

The background features a light pink grid. In the upper center, there are four realistic water droplets. On the right side, there are large, overlapping abstract shapes in blue, red, and yellow. The main title is centered in a bold, dark blue font.

Планирование процессов

L/O/G/O

Основные понятия планирования

- **Планирование** - обеспечение поочередного доступа процессов к одному процессору.
- **Планировщик** - отвечающая за это часть ОС.
- **Алгоритм планирования** - используемый алгоритм для планирования.

Основные понятия планирования

- Ситуации, когда необходимо планирование:
 - Когда создается процесс
 - Когда процесс завершает работу
 - Когда процесс блокируется на операции ввода/вывода, семафоре, и т.д.
 - При прерывании ввода/вывода.

Основные понятия планирования

Виды систем:

- *Системы пакетной обработки* - могут использовать неприоритетный и приоритетный алгоритм
- *Интерактивные системы* - могут использовать только приоритетный алгоритм, нельзя допустить чтобы один процесс занял надолго процессор
- *Системы реального времени* - могут использовать неприоритетный и приоритетный алгоритм

Задачи алгоритмов планирования

Для всех систем

- Справедливость - каждому процессу справедливую долю процессорного времени
- Контроль над выполнением принятой политики
- Баланс - поддержка занятости всех частей системы (например: чтобы были заняты процессор и устройства ввода/вывода)

Задачи алгоритмов планирования

Системы пакетной обработки

- Пропускная способность - количество задач в час
- Оборотное время - минимизация времени на ожидание обслуживания и обработку задач.
- Использование процессора - чтобы процессор всегда был занят.

Задачи алгоритмов планирования

Интерактивные системы

- Время отклика - быстрая реакция на запросы
- Соразмерность - выполнение ожиданий пользователя

Системы реального времени

- Окончание работы к сроку - предотвращение потери данных
- Предсказуемость -

Основные понятия планирования

- **Алгоритм планирования без переключений (неприоритетный)** - не требует прерывание по аппаратному таймеру, процесс останавливается только когда блокируется или завершает работу.

Основные понятия планирования

- **Алгоритм планирования с переключениями (приоритетный)** - требует прерывание по аппаратному таймеру, процесс работает только отведенный период времени, после этого он приостанавливается по таймеру, чтобы передать управление планировщику.

Механизмы планирования

- **Таймер** – позволяет отсчитывать время выполнения процесса в процессоре и регулировать загрузку процессора
- **Переключение** – позволяет подавать сигналы ядру на приостановку / возобновление процесса с переключением контекста
- **Приоритеты** – позволяют установить порядок переключения процессов в зависимости от различных факторов выполнения процессов

Планирование в системах пакетной обработки

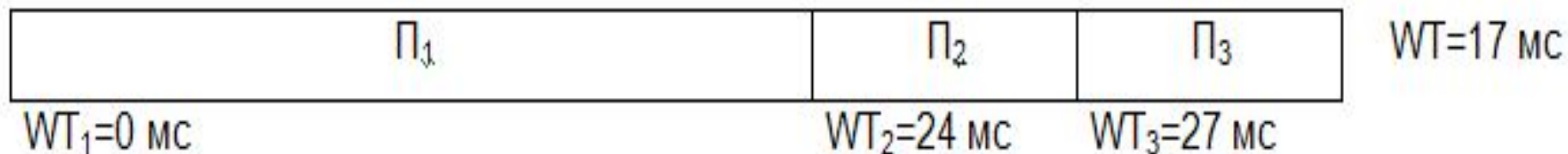
"Первый пришел - первым обслужен" (FIFO - First In First Out)

- процессор передается тому процессу, который раньше всех других его запросил.
- среднее время ожидания для стратегии FIFO часто весьма велико и зависит от порядка поступления процессов в очередь готовых процессов.

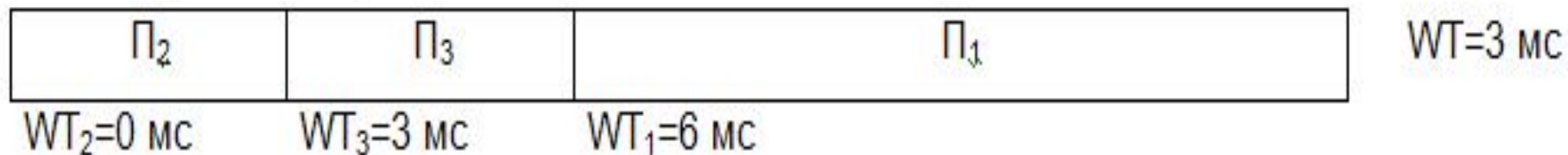
FIFO

- Пусть три процесса попадают в очередь одновременно в момент 0 и имеют следующие значения времени последующего обслуживания на ЦП
- Вариант 1: П1 (24 мс), П2 (3 мс), П2(3 мс)
- Вариант 2: П1 (3 мс), П2 (3 мс), П2(24 мс)

вариант 1:



вариант 2:



FIFO

Преимущества:

- Простота
- Справедливость

Недостатки:

- Процесс, ограниченный возможностями процессора может затормозить более быстрые процессы, ограниченные устройствами ввода/вывода.

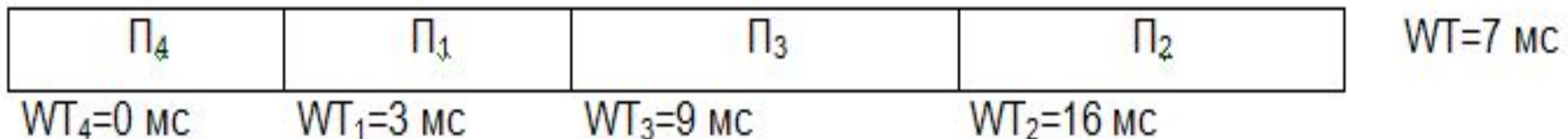
Кратчайшая задача – первая (SJF – Shortest Job First)

Пусть 4 процесса попадают в очередь одновременно в момент 0 имеют следующие значения времени последующего обслуживания на ЦП: П1 (6 мс), П2 (8 мс), П3 (7 мс), П4 (3 мс)

■ FIFO



■ SJF



Кратчайшая задача – первая (SJF – Shortest Job First)

Преимущества:

- Уменьшение оборотного времени
- Справедливость

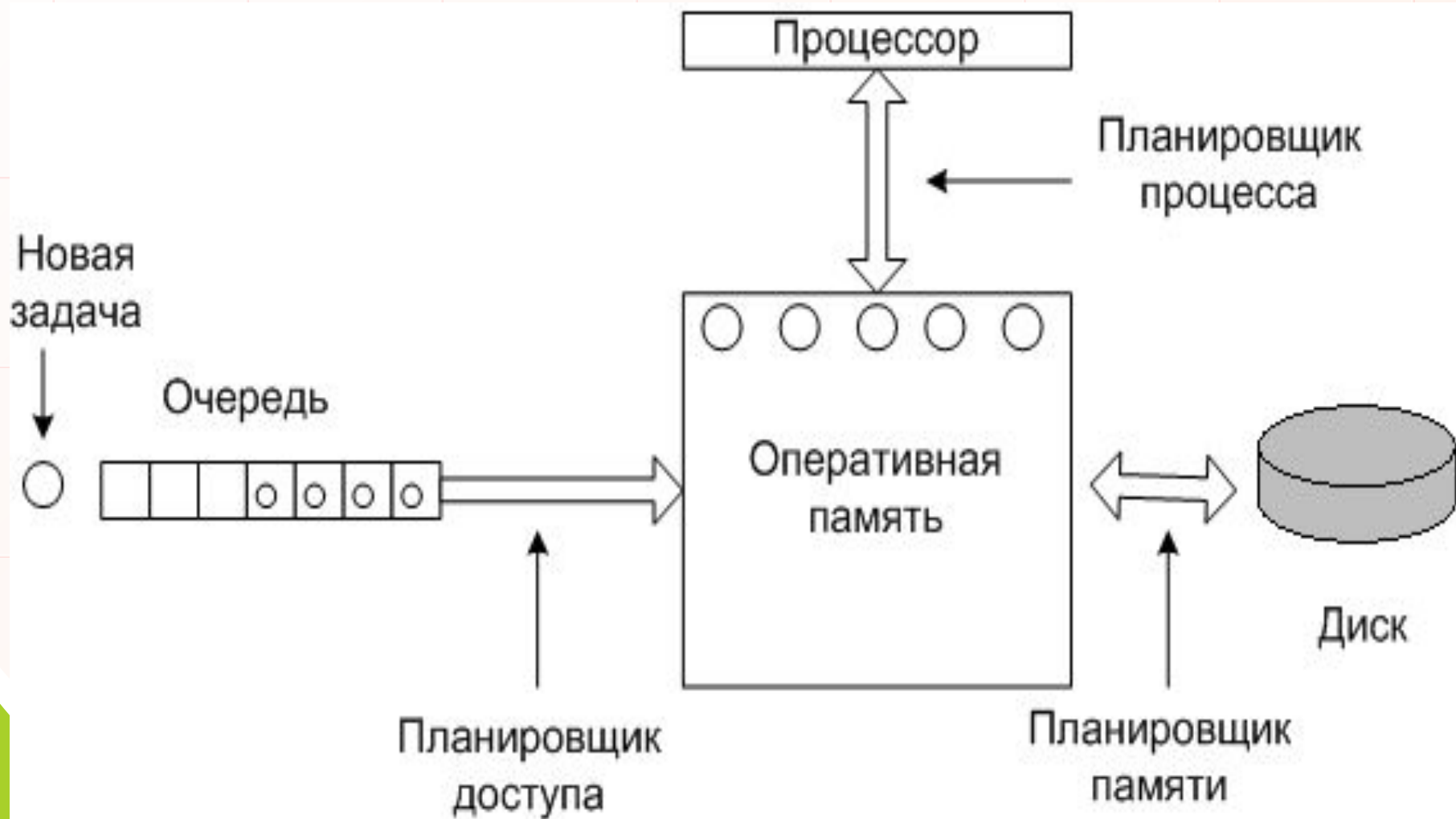
Недостатки:

- Длинный процесс, занявший процессор, не пустит более новые короткие процессы, которые пришли позже.

Наименьшее оставшееся время выполнения (SRT – Shortest Remaining Time)

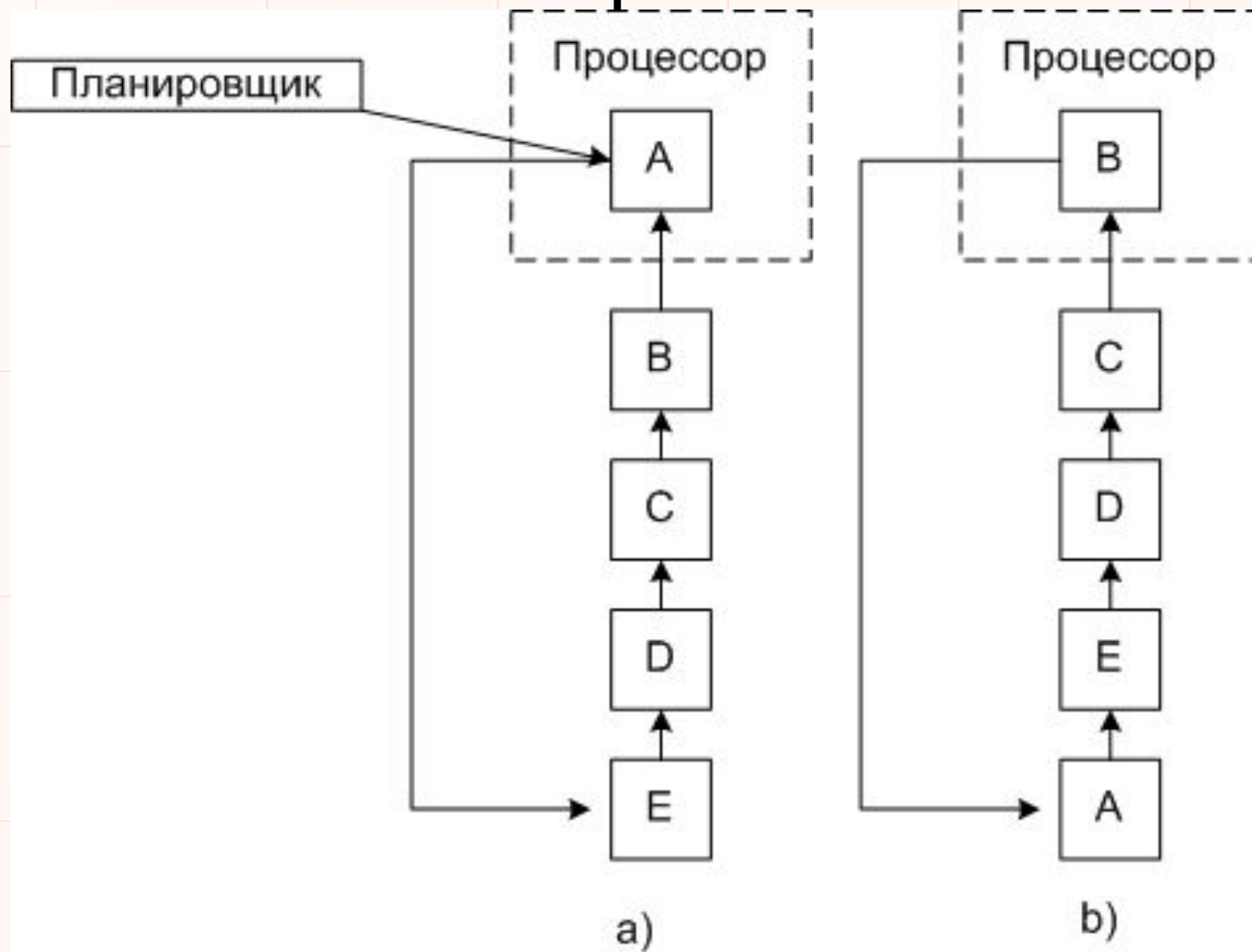
- Аналог SJF, но с переключениями.
- Если приходит новый процесс, его полное время выполнения сравнивается с оставшимся временем выполнения текущего процесса и выполняется тот процесс, которому осталось наименьшее время выполнения

Трехуровневое планирование



Планирование в интерактивных системах

- Циклическое планирование



Циклическое планирование

- Каждому процессу предоставляется квант времени процессора.
- Когда квант заканчивается процесс переводится планировщиком в конец очереди.
- При блокировке процесс выпадает из очереди

Пример:

П1 (24 мс), П2 (3 мс), П3 (3 мс); $q = 4$ мс

П ₁	П ₂	П ₃	П ₁	П ₁	П ₁	П ₁	П ₁
WT ₁ =0 мс	7	10	14	18	22	26	30

Циклическое планирование

Преимущества:

- Простота
- Справедливость

Недостатки:

- При малом кванте - частые переключения, в результате уменьшение производительности
- При большом кванте - редкие переключения, в результате происходит увеличение времени ответа на запрос (приближается к FIFO).

Приоритетное планирование

- Каждому процессу присваивается **приоритет**, и управление передается процессу с самым высоким приоритетом
- Приоритет может быть динамический и статический.

Динамический приоритет может устанавливаться следующим образом:

$$P = 1/T$$

где T- часть использованного кванта

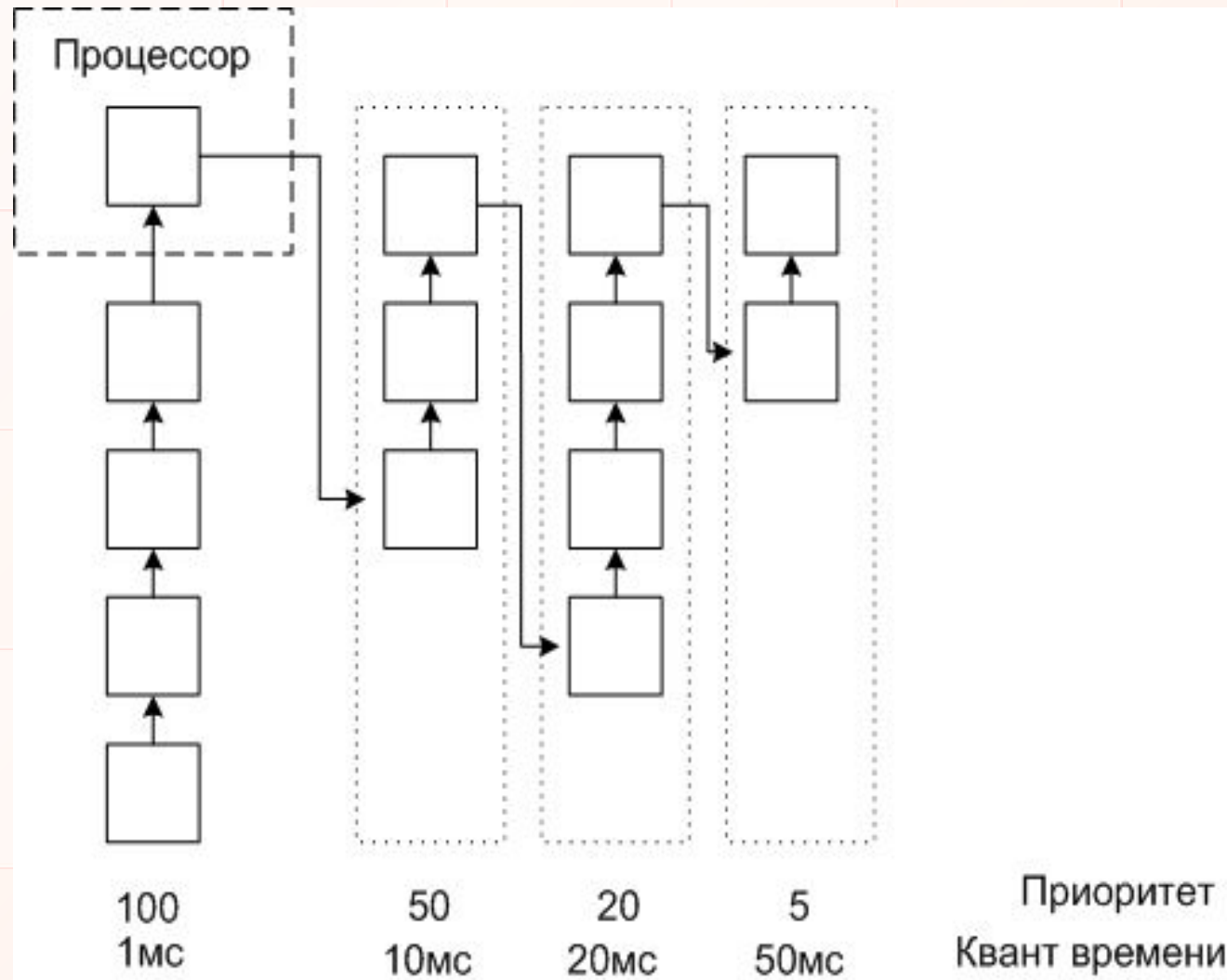
Например, если $T = 1/50$, то приоритет 50, если использован весь квант, то приоритет 1.

Приоритетное планирование

- Часто процессы объединяют по приоритетам в группы, и используют
 - среди групп - приоритетное планирование
 - внутри группы - циклическое планирование
- Методы разделения процессов на группы
 - Группы с разным квантом времени
 - Группы с разным назначением процессов

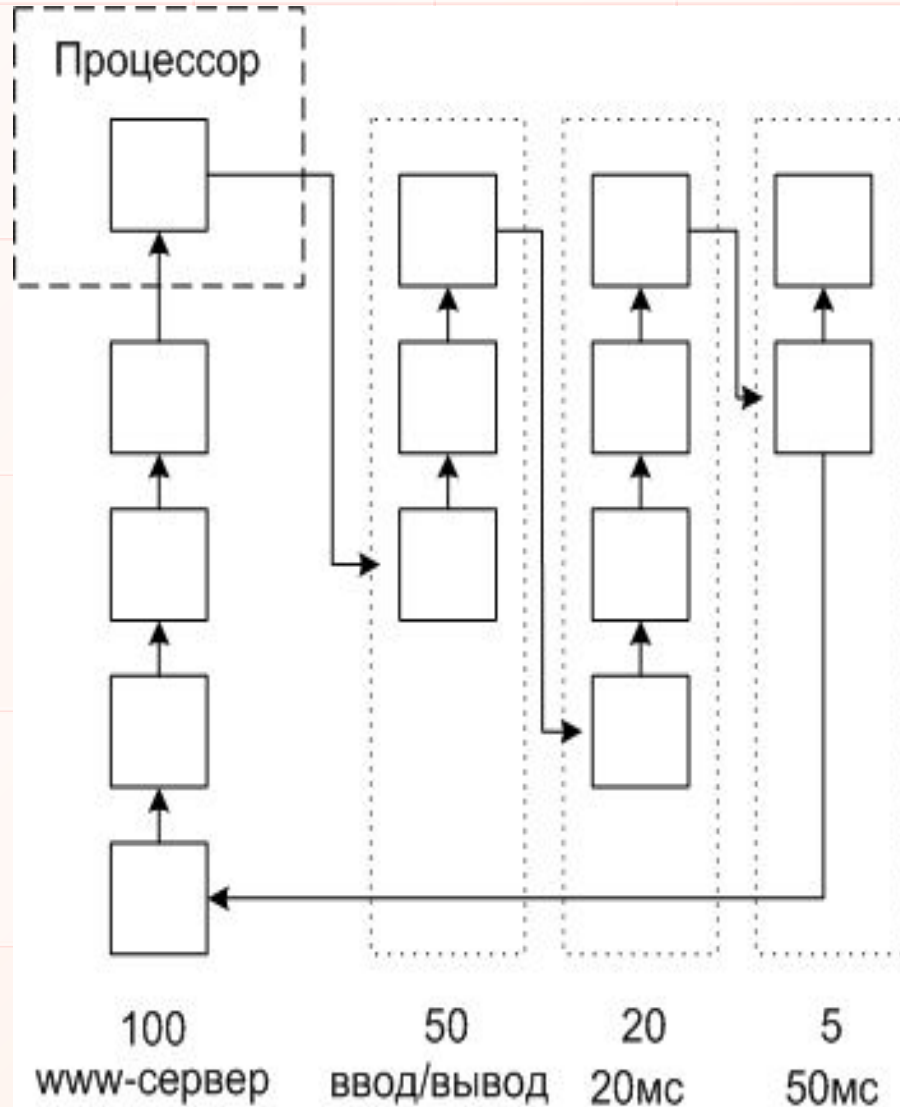
Группы с разным квантом времени

Процесс либо заканчивает работу, либо переходит в другую группу



Группы с разным назначением процессов

Процесс, отвечающий на запрос, переходит в группу с наивысшим приоритетом



Планирование в интерактивных системах

- **Гарантированное планирование**

В системе с n -процессами, каждому процессу будет предоставлено $1/n$ времени процессора.

- **Справедливое планирование**

Процессорное время распределяется среди пользователей, а не процессов.

Планирование в интерактивных системах

- Лотерейное планирование

Процессам раздаются "лотерейные билеты" на доступ к ресурсам.

Планировщик может выбрать любой билет, случайным образом. Чем больше билетов у процесса, тем больше у него шансов захватить ресурс.

Планирование в системах реального времени

Системы реального времени делятся на:

- **жесткие** (жесткие сроки для каждой задачи) - управление движением
- **гибкие** (нарушение временного графика не желательны, но допустимы) - управление видео и аудио

Планирование в системах реального времени

Внешние события, на которые система должна реагировать, делятся:

- **периодические** - потоковое видео и аудио
- **непериодические (непредсказуемые)** - сигнал о пожаре

Планирование в системах реального времени

- Чтобы систему реального времени можно было планировать, нужно чтобы выполнялось условие:

$$\sum_{i=1}^m \frac{T(i)}{P(i)} \leq 1$$

m - число периодических событий

i - номер события

P(i) - период поступления события

T(i) - время, которое уходит на обработку события

Перегруженная система реального времени является **непланируемой**

Общее планирование реального времени

Каждый процесс борется за процессор со своим заданием и графиком его выполнения.

Общее планирование реального времени

Планировщик должен знать:

- **Частоту** , с которой должен работать процесс
- **объем работ**, который ему предстоит выполнить
- **ближайший срок выполнения** очередной порции задания

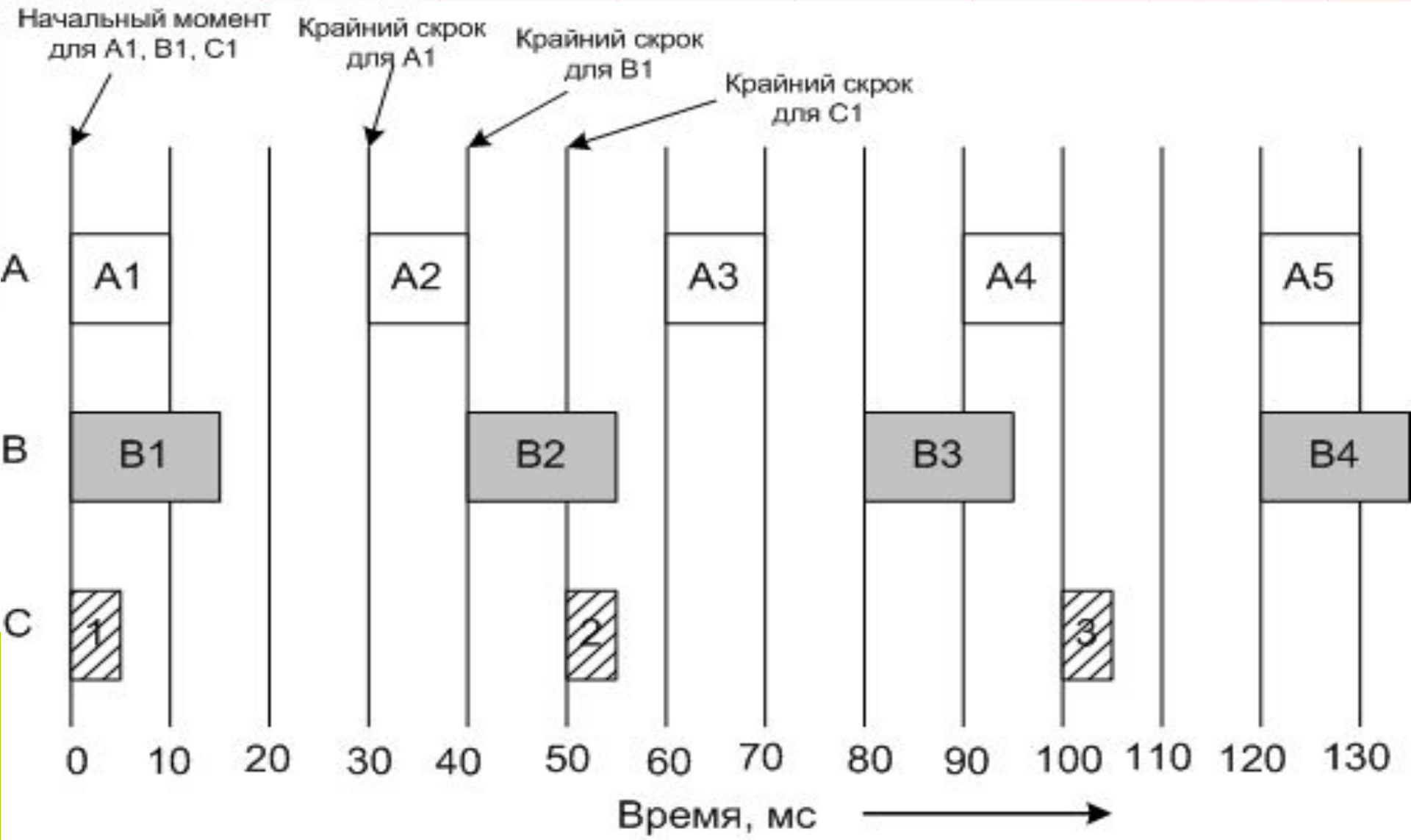
Общее планирование реального времени

Пример: имеются 3 периодических процесса.

- Процесс **A** запускается каждые 30мс, обработка - 10мс
- Процесс **B** частота = 25 (т.е. каждые 40мс), обработка - 15мс
- Процесс **C** частота = 20 (т.е. каждые 50мс), обработка кадра 5мс

$$10/30+15/40+5/50=0.808<1$$

Общее планирование реального времени



Общее планирование реального времени

- Различают 2 алгоритма планирования в системах реального времени:
 - **Статический алгоритм планирования RMS** (Rate Monotonic Scheduling) –
 - Процессы выполняются по приоритету
 - Приоритет пропорционален частоте
 - **Динамический алгоритм планирования EDF** (Earliest Deadline First)
 - Наибольший приоритет выставляется процессу, у которого осталось наименьшее время выполнения

Алгоритм планирования RMS

Процессы должны удовлетворять условиям:

1. Процесс должен быть завершен за время его периода
2. Один процесс не должен зависеть от другого
3. Каждому процессу требуется одинаковое процессорное время на каждом интервале
4. У непериодических процессов нет жестких сроков
5. Прерывание процесса происходит мгновенно

Сравнение RMS и EDF

■ Пример 1

Процесс	A	B	C
Время T	10	15	5
Период P (мс)	30	40	50
Частота (1/с)	33	25	20
Приоритет	33	25	20

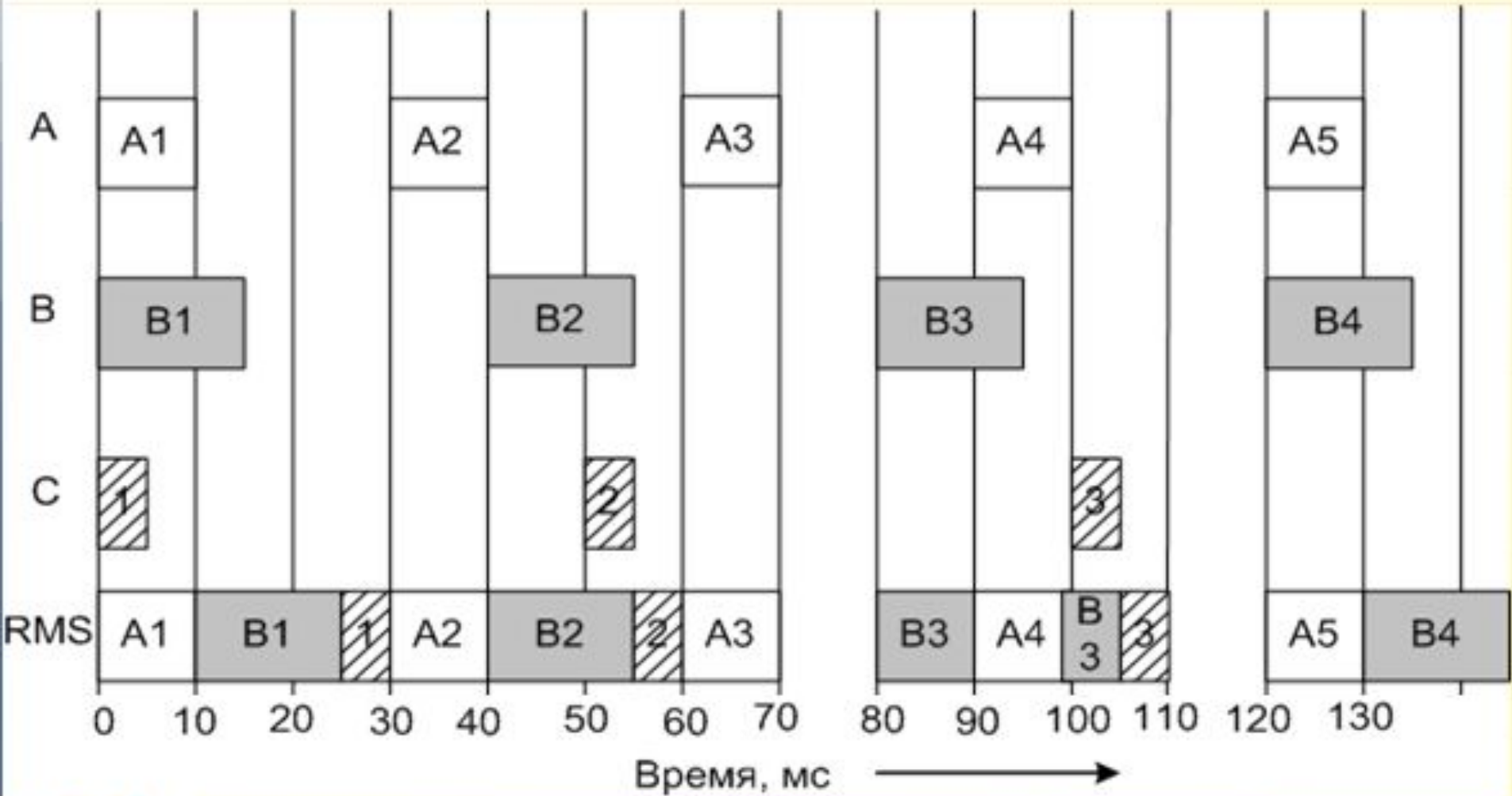
$$10/30 + 15/40 + 5/50 = 0.808 < 1$$

■ Пример 2

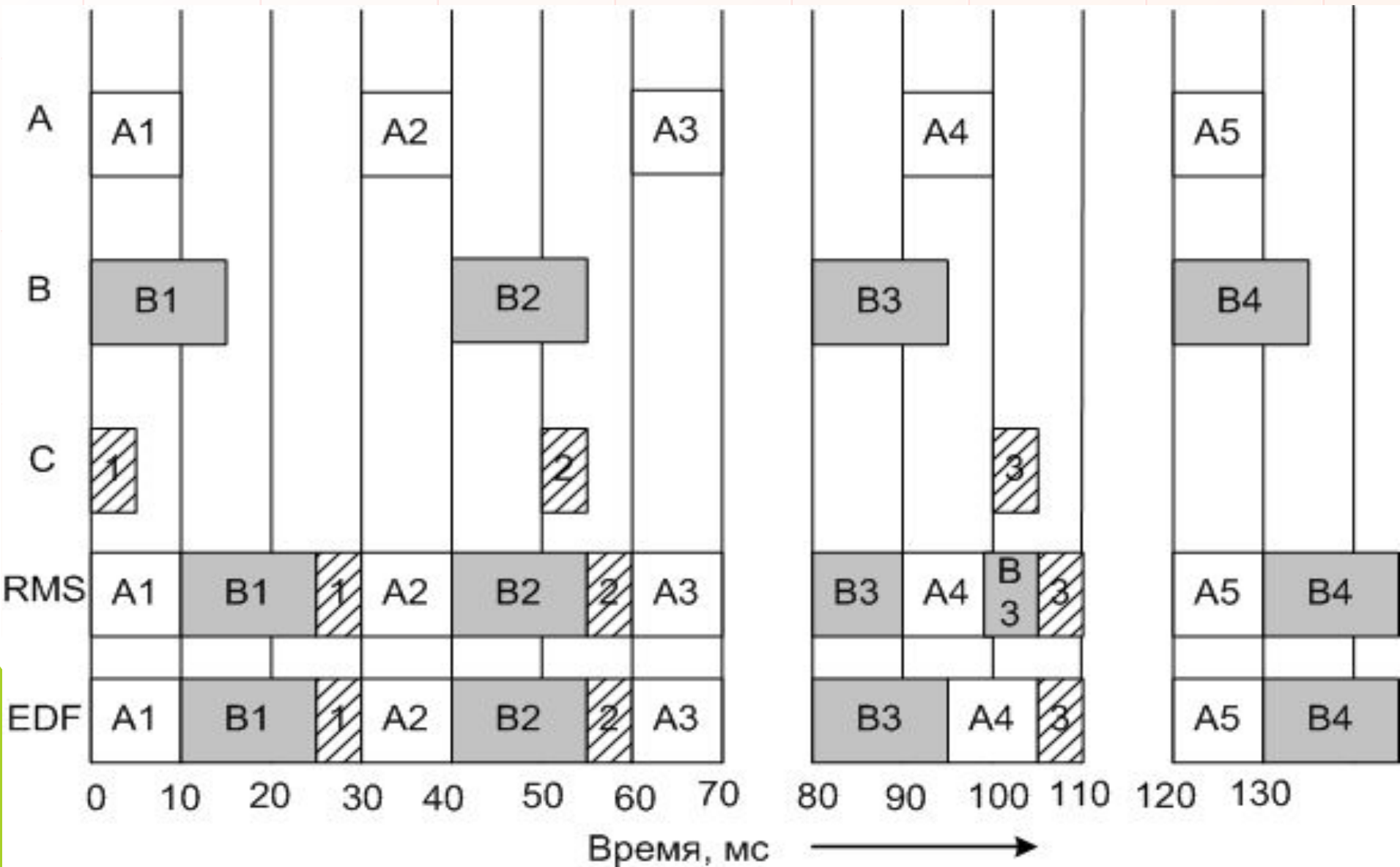
Процесс	A	B	C
Время T	15	15	5
Период P (мс)	30	40	50
Частота (1/с)	33	25	20
Приоритет	33	25	20

$$15/30 + 15/40 + 5/50 = 0.975 < 1$$

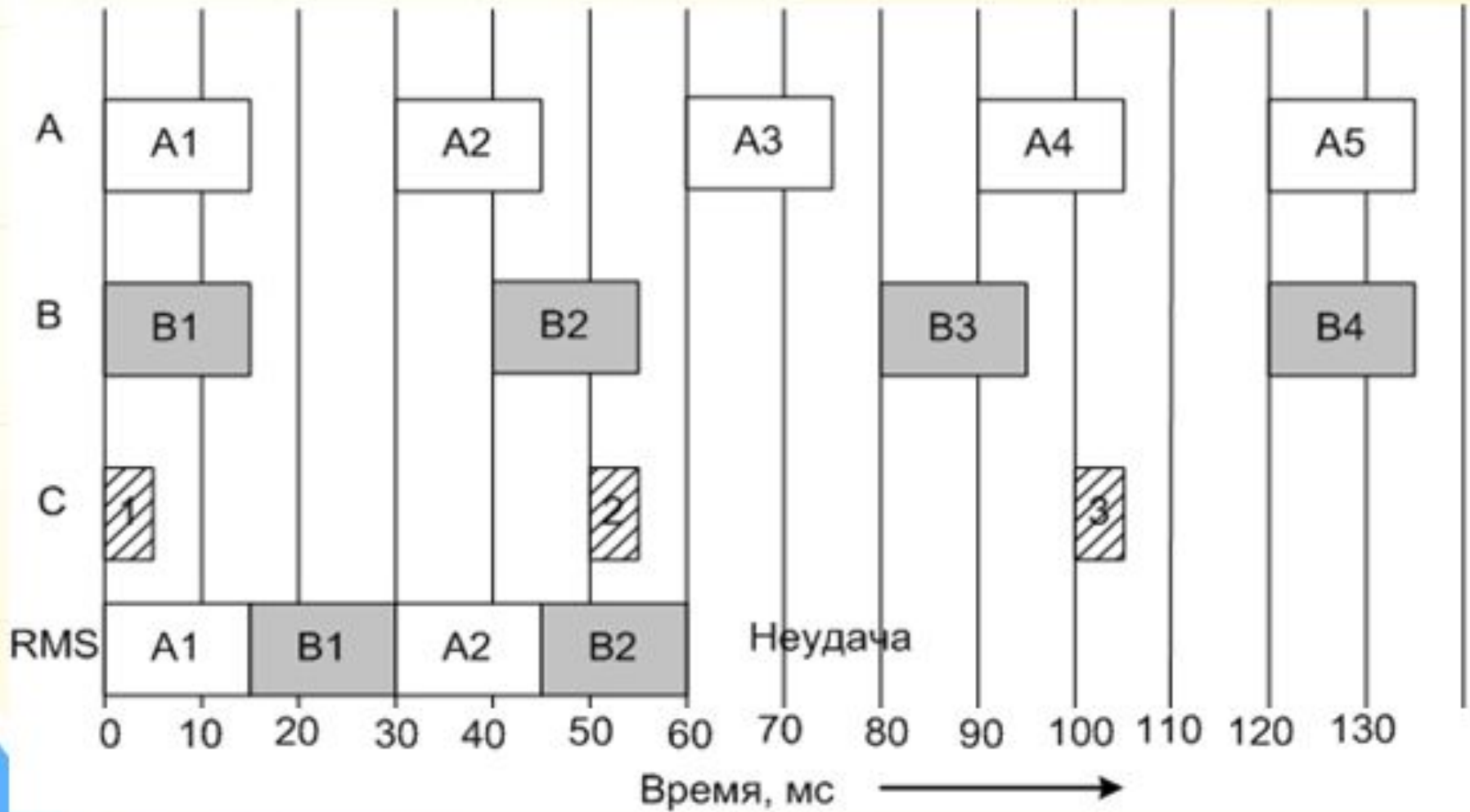
RMS – Пример 1



Сравнение RMS и EDF- Пример 1



RMS - Пример 2



Сравнение RMS и EDF- Пример 2

