Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Кафедра цифрових технологій в енергетиці

ЗВІТ   
з виконання лабораторної роботи №3  
з дисципліни «Технології паралельних обчислень в енергетичних комплексах

»

«Паралельні обчислення для мультикомп’ютерів на основі технології МРІ»

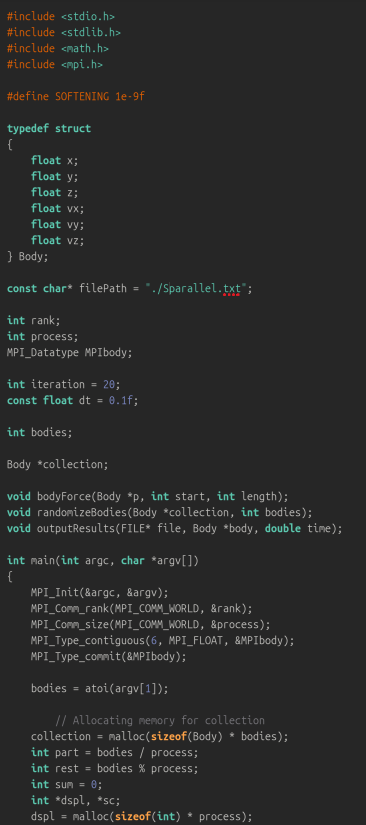
Варіант 21

Виконав: студент групи ТР-12 Руденко В.І

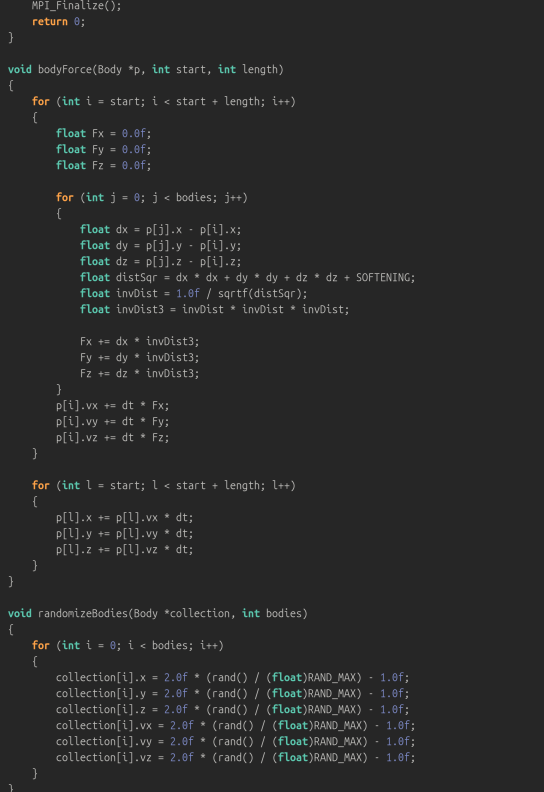
Завдання:

Розробити програмну реалізацію задачі N тіл в середовищі MPI.

Хід Роботи:





\

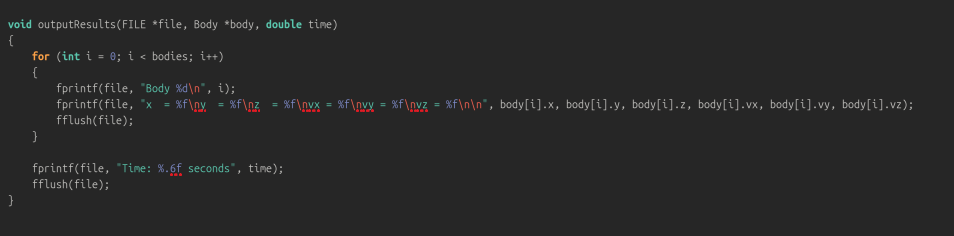


Table 1 Код програми Завдання 1

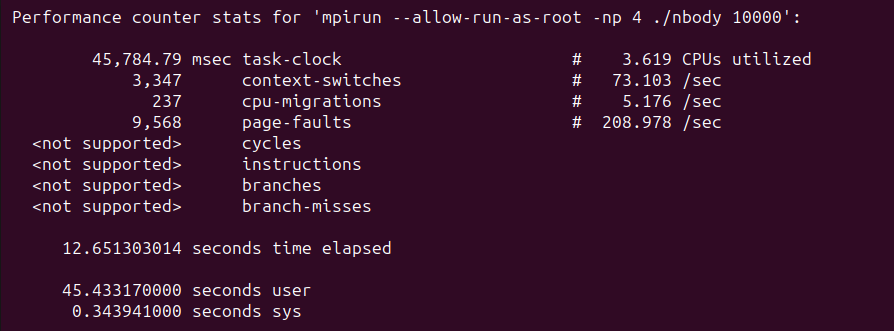


Table 2 Виконання завдання в однопоточному режимі

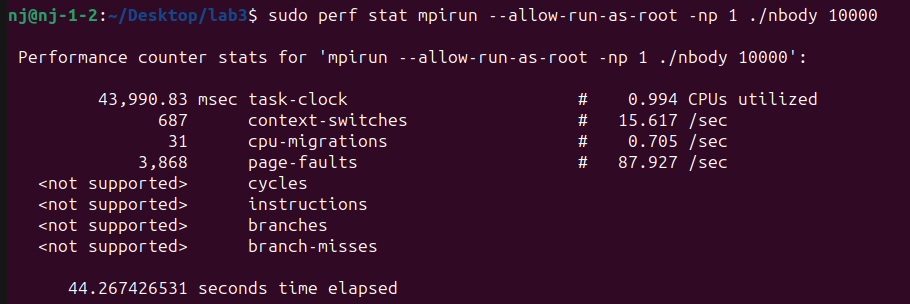


Table 4 Виконання Завдання а багатопоточному режимі

**Контрольні запитання**

1. Що слід розуміти під паралельною програмою?

Паралельна програма – це програма, яка виконується одночасно на декількох процесорах або ядрах, з метою прискорення обчислень. У такій програмі завдання розподіляються між кількома процесами або потоками, які працюють одночасно і взаємодіють між собою для обміну даними та синхронізації.

2. Що розуміють в MPI під комунікатором?

В MPI (Message Passing Interface) комунікатор – це об’єкт, який визначає групу процесів, що можуть обмінюватися повідомленнями між собою. Комунікатори дозволяють обмежити обмін повідомленнями до певної підмножини процесів в програмі.

3. Як можна організувати приймання повідомлень від конкретних процесів?

Для приймання повідомлень від конкретних процесів в MPI використовуються функції, такі як MPI\_Recv, яка має параметр source для вказівки ідентифікатора процесу, від якого очікується повідомлення. Це дозволяє організувати приймання повідомлень від конкретних відправників.

4. Як визначити час виконання MPI-програми?

Для визначення часу виконання MPI-програми використовують функції MPI\_Wtime, яка повертає поточний час у секундах. Для вимірювання часу виконання певної частини програми слід зафіксувати час на початку і в кінці цієї частини, а потім обчислити різницю між отриманими значеннями.

5. У чому відмінність парних і колективних операцій передачі даних?

Парні операції (point-to-point operations): передача даних від одного конкретного процесу до іншого конкретного процесу (наприклад, MPI\_Send та MPI\_Recv).

Колективні операції (collective operations): передача даних між усіма процесами в комунікаторі або групі процесів (наприклад, MPI\_Bcast, MPI\_Allgather).

6. Яка функція MPI забезпечує передачу даних від одного процесу всім процесам?

Функція MPI\_Bcast забезпечує передачу даних від одного процесу (кореневого процесу) всім іншим процесам у групі.

7. Що розуміють під операцією редукції?

Операція редукції (reduction) в MPI – це колективна операція, яка об’єднує дані з усіх процесів групи за допомогою певної операції (наприклад, суми, мінімуму, максимуму) і повертає результат одному процесу або всім процесам. Основна функція для редукції – MPI\_Reduce.

8. У яких ситуаціях слід застосовувати бар'єрну синхронізацію?

Бар’єрну синхронізацію слід застосовувати, коли необхідно гарантувати, що всі процеси досягли певної точки в програмі перед тим, як продовжити виконання. Це корисно для забезпечення синхронного переходу процесів до нового етапу обчислень. Функція для цього – MPI\_Barrier.

9. Які режими передачі даних підтримуються в MPI?

Стандартний режим (standard mode): базовий режим, де система може буферизувати повідомлення.

Синхронний режим (synchronous mode): передача повідомлення завершується тільки тоді, коли приймач підтвердить отримання.

Буферизований режим (buffered mode): відправник використовує попередньо виділений буфер для зберігання повідомлень.

Готовий режим (ready mode): відправник починає передачу лише тоді, коли гарантовано, що приймач готовий прийняти повідомлення.

10. Як організує неблокуючий обмін даними в MPI?

Для організації неблокуючого обміну даними в MPI використовуються неблокуючі операції, такі як MPI\_Isend і MPI\_Irecv. Вони повертають контроль програмі одразу після запуску операції, дозволяючи виконувати інші обчислення під час передачі даних. Для перевірки завершення неблокуючих операцій використовують MPI\_Wait або MPI\_Test.

11. Які колективні операції передачі даних передбачені в MPI?

MPI\_Allgather: Збирає дані від усіх членів групи та надсилає дані всім членам групи.

MPI\_Allgatherv: Збирає змінну кількість даних від кожного члена групи та надсилає дані всім членам групи.

MPI\_Allreduce: Поєднує значення з усіх процесів і розподіляє результат назад для всіх процесів.

MPI\_Alltoall: Збирає дані від усіх членів групи та розповсюджує їх.

MPI\_Barrier: Ініціює синхронізацію бар’єрів між усіма членами групи.

MPI\_Bcast: Передає дані від одного члена групи всім членам групи.

MPI\_Gather: Збирає дані від усіх членів групи до одного члена.

MPI\_Iallgather: Збирає дані від усіх членів групи та надсилає дані всім членам групи без блокування.

MPI\_Iallreduce: Поєднує значення з усіх процесів і розподіляє результат назад до всіх процесів без блокування.

MPI\_Ibarrier: Виконує бар’єрну синхронізацію між усіма членами групи без блокування.

MPI\_Ibcast: Транслює повідомлення від процесу з рангом «root» до всіх інших процесів комунікатора неблокуючим способом.

MPI\_Igather: Збирає дані від усіх членів групи до одного члена без блокування.

MPI\_Ireduce: Виконує глобальну операцію редукції (наприклад, суму, максимум або логічне і) для всіх членів групи без блокування.

MPI\_Iscatter: Розподіляє дані від одного члена між усіма членами групи неблокуючим способом. Ця функція виконує дію, протилежну операції, яка виконується функцією MPI\_Igather .

MPI\_Reduce: Виконує глобальну операцію редукції для всіх членів групи.

MPI\_Scatter: Розподіляє дані від одного члена між усіма членами групи.

MPI\_Exscan: Обчислює ексклюзивне сканування (часткове скорочення) даних для набору процесів.

MPI\_Op\_create: Створює дескриптор комбінованої функції, визначений користувачем.

MPI\_Op\_free: Звільняє дескриптор функції комбінації, визначений користувачем.

MPI\_Reduce\_local: Застосовує оператор скорочення до локальних аргументів.

MPI\_Reduce\_scatter: Поєднує значення та розподіляє результати.

MPI\_Scan: Обчислює сканування (часткове скорочення) даних у наборі процесів.

MPI\_User\_function є заповнювачем для назви функції, визначеної програмою.