65Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |  |
| --- | --- |
| Институт | математики и компьютерных наук |
| Кафедра | компьютерной безопасности |

ОТЧЕТ

Лабораторная работа №4

по дисциплине «Исследование И Мониторинг Производительности ОС Windows С Помощью Системного Монитора»

Выполнил:

Пронин Владимир Иванович,

студент 2 курса

группы КМБ-с-о-23-1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Проверено с оценкой:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_31 03 2025 г.

Ставрополь, 2025

1. Построить графики изменения количества потоков приложений Notepad и Open Office при создании документа, содержащего текст из одного слова.

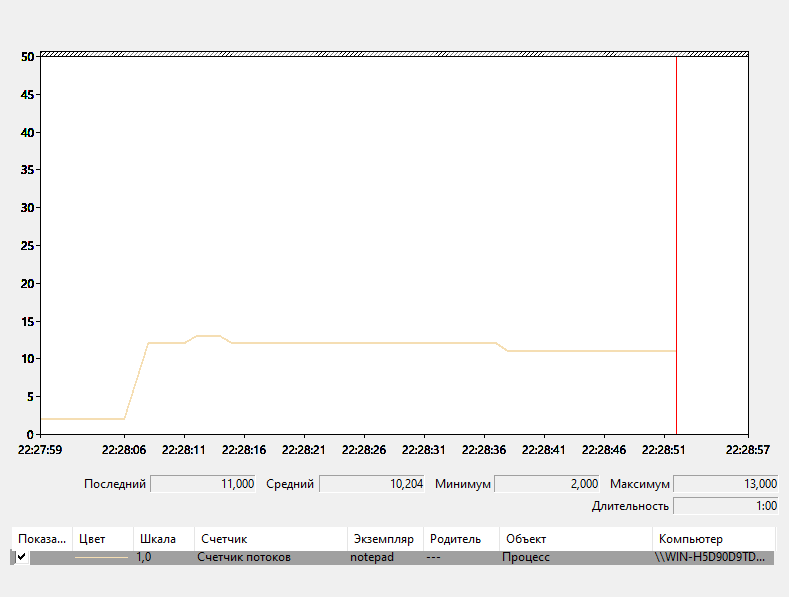


Рисунок 1 – график для NotePad’a

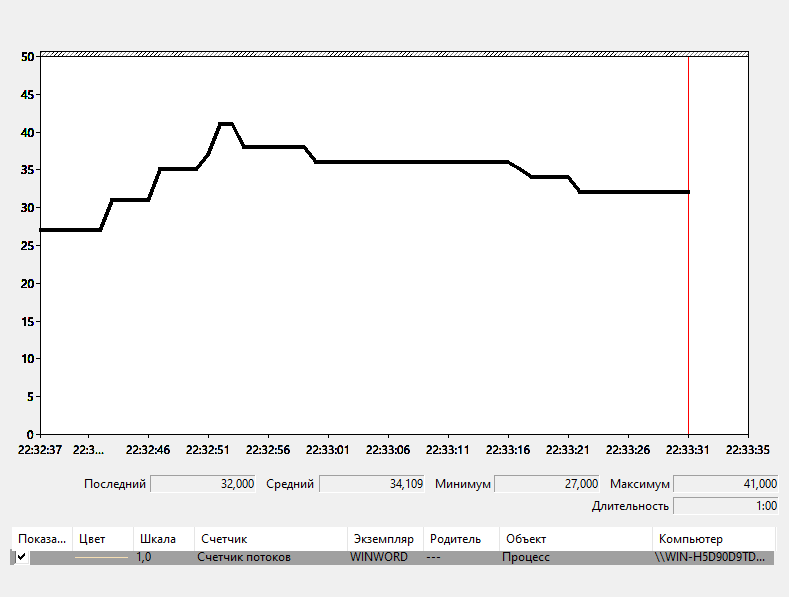


Рисунок 2 – график для Open Office

1. Для приложения Калькулятор построить 2-3 наиболее динамично изменяющихся графика изменения текущего приоритета потоков при вычислении значения арифметического выражения, перемещении калькулятора по экрану, перемещении курсора мыши по экрану в области окна калькулятора.

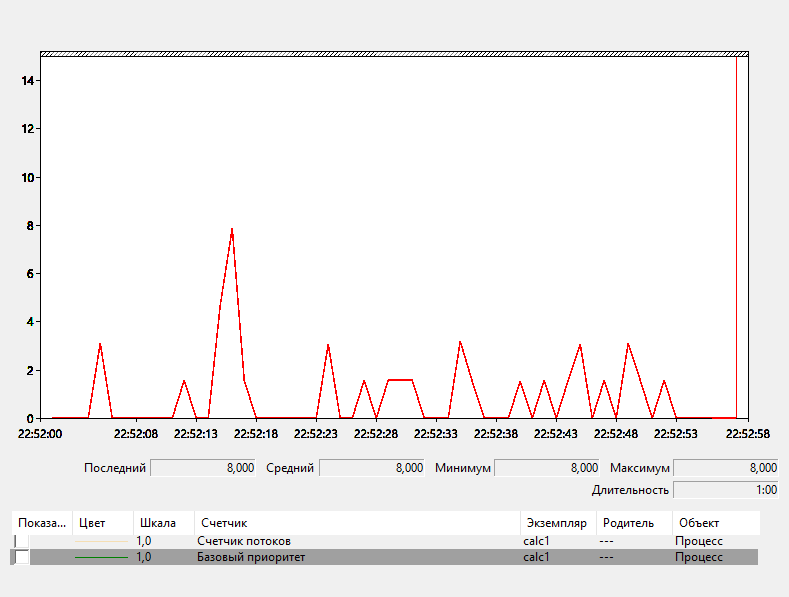


Рисунок 3 – график для калькулятора (перемещении курсора мыши по экрану в области окна калькулятора)

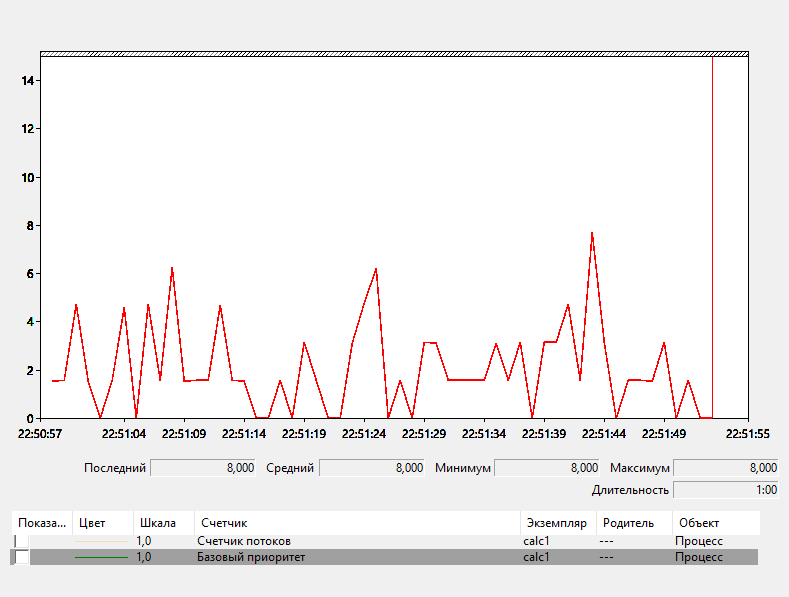


Рисунок 4 – график для калькулятора (при перемещении калькулятора по экрану)

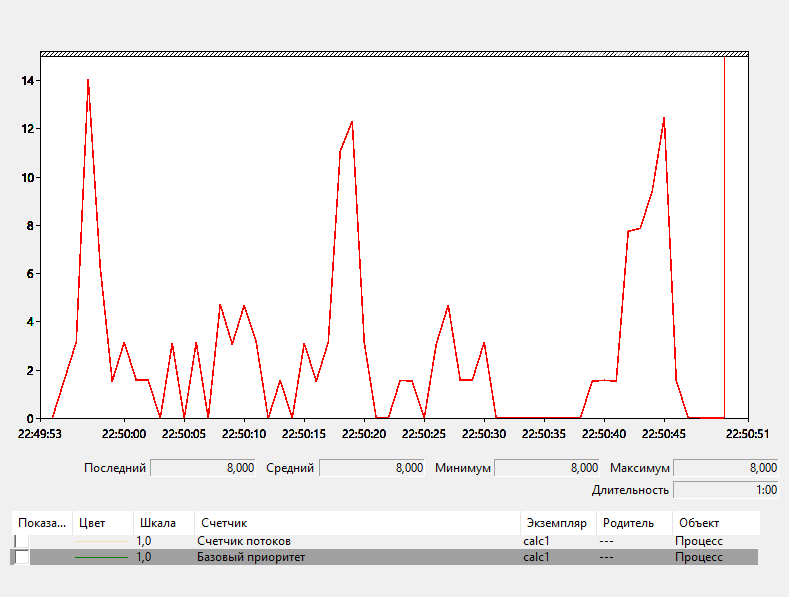


Рисунок 1 – график для калькулятора (при вычислении значения арифметического выражения)

1. Для приложения Open Office построить график изменения объема используемого файла подкачки при последовательном открытии 3-4 файлов увеличивающегося размера.

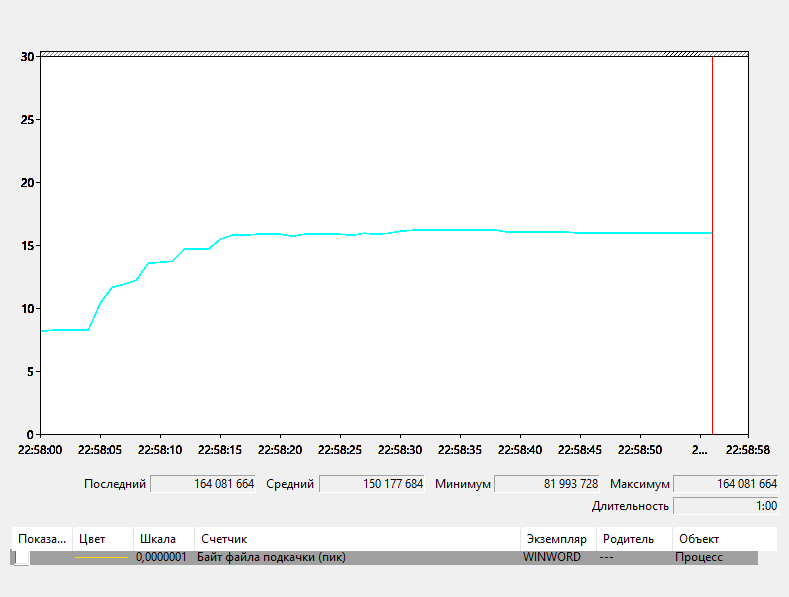
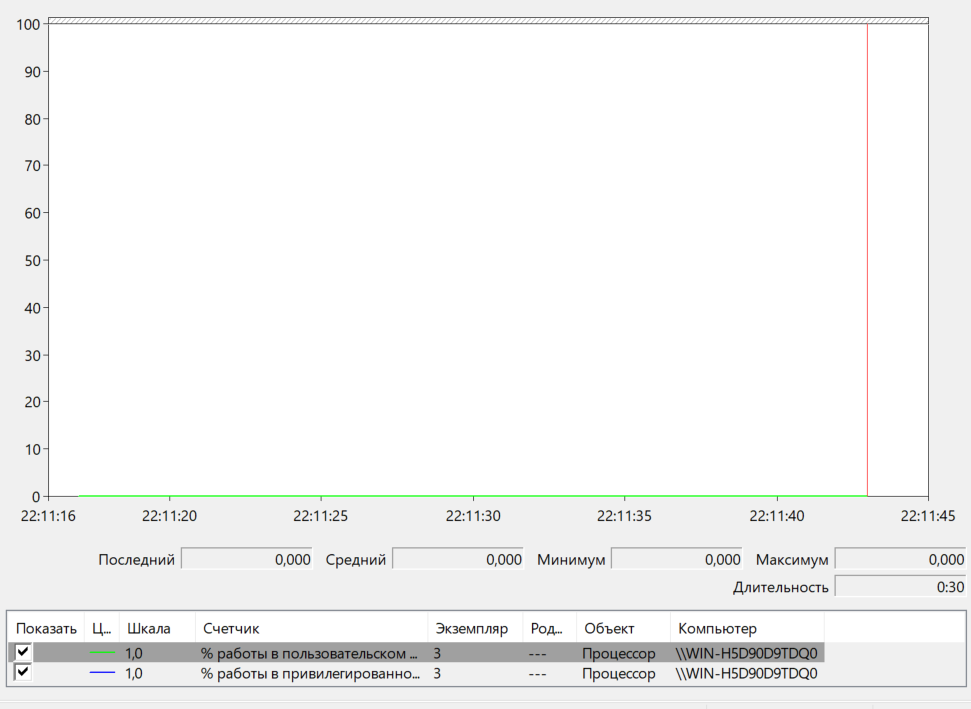
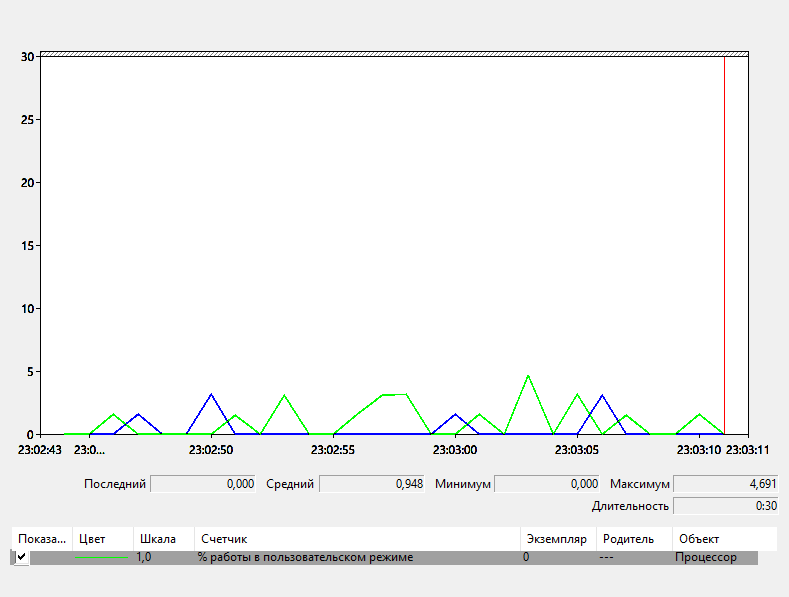
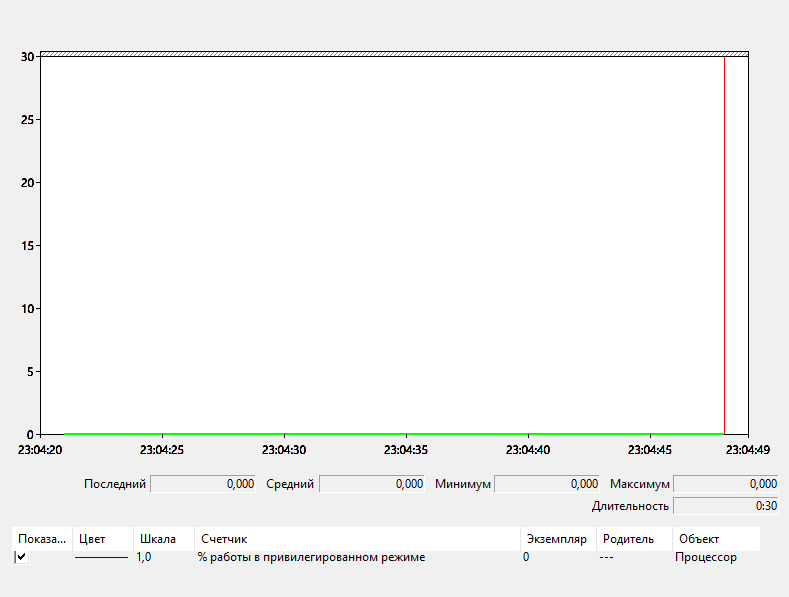
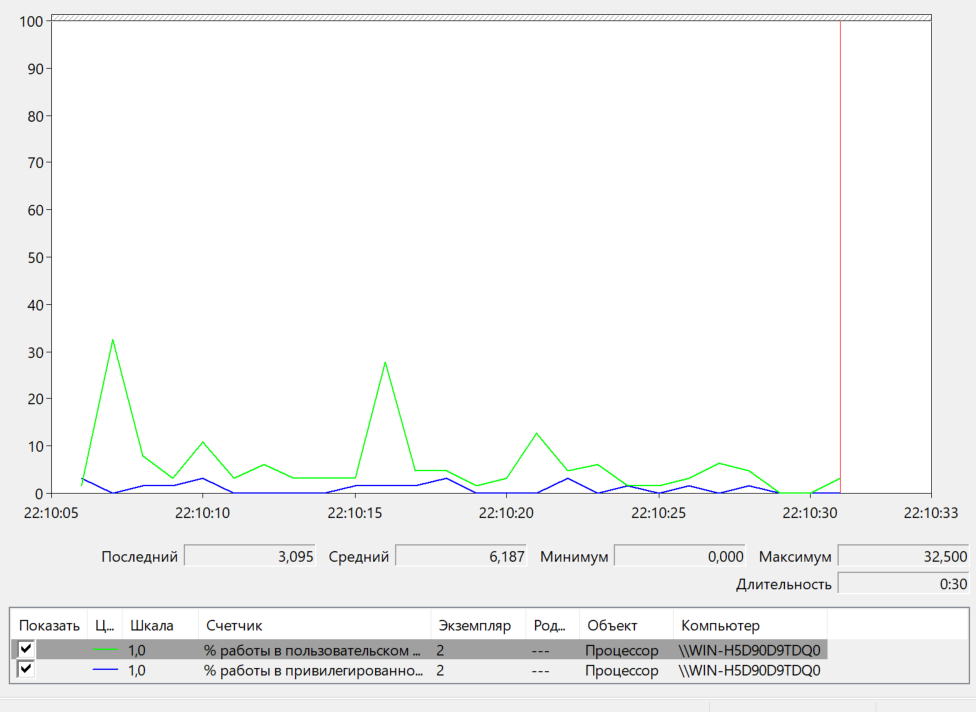
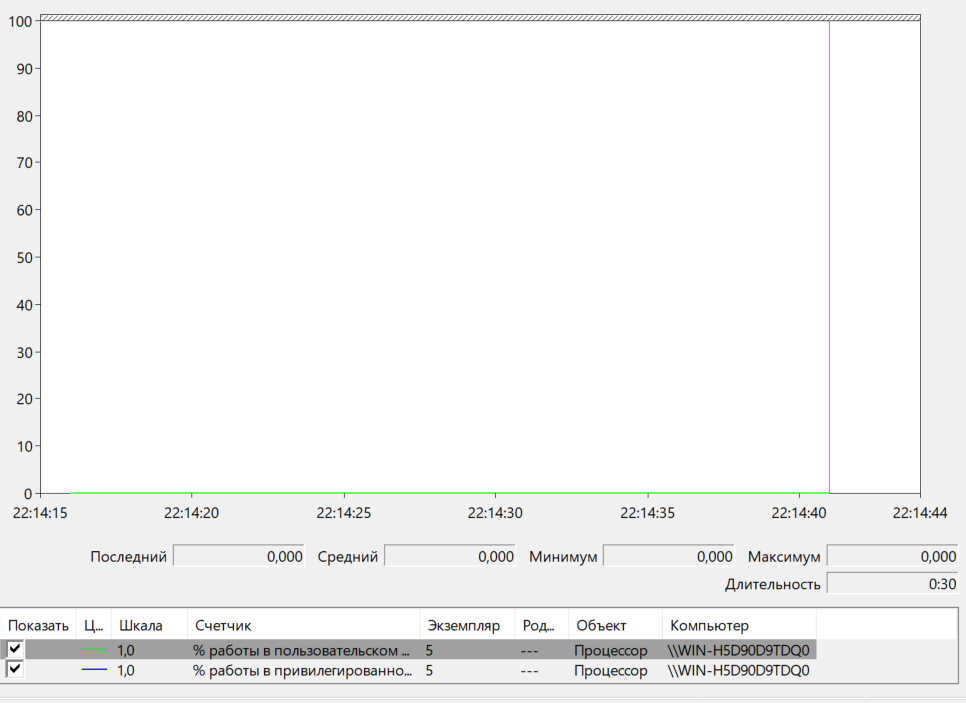
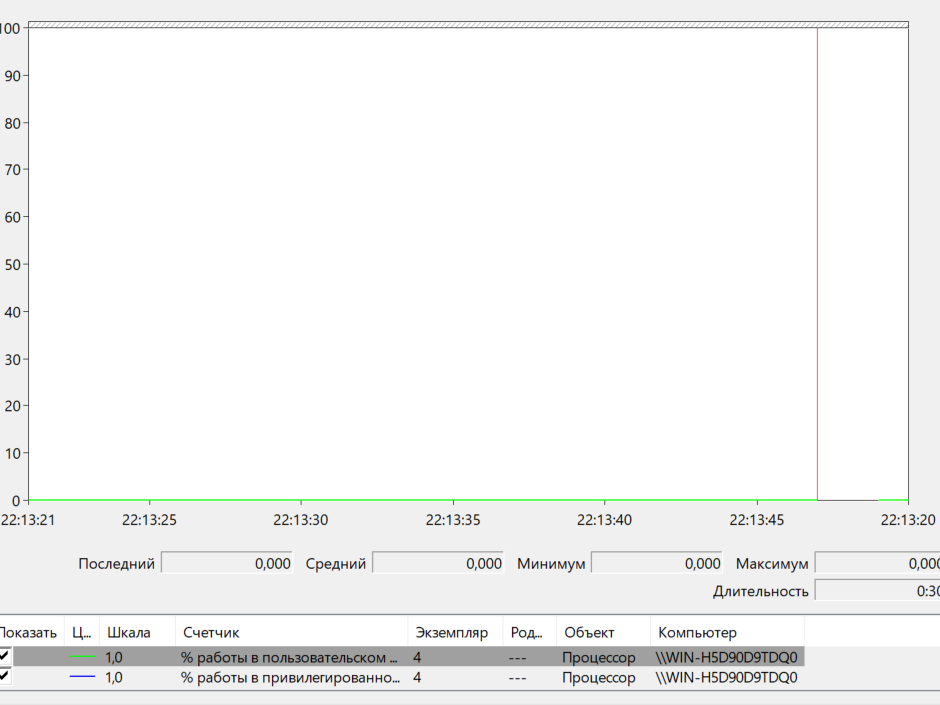


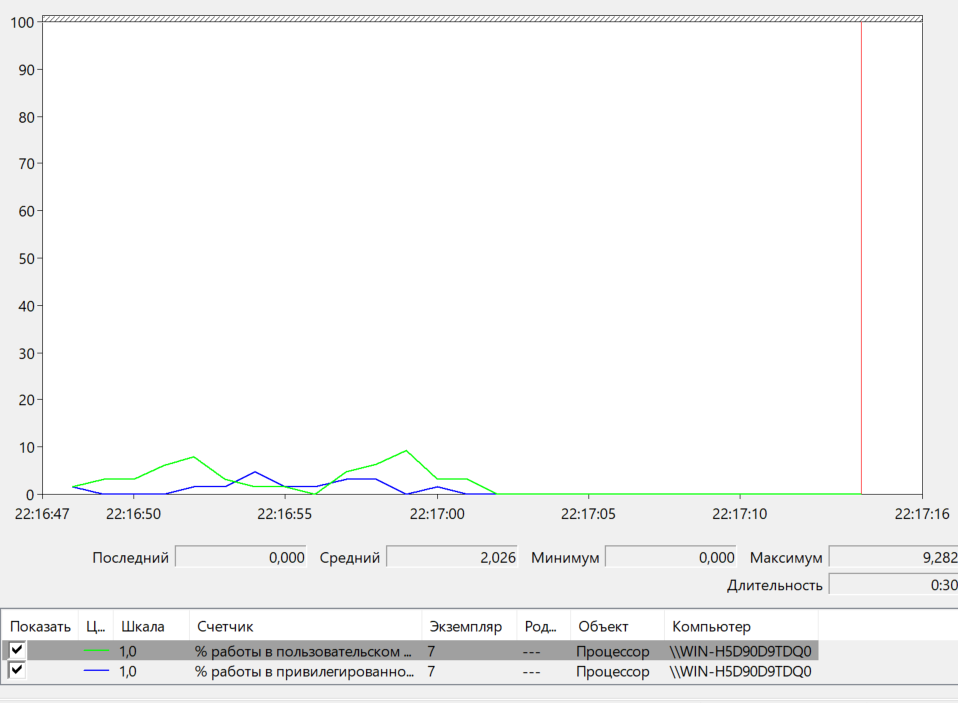
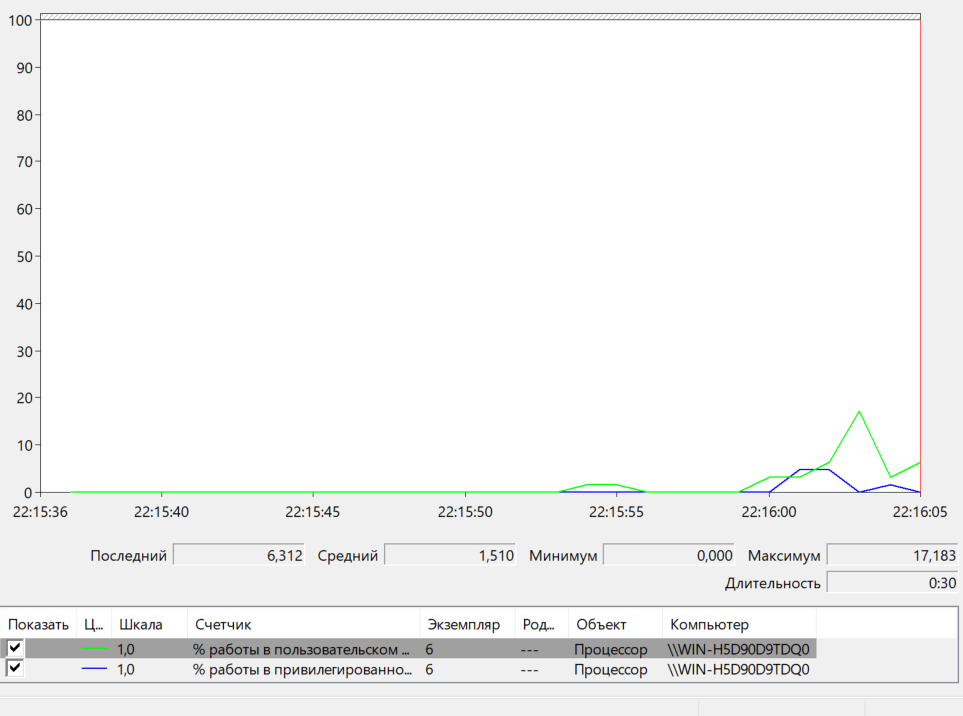
Рисунок 1 – график приложения Open Office

1. Выполнить индивидуальные задания для студентов согласно табл. 1









**1. Назначение счетчиков производительности**

Счетчики производительности (performance counters) — это инструменты, предоставляемые операционной системой, которые позволяют отслеживать различные аспекты производительности системы и отдельных приложений в реальном времени. Они предоставляют числовые данные о различных событиях и характеристиках системы.

**Основные назначения счетчиков производительности:**

* **Мониторинг производительности системы:** Позволяют получить общую картину о состоянии системы, выявить узкие места и аномалии в работе.
* **Отладка и профилирование приложений:** Помогают разработчикам анализировать, как их приложения используют ресурсы системы, выявлять причины низкой производительности и оптимизировать код.
* **Оптимизация использования ресурсов:** Дают информацию, необходимую для балансировки нагрузки, выделения оптимального количества ресурсов (памяти, процессорного времени) для различных приложений.
* **Планирование мощностей:** Предоставляют исторические данные о потреблении ресурсов, которые можно использовать для прогнозирования будущих потребностей и планирования масштабирования инфраструктуры.
* **Диагностика проблем:** Помогают определить причины проблем, таких как зависания, высокая загрузка процессора, нехватка памяти и т. д.
* **Создание отчетов:** Служат источником данных для создания отчетов о производительности системы и отдельных приложений.
* **Автоматизация задач:** Могут использоваться для автоматического реагирования на определенные события, например, для перезапуска приложения при достижении определенного порога загрузки процессора.

**Примеры использования счетчиков производительности:**

* Отслеживание загрузки процессора для определения, не перегружен ли сервер.
* Мониторинг использования памяти для выявления утечек памяти в приложении.
* Анализ дисковой активности для выявления узких мест в подсистеме хранения данных.
* Измерение времени отклика веб-сервера для оценки качества обслуживания.
* Отслеживание количества ошибок в приложении для выявления проблем с кодом.

**2. Категории и экземпляры счетчиков**

Счетчики производительности организованы в иерархическую структуру, состоящую из категорий и экземпляров.

* **Категории счетчиков (Performance Counter Categories):** Это логические группы, объединяющие счетчики, относящиеся к определенному аспекту производительности. Категории позволяют организовать счетчики по функциональному признаку и упростить их поиск.
  + **Примеры категорий:**
    - Processor (процессор)
    - Memory (память)
    - Disk (диск)
    - Network Interface (сетевой интерфейс)
    - Process (процесс)
    - .NET CLR Memory (память .NET CLR)
* **Экземпляры счетчиков (Performance Counter Instances):** Экземпляры представляют конкретные объекты или компоненты, для которых собираются данные. Не все категории имеют экземпляры.
  + **Примеры экземпляров:**
    - Для категории Processor: \_Total (общая загрузка всех ядер), 0, 1, 2 (загрузка отдельных ядер процессора)
    - Для категории Process: chrome (загрузка процесса chrome), notepad (загрузка процесса notepad)
    - Для категории Network Interface: Ethernet (производительность сетевого интерфейса Ethernet), Wi-Fi (производительность сетевого интерфейса Wi-Fi).
  + Если категория не имеет экземпляров, то это означает, что счетчик относится ко всей категории в целом. Например, счетчик Memory\Available MBytes относится ко всей памяти системы и не имеет экземпляров.

**Комбинация Категории, Экземпляра и Счетчика:**

Полный путь к счетчику производительности состоит из трех частей:

1. **Категория (Category):** Общая область, к которой относится счетчик.
2. **Экземпляр (Instance):** Конкретный экземпляр объекта, для которого собираются данные (может отсутствовать).
3. **Счетчик (Counter):** Конкретный показатель производительности, который измеряется.

**Пример:**

\Processor(\_Total)\% Processor Time

* Processor: Категория (процессор).
* \_Total: Экземпляр (все процессоры).
* % Processor Time: Счетчик (процент времени использования процессора).

**3. Управление параметрами создаваемых графиков (масштаб, цвет и толщина линий)**

Большинство инструментов для мониторинга производительности (например, Performance Monitor в Windows) предоставляют возможность управления параметрами создаваемых графиков, такими как масштаб, цвет и толщина линий. Конкретный интерфейс для управления этими параметрами зависит от используемого инструмента, но общие принципы остаются одинаковыми.

**Управление параметрами графиков:**

1. **Выбор счетчиков:**
   * Добавьте нужные счетчики производительности на график.
2. **Масштаб (Scale):**
   * Масштаб позволяет изменить отображение значений счетчика на графике. Это особенно полезно, если значения счетчика находятся в очень узком диапазоне или очень велики.
   * Обычно масштаб задается как множитель, на который умножается значение счетчика. Например, если значения счетчика находятся в диапазоне от 0 до 1, можно установить масштаб 100, чтобы отображать значения в процентах.
3. **Цвет (Color):**
   * Цвет позволяет визуально различать разные счетчики на графике.
   * Обычно для каждого счетчика можно выбрать свой цвет из палитры.
4. **Толщина линии (Line Thickness):**
   * Толщина линии влияет на то, насколько заметен счетчик на графике.
   * Более толстые линии лучше видны, но могут загромождать график, если на нем много счетчиков.

**Пример управления параметрами в Windows Performance Monitor:**

1. Запустите Performance Monitor (perfmon.exe).
2. Добавьте счетчик производительности, щелкнув по значку “+”.
3. В окне добавления счетчика выберите категорию, экземпляр и счетчик, затем нажмите “Add”.
4. Нажмите “OK”, чтобы добавить счетчик на график.
5. Чтобы изменить параметры счетчика (масштаб, цвет, толщину линии), щелкните правой кнопкой мыши на графике и выберите “Properties”.
6. Перейдите на вкладку “Graph”.
7. Выберите счетчик из списка “Counters”.
8. Измените параметры “Scale”, “Color” и “Style” (который включает толщину линии).
9. Нажмите “OK”, чтобы применить изменения.

**4. Влияние активности окна приложения на текущий приоритет его потоков**

Влияние активности окна приложения на текущий приоритет его потоков – это механизм, используемый операционной системой (обычно Windows) для оптимизации работы пользовательского интерфейса и обеспечения его отзывчивости.

**Принцип работы:**

Когда окно приложения находится на переднем плане и активно (т.е. имеет фокус ввода), операционная система может повысить приоритет потоков, связанных с этим окном. Это позволяет приложению быстрее реагировать на действия пользователя (например, щелчки мыши, нажатия клавиш).

Когда окно приложения находится в фоновом режиме (т.е. не имеет фокуса ввода), операционная система может понизить приоритет его потоков. Это позволяет освободить ресурсы системы для других приложений, которые в данный момент активны.

**Механизмы управления приоритетами:**

* **Приоритетные классы процессов (Process Priority Classes):**
  + Определяют общий приоритет процесса.
  + Примеры: Realtime, High, AboveNormal, Normal, BelowNormal, Idle.
  + Влияют на все потоки процесса.
* **Приоритеты потоков (Thread Priorities):**
  + Определяют приоритет конкретного потока внутри процесса.
  + Примеры: TimeCritical, Highest, AboveNormal, Normal, BelowNormal, Lowest, Idle.
  + Могут изменяться динамически операционной системой.
* **Динамическое повышение приоритета (Dynamic Priority Boosting):**
  + Операционная система может временно повышать приоритет потоков, которые обрабатывают ввод пользователя, чтобы обеспечить отзывчивость интерфейса.
  + Этот механизм обычно применяется к потокам, которые обрабатывают сообщения окон (window messages).

**Влияние на производительность:**

* **Повышение приоритета активного окна:**
  + Улучшает отзывчивость интерфейса активного приложения.
  + Может незначительно замедлить работу фоновых приложений.
* **Понижение приоритета фоновых окон:**
  + Освобождает ресурсы для активных приложений.
  + Может замедлить выполнение задач в фоновых приложениях (например, загрузку файлов, рендеринг видео).

**Практическое применение:**

* Операционная система старается сделать так, чтобы активное приложение работало плавно и быстро реагировало на действия пользователя.
* Фоновые приложения могут выполнять менее важные задачи с пониженным приоритетом, не мешая активным приложениям.
* Разработчики приложений должны учитывать влияние активности окна на приоритет потоков и разрабатывать приложения так, чтобы они хорошо работали как в активном, так и в фоновом режимах.
* Следует избегать длительных или ресурсоемких операций в потоках, отвечающих за UI, чтобы не блокировать интерфейс.

В заключение, активность окна приложения оказывает влияние на приоритет его потоков, что позволяет операционной системе оптимизировать работу пользовательского интерфейса и обеспечить его отзывчивость. Это динамический процесс, который постоянно адаптируется к текущим потребностям системы.