**Тема 4. Взаимодействие и планирование процессов**

**Лекция 15 «Вытесняющее и невытесняющее планирование. Алгоритмы планирования».**

Процесс планирования осуществляется частью операционной системы, называемой планировщиком. Планировщик может принимать решения о выборе для исполнения нового процесса, из числа находящихся в состоянии готовность в следующих случаях:

1. Когда процесс переводится из состояния исполнение в состояние завершение.
2. Когда процесс переводится из состояния исполнение в состояние ожидание.
3. Когда процесс переводится из состояния исполнение в состояние готовность (например, после прерывания от таймера).
4. Когда процесс переводится из состояния ожидание в состояние готовность (завершилась операция ввода-вывода или произошло другое событие).

В случаях **1** и **2** процесс, находившийся в состоянии исполнение, не может дальше исполняться, и для выполнения всегда необходимо выбрать новый процесс. В случаях **3** и **4** планирование может не проводиться, процесс, который исполнялся до прерывания, может продолжать свое выполнение после обработки прерывания. Если планирование осуществляется только в случаях **1** и **2**, говорят, что имеет место невытесняющее (nonpreemptive) планирование, в противном случае это вытесняющее (preemptive) планирование. Термин «вытесняющее планирование» возник потому, что исполняющийся процесс помимо своей воли может быть вытеснен из состояния исполнение другим процессом.

**Алгоритмы планирования**

1. **Первый пришел - первый обслуживается**. Процессор передается тому процессу, который раньше всех других его запросил. Среднее время ожидания для этой стратегии часто весьма длительное и зависит от порядка поступления процессов в очередь готовых процессов.
2. **Стратегия «наиболее короткая работа выполняется первой».** Стратегия, позволяющая короткому процессу из очереди выполняться первым. Снижается время ожидания очереди. Наибольшая трудность в практической реализации этой стратегии заключается в невозможности заранее определить величину времени последующего обслуживания.
3. **Приоритетное планирование.** Вышеперечисленные стратегии могут рассматриваться как частные случаи стратегии приоритетного планирования. Эта стратегия предполагает, что каждому процессу приписывается приоритет, определяющий очередность предоставления ему ЦП. Обычно приоритет – это целое положительное число, находящееся в некотором диапазоне, например, от 0 до 7 или от 0 до 1024. Будем считать, что чем меньше значение числа, тем выше приоритет процесса. Приоритеты назначаются, исходя из совокупности внутренних и внешних по отношению к операционной системе факторов.

Внутренние факторы: требования к памяти, количество открытых файлов, отношение среднего времени ввода-вывода к среднему времени использования ресурсов ЦП и так далее. Внешние факторы: важность процесса, тип и величина файлов, используемых для оплаты, отделение, выполняющее работы, и так далее. Внутренние факторы могут использоваться для автоматического назначения приоритетов самой ОС, а внешние для принудительного, с помощью оператора.

Главный недостаток приоритетного планирования заключается в возможности блокирования на неопределенно долгое время низкоприоритетных процессов.

1. **«Карусельная» стратегия планирования** – применяется в системах разделения времени. Процессы циклически перемещаются по очереди, получая ЦП на время, равное одному кванту. Новый процесс добавляется в хвост очереди. Если процесс не завершился в пределах выделенного ему кванта времени, его работа принудительно прерывается, и он перемещается в хвост очереди.
2. **Планирование с использованием многоуровневой очереди.** Эта стратегия разработана для ситуации, когда процессы могут быть легко классифицированы на группы, например, часто процессы разделяют на две группы: интерактивные (процессы переднего плана) и пакетные (фоновые). Стратегия многоуровневой очереди разделяет очередь готовых процессов на несколько очередей, в каждой из которых находятся процессы с одинаковыми свойствами. Взаимодействие очередей осуществляется по следующим правилам: ни один процесс с более низким приоритетом не может быть запущен, пока не выполнятся процессы во всех очередях с более высоким приоритетом. Работа процесса из очереди с более низким приоритетом может быть приостановлена, если в одной из очередей с более высоким приоритетом появился процесс.
3. **Использование многоуровневой очереди с обратными связями.** Обычная многоуровневая очередь не допускает перемещения процессов между очередями. Многоуровневая очередь с обратными связями предполагает, что процессы при определенных условиях могут перемещаться между очередями. Данная система наиболее быстро обслуживает все короткие по времени обслуживания запросы. Недостаток системы заключается в затратах времени на перемещение запросов из одной очереди в другую. Данная стратегия является универсальной и сочетает в себе свойства рассмотренных выше стратегий.
4. **Приоритетная многоочередная дисциплина обслуживания**. Вновь поступающие в систему запросы устанавливаются не обязательно в 1-ю очередь, а в очередь в соответствии с имеющимися приоритетами, которые определяются параметрами обслуживания процессов.