Тема 12. Структуры программных процессов

Процесс в Windows описывается структурой данных **EPROCES**. Поля структуры:

Pcb (Process Control Block – блок управления процессом) – представляет собой структуру KPROCESS, хранящую данные, необходимые для планирования потоков, в том числе указатель на список потоков процесса.

CreateTime и **ExitTime** – время создания и завершения процесса.

UniqueProcessId – уникальный идентификатор процесса.

ActiveProcessLinks – элемент двунаправленного списка (тип LIST_ENTRY), содержащего активные процессы.

QuotaUsage, QuotaPeak, CommitCharge – квоты (ограничения) на используемую память.

ObjectTable – таблица дескрипторов процесса.

Token – маркер доступа.

ImageFileName – имя исполняемого файла.

ThreadListHead – двунаправленный список потоков процесса.

Peb (Process Environment Block – блок переменных окружения процесса) – информация об образе исполняемого файла.

PriorityClass – класс приоритета процесса.

Структура для потока в Windows − **ETHREAD** . Её основные поля следующие:

Tcb (Thread Control Block – блок управления потоком) – поле, которое является структурой типа KTHREAD и необходимо для планирования потоков.

CreateTime и **ExitTime** – время создания и завершения потока.

Cid – структура типа CLIENT_ID, включающая два поля – идентификатор процесса-владельца данного потока и идентификатор самого потока.

ThreadsProcess – указатель на структуру EPROCESS процесса-владельца.

StartAddress – адрес системной стартовой функции потока. При создании потока сначала вызывается системная стартовая функция, которая запускает пользовательскую стартовую функцию.

Win32StartAddress – адрес пользовательской стартовой функции.

Процессы создаются либо пользователем, либо другим процессом, либо автоматически при загрузке операционной системы.

Процесс, создавший другой процесс, называется родителем, а созданный процесс – потомком. Таким образом, формируется иерархия процессов.

Любой процесс начинает свою работу с основного (main), или первичного, потока, который может запускать (порождать) другие потоки – так образуется иерархия потоков.

В Windows для создания процессов обычно используется функция CreateProcess.

Создание процессов в Windows включает 7 этапов.

1. Проверка и преобразование параметров.

Параметры функции CreateProcess проверяются на корректность и преобразуются к внутреннему формату системы.

2. Открытие исполняемого файла.

Происходит поиск файла, который содержит запускаемую программу. Обычно это файл с расширением .EXE, но могут быть также расширения .COM, .PIF, .BAT, .CMD. Определяется тип исполняемого файла:

- Windows приложение (.EXE) – продолжается нормальное создание процесса;
- приложение MS-DOS или Win16 (.EXE, .COM, .PIF) – запускается образ поддержки Ntvdm.exe;
- командный файл (.BAT, .CMD) – запускается образ поддержки Cmd.exe;
- приложение POSIX – запускается образ поддержки Posix.exe.

3. Создание объекта "Процесс".

Формируются структуры данных EPROCESS, KPROCESS, PEB, инициализируется адресное пространство процесса. Для этого вызывается системная функция NtCreateUserProcess.

Основные действия, выполняемые функцией PspCreateProcess:

- Если в параметрах функции PspCreateProcess указан процесс-родитель, то по его дескриптору определяется указатель на объект EPROCESS и наследуется от процесса родителя маска привязки к процессорам.
- Устанавливается минимальный и максимальный размеры рабочего набора (WorkingSetMinimum = 20 МБ и WorkingSetMaximum = 45 МБ.
- Создается объект "Процесс" (структура EPROCESS) при помощи функции ObCreateObject (строка 1108).
- Инициализируется двунаправленный список потоков при помощи функции InitializeListHead.
- Копируется таблица дескрипторов родительского процесса.
- Создается структура KPROCESS при помощи функции KeInitializeProcess. Маркер доступа и другие данные, связанные с безопасностью копируются из родительского процесса.

- Устанавливается приоритет процесса, равный Normal; однако, если приоритет родительского процесса был Idle или Below Normal, то данный приоритет наследуется.
- Инициализируется адресное пространство процесса.
- Генерируется уникальный идентификатор процесса (функция ExCreateHandle) и сохраняется в поле UniqueProcessId структуры EPROCESS).
- Создается блок PEB и записывается в соответствующее поле структуры EPROCESS. Созданный объект вставляется в хвост двунаправленного списка всех процессов и в таблицу дескрипторов. Первая вставка обеспечивает доступ к процессу по имени, вторая – по ID. Определяется время создания процесса (функция KeQuerySystemTime) и записывается в поле CreateTime структуры EPROCESS.

4. Создание основного потока.

Формируется структура данных ETHREAD, стек и контекст потока, генерируется идентификатор потока. Поток создается при помощи функции NtCreateThread, которая вызывает функцию PspCreateThread. При этом выполняются следующие действия:

- Создается объект ETHREAD.
- Заполняются поля структуры ETHREAD, связанные с процессом-владельцем, – указатель на структуру EPROCESS (ThreadsProcess) и идентификатор процесса (Cid.UniqueProcess).
- Генерируется уникальный идентификатор потока (функция ExCreateHandle) и сохраняется в поле Cid.UniqueThread структурыEPROCESS.
- Заполняются стартовые адреса потока, системный (StartAddress) и пользовательский (Win32StartAddress).
- Инициализируются поля структуры KTHREAD при помощи вызова функции KeInitThread.
- Функция KeStartThread заполняет остальные поля структуры ETHREAD и вставляет поток в список потоков процесса.

Если при вызове функции PspCreateThread установлен флаг CreateSuspended ("Приостановлен") поток переводится в состояние ожидания (функция KeSuspendThread); иначе вызывается функция KeReadyThread, которая ставит поток в очередь готовых к выполнению потоков.

5. Уведомление подсистемы Windows.

Подсистеме Windows отправляется сообщение о вновь созданных процессе и его основном потоке, в которое входят их дескрипторы, идентификаторы и другая информация. Подсистема Windows добавляет новый процесс в общий список всех процессов и готовится к запуску основного потока.

6. Запуск основного потока.

Основной поток стартует, но начинают выполняться системные функции, завершающие создание процесса – осуществляется егоинициализация.

7. Инициализация процесса.

- проверяется, не запущен ли процесс в отладочном режиме;
- проверяется, следует ли производить предвыборку блоков памяти (тех участков памяти, которые при прошлом запуске использовались в течение первых 10 секунд работы процесса);
- инициализируются необходимые компоненты и структуры данных процесса, например, диспетчер кучи;
- загружаются динамически подключаемые библиотеки (DLL – Dynamic Link Library);
- начинается выполнение стартовой функции потока.