

## ОСНОВНЫЕ ПОНЯТИЯ [IEEE Std. 610.12-1990]

**ОС** - collection SW, FirmWare and HW элементов, которые управляют исполнением компьютерных программ и обеспечивают распределение компьютерных ресурсов.

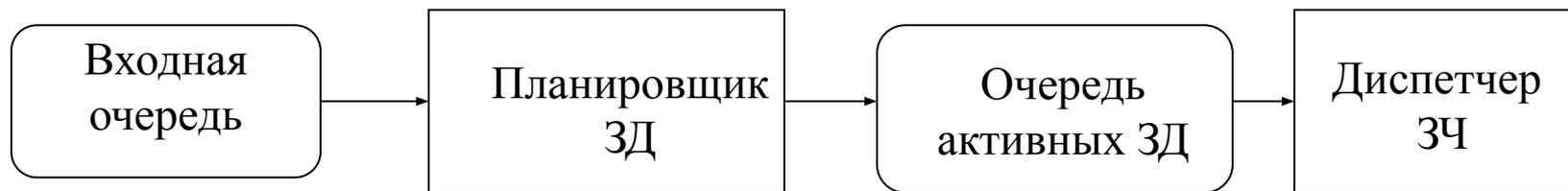
**ПО ОС ВС**, представляет собой комплекс управляющих и обслуживающих программ, обеспечивающих максимальную эффективность системы за счет автоматического управления вычислительными ПРС и ресурсами при обработке потока задач

**task, задача, ЗЧ** - основной объект выполнения работ в ВС, требующий ресурсов: памяти (ОП и ВнП) для размещения ЗЧ, устройств обмена для загрузки/разгрузки ЗЧ, времени ПРС для выполнения операций обработки ЗЧ

**job, задание, ЗД** - единица работ ВС, представленных в виде ЗЧ или последовательности ЗЧ

**job batch, пакет заданий** - последовательность ЗД, вводимых через устройства ввода

Задача ОС - обеспечить минимизацию общего времени обработки потока ЗД.



# ВИРТУАЛЬНАЯ ПАМЯТЬ

**Задание (ЗД)** является источником **Набора Задач (ЗЧ)** или шагов ЗД (ШЗД)

Каждая ЗЧ выполняется, используя ресурсы ВВП и ОП. Поскольку объекты ЗЧ размещены во ВВП, а исполняются в ОП, то Монитор управления памятью действует так, чтобы в процессе выполнения необходимая для реализации вычислительного процесса информация подкачивалась из ВВП в ОП, а полученные результаты по мере окончания их активного использования откачивались из ОП во ВВП.

Таким образом ОП – это окно, на которое проектируется информация ВВП, необходимая для реализации вычислительного процесса.

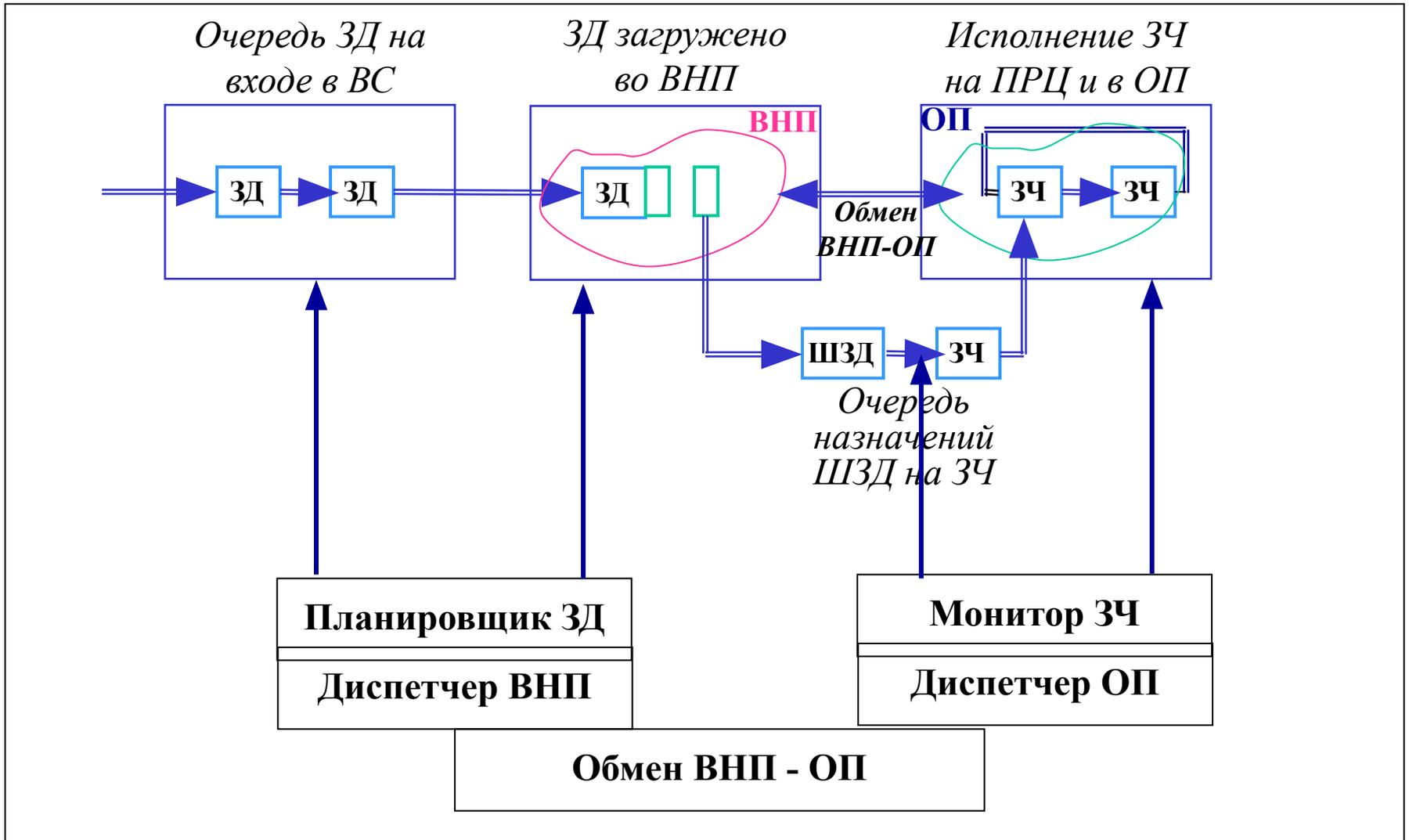
# ВИРТУАЛЬНАЯ ПАМЯТЬ

**Монитор ВИП** – это совокупность Диспетчеров ОП и ВМП и утилит обмена ОП и ВМП. Запросы к Монитору поступают от Монитора ЗЧ и Планировщика ЗД.

Следуя приведенному алгоритму – Монитор ВИП должен различать памяти разных устройств (здесь ОП и ВМП) для того чтобы в пользовательских приложениях (ЗД, ЗЧ) этого уже не надо было делать.

То-есть модель ВИП – это модель пользователя, но не разработчика Монитора управления памятью операционной системы.

# ВИРТУАЛЬНАЯ ПАМЯТЬ

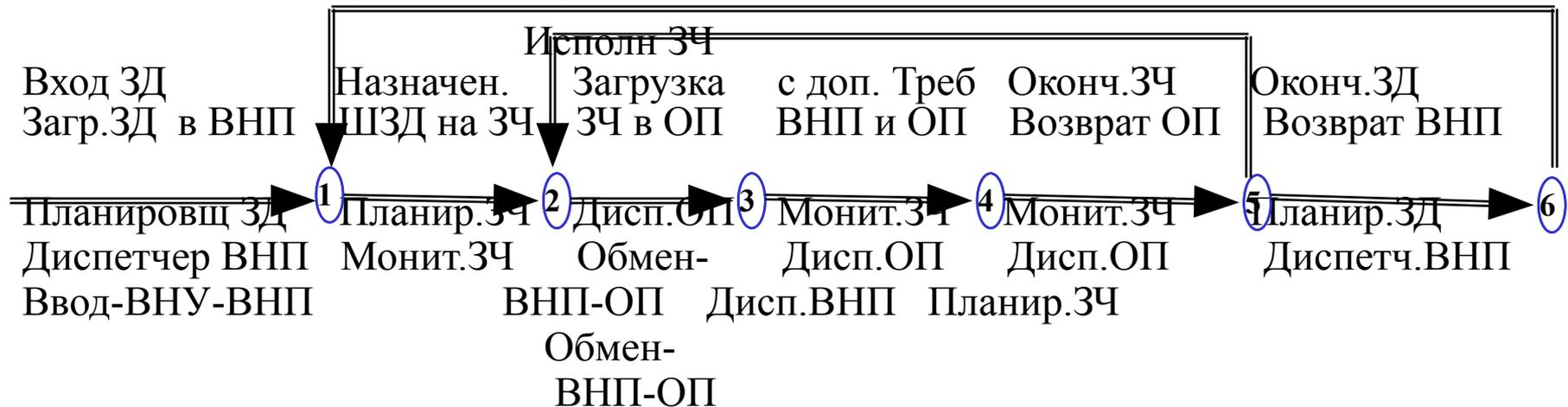


## Управление виртуальной памятью в ОС

# ВИРТУАЛЬНАЯ ПАМЯТЬ

## Задачи Монитора Управления ВИП:

- Планирование ВИП при загрузке ЗД в ВС
- Распределение ВИП между ЗД при поступлении заявок на дополнительную ВВП или на возврат ВВП
- Распределение ВИП между ЗЧ при поступлении заявок на дополнительную ОП или на возврат ОП
- Перераспределение занятой и свободной памяти в целях уничтожения малопригодных для использования областей памяти (сборка мусора)



# МЕТОДЫ ДИСПЕТЧЕРИЗАЦИИ В МУЛЬТИПРОГРАММНОЙ СИСТЕМЕ

Алгоритм (дисциплина) диспетчеризации задач в ОС определяет способ выбора ВПРС из очереди активных и определяет для ВПРС величину кванта.



**Бесприоритетные дисциплины** определяют алгоритмы упорядочения заявок (постановки в очередь) и алгоритмы выбора заявки из очереди.

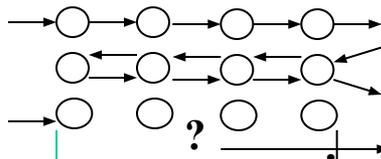
**FIFO** – First In First Out

**LIFO** - Last In First Out

**RND** - Random

**SJN** - Shortest Job Next (упорядочение по **min** времени обслуживания)

**SRN** - Shortest Residuary time Next (упорядочение по **min** времени дообслуживания)



С ростом пропускной способности  $t_{\text{ожидания}}$  падает



С ростом загрузки системы  $t_{\text{ожидания}}$  растет

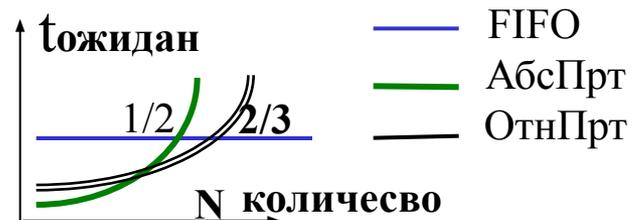
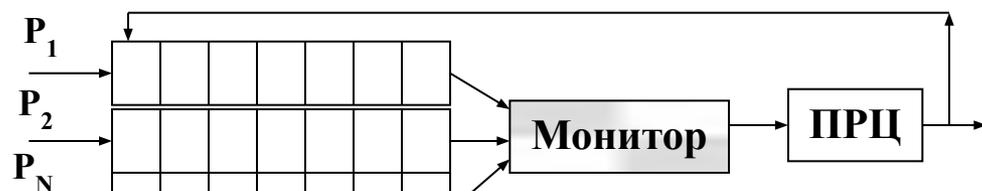
Бесприоритетные дисциплины позволяют регулировать производительность системы, либо путем подключения больших ресурсов ПРЦ к обслуживанию ВПРС, либо путем регулирования нагрузки на процессор

# МЕТОДЫ ПЛАНИРОВАНИЯ В МУЛЬТИПРОГРАММНОЙ СИСТЕМЕ

**Приоритетные дисциплины** характеризуются тем, что ВПРС ставится в соответствие приоритет – число, обозначающее вес (привилегии) ВПРС. При вычислении приоритета учитывают вес ВПРС, занятые ВПРС ресурсы, и т.п. Привилегированность нужна важным ВПРС и дефицитным ресурсам

**Относительные приоритеты** – когда ресурс (ПРЦ) отдается очередному ВПРС и не возвращается даже если в очереди появился более приоритетный ВПРС.

**Абсолютные приоритеты** - очередной ВПРС отдает ресурс (ПРЦ) при появлении в очереди ПРС с более высоким приоритетом. Прерванный процесс впоследствии дообслуживается в соответствии с выбранной дисциплиной (сразу, на общих основаниях, с изменением приоритета)



Относительные приоритеты действуют мягче, они дают преимущество высоким приоритетам, но при этом не сильно замедляют менее приоритетные ВПРС. Абсолютные сильно ускоряют, но и сильно замедляют равно приоритетные ВПРС.

## Методы обслуживания вычислительных процессов

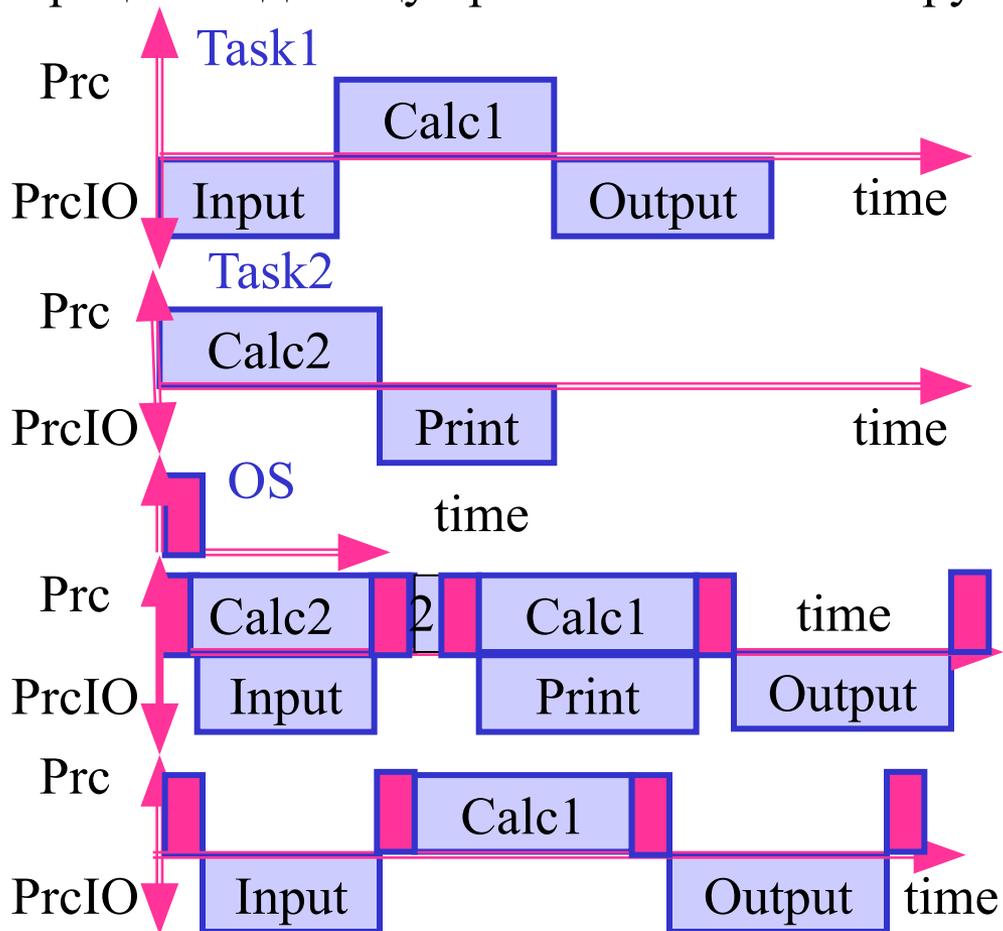
**Пакетная обработка (ПО)** не регулирует очередность ВПРС по какому-либо признаку, т.к. в среднем это означает выполнение одинакового количества операций в единицу времени. Максимизируется загрузка всех ПРЦ-ров.

**Разделение времени (РВ)** ускоряет короткие ВПРС, т.е. делает их приоритетными за счет большего ожидания обслуживания у больших заданий.

**Квантование – метод РВ**, обеспечивающий предоставление  $N$  ВПРС в  $N$  раз более медленных ВМ.

# МЕТОДЫ ДИСПЕТЧЕРИЗАЦИИ В МУЛЬТИПРОГРАММНОЙ СИСТЕМЕ

**Пакетная обработка (ПО)** не регулирует очередность ПРС по какому-либо признаку, т.к. в среднем это означает выполнение одинакового количества операций в единицу времени. Максимизируется загрузка ПРС-ров



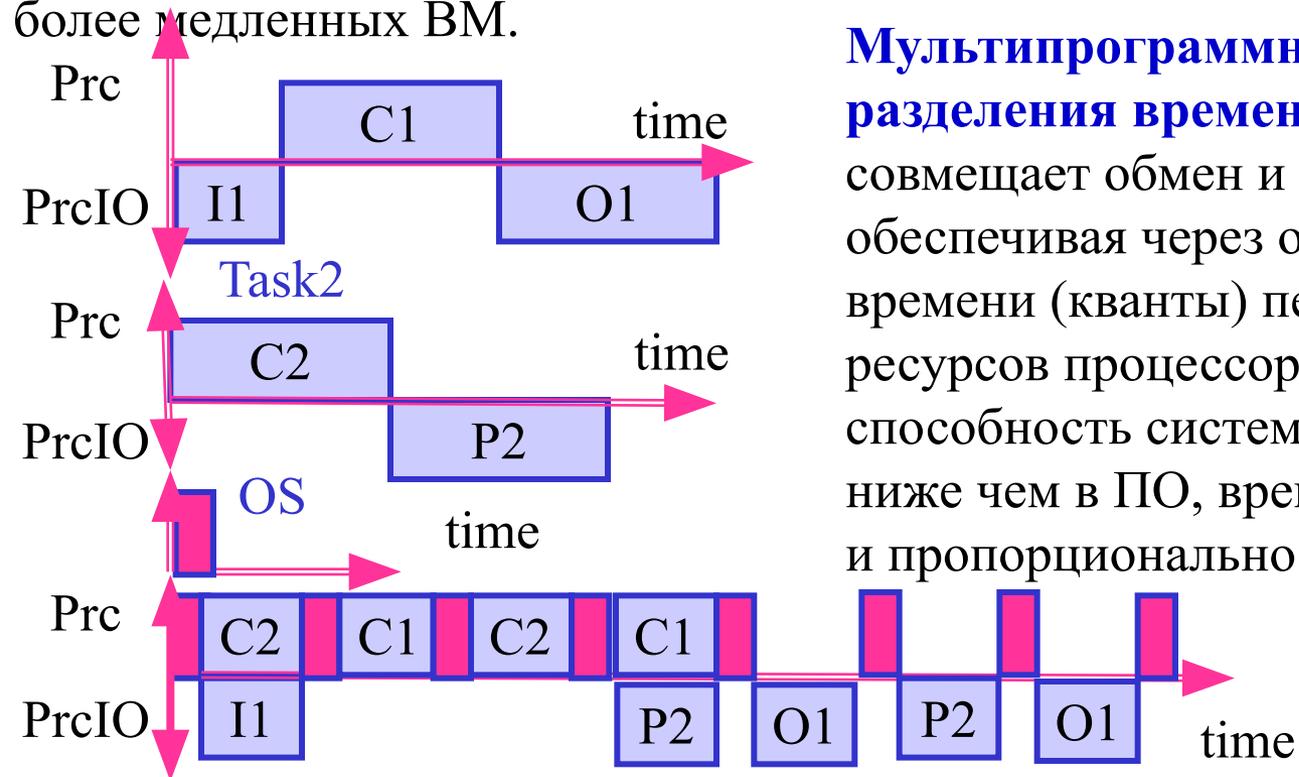
**Мультипрограммная работа – режим ПО** – совмещает обмен и счет на PrcIO и Prc. Пропускная способность системы велика, но и время ожидания ответа увеличивается в сравнении с однопрограммным режимом

Однопрограммная работа дает низкую пропускную способность и малое время ожидания ответа для обслуживаемого ВПРС

# МЕТОДЫ ДИСПЕТЧЕРИЗАЦИИ В МУЛЬТИПРОГРАММНОЙ СИСТЕМЕ

**Разделение времени (РВ)** ускоряет короткие ПРС, т.е. делает их приоритетными за счет большего ожидания обслуживания у больших заданий. **Классическое РВ** - мультипрограммная работа с приоритетом коротких задач.

**Квантование – метод РВ**, обеспечивающий предоставление N ПРС в N раз более медленных ВМ.



**Мультипрограммная работа – режим разделения времени - квантование** –

совмещает обмен и счет на PrcIO и Prc, обеспечивая через определенные интервалы времени (кванты) перераспределение ресурсов процессоров. Пропускная способность системы достаточно велика, но ниже чем в ПО, время ожидания ответа мало и пропорционально величине Задания

## МЕТОДЫ ДИСПЕТЧЕРИЗАЦИИ В МУЛЬТИПРОГРАММНОЙ СИСТЕМЕ

Если квант  $q \rightarrow \infty$ , то квантование переходит в бесприоритетную дисциплину FIFO. Если квант  $q \rightarrow 0$ , то квантование переходит в дисциплину, где  $t_{\text{пробывания}} \sim$  длине ПРГ в ВПРС

Уменьшение кванта  $q$  означает увеличение издержек переключения, поэтому квант выбирают так:

- величина  $q$  должна быть пропорциональна  $\sim$  времени выполнения среднего задания (ВПРС)
- время выполнения ЗЧ в кванте д.б.  $\gg$  издержек переключения с ПРГ на ПРГ

### Квантование с изменяющимся квантом $q$

**Алгоритм Корбато** –  $q_i = q_0 * 2^i$ , т.е. после круга в очереди на обслуживание с приоритетом  $i$ , ВПРС переходит в очередь  $i+1$  приоритета, где получает квант в 2 раза большей величины.

Алгоритм Корбато – **алгоритм очередей с обратными связями**, где ВПРС обслуживаются с такими квантами, которые им более всего подходят. Он дает преимущество коротким Зч, т.е. делает их приоритетными за счет большего ожидания обслуживания больших Зч