

Tema 5

Modificați codul de mai jos pentru a crea subgrupuri de procese conform cu regulile următoare:

- Numărul inițial al proceselor este 4.
- Procesul 2 lansează în execuție 5 procese de tip worker și apoi trimite fiecăruia un șir de caractere.
- Fiecare din cele 5 procese subordonate analizează șirul de caractere primit, calculează numărul de apariții ale literei **s**, transmite acest număr procesului coordonator (parent) și afișează rangul său și numărul de apariții ale literei **s**.
- Procesul 2 afișează rangul său și numărul de apariții ale literei **s** în cele 5 șiruri de caractere trimise proceselor subordonate.

```
/*
=====
Name      : parent.c
=====
*/
#include <stdio.h>
#include <string.h>
#include "mpi.h"

int main(int argc, char* argv[]) {
    int i;
    int my_rank;
    int p;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
    MPI_Comm_size(MPI_COMM_WORLD, &p);
    MPI_Group grp_world;
    MPI_Group grp_new;
    MPI_Comm newComm;
    MPI_Comm_group(MPI_COMM_WORLD, &grp_world);
    MPI_Group_incl(grp_world, 1, &my_rank, &grp_new);
    MPI_Comm_create(MPI_COMM_WORLD, grp_new, &newComm);
    int my_rank_grp;
    int p_grp;
    MPI_Comm_rank(newComm, &my_rank_grp);
    MPI_Comm_size(newComm, &p_grp);
    MPI_Comm workercomm;
    int b = my_rank;
    for (i = 0; i < p; ++i) {
        if (i == my_rank) {
            MPI_Comm_spawn("/home/prof/workspace_test/worker/Debug/worker",
                           MPI_ARGV_NULL, 3, MPI_INFO_NULL, 0, newComm, &workercomm,
                           MPI_ERRCODES_IGNORE );
        }
        MPI_Barrier(MPI_COMM_WORLD );
    }
    MPI_Bcast(&b, 1, MPI_INT, MPI_ROOT, workercomm);
    MPI_Comm_free(&workercomm);
    MPI_Finalize();
    return 0;
}
```

```

*
=====
Name      : worker.c
=====
*/
#include <stdio.h>
#include <string.h>
#include "mpi.h"

int main(int argc, char* argv[]) {
    int my_rank;
    int p;
    int p_remote;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
    MPI_Comm_size(MPI_COMM_WORLD, &p);
    MPI_Comm parentcomm;
    int b;
    MPI_Comm_get_parent(&parentcomm);
    MPI_Comm_remote_size(parentcomm, &p_remote);
    MPI_Bcast(&b, 1, MPI_INT, 0, parentcomm);
    MPI_Comm_free(&parentcomm);
    MPI_Finalize();
    return 0;
}

```

Baza 2 puncte