概念を理解する

～まず細かいことはおいておこう～

情報収集

コピペで行ける世の中　検索技術を競う

パーミッション

ディレクトリ構成

ログ

ログの見方

デーモン

etc.

etc.

～では細かいことはどうなのかな？～

パーミッション（アクセス権）の概念

コピペ元・・・http://begi.net/linux/reading/

パーミッション・・・ファイルやディレクトリに対する操作に権限を与　え、ファイルの保護モードを設定することです。

「読み込み」、「書き込み」、「実行」の 3 種類があり、 それぞれをファイルの「所有者」、「グループユーザー」、「その他のユーザー」に対して設定できます。

パーミッションの確認をするには ls コマンドにオプション -l をつけて確 認します。

----------例----------

$ ls -l /etc/xxxx

-r--------　　1 root　　root　　1085　 8 月 1 日 1:27 /etc/xxxx

ファイルの種別と保護モード（パーミッション）

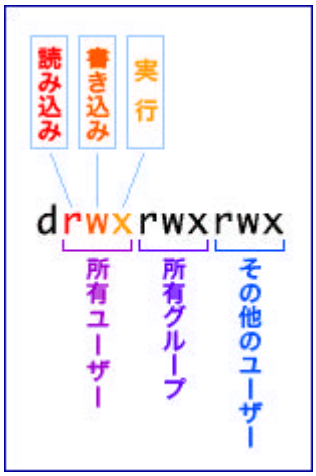
ファイルを所有するユーザー

ファイルの属するグループ

ファイルのサイズ

ファイルの最終更新日時

ファイル名



パーミッションを表しているのは、表示されたうちの一番左にある 10 文 字です。一番左１つと、あとは３つの文字の組み合わせが３つという構成です。

drwxrwxrwx

一番左の d ：ディレクトリかどうか

r ：読み込み(Read)パ ーミッション

w ：書き込み (Write)パーミッション

x ：実行(eXecute)パーミッション

(引用：http://begi.net/linux/reading/)

rwx のセットが 3 つあるのは、それぞれの効果が及ぶ範囲が異なっているから

一番左のセット：「所有 ユーザー」

真ん中のセット：「所有 グループ」

一番右のセット：「その他のユーザー」

●パーミッションの変更

パーミッションの変更ができるのは、そのファイル（ディレクトリ）の所有者、もしくは root 権限者に限られます。パーミッションの変更には　chmod コマンドを使います。

chmod　モード　ファイル(ディレクトリ）名

chmod コマンドの引数の「モード」には、「誰に」「何を」「許可するか禁止 するか」という情報を与えます。

「誰に」	「許可するか禁止するか」	「何を」
u:所有ユーザー(user) g:所有グループ(group) o:その他のユーザー(others) a:全て(all)	+:許可する -:禁止する	r：読み込み(read) w：書き込み(write) x：実行(execute)

●chmod の組み合せ例

u+x　　所有ユーザーに実行を許可

a+rwx　全てのユーザーに全て許可

go-rwx　所有ユーザー以外全て禁止

g+r　　所有グループに読み込みを許可

●ディレクトリのパーミッションについて

ディレクトリのパーミッションの意味は、ファイルと異なり以下のようになる

ｒ：読み込み (Read)

そのディレクトリの情報を見ることができるか （その ディレクトリに対して ls コマンドが実行できるか ただ し、「実行(x)」パーミッションが設定されていないと「-l」 オプションは使えません）

w：書き込み (Write)

そのディレクトリに新しいファイルを作成することがで きるか （同時に「実行(x)」パーミッションの設定が必要 です）

x：実行 (eXecute)

そのディレクトリをカレントディレクトリにできるか （そ のディレクトリに対して cd コマンドが実行できるか）

●パーミッションのもうひとつ設定方法

r、w、x を 2 進数 に置き換え、さらに 8 進数に変換し、数字を用いる方法

モード	---	--x	-w-	-wx	r--	r-x	rw-	rwx
8 進数	0	1	2	3	4	5	6	7

例）パーミッションの確認のために作った、「newfile」のパーミッショ ンを全てのユーザーが全て許可、と設定する場合

$ chmod 777 newfile　パーミッションの設定

$ ls -l newfile　パーミッションの確認

-rwxrwxrwx　1 user　user　11　1 月 17 17:40 newfile

(-rwxrwxrwx = 777)

数字 3 桁でパーミッションを表示し、左側が「所有ユーザー」、真ん中が 「グループユーザー」、右側が「その他のユーザー」を表します。

●ユーザー・グループの変更

chown コマンドで変更します.

chown　ユーザー名.グループ名　ファイル(ディレクトリ）名

例

chown goemon testfile　(testfile の所有者を goemon に変更)

chown goemon.eznetfan testfile　(testfile の所有者を goemon にグループを eznetfan に変更)

chown .eznetfan testfile　(testfile のグループを eznetfan に変更)

●グループの変更

chogrp コマンドで変更します.

chgrp　グループ名　ファイル(ディレクトリ）名

●タイムスタンプの設定

touch コマンド（詳細は省略）

・・・・・・・・・・・・・・・・引用・・・・・・・・・・・・・・・・・・

びぎねっと ＨＰ　（http://begi.net/linux/reading/）

図解でわかる Linux の全て　西村めぐみ著　日本実業出版社

ディレクトリの概念

引用元・・・ http://www.atmarkit.co.jp/flinux/index/indexfiles/index-linux.html

「Windowsユーザーに教えるLinuxの常識」第2回 各ディレクトリの役割を知ろう（ルート編），第3回 各ディレクトリの役割を知ろう（サブ編）

FHS 2.2 におけるディレクトリの構成

/		ルートディレクトリ
├	/bin	基本コマンド
├	/boot	起動に必要なファイル
├	/dev	デバイスファイル
├	/etc	設定ファイル
├	/home	（オプション） ユーザーのホームディレクトリ
├	/lib	共有ライブラリ
├	/lib<qual>	（オプション）
├	/mnt	一時的なマウントポイント
├	/opt	追加アプリケーション
├	/proc	（Linux固有）プロセス情報など
├	/root	（オプション）root用ホームディレクトリ
├	/sbin	システム管理用コマンドなど
├	/tmp	一時的なファイル
├	/usr	各種プログラムなど
└	/var	変更されるデータ

・・・・・・・・・引用元には非常に詳しく載っていますが簡潔にまとめます・・・・・・・・・・・・・・・・・・・・・・・・・・

■/bin

システム管理者と一般ユーザーの両方が使う、極めて基本的なコマンドが入っています。ほかのファイルシステムがマウントされていない、シングルユーザーモードでも一通りの作業が行えるコマンド群です。

■/boot

ブート時に必要なファイルは、このディレクトリに配置されます。初期のLinuxには存在しなかったディレクトリで、/bootはカーネルの再構築を行うとき以外、触る必要のないディレクトリです。

■/dev

/devにはデバイスファイルが配置されています。コンソール画面に何かを出力するなら、/dev/consoleに書き込めばいいわけです。

Linuxの特徴として、「各種のデバイスもファイルとして扱う」ということがよくいわれます。もっとも、これはUNIXの特徴だったのですが、現代のOSは大なり小なりUNIXの影響を受けているためか、割と一般的な機能です。

よほどのことがない限り、/dev内のファイルを変更する必要はありません。訳の分からないうちにいじると、デバイスがグシャグシャになったりするので注意が必要です。

■/etc

さまざまな設定ファイルは、ここ/etcにあります。かなりの数のファイルがあるので、最初は戸惑うかもしれません。しかし、一度にすべてが必要になるわけではないので、1つずつ確認していっても間に合います。関連する設定ファイルが多い場合は、/etc以下にサブディレクトリを作ってその中に配置することもあります。

■/home

/homeは、各ユーザーのホームディレクトリがある場所です。概して容量を必要とするので、/usrの下にシンボリックリンクが張ってあったりします。できれば、このディレクトリは独立したパーティションにしておいた方がいいでしょう（前回のパーティション分割／非分割のセオリー参照）。

■/lib

カーネルのブート時に必要なものと、/ファイルシステムにあるコマンド（/binや/sbin内のコマンド）を実行するのに必要なライブラリはここにあります。

■/lost+found

fsckでディスクをチェックしたときに作られる、破損ファイルの断片を収めるディレクトリです。ルートディレクトリだけでなく、あちこちにあります。とはいえ、普通のユーザーには、ここに残っている情報から元のファイルを復元するのはまず無理でしょう。/lost+foundディレクトリの内容にこだわるよりも、正常なファイルのバックアップをマメに取っておく方が重要です。

■/mnt

一時的にファイルシステムをマウントするためのディレクトリです。ディストリビューションによって、/mntだけの場合と、その下にcdromやfloppyを含んでいる場合があります。一般のユーザーが使うことはなく、root専用と考えた方がいいでしょう。

■/opt
RPM や dpkg といったパッケージ管理システムでプログラムをインストールする場所です。
■/proc
カーネル内部の情報にアクセスするためのファイルが集まっています。CPU や PCI バス、そして文字どおり各種プロセスの情報を読み出せます。
　注：/proc の内容は「ファイル」と呼ばれるが、通常のファイルとは異なり HD 容量を消費することはない。つまり、HD 上にそのようなファイルは存在しないのである。
■/root
　スーパーユーザーのホームディレクトリです。昔はルートディレクトリがそのままスーパーユーザーのホームディレクトリでしたが、「見通しが悪い」「セキュリティ上問題がある」といった理由で専用のディレクトリである /root ができました。/home と別なのは、そちらが壊れても root だけは作業できるようにするためです。
■/sbin
/sbin は、ブートやシステムのリカバリーに必要なシステム標準コマンドが収められています。/bin と違うのは、root が使用するシステムメンテナンス系のコマンドが集まっていることです。
■/tmp
　その名のとおり、一時的な作業用のディレクトリです。リブート時にきれいに掃除されます。だれでも使えますが、持続性を要求されるファイルを置くべきではありません。
■/usr
　ユーザー向けのディレクトリで、多くのサブディレクトリを含んでいます。サブディレクトリの 1 つである/usr/local は、パッケージ管理システムの管轄外となっています。自分でソースからコンパイルしたプログラムなどは、この /usr/local 下に配置するのが一般的です。
　詳しいことは、次回解説します。
■/var
　プリントやメール、ネットニュースのスプール、キャッシュといった作業用エリア、ログファイルなど、変化していく（variable）ファイルを配置するディレクトリです。/tmp と違って、リブートしても削除されません。

詳しくは下記を参照下さい
http://www.atmarkit.co.jp/flinux/index/indexfiles/index-linux.html

ログの概念

syslog.conf の設定を参照　/var/log

引用元：ログをとろう！（kozupon.com）
http://www.kozupon.com/log/log.html

１．不審アクセスを受けた生ログの実例
２．Bind の生ログ　　（省略）
３．自サイトの安全性チェック方法（省略）
４．logrotate の活用（省略）

---

xxx.xxx.xxx.xxx=IPアドレスです

１．不審アクセスを受けた生ログの実例

●　ポートスキャンをくらっている場合

以下、/var/log/messagesの内容

Jun 24 17:54:31 server01 popper[24940]: connect from xxx.xxx.xxx.xxx

Jun 24 17:54:32 server01 in.rlogind[24942]: connect from xxx.xxx.xxx.xxx

Jun 24 17:54:32 server01 in.ftpd[24943]: connect from xxx.xxx.xxx.xxx

Jun 24 17:54:32 server01 in.fingerd[24944]: connect from xxx.xxx.xxx.xxx

Jun 24 17:54:32 server01 in.telnetd[24946]: connect from xxx.xxx.xxx.xxx

Jun 24 17:54:32 server01 in.rshd[24945]: connect from xxx.xxx.xxx.xxx

こんなのが、来たら１００%ポートスキャンくらってます！

あと、nmapでオプションなしのポートスキャンを食らうと、

Jan 22 22:26:34 main2 sshd[87060]: Did not receive ident string from 198.*.***.***.**

●　telnet認証失敗

Jun 24 18:04:24 server01 in.telnetd[24988]: connect from xxx.xxx.xxx.xxx

Jun 24 18:04:29 server01 login: FAILED LOGIN 1 FROM xxx.xxx.xxx.xxx FOR root, Authentication failure

こんなのが有ったら、telnet認証に失敗してます。不正侵入に失敗というわけですねえ。

テキスト形式のログを見るときには、FAIL や ERROR という文字列を常に気にすること！

● **第三者不正中継の場合**

/var/log/maillog ファイルで、「reject」という文字列が目に付いたら、中継を試みられていることになります。こんな感じ。

```
Jun  24 18:23:40 server01 sendmail[25010]: SAA25010: ruleset=check_rcpt, arg1=<foo
@hogehoge.co.jp>, relay=ns.hogehoge.co.jp [xxx.xxx.xxx.xxx], reject=553 <foo@hoge
hoge.co.jp>... Relay operation rejected
```

● **その他の記録の見方**

(詳細は引用元のHPを参照してください)

４．logrotateの活用

ログは、放っておくといくらでも大きくなる。したがって、定期的にログをリセットする必要がある。
しかし、すぐにログを消してしまうと後で不正侵入が判明した場合の調査に利用できなかったりするので、少し昔のものでもとっておいた方が良いであろう。そういう場合に便利なのがこの logrotate である。logrotate は、定期的にログをリセット、３回前のログまで保存する、圧縮してログを保存する、などログに関しての様々な機能を提供する。
logrotate は、Linux で標準にインストールされる。logrotate の設定ファイルは、/etc/logrotate.conf にある。以下に示す。
(設定などの詳細は引用元のHPを参照してください)
-

３．自サイトの安全性チェック方法に行く前に．．

最低チェックが必要なlogファイルとしては、

/var/log/maillog	**mailサーバの動作記録**
/var/log/messages	**様々なプログラムの動作記録**
/var/log/secure	**ユーザ認証関連の記録**

３．自サイトの安全性チェック方法

まず、小生のサーバのsyslog.confの内容。

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
(日)*.info;mail.none;news.none;authpriv.none            /var/log/messages

# The authpriv file has restricted access.
(月)authpriv.*                                          /var/log/secure

# Log all the mail messages in one place.
(火)mail.*                                              /var/log/maillog

# Everybody gets emergency messages, plus log them on another
# machine.
(水)*.emerg                                             *

# Save mail and news errors of level err and higher in a
# special file.
# uucp,news.crit                                        /var/log/spooler

# Save boot messages also to boot.log
(木)local7.*                                            /var/log/boot.log

#
# Popper
#

#local0.notice                                          /var/log/popperlog

#
# Router
#
(金)user.notice;user.info                               /var/log/router
#
# Mail
#
(土)mail.info                                           -/var/log/mail.info
(祭)mail.warn                                           -/var/log/mail.warn
(祝)mail.err                                            /var/log/mail.err
```

(日)は、メール、プライベート認証、cron以外の報告以上の全てを/var/log/messagesに記録する。

(月)は、プライベート認証関連を/var/log/secureに記録する。

(火)は、メール関連のログを/var/log/maillogに記録する。

(水)は、重大な問題をログイン中の全員に知らせる。

(木)は、ブートメッセージを/var/log/bootlogに記録する。

(金)は、ルータの情報を/var/log/routerに記録する。

(土)は、メールの情報を/var/log/mail.infoに記録する。

(祭)は、メールのワーニング情報を/var/log/mail.warnに記録する。

(祝)は、メールエラー情報を /var/log/mail.err に記録する。

FTP 不正アクセス例（じゃないのもあるけど）　　　　抜粋

7:43 PM　02.1.6　80.11.49.204　Ngpuser@home.com　39　ログイン　Guest
7:44 PM　02.1.6　80.11.49.204　Ngpuser@home.com　39　ログアウト　Guest
0:52 AM　02.2.9　218.43.23.224Anonymous　1　拒否されたログイン　no anonymous access allowed
1:13 PM　02.2.10　218.43.23.224Anonymous　2　拒否されたログイン　no anonymous access allowed
11:06 AM　02.7.4　62.5.52.111　Anonymous　9　拒否されたログイン　no anonymous access allowed
11:14 AM　02.7.4　62.5.52.111　Anonymous　10　拒否されたログイン　no anonymous access allowed
11:58 AM　02.7.4　62.5.52.111　Anonymous　11　拒否されたログイン　no anonymous access allowed
0:53 PM　02.7.4　62.5.52.111　Anonymous　12　拒否されたログイン　no anonymous access allowed
1:02 PM　02.7.4　62.5.52.111　Anonymous　13　拒否されたログイン　no anonymous access allowed
6:31 PM　02.7.7　80.11.137.155Anonymous　15　拒否されたログイン　no anonymous access allowed
3:21 AM　02.7.8　217.227.70.212　Anonymous　24　拒否されたログイン　no anonymous access allowed
1:26 PM　02.7.11　217.235.113.29　Anonymous　4　拒否されたログイン　no anonymous access allowed

CodeRed アクセス例　　抜粋

07/18/2002　14:54:47　OK　pd9541d17.dip.t-dialin.net　0
132　200　Configuration
!!CodeRedKiller-PLUGIN 07/18/02　16:01:11　WARNING Code Red !!!!!!!　4039 bytes received.
!!CodeRedKiller-PLUGIN　GET
/default.ida?NN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0
Content-type: text/xml
HOST:www.worm.com
 Accept: */*
Content-length: 3569

U駆 A*

.不明アクセス+α　　抜粋

.07/29/2002　19:10:26　ERR!　210.78.157.248　/scripts/root.exe　522
GET　/c+dir　2809　www　404　Configuration

07/29/2002　19:10:32　ERR!　210.78.157.247　/MSADC/root.exe　522
GET　/c+dir　72　www　404　Configuration

07/29/2002　19:10:36　ERR!　210.78.157.239　/c/winnt/system32/cmd.exe　522
GET　/c+dir　71　www　404　Configuration

07/29/2002　19:11:24　ERR!　210.78.157.228　/d/winnt/system32/cmd.exe　522
GET　/c+dir　2808　www　404　Configuration

07/29/2002　19:12:16　ERR!　210.78.157.246　/scripts/..%255c../winnt/system32/cmd.exe　522
GET　/c+dir　2917　www　404
Configuration
途中省略
07/29/2002　19:17:23　ERR!　210.78.157.229　/scripts/..%252f../winnt/system32/cmd.exe　522
GET　/c+dir　31　www　404
07/30/2002　01:58:54　OK　65.195.133.22　0
2991　200　Configuration
07/30/2002　03:59:52　OK　216.158.225.158　0
2943　200　Configuration
07/30/2002　07:20:29　OK　141.155.241.98　0
162　200　Configuration
07/31/2002　12:17:15　OK　b067135.adsl.hansenet.de　/　1523　GET
5　200　Configuration
07/31/2002　12:42:38　OK　user-119be5a.biz.mindspring.com　0
129　200　Configuration
07/31/2002　22:06:08　OK　adam.nni.com　0
41　200　Configuration

その他．．検索エンジンのアクセス例　　抜粋

08/01/2002　10:10:56　ERR!　66.35.208.59　/robots.txt　524　Mozilla/4.0　compatible　ZyBorg/1.0
(wn.zyborg@looksmart.net; http://www.WISEnutbot.com)　GET　2914
foram.jp　404　Configuration

08/01/2002　12:30:02　OK　65.116.145.45/toyoho/　1030　Mozilla/4.0　compatible　ZyBorg/1.0
(wn.zyborg@looksmart.net; http://www.WISEnutbot.com)　GET　2915
foram.jp　200　Configuration
08/01/2002　12:31:49　OK　65.116.145.45/　1511　Mozilla/4.0　compatible　ZyBorg/1.0
(wn.zyborg@looksmart.net; http://www.WISEnutbot.com)　GET　2813
foram.jp　200　Configuration
08/01/2002　12:40:13　OK　crawl5.googlebot.com　/toyoho/ikikata/dani1.html647　googlebot(at)googlebot.com
Googlebot/2.1 (+http://www.googlebot.com/bot.html)　GET　38
foram.jp　200　Configuration

ごえもん　ログの例
/var/log/boot.log

起動時

```
Jul 25 21:37:58 goemon syslog: syslogd startup succeeded
Jul 25 21:37:58 goemon syslog: klogd startup succeeded
Jul 25 21:37:59 goemon portmap: portmap startup succeeded
Jul 25 21:37:59 goemon nfslock: rpc.statd startup succeeded
Jul 25 21:38:00 goemon autofs: autofs startup succeeded
Jul 25 21:38:00 goemon random: Initializing random number generator:  succeeded
Jul 25 21:37:36 goemon rc.sysinit: Mounting proc filesystem:  succeeded
Jul 25 21:37:36 goemon sysctl: net.ipv4.ip_forward = 1
Jul 25 21:37:36 goemon sysctl: net.ipv4.tcp_syncookies = 1
Jul 25 21:37:36 goemon sysctl: net.ipv4.conf.default.rp_filter = 1
Jul 25 21:37:36 goemon sysctl: kernel.core_uses_pid = 1
Jul 25 21:37:36 goemon rc.sysinit: Configuring kernel parameters:  succeeded
Jul 25 21:37:36 goemon date: Thu Jul 25 21:37:30 JST 2002
Jul 25 21:38:02 goemon netfs: Mounting other filesystems:  succeeded
Jul 25 21:37:36 goemon rc.sysinit: Setting clock  (localtime): Thu Jul 25 21:37:30 JST 2002 succeeded
Jul 25 21:37:36 goemon rc.sysinit: Loading default keymap succeeded
Jul 25 21:37:36 goemon rc.sysinit: Activating swap partitions:  succeeded
Jul 25 21:37:36 goemon rc.sysinit: Setting hostname goemon.ep.sci.hokudai.ac.jp:  succeeded
Jul 25 21:37:36 goemon rc.sysinit: Mounting USB filesystem:  succeeded
Jul 25 21:37:36 goemon fsck: /: clean, 100976/274176 files, 451621/548352 blocks
Jul 25 21:38:03 goemon apmd: apmd startup succeeded
Jul 25 21:37:36 goemon rc.sysinit: Checking root filesystem succeeded
Jul 25 21:37:36 goemon rc.sysinit: Remounting root filesystem in read-write mode:  succeeded
Jul 25 21:37:38 goemon rc.sysinit: Finding module dependencies:  succeeded
Jul 25 21:38:03 goemon identd: identd startup succeeded
Jul 25 21:37:38 goemon modprobe: Warning: loading /lib/modules/2.4.18-0vl3/kernel/drivers/video/encode-eucjp.o will taint the
kernel: no license
Jul 25 21:37:38 goemon modprobe:   See http://www.tux.org/lkml/#s1-18 for information about tainted modules
Jul 25 21:37:38 goemon modprobe: Module encode-eucjp loaded, with warnings
Jul 25 21:37:38 goemon rc.sysinit: Loading unicon module (encode-eucjp):  succeeded
Jul 25 21:37:38 goemon fsck: /boot: clean, 34/13104 files, 13248/52384 blocks
Jul 25 21:38:04 goemon rc: Starting pcmcia:  succeeded
Jul 25 21:37:38 goemon rc.sysinit: Checking filesystems succeeded
Jul 25 21:37:38 goemon rc.sysinit: Mounting local filesystems:  succeeded
Jul 25 21:37:38 goemon rc.sysinit: Enabling local filesystem quotas:  succeeded
Jul 25 21:37:39 goemon rc.sysinit: Enabling swap space:  succeeded
Jul 25 21:37:43 goemon rc: Starting murasaki:  succeeded
Jul 25 21:38:04 goemon inet: inetd startup succeeded
Jul 25 21:37:43 goemon kudzu: Updating /etc/fstab succeeded
Jul 25 21:37:52 goemon kudzu:  succeeded
Jul 25 21:37:52 goemon sysctl: net.ipv4.ip_forward = 1
Jul 25 21:38:05 goemon sshd: Starting sshd:
Jul 25 21:37:52 goemon sysctl: net.ipv4.tcp_syncookies = 1
Jul 25 21:37:52 goemon sysctl: net.ipv4.conf.default.rp_filter = 1
Jul 25 21:37:52 goemon sysctl: kernel.core_uses_pid = 1
Jul 25 21:37:52 goemon network: Setting network parameters:  succeeded
Jul 25 21:37:53 goemon network: Bringing up loopback interface:  succeeded
Jul 25 21:37:55 goemon network: Bringing up interface eth0:  succeeded
Jul 25 21:38:07 goemon sshd:  succeeded
Jul 25 21:37:58 goemon network: Bringing up interface eth1:  succeeded
Jul 25 21:38:07 goemon sshd:
Jul 25 21:38:07 goemon rc: Starting sshd:  succeeded
Jul 25 21:38:09 goemon lpd: lpd startup succeeded
Jul 25 21:38:09 goemon keytable: Loading keymap:
Jul 25 21:38:09 goemon keytable: Loading /usr/lib/kbd/keymaps/i386/qwerty/jp106.kmap.gz
Jul 25 21:38:09 goemon keytable: Loading system font:
Jul 25 21:38:09 goemon rc: Starting keytable:  succeeded
Jul 25 21:38:10 goemon postfix: Starting postfix:
Jul 25 21:38:14 goemon postfix: postfix
```

```
Jul 25 21:38:14 goemon rc: Starting postfix:  succeeded
Jul 25 21:38:14 goemon gpm: gpm startup succeeded
Jul 25 21:38:17 goemon httpd: httpd startup succeeded
Jul 25 21:38:17 goemon FreeWnn: Starting FreeWnn:
Jul 25 21:38:18 goemon FreeWnn:
Jul 25 21:38:18 goemon FreeWnn:
Jul 25 21:38:18 goemon FreeWnn: Nihongo Multi Client Server (FreeWnn 1.1.0pl18)
Jul 25 21:38:18 goemon FreeWnn: Reading /etc/FreeWnn/ja/dic/pubdic/kihon.dic^I Fid = 1
Jul 25 21:38:18 goemon FreeWnn: Reading /etc/FreeWnn/ja/dic/pubdic/setsuji.dic^I Fid = 2
Jul 25 21:38:18 goemon FreeWnn: Reading /etc/FreeWnn/ja/dic/pubdic/koyuu.dic^I Fid = 3
Jul 25 21:38:18 goemon FreeWnn: Reading /etc/FreeWnn/ja/dic/pubdic/chimei.dic^I Fid = 4
Jul 25 21:38:18 goemon FreeWnn: Reading /etc/FreeWnn/ja/dic/pubdic/jinmei.dic^I Fid = 5
Jul 25 21:38:18 goemon FreeWnn: Reading /etc/FreeWnn/ja/dic/pubdic/special.dic^I Fid = 6
Jul 25 21:38:18 goemon FreeWnn: Reading /etc/FreeWnn/ja/dic/pubdic/computer.dic^I Fid = 7
Jul 25 21:38:18 goemon FreeWnn: Reading /etc/FreeWnn/ja/dic/pubdic/symbol.dic^I Fid = 8
Jul 25 21:38:18 goemon FreeWnn: Reading /etc/FreeWnn/ja/dic/pubdic/tankan.dic^I Fid = 9
Jul 25 21:38:18 goemon FreeWnn: Reading /etc/FreeWnn/ja/dic/pubdic/bio.dic^I Fid = 10
Jul 25 21:38:18 goemon FreeWnn: Reading /etc/FreeWnn/ja/dic/gerodic/g-jinmei.dic^I Fid = 11
Jul 25 21:38:18 goemon FreeWnn: Reading /etc/FreeWnn/ja/dic/pubdic/full.fzk^I Fid = 12
Jul 25 21:38:18 goemon FreeWnn: Finished Reading Files
Jul 25 21:38:18 goemon rc: Starting FreeWnn:  succeeded
Jul 25 21:38:18 goemon crond: crond startup succeeded
Jul 25 21:38:20 goemon xfs: xfs startup succeeded
Jul 25 21:38:20 goemon canna: Starting Canna server:
Jul 25 21:38:21 goemon canna:
Jul 25 21:38:21 goemon rc: Starting canna:  succeeded
Jul 25 21:38:21 goemon anacron: anacron startup succeeded
Jul 25 21:38:21 goemon atd: atd startup succeeded
Jul 25 21:38:23 goemon firewall: /etc/rc5.d/S99firewall: line 958: syntax error: unexpected end of file
Jul 25 21:38:23 goemon rc: Starting firewall:  failed
Jul 25 21:38:24 goemon start: Starting Webmin server in /usr/share/webmin
Jul 25 21:38:26 goemon webmin: Starting Webmin:  succeeded
```

終了時

```
Jul 25 21:35:18 goemon stop: Stopping Webmin server in /usr/share/webmin
Jul 25 21:35:18 goemon webmin: Stopping Webmin:  succeeded
Jul 25 21:35:19 goemon atd: atd shutdown succeeded
Jul 25 21:35:19 goemon rc: Stopping keytable:  succeeded
Jul 25 21:35:19 goemon xfs: xfs shutdown succeeded
Jul 25 21:35:19 goemon canna: Shutting down Canna server:
Jul 25 21:35:19 goemon canna:
Jul 25 21:35:19 goemon rc: Stopping canna:  succeeded
Jul 25 21:35:19 goemon gpm: gpm shutdown succeeded
Jul 25 21:35:20 goemon httpd: httpd shutdown succeeded
Jul 25 21:35:20 goemon sshd: sshd -TERM succeeded
Jul 25 21:35:20 goemon postfix: Shutting down postfix:
Jul 25 21:35:20 goemon postfix: postfix
Jul 25 21:35:20 goemon rc: Stopping postfix:  succeeded
Jul 25 21:35:20 goemon inet: inetd shutdown succeeded
Jul 25 21:35:20 goemon crond: crond shutdown succeeded
Jul 25 21:35:21 goemon lpd: lpd shutdown succeeded
Jul 25 21:35:21 goemon identd: identd shutdown succeeded
Jul 25 21:35:22 goemon apmd: apmd shutdown succeeded
Jul 25 21:35:22 goemon dd: 1+0 records in
Jul 25 21:35:22 goemon dd: 1+0 records out
Jul 25 21:35:22 goemon random: Saving random seed:  succeeded
Jul 25 21:35:23 goemon autofs: automount -USR2 succeeded
Jul 25 21:35:26 goemon nfslock: rpc.statd shutdown succeeded
Jul 25 21:35:26 goemon portmap: portmap shutdown succeeded
Jul 25 21:35:27 goemon syslog: klogd shutdown succeeded
```

不要なサービスの停止 http://www.linux.or.jp/JF/JFdocs

改訂版は　http://www.hokudai.ac.jp/hines/HINESworld/No.51/hw51_2.html

---

■はじめに

UNIX(Linux)マシンで外部ネットワークサービスを提供している場合、それ ぞれのサービスを提供するためのデーモンプログラムにセキュリティホールが あると、外部からマシンに不正に侵入される恐れがあります。このため、必要 のないサービスは停止するべきです。また、必要のないデーモンプログラムは インストールするべきではありません。

■デーモンの種類

デーモンプログラムは、起動方法の違いによって、以下の 2 種類にわけら れます。

1.UNIX の起動時に一緒に起動され、クライアントからの要求を待っているもの
2.クライアントからの要求があるたびに、inetd を介して起動されるもの

この文書では、それぞれについて、デーモンを起動させない方法を説明し ます。

■UNIX 起動時に起動するデーモンの停止

まず、現在どのようなデーモンプログラムが動作しているかを確認します。

```
$ ps acux
```

というコマンドを実行すると、現在その UNIX マシンで動いているすべての プロセス(プログラム)が、例えば以下のように表示されます。 これらのプロセスの中には、デーモンも含まれています。

```
ユーザ     PID  %CPU %MEM サイズ 常駐 端末 状態   開始     時間 コマンド
bin        266   0.0  1.6  1488  1044   ?   S    May  6   0:00 cannaserver
daemon     221   0.0  0.6   800   400   ?   S    May  6   0:00 atd
okada     7875   0.0  1.3  1256   832   p0  S     01:49   0:00 bash
okada     7952   0.0  0.9  1436   604   p0  R     01:59   0:00 ps
root         1   0.0  0.6   780   392   ?   S    May  6   0:02 init
root         2   0.0  0.0     0     0   ?   SW   May  6   0:00 kflushd
root         3   0.0  0.0     0     0   ?   SW<  May  6   0:00 kswapd
root         4   0.0  0.0     0     0   ?   SW   May  6   0:00 md_thread
root         5   0.0  0.0     0     0   ?   SW   May  6   0:00 md_thread
root        38   0.0  0.5   752   360   ?   S    May  6   0:00 kerneld
(以下略)
```

Linux が動作する際に最低限必要なデーモンには、一般に init, kflushd, kswaped, update, syslogd, klogd, atd, crond, inetd, getty 族(mingetty な ど) 等があります、また、以下のデーモンも必要な場合が多いです。

■プログラム名	役割、必要な場合
sshd	ssh で外部から接続したい場合
cannaserver	かな漢字変換システム Canna のサーバプログラム
jserver	かな漢字変換システム Wnn のサーバプログラム
lpd	プリンタへの出力を管理する

一方、不要である場合が多いデーモンには以下のものがあります。必要で なかったり、使用していなければ、停止するかアンインストールしてください 特に、sendmail, rpc.mountd, rpc.nfsd にはセキュリティホールがしばしば 発見されてますので、不要であれば必ず停止してください。また apache も 侵入の標的になることが多いです。

■プログラム名	役割、必要な場合
sendmail	メールの配送。メールサーバでないならば不要
rpc.mountd, rpc.nfsd	NFS(ネットワークを介したファイル共有)のためのデーモン
apache	WWW サーバのためのデーモン
ypserv, ypbind など	NIS(ネットワークを介したユーザ情報の共有)を利用するのに必要
portmap	NFS, NIS を利用するなら必要
named	DNS サーバのためのデーモン
smbd, nmbd	Samba(Windows マシンとの共有)サーバのためのデーモン
amd	Auto Mount Daemon、自動的に mount を行う
snmpd	

UNIX が起動する際には、UNIX, Linux ディストリビューションによって異な りますが、/etc/rc, や /etc/rc.local というファイル、あるいは /etc/rc.d/ 以下の各ファイルが読み込まれます。そこに各デーモンの起動コ マンドが書かれており、それに従って各デーモンが起動される仕組みになって います。

UNIX 起動時に 各デーモンを起動しないようにする方法ですが、この読み 込まれるファイルの構成の違いによって、以下の 2 つの方法があります。その 処置を取った後にマシンを再起動させると、デーモンは起動されません。ここ では、その方法の概略を述べます（今回省略)。実際に行う際には、それぞれの UNIX のド キュメント、参考書等で調べてください。

■inetd を介して起動するデーモンの停止

inetd を介して起動するデーモンの設定は、/etc/inetd.conf で行います。 /etc/inetd.conf の例(一部)は以下の通りです。

```
#
# Pop and imap mail services et al
#
pop-2   stream  tcp     nowait  root    /usr/sbin/tcpd ipop2d
pop-3   stream  tcp     nowait  root    /usr/sbin/tcpd ipop3d
imap    stream  tcp     nowait  root    /usr/sbin/tcpd imapd
```

提供している サービス・デーモンがそれぞれ 1 行で書かれています。 # で始まっている行は、コメントです。

inetd を介して提供するサービスのうち、auth は提供した方が良いですが、 その他のサービスは不要ならサービスの提供を止めてください。基本的には必 要かどうかわからなければ止めてしまっても困ることはないと思います。

特に止めた方が良いのは, pop 関連, imap, finger, smtp, systat, netstat, chargen の各サービスです。また、セキュリティ的に安全な sshd のみを利用していていて、外部から telnet, ftp, rlogin, rsh で接続できな くても良いならば、サービスの提供を止めてしまって良いです。

サービスを止めるためには、/etc/inetd.conf の中で 止めるサービスの行 の先頭に# をつけて、コメントにします.

書き換えた後に

```
# ps ax ¦ grep inetd
# kill -HUP inetd の PID
```

とすれば設定の変更が inetd に反映されます。 あるいは、マシンを再起動してください。

■デーモンのアンインストール

不要なデーモンはアンインストールしてください。パッケージ管理システ ムを使用している場合は、簡単にアンインストールすることができます。

パッケージ管理システムによってアンインストールのコマンドが異なりま す。rpm 形式のパッケージを採用しているディストリビューション(Vine Linux も含む)の場合は、

```
# rpm -e パッケージ名
```

とすると、アンインストールされます。

参考文献

* **ネットワーク・アタックに備える、日経 Linux 2000 年 6 月号(日経 BP), 78-99**
* **セキュリティはじめの一歩(日本の Linux 情報)**

この文章の履歴

* 「Linux に関するセキュリティ基本指針」として書かれた. 1998 年 12 月 29 日(岡田直資, 中浦正博, 谷口博)
* オリジナル文書を分割し、「計算機の基礎」ドキュメントシリーズとして 加筆. 2000 年 5 月 12 日(岡田直資)

参考ページ　今回参考にしたところ＋定番いくつか

■Vine Linux のオンライン・マニュアル■

http://www.vinelinux.org/documentations.html

■Linux JF (Japanese FAQ) Project ■←本家

http://www.linux.or.jp/JF/

IPCHAINS-HOWTO 日本語訳 ←隅から隅まで読むとよくわかる

http://www.linux.or.jp/JF/JFdocs/IPCHAINS-HOWTO-1.html

Firewall And Proxy Server HOWTO　日本語訳

http://www.linux.or.jp/JF/JFdocs/Firewall-HOWTO.html

Linux 2.4 NAT HOWTO　日本語訳

http://www.linux.or.jp/JF/JFdocs/NAT-HOWTO.html

■Linux 活用日記 ■ ←調べたい情報が載っている偉いページ．作業記録を作る見本かな

http://www.a-yu.com/

■「びぎねっと」■

http://begi.net/linux/reading/

『Linux のきそ』・『Linux 操作のきほん』第 10 回 パーミッションの基本

http://begi.net/linux/reading/linux_basic_knowledge/permission.html

■MYCOM PCWEB 「Enjoy! Linux」■

http://pcweb.mycom.co.jp/column/linux.html

■Linux Square ■

「各ディレクトリの役割を知ろう」

http://www.atmarkit.co.jp/flinux/index/indexfiles/index-linux.html

「vi の操作を覚えよう」

http://www.atmarkit.co.jp/flinux/rensai/theory05/theory05b.html

「Linux 起動の仕組みを理解しよう［init/inittab 編］」

http://www.atmarkit.co.jp/flinux/rensai/theory10/theory10a.html

「索引」

http://www.atmarkit.co.jp/flinux/index/indexfiles/sakuin-linux.html

■ZDNet Linux How To■

http://www.zdnet.co.jp/help/howto/linux/index.html

システムログの読み方を理解しよう

http://www.zdnet.co.jp/help/howto/linux/0007master/04/index.html

■ログをとろう！■

http://www.kozupon.com/log/log.html

■謎ログ■　←謎のアクセスログを見つけて楽しんでる？

http://iandu.s7.xrea.com/unimama/logwatch.html

■ASH multimedia lab. ■

http://ash.jp/

サーバ構築・運用マニュアル　http://ash.jp/env/index.htm

Linux Tips　http://ash.jp/linux/index.htm

■日経 Linux ■（現在登録制・・・・）

・コマンド集（機能別+アルファベット順）

・コマンド逆引き大全

・/etc リファレンス

・　Linux Q&A

http://itpro.nikkeibp.co.jp/linux/index.shtml

■おまけ■

http://www.debian.org/　（なんか情報あるでしょう）

http://ppc.linux.or.jp/　（ppc 用 linux）

http://www.mac.linux-m68k.org/（68k 用 linux）

http://www.fortunecity.com/business/mars/1542/index.html（HappyLinux / 68k 用 linux）

http://www.linuxppc.ne.jp/ (LinuxPPC 現在 Q4)

http://www.powerbook.org/linux/ (powerbook に linux)

http://www.maconlinux.org/ (Mac on Linux)


ファイル(F) 編集(E) 表示(V) 移動(G) ブックマーク(B) ツール(T) ウィンドウ(W) ヘルプ(H)
https://localhost.localdomain:10000/ip
検索
ホーム
一つ先のページへ進む
Webmin(local)
Webmin
Feedback
Log Out
Webmin システム サーバ ネットワーク ハードウェア Cluster その他
ヘルプ... モジュールの設定
Written by Tim Niemueller Home://page
IP chains ファイアウォール
Name
Inside -> FW
Inside -> Outside
Outside -> FW
Outside -> Inside
FW -> Inside
FW -> Outside
BigBrother
DHCP
DNS
FTP-Active
FTP-Passive
HTTP
HTTPS
IDENT
IMAP
IRC
LDAP
NFS
NNTP
NTP
NetBIOS
POP3
Ping
Proxy
Quake
SMTP
SNMP
SSH
Telnet
Traceroute
Webmin
Actv (Desc)
N/A
Enable the templates by marking the checkbox in the appropriate columns.
(Actv = Active/Enabled, Desc = Description)
Enable Masquerading support (Desc)
Save and Apply
@goemon [0.82.1]
メインメニュー に戻る
localhost.localdomain からデータを転送しています...