

А.В. Макаров, С.Ю. Скоробогатов, А.М. Чеповский

Учебный курс “СIL и системное программирование в Microsoft .NET”



Лекция 19. Многозадачность в Windows

Введение

- Общие понятия и основные подходы к реализации многозадачности
 - Мультипроцессирование
 - Мультипрограммирование
- Основные понятия
 - Процесс
 - Поток
 - Нить
 - Контекст
 - Дескриптор

19.1.1 Мультипроцессирование

■ **Различные устройства**

Использование разных устройств выполняющих специализированные действия;
например: процессор, сопроцессор, контроллер прямого доступа к памяти, видеосистема с графическим ускорителем, PostScript принтер и т.д.

■ **Одинаковые устройства**

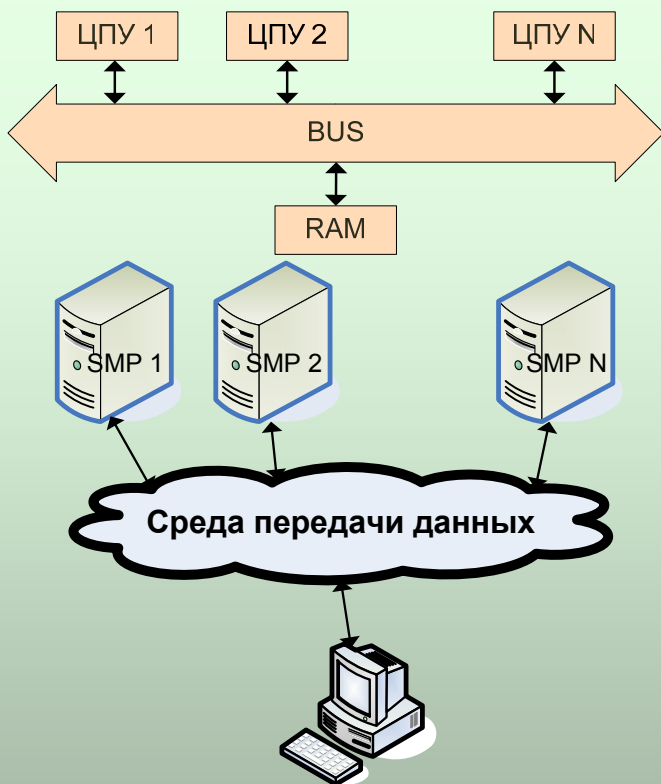
■ **Симметричное мультипроцессирование**

Использование одинаковых равноправных устройств;
например: обычные двух- или четырех- процессорные компьютеры.

■ **Асимметричное мультипроцессирование**

Использование одинаковых неравноправных устройств с выделением ведущих и ведомых.

Многопроцессорные вычислительные системы



- **SMP системы с общей памятью**

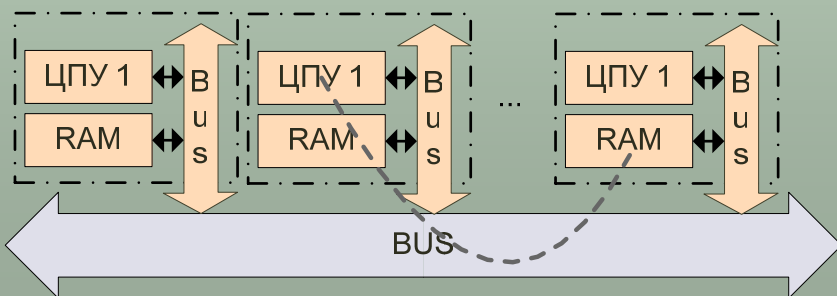
Несколько равноправных процессоров имеют конкурентный доступ к общей памяти

- **MPP массивно-параллельные системы**

Вычислительные узлы соединены в одну систему с помощью среды передачи данных

- **NUMA и cc-NUMA системы**

Вся память системы доступна всем процессорам, но доступ к памяти своего узла осуществляется процессором существенно быстрее.



19.1.2. Мультипрограммирование

Вытесняющая многозадачность

Требует больших расходов на реализацию, однако обеспечивает лучшую изоляцию приложений друг от друга.

Планирование

Статическое

Динамическое

С использованием
приоритетов

С относительными
приоритетами

*Невытесняющая
многозадачность*

С использованием квантов
(постоянных или
динамических)

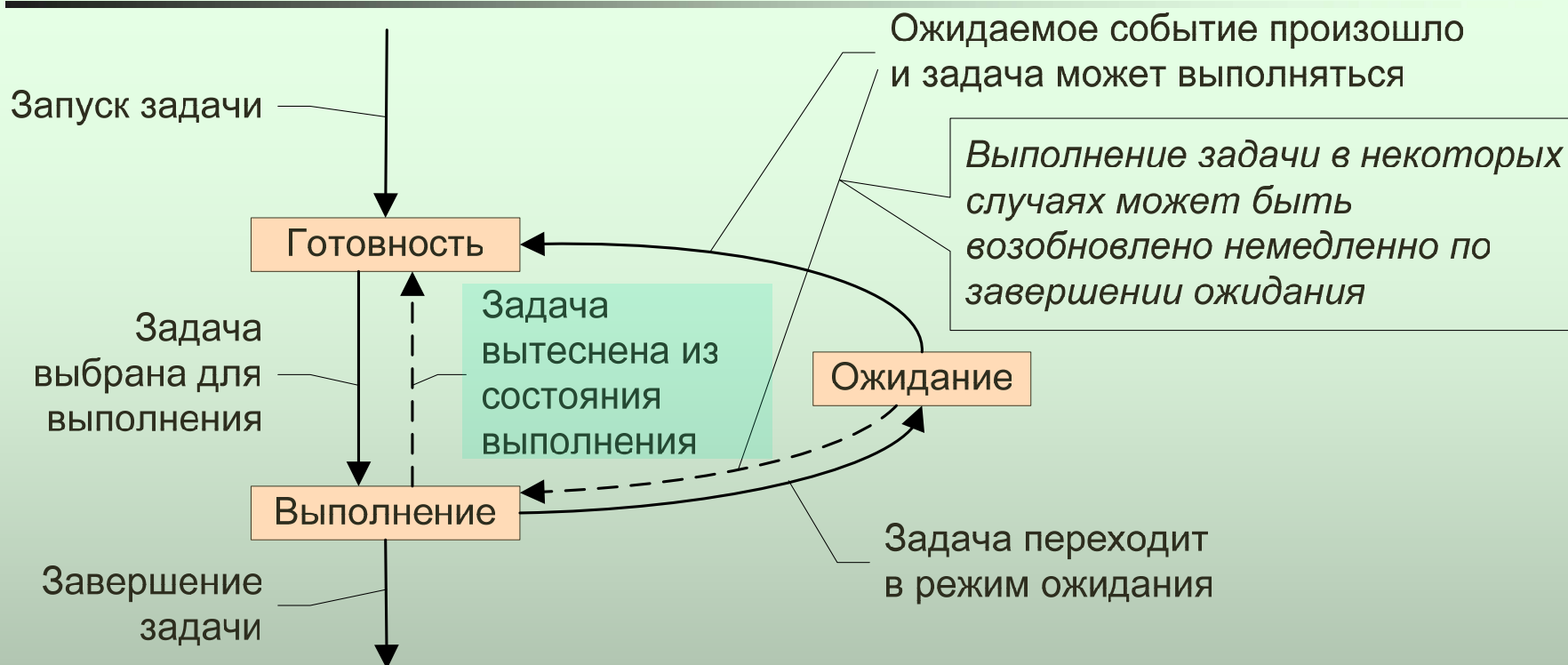
С абсолютными
приоритетами

*Вытесняющая
многозадачность*

Невытесняющая многозадачность

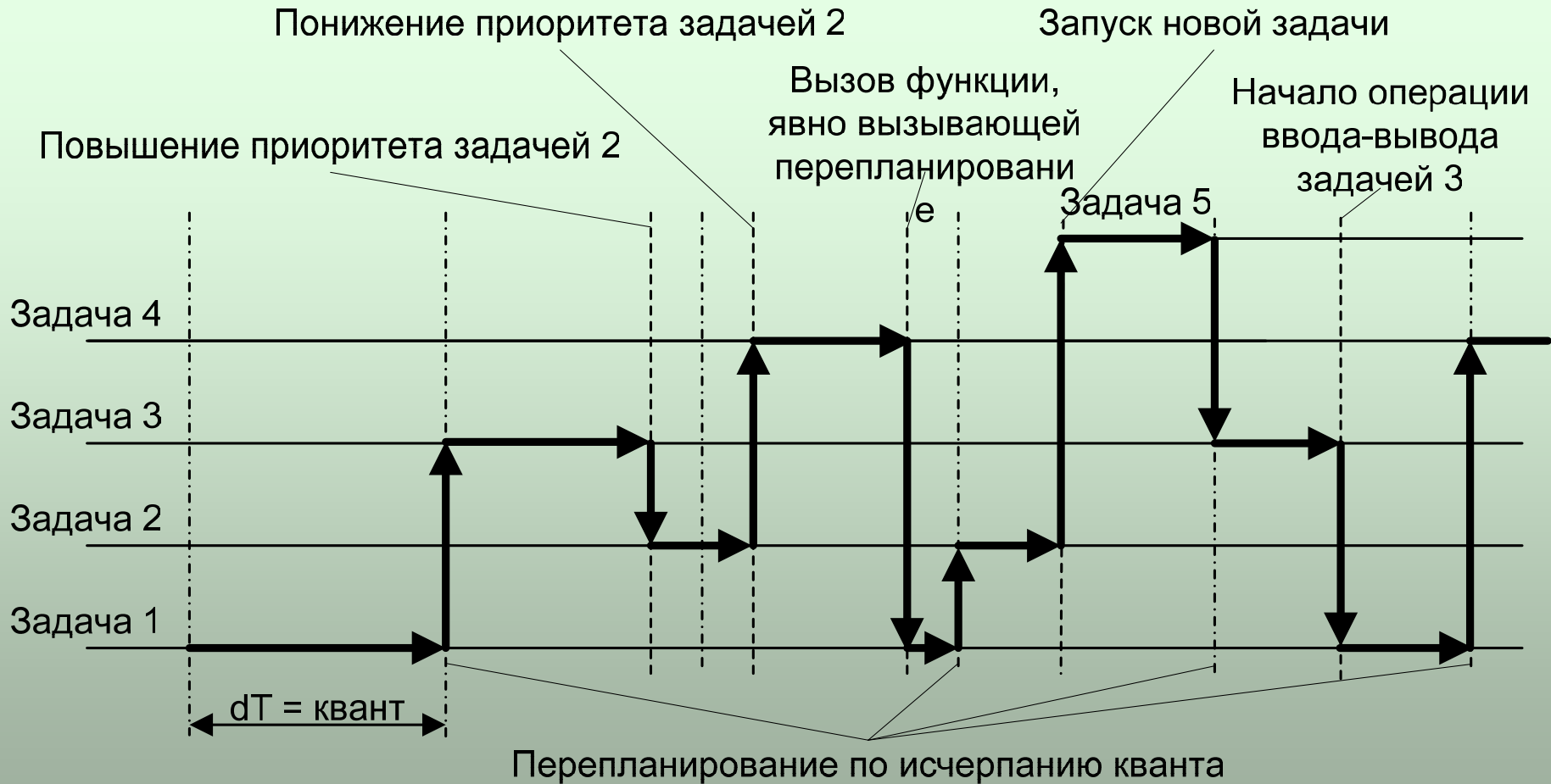
Проста и экономна в реализации, но уязвима и требует тщательной разработки приложений.

Граф состояний задачи



- Граф состояний задачи в случае невытесняющей многозадачности отличается отсутствием перехода из состояния «Выполнение» в состояние «Готовность» (нарисован пунктиром).
- Второй нарисованный пунктиром переход соответствует возможности немедленного возобновления выполнения задачи по наступлении ожидаемого события – это существенно уменьшает время отклика.

Моменты перепланирования



19.1.3. Основные понятия

- **Process** (процесс)
является объектом планирования адресного пространства.
- **Thread** (поток, нить, легковесный процесс)
является объектом планирования процессорного времени;
планирование осуществляется ядром системы.
- **Fiber** (нить, волокно, поток пользователя)
является объектом планирования процессорного времени;
планирование осуществляется приложением.
- **Descriptor** (дескриптор)
содержит информацию, описывающую поток, но не его текущее состояние исполнения (приоритет, права доступа, переменные окружения и т.д.).
- **Context** (контекст)
содержит информацию, описывающую непосредственно состояние исполнения потока (состояние стека, регистров и т.д.).

19.2. Граф состояний потока в Windows

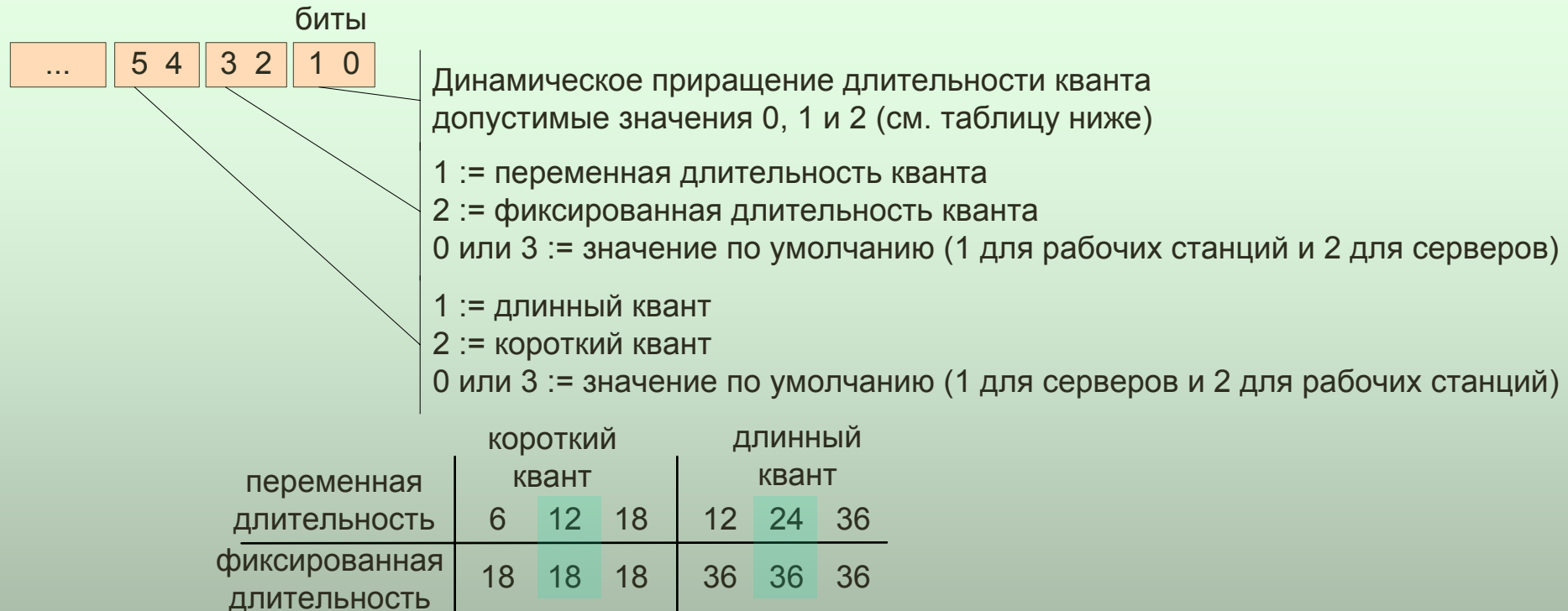
Перепланирование в Windows наступает в случаях:

- Создания и завершения потока
- Исчерпания выделенного потоку кванта
- Выхода потока из состояния ожидания
- Переход потока в состояние ожидания
- Изменения приоритета потока
- Изменения привязки к процессору



19.2.1. Управление квантованием в Windows

HKLM\SYSTEM\CurrentControlSet\Control\PriorityControl\Win32PrioritySeparator



- Типичные значения параметра Win32PrioritySeparation:
0x26 для станций (10 01 10 короткий квант переменной длины, 18 ед.)
0x18 для серверов (01 10 00 длинный квант фиксированной длины, 36 ед.)
- 1 тик таймера равен 3 ед.; период тиков – $10 \div 15$ мс. на разной аппаратуре.

19.2.2. Управление приоритетами в Windows

Влияние класса процесса и относительного приоритета потока на итоговое значение уровня приоритета

	HIGH	ABOVE_NORMAL	NORMAL			BELOW_NORMAL	IDLE
			Foreground	Normal	Background		
15	TIME_CRITICAL	TIME_CRITICAL	TIME_CRITICAL	TIME_CRITICAL	TIME_CRITICAL	TIME_CRITICAL	TIME_CRITICAL
14	ABOVE_NORMAL						
13	NORMAL						
12	BELOW_NORMAL	HIGHEST					
11	LOWEST	ABOVE_NORMAL	HIGHEST				
10		NORMAL	ABOVE_NORMAL	HIGHEST			
9		BELOW_NORMAL	NORMAL	ABOVE_NORMAL	HIGHEST		
8		LOWEST	BELOW_NORMAL	NORMAL	ABOVE_NORMAL	HIGHEST	
7			LOWEST	BELOW_NORMAL	NORMAL	ABOVE_NORMAL	
6				LOWEST	BELOW_NORMAL	NORMAL	HIGHEST
5					LOWEST	BELOW_NORMAL	ABOVE_NORMAL
4						LOWEST	NORMAL
3							BELOW_NORMAL
2							LOWEST
1	IDLE	IDLE	IDLE	IDLE	IDLE	IDLE	IDLE
0	Поток обнуления страниц						

19.2.2. Управление приоритетами в Windows

Влияние класса процесса и относительного приоритета потока на итоговое значение уровня приоритета

REALTIME		<i>NORMAL</i>	<i>NORMAL</i>			<i>BELOW_NORMAL</i>	<i>IDLE</i>
			Foreground	Normal	Background		
31	TIME_CRITICAL	CRITICAL	TIME_CRITICAL	TIME_CRITICAL	TIME_CRITICAL	TIME_CRITICAL	TIME_CRITICAL
30	6						
29	5						
28	4	HIGHEST					
27	3	BELOW_NORMAL	HIGHEST				
26	HIGHEST	NORMAL	ABOVE_NORMAL	HIGHEST			
25	ABOVE_NORMAL	BELOW_NORMAL	NORMAL	ABOVE_NORMAL	HIGHEST		
24	NORMAL	LOWEST	BELOW_NORMAL	NORMAL	ABOVE_NORMAL	HIGHEST	
23	BELOW_NORMAL		LOWEST	BELOW_NORMAL	NORMAL	ABOVE_NORMAL	
22	LOWEST			LOWEST	BELOW_NORMAL	NORMAL	HIGHEST
21	-3				LOWEST	BELOW_NORMAL	ABOVE_NORMAL
20	-4					LOWEST	NORMAL
19	-5						BELOW_NORMAL
18	-6						LOWEST
17	-7	IDLE	IDLE	IDLE	IDLE	IDLE	IDLE
16	IDLE	Поток обнуления страниц					

19.2.3. Выводы

- Windows не является жесткой системой реального времени.
- Windows предоставляет два механизма для реализации мультипрограммирования – вытесняющая многозадачность для потоков (потоки ядра) и невытесняющая для волокон (потоки пользователя).
- Для потоков Windows реализует вытесняющую многозадачность с динамическими абсолютными приоритетами и динамическим квантованием.
- Администратор Windows системы может в некоторой степени влиять на назначение динамических приоритетов и квантование времени.
- Windows содержит встроенную поддержку многопроцессорных SMP и NUMA систем.