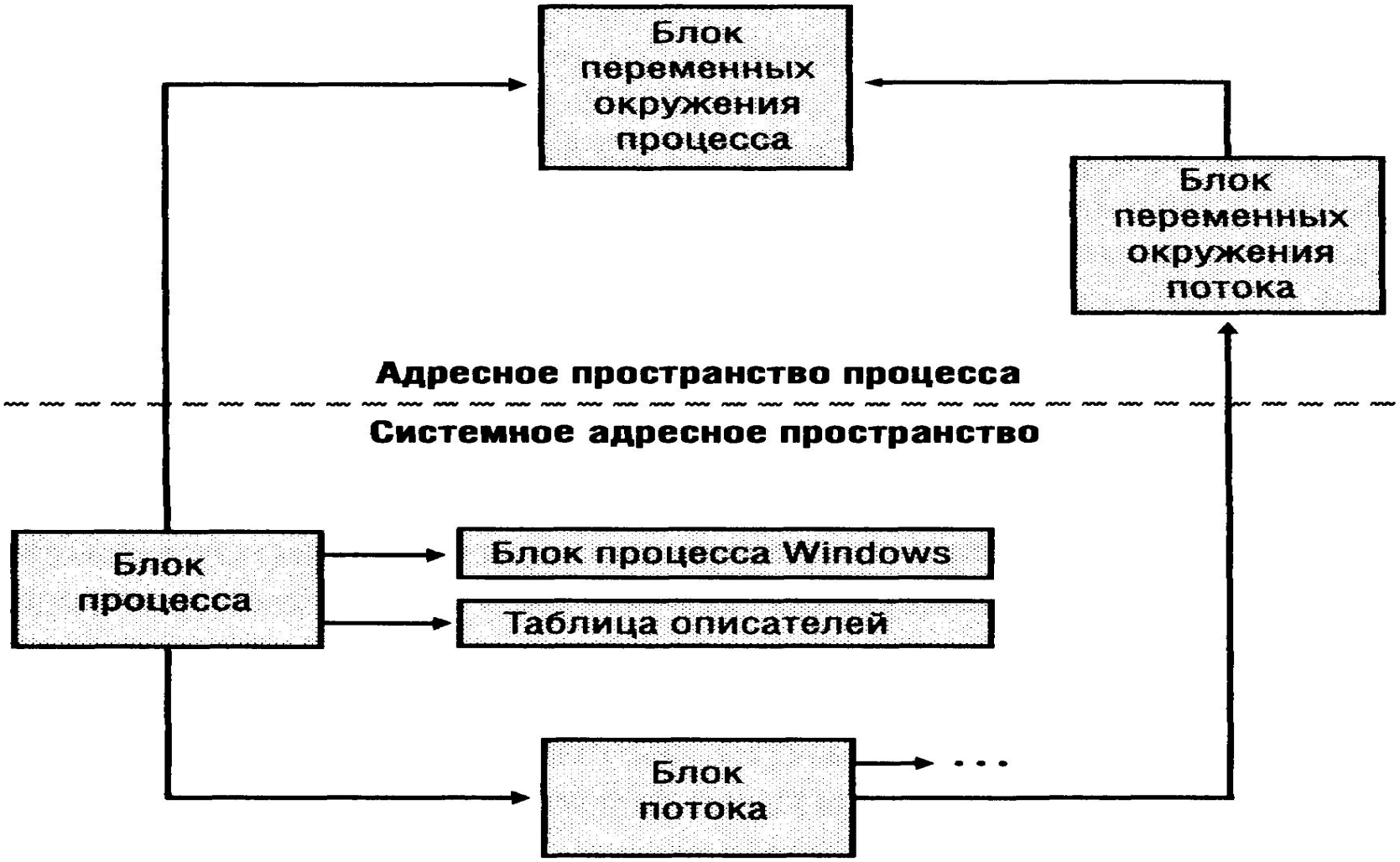
Тема: Процессы и потоки в ОС Windows NT

1. Внутреннее устройство процессов.
2. Внутреннее устройство потоков.
3. Планирование потоков.

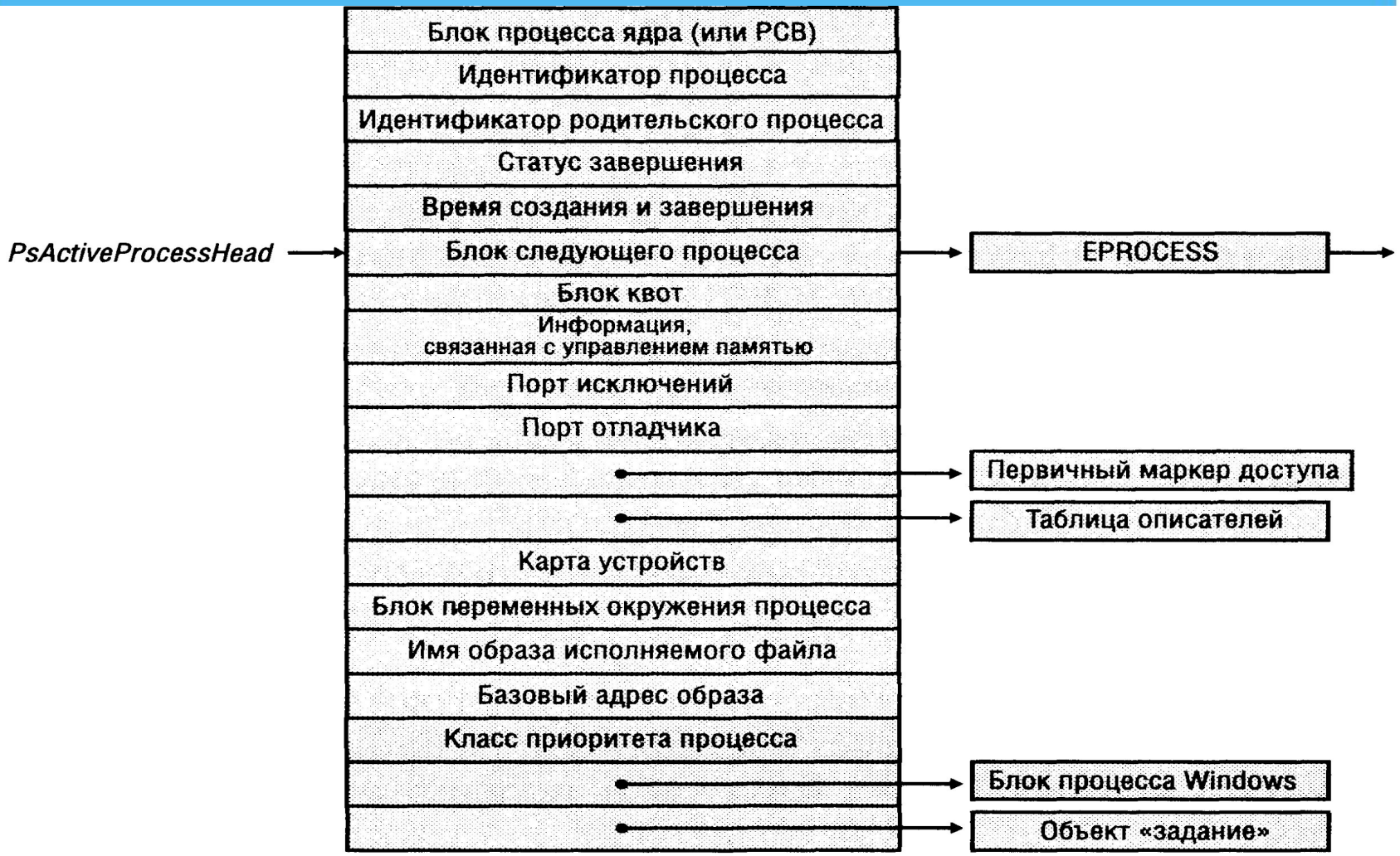
Литература

1. М. Русинович, Д. Соломин Внутреннее устройство Windows: Windows Server 2003, Windows XP, Windows 2000. Мастер-класс. / Пер. с анг. – 4-е изд. – М.: Издательско-торговый дом «Русская редакция»; СПб.: Питер, 2005. – 992 с.
2. Э. Таненбаум Современные операционные системы. 3-е изд.; СПб.: Питер, 2010. – 1120 с.

Структура данных процессов и потоков

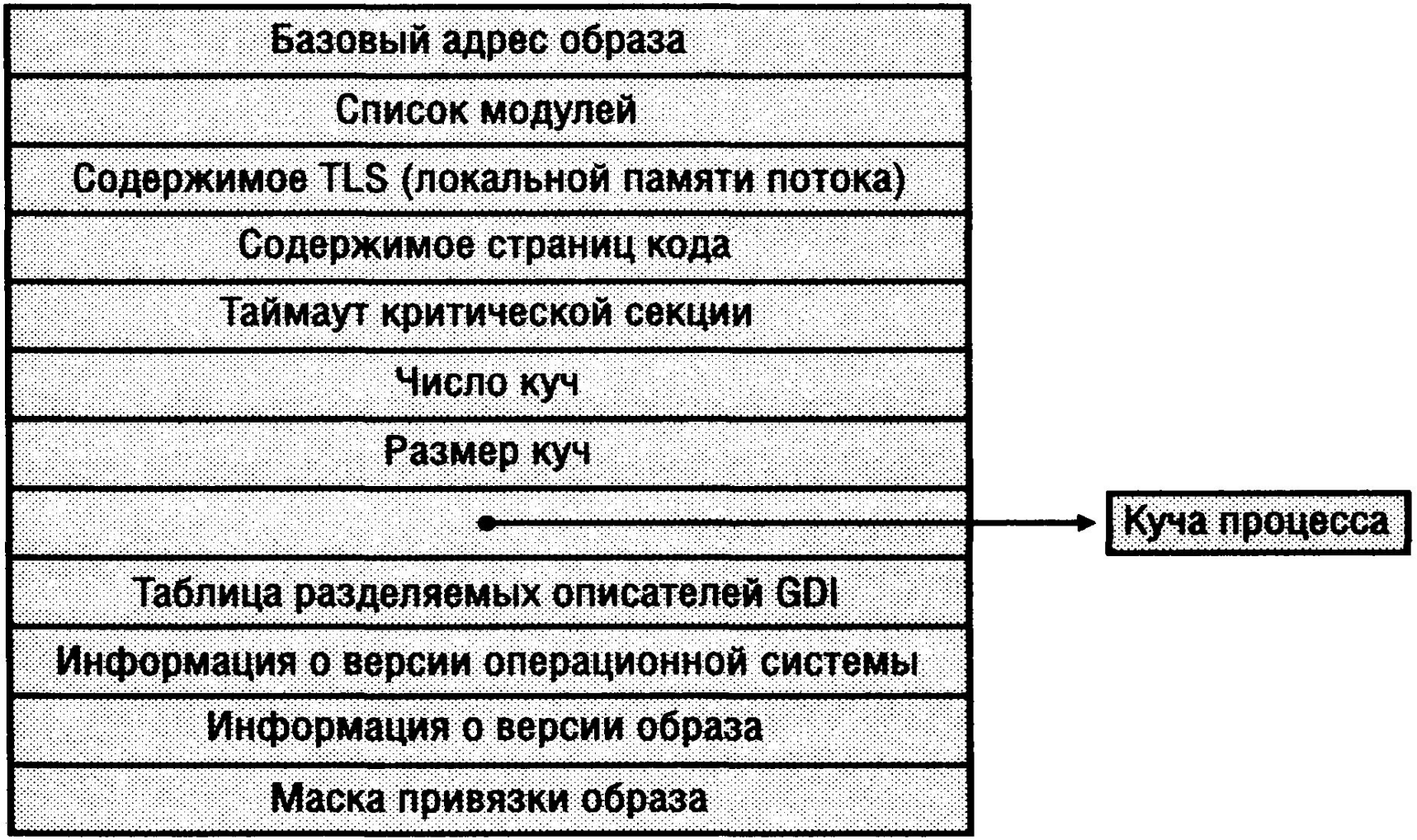


Блок процесса EPROCESS



Блок процесса исполнительной системы

Поля в блоке PEB



Переменные ядра, связанные с производительностью

Переменная	Тип	Описание
PsActiveProcessHead	Заголовок очереди	Заголовок списка блоков процесса
PsIdleProcess	EPROCESS	Блок процесса Idle
PsInitialSystemProcess	Указатель на EPROCESS	Указатель на блок начального системного процесса (с идентификатором 2), который содержит системные потоки
PspCreateProcessNotifyRoutine	Массив указателей	Указатели на процедуры (максимум 8), вызываемые при создании и удалении процесса
PspCreateProcessNotifyRoutineCount	DWORD	Счетчик зарегистрированных процедур уведомления о создании процесса
PspLoadImageNotifyRoutine	Массив указателей	Указатели на процедуры, вызываемые при загрузке образа исполняемого файла
PspLoadImageNotifyRoutineCount	DWORD	Счетчик зарегистрированных процедур уведомления о загрузке образа
PspCidTable	Указатель на HANDLE_TABLE	Таблица описателей для клиентских идентификаторов процесса и потока

Счетчики производительности, связанные с процессами

Объект: счетчик	Описание
Process: % Privileged Time (Процесс: % работы в привилегированном режиме)	Процентная доля времени, в течение которого потоки данного процесса выполнялись в режиме ядра
Process: % Processor Time (Процесс: % загруженности процессора)	Процентная доля процессорного времени, использованная потоками процесса за определенный период времени; вычисляется как сумма % Privileged Time и % User Time
Process: % User Time (Процесс: % работы в пользовательском режиме)	Процентная доля времени, в течение которого потоки данного процесса выполнялись в пользовательском режиме
Process: Elapsed Time (Процесс: Прошло времени)	Суммарное время (в секундах), прошедшее с момента создания процесса
Process: ID Process (Процесс: Идентификатор процесса)	Идентификатор процесса; полученное таким образом значение действительно лишь на время выполнения процесса, поскольку идентификаторы могут использоваться повторно
Process: Creating Process ID [Процесс: Код (ID) создавшего процесса]	Идентификатор родительского процесса; его значение не обновляется после завершения родительского процесса
Process: Thread Count (Процесс: Счетчик потоков)	Число потоков в процессе
Process: Handle Count (Процесс: Счетчик дескрипторов)	Число открытых процессом описателей

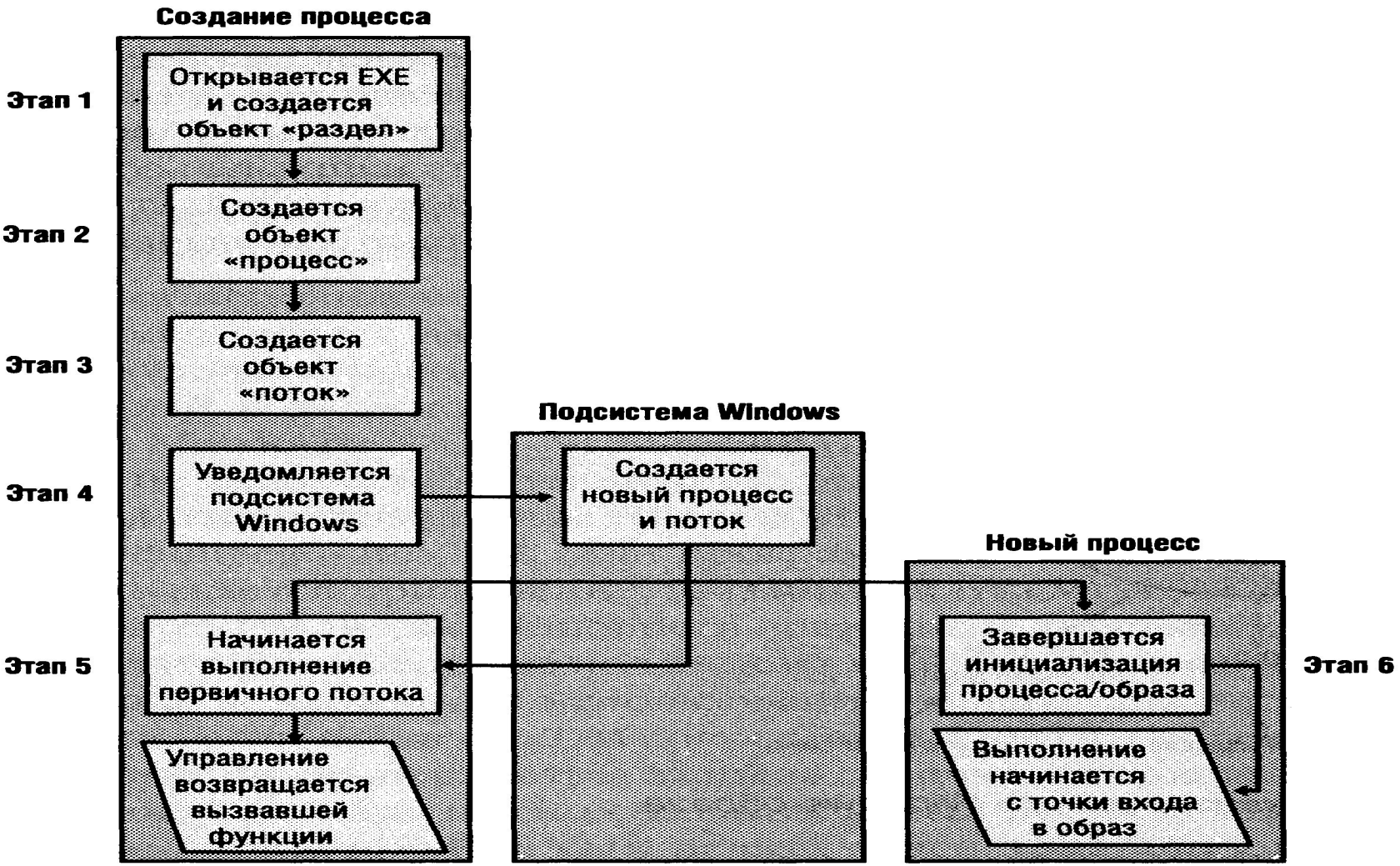
Функции, связанные с процессами

Функция	Описание
CreateProcess	Создает новый процесс и поток с использованием идентификации защиты вызывающего процесса
CreateProcessAsUser	Создает новый процесс и поток с указанным альтер-
CreateProcessWithLogonW	Создает новый процесс и поток для выполнения под учетной записью, соответствующей указанным имени и паролю пользователя
CreateProcessWithTokenW	Создает новый процесс и поток с указанным альтернативным маркером защиты и поддерживает дополнительные возможности, например разрешает загрузку профиля пользователя
OpenProcess	Возвращает описатель указанного объекта «процесс»
ExitProcess	Завершает процесс с уведомлением всех подключенных DLL

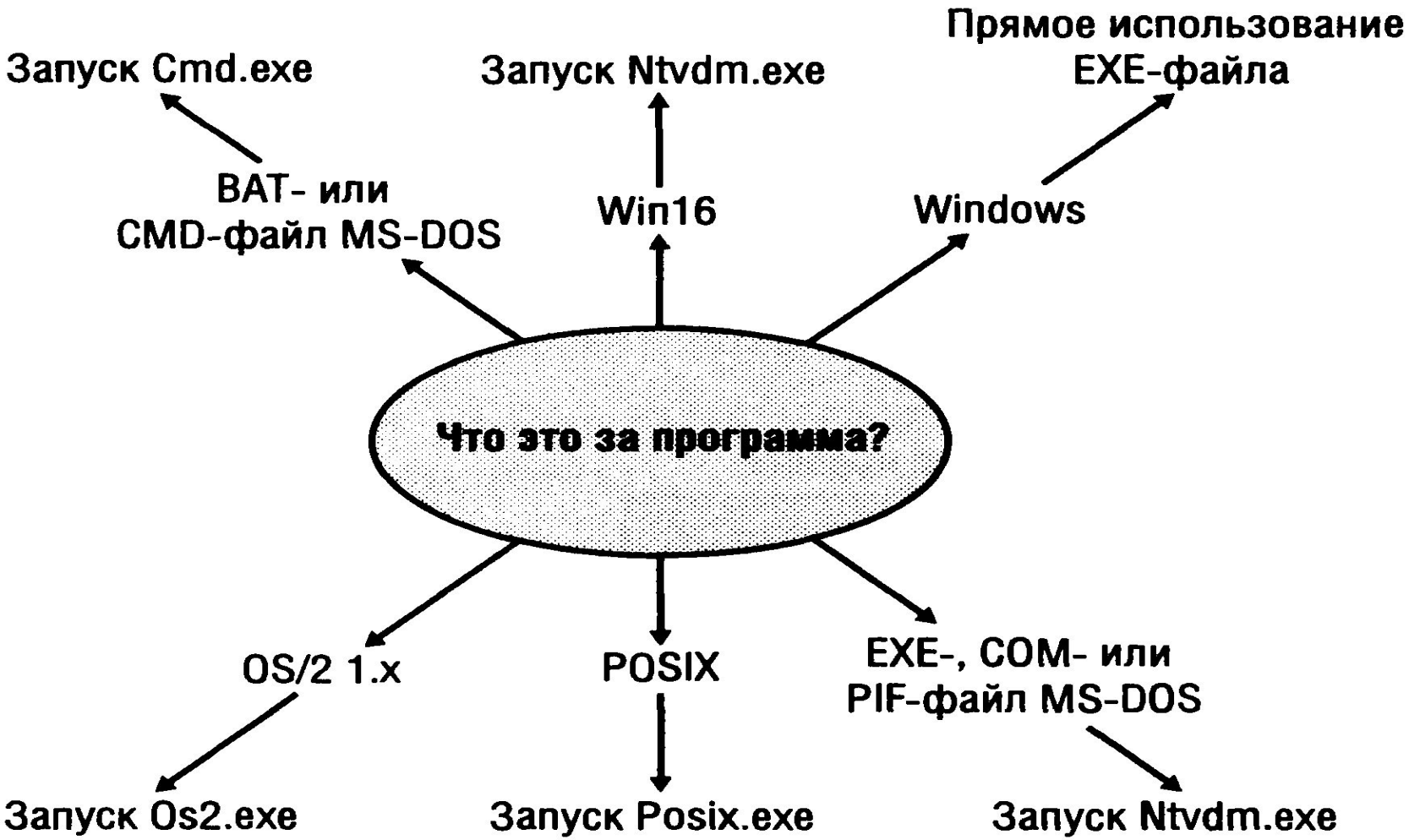
Функции, связанные с процессами

GetExitCodeProcess	Возвращает код завершения процесса, указывающий, как и почему завершился этот процесс
GetCommandLine	Возвращает указатель на командную строку, переданную текущему процессу
GetCurrentProcess	Возвращает псевдоописатель текущего процесса
GetCurrentProcessId	Возвращает идентификатор текущего процесса
GetProcessVersion	Возвращает старший и младший номера версии Windows, необходимой для запуска указанного процесса
GetStartupInfo	Возвращает содержимое структуры STARTUPINFO, заданное при вызове *CreateProcess*
GetEnvironmentStrings	Возвращает адрес блока переменных окружения

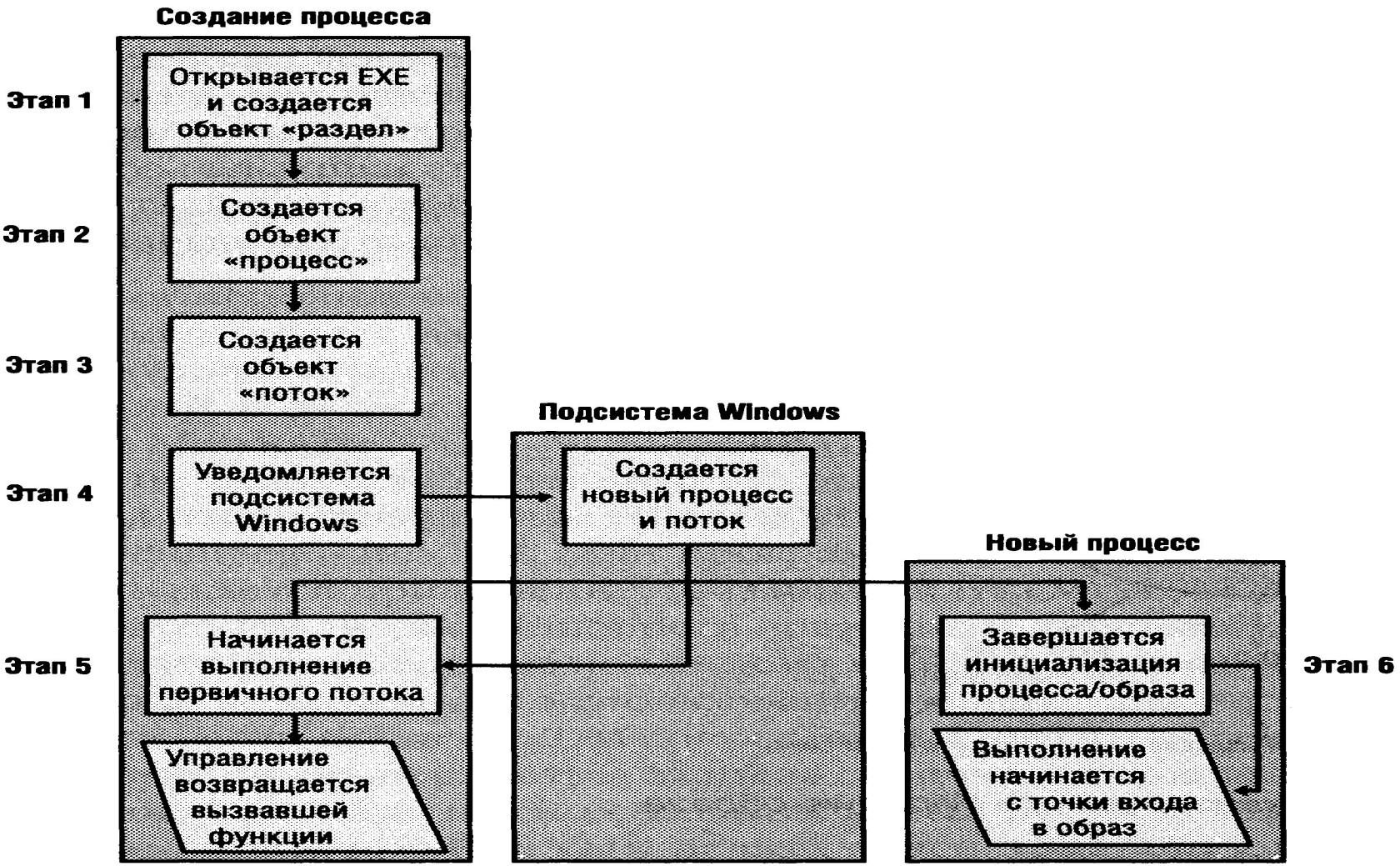
Основные этапы создания процесса



Выбор активируемого Windows- образа



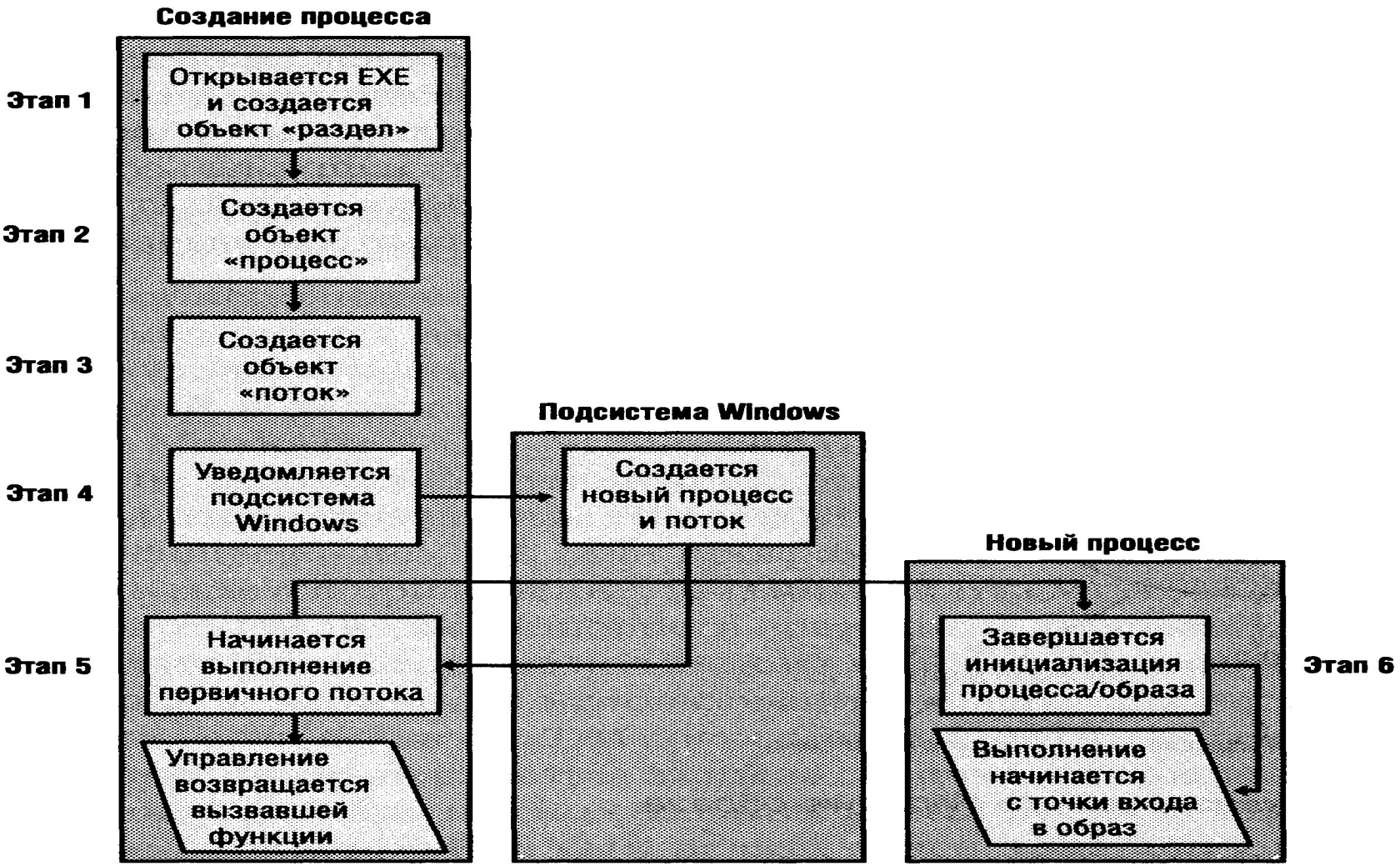
Основные этапы создания процесса



Этап 2: Создание объекта «процесс»

- формируется блок EPROCESS;
- создается начальное адресное пространство процесса;
- инициализируется блок процесса ядра (KPROCESS);
- инициализируется адресное пространство процесса (в том числе список рабочего набора и дескрипторы виртуального адресного пространства), а также проецируется образ на это пространство;
- формируется блок PEB;
- завершается инициализация объекта «процесс» исполнительной системы.

Основные этапы создания процесса



Этап 3: Создание первичного потока, его стека и контекста

1. Увеличивается счетчик потоков в объекте «процесс».
2. Создается и инициализируется блок потока исполнительной системы (ETHREAD).
3. Генерируется идентификатор нового потока.
4. В адресном пространстве пользовательского режима формируется TEB.
5. Стартовый адрес потока пользовательского режима сохраняется в блоке ETHREAD. В случае Windows-потоков это адрес системной стартовой функции потока в Kernel32.dll (*BaseProcessStart* для первого потока в процессе и *BaseThreadStart* для дополнительных потоков). Стартовый адрес, указанный пользователем, также хранится в ETHREAD, но в другом его месте; это позволяет системной стартовой функции потока вызвать пользовательскую стартовую функцию.

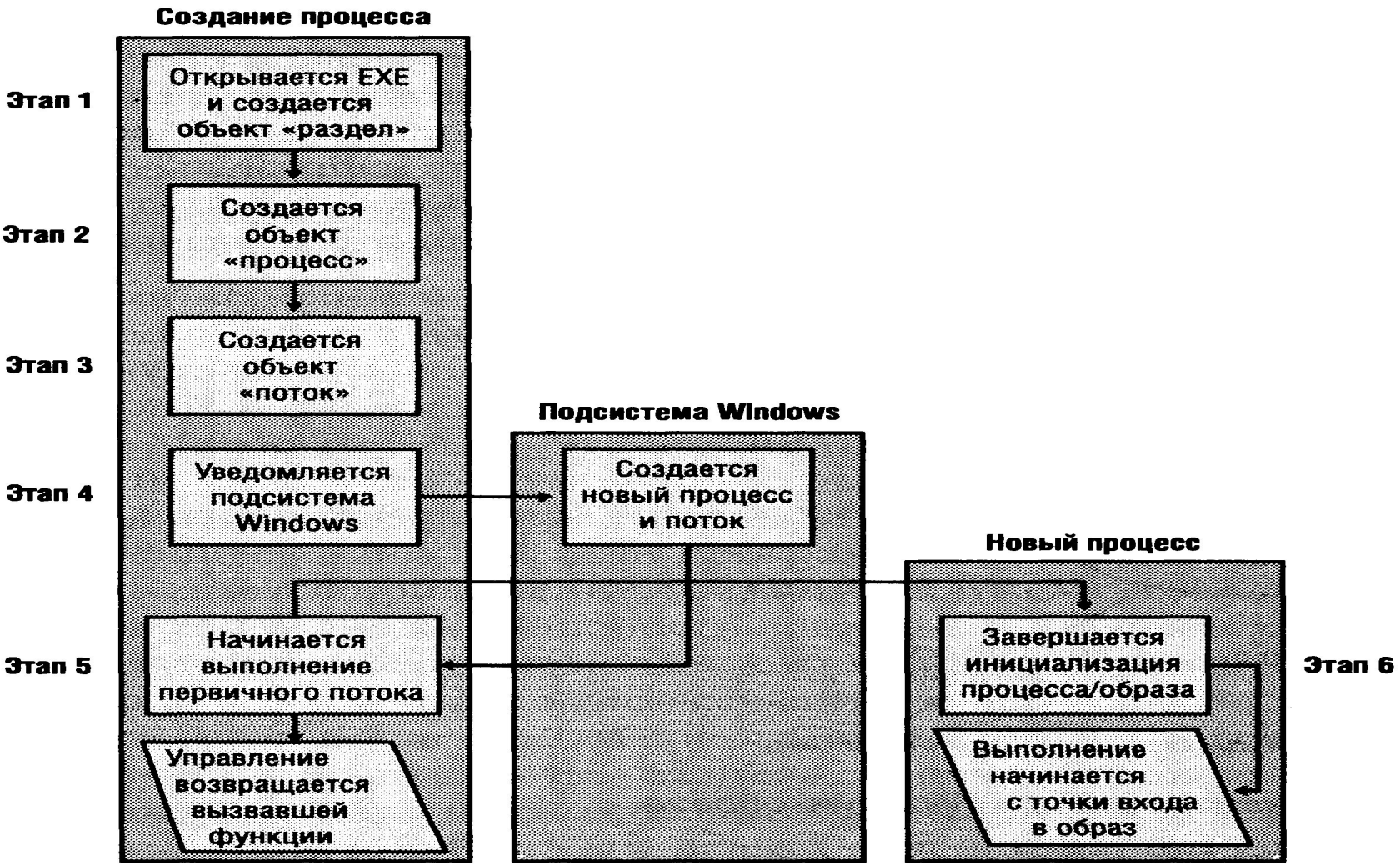
Этап 3: Создание первичного потока, его стека и контекста

6. Для подготовки блока KTHREAD вызывается *KeInitThread*. Начальный и текущий базовые приоритеты потока устанавливаются равными базовому приоритету процесса; привязка к процессорам и значение кванта также устанавливаются по соответствующим параметрам процесса. Кроме того, функция определяет идеальный процессор для первичного потока. (О том, как происходит выбор идеального процессора см. раздел «Идеальный и последний процессоры» далее в этой главе.) Затем *KeInitThread* создает стек ядра для потока и инициализирует его аппаратно-зависимый контекст, включая фреймы ловушек и исключений. Контекст потока настраивается так, чтобы выполнение этого потока началось в режиме ядра в *KiThreadStartup*. Далее *KeInitThread* устанавливает состояние потока в Initialized (инициализирован) и возвращает управление *PspCreateThread*.
7. Вызываются общесистемные процедуры, зарегистрированные на уведомление о создании потока.

Этап 3: Создание первичного потока, его стека и контекста

8. Маркер доступа потока настраивается как указатель на маркер доступа процесса. Затем вызывающая программа проверяется на предмет того, имеет ли она право создавать потоки. Эта проверка всегда заканчивается успешно, если поток создается в локальном процессе, но может дать отрицательный результат, если поток создается в другом процессе через функцию *CreateRemoteThread* и у создающего процесса нет привилегии отладки.
9. Наконец, поток готов к выполнению.

Основные этапы создания процесса



Этап 4: Уведомление подсистемы Windows о новом процессе

- описатели процесса и потока;
- флаги создания;
- идентификатор родительского процесса;
- флаг, который указывает, относится ли данный процесс к Windows-приложениям (что позволяет Csrss определить, показывать ли курсор запуска).

Реакция ОС на уведомление о новом процессе

1. *CreateProcess* дублирует описатели процесса и потока. На этом этапе счетчик числа пользователей процесса увеличивается с 1 (начального значения, установленного в момент создания процесса) до 2.
2. Если класс приоритета процесса не указан, *CreateProcess* устанавливает его в соответствии с алгоритмом, описанным ранее.
3. Создается блок процесса Csrss.
4. Порт исключений нового процесса настраивается как общий порт функций для подсистемы Windows, которая может таким образом получать сообщения при возникновении в процессе исключений (об обработке исключений см. главу 3).
5. Если в данный момент процесс отлаживается (т. е. подключен к процессу отладчика), в качестве общего порта функций выбирается отладочный порт. Такой вариант позволяет Windows пересылать события отладки в новом процессе (генерируемые при создании и удалении потоков, при исключениях и т. д.) в виде сообщений подсистеме Windows, которая затем доставляет их процессу, выступающему в роли отладчика нового процесса.

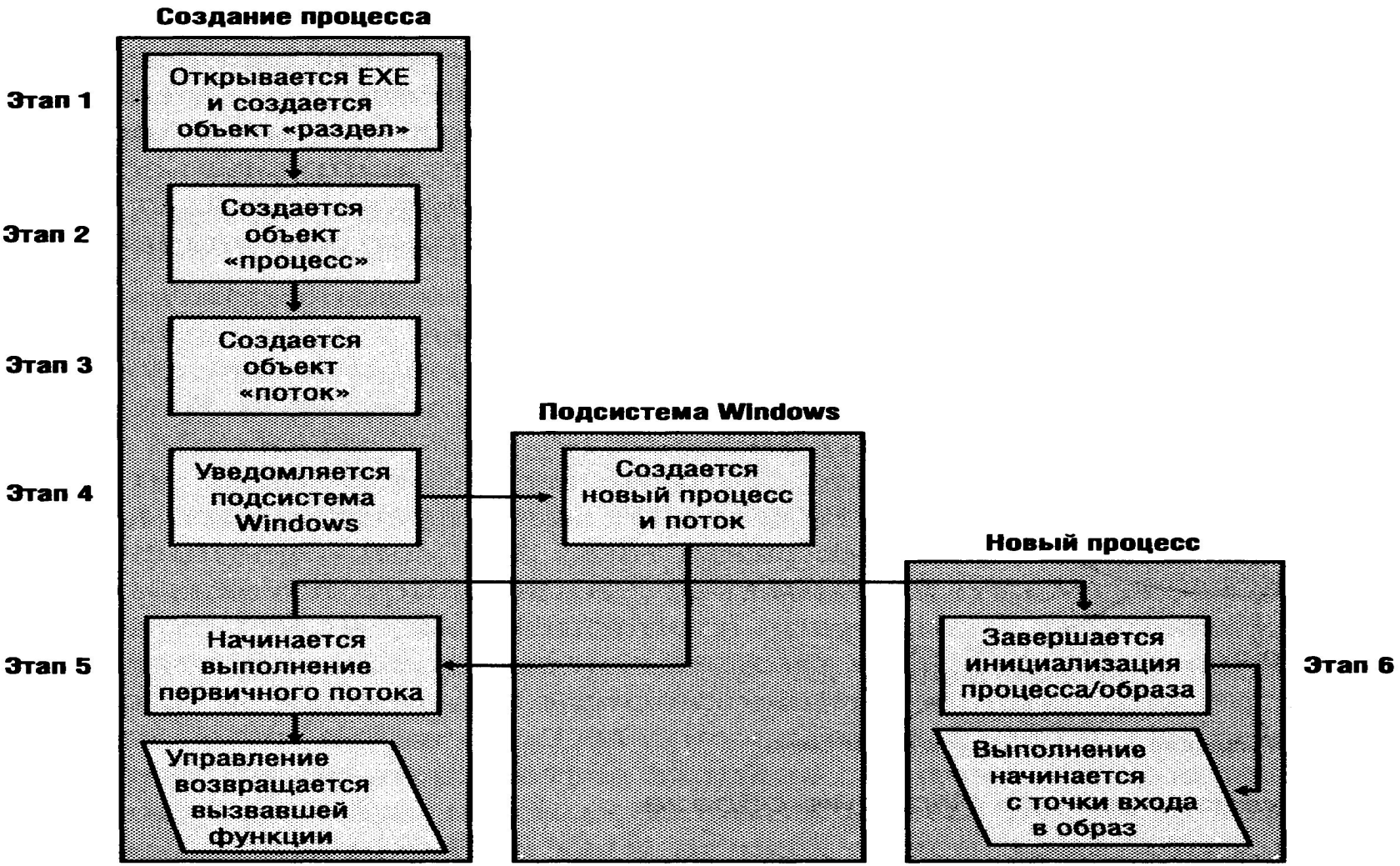
Реакция ОС на уведомление о новом процессе

6. Создается и инициализируется блок потока Csrss.
7. *CreateProcess* включает поток в список потоков процесса.
8. Увеличивается счетчик процессов в данном сеансе.
9. Уровень завершения процесса (process shutdown level) устанавливается как 0x280 (это значение по умолчанию; его описание ищите в документации MSDN Library по ключевому слову *SetProcessShutdownParameters*).
10. Блок нового процесса включается в список общесистемных Windows-процессов.
11. Создается и инициализируется структура данных (W32PROCESS), индивидуальная для каждого процесса и используемая той частью подсистемы Windows, которая работает в режиме ядра.

Реакция ОС на уведомление о новом процессе

12. Выводится курсор запуска в виде стрелки с песочными часами. Тем самым Windows говорит пользователю: «Я запускаю какую-то программу, но ты все равно можешь пользоваться курсором.» Если в течение двух секунд процесс не делает GUI-вызова, курсор возвращается к стандартному виду. А если за это время процесс обратился к GUI, *CreateProcess* ждет открытия им окна в течение пяти секунд и после этого восстанавливает исходную форму курсора.

Основные этапы создания процесса



Тема: Процессы и потоки в ОС Windows NT

2. Внутреннее устройство потоков.

Блок потока исполнительной системы

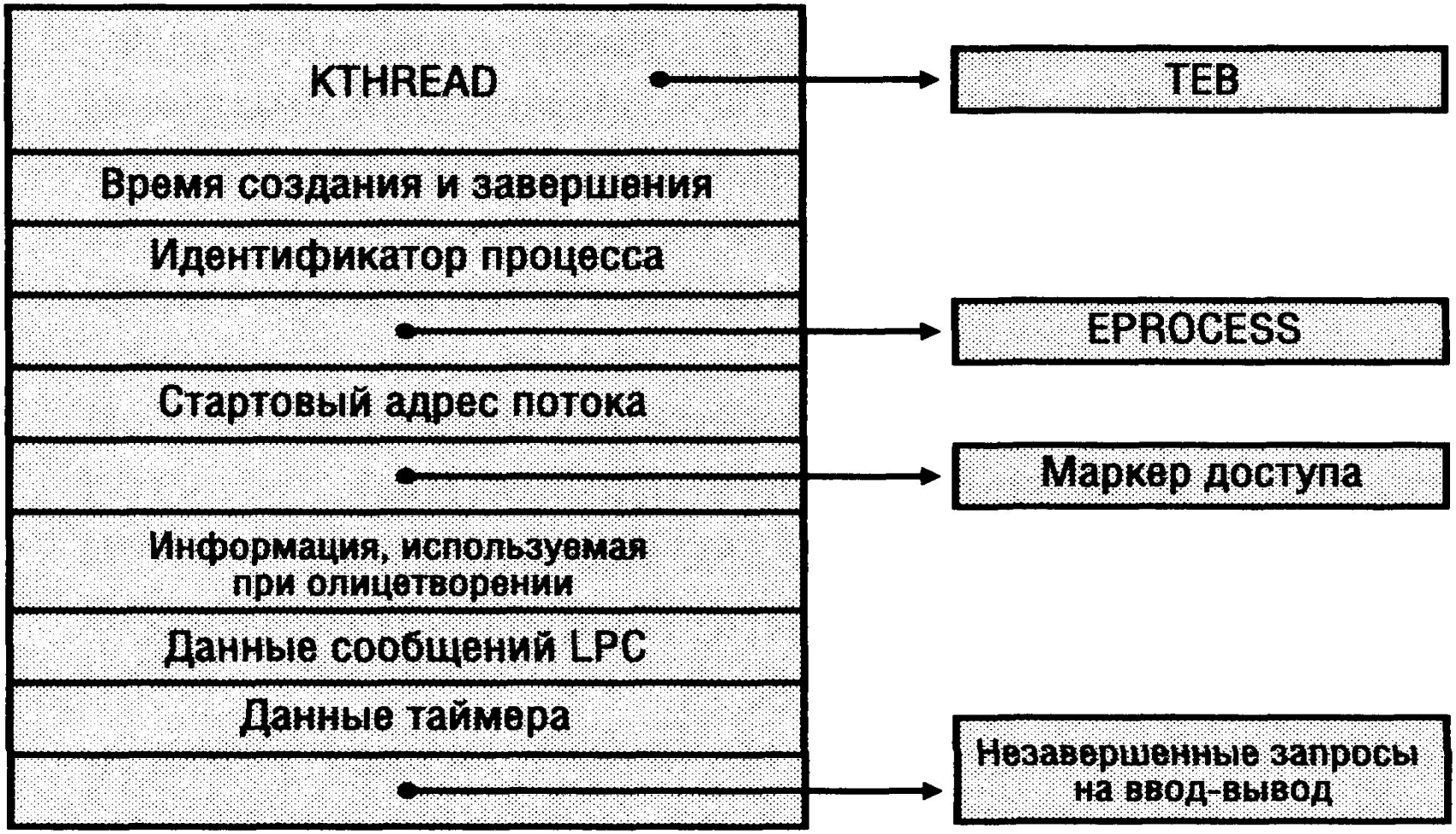
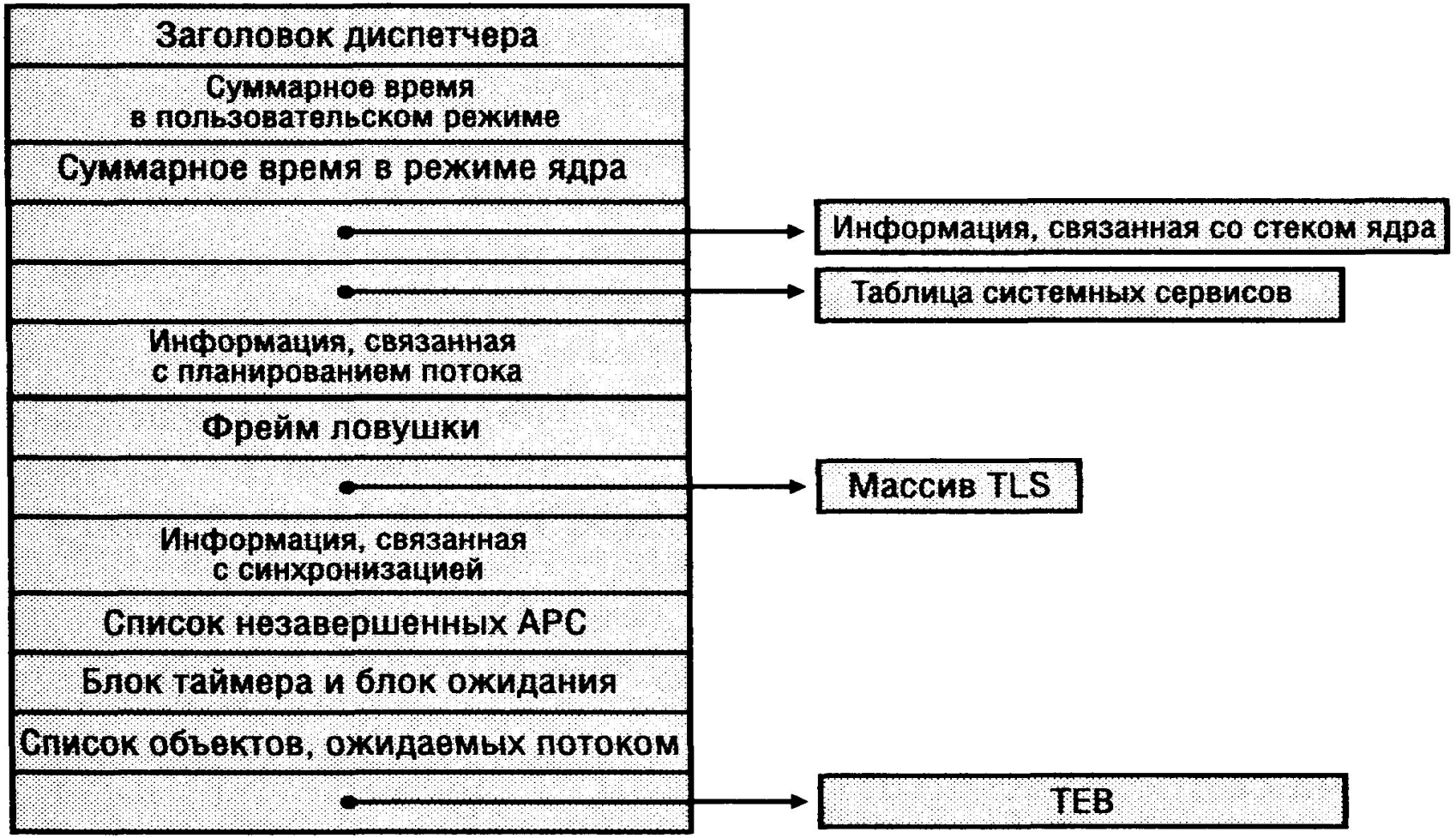
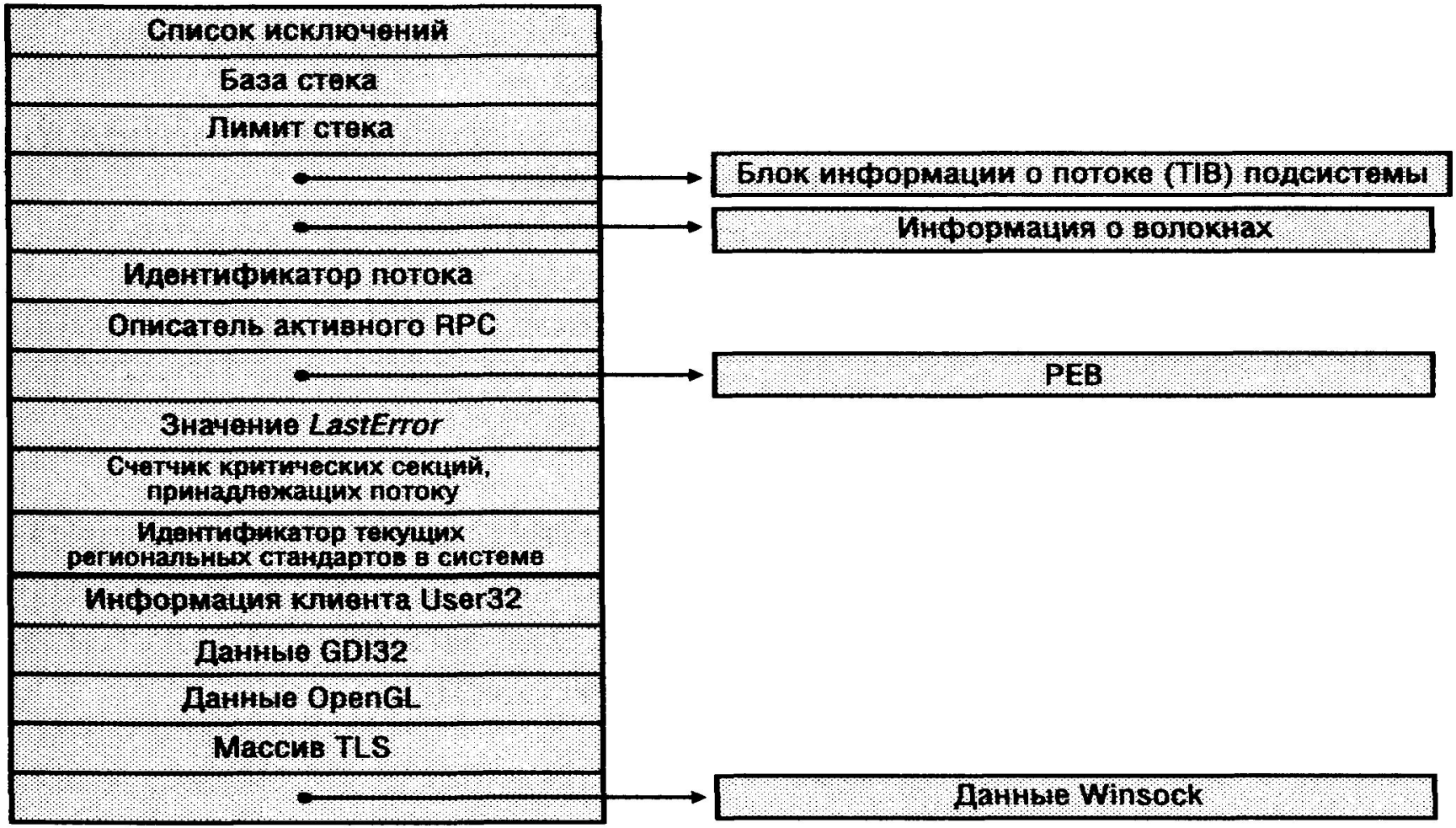


Схема блока потока ядра



Поля блока переменных окружения потока



Утилиты для исследования потоков и функций

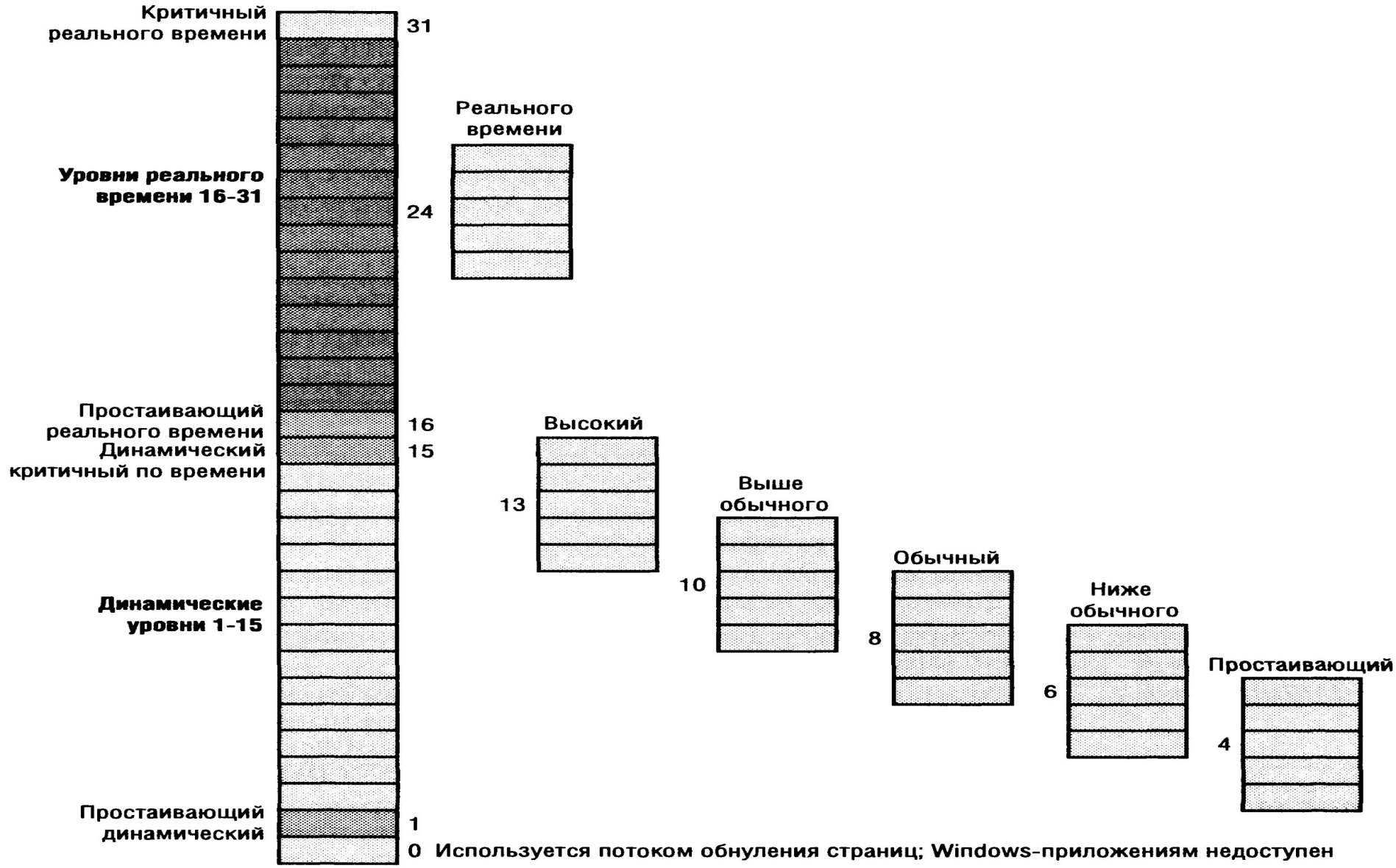
Объект	Perfmon	Pviewer	Pstat	Qslice	Tlist	KD *!thread*	Process Explorer	Pslist
ID потока	✓	✓	✓		✓	✓	✓	✓
Истинный стартовый адрес	✓	✓	✓			✓	✓	✓
Стартовый адрес Win32					✓	✓	✓	
Текущий адрес	✓	✓				✓	✓	
Число переключений контекста	✓	✓	✓				✓	✓
Общее время работы в пользовательском режиме		✓	✓			✓	✓	✓
Общее время работы в привилегированном режиме		✓	✓			✓	✓	✓
Прошедшее время	✓	✓				✓	✓	✓
Состояние потока	✓		✓		✓	✓	✓	✓
Причина перехода в состояние ожидания	✓		✓		✓	✓	✓	✓
Последняя ошибка					✓		✓	
% загруженности процессора	✓			✓			✓	
% работы в пользовательском режиме	✓			✓			✓	
% работы в привилегированном режиме	✓			✓			✓	
Адрес TEB						✓		
Адрес ETHREAD						✓		
Объекты, ожидаемые данным потоком						✓		

Тема: Процессы и потоки в ОС Windows NT

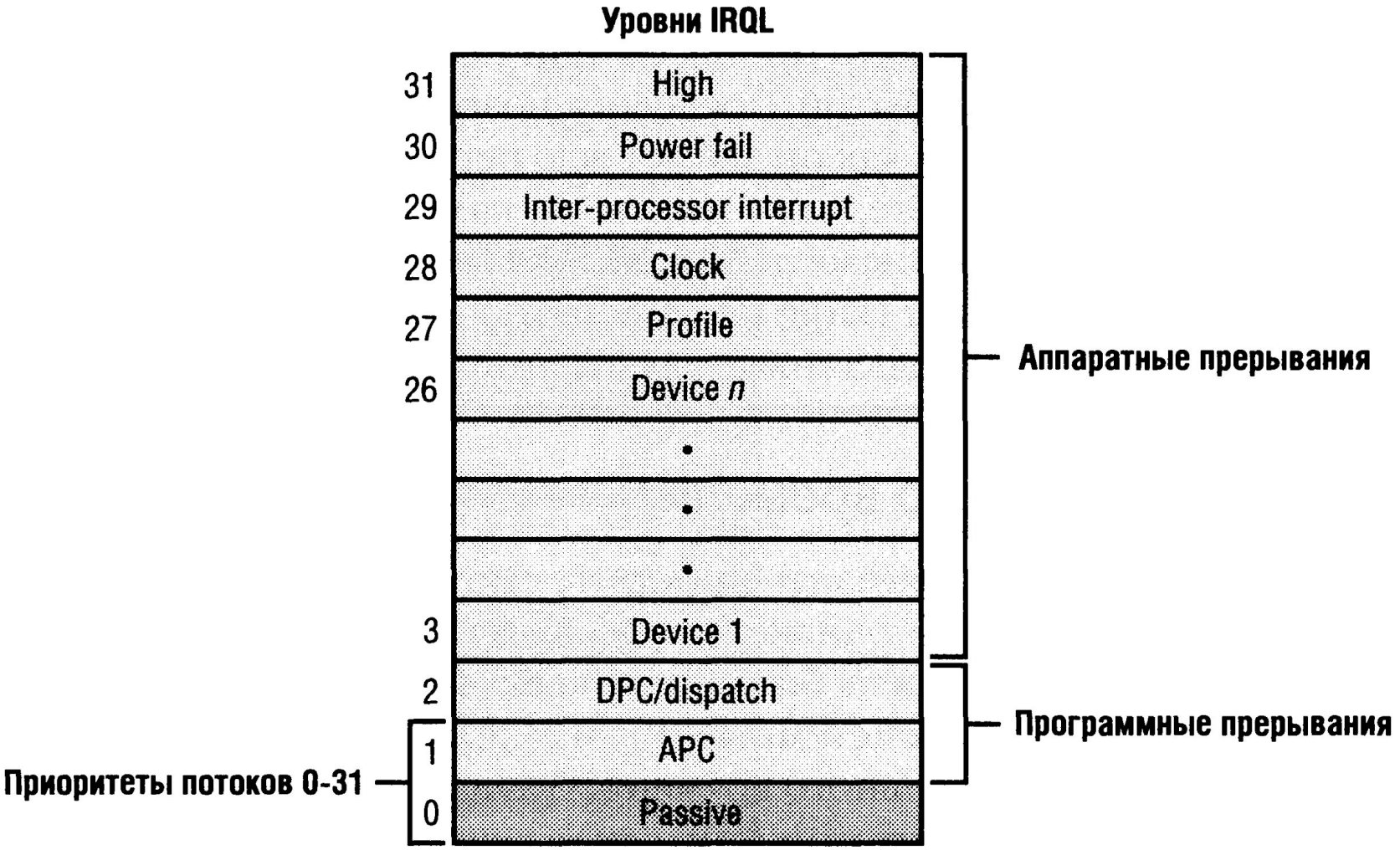
3. Планирование потоков.

Уровни приоритета потоков

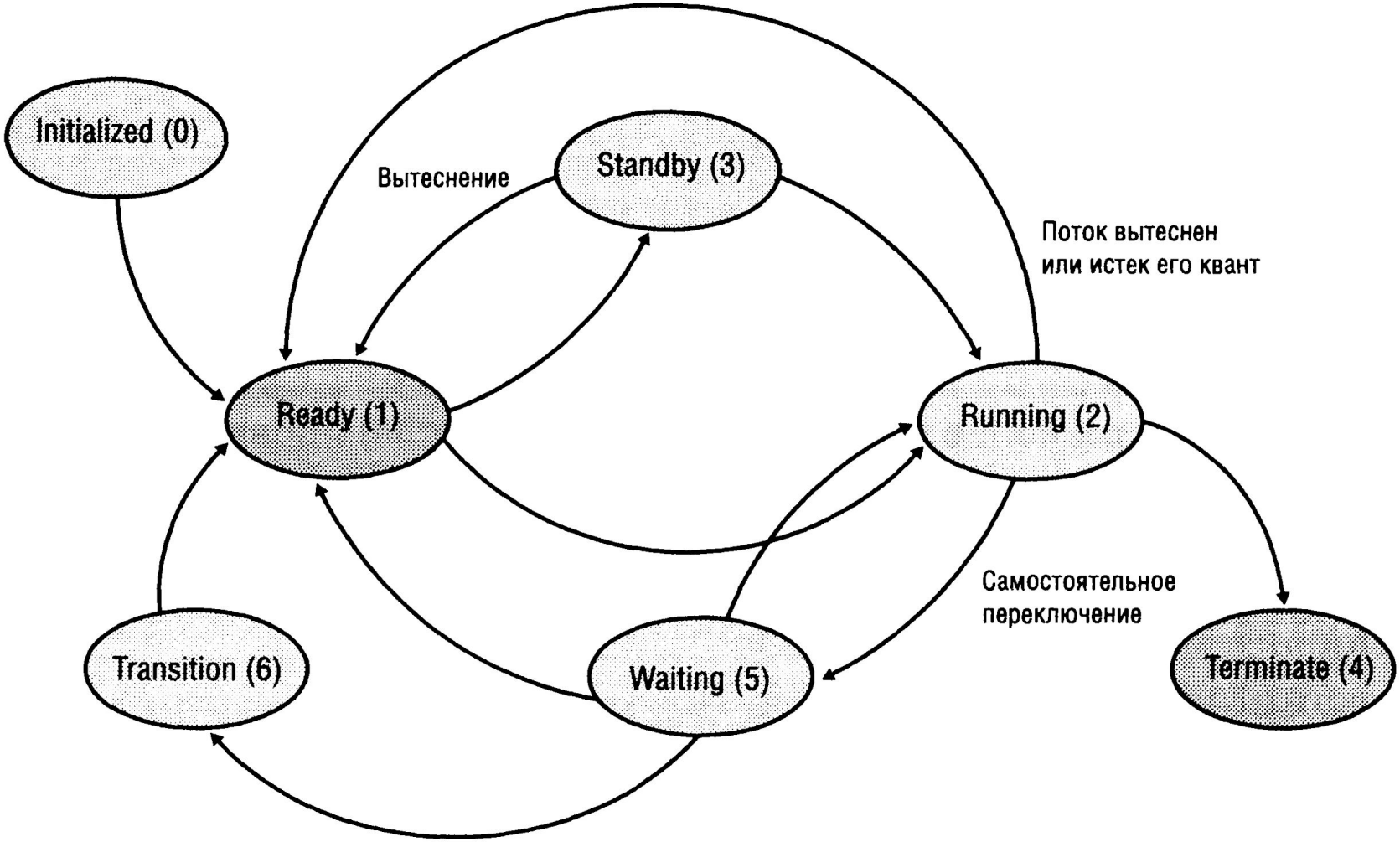
Взаимосвязь приоритетов в ядре и Windows API



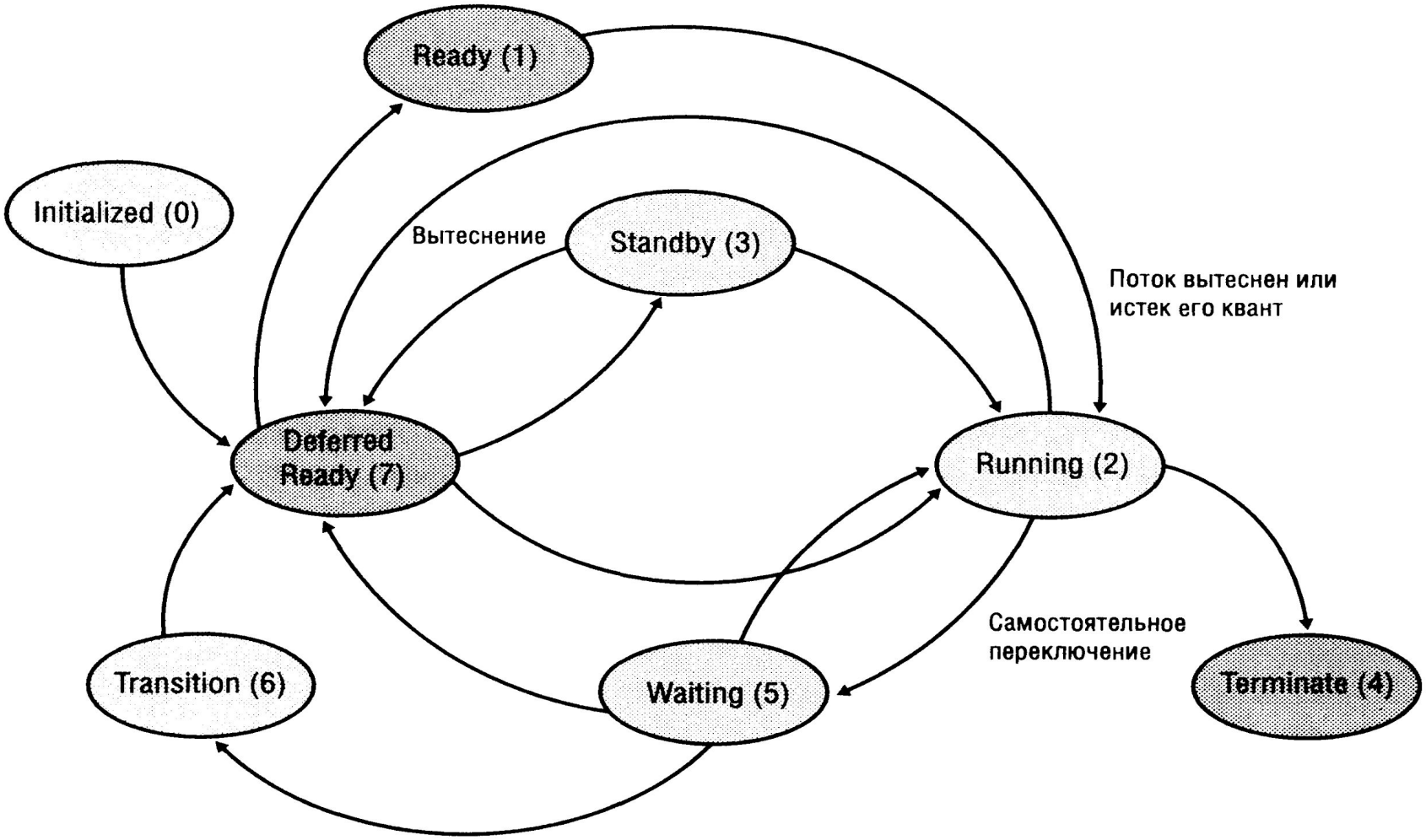
Уровни прерываний и уровни приоритета



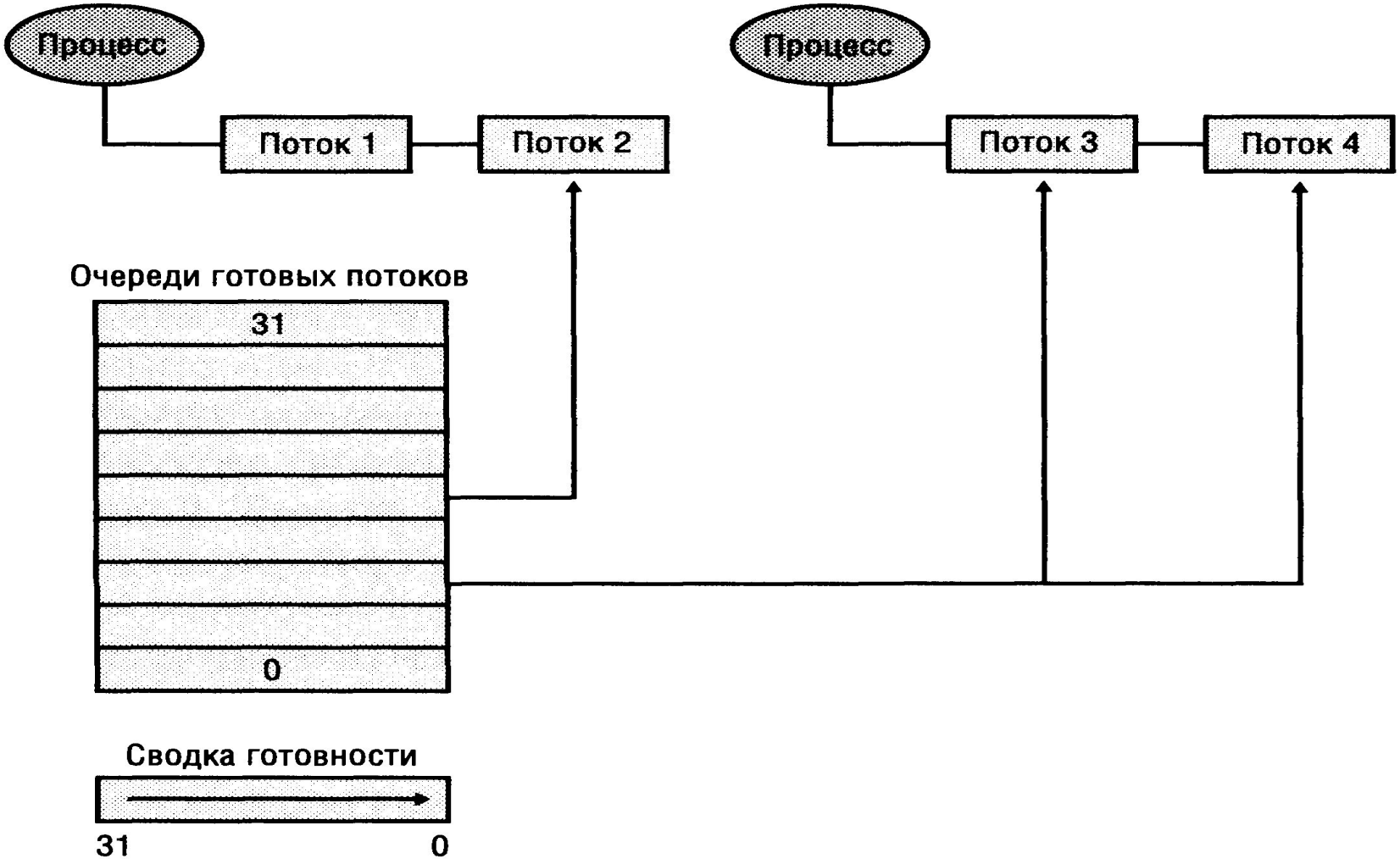
Состояния потоков в Windows XP



Состояния потоков в Windows Server 2003



База данных диспетчера ядра



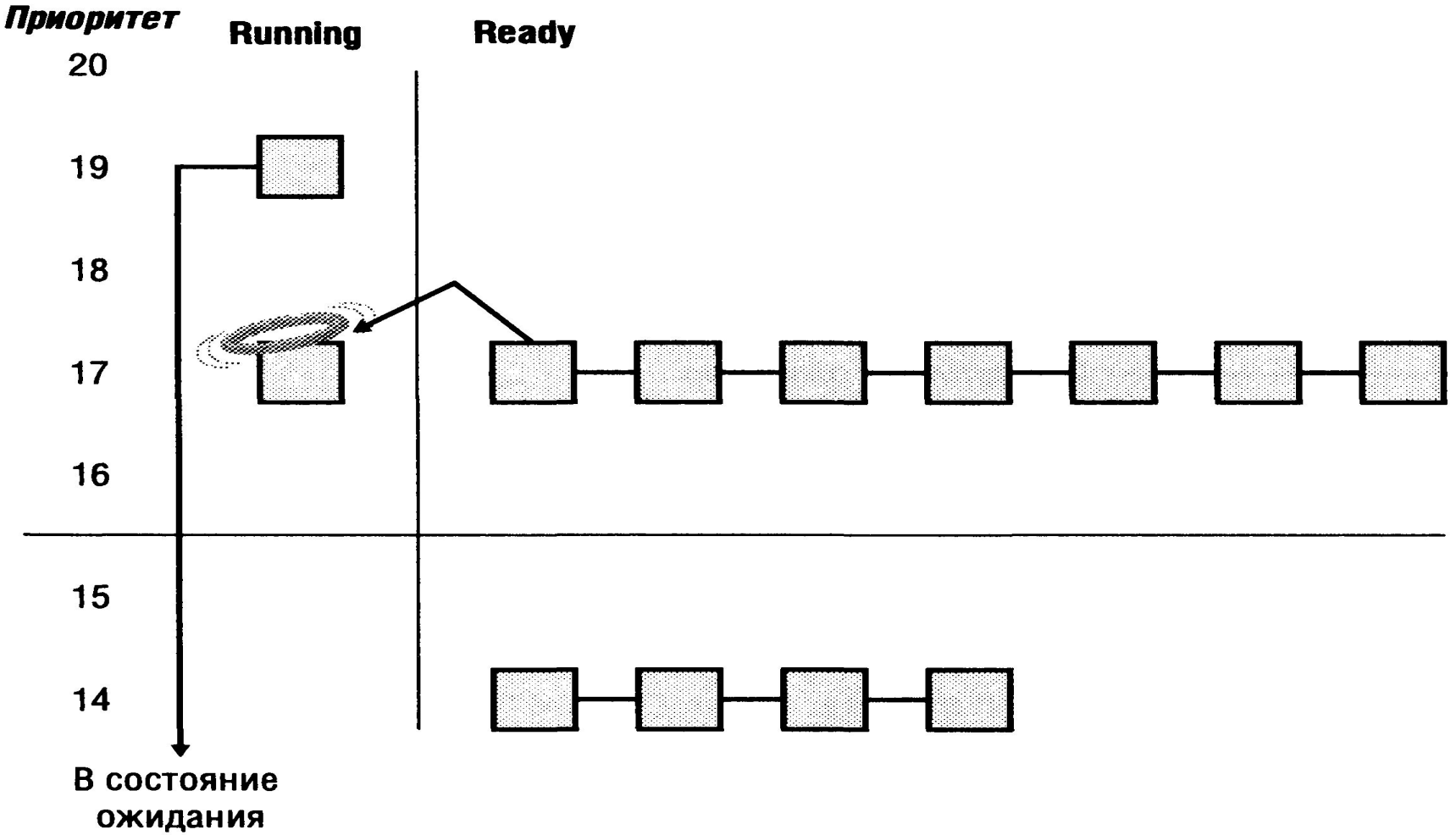
Величины квантов

	Короткие			Длинные		
Переменные	6	12	18	12	24	36
Фиксированные	18	18	18	36	36	36

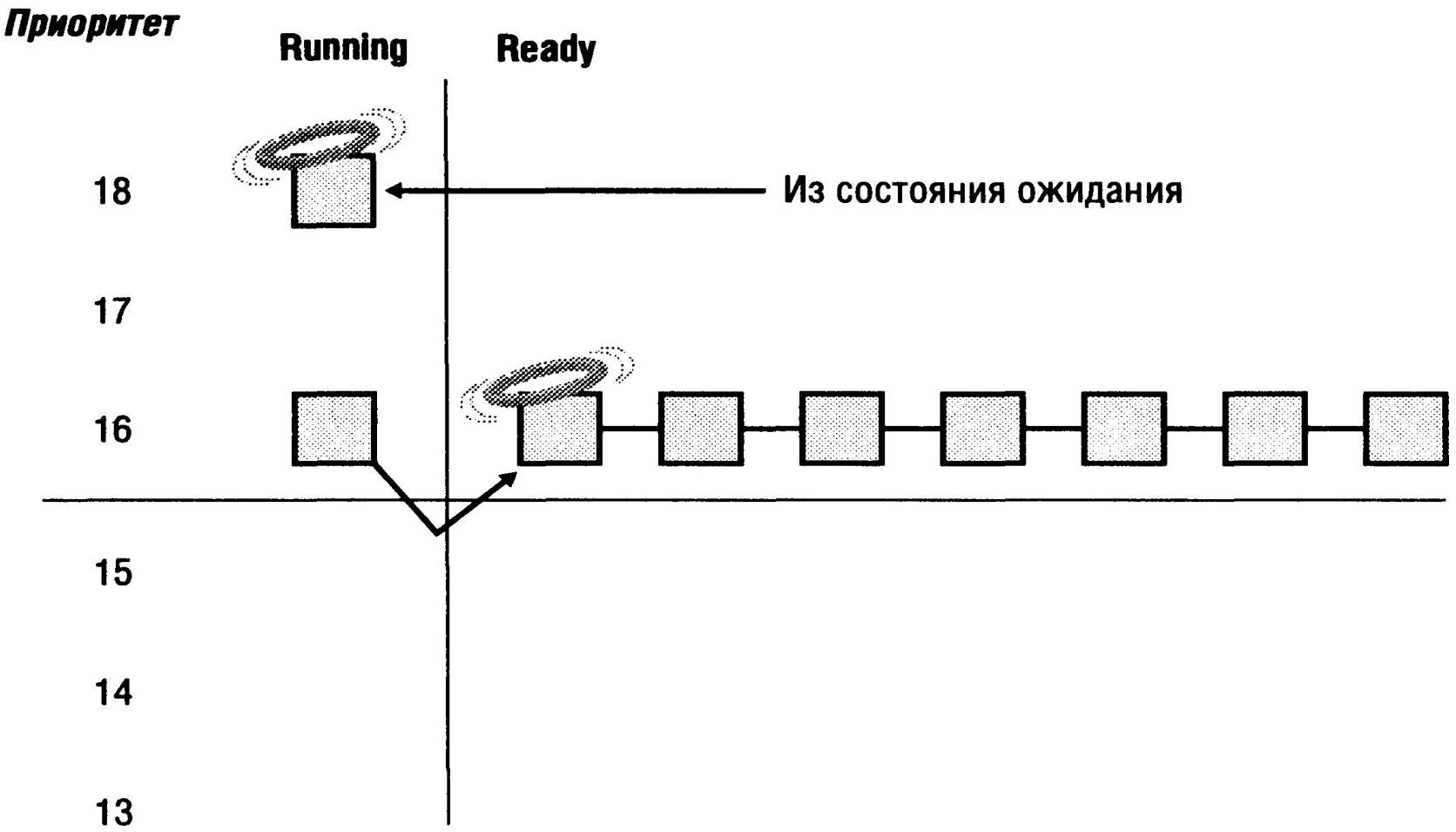
2*3 = 6*10 мс = 60 мс (рабочая станция)

12*3 = 36*15 мс = 540 мс (сервер)

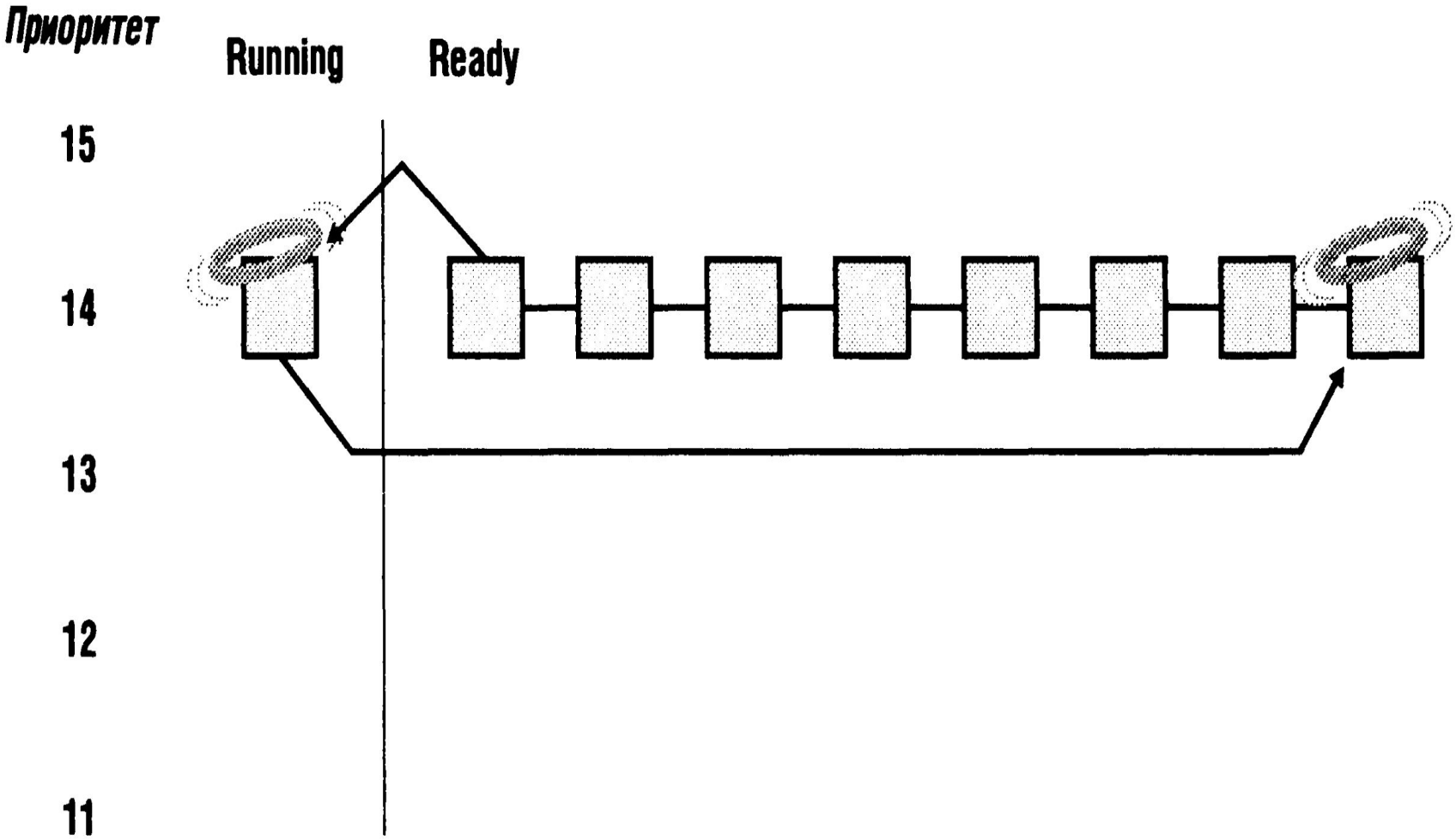
Самостоятельное переключение



Планирование потоков с вытеснением



Планирование потоков в момент завершения кванта текущего потока



Рекомендованные приращения приоритета

Устройство	Приращение приоритета
Диск, CD-ROM, параллельный порт, видео	1
Сеть, почтовый ящик, именованный канал, последовательный порт	2
Клавиатура, мышь	6
Звуковая плата	8

Динамическое изменение приоритета