

Введение в МРІ

(практика)

План семинара

1. Проблема **блокирующих** операций
2. **Неблокирующие** операции
3. Операция **обмена**
4. Получение **(мета)информации** о сообщениях
5. О домашнем задании

Сообщения в стандартном режиме

```
int MPI_Send(const void *, int, MPI_Datatype, int, int, MPI_Comm);
```

и

```
int MPI_Recv(void *, int, MPI_Datatype, int, int, MPI_Comm, MPI_Status *);
```

— блокирующие операции.

Согласно стандарту MPI, сообщение *может быть*:

- Буферизировано
- Доставлено напрямую

Следовательно, *возможен* deadlock:

```
/* rank=0 */
```

```
MPI_Send(..., 1, ...);
```

```
MPI_Recv(..., 1, ...);
```

```
/* rank=1 */
```

```
MPI_Send(..., 0, ...);
```

```
MPI_Recv(..., 0, ...);
```

Решение

- **Правильный порядок операций**
 - Например:
 - чётные процессы вызывают вначале `MPI_Send()`
 - нечётные — вначале `MPI_Recv()`
- **Неблокирующие операции**
- **Операция обмена**

Неблокирующие операции

```
switch(rank)
case 0:
    for (i = 0; i < 5; i++)
    {
        array[i] = i;
    }
    MPI_Isend(array, 5, MPI_INT, 1, 42, MPI_COMM_WORLD, &r_send);
    MPI_Irecv(&array[5], 5, MPI_INT, 1, 42, MPI_COMM_WORLD, &r_recv);

    MPI_Wait(&r_recv, &status);  /* Блокирующая операция */
```

Аналогично для rank=1

Память для данных **не должна модифицироваться** до завершения запроса!
(array в примере выше)

Операция обмена

```
/* rank=0 */  
MPI_Sendrecv(  
    &array[5], 5, MPI_INT, 1, 42, /* приём */  
    array,      5, MPI_INT, 1, 42, /* отправка */  
    MPI_COMM_WORLD, &Status  
);
```

Аналогично для rank=1

- **Блокирующая**
- **Одновременно** выполняется отправка и приём сообщения
 - Буферы отправки и приёма должны **не пересекаться**
- **Взаимодействует с обычными методами** приёма и отправки
- Можно организовать простой **RPC**

Информация о сообщениях

В MPI есть следующие полезные методы (см. [man-страницы](#)):

- `MPI_Waitall(int, MPI_Request[], ...)`
- `MPI_Waitany(...)`
- `MPI_Test(MPI_Request*, int*, ...)`
- `MPI_Probe(int source, int tag, ..., MPI_Status*)`
- `MPI_IProbe(int source, int tag, ..., MPI_Status*)`
- `MPI_Get_count(MPI_Status*, MPI_Datatype, int*)`

Полезные константы

При **приёме** сообщения можно указывать константы:

- `MPI_ANY_SOURCE`
- `MPI_ANY_TAG`

При **отправке** сообщения можно указывать константу:

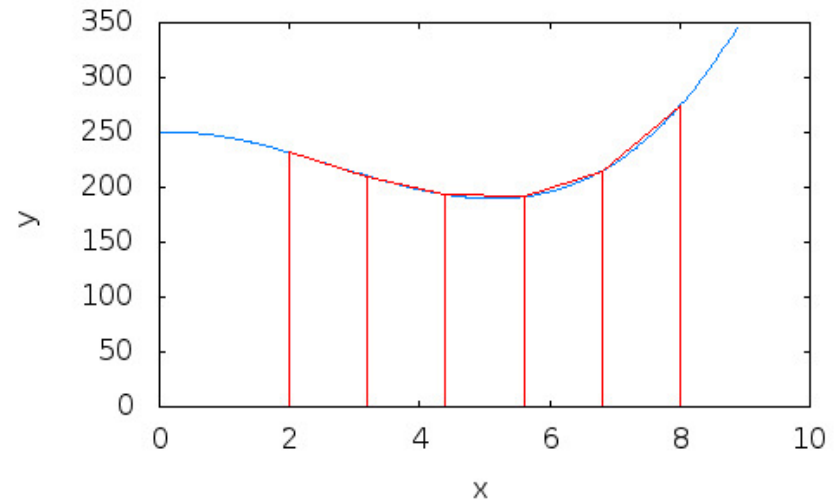
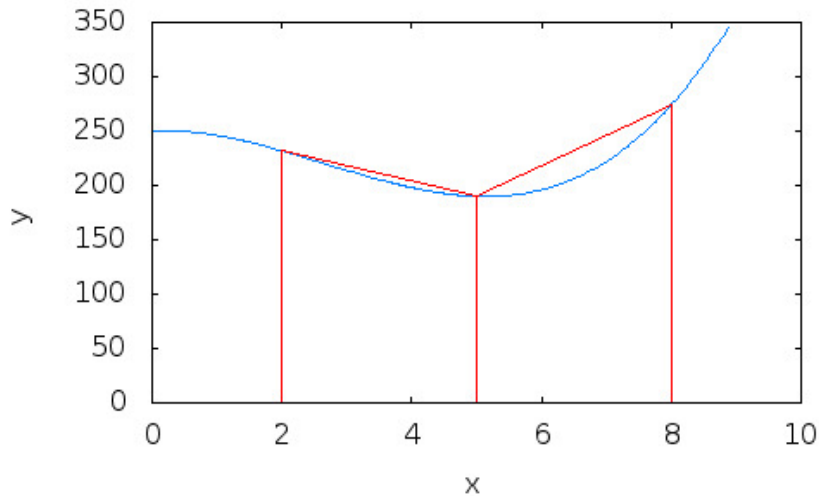
- `MPI_PROC_NULL`

Структура **`MPI_Status`** (согласно стандарту MPI) содержит поля:

- `MPI_SOURCE`
- `MPI_TAG`
- `MPI_ERROR`

Задание: подсчёт интеграла

Метод трапеций:



$$\int_a^b f(x) dx \approx (b - a) \cdot \frac{f(a) + f(b)}{2}.$$

Для измерения elapsed time — `MPI_Wtime()`