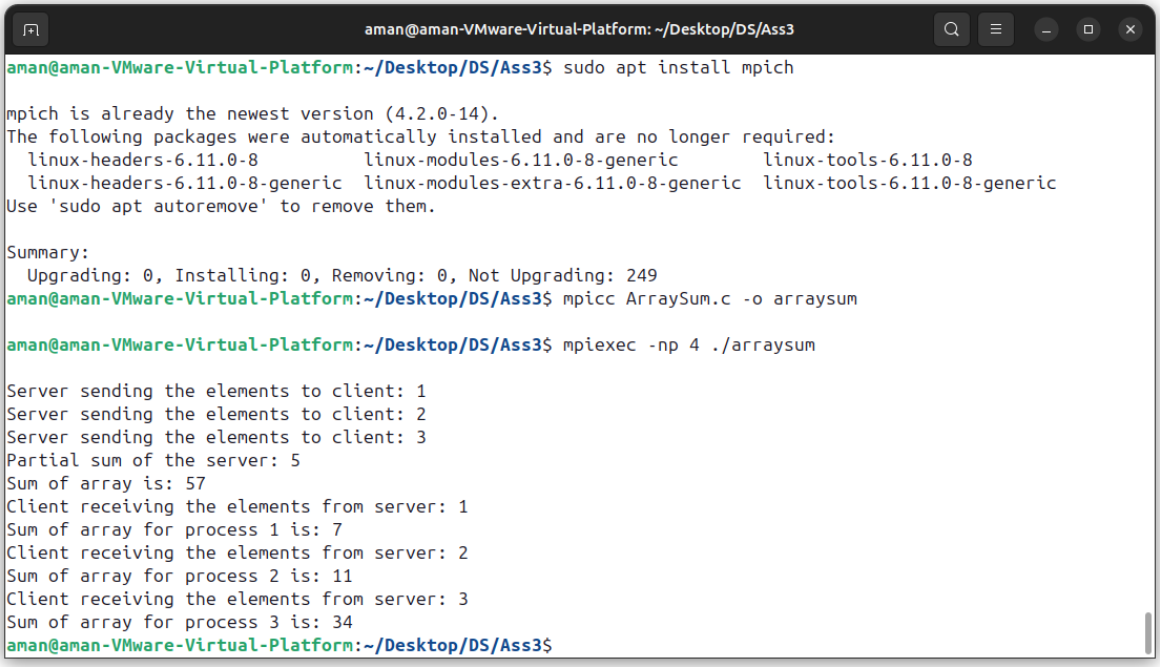
Assignment 3

ArraySum.c

```
#include <mpi.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

// size of array
#define n 10

int a[] = {3, 2, 3, 4, 5, 6, 7, 8, 9, 10};
// Temporary array for slave process
int a2[1000];

int main(int argc, char* argv[]) {
    int pid, np, elements_per_process, n_elements_recieved;
    MPI_Status status;

    // Creation of parallel processes
    MPI_Init(&argc, &argv);

    // Find out process ID and total number of processes
    MPI_Comm_rank(MPI_COMM_WORLD, &pid);
    MPI_Comm_size(MPI_COMM_WORLD, &np);

    // Master process
    if (pid == 0) {
        int index, i;
        elements_per_process = n / np;

        // Check if more than 1 process is running
        if (np > 1) {
            // Distribute portions of the array to child processes
            for (i = 1; i < np - 1; i++) {
                index = i * elements_per_process;
                MPI_Send(&elements_per_process, 1, MPI_INT, i, 0, MPI_COMM_WORLD);
                MPI_Send(&a[index], elements_per_process, MPI_INT, i, 0,
MPI_COMM_WORLD);
                printf("Server sending the elements to client: %d\n", i);
            }

            // Last process handles remaining elements
            index = i * elements_per_process;
            int elements_left = n - index;
```

```
            MPI_Send(&elements_left, 1, MPI_INT, i, 0, MPI_COMM_WORLD);
            MPI_Send(&a[index], elements_left, MPI_INT, i, 0, MPI_COMM_WORLD);
            printf("Server sending the elements to client: %d\n", i);
        }

        // Master process adds its own sub-array
        int sum = 0;
        for (i = 0; i < elements_per_process; i++) {
            sum += a[i];
        }
        printf("Partial sum of the server: %d\n", sum);

        // Collect partial sums from other processes
        int tmp;
        for (i = 1; i < np; i++) {
            MPI_Recv(&tmp, 1, MPI_INT, MPI_ANY_SOURCE, 0, MPI_COMM_WORLD,
&status);
            int sender = status.MPI_SOURCE;
            sum += tmp;
        }

        // Print final sum of array
        printf("Sum of array is: %d\n", sum);
    }
    // Slave processes
    else {
        MPI_Recv(&n_elements_recieved, 1, MPI_INT, 0, 0, MPI_COMM_WORLD, &status);
        MPI_Recv(&a2, n_elements_recieved, MPI_INT, 0, 0, MPI_COMM_WORLD, &status);
        printf("Client receiving the elements from server: %d\n", pid);

        // Compute partial sum
        int partial_sum = 0;
        for (int i = 0; i < n_elements_recieved; i++) {
            partial_sum += a2[i];
        }
        printf("Sum of array for process %d is: %d\n", pid, partial_sum);

        // Send partial sum to root process
        MPI_Send(&partial_sum, 1, MPI_INT, 0, 0, MPI_COMM_WORLD);
    }

    // Cleanup MPI state before exit
    MPI_Finalize();
    return 0;
}
```


aman@aman-VMware-Virtual-Platform: ~/Desktop/DS/Ass3
aman@aman-VMware-Virtual-Platform:~/Desktop/DS/Ass3$ sudo apt install mpich
mpich is already the newest version (4.2.0-14).
The following packages were automatically installed and are no longer required:
linux-headers-6.11.0-8 linux-modules-6.11.0-8-generic linux-tools-6.11.0-8
linux-headers-6.11.0-8-generic linux-modules-extra-6.11.0-8-generic linux-tools-6.11.0-8-generic
Use 'sudo apt autoremove' to remove them.
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 249
aman@aman-VMware-Virtual-Platform:~/Desktop/DS/Ass3$ mpicc ArraySum.c -o arraysum
aman@aman-VMware-Virtual-Platform:~/Desktop/DS/Ass3$ mpiexec -np 4 ./arraysum
Server sending the elements to client: 1
Server sending the elements to client: 2
Server sending the elements to client: 3
Partial sum of the server: 5
Sum of array is: 57
Client receiving the elements from server: 1
Sum of array for process 1 is: 7
Client receiving the elements from server: 2
Sum of array for process 2 is: 11
Client receiving the elements from server: 3
Sum of array for process 3 is: 34
aman@aman-VMware-Virtual-Platform:~/Desktop/DS/Ass3$