



# ARHITECTURI PARALELE

## Tema #1 Termodinamica fluidelor

Termen de predare: 27-Nov-2022 23:55

### Obiective

Scopul acestei teme este de a implementa, în C, folosind thread-uri prin intermediul bibliotecii Pthread, pe Linux, un simulator **scalabil cu numărul de thread-uri**, ce va determina, pe un model simplificat, **cum se va desfășura dinamica temperaturii fluidelor în cadrul unei suprafețe**.

### Date introductive

Comportamentul fluidelor din punct de vedere al temperaturii necesită cunoașterea unui domeniu complex, termodinamica, ce necesită o vastă expertiză, atât teoretică dar și experimentală.

Cu toate acestea, având în vedere cantitatea redusă de timp pe care o avem la dispoziție, dar și scopul de a ne îmbunătăți abilitățile de paralelizare a algoritmilor, divergent de cel al desăvârșirii cunoașterii în profunzime a termodinamicii fluidelor, ne vom permite să ometem mai mulți factori ce ar putea să influențeze dinamica temperaturii, de pildă, temperatura aerului din jur, accelerația gravitațională, ș.a.m.d.. Astfel, vom folosi un model extrem de simplificat, în care definim o suprafața lichidă (de ex.: apă) ce va fi ilustrată sub forma unei matrice. Fiecare celulă a matricei va reprezenta o unitate de suprafață fluidă sau una compusă dintr-un material perfect izolat (care nu permite transferul de temperatură).

Pentru a simula transferul de căldură, la fiecare pas de timp căldura se va transmite de la o unitate de suprafață la unitățile vecine (căsuță vecină în matrice). După un număr dat de unități de timp vom putea observa o uniformizare a temperaturii întregii suprafețe.

### Detalii simulator

Suprafața va fi reprezentată printr-o matrice bi sau tri-dimnesională. În cazul matricei bidimensionale fiecare individ va avea 8 vecini. În cazul celei tridimensionale fiecare individ va avea 26 de vecini. Excepție fac unitățile de pe marginea sau din colțurile matricei. Acestea vor avea vecini lipsă. Tipul și dimensiunile matricei vor fi date în fișierul de intrare.

Pe lângă dimensiunile matricei, în fișierul de intrare vom avea tipul fiecărei unități de suprafață (fluid / non-fluid) și temperatura (pentru tipul fluid, o valoare ce va putea fi stocată într-o variabilă de tip *double* / pentru tipul non-fluid, valoarea 0).

Simularea va rula pentru un număr de unități de timp, dat în fișierul de intrare. Pentru fiecare unitate de timp se va realiza transferul de căldură între celulele fluide. Astfel, temperatura celulei curente va lua valoarea mediei temperaturii celulei și a vecinilor. La finalul simulării, starea suprafeței va fi salvată într-un fișier de ieșire.



Format fișier intrare matrice 2D:

```
1 DIM_MATRICE X Y
2 TIP_ZONA_X1_Y1 TEMP_ZONA_X1_Y1
3 TIP_ZONA_X2_Y1 TEMP_ZONA_X2_Y1
4 TIP_ZONA_X3_Y1 TEMP_ZONA_X3_Y1
5 ...
6 TIP_ZONA_Xn_Y1 TEMP_ZONA_Xn_Y1
7 TIP_ZONA_X1_Y2 TEMP_ZONA_X1_Y2
8 ...
9 TIP_ZONA_X1_Yn TEMP_ZONA_X1_Yn
10 TIP_ZONA_X2_Yn TEMP_ZONA_X2_Yn
11 TIP_ZONA_X3_Yn TEMP_ZONA_X3_Yn
12 ...
13 TIP_ZONA_Xn_Yn TEMP_ZONA_Xn_Yn
14 TIMP_RULARE
```

Format fișier intrare matrice 3D:

```
1 DIM_MATRICE X Y Z
2 TIP_ZONA_X1_Y1_Z1 TEMP_ZONA_X1_Y1_Z1
3 TIP_ZONA_X2_Y1_Z1 TEMP_ZONA_X2_Y1_Z1
4 TIP_ZONA_X3_Y1_Z1 TEMP_ZONA_X3_Y1_Z1
5 ...
6 TIP_ZONA_Xn_Y1_Z1 TEMP_ZONA_Xn_Y1_Z1
7 TIP_ZONA_X1_Y2_Z1 TEMP_ZONA_X1_Y2_Z1
8 ...
9 TIP_ZONA_Xn_Yn_Z1 TEMP_ZONA_Xn_Yn_Z1
10 TIP_ZONA_X1_Y1_Z2 TEMP_ZONA_X1_Y1_Z2
11 TIP_ZONA_X2_Y1_Z2 TEMP_ZONA_X2_Y1_Z2
12 ...
13 TIP_ZONA_X1_Y1_Zn TEMP_ZONA_X1_Y1_Zn
14 TIP_ZONA_X2_Y1_Zn TEMP_ZONA_X2_Y1_Zn
15 TIP_ZONA_X3_Y1_Zn TEMP_ZONA_X3_Y1_Zn
16 ...
17 TIP_ZONA_Xn_Yn_Zn TEMP_ZONA_Xn_Yn_Zn
18 TIMP_RULARE
```

Unde:

- **DIM\_MATRICE** - poate avea valoarea 2 (matrice bidimensională) sau 3 (matrice tridimensională);
- **X, Y, Z** - dimensiunea pe axa o(X,Y,Z) a matricei; Z va fi în fișier doar pentru matrici tridimensionale;
- **TIP\_ZONA\_Xg\_Yr\_Zc** - tipul zonei cu coordonatele X=g, Y=r, Z=c;
- **TEMP\_ZONA\_Xg\_Yr\_Zc** - temperatura zonei cu coordonatele X=g, Y=r, Z=c;
- **TIMP\_RULARE** - numărul de unități de timp (iterații);

Fișierul de ieșire va avea același format cu fișierul de intrare exceptând ultima linie (TIMP\_RULARE) care va lipsi.



## Observații

Dacă unul din puncte nu este implementat avem rugămintea ca programul să returneze instant. Astfel timpul de testare al temelor incomplete va scădea considerabil și colegii nu vor trebui să aștepte degeaba.

În fișierele demo din schelet ce conțin exemple de matrice generate, fiecare tip de zonă este asociat cu un caracter: f - zonă fluidă, n - zonă non-fluidă.

## Rularea programului

Rularea programului se va realiza astfel:

```
1 ./homework INPUT_FILE OUTPUT_FILE NUM_THREADS
```

Unde:

- **INPUT\_FILE** - fișierul cu parametrii de intrare;
- **OUTPUT\_FILE** - fișierul cu date de ieșire;
- **NUM\_THREADS** - numărul de thread-uri cu care programul să ruleze.

Exemplu rulare:

```
1 ./homework input.txt output.txt 2
```

## Trimitere și punctare

Distribuția punctajului este următoarea:

- 50 puncte - Output corect program paralelizat și scalabil cu matrice bidimensională și mărimi divizibile la numărul de thread-uri;
- 10 puncte - Output corect program paralelizat și scalabil cu matrice bidimensională și mărimi nedivizibile la numărul de thread-uri;
- 30 puncte - Output corect program distribuit și scalabil cu matrice tridimensională și mărimi divizibile la numărul de thread-uri;
- 10 puncte - Output corect program paralelizat și scalabil cu matrice tridimensională și mărimi nedivizibile la numărul de thread-uri;

Arhiva .zip trebuie să conțină fișierul "homework.c". O temă care nu compilează va primi 0 puncte. Tema va fi rulată și testată pentru corectitudine și scalabilitate pe un checker. Link-ul către acest checker va fi transmis ulterior.

**Orice încercare de a abuza checker-ul va duce la un punctaj de 0 pe toate temele.**