Операционные системы.
Лабораторная работа №2.

Древовидная структура процессов в ОС Windows
Объектная модель в ОС Windows

Цель работы: Изучить древовидную структуру процессов в ОС Windows. Исследовать зависимость дочернего процесса от родительского. Рассмотрение особенностей реализации объектной модели в ОС Windows.

Методические указания.

Операционные системы типа Windows или Linux содержат множество процессов. Кроме них в системе присутствуют процессы драйверов устройств и прикладных программ. При старте операционной системы создаются системные процессы и запускаются на выполнение. При этом процессы создаются не одновременно, а последовательно. Первые процессы создаёт загрузчик ОС, который находится в загрузочном секторе жёсткого диска. Загрузчик запускает один или несколько процессов и заканчивает свою работу. Запущенные процессы создают другие процессы, а те в свою очередь создают ещё процессы. Так продолжается пока не будут созданы все процессы, необходимые для функционирования операционной системы. Процесс, инициировавший создание нового процесса, принято называть родительским процессом, а созданный процесс – дочерним процессом. Как правило один родительский процесс при загрузке ОС рождает несколько дочерних процессов. Таким образом образуется древовидная структура процессов.

При запуске пользователем прикладных программ на выполнение создаются новые дочерние процессы. При этом для нового процесса родительским является один из уже запущенных процессов. В Windows родительским процессом для прикладных программ, как правило, является Explorer (Проводник).

Explorer является основой графической оболочки пользователя в Windows. Explorer создаёт для пользователя значки рабочего стола, панель задач, меню «Пуск». Explorer также реализует графический интерфейс доступа пользователя к файлам. Explorer или проводник предоставляет пользователю возможность выполнять различные операции над файлами. Таким образом, Explorer является также файловым менеджером.

Дочерние процессы наследуют класс динамического приоритета родительского процесса, если приоритет родительского процесса Normal, Below Normal или Idle. Если динамический приоритет родительского процесса выше Normal, то приоритет дочернего процесса по умолчанию будет Normal. При создании дочернего процесса программист может присвоить ему требуемый приоритет соответствующей командой. Изменить приоритет работающего процесса можно используя специальную функцию API Windows или диспетчер задач.

Windows реализует объектную модель. Это значит, всё с чем тем или иным образом взаимодействует процесс, является объектом. Объект – это файл, папка, процесс, поток, таймер, семафор, динамически выделенная область памяти, периферийное устройство, рабочий стол, окно, меню и т. д. Объектная модель позволяют абстрагироваться от структуры конкретного объекта и стандартизировать методы управления объектами и обмена данными с ними. Есть три категории объектов: объекты пользователя, объекты графического интерфейса, объекты ядра. Количество типов объектов разных категорий увеличивается с каждой новой версией Windows. В настоящее время в Windows 10 определено 46 типов объектов.

Для повышения надёжности и работы ОС, а также для обеспечения разграничения прав доступа обращение к объектам напрямую запрещено. Диспетчер объектов создаёт ссылку на объект, которая называется описателем (handle). Этот описатель позволяет обратиться к объекту, используя функции ОС. Нередко описатель называют дескриптором. В частности, в русифицированной версии Windows описатель называется дескриптором. Это не очень удачный перевод, т. к. слово дескриптор имеет ещё другие значения, относящиеся к операционным системам и программированию.

Каждый поток при обращении к объекту получает свой описатель. Это сделано, с одной

стороны, для обеспечения разграничения прав доступа. С другой стороны, такой подход защищает объект от одновременного обращения к объекту нескольких разных потоков.

Для выполнения лабораторной работы понадобится утилита **Process Explorer** (procexp.exe). По результатам выполнения лабораторной работы надо написать отчёт и показать его преподавателю. Для облегчения оценки выполненной работы рекомендуется по пунктам приводить текст задания и тут же результат выполнения.

Указания по выполнению лабораторной работы

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.

1.1. 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.

1.2. 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.

1.3. 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.

1.4. В отчёте приведите описание любой системной службы, представленной на графе. Объясните для чего она нужна и особенности работы.

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

Имя системной службы	Описание	Имя процесса	PID

2.1. Процессы системных служб в программе **Process Explorer** выделены розовым цветом. Сопоставление процесса со службой можно выполнить, используя диспетчер задач.

2.2. Запустите диспетчер задач и выберите вкладку Службы. На вкладке представлен список служб и их описание.

2.3. Используя PID, можно сопоставить процесс и службу, выполняющуюся в процессе. В таблицу требуется занести информацию о 10 системных службах.

2.4. 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?

3. Исследование связи родительских и дочерних процессов.

3.1. Запустите окно командной строки **cmd.exe**.

3.2. Измените заголовок окна на Parent, набрав команду **title Parent**.

3.3. Из запущенного окна командной строки запустим ещё одно окно командной строки. Для этого наберите команду **start cmd**.

3.4. Переименуйте новое окно, набрав команду **title Child.**

3.5. 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.

3.6. Щёлкните правой кнопкой мыши на процессе cmd.exe и выберите пункт "Завершить дерево процессов". Почему завершились оба процесса?

3.7. Запустим опять окно командной строки: **Пуск – Выполнить – cmd**.

3.8. Повторите пункты 3.2 – 3.4.

3.9. В окне **Child** запустите программу Paint, набрав **mspaint**.

3.10. 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?

3.11. Перейдите в «Диспетчер задач». Откройте вкладку «Процессы» («Приложения»). Щёлкните правой кнопкой мыши на задаче Parent и выберите пункт «Подробно» ("Перейти к процессу").

3.12. Щёлкните правой кнопкой мыши на процессе cmd.exe и выберите пункт "Завершить дерево процессов". Почему программа Paint продолжила работу?

4. Исследование динамического приоритета дочерних процессов.
   4.1. Запустите окно командной строки **cmd.exe**.
   4.2. Измените приоритет процесса **cmd.exe** на Below Normal.
   4.3. Командой **start cmd** создайте дочерний процесс.
   4.4. Определите динамический приоритет дочернего процесса. Объясните получившийся результат.
   4.5. Закройте дочерний процесс, а приоритет родительского процесса измените на Above Normal.
   4.6. Командой **start cmd** создайте дочерний процесс.
   4.7. Определите динамический приоритет дочернего процесса. Объясните получившийся результат.

5. Исследование объектов, принадлежащих процессу.

Объекты, принадлежащие процессу, описываются дескрипторами или описателями (handles). Описатели каждого процесса представлены в нижнем окне утилиты **Process Explorer.**

   5.1. Запустите программу Microsoft Word.
   5.2. 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.
   5.3. Определите количество разных типов объектов, принадлежащих процессу **WINWORD.EXE**.
   5.4. Откройте документ в запущенной программе **WINWORD.EXE**. Как изменилось количество объектов, принадлежащих данному процессу? Какие новые объекты типа **File**, связанные с открытым файлом, появились?
   5.5. 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?