

Se consideră un fișier numbers.txt care conține N numere întregi (câte un număr pe fiecare linie). Scrieți un program MPI care citește un număr natural X și pornește P procese care prelucrează numerele în mod distribuit utilizând un mecanism de transmitere în cascadă între procese, asigurând paralelism.

h(numar,X):

- Dacă $\text{numar} < X$, se adaugă X la număr (cazul 1).
- Dacă $\text{numar} \geq X$, se împart valorile la 2 (cazul 2).

Cerințe

Etapa 1:

1. Procesul 0 citește valoarea X și toate numerele din fișierul numbers.txt, stocându-le într-un vector numbers.
2. Procesul 0 împarte vectorul numbers în P segmente egale (sau aproximativ egale) și transmite celorlalte procese.

Etapa 2:

1. Fiecare proces primește lista de numere de la procesul 0.
2. În paralel:
 - a. Fiecare proces prelucrează numerele conform funcției h(numar,X) și calculează suma numerelor originale într-o variabilă **suma**.
 - b. Transmite numerele prelucrate înapoi la procesul cu Rank = 0
 - c. Așteaptă de la procesul cu Rank – 1 suma numerelor calculate de acesta (procesul cu rank 0 nu așteaptă, și inițiază transmiterea cu valoarea 0).
 - d. Adună suma primită, la suma calculată de proces.
 - e. Transmite noua suma către procesul cu Rank + 1
 - f. Ultimul proces din cascadă (ID = P–1) returnează suma calculată înapoi către procesul 0.

Etapa 3:

1. Procesul 0 primește toate segmentele procesate, reconstruiește vectorul complet și scrie rezultatul în fișierul result.txt.
2. Procesul 0 afișează pe ecran suma primită de la procesul cu ID=P-1

