МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Системы параллельной обработки данных»

Тема: ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ ОБМЕНА ДАННЫМИ «ТОЧКА-ТОЧКА» В БИБЛИОТЕКЕ MPI

Студентка гр. 5303 _____ Допира В.Е.

Преподаватель _____ Татаринов Ю.С.

Санкт-Петербург

2019

Формулировка задания

Вариант 9. Игра с ведущим (съедобное − несъедобное). Процесс 0 поочередно посылает остальным процессам сообщение. Получив сообщение, процесс-приемник информирует об этом процесс 0.

Описание алгоритма с использованием аппарата Сетей Петри

P_0 - начальное состояние. Нулевой процесс из P_0 передает поочередно сообщения остальным процессам, тем самым осуществляются переходы t_1, t_2, t_3 к состояниям P_1, P_2, P_3 соответственно. Количество переходов соответствует количеству заданных процессов минус 1. Здесь показано на трех. Затем процессы извещают нулевой о получении сообщения, срабатывают переходы t_4, t_5, t_6 к состоянию P_4. Программа завершает выполнение.

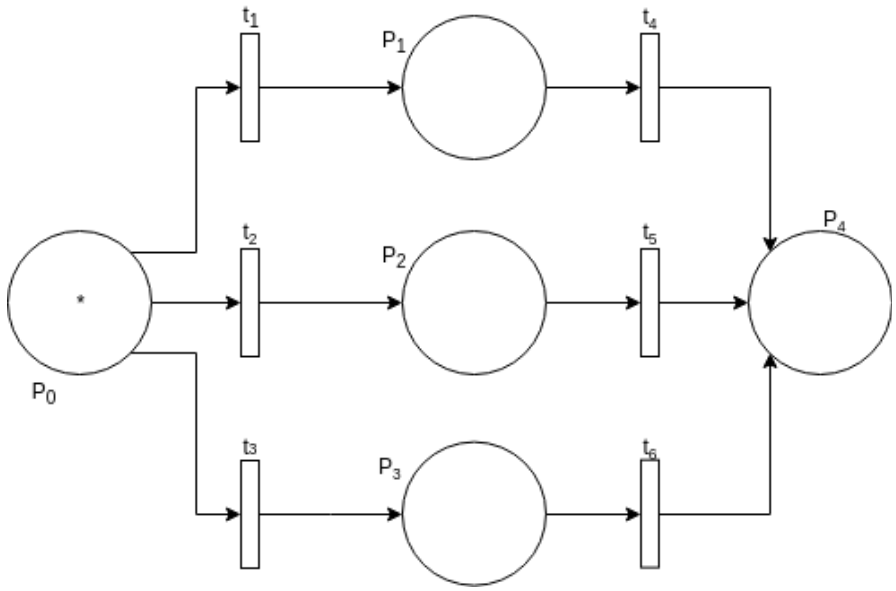

Рисунок 1 — Сеть Петри

Результаты работы программы на 1,2 …. N процессорах

```
$ mpirun -np 2 1.x
Task received from 0 to 1
Task received from 1 to 0

$ mpirun -np 3 1.x
Task received from 0 to 1
Task received from 0 to 2
Task received from 1 to 0
Task received from 2 to 0
```

Вывод

В ходе выполнения лабораторной работы была написана программа с использованием MPI, моделирующая игру с ведущим.

Листинг программы

```
#include "mpi.h"
#include <stdio.h>

int main(int argc, char* argv[]){
  int ProcNum, ProcRank, RecvRank, dest, source, count, tag = 1;
  MPI_Status Status;
  MPI_Init(&argc, &argv);
  MPI_Comm_size(MPI_COMM_WORLD, &ProcNum);
  MPI_Comm_rank(MPI_COMM_WORLD, &ProcRank);
  if ( ProcRank == 0 ){ //Действия, выполняемые только процессом с
рангом 0
    for ( int i=1; i < ProcNum; i++ ) {
      MPI_Send(&ProcRank, 1, MPI_INT, i, tag, MPI_COMM_WORLD);
      MPI_Recv(&RecvRank, 1, MPI_INT, i, tag, MPI_COMM_WORLD,
&Status);
     printf("Task received from %d to %d \n", ProcRank, RecvRank);
    }
  }
  else {// Сообщение, отправляемое всеми процессами, кроме
процесса с рангом 0
    MPI_Recv(&RecvRank, 1, MPI_INT, 0, tag, MPI_COMM_WORLD,
&Status);
    MPI_Send(&ProcRank, 1, MPI_INT, 0, tag, MPI_COMM_WORLD);
    printf("Task received from %d to %d \n", ProcRank, RecvRank);
  }
  MPI_Finalize();
  return 0;
}
```