

## 26. ОС. Управление процессом. Планирование и диспетчеризация процессов. Потоки. Многопоточная модель ОС

*Процесс* (или по-другому, задача) - абстракция, описывающая выполняющуюся программу. Для операционной системы процесс представляет собой единицу работы, заявку на потребление системных ресурсов. Подсистема управления процессами планирует выполнение процессов, то есть распределяет процессорное время между несколькими одновременно существующими в системе процессами, а также занимается созданием и уничтожением процессов, обеспечивает процессы необходимыми системными ресурсами, поддерживает взаимодействие между процессами.

### Управление процессами:

Для того чтобы операционная система могла выполнять операции над процессами, каждый процесс представляется в ней некоторой структурой данных.

Эта структура содержит информацию, специфическую для данного процесса:

- состояние, в котором находится процесс;
- программный счетчик процесса или, другими словами, адрес команды, которая должна быть выполнена для него следующей;
- содержимое регистров процессора;
- данные, необходимые для планирования использования процессора и управления памятью (приоритет процесса, размер и расположение адресного пространства и т. д.);
- учетные данные (идентификационный номер процесса, какой пользователь инициировал его работу, общее время использования процессора данным процессом и т. д.);
- сведения об устройствах ввода-вывода, связанных с процессом (например, какие устройства закреплены за процессом, таблицу открытых файлов).

Для обеспечения **синхронизации** процессов и потоков, выполняющих общие задачи и совместно использующих общие ресурсы, в операционных системах используются специальные механизмы:

- Критические секции;
- Сигналы;
- Семафоры;
- События;
- Таймеры и др.

Планирование заключается в назначении приоритетов процессам в очереди с приоритетами. Программный код, выполняющий эту задачу, называется **планировщиком**.

**Планирование** может быть *динамическим* – решения принимаются системой на основе анализа текущей ситуации (характерен для универсальных ОС) или *статическим* – если потоки запускаются на основе заранее разработанного расписания (характерен для ОС реального времени).

### Типы планировщиков:

Операционные системы могут включать до трёх различных типов планировщиков:

*долговременный планировщик* (добавляет новый процесс в очередь готовых процессов (допускает к выполнению), или откладывает это действие; решает, какие процессы будут выполняться одновременно),

*среднесрочный планировщик* (временно перемещает (выгружает) процессы из основной памяти во вторичную (например, на жёсткий диск), и наоборот. Эти действия называются подкачкой)

*краткосрочный планировщик* (также известный как *диспетчер*, планировщик на этом уровне решает, какие из готовых и загруженных в память процессов будут запущены на ЦПУ после прерывания).

**Поток** — наименьшая единица обработки, исполнение которой может быть назначено ядром

операционной системы. Реализация потоков выполнения и процессов в разных операционных системах отличается друг от друга, но в большинстве случаев поток выполнения находится внутри процесса. Несколько потоков выполнения могут существовать в рамках одного и того же процесса и совместно использовать ресурсы, такие как память, тогда как процессы не разделяют этих ресурсов.

На одном процессоре многопоточность обычно происходит путём временного мультиплексирования (как и в случае многозадачности): процессор переключается между разными потоками выполнения. Это переключение контекста обычно происходит достаточно часто, чтобы пользователь воспринимал выполнение потоков или задач как одновременное. В многопроцессорных и многоядерных системах потоки или задачи могут реально выполняться одновременно, при этом каждый процессор или ядро обрабатывает отдельный поток или задачу.

**Многопоточность** — свойство платформы (например, операционной системы, виртуальной машины и т. д.) или приложения, состоящее в том, что процесс, порождённый в операционной системе, может состоять из нескольких потоков, выполняющихся «параллельно», то есть без предписанного порядка во времени. При выполнении некоторых задач такое разделение может достичь более эффективного использования ресурсов вычислительной машины.

Сутью многопоточности является квазимногозадачность на уровне одного исполняемого процесса, то есть все потоки выполняются в адресном пространстве процесса. Кроме этого, все потоки процесса имеют не только общее адресное пространство, но и общие дескрипторы файлов. Выполняющийся процесс имеет как минимум один (главный) поток.

Многопоточность (как доктрину программирования) не следует путать ни с многозадачностью, ни с многопроцессорностью, несмотря на то, что операционные системы, реализующие многозадачность, как правило, реализуют и многопоточность.

К достоинствам многопоточности в программировании можно отнести следующее:

- Упрощение программы в некоторых случаях за счет использования общего адресного пространства.
- Меньшие относительно процесса временные затраты на создание потока.
- Повышение производительности процесса за счет распараллеливания процессорных вычислений и операций ввода-вывода.