ПЗ 3. Мониторинг процессов в Windows

Цель:
Изучить инструменты мониторинга процессов и потоков, научиться анализировать их характеристики и влияние на систему.

Задание 1: Базовый анализ через Диспетчер задач

1. Запуск Диспетчера задач

- Нажмите сочетание клавиш **Ctrl + Shift + Esc** или **Ctrl + Alt + Delete** → выберите «Запустить диспетчер задач».

2. Анализ вкладки «Процессы»

- Перейдите на вкладку **Процессы**.
- В верхнем меню нажмите **«Вид»** → **«Выбрать столбцы»** и убедитесь, что отмечены:
  - **PID (Идентификатор процесса)**
  - **ЦП (Загрузка процессора)**
  - **Память (Потребление оперативной памяти)**
  - **Описание**

Шаг 2.1: Сортировка процессов по загрузке CPU

- Нажмите на заголовок столбца **«ЦП»**, чтобы отсортировать процессы по убыванию загрузки процессора.
- Сделайте скриншот для отчёта.
- Запишите в отчёт:
  - Названия трёх процессов с наибольшей загрузкой CPU.
  - Объясните, почему эти процессы могут потреблять много ресурсов.

Шаг 2.2: Анализ процесса svchost.exe

- Найдите все экземпляры процесса **svchost.exe** (используйте поиск или сортировку по имени).
- Сделайте скриншот для отчёта.
- Откройте вкладку **Службы**, и выполните сортировку по **PID**.
- Сделайте скриншот для отчёта.
- Ответьте на вопросы:
  - Сколько экземпляров **svchost.exe** запущено?
  - Почему их несколько?
  - Как определить, какие службы связаны с каждым экземпляром?

Шаг 2.3: Определение процесса с наибольшим потреблением памяти

- Нажмите на заголовок столбца **«Память»**, чтобы отсортировать процессы по убыванию потребления ОЗУ.
- Сделайте скриншот для отчёта.
- Запишите:
  - Название процесса, который использует больше всего памяти.
  - Сколько мегабайт (МБ) он занимает.

- Предположите, почему этот процесс требует столько памяти.

3. Анализ вкладки «Быстродействие»

- Перейдите на вкладку **Быстродействие**.

Шаг 3.1: Фиксация системных показателей

- Сделайте скриншот для отчёта.
- Запишите текущие значения в таблицу (представленна ниже):
  - **Потоков**
  - **Процессов**
  - **Дескрипторов**
- Объясните, что означают эти термины.

Шаг 3.2: Наблюдение за динамикой при запуске браузера

- Закройте все приложения, кроме Диспетчера задач.
- Запустите браузер (например, Google Chrome).
- Вернитесь в Диспетчер задач, сделайте скриншот для отчёта и снова зафиксируйте значения в таблицу:
  - На сколько увеличилось количество потоков и процессов?
  - Как изменилось количество дескрипторов?

Пример таблицы для отчёта:

Показатель	До запуска браузера	После запуска браузера	Изменение
Количество процессов			
Количество потоков			
Количество дескрипторов			

Задание 2: Углублённый анализ процессов через командную строку

1. Запуск командной строки

- Нажмите **Win + R**, введите `cmd.exe` и нажмите **Enter**.

2. Команда `tasklist` — просмотр списка процессов
Шаг 2.1: Изучение команды

- Введите команду:

```
tasklist /?
```

- Сделайте скриншот для отчёта.
- Прочитайте справку, ознакомьтесь с флагами команды

Шаг 2.2: Получение списка процессов

- Введите команду:

```
tasklist
```

Вы увидите таблицу с колонками:

- **Image Name** — имя процесса.
- **PID** — уникальный идентификатор процесса.
- **Session Name** — сессия пользователя.
- **Mem Usage** — потребление памяти.

- Сделайте скриншот для отчёта.

Шаг 2.2: Фильтрация вывода

- Чтобы получить информацию о конкретном процессе (например, `svchost.exe`), выполните:

```
tasklist | findstr "svchost.exe"
```

- Сделайте скриншот для отчёта.
- Запишите в отчёт:
  - Сколько экземпляров `svchost.exe` запущено?
  - Какие PID они имеют?

---

3. Команда `taskkill` — завершение процесса
Шаг 3.1: Запуск Блокнота

- Откройте **Блокнот** через меню «Пуск» или командой:

```
notepad.exe
```

Шаг 3.2: Определение PID Блокнота

- Выполните команду:

```
tasklist | findstr "notepad.exe"
```

Запишите PID процесса (например, `1234`).

Шаг 3.3: Завершение процесса

- Введите команду (замените `<PID>` на реальный номер):

```
taskkill /PID <PID> /F
```

  - Параметр `/F` принудительно завершает процесс.
- Сделайте скриншот для отчёта.
- Убедитесь, что Блокнот закрылся.

4. Практическое задание: Сравнение процессов до и после
Шаг 4.1: Фиксация исходного состояния

- Выполните:

```
tasklist > processes_before.txt
```

Это сохранит список процессов в файл.

Шаг 4.2: Создание нагрузки

- Запустите три окна **Блокнота** и браузер.

Шаг 4.3: Сравнение изменений

- Выполните:

```
tasklist > processes_after.txt
```

- Сравните файлы через команду:

```
fc processes_before.txt processes_after.txt
```

- Сделайте скриншот для отчёта.
- Запишите в отчёт:
  - Какие новые процессы появились?
  - Сколько памяти потребляет браузер?

Задание 3: Детальный анализ процессов и потоков в Process Explorer

1. Установка и запуск Process Explorer

1. Скопируйте **Process Explorer** из сетевой папки (рядом с заданием) к себе на локальный компьютер, например на диск D.
2. В данной работе используется устаревшая portable версия.
3. Запустите программу.

2. Интерфейс Process Explorer

- В верхней части окна отображается дерево процессов. Каждый процесс имеет дочерние элементы (потоки, дескрипторы).
- Цветовая маркировка:
  - **Розовый** — процессы, запущенные от имени администратора.
  - **Синий** — службы Windows.
  - **Жёлтый** — .NET-приложения.
- Сделайте скриншот для отчёта.

3. Анализ процесса svchost.exe

1. В списке процессов найдите **svchost.exe**. Их будет несколько.
2. Раскройте дерево одного из них, нажав на значок `+`.
3. Наведите курсор на процесс **svchost.exe** — во всплывающем окне отобразятся **службы**, связанные с этим экземпляром.
4. Сделайте скриншот для отчёта.
5. Запишите в отчёт:
   - Сколько экземпляров **svchost.exe** запущено?
   - Какие службы связаны с выбранным экземпляром?

- Почему Windows использует один хост-процесс для нескольких служб?

4. Изучение потоков процесса explorer.exe

1. Найдите процесс **explorer.exe** (проводник Windows).
2. Двойной клик по нему → откроется окно свойств.
3. Перейдите на вкладку **Threads** (Потоки):
   - Здесь отображаются все потоки процесса.
   - Сделайте скриншот для отчёта.
   - Колонки:
     - **TID** — идентификатор потока.
     - **Start Address** — адрес в памяти, с которого начал выполняться поток.
     - **Cycles Delta** — количество циклов процессора, использованных потоком.
4. Выберите любой поток и нажмите **Stack** → откроется стек вызовов.
   - Сделайте скриншот для отчёта.
   - Запишите, какие модули (DLL или EXE) участвуют в работе потока.
5. Ответьте на вопросы:
   - Сколько потоков у процесса **explorer.exe**?
   - Какую задачу выполняет поток с наибольшим значением **Cycles Delta**?

5. Определение родительского процесса

1. Запустите приложение **Калькулятор**.
2. В Process Explorer найдите процесс **calc.exe**.
3. Наведите курсор на него → во всплывающем окне будет указан **родительский процесс**.
4. Сделайте скриншот для отчёта.
5. Запишите в отчёт:
   - Какой процесс запустил **calc.exe**?
   - Почему некоторые процессы имеют родительский процесс `services.exe` или `wininit.exe`?

6. Анализ потоков при высокой нагрузке

1. Запустите ресурсоёмкое приложение (например, браузер с видео).
2. В Process Explorer найдите его процесс.
3. Перейдите на вкладку **Threads** и отсортируйте потоки по **CPU Usage**.
4. Сделайте скриншот для отчёта.
5. Запишите:
   - Сколько потоков активно используют CPU?
   - Какие модули (DLL) связаны с этими потоками?

Вопросы для защиты:

1. Чем отличается процесс от потока? Приведите примеры из ваших наблюдений.
2. Какой инструмент точнее показывает зависимость процессов: Process Explorer или Диспетчер задач? Почему?
3. Почему при завершении процесса через `taskkill` могут оставаться "висячие" потоки?
4. Как связаны количество потоков и производительность системы?
5. Почему для завершения процесса используется PID, а не имя?
6. Чем отличается `taskkill /IM notepad.exe /F` от `taskkill /PID 1234 /F`?
7. Какие процессы нельзя завершить через `taskkill`? Почему?

8. Как команда `wmic` помогает обнаружить подозрительные процессы?
9. Почему количество потоков растёт быстрее, чем количество процессов?
10. Как связаны дескрипторы и работа приложений? Приведите примеры ресурсов, которые могут быть связаны с дескрипторами.
11. Почему процесс **System Idle Process** часто показывает высокую загрузку CPU?
12. Почему службы Windows группируются в **svchost.exe**?
13. Чем отличается **Start Address** потока от его TID?
14. Как Process Explorer помогает найти скрытые или вредоносные процессы?
15. Почему при закрытии родительского процесса (например, `explorer.exe`) завершаются и его дочерние процессы?