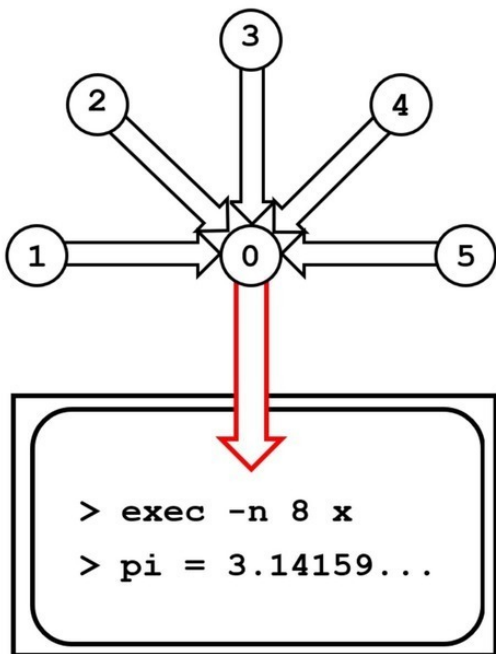


Валентин Юльевич Арьков
*Организация параллельных
процессов*

Учебное пособие



Валентин Юльевич Арьков
Организация параллельных
процессов. Учебное пособие

http://www.litres.ru/pages/biblio_book/?art=53661290

ISBN 9785449862044

Аннотация

В данной работе мы познакомимся с организацией высокопроизводительных вычислений на уровне параллельных процессов. При этом большую задачу приходится разделить на несколько частей, а выполнение каждой части работы поручить отдельному процессу. Первые шаги мы рассмотрим подробно, а потом отпустим читателя в свободное плавание.

Содержание

Введение	5
0. Организация работы	6
0.1. Отчёт	7
0.2. Варианты заданий	8
0.3. Среда разработки	9
1. Технология MPI	10
2. Установка библиотеки MPI	12
3. Программа «Всем привет!»	26
Конец ознакомительного фрагмента.	27

**Организация
параллельных процессов
Учебное пособие**

Валентин Юльевич Арьков

© Валентин Юльевич Арьков, 2020

ISBN 978-5-4498-6204-4

Создано в интеллектуальной издательской системе Ridero

Введение

В данной работе мы познакомимся с организацией высокопроизводительных вычислений на уровне параллельных процессов. При этом большую задачу приходится разделить на несколько частей, а выполнение каждой части работы поручить отдельному процессу.

Это будет самое первое знакомство на очень простых примерах.

Задание. Прочитайте в Википедии статью «Процесс (информатика)» и выясните, что такое ПРОЦЕСС и как он связан с ПРОГРАММОЙ.

Параллельные процессы могут выполняться на одном локальном персональном компьютере, или на другом (удалённом) персональном компьютере, или на нескольких отдельных компьютерах, или на большом вычислительном кластере, который состоит из некоторого количества самостоятельных материнских плат, объединённых скоростной локальной сетью.

Задание. Прочитайте в Википедии статью «Кластер (группа компьютеров)» и найдите определение КЛАСТЕРА и виды кластеров.

0. Организация работы

Для серьёзного изучения предмета параллельных вычислений можно найти несколько книг на русском языке [1 – 9]. Перед этим имеет смысл получить общее представление об изучаемой технологии. Разобрать какой-нибудь сквозной пример с пошаговым руководством. На западе это называется TUTORIAL. Мы добавили к такому практическому подходу простенькие задания для расширения кругозора и общего понимания материала.

Данная работа продолжает серию работ по параллельному программированию [10, 11].

В работе [10] мы познакомились со стандартными методами работы с потоками в операционной системе **Microsoft Windows**.

Далее в работе [11] мы освоили основы работы с параллельными потоками по технологии **OpenMP**. Теперь пришло время для параллельных процессов по технологии **MPI**. Оформление отчёта мы подробно описали в работе [12]. Анализ данных с помощью сводных таблиц мы подробно рассмотрели в работе [13].

Задание. Скачайте пособия [10 – 13] по ссылкам в списке литературы.

0.1. Отчёт

Отчёт будем оформлять в виде многостраничного файла **Microsoft Excel**. Напомним, что студентам нашего вуза предоставляется лицензия на установку **Microsoft Office 365**. Подробности на сайте:

<https://www.ugatu.su/>

Желающие могут использовать любой другой пакет типа электронных таблиц, например, **Libre Office**, **Таблицы Гугл** или что угодно ещё.

Страницы отчёта нумеруем следующим образом: **01**, **02**, **03** и так далее. Номер страницы указываем на вкладке очередного листа внизу окна **Excel**.

Первая страница отчёта – титульный лист со всеми необходимыми сведениями.

Вторая страница отчёта – оглавление со ссылками на все страницы документа.

Далее идёт сам отчёт. Все подробности и инструкции – в работе [12].

Задание. Создайте файл отчёта, заполните титульный лист и сохраните под коротким информативным названием.

0.2. Варианты заданий

Во всех работах мы используем одно и то же задание – параллельное численное интегрирование. Номер варианта задания – последняя цифра номера зачётки. Если последняя цифра ноль, берём вариант **10**. Нулевой вариант используется только для демонстрации технологии и разбора сквозного примера.

Задание. Определите номер своего варианта и укажите его на титульном листе отчёта.

Исходные данные для численного интегрирования мы используем одни и те же во всей серии работ. Это нелинейная функция с индивидуальным набором коэффициентов и пределы интегрирования. Таблица с вариантами заданий есть в предыдущем пособии [11].

Задание. Опишите свой вариант задания в отчёте на странице 03, включая точное аналитическое решение задачи.

0.3. Среда разработки

В данной работе мы используем среду разработки **Visual Studio Community Edition**. Она бесплатна для учебных и некоммерческих целей. Мы будем использовать компилятор языка **Си**. Процесс установки и компиляции мы описали в работе [10].

Желающие могут использовать любой другой компилятор языка **Си**. Главное, чтобы компилятор поддерживал технологию **MPI**.

Задание. Запустите среду разработки и убедитесь в её работоспособности.

1. Технология MPI

В предыдущей работе мы рассмотрели технологию **OpenMP**. В данной работе нам предстоит познакомиться с одной из наиболее распространённых технологий распараллеливания под названием **MPI**.

Задание. Прочитайте в Википедии начало статьи «Message Passing Interface» и выясните, на какие платформы ориентированы технологии **MPI** и **OpenMP**.

Можно сказать, что **MPI** – это только общий стандарт. Для написания программы нам потребуются соответствующие средства программирования (библиотека). Это будет называться РЕАЛИЗАЦИЯ **MPI**.

Задание. Прочитайте в Википедии окончание статьи «Message Passing Interface» и выясните, сколько реализаций **MPI** существует и для каких операционных систем.

Для организации совместной работы потребуется наладить общение между вычислительными процессами. В технологии **MPI** это коммуникация в форме ОБМЕНА СООБЩЕНИЯМИ.

Задание. Прочитайте в Википедии статью «Межпроцессное взаимодействие» и выясните, какие механизмы взаимодействия процессов может обеспечить операционная систе-

ma.

2. Установка библиотеки MPI

В данной работе мы рассмотрим работу в среде **Microsoft Visual Studio Community Edition** – бесплатной для учебных и некоммерческих целей. В предыдущих работах мы рассмотрели процедуру установки **Visual Studio** [10]. Желающие могут использовать любой другой компилятор с поддержкой **MPI**.

Компилятор позволяет организовать вызовы библиотечных функций **MPI**. Но для этого потребуется установить библиотеку **MPI**. В данной работе мы рассмотрим работу с библиотекой **MPICH**. Это одна из реализаций стандарта **MPI**. Желающие могут использовать любую другую библиотеку **MPI**.

Задание. Прочитайте в Википедии статью «**MPICH**» и выясните, какие известные компании участвовали в разработке и как расшифровывается название библиотеки.

Процедура установки библиотеки **MPI** в общих чертах демонстрируется в ролике [15]. Далее мы разберём каждый шаг установки более подробно.

Задание. Посмотрите видеоролик [15].

Нам нужно скачать и установить библиотеку **MPICH2**. Переходим на сайт проекта:

<http://www.mpich.org/>

Переходим в раздел **Downloads**.

Находим ссылку на предыдущие выпуски:
Older releases are available here (рис.2.1).

MPICH High-Performance Portable MPI

Home About **Downloads** Documentation Support ABI Compatibility Initiative Supported Compilers

Downloads

MPICH is distributed under a [BSD-like license](#). NOTE: MPICH binary packages are available in many UNIX distributions and for Windows. For example, you can search for it using "yum" (on Fedora), "apt" (Debian/Ubuntu), "pkg_add" (FreeBSD) or "port"/"brew" (Mac OS). If available for your platform, this is likely the easiest installation method since it automatically checks for dependency packages and installs them. Otherwise you can use the [installation guide](#) for installing MPICH from the source code below.

Release	Platform	Download	Size
mpich-3.3.2 (stable release)	MPICH	[http]	26 MB
hydra-3.3.2 (stable release)	Hydra (mpixec)	[http]	4 MB
mpich-3.4a2 (alpha release)	MPICH	[http]	26 MB
hydra-3.4a2 (alpha release)	Hydra (mpixec)	[http]	4 MB

Older releases are available [here](#). Nightly snapshots are available [here](#). MPE releases are available [here](#).

Рис. 2.1. Сайт проекта

Переходим по указанной ссылке и получаем список разных версий (рис. 2.2).

Index of /static/downloads

Name	Last modified	Size	Description
 Parent Directory			-
 0.9.1/	13-Nov-2016 10:20		-
 0.9.2/	13-Nov-2016 10:20		-
 0.9.3/	13-Nov-2016 10:20		-

Рис. 2.2. Версии библиотеки

При первом знакомстве с библиотекой начнём с чего-нибудь простого. Последние версии предлагаются в виде исходных текстов. Мы будем искать предыдущие версии с готовой программой установки **MSI**. Но для начала разберёмся, что означает это слово из трёх букв.

Задание. Прочитайте в Википедии статью «Установщик Windows» и выясните, что такое **MSI**.

Нас интересует программа-установщик приложения для ОС **Microsoft Windows**.

Мало того, нас интересует 32-разрядная версия библиотеки.

Даже если на компьютере 64-разрядная операционная система, мы всё-таки поставим 32-разрядную версию. И в среде разработки мы будем разрабатывать 32-разрядные приложения **Win32**.

Конечно, желающие могут установить что-нибудь 64-раз-

рядное. Мы здесь просто разбираем сквозной работающий пример, на базе которого можно развить свои собственные идеи – в любом направлении.

Итак, находим что-нибудь 32-разрядное (рис. 2.3).

В названии файла видим знакомые ключевые слова:

- Win;
- 32;
- msi.



The screenshot shows a web browser window with the address bar containing 'www.mpich.org/static/downloads/1.0.1/'. Below the address bar is the title 'Index of /static/downloads/1.0.1'. A table lists the files in the directory. The first row is 'Parent Directory'. The second row, 'mpich2-1.0.1-1-win32-ia32.msi', is highlighted with a red box. The third row is 'mpich2-1.0.1-1-win64-x86-64.zip' and the fourth row is 'mpich2-1.0.1.tar.gz'.

Name	Last modified	Size	Description
 Parent Directory			-
 mpich2-1.0.1-1-win32-ia32.msi	14-Apr-2013 21:03	12M	
 mpich2-1.0.1-1-win64-x86-64.zip	24-Apr-2013 21:03	6.1M	
 mpich2-1.0.1.tar.gz	24-Apr-2013 21:03	12M	

Рис. 2.3. Выбор 32-разрядного установщика

Задание. Найдите и скачайте 32-разрядный установщик формата MSI.

Скачиваем установщик и запускаем его (рис. 2.4).

Может потребоваться дополнительное согласие на установку, если **Windows 10** будет сомневаться в чистоте наших намерений и задавать разные вопросы.

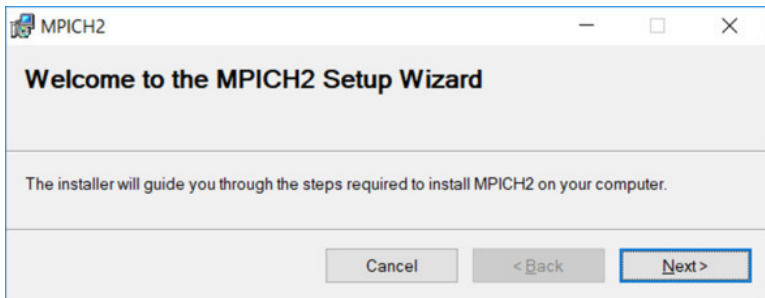


Рис. 2.4. Начало установки

Нажимаем **Next**. Появляется привычное окно установки программы (рис. 2.5).

Но не будем торопиться и щёлкать по кнопкам **ОК – Next – Next – ОК**.

В данном случае можно узнать много полезного, если пролистать текст и попытаться понять написанное (см. задание ниже).

Только после этого можно с облегчением вздохнуть и нажать-таки **Next**.

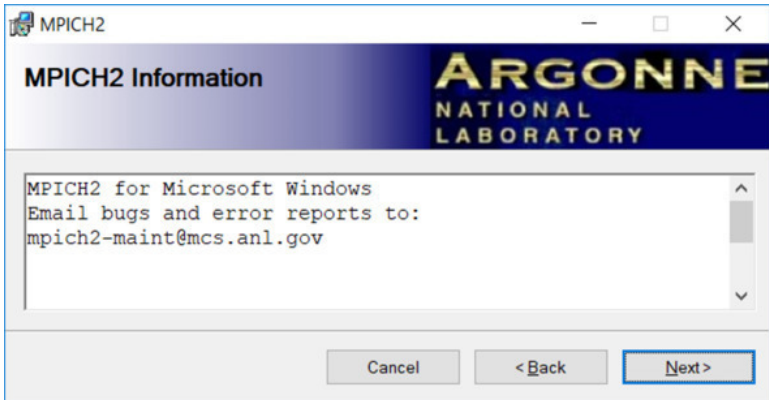


Рис. 2.5. Окно установщика

Задание. Просмотрите текст в окне установщика и выясните следующее:

- Какие компиляторы C/C++ поддерживает данная библиотека?
- Какие каталоги будут созданы?
- Где будут размещены библиотеки DLL?
- Какие настройки нужно сделать в компиляторе?
- Что нужно сделать при запуске программы на нескольких машинах?
- Как запустить параллельное приложение?
- В чём заключается минимальная установка?
- Как выполнить запуск из сетевого каталога?

В следующем окне появляется лицензионное соглашение

(рис. 2.6).

Такой текст мало кто дочитывает до конца, а зря.

Здесь тоже можно кое-что для себя почерпнуть (см. задание).

Только после выполнения задания выбираем согласие с условиями **I agree** и нажимаем кнопку **Next**.



Рис. 2.6. Лицензионное соглашение

Задание. Изучите лицензионное соглашение и выясните следующее:

- Что нам разрешается делать с данной библиотекой?
- От каких претензий отказываются разработчики в разделе **Disclaimer**?

На следующем нам предлагают установить пароль **Pass Phrase** (рис. 2.7).

При этом пояснения занимают аж четыре строчки. И это не просто слова, а полезная информация.

Здесь нам сообщают несколько важных моментов:

Установку нужно делать с правами администратора.

Будет установлена системная служба (service) под названием **SMPD**. Она нужна для работы параллельного приложения **MP1**.

Пароль должен быть одинаковым на всех компьютерах, на которых будем запускать нашу параллельную программу.

Другими словами, мы запускаем одну программу, а выполняются несколько процессов на нескольких компьютерах. Но работать они будут как одна большая программа. И мы это проделаем.

Пароль запомним на будущее. А лучше запишем на стикере и приклеим на монитор. Так обычно поступают начинающие пользователи. Так они надеются обеспечить безопасность компьютерной системы.

Чтобы легче было запомнить этот сложный пароль, можно послушать популярную песенку «Don't Worry, Be Happy» и почитать её историю где-нибудь в сети.

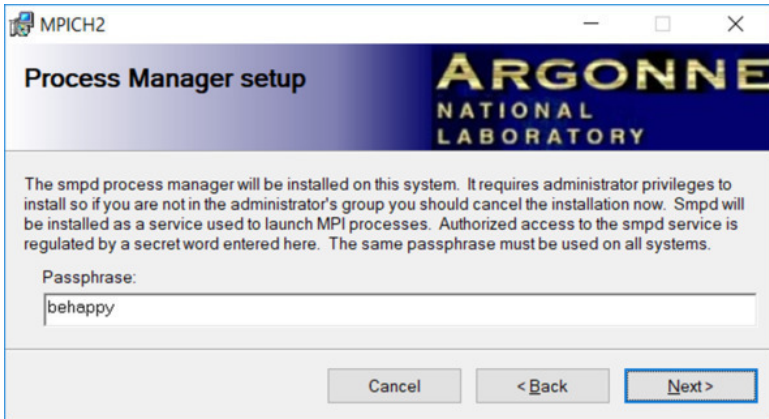


Рис. 2.7. Задание пароля

Задание. Установите и запишите пароль.

Судя по непроверенным сообщениям из достоверных источников в интернете, **SMPD** расшифровывается как **Simple Multi-Purpose Daemon**. Они же утверждают, что эта служба относится к средствам управления процессами – **Process managers**. Эта часть названия более-менее понятна.

Название заканчивается словом **Daemon**. Интересно, про каких таких «демонов» здесь идёт речь? Программисты обожают образные названия. А ещё в компьютерах встречаются «сироты», «зомби» и другие явления природы.

Задание. Прочитайте в Википедии статью «Демон (программа)» и выясните, что компьютерщики называют словом

«демон» и как появился этот термин.

Нажимаем кнопку **Next**.

Получаем новое диалоговое окно (рис. 2.8).

Мы выбираем каталог для установки библиотеки.

Обратим внимание на выбор по умолчанию:

C:\Program Files (x86)\MPICH2

Это ещё одно подтверждение того, что мы устанавливаем именно 32-разрядную версию.

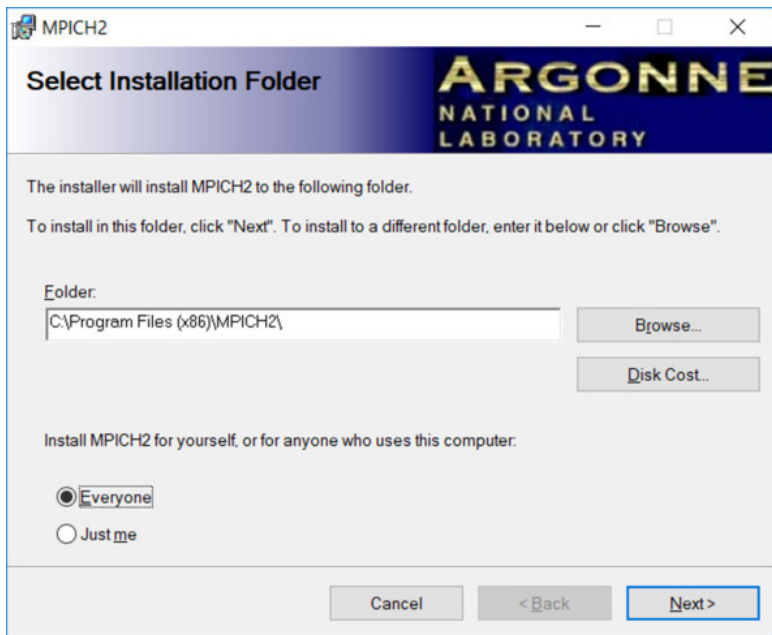


Рис. 2.8. Каталог для установки

Задание. Обратите внимание на каталог для установки и запишите его в отчёт.

Напомним, что нам желательно следить за разрядностью во всех её проявлениях:

- компьютер;
- операционная система;
- компилятор;
- библиотека;
- прикладная программа.

При этом на 64-разрядный компьютер можно установить 64-разрядную ОС. А прикладные программы могут быть и 32- и 64-разрядными.

Что бы всё окончательно запутать, прикладные программы устанавливаются в каталог **Program Files**.

Если у нас 64-разрядная ОС, в этот каталог будут установлены 64-разрядные приложения. А вот каталог **Program Files (x86)** предназначен для 32-разрядных программ (рис. 2.9).

При этом в 32-разрядной ОС программы 32-разрядные, и они устанавливаются в каталог **Program Files**.

В нашем примере (рис. 2.9) можно видеть, что компьютер и ОС 64-разрядные, а каталоги есть для обоих типов программ.

Device specifications

Processor	Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz	3.40 GHz
Installed RAM	16.0 GB (15.9 GB usable)	
System type	64-bit operating system, x64-based processor	

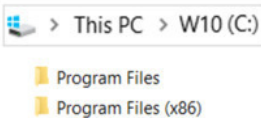


Рис. 2.9. Разрядность ОС и программ

Задание. Изучите свойства системы и каталоги для установки прикладных программ на своём компьютере.

Если говорить об установке приложений, **x86** прозрачно намекает на 32-разрядные программы. Эта традиция сложилась исторически. Дело было давно, когда невозможно было даже мечтать о 64 разрядах.

Задание. Прочитайте в Википедии статью «x86» и выясните, как появилось такое обозначение.

Нажимаем кнопки **Next – Next** и следим за процессом установки.

Читаем сообщение о завершении установки (рис. 2.10).

Похоже, в дальнейшем нам придётся установить ещё кое-что, но об этом потом.

Нажимаем кнопку **Close**.

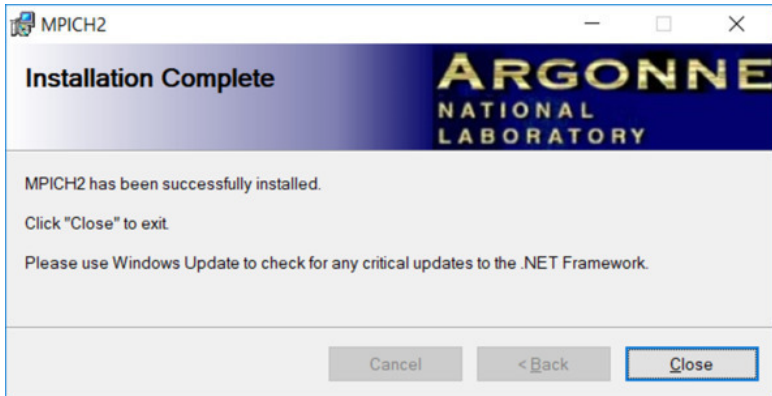


Рис. 2.10. Завершение установки

Задание. Установите библиотеку MPICH.

Рассмотрим содержимое каталога MPICH2 (рис. 2.11).

Нас будут интересовать три каталога:

- **bin**;
- **include**;
- **lib**.

Они нам понадобятся при настройке компилятора.

Ещё один полезный каталог называется **examples**. Судя по названию, здесь находятся примеры использования данной библиотеки. Примеры – во множественном числе. Угадайте с одного раза, сколько примеров можно найти в каталоге **examples**?

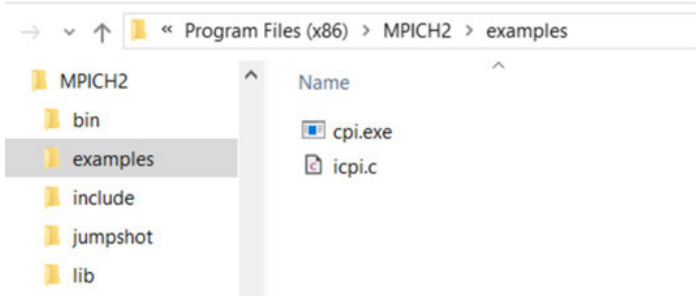


Рис. 2.11. Каталог MPICH2

Задание. Изучите содержимое каталога examples.

3. Программа «Всем привет!»

Начнём знакомство с MPI с самой простой, стандартной программы «Hello, World!»

Для начала посмотрим видеоролик [16], чтобы представить себе общую схему дальнейшей работы. В этом разделе мы подробно разберём все необходимые шаги.

Задание. Посмотрите ролик [16].

Текст программы приводится на рис. 3.1.

Главная часть программы – вывод на экран короткого стандартного приветствия с помощью команды **printf**

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.