

time slice = max waiting time (در قی)

تعداد پردازشهای آماده اجرا

از برآورد max waiting time که در دوره قبلی بدست آورده می شود و می توانیم
ضرایب پردازشهای که CPU intensive یا CPU burst بودند، مدت زمان طولانی
مدت و اجبار دارند را بفهمیم.

مثلاً اگر پردازشهای که CPU burst است اول کار قرار بگیرد
مدت زمانی که پردازشها صبر می کنند زیاد می شود. چون time slice یا مدت زمانی که
به هر پردازش می دهیم وابسته به این مقدار است → هرچه به پردازشها بزرگتر شوند و
time slice هم بزرگتر شود → این به دفع به پردازشهای CPU burst می خورد
چون ~~تعداد~~ context switch ها کم می شود → بهره داری هم در CPU
زیادتر می شود (CPU utilization ↑)

وقتی time slice زیاد شود → Response time هم زیاد می شود
به این CPU bound ها فرقی می کند
می توانیم ببینیم که throughput هم زیاد می شود.

اما اگر پردازشهای IO bound هستند یعنی این که، زمان response
زیاد شود → بعد از آن کم می شود و در بر ریزش جواب داده می شود
اگر max wt کم باشد → می توانیم ببینیم به این پردازشها
IO burst یا IO bound بودند → که تازه مدت اجبار است
و برای سیستم های که interactive هستند مناسب است.

↓ این دلیلی است که time slice بزرگ هم کم می شود → بین پردازشها
در یک این نوع آزمون به زمانها ضربه می خورد که به این پردازشها از یک نوع باشند
یعنی با مقدار زیاد از IO bound یا IO bound CPU
اگر نصف ~~نصف~~ نصف باشند به قدری می شود.

Completely fair scheduler ← CFS

هدف از این آلوگوریتم ← به هر task سهم عادلانه ای از CPU processing time ببرد.

این آلوگوریتم ۲ تا چیز را باید تعیین کند

① از بین پردازشهای که در صف ready داریم کدام را انتخاب کنیم تا بسوزد CPU دهیم

② اذن quantum زمانی که به هر پردازش می دهیم و مقدار باید باشد

مقدار به هر پردازش زمان برای اجرا دهیم

این آلوگوریتم برای جواب به سوال ۱ و ۲

Scheduling decision

چگونه اذن پردازشهای را انتخاب کنیم

vruntime - پیش (virtual runtime) کمتر باشند

برای پاسخ به سوال دوم

Scheduling quantum
= time slice

یعنی این آلوگوریتم task های را انتخاب

می کند که nice کمتری دارند

↓ nice value → ↑ اولویت

اولویت بالاتری برای processing CPU دارند

این آلوگوریتم میار به اولویت پردازشها برای انتخاب بین اهمیت و زمان

برای انتخاب پردازش بعدی به vruntime یک پردازش وقت می کند

اگر اولویت پردازشهای برابر باشد ← virtual runtime = actual runtime

اگر ~ ~ ~ با برابر باشند ~ ~ ~ vruntime < actual runtime

اگر ~ ~ ~ پایین ~ ~ ~ > ~ ~ ~

صورت متغیر است

طبق جدول های این الگوریتم

سوال ۶ ب

$$\text{time_slice}_k = \frac{\text{weight}_k}{\sum_{i=0}^{n-1} \text{weight}_i} \cdot \frac{\text{sched_latency}}{\text{time interval}}$$

مخرج ثابت است

یک آرایه به نام prio_to_weight داریم که nice values در آن قرار دارد.
 به weight در این آرایه map می کنیم.
 weight هر پروسه و یک متغیر با nice values در آن یعنی اگر
 weight یک پروسه بیشتر باشد + هم بیشتر از sched_latency می گیرد.
 time interval

در واقع $\text{time slice} \sim \text{weight}$
 هر پروسه

پس هر چه nice value کمتر باشد + weight بیشتر می شود.
 time slice هم بیشتر می شود.

$$\text{vruntime} = \text{vruntime}_0 + \frac{\text{weight}_i}{\text{weight}_i} \cdot \text{runtime}_i$$

virtual runtime (مجاز)

actual runtime (واقعی)

$\text{weight}_0 = \text{nice_value}$ و weight_i در آرایه دارد +
 رابطه weight و vruntime و nice_value

$$\text{vruntime} \sim \frac{1}{\text{weight_پروسه}}$$

پس هر چه nice value کمتر باشد +
 weight کمتر می شود +
 بزرگتر

vruntime کمتر + زمان بیشتری در صف انتظار می ماند و بهتری
 time slice بیشتری هم دارد.
 در واقع در این الگوریتم هر چه weight کمتر باشد + زمان بیشتری در صف انتظار می ماند +
 زمان بیشتری در صف انتظار می ماند.

انواع ۲ -
پارامترهای که در این آلگوریتم استفاده شده از جمله sched_latency که

مقدار بازه زمانی را مشخص می کند که تا چقدر طول می کشد تا پردازش های آماده به کار ready

وجود دارند و حداقل یک بار CPU می گیرند و فصول های مشابهی

virtual runtime و time_slice و به نشان از مدت زمانی که آلگوریتم

در sched_latency

اگر بازه خیلی کوچک باشد → مقدار Context switch کاهش می یابد

→ بهینه نیست

راه حل ۲

یک حداقل → min granularity

را برای این time interval مشخص می کنند

برای وقتی که یک پروسه میوه IO انجام دهد وقتی که پروسه در runtime - نیازی نیست

نسبت به بقیه کم است → تمام CPU می گیرد → باید گرسنگی بقیه پروسه ها

می خورد اما یک راه حل که ارائه دارند ۲

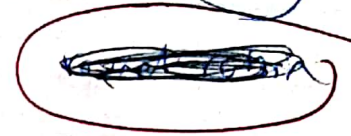
$$vruntime = \min vruntime$$

که در وقت پروسه ها هست

نیازی برای رفع این مشکل هم راه حل هست

	priority	burst
P1	1	0
P2	0	10 9
P3	2	0
P4	3	0

SU1
(2)



proemotive ← مادم قبله كذا وبقية

SRF

P1	P1	P3	P4	P2
0	1	2	3	10

$$\text{avg wtp} = \frac{0 + (10-1) + 0 + (3-2) + (3-1)}{5} = \frac{9}{5}$$

$\frac{10}{5} = \frac{2}{1} \rightarrow$ متوسطه، كذا، انكاف، ع

$$\text{avg RTP} = \frac{0 + 0 + (3-2) + (3-1)}{5} = \frac{2}{2}$$

ماتنن، متوسطه، كذا، باغ

	زمان	burst
P_2	0	$10 \rightarrow 4 \rightarrow 1 \rightarrow 0$
P_1	1	$5 \rightarrow 0$
P_3	3	$1 \rightarrow 0$
P_4	4	$2 \rightarrow 0$

این استوریج RT
که به صورت preemptive
کار می کند

P_2	P_1	P_3	P_4	P_1	P_2
0	5	1	10	14	19

$$\text{avg wt} = \frac{0 + (14 - 5) + (5 - 1) + (1 - 3) + (10 - 4)}{5} =$$

$$\frac{14 + 5}{5} = \left\lceil \frac{19}{5} \right\rceil$$

$$\text{avg RT} = \frac{0 + (5 - 1) + (1 - 3) + (10 - 4)}{5} = \frac{14}{5} = \left\lceil \frac{14}{5} \right\rceil$$

$$\rightarrow \text{avg } \left\lceil \text{RT} \times \frac{V}{r} \right\rceil$$