

16. プロセススケジューリング

Linux におけるスケジューリングには TSS(Time Sharing System)を採用しています。TSS はプロセスから他のプロセスへ短い間隔で切り替えることにより、あたかも複数のプロセスを同時に実行しているように見えるシステムです。
※Linux のスケジューリング方針は、TSS 以外にも存在しています。
(リアルタイムスケジューラ)
リアルタイムスケジューラについては、後ほど説明します。

スケジューリングの対象となるものはプロセスとスレッドになります。

プロセスのスケジューリングを行うスケジューラの動きは Linux 2.4 と Linux 2.6 で異なります。

Linux 2.4 スケジューラ

全プロセスを 1 本の線形リストで管理しています。マルチプロセッサ環境の場合でも線形リストは 1 本です。

スケジューラの動作は以下ようになります。

タイマー割り込みでの動作

- ・ タイムスライスをデクリメント
- ・ タイムスライスが 0 になった場合、プロセスディスパッチが発生

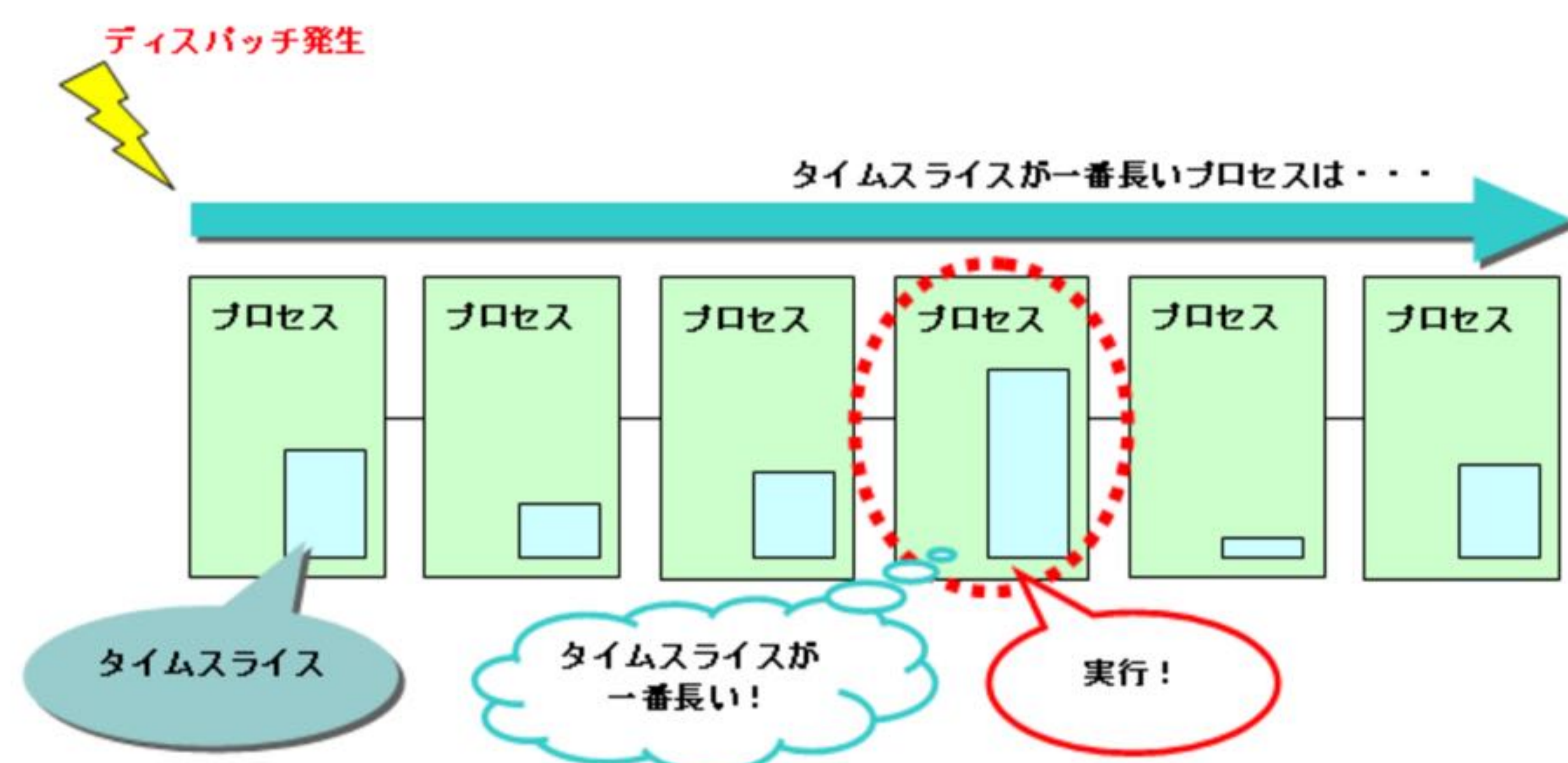
プロセスディスパッチが発生した場合

- ・ 線形リストの中で最もタイムスライス(プロセスの CPU 使用可能残り時間)が長いプロセスに実行権限を与える

新しいプロセスが生成された場合

- ・ 新しいプロセスが生成された場合は線形リストの一番最後に接続

以下の図は Linux 2.4 スケジューラのプロセスディスパッチ時の動作を表したものです。



問題点

Linux 2.4 スケジューラでは、以下の場合にプロセスディスパッチのオーバーヘッドが増大するという問題がありました。

- ・ プロセス数が増えた場合
例えば、プロセス数が 100 であった場合はプロセスのタイムスライスを 100 回調べなければならないため、時間が掛かる。
- ・ カーネルの処理が長時間かかる場合

Linux 2.6 スケジューラでは上記の点が改良されました。

Linux 2.6 スケジューラ

Linux 2.6 ではプロセスディスパッチのオーバーヘッドを改善するため、以下のスケジューラが採用されました。

2.6.22 以前

O(1) (オーダワン)スケジューラ

2.6.23以降

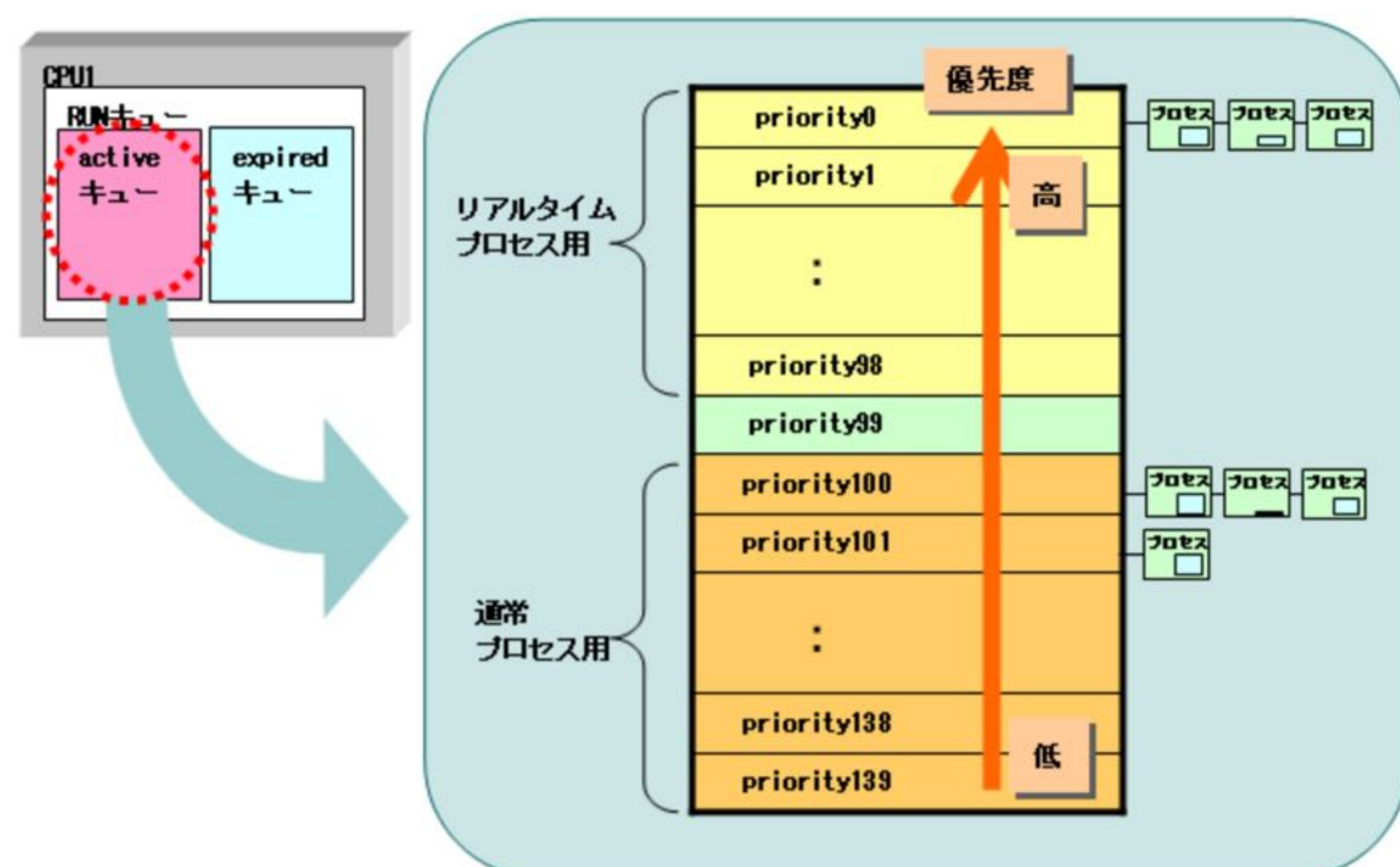
CFS (Completely Fair Scheduler)

O(1)スケジューラ

O(1)スケジューラは、実行可能なプロセスの中で一番優先度の高いプロセスを 1 アクションで特定することで、Linux 2.4 スケジューラの問題点を解決しています。

O(1)スケジューラでは、優先度ごとに用意した線形リストでプロセスを管理します。優先度は数字が小さいほど高くなります。

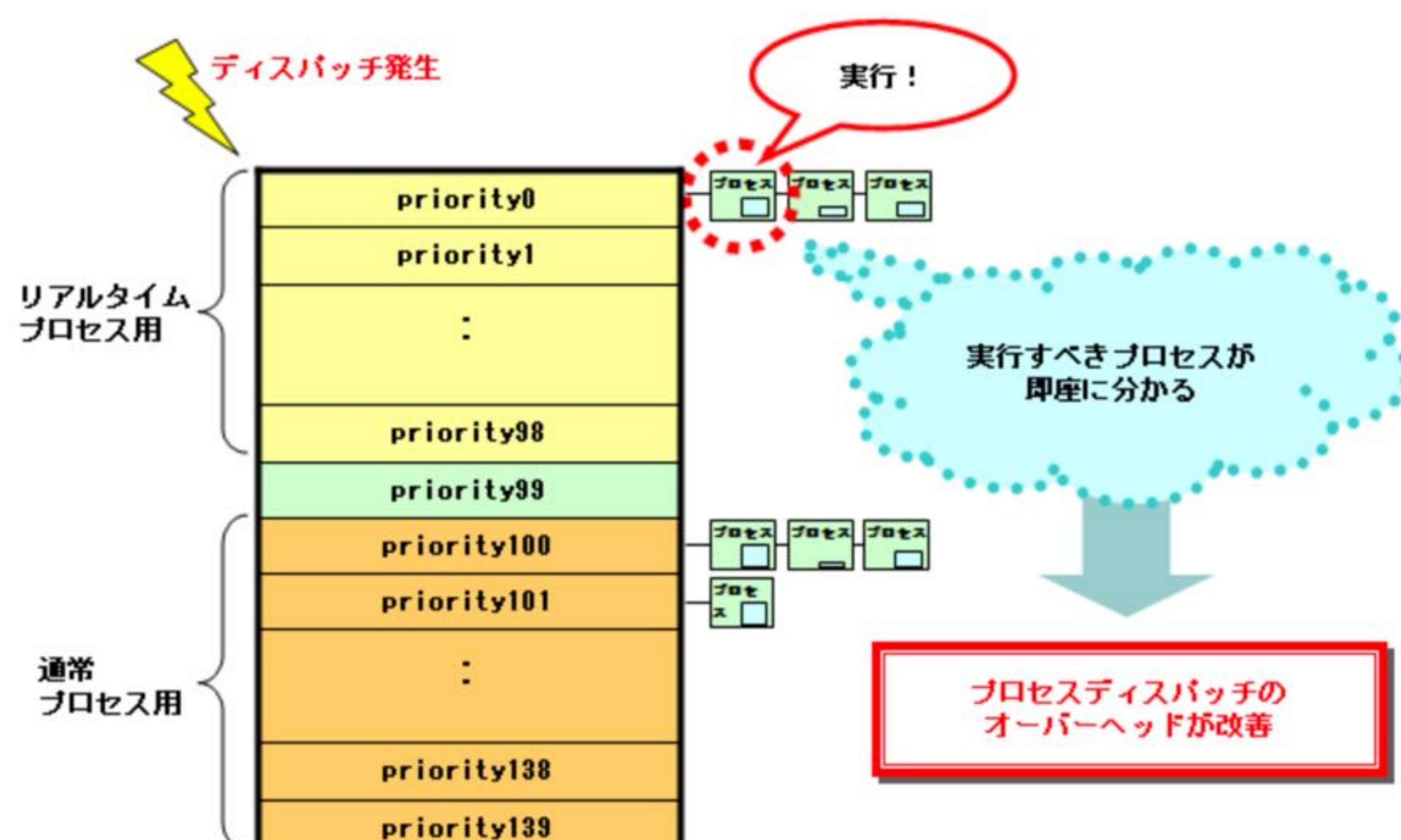
以下の図は優先度ごとに線形リストでプロセスを管理している様子を表したものです。



ディスパッチが発生した場合、実行可能状態にあるプロセスの中で最も優先度が高いプロセスが実行されます。O(1)スケジューラは優先度ごとに線形リストを持ちますが、線形リストの先頭にあるプロセスが最初に実行されます。そのため、プロセス数が増えてもディスパッチのオーバーヘッドが増大することはありません。(Linux 2.4 スケジューラはプロセス数に比例してディスパッチのオーバーヘッドが増えるので、O(n)スケジューラといえます)

O(1)スケジューラでもプロセスごとにプロセスのタイムスライスが存在します。Linux 2.4 スケジューラと同じく、タイマー割り込みごとにタイムスライスをデクリメントし、タイムスライスが 0 になるとプロセスディスパッチが発生します。

以下の図はプロセスディスパッチ時に実行すべきプロセスが即座に分かる様子を表したものです。



実行権を与えられるのは最も優先度の高いプロセスなので、優先度の低いプロセスはいつまで経っても実行されないのではないのでしょうか？

O(1)スケジューラはリアルタイムスケジューラであるため、優先度の低いプロセスは、自分より優先度の高いプロセスが存在する限り実行されません。