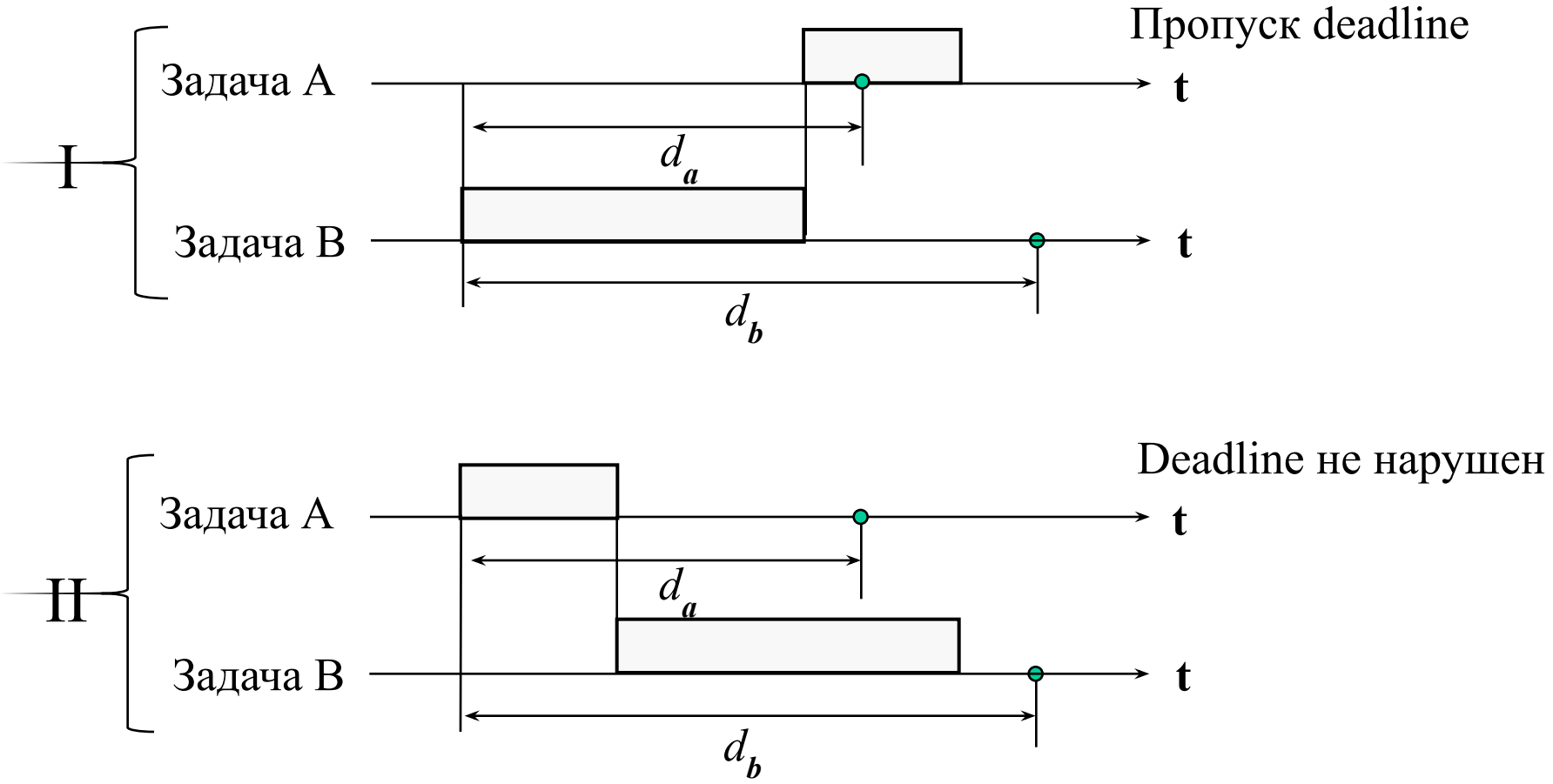
9. Планирование задач (scheduling)

Пример



Планирование задач, определения

Алгоритм планирования (scheduling algorithm) - правило, определяющее порядок выполнения задач

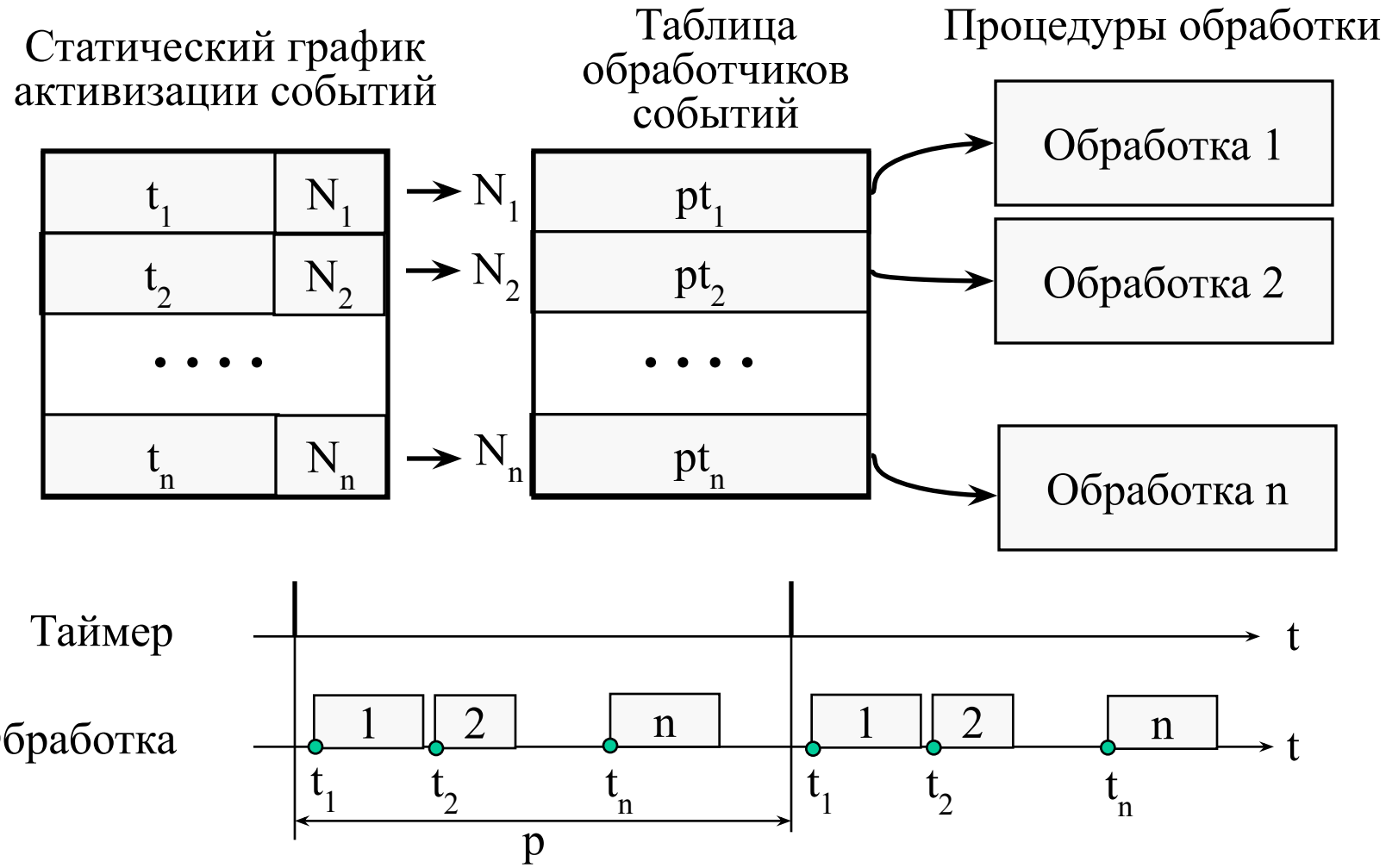
Статическое планирование (off-line) – условия активизации задач определяются до начала работы системы

Динамическое планирование (on-line) – условия активизации определяются в процессе работы системы на основе текущих условий

Приоритетное планирование (статическое /динамическое) – в качестве параметра, определяющего порядок активизации задач, выступает приоритет задачи

Предсказуемость (predictability) - для всех действий определимы временные границы их завершения

Циклический исполнитель (Round Robin)



Циклический исполнитель (2)

Достоинство – простота реализации

Недостаток – при большом количестве задач «off-line» планирование может вызвать трудности

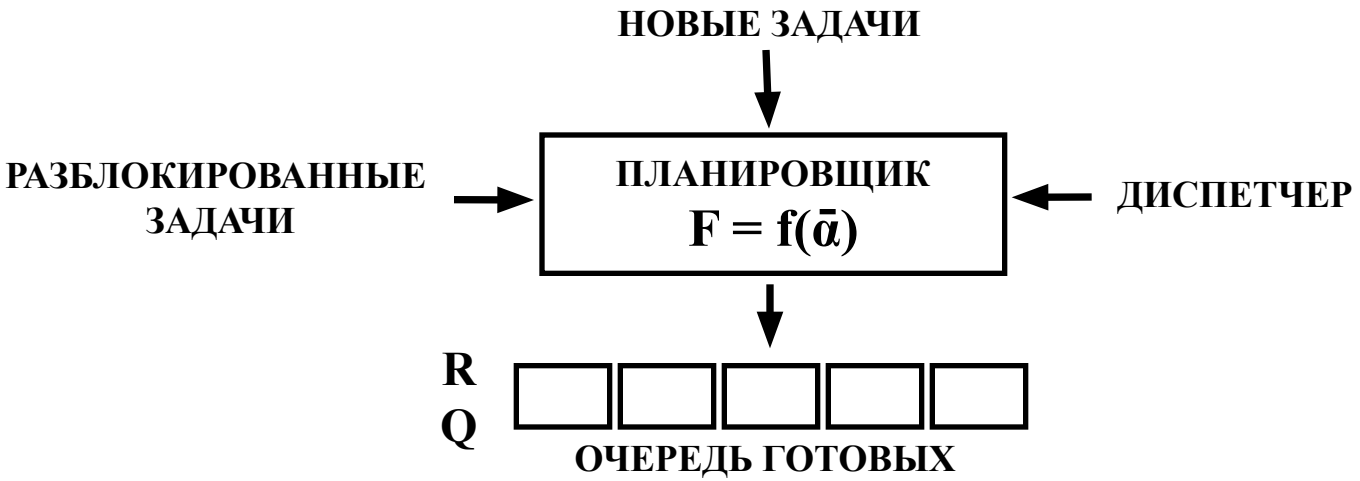
Проблема – планирование обработки асинхронных событий

- Возможное решение – каждому асинхронному событию *i* ставится в соответствие момент времени t_i, индекс N_i и *Обработчик i*. Если событие *i* происходит раньше момента t_i, то запоминается факт его возникновения, а обработка активизируется в момент t_i; если позже, то активизация осуществляется на следующем цикле
- Недостаток решения – большие *latency*

Приоритетное планирование

Статические алгоритмы – приоритеты задач определяются на этапе проектирования (off-line) и не изменяются в процессе работы системы

Динамические алгоритмы – приоритеты задач изменяются в процессе работы (on-line) в зависимости от текущих условий



F – приоритет задачи, α - параметры задачи, состояние системы

Rate monotonic планирование

Rate monotonic (RM) – правило статического (off-line) назначения приоритетов:

Пусть имеется n независимых периодических задач реального времени (p_i - период i-й задачи, C_i – время выполнения в наихудшем случае), у каждой из которых которых значение *deadline* совпадает с периодом активизации p_i

Суммарная загрузка системы в этом случае определяется как

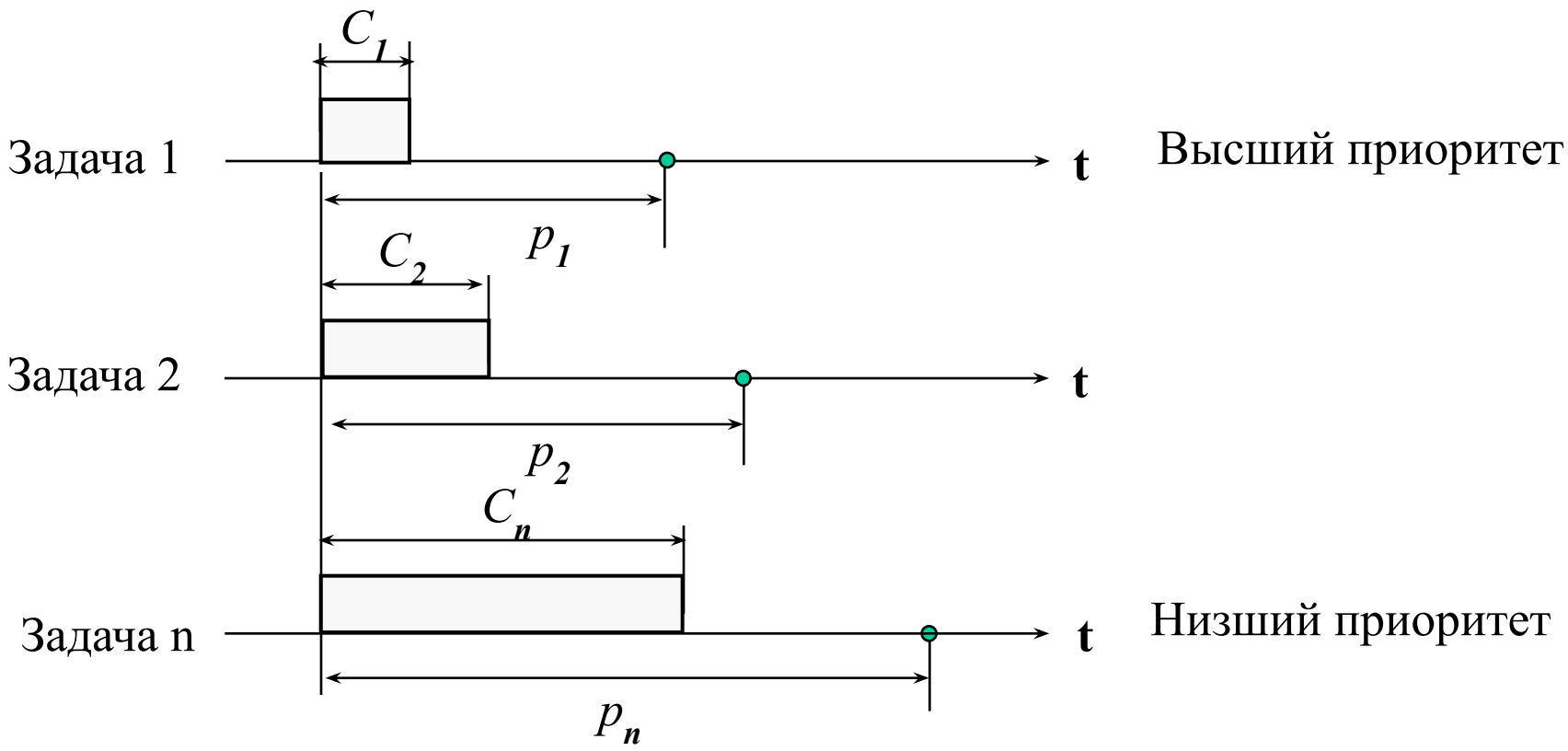
$$R = \sum_{I=1}^{n} C_i / p_i$$

Тогда, если приоритеты задач будут назначены обратно пропорционально их p_i, и будет выполнено условие

$$R <= n * (2^{1/n} - 1)$$

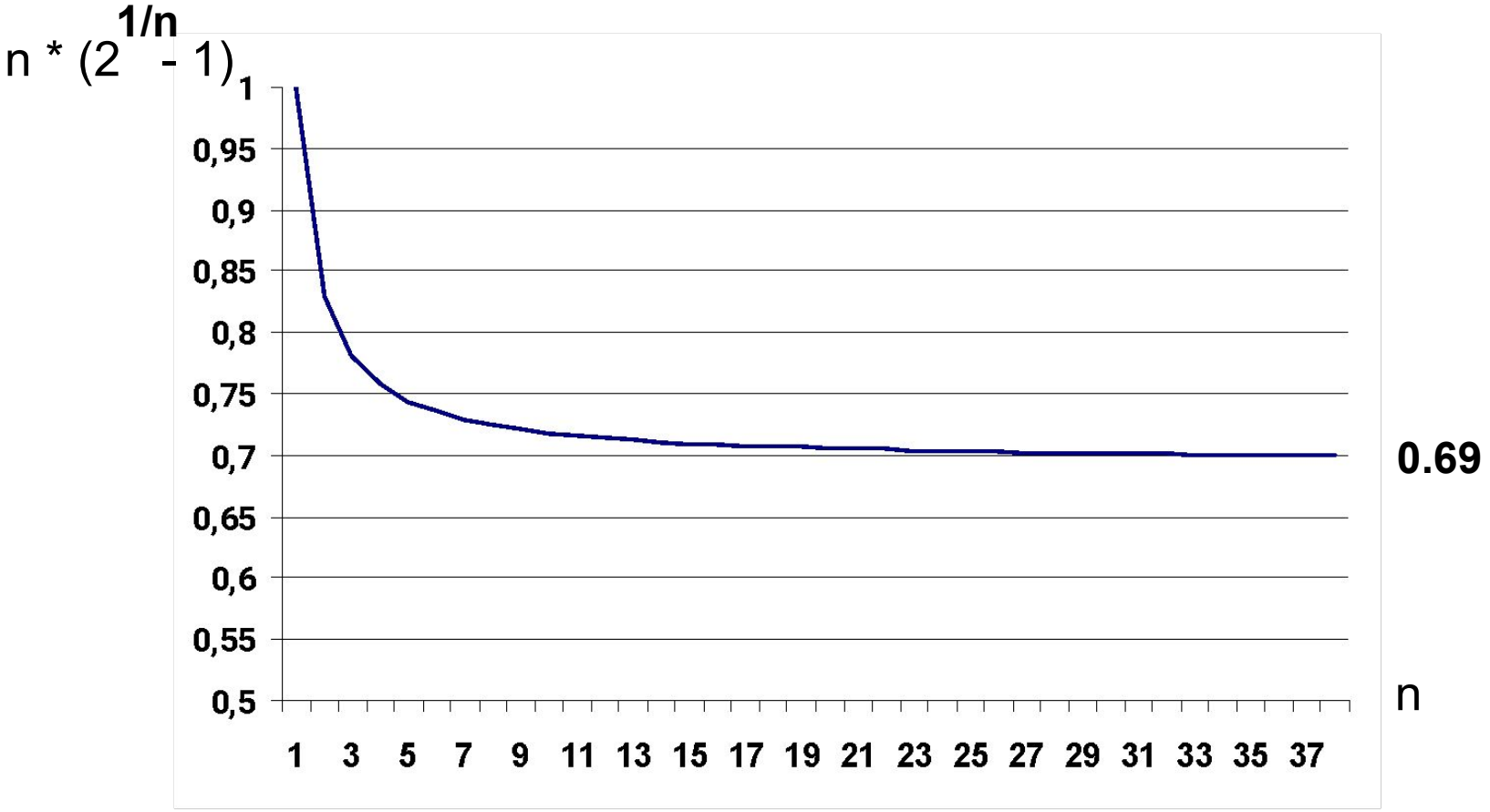
то выполнение задач будет проходить без нарушения их *deadline* (1973 год, Liu)

Rate monotonic планирование (2)



$$\sum_{I=1}^{I=n} C_i / p_i <= n * (2^{1/n} - 1)$$

Тест *Rate Monotonic*



Тест *Rate Monotonic* (2)

Набор из n периодических задач будет «шедулируемым» (выполнимым) если приоритеты назначены в соответствии с RM и выполняется условие:

$$R = \sum_{I=1}^{I=n} C_i / p_i <= 0.69 \qquad (*)$$

- Зависимость (*) представляет нижнюю границу значения R
- В случае, когда частоты активизации задач находятся в гармонической зависимости, R = 1

(*Гармоническая зависимость – частота активизации любой задачи приложения кратна частоте каждой задачи с меньшей частотой; например - 1Hz, 5Hz, 10Hz, 20Hz*)

- В большинстве практических случаев R < 0.88

Особенности *Rate Monotonic*

- Rate Monotonic называют «устойчивым» (stable) алгоритмом – подмножество задач, которое удовлетворяет тесту, всегда будет выполняться без нарушения deadline

- Rate Monotonic называют «оптимальным» алгоритмом – в том смысле, что если набор задач выполним при любой другой дисциплине планирования, то он выполним и при RM планировании

Приоритетное планирование, динамические алгоритмы

Earliest Deadline First (EDF) – значения приоритетов, которые назначаются задачам, обратно пропорциональны длительности временных интервалов до наступления deadline

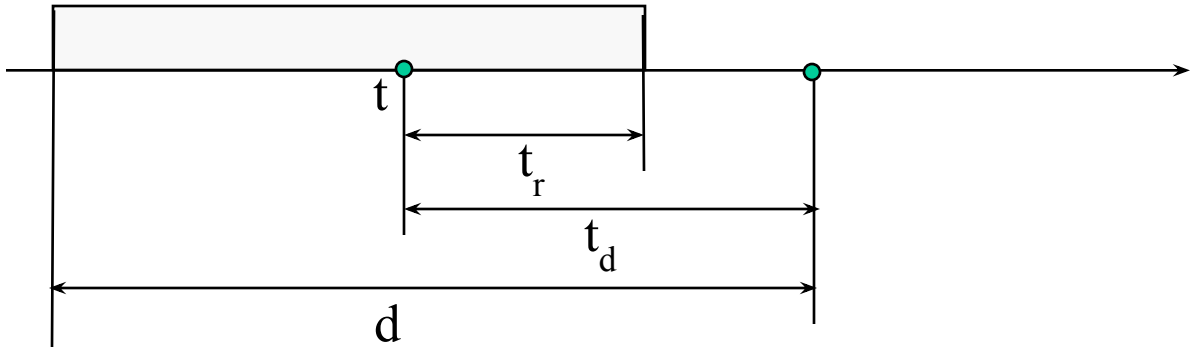
EDF scheduling test:

$$\sum_{I=1}^{I=n} C_i / p_i < 1$$

EDF - характеризуется наименьшим количеством переключения задач

Приоритетное планирование, динамические алгоритмы (2)

Least Laxity First (LLF) – (laxity – (t_d - t_r) - «расхлябанность») наивысший приоритет присваивается задаче с наименьшим laxity)



LLF scheduling test – аналогичен EDF

Признак laxity<0 - может быть использован для раннего обнаружения нарушения deadline (исключение)