Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

**Кафедра «Компьютерные интеллектуальные технологии»**

**Лабораторная работа № 3**

**Исследование особенностей реализации многозадачности в ОС семейства Windows**

по дисциплине «Операционные системы»

Выполнил

студент гр. з3530903/90001 А.Ю. Леснов

<*подпись*>

Руководитель

доцент, к.т.н. А.В. Сергеев

<*подпись*>

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 г.

Санкт-Петербург

2021

Указания по выполнению лабораторной работы.

1. Исследование сбоев и зависаний программ при работе компьютера.
   1. Для интерпретации результатов, получающихся при выполнении лабораторной работы, необходимо знать некоторые определения. Запишите в отчёт определение следующих терминов: **контекст потока**, **переключение контекста (Context Switch)**. Укажите о чём свидетельствует увеличение числа переключений контекста, а о чём наоборот отсутствие переключений контекста. Также запишите, что значит **сбой** выполнения программы. Приведите конкретные примеры ситуаций, приводящих к сбою. Напишите, что значит **зависание** программы и когда оно возникает.

**Контекст потока:** Особую роль в структурах данных, описывающих потоки, играет контекст потока. Информацию, входящую в состав контекста, необходимо периодически сохранять и восстанавливать в случае возникновения различных событий, например, при переключении потоков. Обычно сохранению и последующему восстановлению подлежат:

* программный счетчик, регистр состояния и содержимое остальных регистров процессора;
* указатели на стек ядра и пользовательский стек;
* указатели на адресное пространство, в котором выполняется поток (каталог таблиц страниц процесса).

**Переключение контекста:** при исполнении процессором некоторого процесса возникает прерывание от устройства ввода-вывода, сигнализирующее об окончании операций на устройстве. Над выполняющимся процессом производится операция приостановки. Далее ОС разблокирует процесс, инициировавший запрос на ввод-вывод и осуществляет запуск приостановленного или нового процесса, выбранного при выполнении планирования.

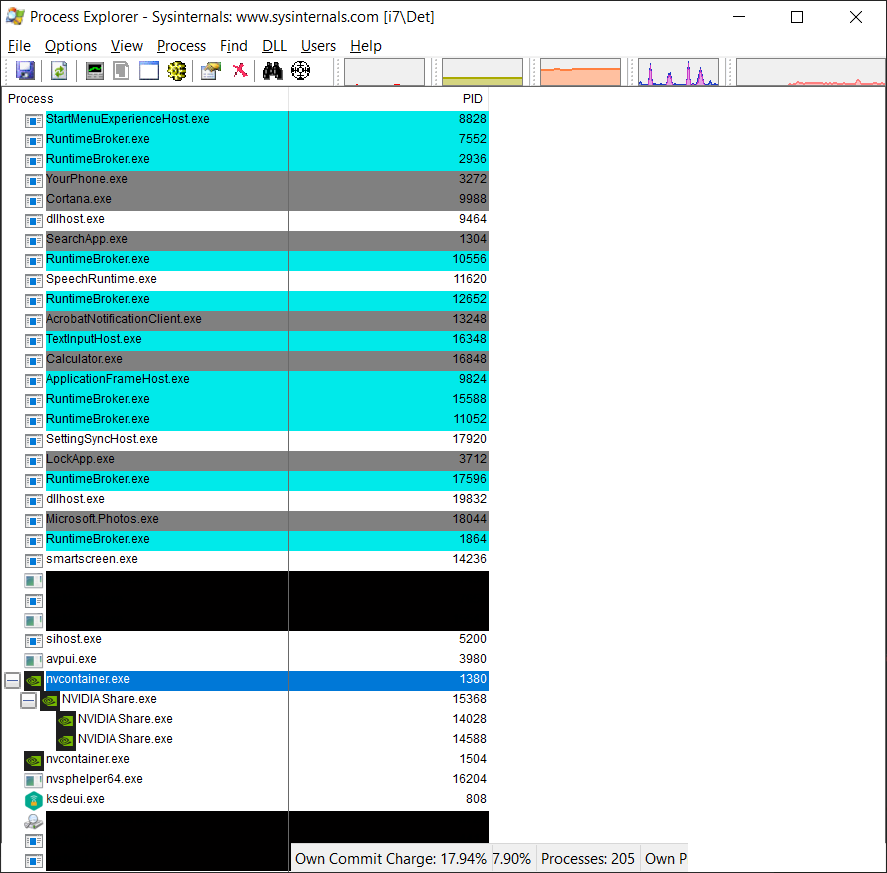
Для корректного переключения процессора с одного процесса на другой необходимо сохранить контекст исполнявшегося процесса и восстановить контекст процесса, на который будет переключен процессор. Такая процедура сохранения/восстановления работоспособности процессов называется переключением контекста.

В процедуру переключения контекста входит т. н. планирование задачи –

процесс принятия решения, какой задаче передать управление.

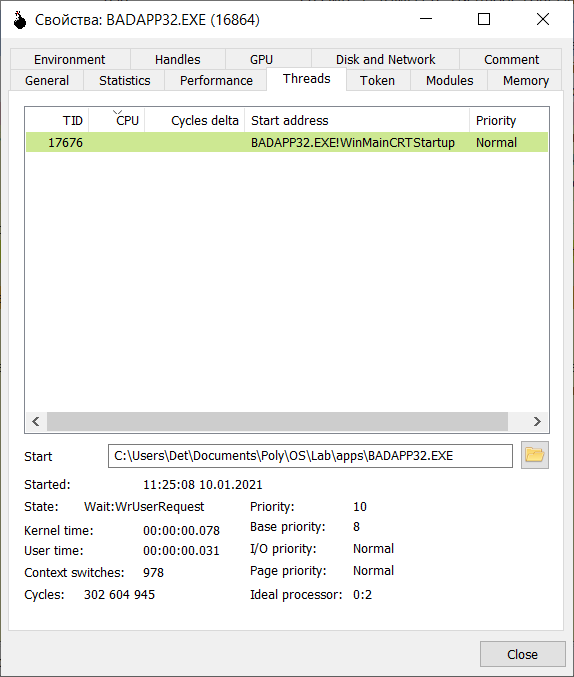
* 1. Для наблюдения за процессами и потоками запустите программу Process Explorer (**procexp.exe**).

Почему-то мой Process Explorer стал выглядеть вот так:



Решения проблемы найти не смог, ничего не помогло. Пользуюсь Process Hacker

* 1. Запустите программу **badapp32.exe**. Эта программа предназначена для моделирования выполнения программой некорректной операции (сбой) и моделирования зависания программы.
  2. Запишите приоритет процесса **badapp32** и приоритет основного потока, принадлежащего этому процессу. Для этого найдите в списке процессов процесс badapp32, щёлкните по нему правой кнопкой мыши и выберите пункт меню **Properties**. В открывшемся окне выберите закладку **Threads**.



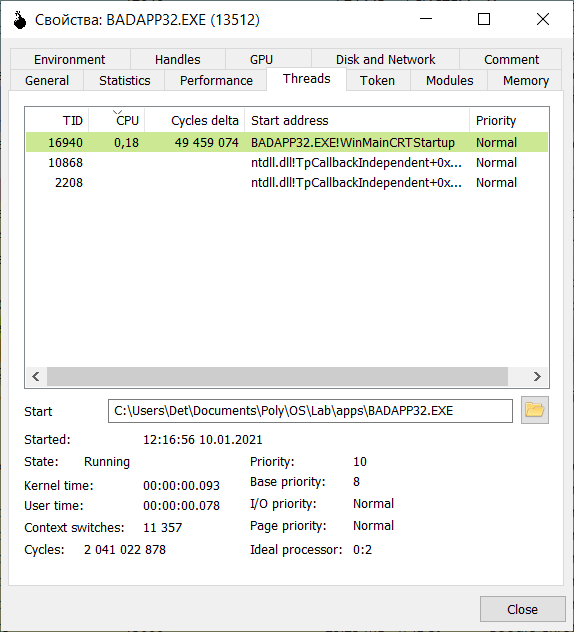
* 1. Проследите, как меняется количество изменений контекста потока (Context Switches) если программа не активна. Каков её статус? Как меняется Context Switches при наведении курсора мыши на окно программы **badapp32** и выборе какого-либо меню? В отчёте приведите объяснение наблюдаемым явлениям.

Статус: Wait:WrUserRequest

Состояние: Выполняется

Context Switches: Без изменений

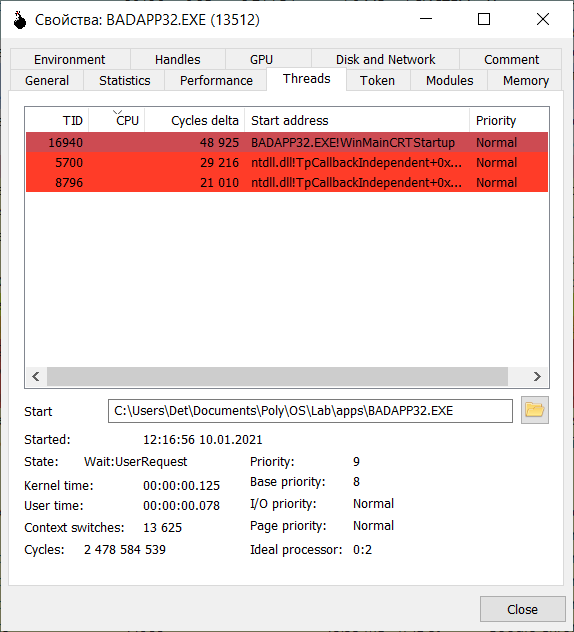
При наведении курсора или выборе меню количество context switches начинает расти. Это происходит в результате пользовательского обращения к процессу (наведение курсора)



* 1. Промоделируйте сбой программы **badapp32**, выбрав пункт меню **Action – GPFault**. В программе Process Explorer отследите изменение состояния потока. Каков его статус? Как меняется Context Switches? Объясните произошедшие изменения.

Статус: Wait:UserRequest, n/a

Context Switches: медленный рост



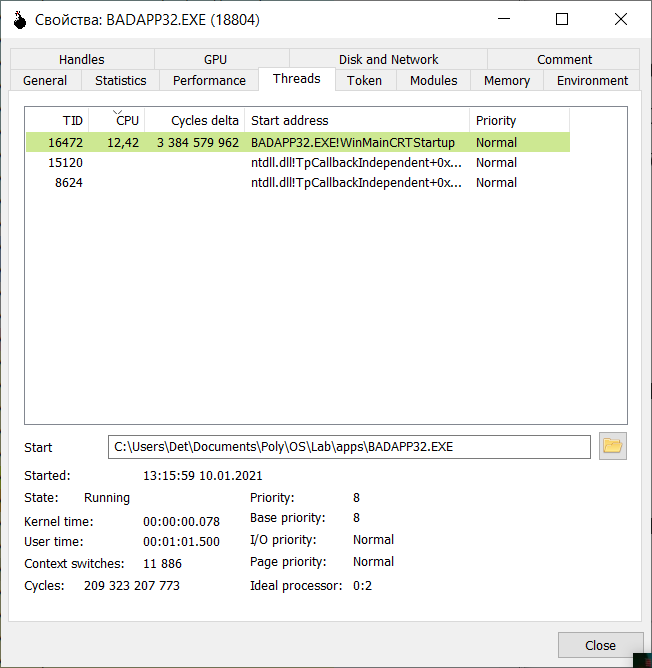
* 1. Объясните, почему сбой одной программы не влияет на другие программы?
  2. Закройте программу **badapp32.exe** и запустите её вновь. Промоделируйте зависание программы, выбрав пункт меню **Action – Hang**. Запишите какой приоритет стал у процесса и потока. Запишите также, какой стал статус у потока и как меняется Context Switches? Объясните получившиеся результаты.

Статус: Running

Состояние: Не отвечает

Context Switches: быстрый рост

Зависание программы, повторяющийся цикл выполнения.



* 1. Объясните, почему зависание одной программы не влияет на другие программы?

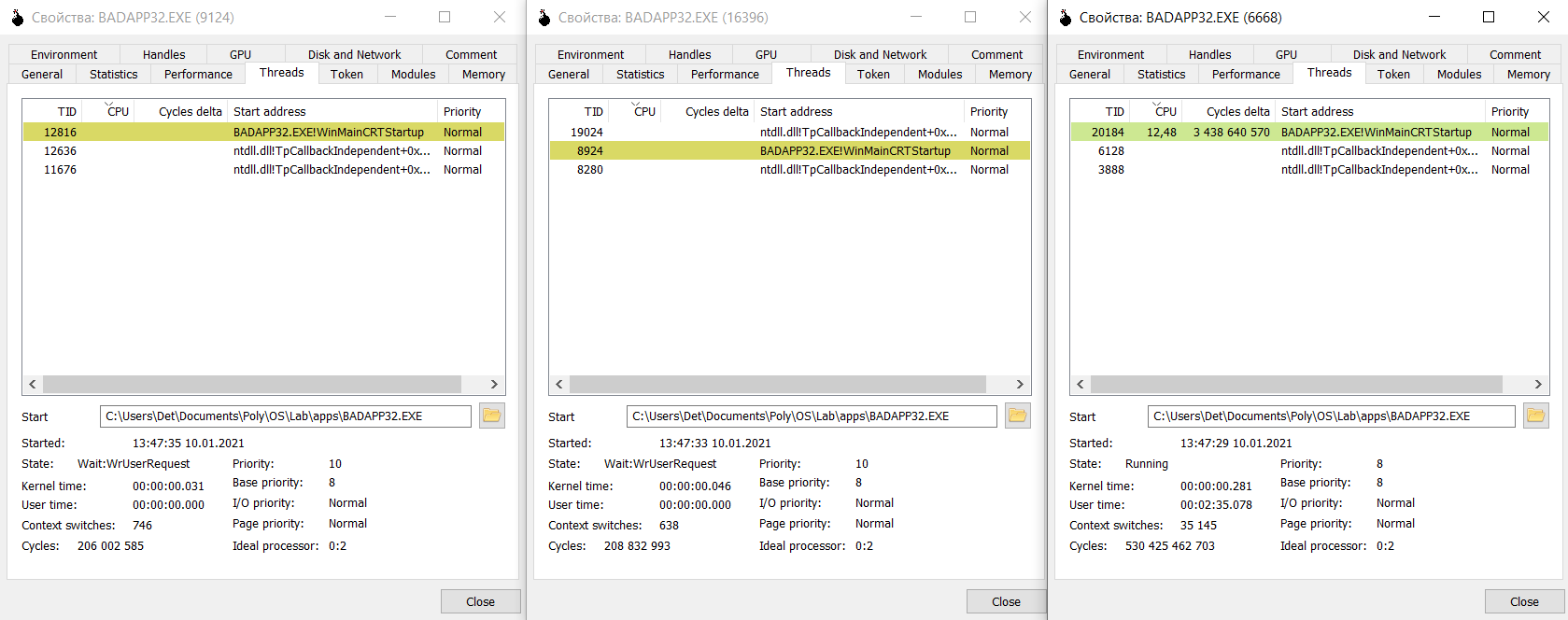
Система поочерёдно выполняет другие программы

1. Исследование свойств кооперативной (невытесняющей) и вытесняющей многозадачности Win32.
   1. Попробуйте запустить программу **badapp16.exe** в 64-х разрядной операционной системе.Почему не удалось её запустить?

64-разрядная версия Windows не поддерживает запуск 16-разрядных приложений, основная причина в том, что дескрипторы имеют 32 значащих бита в 64-битной Windows.

Поэтому дескрипторы не могут быть усечены и переданы 16-разрядным приложениям без потери данных.

* 1. Запустите 3 экземпляра приложения **badapp16** в 32-х разрядной операционной системе и произведите операцию искусственного зависания (Hang) одного из них. Зафиксируйте поведение остальных экземпляров программы. Сделайте вывод и запишите его в отчёт.
  2. Запустите 3 экземпляра приложения **badapp32** и произведите операцию искусственного зависания (Hang) одного из них. Зафиксируйте поведение остальных экземпляров программы. Объясните, получившиеся различия в поведении 16-разрядных и 32разрядных приложений.

Не влияет на поведение остальных экземпляров

1. Исследование поведения процесса в режиме реального времени.
   1. Переведите программу **badapp32** в режим зависания (Hang).
   2. Используя **Диспетчер задач**, задайте процессу badapp32.exe приоритет реального времени.
   3. Используя программу Process Explorer, убедитесь, что приоритет потока управления равен 24.
   4. Запишите, какой статус у потока? Как меняется Context Switches? На каком ядре запущен поток, и какой процент процессорного времени ядра занимает поток? Объясните получившиеся результаты.

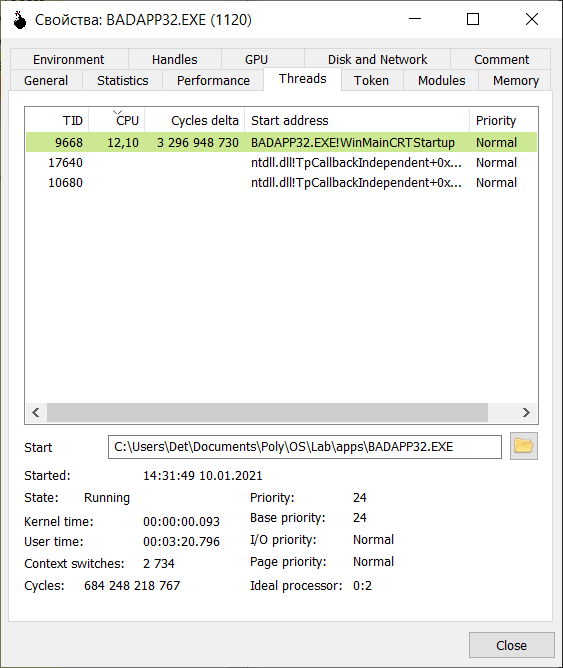
Статус: Running

Состояние: Не отвечает

Context Switches: незначительный рост

Ядро: Ideal processor 2 (3-е ядро)

Незначительный рост Context Switches связан с приоритетом реального времени процесса (над процессом не производится операция приостановки)



1. Определение влияния приоритета процесса на выделяемое этому процессу процессорное время. Для дальнейшей работы надо использовать программу counter. Для того, чтобы программа заработала, надо зарегистрировать в операционной системе библиотеку VB40032.DLL. Для этого надо обладать правами администратора. Если у Вас нет таких прав, то вместо counter можно использовать программу badapp32. В этом случае надо перейти к п. 5. В Windows 7 при запуске программы counter пользователем с правами администратора библиотека зарегистрируется автоматически. В Windows 10 для регистрации библиотеки надо нажать правой кнопкой мыши на файл counter.exe и в контекстном меню выбрать пункт «Запуск от имени администратора». После этого библиотека зарегистрируется, и программа counter запустится. Если программа counter запустилась, то п. 5 выполнять не надо. Запустите два экземпляра приложения **counter.** Определите при помощи утилиты «Диспетчер задач» доли процессорного времени, выделяемого каждому из полученных процессов. Процент процессорного времени, отведённому каждому процессу, представлен в колонке ЦП «Диспетчера задач».

Каждый процесс занимает ~10% процессорного времени

* 1. Измените приоритет одного процесса **counter** на **Высокий**, а другого на **Низкий** и определите доли процессорного времени, выделяемого каждому из полученных процессов. Объясните получившийся результат.

Процессы по-прежнему занимают ~10% процессорного времени. Это связано с тем, что процессы выполняются на разных ядрах.

* 1. Запустите оба процесса **counter** на одном ядре процессора. Для этого в «Диспетчере задач» надо нажать правой кнопкой мыши на процесс **counter** и выбрать из контекстного меню «**Задать соответствие»** (**Задать сходство**). Как изменился процент процессорного времени, отводимый каждому процессу counter? Почему?

Процессорное время приоритета **Высокий**: 6-7%

Процессорное время приоритета **Низкий**: 5-6%

---

* 1. Запустите третий экземпляр программы **counter** на том же процессорном ядре, что и остальные процессы **counter**. Как изменилась доля процессорного времени, отводимого каждому процессу counter? Почему?

Процессорное время приоритета **Высокий**: 6-7%

Процессорное время приоритета **Обычный**: 4-5%

Процессорное время приоритета **Низкий**: 0-1%

* 1. Установите у нового процесса приоритет **Высокий.** Как изменилась доля процессорного времени, отводимого каждому процессу **counter**? Почему?

Процессорное время приоритета **Высокий**: 5-6%

Процессорное время приоритета **Высокий**: 5-6%

Процессорное время приоритета **Низкий**: 0-1%

* 1. Оцените качественно возможность реакции процессов **counter** на интерактивное событие – перетаскивание окна. Есть ли разница при перетаскивании разных окон **counter**? Объясните результат. Объясните, почему при перетаскивании окна **counter** останавливается счётчик.

Процессорное время приоритета **Высокий**: повышается до ~10%

Процессорное время приоритета **Высокий**: повышается до ~10%

Динамический приоритет повышается до 15, что выше, чем 13 у другого процесса с высоким уровнем приоритета. Во время повышения процессорного времени одного из процессов до ~10%, время процессорного времени другого понижается до ~1-2%

Процессорное время приоритета **Низкий**: повышается до ~1%

Динамический приоритет повышается до 6-7, это меньше чем у процессов с высоким приоритетом. По этой причине процессорное время остается приблизительно на прежнем уровне