

ALGORITMI PARALELI

Tema #1 Seeds

Termen de predare: 06-Dec-2024 23:55

Obiective

Scopul acestei teme este de a implementa, în C, folosind thread-uri prin intermediul bibliotecii Pthread, pe Linux, un program **scalabil cu numărul de thread-uri**, ce va simula automatul celular [Seeds](#).

Date introductive

Simulatorul presupune existența unei hărți sub forma unei matrice bidimensionale sau tridimensionale, fiecare element având una din următoarele două stări:

- **ON**
- **OFF**

Simularea va fi executată pentru un număr finit de etape (pași). Astfel, regulile de modificare a stării celulelor la fiecare pas sunt următoarele:

- O celulă trece din starea **OFF** în starea **ON** dacă are exact doi vecini aflați în starea **ON**;
- O celulă aflată în starea **ON** intră implicit în starea **OFF**.

În cazul matricei bidimensionale, o celulă va avea tot timpul 8 vecini. Pentru cazul tridimensional, o celulă va avea 26 de vecini. Această regulă se aplică și în cazul celulelor aflate pe marginile suprafeței de simulare, cu mențiunea că vecinii ce depășesc granița matricei se vor afla tot timpul în starea **OFF**.

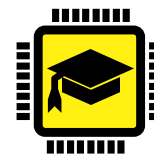
Procesul de simulare

Simularea este constituită dintr-un număr fix de etape, starea celulelor schimbându-se 'simultan' în funcție de celulele din etapa anterioară.

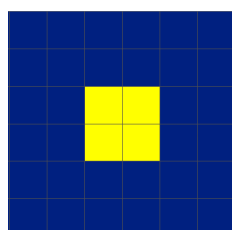
Programul va primi ca input un fișier în care se vor regăsi informațiile legate de suprafața simulării și stările inițiale pentru toate celulele.

Pe prima linie vom avea:

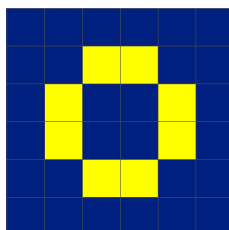
- Un număr întreg **D** ce indică numărul de dimensiuni ale matricei. 2 pentru matrice bidimensională, 3 pentru matrice tridimensională;
- Un număr întreg **L** ce indică lungimea hărții (există doar pentru cazul 3D);
- Un număr întreg **H** ce indică înălțimea hărții;
- Un număr întreg **W** ce indică lățimea hărții;
- Un număr întreg **S** ce indică numărul total de pași ce vor fi efectuați în cadrul simulării. Acesta există doar în cazul fișierului de intrare.



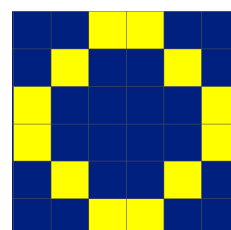
Exemplu simulare pe o hartă de dimensiune 6 x 6



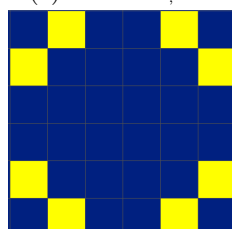
(a) Harta inițială



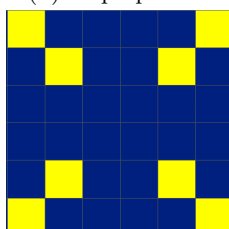
(b) După pasul 1



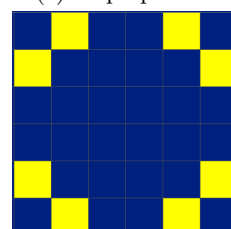
(c) După pasul 2



(d) După pasul 3



(e) După pasul 4



(f) După pasul 5

Apoi vor urma $H \times W$ valori (sau $L \times H \times W$ pentru varianta 3D), fiecare element având o valoare numerică reprezentată de starea celulei, unde:

- Valoarea 0 reprezintă starea OFF;
- Valoarea 1 reprezintă starea ON.

Format fișier intrare matrice 2D:

```
1 2 H W S
2 STARE_INITIALA_CELULA_Y1_X1 STARE_INITIALA_CELULA_Y1_X2 ... STARE_INITIALA_CELULA_Y1Xn
  STARE_INITIALA_CELULA_Y2X1 ... STARE_INITIALA_CELULA_YmXn
```

Format fișier intrare matrice 3D:

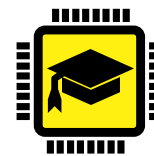
```
1 3 L H W S
2 STARE_INITIALA_CELULA_Z1_Y1_X1 STARE_INITIALA_CELULA_Z1_Y1_X2 ...
  STARE_INITIALA_CELULA_Z1_Y1_Xn STARE_INITIALA_CELULA_Z1_Y2_X1 ...
  STARE_INITIALA_CELULA_Z1_Ym_Xn STARE_INITIALA_CELULA_Z2_Y1_X1 ...
  STARE_INITIALA_CELULA_Zk_Ym_Xn
```

Format fișier ieșire matrice 2D:

```
1 2 H W
2 STARE_INITIALA_CELULA_Y1_X1 STARE_INITIALA_CELULA_Y1_X2 ... STARE_INITIALA_CELULA_Y1Xn
  STARE_INITIALA_CELULA_Y2X1 ... STARE_INITIALA_CELULA_YmXn
```

Format fișier ieșire matrice 3D:

```
1 3 L H W
2 STARE_FINALA_CELULA_Z1_Y1_X1 STARE_FINALA_CELULA_Z1_Y1_X2 ...
  STARE_FINALA_CELULA_Z1_Y1_Xn STARE_FINALA_CELULA_Z1_Y2_X1 ...
  STARE_FINALA_CELULA_Z1_Ym_Xn STARE_FINALA_CELULA_Z2_Y1_X1 ...
  STARE_FINALA_CELULA_Zk_Ym_Xn
```



Unde:

- **L, H, W** - dimensiunile matricei;
- **STARE_CELULA_Xg_Yr_Xc** - starea celulei cu coordonatele $Z=g$, $Y=r$, $X=c$;
- **S** - numărul de unități de timp (iterații/pași) ale simulării.

Rularea programului

Rularea programului se va realiza astfel:

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

Unde:

- **INPUT_FILE** - fișierul cu date 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.

- **Va fi luată în considerare ultima încărcare înainte de termenul limită!**
- **Orice încercare de a abuza checker-ul va duce la un punctaj de 0 pe toate temele!**