2个 CPU

7th task

upper CPU help t7

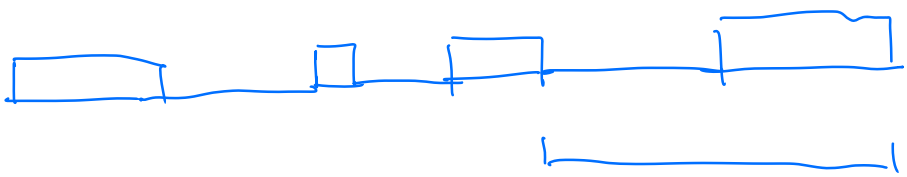
$$R_7 = C_7 + S_7 + \sum_{i=2\sim6} W_i(R_7) + \left[R_7 - \sum_{i=1\sim6} W_i(R_7)\right] + (n-1)(C_7 + S_7)$$

lower CPU $t_2 \sim t_6$ 对 t7 的阻

task 3 的 $W_3(R_7)$

待定

W

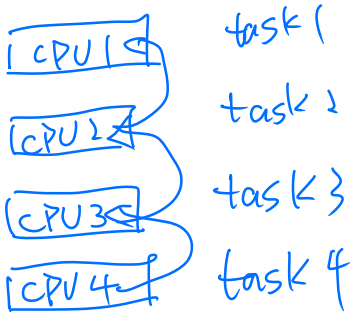


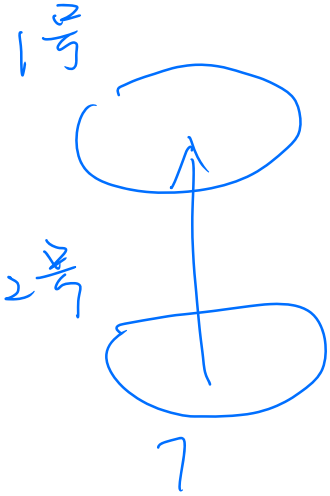
即 task 2~6 对 task 7 的阻碍被计算两遍

且没有考虑

"CPU1 能多做一点 task4，从而减轻 task4 在 CPU2 上对 task7 的阻碍作用."

4个





core 0 ← task1

core 1 ← task2

core 2 ← task3

core 3 ← task4

task1~6

task5: $\sum C_5 + \sum S_5$ t

2个 CPU　　4个task.　task 1 ~ 4

$R_4 = \sum C_4 + \sum S_4 + W_2(R_4) + W_3(R_4) - [R_4 - W_1(R_4) - W_3(R_4)]$

$+ C_4 + S_4 - W_1(R_4)$

1号CPU帮task 4

之前多计算的

a

b

在CPU2上task3对t4

在CPU1上task3对t4

若 task 3 一部分被CPU 1做了，b会大一点.

那a就该小一点.

W_1 定义：在一段时间 t 内 task 1 最多能有多少时间在干活

task 2 还 task 3

t_k 内　对 task 3 的阻碍是 $W_1(t_k)$

100　　$W_1(t_k)$

：task 1 在 CPU 1 上对 task 3 的阻碍.

$-W_1(R_4)$ 意义 对两个CPU整体来说，task 1 对 task 3 的

0　　0