

# МОДЕЛЬ ОРГАНИЗАЦИИ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА С ДИНАМИЧЕСКИ ИЗМЕНЯЕМЫМИ ПРИОРИТЕТАМИ

## Шурхаленко П.Г.

*Шурхаленко Павел Геннадьевич – студент,  
кафедра электротехники и информационно-измерительных систем,  
Национальный исследовательский технологический университет «МИСиС», г. Москва*

Главной целью при планировании вычислительного процесса является распределение времени процессора между заданиями пользователей таким образом, чтобы удовлетворить все требования пользователя и системы. Основными характеристиками могут быть пропускная способность, время отклика, загрузка процессора.

В современных ОС используется несколько видов планирования, различаемые по временному масштабу: долгосрочное планирование, среднесрочное планирование, краткосрочное и планирование ввода-вывода. Диспетчер, он же краткосрочный планировщик (dispatcher), обычно работает, определяя, какой процесс или поток будет выполняться следующим.

На сегодняшний день, в большинстве операционных систем планирование осуществляется динамически, т.е. решения принимаются во время работы системы на основании проведенного анализа текущей ситуации, не задействовав сторонних предложений о мультипрограммной смеси.

Другой тип планирования – статический, используется только в специализированных системах заранее определенным набором задач. При таком типе планирования решение принимается заранее. Результатом является расписание – таблица, в которой указано, какому процессу, когда и на какое время должен быть предоставлен процессор.

Среднесрочное планирование является частью системы свопинга. Обычно решение о загрузке процесса в память принимается в зависимости от степени многозадачности (например, OS MFT, OS MVT). Кроме того, в системе с отсутствием виртуальной памяти среднесрочное планирование тесно связано с вопросами управления памятью [1].

Алгоритмы планирования потоков можно разделить на два класса: вытесняющие и не вытесняющие алгоритмы планирования [2].

- Не вытесняющие (non-preemptive) алгоритмы основаны на том, что активному потоку позволяет выполняться, пока он сам не отдаст управление операционной системе, для того чтобы она выбрала из очереди готовый к выполнению поток.

- Вытесняющие (preemptive) алгоритмы – это такие способы планирования потоков, в которых решение о переключении процессора с выполнения одного потока на выполнение другого потока принимается операционной системой, а не активной задачей.

В большинстве ОС (UNIX, Windows NT/2000/2003, OS/2, VAX/VMS и др.) реализованы вытесняющие алгоритмы планирования. В основе многих таких алгоритмов лежит концепция квантования. Т.е. каждому потоку поочередно предоставляется ограниченный период процессорного времени – квант [2].

Смена активного потока происходит, если:

- поток завершается и покинул систему;
- произошла ошибка;
- поток перешел в состояние ожидания;
- исчерпан квант времени, отведенный данному потоку.

В случае, если поток, исчерпав свой квант времени, не успел обработаться полностью, он переводится в состояние готовности и ожидает, когда ему будет предоставлен новый квант времени от процессора, а на выполнение в соответствии с определенным правилом выбирается новый поток из очереди готовых потоков.

Различают два больших класса дисциплин обслуживания: бесприоритетные и приоритетные. При бесприоритетном обслуживании выбор задач производится в заранее установленном. Приоритетные же делятся на [3]:

- приоритет, присвоенный задаче, является величиной постоянной;
- приоритет изменяется в течение времени решения задачи (динамический приоритет).

В большинстве ОС, поддерживающих потоки, приоритет потока связан с приоритетом процесса, в рамках которого выполняется поток. При создании процесса, ему назначается приоритет операционной системой и его значение внедряется в заголовки процесса, содержащий описание последнего и используется при назначении приоритета потоком этого процесса. Изменения приоритета могут происходить по инициативе самого потока, когда он обращается с соответствующим вызовом к ОС, или по инициативе пользователя, когда он выполняет соответствующую команду. Кроме этого, сама ОС может изменить приоритеты потоков в зависимости от ситуации, складывающейся в системе. В последнем случае приоритеты называются динамическими в отличие от неизменяемых, фиксированных

приоритетов. Возможности пользователей влиять на приоритеты процессов и потоков ограничены ОС. Обычно это разрешается администраторам, и то в определенных пределах. В большинстве случаев ОС присваивает приоритеты потокам по умолчанию.

Различают два вида приоритетного планирования по типу приоритета: относительные и абсолютные. И в том, и в другом случае на вход подается поток с наивысшим приоритетом. Однако определение момента смены активного потока решается по-разному. При первом типе планирования активный поток выполняется до тех пор, пока он сам не покинет процессор, перейдя в состояние ожидания (по вводу-выводу, например), или не завершится, или не произойдет ошибка.

Приоритеты из разных классов обрабатываются несколько по-разному. В классе приоритетов реального времени все потоки имеют фиксированный приоритет (от 16 до 31), который никогда не изменяется, и все активные потоки с определенным уровнем приоритета располагаются в круговой очереди данного класса [1].

В классе переменных приоритетов поток начинает работу с базового приоритета процесса, который может принимать значение от 1 до 15. Каждый поток, связанный с процессом имеет, свой базовый приоритет, равный базовому приоритету процесса, или отличающийся от него не более чем на 2 уровня в большую или меньшую сторону. После активации потока его динамический приоритет может колебаться в определенных пределах – он не может упасть ниже наименьшего базового приоритета данного класса, т.е. 15 (для потоков с приоритетом 16 и выше никогда не делается никаких изменений приоритетов) [1].

### ***Список литературы***

1. НОУ ИНТУИТ Современные операционные системы. [Электронный ресурс]. Режим доступа: <http://www.intuit.ru/studies/courses/631/487/info/> (дата обращения: 15.05.2017).
2. *Таненбаум Э.* Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
3. *Олифер В.Г., Олифер Н.А.* Сетевые операционные системы. СПб.: Питер, 2007. 544 с.