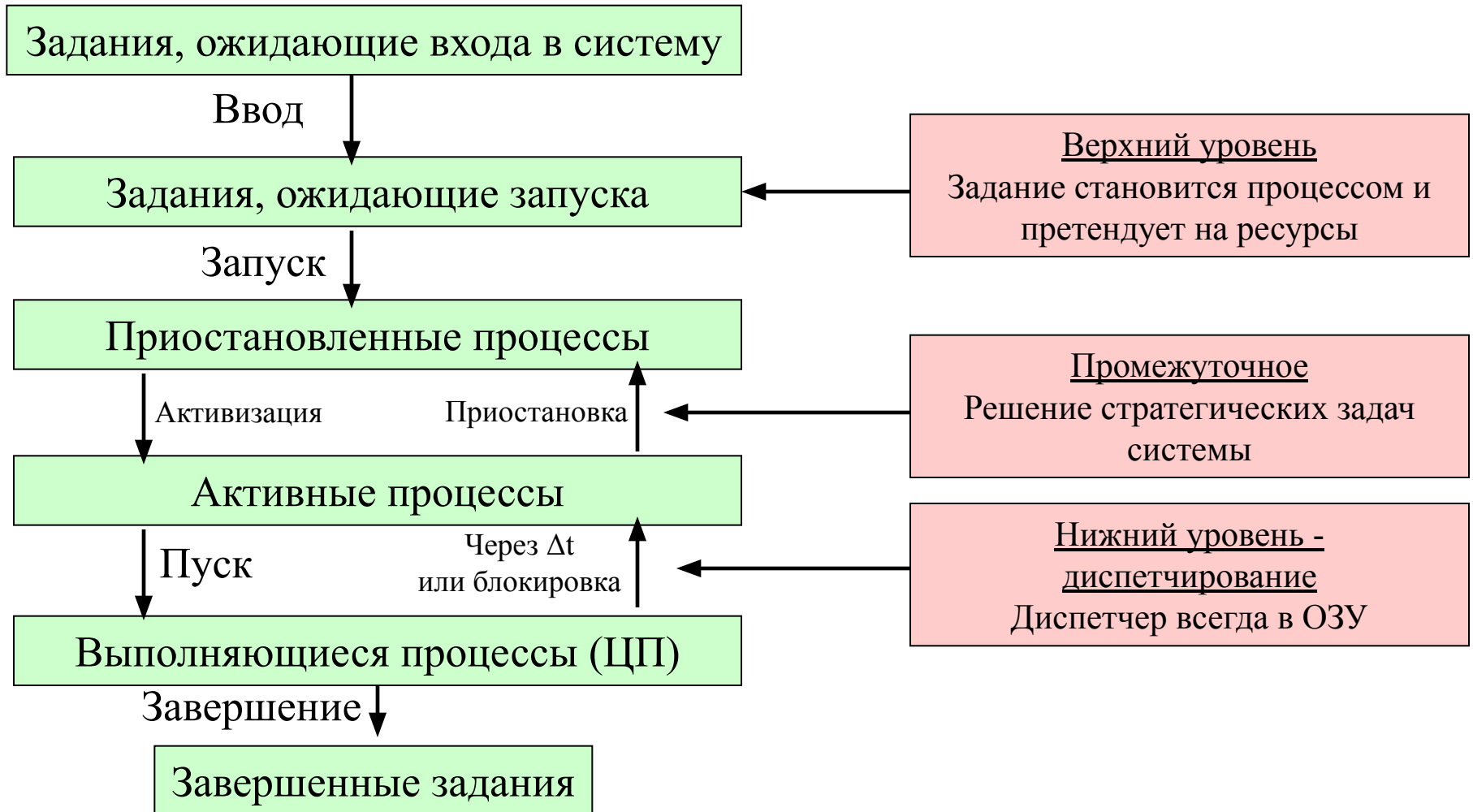


# Управление процессорами

**Планирование загрузки процессоров** – это распределение процессоров (или процессорного времени) между процессами.

## Уровни планирования



# Дисциплина планирования должна преследовать следующие цели:

Быть справедливой ко всем процессам

Повышать производительность системы (число процессов/время)

Уменьшать время реакции системы для пользователей

Быть предсказуемой (время решения задания не должно зависеть от нагрузки)

Минимизировать потери ресурсов

Загружать простаивающие ресурсы

Исключать бесконечное откладывание

Учитывать приоритеты

Выделять процессы, занимающие ключевые ресурсы (не прерывать)

Создавать хорошие условия для «правильных» процессов

Балансировать между  $\min(\text{время реакции})$  –  $\max(\text{загрузка ресурсов})$

Иметь плавную зависимость параметров от нагрузки

# Дисциплина планирования должна учитывать: (критерии планирования)

Лимитируется ли процесс I/O операциями

Освобождает ли процесс ЦП до окончания кванта времени

Режим работы процесса – пакетный или интерактивный

Необходимость «немедленной» реакции на события (real time)

Приоритет процесса

Частоту прерываний из-за отсутствия страниц в памяти (раб. множество)

Частоту прерываний из-за низкого приоритета

Выделенное время

Время ожидания в очереди

Время, необходимое для завершения процесса

И многое другое

## Дисциплины планирования делятся на:

### **ДИСЦИПЛИНЫ БЕЗ ПЕРЕКЛЮЧЕНИЯ**

После выделения процессу ЦП его нельзя отобрать до завершения

### **ДИСЦИПЛИНЫ С ПЕРЕКЛЮЧЕНИЕМ**

У процесса можно отобрать ЦП  
используя таймер или с приходом другого процесса

---

## Приоритеты бывают:

**СТАТИЧЕСКИЕ** – не меняются при работе процесса  
Жесткий алгоритм управления, низкая эффективность

**ДИНАМИЧЕСКИЕ** – могут учитывать показания системных «сенсоров»  
Гибкое управление, высокие издержки на обслуживание

**ПОКУПНЫЕ** – изменяется (повышается) по инициативе пользователя  
Экзотика.

# 1. Планирование по сроку завершения

Все процессы должны закончиться к указанному сроку.

## ТРУДНОСТИ:

Надо точно указать какие ресурсы нужны процессу (как узнать?).

Нельзя дискриминировать отдельные процессы!

Как учесть приход различных запросов во время выполнения процесса?

Если процессов с «жесткими» сроками завершения несколько, как выбрать?

Стратегия требует активного управления, а это дорого.

На практике редко используется, только для «закрытых» систем.

## 2. Планирование по принципу FIFO (First Input First Output)

Из очереди выбирается тот процесс, который раньше пришел в систему.  
БЕЗ ПЕРЕКЛЮЧЕНИЯ.

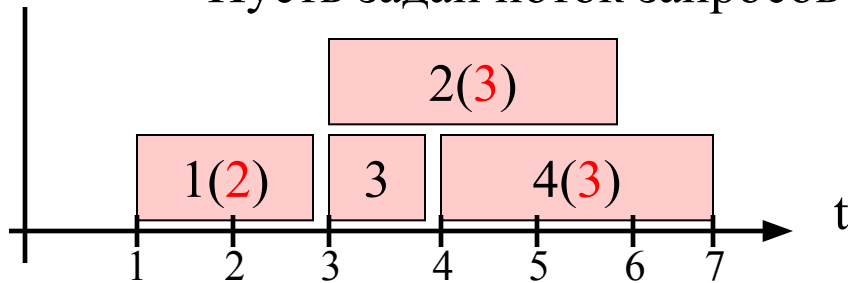
ОСОБЕННОСТИ:

Простота  
реализации.

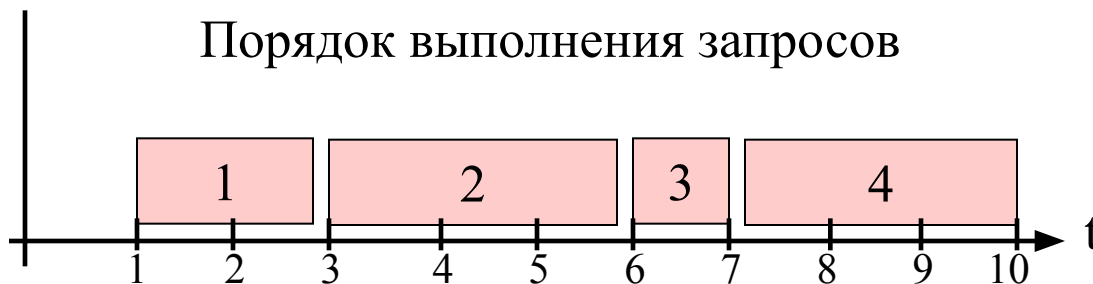
Длинные процессы блокируют ЦП.

Нельзя использовать в интерактивных системах

Пусть задан поток запросов



Порядок выполнения запросов



Pr	$t_{\text{real}} = t_{\text{out}} - t_{\text{inp}}$		$\Delta = t_{\text{real}} - t_{\text{exe}}$		$\Delta$
	$t_{\text{inp}}$	$t_{\text{exe}}$	$t_{\text{out}}$	$t_{\text{real}}$	
1	1	2	3	2	0
2	3	3	6	3	0
3	3	1	7	4	3
4	4	3	10	6	3

### 3. Планирование по принципу SJF (Shortest Job First)

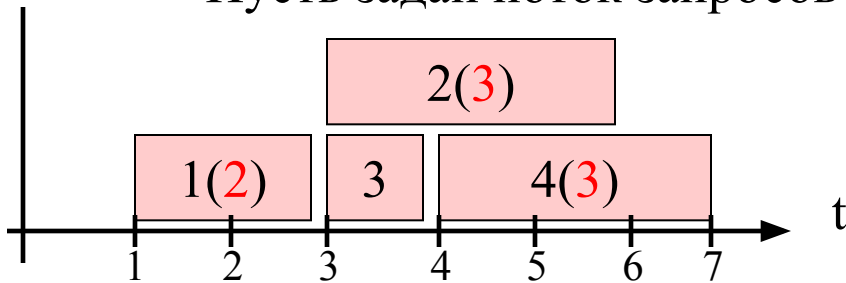
Из очереди выбирается процесс с наименьшим временем выполнения.  
БЕЗ ПЕРЕКЛЮЧЕНИЯ.

ОСОБЕННОСТИ:

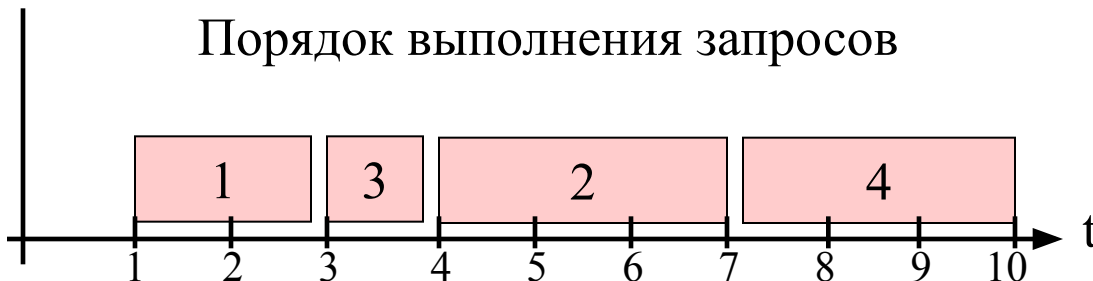
Снижает длину очереди.

Как оценить время выполнения?

Пусть задан поток запросов



Порядок выполнения запросов



Pr	$t_{\text{inp}}$	$t_{\text{exe}}$	$t_{\text{out}}$	$t_{\text{real}}$	$\Delta$
1	1	2	3	2	0
2	3	3	7	4	1
3	3	1	4	1	0
4	4	3	10	6	3

# 4. Планирование по принципу SRTF (Shortest Remaining Time First)

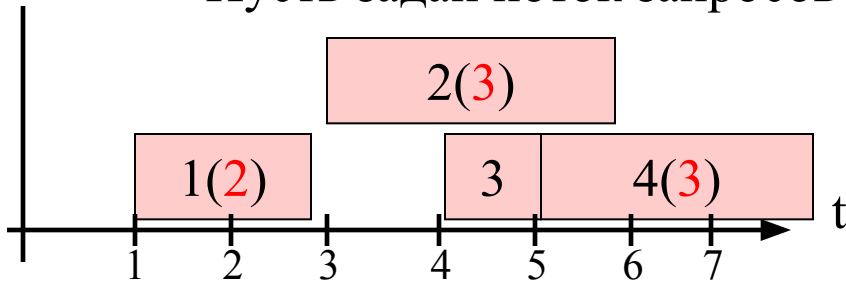
Из очереди выбирается процесс с наименьшим временем завершения.  
С ПЕРЕКЛЮЧЕНИЕМ.

ОСОБЕННОСТИ:

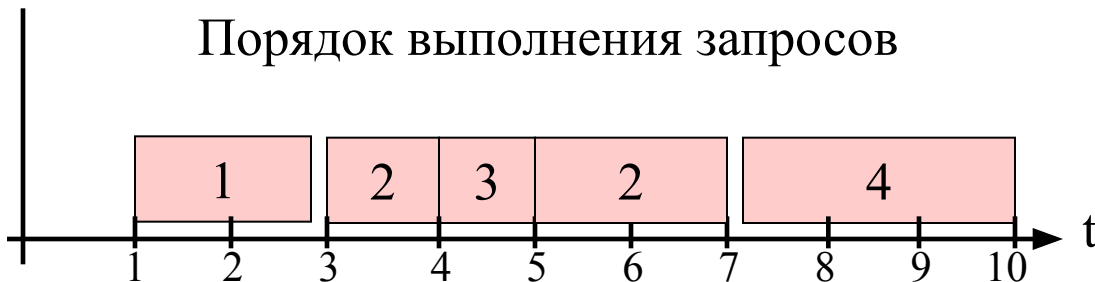
Минимальное время ожидания.

Как оценить время выполнения?

Пусть задан поток запросов



Порядок выполнения запросов



Pr	$t_{inp}$	$t_{exe}$	$t_{out}$	$t_{real}$	$\Delta$
1	1	2	3	2	0
2	3	3	7	4	1
3	4	1	5	1	0
4	5	3	10	5	2



# 5. Циклическое планирование (RR) (Round Robin)

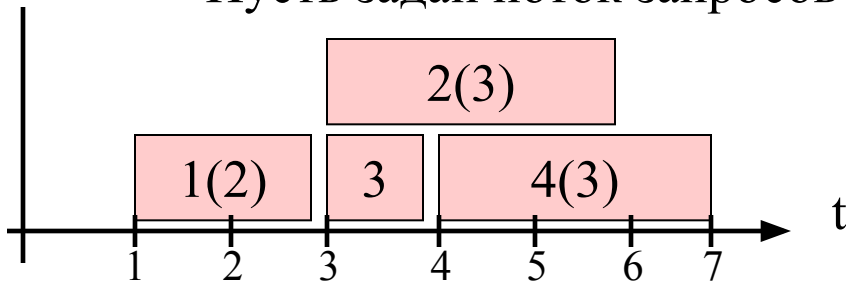
Каждый квант времени из очереди выбирается очередной процесс.  
Работавший процесс становится последним в очереди (цикл).  
С ПЕРЕКЛЮЧЕНИЕМ.

ОСОБЕННОСТИ:

Для интерактивных систем.

Любит ОЗУ.  
Размер кванта?

Пусть задан поток запросов



Порядок выполнения запросов



Pr	$t_{inp}$	$t_{exe}$	$t_{out}$	$t_{real}$	$\Delta$
1	1	2	3	2	0
2	3	3	9	6	3
3	3	1	5	2	1
4	4	3	10	6	3

## 6. Планирование по принципу HRN (Highest Response ratio Next)

Каждый квант времени из очереди выбирается процесс с наибольшим приоритетом.  
С ПЕРЕКЛЮЧЕНИЕМ.

ОСОБЕННОСТИ:

Приоритет процесса – динамический.

$\text{ПРИОРИТЕТ} = (\text{время ожидания} + \text{время обслуживания}) / (\text{время обслуживания})$

Справедливая стратегия

Размер кванта?

# 7. Многоуровневые очереди с обратными связями

Отношение системы к процессу зависит от его поведения.  
Адаптивная стратегия. Требует значительных ресурсов системы.  
Самая «правильная».  
С ПЕРЕКЛЮЧЕНИЕМ.

