

停止と再起動

この文書では Unix に類似したシステムでの Apache の停止と再起動について扱っています。Windows ユーザ (Windows で Apache を使う場合) は、実行中の Apache にシグナルを送る¹もご覧下さい。

トピック

イントロダクション.....	1
急な停止.....	1
緩やかな再起動.....	1
急な再起動.....	3
付録: シグナルと競合状態.....	3
URI References.....	3

イントロダクション

たくさんの実行形式 httpd がシステム上で 実行されているのに気が付くでしょうが、シグナルを送るのは 親プロセスだけで、それ以外の個々のプロセスには シグナルを送らないで下さい。その親プロセスの pid は PidFile に書かれています。これはつまり、親以外のプロセスに シグナルを送る必要すらない、ということです。親プロセスに送ることができる 3 種類のシグナルがあります: TERM, HUP, USR1 です。これらの説明については続きをご覧下さい。

親プロセスにシグナルを送るには、次のようなコマンドを発行して下さい:

```
kill -TERM `cat /usr/local/apache/logs/httpd.pid`
```

これの実行状況は次のコマンドで読むことができます:

```
tail -f /usr/local/apache/logs/error_log
```

ここに挙げた例は、各自の ServerRoot と PidFile の設定に適合するように適宜修正して下さい。

apachectl² と呼ばれるシェルスクリプトで、Apache にシグナルを送る手順を 自動化することができます。このスクリプトの詳細に関しては、Apache の起動³の文書をご覧下さい。

急な停止

シグナル: TERM
apachectl stop

TERM シグナルを親プロセスに送ると、即座に子プロセス全てを kill しようとします。子プロセスを完全に kill し終わるまでに数秒かかるかもしれません。その後、親プロセス自身が終了します。処理中のリクエストは全て停止され、もはやリクエストに対する 応答はされません。

緩やかな再起動

停止と再起動

シグナル: USR1

```
apachectl graceful
```

親プロセスは USR1 シグナルを受け取ると、子プロセスに現在のリクエストの処理の後に終了する（あるいは何もしていなければすぐに終了する）ように助言します。親プロセスは設定ファイルを再読込して、ログファイルを開き直します。子プロセスが徐々になくなるに従って、新しい世代の設定による子プロセスに置き換えていきます。そして、これらが新たなリクエストに即座に応答し始めます。

特定のプラットフォームでは USR1 を緩やかな再起動のために使うことができませんが、代替のシグナル（例えば WINCH）が使用できるでしょう。apachectl graceful というコマンドはプラットフォームに合ったシグナルを送ります。

このコードは常に MPM のプロセス制御ディレクティブの設定を重視しますので、クライアントのリクエストを扱うプロセスとスレッドの数を再起動の処理中も適切な値に維持されます。また、次のようにして `StartServers` を守ります: 少なくとも 1 秒後に `StartServers` 個の新しい子プロセスが生成されていなければ、その数になるように適宜プロセスを生成します。この挙動は現在の負荷に対して適切な子プロセスの数と `StartServers` パラメータでの希望の数の両方を維持しようとしています。

`mod_status` を使用している場合は、USR1 シグナルが送られた際にサーバ統計がゼロに設定されないことに注意してください。サーバが新しいリクエストに応答不能な時間を最小にするように（リクエストは OS によってキューに追加されるので絶対に紛失はしません）、また同時に、希望のチューニングパラメータを守るようにコードは書かれています。このようにするために、世代をまたがった全子プロセスの追跡に使われているスコアボードを維持しなければなりません。

`status` モジュールは、緩やかな再起動以前から開始してリクエストに応答し続けている子プロセスを特定するために、`G` を使うこともします。

現在、USR1 を使うログ移動スクリプトでは、再起動前の子プロセスがログを書き終わったことを確認する方法がありません。古いログに対して何かする前に、USR1 シグナルを送った後いくらか適当な時間待つことを提案します。例えば、帯域の狭い通信路のユーザのリクエストのほとんどが 10 分以下で完了しているということが分かっていたら、古いログに何かする前に 15 分待つということです。

再起動時に設定ファイルに誤りがあると、親プロセスは再起動せずにエラーとともに終了します。緩やかな再起動の場合は、親プロセスが終了した後も子プロセスが実行されたまま放置されたりもします。（最後のリクエストを処理した後「緩やかに終了」する子プロセスとなります。）サーバを再起動する際に、これが問題になるかもしれません — サーバは `listen` するポートにバインドできないかもしれません。再起動する前に、設定ファイルの構文を `-t` コマンドライン引数 (`httpd4` をご覧ください) を使って検証することができます。設定ファイルの意味的な内容を構文と同様に検証したい場合は、非 root ユーザで `httpd` を起動しようとするればわかります。もしエラーがなければ、ソケットやログを開こうとして root でないため（もしくは `httpd` が既に必要なポートにバインドしているため）に失敗するでしょう。これ以外の理由で起動に失敗したのであれば、それは設定ファイルのエラーで、緩やかな再起動を行う前にその誤りを修正しなければなりません。

急な再起動

シグナル: HUP

```
apachectl restart
```

HUP シグナルを親プロセスに送ると、TERM と同様に子プロセスを kill しますが、親プロセスは終了しません。設定ファイルを再読み込みして、ログファイル全てを開き直します。その後、新しい子プロセスを起動して応答を続けます。

`mod_status` を使っている場合は、HUP が送られた場合にサーバ統計がゼロに設定されることに注意してください。

再起動時に設定ファイルに誤りがあると、親プロセスは再起動せずにエラーとともに終了します。これを避けるには次の方法をご覧ください。

付録: シグナルと競合状態

Apache 1.2b9 以前は、再起動や停止のシグナルを含む競合状態（競合状態を簡単に説明すると：タイミンにによる問題で、具合の悪い時間帯にちょうど何かが起こると予想外の動作をするようなことを指します）がありました。「正しい」機能を持っているアーキテクチャでは、できるだけこのようなことが起こらないようにしています。しかし、ある種のアーキテクチャでは競合状態は未だ確実に起こりえるということに注意してください。

ディスク上で `ScoreBoardFile` を使用しているアーキテクチャでは、潜在的にスコアボードが壊れる可能性があります。スコアボードが壊れた場合は、“bind: Address already in use” (HUP 後) や “long lost child came home!” (USR1 後) といった結果になります。前者は致命的なエラーですが、後者はスコアボードスロットを失うだけです。ですから緩やかな再起動は、たまに確実な再起動 (HUP) も併用して使った方が良いでしょう。これらの問題を克服するのは非常に難しいのですが、幸いなことに大部分のアーキテクチャではスコアボードのファイルは必要ありません。これを使用するアーキテクチャは、`ScoreBoardFile` をご覧ください。

全てのアーキテクチャにおいて、個々の子プロセスで継続的な HTTP コネクション (KeepAlive) に関する小さな競合状態が起こりえます。リクエスト行を読んだ後、そしてリクエストヘッダを読む前に子プロセスは終了するかも知れません。これに対する修正がありますが 1.2 で修正するには発見が遅すぎました。理論的には、これは問題ではありません。なぜなら KeepAlive のクライアントは、ネットワーク遅延やサーバのタイムアウトなどに備えていなければならないからです。実際にも何か影響があるようには見えません -- テストケースでサーバを 1 秒間に 20 回再起動してもクライアントは壊れた画像や空のドキュメントを受け取ることなく正常に閲覧できています。

URI References

- [1] <http://httpd.apache.org/docs-2.1/platform/windows.html#signal>
- [2] <http://httpd.apache.org/docs-2.1/programs/apachectl.html>
- [3] <http://httpd.apache.org/docs-2.1/invoking.html>
- [4] <http://httpd.apache.org/docs-2.1/programs/httpd.html>