

SISTEME TOLERANTE LA DEFECTE

Tema #1 Game of Life

Termen de predare: 17-Apr-2022 23:55

Obiective

Scopul acestei teme este de a implementa, în C, folosind biblioteca MPI, un program distribuit, scalabil cu numărul de procese, având ca scop simularea jocului Game of Life - B3/S23 introdus de John Conway.

Date introductive

Jocul presupune existența unei hărți sub forma unei matrici de dimensiune $H \times W$. Fiecare element având una din următoarele două valori:

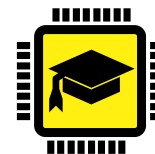
- **Individ**, valoarea **1**
- **Gol**, valoarea **0**

Simularea va fi executată pentru un număr dat de etape. În fiecare etapă elementele din matrice își schimbă valoarea după setul de reguli B3/S23:

- (a) Un individ nou este creat dacă are 3 vecini;
- (b) Un individ continuă să existe dacă are 2 sau 3 vecini;
- (c) Orice individ cu 1, 0 sau mai mult de 3 vecini dispare.

Vecinii unui element sunt cele 8 căsuțe reprezentate în tabelul 1 cu verde.

Table 1: Exemplu hartă, vecini



Reprezentare hartă

Există două metode de a reprezenta harta:

- (a) Ca un plan, marginile vor fi alcătuite (tot timpul) din elemente goale;
- (b) Ca un toroid: astfel, prima linie/coloană este lipită de ultima. Deci, zona simulată nu va conține margini.

Table 2: Plan

0	0	0	0	0	0
0	A	B	C	D	0
0	E	F	G	H	0
0	I	J	K	L	0
0	0	0	0	0	0

Table 3: Toroid

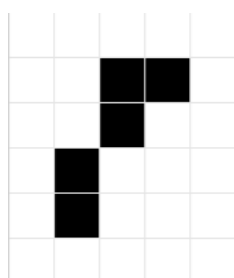
L	I	J	K	L	I
D	A	B	C	D	A
H	E	F	G	H	E
L	I	J	K	L	I
D	A	B	C	D	A

Procesul de simulare

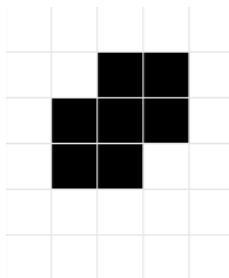
Simularea va fi constituită dintr-un număr de etape, starea celulelor schimbându-se simultan în funcție de celulele din etapa anterioară.

Programul va primi ca input un fișier și un număr de pași care va reprezenta numărul de etape ale simulării, iar rezultatul final va fi scris într-un fișier de ieșire.

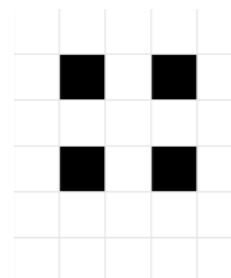
Exemplu simulare pe o hartă de dimensiune 6 x 5



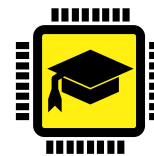
(a) Harta inițială



(b) După pasul 1



(c) După pasul 2



Rularea programului

Rularea programului se va realiza astfel:

```
1 mpirun --oversubscribe -np NUM_PROCS ./homework IN_FILENAME OUT_FILENAME NUM_STEPS
```

Unde:

- **NUM_PROCS** - numărul de procese;
- **IN_FILENAME** - numele (path-ul) fișierului de intrare;
- **OUT_FILENAME** - numele (path-ul) fișierului de ieșire;
- **NUM_STEPS** - numărul de pași ai simulării.

Exemplu:

```
1 mpirun --oversubscribe -np 2 ./homework in1.txt out1.txt 567
```

Fișiere de intrare/ieșire

Pe prima linie vom avea:

- O literă P (plan) sau T (toroid) ce indică tipul hărții;
- Un număr întreg H ce indică lățimea hărții;
- Un număr întreg W ce indică lungimea hărții;
- H linii cu W valori, pentru fiecare celulă din hartă (1 - individ, 0 - gol).

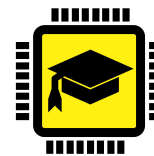
Exemplu:

Table 4: Exemplu fișier de intrare

P	5	4	
0	1	1	0
0	1	0	0
1	0	0	0
1	0	0	0
0	0	0	0

Table 5: Exemplu fișier de ieșire

P	5	4	
0	1	1	0
1	1	1	0
1	1	0	0
0	0	0	0
0	0	0	0



Trimitere și punctare

- Este necesară crearea unei arhive **.zip** care trebuie să conțină fișierul “homework.c” și uploadarea acesteia în cadrul checker-ului.
- O temă care nu compilează va primi 0 puncte. Tema va fi rulată și testată pentru corectitudine și scalabilitate pe un checker al ATM.
- 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.

Distribuția punctajului este următoarea:

- 70 puncte - Output corect program distribuit și scalabil pentru hărți de tipul PLAN.
- 30 puncte - Output corect program distribuit și scalabil pentru hărți de tipul TOROID.

Arhiva .zip trebuie să conțină fișierele “homework.c”. O temă care nu compilează va primi 0 puncte. Tema va fi rulată și testată pentru corectitudine și scalabilitate pe un checker al ATM. 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.