

## Описание задачи

Доработать MPI-программу, реализованную в рамках курса “Суперкомпьютеры и параллельная обработка данных”. Добавить контрольные точки для продолжения работы программы в случае сбоя. Реализовать один из 3-х сценариев работы после сбоя: а) продолжить работу программы только на “исправных” процессах; б) вместо процессов, вышедших из строя, создать новые MPI-процессы, которые необходимо использовать для продолжения расчетов; в) при запуске программы на счет сразу запустить некоторое дополнительное количество MPI-процессов, которые использовать в случае сбоя.

Подготовить отчет о выполнении задания, включающий описание алгоритма, детали реализации, а также временные оценки работы алгоритма.

Отчет и текст программы необходимо разместить в трекере.

## Реализация алгоритма

Прошлогодной задачей является:

Разработка параллельной версии программы для умножения матрицы на вектор.

Реализован сценарий работы а) – в случае сбоя продолжить работу программы только на “исправных” процессах.

MPI программа, разработанная в рамках курса “Суперкомпьютеры и параллельная обработка данных”, реализовывала передачу данных между процессами через MPI\_Gather. Для корректной обработки буферов при аварийном завершении процессов, вместо MPI\_Gather теперь используются MPI\_Send и MPI\_Recv. Обработку ошибок вынесем в отдельную функцию verbose\_errhandler. В случае получения ошибки в функции MPI\_Recv (например, i-й процесс упал в состоянии SIGKILL) обработчик ошибок передаст в root-процесс ранг неисправного процесса. Для того, чтобы не начинать **все** вычисления умножения матрицы на вектор заново, root-процессом будут пересчитаны только результаты упавшего процесса.

## Команды для запуска

```
ompi/forbin/bin/mpicc MPI_with_err_handler.c -o ./MPI_with_err_handler
```

```
ompi/forbin/bin/mpiexec -with-ft ulfm --oversubscribe -np 3 ./MPI_with_err_handler
```

где ompi/forbin/ - папка с установленным ulfm