

УДК 621.3

З.Б. Холодная, М.А. Трусов

*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков*

## **ИССЛЕДОВАНИЕ АЛГОРИТМОВ УПРАВЛЕНИЯ ПРИОРИТЕТНОЙ МНОГОЗАДАЧНОСТЬЮ ОПЕРАЦИОННОЙ СИСТЕМЫ WINDOWS XP**

*Исследуются проблемы управления приоритетной многозадачностью ОС Windows XP и предлагается механизм контроля потоков с приоритетом реального времени.*

*операционная система, многозадачность, приоритетность, реальное время*

### **Введение**

**Анализ проблемы.** Из года в год возрастают темпы развития компьютерных технологий – всё интенсивнее происходит интеграция отдельных компьютеров в локальные сети и рост сети Internet, сложность решаемых задач неуклонно возрастает, проблемы надёжности, быстродействия, функциональности выходят на первый план. Наряду с развитием инструментальных средств, непрерывно совершенствуются и операционные системы. На смену однозадачным операционным системам пришли многозадачные многопроцессорные системы. Трудно представить объём задач, решаемых системой, многообразие и сложность процессов, протекающих в ней.

Современная операционная система – это сложный программный комплекс, под управлением которого работают десятки, а иногда, и сотни процессов. Вполне понятно, что их совместная работа невозможна без применения гибких и эффективных сценариев планирования.

Алгоритмы планирования являются важной составляющей любой многозадачной системы. Основная задача, которую они решают – это распределение процессорного времени между процессами. Эффективность работы алгоритмов планирования во многом определяет эффективность работы самой операционной системы. Проанализируем алгоритмы управления приоритетной многозадачностью на примере операционной системы Windows XP как наиболее приемлемой альтернативой в виду ее распространённости и доступности.

**Обзор механизмов планирования.** Планирование в операционной системе Windows XP происходит на уровне потоков, а сами процессы не выполняются, а лишь предоставляют ресурсы и контекст для выполнения потоков. Система не обращает внимания на то, какому процессу принадлежит тот или иной поток, и не производит разделение на системные и пользовательские потоки. В Windows реализована подсистема вытесняющего планирования на основе

уровней приоритета, в которой всегда выполняется поток с наибольшим приоритетом, готовый к выполнению. Код Windows, отвечающий за планирование, реализован в ядре. Поскольку этот код рассредоточен по ядру, единого модуля или процедуры с названием «планировщик» нет. Совокупность процедур, выполняющих эти обязанности, называется диспетчером ядра. Для более детального понимания алгоритмов управления необходимо рассмотреть допустимые состояния потоков и уровни приоритетов, используемые в операционной системе Windows XP.

Поток может находиться в одном из 7 состояний, соответствующая схема представлена на рис. 1.

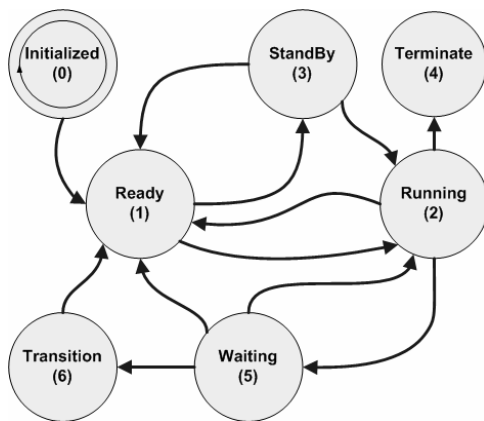


Рис. 1. Состояния потоков в Windows XP

Краткое описание состояний потока:

**Ready (Готов)** – поток в состоянии готовности ожидает выполнения. Выбирая следующий поток для выполнения, диспетчер принимает во внимание только пул потоков, готовых к выполнению;

**StandBy (Простаивает)** – поток в этом состоянии уже выбран следующим для выполнения на конкретном процессоре. В подходящий момент диспетчер переключает контекст на этот поток;

**Running (Выполняется)** – поток переходит в это состояние и начинает выполняться сразу после того, как диспетчер переключает на него контекст;

**Waiting (Ожидает)** – по окончании ожидания поток – в зависимости от приоритета – либо немедленно начинает выполняться, либо переходит в состояние Ready;

**Transition (Переходное состояние)** – поток переходит в это состояние, если он готов к выполнению, но его стек ядра выгружен из памяти. Как только этот стек загружается в память, поток переходит в состояние Ready;

**Terminated (Завершён)** – заканчивая выполнение, поток переходит в это состояние;

**Initialized (Инициализирован)** – в это состояние поток входит в процессе своего создания.

Как показано на рис. 2, в Windows предусмотрено 32 уровня приоритета – от 0 до 31. Эти значения группируются следующим образом:

- 16–31 – уровни реального времени;
- 1–15 – динамические уровни;
- 0 – системный уровень, зарезервированный для потока обнуления страниц.

У процесса только одно значение приоритета – базовое, у потока два значения: базовое и текущее. Базовый приоритет потока устанавливается, исходя из приоритета процесса, которому данный поток принадлежит, текущий приоритет изменяется в зависимости от состояния потока. Решения, связанные с планированием, принимаются на основе текущего приоритета. Нужно отметить, что уровни приоритета назначаются с учётом двух разных точек зрения – Windows API и ядра Windows.

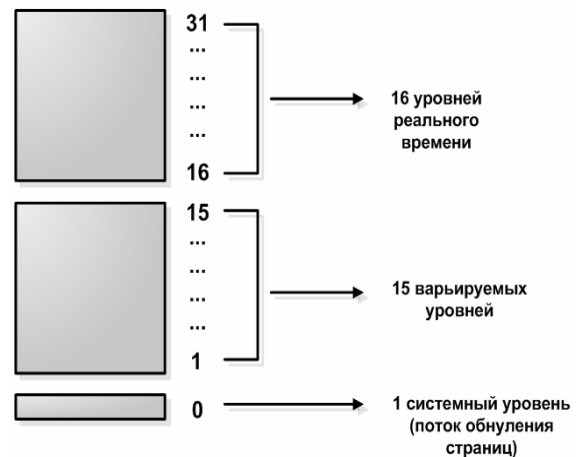


Рис. 2. Уровни приоритета потока

Windows API сначала упорядочивает процессы по классам приоритета, назначенным при их создании [Real-time, High, Above Normal, Normal, Below Normal, Idle], а затем – по относительному приоритету индивидуальных потоков в рамках этих процессов [Time-Critical, Highest, Above-normal, Normal, Below-normal, Lowest, Idle].

В определённой ситуации Windows может на короткое время повышать приоритеты потоков в динамическом диапазоне (1–15). Система никогда не изменяет приоритеты потоков в диапазоне реального времени (16–31), поэтому у таких потоков базовый приоритет всегда идентичен текущему. Более того, планирование потоков реального времени в ОС Windows происходит под лозунгом: «Тот, кто использует потоки реального времени, знает, что делает», т.е. нет никакого динамического регулирования со стороны системы.

Windows может динамически повышать значение текущего приоритета потока в следующих случаях:

**После завершения операций ввода-вывода.**

Windows временно повышает приоритет потоков по окончании операций ввода-вывода, поэтому у потоков, ожидавших завершения таких операций, больше шансов возобновить выполнение и обработать полученные данные.

**По окончании ожидания на событии или семафоре исполнительной системы.**

Когда ожидание потока на событии исполнительной системы или объекте «семафор» успешно завершается, его приоритет повышается на 1 уровень.

**По окончании операции ожидания потоками активного процесса.**

Происходит повышение значения текущего приоритета на заданную системой величину. В отличие от других видов динамического повышения приоритета этот поддерживается всеми системами Windows и его нельзя отключить.

**При пробуждении GUI-потоков из-за операций с окнами.**

Приоритет потоков, владеющих окнами, дополнительно повышается на 2 уровня после их пробуждения из-за активности подсистемы управления окнами. Повышение происходит для создания преимуществ интерактивным приложениям.

**Если поток, готовый к выполнению, задерживается из-за нехватки процессорного времени.**

Раз в секунду диспетчер настройки баланса, системный поток, предназначенный главным образом для выполнения функций управления памятью, сканирует очереди готовых потоков и ищет потоки, которые находятся в состоянии Ready в течение примерно 4 секунд. Обнаружив такой поток, диспетчер настройки баланса повышает его приоритет до 15, квант потока удваивается относительно кванта процесса. Как только квант потока истекает, приоритет потока немедленно снижается до исходного уровня.

Отметим, что независимо от приращения приоритета потока никогда не будет больше 15.

**Проблемы планирования в операционной системе Windows XP.** Проанализировав алгоритмы управления приоритетной многозадачностью, можно сказать, что в модели приоритетного планирования потоков в операционной системе Windows XP, несмотря на её гибкость и функциональность, существует ряд слабых мест, в частности:

- отсутствие контроля над работой процессов реального времени;
- возможность блокирования пользовательскими потоками важных системных потоков;
- отсутствие встроенных инструментов по выявлению и ликвидации сбойных процессов, которые в результате неправильной работы блокируют работу системы.

Как следствие возникает потребность в разработке программного продукта, который сможет производить контроль над работой потоков реального времени, и тем самым, повысить отказоустойчивость системы.

**Разработка механизма контроля потоков с приоритетом реального времени.** Разработанный программный продукт ориентирован на работу с потоками, работающими в диапазоне реального времени, в связи с этим поток, в котором происхо-

дит работа – получение и анализ данных, должен и сам обладать приоритетом реального времени. Задачи разрабатываемой программы – выявление и ликвидация «нечестных» сценариев планирования. Рассмотрим алгоритм работы данного продукта.

В процессе сканирования системы производится получение информации о процессах и потоках в системе и анализ полученных данных. Если поток удовлетворяет установленным условиям, то он заносится в список подозрительных потоков. Подозрительным считается поток, который удовлетворяет следующим требованиям: обладает приоритетом реального времени; находится в состоянии Ready.

При сканировании учитывается состояние всех потоков в системе и на основе этих данных устанавливается степень загруженности системы. Загруженность характеризуется количеством потоков находящихся в состоянии Ready более 3 секунд и уровнем приоритета у этих процессов. Если степень загруженности превышает установленный предел, то система включает механизм блокировки. Рассмотрим принцип работы данного механизма.

Первым делом система обращается к списку подозрительных потоков, понижает уровень их приоритета до базового, затем вычисляется родительский процесс, которому принадлежит подозрительный поток, после этого производится запрос пользователя на дальнейшие действия: убить процесс, понизить его приоритет или игнорировать. В зависимости от выбора пользователя система запускает ту или иную команду. В случае, если пользователь игнорирует запрос, система возвращает потокам их уровень приоритета.

## Выводы

В данной статье были рассмотрены механизмы планирования, используемые в операционной системе Windows XP. В результате анализа алгоритмов управления приоритетной многозадачностью, выявлены недостатки этой методики. В качестве решения проблемы был разработан механизм контроля, позволяющий определить и корректно обработать ошибочную ситуацию.

## Список литературы

1. Руссинович М., Соломон Д. *Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP и Windows 2000.* – СПб.: Русская Редакция, 2006. – 968 с.
2. Интернет-проект «Russian Software Developer Network». [Электронный ресурс]. – Режим доступа: <http://rsdn.ru>.

Поступила в редколлегию 11.12.2007

**Рецензент:** д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.