

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА  
Факультет електроніки і комп'ютерних технологій

**Звіт**  
**Про виконання лабораторної роботи №8**  
**“Групи і комунікатори”**

Виконав:  
Ст. групи Фес-32  
Молібожко Олександр

Львів 2024

## Хід роботи:

У даному коді було реалізовано створення нової MPI комунікатора з групою процесів, які містять тільки процеси з рангами 1 та 3. Для цього було використано функції `MPI_Comm_group()` та `MPI_Group_incl()` для створення нової групи процесів, `MPI_Comm_create()` для створення нового комунікатора на основі створеної групи, а також функції `MPI_Comm_size()` та `MPI_Comm_rank()` для отримання розміру та рангу процесів в новому комунікаторі. Далі, було використано функцію `MPI_Bcast()` для розсилки повідомлення від процесу з рангом 0 до всіх процесів у новому комунікаторі.

## Код:

```
#include <mpi.h>
#include <iostream>
#include <cstring>

int main(int argc, char** argv) {
    int rank, size, newrank, newsize;
    char message[32] = "Привіт, MPI!";
    MPI_Comm newcomm;
    MPI_Group oldgroup, newgroup;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    MPI_Comm_group(MPI_COMM_WORLD, &oldgroup);

    int ranks[2] = {1, 3};
    MPI_Group_incl(oldgroup, 2, ranks, &newgroup);

    MPI_Comm_create(MPI_COMM_WORLD, newgroup, &newcomm);
    MPI_Comm_size(newcomm, &newsize);
    MPI_Comm_rank(newcomm, &newrank);

    if (newcomm != MPI_COMM_NULL) {
        if (newrank == 0) {
            std::cout << "Введіть повідомлення (макс. 32 символи): ";
            std::cin >> message;
        }

        MPI_Bcast(message, 32, MPI_CHAR, 0, newcomm);

        if (newcomm != MPI_COMM_NULL) {
            std::cout << "Новий комунікатор: процес " << rank << " з rank = " << newrank
                << " має доступ; розмір комунікатора: " << newsize << std::endl;
            std::cout << "Повідомлення: " << message << std::endl;
        }
    }

    if (rank == 0) {
        std::cout << "MPI_COMM_WORLD: процес " << rank << " з rank = " << newrank
            << "; розмір комунікатора: " << size << std::endl;
        std::cout << "Новий комунікатор: немає доступу " << std::endl;
    }

    MPI_Group_free(&newgroup);
    MPI_Group_free(&oldgroup);

    if (newcomm != MPI_COMM_NULL) {
        MPI_Comm_free(&newcomm);
    }

    MPI_Finalize();
    return 0;
}
```

## Результат:

```
Введіть повідомлення (макс. 32 символи): Новий комунікатор: процес 3 з rank = 1 має доступ; розмір комунікатора: 2
Повідомлення: Привіт, MPI!
Новий комунікатор: процес 1 з rank = 0 має доступ; розмір комунікатора: 2
Повідомлення: Привіт, MPI!
```

**Висновок:** У цій лабораторній роботі було реалізовано створення нового комунікатора, який об'єднує процеси з рангами 1 та 3, і організовано розсилку повідомлень всередині цієї групи процесів за допомогою функції `MPI_Bcast()`.