MPI

1. Создать коммуникатор, в котором нумерация процессов будет вестись в обратном порядке по сравнению с коммуникатором MPI_COMM_WORLD и напечатать ранги процессов в обоих коммуникаторах.
2. Создать две непересекающихся группы процессов и организовать обмен сообщениями через коммуникатор MPI_COMM_WORLD процессов с одинаковым рангом в этих группах.
3. Разбить все процессы приложения на три произвольных группы и напечатать ранги в MPI_COMM_WORLD тех процессов, что попали в первые две группы, но не попали в третью.
4. Переслать нулевому процессу от всех процессов приложения структуру (структуру создать на уровне MPI), состоящую из ранга процесса и названия узла, на котором данный процесс запущен (полученного с помощью MPI_GET_PROCESSOR_NAME).
5. Сравнить эффективность реализации функции MPI_SENDRECV с моделированием той же функциональности при помощи неблокирующих операций.
6. Смоделировать глобальное суммирование методом сдваивания и сравнить эффективность такой реализации с функцией MPI_Reduce.
7. Сравнить эффективность широковещательной посылки данных с их эквивалентом из нескольких пересылок типа точка-точка.
8. Использовать двумерную декартову топологию процессов при реализации параллельного перемножения матриц. Каждый процесс считает элемент результирующей матрицы.

OpenMP

1. Реализовать метод Гаусса для систем произвольной размерности на OpenMP. Показать ускорение.
2. Распараллелить алгоритм чет-нечетной перестановки. Показать ускорение. Сделать двумя способами – просто распределять итерации между потоками (при сравнении пар), и с помощью поблочного распределения массива между потоками и чередованием сортировки внутри блока и сравнением-разделением между потоками.

1. Придумать пример (можно искусственный), в котором динамическое планирование итераций имело бы преимущество в производительности над статической.
2. Придумать пример (можно искусственный), в котором множество потоков имеют доступ к разделяемому ресурсу через критическую секцию.