) λειτουργεί ως *interpreter* των εντολών που γράφουμε εκεί. Κάθε εντολή που δακτυλογραφούμε μεταφράζεται σε γλώσσα μηχανής και εκτελείται μόλις πατήσουμε <ENTER>. Με την ίδια ακριβώς λογική λειτουργούν και τα προγράμματα *script*, δηλαδή είναι σαν να γράφαμε στο *command line* του *MATLAB* κάθε εντολή που είναι γραμμένη στο αρχείο *script* και πατούσαμε το <ENTER>. Έτσι κάθε εντολή μεταφράζεται και εκτελείται ξανά και ξανά ακόμη και εάν βρίσκεται σε ένα βρόχο. Αυτό δημιουργεί καθυστερήσεις στους βρόχους και είναι λιγότερο αποδοτική μέθοδος σε σχέση με την μετάφραση του κώδικα μια φορά στην αρχή και την δημιουργία *binary* εκτελέσιμου αρχείου τύπου *exe*. Η μέθοδος του *interpreter* είναι βέβαια ιδανική όταν θέλουμε να τεστάρουμε γρήγορα μια ιδέα ή έναν αλγόριθμο χωρίς να χρειάζεται κάθε φορά να μεσολαβεί το χρονοβόρο στάδιο της μετάφρασης και της εκτέλεσης του εκτελέσιμου αρχείου.

Το *MATLAB* προσφέρει έναν **compiler με το όνομα *mcc*** οποίος δημιουργεί εκτελέσιμο κώδικα σε μορφή αρχείου ***exe***.

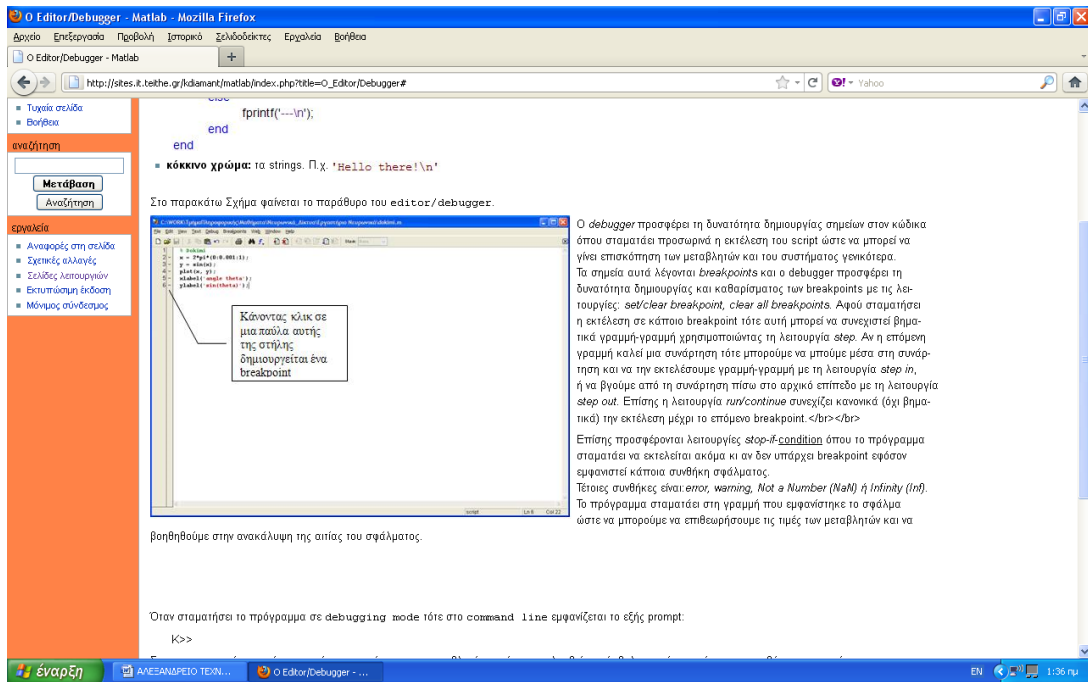
5.5.11 Ο Editor/Debugger

Ο *editor/debugger* είναι ένα ολοκληρωμένο περιβάλλον επεξεργασίας *scripts*. Προσφέρει αυτόματη μορφοποίηση του κώδικα που γράφετε ενώ με διαφορετικά χρώματα καταδεικνύονται οι διαφορετικές εντολές:

- **πράσινο χρώμα:** σχόλια. Ό,τι ακολουθεί το σύμβολο % σε μια γραμμή θεωρείται σχόλιο. Π.χ. `%Dokimi`
- **μπλε χρώμα:** εντολές *built-in*. Π.χ, οι εντολές `for`, `if`, `else`, και `end`

```
for (i=1:10)
    if (a(i) == 1)
        fprintf('Hello there!\n');
    else
        fprintf('---\n');
    end
end
```

- **κόκκινο χρώμα:** τα *strings*. Π.χ. `'Hello there!\n'`



Εικόνα 26: Editor/Debugger matlab μέσα στο wiki

Ο *debugger* προσφέρει τη δυνατότητα δημιουργίας σημείων στον κώδικα όπου σταματάει προσωρινά η εκτέλεση του script ώστε να μπορεί να γίνει επισκόπηση των μεταβλητών και του συστήματος γενικότερα. Τα σημεία αυτά λέγονται *breakpoints* και ο debugger προσφέρει τη δυνατότητα δημιουργίας και καθαρίσματος των breakpoints με τις λειτουργίες: *set/clear breakpoint*, *clear all breakpoints*. Αφού σταματήσει η εκτέλεση σε κάποιο breakpoint τότε αυτή μπορεί να συνεχιστεί βηματικά γραμμή-γραμμή χρησιμοποιώντας τη λειτουργία *step*. Αν η επόμενη γραμμή καλεί μια συνάρτηση τότε μπορούμε να μπούμε μέσα στη συνάρτηση και να την εκτελέσουμε γραμμή-γραμμή με τη λειτουργία *step in*, ή να βγούμε από τη συνάρτηση πίσω στο αρχικό επίπεδο με τη λειτουργία *step out*. Επίσης η λειτουργία *run/continue* συνεχίζει κανονικά (όχι βηματικά) την εκτέλεση μέχρι το επόμενο breakpoint.

Επίσης προσφέρονται λειτουργίες *stop-if-condition* όπου το πρόγραμμα σταματάει να εκτελείται ακόμα κι αν δεν υπάρχει breakpoint εμφανιστεί κάποια συνθήκη σφάλματος. Τέτοιες συνθήκες είναι: *error*, *warning*, *Not a Number (NaN)* ή *Infinity (Inf)*. Το πρόγραμμα σταματάει στη γραμμή που εμφανίστηκε το σφάλμα ώστε να μπορούμε να επιθεωρήσουμε τις τιμές των μεταβλητών και να βοηθηθούμε στην ανακάλυψη της αιτίας του σφάλματος.

Επίσης προσφέρονται λειτουργίες *stop-if-condition* όπου το πρόγραμμα σταματάει να εκτελείται ακόμα κι αν δεν υπάρχει breakpoint εφόσον εμφανιστεί κάποια συνθήκη σφάλματος. Τέτοιες συνθήκες είναι: ***error***, ***warning***, ***Not a Number (NaN)*** ή ***Infinity (Inf)***. Το πρόγραμμα σταματάει στη γραμμή που εμφανίστηκε το σφάλμα ώστε να μπορούμε να επιθεωρήσουμε τις τιμές των μεταβλητών και να βοηθηθούμε στην ανακάλυψη της αιτίας του σφάλματος.

Όταν σταματήσει το πρόγραμμα σε debugging mode τότε στο command line εμφανίζεται το εξής prompt:

```
K>>
```

Στο prompt αυτό μπορούμε να γράψουμε το όνομα μιας μεταβλητής χωρίς να ακολουθεί το σύμβολο ; οπότε τυπώνεται στην οθόνη το περιεχόμενο της μεταβλητής αυτής. Πχ,

```
K>> x
```

```
x =
```

```
3
```

```
K>> y
```

```
y =
```

```
1 -2 3 0.1 5
```

Με την εντολή return επιστρέφουμε στην κανονική εκτέλεση του προγράμματος (βγαίνουμε από τον debugger).

5.6 ΓΡΑΦΙΚΕΣ ΠΑΡΑΣΤΑΣΕΙΣ

Η MATLAB έχει εξαιρετικές δυνατότητες για γραφικά και είναι εφοδιασμένη με αρκετές συναρτήσεις για εύκολο και ευέλικτο σχεδιασμό επίπεδων καμπυλών, τρισδιάστατων επιφανειών, ισούψων, παραμετρικών δισδιάστατων αλλά και τρισδιάστατων καμπυλών κα.

5.6.1 Γραφικές Παραστάσεις 2 Διαστάσεων

5.6.1.1 Plot

Η βασική συνάρτηση για 2Δ γραφικές παραστάσεις είναι η plot. Η plot σαν κύριο όρισμα παίρνει δύο vectors ή matrices με σημεία στον 2Δ χώρο και σχεδιάζει την γραφική παράσταση που ορίζουν τα σημεία. Η συνάρτηση plot χρησιμοποιείται για την κατασκευή του γραφήματος μιας επίπεδης καμπύλης τα σημεία της είναι αποθηκευμένα στα ισομήκη διανύσματα x και y.

παράδειγμα:

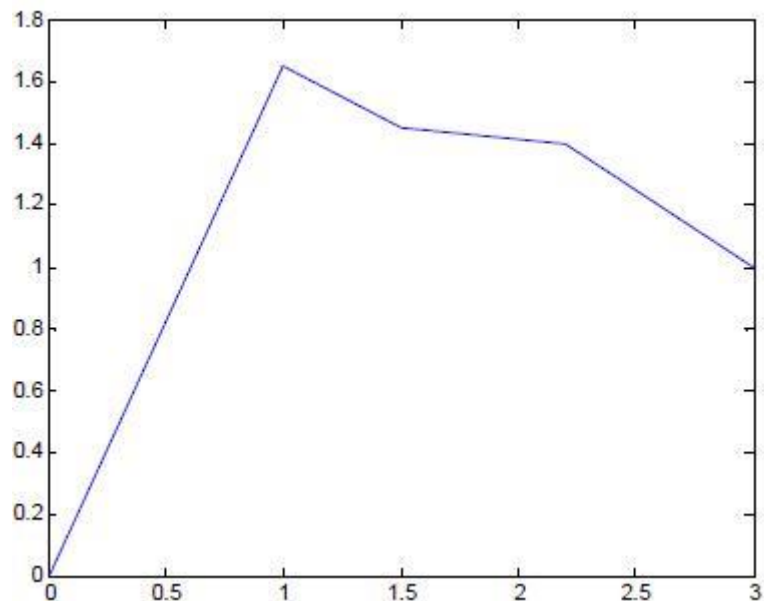
Αν έχουμε τα διανύσματα

```
>> x=[0 1 1.5 2.2 3.]  
x =  
    0    1.0000    1.5000    2.2000    3.0000  
>> y=[0 1.65 1.45 1.4 1]  
y =  
    0    1.6500    1.4500    1.4000    1.0000
```

με την εντολή

```
>> plot(x,y)
```

εμφανίζεται αυτόματα ένα παράθυρο γραφικών με το πιο κάτω γράφημα:



Εικόνα 27: Γραφική Παράσταση Plot

Παρατηρούμε ότι η plot ενώνει τα 5 σημεία που ορίζουν τα x και y με συνεχείς γραμμές.

5.6.1.2 Ezplot

Η συνάρτηση Ezplot καθώς και η συνάρτηση Fplot παράγουν πιο εύκολα απ' ότι η plot τη γραφική παράσταση της $y = f(x)$, ειδικά αν η f είναι συνάρτηση βιβλιοθήκης ή ανώνυμη συνάρτηση ή έχει οριστεί μέσω της εντολής *inline*. Η συνάρτηση ezplot (easy plot) έχει σημαντικά πλεονεκτήματα σε σχέση με τις plot και fplot:

- βρίσκει αυτόματα τα διαστήματα των αξόνων και δίνει επίσης τη δυνατότητα στο χρήστη να τα επιλέξει ο ίδιος,
- μπορεί να κάνει το γράφημα πεπλεγμένης συνάρτησης, και
- μπορεί να κάνει το γράφημα παραμετρικής καμπύλης.

Αν $f(x)$ είναι μια συνάρτηση βιβλιοθήκης ή συνάρτηση ορισμένη από το χρήστη, τότε

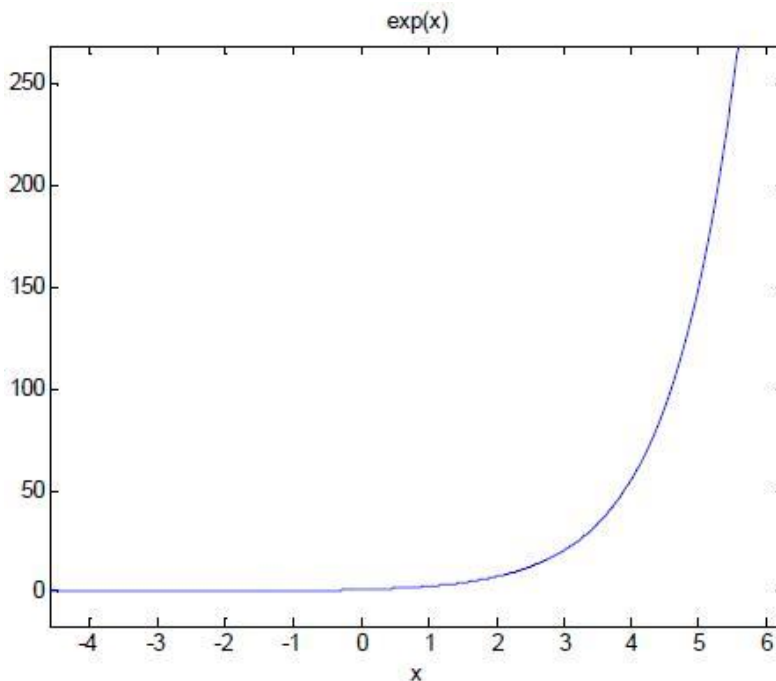
- η `ezplot(f)` παράγει το γράφημα της f επιλέγοντας τα διαστήματα των αξόνων των x και y .
Σύμφωνα με το εγχειρίδιο της MATLAB η προεπιλογή για το x είναι $-2\pi < x < 2\pi$.
- η `ezplot(f, a, b)` ή η `ezplot(f, [a, b])` παράγει το γράφημα της f έτσι ώστε οι τιμές του x να βρίσκονται στο $[a, b]$.

παράδειγμα:

Με την εντολή

```
>> ezplot('exp(x)')
```

παίρνουμε το γράφημα



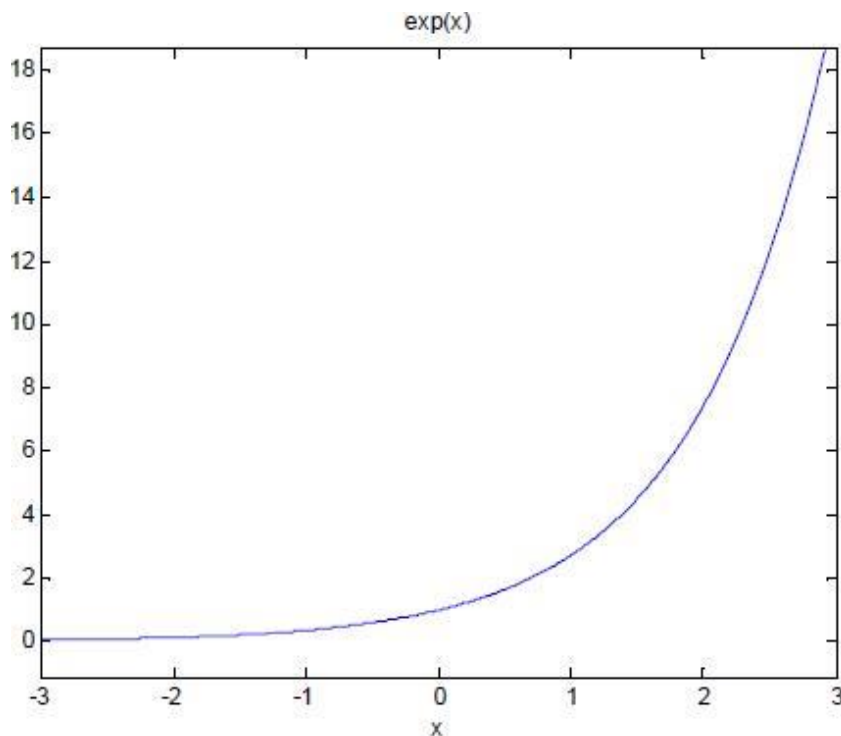
Εικόνα 28: Γραφική Παράσταση Ezplot

Παρατηρούμε ότι η η συνάρτηση προσθέτει επίσης και τίτλο στο γράφημα καθώς και ετικέτα στον άξονα των x.

Αν επιλέξουμε τώρα το διάστημα $[-3, 3]$ για τις τιμές του x,

```
>> ezplot('exp(x)', -3, 3)
```

παίρνουμε το γράφημα



Εικόνα 29: Γραφική Παράσταση Plot

5.6.1.3 Fplot

Σε αντίθεση με την ezplot, στη συνάρτηση **fplot** είναι υποχρεωτική η εισαγωγή του πεδίου σχεδίασης του γραφήματος μιας συνάρτησης. Δύο χαρακτηριστικές περιπτώσεις είναι οι εξής:

- η **fplot(f, [xmin, xmax])** παράγει το γράφημα της f με $x_{min} < x < x_{max}$
- η **fplot(f, [xmin, xmax, ymin, ymax])** παράγει το γράφημα της f στο χωρίο που ορίζεται από τις $x_{min} < x < x_{max}$ και $y_{min} < y < y_{max}$.

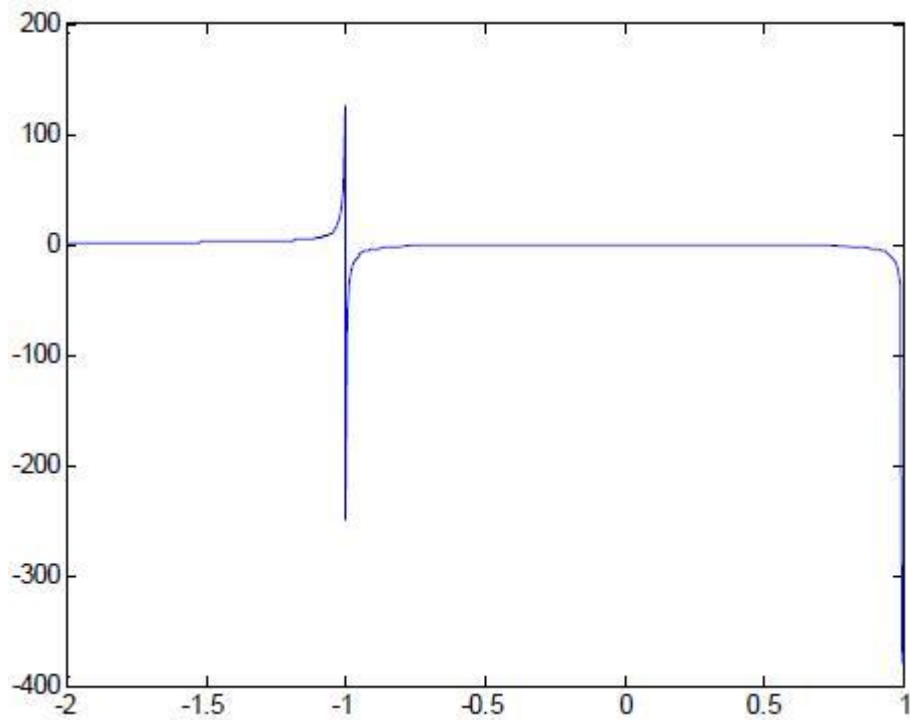
παράδειγμα:

Με τις εντολές

```
>> f = @(x) x.^2./(x.^2-1);  
>> fplot(f, [-2,1])
```

παίρνουμε την όχι και τόσο ευπαρουσίαστη γραφική παράσταση της

$$f(x) = x^2 / x^2 - 1$$

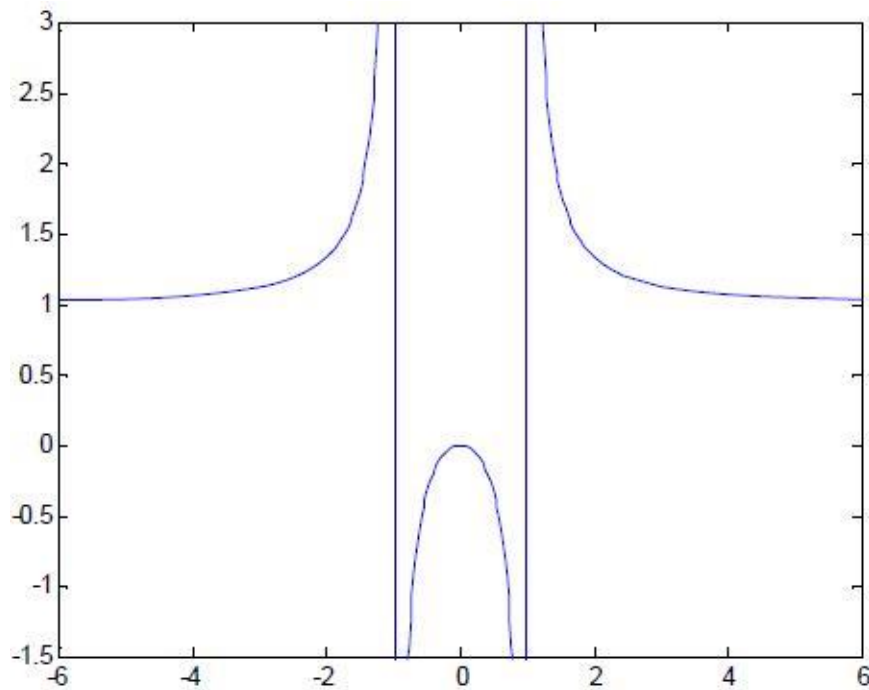


Εικόνα 30: Γραφική Παράσταση Fplot

Αν ορίσουμε το χωρίο του γραφήματος με την εντολή

```
>> fplot(f, [-6 6 -1.5 3])
```

παίρνουμε το εξής γράφημα



Εικόνα 31: Γραφική Παράσταση Fplot

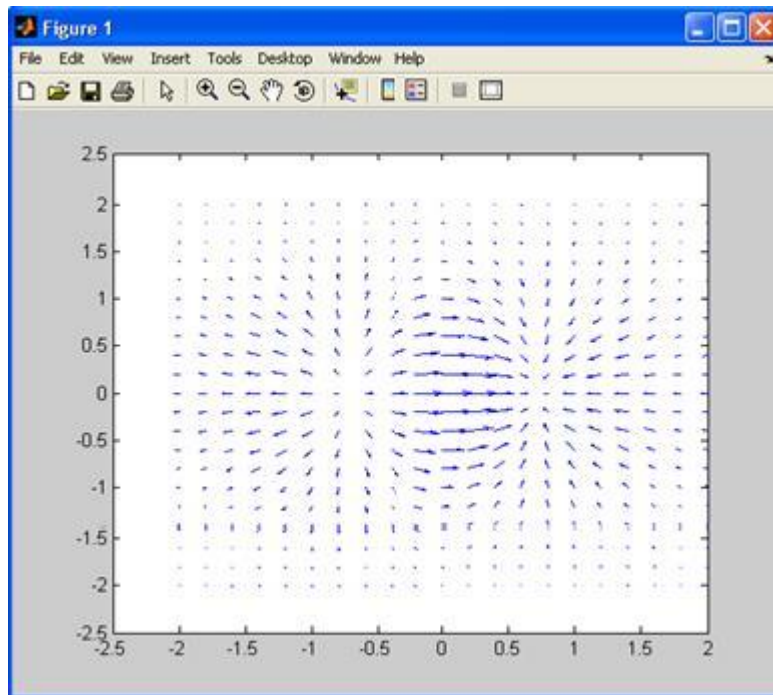
!!!Το γράφημα αυτό διαφέρει από αυτό που παίρνουμε με τη συνάρτηση ezplot

5.6.1.4 Quiver

Η quiver σχεδιάζει ένα γράφημα διανυσμάτων. Το κάθε διάνυσμα αναπαριστάται ως ένα βέλος.

παράδειγμα:

```
[X, Y] = meshgrid(-2:.2:2); %ορισμός πλέγματος
Z = X.*exp(-X.^2 - Y.^2);
[DX, DY] = gradient(Z, .2, .2);
quiver(X, Y, DX, DY)
```



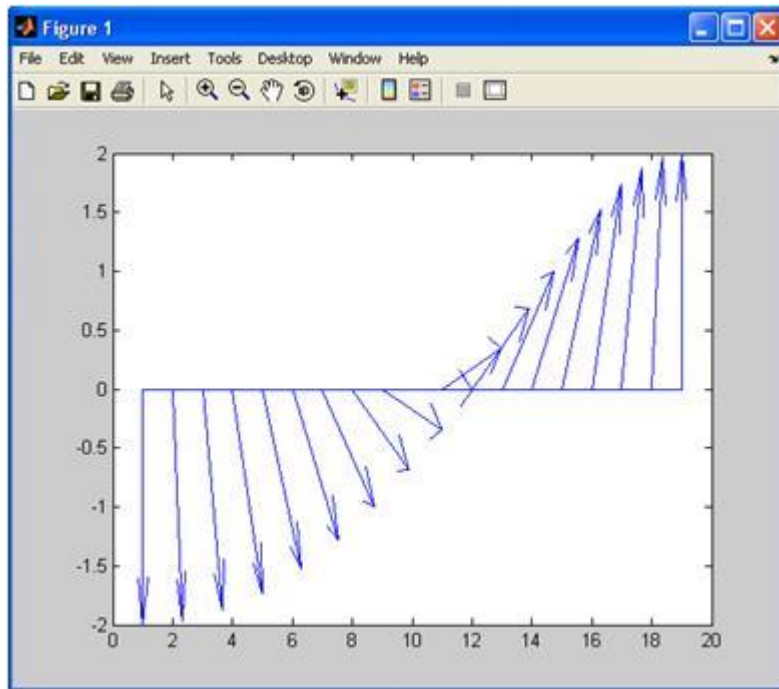
Εικόνα 32: Γραφική Παράσταση Quiver

5.6.1.5 Feather

Η feather σχεδιάζει διανύσματα που προέρχονται σημεία που ισαπέχουν στον οριζόντιο άξονα

παράδειγμα:

```
theta = (-90:10:90)*pi/180;           %κατεύθυνση διανυσμάτων  
r = 2*ones(size(theta));              %μέτρο διανυσμάτων  
[u,v] = pol2cart(theta,r);            %μετατροπή πολικών  
συντεταγμένων σε καρτεσιανών  
feather(u,v);
```



Εικόνα 33: Γραφική Παράσταση Feather

5.6.2 Γραφικές Παραστάσεις 3 Διαστάσεων

Οι παρακάτω εντολές χρησιμοποιούνται στη σχεδίαση της γραφικής παράστασης μιας συνάρτησης δύο μεταβλητών

$$z = f(x, y)$$

Είναι γνωστό ότι η γραφική παράσταση της πιο πάνω συνάρτησης είναι μια επιφάνεια στον τρισδιάστατο χώρο.

5.6.2.1 Ezsurf

Αν η $f(x, y)$ έχει οριστεί σαν ανώνυμη συνάρτηση (ή μέσω της εντολής `inline`), τότε ο πιο εύκολος τρόπος για να πάρουμε την γραφική της παράσταση είναι με την εντολή **ezsurf** (που δουλεύει με ανάλογο τρόπο όπως η `ezplot`).

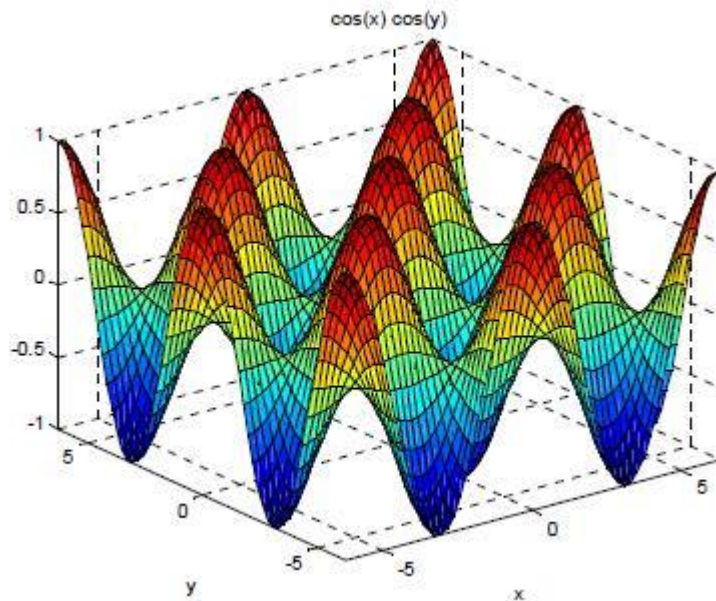
παράδειγμα:

Οι εντολές

```
>> z = @(x,y) cos(x) .* cos(y);
```

```
>> ezsurf(z)
```

δίνουν το πιο κάτω γράφημα:



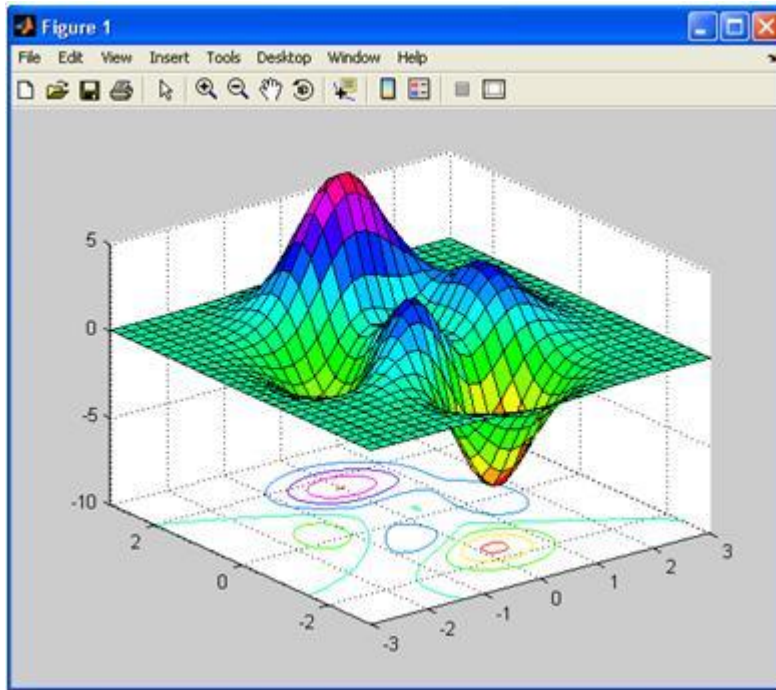
Εικόνα 34: Γραφική Παράσταση Ezsurf

5.6.2.2 Surf

Η **surf** σχεδιάζει την γραφική παράσταση μιας επιφάνειας με την διαφορά ότι η επιφάνεια θα είναι σκιασμένη.

παράδειγμα:

```
[X,Y] = meshgrid(-3:125:3);  
Z = peaks(X,Y);  
surfc(X,Y,Z);  
axis([-3 3 -3 3 -10 5]);
```



Εικόνα 35: Γραφική Παράσταση Surf

5.6.2.3 Meshgrid

Στην περίπτωση που η $f(x, y)$ έχει οριστεί από ένα m-file, ή αν θέλουμε να επιλέξουμε τα σημεία στους άξονες των x και y , τότε πρέπει πρώτα να κατασκευάσουμε ένα πλέγμα στο επίπεδο xy . Η συνάρτηση **meshgrid** δημιουργεί matrices που ορίζουν το πλέγμα των γραφημάτων. Η meshgrid μπορεί να χρησιμοποιηθεί μόνο για δισδιάστατο ή τρισδιάστατο καρτεσιανό χώρο. Παίρνει για ορίσματα δύο (x,y) ή τρία $vectors(x,y,z)$ και επιστρέφει δύο (X,Y) ή αντίστοιχα τρία (X,Y,Z) vectors. Οι γραμμές του X είναι αντίγραφα του x και οι στήλες του Y είναι αντίγραφα του y .

παράδειγμα:

```
[X,Y] = meshgrid(1:3,10:14)
```

X =

1	2	3
1	2	3
1	2	3
1	2	3
1	2	3

Y =

```
10    10    10
11    11    11
12    12    12
13    13    13
14    14    14
```

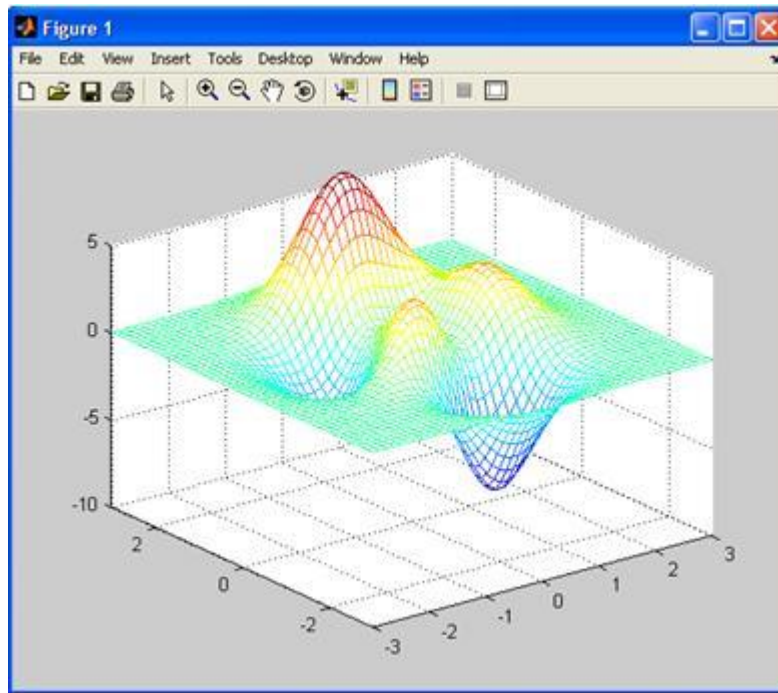
5.6.2.4 Mesh

Η συνάρτηση **mesh** σχεδιάζει την γραφική παράσταση μιας επιφάνειας. Παίρνει τρία ορίσματα, δύο vectors ή matrices που περιέχουν τις συντεταγμένες του οριζόντιου επιπέδου και ένα τρίτο που περιέχει τις συντεταγμένες της επιφάνειας στην τρίτη διάσταση.

παράδειγμα:

```
[X,Y] = meshgrid(-3:125:3);
Z = peaks(X,Y);           %δίνουμε τιμές στα ύψη σε
                           %σχέση με τα x και y

mesh(X,Y,Z);
axis([-3 3 -3 3 -10 5]); %δίνουμε τα όρια των axes
```



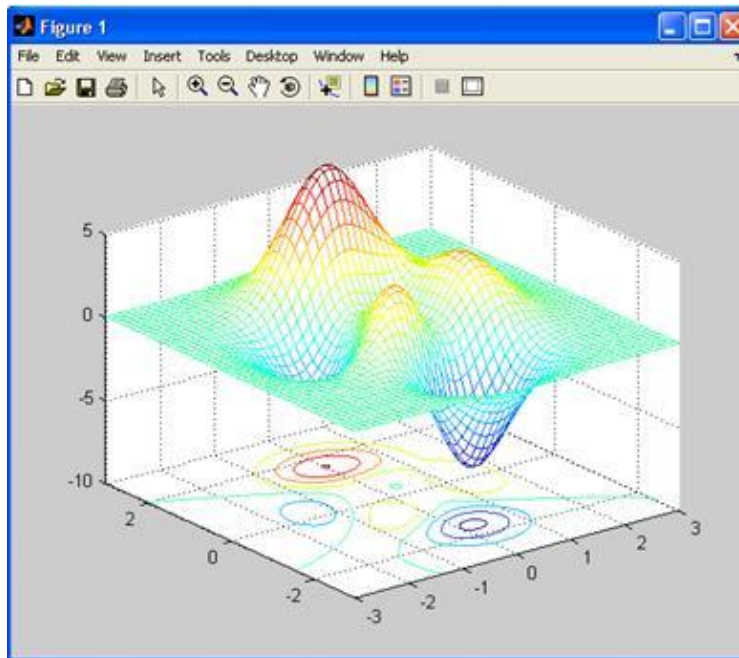
Εικόνα 36: Γραφική Παράσταση Mesh

5.6.2.5 Meshc

Η **meshc** είναι απλά ο συνδυασμός της mesh με την contour. Ο παρακάτω κώδικας δίνει ένα παράδειγμα χρήσης της meshc.

παράδειγμα:

```
[X,Y] = meshgrid(-3:.125:3);  
Z = peaks(X,Y);  
meshc(X,Y,Z);  
axis([-3 3 -3 3 -10 5]);
```

Εικόνα 37: Γραφική Παράσταση Meshc

5.6.2.6 Contour (Ισοϋψείς καμπύλες)

Η συνάρτηση **contour** σχεδιάζει διαγράμματα ισοϋψών. Η σύνταξη της φαίνεται παρακάτω.

```
[C,h] =contour(X,Y,Z,n) ;
```

όπου:

X -- τα x του γραφήματος

Y -- τα y του γραφήματος

Z -- τα ύψη του γραφήματος σχέση με τα x και y

n -- τα επίπεδα του γραφήματος

και επιστρέφει:

C -- το matrix με τα ισοϋψή

h -- το handle για τα contourgroup αντικείμενα του γραφήματος

παράδειγμα:

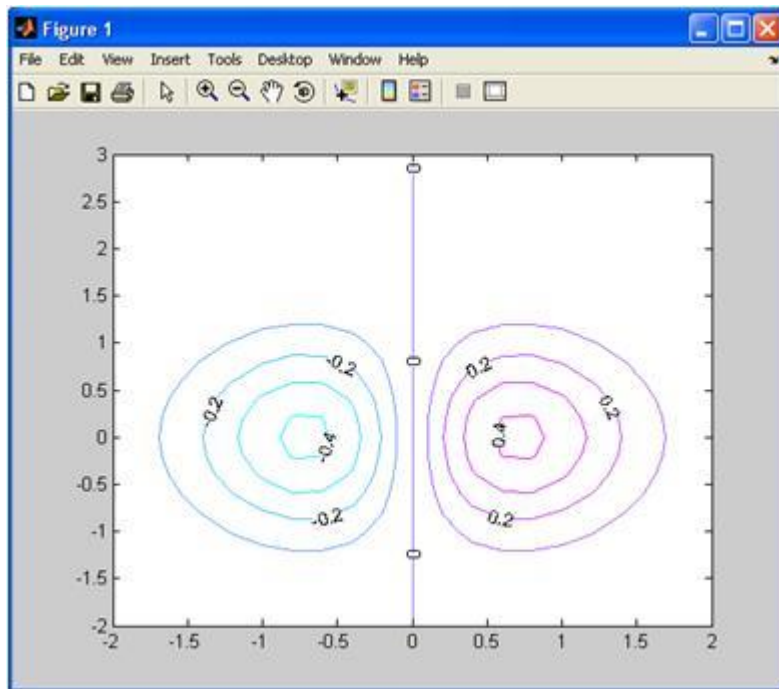
```
[X,Y] = meshgrid(-2:.2:2,-2:.2:3) ;
```

```
Z = X.*exp(-X.^2-Y.^2) ;
```

```
%δίνουμε τιμές στα  
% ύψη σε σχέση με τα  
% x και y
```

```
[C,h] = contour(X,Y,Z) ;
```

```
set(h,'ShowText','on','TextStep',get(h,'LevelStep')*2)  
colormap cool
```



Εικόνα 38: Γραφική Παράσταση Countour

5.6.2.7 Plot3 (Τρισδιάστατες καμπύλες)

Στη MATLAB μπορούμε να σχεδιάσουμε τρισδιάστατες καμπύλες με την εντολή **plot3** η οποία είναι το τρισδιάστατο ανάλογο της plot. Η γενική μορφή της plot3 είναι

```
plot3(x, y, z )
```

παράδειγμα:

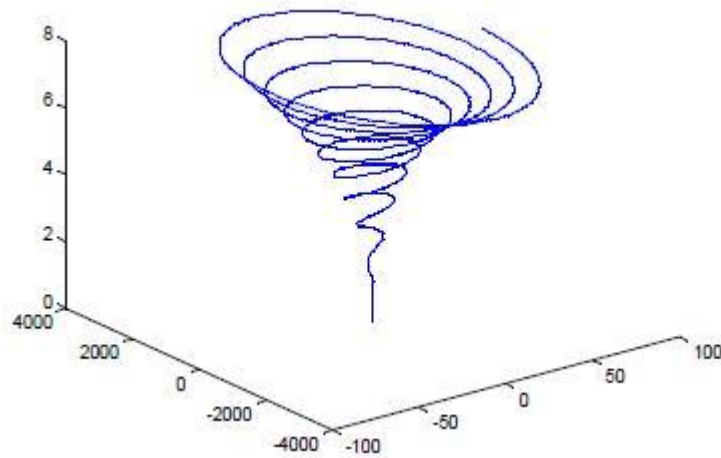
Για να σχεδιάσουμε την τρισδιάστατη καμπύλη

$$x(t) = t \cos t, \quad y(t) = t^2 \sin t, \quad z(t) = \sqrt{t}, \quad t \in [0, 20\pi]$$

χρησιμοποιούμε τις εντολές

```
>> t=0:pi/100:20*pi;  
>> x=t.*cos(t);  
>> y=t.^2.*sin(t);  
>> z=sqrt(t);  
>> plot3(x,y,z)
```

οι οποίες παράγουν το πιο κάτω γράφημα:



Εικόνα 39: Γραφική Παράσταση Plot3

5.7 ΑΝΑΠΤΥΞΗ ΓΡΑΦΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

Το MATLAB προσφέρει στο χρήστη τη δυνατότητα να κατασκευάσει δικές του γραφικές διεπιφάνειες, (Graphical User Interfaces). Η χρησιμότητα αυτής της λειτουργίας είναι μεγάλη, επειδή τα προγράμματα-εφαρμογές που περιέχουν γραφική διεπιφάνεια γίνονται πιο φιλικές στον τελικό χρήστη. Το MATLAB διαθέτει μια ικανοποιητική εργαλειοθήκη, η οποία διευκολύνει πολύ τη δημιουργία μιας γραφικής διεπιφάνειας χρήστη. Αυτή η εργαλειοθήκη ή αλλιώς GUIDE περιέχει μια πληθώρα χρήσιμων εργαλείων ελέγχου όπως κουμπιά, πλαίσια κ.α.

Ο χρήστης έχει τη δυνατότητα μη-χρησιμοποίησης του GUIDE και αντί αυτού να δημιουργήσει μόνος του κάποιο *figure* προγραμματίζοντας κατάλληλα ένα m-αρχείο. Επίσης μπορεί να εισάγει και να προγραμματίσει τα διάφορα στοιχεία ελέγχου εκ του μηδενός (from scratch). Η τεχνική αυτή αποτελεί μια τεχνική χαμηλού επιπέδου σε αντίθεση με τη χρήση του GUIDE που αποτελεί μια τεχνική υψηλού επιπέδου.

5.7.1 Παρουσίαση του GUIDE (GUI Layout Editor)

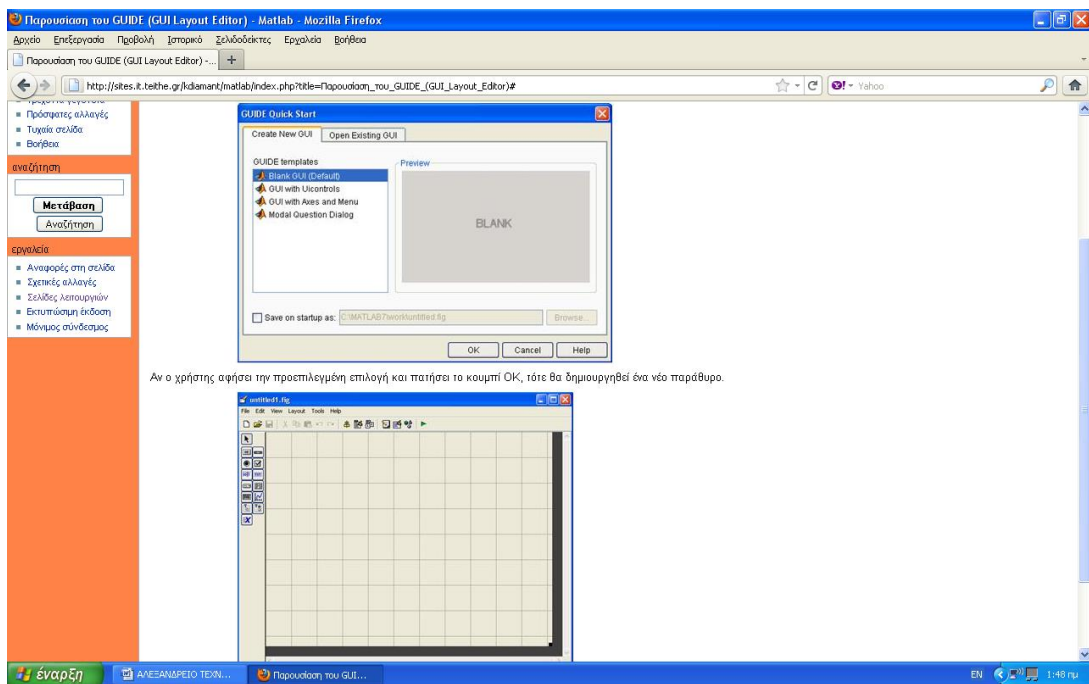
Το Matlab δίνει την δυνατότητα στο χρήστη να δημιουργήσει γραφικό περιβάλλον για τις εφαρμογές του. Το εργαλείο GUIDE περιέχει όλες τις

ευκολίες για το σκοπό αυτό. Η εκκίνηση του GUIDE γίνεται εύκολα, με δύο τρόπους. Πρώτον, με την κλήση της ομώνυμης συνάρτησης από τη γραμμή εντολών του MATLAB.

```
>>guide
```

Δεύτερον, επιλέγοντας από τη γραμμή μενού **File**→**New**→**GUI**. Όποιο τρόπο και να επιλέξει ο χρήστης θα εμφανιστεί ένα dialog που επιτρέπει να διαλέξουμε τον τύπο figure που θέλουμε. Ο οδηγός ρωτάει τον χρήστη αν θέλει να δημιουργήσει ένα κενό παράθυρο(Blank GUI), αν θέλει να δημιουργήσει ένα κενό παράθυρο βασισμένο σε κάποια πρότυπα, (π.χ. με άξονες και μενού) ή αν θέλει να ανοίξει ένα έτοιμο παράθυρο.

Αν ο χρήστης αφήσει την προεπιλεγμένη επιλογή και πατήσει το κουμπί OK, τότε θα δημιουργηθεί ένα νέο παράθυρο.



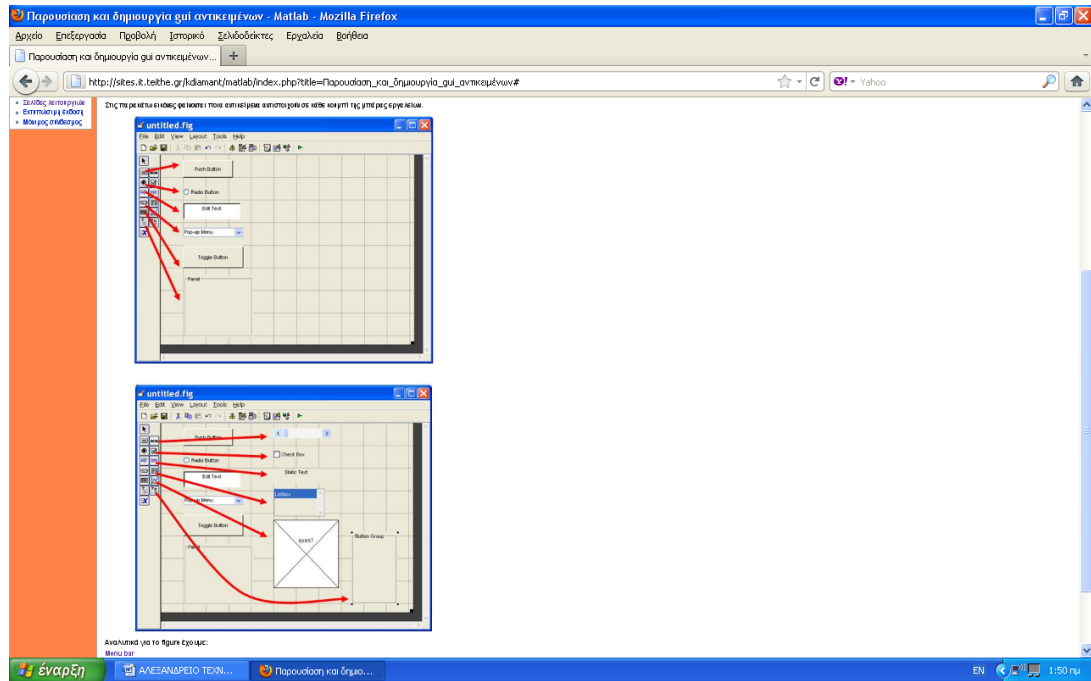
Αν ο χρήστης αφήσει την προεπιλεγμένη επιλογή και πατήσει το κουμπί OK, τότε θα δημιουργηθεί ένα νέο παράθυρο.

Εικόνα 40: Guide του matlab μέσα στο wiki

5.7.2 Παρουσίαση και δημιουργία gui αντικειμένων


Στο κεντρικό παράθυρο του GUIDE περιέχετε ένα figure το οποίο είναι αρχικά άδειο. Στα αριστερά υπάρχει η μπάρα εργαλείων (toolbar) η οποία περιέχει διάφορα αντικείμενα (objects) –ως επί το πλείστον κουμπιά– τα οποία μπορούμε να τοποθετήσουμε στην εικόνα μας. Τα αντικείμενα–κουμπιά στην μπάρα εργαλείων είναι τα εξής:

Στις παρακάτω εικόνες φαίνονται ποια αντικείμενα αντιστοιχούν σε κάθε κουμπί της μπάρας εργαλείων.

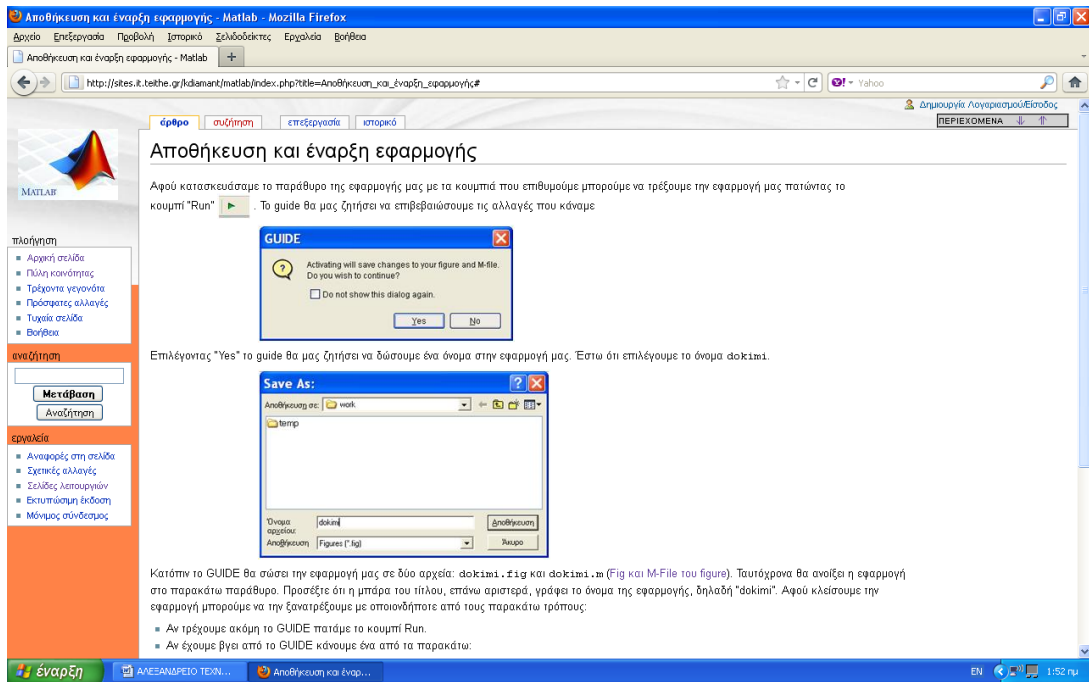


Εικόνα 41: Gui αντικείμενα του matlab μέσα στο wiki

5.7.3 Αποθήκευση και έναρξη εφαρμογής

Αφού κατασκευάσαμε το παράθυρο της εφαρμογής μας με τα κουμπιά που επιθυμούμε μπορούμε να τρέξουμε την εφαρμογή μας πατώντας το κουμπί “Run” . Το guide θα μας ζητήσει να επιβεβαιώσουμε τις αλλαγές που κάναμε

Επιλέγοντας “Yes” το guide θα μας ζητήσει να δώσουμε ένα όνομα στην εφαρμογή μας. Έστω ότι επιλέγουμε το όνομα dokimi.



Εικόνα 42: Guide του matlab μέσα στο wiki

Κατόπιν το GUIDE θα σώσει την εφαρμογή μας σε δύο αρχεία: dokimi.fig και dokimi.m ([Fig και M-File του figure](#)). Ταυτόχρονα θα ανοίξει η εφαρμογή στο παρακάτω παράθυρο. Προσέξτε ότι η μπάρα του τίτλου, επάνω αριστερά, γράφει το όνομα της εφαρμογής, δηλαδή “dokimi”. Αφού κλείσουμε την εφαρμογή μπορούμε να την ξανατρέξουμε με οποιοδήποτε από τους παρακάτω τρόπους:

- Αν τρέχουμε ακόμη το GUIDE πατάμε το κουμπί Run.
- Αν έχουμε βγει από το GUIDE κάνουμε ένα από τα παρακάτω:
 - ~Γράφουμε στο command line την εντολή
>> dokimi
 - οπότε ανοίγει η εφαρμογή αλλά όχι το παράθυρο σχεδίασης
 - ~Γράφουμε στο command line την εντολή
>> guide (dokimi)
 - οπότε ανοίγει και η εφαρμογή και το παράθυρο σχεδίασης
 - ~Γράφουμε στο command line την εντολή
>> guide

και όταν βγει το παράθυρο του guide επιλέγουμε “Open Existing GUI” για να ανοίξει το παράθυρο σχεδίασης και κατόπιν η εφαρμογή τρέχει με Run.

5.7.4 Fig και M-File του figure

Μια Γραφική Διεπιφάνεια Χρήστη αποτελείται συνήθως από διάφορα παράθυρα τα οποία περιέχουν ποικίλα στοιχεία ελέγχου όπως πεδία κειμένου, γραμμές κύλισης κ.α. Τα παράθυρα αυτά είναι δυνατόν να καλούν το ένα το άλλο, να δέχονται δεδομένα από το χρήστη, να μεταβιβάζουν πιθανώς τα δεδομένα από το ένα παράθυρο στο άλλο και γενικά να επιτελούν διάφορες λειτουργίες. Το GUIDE για να το πετύχει αυτό δημιουργεί για κάθε νέο παράθυρο, δύο αρχεία. Τα αρχεία αυτά είναι το FIG-αρχείο και το m-αρχείο.

- Το αρχείο fig περιέχει μία ολοκληρωμένη περιγραφή του GUI figure και όλων των GUI αντικειμένων του (uicontrols και axes) και τις τιμές όλων των ιδιοτήτων των αντικειμένων που μπορεί να υπάρχουν σε αυτό. Μπορούμε να κάνουμε αλλαγές σε αυτό το αρχείο με τον GUIDE.
- Το m αρχείο περιέχει τις μεθόδους που τρέχουν και ελέγχουν το GUI και τις callbacks. Αυτό το αρχείο αναφέρεται σαν GUI M-file.

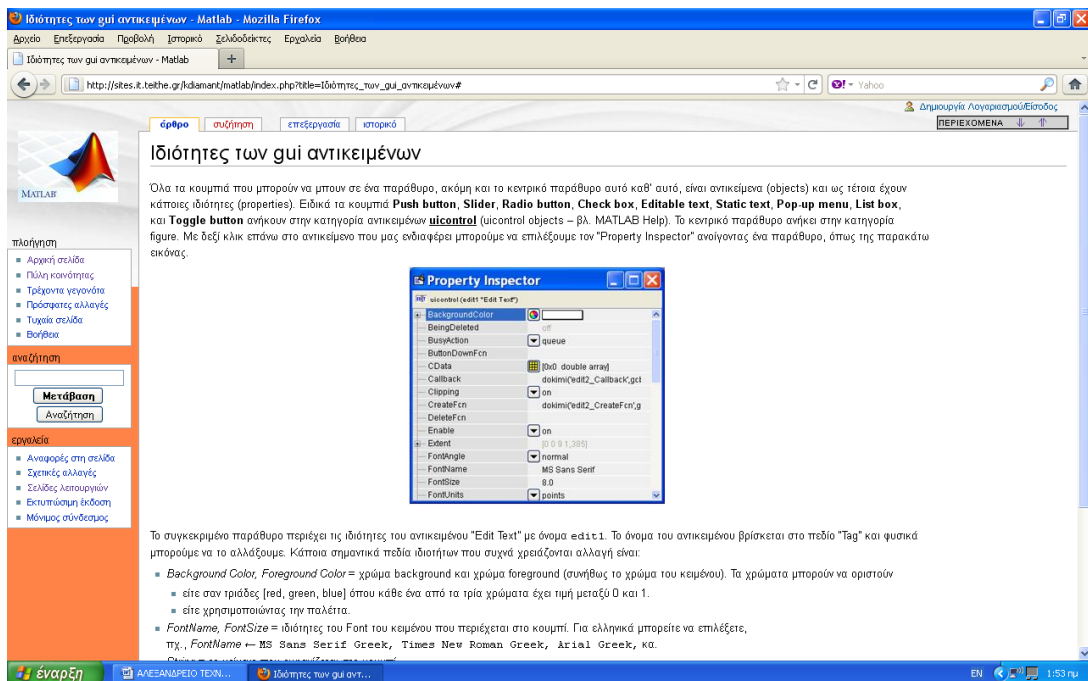
Ο GUIDE για λόγους ευκολίας δημιουργεί το m-file απευθείας από το layout(το σχέδιο του figure που έχουμε σχεδιάσει με τον GUIDE όπως τα κουμπιά, τα plot ή οποιοδήποτε άλλο γραφικό αντικείμενο που έχουμε προσθέσει). Επίσης παράγονται αυτόματα και οι callbacks για τα GUI αντικείμενα που τους χρειάζονται. Αρχικά δημιουργούνται δύο callbacks η *Opening function* και η *Output function*. Η πρώτη εκτελείται πριν το figure γίνει ορατό στο χρήστη. Η Output function επιστρέφει δεδομένα στο command line. Τα default ονόματα των δύο αυτών μεθόδων είναι my_gui_OpeningFcn και my_gui_OutputFcn αντίστοιχα(όπου my_gui είναι το όνομα που δώσαμε στο figure μας). Και στις δύο μεθόδους μπορούμε να προσθέσουμε δικό μας κώδικα όπως π.χ. στην Opening function μπορούμε να αρχικοποιήσουμε δεδομένα. Κάθε φορά που εισάγουμε ένα γραφικό αντικείμενο με το GUIDE παράγετε και στο m-File το αντίστοιχο callback.

!!! Κάθε αρχείο *.fig πρέπει να συνοδεύεται από το αντίστοιχο αρχείο *.m,

(με το ίδιο όνομα). Αν για κάποιο λόγο, χαθεί ή καταστραφεί το *.m αρχείο, τότε το παράθυρο *.fig δεν θα είναι λειτουργικό.

5.7.5 Ιδιότητες των gui αντικειμένων

Όλα τα κουμπιά που μπορούν να μπουν σε ένα παράθυρο, ακόμη και το κεντρικό παράθυρο αυτό καθ' αυτό, είναι αντικείμενα (objects) και ως τέτοια έχουν κάποιες ιδιότητες (properties). Ειδικά τα κουμπιά **Push button**, **Slider**, **Radio button**, **Check box**, **Editable text**, **Static text**, **Pop-up menu**, **List box**, και **Toggle button** ανήκουν στην κατηγορία αντικειμένων **uicontrol** (uicontrol objects – βλ. MATLAB Help). Το κεντρικό παράθυρο ανήκει στην κατηγορία figure. Με δεξί κλικ επάνω στο αντικείμενο που μας ενδιαφέρει μπορούμε να επιλέξουμε τον “Property Inspector” ανοίγοντας ένα παράθυρο, όπως της παρακάτω εικόνας.



Εικόνα 43: Gui αντικείμενα του matlab μέσα στο wiki

Το συγκεκριμένο παράθυρο περιέχει τις ιδιότητες του αντικειμένου “Edit Text” με όνομα edit1. Το όνομα του αντικειμένου βρίσκεται στο πεδίο “Tag” και φυσικά μπορούμε να το αλλάξουμε. Κάποια σημαντικά πεδία ιδιοτήτων που συχνά χρειάζονται αλλαγή είναι:

- *Background Color, Foreground Color* = χρώμα background και χρώμα foreground (συνήθως το χρώμα του κειμένου). Τα χρώματα μπορούν να οριστούν
 - είτε σαν τριάδες [red, green, blue] όπου κάθε ένα από τα τρία χρώματα έχει τιμή μεταξύ 0 και 1.
 - είτε χρησιμοποιώντας την παλέττα.
- *FontName, FontSize* = ιδιότητες του Font του κειμένου που περιέχεται στο κουμπί. Για ελληνικά μπορείτε να επιλέξετε, πχ., *FontName* ← MS Sans Serif Greek, Times New Roman Greek, Arial Greek, κα.
- *String* = το κείμενο που εμφανίζεται στο κουμπί.
- *Callback* = η συνάρτηση ή το script που καλείται όταν πατηθεί το κουμπί αυτό
- κλπ

!!!Φυσικά το πλήθος και το είδος των πεδίων εξαρτάται από τον τύπο του κουμπιού. Η ανάγκη μας να διαβάσουμε ή να αλλάξουμε τις τιμές που περιέχει ένα πεδίο εξαρτάται προφανώς από την συγκεκριμένη εφαρμογή που θέλουμε να υλοποιήσουμε.

5.7.6 Δέντρο αντικειμένων(handles, findall, guihandles)

Η δημιουργία μιας παραθυρικής εφαρμογής GUI δημιουργεί αυτομάτως μια δομή δέντρου ανάμεσα στα αντικείμενα που περιλαμβάνει η εφαρμογή αυτή. Η *ρίζα του δέντρου* είναι το κεντρικό παράθυρο το οποίο είναι *πάντα αντικείμενο τύπου figure*. Παιδιά του αντικειμένου αυτού είναι όλα τα αντικείμενα που περιέχει το παράθυρο. Σε κάθε αντικείμενο αντιστοιχεί ένας **δείκτης (handle)**. Κάποιοι χρήσιμοι αυτόματοι δείκτες (δεσμευμένες μεταβλητές στο MATLAB) είναι οι παρακάτω:

- **gcf** = δείκτης στο τρέχον figure (get current figure)
- **gco** = δείκτης στο τρέχον object (get current object). Τρέχον είναι το παραθυρικό αντικείμενο (κουμπί, παράθυρο, μενού, κλπ) που έχει το focus. Ένα αντικείμενο αποκτάει focus όταν κάνουμε κλικ επάνω του.
- **gca** = δείκτης στο τρέχον axis (get current axis)
- **gcbo** = δείκτης στο object του οποίου το callback εκτελείται. Το αντικείμενο gcbo είναι ίδιο με το αντικείμενο gco εαν το callback

αντιστοιχεί στην ενέργεια «κλικ». Αν όμως το callback αντιστοιχεί σε άλλο event εκτός από κλικ μπορεί τα δύο αντικείμενα να μην είναι ίδια

Η εντολή **findall** βρίσκει όλα τα handles όλων των παραθυρικών αντικειμένων είτε αυτά είναι figures είτε είναι παιδιά άλλων αντικειμένων. Τα handles αυτά επιστρέφονται στο array h.

```
>> h = findall(0);
```

Η εντολή **guihandles** φορτώνει στη δομή handles τους δείκτες όλων των αντικειμένων που βρίσκονται μέσα στο δέντρο του τρέχοντος figure.

```
>> handles = guihandles(gcf);
```

5.7.7 Οι εντολές set, get

Οι εντολές set και get επιτρέπουν την πρόσβαση στις ιδιότητες ενός παραθυρικού αντικειμένου μέσα από το command line αρκεί να γνωρίζουμε το handle του αντικειμένου.

- Για παράδειγμα, εκτελώντας την εντολή

```
>> x = get(fh, 'Tag')
```

παίρνουμε ως απάντηση την τιμή της ιδιότητας 'Tag' του αντικειμένου με handle=fh.

```
x =
```

```
figure1
```

- Αντίστοιχα εκτελώντας την εντολή

```
>> x = get(fh);
```

παίρνουμε μια ολόκληρη δομή x που περιέχει όλες τις ιδιότητες του αντικειμένου με handle=fh.

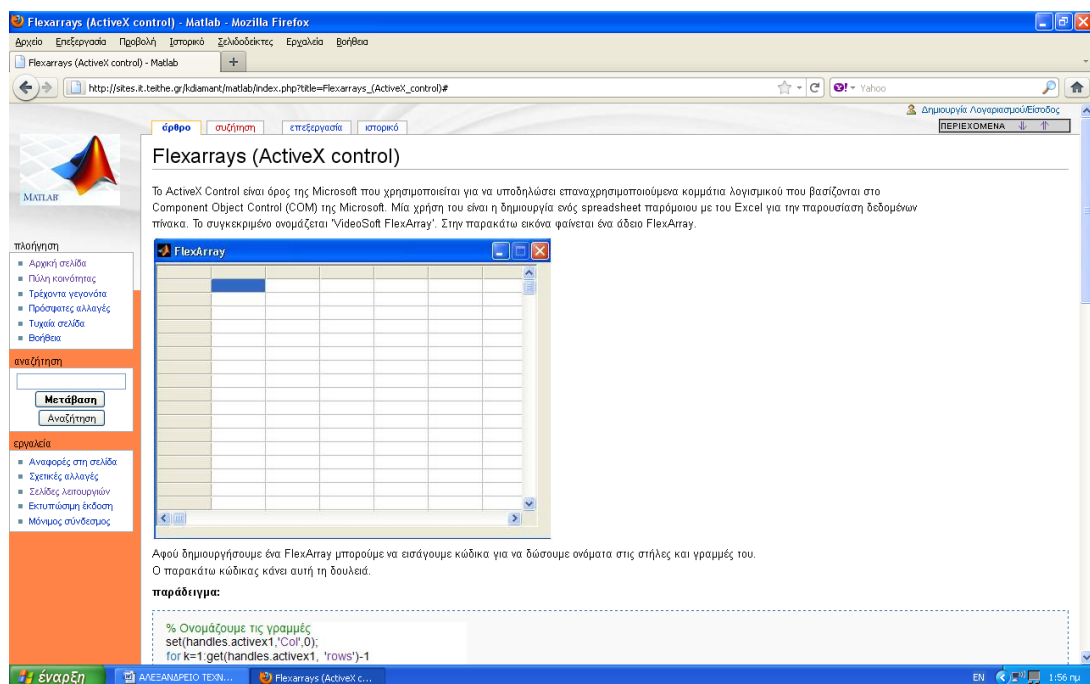
5.7.8 Callbacks

Τα κουμπιά που βάζουμε σε ένα παράθυρο έχουν κάποιες ελάχιστες (default) λειτουργίες: πχ. το toggle button αφού πατηθεί θα παραμείνει πατημένο. Ωστόσο πρέπει να δώσουμε στα κουμπιά τη λειτουργικότητα που

θα θέλαμε έτσι ώστε να κάνουν κάτι χρήσιμο για την εφαρμογή μας. Η λειτουργία που εκτελείται όταν πατηθεί ένα κουμπί, καλείται **callback**. Το MATLAB προσφέρει *έτοιμες συναρτήσεις callback* για κάθε κουμπί της εφαρμογής μας. Αν υποθέσουμε ότι η εφαρμογή μας είναι η `neuralnet.fig` τότε οι συναρτήσεις-callback βρίσκονται μέσα στο αρχείο `neuralnet.m`.

5.7.9 Flexarrays (ActiveX control)

Το ActiveX Control είναι όρος της Microsoft που χρησιμοποιείται για να υποδηλώσει επαναχρησιμοποιούμενα κομμάτια λογισμικού που βασίζονται στο Component Object Control (COM) της Microsoft. Μία χρήση του είναι η δημιουργία ενός spreadsheet παρόμοιου με του Excel για την παρουσίαση δεδομένων πίνακα. Το συγκεκριμένο ονομάζεται 'VideoSoft FlexArray'. Στην παρακάτω εικόνα φαίνεται ένα άδειο FlexArray.



Εικόνα 44: Flexarrays του matlab μέσα στο wiki

Αφού δημιουργήσουμε ένα FlexArray μπορούμε να εισάγουμε κώδικα για να δώσουμε ονόματα στις στήλες και γραμμές του.

Με τις παρακάτω εντολές μπορούμε να αλλάξουμε το μέγεθος του FlexArray.

```
set(handles.activex1,'Rows', arithmosGrammwon );
```

```
set(handles.activex1,'Cols', arithmosSthlwn );
```

Αξίζει να σημειωθεί ότι στον αριθμό των γραμμών και στηλών συμπεριλαμβάνεται και τα κελιά με τα ονόματα τους, δηλαδή τα Row 1, Row2,..., και τα Column 1, Column 2,...

Άρα στην περίπτωση που θέλουμε να εμφανίσουμε ένα πίνακα 2x4 σε ένα FlexArray τότε δημιουργούμε ένα με 3 γραμμές και 5 στήλες.

5.8 NEURAL NETWORK TOOLKIT

Το **NNToolKit** απλοποιεί την δημιουργία και χρησιμοποίηση των νευρωνικών δικτύων στο Matlab. Το toolkit περιέχει functions και structures που χειρίζονται τα νευρωνικά δίκτυα ώστε να μην χρειάζεται να γράψουμε κώδικα για τις functions ενεργοποίησης, για τους αλγόριθμους εκπαίδευσης, κτλ που θέλουμε ναχρησιμοποιήσουμε.

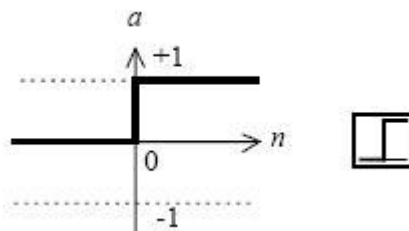
5.8.1 ΔΟΜΕΣ ΚΑΙ ΣΥΝΑΡΤΗΣΕΙΣ ΤΟΥ NEURAL NETWORK TOOLKIT

5.8.1.1 Συναρτήσεις Ενεργοποίησης

Οι συναρτήσεις ενεργοποίησης των νευρώνων που περιέχει το NNToolKit είναι πάρα πολλές. Μία πλήρες λίστα υπάρχει στο help του Matlab. Παρακάτω παρουσιάζονται οι πιο κοινές από αυτές.

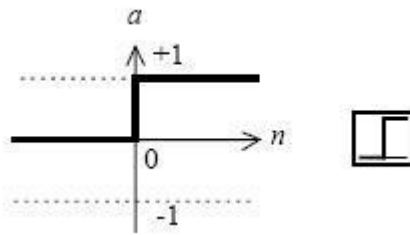
Βηματική(0/1)

```
a=hardlim(n)
```



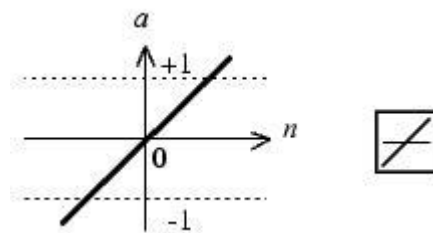
Βηματική(-1/1)

$a = \text{hardlims}(n)$



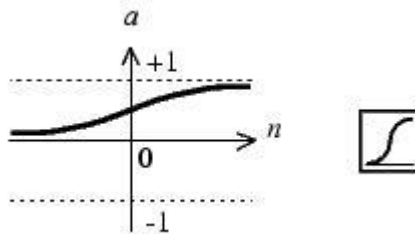
Γραμμική

$a = \text{purelin}(n)$



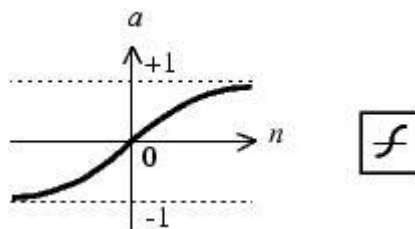
Σιγμοειδείς

$a = \text{logsig}(n)$



Υπερβολική Εφαπτομένη

$a = \text{tansig}(n)$



5.8.1.2 Δομές και Συναρτήσεις δημιουργίας δικτύων

Στο NNToolKit περιέχονται δομές(structs) για την αποθήκευση των ιδιοτήτων των δικτύων. Αυτές οι δομές δημιουργούνται από συγκεκριμένες συναρτήσεις.

Perceptron

Ένα δίκτυο Perceptron δημιουργείται με την συνάρτηση `newp`.

```
net = newp(PR, S, TF, LF)
```

όπου:

PR -- ένας πίνακας R_x2 όπου περιέχει το ελάχιστο και μέγιστο για R εισόδους.

S -- ο αριθμός των νευρώνων.

TF -- η συνάρτηση ενεργοποίησης (προεπιλογή είναι η `hardlim`)

LF -- η συνάρτηση εκπαίδευσης (προεπιλογή είναι η `learnp`)

Η TF μπορεί να πάρει τις τιμές `'hardlim'` ή `'hardlims'` και η LF τις τιμές `'learnp'` και `'learnpn'`. Στη μεταβλητή `net` επιστρέφεται το struct με τις ιδιότητες του δικτύου που δημιουργήσαμε. Περιέχει την αρχιτεκτονική του δικτύου, τις συναρτήσεις ενεργοποίησης, εκπαίδευσης, τις παραμέτρους εκπαίδευσης, τα βάρη κτλ.

Στο command window μπορείτε να γράψετε το όνομα του δικτύου που δημιουργήσαμε για να δείτε λεπτομερώς τη δομή του struct.

Adaline

Ένα δίκτυο adaline δημιουργείται με την συνάρτηση `newlin` (linear layer).

```
net = newlin(PR,S,ID,LR)
```

όπου:

PR -- ένας πίνακας R_x2 όπου περιέχει το ελάχιστο και μέγιστο για R εισόδους.

S -- ο αριθμός των νευρώνων.

ID -- ένα διάνυσμα που περιέχει την χρονική καθυστέρηση (προεπιλογή είναι

το διάνυσμα [0]).

LR -- ο ρυθμός εκπαίδευσης (προεπιλογή είναι η τιμή 0.01).

Multi-layer Perceptron

Ένα δίκτυο MLP δημιουργείται με την συνάρτηση `newff`.

```
net = newff(PR, [S1 S2...SN1], {TF1 TF2...TFN1}, BTF, BLF, PF)
```

όπου:

PR -- ένας πίνακας R_x2 όπου περιέχει το ελάχιστο και μέγιστο για R εισόδους.

S_i -- το μέγεθος του ith στρώματος, για N_i στρώματα.

TF_i -- η συνάρτηση ενεργοποίησης του ith στρώματος (προεπιλογή είναι η συνάρτηση `tansig`).

BTF -- ο Back – Propagation αλγόριθμος εκπαίδευσης (προεπιλογή είναι η συνάρτηση `traingdx`).

BLF -- ο Back – Propagation weight/bias αλγόριθμος εκπαίδευσης (προεπιλογή είναι η συνάρτηση `learnqdm`).

PF -- η συνάρτηση μέτρησης απόδοσης του δικτύου (προεπιλογή είναι η συνάρτηση `mse`).

Οι συναρτήσεις ενεργοποίησης TF_i μπορεί να πάρει τις τιμές `tansig`, `logsig` ή η `purelin`. Η συνάρτηση εκπαίδευσης BTF μπορεί να πάρει τις τιμές `trainlm`, `trainbfg`, `trainrp`, `traingd`, κτλ. Η συνάρτηση BLF μπορεί να πάρει τις τιμές `learnq` ή `learnqdm`.

Εκπαίδευση Δικτύου

Η εκπαίδευση ενός δικτύου γίνεται με την συνάρτηση `train`. Η `train` εκπαιδεύει το δίκτυο ανάλογα με την συνάρτηση εκπαίδευσης που είναι δηλωμένη στο `struct` του δικτύου δηλαδή το `net.trainFcn` και σύμφωνα με τις παραμέτρους εκπαίδευσης που είναι δηλωμένες στο `net.trainParam`. Όταν τελειώσει η εκπαίδευση εμφανίζεται ένα `figure` με την γραφική παράσταση του σφάλματος σε σχέση με την εποχή εκπαίδευσης.

```
[net, tr, Y, E, Pf, Af] = train(net, P, T, Pi, Ai, VV, TV)
```

όπου:

net -- το νευρωνικό δίκτυο

P – οι είσοδοι του δικτύου

T – οι στόχοι του δικτύου, (προεπιλογή zeros)

Pi -- αρχικές χρονικές καθυστερήσεις των εισόδων (προεπιλογή zeros).

Ai -- αρχικές χρονικές καθυστερήσεις των στρωμάτων (προεπιλογή zeros).

VV -- Structure of validation vectors, default = []

TV -- Structure of test vectors, default = []

και επιστρέφει:

net -- το νέο δίκτυο.

TR -- πληροφορίες της εκπαίδευσης (εποχές και απόδοση).

Y -- οι έξοδοι του δικτύου.

E -- τα σφάλματα του δικτύου.

Pf -- οι τελικές χρονικές καθυστερήσεις των εισόδων.

Af -- οι τελικές χρονικές καθυστερήσεις των στρωμάτων.

Ανάκληση Δικτύου

Η ανάκληση ενός νευρωνικού δικτύου γίνεται με την συνάρτηση sim.

$[Y, Pf, Af, E, perf] = sim(net, P, Pi, Ai, T)$

όπου:

net -- το νευρωνικό δίκτυο.

P -- οι είσοδοι του δικτύου.

Pi -- περιέχει τις αρχικές χρονικές καθυστερήσεις των εισόδων (προεπιλογή zeros).

Ai -- περιέχει τις αρχικές χρονικές καθυστερήσεις των στρωμάτων (προεπιλογή zeros).

T -- οι στόχοι του δικτύου (προεπιλογή zeros).

και επιστρέφει:

Y -- οι έξοδοι του δικτύου.

Pf -- οι τελικές χρονικές καθυστερήσεις των εισόδων.

Af -- οι τελικές χρονικές καθυστερήσεις των στρωμάτων.

E -- τα σφάλματα του δικτύου.

perf -- η απόδοση του δικτύου.

5.8.2 NEURAL NETWORK TOOLKIT PLOTS

Το NNToolKit προσφέρει έτοιμα plots ειδικά για νευρωνικά δίκτυα.

Plotrv

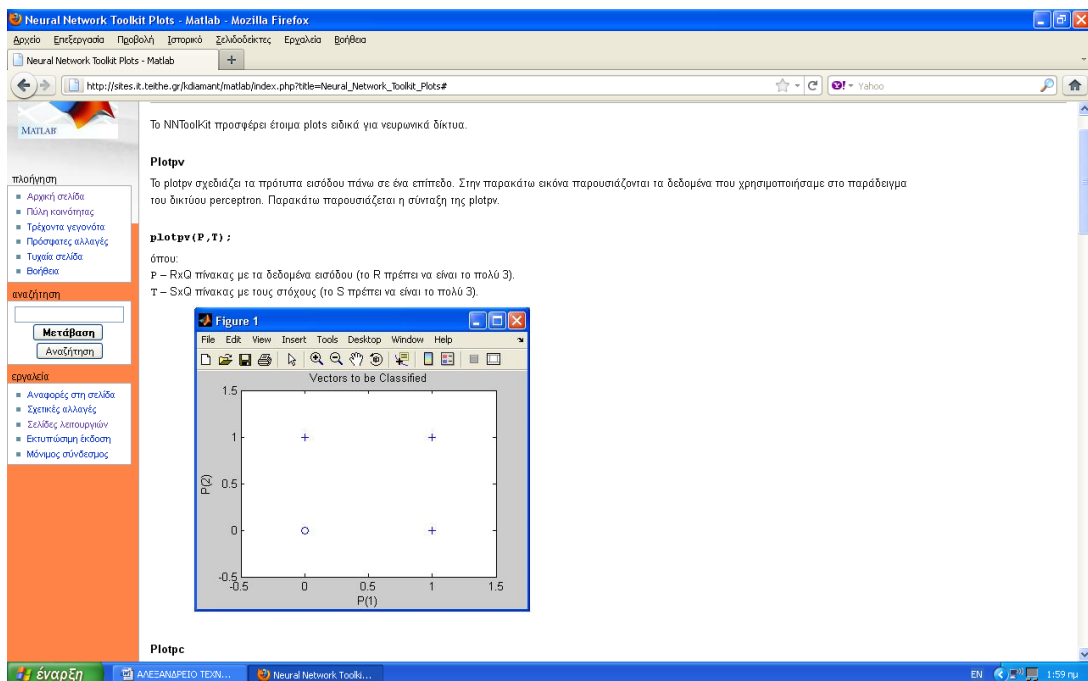
Το plotrv σχεδιάζει τα πρότυπα εισόδου πάνω σε ένα επίπεδο. Στην παρακάτω εικόνα παρουσιάζονται τα δεδομένα που χρησιμοποιήσαμε στο παράδειγμα του δικτύου perceptron. Παρακάτω παρουσιάζεται η σύνταξη της plotrv.

plotrv(P, T) ;

όπου:

P – R \times Q πίνακας με τα δεδομένα εισόδου (το R πρέπει να είναι το πολύ 3).

T – S \times Q πίνακας με τους στόχους (το S πρέπει να είναι το πολύ 3).



Εικόνα 45: NEURAL NETWORK TOOLKIT PLOTS

Plotpc

Το `plotpc` σχεδιάζει την γραμμή διαχωρισμού των προτύπων σε ένα δίκτυο `perceptron`. Στην επόμενη εικόνα φαίνεται η γραμμή διαχωρισμού του παραδείγματος του `perceptron`. Παρακάτω παρουσιάζεται η σύνταξη της `plotpc`.

`plotpc (W, B) ;`

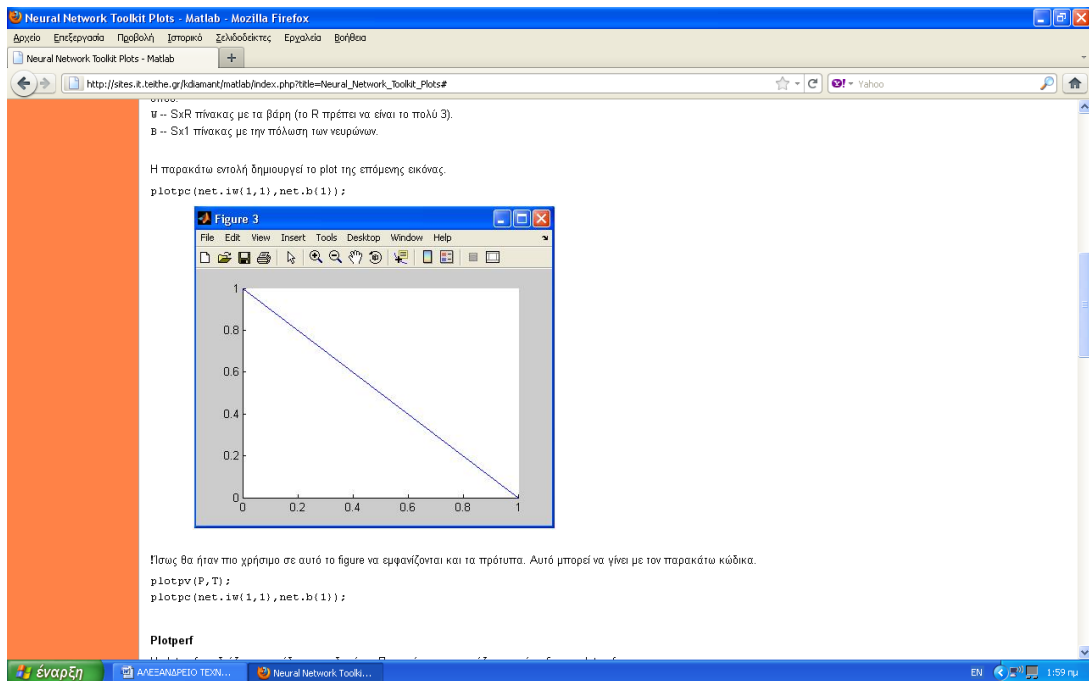
όπου:

W -- SxR πίνακας με τα βάρη (το R πρέπει να είναι το πολύ 3).

B -- Sx1 πίνακας με την πόλωση των νευρώνων.

Η παρακάτω εντολή δημιουργεί το plot της επόμενης εικόνας.

```
plotpc (net.iw{1,1}, net.b{1}) ;
```



Εικόνα 46: NEURAL NETWORK TOOLKIT PLOTS

!!! Ίσως θα ήταν πιο χρήσιμο σε αυτό το figure να εμφανίζονται και τα πρότυπα. Αυτό μπορεί να γίνει με τον παρακάτω κώδικα.

```
plotpcn (P, T) ;
```

```
plotpc (net.iw{1,1}, net.b{1}) ;
```

Plotperf

Η `plotperf` σχεδιάζει την απόδοση του δικτύου. Παρακάτω παρουσιάζεται η σύνταξη της `plotperf`.

`plotperf (TR, goal , name , epoch)`

όπου:

TR -- Η δομή της εκπαίδευση που επιστρέφει η `train`.

goal -- Στόχος απόδοσης (προεπιλογή NaN).

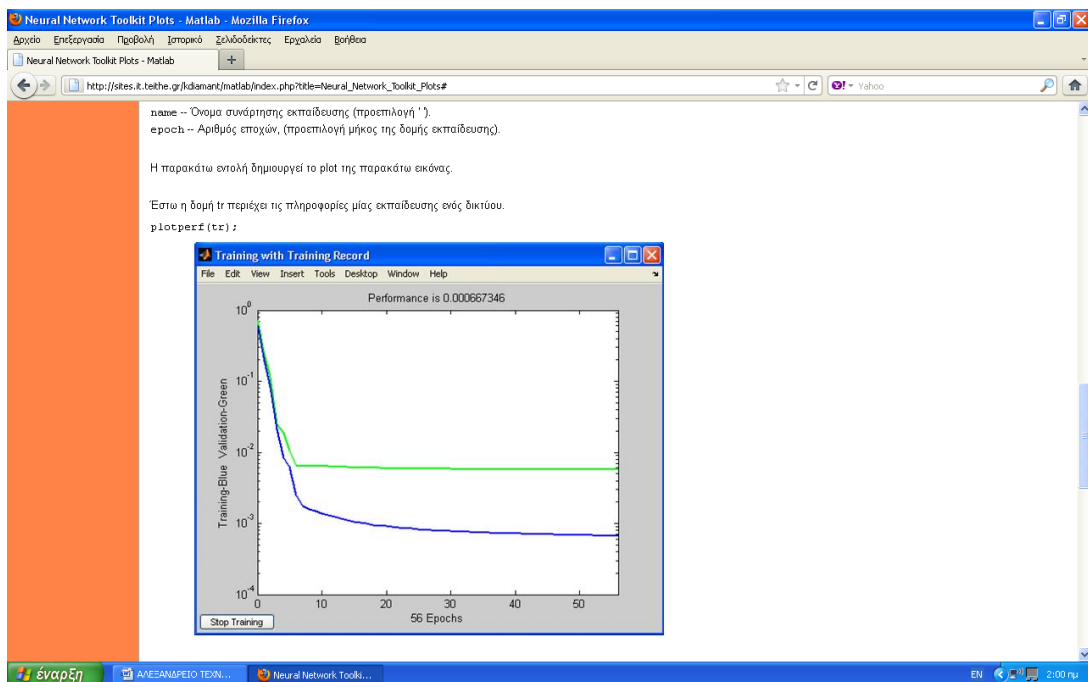
name -- Όνομα συνάρτησης εκπαίδευσης (προεπιλογή ' ').

epoch -- Αριθμός εποχών, (προεπιλογή μήκος της δομής εκπαίδευσης).

Η παρακάτω εντολή δημιουργεί το plot της παρακάτω εικόνας.

Έστω η δομή `tr` περιέχει τις πληροφορίες μίας εκπαίδευσης ενός δικτύου.

`plotperf (tr) ;`



Εικόνα 47: NEURAL NETWORK TOOLKIT PLOTS

Plotsom

Το `plotsom` σχεδιάζει τον τοπογραφικό χάρτη ενός δικτύου SOM. Παρακάτω παρουσιάζεται η σύνταξη της `plotsom`.

`plotsom (pos) ;`

όπου:

pos -- NxS πίνακας με S N-διαστατές συντεταγμένες θέσεων νευρώνων.

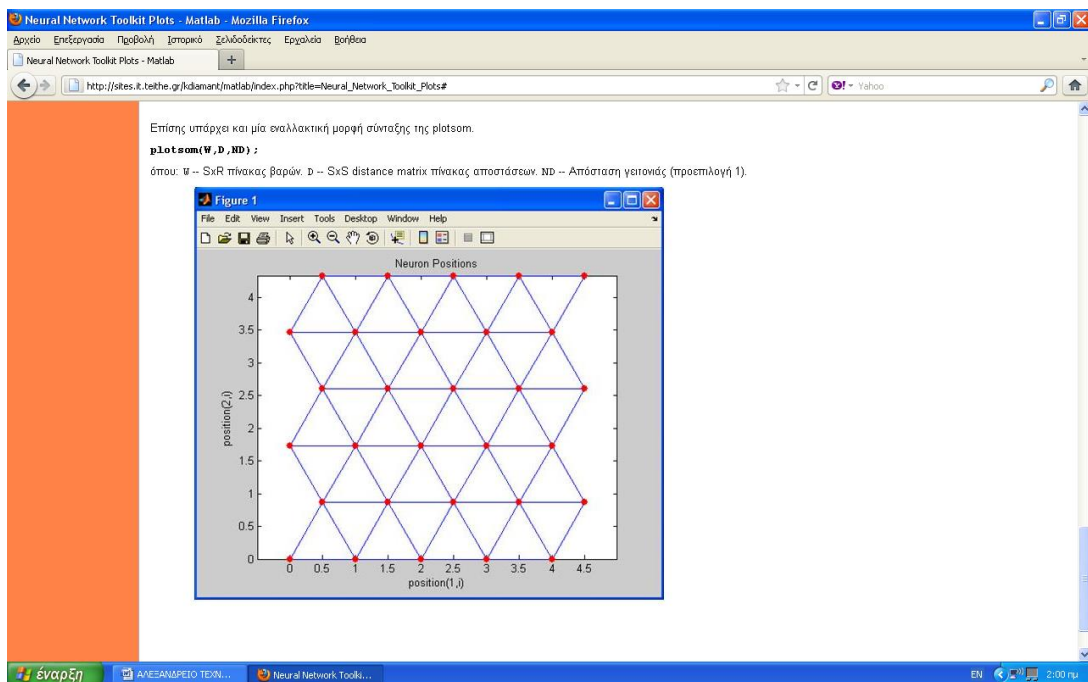
Η παρακάτω εντολή δημιουργεί το plot της επόμενης εικόνας. Σχεδιάζει τις θέσεις των νευρώνων με κόκκινες βούλες και ενώνει του νευρώνες με απόσταση 1.

```
pos = hextop(5, 6);  
plotsom(pos);
```

Επίσης υπάρχει και μία εναλλακτική μορφή σύνταξης της plotsom.

plotsom(W,D,ND);

όπου: W -- SxR πίνακας βαρών. D -- SxS distance matrix πίνακας αποστάσεων. ND -- Απόσταση γειτονιάς (προεπιλογή 1).



Εικόνα 48: NEURAL NETWORK TOOLKIT PLOTS

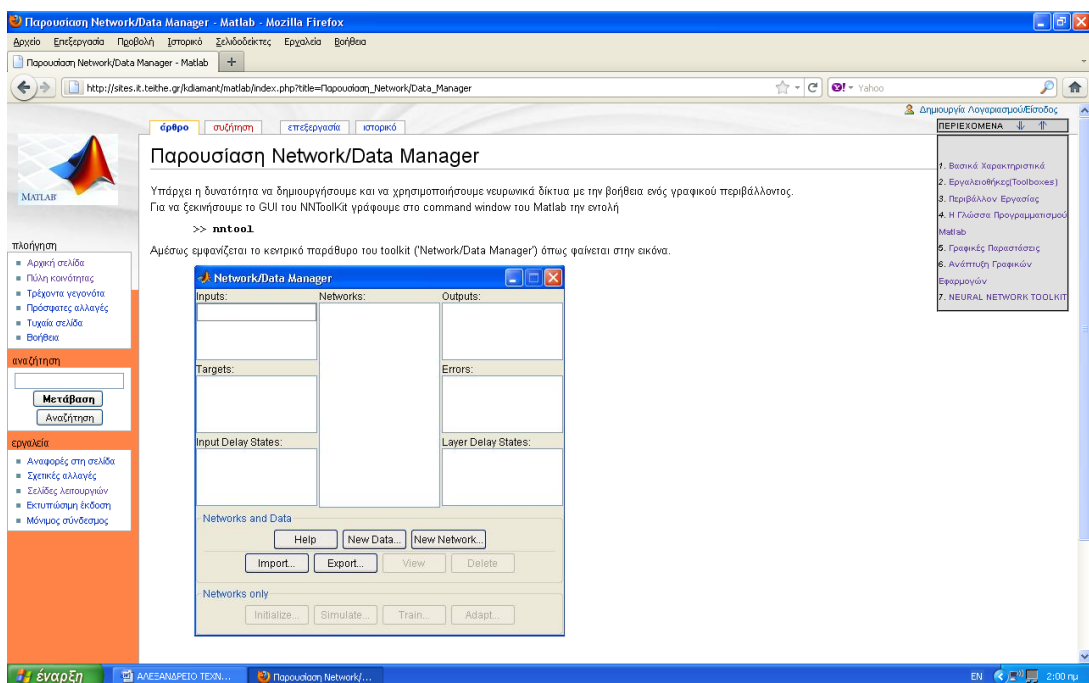
5.8.3 INTERFACE TOY NEURAL NETWORK TOOLKIT

5.8.3.1 Παρουσίαση Network/Data Manager

Υπάρχει η δυνατότητα να δημιουργήσουμε και να χρησιμοποιήσουμε νευρωνικά δίκτυα με την βοήθεια ενός γραφικού περιβάλλοντος. Για να ξεκινήσουμε το GUI του NNToolKit γράφουμε στο command window του Matlab την εντολή

```
>> nntool
```

Αμέσως εμφανίζεται το κεντρικό παράθυρο του toolkit ('Network/Data Manager') όπως φαίνεται στην εικόνα.



Εικόνα 49: NEURAL NETWORK TOOLKIT PLOTS

Όπως βλέπουμε υπάρχουν λειτουργίες που μας επιτρέπουν να δημιουργήσουμε δίκτυα και δεδομένα. Με το κουμπί 'New Network' προφανώς μπορούμε να δημιουργήσουμε ένα δίκτυο. Με τα κουμπιά 'New Data' και 'Import' μπορούμε να δημιουργήσουμε ή να εισάγουμε αντίστοιχα δεδομένα εκπαίδευσης ή ανάκλησης για το δίκτυο ή δίκτυα που μπορεί να έχουμε δημιουργήσει.

6 Μπι-Wiki

6.1 ΠΑΡΑΛΛΗΛΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

6.1.1 Εισαγωγή

Πέραν της, παρατήρησης, της θεωρίας και του πειράματος, η σύγχρονη επιστήμη χαρακτηρίζεται και από την αριθμητική προσομοίωση, η οποία αποτελεί το σημαντικότερο εργαλείο της. Όπου η παρατήρηση και το αίνιγμα κρίνονται ως ανέφικτα ή ανεπαρκή προς επιβεβαίωση ή απόρριψη μιας θεωρητικής υπόθεσης η οποία αφορά ένα πολύπλοκο επιστημονικό πρόβλημα, η καταφυγή σε μεθόδους αριθμητικής προσομοίωσης αποτελεί σήμερα κοινή επιστημονική πρακτική.

Τα σύγχρονα πολύπλοκα επιστημονικά προβλήματα συνήθως ονομάζονται "μεγάλες προσκλήσεις". Η κατηγοριοποίηση τους σύμφωνα με τον Levin είναι:

1. Κβαντοχημεία, στατική μηχανική και σχετικιστική φυσική.
2. Κοσμολογία και αστροφυσική
3. Υπολογιστική δυναμική των ρευστών
4. Σχεδιασμός υλικών και υπεραγωγιμότητα
5. Γενετική μηχανική
6. Ιατρική
7. Καιρός της υδρογείου

Η αριθμητική προσομοίωση των μεγάλων προκλήσεων απαιτεί τεράστια υπολογιστική ισχύ, την οποία οι ακολουθιακοί υπολογιστές δεν διαθέτουν. Η ανάγκη για αύξηση της ταχύτητας των υπολογιστών έχει οδηγήσει τα τελευταία χρόνια στην ανάπτυξη ενός νέου τρόπου επίτευξης καλύτερης απόδοσης : την **παράλληλη επεξεργασία** .

Η έννοια της παραλληλίας, την οποία υπο μορφή αξιώματος ενσωμάτωσε ο Ευκλείδης στην ομόνυμη γεωμετρία του, είναι πρωτογενής. Σε οποιαδήποτε κλίμακα μεταξύ της κοσμολογίας και της κβαντομηχανικής ο κόσμος συμπεριφέρεται οργανωμένος μέσα σε τρεις διαστάσεις χώρου και μια διάσταση χρόνου. Τα γεγονότα, με τα οποία συνυπάρχουν αντικείμενα

και δραστηριότητες, συμβαίνουν στο χώρο και στο χρόνο. Είναι δυνατόν σε γεγονότα να συμβαίνουν στον ίδιο χώρο, το ένα μετά το άλλο μέσα στο χρόνο, δηλαδή κατ'ακολουθία, και εξίσου δυνατόν είναι σε γεγονότα να συμβαίνουν σε διαφορετικούς χώρους στον ίδιο χρόνο, δηλαδή παράλληλα. Κατά φυσική συνέπεια, οι υπολογιστικές μηχανές με δυνατότητες παράλληλης επεξεργασίας δεδομένων χαρακτηρίζονται ως παράλληλοι υπολογιστές.

Γενικά μιλώντας, παράλληλη επεξεργασία έχουμε όταν πολλοί υπολογιστικοί κόμβοι (επεξεργαστές συνήθως) συνεργάζονται για την γρηγορότερη επίλυση ενός προβλήματος – αλγορίθμου. Η ανάπτυξη αυτού του μοντέλου επεξεργασίας υποβοηθήθηκε από διάφορες καταστάσεις. Ο ρυθμός αύξησης της απόδοσης των απλών υπολογιστικών συστημάτων, ακολουθεί φθίνουσα πορεία λόγω φυσικών περιορισμών όπως η πεπερασμένη ταχύτητα μετάδοσης των σημάτων. Επίσης, από ένα σημείο και μετά, το κόστος ενός τέτοιου συμβατικού παράλληλου συστήματος αυξάνει υπερβολικά σε σχέση με την απόδοσή του. Έτσι, άρχισε να ωριμάζει η ιδέα της συνεργασίας πολλών επεξεργαστών για την επίτευξη ενός κοινού στόχου (δηλαδή την λύση ενός προβλήματος σε μικρότερο χρόνο) και έχουμε φτάσει πλέον σε ένα ικανοποιητικό επίπεδο κόστους, τέτοιο ώστε ακόμα και μια μικρή σχετικά επιχείρηση να μπορεί να προμηθευτεί ένα παράλληλο σύστημα και να επωφεληθεί από την χρήση του.

6.1.2 Ταξινομήσεις Παράλληλων Αρχιτεκτονικών

Η πιο γνωστή ταξινόμηση παραλλήλων αρχιτεκτονικών είναι αυτή του Flynn. Όλες οι ταξινομήσεις αποτελούν παραλληλοποιήσεις της υπολογιστικής μηχανής Von Neumann με την εξής έννοια: και στην παράλληλη επεξεργασία εξακολουθεί να ισχύει η αρχή Von Neumann για την ροή και τον έλεγχο του προγράμματος. Εξακολουθεί να ισχύει ο παραδοσιακός κύκλος παραλαβή-εκτέλεση-αποθήκευση και το πρόγραμμα εξακολουθεί να εκτελείται με μια ακολουθία εντολών, σε περισσότερους βέβαια επεξεργαστές ή σε αγωγούς.

Η ταξινόμηση κατά Flynn βασίζεται στο είδος των εντολών και των προς επεξεργασία δεδομένων. Ρεύμα (stream) είναι μια ακολουθία από

εντολές ή δεδομένα. Σ' ένα ρεύμα μπορεί να επαναλαμβάνεται η ίδια εντολή ή η ακολουθία να περιέχει διαφορετικά είδη εντολών. Στην πρώτη περίπτωση έχουμε ρεύμα μοναδικής εντολής (SI: Single Instruction stream) και στην δεύτερη ρεύμα πολλαπλής εντολής (MI: Multiple Instruction stream). Αντίστοιχα για τα δεδομένα, αν επαναλαμβάνονται τα ίδια δεδομένα στο ρεύμα δεδομένων ή αν είναι διαφορετικά έχουμε αντίστοιχα ρεύμα μοναδικών δεδομένων (SD: Single Data) ή πολλαπλών δεδομένων (MD: Multiple Data).

Με αυτή τη λογική οι αρχιτεκτονικές ταξινομούνται σε τέσσερις κατηγορίες:

SISD (Single Instruction stream Single Data stream):

Κάθε φορά εκτελείται μια εντολή σ' ένα αντίστοιχο στοιχείο δεδομένων, διαδοχικά. Η αρχιτεκτονική αυτή για την υλοποίησή της χρησιμοποιεί συμβατικούς, σειριακούς ή ακολουθιακούς υπολογιστές.

SIMD (Single Instruction stream Multiple Data stream):

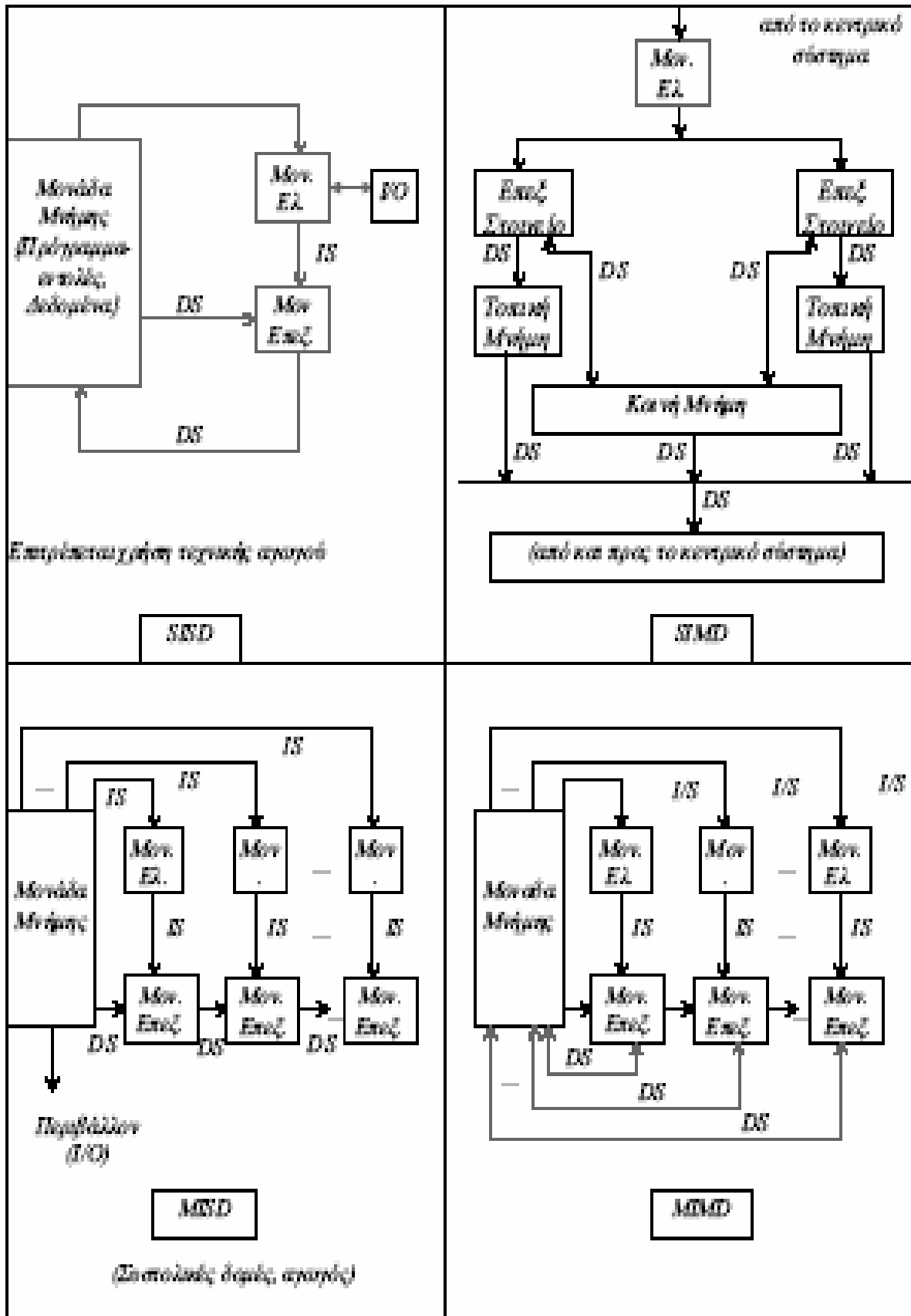
Σε αυτήν την αρχιτεκτονική, η ίδια εντολή εφαρμόζεται σε διαφορετικά δεδομένα. Εδώ εντάσσονται αρκετά είδη παράλληλων αρχιτεκτονικών όπως οι διανυσματικοί υπολογιστές (αγωγοί, υπερυπολογιστές), τα διανύσματα επεξεργαστών, οι πολλαπλοί SIMD κ.ά.

MISD (Multiple Instruction stream Single Data stream):

Εδώ πολλές εντολές ενεργούν ταυτόχρονα σ' ένα στοιχείο δεδομένων. Εδώ μπορούν να καταταγούν ορισμένες συστολικές δομές.

MIMD (Multiple Instruction stream Multiple Data stream):

Εδώ, πολλές εντολές εκτελούνται παράλληλα, η κάθε μια σε διαφορετικά δεδομένα. Τυπικό μέλος της κατηγορίας αυτής αποτελούν τα πολυεπεξεργαστικά συστήματα. Εδώ χρησιμοποιείται ο όρος “πρόγραμμα” αντί του όρου ρεύματος, οπότε επικρατούν και οι αντίστοιχοι όροι, όπως π.χ. SPMD (Single Program Multiple Data) αντί του SIMD. Στην παρακάτω εικόνα γίνεται σχηματική αναπαράσταση των τεσσάρων .



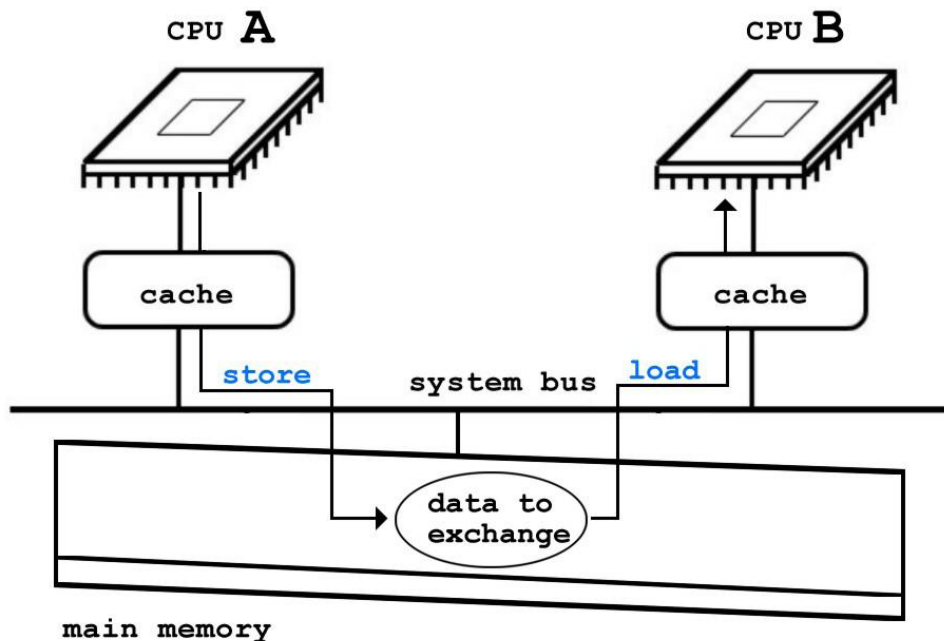
Εικόνα 50: Παράλληλες Αρχιτεκτονικές Υπολογιστών

6.1.3 Μοντέλα Παράλληλης Επεξεργασίας

Υπάρχουν διάφορα μοντέλα παράλληλης επεξεργασίας , μερικά απο τα οποία θα παρουσιάσουμε επιγραμματικά παρακάτω .

✓ Κοινή μνήμη

Οι υπολογιστικοί κόμβοι , λειτουργούν παράλληλα κάνοντας χρήση μιας **κοινής μνήμης** στην οποία όλοι έχουν πρόσβαση και όλοι έχουν ένα κοινό χώρο διευθύνσεων . Έτσι , η επικοινωνία μεταξύ των κόμβων επιτυγχάνεται με την εγγραφή στην κοινή μνήμη .



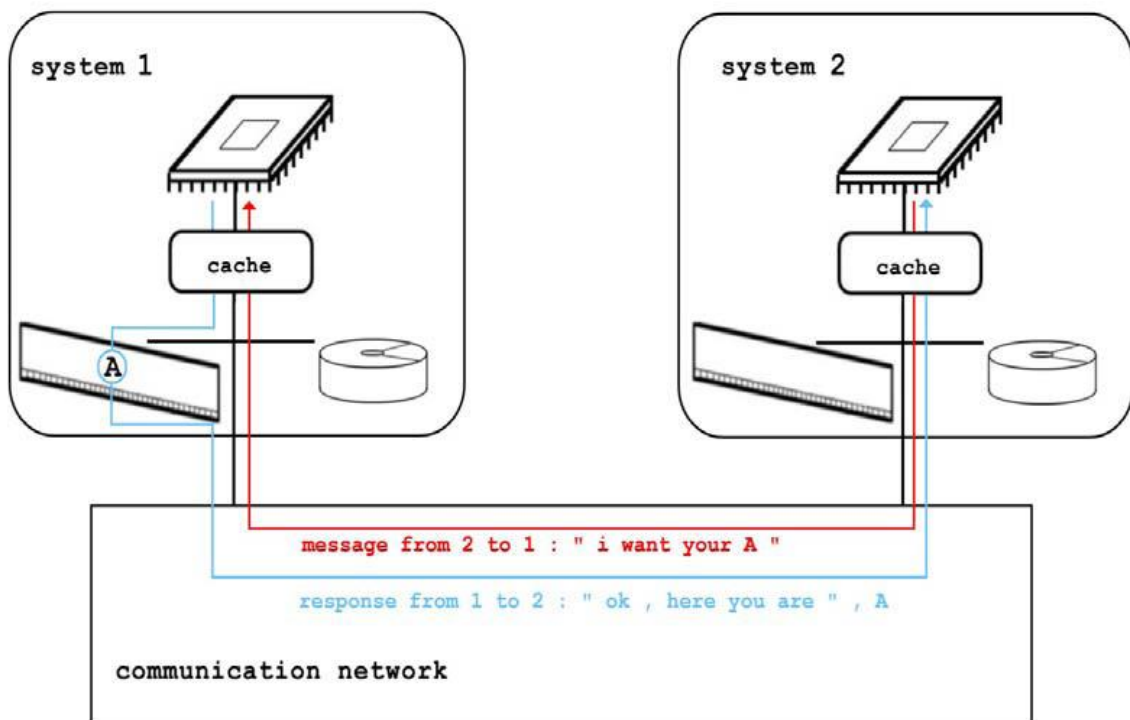
Εικόνα 51: Επικοινωνία σε σύστημα με κοινή μνήμη

Εκτός της φυσικά αλλά και λογικά κοινής μνήμης , υπάρχει και μια παραλλαγή : αυτή της κοινής κατακευματισμένης μνήμης όπου η μνήμη είναι φυσικά διαχωρισμένη σε κομμάτια αλλά υπάρχει ένας κοινός χώρος διευθύνσεων . Σε αυτή την τελευταία περίπτωση ο κάθε κόμβος όταν προσπελάει την μνήμη δεν γνωρίζει εάν θα χρειαστεί

επικοινωνία μεταξύ των κόμβων για να ανακτηθούν τα δεδομένα ή όχι. Πάντως στο μοντέλο της κοινής μνήμης υπάρχει μεγάλος περιορισμός στον αριθμό των κόμβων που μπορούν να προστεθούν στο όλο σύστημα πριν να αρχίσει να υπάρχει επιδείνωση λόγω των συχνών συγκρούσεων για την πρόσβαση στην μνήμη . Επίσης , χρειάζονται ειδικές διαδικασίες για να διατηρηθεί η συνέπεια μεταξύ των τοπικών cache και της κεντρικής μνήμης .

✓ Πέρασμα μηνυμάτων

Σε αυτό το μοντέλο , ο κάθε υπολογιστικός κόμβος έχει την δικιά του τοπική μνήμη και η επικοινωνία επιτυγχάνεται με την **ανταλλαγή μηνυμάτων** μέσω του δικτύου διασύνδεσης . Βασικό στοιχείο αυτής της ιδέας είναι οτι στην μεταφορά δεδομένων μεταξύ δύο κόμβων , πρέπει να συμμετάσχουν και οι δύο επικοινωνούντες κόμβοι .



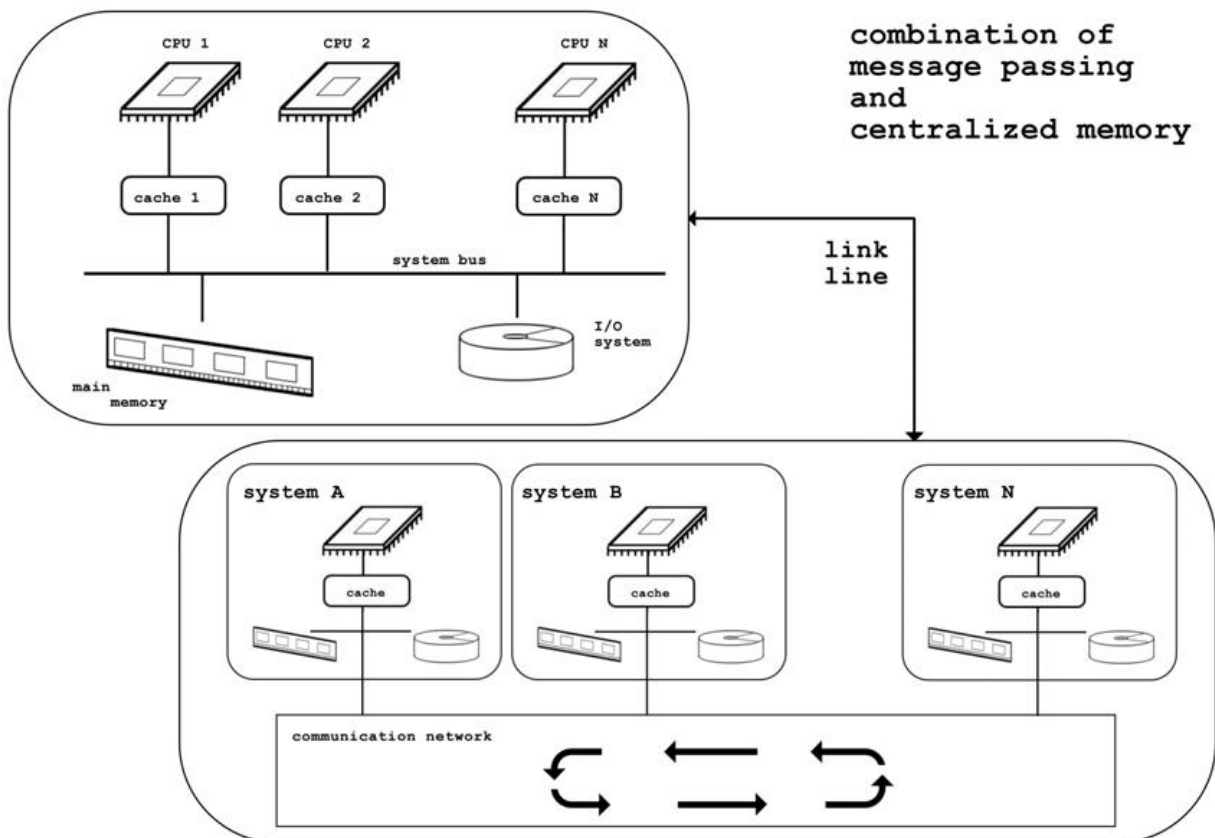
Εικόνα 52: Επικοινωνία με ανταλλαγή μηνυμάτων

✓ **Διαδικασίες απομακρυσμένης μνήμης**

Κάπου μεταξύ των δύο παραπάνω μοντέλων βρίσκεται το μοντέλο των **διαδικασιών απομακρυσμένης μνήμης** . Εδώ , υπάρχουν πάλι τοπικές μνήμες σε κάθε κόμβο , αλλά για την πρόσβαση σε δεδομένα άλλου κόμβου δεν χρειάζεται να συμμετάσχει ο τελευταίος .

✓ **Συνδυασμοί**

Υπάρχουν και διάφοροι συνδυασμοί των παραπάνω μοντέλων , όπως για παράδειγμα σε έναν αριθμό υπολογιστικών κόμβων να γίνεται η επικοινωνία μέσω μηνυμάτων , ενώ οι υπόλοιποι να μοιράζονται μια κοινή περιοχή μνήμης .



Εικόνα 53: Συνδυασμός κοινής μνήμης και ανταλλαγής μηνυμάτων

6.1.4 Πέρασμα μηνυμάτων

✓ Πλεονεκτήματα του μοντέλου ανταλλαγής μηνυμάτων

Παρόλο που δεν μπορούμε να πούμε ότι το μοντέλο ανταλλαγής μηνυμάτων είναι καλύτερο από τα υπόλοιπα μοντέλα σε όλους τους τομείς, έχει επικρατήσει στις τάξεις των προγραμματιστών παράλληλων εφαρμογών. Οι λόγοι που συντέλεσαν σε αυτήν την επικράτηση είναι αρκετοί και σαν χαρακτηριστικούς αναφέρουμε:

- Την ευκολία αποσφαλμάτωσης (*debugging*). Η αποσφαλμάτωση παράλληλων προγραμμάτων είναι μια πολύ δύσκολη υπόθεση. Τα περισσότερα λάθη προκύπτουν από γράψιμο σε ήδη χρησιμοποιούμενη θέση μνήμης. Το μοντέλο ανταλλαγής μηνυμάτων ελέγχει τις αναφορές στην μνήμη με μεγαλύτερη σαφήνεια και κάνει έτσι την ανίχνευση σφαλμάτων πιο εύκολη.

- Την ελαστικότητα και συμβατότητα του με όλα τα αρχιτεκτονικά μοντέλα παράλληλων συστημάτων, είτε αυτά είναι παράλληλοι υπερ-υπολογιστές είτε είναι δίκτυα υπολογιστών (*networks of workstations*, NOW). Σε περίπτωση που υπάρχει επιπλέον hardware το μοντέλο κάνει χρήση αυτού για να επιταχύνει την μεταφορά δεδομένων.

- Τις επιδόσεις του μοντέλου, που είναι φυσικά και ο πιο σημαντικός λόγος που το μοντέλο αυτό θα μείνει στο προσκήνιο για αρκετό καιρό ακόμα. Ένα πλεονέκτημα που έχουν τα συστήματα κατανεμημένων δικτυωμένων υπολογιστών (NOWs) έναντι των παράλληλων υπερ-υπολογιστών είναι η μεγάλη ποσότητα διαθέσιμης μνήμης. Το μοντέλο ανταλλαγής μηνυμάτων, εκμεταλλεύεται αυτό το γεγονός δίνοντας στον προγραμματιστή την δυνατότητα να συσχετίσει συγκεκριμένα δεδομένα του προβλήματος με συγκεκριμένες διεργασίες (*processes*), επιτρέποντας έτσι την καλύτερη διαχείριση της ιεραρχίας της μνήμης, κυρίως σε εφαρμογές που έχουν μεγάλες απαιτήσεις. Όλα τα παραπάνω, δικαιολογούν την ευρεία αποδοχή του μοντέλου ανταλλαγής μηνυμάτων από την προγραμματιστική κοινότητα και προδικάζουν την παραμονή αυτού του μοντέλου στην κορυφή των προτιμήσεων για αρκετό καιρό ακόμα, παρά τα διάφορα μικροπροβλήματα που υπάρχουν.

6.2 MPI

6.2.1 Ιστορική Εξέλιξη

Υπάρχουν πολλές διαφορετικές προσεγγίσεις τις οποίες μπορούμε να ακολουθήσουμε για να προχωρήσουμε στη δημιουργία παράλληλων εφαρμογών εμείς θα περιγράψουμε το πρότυπο MPI . Χωρίς την ύπαρξη κάτι σταθερού στο πέρασμα μηνυμάτων , η ζωή των προγραμματιστών γινόταν πολύ δύσκολη .



Εικόνα 54: Λογότυπο mpi

Οι προτάσεις και οι υλοποιήσεις ήταν πάρα πολλές και πολύ διαφορετικές μεταξύ τους με αποτέλεσμα η μεταφορά ενός παράλληλου προγράμματος από μια πλατφόρμα σε άλλη να είναι από πολύ δύσκολη έως αδύνατη . Η μόνη λύση έμοιαζε να είναι ο ορισμός κάτι σταθερού όσον αφορά το πέρασμα μηνυμάτων . Έτσι οι χρήστες (στον όρο χρήστες συμπεριλαμβάνονται και οι προγραμματιστές και προμηθευτές παράλληλων βιβλιοθηκών ή/και hardware) αποφάσισαν να διευθετήσουν το πρόβλημα .

Αν και οι πρώτες προσπάθειες για την ανάπτυξη παράλληλων εφαρμογών πραγματοποιήθηκαν στις αρχές της δεκαετίας του 1980 με την εισαγωγή της έννοιας της κατανεμημένης μνήμης και του παράλληλου προγραμματισμού, το πρότυπο MPI χρησιμοποιήθηκε για πρώτη φορά τον Απρίλιο του 1992 από το πανεπιστήμιο της Virginia, ενώ το Νοέμβριο του ίδιου έτους πραγματοποιήθηκε η πρώτη συνάντηση της ομάδας εργασία στη Minneapolis. Μια σειρά από συγκεντρώσεις και συζητήσεις οδήγησαν στα τέλη του 1992 στην δημιουργία μιας επιτροπής (***MPI Forum***) που σαν σκοπό θα είχε τον ορισμό ενός προτύπου . Στην όλη προσπάθεια συμμετείχαν περισσότεροι απο 60 άνθρωποι από περισσότερους από 40 οργανισμούς . Μεταξύ αυτών οι περισσότεροι προμηθευτές παράλληλου hardware , πολλοί ερευνητές από πανεπιστήμια καθώς και πολλοί εκπρόσωποι της βιομηχανίας . Εκτός των ανά 6 εβδομάδων τακτικών συναντήσεων υπήρχε και η συνεχής ηλεκτρονική επικοινωνία , στην οποία

μπορούσαν να συμμετέχουν όλα τα μέλη της high performance computing society .

Στον σχεδιασμό του προτύπου η φιλοσοφία ήταν όχι να επιλεχθεί ένα από τα υπάρχοντα συστήματα ανταλλαγής μηνυμάτων και να καθιερωθεί - με κάποιες βελτιώσεις - ως πρότυπο , αλλά η εξαγωγή των κυριότερων χαρακτηριστικών και πλεονεκτημάτων ενός αριθμού διαφορετικών συστημάτων και η χρησιμοποίησή τους για την υλοποίηση του προτύπου .

Στόχος αυτής της συνάντησης ήταν ο καθορισμός των βασικών χαρακτηριστικών του προτύπου και η δημιουργία του MPI Forum μέσα από το οποίο ήταν δυνατή η περαιτέρω ανάπτυξη του εν λόγω περιβάλλοντος και η προτυποποίησή του. Το τελευταίο σημαντικό στάδιο της σύντομης ιστορίας του MPI είναι η διάσκεψη Supercomputing 93 όπου και παρουσιάζονται προσχέδια του MPI, Το εγχείρημα ολοκληρώθηκε τον Μάιο του 1994 . Παρόλα αυτά , στην προσπάθεια να δημιουργηθεί το πρότυπο μέσα σε μικρό χρονικό διάστημα – ένας χρόνος ήταν η αρχική ‘απαίτηση’ , οι διευθετήσεις πολλών θεμάτων μετατοπίστηκαν για αργότερα και το forum ασχολήθηκε κυρίως με την δημιουργία ενός συμπαγούς πυρήνα έτοιμου προς χρήση – που περιελάμβανε κυρίως τις ρουτίνες για point to point επικοινωνία . Η τελική έκδοση του εν λόγω προτύπου παρουσιάστηκε το Μάιο του 1994. Οι παραπάνω αναφορές αφορούν την πρώτη έκδοση του προτύπου MPI που είναι γνωστή ως MPI1 . Το forum επέκτεινε το πρότυπο να συμπεριλαμβάνει και άλλες λειτουργίες κατά την περίοδο 1995 – 1997 – όπως διαχείριση της μνήμης , παράλληλη είσοδος / έξοδος και άλλα . Το Δεκέμβριο του 1995, το πρότυπο αυτό επεκτάθηκε και μετονομάστηκε σε MPI2, που είναι και το μοντέλο που χρησιμοποιείται μέχρι σήμερα .

6.2.2 Οι στόχοι του MPI

Ο κύριος στόχος του όλου εγχειρήματος ήταν η τριάδα απόδοση , μεταφερσιμότητα και λειτουργικότητα , χωρίς όμως για χάρη του ενός να υπονομεύεται κάποιο από τα άλλα χαρακτηριστικά – πχ. να μην μειώνεται η λειτουργικότητα για χάρη αυξημένης απόδοσης ή το αντίθετο – πράγμα που έκανε την όλη διαδικασία εισήγησης ενός προτύπου πολύ δύσκολη . Πιο

ειδικά , οι στόχοι του MPI , όπως αυτοί παρατίθενται στο MPI forum documentation είναι οι εξής :

- Ο σχεδιασμός ενός interface ανάπτυξης εφαρμογών .
- Η ύπαρξη αποδοτικής επικοινωνίας με την αποφυγή memory to memory αντιγραφής δεδομένων και την δυνατότητα επικάλυψης επικοινωνίας και επεξεργασίας .
- Δυνατότητα χρήσης των εφαρμογών βασισμένες στο πρότυπο , σε ετερογενή περιβάλλοντα .
- Η υποστήριξη απο το interface των C και Fortran 77 .
- Η ύπαρξη αξιόπιστου επικοινωνιακού interface : ο χρήστης δεν χρειάζεται να αντιμετωπίζει πιθανές αποτυχίες στην επικοινωνία καθώς αυτές θα αντιμετωπίζονται απο το υποσύστημα επικοινωνίας του προτύπου .
- Ο ορισμός ενός interface που δεν θα διαφέρει πολύ απο τα υπάρχοντα συστήματα αλλά ταυτόχρονα θα προσθέτει και μεγαλύτερη ελαστικότητα και δυνατότητες .
- Ο ορισμός ενός interface που θα μπορεί να υλοποιηθεί σε διαφορετικές πλατφόρμες χωρίς σημαντικές αλλαγές στα λογισμικό συστήματος και επικοινωνίας .
- Η σημασιολογία του interface θα είναι ανεξάρτητη γλωσσών προγραμματισμού .
- Το interface πρέπει να σχεδιαστεί έτσι ώστε να είναι ασφαλές σε επίπεδο threads .

Σημαντικό επίτευγμα είναι η πολυμορφία των διαφορετικών πλατφόρμων πάνω στις οποίες μπορεί να υλοποιηθεί μια εφαρμογή κάνοντας χρήση του προτύπου MPI . Πιο συγκεκριμένα μπορεί να εκτελεστεί σε πολυυπολογιστές με καταμεμημένη μνήμη , σε δίκτυα υπολογιστών ή σε κάποιο συνδυασμό αυτών . Δυνατές όμως είναι και υλοποιήσεις σε μοντέλα με κοινή μνήμη καθώς και σε συνδυασμούς αρχιτεκτονικών κοινής και καταμεμημένης μνήμης . Τέλος το πρότυπο εκμεταλλεύεται πλήρως τυχόν επιπλέον hardware για αυξημένη επικοινωνία μεταξύ των επεξεργαστών, προσφέροντας αυξημένες επιδόσεις .

6.2.3 Βασικές έννοιες

- **non – blocking διαδικασίες** , σημαίνει ότι η διαδικασία επιστρέφει τον έλεγχο στον χρήστη πριν ολοκληρωθεί και πριν ο χρήστης μπορεί εκ του ασφαλούς να επαναχρησιμοποιήσει τα αγαθά που αναφέρθηκαν στην κλήση της διαδικασίας .
 - **blocking διαδικασία** αντίθετα με τις non – blocking διαδικασίες, σημαίνει ότι η διαδικασία επιστρέφει ταυτόχρονα και την απελευθέρωση των αγαθών που χρησιμοποιήθηκαν στην κλήση της και φυσικά ο χρήστης μπορεί να τα ξαναδιαθέσει όπως νομίζει με ασφάλεια .
 - **τοπικές** λέγονται οι διαδικασίες που για την ολοκλήρωσή τους δεν χρειάζεται επικοινωνία με άλλες διεργασίες , αλλά βασίζεται μόνο στην τοπική διεργασία .
 - **μη τοπική** διαδικασία λέγεται αυτή που για την ολοκλήρωσή της χρειάζεται και επικοινωνία με κάποια/ες άλλη/ες διεργασία/ες μέσω κλήσεων κάποιων MPI ρουτινών .
 - **συλλογικές** ονομάζονται οι διαδικασίες που για την ολοκλήρωσή τους πρέπει να συμμετάσχουν όλες οι διεργασίες σε μια ομάδα διεργασιών (group) .

✓ Εκτέλεση

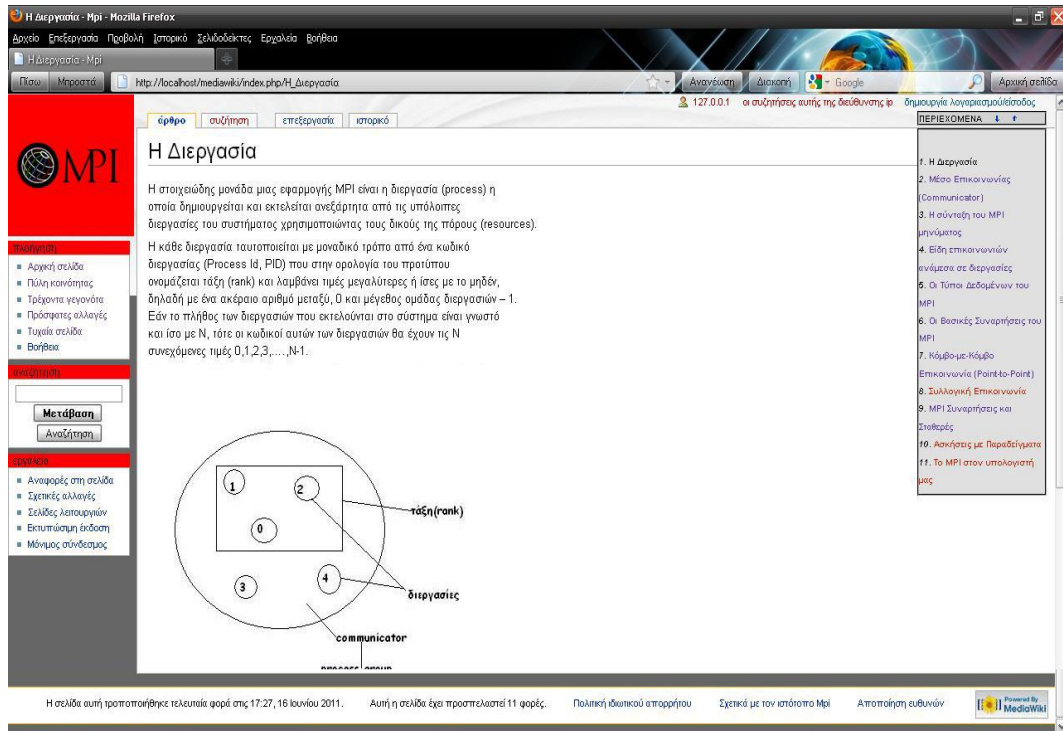
Ένα MPI πρόγραμμα αποτελείται από πολλές αυτόνομες διεργασίες που εκτελούνται ταυτόχρονα και η κάθε μια έχει τον δικό της χώρο διευθύνσεων και επικοινωνούν μέσω κλήσεων σε MPI διαδικασίες , αν και άλλου τύπου τρόποι επικοινωνίας όπως με χρήση κοινής μνήμης επιτρέπονται . Παρόλα αυτά στον παρόν πόνημα δεν θα ασχοληθούμε με άλλους τρόπους επικοινωνίας παρά μόνο με πέρασμα μνημάτων μέσω του προτύπου MPI .

Δεν είναι απαραίτητο η κάθε διεργασία να εκτελεί τον ίδιο κώδικα .

Όσον αφορά την εκτέλεση της κάθε διεργασίας , δεν ορίζεται κάποιο συγκεκριμένο μοντέλο . Κάθε διεργασία μπορεί να εκτελείται σειριακά ή να αποτελείται από πολλά νήματα (threads) τα οποία είναι δυνατόν να εκτελούνται ταυτόχρονα . Πάντως έχει ληφθεί μέριμνα για να είναι το MPI ασφαλές σε αυτόν τον τομέα των threads .

6.2.4 Η Διεργασία

Η στοιχειώδης μονάδα μιας εφαρμογής MPI είναι η διεργασία (process) η οποία δημιουργείται και εκτελείται ανεξάρτητα από τις υπόλοιπες διεργασίες του συστήματος χρησιμοποιώντας τους δικούς της πόρους (resources).

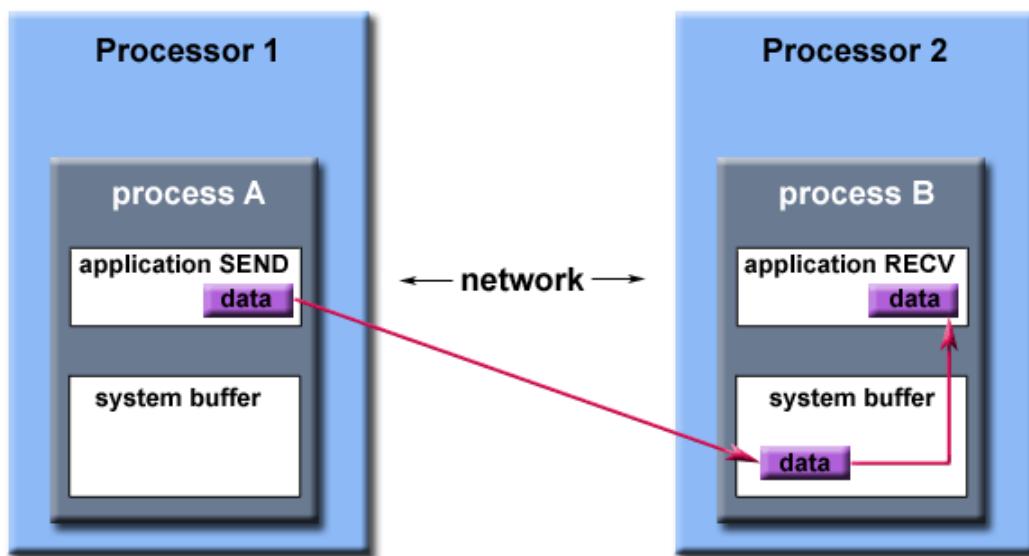


Εικόνα 55A: Η Διεργασία (εμφάνιση μέσα από το Wiki)

Η κάθε διεργασία ταυτοποιείται με μοναδικό τρόπο από ένα κωδικό διεργασίας (Process Id, PID) που στην ορολογία του προτύπου ονομάζεται τάξη (rank) και λαμβάνει τιμές μεγαλύτερες ή ίσες με το μηδέν, δηλαδή με ένα ακέραιο αριθμό μεταξύ, 0 και μέγεθος ομάδας διεργασιών – 1. Εάν το πλήθος των διεργασιών που εκτελούνται στο σύστημα είναι γνωστό και ίσο με N, τότε οι κωδικοί αυτών των διεργασιών θα έχουν τις N συνεχόμενες τιμές 0,1,2,3,.....,N-1. Ένα σύνολο N διατεταγμένων διεργασιών που χαρακτηρίζεται από τιμές τάξεων 0,1,2,....,N-1 είναι γνωστό ως ομάδα (group). Σε μια εφαρμογή MPI μπορούμε να ορίσουμε πολλές ομάδες διεργασιών κάθε μια από τις οποίες χαρακτηρίζεται από το δικό της κωδικό (group id). Ας σημειωθεί πως αυτός ο κωδικός είναι άγνωστος στο χρήστη, και υπό αυτή την έννοια μια ομάδα διεργασιών χαρακτηρίζεται ως ένα αδιαφανές αντικείμενο με άγνωστη εσωτερική δομή.

Εάν ο χρήστης επιθυμεί να προσπελάσει τις τιμές των ιδιοτήτων μιας ομάδας διεργασιών θα πρέπει να χρησιμοποιήσει ειδικές συναρτήσεις του MPI προκειμένου να ανακτήσει αυτές τις τιμές.

Ένα τελευταίο χαρακτηριστικό που αξίζει να αναφερθεί, είναι η δυνατότητα που έχει μια διεργασία να αποστείλει ένα μήνυμα στον εαυτό της. Στην περίπτωση αυτή ωστόσο απαιτείται η αποθήκευση του μηνύματος σε ένα χώρο ενδιάμεσης αποθήκευσης του συστήματος (system buffer). Αυτό σημαίνει πως το μήνυμα δεν θα μεταφερθεί απευθείας στην περιοχή μνήμης που έχει δεσμευθεί για την παραλαβή του, αλλά θα διατηρηθεί προσωρινά σε μία περιοχή ενδιάμεσης μνήμης. Ο τρόπος δέσμευσης αυτής της μνήμης καθώς και το μέγιστο μέγεθός της εξαρτώνται από την υλοποίηση που χρησιμοποιείται σε κάθε περίπτωση. Επομένως γενικά δεν συνίσταται η αποστολή μηνυμάτων από μια διεργασία στον εαυτό της, κάτι που ισχύει ιδιαίτερα για τις παρεμποδιστικές (blocking) επικοινωνίες που θα παρουσιαστούν στην επόμενη ενότητα.



Εικόνα 56: Μονοπάτι αποθήκευσης του μηνύματος

6.2.5 Μέσο Επικοινωνίας (Communicator)

Σημαντικό χαρακτηριστικό του προτύπου MPI είναι η έννοια του μέσου επικοινωνίας (communicator) δια της χρήσεως του οποίου λαμβάνει χώρα η ανταλλαγή μηνυμάτων ανάμεσα σε δύο διεργασίες. Αυτή η ενδιάμεση οντότητα χαρακτηρίζεται και αυτή από την ύπαρξη κάποιου κωδικού, η ανάκτηση του οποίου μπορεί να λάβει χώρα χρησιμοποιώντας κάποια από τις συναρτήσεις του MPI. Είναι προφανές πως για να είναι δυνατή η επικοινωνία ανάμεσα σε δύο διεργασίες αυτές θα πρέπει να ανήκουν στον ίδιο communicator.

Για τα περισσότερα προγράμματα MPI, αρκεί ένας βασικός communicator, ο οποίος ονομάζεται MPI_COMM_WORLD. Αυτός ο communicator δημιουργείται αυτόματα από το MPI και περιέχει όλες τις διεργασίες, οι οποίες δημιουργούνται, όταν αρχίζει η εκτέλεση του προγράμματος. Η χρήση αυτού του τρόπου επικοινωνίας διασφαλίζει το σωστό τρόπο διακίνησης της πληροφορίας και ελαχιστοποιεί την πιθανότητα εμφάνισης προβλημάτων όσον αφορά την αποστολή και λήψη των μηνυμάτων από τις διεργασίες της εφαρμογής.

Μιλώντας γενικά, ένας communicator ορίζεται ως ένα τοπικό αντικείμενο (local object) που χρησιμοποιείται για την αναπαράσταση ενός χώρου μέσα στον οποίο ανήκουν οι διεργασίες που έρχονται σε επικοινωνία (communication domain). Αυτός ο χώρος ορίζεται ως μια καθολική (global) και κατανεμημένη δομή που επιτρέπει στις διεργασίες μιας ομάδας να επικοινωνήσουν είτε μεταξύ τους, είτε με τις διεργασίες μιας άλλης ομάδας.

Στην περίπτωση κατά την οποία οι διεργασίες που επικοινωνούν ανήκουν στην ίδια ομάδα, ο communicator είναι γνωστός ως intracommunicator ενώ στην περίπτωση κατά την οποία οι διεργασίες που επικοινωνούν ανήκουν σε διαφορετικές ομάδες, ο communicator είναι γνωστός ως intercommunicator. Μια διεργασία μπορεί να συμμετάσχει σε πολλούς επικοινωνιακούς χώρους (communication domains). Με τον ίδιο τρόπο, δύο επικοινωνιακοί χώροι μπορεί να περιέχουν ομάδες διεργασιών που να περιέχουν κοινές διεργασίες ή να επικαλύπτονται πλήρως (να περιέχουν τις ίδιες διεργασίες ή η μια να είναι υποσύνολο ή υπερσύνολο της άλλης). Αυτό το χαρακτηριστικό επιτρέπει σε μια διεργασία να επικοινωνήσει με μια άλλη διεργασία δια μέσου δύο διαφορετικών και διακριτών μεταξύ

τους επικοινωνιακών χώρων. Αυτό σημαίνει πως μια διεργασία μπορεί να ανήκει ταυτόχρονα σε πολλούς επικοινωνιακούς χώρους, σε κάθε έναν από τους οποίους θα περιγράφεται και από διαφορετική τιμή τάξης.

6.2.6 Είδη επικοινωνιών ανάμεσα σε διεργασίες

Τα είδη επικοινωνιών ανάμεσα σε διεργασίες και συνεπώς οι MPI συναρτήσεις μπορούν να χωριστούν στις εξής κατηγορίες :

✓ **Επικοινωνία Κόμβου με Κόμβου :**

Ο βασικός μηχανισμός επικοινωνίας του MPI είναι η μετάδοση δεδομένων μεταξύ ενός ζεύγους διεργασιών (όπου η μια στέλνει και άλλη λαμβάνει). Αυτός ο τρόπος επικοινωνίας ονομάζεται επικοινωνία από σημείο σε σημείο (point to point communication) και αποτελεί το μοντέλο επικοινωνίας για ένα πολύ μεγάλο εύρος εφαρμογών MPI.

Εδώ οι συναρτήσεις του MPI εμφανίζονται σε δύο διαφορετικές μορφές που φέρουν τα ονόματα αναστέλλουσες-παρεμποδιστικές (blocking) και μη αναστέλλουσες-μη παρεμποδιστικές (non blocking) συναρτήσεις.

✓ **Μια συνάρτηση λέγεται παρεμποδιστική (blocking) :**

Όταν η κλήση της μέσα από κάποια διεργασία αναστέλλει τη λειτουργία αυτής της διεργασίας, που την καλεί, μέχρι την ολοκλήρωση της λειτουργίας της συνάρτησης που έχει κληθεί.

✓ **Μια συνάρτηση λέγεται μη παρεμποδιστική (non blocking) :**

Κατά την κλήση της δεν προκαλεί την αναστολή της λειτουργίας της διεργασίας.

✓ **Συλλογική Επικοινωνία :**

Χαρακτηρίζονται από την ύπαρξη περισσότερων από δυο διεργασιών. Θεωρητικά, για την πραγματοποίηση της επικοινωνίας πολλών διεργασιών, μπορούν να χρησιμοποιηθούν οι συναρτήσεις send και receive, οι οποίες χρησιμοποιούνται στην από κόμβο σε κόμβο επικοινωνία. Όμως έτσι αυξάνεται η πολυπλοκότητα του κώδικα. Δηλαδή αν έπρεπε μια διεργασία να στείλει το ίδιο μήνυμα σε 100 άλλες διεργασίες, θα χρειασθούν 100

κλήσεις από κόμβο σε κόμβο επικοινωνίας. Θα ήταν πολύ πιο εύκολο αν η αποστολή του μηνύματος στις 100 διεργασίες γίνει με την κλήση μιας συνάρτησης.

Εδώ χρησιμοποιείται η συλλογική επικοινωνία. Για παράδειγμα την αποστολή του ίδιου μηνύματος σε όλες τις διεργασίες μίας ομάδας (broadcast).

6.2.7 Η σύνταξη του MPI μηνύματος

Κάθε MPI-μήνυμα αποτελείται από δύο μέρη: από τον φάκελο (envelope) και από το κυρίως μήνυμα (message body).

Ο φάκελος ενός MPI-μηνύματος περιλαμβάνει την εξής πληροφορία:

- Το όνομα της διεργασίας-αφετηρίας, από την οποία στέλνεται το μήνυμα (source).
- Το όνομα της διεργασίας-προορισμού, στην οποία κατευθύνεται το μήνυμα (destination).
- Τον communicator, στον οποίον ανήκουν οι διεργασίες αφετηρίας και προορισμού (communicator).
- Μία ετικέτα (tag), η οποία μπορεί να χρησιμοποιηθεί προκειμένου η διεργασία-παραλήπτης να ξεχωρίσει το είδος του μηνύματος. Έτσι, αν ληφθούν δύο μηνύματα από την ίδια διεργασία, με τη χρήση της ετικέτας, ο παραλήπτης μπορεί να τα διακρίνει.

Για να ληφθεί ένα μήνυμα, πρέπει ο φάκελος του να περιέχει τις ίδιες τιμές για τις μεταβλητές source, tag και comm., με αυτές που καθορίζονται από τη διεργασία-παραλήπτη, στη συνάρτηση που λαμβάνει το μήνυμα. Στη θέση της source ή/και της tag μπορούμε να χρησιμοποιήσουμε τις σταθερές MPI_ANY_TAG, MPI_ANY_SOURCE. Με τη χρήση της ετικέτας MPI_ANY_TAG επιτρέπεται η λήψη μηνυμάτων με οποιοδήποτε tag. Η σταθερά MPI_ANY_SOURCE επιτρέπει τη λήψη μηνυμάτων από οποιοδήποτε αποστολέα. Όμως, δεν μπορούμε να χρησιμοποιήσουμε μία σταθερά για την comm. Ο communicator πρέπει πάντα να ορίζεται με σαφήνεια, π.χ MPI_COMM_WORLD

6.2.8 Οι Τύποι Δεδομένων του MPI

Αν και οι συναρτήσεις του προτύπου MPI μπορούν να κληθούν μέσα από ένα πρόγραμμα γραμμένο στη γλώσσα προγραμματισμού C (η γλώσσα προγραμματισμού Fortran επίσης υποστηρίζεται), οι τύποι δεδομένων των ορισμάτων των εν λόγω συναρτήσεων, δεν είναι αυτοί που χρησιμοποιούνται στη γλώσσα C, καθώς το MPI χρησιμοποιεί τους δικούς του τύπους δεδομένων. Ο επόμενος πίνακας περιέχει τους στοιχειώδεις τύπους δεδομένων που χρησιμοποιούνται από τις συναρτήσεις του MPI και τους τύπους δεδομένων της γλώσσας C που αντιστοιχούν στους τύπους δεδομένων του MPI.

Πίνακας 3: Τύποι Δεδομένων του MPI

The screenshot shows a web browser window with the URL http://localhost/mediawiki/index.php/O_Τύποι_Δεδομένων_του_MPI. The page title is "Οι Τύποι Δεδομένων του MPI". The main content area contains a table with two columns: "MPI Datatype" and "C Datatype". The table lists the following mappings:

MPI Datatype	C Datatype
MPI_CHAR	signed char
MPI_SHORT	signed short int
MPI_INT	signed int
MPI_LONG	signed long int
MPI_UNSIGNED_CHAR	unsigned char
MPI_UNSIGNED_SHORT	unsigned short int
MPI_UNSIGNED	unsigned int
MPI_UNSIGNED_LONG	unsigned long int
MPI_FLOAT	float
MPI_DOUBLE	double
MPI_LONG_DOUBLE	long_double

Below the table, there is a footer with the text: "Η σελίδα αυτή προπονήθηκε τελευταία φορά στις 17:27, 16 Ιουνίου 2011. Αυτή η σελίδα έχει προσπελάσει 11 φορές. Πολιτική κινητικού απορρήτου Σχετικά με τον ιστότοπο MPI Αποποίηση ευθυνών Powered By MediaWiki".

Οι τύποι δεδομένων MPI_BYTE και MPI_PACKED δεν αντιστοιχούν σε κάποιο τύπο της C. Το MPI_BYTE παριστάνει 8 δυαδικά ψηφία . Χρησιμοποιείται, όταν θέλουμε να αποστείλουμε μηνύματα με bit-πεδία, όπου η τιμή κάθε bit παριστάνει την τιμή μιας σημαίας (flag). Αυτός ο τύπος μπορεί να χρησιμοποιηθεί για την αποστολή πληροφοριών ελέγχου στον παραλήπτη. Αν και αυτό μπορεί να γίνει και από το MPI_INT, το MPI_BYTE είναι μικρότερο, άρα η μετάδοση θα γίνει γρηγορότερα.

Ο τύπος MPI_PACKED χρησιμοποιείται για την αποστολή και τη λήψη συνεπτυγμένων μηνυμάτων.

6.3 ΟΙ ΒΑΣΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ MPI

Το πρότυπο MPI, μπορεί να υλοποιηθεί χρησιμοποιώντας μόνο ένα μικρό πλήθος συναρτήσεων. Όλα τα ονόματα των συναρτήσεων και σταθερών του MPI αρχίζουν με το πρόθεμα MPI_. Ο πρώτος χαρακτήρας των ονομάτων των ρουτινών στη C γράφεται με κεφαλαίο γράμμα ενώ οι υπόλοιποι με μικρά. Τα ονόματα των σταθερών γράφονται με κεφαλαία γράμματα. Για να χρησιμοποιηθεί το MPI σε ένα πρόγραμμα C, πρέπει στην αρχή του να υπάρχει η πάρακατω δήλωση προεπεξεργαστή

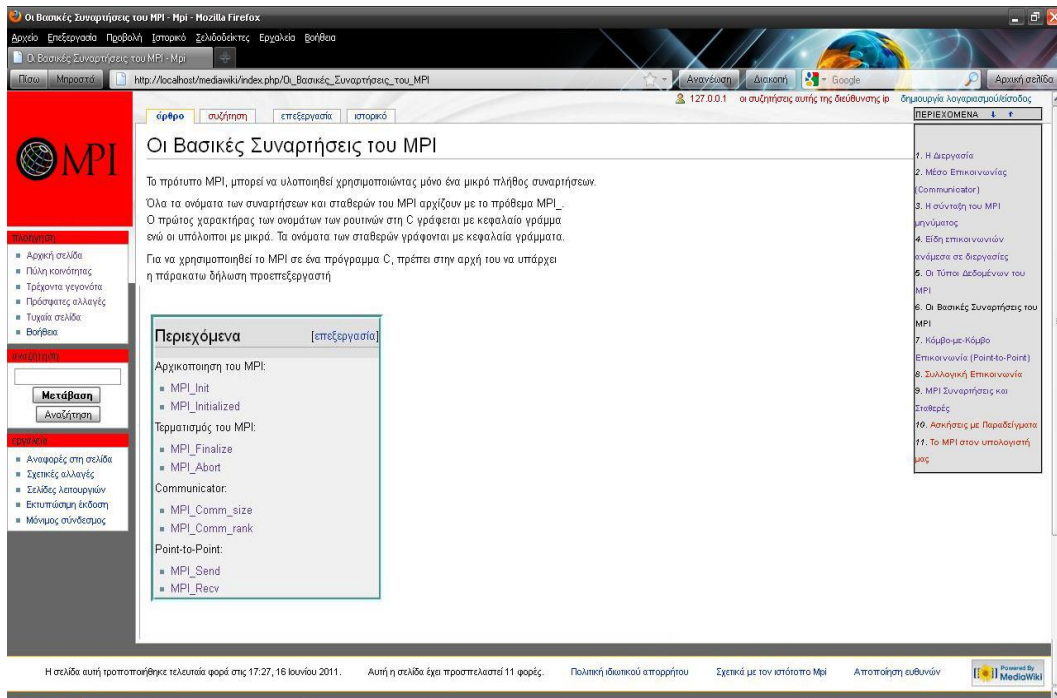
```
include "mpi.h
```

είτε

```
include <mpi.h>
```

ανάλογα με τον αν το αρχείο mpi.h βρίσκεται στον τρέχοντα κατάλογο ή στον κατάλογο που περιέχει τα [[header]] αρχεία του μεταγλωτιστή . Οι συναρτήσεις του MPI, ορίζονται στο αρχείο επικεφαλίδας του mpi.h και για αυτό θα πρέπει να συμπεριλαμβάνεται στον πηγαίο κώδικα, όλων των εφαρμογών που χρησιμοποιούν το πρότυπο διαβίβασης μνημάτων.

Για την διαχείριση του περιβάλλοντος το MPI χρησιμοποιεί ρουτίνες για μια κατάταξη των σκοπών, όπως η αρχικοποίηση και ο τερματισμός του περιβάλλοντος του MPI.



Εικόνα: Βασικές Συναρτήσεις του MPI

6.3.1 Αρχικοποίηση του MPI

✓ MPI_Init

Πριν την χρήση οποιασδήποτε εντολής MPI, θα πρέπει να λάβει χώρα η αρχικοποίηση αυτού του περιβάλλοντος, η οποία θα γίνει χρησιμοποιώντας την συνάρτηση MPI_Init.

Αυτό σημαίνει πως αυτή η συνάρτηση, θα πρέπει να καλείται πριν απο οποιαδήποτε συνάρτηση MPI. Η MPI_Init πρέπει να κληθεί μόνο μια φορά και πριν από κάθε άλλη κλήση συνάρτησης MPI. Δεύτερη κλήση της αποτελεί λάθος. Η κλήση της γίνεται με τις παραμέτρους, που περνιούνται αυτόματα απο τον μεταγλωττιστή της C στη main, δηλαδή,

```
MPI_Init(&[[argc]], &[[argv]]);
```

Μετά την κλήση της MPI_Init, όλες οι διεργασίες που δημιουργήθηκαν, θα έχουν ένα αντίγραφο των ορισμάτων της main. Ο αριθμός των διεργασιών δίνεται ως παράμετρος στη γραμμή εντολών, ενώ ο τρόπος της ανάθεσης των διεργασιών στους επεξεργαστές εξαρτάται απο την συγκεκριμένη υλοποίηση του MPI.

✓ **MPI_Initialized**

Δείχνει εάν το `[[MPI_Init]]` έχει καλεστεί και επιστρέφει μια σημαία με λογικές τιμές 1(αλήθεια)εάν έχει κληθεί η `[[MPI_Init]]` ή 0(ψέμα). Το MPI απαιτεί ότι το `[[MPI_Init]]` έχει καλεστεί μια και μόνο μια φορά σε ένα πρόγραμμα . Αυτό μπορεί να δημιουργήσει ένα πρόβλημα για μια ενότητα που θέλει να χρησιμοποιήσει MPI και είναι έτοιμο να καλέσει `[[MPI_Init]]` εάν είναι απαραίτητο και έτσι το `MPI_Initialized` λύνει το πρόβλημα αυτό.

Είναι χρήσιμη σε προγράμματα, όπου ένα μέρος τους εκτελείται τοπικά σε έναν επεξεργαστή, ενώ το υπόλοιπο εκτελείται παράλληλα με τη βοήθεια του MPI.

Η δήλωση είναι : `int MPI_Initialized(int *flag);`

Το αποτέλεσμα επιστρέφεται στην παράμετρο `flag`.

6.3.2 Τερματισμός του MPI

✓ **MPI_Finalize**

Η συνάρτηση `MPI_Finalize` είναι η τελευταία συνάρτηση που θα πρέπει να κληθεί και ο ρόλος της είναι ο τερματισμός της λειτουργίας του προτύπου MPI. Αυτό σημαίνει πως κάθε φορά που χρησιμοποιούμε συναρτήσεις MPI μέσα από ένα πρόγραμμα C, θα πρέπει υποχρεωτικά να χρησιμοποιούμε τη συνάρτηση `MPI_Finalize` για να απελευθερώσουμε τη μνήμη που δεσμεύεται από τις δομές δεδομένων του προτύπου που είχαν δεσμευθεί απο την `[[MPI_Init]]` και γενικά για να τερματίσουμε τη λειτουργία του.

Η συνάρτηση `MPI_Finalize` καλείται χωρίς ορίσματα **`MPI_Finalize();`**

✓ **MPI_Abort**

Τερματίζει όλες τις MPI διεργασίες που σχετίζονται με το συγκεκριμένο communicator.

Στις περισσότερες εφαρμογές τερματίζει όλες τις διεργασίες ασχέτως από το καθορισμένο από το προγραμματιστή communicator.

Η σύνταξη της είναι η παρακάτω:

```
int MPI_Abort([[MPI_Comm]] [[comm]], int [[err]])
```

Ο κωδικός λάθους δεν χρησιμοποιείται, στις περισσότερες υλοποιήσεις του MPI. Αν δοθεί `comm=MPI_COMM_WORLD`, τότε προκαλείται τερματισμός, ολόκληρου του παράλληλου προγράμματος.

6.3.3 Communicator

MPI Comm rank

Η συνάρτηση `MPI_Comm_rank` επιστρέφει την τάξη της τρέχουσας διεργασίας.

Η κλήση της συνάρτησης χαρακτηρίζεται από τη σύνταξη

```
MPI_Comm_rank (MPI_comm comm, int * rank)
```

όπου η μεταβλητή `comm` χρησιμοποιείται για τον καθορισμό του Communicator στον οποίο ανήκει η διεργασία, ενώ η μεταβλητή `rank` είναι μια ακέραια μεταβλητή που μετά την επιτυχή κλήση της συνάρτησης θα περιέχει την τάξη της τρέχουσας διεργασίας. Τυπικό παράδειγμα χρήσης της συνάρτησης `MPI_Comm_rank` είναι η κλήση της με τη μορφή

```
MPI_Comm_rank (MPI_COMM_WORLD, &rank)
```

η οποία επιστρέφει στη μεταβλητή `rank` την τάξη της τρέχουσας διεργασίας που ανήκει στον προεπιλεγμένο Communicator `MPI_COMM_WORLD`.

MPI Comm size

Η συνάρτηση `MPI_Comm_size` επιστρέφει το μέγεθος ενός Communicator, δηλαδή το πλήθος των διεργασιών που περιλαμβάνονται σε αυτόν. Η κλήση αυτής της συνάρτησης χαρακτηρίζεται από τη σύνταξη

```
MPI_Comm_size (MPI_comm comm, int * size)
```

όπου η μεταβλητή `comm` χρησιμοποιείται για τον καθορισμό του Communicator του οποίου το μέγεθος θέλουμε να ανακτήσουμε, ενώ η μεταβλητή `size` είναι μια ακέραια μεταβλητή που μετά την επιτυχή κλήση της συνάρτησης θα περιέχει το πλήθος των διεργασιών που

περιλαμβάνονται στον τρέχοντα Communicator. Τυπικό παράδειγμα χρήσης της συνάρτησης `MPI_Comm_size` είναι η κλήση της με τη μορφή

```
MPI_Comm_size (MPI_COMM_WORLD, &size)
```

η οποία επιστρέφει στη μεταβλητή `size` το πλήθος των διεργασιών που ανήκουν στον προεπιλεγμένο Communicator `MPI_COMM_WORLD`.

6.3.4 Point-to-Point

✓ **MPI_Send**

Η συνάρτηση `MPI_Send` χρησιμοποιείται για την αποστολή ενός μηνύματος σε κάποια από τις διεργασίες που περιλαμβάνονται στον τρέχοντα Communicator. Το πρωτότυπο της συνάρτησης `MPI_Send` έχει τη μορφή

```
MPI_Send (void * buf, int count, MPI_Datatype dataType,  
int destRank, int messageTag, MPI_Comm comm)
```

Από τα ορίσματα αυτά της συνάρτησης, τα τρία πρώτα ταυτοποιούν τα χαρακτηριστικά του μηνύματος προς αποστολή. Πιο συγκεκριμένα:

- το πρώτο όρισμα περιέχει τη διεύθυνση της περιοχής μνήμης στην οποία βρίσκονται τα δεδομένα προς αποστολή,
- το δεύτερο όρισμα περιέχει το πλήθος των στοιχείων προς αποστολή,
- ενώ το τρίτο όρισμα περιέχει τον τύπο δεδομένων για αυτά τα στοιχεία.

Τα επόμενα τρία ορίσματα αφορούν (με τη σειρά με την οποία αναγράφονται) :

- την τάξη της διεργασίας προς την οποία πρόκειται να αποσταλεί το μήνυμα (διεργασία παραλήπτης),
- την ετικέτα του μηνύματος που μας επιτρέπει να ξεχωρίσουμε αυτό το μήνυμα από τα υπόλοιπα (εάν δεν θέλουμε να καθορίσουμε κάποια ετικέτα μπορούμε να καταχωρήσουμε σε αυτό το όρισμα την τιμή `MPI_ANY_TAG`)
- και τον Communicator στον οποίο ανήκουν οι διεργασίες αποστολέας

και παραλήπτης.

Η ετικέτα του μηνύματος είναι ένας ακέραιος αριθμός που παίρνει τιμές στο διάστημα $[0, UB]$, όπου η τιμή της σταθεράς UB εξαρτάται από την υλοποίηση που χρησιμοποιείται σε κάθε περίπτωση.

Προκειμένου να ανακτήσουμε την τιμή αυτής της σταθεράς, θα πρέπει να ζητήσουμε από το σύστημα να μας επιστρέψει την τιμή του μεγέθους `MPI_TAG_UB`. Είναι προφανές πως εφ' όσον η ετικέτα είναι ένας ακέραιος αριθμός, η τιμή αυτού του μεγέθους δεν μπορεί να είναι μικρότερη από 32767.

Τυπικό παράδειγμα χρήσης της συνάρτησης `MPI_Send` είναι η κλήση της με τη μορφή

```
MPI_Send (buffer, 100, MPI_CHAR, 1, 0, MPI_COMM_WORLD)
```

η οποία προκαλεί την αποστολή μιας συμβολοσειράς 100 χαρακτήρων στη διεργασία με τιμή τάξης ίση με 1. Αυτή η συμβολοσειρά περιλαμβάνεται σε ένα μήνυμα η ετικέτα του οποίου έχει την τιμή 0, ενώ και οι δύο διεργασίες ανήκουν στον προεπιλεγμένο Communicator `MPI_COMM_WORLD`.

Είναι προφανές πως η περιοχή της ενδιάμεσης μνήμης στην οποία βρίσκονται αποθηκευμένα τα δεδομένα προς αποστολή, αποτελείται από `count` συνεχόμενα στοιχεία τύπου `dataType`, με το πρώτο όρισμα της συνάρτησης `MPI_Send` (που περιγράφεται από το όνομα `buf`) να είναι ένας δείκτης προς το πρώτο από αυτά τα συνεχόμενα στοιχεία. Εφ' όσον ο τύπος δεδομένων για αυτά τα συνεχόμενα στοιχεία είναι γνωστός, είναι πάρα πολύ εύκολο να υπολογίσουμε το συνολικό μέγεθος των δεδομένων προς αποστολή. Στο παραπάνω παράδειγμα η ενδιάμεση περιοχή μνήμης που περιέχει τα δεδομένα του μηνύματος, περιλαμβάνει 100 συνεχόμενους χαρακτήρες, και επομένως το μέγεθος της περιοχής `buffer` είναι ίσο με 100 bytes.

Είναι προφανές πως εφ' όσον το δεύτερο όρισμα της συνάρτησης `MPI_Send` αναφέρεται στο πλήθος των στοιχείων που περιλαμβάνονται στην ενδιάμεση μνήμη, και όχι στο μέγεθός της σε bytes, είναι δυνατή η

μεταφορά των προγραμμάτων MPI και σε αρχιτεκτονικές που δεν είναι οργανωμένες με βάση το byte.

Το μήνυμα που στέλνεται από MPI_Send μπορεί να παραληφθεί από MPI_Recv είτε MPI_Irecv.

✓ **MPI_Recv**

Η συνάρτηση MPI_Recv χρησιμοποιείται για την παραλαβή ενός μηνύματος από κάποια από τις διεργασίες που ανήκουν στον τρέχοντα Communicator. Η χρήση της συνάρτησης MPI_Recv χαρακτηρίζεται από μια κλήση της μορφής

```
MPI_Recv (void * buf, int count, MPI_Datatype dataType,  
int sourceRank, int messageTag, MPI_Comm comm,  
[[MPI_Status]] status)
```

και επομένως η εν λόγω συνάρτηση χρησιμοποιείται με τον ίδιο σχεδόν τρόπο όπως η MPI_Send. Στην περίπτωση αυτή βέβαια το τέταρτο όρισμα αναφέρεται στην τάξη της διεργασίας αποστολέα, ενώ υπάρχει και ένα επιπλέον όρισμα που δεν εμφανίζεται στη συνάρτηση MPI_Send, και το οποίο επιτρέπει την ανάκτηση πληροφοριών σχετικά με την κατάσταση παραλαβής του μηνύματος.

Τυπικό παράδειγμα χρήσης της συνάρτησης MPI_Recv είναι η κλήση της με τη μορφή

```
MPI_Recv (buffer, 50, MPI_DOUBLE, 4, MPI_ANY_TAG,  
MPI_COMM_WORLD, status)
```

η οποία διαβάζει ένα μήνυμα που περιέχει 50 πραγματικούς αριθμούς διπλής ακρίβειας από τη διεργασία με τιμή τάξης ίση με 4. Οι δύο διεργασίες του εν λόγω παραδείγματος ανήκουν στον προεπιλεγμένο Communicator MPI_COMM_WORLD ενώ η ετικέτα που μηνύματος που

παραλαμβάνεται δεν μας ενδιαφέρει, και για το λόγο αυτό ως τιμή στο αντίστοιχο όρισμα χρησιμοποιείται η σταθερά `MPI_ANY_TAG`.

Είναι προφανές πως η επιτυχής παραλαβή ενός μηνύματος απαιτεί τη δέσμευση του κατάλληλου σε κάθε περίπτωση αποθηκευτικού χώρου. Πιο συγκεκριμένα, το μέγεθος του μηνύματος προς παραλαβή θα πρέπει να είναι μικρότερο ή ίσο με το μέγεθος του χώρου ενδιάμεσης αποθήκευσης το όνομα του οποίου περνάμε ως πρώτο όρισμα στη συνάρτηση `MPI_Recv`. Εάν το μέγεθος του μηνύματος (το οποίο ας υποθέσουμε πως είναι ίσο με N) είναι μικρότερο από το μέγεθος του αποθηκευτικού χώρου, το μήνυμα θα καταλάβει τις N πρώτες θέσεις της μνήμης, ενώ το υπόλοιπο τμήμα του ενδιάμεσου αποθηκευτικού χώρου θα παραμείνει αμετάβλητο. Αντίθετα εάν το μήνυμα είναι μεγαλύτερο από το μέγεθος του εν λόγω χώρου, η παραλαβή του δεν θα ολοκληρωθεί σωστά, και η `MPI_Recv` θα επιστρέψει ένα κωδικό σφάλματος.

Το μήνυμα που στέλνεται από `MPI_Recv` μπορεί να παραληφθεί `MPI_Send` είτε `MPI_Isend`

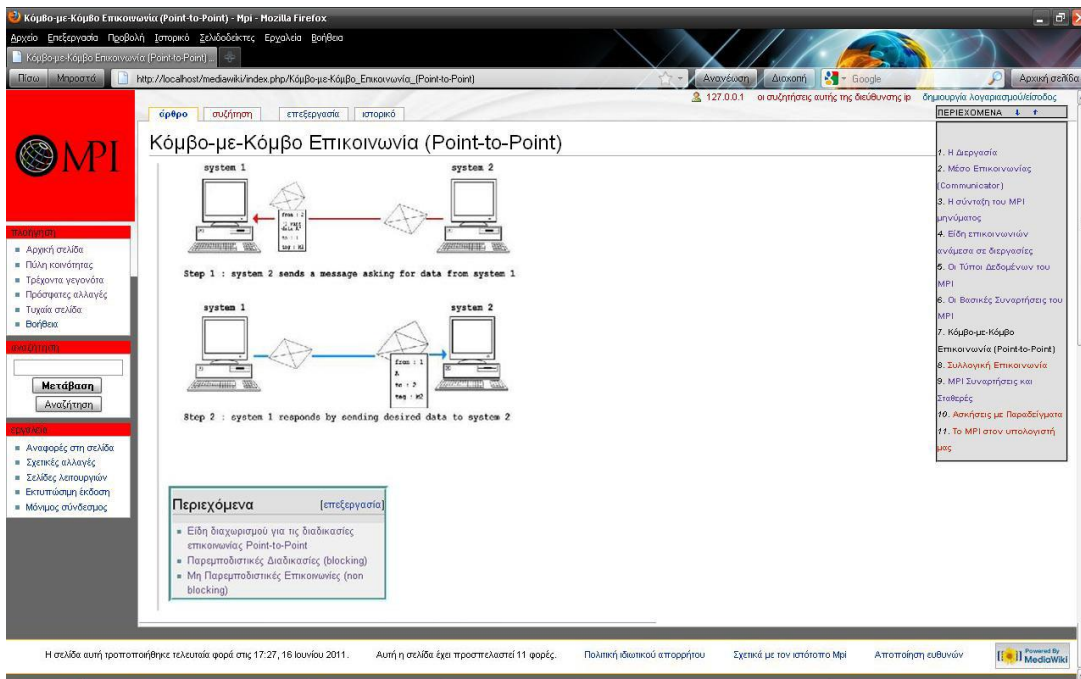
6.4 KOMBO - ME – KOMBO ΕΠΙΚΟΙΝΩΝΙΑ (Point-to-Point)

Η πιο βασική μορφή επικοινωνίας στο πρότυπο MPI – και φυσικά σε κάθε σχετικό πρότυπο ανταλλαγής μηνυμάτων – και τα θεμέλια πάνω στα οποία χτίζεται, είναι η επικοινωνία μεταξύ δύο σημείων – διεργασιών. Η point-to-point επικοινωνία ουσιαστικά είναι η αντιγραφή ενός κομματιού δεδομένων από τον χώρο διευθύνσεων μιας διεργασίας στον χώρο διευθύνσεων μιας άλλης. Η επικοινωνία αυτή είναι με την μορφή συνεργασίας και πρέπει και οι δύο συμβαλλόμενες διεργασίες να συμμετάσχουν. Η διεργασία που επιθυμεί να στείλει δεδομένα στην άλλη, καλεί την MPI ρουτίνα αποστολής δεδομένων `send`, ενώ η διεργασία που θα δεχτεί τα δεδομένα καλεί την αντίστοιχη `receive`.

Ένα άλλο πολύ σημαντικό θέμα στην point-to-point επικοινωνία είναι η αναγνώριση των πακέτων που μεταδίδονται. Για παράδειγμα μπορεί ταυτόχρονα δύο ή περισσότερες διεργασίες να στείλουν δεδομένα σε μια συγκεκριμένη διεργασία. Η τελευταία θα πρέπει να έχει ένα τρόπο για να

ξεχωρίζει τα πακέτα δεδομένων που παραλαμβάνει καθώς και την διεργασία που τα έστειλε έτσι ώστε να υπάρχει έλεγχος στην διαδικασία λήψης μηνυμάτων . Η αναγνώριση αυτή των μηνυμάτων γίνεται με διπλό τρόπο : το κάθε μήνυμα συμπεριλαμβάνει και έναν αναγνωριστικό αριθμό που καλείτε **tag** και στις ρουτίνα receive προσδιορίζεται ο βαθμός της διεργασίας που έστειλε τα δεδομένα . Έτσι , εαν για παράδειγμα δύο διεργασίες στείλουν ταυτόχρονα πακέτα δεδομένων στην ίδια διεργασία , η τελευταία θα μπορεί να ξεχωρίσει τα πακέτα και να ελέγξει την παραλαβή τους χωρίς λάθη .

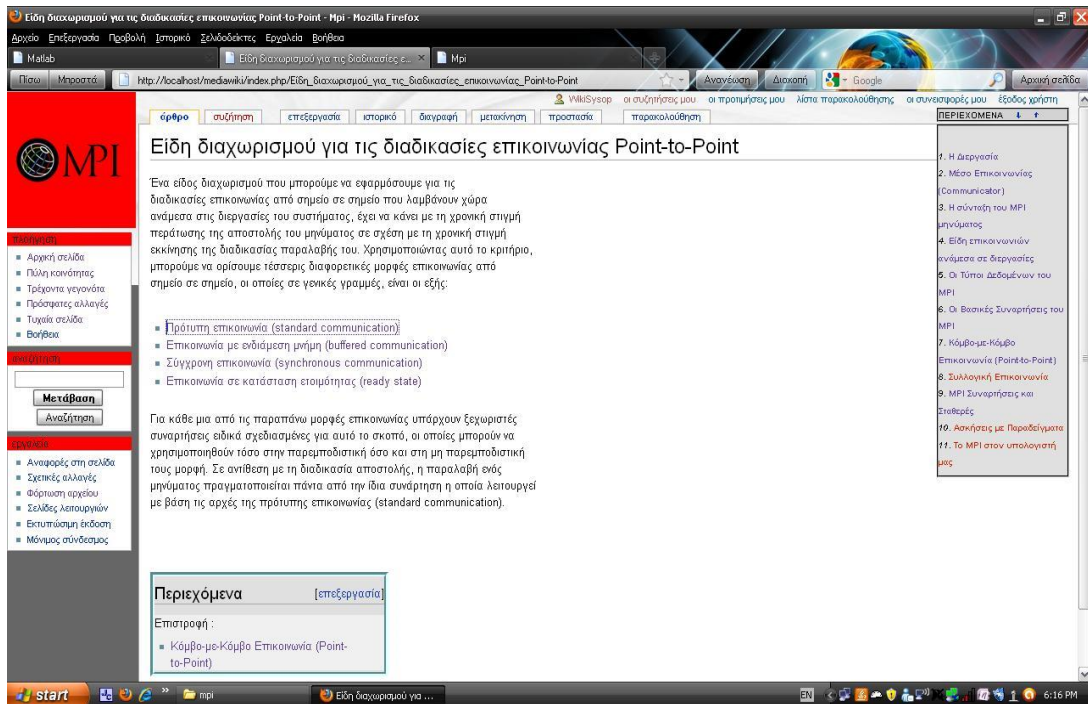
Παρόλα αυτά , μέχρι την εμφάνιση του MPI υπήρχε ένα πολύ σημαντικό πρόβλημα : ο μηχανισμός αυτός ήταν ανεπαρκής γιατί αυτή η αυθαιρεσία στον ορισμό των tags σημαίνει οτι σε όλο το πρόγραμμα θα πρέπει να χρησιμοποιούνται με ένα συγκεκριμένο και συνεπή τρόπο ώστε να αποφεύγονται τα λάθη . Ακόμα και έτσι όμως υπάρχει το πολύ σοβαρό πρόβλημα των παράλληλων βιβλιοθηκών , που τα μηνύματα που αυτές χρησιμοποιούν δεν θα πρέπει να συγχέονται με τα μηνύματα της εκάστοτε εφαρμογής που χρησιμοποιεί την βιβλιοθήκη . Στο MPI το ζήτημα αυτό διευθετήθηκε με την εισαγωγή της έννοιας των **contexts** (πλαίσια) . Αυτά καθορίζονται απο το σύστημα κατά την εκτέλεση του προγράμματος και χρησιμοποιούνται για την αντιστοίχιση των μηνυμάτων . Διαφέρουν απο τη λογική των tags αφού διανέμονται απο το σύστημα και όχι απο τον χρήστη .



Εικόνα 57: point-to-point

6.4.1 Είδη διαχωρισμού για τις διαδικασίες επικοινωνίας Point-to-Point

Ένα είδος διαχωρισμού που μπορούμε να εφαρμόσουμε για τις διαδικασίες επικοινωνίας από σημείο σε σημείο που λαμβάνουν χώρα ανάμεσα στις διεργασίες του συστήματος, έχει να κάνει με τη χρονική στιγμή περάτωσης της αποστολής του μηνύματος σε σχέση με τη χρονική στιγμή εκκίνησης της διαδικασίας παραλαβής του. Χρησιμοποιώντας αυτό το κριτήριο, μπορούμε να ορίσουμε τέσσερις διαφορετικές μορφές επικοινωνίας από σημείο σε σημείο, οι οποίες σε γενικές γραμμές, είναι οι εξής:



✓ **Πρότυπη επικοινωνία (standard communication)**

Σε αυτόν τον τύπο επικοινωνίας η απόφαση για τη δέσμευση ή όχι ενδιάμεσης περιοχής μνήμης στην οποία θα αποθηκευτεί προσωρινά το εξερχόμενο μήνυμα, λαμβάνεται από το MPI και όχι από το χρήστη. Στην περίπτωση κατά την οποία πραγματοποιηθεί αυτή η διαδικασία της ενδιάμεσης αποθήκευσης του μηνύματος (buffer), η διαδικασία αποστολής του δύναται να περατωθεί ακόμη και εάν η αντίστοιχη διαδικασία παραλαβής του μηνύματος δεν έχει ξεκινήσει ακόμη. Στην αντίθετη περίπτωση όμως, κατά την οποία δεν είναι διαθέσιμη η περιοχή ενδιάμεσης μνήμης, η διαδικασία αποστολής θα ολοκληρωθεί μόνο όταν έχει ξεκινήσει η αντίστοιχη διαδικασία παραλαβής και το μήνυμα έχει μεταφερθεί στην περιοχή αποθήκευσης της διεργασίας παραλήπτη.

✓ **Επικοινωνία με ενδιάμεση μνήμη (buffered communication)**

Σε αυτόν τον τύπο επικοινωνίας η διαδικασία αποστολής μπορεί να ξεκινήσει ανεξάρτητα από το εάν έχει ξεκινήσει η αντίστοιχη διαδικασία παραλαβής, ενώ μπορεί ακόμη και να ολοκληρωθεί ανεξάρτητα από την εκκίνηση ή όχι της τελευταίας διαδικασίας. Στην περίπτωση αυτή ωστόσο είναι πιθανή η απαίτηση της τοπικής αποθήκευσης του εξερχόμενου μηνύματος κάτι που σημαίνει πως η εφαρμογή θα πρέπει να μεριμνήσει για

την αναζήτηση και δέσμευση του κατάλληλου αποθηκευτικού χώρου. Εάν η μνήμη που δεσμεύεται για το λόγο αυτό είναι ανεπαρκής όσον αφορά το μέγεθός της και δεν δύναται να διατηρήσει το διακινούμενο μήνυμα, λαμβάνει χώρα εμφάνιση σφάλματος. Στην αντίθετη περίπτωση το μήνυμα θα μεταφερθεί με επιτυχία στη διεργασία παραλήπτη, κάτι που μας επιτρέπει να αποδεσμεύσουμε πλέον αυτή την τοπική περιοχή μνήμης που είχε δεσμευθεί για τη διακίνηση του μηνύματος. Η διάθεση των buffer ελέγχεται από τον χρήστη μέσω κλήσεων σε ειδικές ρουτίνες του προτύπου όπως: MPI_Bsend , MPI_lbsend

✓ **Σύγχρονη επικοινωνία (synchronous communication)**

Σε αυτόν τον τύπο επικοινωνίας μια διαδικασία αποστολής μπορεί να ξεκινήσει ανεξάρτητα από την εκκίνηση της διαδικασίας παραλαβής από τη διεργασία παραλήπτη. Ωστόσο στην περίπτωση αυτή η αποστολή θα ολοκληρωθεί με επιτυχία μόνο όταν η διαδικασία παραλαβής έχει ξεκινήσει. Στην περίπτωση αυτή είναι δυνατή η χρήση του αποθηκευτικού χώρου της διαδικασίας αποστολέα για την αποστολή του επόμενου μηνύματος. Εδώ χρησιμοποιούμε τις ειδικές ρουτίνες πρότυπου : MPI_Ssend, MPI_Issend

✓ **Επικοινωνία σε κατάσταση ετοιμότητας (ready state)**

Σε αυτόν τον τύπο επικοινωνίας η διαδικασία επικοινωνίας μπορεί να ξεκινήσει μόνο όταν έχει ξεκινήσει και η αντίστοιχη διαδικασία παραλαβής. Στην αντίθετη περίπτωση η επικοινωνία χαρακτηρίζεται από την εμφάνιση σφαλμάτων και από εξαιρετικά χαμηλό βαθμό αξιοπιστίας. Εδώ χρησιμοποιούμε τις ειδικές ρουτίνες πρότυπου : MPI_Rsend, MPI_lrsend

6.4.2 Παρεμποδιστικές Διαδικασίες (blocking)

Η ανταλλαγή δεδομένων ανάμεσα σε δύο διεργασίες προϋποθέτει την εκκίνηση μιας διαδικασίας αποστολής από τη διεργασία αποστολέα και την εκκίνηση μιας διαδικασίας παραλαβής από τη διεργασία παραλήπτη. Αυτό το είδος της μεταφοράς μπορεί να πραγματοποιηθεί με πολλούς διαφορετικούς τρόπους ανάλογα με τις ανάγκες που υφίστανται σε κάθε περίπτωση. Μιλώντας γενικά, υπάρχουν δύο διαφορετικές προσεγγίσεις τις οποίες μπορούμε να ακολουθήσουμε: είτε θα ελαχιστοποιήσουμε τις

διαδικασίες αντιγραφής και ενδιάμεσης αποθήκευσης των διακινούμενων δεδομένων εξαναγκάζοντας τη διεργασία παραλαβής να ξεκινήσει αμέσως μετά την περάτωση της διαδικασίας αποστολής, είτε θα ελαχιστοποιήσουμε το χρονικό διάστημα αναστολής της λειτουργίας των διεργασιών – ας μην ξεχνάμε πως αναφερόμαστε σε παρεμποδιστικές διαδικασίες οι οποίες αναστέλλουν την εκτέλεση των διεργασιών που τις χρησιμοποιούν μέχρι την ολοκλήρωσή τους – αντιγράφοντας τα διακινούμενα δεδομένα σε ενδιάμεσους χώρους αποθήκευσης του συστήματος (system buffers) από όπου στη συνέχεια μπορούν να παραληφθούν από τη διεργασία παραλήπτη.

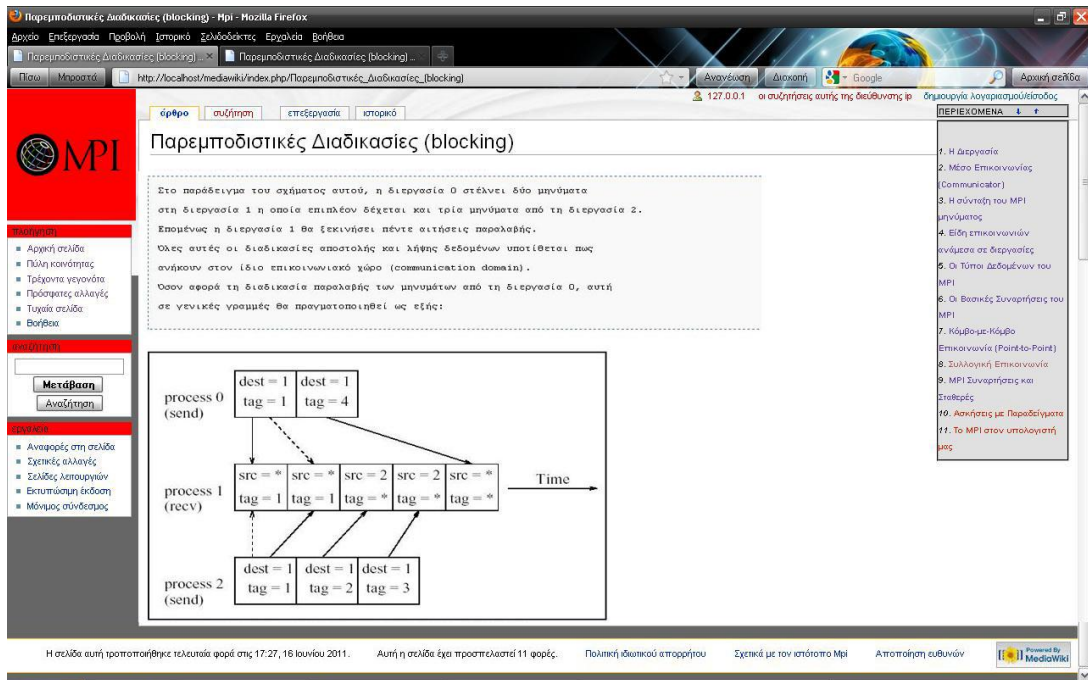
Σε περιπτώσεις κατά τις οποίες η εφαρμογή μας δεν έχει σχεδιαστεί σωστά, οι παρεμποδιστικές διεργασίες μπορούν όπως έχουμε ήδη αναφέρει να οδηγήσουν στην εμφάνιση αδιεξόδων (deadlocks) που χαρακτηρίζονται από την επ' αόριστον αναστολή της λειτουργίας όλων των διεργασιών του συστήματος. Τυπική αιτία εμφάνισης αδιεξόδων είναι οι τεράστιες απαιτήσεις σε μνήμη μιας διεργασίας: εάν υπάρχει η αναγκαία ποσότητα μνήμης, η εφαρμογή λειτουργεί κανονικά, αλλά όταν το μέγεθος των μηνυμάτων και των διακινούμενων δεδομένων αυξάνει απεριόριστα, είναι δυνατόν να εμφανιστούν αυτού του είδους οι προβληματικές καταστάσεις.

Για το λόγο αυτό, το MPI στηρίζεται στη λογική πως ένα ασφαλές πρόγραμμα θα πρέπει να μπορεί να ολοκληρωθεί με επιτυχία σε κάθε περίπτωση, ανεξάρτητα από το μέγεθος και την πολιτική δέσμευσης της απαιτούμενης μνήμης. Αυτό σημαίνει πως η χρήση ενδιάμεσης μνήμης θα επηρεάσει πιθανόν την απόδοση του προγράμματος, αλλά σε καμιά περίπτωση δεν θα πρέπει να είναι η αιτία της μη επιτυχούς εκτέλεσής του.

Ένα δεύτερο σημαντικό ζήτημα που συσχετίζεται με τις παρεμποδιστικές μορφές επικοινωνίας, είναι η σειρά με την οποία λαμβάνει χώρα η ανάγνωση και η παραλαβή των μηνυμάτων από τις διεργασίες της εφαρμογής. Εάν μια διεργασία αρχίσει να στέλνει πολλά μηνύματα το ένα πίσω από το άλλο, αυτά μπορεί να θεωρηθούν πως συνιστούν μια ταξινομημένη ακολουθία μηνυμάτων.

Με εντελώς ανάλογο τρόπο, οι λειτουργίες παραλαβής που ξεκινούν από μία διεργασία, μπορεί να θεωρηθούν πως ανήκουν και αυτές σε μια δεύτερη ταξινομημένη ακολουθία. Η κατάσταση αυτή παρουσιάζεται στο

επόμενο σχήμα. Στο παράδειγμα του σχήματος αυτού, η διεργασία 0 στέλνει δύο μηνύματα στη διεργασία 1 η οποία επιπλέον δέχεται και τρία μηνύματα από τη διεργασία 2. Επομένως η διεργασία 1 θα ξεκινήσει πέντε αιτήσεις παραλαβής. Όλες αυτές οι διαδικασίες αποστολής και λήψης δεδομένων υποτίθεται πως ανήκουν στον ίδιο επικοινωνιακό χώρο (communication domain). Όσον αφορά τη διαδικασία παραλαβής των μηνυμάτων από τη διεργασία 0, αυτή σε γενικές γραμμές θα πραγματοποιηθεί ως εξής:



Εικόνα 58: blocking

Το πρώτο μήνυμα της διεργασίας 0 και το πρώτο μήνυμα της διεργασίας 2 μπορούν να παραληφθούν από τη διεργασία 1 με οποιαδήποτε σειρά, καθώς αντιστοιχούν στις δύο πρώτες διαδικασίες παραλαβής που έχουν ξεκινήσει από τη διεργασία 1. Αντίθετα, το δεύτερο μήνυμα της διεργασίας 2, θα παραληφθεί πριν από το τρίτο μήνυμα, ακόμη και αν μπορεί να παραληφθεί από την τρίτη και τέταρτη διαδικασία παραλαβής της διεργασίας 1. Επομένως εάν η διεργασία αποστολέας αποστείλει διαδοχικά μηνύματα στην ίδια διεργασία παραλήπτη τα οποία και τα δύο μπορούν να ικανοποιηθούν από την ίδια διαδικασία παραλαβής, το δεύτερο μήνυμα δεν θα μπορέσει να παραληφθεί εάν προηγουμένως δεν παραληφθεί το πρώτο μήνυμα.

Με τον ίδιο τρόπο εάν η διεργασία παραλήπτης ξεκινήσει δύο διαδικασίες παραλαβής τη μία μετά την άλλη οι οποίες μπορούν να

παραλάβουν το ίδιο εισερχόμενο μήνυμα, η δεύτερη διαδικασία παραλαβής δεν θα μπορέσει να προχωρήσει εάν η πρώτη βρίσκεται ακόμη σε εξέλιξη. Με τον τρόπο αυτό διασφαλίζεται η αιτιοκρατική φύση της διαδικασίας διαβίβασης μηνυμάτων καθώς ένα μήνυμα θα πρέπει προφανώς να αποστέλλεται σε προγενέστερη χρονική στιγμή από εκείνη της παραλαβής του.

6.4.2.1 Χαρακτηριστικά των blocking διαδικασιών (με παραδείγματα)

Τα μηνύματα έχουν μια συγκεκριμένη σειρά ανάλογα με την χρονική στιγμή την οποία εστάλησαν (σε τελική ανάλυση ανάλογα με την θέση τους στον κώδικα) .

1) Εάν , για παράδειγμα , μια διεργασία A στείλει δύο συνεχόμενα μηνύματα στην ίδια διεργασία B , τα οποία μπορούν να εξυπηρετηθούν απο την ίδια receive , τότε το δεύτερο μήνυμα δεν πρόκειται ποτέ να αντιστοιχηθεί με αυτήν την receive εφόσον το πρώτο μήνυμα δεν έχει παραδοθεί . Αντίστοιχα , εάν μια διεργασία B έχει δύο διαδοχικές receive που ταιριάζουν με την ίδια send της διεργασίας A , τότε η δεύτερη receive δεν θα αντιστοιχηθεί ποτέ με το μήνυμα όσο δεν έχει ολοκληρωθεί η πρώτη receive

```
. . .
MPI_Comm_rank (MPI_COMM_WORLD, &rank) ;
. . .
if ( rank == 0 )
{
MPI_Bsend(buffer1, count, MPI_INT, 1, SOME_TAG,
MPI_COMM_WORLD) ;
MPI_Bsend(buffer2, count, MPI_INT, 1, SOME_TAG,
}
else
{
MPI_Recv (buffer1, count, MPI_INT, 0, MPI_ANY_TAG,
MPI_COMM_WORLD, &status) ;
MPI_Recv (buffer2, count, MPI_INT, 0, MPI_ANY_TAG,
```

```
MPI_COMM_WORLD, &status);  
}  
. . .
```

Στο παραπάνω παράδειγμα , είναι αδύνατον το πρώτο μήνυμα που στέλνει η διεργασία '0' , να παραληφθεί απο άλλη εκτός της πρώτης receive που καλεί η διεργασία 1 . Και οι δύο send ταιριάζουν και με τις δύο receive , παρόλα αυτά η σειρά των μηνυμάτων δεν

μπορεί να αλλάξει και η πρώτη receive θα παραλάβει το πρώτο μήνυμα και η δεύτερη το δεύτερο μήνυμα .

2) Εάν ένα ζευγάρι send και receive έχουν αρχικοποιηθεί σε δύο διεργασίες , τότε τουλάχιστον μια απο αυτές τις διαδικασίες θα ολοκληρωθεί επιτυχώς . Η διαδικασία αποστολής θα ολοκληρωθεί εκτός και αν η receive 'ικανοποιηθεί' απο άλλη send οπότε θα ολοκληρωθεί εκείνη . Η διαδικασία παραλαβής θα ολοκληρωθεί εκτός και αν η send εξυπηρετηθεί απο άλλη αντίστοιχη receive στην ίδια διεργασία , οπότε θα ολοκληρωθεί εκείνη .

```
. . .  
MPI_Comm_rank(MPI_COMM_WORLD, &rank);  
. . .  
if ( rank == 0 )  
{  
MPI_Send(buffer1, count1, MPI_INT, 1, tag1,  
MPI_COMM_WORLD);  
MPI_Send(buffer2, count2, MPI_INT, 1, tag2,  
MPI_COMM_WORLD);  
}  
else  
{  
MPI_Recv(buffer2, count2, MPI_INT, 0, tag2,  
MPI_COMM_WORLD, &status1);  
MPI_Recv(buffer1, count1, MPI_INT, 0, tag1,  
MPI_COMM_WORLD, &status2);  
}
```

. . .

Στο παραπάνω παράδειγμα , υπάρχουν δύο χιαστί ζευγάρια ταιριαστών `send` και `receive` . Το παραπάνω πρόγραμμα θα καταλήξει σε αδιέξοδο , εκτός και αν η πρώτη `send` της διεργασίας '0' , εκτελεστεί σε `buffered mode` οπότε το μήνυμα μεταφέρεται σε ένα προσωρινό χώρο αποθήκευσης , δίνοντας έτσι την δυνατότητα στην διαδικασία να ολοκληρωθεί επιτυχώς . Στην συνέχεια , εκτελείται η δεύτερη `send` η οποία αντιστοιχεί με την πρώτη `receive` που έχει είδη καλέσει η διεργασία '1' . Επομένως η πρώτη μετάδοση δεδομένων ολοκληρώνεται επιτυχώς . Τέλος , η διεργασία '1' καλεί την δεύτερη `receive` , η οποία αντιστοιχεί με την πρώτη `send` -που εκτελέστηκε σε `buffered mode` , επιτρέποντας έτσι την επιτυχή ολοκλήρωση και της δεύτερης μετάδοσης δεδομένων . Είναι σημαντικό να παρατηρήσουμε ότι στην παραπάνω περίπτωση , τα μηνύματα λαμβάνονται από την διεργασία '1' με την αντίστροφη σειρά από ότι εστάλησαν από την διεργασία '0' .

3) Ιδιαίτερη προσοχή χρειάζεται ώστε να μην υπάρξει πρόβλημα με τους διαθέσιμους πόρους του συστήματος . Τα μηνύματα που αποστέλονται σε `buffered mode` , όπως είναι φυσικό καταναλώνουν χώρο μνήμης ο οποίος έχει και κάποια όρια.Εάν συνεχίσουν να 'παράγονται' `buffered` διαδικασίες αποστολής, τότε είναι πολύ πιθανόν σε κάποια στιγμή να εξαντληθεί η μνήμη , να παραχθεί ένα `overflow` μήνυμα λάθους και το πρόγραμμα να τερματιστεί 'ανώμαλα'. Είναι σημαντικό να παρατηρήσουμε , ότι ενώ στις `buffered` αποστολές το σύστημα ενδεχομένως να αντιμετωπίσει πρόβλημα διαθέσιμης μνήμης , στις `standard mode` αποστολές , σε περίπτωση εξάντλησης των πόρων του συστήματος το πρόγραμμα δεν θα κλείσει ανώμαλα : απλώς η διαδικασία αποστολής θα μπλοκάρει , περιμένοντας είτε να ελευθερωθεί χώρος στην μνήμη για `buffering` του μηνύματος , είτε να κληθεί η αντίστοιχη `receive` οπότε και θα αρχίσει να μεταδίδει τα δεδομένα κατευθείαν στην διεργασία – παραλήπτη .

. . .

```
MPI_Comm_rank(MPI_COMM_WORLD, &rank) ;
```

. . .

```
if ( rank == 0 )
```



```
{
MPI_Send(send, count1, MPI_INT, 1, SOME_TAG,
MPI_COMM_WORLD);
MPI_Recv(recv, count2, MPI_INT, 1, SOME_TAG,
MPI_COMM_WORLD, &status);
}
else
{
MPI_Recv(recv, count1, MPI_INT, 0, SOME_TAG,
MPI_COMM_WORLD, &status);
MPI_Send(send, count2, MPI_INT, 0, SOME_TAG,
MPI_COMM_WORLD);
}
```

Στο παραπάνω παράδειγμα , οι δύο διεργασίες θα στείλουν απο ένα μήνυμα η κάθε μια στην άλλη (θα 'ανταλλάξουν' δεδομένα) . Η όλη διαδικασία θα ολοκληρωθεί με επιτυχία , ακόμα και αν δεν υπάρχει ελεύθερος χώρος στην μνήμη για buffering .

```
4) . . .
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
. . .
if ( rank == 0 )
{
MPI_Recv(recv, count2, MPI_INT, 1, SOME_TAG,
MPI_COMM_WORLDS, &status);
MPI_Send(send, count1, MPI_INT, 1, SOME_TAG,
MPI_COMM_WORLD);
}
else
{
MPI_Recv(recv, count1, MPI_INT, 0, SOME_TAG,
MPI_COMM_WORLD, &status);
MPI_Send(send, count2, MPI_INT, 0, SOME_TAG,
MPI_COMM_WORLD);
}
```

}

Ο συγκεκριμένος κώδικας , θα καταλήγει σε αδιέξοδο μόνιμα , ανεξάρτητα του mode επικοινωνίας που θα χρησιμοποιήσουμε . Η διαδικασία αποστολής κάθε διεργασίας δεν μπορεί να αρχίσει , παρά μόνο όταν ολοκληρωθούν οι διαδικασίες receive που προηγούνται . Οι διαδικασίες receive , με την σειρά τους , δεν μπορούν να ολοκληρωθούν εάν δεν σταλούν τα αντίστοιχα μηνύματα . Έτσι προκύπτει το αδιέξοδο .

```
5) . . .
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
. . .
if ( rank == 0 )
{
MPI_Send(send, count1, MPI_INT, 1, SOME_TAG,
MPI_COMM_WORLD);
MPI_Recv(recv, count2, MPI_INT, 1, SOME_TAG,
MPI_COMM_WORLD, &status);
}
else
{
MPI_Send(send, count2, MPI_INT, 0, SOME_TAG,
MPI_COMM_WORLD);
MPI_Recv(recv, count1, MPI_INT, 0, SOME_TAG,
MPI_COMM_WORLD, &status);
}
```

Σε αυτήν την περίπτωση , ο κώδικας θα εκτελεστεί σωστά , μόνο εάν τουλάχιστον μια απο τις δύο διαδικασίες αποστολής (send) , εκτελεστούν σε buffered mode έτσι ώστε να ολοκληρωθεί και να αρχίσει η εκτέλεση της receive που ακολουθεί . Σε περίπτωση που το σύστημα δεν μπορεί να αντιγράψει σε ένα buffer κανένα απο τα δύο μηνύματα (δηλαδή τουλάχιστον min(count1, count2) bytes δεδομένων) πιθανώς λόγω έλλειψης διαθέσιμου χώρου , τότε το πρόγραμμα θα οδηγηθεί σε αδιέξοδο και θα 'κολλήσει' .

6.4.3 Μη Παρεμποδιστικές Επικοινωνίες (non blocking)

Το πρότυπο MPI παρέχει μη παρεμποδιστικές συναρτήσεις αποστολής και λήψης δεδομένων οι οποίες δεν αναστέλλουν τη λειτουργία της διεργασίας που τις καλεί, αλλά επιτρέπουν τη συνέχιση της εκτέλεσης τους ταυτόχρονα με την αποστολή ή την παραλαβή του μηνύματος. Οι συναρτήσεις αυτές φέρουν τα ονόματα MPI_Isend και MPI_Irecv και είναι οι μη παρεμποδιστικές εκδόσεις των παρεμποδιστικών συναρτήσεων MPI_Send και MPI_Recv. Σε μια πιο λεπτομερή περιγραφή, το βασικό χαρακτηριστικό των μη παρεμποδιστικών συναρτήσεων, είναι η χρήση κάποιας από αυτές (post-send ή post-receive operation) για την εκκίνηση της διαδικασίας αποστολής ή παραλαβής κάποιου μηνύματος. Σε αντίθεση όμως με τα παρεμποδιστικά ισοδύναμά τους, οι συναρτήσεις αυτές δεν αναμένουν την ολοκλήρωση της διαδικασίας μετάδοσης του μηνύματος, αλλά τερματίζουν άμεσα τη λειτουργία τους. Αυτός ο τερματισμός επιτρέπει τη λειτουργία της διεργασίας που κάλεσε τη μη παρεμποδιστική συνάρτηση να συνεχιστεί κανονικά. Όσον αφορά τον τερματισμό της διαδικασίας αποστολής ή λήψης του μηνύματος αυτός γίνεται με τη βοήθεια κάποιων άλλων συναρτήσεων (complete-send ή complete-receive operations) οι οποίες επιπλέον ελέγχουν εάν το μήνυμα έχει αποσταλεί ή παραληφθεί σωστά.

Η λογική της non-blocking επικοινωνίας ακολουθεί τις παρακάτω φάσεις :

A. Έναρξη της διαδικασίας αποστολής / παραλαβής και άμεση επιστροφή του ελέγχου στον χρήστη . Η διαδικασία δεν έχει ολοκληρωθεί και η άμεση χρήση του buffer αποστολής / παραλαβής είναι επικίνδυνη .

B. Πιθανόν η διεργασία να εκτελέσει κάποιον κώδικα , μετά την έναρξη της διαδικασίας αποστολής / παραλαβής και πριν την ολοκλήρωση της .

Γ. Η ολοκλήρωση της διαδικασίας αποστολής / παραλαβής ελέγχεται με την κλήση κάποιων συγκεκριμένων ρουτινών. Απο την πλευρά του αποστολέα , αυτό δεν σημαίνει ότι το μήνυμα μεταδόθηκε στην διεργασία – παραλήπτη , αλλά ότι έχει αντιγραφεί σε κάποιο buffer του συστήματος και είναι έτοιμο να μεταδοθεί όταν εμφανιστεί μια αντίστοιχη receive . Απο αυτό το σημείο και μετά , ο buffer αποστολής / παραλαβής μπορεί άνετα να επαναχρησιμοποιηθεί .

6.4.3.1 Χαρακτηριστικά των Μη Παρεμποδιστικών Επικοινωνιών (με παραδείγματα)

Το βασικό χαρακτηριστικό των μη παρεμποδιστικών διαδικασιών, είναι η χρήση ειδικών αντικειμένων (request objects) για την ταυτοποίηση των λειτουργιών που πραγματοποιούνται σε κάθε περίπτωση και τη συσχέτιση της διαδικασίας αποστολής του μηνύματος με τη διαδικασία ολοκλήρωσής της. Αυτά τα ειδικού τύπου αδιαφανή αντικείμενα δημιουργούνται από το MPI κατά την εκκίνηση της λειτουργίας του, διατηρούνται στη μνήμη του συστήματος ,και είναι προσπελάσιμα μόνο δια της χρήσεως ειδικών συναρτήσεων οι οποίες επιστρέφουν ένα χειριστή (handle) προς αυτά. Ο ρόλος αυτών των ενδιάμεσων αντικειμένων είναι πάρα πολύ σημαντικός καθώς χρησιμοποιούνται από το σύστημα για την ταυτοποίηση των ιδιοτήτων της διαδικασίας αποστολής ή παραλαβής και την αποθήκευση πληροφοριών που συσχετίζονται με την κατάσταση (status) της διαδικασίας επικοινωνίας. Η προσπέλαση αυτών των αντικειμένων μέσα από μια μη παρεμποδιστική συνάρτηση είναι δυνατή δια της χρήσεως ενός ειδικού τύπου δεδομένων του MPI που φέρει το όνομα MPI_Request.

1) Η ιδιότητα των μηνυμάτων να διατηρούν την σειρά με την οποία εστάλησαν , στην οποία αναφερθήκαμε στις ιδιότητες των blocking διαδικασιών , ισχύει και στην περίπτωση των non-blocking διαδικασιών επικοινωνίας .

```
. . .
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
. . .
if ( rank == 0 )
{
MPI_Isend(send1, 1, MPI_INT, 1, SOME_TAG,
MPI_COMM_WORLD, &request1);
MPI_Isend(send2, 1, MPI_INT, 1, SOME_TAG,
MPI_COMM_WORLD, &request2);
}
else
{
```

```
MPI_Irecv(recv1, 1, MPI_INT, 0, SOME_TAG,  
MPI_COMM_WORLD, &request1);  
MPI_Irecv(recv2, 1, MPI_INT, 0, SOME_TAG,  
MPI_COMM_WORLD, &request2);  
}
```

Εδώ , παρατηρούμε οτι και οι δύο send της διεργασίας '0' , 'ταιριάζουν' με την πρώτη receive τη διεργασίας '1' . Παρόλα αυτά , αυτή ή receive θα δέχεται πάντα το μήνυμα της πρώτης send , ακόμα και αν οι δύο non blocking send είχαν επιστρέψει και οι δύο πριν απο την κλήση της πρώτης receive .

6.5 ΣΥΛΛΟΓΙΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ

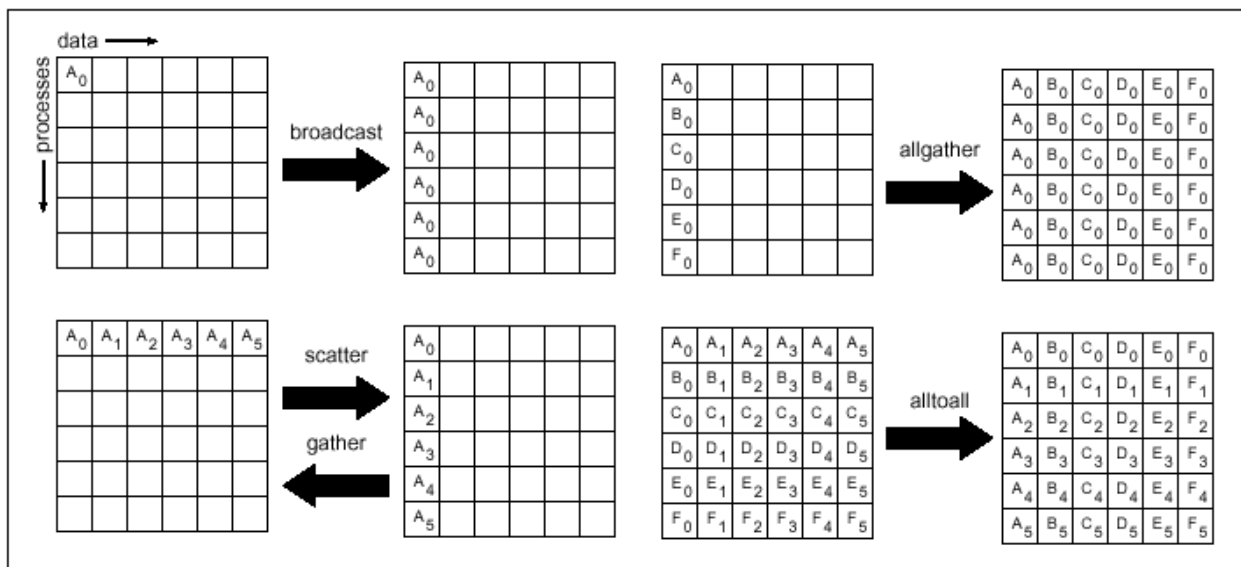
Το βασικό χαρακτηριστικό αυτού του ιδιαίτερου τύπου επικοινωνίας, είναι πως η ανταλλαγή των δεδομένων πραγματοποιείται ανάμεσα σε δύο μόνο διεργασίες εκ των οποίων η μία νοείται ως η διεργασία αποστολέας ενώ η δεύτερη νοείται ως η διεργασία παραλήπτης. Εκτός όμως από τις παραπάνω διαδικασίες επικοινωνίας από σημείο σε σημείο, υπάρχουν και άλλες διαδικασίες οι οποίες χαρακτηρίζονται από τη μετάδοση των δεδομένων της εφαρμογής σε όλες τις διεργασίες που περιλαμβάνονται στον τρέχοντα intracommunicator. Οι επικοινωνίες που πραγματοποιούνται με τον τρόπο αυτό, είναι γνωστές ως συλλογικές επικοινωνίες (collective communications).

Μιλώντας γενικά, η ανταλλαγή δεδομένων ανάμεσα σε όλες τις διεργασίες που υπάρχουν στο σύστημα, μπορεί να ανήκει σε μία από τις ακόλουθες μορφές επικοινωνίας:

- Εκπομπή δεδομένων (broadcasting) όπου μια διεργασία αποστέλλει το κατάλληλο σε κάθε περίπτωση μήνυμα σε όλες τις διαθέσιμες διεργασίες.
- Συλλογή δεδομένων (gathering) όπου μια διεργασία συλλέγει δεδομένα που έχουν αποσταλεί από όλες τις διαθέσιμες διεργασίες.
- Διασπορά δεδομένων (scattering) όπου τα δεδομένα ενός μηνύματος διαμοιράζονται με τον κατάλληλο τρόπο σε όλες τις υπόλοιπες διεργασίες.

Οι παραπάνω συναρτήσεις επιτελούν τις θεμελιώδεις μορφές επικοινωνίας που δύνανται να χρησιμοποιηθούν σε ένα σύστημα πολλών παράλληλων διεργασιών.

Υπάρχουν ωστόσο και αρκετές παραλλαγές αυτών των συναρτήσεων όπως είναι για παράδειγμα η συνάρτηση `MPI_Alltoall` που συνδυάζει τις συναρτήσεις της συλλογής και της διασποράς σε μια απλή κλήση, και η συνάρτηση `MPI_Allgather` που επεκτείνει τη λειτουργία της συλλογής, επιτρέποντας την παραλαβή των δεδομένων από όλες τις διαθέσιμες διεργασίες. Ο τρόπος λειτουργίας των συλλογικών επικοινωνιών που παρουσιάστηκαν στην προηγούμενη παράγραφο, απεικονίζεται διαγραμματικά στο επόμενο σχήμα.



Εικόνα 59: Λειτουργία των συλλογικών επικοινωνιών

Στο σχήμα αυτό, η πρώτη γραμμή αναπαριστά τις τιμές των δεδομένων που είναι διαθέσιμες, ενώ η πρώτη στήλη αντιστοιχεί στις διεργασίες που τρέχουν στο σύστημα. Σύμφωνα με την παραπάνω εικόνα, υπάρχουν έξι μεταβλητές που περιέχουν δεδομένα και φέρουν τα ονόματα A,B,C,D,E και F, και ισάριθμο πλήθος διεργασιών με τιμές τάξης 0,1,2,3,4 και 5.

Προκειμένου να αναφερθούμε στη τιμή που αποστέλλεται σε κάποια διεργασία χρησιμοποιούμε το γράμμα που αντιστοιχεί στην εν λόγω τιμή

συνοδευόμενο από την τάξη της θεωρούμενης διεργασίας, η οποία εμφανίζεται υπό μορφή δείκτη. Έτσι το σύμβολο A_0 αναφέρεται στην τιμή της ποσότητας A η οποία ανήκει στη διεργασία με τάξη 0, το σύμβολο B_3 αναφέρεται στην τιμή της ποσότητας B η οποία ανήκει στη διεργασία με τάξη 3 κ.ο.κ. Χρησιμοποιώντας αυτόν τον συμβολισμό, μπορούμε να προχωρήσουμε σε μια ποιοτική περιγραφή των συλλογικών επικοινωνιών με τον ακόλουθο τρόπο: Θεωρώντας πως η πρώτη διεργασία διαθέτει την ποσότητα A_0 , η λειτουργία της εκπομπής αποστέλλει την τιμή της εν λόγω ποσότητας σε όλες τις διαθέσιμες διεργασίες, οι οποίες μετά την ολοκλήρωση της διαδικασίας περιέχουν όλες την τιμή A_0 . Αντίθετα, στην λειτουργία της διασποράς, η κάθε μια από τις έξι διαφορετικές τιμές που είναι διαθέσιμες αποστέλλεται και σε μια διαφορετική διεργασία και με τέτοιο τρόπο ώστε η κάθε διεργασία να παραλάβει μόνο ένα στοιχείο. Θεωρώντας τις έξι προαναφερόμενες τιμές ως τις συνιστώσες ενός διάνυσματος, είναι προφανής ο χαρακτηρισμός της παραπάνω διαδικασίας ως διασπορά: η διεργασία αποστολέας αποσυνθέτει το εν λόγω διάνυσμα στις συνιστώσες από τις οποίες αποτελείται, και στη συνέχεια τις διασκορπίζει, αποστέλλοντας κάθε μια από αυτές και σε μια διαφορετική διεργασία.

Η τρίτη θεμελιώδης μορφή συλλογικής επικοινωνίας είναι η συλλογή – η διαδικασία αυτή χαρακτηρίζεται από την παραλαβή των δεδομένων όλων των διεργασιών του συστήματος, από μια απλή διεργασία που ονομάζεται διεργασία ρίζα (root process). Η διαδικασία της συλλογής επιτελεί την αντίστροφη λειτουργία από εκείνη της διασποράς καθώς δεν διασκορπίζει αλλά συλλέγει δεδομένα και μπορεί να επεκταθεί με τέτοιο τρόπο ώστε τα διακινούμενα δεδομένα να παραληφθούν από όλες τις διαθέσιμες διεργασίες. Μια άλλη λειτουργία που αξίζει να αναφερθεί περιλαμβάνει μια συνδυασμένη χρήση των λειτουργιών διασποράς και παραλαβής και επιτρέπει την προσπέλαση του συνόλου των δεδομένων της εφαρμογής από όλες τις διαθέσιμες διεργασίες του συστήματος – οι δύο τελευταίες περιπτώσεις παρουσιάζονται και αυτές στο παραπάνω σχήμα.

Εκτός από τις παραπάνω συναρτήσεις αποστολής και λήψης δεδομένων σε συλλογικό επίπεδο, το πρότυπο MPI παρέχει και ορισμένες άλλες χρήσιμες συναρτήσεις, οι πιο ενδιαφέρουσες από τις οποίες είναι οι επανομαζόμενες συναρτήσεις αναγωγής (reduction functions). Το βασικό

χαρακτηριστικό αυτών των συναρτήσεων είναι ο υπολογισμός μιας απλής τιμής από ένα σύνολο τιμών που παραλαμβάνονται από τις υπόλοιπες διεργασίες. Για παράδειγμα, μπορούμε να συλλέξουμε δέκα διαφορετικές τιμές από δέκα διαφορετικές διεργασίες, και στη συνέχεια να υπολογίσουμε τη μεγαλύτερη ή τη μικρότερη από αυτές.

Τέλος υπάρχει η δυνατότητα να συγχρονίσουμε την εκτέλεση όλων των ενεργών διεργασιών, καλώντας την κατάλληλη για αυτό το σκοπό συνάρτηση του MPI.

Η πιο σημαντική διαφορά που υφίσταται ανάμεσα στις συλλογικές επικοινωνίες και στις επικοινωνίες από σημείο σε σημείο, είναι πως οι πρώτες πραγματοποιούνται δια της χρήσεως μόνο παρεμποδιστικών διαδικασιών – η χρήση μη παρεμποδιστικών συλλογικών διαδικασιών δεν υποστηρίζεται (ακόμη) από το πρότυπο του MPI. Επιπλέον, η ποσότητα δεδομένων που αποστέλλεται προς κάποια διεργασία πρέπει να είναι ακριβώς η ίδια με εκείνη που θα δηλωθεί πως θα παραληφθεί από τη διεργασία παραλήπτη. Η κατάσταση αυτή δεν παρατηρείται στις επικοινωνίες από σημείο σε σημείο, όπου η διεργασία παραλήπτης είχε τη δυνατότητα να παραλάβει μικρότερη ποσότητα πληροφορίας από εκείνη που είχε σταλεί.

Τέλος, στις διαδικασίες που χρησιμοποιούνται στις συλλογικές επικοινωνίες, το χαρακτηριστικό της ετικέτας (tag) του μηνύματος δεν υφίσταται, και η σειρά εξυπηρέτησης των διαδικασιών που εκκρεμούν, καθορίζεται μονοσήμαντα από τη σειρά της εκκίνησής τους.

Ολοκληρώνουμε τη συνοπτική θεωρητική εισαγωγή μας στις συλλογικές διαδικασίες, αναφέροντας πως σε αντίθεση με την επικοινωνία από σημείο σε σημείο η οποία εμφανίζεται σε τέσσερις διαφορετικές μορφές (standard, buffered, synchronous και ready), οι συλλογικές επικοινωνίες χαρακτηρίζονται από την ύπαρξη ενός και μοναδικού τύπου που αντιστοιχεί στον πρότυπο (standard) τύπο επικοινωνίας από σημείο σε σημείο. Πιο συγκεκριμένα, μια συλλογική συνάρτηση μπορεί να επιστρέψει όταν η δική της συμμετοχή στη γενικότερη διαδικασία επικοινωνίας έχει ολοκληρωθεί.

Σε πλήρη αναλογία με τις προηγούμενες περιπτώσεις, η ολοκλήρωση μιας διαδικασίας επιτρέπει τη διεργασία αποστολέα να προσπελάσει και να τροποποιήσει τα περιεχόμενα της περιοχής μνήμης που χρησιμοποιεί για

την ανταλλαγή δεδομένων (communication buffer). Αντίθετα δεν παρέχει καμιά πληροφορία σχετικά με το εάν οι άλλες διεργασίες έχουν ξεκινήσει τη λειτουργία τους ή έχουν ολοκληρωθεί.

6.5.1 Οι θεμελιώδεις συναρτήσεις της συλλογικής επικοινωνίας

Μετά την περιγραφή των βασικών χαρακτηριστικών που περιγράφουν τη διαδικασία της συλλογικής επικοινωνίας, ας περάσουμε τώρα στην παρουσίαση των συναρτήσεων που χρησιμοποιούμε σε αυτές τις περιπτώσεις.

Η πρώτη από τις συναρτήσεις που θα περιγράψουμε, είναι η MPI_Bcast η οποία αποστέλλει ένα μήνυμα από κάποια διεργασία – που ονομάζεται διεργασία ρίζα (root process) – σε όλες τις υπόλοιπες διεργασίες της τρέχουσας ομάδας.

Η χρήση της συνάρτησης MPI_Bcast χαρακτηρίζεται από μία σύνταξη της μορφής

```
int MPI_Bcast (void * buffer, int count,  
MPI_Datatype  dataType, int root, MPI_Comm comm)
```

Στην παραπάνω σύνταξη, τα τρία πρώτα ορίσματα ταυτοποιούν το μήνυμα που αποστέλλεται: το πρώτο όρισμα αναφέρεται στην περιοχή μνήμης που περιέχει το εν λόγω μήνυμα, ενώ το δεύτερο και το τρίτο όρισμα επιτρέπουν τον καθορισμό του πλήθους και του τύπου δεδομένων των στοιχείων που περιλαμβάνονται στο μήνυμα.

Στη συνέχεια η συνάρτηση περιέχει την τάξη της διεργασίας που θα αναλάβει την αποστολή του μηνύματος (το όρισμα root) ενώ το τελευταίο όρισμα αναφέρεται κατά τα γνωστά στον communicator που χρησιμοποιείται για τη διακίνηση των μηνυμάτων της εφαρμογής. Ας σημειωθεί πως η επικοινωνία των διεργασιών δια της χρήσεως συλλογικών διαδικασιών είναι δυνατή μόνο όταν αυτές ανήκουν στην ίδια ομάδα. Για το λόγο αυτό το τελευταίο όρισμα της συνάρτησης MPI_Bcast θα πρέπει να αναφέρεται μόνο σε intracommunicator και όχι σε intercommunicator ο οποίος επιτρέπει την επικοινωνία διεργασιών που ανήκουν σε διαφορετικές ομάδες.

Είναι ενδιαφέρον να αναφερθεί στο σημείο αυτό, πως η συνάρτηση `MPI_Bcast` λειτουργεί με τέτοιο τρόπο, ώστε κατά τον τερματισμό της λειτουργίας της να έχει λάβει χώρα αντιγραφή των δεδομένων του μηνύματος στις περιοχές αποθήκευσης των υπόλοιπων διεργασιών του συστήματος, έτσι ώστε να μην είναι αναγκαία η κλήση κάποια συνάρτησης παραλαβής, όπως είναι για παράδειγμα η `MPI_Recv`.

Τυπικό παράδειγμα χρήσης της συνάρτησης `MPI_Bcast` είναι η κλήση της με τη μορφή

```
MPI_Bcast (buffer, 500, MPI_DOUBLE, 0, MPI_COMM_WORLD)
```

Η οποία επιτρέπει την αποστολή ενός πίνακα 500 πραγματικών αριθμών από τη διεργασία με τάξη μηδέν προς όλες τις διαθέσιμες διεργασίες του συστήματος.

Η δεύτερη σημαντική συνάρτηση που χρησιμοποιείται στις συλλογικές επικοινωνίες είναι η `MPI_Gather` η οποία επιτρέπει σε μια διεργασία να συλλέξει δεδομένα που έχουν αποσταλεί προς αυτή από όλες τις υπόλοιπες διεργασίες της τρέχουσας ομάδας. Η `MPI_Gather` καλείται τόσο από τις διεργασίες που αποστέλλουν τα δεδομένα όσο και από τη διεργασία ρίζα η οποία θα πραγματοποιήσει την παραλαβή των δεδομένων. Σε αυτόν τον τρόπο επικοινωνίας, η κάθε διεργασία (συμπεριλαμβανομένης και της διεργασίας ρίζα) αποστέλλει τα δεδομένα της στη διεργασία ρίζα η οποία τα παραλαμβάνει και τα αποθηκεύει, ανάλογα με την τιμή της τάξης της διεργασίας που τα απέστειλε. Αξίζει να σημειωθεί, πως το αποτέλεσμα της συνάρτησης `MPI_Gather` είναι ακριβώς το ίδιο με εκείνο που θα προέκυπτε εάν κάθε μια από τις διεργασίες της ομάδας (συμπεριλαμβανομένης και της διεργασίας ρίζα) χρησιμοποιούσε την `MPI_Send` για να αποστείλει τα δεδομένα της στη διεργασία ρίζα, και στη συνέχεια η διεργασία ρίζα χρησιμοποιούσε τη συνάρτηση `MPI_Recv` για να παραλάβει αυτά τα δεδομένα.

Η χρήση της συνάρτησης `MPI_Gather` ακολουθεί τη σύνταξη

```
int MPI_Gather(void * sendBuf, int sendCount,  
MPI_Datatype sendType, void * recvBuf,
```

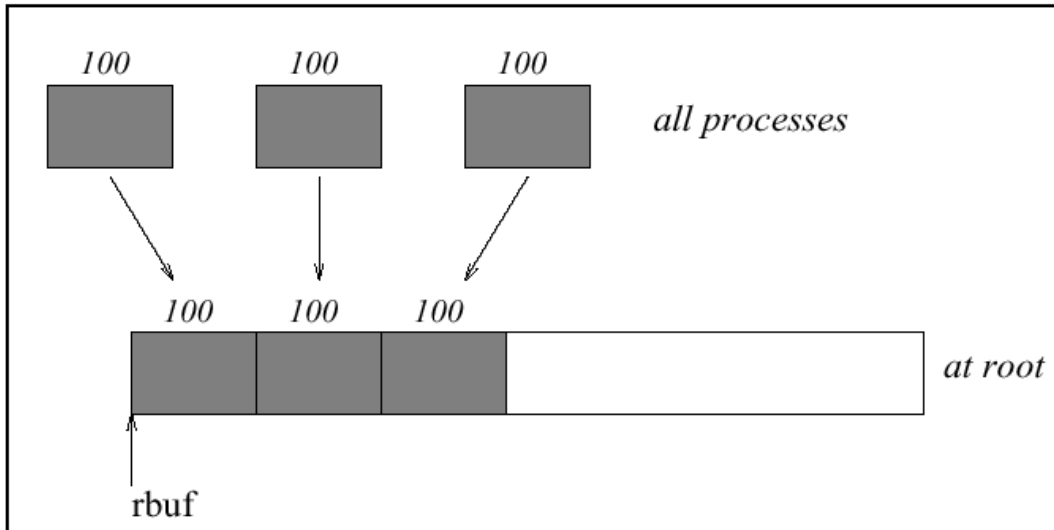
```
int recvCount, MPI_Datatype recvType, int root,  
MPI_Comm comm)
```

Στην παραπάνω σύνταξη, τα ορίσματα `sendBuf`, `sendCount` και `sendType` αναφέρονται κατά τα γνωστά στην περιοχή μνήμης της διαδικασίας αποστολέα καθώς επίσης και στο πλήθος και στον τύπο δεδομένων των στοιχείων προς αποστολή. Τα επόμενα τρία ορίσματα συσχετίζονται με τη διαδικασία παραλαβής και περιγράφουν την περιοχή μνήμης στην οποία θα αποθηκευτούν αυτά τα δεδομένα από τη διεργασία ρίζα, καθώς επίσης το πλήθος των στοιχείων που θα παραληφθούν από κάθε διεργασία και τον τύπο δεδομένων τους. Ας αναφερθεί στο σημείο αυτό πως αυτά τα τρία ορίσματα λαμβάνονται υπ' όψιν μόνο από τη διεργασία ρίζα – που θα παραλάβει τα διακινούμενα δεδομένα – και θα αγνοηθούν από τις υπόλοιπες διεργασίες του συστήματος. Η τιμή της τάξης της διεργασίας ρίζα καθορίζεται από το όρισμα `root`, ενώ το τελευταίο όρισμα αναφέρεται ως συνήθως στον `communicator` δια μέσου του οποίου λαμβάνει χώρα η διακίνηση των δεδομένων ανάμεσα στις διεργασίες του συστήματος.

Τυπικό παράδειγμα χρήσης της συνάρτησης `MPI_Gather` ακολουθεί στη συνέχεια:

```
int rank, size, sendBuf[100], * recvBuf;  
MPI_Comm_rank (MPI_COMM_WORLD, &rank);  
if (rank==0) {  
MPI_Comm_size (MPI_COMM_WORLD, &size);  
recvBuf = (int *) malloc (size*100*sizeof(int)); }  
MPI_Gather(sendBuf,100,MPI_INT,recvBuf,100,  
MPI_INT, 0, MPI_COMM_WORLD);
```

Στο παραπάνω παράδειγμα η διεργασία ρίζα – που χαρακτηρίζεται από τιμή τάξης ίση με το μηδέν – διαβάζει ένα πίνακα 100 ακεραίων από κάθε μια από τις διεργασίες που τρέχουν στο σύστημα. Η κατάσταση αυτή παρουσιάζεται στο επόμενο σχήμα.



Εικόνα 60: παράδειγμα MPI_Gather

Η χρήση της συνάρτησης MPI_Scatter χαρακτηρίζεται από μία σύνταξη της μορφής

```
int MPI_Scatter (void * sendBuf, int sendCount,
MPI_Datatype sendType, void * recvBuf,
int recvCount, MPI_Datatype recvType, int root,
MPI_Comm comm)
```

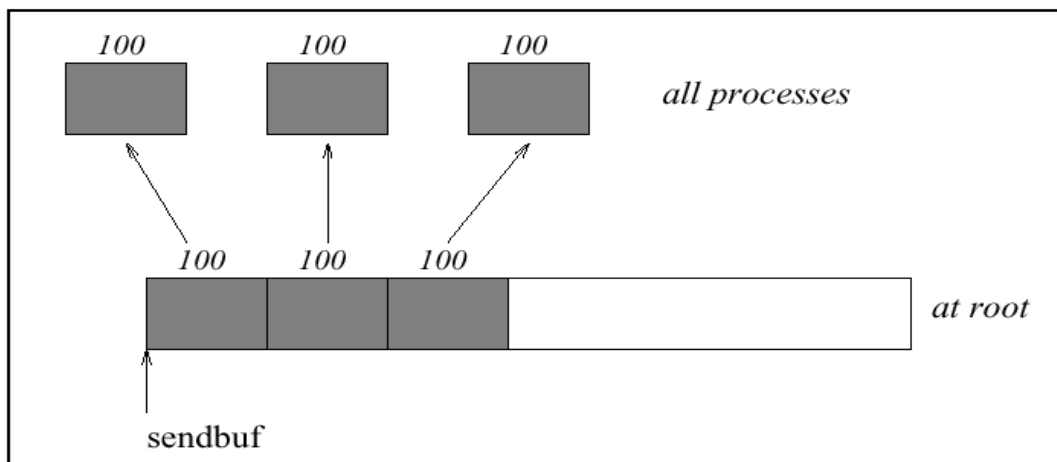
Τα ορίσματα στην παραπάνω σύνταξη της MPI_Scatter έχουν την ίδια σημασία με εκείνη που περιγράψαμε για τα αντίστοιχα ορίσματα της συνάρτησης MPI_Gather. Έτσι τα τρία πρώτα ορίσματα αναφέρονται στην περιοχή μνήμης στην οποία αποθηκεύεται το μήνυμα προς διασπορά, καθώς επίσης στο πλήθος και στον τύπο αυτών των δεδομένων. Το όρισμα recvBuff αναφέρεται στην περιοχή μνήμης στην οποία θα αποθηκευθούν τα δεδομένα από τη διεργασία παραλήπτη, η παράμετρος recvCount είναι το πλήθος των στοιχείων που παραλαμβάνονται, ενώ το όρισμα recvType αναφέρεται στον τύπο δεδομένων αυτών των στοιχείων.

Τέλος, το όρισμα root ταυτοποιεί τη διεργασία ρίζα που πρόκειται να αποστείλει το μήνυμα, ενώ η παράμετρος comm ορίζει τον communicator δια της χρήσεως του οποίου λαμβάνει χώρα η διακίνηση της πληροφορίας στην τρέχουσα εφαρμογή.

Τα παραπάνω ορίσματα, στο σύνολό τους χρησιμοποιούνται μόνο από τη διεργασία ρίζα, ενώ οι υπόλοιπες διεργασίες λαμβάνουν υπ' όψιν μόνο τα ορίσματα `recvBuf`, `recvCount` και `recvType` και αγνοούν όλα τα υπόλοιπα. Το αποτέλεσμα της κλήσεως της `MPI_Scatter` είναι το ίδιο με εκείνο που θα προέκυπτε εάν η διεργασία ρίζα καλούσε N φορές τη συνάρτηση `MPI_Send` για να στείλει το υπ' αριθμόν k τμήμα του μηνύματος στη διεργασία με τιμή τάξης ίση με k ($k=0,1,2,\dots,N-1$) ενώ η κάθε μια από τις υπόλοιπες διεργασίες καλούσε μια φορά τη συνάρτηση `MPI_Recv` για να παραλάβει το τμήμα του μηνύματος που της έχει αποσταλεί. Τυπικό παράδειγμα χρήσης της συνάρτησης `MPI_Scatter` ακολουθεί στη συνέχεια. Στο παράδειγμα αυτό η διεργασία ρίζα αποστέλλει ένα πίνακα 100 πραγματικών αριθμών σε κάθε μια από τις διεργασίες της εφαρμογής.

```
int size, * sendBuf;
int root, recvBuf [100];
-----
MPI_Comm_size (MPI_COMM_WORLD, &size);
sendBuf = (int *) malloc (size*100*sizeof(int));
-----
MPI_Scatter (sendBuf, 100, MPI_INT, recvBuf, 100, MPI_INT, 0,
MPI_COMM_WORLD);
```

Το αποτέλεσμα της εκτέλεσης του παραπάνω κώδικα παρουσιάζεται στο επόμενο σχήμα.



Εικόνα 61: παράδειγμα `MPI_Scatter`

6.6 MPI ΣΥΝΑΡΤΗΣΕΙΣ ΚΑΙ ΣΤΑΘΕΡΕΣ

✓ Συναρτήσεις Διαχείρισης του Περιβάλλοντος :

MPI Abort
MPI Errhandler create
MPI Errhandler free
MPI Errhandler get
MPI Errhandler set
MPI Error class
MPI Error string
MPI Finalize
MPI Get processor name
MPI Init
MPI Initialized
MPI Wtick
MPI Wtime

✓ Συναρτήσεις Επικοινωνίας Point-to-Point :

MPI Bsend
MPI Bsend init
MPI Buffer attach
MPI Buffer detach
MPI Cancel
MPI Get count
MPI Get elements
MPI Ibsend
MPI Iprobe
MPI Irecv
MPI Isend
MPI Issend
MPI Irsend
MPI Probe
MPI Recv

MPI Recv_init

MPI Request_free

MPI Rsend

MPI Rsend_init

MPI Send

MPI Send_init

MPI Sendrecv

MPI Sendrecv_replace

MPI Ssend

MPI Ssend_init

MPI Start

MPI Startall

MPI Test

MPI Test_cancelled

MPI Testall

MPI Testany

MPI Testsome

MPI Wait

MPI Waitall

MPI Waitany

MPI Waitsome

✓ **Συναρτήσεις Συλλογικής Επικοινωνίας :**

MPI Allgather

MPI Allgatherv

MPI Allreduce

MPI Alltoall

MPI Alltoallv

MPI Barrier

MPI Bcast

MPI Gather

MPI Gatherv

MPI Op_create

MPI Op_free

MPI Reduce

MPI Reduce scatter

MPI Scan

MPI Scatter

MPI Scatterv

✓ **Συναρτήσεις Διαχείρισης Ομάδων :**

MPI Group compare

MPI Group difference

MPI Group excl

MPI Group free

MPI Group incl

MPI Group intersection

MPI Group range excl

MPI Group range incl

MPI Group rank

MPI Group size

MPI Group translate ranks

MPI Group union

✓ **Συναρτήσεις Διαχείρισης Communicators :**

MPI Comm compare

MPI Comm create

MPI Comm dup

MPI Comm free

MPI Comm group

MPI Comm rank

MPI Comm remote group

MPI Comm remote size

MPI Comm size

MPI Comm split

MPI Comm test inter

MPI Intercomm create

MPI Intercomm_merge

✓ **Συναρτήσεις για τους Παραγόμενους Τύπους Δεδομένων :**

MPI Type_commit

MPI Type_contiguous

MPI Type_extent

MPI Type_free

MPI Type_hvector

MPI Type_lb

MPI Type_size

MPI Type_struct

MPI Type_ub

MPI Type_vector

✓ **Συναρτήσεις Διαχείρισης Εικονικών Τοπολογιών :**

MPI Cart_coords

MPI Cart_create

MPI Cart_get

MPI Cart_map

MPI Cart_rank

MPI Cart_shift

MPI Cart_sub

MPI Cartdim_get

MPI Dims_create

MPI Graph_create

MPI Graph_get

MPI Graph_map

MPI Graph_neighbors

MPI Graph_neighbors_count

MPI Graphdims_get

MPI Topo_test

✓ **Διάφορες Συναρτήσεις :**

MPI_Address

MPI_Attr_delete

MPI_Attr_get

MPI_Attr_put

MPI_Keyval_create

MPI_Pack

MPI_Pack_size

MPI_Pcontrol

MPI_Unpack

✓ **Αποτελέσματα Συναρτήσεων :**

MPI_CART

MPI_COMM_NULL

MPI_CONGRUENT

MPI_DATATYPE_NULL

MPI_ERRHANDLER_NULL

MPI_ERROR

MPI_GRAPH

MPI_IDENT

MPI_OP_NULL

MPI_PENDING

MPI_PROC_NULL

MPI_REQUEST_NULL

MPI_SIMILAR

MPI_SUCCESS

MPI_UNDEFINED

MPI_UNEQUAL

✓ **Μηνύματα Λάθους :**

MPI_ERR_ARG

MPI_ERR_BUFFER

MPI_ERR_COMM

MPI_ERR_COUNT

MPI_ERR_DIMS

MPI_ERR_GROUP

MPI_ERR_IN_STATUS

MPI_ERR_INTERN
MPI_ERR_LASTCODE
MPI_ERR_OP
MPI_ERR_OTHER
MPI_ERR_PENDING
MPI_ERR_RANK
MPI_ERR_REQUEST
MPI_ERR_ROOT
MPI_ERR_TAG
MPI_ERR_TOPOLOGY

✓ **Ορίσματα Κλήσης :**

MPI_ANY_SOURCE
MPI_ANY_TAG

✓ **Συμβολικές Σταθερές :**

MPI_2DOUBLE_PRECISION
MPI_2INIT
MPI_2INTEGER
MPI_2REAL
MPI_BAND
MPI_BOR
MPI_BOTTOM
MPI_BSEND_OVERHEAD
MPI_BXOR
MPI_COMM_SELF
MPI_COMM_WORLD
MPI_GROUP_EMPTY
MPI_ERRORS_ARE_FATAL
MPI_ERRORS_RETURN
MPI_GROUP_NULL
MPI_HOST
MPI_IO

MPI_KEYVAL_INVALID
MPI_LAND
MPI_LB
MPI_LOR
MPI_LXOR
MPI_MAX
MPI_MAX_ERROR_STRING
MPI_MAX_PROCESSOR_NAME
MPI_MAXLOC
MPI_MIN
MPI_MINLOC
MPI_PROD
MPI_SOURCE
MPI_STATUS_SIZE
MPI_SUM
MPI_TAG
MPI_TAG_UB
MPI_UB
MPI_WTIME_IS_GLOBAL

ΠΑΡΑΡΤΗΜΑ Α

Πηγαίος Κώδικας

Στο παράρτημα αυτό παρουσιάζεται ο κώδικας που χρησιμοποιήθηκε για τη ρύθμιση σημαντικών παραμέτρων των wikis.

LocalSettings.php

```
<?php

# This file was automatically generated by the MediaWiki installer.
# If you make manual changes, please keep track in case you need to
# recreate them later.
#
# See includes/DefaultSettings.php for all configurable settings
# and their default values, but don't forget to make changes in
# _this_
# file, not there.
#
# Further documentation for configuration settings may be found at:
# http://www.mediawiki.org/wiki/Manual:Configuration_settings

# If you customize your file layout, set $IP to the directory that
# contains
# the other MediaWiki files. It will be used as a base to locate
# files.
if( defined( 'MW_INSTALL_PATH' ) ) {
    $IP = MW_INSTALL_PATH;
} else {
    $IP = dirname( __FILE__ );
}

$path = array( $IP, "$IP/includes", "$IP/languages" );
set_include_path( implode( PATH_SEPARATOR, $path ) . PATH_SEPARATOR
. get_include_path() );

require_once( "$IP/includes/DefaultSettings.php" );

# If PHP's memory limit is very low, some operations may fail.
# ini_set( 'memory_limit', '20M' );
```

```
if ( $wgCommandLineMode ) {
    if ( isset( $_SERVER ) && array_key_exists( 'REQUEST_METHOD',
$_SERVER ) ) {
        die( "This script must be run from the command line\n"
);
    }
}

## Uncomment this to disable output compression
# $wgDisableOutputCompression = true;

$wgSitename          = "Matlab";

## The URL base path to the directory containing the wiki;
## defaults for all runtime URL paths are based off of this.
## For more information on customizing the URLs please see:
## http://www.mediawiki.org/wiki/Manual:Short_URL
$wgScriptPath        = "/kdiamant/matlab";
$wgScriptExtension   = ".php";

## UPO means: this is also a user preference option

$wgEnableEmail       = true;
$wgEnableUserEmail   = true; # UPO

$wgEmergencyContact  = "v13@it.teithe.gr";
$wgPasswordSender    = "v13@it.teithe.gr";

$wgEnotifUserTalk    = true; # UPO
$wgEnotifWatchlist   = true; # UPO
$wgEmailAuthentication = true;

## Database settings
$wgDBtype             = "mysql";
$wgDBserver           = "localhost";
$wgDBname             = "matlab";
$wgDBuser             = "kdiamant";
$wgDBpassword        = "mp1mp1!@#";

# MySQL specific settings
$wgDBprefix           = "";
```

```
# MySQL table options to use during installation or update
$wgDBTableOptions = "ENGINE=InnoDB, DEFAULT CHARSET=binary";

# Experimental charset support for MySQL 4.1/5.0.
$wgDBmysql5 = true;

## Shared memory settings
$wgMainCacheType = CACHE_NONE;
$wgMemCachedServers = array();

## To enable image uploads, make sure the 'images' directory
## is writable, then set this to true:
$wgEnableUploads = true;
$wgFileExtensions = array('png', 'gif', 'jpg', 'jpeg', 'doc',
'xls', 'mpp', 'pdf', 'ppt', 'tiff', 'bmp', 'docx', 'xlsx', 'pptx',
'ps');

# $wgUseImageMagick = true;
# $wgImageMagickConvertCommand = "/usr/bin/convert";

## If you use ImageMagick (or any other shell command) on a
## Linux server, this will need to be set to the name of an
## available UTF-8 locale
$wgShellLocale = "en_US.utf8";

## If you want to use image uploads under safe mode,
## create the directories images/archive, images/thumb and
## images/temp, and make them all writable. Then uncomment
## this, if it's not already uncommented:
$wgHashedUploadDirectory = false;

## If you have the appropriate support software installed
## you can enable inline LaTeX equations:
$wgUseTeX = false;

$wgLocalInterwiki = strtolower( $wgSitename );

$wgLanguageCode = "el";
```

Πτυχιακή εργασία των φοιτητριών Βαλιαντή Βασιλική - Πολυχρονιάδου Βασιλική

```
$wgSecretKey =
"f11ce767e55fdabe82eb83495b0c163bb5ba0a4a92f8d0f1ac06000a3f2a8e3a";

## Default skin: you can change the default skin. Use the internal
symbolic
## names, ie 'standard', 'nostalgia', 'cologneblue', 'monobook':
$wgDefaultSkin = 'monobook';

## For attaching licensing metadata to pages, and displaying an
## appropriate copyright notice / icon. GNU Free Documentation
## License and Creative Commons licenses are supported so far.
# $wgEnableCreativeCommonsRdf = true;
$wgRightsPage = ""; # Set to the title of a wiki page that
describes your license/copyright
$wgRightsUrl = "";
$wgRightsText = "";
$wgRightsIcon = "";
# $wgRightsCode = ""; # Not yet used

$wgDiff3 = "/usr/bin/diff3";

# When you make changes to this configuration file, this will make
# sure that cached pages are cleared.
$wgCacheEpoch = max( $wgCacheEpoch, gmdate( 'YmdHis', @filemtime(
__FILE__ ) ) );
```

MonoBook.php

```
<?php
/**
 * MonoBook nouveau
 *
 * Translated from gwicke's previous TAL template version to remove
 * dependency on PHPTAL.
 *
 * @todo document
 * @file
 * @ingroup Skins
 */

if( !defined( 'MEDIAWIKI' ) )
    die( -1 );
```



```
/**
 * Inherit main code from SkinTemplate, set the CSS and template
filter.
 * @todo document
 * @ingroup Skins
 */
class SkinMonoBook extends SkinTemplate {
    /** Using monobook. */
    function initPage( OutputPage $out ) {
        parent::initPage( $out );
        $this->skinname = 'monobook';
        $this->stylename = 'monobook';
        $this->template = 'MonoBookTemplate';
    }

    function setupSkinUserCss( OutputPage $out ) {
        global $wgHandheldStyle;

        parent::setupSkinUserCss( $out );

        // Append to the default screen common & print
styles...
        $out->addStyle( 'monobook/main.css', 'screen' );
        if( $wgHandheldStyle ) {
            // Currently in testing... try 'chick/main.css'
            $out->addStyle( $wgHandheldStyle, 'handheld' );
        }

        $out->addStyle( 'monobook/IE50Fixes.css', 'screen', 'lt
IE 5.5000' );
        $out->addStyle( 'monobook/IE55Fixes.css', 'screen', 'IE
5.5000' );
        $out->addStyle( 'monobook/IE60Fixes.css', 'screen', 'IE
6' );
        $out->addStyle( 'monobook/IE70Fixes.css', 'screen', 'IE
7' );

        $out->addStyle( 'monobook/rtl.css', 'screen', '', 'rtl'
);
    }
}
```

```
    }  
}  
  
/**  
 * @todo document  
 * @ingroup Skins  
 */  
class MonoBookTemplate extends QuickTemplate {  
    var $skin;  
    /**  
     * Template filter callback for MonoBook skin.  
     * Takes an associative array of data set from a  
SkinTemplate-based  
     * class, and a wrapper for MediaWiki's localization  
database, and  
     * outputs a formatted page.  
     *  
     * @access private  
     */  
    function execute() {  
        global $wgRequest;  
        $this->skin = $skin = $this->data['skin'];  
        $action = $wgRequest->getText( 'action' );  
  
        // Suppress warnings to prevent notices about missing  
indexes in $this->data  
        wfSuppressWarnings();  
  
?><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="<?php $this->text('xhtmldefaultnamespace') ?>" <?php  
    foreach($this->data['xhtmlnamespaces'] as $tag => $ns) {  
        ?>xmlns:<?php echo "{$tag}=\"{$ns}\" ";  
    } ?>xml:lang="<?php $this->text('lang') ?>" lang="<?php  
$this->text('lang') ?>" dir="<?php $this->text('dir') ?>">  
    <head>  
        <meta http-equiv="Content-Type" content="<?php $this->  
>text('mimetype') ?>; charset="<?php $this->text('charset') ?>" />  
        <?php $this->html('headlinks') ?>  
        <title><?php $this->text('pagetitle') ?></title>  
        <?php $this->html('csslinks') ?>
```

```
<!--[if lt IE 7]><script type="<?php $this->text('jsmimetype') ?>" src="<?php $this->text('stylepath') ?>/common/IEFixes.js?<?php echo $GLOBALS['wgStyleVersion'] ?>"></script>
<meta http-equiv="imagetoolbar" content="no" /><![endif]-->

<?php print Skin::makeGlobalVariablesScript( $this->data ); ?>

<script type="<?php $this->text('jsmimetype') ?>" src="<?php $this->text('stylepath') ?>/common/wikibits.js?<?php echo $GLOBALS['wgStyleVersion'] ?>"><!-- wikibits js --></script>
<!-- Head Scripts -->
<?php $this->html('headscripts') ?>
<?php if($this->data['jsvarurl']) { ?>
    <script type="<?php $this->text('jsmimetype') ?>" src="<?php $this->text('jsvarurl') ?>"><!-- site js --></script>
<?php } ?>
<?php if($this->data['pagecss']) { ?>
    <style type="text/css"><?php $this->html('pagecss') ?></style>
<?php }

    if($this->data['usercss']) { ?>
        <style type="text/css"><?php $this->html('usercss') ?></style>
<?php }

    if($this->data['userjs']) { ?>
        <script type="<?php $this->text('jsmimetype') ?>" src="<?php $this->text('userjs') ?>"></script>
<?php }

        if($this->data['userjsprev']) { ?>
            <script type="<?php $this->text('jsmimetype') ?>"><?php $this->html('userjsprev') ?></script>
<?php }

            if($this->data['trackbackhtml']) print $this->data['trackbackhtml']; ?>
```

```
<script type="text/javascript">
function hideDiv(divid) {
    if(document.getElementById) {
        document.getElementById(divid).style.visibility =
'hidden';
    }else{
        if(document.layers){
            document.divid.visibility = 'hidden';
        }else{
            document.all.divid.style.visibility =
'hidden';
        }
    }
}
function showDiv(divid) {
    if(document.getElementById) {
        document.getElementById(divid).style.visibility =
'visible';
    }else{
        if (document.layers){
            document.divid.visibility = 'visible';
        }else{
            document.all.divid.style.visibility =
'visible';
        }
    }
}
</script>
```

</head>

```
<body<?php if($this->data['body_ondblclick']) { ?>
ondblclick="<?php $this->text('body_ondblclick') ?>"<?php } ?>
<?php if($this->data['body_onload']) { ?> onload="<?php $this-
>text('body_onload') ?>"<?php } ?>
```

```
class="mediawiki <?php $this->text('dir') ?> <?php $this-
>text('pageclass') ?> <?php $this->text('skinnameclass') ?>">
    <div id="globalWrapper">
        <div id="column-content">
            <div id="content">
                <a name="top" id="top"></a>
                <?php if($this->data['sitenotice']) { ?><div
id="siteNotice"><?php $this->html('sitenotice') ?></div><?php } ?>
                <h1 id="firstHeading" class="firstHeading"><?php $this-
>data['displaytitle']!=""?$this->html('title'):$this->text('title')
?></h1>
                <div id="bodyContent">
                    <h3 id="siteSub"><?php $this->msg('tagline')
?></h3>
                    <div id="contentSub"><?php $this-
>html('subtitle') ?></div>
                    <?php if($this->data['undelete']) { ?><div
id="contentSub2"><?php $this->html('undelete') ?></div><?php }
?>
                    <?php if($this->data['newtalk'] ) { ?><div
class="usermessage"><?php $this->html('newtalk') ?></div><?php }
?>
                    <?php if($this->data['showjumplinks']) { ?><div
id="jump-to-nav"><?php $this->msg('jumpto') ?> <a href="#column-
one"><?php $this->msg('jumptonavigation') ?></a>, <a
href="#searchInput"><?php $this->msg('jumptosearch')
?></a></div><?php } ?>
                    <!-- start content -->
                    <?php $this->html('bodytext') ?>
                    <?php if($this->data['catlinks']) { $this-
>html('catlinks'); } ?>
                    <!-- end content -->
                    <?php if($this->data['dataAfterContent']) {
$this->html ('dataAfterContent'); } ?>
                    <div class="visualClear"></div>
                </div>
            </div>
        </div>
    <div id="column-one">
        <div id="p-cactions" class="portlet">
            <h5><?php $this->msg('views') ?></h5>
```

```
<div class="pBody">
    <ul>
        <?php
foreach($this->data['content_actions'] as $key =>
$tab) {
            echo '
            <li id="' . Sanitizer::escapeId( "ca-$key"
) . '"';
                if( $tab['class'] ) {
                    echo '
class="' . htmlspecialchars($tab['class']) . '"';
                }
                echo '><a
href="' . htmlspecialchars($tab['href']) . '"';
                # We don't want to give the watch tab
an accesskey if the
                # page is being edited, because that
conflicts with the
                # accesskey on the watch checkbox.
We also don't want to
                # give the edit tab an accesskey,
because that's fairly su-
                # perfluous and conflicts with an
accesskey (Ctrl-E) often
                # used for editing in Safari.
                if( in_array( $action, array( 'edit',
'submit' ) )
                && in_array( $key, array( 'edit',
'watch', 'unwatch' ))) {
                    echo $skin->tooltip( "ca-$key"
);
                } else {
                    echo $skin-
>tooltipAndAccesskey( "ca-$key" );
                }
                echo
                '>' . htmlspecialchars($tab['text']) . '</a></li>';
            } ?>
        </ul>
    </div>
</div>
<div class="portlet" id="p-personal">
```

```

        <h5><?php $this->msg('personaltools') ?></h5>
        <div class="pBody">
            <ul>
<?php                foreach($this->data['personal_urls'] as $key =>
$item) { ?>
                    <li id="<?php echo Sanitizer::escapeId(
"pt-$key" ) ?>"<?php
                        if ($item['active']) { ?>
class="active"<?php } ?><a href="<?php
                            echo htmlspecialchars($item['href'])
?>"<?php echo $skin->tooltipAndAccesskey('pt-'. $key) ?><?php
                                if(!empty($item['class'])) { ?>
class="<?php
                                    echo htmlspecialchars($item['class'])
?>"<?php } ?><?php
                                        echo htmlspecialchars($item['text'])
?></a></li>
<?php                } ?>
                    </ul>
                </div>
            </div>
            <div class="portlet" id="p-logo">
                <a style="background-image: url(<?php $this-
>text('logopath') ?>);" <?php
                    ?>href="<?php echo htmlspecialchars($this-
>data['nav_urls']['mainpage']['href'])?>"<?php
                        echo $skin->tooltipAndAccesskey('p-logo') ?></a>
                </div>
                <script type="<?php $this->text('jsmime') ?>"> if
(window.isMSIE55) fixalpha(); </script>
<?php
                $sidebar = $this->data['sidebar'];
                if ( !isset( $sidebar['SEARCH'] ) ) $sidebar['SEARCH']
= true;
                if ( !isset( $sidebar['TOOLBOX'] ) )
$sidebar['TOOLBOX'] = true;
                if ( !isset( $sidebar['LANGUAGES'] ) )
$sidebar['LANGUAGES'] = true;
                foreach ( $sidebar as $boxName => $cont) {
                    if ( $boxName == 'SEARCH' ) {
                        $this->searchBox();

```

```
        } elseif ( $boxName == 'TOOLBOX' ) {
            $this->toolbox();
        } elseif ( $boxName == 'LANGUAGES' ) {
            $this->languageBox();
        } else {
            $this->customBox( $boxName, $cont );
        }
    }
?>

</div><!-- end of the left (by default at least) column
-->

<div class="visualClear"></div>
<div id="footer">
<?php
    if($this->data['poweredbyico']) { ?>
        <div id="f-poweredbyico"><?php $this-
>html('poweredbyico') ?></div>
<?php }

    if($this->data['copyrightico']) { ?>
        <div id="f-copyrightico"><?php $this-
>html('copyrightico') ?></div>
<?php }

    // Generate additional footer links
    $footerlinks = array(
        'lastmod', 'viewcount', 'numberofwatchingusers',
'credits', 'copyright',
        'privacy', 'about', 'disclaimer', 'tagline',
    );
    $validFooterLinks = array();
    foreach( $footerlinks as $aLink ) {
        if( isset( $this->data[$aLink] ) && $this-
>data[$aLink] ) {
            $validFooterLinks[] = $aLink;
        }
    }
    if ( count( $validFooterLinks ) > 0 ) {
?>
        <ul id="f-list">
<?php
            foreach( $validFooterLinks as $aLink ) {
```


Πτυχιακή εργασία των φοιτητριών Βαλιαντή Βασιλική - Πολυχρονιάδου Βασιλική

```

        if( isset( $this->data[$aLink] ) && $this-
>data[$aLink] ) {
?>
        <li id="<?php echo$aLink?>"><?php
$this->html($aLink) ?></li>
<?php
        }
        }
?>
        </ul>
<?php }
?>
        </div>
</div>
<?php $this->html('bottomscripts'); /* JS call to runBodyOnloadHook
*/ ?>
<?php $this->html('reporttime') ?>
<?php if ( $this->data['debug'] ): ?>
<!-- Debug output:
<?php $this->text( 'debug' ); ?>

-->
<?php endif; ?>

<div id="hideshow" style="font-size:100%; border:2px groove;
background-color:#e0e0e0; font-weight:bold; position: fixed;
right:10px; width: 152px; top:18px; ">
<div style="font-size:100; border:0px solid; background-
color:#D3D3D3 ; ">

&Pi; &Epsilon; &Rho; &Iota; &Epsilon; &Chi; &Omicron; &Mu; &Epsilon; &Nu; &Al
pha;
&nbsp;
<a href="#" onclick="showDiv('hideshow');">&dArr;</a>

&nbsp;
<a href="#" onclick="hideDiv('hideshow');">&uArr;</a>

</div>
</div>
```

```
</body></html>
<?php
    wfRestoreWarnings();
    } // end of execute() method

    /*****
    *****/
    function searchBox() {
        global $wgUseTwoButtonsSearchForm;
?>
    <div id="p-search" class="portlet">
        <h5><label for="searchInput"><?php $this->msg('search')
?></label></h5>
        <div id="searchBody" class="pBody">
            <form action="<?php $this->text('wgScript') ?>"
id="searchform"><div>
                <input type='hidden' name="title"
value="<?php $this->text('searchtitle') ?>"/>
                <input id="searchInput" name="search"
type="text"<?php echo $this->skin->tooltipAndAccesskey('search');
                    if( isset( $this->data['search'] ) ) )
{
                        ?> value="<?php $this-
>text('search') ?>"<?php } ?> />
                <input type='submit' name="go"
class="searchButton" id="searchGoButton" value="<?php $this-
>msg('searcharticle') ?>"<?php echo $this->skin-
>tooltipAndAccesskey( 'search-go' ); ?> /><?php if
($wgUseTwoButtonsSearchForm) { ?>&nbsp;
                    <input type='submit' name="fulltext"
class="searchButton" id="mw-searchButton" value="<?php $this-
>msg('searchbutton') ?>"<?php echo $this->skin-
>tooltipAndAccesskey( 'search-fulltext' ); ?> /><?php } else { ?>

                <div><a href="<?php $this-
>text('searchaction') ?>" rel="search"><?php $this-
>msg('powersearch-legend') ?></a></div><?php } ?>
```

```
        </div></form>
    </div>
</div>
<?php
}

/*****
*****/

function toolbox() {
?>
<div class="portlet" id="p-tb">
    <h5><?php $this->msg('toolbox') ?></h5>
    <div class="pBody">
        <ul>
<?php
            if($this->data['notspecialpage']) { ?>
                <li id="t-whatlinkshere"><a href="<?php
                    echo htmlspecialchars($this-
>data['nav_urls']['whatlinkshere']['href'])
                    ?>"<?php echo $this->skin-
>tooltipAndAccesskey('t-whatlinkshere') ?>><?php $this-
>msg('whatlinkshere') ?></a></li>
<?php
                if( $this-
>data['nav_urls']['recentchangeslinked'] ) { ?>
                    <li id="t-recentchangeslinked"><a
href="<?php
                        echo htmlspecialchars($this-
>data['nav_urls']['recentchangeslinked']['href'])
                        ?>"<?php echo $this->skin-
>tooltipAndAccesskey('t-recentchangeslinked') ?>><?php $this-
>msg('recentchangeslinked') ?></a></li>
<?php                }
                }
                if(isset($this->data['nav_urls']['trackbacklink'])) {
?>
                    <li id="t-trackbacklink"><a href="<?php
                        echo htmlspecialchars($this-
>data['nav_urls']['trackbacklink']['href'])
```

```

        ?>"<?php echo $this->skin-
>tooltipAndAccesskey('t-trackbacklink') ?><?php $this-
>msg('trackbacklink') ?></a></li>
<?php }

        if($this->data['feeds']) { ?>
            <li id="feedlinks"><?php foreach($this-
>data['feeds'] as $key => $feed) {
                ?><a id="<?php echo
Sanitizer::escapeId( "feed-$key" ) ?>" href="<?php
                    echo htmlspecialchars($feed['href'])
?>" rel="alternate" type="application/<?php echo $key ?>+xml"
class="feedlink"<?php echo $this->skin->tooltipAndAccesskey('feed-
'.$key) ?><?php echo htmlspecialchars($feed['text'])?></a>&nbsp;
                    <?php } ?></li><?php
            }

            foreach( array('contributions', 'log', 'blockip',
'emailuser', 'upload', 'specialpages') as $special ) {

                if($this->data['nav_urls'][$special]) {
                    ?><li id="t-<?php echo $special ?>"><a
href="<?php echo htmlspecialchars($this-
>data['nav_urls'][$special]['href'])
                    ?>"<?php echo $this->skin-
>tooltipAndAccesskey('t-'.$special) ?><?php $this->msg($special)
?></a></li>
<?php }
                }

                if(!empty($this->data['nav_urls']['print']['href'])) {
?>
                    <li id="t-print"><a href="<?php echo
htmlspecialchars($this->data['nav_urls']['print']['href'])
                    ?>" rel="alternate"<?php echo $this->skin-
>tooltipAndAccesskey('t-print') ?><?php $this-
>msg('printableversion') ?></a></li><?php
                }

                if(!empty($this-
>data['nav_urls']['permalink']['href'])) { ?>

```

```

        <li id="t-permalink"><a href="<?php echo
htmlspecialchars($this->data['nav_urls']['permalink']['href'])
        ?>"<?php echo $this->skin-
>tooltipAndAccesskey('t-permalink') ?>><?php $this-
>msg('permalink') ?></a></li><?php
        } elseif ($this->data['nav_urls']['permalink']['href']
=== '') { ?>
        <li id="t-ispermalink"<?php echo $this-
>skin->tooltip('t-ispermalink') ?>><?php $this->msg('permalink')
?></li><?php
        }

        wfRunHooks( 'MonoBookTemplateToolboxEnd', array( &$this
) );

        wfRunHooks( 'SkinTemplateToolboxEnd', array( &$this )
);
?>

        </ul>
    </div>
</div>
<?php
}

/*****
*****/
function languageBox() {
    if( $this->data['language_urls'] ) {
?>
    <div id="p-lang" class="portlet">
        <h5><?php $this->msg('otherlanguages') ?></h5>
        <div class="pBody">
            <ul>
<?php        foreach($this->data['language_urls'] as $langlink) { ?>
                <li class="<?php echo
htmlspecialchars($langlink['class']) ?>"><?php
                    ?><a href="<?php echo
htmlspecialchars($langlink['href']) ?>"><?php echo
$langlink['text'] ?></a></li>
<?php        } ?>
            </ul>
        </div>
    </div>

```

```

        </div>
<?php
        }
    }

    /*****
    *****/
    function customBox( $bar, $cont ) {
?>
        <div class='generated-sidebar portlet' id='<?php echo
Sanitizer::escapeId( "p-$bar" ) ?>'<?php echo $this->skin-
>tooltip('p-'. $bar) ?>>
            <h5><?php $out = wfMsg( $bar ); if (wfEmptyMsg($bar,
$out)) echo $bar; else echo $out; ?></h5>
            <div class='pBody'>
<?php    if ( is_array( $cont ) ) { ?>
                <ul>
<?php                foreach($cont as $key => $val) { ?>
                    <li id="<?php echo
Sanitizer::escapeId($val['id']) ?>"<?php
                        if ( $val['active'] ) { ?>
class="active" <?php }
                            ?>><a href="<?php echo
htmlspecialchars($val['href']) ?>"<?php echo $this->skin-
>tooltipAndAccesskey($val['id']) ?>><?php echo
htmlspecialchars($val['text']) ?></a></li>
<?php                } ?>
                </ul>
<?php    } else {
                # allow raw HTML block to be defined by
extensions
                print $cont;
            }
?>
        </div>
    </div>
<?php
    }

} // end of class

```

ΠΑΡΑΡΤΗΜΑ Β

ΒΙΒΛΙΟΓΡΑΦΙΑ

References

- [1] Γ. Γεωργίου - Χ. Ξενοφώντος(2007)- Εισαγωγή στη *MATLAB*, Λευκωσία
- [2] Κωνσταντίνος Διαμαντάρας, Κωνσταντίνος Γουλιάνας - *Εργαστηριακές Ασκήσεις Νευρωνικών Δικτύων και Σύντομο Εγχειρίδιο Matlab*, Αλεξάνδρειο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Θεσσαλονίκης
- [3]Κωνσταντίνος Παπαρίζος - *MATLAB 6.5*, Πανεπιστήμιο Μακεδονίας, ΖΥΓΟΣ, Θεσσαλονίκη
- [4]Ευάγγελος Β.Χατζίκος –*MATLAB Για Επιστήμονες και Μηχανικούς*, Εκδόσεις Τζιόλα
- [5] Αθανάσιος Ι. Μάργαρης – *MPI Θεωρία και Εφαρμογες*
- [6]Γραμματη Παντζιου – Αλεξανδρος Τομαρας- *Στοιχεία Παράλληλου Υπολογισμού*
- [7] *Jun Nakano, Yukiya Aoyama - Practical MPI Programming*
- [8] *Using MPI, by William Gropp, Erwin Lusk, and Anthony Skjellum*
- [9] *MPI Annotated Reference manual, by Marc Snir, et al.*
- [10] Ησαΐάδης Σταύρος - *MPI Introductions*
- [11] Ιερεμίας Ντουμπάκης - *Προχωρημένες Αρχιτεκτονικές Συστημάτων Υπολογιστών Υψηλής Απόδοσης και Τεχνικές απεικόνισης Αλγορίθμων*
- [12] Ρογκάκος Γεώργιος -*MPI, Δομή Open MPI, Συλλογικές επικοινωνίες .Και πρόσθεση νέου component συλλογικών επικοινωνιών*
- [13] Προγραμματισμός Συστημάτων Υψηλών Επιδόσεων Ειδικά Θέματα Λογισμικού – Εισαγωγή στο MPI - Παναγιώτης Χατζηδούκας

Electronic Links

<http://el.wikipedia.org/wiki>

<http://www.mediawiki.org/wiki/MediaWiki>

<http://hydra.it.teithe.gr/vpn>

<http://www.mathworks.com>

<http://aetos.it.teithe.gr/~kdiamant>

<http://www.mpi-forum.org/> (MPI forum)

<http://www.mcs.anl.gov/research/projects/mpi/> (Site για το MPI στο Argonne Labs)

<http://static.msi.umn.edu/tutorial/scicomp/general/MPI/>

http://en.wikipedia.org/wiki/Message_Passing_Interface/

<http://computing.llnl.gov/tutorials/mpi/>

www.mcs.anl.gov/mpi/

