


Παράλληλα Συστήματα

Χειμερινό εξάμηνο 2024–2025

#1

Παράλληλα Συστήματα


► Προαπαιτούμενα

- Γλώσσα C
 - Linux
 - Μαθήματα
- 


Παράλληλα Συστήματα

- ▶ Ύλη Α.Π. / εργαστηρίου
 - OpenMP και CUDA
- ▶ Πλάνο μαθημάτων


Παράλληλα Συστήματα

- ▶ Δομή μαθημάτων
 - ▶ Χώρος / εργαλεία εργασίας
 - ▶ Βαθμολογία
- 

Παράλληλα Συστήματα

- ▶ Υποστηρικτικό υλικό
 - ▶ Επικοινωνία
 - ▶ Απορίες
- 

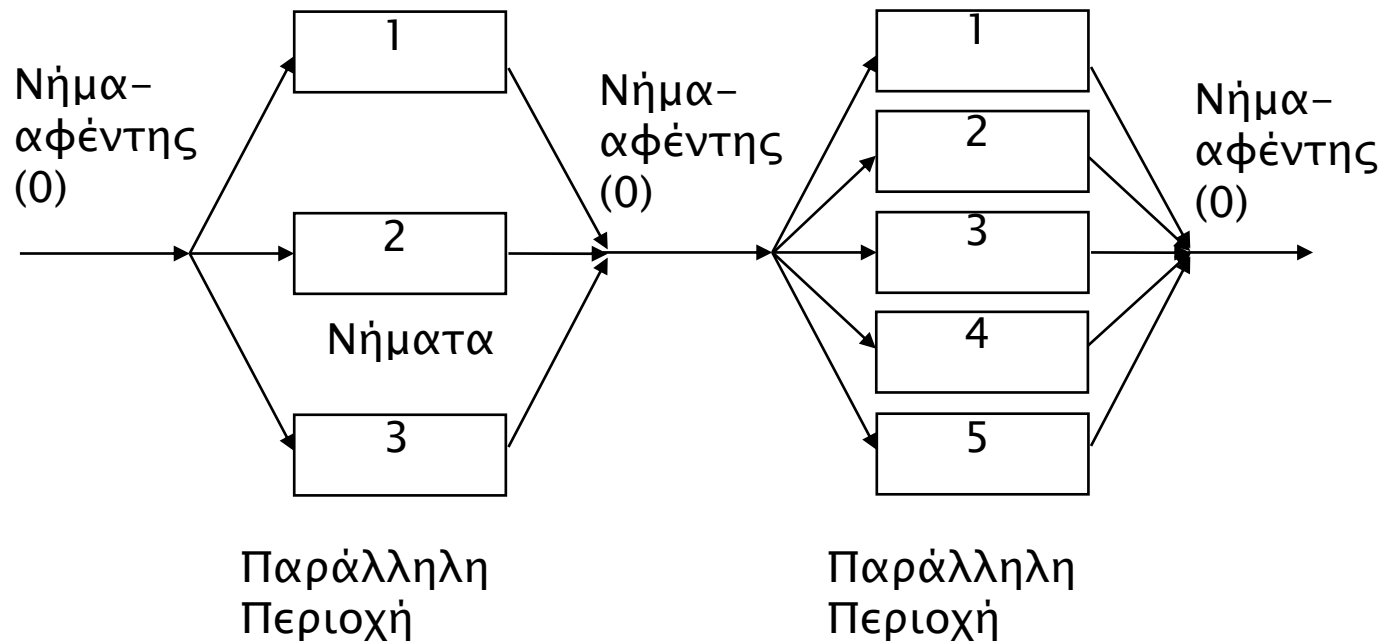
OpenMP – Εισαγωγή

- ▶ Τι είναι;
 - ▶ Εκδόσεις
 - ▶ Γλώσσες
- 

OpenMP – Πρότυπο / Μοντέλο

- ▶ Πολυνηματικό (multithreaded) πρότυπο παράλληλου προγραμματισμού
- ▶ Νήμα-αφέντης (master thread)
- ▶ Σειριακή εκτέλεση => Δήλωση παράλληλης περιοχής
- ▶ Εκκίνηση νημάτων από το νήμα-αφέντης

Μοντέλο fork-join



Το μοντέλο εκτέλεσης *fork-join* του OpenMP

Διαταγές μεταγλωττιστή

`#pragma omp directive [clause ...] new-line`

`#pragma omp parallel [clause ...] new-line`

`structured-block`



Βασικό Παράδειγμα Εκτέλεσης

```
#include <omp.h>
#include <stdio.h>
int main(int argc, char **argv)
{
    #pragma omp parallel
    {
        printf("Hello OpenMP!\n");
    }
}
```

Αν υποθέσουμε ότι βρισκόμαστε σε ένα υπολογιστικό σύστημα με δύο (2) επεξεργαστές/πυρήνες:


Hello OpenMP!
Hello OpenMP!

```
gcc -o program program.c -fopenmp
export OMP_NUM_THREADS=2
./program
```

Παράδειγμα omp2.c

```
#include <omp.h>
#include <stdio.h>
```


```
main () {
    int nthreads, tid;
    omp_set_num_threads(4);
    #pragma omp parallel private(tid) {
        tid = omp_get_thread_num();
        printf("Hello World from thread %d\n", tid);
        if (tid == 0) {
            nthreads = omp_get_num_threads();
            printf("Number of threads = %d\n", nthreads);
        }
    }
}
```



Επιπλέον δυνατότητες δήλωσης

- ▶ `if (...)`
- ▶ `firstprivate (...)`
 - Διαφορά με `private` = ?
- ▶ `num_threads (...)`
- ▶ `default (shared | none)`

Βασικές διαταγές διαμοιρασμού εργασίας

- ▶ Διαταγή 'for'
 - ▶ Διαταγή 'sections'
 - ▶ Διαταγή 'single'
- 

Διαταγή 'for'

- ▶ Αυτόματη παραλληλοποίηση ενός συνόλου επαναλήψεων του προγράμματός μας οι οποίες καθορίζονται μέσα σε μία δομή for της C/C++.

```
#pragma omp for [clause ...] new-line  
for-loop
```

Παράδειγμα omp3.c

- ▶ Άθροισμα δύο διανυσμάτων μήκους N
 - Σειριακή εκδοχή
 - Παράλληλη εκδοχή

Παράδειγμα omp3.c


- ▶ .. parallel shared (a, b, c, chunk) private(i, tid)
- ▶ ... for schedule (static, chunk)
 - static / dynamic / ...
 - chunk size
- ▶ Δοκιμές!

Παράμετροι σε parallel 'for'

- ▶ private (list)
 - ▶ shared (list)
 - ▶ firstprivate (list)
 - ▶ lastprivate (list)
 - ▶ reduction
 - ▶ collapse (n)
 - ▶ ordered
 - ▶ nowait
- 


Συνδυασμός

- ▶ `#pragma omp parallel ...`
 - ▶ `#pragma omp for ...`

 - ▶ `#pragma omp parallel for ...`
- 

Διαταγή 'sections'

```
#pragma omp sections [clause ...] new-line  
{  
    #pragma omp section new-line  
        structured-block  
    #pragma omp section new-line  
        structured-block  
    ...  
}
```




Παράδειγμα omp5.c

- ▶ Έστω ότι μας ζητείται να υπολογίσουμε παράλληλα τα ακόλουθα δύο αθροίσματα:
 - (α) το άθροισμα δύο διανυσμάτων a και b (μήκους N), και
 - (β) το άθροισμα δύο άλλων διανυσμάτων e και f (επίσης μήκους N)
- ▶ Παράμετροι σε διαταγή 'section'

Διαταγή 'single'

#pragma omp single [clause ...] new-line
structured-block

```
#pragma omp parallel
{
    Do_Job1();
    #pragma omp single
    {
        Do_Job2();
    }
    Do_Job3();
}
```



Πρόγραμμα ask.c

- ▶ Να γραφτεί ένα πρόγραμμα με χρήση OpenMP που να πολλαπλασιάζει 3 διανύσματα $d1$, $d2$ και $d3$, όλα μήκους M
 - Σειριακή εκδοχή
 - Παράλληλη εκδοχή