



**Πανεπιστήμιο Δυτικής Αττικής
Σχολή Μηχανικών
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών**

**Εργαστήριο Εισαγωγής
στον παράλληλο υπολογισμό**

**ΝΙΚΟΛΑΟΣ ΘΩΜΑΣ
ΑΜ: 21390068
ΤΜΗΜΑ: Ε8**

**ΑΘΗΝΑ
Τρίτη, 5 Δεκεμβρίου 2023**

```
invalid@invalid:/media/sf_OneDrive_-_University_of_West_Attica/Εισαγωγή στο παράλληλο υπολογισμό$ mpicc -o thomasex thomas.c
thomas.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
5 | main(int argc, char **argv)
  | ^~~~~
invalid@invalid:/media/sf_OneDrive_-_University_of_West_Attica/Εισαγωγή στο παράλληλο υπολογισμό$
```

Εικόνα 1: C file in to ex.

```
invalid@invalid:/media/sf_OneDrive_-_University_of_West_Attica/Εισαγωγή στο παράλληλο υπολογισμό$ mpiexec -n 3 ./thomasex
Type 1 to continue or anything else to exit!
```

Εικόνα 2: Running the ex.

Στο παρακάτω στιγμιότυπο οθόνης εμφανίζεται ο κώδικας που χρησιμοποιείται για το menu επιλογών. Ξεκινάει με την συνάρτηση `while()` η οποία θα εκτελείται μέχρι το key να είναι διαφορετικό του 1. Διαβάζει με την `scanf()` το input από τον χρήστη, το στέλνει σε κάθε διεργασία (ώστε να μπορεί να εκτελέσει την `MPI_FINALIZE()` σε περίπτωση που δεν είναι 1) και αν είναι 1 συνεχίζει, αλλιώς τερματίζει.

```
while (key)
{
    if (my_rank == 0)
    {
        printf("\nType 1 to continue or anything else to exit!\n");
        scanf("%d", &key);
        for (target = 1; target < p; target++)
            MPI_Send(&key, 1, MPI_INT, target, tag1, MPI_COMM_WORLD);
    }
    else
    {
        MPI_Recv(&key, 1, MPI_INT, 0, tag1, MPI_COMM_WORLD, &status);
    }

    if (key != 1)
    {
        MPI_Finalize();
        exit(1);
    }
}
```

Εικόνα 3: Menu επιλογών.

Στο παρακάτω στιγμιότυπο οθόνης εμφανίζεται ο κώδικας που χρησιμοποιείται για να δώσει ο χρήστης το διάνυσμα. Αν είναι η διεργασία 0, τότε ξεκινάει ζητώντας από τον χρήστη το πλήθος των αριθμών του διανύσματος και στην συνέχεια τους αριθμούς του διανύσματος. Διαβάζει όλους τους αριθμούς και τους βάζει σε ένα int array. Στην συνέχεια στέλνει το πλήθος των αριθμών του διανύσματος σε κάθε διεργασία και δηλώνει μία νέα μεταβλητή η οποία διαιρεί το πλήθος των αριθμών του διανύσματος με τον αριθμό των επεξεργασιών.

Χρησιμοποιεί λοιπόν την `num` και την `k` για να στείλει το διάνυσμα σε όλες τις διεργασίες. Η 0 διεργασία αποθηκεύει τώρα τους αριθμούς που έχει σε ένα νέο array `data_loc`. Όταν έρθει και η σειρά των υπόλοιπων διεργασιών βάζουν και αυτές στο αντίστοιχο array τους αριθμούς.

```
if (my_rank == 0)
{
    printf("Give amount of numbers:\n");
    scanf("%d", &plithos);
    printf("Give %d numbers:\n", plithos);
    for (k = 0; k < plithos; k++)
        scanf("%d", &data[k]);
    for (target = 1; target < p; target++)
        MPI_Send(&plithos, 1, MPI_INT, target, tag2, MPI_COMM_WORLD);
    num = plithos / p;
    k = num;
    for (target = 1; target < p; target++)
    {
        MPI_Send(&data[k], num, MPI_INT, target, tag3, MPI_COMM_WORLD);
        k += num;
    }
    for (k = 0; k < num; k++)
        data_loc[k] = data[k];
}
else
{
    MPI_Recv(&plithos, 1, MPI_INT, 0, tag2, MPI_COMM_WORLD, &status);
    num = plithos / p;
    MPI_Recv(&data_loc[0], num, MPI_INT, 0, tag3, MPI_COMM_WORLD, &status);
}
```

```
Type 1 to continue or anything else to exit!
1
Give amount of numbers:
6
Give 6 numbers:
1
2
3
4
5
6
```

Εικόνα 4: Ορισμός Διανύσματος.

Στο παρακάτω στιγμιότυπο οθόνης εμφανίζεται ο κώδικας που χρησιμοποιείται για το μ και το μέγιστο αριθμό του διανύσματος.

```
/*About M and Max*/

m = 0;
max = 0;
for (k = 0; k < num; k++)
{
    m = m + data_loc[k];
    if (data_loc[k] > max)
    {
        max = data_loc[k];
    }
}

/*Sending M and Max to (0) process.*/

if (my_rank != 0)
{
    MPI_Send(&m, 1, MPI_INT, 0, tag4, MPI_COMM_WORLD);
    MPI_Send(&max, 1, MPI_INT, 0, tag4, MPI_COMM_WORLD);
}
else
{
    /*When in 0 process, receiving M and Max. After the calculations,
    we get the final results.*/

    finres = m;
    printf("\n Result of process %d: %d\n", my_rank, m);
    for (source = 1; source < p; source++)
    {
        MPI_Recv(&m, 1, MPI_INT, source, tag4, MPI_COMM_WORLD, &status);
        MPI_Recv(&max, 1, MPI_INT, source, tag4, MPI_COMM_WORLD, &status);
        finres = finres + m;
        if (finresmax < max)
        {
            finresmax = max;
        }
        printf("\n Result of process %d for m: %d\n", source, m);
        printf("\n Result of process %d for Max: %d\n", source, max);
    }
    finresmdiv = finres / (float)plithos;
    printf("\n\n** a) M: %f **\n", finresmdiv);
    printf("\n\n** b) Max number: %d **\n\n", finresmax);
}
```

Result of process 0: 3
Result of process 1 for m: 7
Result of process 1 for Max: 4
Result of process 2 for m: 11
Result of process 2 for Max: 6

$$\mu = (x_0 + x_1 + x_2 + \dots + x_{n-1}) / n$$

** a) M: 3.500000 **
** b) Max number: 6 **

Εικόνα 5: M and Max.

Στο παρακάτω στιγμιότυπο οθόνης εμφανίζεται ο κώδικας που χρησιμοποιείται για τον υπολογισμό του Var (Διαβάστε τα σχόλια του κώδικα).

```
/*About Var.
When in process 0, send M to all other process.*/

if (my_rank == 0)
{
    for (target = 1; target < p; target++)
    {
        MPI_Send(&finresmdiv, 1, MPI_FLOAT, target, tag5, MPI_COMM_WORLD);
    }
}
else
{
    MPI_Recv(&finresmdiv, 1, MPI_FLOAT, 0, tag5, MPI_COMM_WORLD, &status);
}

/*Calculating Var with M.*/

var = 0;
for (k = 0; k < num; k++)
{
    var = var + ((data_loc[k] - finresmdiv) * (data_loc[k] - finresmdiv));
}

/*Sending Var to (0) process.*/

if (my_rank != 0)
{
    MPI_Send(&var, 1, MPI_FLOAT, 0, tag6, MPI_COMM_WORLD);
}
else
{
    /*When in 0 process, receiving Var. After the calculations,
    we get the final results.*/

    finresvar = var;
    printf("\n Result of process %d: for Var %f\n", my_rank, var);
    for (source = 1; source < p; source++)
    {
        MPI_Recv(&var, 1, MPI_FLOAT, source, tag6, MPI_COMM_WORLD, &status);
        finresvar = finresvar + var;
        printf("\n Result of process %d for Var: %f\n", source, var);
    }
    printf("\n\n** c) Var: %f **\n\n", finresvar / (float)plithos);
}
```

```
Result of process 0: for Var 8.500000
Result of process 1 for Var: 0.500000
Result of process 2 for Var: 8.500000
```

```
** c) Var: 2.916667 **
```

$$var = ((x_0 - \mu)^2 + (x_1 - \mu)^2 + (x_2 - \mu)^2 + \dots + (x_{n-1} - \mu)^2) / n$$

Εικόνα 6: Var.

```
if (my_rank == 0)
{
    for (target = 1; target < p; target++)
    {
        MPI_Send(&finresmax, 1, MPI_INT, target, tag7, MPI_COMM_WORLD);
    }
}
else
{
    MPI_Recv(&finresmax, 1, MPI_INT, 0, tag7, MPI_COMM_WORLD, &status);
}

for (k = 0; k < num; k++)
{
    d[k] = abs((data_loc[k] - finresmax) * (data_loc[k] - finresmax));
}

if (my_rank != 0)
{
    MPI_Send(&d, num, MPI_INT, 0, tag8, MPI_COMM_WORLD);
}
else
{
    for (k = 0; k < num; k++)
    {
        lastd[k] = d[k];
    }
    i = k;
    for (source = 1; source < p; source++)
    {
        MPI_Recv(&lastd[i], num, MPI_INT, source, tag8, MPI_COMM_WORLD, &status);
        i += num;
    }
    for (k = 0; k < plithos; k++)
    {
        printf("\n** d) D: %d **\n", lastd[k]);
    }
}

MPI_Finalize();
return 0;
}
```

$$\delta_i = |x_i - m|^2$$

```
** d) D: 25 **
** d) D: 16 **
** d) D: 9 **
** d) D: 4 **
** d) D: 1 **
** d) D: 0 **
Type 1 to continue or anything else to exit!
```

```
        finresvar = var;
        printf("\n Result of\n");
        for (source = 1; source < p; source++)
        {
            MPI_Recv(&var, 1, MPI_INT, source, tag7, MPI_COMM_WORLD, &status);
            finresvar = finresvar + var;
        }
        finresvar = finresvar / (p - 1);
        MPI_Send(&finresvar, 1, MPI_INT, 0, tag7, MPI_COMM_WORLD);
    }
}
```

Εικόνα 7: D.

Στην εικόνα 7, γίνεται ο υπολογισμός του τελευταίου ερωτήματος. Για αρχή, από την διεργασία 0, στέλνει το μέγιστο αριθμό του διανύσματος σε κάθε

διεργασία. Κάθε διεργασία το λαμβάνει. Στην συνέχεια, γίνονται οι υπολογισμοί του ερωτήματος (**abs()** η συνάρτηση για την απόλυτη τιμή). Κάθε διεργασία στέλνει στην 0 το/τα d που υπολόγισε. Η 0 βάζει αυτά που υπολόγισε η ίδια και ότι λαμβάνει στο **lastd** array. Έπειτα τα εμφανίζει.

```
invalid@invalid:/media/sf_OneDrive_-_University_of_West_Attica/Εισαγωγή στο παράλληλο υπολογισμό$ mpiexec -n 3 ./thomasex
Type 1 to continue or anything else to exit! Διατάξη Αναφορές Στοιχεία αλληλογραφίας Αναθεώρηση Πρόβολη Βοήθεια
0
invalid@invalid:/media/sf_OneDrive_-_University_of_West_Attica/Εισαγωγή στο παράλληλο υπολογισμό$
```

Εικόνα 8: Έξοδος (menu επιλογών).