Создание групп и коммуникаторов

Параллельное программирование

Группы

Группа представляет собой *упорядоченное множество процессов*. Каждый процесс идентифицируется переменной целого типа. Идентификаторы процессов образуют непрерывный ряд, начинающийся с *нуля*.

В MPI вводится специальный тип данных **MPI_Group** и набор функций для работы с переменными и константами этого типа.

Существует две предопределенных группы:

MPI_GROUP_EMPTY- группа, не содержащая ни одного процесса;

MPI_GROUP_NULL - возвращаемое значение функций, когда группа не может быть создана.

Созданная группа не может быть модифицирована - *расширена* или *усечена*, может быть только создана новая группа. Интересно отметить, что при инициализации MPI не создается группы, соответствующей коммуникатору MPI_COMM_WORLD. Она должна создаваться специальной функцией явным образом.

Функция MPI_Comm_group

создает группу *group* для множества процессов, входящих в область связи коммуникатора *comm*.

```
int MPI_Comm_group (MPI_Comm comm, MPI_Group *group)
```

- Входные параметры:

`comm`	коммуникатор области связи

- Выходные параметры:

`group`	созданная группа

Функция MPI_Group_incl

создает новую группу *newgroup* из *n* процессов, входящих в группу *oldgroup*. Ранги этих процессов содержатся в массиве *ranks*

```
int MPI_Group_incl (MPI_Group oldgroup, int n,
int *ranks, MPI_Group *newgroup)
```

- Входные параметры:

`oldgroup`	существующая группа, на основе которой создается новая
`n`	количество процессов в новой группе
`ranks`	ранги процессов, вошедших в эту группу

- Выходные параметры:

`newgroup`	созданная группа

Функция MPI_Group_incl

создает новую группу *newgroup* из *n* процессов, входящих в группу *oldgroup*. Ранги этих процессов содержатся в массиве *ranks*

```
int MPI_Group_incl (MPI_Group oldgroup, int n,
int *ranks, MPI_Group *newgroup)
```

В НОВУЮ группу войдут процессы с рангами ranks[0], ..., ranks[n-1], причем рангу *i* в новой группе соответствует ранг *ranks(i)* в старой группе. При *n = 0* создается пустая группа MPI_GROUP_EMPTY.

С помощью данной функции можно не только создать новую группу, но и изменить порядок процессов в старой группе.

Функция MPI_Group_excl

создает группу *newgroup*, исключая из исходной группы (*oldgroup*) процессы с рангами ranks[0], ..., ranks[n-1]

```
int MPI_Group_excl (MPI_Group oldgroup, int n,
int *ranks, MPI_Group *newgroup)
```

- Входные параметры:

`oldgroup`	существующая группа, на основе которой создается новая
`n`	количество удаляемых из группы процессов
`ranks`	ранги процессов, которые не войдут в группу

- Выходные параметры:

`newgroup`	созданная группа

Функция MPI_Group_union

формирует новую группу из всех элементов 1-й и 2-й групп (объединение множеств)

```
int MPI_Group_union (MPI_Group group1,
MPI_Group group2, MPI_Group *newgroup)
```

- Входные параметры:

group1	первая группа
group2	вторая группа

- Выходные параметры:

newgroup	созданная группа

Функция MPI_Group_intersection

новая группа формируется из элементов 1-й группы, которые входят также и во 2-ю. Упорядочивание, как в 1-й группе (пересечение множеств)

```
int MPI_Group_intersection (MPI_Group group1,
MPI_Group group2, MPI_Group *newgroup)
```

- Входные параметры:

group1	первая группа
group2	вторая группа

- Выходные параметры:

newgroup	созданная группа

Функция MPI_Group_difference

формирует новую группу из всех элементов 1-й группы, которые не входят во 2-ю. Упорядочивание, как в 1-й группе (дополнение множеств)

```
int MPI_Group_difference (MPI_Group group1,
MPI_Group group2, MPI_Group *newgroup)
```

- Входные параметры:

group1	первая группа
group2	вторая группа

- Выходные параметры:

newgroup	созданная группа

Функция MPI_Group_size

определяет количество (*size*) процессов в группе *group*

```
int MPI_Group_size (MPI_Group group, int *size)
```

- Входные параметры:

`group`	группа

- Выходные параметры:

`size`	количество процессов

Функция MPI_Group_rank

возвращает ранг (*rank*) процесса в группе *group*

```
int MPI_Group_rank (MPI_Group group, int *rank)
```

- Входные параметры:

`group`	группа

- Выходные параметры:

`rank`	ранг процесса в данной группе (или MPI_UNDEFINED)

Если процесс не входит в указанную группу, вторым аргументом возвращается значение **MPI_UNDEFINED**

MPI_Group_translate_ranks -

Функция установки соответствия между номерами процессов в двух группах.

```
int MPI_Group_rank (MPI_Group group1, int n, int *ranks1,
MPI_Group group2, int *ranks2)
```

- Входные параметры:

group1	группа1
n	число процессов, для которых устанавливается соответствие
ranks1	Массив номеров процессов из первой группы
group2	группа2

- Выходные параметры:

ranks2	номера тех же процессов во второй группе

Функция MPI_Group_free

уничтожает группу *group*

```
int MPI_Group_free (MPI_Group *group)
```

- Входные параметры:

`group`	уничтожаемая группа

- Выходные параметры:

`group`	указатель на пустую группу MPI_GROUP_NULL

Коммуникаторы

Коммуникатор представляет собой скрытый объект с некоторым набором атрибутов, а также правилами его создания, использования и уничтожения. Коммуникатор описывает некоторую область связи. Одной и той же области связи может соответствовать несколько коммуникаторов, но даже в этом случае они не являются тождественными и не могут участвовать во взаимном обмене сообщениями. Если данные посылаются через один коммуникатор, процесс-получатель может получить их только через тот же самый коммуникатор.

При инициализации MPI создается два предопределенных коммуникатора:

- MPI_COMM_WORLD – описывает область связи, содержащую все процессы;
- MPI_COMM_SELF – описывает область связи, состоящую из одного процесса.

Функция MPI_Comm_create

создает новый коммуникатор из группы group

```
int MPI_Comm_create (MPI_Comm comm, MPI_Group group,
MPI_Comm *newcomm)
```

- Входные параметры:

comm	родительский коммуникатор
group	группа, для которой создается коммуникатор

- Выходные параметры:

newcomm	новый коммуникатор

Функция MPI_Comm_dup

Функция дублирования коммуникатора

```
int MPI_Comm_dup (MPI_Comm comm, MPI_Comm *newcomm)
```

- Входные параметры:

comm	родительский коммуникатор

- Выходные параметры:

newcomm	новый коммуникатор

Функция MPI_Comm_split

расщепляет группу, связанную с родительским коммуникатором, на непересекающиеся подгруппы

```
int MPI_Comm_split (MPI_Comm comm, int color,
int key, MPI_Comm *newcomm)
```

- Входные параметры:

`comm`	родительский коммуникатор
`color`	признак подгруппы
`key`	управление упорядочиванием

- Выходные параметры:

`newcomm`	новый коммуникатор

Функция MPI_Comm_split

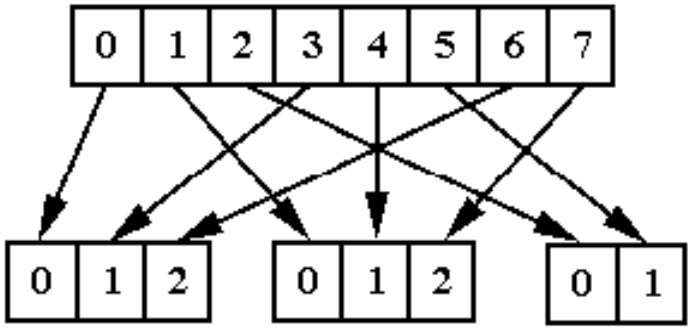
```
int MPI_Comm_split (MPI_Comm comm, int color,
int key, MPI_Comm *newcomm)
```

Функция расщепляет группу, связанную с родительским коммуникатором, на непересекающиеся подгруппы по одной на каждое значение признака подгруппы *color*. Значение *color* должно быть неотрицательным. Каждая подгруппа содержит процессы с одним и тем же значением *color*. Параметр *key* управляет упорядочиванием внутри новых групп: меньшему значению *key* соответствует меньшее значение идентификатора процесса. В случае равенства параметра *key* для нескольких процессов упорядочивание выполняется в соответствии с порядком в родительской группе.

Функция MPI_Comm_split

Алгоритм расщепления группы из восьми процессов на три подгруппы:

```
MPI_comm comm, newcomm;
int myid, color;
……
MPI_Comm_rank(comm, &myid);
color = myid%3;
MPI_Comm_split(comm, color, myid, &newcomm);
```



В данном примере первую подгруппу образовали процессы, номера которых делятся на 3 без остатка, вторую, для которых остаток равен 1, и третью, для которых остаток равен 2. Отметим, что после выполнения функции MPI_Comm_split значения коммуникатора newcomm в процессах разных подгрупп будут отличаться.

Функция MPI_Comm_free

уничтожает коммуникатор comm

```
int MPI_Comm_free (MPI_Comm *comm)
```

- Входные параметры:

comm	уничтожаемый коммуникатор