Дополнительные материалы к лабораторным работам по курсу «Основы ПП»

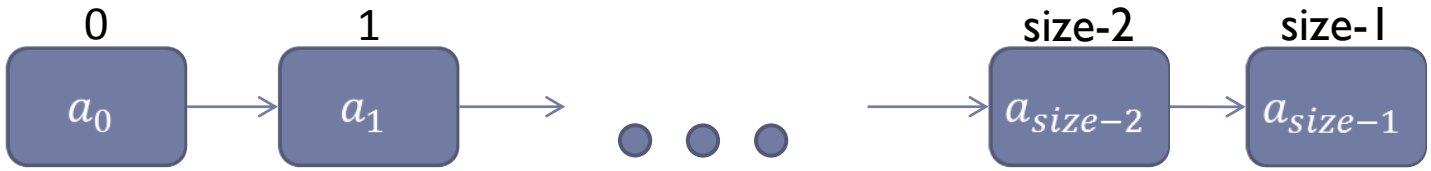
Черникова Анна

Рекомендации

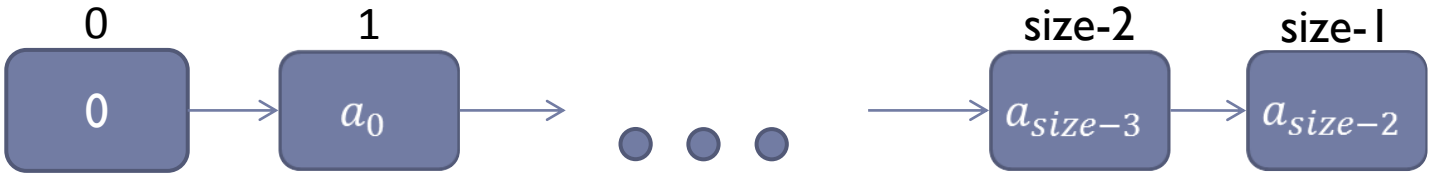
- Сравнить результаты работы параллельной программы и последовательной для одной и той же задачи. Как правило, в последовательной реализации допускается меньше ошибок.
- Не рекомендуется для тестирования программы использовать переменные с одинаковыми значениями.
- Необходимо исключить переполнение памяти и переполнение разряда при многократном суммировании в одну и ту же переменную.

Конвейерная передача информации. Пример.

До передачи



После передачи



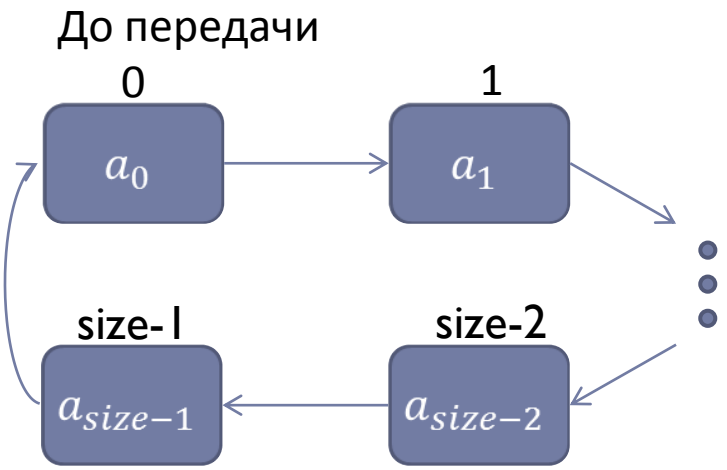
Задача. Передать значение по конвейеру и убедиться в том, что программа работает корректно. Чтобы тест был показательным необходимо выводить значения переменных до и после передачи.

Замечание!

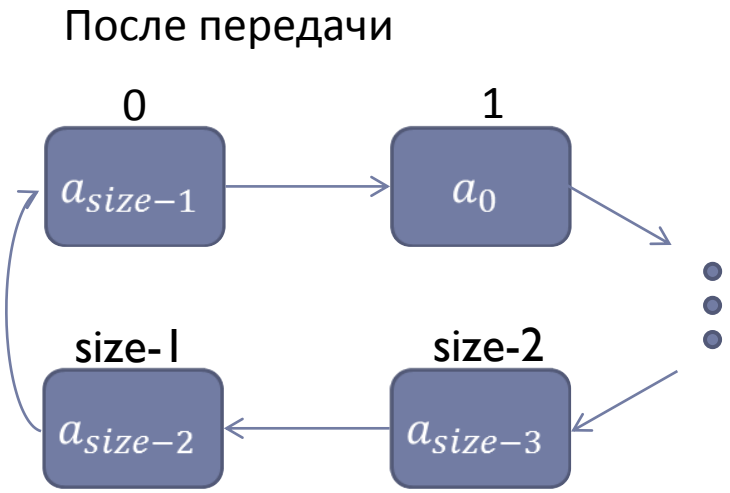
Если rank = 0, не принимает значение;

Если rank = size-1, не передает значение;

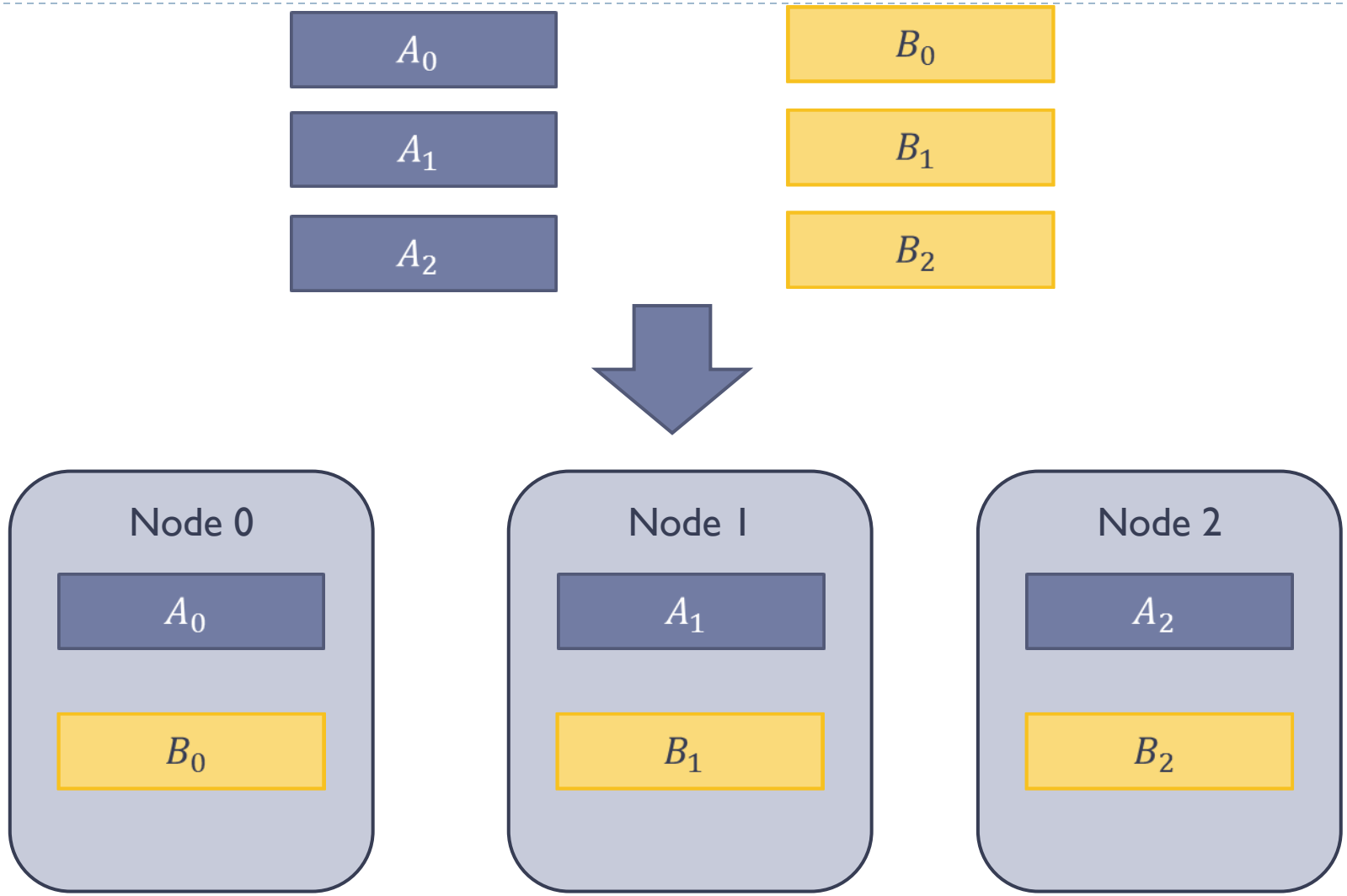
Кольцевая передача информации.



Задача. Передать значение по кольцу, убедиться, что программа работает верно.



Умножение матриц



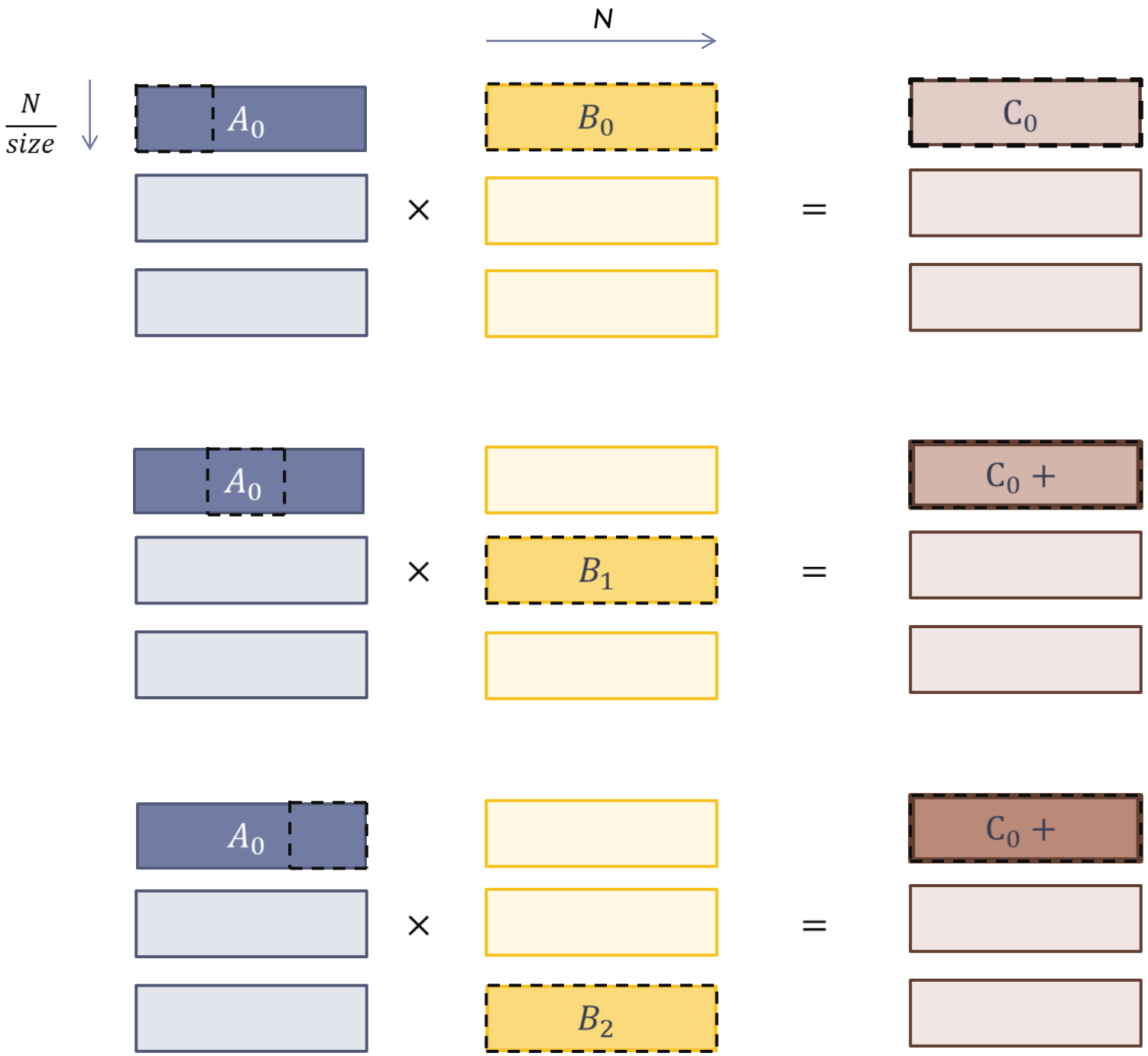
Умножение матриц. Последовательная программа.

Вариант 1

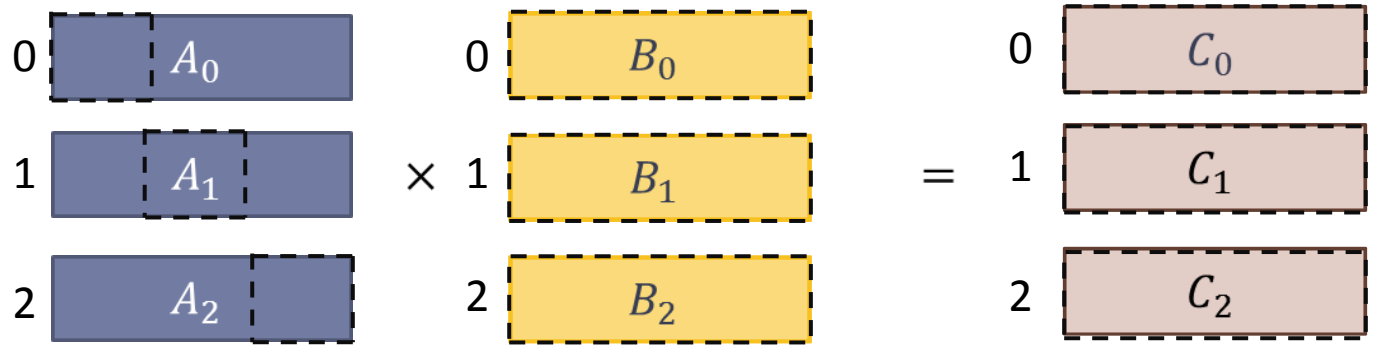
```
for(i=0; i<N; i++)
  for(j=0; j<N; j++)
    for(k=0; k<N; k++)
      C[i][j] += A[i][k]*B[k][j];
```

Вариант 2

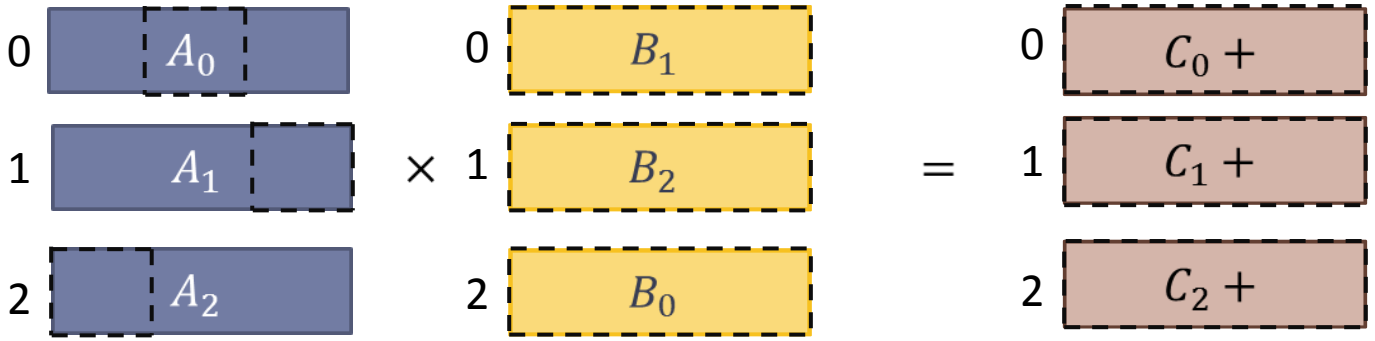
```
for(i=0; i<N; i++)
  for(k=0; k<N; k++)
    for(j=0; j<N; j++)
      C[i][j] += A[i][k]*B[k][j];
```


N
N
size
A_0
B_0
C_0
×
=
A_0
B_1
C_0 +
×
=
A_0
B_2
C_0 +
×
=

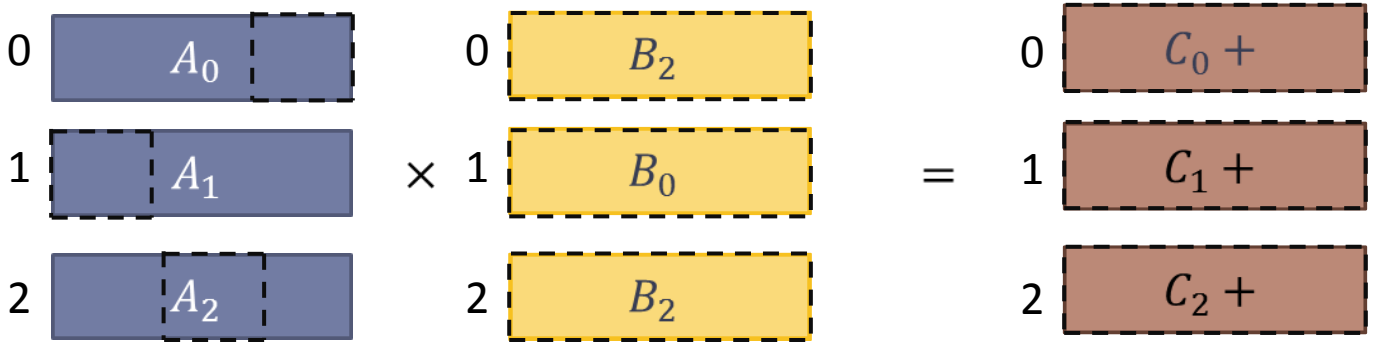
Сдвиг по кольцу: 0



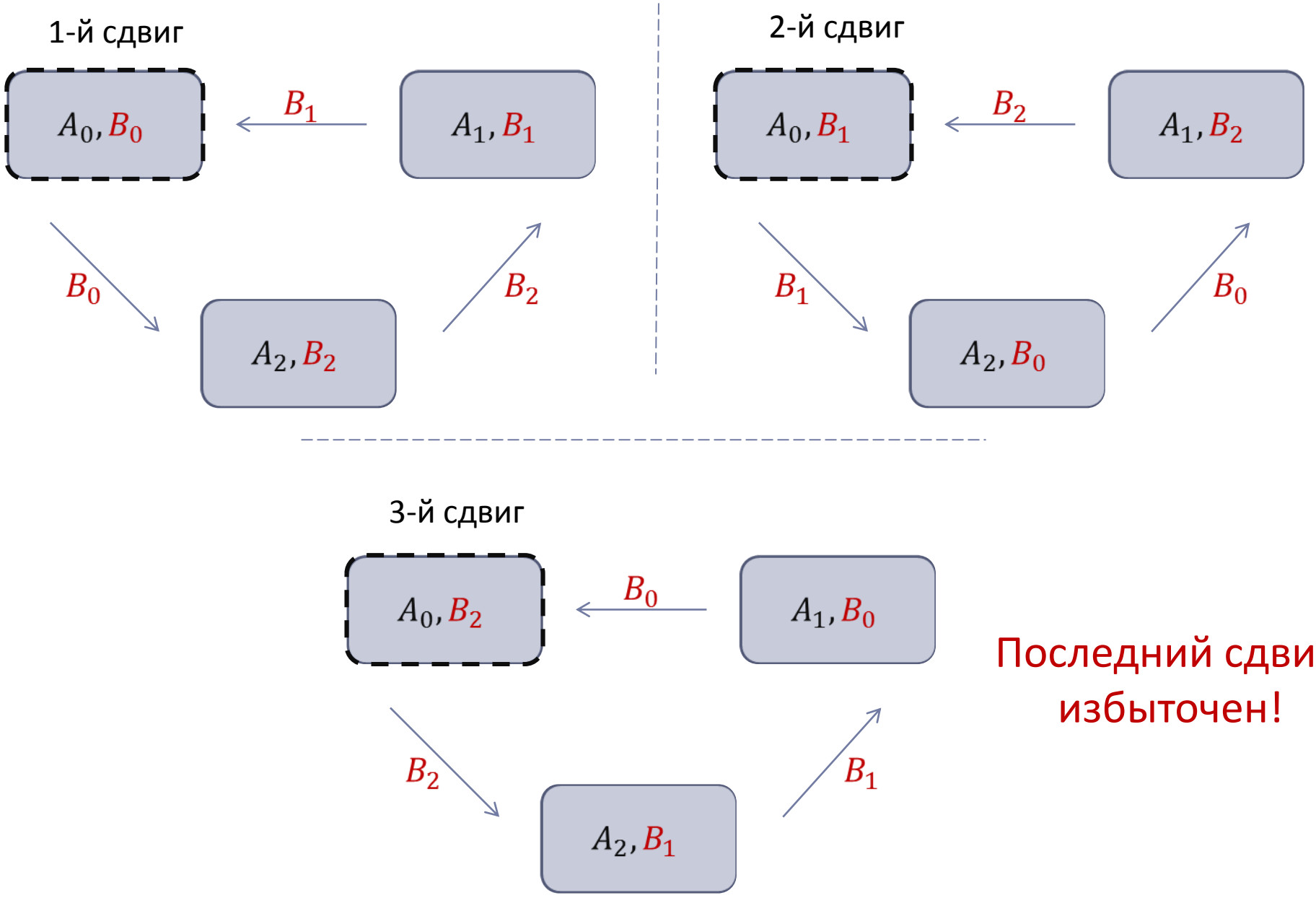
Сдвиг по кольцу: I



Сдвиг по кольцу: 2



Передача по кольцу



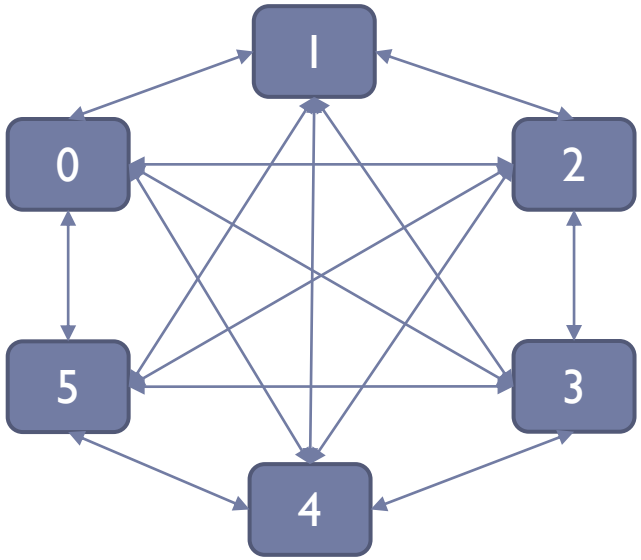
Последний сдвиг избыточен!

Умножение матриц. Сдвиг по кольцу.

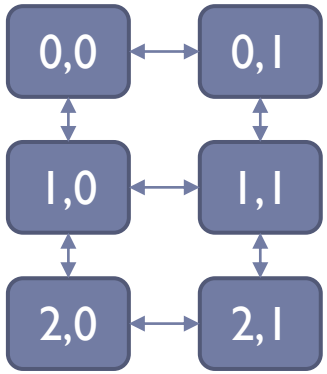
```
for(i=0; i<N/size; i++)
    for(j=0; j<N; j++)
      for(k=0; k<N/size; k++)
      {
        c[i][j] += a[i][?]*b[k][j];
        сдвинуть B против часовой стрелки;
      }
```

Виртуальные топологии. Декартовы топологии.

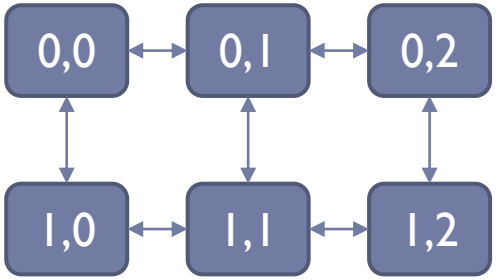
MPI_COMM_WORLD



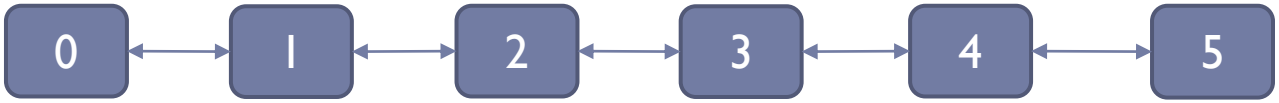
MPI_GRID1



MPI_GRID2



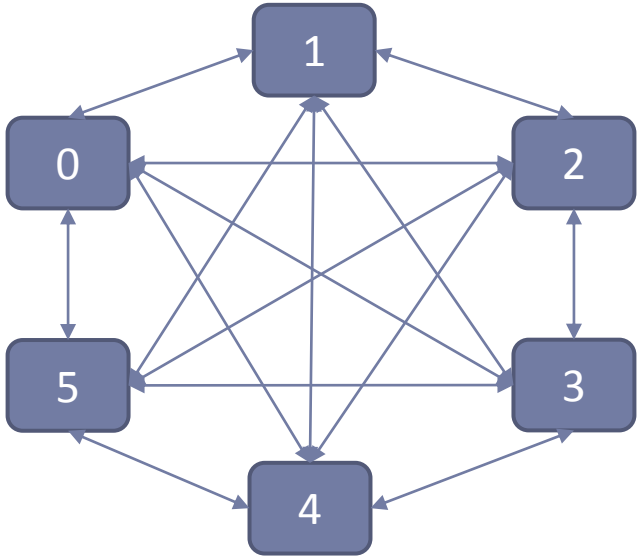
MPI_LINE



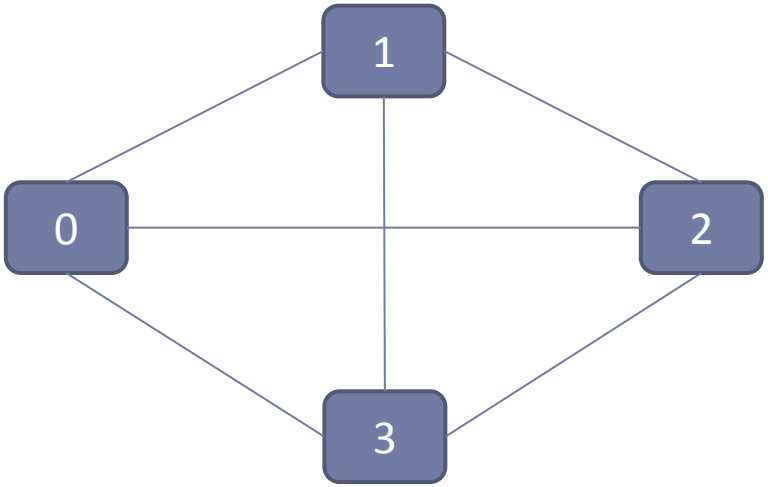
Виртуальные топологии. Декартовы топологии. Перекрытие топологий.

Виртуальные топологии. Топология графа.

MPI_COMM_WORLD



MPI_GRAPH



Матрица смежности

Процессы	Соседи
0	1,2,3
1	0,2
2	0,1,3
3	0,1,2

Виртуальные топологии.
Создание топологии «линейка». Пример.

```
int main(int argv, char **argv)
{
  int rank,  old_rank, size;
  MPI_Init(&argv, &argc);
  MPI_Comm_size(MPI_COMM_WORLD, &size);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);

  MPI_Comm new_comm;
  int ndim = 1s; //размерность новой декартовой топологии
  int dims[ndims]; //хранит число процессов вдоль каждого измерения
  int periods[ndims]; //хранит значение периодичности вдоль каждого измерения
  int coords[ndims]; //координаты процесса внутри декартовой топологии
  int reorder = 1; //разрешить или запретить переупорядочивание процессов
  dims[0] = 3; periods[0] = 0;

  old_rank = rank;
  MPI_Cart_create(MPI_COMM_WORLD,  ndims, dims, periods, reorder, &new_comm);
  MPI_Cart_coords(new_comm, rank, ndims, coords);

  printf("old rank = %d, new_rank = %d, coord = %d\n", old_rank, rank, coords[0]);
  MPI_Finalize();
 return 0;
}
```

Задание. Скомпилировать и запустить программу на 2, 3, 4 процессорах.
Проанализировать полученные результаты.