|  |  |
| --- | --- |
| |  | | --- | | МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | |

Институт Информационных технологий

Кафедра МОСИТ

**ПРАКТИЧЕСКАЯ РАБОТА №1**

по дисциплине «Параллельное программирование»

**Студент группы** ИКБО-11-17 Алиев Ю. А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись студента)*

**Руководитель работы** Сыромятников В. П. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись руководителя)*

Москва, 2020

Оглавление

[Тема работы 2](#_Toc53946442)

[Цель работы 2](#_Toc53946443)

[Постановка задачи 2](#_Toc53946444)

[Теоретическое введение 3](#_Toc53946445)

[Ход выполнения задачи 1 5](#_Toc53946446)

[Ход выполнения задачи 2 6](#_Toc53946447)

[Ход выполнения задачи 3 7](#_Toc53946448)

[Ход выполнения задачи 4 8](#_Toc53946449)

[Ход выполнения задачи 5 9](#_Toc53946450)

[Ход выполнения задачи 5.1 9](#_Toc53946451)

[Ход выполнения задачи 5.2 10](#_Toc53946452)

[Ход выполнения задачи 5.3 11](#_Toc53946453)

[Ход выполнения задачи 6 12](#_Toc53946454)

[Ход выполнения задачи 7 14](#_Toc53946455)

[Ход выполнения задачи 9 15](#_Toc53946456)

[Выводы 17](#_Toc53946457)

[Список используемых информационных источников 18](#_Toc53946458)

# **Тема работы**

Установка и настройка интегрированной среды параллельного программирования C++ и OpenMP

# **Цель работы**

Формирование компетенции установки, тестирования и адаптации существующих пакетов программ на примере IDE Microsoft Visual Studio и API OpenMP.

# **Постановка задачи**

1. Провести обоснованный выбор версии (издания) IDE MVS на основе сопоставления технических характеристик компьютера и версии используемой операционной системы Windows с требованиями, предъявляемой IDE MVS.
2. Скопировать (загрузить) с официального сайта производителя интегрированной среды разработки MVS бесплатно распространяемую версию дистрибутива IDE MVS, например, [2] и произвести ее установку на свой компьютер. Ссылку на официальный сайт в обязательном порядке привести в списке используемых информационных источников.
3. Инсталлировать IDE MVS и настроить интегрированную среду параллельного программирования C++ и OpenMP. В случае, если IDE MVS уже установлена на компьютере, необходимо только расширить ее настройку подключением OpenMP.
4. Создать тестовый проект параллельного приложения для работы с библиотекой OpenMP, который выводит на экран монитора сообщение «Hello world! \n».
5. Модифицировать текст исходного кода теста, установив заданное количество параллельных потоков тремя разными способами с помощью:

* Переменной окружения OMP\_NUM\_THREADS,
* Опции num\_threads() директивы #pragma omp parallel,
* Функции set\_num\_threads()

1. Определить приоритет каждого из перечисленных средств.
2. Определить максимальное количество параллельных потоков, которое может быть установлено на компьютере.
3. Сделать выводы по полученным результатам и оформить отчет по выполненной работе.
4. Привести характеристики используемого компьютера и его процессора (в том числе, в обязательном порядке указать производителя, марку, тактовую частоту, количество ядер, наличие технологии hyper treading).

# **Теоретическое введение**

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

OpenMP (Open Multi-Processing) — открытый стандарт для распараллеливания программ на языках Си, Си++ и Фортран. Дает описание совокупности директив компилятора, библиотечных процедур и переменных окружения, которые предназначены для программирования многопоточных приложений на многопроцессорных системах с общей памятью.

Разработку спецификации OpenMP ведут несколько крупных производителей вычислительной техники и программного обеспечения, чья работа регулируется некоммерческой организацией, называемой OpenMP Architecture Review Board (ARB) [1]).

Первая версия появилась в 1997 году, предназначалась для языка Fortran. Для С/С++ версия разработана в 1998 году. В 2008 году вышла версия OpenMP 3.0. В июле 2014-го вышла версия 4.0

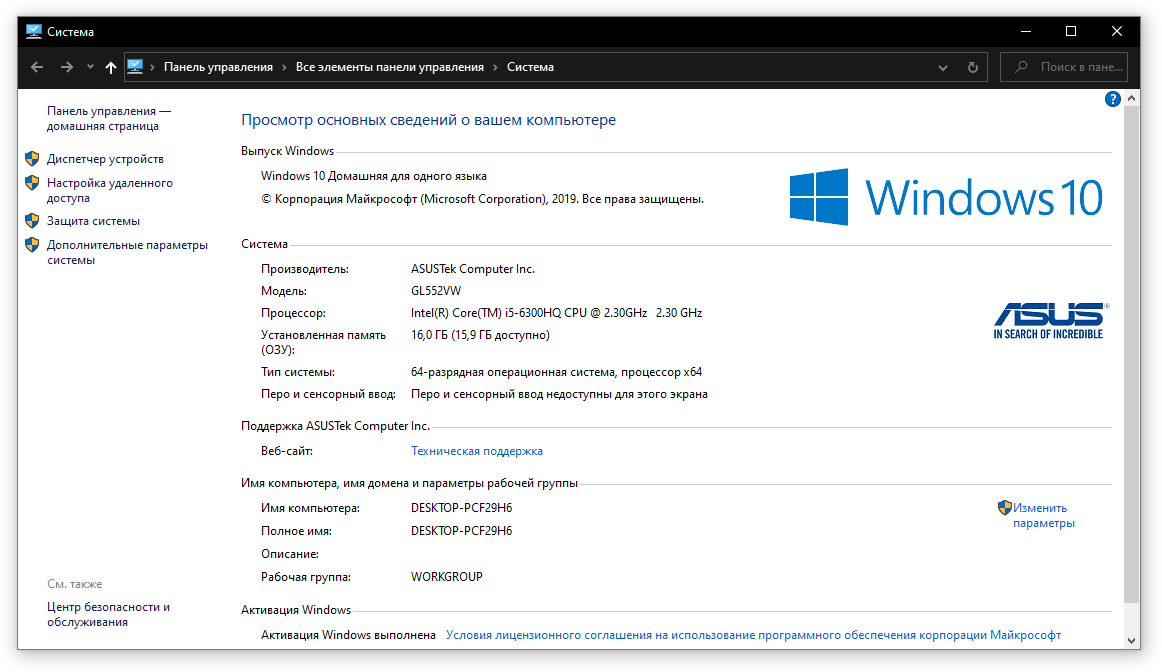
OpenMP реализует параллельные вычисления с помощью многопоточности, в которой «главный» (master) поток создает набор подчиненных (slave) потоков и задача распределяется между ними. Предполагается, что потоки выполняются параллельно на машине с несколькими процессорами (количество процессоров не обязательно должно быть больше или равно количеству потоков).

Задачи, выполняемые потоками параллельно, так же, как и данные, требуемые для выполнения этих задач, описываются с помощью специальных директив препроцессора соответствующего языка — прагм. Например, участок кода на языке Fortran, который должен исполняться несколькими потоками, каждый из которых имеет свою копию переменной N, предваряется следующей директивой:$OMP PARALLEL PRIVATE(N)

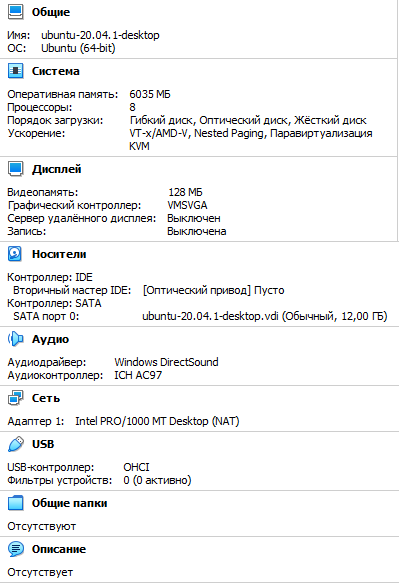
Количество создаваемых потоков может регулироваться как самой программой при помощи вызова библиотечных процедур, так и извне, при помощи переменных окружения.

# **Обосновать выбор**

Практическая работа выполняется на виртуальной машине в среде ОС Linux с дистрибутивом Ubuntu 20.04. Технические характеристики компьютера основываются на технических характеристик виртуальной машины, куда установлена ОС (рисунок 1).

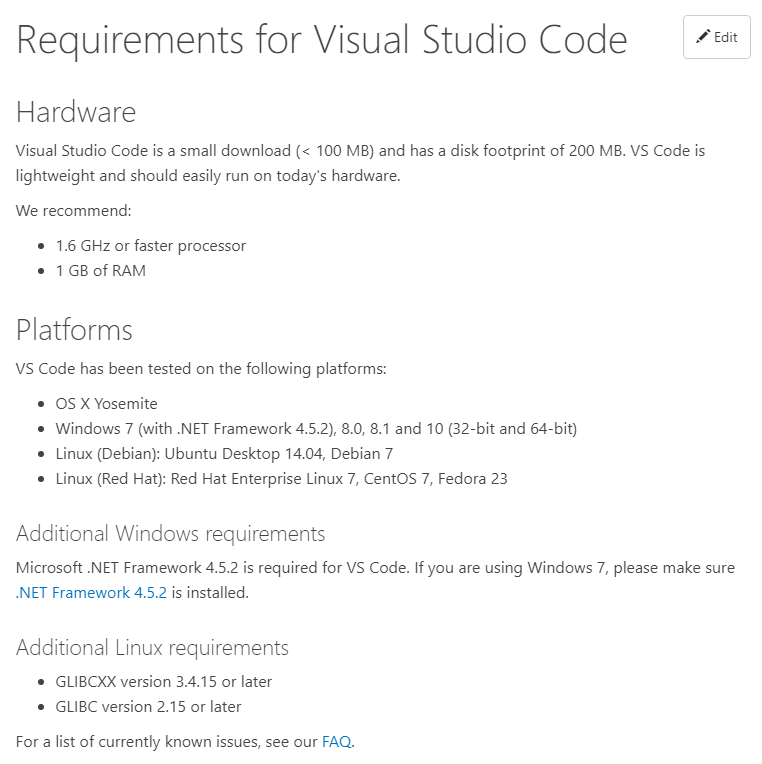


*Рисунок 1. Технические характеристики ПК, на котором установлена ОС.*



*Рис 2. Характеристики ПК (виртуальной машины), используемой при выполнении данной практической работы.*

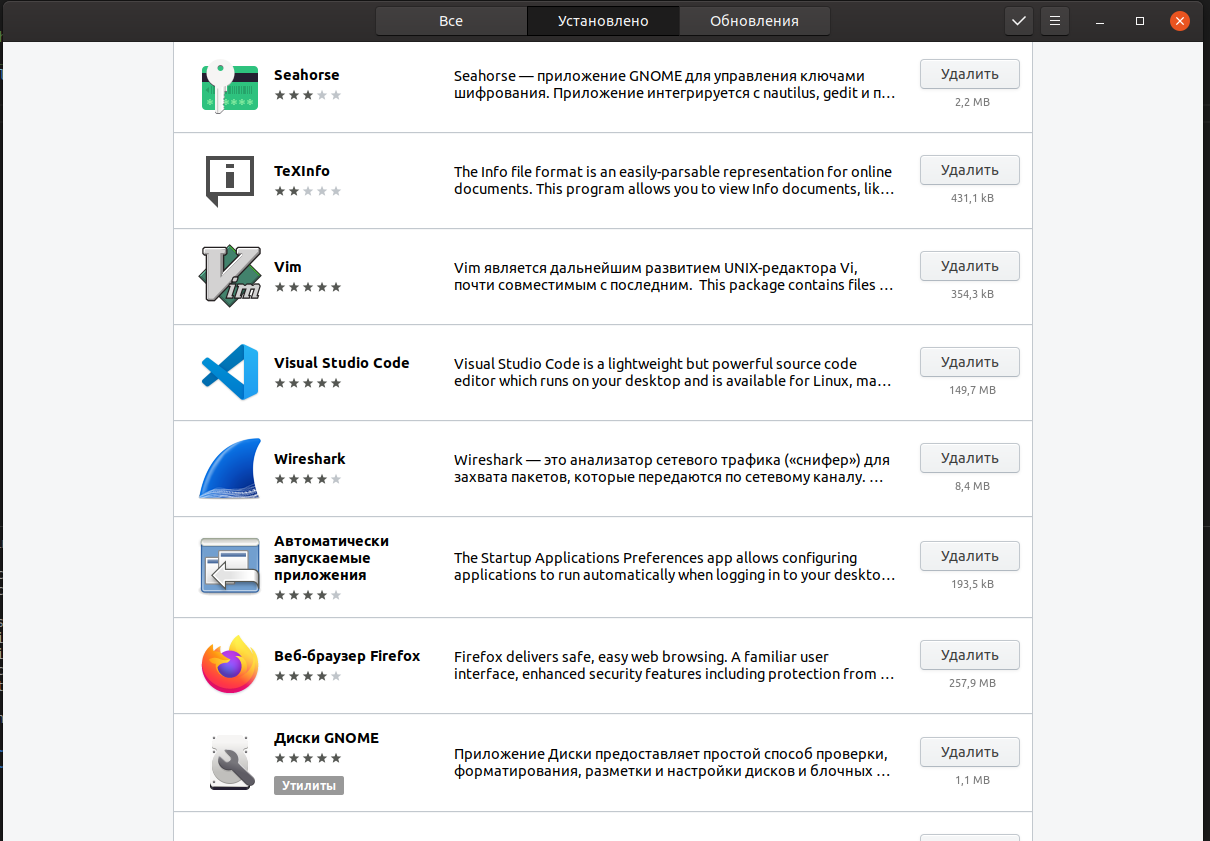
В качестве IDE для разработки будет использоваться Microsoft Visual Studio Code ver 1.50 с поддержкой интерпретатора кода C++ и установленным OpenMP. Системные требования IDE MVS Code представлены на рисунке 2.



*Рисунок 3. Системные требования MVS Code последней версии 1.50.*

# **Установка**

Так как используется ОС Linux с дистрибутивом Ubuntu 20.04, установка MVS Code производилась при помощи внутреннего магазина приложений (рисунок 3).



*Рис 4. Магазин приложений OC Linux Ubuntu 20.04.*

Так же можно установить с помощью команды представленной на официальном сайте VS Code

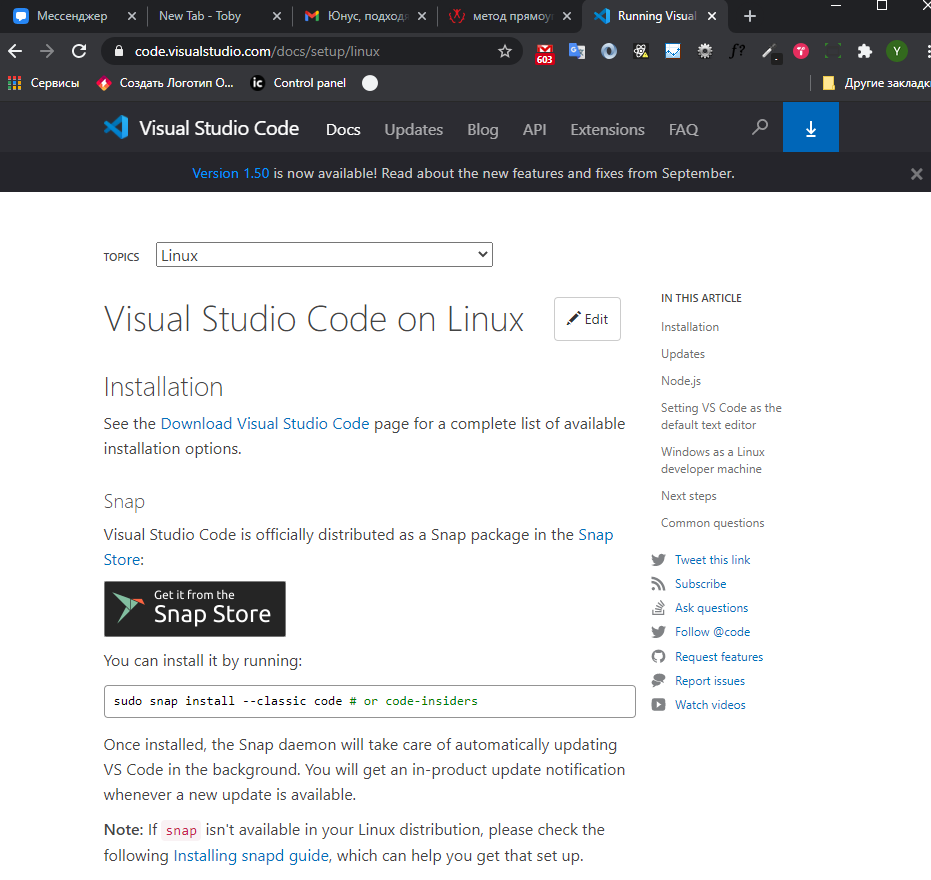
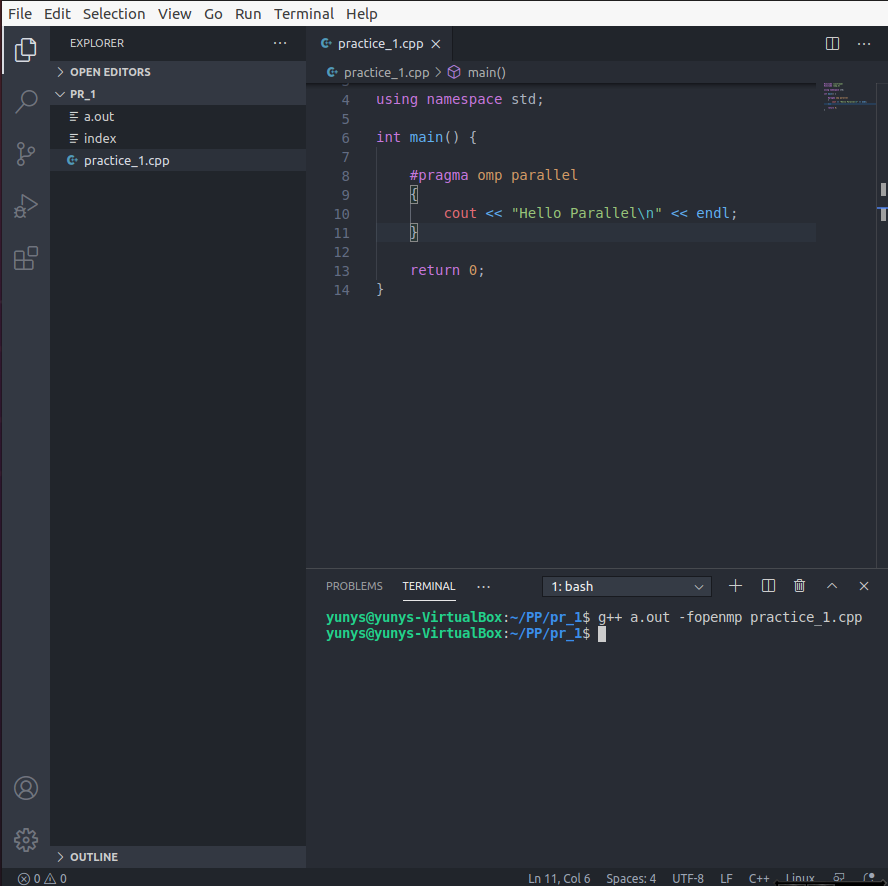


Рис 5 Установка с помощью команды через терминал OC Ubuntu

# **Подключение**

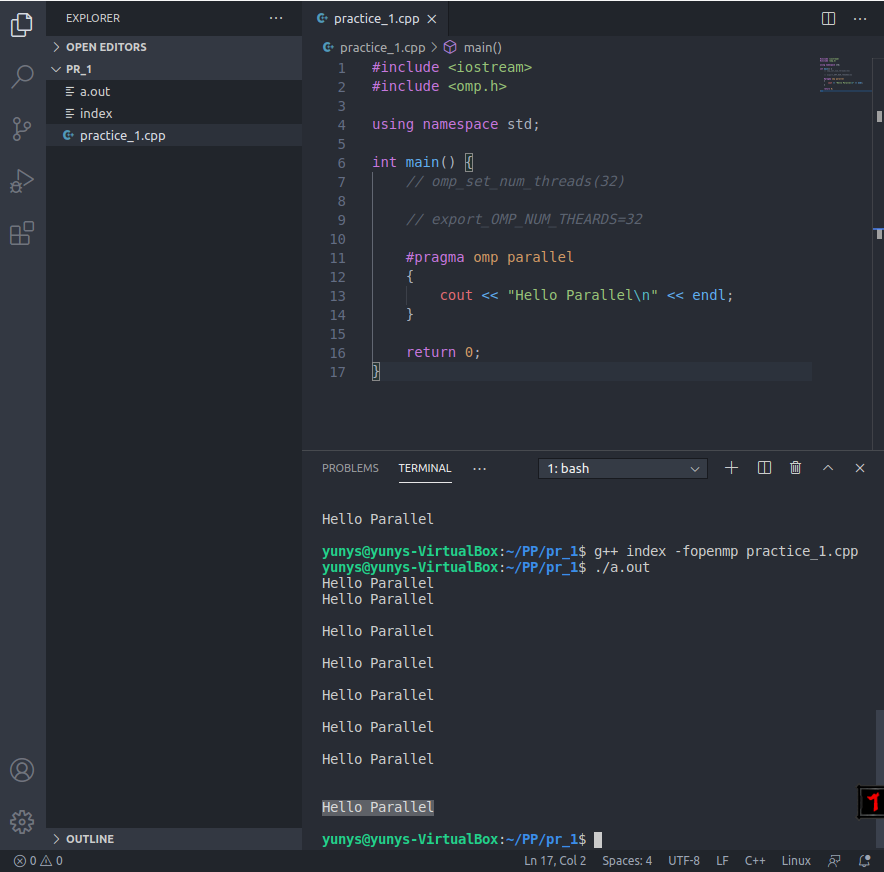
После установки MVS Code ver 1.50 при помощи встроенного терминала была создана директория выполнения для первой практической работы, а также главный файл practice.cpp (рисунок 4).



*Рис 6. Скриншот предварительная настройка MVS Code для выполнения последующих задач практической работы.*

# **Тестирование установки**

В файле practice.cpp необходимо написать код программы, а также скомпилировать исходный файл при помощи команды *g++ -o Имя выходного файла -fopenmp Имя компилируемого файла.* Результат проделанных действий предоставлен на рисунке 5.

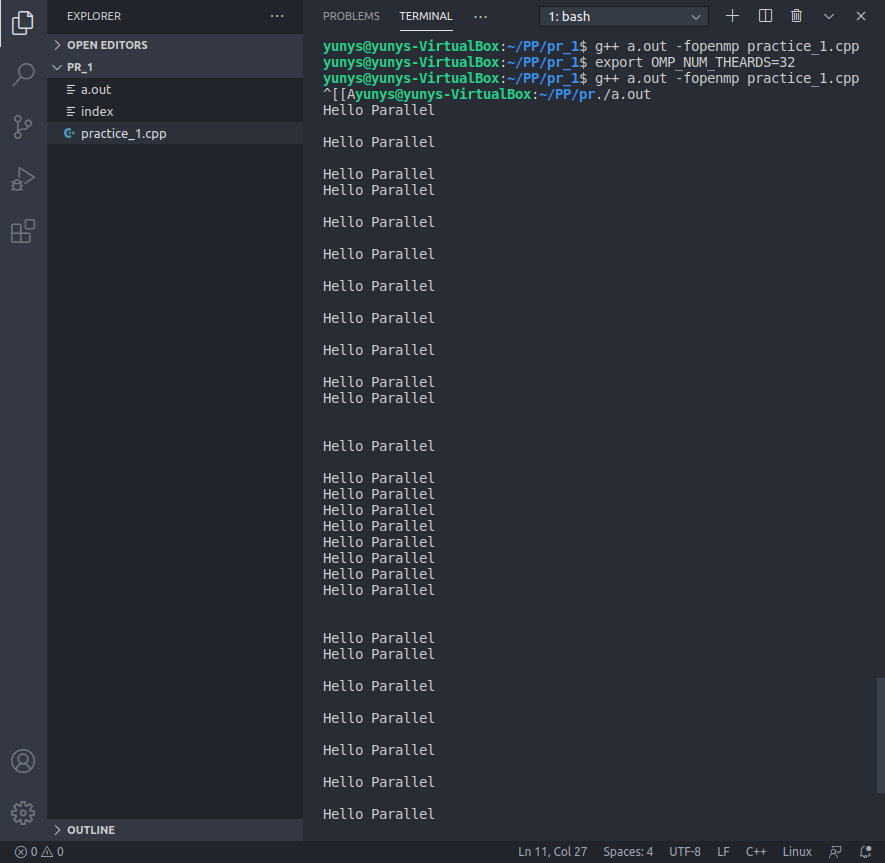


*Рис 7. Скриншот результат работы программы по пункту 4.*

# **Управление количество потоков**

## **Использование переменной окружения**

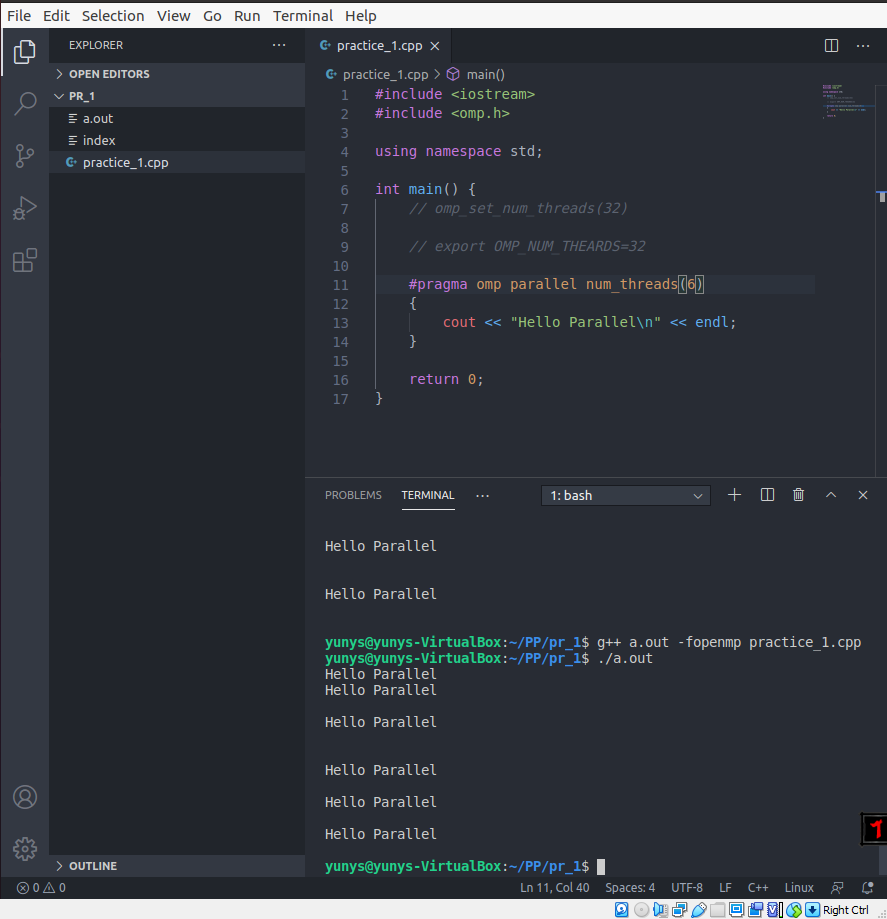
В Unix системах для определения переменной окружения OMP\_NUM\_THREADS достаточно написать export OMP\_NUM\_THREADS = кол-во ядер (в данном случае необходимо написать 32). После выполнения этих действий снопа компилируем программу и смотрим результат (рисунок 6).



*Рис 8. Скриншот результат работы программы по пункту 5.1*

## **Использовании опции**

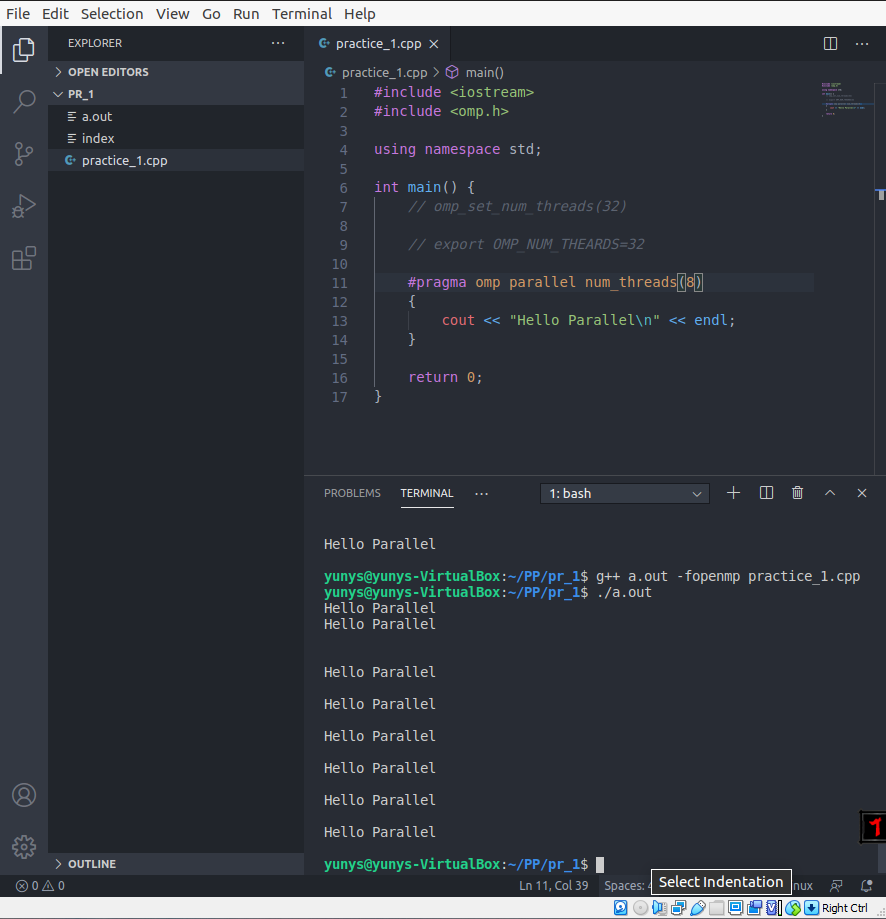
Для выполнения данного пункта задачи, достаточно после #pragma omp parallel написать опция num\_threads (число потоков). В данном примере я выставил 6 потоков. Результат выполнения программы представлен на рисунке 6.



*Рис 9. Скриншот результат работы программы по пункту 5.2*

## **Использование функции**

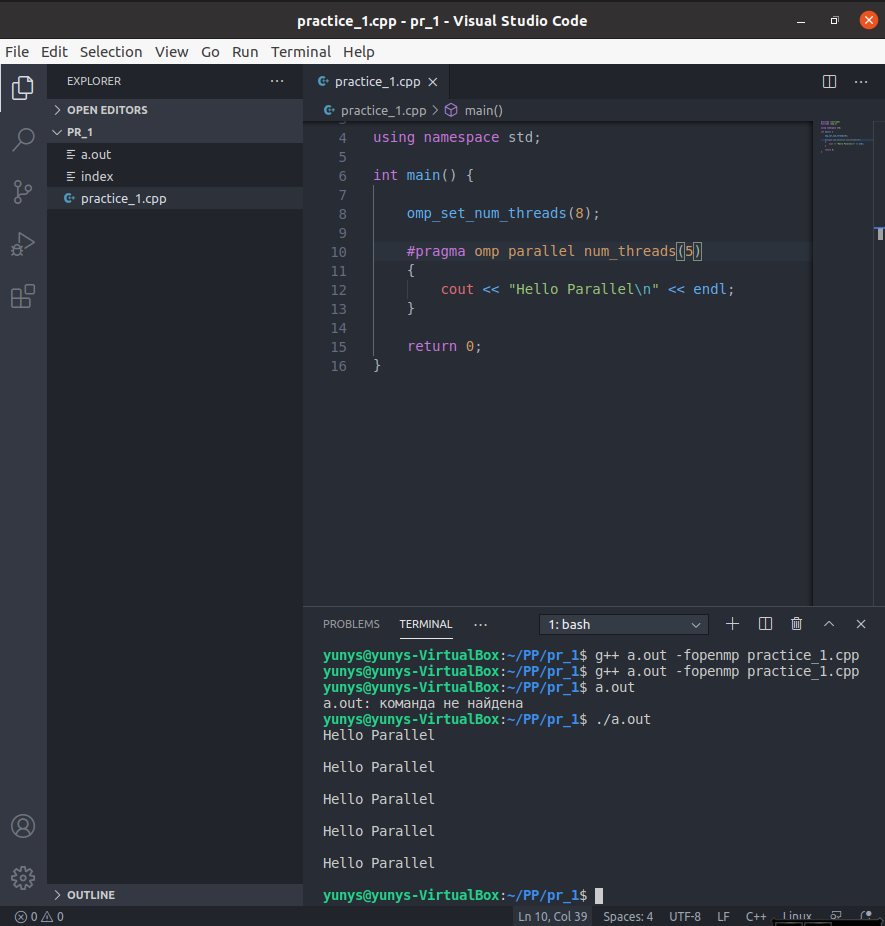
Для выполнения данного пункту задачи 5 достаточно перед строкой #pragma omp parallel написать функцию omp\_set\_num\_threads(количество потоков). В данном примере количество равно 8. Результат работы представлен на рисунке 7.



*Рис 10. Скриншот результат работы программы по пункту 5.3*

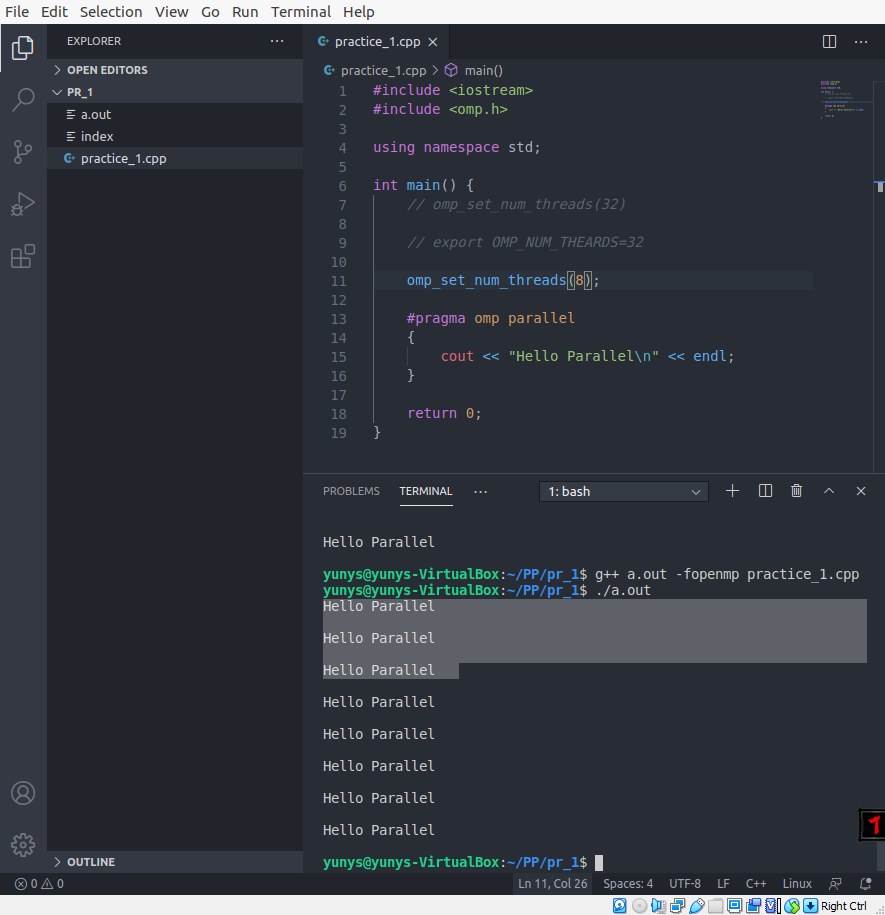
# **Выявить приоритет**

Для проверки приоритета проведем следующие действия: объявим изначально переменную окружения OMP\_NUM\_THREADS = 10, а в коде программы выставим функцию omp\_set\_num\_threads(8) и опцию в директиве num\_threads(5). Результат представлен на рисунке 8.



*Рисунок 11. Скриншот результат работы программы по пункту 6(1).*

Исходя из представленного рисунка, можем сделать вывод, что по приоритету выполнения вначале будет стоять опция директивы. Далее, уберем эту опцию и повторим процедуру для окончательного формирования вывода по приоритету каждого из перечисленных средств изменения номеров потоков. Результат представлен на рисунке 9.

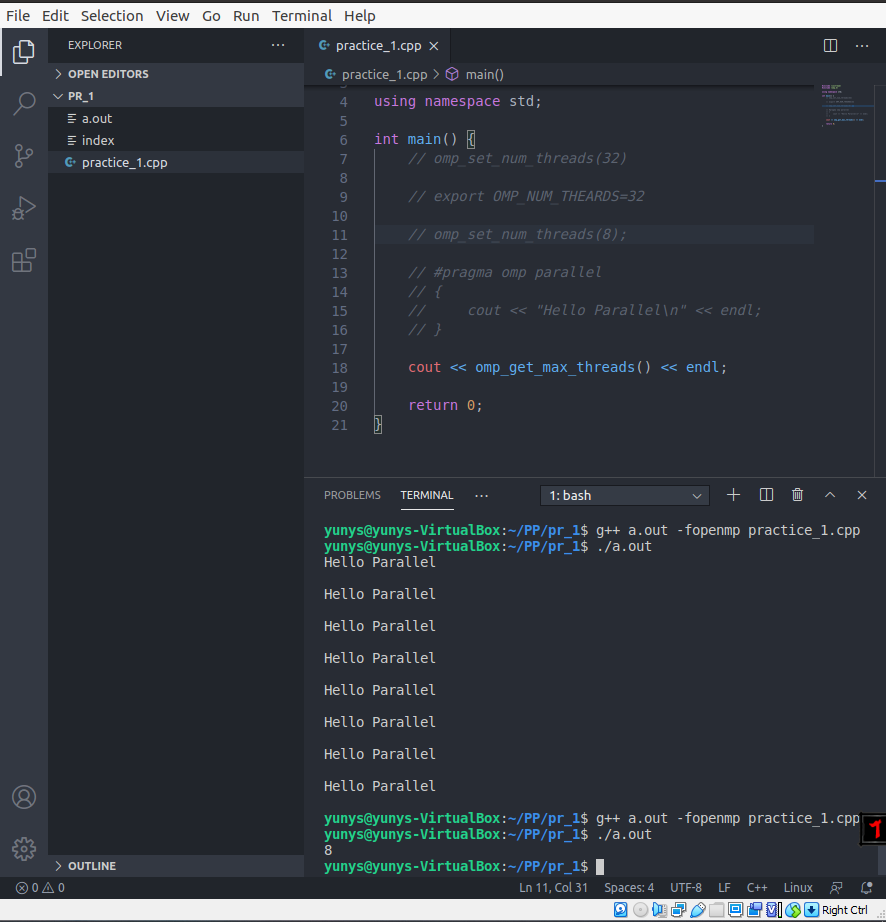


*Рисунок 12. Скриншот результат работы программы по пункту 6(2).*

Исходя из этого результата, делаем вывод что приоритет способа потоков следующий: 1 – опция директивы, 2 – функция omp\_set\_num\_threads(), 3 – переменная окружения OMP\_NUM\_THREADS

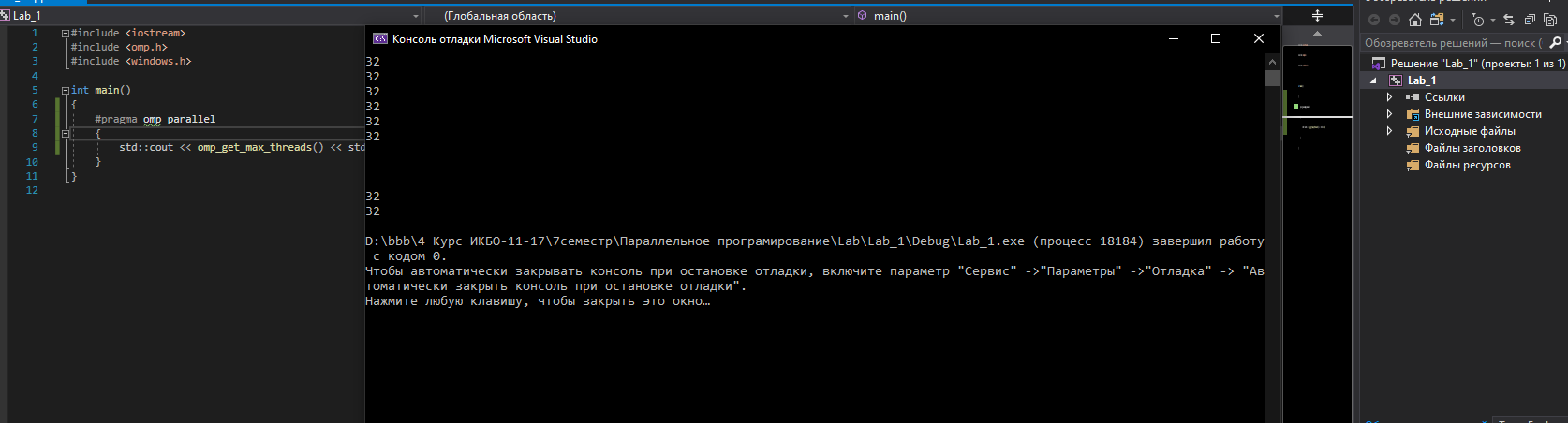
# **Вывести количество потоков**

Для выполнения данного пункта задачи необходимо вывести в коде значение функции omp\_get\_max\_threads(). Исходя из результатов рисунка 10 убеждаемся, что количество потоков равно количеству ядер используемого процессора. В моем случае – это число 8. Число 8 потому что виртуальная машина настроена на 8 потоков.



*Рисунок 13. Скриншот результат работы программы задачи 7.*

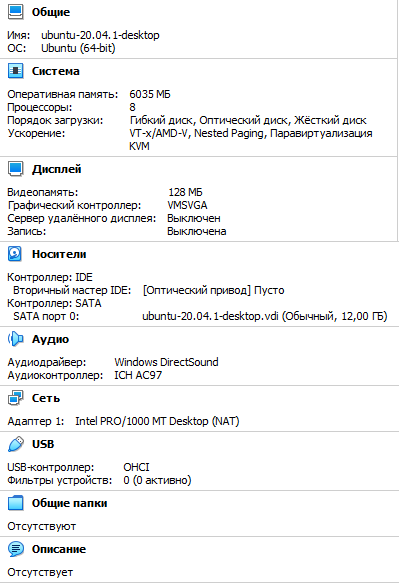
На физической машине максимальное количество потоков = 32, это можно определить по рисунку 14.



*Рисунок 14. Скриншот результат работы программы задачи 8.*

# **Ход выполнения задачи 9**

Характеристики ПК, используемого в практической работе представлены на рисунке 11. Полное наименование процессора, на котором запущена виртуальная машина – Intel Core i5-6300HQ 2.30MHz. Технология hyper threading отсутствует.



*Рис 13. Характеристики ПК (виртуальной машины), используемой при выполнении данной практической работы.*

# **Выводы**

В ходе выполнения данной практической работы была установлена и настроена IDE MVS Code ver 1.50. Так же рассмотрена работа с пакетом для языка C++ OpenMp, выполнены задачи настоящей практической работы, по каждой задаче проведены соответствующие выводы.

# **Список используемых информационных источников**

1. Сыромятников В. П. Курс лекций по дисциплине «Параллельное программирование». – РТУ МИРЭА, 2020-2021 г.

2. Руководство по использованию модуля OpenMp [Электронный ресурс]. – Режим доступа: <http://ccfit.nsu.ru/arom/data/openmp.pdf> (дата обращения 17.10.2020)

3. Официальные характеристики процессора – Intel Core i5-6300HQ [Электронный ресурс]. – Режим доступа: <https://ark.intel.com/content/www/ru/ru/ark/products/80815/intel-core-i5-4590-processor-6m-cache-up-to-3-70-ghz.html> (дата обращения 17.10.2020)

4. Официальный сайт дистрибутива MVS Code [Электронный ресурс]. – Режим доступа: <https://code.visualstudio.com> (дата обращения 17.10.2020)