

Задание 14

MPI. Динамическое управление процессами. Порождение процессов

Для выполнения задания необходимо создать и скомпилировать две программы: Master (мастер) и Slave (рабочий). Мастер должен запускать рабочего, поэтому **будьте внимательны с именами выполняемых файлов**.

Мастер запускать через команду **mpirun** для одного процесса.

Пример запуска: **mpirun -n 1 ./master**

Разобраться с новыми функциями, объяснить выполнение программы.

Добавить третий процесс, который будет передавать от рабочего мастеру число запущенных процессов, мастер должен принять и вывести на экран.

Код программы Master

```
#include "mpi.h"
int main(int argc, char **argv)
{
    int size, rank1, rank2;
    MPI_Status status;
    MPI_Comm intercomm;
    char slave[10]="./slave";
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Comm_spawn(slave, MPI_ARGV_NULL, 2, MPI_INFO_NULL, 0, MPI_COMM_SELF, &intercomm,
MPI_ERRCODES_IGNORE);
    MPI_Recv(&rank1, 1, MPI_INT, 0, 0, intercomm, &status);
    MPI_Recv(&rank2, 1, MPI_INT, 1, 1, intercomm, &status);
```

Вывести на экран "Slaves **rank1** and **rank2** are working", на экране вместо слов **rank1** и **rank2** должны выводиться их значения.

```
    MPI_Finalize();
    return 0;
}
```

Код программы Slave

```
#include "mpi.h"
int main(int argc, char **argv)
{
    int rank;
    MPI_Comm intercomm;
    MPI_Init(&argc, &argv);
    MPI_Comm_get_parent(&intercomm);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Send(&rank, 1, MPI_INT, 0, rank, intercomm);
    MPI_Finalize();
    return 0;
}
```