FreeBSDにおける閏秒のサポートについて

目次

1. イントロダクション 1
2. FreeBSDにおけるデフォルトの閏秒のハンドリング方法 1
3. 注意事項 2
4. テスト方法 2
5. 結論 2

1. イントロダクション

leap second(閏秒)は地球の自転と時刻を同期させるために使用される特別な秒のことです。この記事はFreeBSDがどのように閏秒を扱っているかを解説します。

執筆段階ですと、次に閏秒を挿入するのは協定世界時で2015年6月30日 23時59分60秒になります。この閏秒は北アメリカ、南アメリカ、アジア太平洋地域の営業日に実施されます。

閏秒はBulletin CにおけるIERSにおいて発表されています。

閏秒の一般的な動作に関してはRFC 7164で解説されています。time2posix(3)に関しても参照してください。

2. FreeBSDにおけるデフォルトの閏秒のハンドリング方法

閏秒のもっとも簡単な取り扱い方法はFreeBSDがデフォルトで使っている POSIX のタイムルールとNTPを組み合わせる方法です。ntpd(8)が上位の NTP サーバと同期している場合には閏秒は適切に処理され、閏秒は日の最後の秒をもう一度繰り返すという方法を自動的に実施します。これ以外の調整は必要ありません。

アップストリーム NTP サーバが閏秒を適切に処理していない場合、ntpd(8)は時刻のずれに気づいたアップストリームサーバが時刻を修正したあとに時刻を合わせることになります。

NTP を使っていない場合、閏秒が経過したあとに手動でシステムクロックを変更する必要があります。

3. 注意事項

閏秒は UTC (協定世界時)での真夜中に世界中で同時に挿入されます。日本では午前の半ば、太平洋地域では日中、米国では午後の遅いタイミング、欧州は夜です。

FreeBSDでは適切で安定した NTP サービスが提供されていれば先ほど説明したように閏秒のタイミングで設計通りに処理が行われることになると思います。

しかしながら、実際のところカーネルに対して閏秒について尋ねてくるアプリケーションは存在しないことに注意してください。我々の経験からしますと、想定されているように、閏秒の処理は閏秒のタイミングで1秒を1度繰り返すというもので、これはほとんどのアプリケーションプログラマにとっては想定していないものだと思います。

FreeBSDと同じ方法で閏秒を処理しているしていないに関わらずほかのオペレーティングシステムやほかのコンピュータと、適切で安定した NTP サービスを使用していないシステムは閏秒に関してはまったく関知してくれません。

コンピュータが閏秒が原因でクラッシュするという話は聞いたことがありませんが、経験からしますとパブリックに利用されている NTP サーバの一部は不適切に閏秒を処理して報告をおこなっています。

閏秒が原因でなにか問題が発生しないことを確認するようにしてください。

4. テスト方法

閏秒が使われるかどうかをテストする方法があります。NTP の特性から、テストは閏秒が発生する24時間前から行います。いくつかの有名な時刻の参照ソースは閏秒発生の1時間前にアナウンスを行います。NTP デーモンに次のクエリを発行します:

```
% ntpq -c 'rv 0 leap'
```

`leap_add_sec`インディケータを含んだ出力は閏秒を適切にサポートしていることを意味しています。閏秒が発生するよりも24時間前、または閏秒が発生した後には`leap_none`が表示されます。

5. 結論

実際のところ、閏秒がFreeBSDで問題になることはありません。この要約がどのように閏秒の処理で何が行われるのか、どうやって閏秒の処理を問題なく済ませればよいのかという考えを明確にする手助けになればと思います。