Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Модели данных и системы управления базами данных

*К защите допустить*:

Заведующий кафедрой информатики

\_\_\_\_\_\_\_\_\_\_\_\_С. И. Сиротко

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

**Программное средство для коммуникации пользователей (соцсеть)**

БГУИР КП 1-40 04 01 035 ПЗ

|  |  |  |
| --- | --- | --- |
| Студент |  | М. А. Дубинка |
| Руководитель |  | А. В. Давыдчик |
| Нормоконтролер |  | А. В. Давыдчик |

Минск 2024

# СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc153150112)

[1 АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ 5](#_Toc153150113)

[1.1 Архитектура](#_Toc153150114) *[CPU](#_Toc153150114)* [5](#_Toc153150114)

[1.2 Архитектура](#_Toc153150115) *[GPU](#_Toc153150115)* [7](#_Toc153150115)

[1.3 Обоснование выбора систем 10](#_Toc153150116)

[2 ПЛАТФОРМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 11](#_Toc153150117)

[ЗАКЛЮЧЕНИЕ 40](#_Toc153150135)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 41](#_Toc153150136)

[ПРИЛОЖЕНИЕ Б 43](#_Toc153150137)

[ПРИЛОЖЕНИЕ В 45](#_Toc153150138)

[ПРИЛОЖЕНИЕ Г 46](#_Toc153150139)

[ПРИЛОЖЕНИЕ Д 47](#_Toc153150140)

# ВВЕДЕНИЕ

С развитием технологий социальные платформы для обмена контентом стали играть важную роль в повседневной жизни, позволяя пользователям делиться фотографиями, видео и другими формами цифрового контента. С появлением блокчейна и NFT (невзаимозаменяемых токенов) возможности цифровой собственности вышли на новый уровень, позволяя пользователям не только публиковать контент, но и обладать подтвержденными правами на него. Тем не менее, большинство существующих решений для создания и продажи NFT требуют значительных усилий и финансовых затрат от пользователей, что ограничивает их массовое использование в социальных сетях.

Данная курсовая работа направлена на устранение этих ограничений, интегрируя блокчейн-технологию и сжатые NFT (cNFT) в социальную платформу. Каждая фотография, загруженная пользователем, автоматически преобразуется в cNFT, позволяя владельцу контента управлять, продавать и делиться им. Для обеспечения надежного хранения данных о пользователях, cNFT, транзакциях и взаимодействиях требуется эффективная и масштабируемая база данных, которая станет основой для успешного функционирования.

Целью данной курсовой работы является проектирование и разработка базы данных для социальной платформ, которая объединяет традиционные функции социальных сетей и возможности блокчейна для работы с cNFT. База данных должна эффективно хранить и обрабатывать данные пользователей, cNFT, транзакций, а также данные о взаимодействиях между пользователями, таких как лайки, комментарии и подписки.

Задачи курсовой работы:

− Определить требования к структуре и функциональности базы данных для поддержки всех основных функций MintGram;

− Выбрать и обосновать выбор основной системы управления базами данных (СУБД);

− Спроектировать и реализовать схему базы данных, включая таблицы и связи для хранения информации о пользователях, публикациях, транзакциях, cNFT и социальных взаимодействиях;

− Оптимизировать базу данных для обеспечения высокой производительности, безопасности и масштабируемости;

− Реализовать основные функции взаимодействия с базой данных, такие как регистрация и авторизация, публикация и продажа cNFT, а также хранение и обработка данных о социальных взаимодействиях (лайки, комментарии и подписки);

− Проверить работоспособность базы данных в условиях моделирования высокой нагрузки, чтобы убедиться в её устойчивости и эффективности.

Разрабатываемая база данных должна быть структурирована таким образом, чтобы поддерживать все функции социальной платформы: регистрация и авторизация через криптокошельки и электронную почту, публикация и продажа cNFT, взаимодействие с контентом и другими пользователями, а также хранение статистики и данных о транзакциях. База данных должна быть готова к высокой нагрузке, так как платформа рассчитана на большое количество активных пользователей. Выбранная СУБД должна обеспечить возможность эффективного хранения и обработки данных, а также поддерживать функции для быстрой выборки, обеспечения безопасности и масштабируемости.

# 1 АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

## 1.1 Архитектура *CPU*

Процессор *Intel Core i*5-9300*h* (рисунок 1.1), выпущенный в 2019 году, является представителем популярной линейки процессоров *Intel Core i*5, предназначенных для ноутбуков и портативных устройств. Этот четырехъядерный процессор, основанный на архитектуре *Intel Coffee Lake HR*, предлагает выдающуюся производительность для широкого спектра вычислительных задач. [1] Для более глубокого понимания его архитектуры и возможностей проведем детальный анализ.

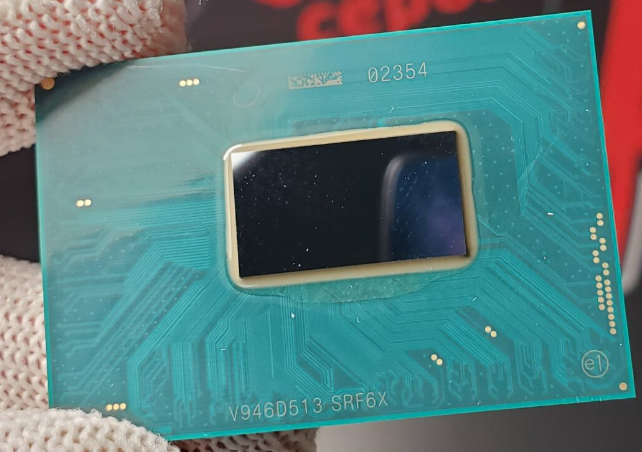


Рисунок 1.1 – Внешний вид процессора *Intel Core i5*-9300*h*

**1.1.1** Архитектура. Процессор *Intel Core i*5-9300*h* построен на архитектуре Intel Coffee Lake *HR* (рисунок 1.2), которая была выпущена в 2017 году. *Coffee Lake HR* (*CFL-HR*, *Coffee Lake H Refresh*) – кодовое название линейки высокопроизводительных мобильных процессоров *Intel*, основанных на микроархитектуре *Coffee Lake*, являющейся преемницей *Coffee Lake H*. [2] Эти чипы в первую очередь предназначены для обеспечения максимальной производительности мобильных устройств, мобильных рабочих станций и высокопроизводительных встраиваемых устройств.

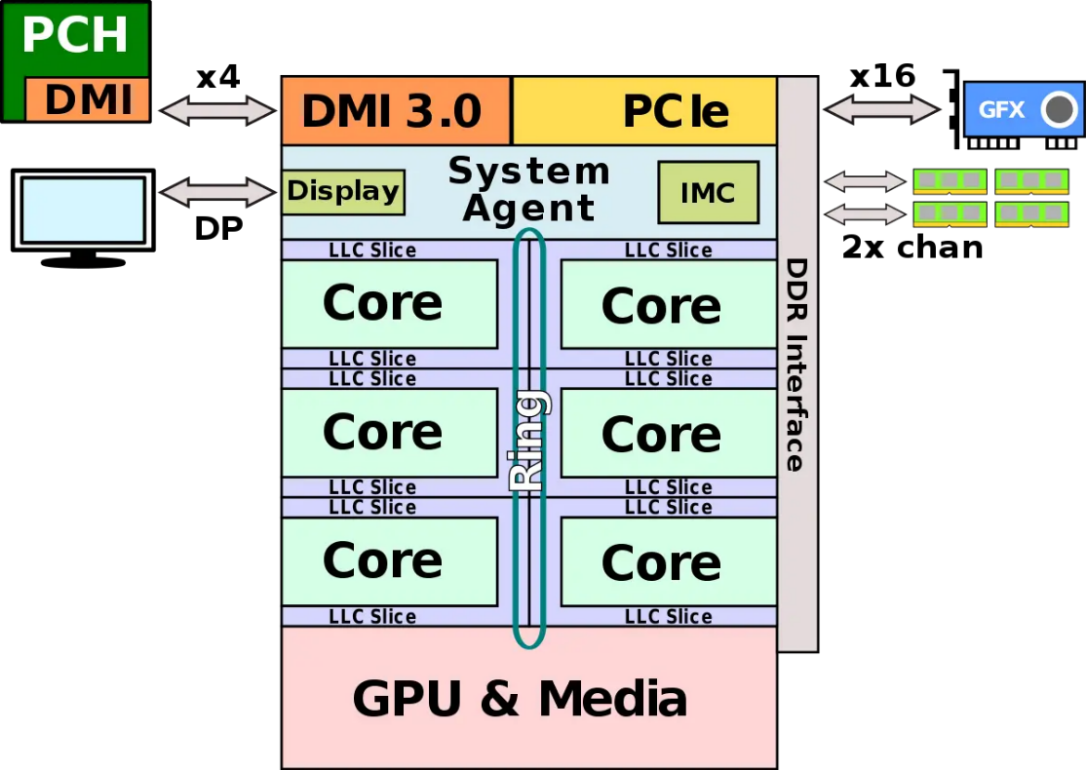


Рисунок 1.2 – Архитектура *Coffe* *Lake*

Процессоры *Coffee Lake HR* изготавливаются по усовершенствованному 14-нм техпроцессу Intel и представляют собой двухчиповое решение – микропроцессор и набор микросхем. *Coffee Lake H* имеют *BGA*-1440 и используют чипсет 300-й серии (концентратор). Микропроцессор подключен к чипсету через 4 из 20 каналов *PCI* чипа с использованием фирменного интерфейса *Direct Media* 3.0 от *Intel* (*DMI* 3.0), обеспечивающего скорость передачи данных 8 ГТ/с на канал.

**1.1.2** Технические характеристики: Количественные параметры *Core i*5-9300*h*:

– Архитектура: *x*86;

– Количество ядер: 4;

– Количество потоков: 8;

– Базовая тактовая частота: 2.7 ГГц;

– Максимальная тактовая частота: 4.5 ГГц;

– Кэш-память *L*1: 256 КБ;

– Кэш-память *L*2: 1 MБ;

– Кэш-память *L*3: 8 МБ;

– Техпроцесс: 14 нм;

– Интегрированная графика: *Intel* *UHD* *Graphics* 630*;*

– Поддержка памяти: *DDR*4 (до 2666 МГц);

– *TDP*: 45 Вт. [3]

Число ядер и потоков, тактовые частоты, техпроцесс, объем кэша и состояние блокировки множителя, косвенным образом говорят о производительности процессора. Пропускная способность памяти зависит не только от процессора. Данный курсовой проект будет выполняться на устройстве Lenovo L340-Gaming, пропускная способность памяти в котором 18.75 ГБ/c.

**1.1.3** Технологии и особенности. Технология *Intel Turbo Boost* позволяет процессору автоматически увеличивать тактовую частоту при необходимости для обеспечения более высокой производительности. Технология *Hyper-Threading* позволяет каждому физическому ядру обрабатывать два потока данных, что улучшает производительность в многозадачных приложениях. *TDP* указан на уровне 45 Вт, потому этот процессор подходит только для ноутбуков с эффективной системой охлаждения. Однако с помощью механизма *TDP-down* производитель снижает *TDP* процессора до 35 Вт, в результате чего снижается и производительность, и температура. [4]

## 1.2 Архитектура *GPU*

Графический процессор (*GPU*): *GTX* 1650 (рисунок 1.3) представляет собой дискретную графическую карту от *Nvidia*. Графические процессоры разработаны специально для параллельных вычислений и обработки графики.

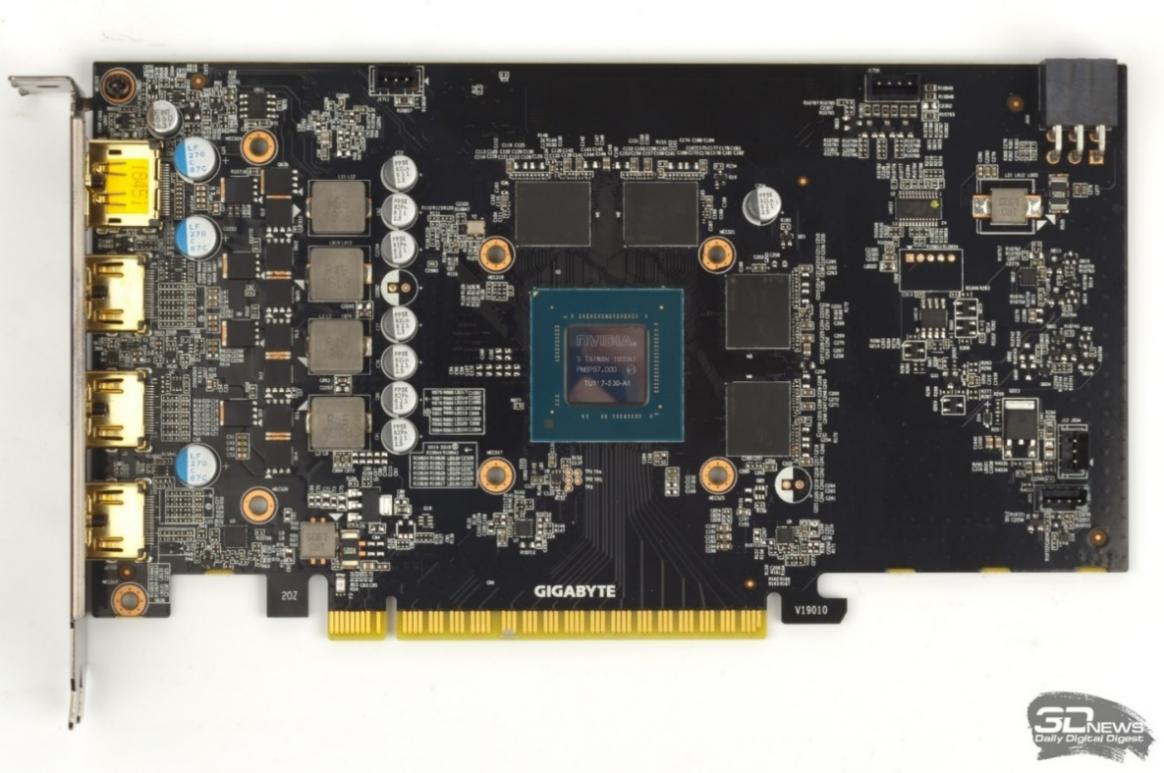


Рисунок 1.3 – Внешний вид *Nvidia* *GTX* 1650

Графический процессор, который будет использоваться в рамках работы, имеет следующие характеристики:

– Архитектура *Turing*;

– Количество *CUDA*-ядер: 1024;

– Частота ядра: базовая частота 1380 МГц, *boost*-частота 1515 МГц;

– Техпроцесс: 12 нм;

– Видеопамять: 4*GB* *GDDR*5;

– Ширина шины памяти: 128 бит;

– Пропускная способность памяти: 128 ГБ/с;

– *TDP*: 50 Вт. [5]

*GTX* 1650 построена на архитектуре *Turing*, которая включает в себя ядра *CUDA* для выполнения параллельных вычислений. Эта архитектура была разработана с учетом высокой производительности в задачах машинного обучения и графики. Архитектура *NVIDIA* *Turing* является одной из последних и наиболее значимых архитектур, разработанных для графических процессоров (*GPU*) компании *NVIDIA*. Она была анонсирована в 2018 году и представляет собой существенное обновление предыдущей архитектуры *Pascal*. Архитектура *Turing* внесла ряд инноваций и технологических усовершенствований, которые сделали её ключевой для вычислительных и графических задач высокой производительности. [5] Некоторые из основных особенностей и преимуществ архитектуры *Turing* (рисунок 1.4):

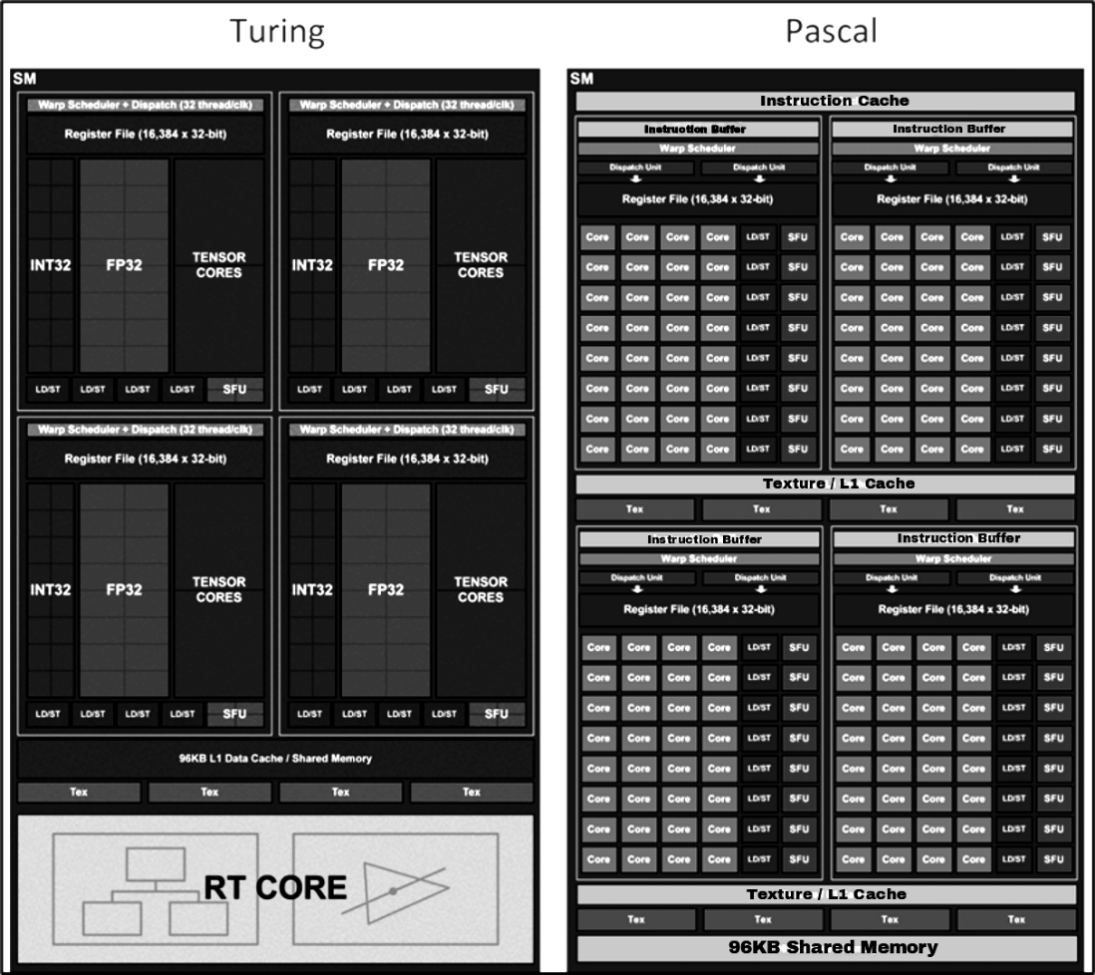


Рисунок 1.4 – Сравнение *SM* в *Turing* и *Pascal*

1 Усовершенствованные ядра *CUDA*: aрхитектура *Turing* включает в себя ядра *CUDA* нового поколения, что позволяет более эффективно обрабатывать параллельные вычисления и ускоряет выполнение задач искусственного интеллекта, машинного обучения и графики.

2 Тензорные ядра для искусственного интеллекта: архитектура *Turing* включает в себя тензорные ядра, предназначенные для обработки операций искусственного интеллекта. Эти ядра способствуют ускорению процессов машинного обучения и глубокого обучения.

3 Графические ядра *Turing*: графические ядра, благодаря одновременному выполнению операций с плавающей запятой и целочисленных операций, в архитектуре *Turing* обеспечивают выдающуюся производительность в графических задачах, включая игры и 3*D*-моделирование.

4 Дополнительные технологии: архитектура *Turing* поддерживает технологии, такие как *DLSS* (*Deep Learning Super Sampling*), которая использует искусственный интеллект для улучшения качества графики при снижении нагрузки на *GPU*, и *GDDR*6 видеопамять для высокой скорости передачи данных.

Архитектура *Turing* стала значимым шагом в развитии графических технологий, предоставив более высокую производительность и возможности для создания реалистичных и красочных графических сценариев, а также для ускорения вычислений в области искусственного интеллекта.

*CUDA* (*Compute Unified Device Architecture*) – это технология, разработанная компанией *NVIDIA*, которая позволяет использовать процессоры графических карт для выполнения вычислений общего назначения. Она предоставляет программистам доступ к вычислительным ресурсам графического процессора (*GPU*) для решения сложных задач, которые требуют большой вычислительной мощности.

*CUDA* предоставляет интерфейс программирования приложений (*API*) и набор инструментов, которые позволяют разработчикам написать код, который выполняется на графическом процессоре. Типичные области применения *CUDA* включают в себя научные и инженерные вычисления, машинное обучение и глубокое обучение, обработку изображений и видео, финансовый анализ, анализ данных и многое другое.

*CUDA* поддерживает языки программирования *C*, *C*++, *Python*. Разработчикам доступны специальные расширения языка, такие как *CUDA* *C*, которые позволяют использовать параллельные вычисления и многопоточность для максимальной эффективности работы с графическими процессорами.

Основной преимуществом использования *CUDA* является возможность значительно ускорить выполнение сложных вычислений за счет использования параллельной обработки на графическом процессоре. Графические карты имеют большое количество ядер и параллельных вычислений, что позволяет обрабатывать большие объемы данных и выполнять операции намного быстрее, чем на центральном процессоре.

*CUDA* также предоставляет различные инструменты и библиотеки, такие как *cuBLAS* и *cuDNN*, которые упрощают программирование и оптимизацию вычислений. Они предоставляют предопределенные функции и алгоритмы, которые можно использовать для решения специфических задач без необходимости писать весь код с нуля.

В целом, *NVIDIA GTX* 16*5*0 представляет собой мощное графическое решение для ноутбуков, обеспечивающее высокую производительность и передовые графические возможности.

## 1.3 Обоснование выбора систем

Выбор системы с *GPU GTX* 1650 и процессором *i*5-9300*h* для данной курсовой работы обоснован следующими причинами:

1 Доступность и популярность оборудования: *GTX* 1650 и процессор *i*5-9300*h* являются достаточно популярными и доступными компонентами для сборки компьютера. Это облегчает работу с этой системой и удобство получения результата.

2 Баланс производительности и стоимости: *GTX* 1650 и *i*5-9300*h* являются частью более доступных вариантов оборудования, которые обладают приемлемой производительностью по сравнению с более дорогими альтернативами.

3 Широкая поддержка и документация: Оборудование, такое как *GTX* 1650 и *i*5-9300*h*, имеет широкую базу пользователей и разработчиков, что означает наличие подробной документации, онлайн-ресурсов и сообщества, готового поддержать в случае возникновения проблем или вопросов во время написания курсовой работы.

В целом, выбор системы с *GTX* 1650 и *i*5-9300*h* обусловлен их эффективным сочетанием производительности, доступности, поддержки и масштабируемости, а также их способностью выполнять задачи параллельных вычислений и обеспечивать баланс между производительностью и стоимостью.

# 2 ПЛАТФОРМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для сравнения производительности процессора и видеокарты нужно определиться с таким фактором как операционная система. Для проведения такого сравнения необходимо создать одинаковые условия. Также от операционной системы требуется свойство использовать минимум ресурсов для того, чтобы обеспечить максимальную производительность при выполнении вычисления.

Так как сравнение будет выполняться на устройстве на базе *CPU* *i*5-9300*h* и *GPU* *GTX* 1650, то такие характеристики, как время обращения к памяти и работы сетью можно считать одинаковыми. Это позволяет использовать операционную систему без установки виртуальных машин для приравнивания окружения.

Если учесть все факторы упомянутые выше, то выбор можно свести к операционным системам *Windows* и дистрибутив *GNU*/*Linux* *Ubuntu*. Данные две операционные системы обладают большой популярностью, что в свою очередь означает широкую поддержку производителем и сторонними разработчиками. Таким образом, данные две операционные систему поддерживают поддержку выбранного аппаратного обеспечения, а также обеспечивают драйверов и библиотек. Однако *Ubuntu* имеет ряд преимуществ. Данная операционная система предоставляет большую гибкость и контроль над системой. Пользователи могут легко настраивать различные аспекты операционной системы, чтобы соответствовать заданным потребностям. Это особенно важно при сравнении производительности *CPU* и *GPU*, так как различные факторы, такие как фоновые процессы, службы и настройки ядра, могут влиять на точность и последовательность измерений производительности. *Ubuntu* предоставляет более прямой контроль над этими факторами, чем, операционная система *Windows*, где есть больше фоновых процессов, и виртуальные машины, которые также могут вносить некоторые накладные расходы и ограничения. Также к преимуществам Ubuntu можно отнести наличие возможности отключения графического интерфейса, что позволяет снизить нагрузку на процессор до минимума (рисунок 2.1).

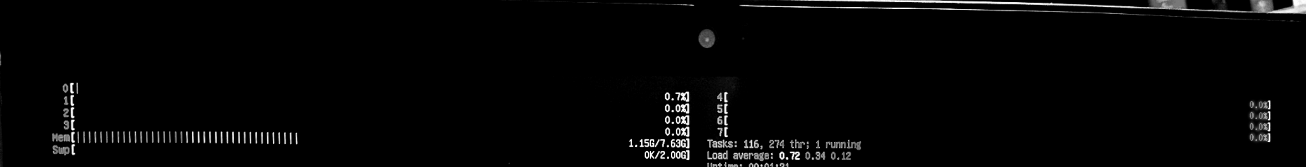


Рисунок 2.1– Нагрузка на ядра процессора при отключенном графическом интерфейсе *Ubuntu*

Рассмотрим операционную систему *Ubuntu* более подробно. *Ubuntu* – это один из самых популярных бесплатных дистрибутивов *Linux*, широко используемый как в коммерческих, так и в академических средах. [6] Он предоставляет мощную и гибкую платформу для проведения исследований, разработки и сравнения производительности различных аппаратных компонентов, включая процессоры и графические ускорители. Ubuntu разработана с упором на простоту использования. У нее простой и интуитивно понятный пользовательский интерфейс, что делает ее доступной для новичков и опытных пользователей. Она также предлагает множество предустановленных приложений и инструментов, которые позволяют начать работу сразу после установки. Большинство пользователей *Ubuntu* – это разработчики, поэтому данная операционная система имеет активное сообщество. Благодаря чему упрощается и ускоряется процесс разработки.

Использование *C*/*C*++ для курсовой работы обосновывается следующими причинами:

1 Уровень абстракции: *C*++ является более высокоуровневым языком программирования, чем *C*, что позволяет разработчику использовать более удобные и гибкие конструкции языка. Это может облегчить написание кода и управление алгоритмами вычислений числа Пи.

2 Удобство работы с параллельными вычислениями: Оба языка, *C*++ и *C*, обладают мощной поддержкой многопоточности и параллельных вычислений. Это позволяет эффективно использовать вычислительные ресурсы как *CPU*, так и *GPU*, распределять задачи между ними и контролировать их выполнение.

3 Библиотеки и инструменты: *C*++ и *C* являются широко используемыми языками программирования, для которых разработано множество библиотек и инструментов для работы с вычислительными задачами. Например, для работы с параллельными вычислениями и использования *GPU* можно использовать библиотеки, такие как *CUDA* для *NVIDIA* *GPU* или *OpenCL* для общего использования.

4 Производительность и управление ресурсами: Языки *C*++ и *C* предоставляют возможность более низкоуровневого программирования, что позволяет более точно управлять ресурсами и оптимизировать код для достижения максимальной производительности.

В целом, оба языка обладают достоинствами и будут использованы для эффективного решения поставленной в данной курсовой работе задачи.

Для того, чтобы обработать полученные данные и построить графики после выполнения программы выбран язык *Python*. *Python* является одним из самых популярных языков программирования для обработки данных, благодаря своей простоте, элегантности и богатой экосистеме библиотек для анализа данных. Основные преимущества *Python*:

1 Простота и читаемость: *Python* обладает простым и понятным синтаксисом, который делает его легким для изучения и использования. Он ставит акцент на читаемости кода, что облегчает сопровождение и коллаборацию между разработчиками.

2 Переносимость: *Python* является переносимым языком, что означает, что программы, написанные на *Python*, могут работать на различных операционных системах, таких как *Windows*, *macOS* и *Linux*, без необходимости изменения исходного кода. Это делает *Python* удобным выбором для разработки кросс-платформенных приложений.

3 Быстрая разработка: *Python* обладает простым и прямолинейным синтаксисом, а также богатым комплектом инструментов и библиотек, что позволяет разработчикам быстро создавать прототипы и экспериментировать с идеями. Это особенно полезно в сферах, таких как веб-разработка и научные исследования.

4 Интеграция: *Python* легко интегрируется с другими языками программирования, такими как *C*, *C*++, *Java* и другими. Это позволяет использовать существующий код на других языках, включать расширения и использовать *Python* в качестве скриптового языка для автоматизации и расширения функциональности других приложений.

5 Большое количество библиотек для анализа данных: *Python* имеет множество мощных библиотек, таких как *NumPy*, *Pandas*, *Matplotlib*, *SciPy* и *Scikit*-*learn*, которые предоставляют широкий набор функций и инструментов для обработки, анализа и визуализации данных. Эти библиотеки значительно упрощают выполнение сложных операций с данными.

6 Мощные инструменты визуализации данных: *Python* предоставляет различные библиотеки для визуализации данных, такие как *Matplotlib*, *Seaborn*, *Plotly* и другие. Они позволяют создавать качественные и информативные графики, диаграммы и визуализации, что облегчает анализ данных и коммуникацию результатов. [7]

Исходя из этого *Python* был выбран для построения графиков и обработки данных в курсовом проекте.

# 3 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

# ПРИЛОЖЕНИЕ Б

(**обязательное**)

**Исходный** **код**