1-1 ログの基本をおさえておこう

Author 近藤 成(こんどう じょう)
Mail jj2kon@gmail.com Web http://server-setting.info/

ログは、システムの稼働状況をはじめとして、さまざまな情報を蓄積しています。本章では、システムログ、アプリケーションのログ、それぞれのログの読み方・扱い方を解説します。これによって、システムに障害が起きたときに原因追及ができるようになったり、分析ができるようになります。まず本節1-1では、ログの基本をしっかりと学びます。

ログは、いつ、だれが、どこに収集するの?

そもそもログは、ソフトウェア[注1](プログラムの意味)が実行された経過情報を出力したものです。つまり、**プログラムが実行されたときに、あらかじめ決められている出力先(ファイルなど)へ、プログラムの処理情報を書き込みます。**

一般的なUNIX系のアプリケーション[注2]のほとんどは、ログの出力先(ファイル名など)を指定ができるようになっており、そこにあらかじめ設定しておけば、そのアプリケーションが起動した際に、その設定情報を読み込み、指定してあるログの出力先(ファイルなど)へ出力されるようになります(**図1**)。

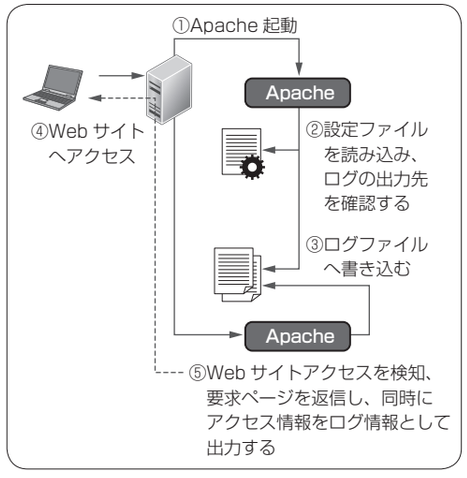
ヒント ここで紹介するCentOSのデフォルトの設定では、ログの出力先は、ほとんどの場合、**/var/log/**配下のディレクトリおよびファイルへ出力するようになっています。

たとえば、**図1**はWebサーバとして有名なApacheのログ出力までの大きな流れです。

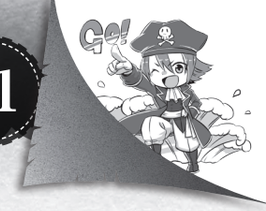
①OSからApacheが起動される
②Apacheは、起動時に設定ファイルの情報を読み込む。そこで、ログの出力先を確認する
③ログの出力先にApacheが起動した時のログ情報を出力する
④Apacheが起動完了し、Webサイトへのアクセスが可能となる。そこへ、ユーザからそのWebサイトへアクセスがあったとする
⑤ユーザからWebサイトへのアクセスをApacheは検出し、ユーザへ要求ページを返信すると同時に、日時情報(いつ)とともにユーザ情報(だれが)、要求されたページ情報(何をしたか)をログ情報として出力する

このような流れでApacheのログ情報は出力されます。Apacheのログについては、「1-2 Webサーバのログを見てみよう」で解説します。

▼図1 Apacheのログ出力までの流れ



注1) ここでは、ハードウェアとの違いを強調するためソフトウェアという表現を使っています。ここでの意味は、プログラムと同義です。
注2) プログラムの中で一般的にサービス(デーモン)として提供されているプログラムをここではアプリケーションと呼ぶことにします。



ヒント **図1**では、ログの出力先をファイルとしていますが、ログの出力先はファイルとは限りません。データベースやメールなどへ出力されることもあります。

syslogはログの基本です

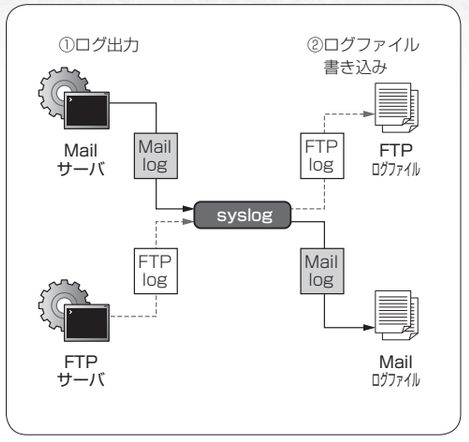
Unix系OS(Linuxも含む)では、何といってもログと言えばsyslog(シスログと呼ばれる)です。

そもそもsyslogは、Mail(SMTP:Simple Mail Transfer Protocol)サーバの代表的なアプリケーションであるsendmail[注3]のログアプリケーションとして開発されたものです。つまり、最初は、単純にMailサーバ専用のロガープログラムだったわけです。これが便利だと気づいた開発者たちが、こぞってほかのアプリケーション(FTPサーバなど)でも取り入れて、sendmailと同じようにsyslogアプリケーションを使ってログ情報の出力を行ったので、デファクトスタンダードになりました。このデファクトスタンダードになったsyslogを体系づけてRFC 3164[注4]にまとめたのがsyslogプロトコルと言われるものです。

ヒント 一般的にsyslogというと、広義のアプリケーションの総称(たとえば、以降で解説するrsyslogやsyslog-ngなどのアプリケーションを含めたもの)として使うことが多いと思っていました。しかし、最近のWeb情報ではsyslogとはプロトコルだという記事をよく見かけるようになりました。個人的に、少し違和感を感じて調べてみるとwikipediaにそれに近い表現で書かれているのを見つけました。想像ですが、その情報を元に、いろいろな方々が書かれたのではないかと思います。このような呼称は、広く普及したほうが正しくなってしまうので、今では、どっちが正しいとは言えなくなってきているのかもしれません。

注3) http://www.sendmail.com/sm/open_source/
注4) RFCとは、Request for Commentsの略で直訳すれば「コメント募集」となります。もともとは、広く意見を吸い上げる意味合いで使用されたようですが、今では少し異なり、インターネットに関する技術の標準を定める団体であるIETFが正式に発行する公開文書を意味します。RFC 3164では、The BSD syslog Protocolが定義、公開されています。

▼図2 Syslogのログ書き込みの流れ



syslogの基本機能①「ログを書き込む」

1つ目の機能はログを書き込む機能です。言い換えれば、ログの出力を管理する機能のことです。たとえば、ログの出力先がファイルの場合は、ログファイルへの書き込み、ログファイルの管理を行う機能のことです。

たとえば**図2**は、Mailサーバ(sendmailやpostfix[注5]など)やFTPサーバ(vsftpd[注6]など)からのログ情報をsyslogがログファイルへ書き込むまでの大きな流れです。

①MailサーバやFTPサーバなどでログ情報を書き込みたい場合、syslogへログ情報を渡す
②syslogはそのログ情報を受け取り、誰からのログ情報か確認し、各アプリケーション用のログファイルへ書き込む

このように非常に単純な流れです。これによってMailサーバやFTPサーバは、ログファイルへの書き込みおよびそのファイルの管理を行う必要もなく、手間が省けて大助かりというわけです。ただ、必ずsyslogがインストールされているとも限らないので、一般的なアプリケーショ

注5) http://www.postfix.org/ [日本語]http://www.postfix-jp.info/
注6) https://security.appspot.com/vsftpd.html

ンは、自前のログ情報出力・管理機能を持っています。もちろん、syslogへの出力機能も、ほとんどのアプリケーションが有しています。

ヒント Windowsでは、このsyslogのログの出力管理機能をイベントログが行っています。

syslogの基本機能②「ログを収集する」

2つ目の機能は、ログの収集管理機能で、複数のサーバのログ情報を1台のログ専用サーバで集中的に収集し管理する機能です。

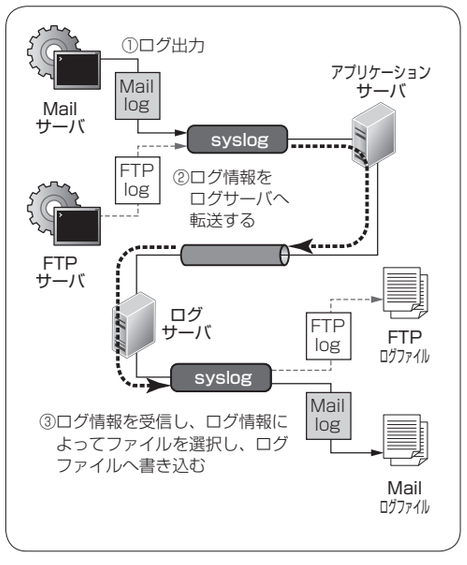
たとえば**図3**は、**図2**の例に倣ってMailサーバ(sendmailもしくはpostfixなど)や、FTPサーバ(vsftpdなど)からのログ情報をsyslog(アプリケーションサーバ)がログの専用サーバ(ログサーバ)へ送信し一括管理するという大きな流れです。

①MailサーバやFTPサーバなどでログ情報を書き込みたい場合、アプリケーションサーバ内のsyslogへログ情報を渡す
②アプリケーションサーバ内のsyslogは、そのログ情報を受け取りログサーバへ転送する
③ログサーバ内のsyslogは、そのログ情報をアプリケーションサーバから受け取り、誰からのログ情報かを確認し、各アプリケーション用のログファイルへ書き込む

このようにネットワークを介してログ情報を一ヵ所に集めて管理できます。大規模なシステムでも、この機能を利用し、ログサーバの情報管理をしっかり行えば、運用の手助けになることは間違いありません。

このようにアプリケーションを開発する側としたら、ログ情報をsyslogへポイポイと投げておけば、あとはsyslogが何とかしてくれるわけです。確かに便利ですよね。

▼図3 syslogのログ収集の流れ



syslogというアプリケーションは今や利用されていない

CentOS 5では、アプリケーションとしてのsyslogd(syslogデーモン)は、

・syslogd
・klogd (カーネルログデーモン)

の2つのアプリケーション(デーモン)をsysklogdという1つのパッケージで提供していました。しかし、syslogdは、バージョン1.5(2007年リリース)で更新が止まってしまいました[注7]。また、syslogdは、ログ情報の紛失の可能性やネットワーク上のログ情報が暗号化ができないなどのいくつかの問題が指摘されていました。

これらの問題から、とくに多くのLinuxディストリビューションでは、sysklogd(つまり、syslogd、klogd)パッケージは採用されなくなっています。

代わりに採用されているsyslogアプリケーションの後継者には、大きくsyslog-ngとrsyslogの2つがあります。以降、この2つのアプリケーションについて解説します。

Column syslogがなかった時代

syslogがなかった、あるいはアプリケーションがまだsyslogに対応していなかった時代では、ログの一括管理は自前のUDPによるファイル転送などを駆使して行っていました。その当時は、まだまだ、電話回線＋モデムでの接続がメインで、TCP接続での常時接続などは夢物語のような時代でした。UDP接続で1日に数回、コンピュータ同士を接続するのが普通の時代です。メールもそのUDP接続で1日に数回送られていた時代でもありましたから、メールのやりとりは、文字どおり郵送の手紙感覚でした。今でこそ、ほぼリアルタイムに送受信できるメールもそんな感じだったんですよ。このsyslogも例外でなく、開発された当時は、UDPがメインでした。そのため、ここでの転送もUDPがメインです。

▼写真1 スイス・アーミーナイフ

このようになんでもかんでも1つのナイフにぶら下がっている様を言いたいんでしょうか、それとも、これだけ機能が豊富だよということでしょうか。いずれにせよ、便利であることは間違いないと思います。

syslog-ngの機能とは

syslog-ng[注8]は、syslog New Generationの略で、直訳すれば「次世代syslog」ぐらいの意味でしょうか。この名前のとおり先のsyslogアプリケーションの問題を解決すべく開発されたアプリケーションです。syslogプロトコルのサポートはもちろんのこと、次のような主だった機能が追加されています。

・ログの分類機能
・TCPによるログ情報の送受信(ログ情報の紛失の回避)
・SSL/TLSを使用してセキュアログ(ネットワーク経由の暗号化の実現)
・データベースへのログ出力

などさまざまな機能が盛り込まれています。

rsyslogの機能とは

rsyslog[注9]は、rocket-fast system for log processingの略で、直訳すれば「猛烈に早いsyslog」ぐらいの意味でしょうか。そもそもrsyslogは、標準のsyslogdの後継として始まりましたが、多種多様なソースコードからの入力を変換、結果をさらに多様な出力先への書き込みを可能にすることでロギングのスイス・アーミーナイフのようなもの(**写真1**)へと進化しました(このキャプションは公式サイトの訳です)。

先のsyslog-ng同様、rsyslogもsyslogプロトコルのサポートはもちろんのこと、次のような主だった機能が追加されています。

・ログの分類機能
・TCPによるログ情報の送受信(ログ情報の紛失の回避)
・SSL/TLSを使用してセキュアログ(ネットワーク経由の暗号化の実現)
・データベースへのログ出力

ヒント Web情報でrsyslogを検索すると、rsyslogは、reliable syslog(信頼性の高いsyslogの意味)の略だというページを多く見つけました。reliable syslogを目指したのは確かですが、名前の由来は、公式サイトで上記のように記載があったので、プロトコルと混同されているのではないかと思います。

CentOS 6では、このrsyslogが採用されています。また、Debianでもrsyslogが採用されており、主要な2つのLinuxディストリビューションで採用されたことによって、rsyslogが今やデファクトスタンダードになりつつあると言っても良いかもしれません。

注7) http://freecode.com/projects/sysklogd
注8) ライセンス:LGPL(core部)、LGPLv2(plugin部)http://www.balabit.com/network-security/syslog-ng/opensource-logging-system
注9) ライセンス:GPLv3(http://www.rsyslog.com)

ここまで、ログ、syslogの基本的な機能を解説してきました。ここからは、実際にCentOS 6を使って具体的な設定、使い方を解説します。また、以降syslogと表記する場合は、狭義の意味でのrsyslogを指すのではなく、広義の意味でsyslog対応アプリケーションの意味として表記することに注意してください。

syslog へコマンドを使って出力してみよう

ここでCentOS 6 + rsyslog環境で実践することで、まずはsyslogが、どのようなものかを肌で感じてもらおうと思います。CentOS 6は、2014年5月現在の最新版6.5です。rsyslogのバージョンは、5.8.10です。また、CentOS 6は便宜上リモートアクセス(SSH接続)して使用します。SSH接続するためには、SSHサーバ(openssh-server)がCentOS側で起動していなければいけません。

もし、SSHサーバ(openssh-server)がインストールされていないようなら、サーバのコンソール画面から次の要領で簡単にインストールできますので、インストールしてみましょう。

```
$ yum install openssh-server
……(省略)……
Is this ok [y/N]: y
……(省略)……
Complete!
```

インストールを終えたら、起動しておきましょう。

```
$ /etc/init.d/sshd start
sshd を起動中:                             [  OK  ]
```

リモートアクセスしてみよう

まずは、CentOSへログインしてみましょう。SSHによるリモートログインを行うには、Macでは、Mac OS Xターミナルを使うと良いでしょう。Windowsのターミナルソフトは、コマンドプロンプトです。そもそもデフォルトで**ssh**コマンドが存在しませんので、**ssh**コマンドのインストールを行うか、別のターミナルソフトを使うことになります。ここではターミナルソフトのTeraTermを使ってリモートログインを行ってみます(TeraTermのインストールおよび設定は多くのWebサイトで紹介されていますので、ここでは割愛します)。

コマンドを使ってみよう

まずは、簡単な**ls**コマンド(Windowsでいうところのdirコマンド)を使ってみます(**図4**)。

パラメータの"**-l**"はリスト出力、"**-a**"は、すべてを意味します。実行すると上記のようにすべての情報をリスト形式で出力します。

次に**ps**コマンドを使って、syslogのプロセスを確認してみます(**図5**)。

psコマンドは、現在のプロセスの状態を出力するコマンドです。"**x**"は、呼び出したユーザの所有する全プロセスを出力するという意味で、"**a**"は、端末(tty)を持つすべてのプロセスをリストで出力するという意味になります。ちょっとわかり難いですが、"**ax**"を指定することで全プロセスを出力してくれると覚えてお

▼図4　lsコマンドを試す

```
$ ls -al
合計 20
drwx------. 2 hoge hoge 4096  5月 25 06:33 2014 .
drwxr-xr-x. 3 root root 4096  5月 25 06:33 2014 ..
-rw-r--r--. 1 hoge hoge   18  7月 18 22:15 2013 .bash_logout
-rw-r--r--. 1 hoge hoge  176  7月 18 22:15 2013 .bash_profile
-rw-r--r--. 1 hoge hoge  124  7月 18 22:15 2013 .bashrc
……(省略)……
```

Column ログはコンピュータの行動記録

ログとは、そもそも英語ではlogと書きます。英和辞典を調べてみると最初に出てくるのが、丸太、材木を切り出すという意味です。次が、測程儀(船の速度を測る器具)や航海(航路)日誌(**写真2**)に記入するなどの意味が出てきます。この2つは、まったく意味が異なるもののようですが、これらを結びつけるものは"船"であり"測程儀"です。船の速度を測る器具(測程儀)に手用測程儀(hand log)というものがあります。これは、木片に長い紐を括り付けた簡単な道具です。使い方も簡単で、木片を海に浮かべ、紐を船上から垂らし、その紐がスルスルと簡単に流れ出るようにして船を走らせるだけです。一定時間内に紐がどれだけ流れ出たかで船の速度を測定するというものです。船の速度でノット(英語：knot、日本語：結び目)という単位が使われるのは、この紐に一定間隔で結び目をつけ、先の計測方法で流れ出た結び目の数を船の速度としたことに由来します。この木片が丸太(log)であり、船の計測が航海日誌(logbook)へと結びついたとされています。

また、コンピュータ、とくにプログラミングの世界では、プログラムが経時(あるいは処理経過)ごとに(「いつ、だれが、どこで、何をした」という情報を)記録することを「ロギングする」、記録したものを「ログ」と呼びます。英語で"記録"はrecordという単語を使うことが多いですが、まさにlogが使われているのは、航海日誌(logbook)のように時間経過(あるいは作業経過)とともに記録を残したことにたとえてのことだと言われています。

さて、本題に入りましょう。なぜログが必要なのでしょう。

それは、大きく2つの理由があります。1つは、不具合の修理、改善のためです。コンピュータシステムに完璧なものはありません。むしろ、コンピュータシステムほどよく壊れるものはないとさえ言えるかもしれません。今でこそ、少なくなりましたが、パソコンが固まる(突然、動かなくなる)のは日常茶飯事でした。もし、コンピュータシステムが完璧で壊れないものであれば、そのような過去を記録した情報(ログ)は、必要ないのかもしれません。しかし、コンピュータシステムに限らず完璧なもの、壊れないものはありません。安全神話が妄想であるように、それを正しく理解していれば壊れたときにどうしようかと考えるでしょう。もし壊れれば、その原因を究明し、修復・修繕し、二度と同じような壊れ方をしないように(英語にもなった)改善(カイゼン)を図るでしょう。その原因究明には、壊れた時の状態・情報が非常に大事です。その貴重な情報がログです。そのログが経時的(あるいは経過的)に記録された情報であることから、壊れた時に何が起こったか、ハードウェア・ソフトウェアを含めてシステムの状態を時間をさかのぼって把握することができるのです。その昔、大航海時代には、航海日誌が安全な航海のための貴重な情報だったように、このログも安定したシステム運用のための貴重な情報なのです。

もう1つは、昨今、注目されているビッグデータに代表される痕跡情報(アクセス情報)としてのログの必要性が高まったことにあるでしょう。これは、ログの特徴である「いつ、だれが、どこで、何をした」という情報から、人の動向や意識、マーケティングの調査などのためのデータマイニングの元データとして用いられたりします。具体的に身近な例として、Webサイトのアクセスログの解析があります。Webサイトのアクセスログ解析では、どのページから入ってきて、どのページで離脱したか、どの地域の人がどれくらいアクセスしたかなどさまざまな分析が行われます。その分析結果は、より人が興味を持つようなページ作りや人を見せたいページへどのように導いていくか、いわゆる導線の張り方を考える材料などに用いられたりします。これらは、先の不具合の修理のように過去のデータから現在を改善するのではなく、過去のデータから未来を予測(改善)するという、同じログの情報でも活用の範囲を広げた1つの解析(分析)方法でもあります。最近では、インターネットの発展とともに、これらの情報活用が非常に注目を集めているだけに、ログというとこちらのイメージが強い方も多いようです。このように、ログが、さまざまな切り口で利用され、改修、改善を図る貴重なデータであることは間違いありません。

▼写真2　大航海時代の海図