

Параллельные процессы

MPI

MPI

- Message Passing Interface
- Компилятор поддерживает вызовы библиотечных функций
- Требуется установка дополнительной библиотеки MPI
 - MPICH – OpenSource библиотека MPI
 - <http://www.mcs.anl.gov/research/projects/mpich2/>
 - скачать MPICH2 Windows IA32 (binary)

Установка MPI

- SPMD process manager – привилегии администратора для установки
 - SPMD – служба запуска MPI процессов
- Passphrase (пароль по умолчанию)
- behappy
- Установка в папку (по умолчанию)
 - C:\Program Files\MPICH2\

Установка

- MPICH2 устанавливается на все машины, где будут запускаться процессы MPI
- При установке создаются каталоги
 - mpich2\bin
 - smpd.exe – менеджер MPI процессов
 - mpiexec.exe – программа для запуска заданий
 - mpich2\include
 - mpich2\lib
 - библиотеки для компиляции
- Библиотеки dll копируются в Windows\system32

Компиляция

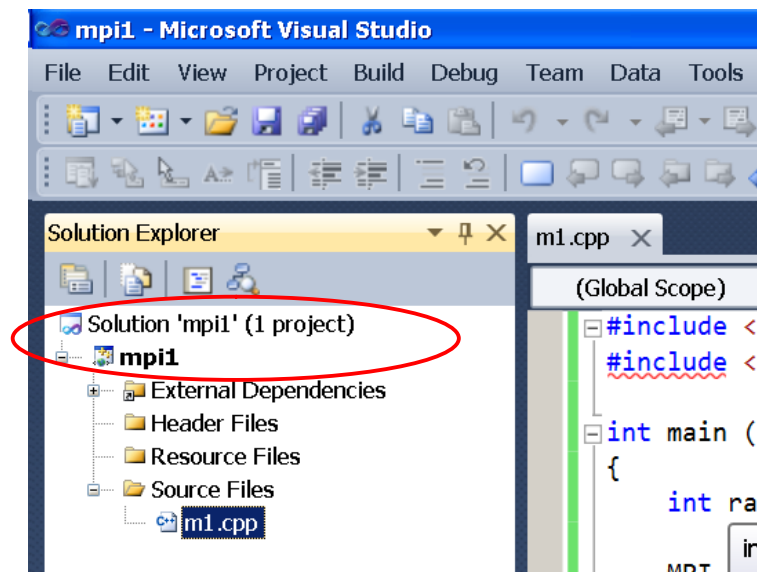
- 2) Add mpich2\include to the include path
- 3) Add mpich2\lib to the library path
- 4) For C applications add mpi.lib to your target link command.
- 6) Compile
- 7) Place your application and all the dlls it depends on in a shared location or copy them to all the nodes.
- 8) Run the application using mpiexec
- RUNNING MPI JOBS:
- mpiexec is a command line application used to launch MPI jobs. Bring up a command prompt to run it. Execute "mpiexec" to see the available options.
- The simplest mpiexec command is like this:
- mpiexec -n 3 myapp.exe

Минимальная установка

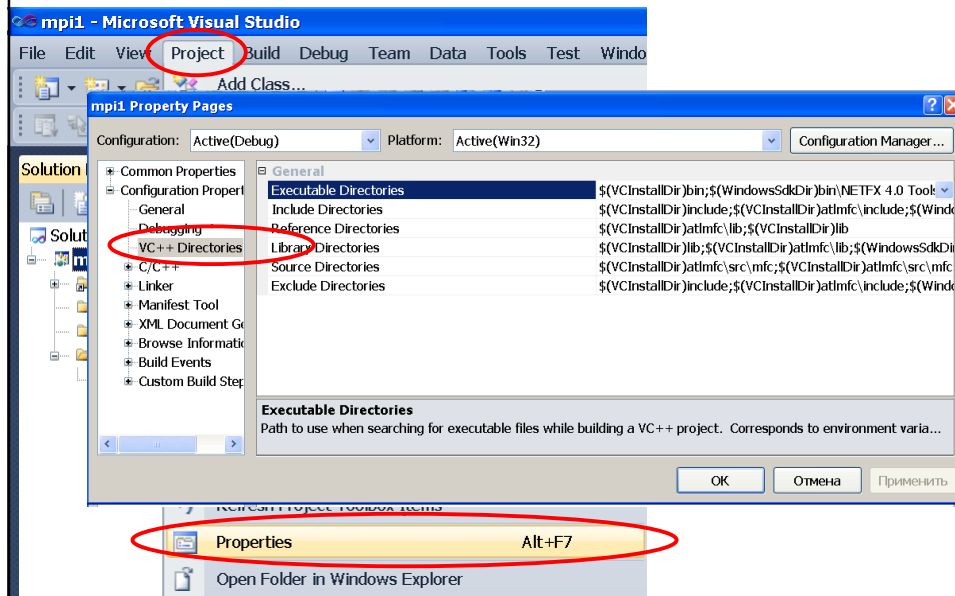
- Вычислительные узлы без компиляции, только для выполнения параллельной программы (worker nodes)
- Скопировать файл smpd.exe
- Выполнить команду
 - smpd.exe -install
- Скопировать dll библиотеки
 - в каталог windows\system32
 - в каталог, из которого запускается MPI программа
- For example, if you have a directory called \\myserver\mysharedfolder and you have myapp.exe and *mpich2*.dll in that directory then you can execute this command: "mpiexec -n 4 \\myserver\mysharedfolder\myapp.exe"
- Note: There are several mpich2 dlls and depending on your build target (Fortran, C, Debug or Release) you will need the corresponding dlls in the application directory.

- To specify a per-project directory list
- Project menu
- click Properties
- Configuration Properties
- VC++ Directories
- To edit one of the directory lists,
 - click its name
 - click the arrow that appears
 - click Edit
- You can add or remove values, and you can reorder any values that have been added. You can also select or clear Inherit from parent or project defaults.

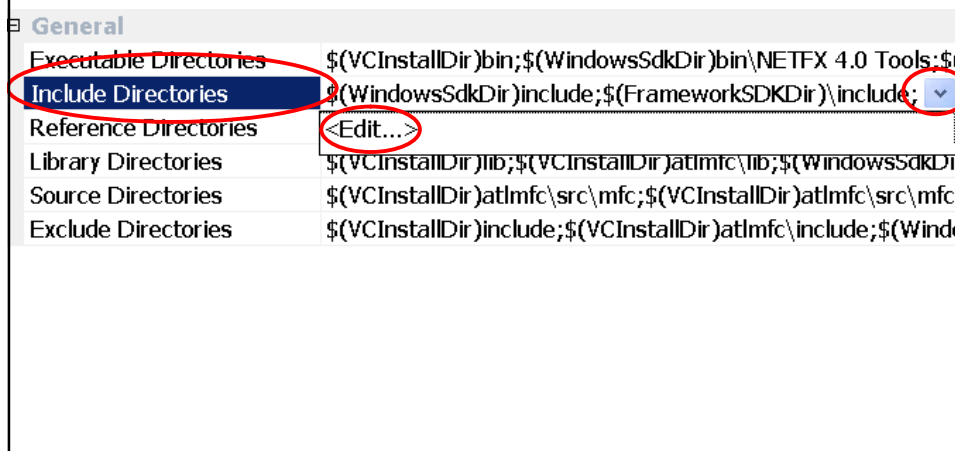
Создаем проект



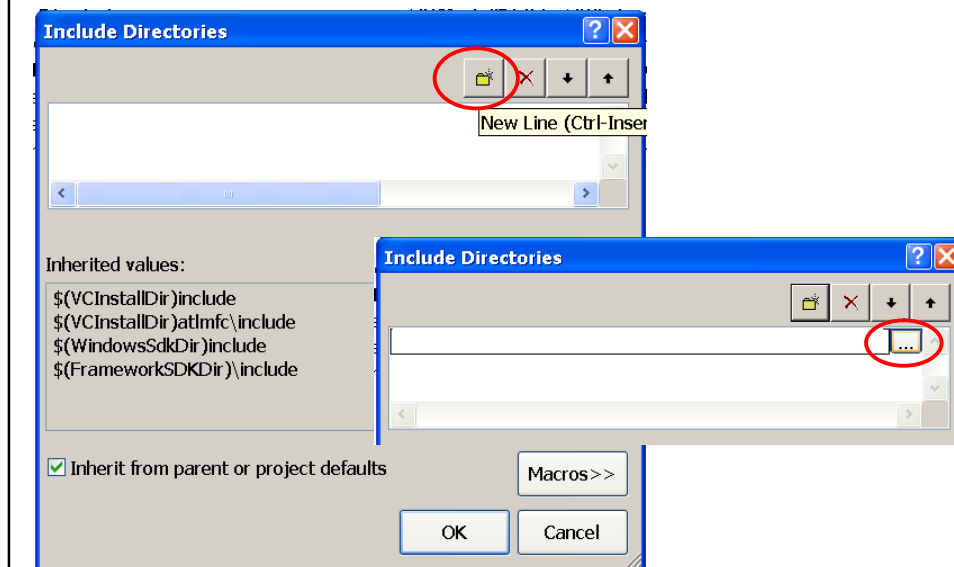
Project ► Properties ► VC++ Directories



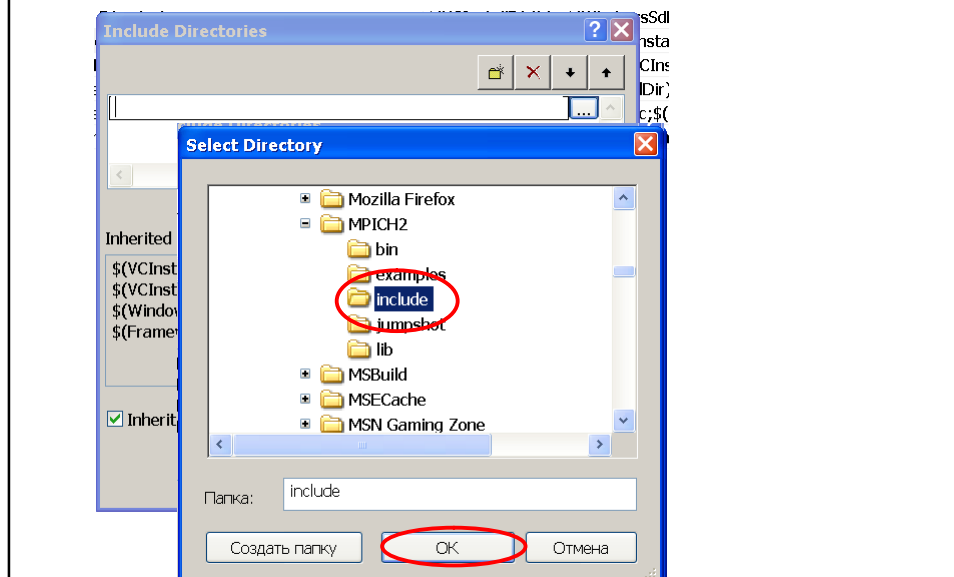
Include Directories ► Список ► Edit



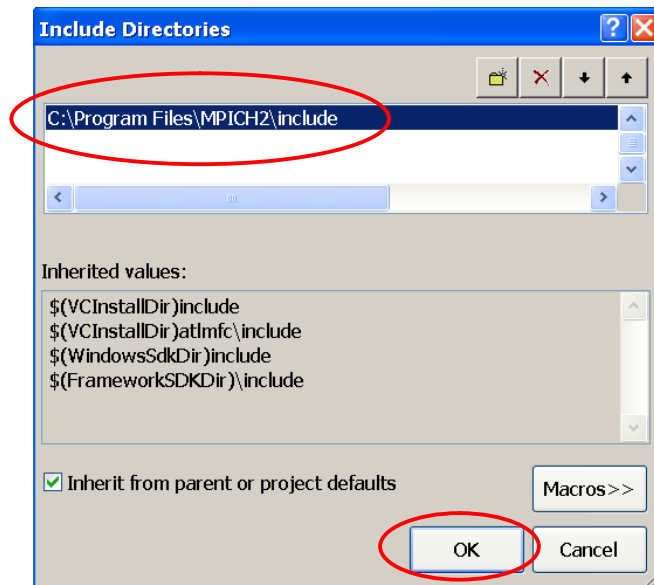
Include Directories ► New Line ► ...



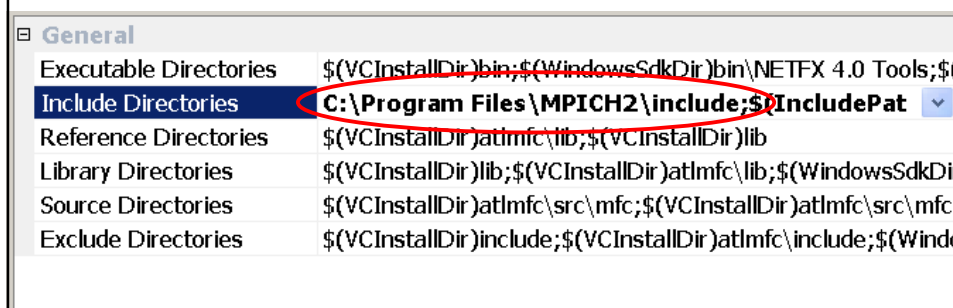
Select Directory ► C:\Program Files\MPICH2\include ► OK



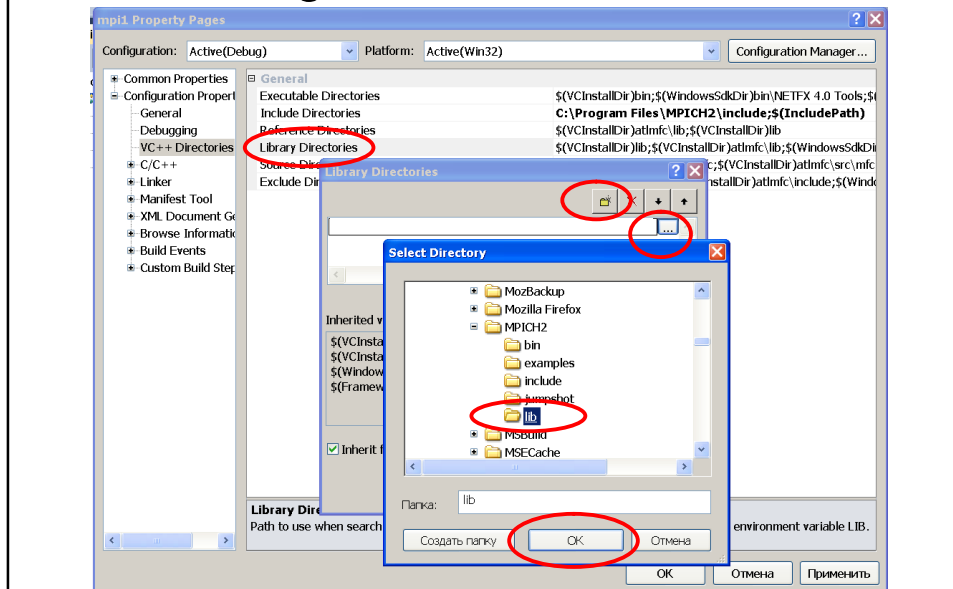
Include Directories ► OK



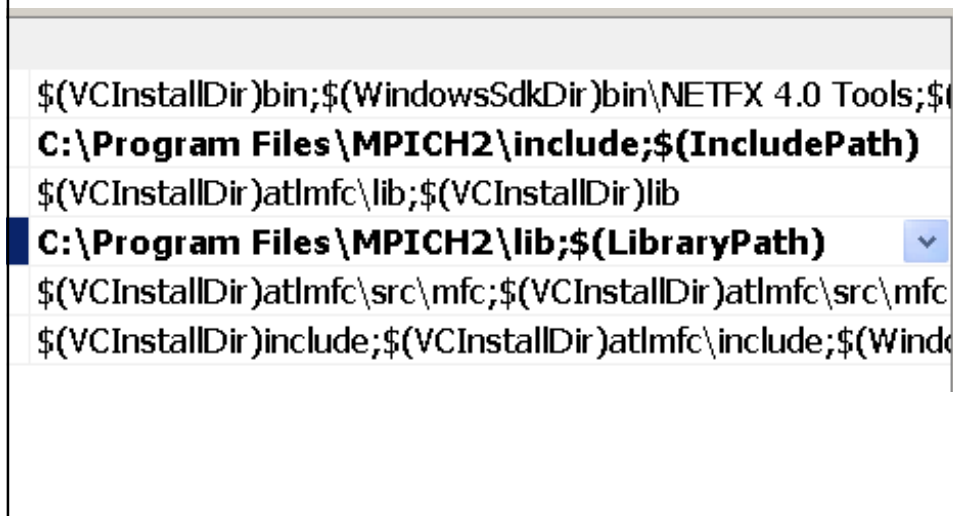
Каталог показан в списке



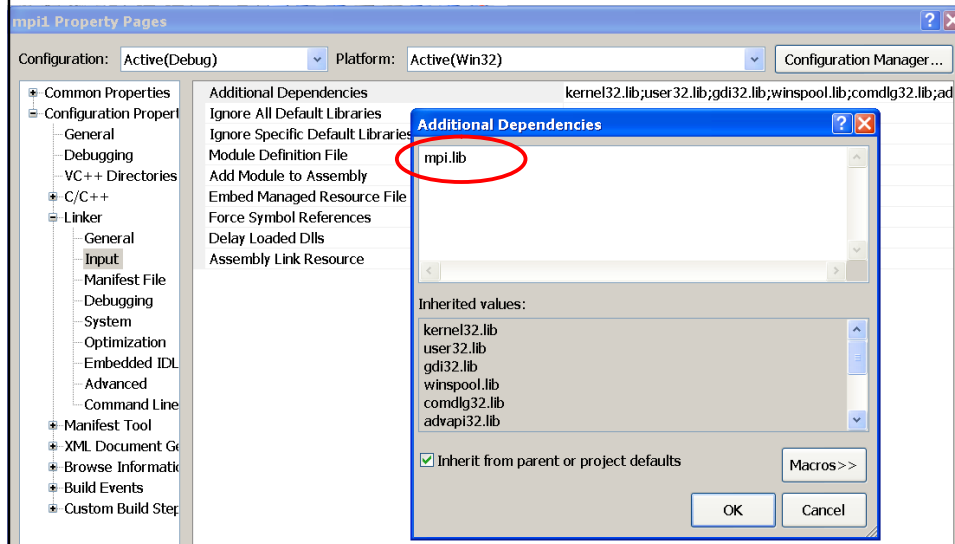
Library Directories ► C:\Program Files\MPICH2\lib



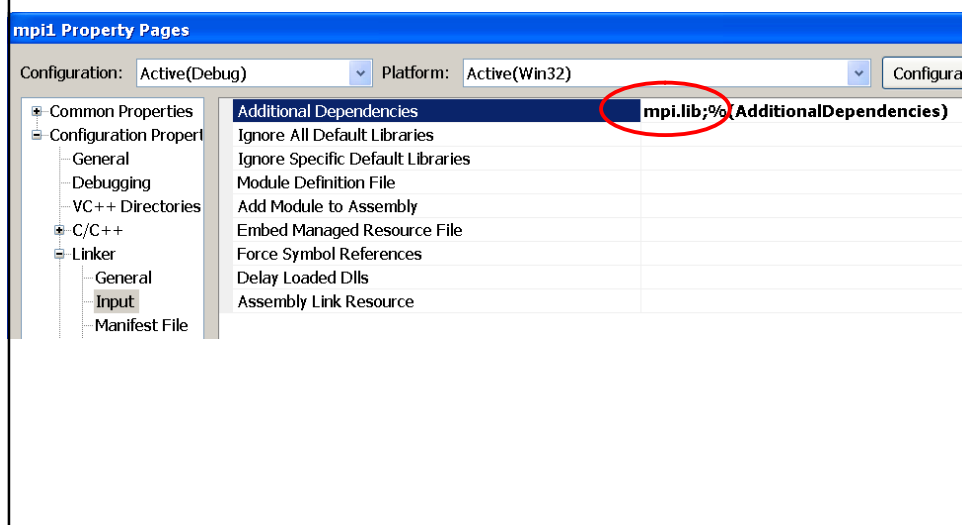
Проверяем каталоги MPICH2\include MPICH2\lib



Linker ► Input ► Additional Dependencies ► mpi.lib



mpi.lib



Debug / Release

- Повторить установку для конфигурации Release
 - Include Dir - MPICH2\include
 - Lib Dir - MPICH2\lib
 - Linker - mpi.lib

- Для работы smrd на всех компьютерах должны быть одинаковые логин и пароль пользователей
- Пароль не должен быть пустым

Запуск MPI приложения

- Запускаем CMD
- Переходим в рабочий каталог
- `mpiexec -n 3 myapp.exe`
- Для выполнения команды скопировать `mpiexec.exe` в рабочий каталог либо установить путь Path

Hello, World!

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

int main (int argc, char* argv[])
{
    int rank, size;
    MPI_Init (&argc, &argv);
    MPI_Comm_rank (MPI_COMM_WORLD, &rank);
    MPI_Comm_size (MPI_COMM_WORLD, &size);
    printf( "Hello world from process %d of
%d\n", rank, size );
    MPI_Finalize();
    return 0;
}
```

Запуск MPI

- Запуск параллельного приложения MPI
 - Одна и та же программа запускается несколько раз
- Запускается заданное число идентичных процессов
 - на локальной машине
 - на любой машине в сети
 - на машинах с заданными адресами

Коммуникатор

- Название группы процессов, которые могут обмениваться сообщениями
- По умолчанию, все процессы входят в коммуникатор
- **MPI_COMM_WORLD**

MPI_Init

```
MPI_Init (&argc, &argv);
```

- Инициализация
- Запуск системы обмена сообщениями MPI

MPI_Finalize

```
MPI_Finalize();
```

- Завершение
- Окончание работы системы обмена сообщениями MPI

MPI_Comm_size

- **MPI_Comm_size**
(MPI_COMM_WORLD, &size);
- Количество процессов в коммутикаторе

MPI_Comm_rank

MPI_Comm_rank
(MPI_COMM_WORLD, &rank)

Номер процесса в коммутикаторе

Можно поручить нулевому процессу
собирать результаты вычислений и
выводить на экран

Вывод на экран

- **printf**
- вызывается из любого процесса
- производится на экран машины, с которой запущено задание

MPI_Reduce

```
MPI_Reduce(&PartSum, &MyPi, 1,  
MPI_DOUBLE, MPI_SUM, 0,  
MPI_COMM_WORLD);
```

- Операция редукции
- Выполняется операция суммирования MPI_SUM над локальными переменными PartSum всех параллельных процессов
- Результат записывается в переменную MyPi процесса номер 0 в коммуникаторе MPI_COMM_WORLD

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\mpiuser>w:

W:\>cd 1

W:\1>dir
Том в устройстве W имеет метку Disk-1-W 400 GB
Серийный номер тома: 2032-3546

Содержимое папки W:\1

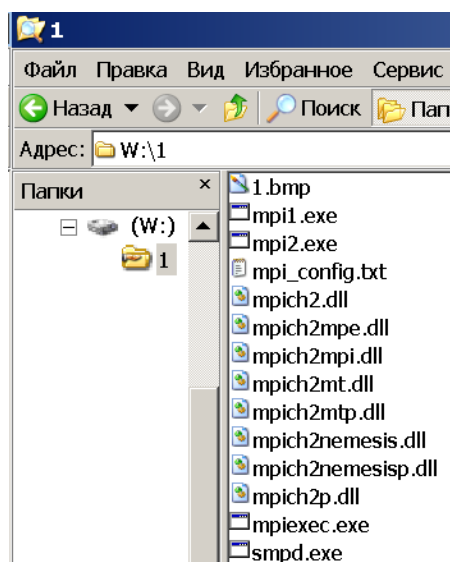
22.01.2012  15:09    <DIR>          .
22.01.2012  15:09    <DIR>          ..
22.01.2012  15:00             7a168 mpi1.exe
01.09.2011  15:25          491a520 mpiexec.exe
                2 файлов             498a688 байт
                2 папок          4a947a578a880 байт свободно

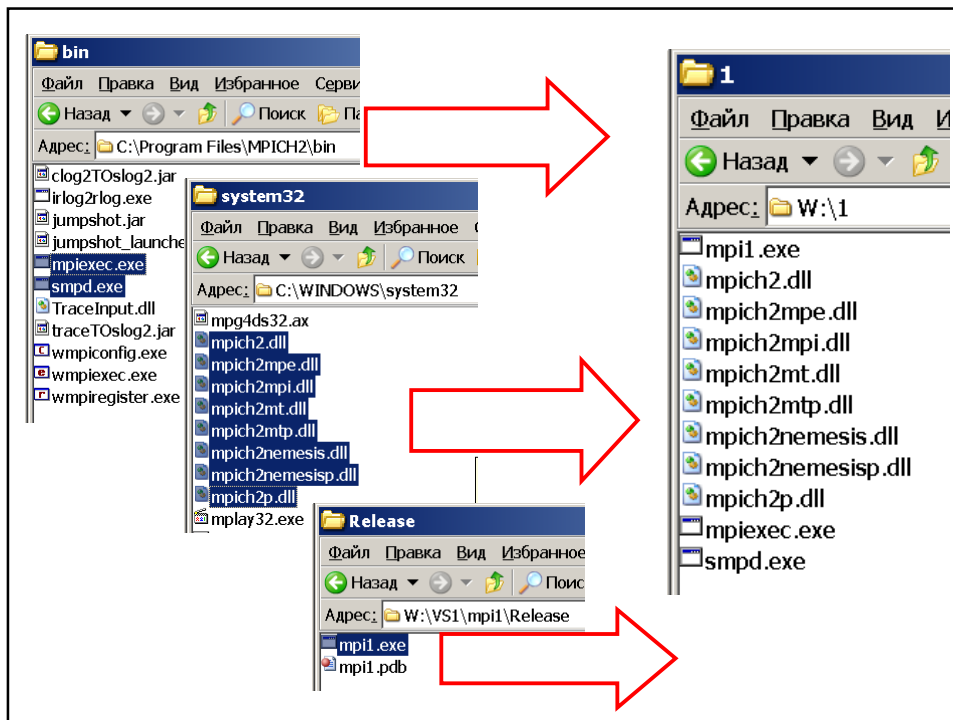
W:\1>mpiexec -n 3 mpi1.exe
User credentials needed to launch processes:
account (domain\user) [WALL\mpiuser]:
password:
Hello world from process 1 of 3
Hello world from process 2 of 3
Hello world from process 0 of 3

W:\1>_
```

Компиляция – Запуск

- Файлы скопировать в разделяемую сетевую папку





mpiexec.exe

Запуск 4 процессов на локальной машине

`mpiexec -np 4 mpi1.exe`

```

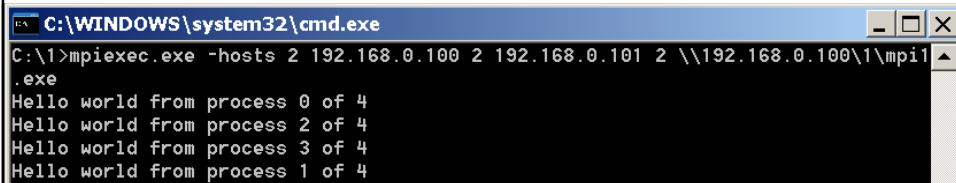
C:\WINDOWS\system32\cmd.exe

C:\1>mpiexec -np 4 mpi1.exe
Hello world from process 2 of 4
Hello world from process 3 of 4
Hello world from process 0 of 4
Hello world from process 1 of 4
  
```

mpiexec.exe

Запуск 2 процессов на 2 машинах с сетевого диска

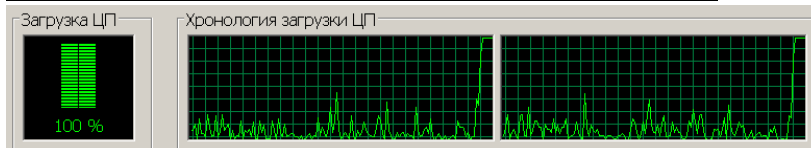
```
mpiexec.exe -hosts 2 192.168.0.100 2  
192.168.0.101 2  
\\192.168.0.100\1\mpi1.exe
```



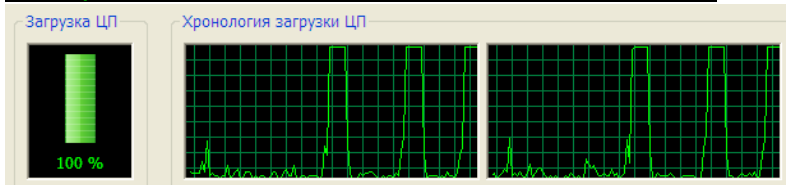
```
C:\WINDOWS\system32\cmd.exe  
C:\>mpiexec.exe -hosts 2 192.168.0.100 2 192.168.0.101 2 \\192.168.0.100\1\mpi1.exe  
Hello world from process 0 of 4  
Hello world from process 2 of 4  
Hello world from process 3 of 4  
Hello world from process 1 of 4
```

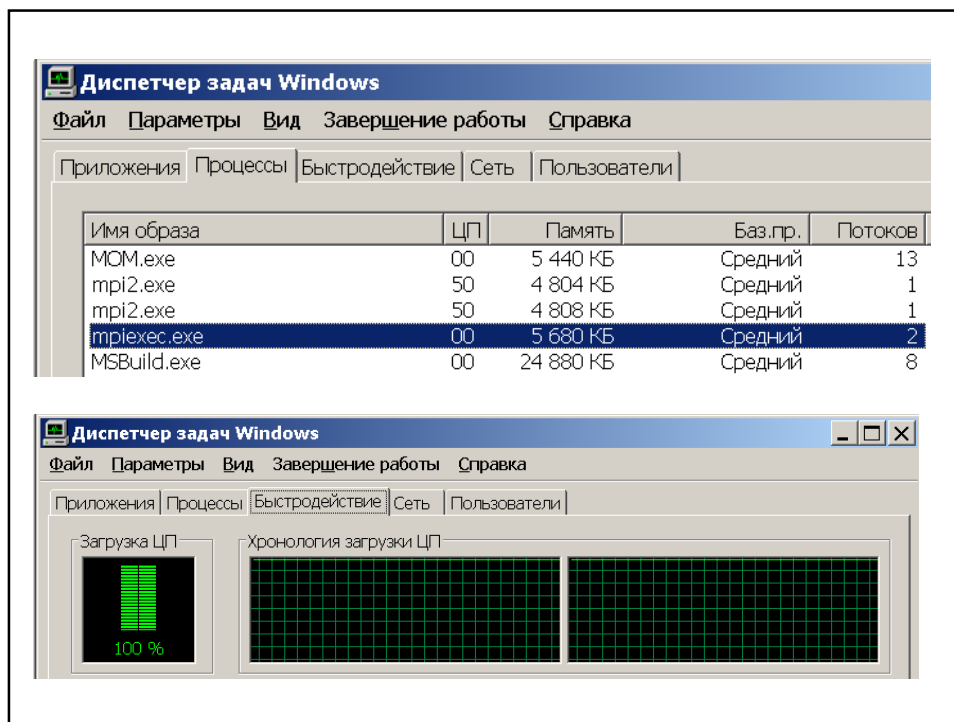
```
C:\>ccc  
  
C:\>mpiexec.exe -hosts 2 192.168.0.100 2 192.168.0.101 2 \\192.168.0.100\1\mpi2.exe  
Number of rectangles = 1000000000  
MyPi = 3.14159265358976820000  
Pi = 3.14159265358979310000  
Err = 0.00000000000079160472%  
MyTime = 8.421000 sec
```

IP-адрес : 192.168.0.100



IP-адрес : 192.168.0.101





Оценка числа Пи

- Интегрирование методом прямоугольников

$$\pi = \int_0^1 \frac{4}{1+x^2} dx$$

```
// Вычисление числа Пи: интеграл методом
// прямоугольников
#include <stdio.h>
#include "mpi.h"
#include <time.h>
#include <math.h>
#define NUMRECT 100000000
int main () {
    int ProcNum, ProcRank, i;
    double x, MyPi = 0;
    double PartSum = 0;
    double TotalTime;
    clock_t StartClock, EndClock;
    StartClock = clock ();
    MPI_Init( NULL, NULL );
    MPI_Comm_size (MPI_COMM_WORLD, &ProcNum);
    MPI_Comm_rank (MPI_COMM_WORLD, &ProcRank);
    double h = 1.0 / NUMRECT;
```

```
for (i = ProcRank; i < NUMRECT; i+= ProcNum)
{
    x = ( i + 0.5 ) * h;
    PartSum += 4.0 / ( 1 + x*x) ;
}
MPI_Reduce(&PartSum, &MyPi, 1, MPI_DOUBLE, MPI_SUM, 0,
MPI_COMM_WORLD);
MPI_Finalize ();
if (ProcRank == 0)
{
    MyPi *= h;
    EndClock = clock ();
    printf("Number of rectangles = %i\n", NUMRECT);
    printf("MyPi = %3.20f\n", MyPi);
    double pi = 3.1415926535897932384626433832795;
    printf(" Pi = %3.20f\n", pi);
    printf(" Err = %3.20f\n", abs(MyPi-pi));
    TotalTime = (double) ( EndClock - StartClock ) /
CLOCKS_PER_SEC;
    printf("MyTime = %f sec\n", TotalTime);
}
}
```