



Πανεπιστήμιο Δυτικής Μακεδονίας  
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

---

# Συστήματα Παράλληλης & Κατανεμημένης Επεξεργασίας

## Ενότητα 2: OpenMPI

Δρ. Μηνάς Δασυγένης

[mdasyg@ieee.org](mailto:mdasyg@ieee.org)

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

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

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

---



Πανεπιστήμιο Δυτικής Μακεδονίας



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

# Άδειες Χρήσης

---

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
*επένδυση στην κοινωνία της γνώσης*  
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
Πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# Σκοπός της Ενότητας

---

- Η κατανόηση της αποστολής και λήψης μηνυμάτων με το MPI.



# Υλοποιήσεις του MPI

---

- OpenMPI (The Open MPI Development Team).
- MPICH, MPICH2 (Argonne National Laboratory).
- LAM MPI (Local Area Multicomputer - ohio Supercomputer Center).
- CHIMP (Common High Level Interface to Message Passing -Edinburg Parallel Computing Center).
- Εμπορικές Υλοποιήσεις: IBM MPI, HP MPI, Sun MPI, Digital MPI, SGI MPI – προσαρμοσμένες στις ανάγκες κάθε εταιρίας.



# Πότε θα χρησιμοποιήσουμε τη συνάρτηση MPI;

---

- Όταν χρειαζόμαστε ένα portable παράλληλο πρόγραμμα.
- Όταν θέλουμε να γράψουμε μια παράλληλη βιβλιοθήκη.
- Όταν μας ενδιαφέρει η απόδοση.



# Πότε δε θα χρησιμοποιήσουμε το OpenMPI;

---

- Όταν μπορούμε να χρησιμοποιήσουμε κάποια παράλληλη γλώσσα, π.χ. Paralell Fortran 90.
- Όταν δεν χρειαζόμαστε παραλληλισμό.



# Τι άλλο υπάρχει εκτός από το MPI;

---

- PVM – Parallel Virtual Machine.
- Κατάλληλο για ετερογενή δίκτυα.
- Εφαρμογές μεγάλης σταθερότητας.
- Επιτρέπει προσθήκη / αφαίρεση υπολογιστικών κόμβων.





# Στο προηγούμενο εργαστήριο...

---

```
#include <stdio.h>
#include <mpi.h>
int main(int argc, char *argv[])
{
int rank, size;
MPI_Init(&argc, &argv); MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
MPI_Comm_size(MPI_COMM_WORLD, &size);
printf ("Hello world! I am %d of %d \n", rank, size);
MPI_Finalize();
return 0;
}
```



# Είδη επικοινωνίας στο OpenMPI

---

- Επικοινωνία από σημείο σε σημείο (point to point).
- communication: blocking, non blocking).
- Συλλογικές επικοινωνίες (collective communications: broadcasting scattering, gather, reduce).



# Τύποι δεδομένων στο OpenMPI

---

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



# Πρότυπα συναρτήσεων

## MPI\_Abort

---

- Το πρότυπό έχει την μορφή:
  - `Int MPI_Abort (MPI_Comm comm, int * err);`
- Καλείται στην περίπτωση που η `MPI_Init` αποτυγχάνει να αρχικοποιήσει το περιβάλλον MPI.
- Παράδειγμα:
  - **`MPI_Abort (MPI_COMM_WORLD, err);`**
  - Η `MPI_Init` αν επιτύχει την αρχικοποίηση επιστρέφει 0.
  - Συνήθως ελέγχουμε την τιμή που επιστρέφει η `MPI_Init` και αν είναι διαφορετική από 0 (με μια δομή `if`) τότε καλείται η `MPI_Abort`.



# MPI\_Init και MPI\_Abort

---

- Για να γίνει ο έλεγχος της αρχικοποίησης, ελέγχουμε την τιμή που επιστρέφει η συνάρτηση MPI\_Init (ή οποιαδήποτε άλλη συνάρτηση).
- Ακολουθεί μια δομή ελέγχου:

```
if (error_code != MPI_SUCCESS )  
{ .... }
```

- Η σταθερά MPI\_SUCCESS ορίζεται στο include αρχείο. Μπορεί να χρησιμοποιηθεί για να ελεγχθούν όλες τις συναρτήσεις του MPI.
- Πρέπει να έχει οριστεί:

```
int error_code;
```



# Ρουτίνα αποστολής μηνυμάτων MPI Send

---

MPI\_Send (void \*buffer , int count, MPI\_Datatype data\_type, int destination, int tag, MPI\_Comm communicator):

- **buffer**: Δεδομένα που θα αποσταλούν.
- **count**: Αριθμός στοιχείων του buffer.
- **data\_type**: Τύπος των δεδομένων (π.χ. floats, integers).
- **destination**: Παραλήπτης δεδομένων.
- **tag**: Ο δείκτης της επικοινωνίας.
- **communicator**: Το όνομα της ομάδας των διεργασιών.



# Ρουτίνα αποστολής μηνυμάτων MPI\_Send

---

- Η συνάρτηση MPI Send χρησιμοποιείται για αποστολή μηνύματος σε κάποια από τις διεργασίες του communicator.
- Παράδειγμα χρήσης:  
**MPI\_Send (buffer, 100,  
MPI\_CHAR, 1, 0,  
MPI\_COMM\_WORLD);**



# Ρουτίνα λήψης μηνυμάτων

## MPI\_Recv (1/2)

---

MPI\_Recv(void \*buffer , int count, MPI\_Datatype data\_type, int source, int tag, MPI\_Comm communicator , MPI\_Status status):

- **buffer**: Δεδομένα που θα αποσταλούν.
- **count**: Αριθμός στοιχείων του buffer.
- **data\_type**: Τύπος των δεδομένων (π.χ. floats, integers).
- **source**: Αποστολέας δεδομένων.
- **tag**: Ο δείκτης της επικοινωνίας.
- **communicator**: Το όνομα της ομάδας των διεργασιών.
- **status**: Χρησιμοποιείται για έλεγχο σε περίπτωση σφάλματος.





# Ρουτίνα λήψης μηνυμάτων

## MPI\_Recv (2/2)

---

- Η συνάρτηση MPI Recv χρησιμοποιείται για παραλαβή μηνύματος από κάποια από τις διεργασίες του communicator.
- Παράδειγμα χρήσης:  
**MPI\_Recv (buffer, 50,  
MPI\_DOUBLE, 3, 0,  
MPI\_COMM\_WORLD, &status);**
- Απαιτείται δήλωση της μορφής:  
**MPI\_Status status;**



# Χρήση MPI\_Send, MPI\_Receive

---

- Δηλώνουμε μια μεταβλητή για μήνυμα αποστολής:
  - **char outMessage[]='hi there'**
- Δηλώνουμε μια μεταβλητή για μήνυμα λήψης.  
Προσοχή στο μέγεθος του εισερχόμενου μηνύματος:
  - **char inMessage[7]**
- Δηλώνουμε το **MPI\_Status status;**



---

# Τέλος Ενότητας



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

