Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

**Кафедра «Компьютерные интеллектуальные технологии»**

**Лабораторная работа № 2**

**Древовидная структура процессов в ОС Windows**

**Объектная модель в ОС 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

1. При помощи утилиты **Process Explorer** (procexp.exe) исследуйте дерево процессов и постройте соответствующий граф. Так как полный граф получается большой, то на каждом уровне достаточно изобразить 4 процесса. На графе должны быть указаны наименования процессов, идентификаторы процессов и количество потоков в каждом процессе. Всего процессов в графе должно быть не менее 20.

Memory Compression:

Функция сжатия оперативной памяти в Windows 10 предназначена для ускорения работы системы за счет хранения части страниц в оперативной памяти в сжатом виде. Тем самым достигается уменьшение количества обращений на чтение и запись страниц памяти из медленного (по сравнению с RAM) файла подкачки на жестком диске. Нужные данные извлекаются из более быстрой оперативной памяти быстрее, даже с учетом того, что на их сжатие/декомпрессию тратятся дополнительные ресурсы процессора.

Количество потоков считал в командной строке с помощью команды:

wmic process where (Caption like '%explorer%') get \* /format:list

(строка ThreadCount=..)

Name: NVIDIA Web Helper.exe

PID: 10720

Threads: 94

Name: explorer.exe

PID: 7392

Threads: 90

Name: System Idle Process

PID: 0

Threads: 8

Name: conhost.exe

PID: 11520

Threads: 4

Name: SecurityHealthSystray.exe

PID: 9848

Threads: 1

Name: System

PID: 4

Threads: 247

Name: RtkNGUI64.exe

PID: 9880

Threads: 6

Name: smss.exe

PID: 536

Threads: 2

Name: wininit.exe

PID: 88

Threads: 1

Name: RAVBg64.exe

PID: 10000

Threads: 4

Name: Memory Compression

PID: 1692

Threads: 42

Name: fontdrvhost.exe

PID: 696

Threads: 5

Name: AdobeCollabSync.exe

PID: 8620

Threads: 13

Name: lsass.exe

PID: 720

Threads: 9

Name: winlogon.exe

PID: 884

Threads: 7

Name: services.exe

PID: 644

Threads: 7

Name: nusb3mon.exe

PID: 10536

Threads: 4

Name: fontdrvhost.exe

PID: 836

Threads: 5

Name: acrotray.exe

PID: 10712

Threads: 2

Name: dwm.exe

PID: 1240

Threads: 19

* 1. Запустите программу procexp.exe. Появится окно, разделённое на две части. В верхней части представлен список процессов. В нижней части находится список описателей объектов (handle) для процесса или список используемых dll в зависимости от настройки.
  2. Все процессы надо упорядочить в виде дерева для этого надо выбрать пункт меню **View-Show Process Three.** Если этот пункт меню не активный, то процессы уже представлены в виде дерева.
  3. Для определения потоков процесса надо выделить требуемый процесс и нажать правую кнопку мыши. В контекстном меню надо выбрать **Properties** и закладку **Threads.** На этой вкладке показаны стартовые адреса всех потоков процесса, а также их ID.
  4. В отчёте приведите описание любой системной службы, представленной на графе. Объясните для чего она нужна и особенности работы.

1. Определение имён системных служб, использующих те или иные процессы. Требуется заполнить следующую таблицу.

svchost.exe – это главный процесс (Service Host process) для системных служб (сервисов), загружаемых из динамических библиотек (файлов .dll)

Использование единого процесса для работы нескольких сервисов позволяет существенно уменьшить затраты оперативной памяти и процессорного времени.

Все копии svchost.exe запускаются системным процессом services.exe (родительским процессом) в ходе начальной загрузки, инициализации системы и при необходимости, в ходе выполнения запросов пользовательских или системных задач. Таким образом, каждый процесс svchost.exe – это одна из системных служб Windows.

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя системной службы** | **Описание** | **Имя процесса** | **PID** |
| DcomLaunch  PlugPlay  BrokerInfrastructure  SystemEventsBroker  Power | Модуль запуска процессов DCOM-сервера  Plug and Play  Служба инфраструктуры фоновых задач  Брокер системных событий  Питание | svchost.exe | 1016 |
| AtherosSvc | AtherosSvc | AdminService.exe | 4000 |
| NetPipeActivator | Адаптер прослушивателя Net.Pipe | SMSvcHost.exe | 3636 |
| KSDE5.1 | Kaspersky Secure Connection Service 5.1 | ksde.exe | 2788 |
| WO\_LiveService2 | Ashampoo LiveTuner 2 Service | LiveTunerService.exe | 5776 |
| SecurityHealthService | Служба "Безопасность Windows" | SecurityHealthService.exe | 7852 |
| KeyIso  SamSs  VaultSvc | Изоляция ключей CNG  Диспетчер учетных записей безопасности  Диспетчер учетных данных | lsass.exe | 720 |
| NVDisplay.ContainerLocalSystem | NVIDIA Display Container LS | NVDisplay.Container.exe | 1872 |
| Spooler | Диспетчер печати | spoolsv.exe | 3792 |
| AdobeARMservice | Adobe Acrobat Update Service | armsvc.exe | 4020 |
| MSMQ | Очередь сообщений | mqsvc.exe | 4504 |

* 1. Процессы системных служб в программе **Process Explorer** выделены розовым цветом. Сопоставление процесса со службой можно выполнить, используя диспетчер задач.
  2. Запустите диспетчер задач и выберите вкладку Службы. На вкладке представлен список служб и их описание.
  3. Используя PID, можно сопоставить процесс и службу, выполняющуюся в процессе. В таблицу требуется занести информацию о 10 системных службах.
  4. Как видно запущено несколько процессов svchost.exe. Каждый процесс svchost.exe содержит несколько системных служб. Объясните, что из себя представляет процесс svchost.exe, зачем он нужен? Почему один процесс содержит несколько системных служб?

1. Исследование связи родительских и дочерних процессов.
   1. Запустите окно командной строки **cmd.exe**.
   2. Измените заголовок окна на Parent, набрав команду **title Parent**.
   3. Из запущенного окна командной строки запустим ещё одно окно командной строки. Для этого наберите команду **start cmd**.
   4. Переименуйте новое окно, набрав команду **title Child.**
   5. Перейдите в «Диспетчер задач». Откройте вкладку «Процессы», если используется ОС Windows 10, или «Приложения», если используется Windows 7. Щёлкните правой кнопкой мыши на задаче Parent и выберите пункт «Подробно» ("Перейти к процессу"). Диспетчер задач Windows 10 не показывает имя окна в отличие от Windows 7. В этом случае следует учитывать, что процесс Parent располагается ниже Child в списке приложений.
   6. Щёлкните правой кнопкой мыши на процессе cmd.exe и выберите пункт "Завершить дерево процессов". Почему завершились оба процесса?

Процессы **Parent** и **Child** были частью одного дерева процессов, завершив дерево мы завершили все процессы внутри.

* 1. Запустим опять окно командной строки: **Пуск – Выполнить – cmd**.
  2. Повторите пункты 3.2 – 3.4.
  3. В окне **Child** запустите программу Paint, набрав **mspaint**.
  4. После запуска программы Microsoft Paint завершите работу окна **Child.** Для этого в окне **Child** надо набрать команду **exit**. Окно **Child** закрылось, а программа Paint продолжает работать. Почему?

**Paint** – не зависящий от **cmd** процесс, поэтому закрывая **cmd** мы не влияем на **paint**. К тому же был закрыт только процесс, а не всё дерево.

* 1. Перейдите в «Диспетчер задач». Откройте вкладку «Процессы» («Приложения»). Щёлкните правой кнопкой мыши на задаче Parent и выберите пункт «Подробно» ("Перейти к процессу").
  2. Щёлкните правой кнопкой мыши на процессе cmd.exe и выберите пункт "Завершить дерево процессов". Почему программа Paint продолжила работу?

Закрыв **Child** на предыдущем шаге, мы сделали **paint** самостоятельным процессом (не принадлежащим какому-либо дереву). Поэтому **paint** продолжил работу. Если бы мы на предыдущем шаге завершили дерево процессов, **paint** был бы тоже завершён.

1. Исследование динамического приоритета дочерних процессов.
   1. Запустите окно командной строки **cmd.exe**.
   2. Измените приоритет процесса **cmd.exe** на Below Normal.
   3. Командой **start cmd** создайте дочерний процесс.
   4. Определите динамический приоритет дочернего процесса. Объясните получившийся результат.

Дочерний процесс был создан с тем же (Below Normal) приоритетом. Родительский процесс не может создать дочерний с уровнем приоритета выше своего.

* 1. Закройте дочерний процесс, а приоритет родительского процесса измените на Above Normal.
  2. Командой **start cmd** создайте дочерний процесс.
  3. Определите динамический приоритет дочернего процесса. Объясните получившийся результат.

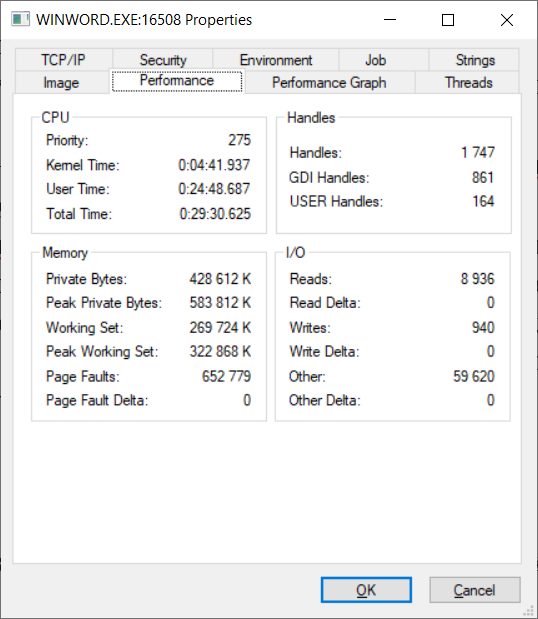
Дочерний процесс был создан с приоритетом Normal. Родительский процесс может создать дочерний с уровнем приоритета ниже своего, приоритет по-умолчанию для **cmd**: Normal.

1. Исследование объектов, принадлежащих процессу.

Объекты, принадлежащие процессу, описываются дескрипторами или описателями (handles).

Описатели каждого процесса представлены в нижнем окне утилиты **Process Explorer.**

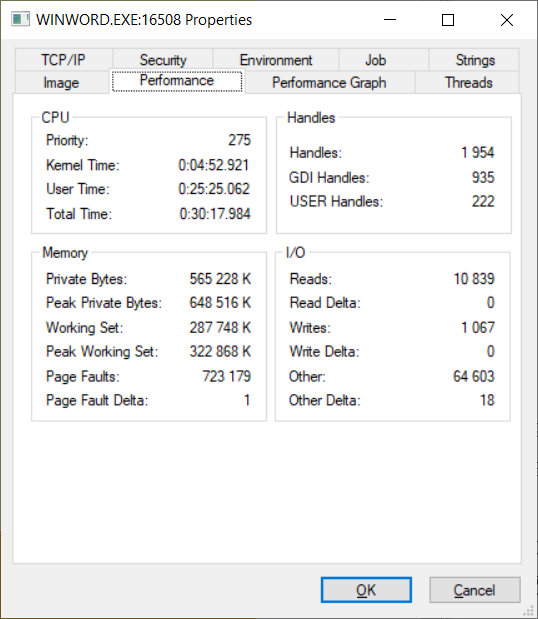
* 1. Запустите программу Microsoft Word.
  2. Найдите в **Process Explorer** процесс **WINWORD.EXE**. Определите общее количество объектов, принадлежащих данному процессу. Для этого надо выделить требуемый процесс и нажать правую кнопку мыши. В контекстном меню надо выбрать **Properties** и закладку **Performans**. Параметр **Handles** показывает число описателей. Каждый описатель определяет объект.



* 1. Определите количество разных типов объектов, принадлежащих процессу **WINWORD.EXE**.

Типы Handles:

1. WindowStation
2. Thread
3. Semaphore
4. Section
5. Process
6. Mutant
7. Key
8. File
9. Event
10. Directory
11. Desktop
12. ALPC Port
    1. Откройте документ в запущенной программе **WINWORD.EXE**. Как изменилось количество объектов, принадлежащих данному процессу? Какие новые объекты типа **File**, связанные с открытым файлом, появились?



Появились новые объекты, связанные с открытым документом.

Запустите программу **Блокнот (Notepad.exe)**. Откройте текстовый файл. Определите, какие появились объекты типа **File**, связанные с открытым файлом. Почему отсутствует объект, связанный с открытым файлом?

Процесс **Notepad.exe** поддерживает лишь один открытый файл (процесс и является объектом).  
(Вероятно, я тут в чём-то ошибаюсь. Не смог до конца разобраться с со списком **Handles)**

