|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет экономики, менеджмента и бизнес-информатики* |
|  |
| Чепоков Елизар Сергеевич  **ВОЛНОВЫЕ АЛГОРИТМЫ**  *Лабораторная работа №6*  *по дисциплине «Распределенные вычисления»*  студента образовательной программы «Программная инженерия»  по направлению подготовки *09.03.04 Программная инженерия*   |  |  | | --- | --- | |  |  | |

Пермь, 2022 год

**Оглавление**

[Постановка задачи 4](#_Toc99333534)

[1. Алгоритм Эха 4](#_Toc99333535)

[2. Алгоритм Финна 5](#_Toc99333536)

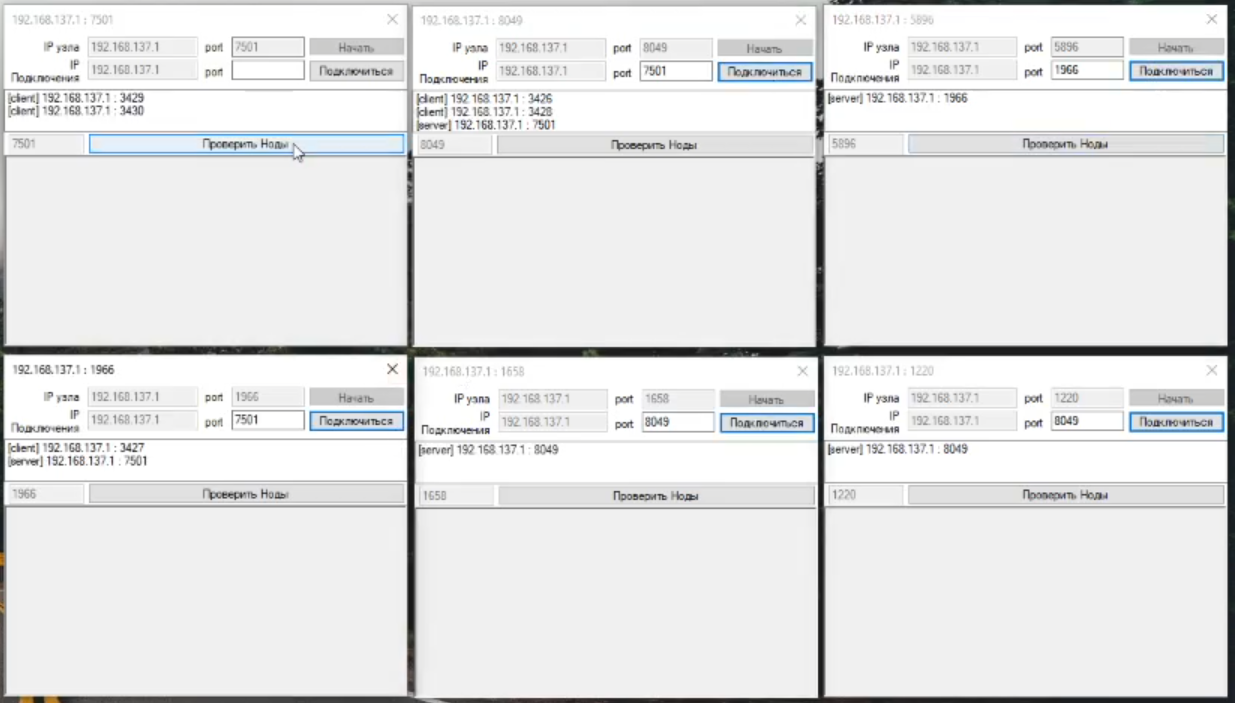
Постановка задачи

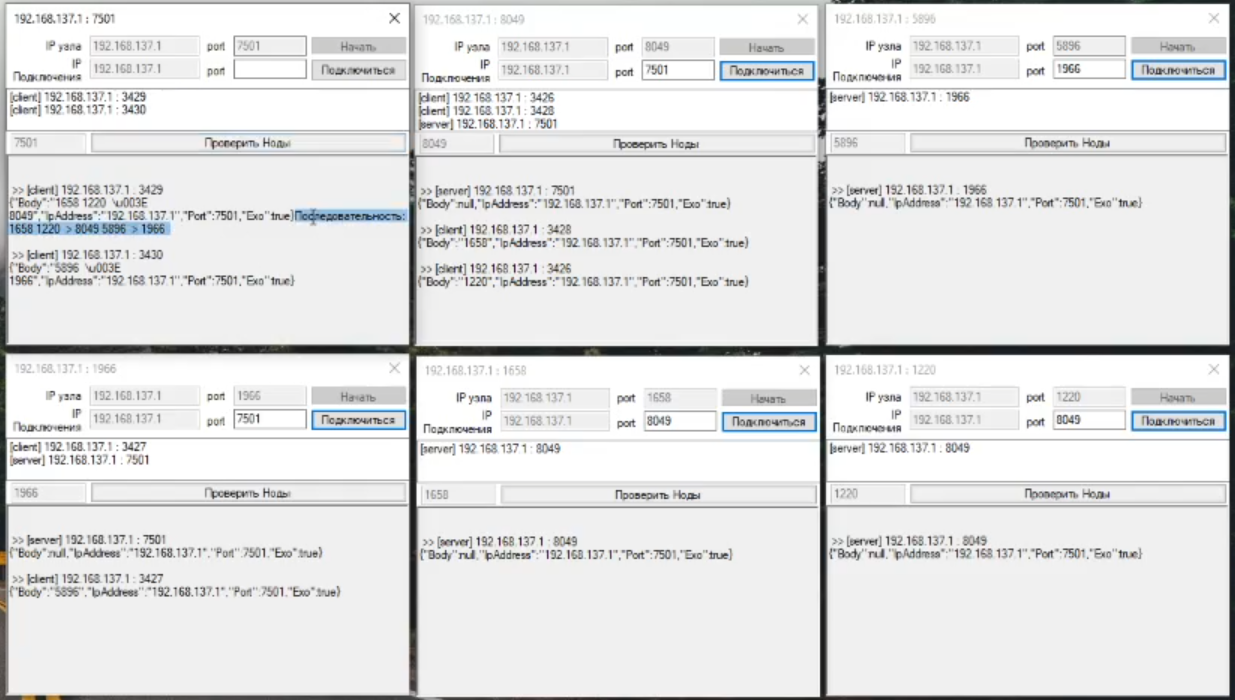
Разработать приложения, реализующие:

1. Алгоритм Эха.

2. Алгоритм Финна.

1. Алгоритм Эха

Результат работы программы:



***Рисунок 2 – Конец работы программы***

***Рисунок 1 – Начало работы программы***

Листинг:

private void Echo\_algorithm(Socket socket, MsgBody msgBody)

{

if (exoInitiator)

{

exoCount += 1;

exoMessage += $"{msgBody.Body} ";

if (exoCount == Neighbours.Count)

{

rtbMessages.Text += $"Последовательность: {exoMessage}";

exoCount = 0;

exoInitiator = false;

btnCheck.BackColor = Color.FromName("Control");

exoMessage = "";

}

}

else

{

btnCheck.BackColor = Color.Green;

if (exoParentIndex == -1)

{

if (Neighbours.Count == 1)

{

exoCount = 0;

Thread.Sleep(DELAY);

btnCheck.BackColor = Color.FromName("Control");

msgBody.Body = tbMessage.Text;

SendMessages((Socket)Neighbours[0][1], msgBody);

}

else

{

exoParentIndex = Neighbours.FindIndex(a => (Socket)a[1] == socket);

Thread.Sleep(DELAY);

btnCheck.BackColor = Color.Red;

for (int i = 0; i < Neighbours.Count; i++)

if (i != exoParentIndex)

SendMessages((Socket)Neighbours[i][1], msgBody);

}

}

else

{

exoCount += 1;

exoMessage += $"{msgBody.Body} ";

if (exoCount == Neighbours.Count - 1)

{

exoCount = 0;

Thread.Sleep(DELAY);

btnCheck.BackColor = Color.FromName("Control");

msgBody.Body = $"{exoMessage} > {tbMessage.Text}";

SendMessages((Socket)Neighbours[exoParentIndex][1], msgBody);

exoParentIndex = -1;

exoMessage = "";

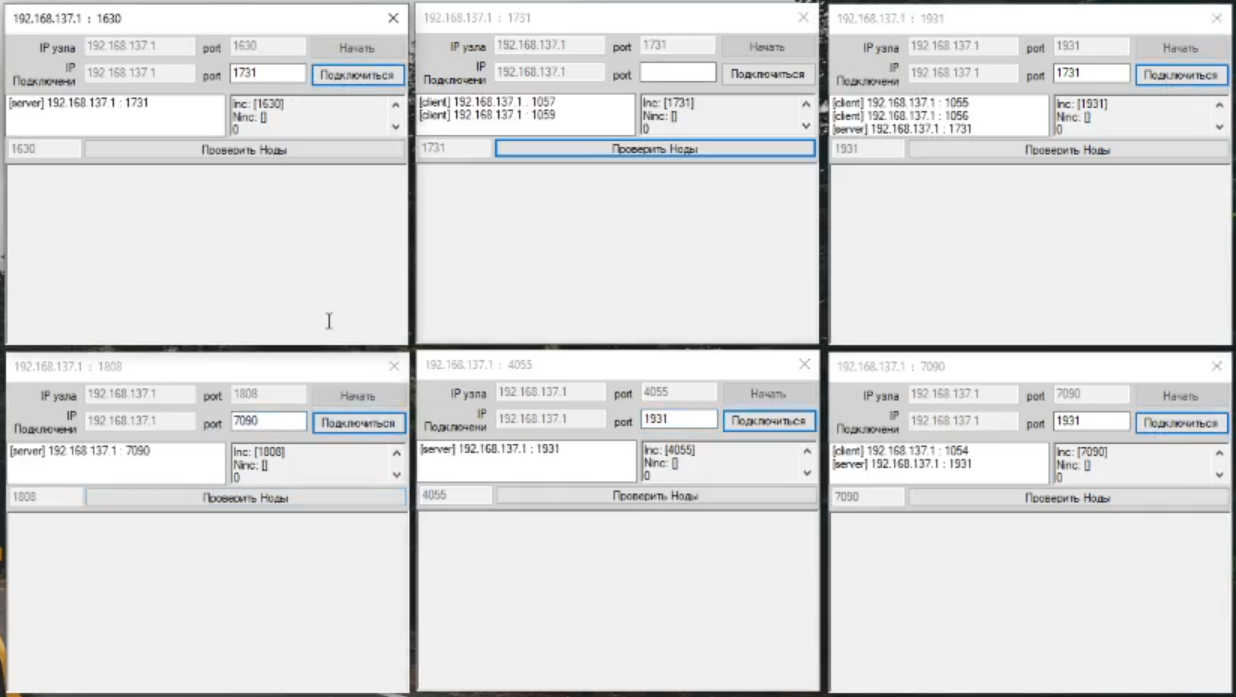
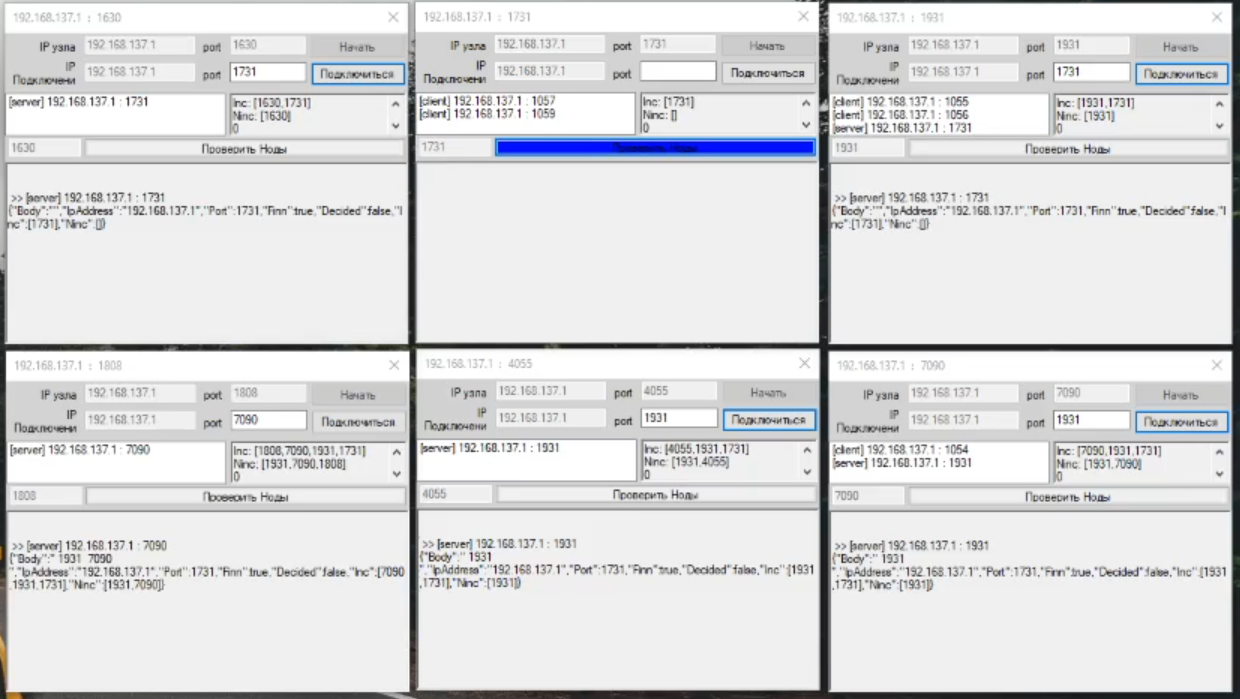
}

}

}

}

1. Алгоритм Финна

Результат работы программы:

***Рисунок 4 – Конец работы программы***

***Рисунок 3 – Начало работы программы***

Листинг:

private void Finn\_algorithm(Socket socket, MsgBody msgBody)

{

btnCheck.BackColor = Color.Red;

if (msgBody.Decided)

{

if (finnInitiator)

{

rtbMessages.Text += $"Собрано сообщение: \r\n{msgBody.Body}";

finnCount = 0;

finnInitiator = false;

btnCheck.BackColor = Color.FromName("Control");

}

else

{

finnCount = 0;

Thread.Sleep(DELAY);

SendToAllClients(msgBody);

btnCheck.BackColor = Color.FromName("Control");

}

}

else

{

Inc.UnionWith(msgBody.Inc);

Ninc.UnionWith(msgBody.Ninc);

msgBody.Inc = Inc;

msgBody.Ninc = Ninc;

UpdateInfo();

finnCount++;

finnMsgBody = msgBody;

finnMessage += $"{msgBody.Body} ";

int serversCount = Neighbours.FindAll(a => a[0].ToString() == "server").Count;

if (finnCount == serversCount)

{

finnCount = 0;

Ninc.Add(Port);

finnMsgBody.Inc = Inc;

finnMsgBody.Ninc = Ninc;

UpdateInfo();

finnMsgBody.Body = $"{finnMessage}{tbMessage.Text} ";

finnMessage = "";

if (timer != null)

{

timer.Dispose();

timer = null;

}

}

else

{

timer = new Timer(new TimerCallback(WaitingTimeout));

timer.Change(WAITING\_TIME, 0);

return;

}

if (Inc.SetEquals(Ninc))

{

btnCheck.BackColor = Color.Purple;

finnMsgBody.Decided = true;

if (finnInitiator)

{

MessageBox.Show($"Собрано сообщение: \r\n{finnMsgBody.Body}");

finnCount = 0;

finnInitiator = false;

btnCheck.BackColor = Color.FromName("Control");

return;

}

}

Thread.Sleep(DELAY);

SendToAllClients(finnMsgBody);

if (finnInitiator)

btnCheck.BackColor = Color.Blue;

else

btnCheck.BackColor = Color.FromName("Control");

}

}