

16.8. O(1)スケジューラのプロセスの優先度

プロセスの優先度には、「静的優先度」と「動的優先度」があります。

静的優先度

- ユーザがシステムコールにて設定可能な値
- 通常プロセスの場合は nice 値から算出

動的優先度

- スケジューラが次に実行すべき通常プロセスを選ぶ時に評価する値
- プロセスの静的優先度と CPU 使用状況などから算出
- 静的優先度とは異なり、ユーザは値の変更不可
- 決まった契機で再計算される

静的優先度

静的優先度の指定方法を以下に示します。

	静的優先度	リアルタイム優先度	nice
リアルタイムプロセス	0	99	
	1	98	
	:	:	
	98	1	
	99	0	
通常プロセス	100	0	20
	101		-19
	:		:
	139		18
	139		19

リアルタイムプロセスの場合は、リアルタイム優先度(0 ~ 99) を指定することで静的優先度を変更します。
通常プロセスの場合は、nice 値(19 ~ -20) を指定することで静的優先度を変更します。

動的優先度

動的優先度は通常プロセスのみ割り当てられます。

通常プロセスは決まった契機で再計算されます。
再計算時には、以下の内容を考慮し、動的優先度が割り当てられます。

- 通常プロセスの種類
- ボーナス値
- 静的優先度(nice 値)

SCHED_OTHERの種類

通常プロセスには、I/O バウンド型と CPU バウンド型という種類があります。
I/O バウンド型は、頻繁に I/O デバイスにアクセスし、多くの時間を I/O 操作の完了待ちに費やすアプリケーションです。
CPU バウンド型は、多くの CPU 時間を必要とする大量の演算を行うアプリケーションが該当します。
スケジューラは I/O バウンド型のプロセスを優先する傾向にあります。

次の式を満たす場合、スケジューラは I/O バウンド型プロセスと判断します。

$$\text{動的優先度} \leq 3 \times \text{静的優先度} \div 4 + 28$$

静的優先度 139 のプロセスが I/O バウンド型プロセスとみなされることはありません。
※I/O バウンド型プロセスと判断するための式は以下のようにも書けます。

$$\begin{aligned} \text{ボーナス値} - 5 &\geq \text{静的優先度}(139) \div 4 - 28 \\ \text{ボーナス値} &\geq \text{静的優先度}(139) \div 4 - 28 + 5 \\ \text{ボーナス値} &\geq 11.75 \end{aligned}$$

ボーナス値が 11 を越えることはありません。
そのため、静的優先度 139 のプロセスが I/O バウンド型プロセスとみなされることはありません。

ボーナス値

ボーナス値は、平均休止時間により求められます。
平均休止時間とボーナス値の対応を以下に示します。

平均休止時間	ボーナス値
0ms 以上 ~ 100ms 未満	0
100ms 以上 ~ 200ms 未満	1
200ms 以上 ~ 300ms 未満	2
300ms 以上 ~ 400ms 未満	3
400ms 以上 ~ 500ms 未満	4
500ms 以上 ~ 600ms 未満	5
600ms 以上 ~ 700ms 未満	6
700ms 以上 ~ 800ms 未満	7
800ms 以上 ~ 900ms 未満	8
900ms 以上 ~ 1000ms 未満	9
1000ms 以上	10

平均休止時間は、プロセスが休止している間に経過した平均ミリ秒時間を表します。
しかし、平均休止時間は以下に示す要因により、プロセスが起動してから現在までの休止時間を単純に平均するわけではありません。

- TASK_INTERRUPTIBLE 状態での平均休止時間と TASK_UNINTERRUPTIBLE 状態での平均休止時間は算出方法が異なる
- プロセス実行中は実行平均休止時間は減算される
- 平均休止時間は 1 秒よりも大きくなることはない(1 秒よりも大きくなると切り捨てられる)

静的優先度やスケジューリングポリシーを変更した場合は、ボーナス値は 0 となります。

動的優先度の算出

動的優先度は以下の計算式により求められます。

$$\text{動的優先度} = \max(100, \min(\text{静的優先度} - \text{ボーナス値} + 5, 139))$$

動的優先度は、プロセスが休止している間に経過した時間を考慮して決定されます。
そのため、待ち時間の長い I/O バウンド型プロセスほど動的優先度が高くなります。

動的優先度の再計算契機

以下の場合、動的優先度が再計算されます。

- タイムスライスを使い果たした場合
active キューから expired キューに接続
- 待ち状態から復帰した場合
wait キューから active キューに接続
- 通常プロセスで静的優先度を変更した場合
expired キューに接続
動的優先度は静的優先度と同じ

以下の場合、動的優先度は再計算されません。

- プロセスを生成した場合
active キューに接続
子プロセスは親プロセスより後に active キューに接続される
- 他のプロセスに実行権を譲った場合(sched_yield(2))
active キューから expired キューに接続
動的優先度は変更されない

nice 値を変更する際の注意点

nice 値を変更する際には、動的優先度により意図しない優先度逆転が発生する可能性があることに注意する必要があります。

priority100	prio=100(-4) static_prio=104	
priority101	prio=101 static_prio=101	
priority102	prio=102(+1) static_prio=101	prio=102 static_prio=102
priority103	prio=103(+2) static_prio=101	
priority104		
:		
priority139		

優先度が逆転しても問題が発生しない
アプリケーション設計を行う必要がある

※図中の prio は動的優先度を、static_prio は静的優先度を表します。