


3 ПРОЦЕССЫ И ПОТОКИ

3.2.1 Понятия «процесс» и «поток»

Чтобы поддерживать мультипрограммирование, ОС должна определить и оформить для себя те внутренние единицы работы, между которыми будет разделяться процессор и другие ресурсы компьютера. В настоящее время в большинстве операционных систем определены два типа единиц работы.

Под **процессом** понимается программа в стадии выполнения.



Процесс можно рассматривать также как единицу работы для процессора. **Процесс** представляет собой объект – структуру данных, содержащую информацию, необходимую для выполнения программы. Для современных типов процессоров существует и более мелкая единица работы – поток или нить. Другими словами процесс может породить один и более потоков. В операционных системах, где существуют и процессы, и потоки, процесс рассматривается операционной системой как заявка на потребление всех видов ресурсов, кроме одного – процессорного времени.

Этот ресурс распределяется операционной системой между другими единицами работы – **потоками**, которые и получили свое название благодаря тому, что они **представляют собой последовательности (потоки выполнения) команд**.

Потоки (thread) – объекты, которым операционная система предоставляет процессорное время.


Для того чтобы процессы не могли вмешаться в распределение ресурсов, а также не могли повредить коды и данные друг друга, важнейшей задачей ОС является изоляция одного процесса от другого. Для этого операционная система обеспечивает каждый процесс отдельным виртуальным адресным пространством, так что ни один процесс не может получить прямого доступа к командам и данным другого процесса.

Виртуальное адресное пространство процесса – это совокупность адресов, которыми может манипулировать программный модуль процесса.

Потоки возникли в операционных системах как средство распараллеливания вычислений.

Современные ОС используют механизм многопоточной обработки (multithreading).

Понятию «поток» соответствует последовательный переход процессора от одной команды программы к другой. ОС распределяет процессорное время между потоками.



Процессу ОС назначает адресное пространство и набор ресурсов, которые совместно используются всеми его потоками.

Задача, оформленная в виде нескольких потоков в рамках одного процесса, может быть выполнена быстрее за счет псевдопараллельного (или параллельного в мультипроцессорной системе) выполнения отдельных ее частей.

Особенно эффективно можно использовать многопоточность для выполнения распределенных приложений.


Примеры.

- В современных браузерах каждой вкладке соответствует свой поток.
- В текстовом редакторе один поток может управлять вводом текста, второй – проверять орфографию, третий – выводить документ на принтер.
- В компьютерной игре-стратегии за обработку действий каждого игрока отвечает отдельный поток.

3.2.2 Создание процессов и потоков

Когда операционная система собирается добавить новый процесс к тем, которые уже состоят на учете, она создает структуры данных, используемые при управлении этим процессом и размещает его адресное пространство в основной памяти. С помощью этих действий и создается новый процесс.


В среде пакетной обработки процесс создается в ответ на поступление задания; в интерактивной среде процесс создается при попытке нового пользователя войти в систему.



Кроме того, операционная система может создавать процесс по требованию приложения. Например, если пользователь отправляет запрос на распечатку файла, операционная система может создать процесс, управляющий печатью.

Традиционно операционная система создает все процессы незаметно для пользователя или приложения; такой способ принят во многих современных операционных системах.


Однако иногда требуется, чтобы один процесс мог послужить причиной создания другого процесса.



Создание операционной системой процесса по явному запросу другого процесса называется **порождением процесса** (process spawning).


Когда один процесс порождает другой, то порождающий процесс называется **родительским**, или **предком** (parent), а порождаемый процесс — **дочерним**, или **потомком** (child). Обычно «родственные» процессы обмениваются между собой информацией и взаимодействуют друг с другом.

Создать процесс – это, прежде всего, означает создать **описатель** (handle – дескриптор) процесса, в качестве которого выступает одна или несколько информационных структур, содержащих все сведения о процессе, необходимые операционной системе для управления им. В число таких сведений могут входить, например, идентификатор процесса, данные о расположении в памяти исполняемого модуля, степень привилегированности процесса (приоритет и права доступа) и т. п.



Создание процесса включает загрузку кодов и данных исполняемой программы данного процесса с диска в оперативную память.

В многопоточной системе при создании процесса ОС создает для каждого процесса как минимум один поток выполнения. При создании потока так же, как и при создании процесса, операционная система генерирует специальную информационную структуру – **описатель потока**, который содержит идентификатор потока, данные о правах доступа и приоритете, о состоянии потока и другую информацию.



При управлении процессами операционная система использует два основных типа информационных структур: **дескриптор процесса** и **контекст процесса**. **Дескриптор процесса** содержит такую информацию о процессе, которая необходима ядру в течение всего жизненного цикла процесса независимо от того, находится он в активном или пассивном состоянии, находится образ процесса в оперативной памяти или выгружен на диск.

Дескрипторы отдельных процессов объединены в список, образующий таблицу процессов.

Память для таблицы процессов отводится динамически в области ядра. На основании информации, содержащейся в таблице процессов, операционная система осуществляет планирование и синхронизацию процессов. В дескрипторе содержится информация о состоянии процесса, о расположении образа процесса в оперативной памяти и на диске, о значении отдельных составляющих приоритета, а также о его итоговом значении — глобальном приоритете, об идентификаторе пользователя, создавшего процесс, о родственных процессах, о событиях, осуществления которых ожидает данный процесс, и другая информация.

Контекст процесса содержит менее оперативную, но более объемную часть информации о процессе, необходимую для возобновления выполнения процесса с прерванного места: содержимое регистров процессора, коды ошибок выполняемых процессором системных вызовов, информация обо всех открытых данным процессом файлах и незавершенных операциях ввода-вывода и другие данные, характеризующие состояние вычислительной среды в момент прерывания. Контекст, так же как и дескриптор процесса, доступен только программам ядра, то есть находится в виртуальном адресном пространстве операционной системы.

3.2.3 Завершение процессов

В любой компьютерной системе должны быть средства, позволяющие определить, закончилось выполнение процесса или нет.

Например, в системе с разделением времени процесс пользователя должен быть завершен, когда пользователь выходит из системы или выключает терминал. На персональном компьютере пользователь может выйти из приложения (например, закрыть программу обработки текста или электронную таблицу). Все эти действия в конечном счете приведут к тому, что будет вызвана служба операционной системы, завершающая процесс.

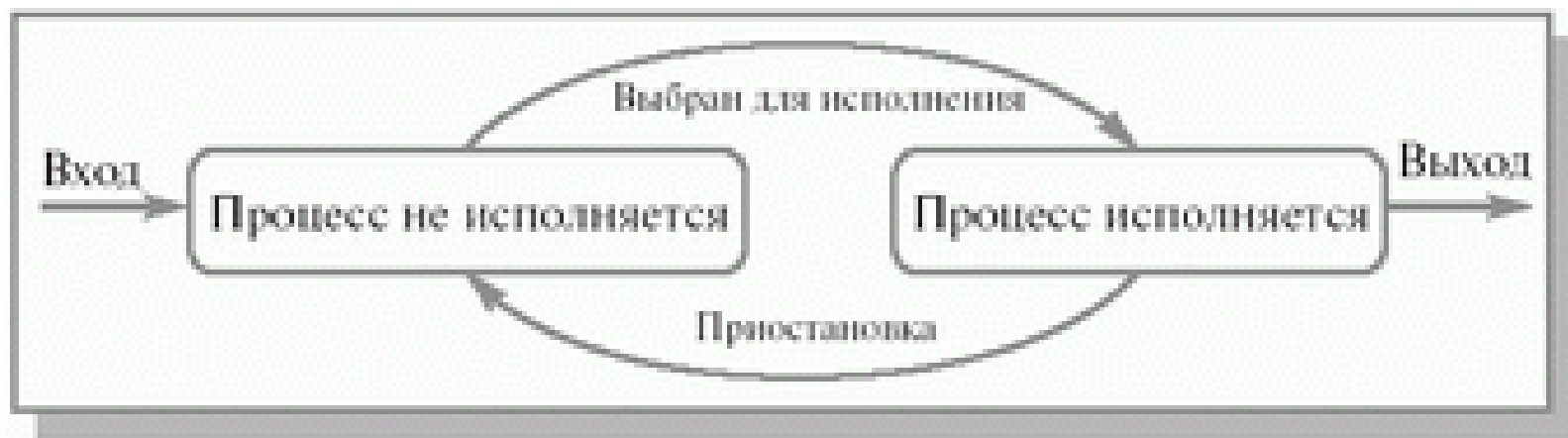
- Обычное завершение – процесс вызывает службу операционной системы, чтобы сообщить, что он завершил свою работу.
- Недостаточный объем памяти – для работы процесса требуется больше памяти, чем имеется в системе.
- Выход за пределы отведенной области памяти – процесс пытается получить доступ к ячейке памяти, к которой у него нет прав доступа.
- Ошибка защиты – процесс пытается использовать недоступный для него ресурс или файл, или пытается сделать это недопустимым образом, например, производит попытку записи в файл, открытый только для чтения.

- Арифметическая ошибка – процесс пытается выполнить запрещенную арифметическую операцию, например, деление на ноль, или пытается использовать число, превышающее возможности аппаратного обеспечения.
- Ошибка ввода-вывода – во время ввода или вывода происходит ошибка. Например, не удается найти нужный файл или производится попытка выполнить недопустимую операцию (например, чтение с печатающего устройства).
- Неверная команда – процесс пытается выполнить несуществующую команду.
- Команда с недоступными привилегиями – процесс пытается использовать команду, зарезервированную для операционной системы.

- Вмешательство оператора или операционной системы – по какой-либо причине операционная система может завершить процесс (например, в случае взаимоблокировки).
- Завершение родительского процесса – при завершении родительского процесса операционная система может автоматически прекращать все его дочерние процессы.
- Запрос со стороны родительского процесса – обычно родительский процесс имеет право прекращать любой из своих дочерних процессов.


3.2.4 Состояния процесса

ОС планирует потоки, принимая во внимание их состояние. Каждый процесс может находиться как минимум в двух состояниях: процесс выполняется и процесс не выполняется.



В многозадачной среде каждый процесс (поток) может находиться в 3-х состояниях.

- **ВЫПОЛНЕНИЕ** – *активное* состояние потока, во время которого поток обладает всеми необходимыми ресурсами и непосредственно выполняется процессором;
- **ОЖИДАНИЕ** – *пассивное* состояние потока, в котором, поток заблокирован по своим внутренним причинам (ждет завершения операции ввода-вывода, получения сообщения от другого потока и т. д.);
- **ГОТОВНОСТЬ** – *пассивное* состояние потока, но в этом случае он заблокирован в связи с внешним по отношению к нему обстоятельством (имеет все требуемые ресурсы, готов выполняться, однако процессор занят выполнением другого потока).



В течение своей жизни каждый поток переходит из одного состояния в другое в соответствии с алгоритмом планирования потоков, принятым в данной операционной системе.

Рассмотрим типичный граф состояния потока.



Рисунок 3.1 – Граф состояний потока в многозадачной среде

Кроме основных существует ещё несколько состояний – Инициализация (Init) или Рождение потока, Завершение (Terminate) или Закончил выполнение.




В состоянии выполнения в однопроцессорной системе может находиться не более одного потока, а в каждом из состояний ожидания и готовности – несколько потоков. Эти потоки образуют очереди соответственно ожидающих и готовых потоков. Очереди потоков организуются путем объединения в списки описателей отдельных потоков. Таким образом, каждый описатель потока, кроме всего прочего, содержит, по крайней мере, один указатель на другой описатель, соседствующий с ним в очереди. Такая организация очередей позволяет легко их переупорядочивать, включать и исключать потоки, переводить потоки из одного состояния в другое.

Процессы пользовательского режима ОС Windows

В пользовательском режиме работают следующие виды процессов:

□ системные процессы (system processes) – компоненты Windows, отвечающие за решение критически важных системных задач (т. е. аварийное завершение одного из этих процессов вызывает крах или нестабильную работу всей системы). Основные системные процессы:

■ Winlogon.exe – процесс входа в систему и выхода из неё;



■ `Smss.exe` (Session Manager – диспетчер сеансов) – процесс выполняет важные операции при инициализации системы (загрузка необходимых DLL, запуск процессов `Winlogon` и `Csrss` и др.), а затем контролирует работу `Winlogon` и `Csrss` (отвечает за подсистему Windows);

■ `Lsass.exe` (Local Security Authentication Subsystem Server – сервер подсистемы локальной аутентификации) – процесс проверяет правильность введенных имени пользователя и пароля;

- Wininit.exe – процесс инициализации системы (запускает процессы Lsass и Services);
- Userinit.exe – процесс инициализации пользовательской среды (например, запускает системную оболочку – по умолчанию, Explorer.exe);
- Services.exe (SCM, Service Control Manager – диспетчер управления службами) – процесс, отвечающий за выполнение служб;
- службы (сервисы, services) – приложения, работающие в фоновом режиме и не требующие взаимодействия с пользователем.

Службы могут быть как частью операционной системы (например, Windows Audio – служба для работы со звуком, или Print Spooler – диспетчер печати), так и частью пользовательского приложения (например, служба СУБД SQL Server);

□ пользовательские приложения (user applications) – прикладные программы, запускаемые пользователем;

□ подсистемы окружения (environment subsystems) – компоненты, предоставляющие доступ приложениям к некоторому подмножеству системных функций.