



ПЛАНИРОВАНИЕ



Сезон
2019-2020



ПЛАНИРОВАНИЕ ЗАДАЧ

Проблема планирования

задач

В многозадачных системах задачи (процессы, потоки) вынуждены делить между собой процессорное время, где важно ещё и время получения результата, задачи могут «ждать», чтобы им позволили бы поделить процессорное время между задачам

плниться.



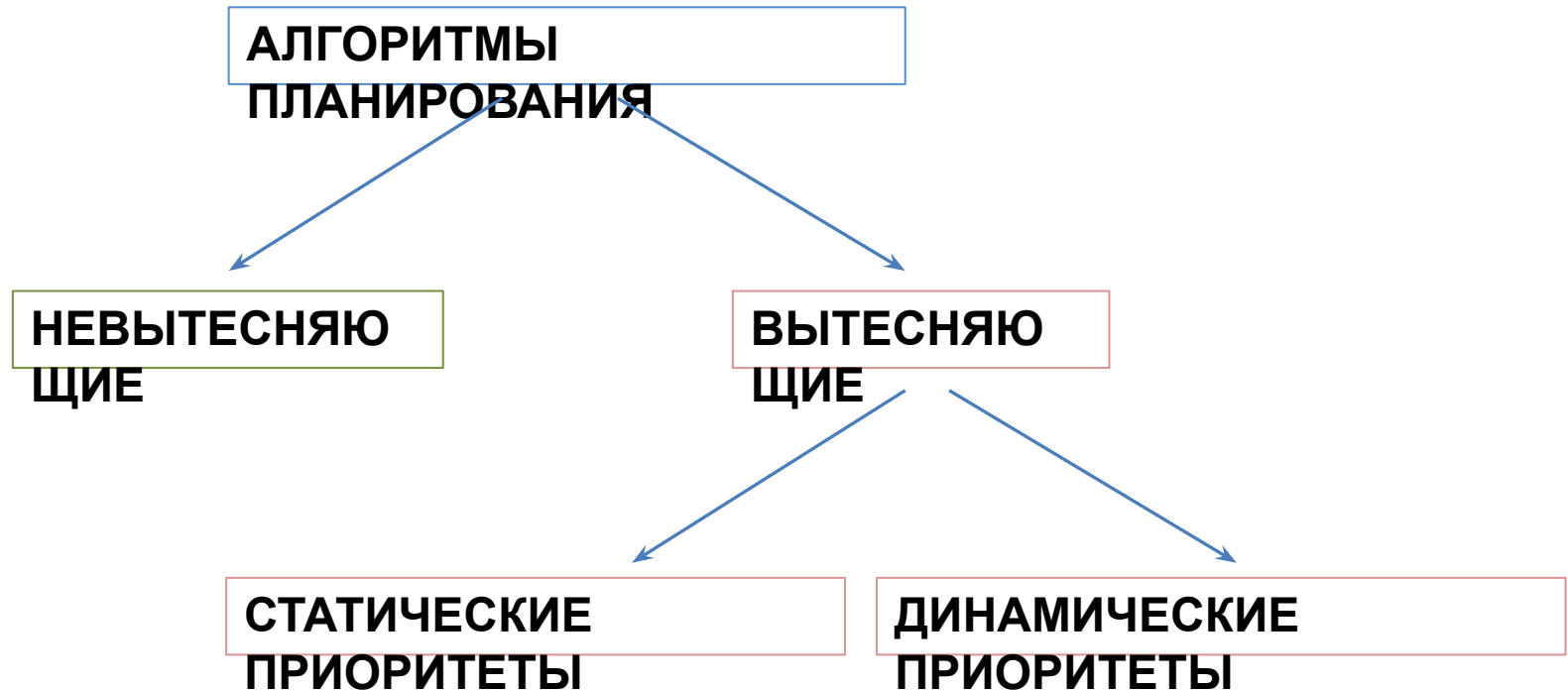


ПЛАНИРОВАНИЕ ЗАДАЧ

Виды алгоритмов

планирования

В зависимости от того, какой тип многозадачности реализуется в системе, алгоритмы планирования можно разделить на вытесняющие и невытесняющие.





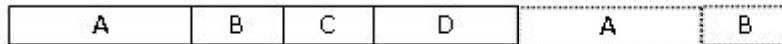
Невытесняющие алгоритмы

планирования

неприоритетные алгоритмы не находят применения в системах реального времени, так как не позволяют системе гибко реагировать на возникающие события. Более простым неприоритетным алгоритмом (без вытеснения)

является дисциплина очереди их поступления, выполнившаяся или новая задача

встаёт в конец очереди.



```
while(1)
{
    Task_A();
    Task_B();
    Task_C();
    Task_D();
}
```



ПЛАНИРОВАНИЕ ЗАДАЧ

Невытесняющие алгоритмы

планирования

- более совершенным вариантом является алгоритм «кратчайшая задача - первая»;
- Положим для нашего примера следующие длительности времени выполнения задач:

$$c_A = 10\text{мкс}, c_B = 4\text{мкс}, c_C = 4\text{мкс}, c_D = 7\text{мкс}$$

Циклическая

очередь:	B	C	D
----------	---	---	---

$$c_{\text{общ}A} = 10\text{мкс}$$

$$c_{\text{общ}B} = 14\text{мкс}$$

$$c_{\text{общ}C} = 18\text{мкс}$$

$$c_{\text{общ}D} = 25\text{мкс}$$

«Кратчайшая задача -

пе	B	C	D	A
----	---	---	---	---

$$c_{\text{общ}B} = 4\text{мкс}$$

$$c_{\text{общ}C} = 8\text{мкс}$$

$$c_{\text{общ}D} = 15\text{мкс}$$

$$c_{\text{общ}A} = 25\text{мкс}$$



ПЛАНИРОВАНИЕ ЗАДАЧ

Вытесняющие алгоритмы

планирования

наиболее популярным алгоритмом является алгоритм с разделением времени («time slice»). Алгоритм состоит в том, что каждой задаче отводится свой КВАНТ процессорного времени, по истечении которого активизируется планировщик. Алгоритмы широко применяются в ОС общего назначения и периодически передают задачу на другую очередь своей простоты.



ПЛАНИРОВАНИЕ ЗАДАЧ

Условие планируемости

системы

для планирования периодических задач существуют более эффективные

алгоритмы, учитывающие приоритеты, блокировку задач, активно

используемые в операционных системах, могут не работать при высокой загрузке

- если система перегружена, то она может не поддаваться планированию ни одним

известным алгоритмом.

Обозначим c_i время выполнения одной задачи в системе, а T_i её период. Тогда условие планируемости имеет следующий вид:

$$\sum_i \frac{c_i}{T_i} \leq 1$$



ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических задач:

RMS

RMS – rate-monotonic scheduling;

- приоритет задачи обратно пропорционален длительности
- ~~длительности~~ периода; каждой задачи должен совпадать с периодом
- ~~задачи должны~~ быть
- ~~задачи должны~~ быть
- ~~время выполнения~~ задачи на каждом периоде должно быть одинаково.

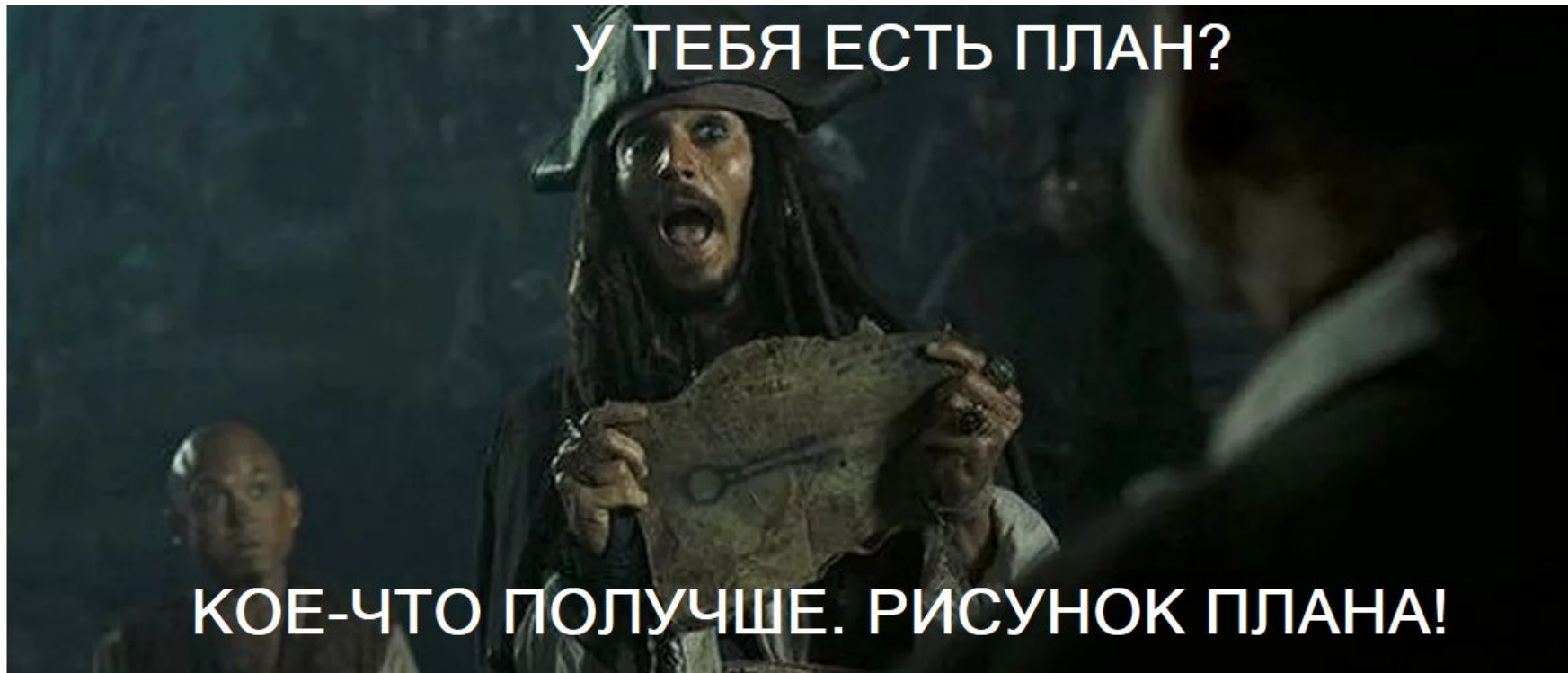


ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических задач:
RMS

У ТЕБЯ ЕСТЬ ПЛАН?

КОЕ-ЧТО ПОЛУЧШЕ. РИСУНОК ПЛАНА!





ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических задач:

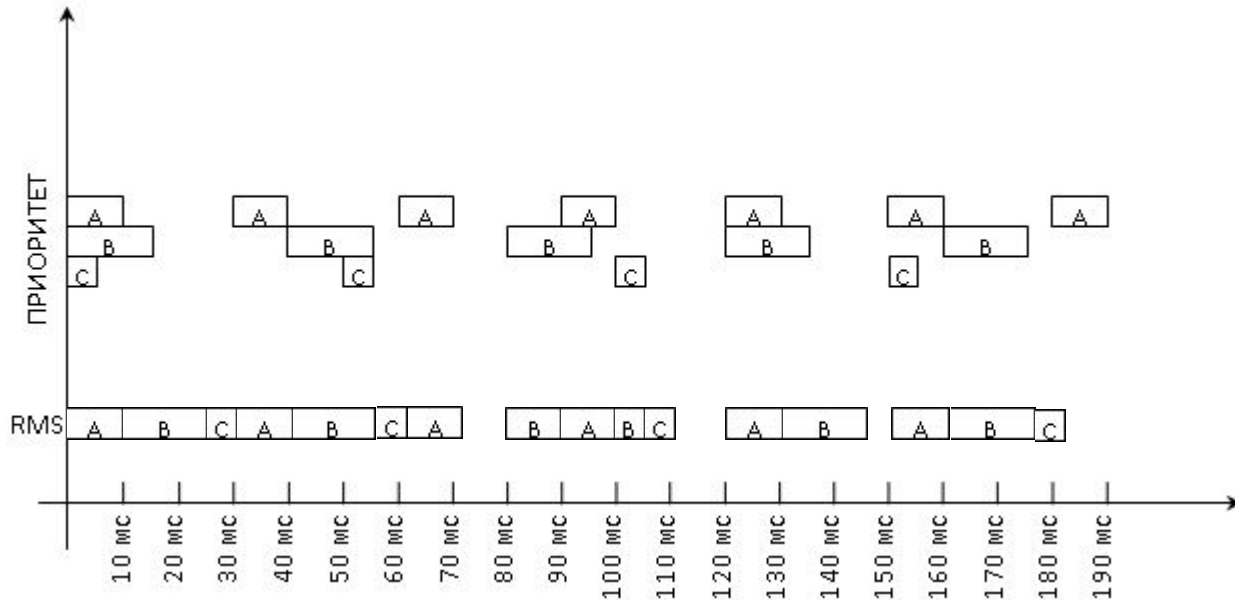
RMS. Рассмотрим систему с тремя периодическими задачами А, В и С (и относительно слабым процессором 😊).

$$c_A = 10 \text{ мкс}, T_A = 30 \text{ мкс}$$

$$c_B = 15 \text{ мкс}, T_B = 40 \text{ мкс}$$

$$c_C = 5 \text{ мкс}, T_C = 50 \text{ мкс}$$

$$U = \frac{10}{30} + \frac{15}{40} + \frac{5}{50} = 0.8083(3) < 1$$



ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических задач:

RMS

При высоких нагрузках на систему RMS начинает «фейлить» и не может нормально

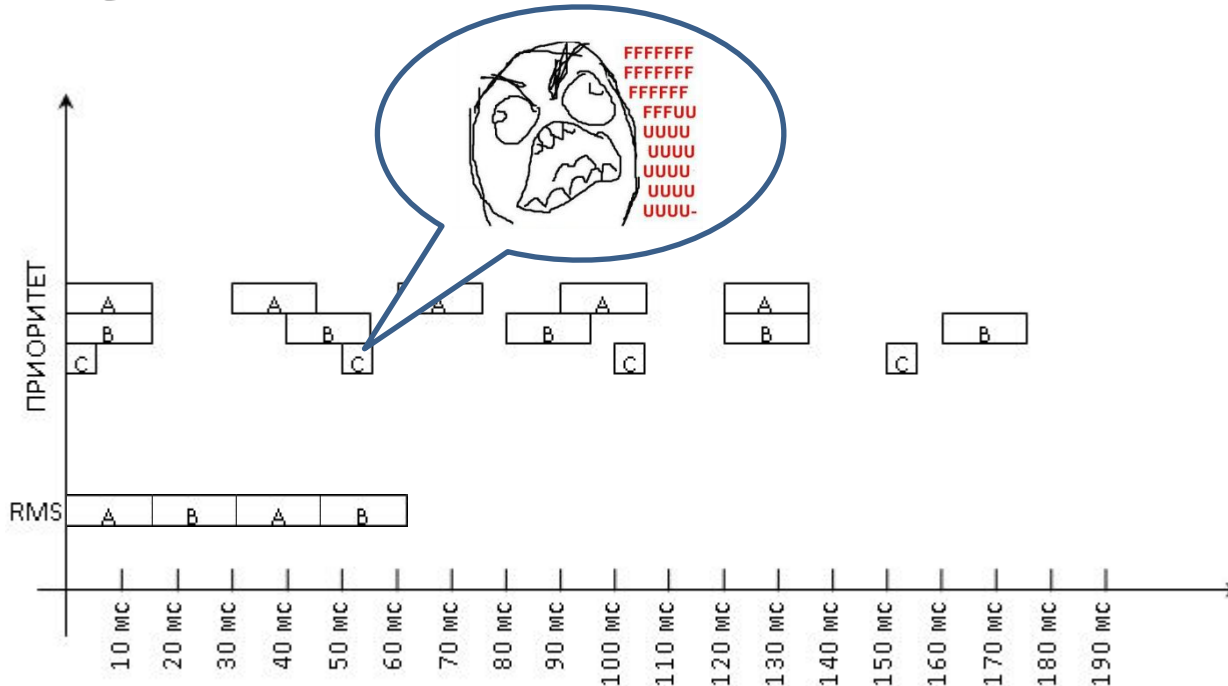
работать.

$$c_A = 15 \text{ мкс}, T_A = 30 \text{ мкс}$$

$$c_B = 15 \text{ мкс}, T_B = 40 \text{ мкс}$$

$$c_C = 5 \text{ мкс}, T_C = 50 \text{ мкс}$$

$$U = \frac{15}{30} + \frac{15}{40} + \frac{5}{50} = 0.975 < 1$$





ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических задач:

RMS

Очевидно, для RMS общее условие планируемости не всегда точно показывает

возможность его надёжной работы. Необходимо точно установить предельную загрузку системы для применения RMS

$$U = \sum_{i=1}^N \frac{c_i}{T_i} \leq N(2^{\frac{1}{N}} - 1)$$

(Чунг Лью, Джеймс
Лейланд)

UB-тест (*utilisation bound test*)

При $n \rightarrow \infty$ получаем $N(2^{\frac{1}{N}} - 1) \rightarrow \ln 2$.



ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических задач:

RMS

однако в первом примере нагрузка была выше $\ln 2$, и RMS надёжно работал
- очевидно, что UB-тест является достаточным, но не необходимым
условием для

понимания возможности прихода в пределы точной возможности применения
RMS

**МЫ ПОЙДЁМ
ДРУГИМ ПУТЁМ!**





ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических задач:

- RMS** необходимым условием является прохождение **RT-теста** (*response time test*)
- тест предложен Паритошем Пандья и Матайей Джозефом и основан на следующей идее:
рассматривается ситуация наибольшего риска несоблюдения дэдлайнов – когда все задачи активизируются одновременно (так называемый критический момент);
для каждой задачи системы рассчитываем наихудшее время отклика (*worst case response time, WCRT*) R_i ;
 - если $R_i < d_i$, то задача прошла тест и поддается планированию;
 - если все задачи прошли тест, то система поддается планированию;
 - если в наихудшем случае, то она поддается планированию и в любом другом случае.



ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических задач:

RMS время отклика для каждой из задач вычисляется по рекуррентной формуле:

$$R_i^0 = c_i + \sum_{j \in PH(i)} c_j$$
$$R_i^{k+1} = c_i + \sum_{j \in PH(i)} \left\lceil \frac{R_i^k}{T_j} \right\rceil \cdot c_j$$

множество задач с приоритетом,
большим,
чем у i -ой

- критерии остановки

вычисления прошла тест: $R_i^{k+1} = R_i^k$ и $R_i^{k+1} < d_i$

- задача провалила тест: $R_i^{k+1} > d_i$



ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических задач:

RMS применим RT-тест к нашим

- ~~пример~~ для случая, когда RMS работает:

$$c_A = 10\text{мкс}, T_A = 30\text{мкс}$$

$$c_B = 15\text{мкс}, T_B = 40\text{мкс}$$

$$c_C = 5\text{мкс}, T_C = 50\text{мкс}$$

$$U = \frac{10}{30} + \frac{15}{40} + \frac{5}{50} = 0.8083(3) < 1$$

- для задачи

$$A: R_A^0 = c_A = 10$$

$$R_A^1 = c_A = 10 = R_A^0 < 30$$



ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических задач:

RMS

- для задачи

$$R_B^0 = c_B + \sum_{j \in PH(i)} c_j = c_B + c_A = 15 + 10 = 25$$

$$R_B^1 = c_B + \sum_{j \in PH(i)} \left[\frac{R_i^k}{T_j} \right] \cdot c_j = c_B + \left[\frac{R_B^0}{T_A} \right] \cdot c_A = 15 + \left[\frac{25}{30} \right] \cdot 10 = 25$$

$$R_B^1 = R_B^0 < 40$$



Планирование периодических задач:

RMS

- для задачи

$$C: R_C^0 = c_C + \sum_{j \in PH(i)} c_j = c_C + c_B + c_A = 5 + 15 + 10 = 30$$

$$\begin{aligned} R_C^1 &= c_C + \sum_{j \in PH(i)} \left\lfloor \frac{R_i^k}{T_j} \right\rfloor \cdot c_j = c_C + \left\lfloor \frac{R_C^0}{T_A} \right\rfloor \cdot c_A + \left\lfloor \frac{R_C^0}{T_B} \right\rfloor \cdot c_B = \\ &= 5 + \left\lfloor \frac{30}{30} \right\rfloor \cdot 10 + \left\lfloor \frac{30}{40} \right\rfloor \cdot 15 = 30 = R_C^0 \end{aligned}$$

$$R_C^1 < 50$$

Все задачи прошли тест, следовательно, вся система поддается планированию



ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических задач:

RMS

- теперь рассмотрим случай, когда RMS не работал;

$$c_A = 15\text{мкс}, T_A = 30\text{мкс}$$

$$c_B = 15\text{мкс}, T_B = 40\text{мкс}$$

$$c_C = 5\text{мкс}, T_C = 50\text{мкс}$$

$$U = \frac{15}{30} + \frac{15}{40} + \frac{5}{50} = 0.975 < 1$$

- для задачи

$$\begin{aligned} \text{A: } R_A^0 &= c_A = 15 \\ R_A^1 &= c_A = 15 = R_A^0 < 30 \end{aligned}$$



Планирование периодических задач:

RMS

- для задачи

$$R_B^0 = c_B + \sum_{j \in PH(i)} c_j = c_B + c_A = 15 + 15 = 30$$

$$R_B^1 = c_B + \sum_{j \in PH(i)} \left\lceil \frac{R_i^k}{T_j} \right\rceil \cdot c_j = c_B + \left\lceil \frac{R_B^0}{T_A} \right\rceil \cdot c_A = 15 + \left\lceil \frac{30}{30} \right\rceil \cdot 15 = 30$$

$$R_B^1 = R_B^0 < 40$$



ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических задач:

RMS

- для задачи

$$R_C^0 = c_C + \sum_{j \in PH(i)} c_j = c_C + c_B + c_A = 5 + 15 + 15 = 35$$

$$\begin{aligned} R_C^1 &= c_C + \sum_{j \in PH(i)} \left\lfloor \frac{R_i^k}{T_j} \right\rfloor \cdot c_j = c_C + \left\lfloor \frac{R_C^0}{T_A} \right\rfloor \cdot c_A + \left\lfloor \frac{R_C^0}{T_B} \right\rfloor \cdot c_B = \\ &= 5 + \left\lfloor \frac{35}{30} \right\rfloor \cdot 15 + \left\lfloor \frac{35}{40} \right\rfloor \cdot 15 = 50 \end{aligned}$$

$$\begin{aligned} R_C^2 &= c_C + \sum_{j \in PH(i)} \left\lfloor \frac{R_i^k}{T_j} \right\rfloor \cdot c_j = c_C + \left\lfloor \frac{R_C^1}{T_A} \right\rfloor \cdot c_A + \left\lfloor \frac{R_C^1}{T_B} \right\rfloor \cdot c_B = \\ &= 5 + \left\lfloor \frac{50}{30} \right\rfloor \cdot 15 + \left\lfloor \frac{50}{40} \right\rfloor \cdot 15 = \end{aligned}$$

EPIC FAIL



ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических

задач: EDF

алгоритм EDF (*Earliest Deadline First*) является алгоритмом с динамическими

приоритетами. RMS приоритет задачи зависит от её относительного дедлайна – того

дедлайна, который используется алгоритмом в данный момент работы планировщика; поддаётся

планированию, то она может быть обслужена алгоритмом EDF даже при самых

высоких нагрузках;



ПЛАНИРОВАНИЕ ЗАДАЧ

Планирование периодических задач: EDF

задач: EDF

$$c_A = 15 \text{ мкс}, T_A = 30 \text{ мкс}$$

$$c_B = 15 \text{ мкс}, T_B = 40 \text{ мкс}$$

$$c_C = 5 \text{ мкс}, T_C = 50 \text{ мкс}$$

$$U = \frac{15}{30} + \frac{15}{40} + \frac{5}{50} = 0.975 < 1$$

