Linuxサーバ設定

サーバ設定と運用

Linuxサーバ設定　目次

- 基本設定
  - ユーザ追加
  - SSH設定
  - ネットワーク設定
  - シェル
  - ・・・
- サーバ設定
  - DNSサーバ
  - WEBサーバ
  - メールサーバ
- 運用
  - ログローテーション
  - バックアップ
  - リソース監視
  - ・・・

環境

- 本講座で使用するサーバ環境
  - OS: Fedora Core 4
  - CPU: Pentium4 2.8GHz
  - Memory: 512MB
  - Network: 192.168.0.0/24
  - Remote Access: SSH

Linuxディストリビューション

- RPM系
  - Cent OS : フリー、Red Hat Enterprise Linuxのクローン
  - Fedora Core : フリー、Red Hat Linux後継のコミュニティによるプロジェクト、非商用では最もよく使用されている。
  - Red Hat Enterprise Linux : 商用、Red Hat社提供、商用では最もよく使用されている。
  - SUSE Linux : 商用、Novell社提供、ヨーロッパで強い。
  - Turbolinux : 商用、Turbolinux社提供、デスクトップ製品を強化
  - Vine Linux : フリー、日本人作成、日本語環境に優れる
- Debian系
  - Debian GNU/Linux : 100%フリー、コミュニティベース、自由な構成がしやすい
  - KNOPPIX : フリー、CDからブートして利用する
  - Linspire (Lindows) : 製品、WindowsライクにGUIが充実
- Slackware系
  - Slackware :フリー、初期のディストリビューション
  - Plamo Linux : フリー、Slackwareを日本語化

Linuxサーバ基本設定

- 基本設定項目
  - ユーザ、グループ
  - リモートアクセス（SSH）
  - ネットワーク
  - シェル
  - 時刻・タイムゾーン
  - 文字コード

※OSインストールについては省略します。
※基本コマンド（ls, cd, vi等）は解説しません。

ユーザ、グループ

- ポリシー
  - できる限り一般ユーザで作業する。
  - 必要な時だけrootユーザで作業する。
  - 基本的には1人1ユーザアカウントを割り当てる。

- ユーザの作成

```
# useradd user01
# passwd user01
```

※グループも自動的に作成される。

- rootパスワードの変更

```
# passwd
```

ユーザ、グループ

- グループの作成

```
# groupadd -g 1000 admingroup
```

- ユーザの作成（オプション指定）

```
# useradd -u 501 -g admingroup -d /home/user02 -s /bin/bash -m -c “User 02" user02
# passwd user02
```

- （参考）ログイン不可能なユーザの作成

```
# useradd -u 900 -d /home/systemuser01 -s /sbin/nologin -m -c “No Login User" systemuser01
```

- （参考）パスワードなしでログイン可能なユーザの確認

```
# awk -F ':' '{if ($2=="") print $1}' /etc/shadow
```

- （参考）

/etc/default/useradd: useraddのデフォルト内容
/etc/skel/: ファイルの雛形が置かれるディレクトリ

ユーザ、グループ

/etc/passed

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
...
user01:x:500:500::/home/user01:/bin/bash
```

```
user02:x:501:1000:User 02:/home/user02:/bin/bash
   (1)(2)(3) (4)     (5)       (6)        (7)
```

(1) ユーザー名
(2) パスワード。最近のLinuxではシャドウ・パスワードという機能が有効になっているため、ここにはパスワードは表示されず「x」という文字が記述されている
(3) Linuxがユーザーを識別・管理するためのユーザーID
(4) Linuxがユーザーの所属するグループを識別・管理するためのグループID
(5) コメント欄（ユーザーのフルネームなどを記述する）
(6) ホーム・ディレクトリ（ユーザーがログインしたときの最初のディレクトリ）
(7) ログイン・シェル（ユーザーがログインすると、ここに記述したシェルが起動する）

/etc/shadow

```
root:$1$YEplOvhM$MqKvy3DIGQJOaWfjT6nkP0:131
81:0:99999:7:::
bin:*:13181:0:99999:7:::
daemon:*:13181:0:99999:7:::
adm:*:13181:0:99999:7:::
...
user01:$1$QAwhq3gV$PcVzbD/GQwqUCVNvqTGu71
:13181:0:99999:7:::
```

```
user02:$1$QAwhq:13181:0:99999:7:::
  (1)  (2)       (3)    (4) (5)   (6)(7)(8)(9)
```

(1) ユーザー名
(2) 暗号化されたパスワード（MD5という暗号化アルゴリズムで暗号化されている）
(3) パスワードの最終変更日（1970年1月1日からの日数）
(4) パスワードの最終変更日から次にパスワードを変更できるようになるまでの期間（日数）
(5) パスワードの有効期間（日数）
(6) パスワードの有効期限が迫っていることを何日前からユーザーに知らせるかを指定（日数）
(7) パスワードの有効期限が切れてもパスワードが変更されなかったとき、そのユーザーのアカウント（利用権）を無効にするまでの猶予期間（日数）
(8) ユーザー・アカウントの有効期限（1970年1月1日からの日数）
(9) 未使用の領域

ユーザ、グループ

/etc/group

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
...
user01:x:500:
```

```
admingroup:x:1000:user02,user03
      (1)        (2)(3)    (4)
```

(1) グループ名
(2) グループ・パスワード（現在は使用されていない）
(3) グループID（Linuxが各グループを識別するための番号）
(4) このグループに属するメンバー（ユーザー名）

リモートアクセス（SSH）

- ポリシー
  - telnetでのアクセスを禁止する。
  - rloginやrshも禁止する。

（最近はデフォルトで禁止されている場合が多い。）

  - rootユーザでのSSHアクセスを禁止する。

/etc/ssh/sshd_config

```
…
PermitRootLogin no
…
```

/etc/xinetd.d/telnet

```
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        disable         = yes
}
```

ネットワーク

- 設定項目
  - 通常はGUIから行うが、直接ファイルを編集できると便利
  - デフォルトルート(default router)を設定する。

- IPアドレスの設定

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=static
HWADDR=00:90:99:xx:xx:xx
BROADCAST=192.168.1.255
IPADDR=192.168.1.2
NETMASK=255.255.255.0
NETWORK=192.168.1.0
ONBOOT=yes
TYPE=Ethernet
```

- デフォルトルートの設定

/etc/sysconfig/network

```
NETWORKING=yes
HOSTNAME=localhost.localdomain
GATEWAY=192.168.1.1
```

※設定されていないと、よくネットワークトラブルの要因になるので要注意

- ネットワーク設定の反映

```
# service network restart
```

ネットワーク

- ホスト名の設定

```
# hostname server.mydomain
```

- ホスト名とIPアドレスとの対応を定義

/etc/hosts

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1        localhost.localdomain localhost
192.168.1.2      www.mydomain   www
192.168.1.3      client.mydomain    client
```

- DNSリゾルバの設定

/etc/resolv.conf

```
nameserver 192.168.1.1
search localdomain
```

※設定されていないと、名前解決ができない。

ネットワーク

- ルーティングテーブルの確認

```
# route (netstat -r)
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eth0
default         192.168.1.1     0.0.0.0         UG        0 0          0 eth0
```

- 一時的なデフォルトルートの追加

```
# route add default gw 192.168.1.254
```

※ただし、サーバを再起動すると、設定が消えます。

- スタティックルート(static router)の設定

/etc/sysconfig/static-routes

```
any net 172.26.1.0 netmask 255.255.255.0 gw 192.168.1.1 dev eth1
any net 10.100.1.0 netmask 255.255.255.0 gw 192.168.1.1 dev eth1
```

ネットワーク

- NIC (Network Interface Card)の確認

```
# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:90:99:22:B5:08
          inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::290:99ff:fe22:b508/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13221246 errors:0 dropped:1 overruns:0 frame:0
          TX packets:5644635 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2618301881 (2.4 GiB)  TX bytes:1771825470 (1.6 GiB)
          Interrupt:11 Base address:0x8000

…
```

- NICのダウン、アップ

```
# ifconfig eth0 down  (ifdown eth0)
# ifconfig eth0 up  (ifup eth0)
```

シェル

- シェル
  - LinuxではbashがよくR使われます。デフォルトのシェルです。
  - Cシェル系(csh, tcsh, zsh)も好みに応じて。他にも多数。
  - ログインシェルについてbashの場合、/etc/profileが存在すればまずここから設定を読み、次に、~/.bash_profile, ~/.bash_login, ~/.profileをこの順で探して設定を読みます。終了時には、~/.bash_logoutがあればこれを読み込んで実行します。
  - ログインシェル以外について、~/.bashrcがあれば、これを読み込んで実行します。
- 環境変数の表示

```
$ export
```

- 一時的な環境変数の設定

```
$ export UMASK=022 ← umaskの値を022に設定
$ export TMPUT=600 ← 一定時間コマンドラインからの入力を待機し、これを超えるとユーザは自動的にログアウトする。
```

シェル

- 恒久的な設定

~/.bash_profile

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
unset USERNAME
```

~/.bashrc

```
# .bashrc

# User specific aliases and functions

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
```

時刻･タイムゾーン

- 時刻設定

```
# date –s 09:00
```

- NTPサーバと同期

```
# ntpdate clock.redhat.com
```

※ただし、NTPサーバが動作していると、このコマンドは使用できません。

- タイムゾーン
  - 通常のインストールであれば設定変更の必要はありません。
  - もしタイムゾーンが違う場合には以下の設定を行います。

- タイムゾーンの設定

```
# ｌn -fs /usr/share/zoneinfo/Japan /etc/localtime
```

タイムゾーンをJapanに変更する場合には、/etc/localtime に日本のタイムゾーン情報のシンボリックリンクを張ります。

文字コード

- 文字コード
  - UNIXでは通常文字コードはEUC-JPですが、Fedora Coreではデフォルトで UTF-8になっています。
  - 使いにくい場合はUTF-8に変更します。
- 文字コードの設定

/etc/sysconfig/i18n

```
#LANG="ja_JP.UTF-8“
LANG="ja_JP.eucJP"
```

/etc/man.config

```
#PAGER          /usr/bin/less -is
PAGER          /usr/bin/lv"
```

Linuxサーバ設定

- サーバ設定
  - DNSサーバ
  - WEBサーバ
  - メールサーバ

別途資料を参照

Postfix

- Postfixの設定

/etc/postfix/main.cf

```
# ホスト名
myhostname = mail.sensaba.net
# ドメイン名
mydomain = sensaba.net
myorigin = $mydomain
# メールを受信するインタフェース
inet_interfaces = $myhostname
# メッセージの最終的な宛先として受けつけるホスト名
mydestination = $myhostname, localhost.$mydomain $mydomain
unknown_local_recipient_reject_code = 450
# メール中継を許可するクライアントのIPアドレス
mynetworks = 125.100.241.0/24
# メール中継を許可する宛先のドメイン
relay_domains = $mydestination
# エイリアスファイル
alias_maps = hash:/etc/postfix/aliases
sendmail_path = /usr/sbin/sendmail.postfix
newaliases_path = /usr/bin/newaliases.postfix
mailq_path = /usr/bin/mailq.postfix
setgid_group = postdrop
manpage_directory = /usr/share/man
sample_directory = /usr/share/doc/postfix-2.0.11/samples
readme_directory = /usr/share/doc/postfix-2.0.11/README_FILES
# エイリアスデータベース
alias_database = hash:/etc/postfix/aliases
# メール配信データベース（特定のホストおよびドメインの配信先の制御）
transport_maps = hash:/etc/postfix/transport
```

その他のサーバ

- 各種サーバについて
  - メールサーバ:qmail
  - WEBメール:squirrelmail
  - メーリングリスト:mailman, fml, majordomo
  - データベース:MySQL, PostgreSQL
  - ニュースサーバ:INN
  - プロキシサーバ:Squid
  - DHCPサーバ:ISC DHCP
  - 情報共有(NIS):ypserv
  - LDAPディレクトリ:OpenLDAP
  - ファイルシステム共有(NFS):nfs
  - Windows共有:Samba
  - Mac OS共有:Netatalk

Linuxサーバ運用

- 運用項目
  - FTPサーバ
  - NTP
  - ログ
  - バックアップ
  - アップデート
  - パッケージ管理
  - 監視
  - サーバ設定確認

FTPサーバ

- FTPサーバについて
  - リモートとローカルサーバのファイルのやりとりにFTPがよく使用されます。
  - FTPサーバには、フリーソフトでvsftpdやWU-FTPD、ProFTPDなどあります。
  - FTPサーバは、かつては、スーパーデーモン(inetd)から逐次起動されることが多かったようですが、最近では単独でデーモンとして動作する場合が多いようです。
  - 一般ユーザに対しては通常chrootの設定をして、自分のディレクトリより上位のディレクトリに移動することができないようにします。
  - FTPプロトコルではログイン時のパスワードが平文で流れ、データもそのまま流れますので情報漏洩には注意が必要です。

FTPサーバ

- vsftpdの設定

/etc/vsftpd/vsftpd.conf

```
...
anonymous_enable=NO # anonymousユーザを禁止
...
xferlog_std_format=NO # 詳細なログを記録する
xferlog_file=/var/log/vsftpd.log # 詳細なログ記録の場合のログファイルの指定（デフォルト）
...
ascii_upload_enable=YES # ASCIIモードでのアップロードを許可
ascii_download_enable=YES # ASCIIモードでのダウンロードを許可
...
chroot_list_enable=YES # chroot有効（ホームディレクトリより上層への移動を禁止）
chroot_list_file=/etc/vsftpd.chroot_list # chroot対象ユーザをこのファイルに指定
...
（最下行に一行追加）
use_localtime=YES # ファイルのタイムスタンプを日本時間にする。（デフォルトはGMTによる表示）
```

ログ記録について、デフォルトでは、"xferlog_std_format=NO"となっており、/var/log/xferlogにログが出力されます。詳細なログを記録するように設定すると、FTPでのログイン情報を記録されるようになります。（設定ではYESをNOに変更する点に注意。）

FTPサーバ

chroot対象（ホームディレクトリより上層への移動を禁止）のユーザを以下のファイルに記述します。（デフォルトではファイルがありませんので作成します。）

/etc/vsftpd.chroot_list

```
limiteduser1 # 以下のユーザがchroot対象
limiteduser2
...
```

FTPでログイン不可能なユーザを追加します。

/etc/vsftpd.ftpusers

```
systemuser1 # 以下のユーザはFTPログイン不可
systemuser2
...
```

設定が終了したらvsftpdを起動します。

```
# service vsftpd start
```

サーバ再起動時にvsftpdを実行するには、以下のコマンドを実行します。

```
# chkconfig vsftpd on
```

FTPサーバ

chrootについて

chrootについて、通常/etc/vsftpd.chroot_listファイルにはchrootさせるユーザを記述します。
すべてのユーザをまずchrootさせて一部の管理系のユーザだけchrootさせないようにしたい（ホームディレクトリより上層への移動を許可する）という場合もあります。すべてのユーザを記述するのは大変ですので、この場合は次のようにすると、/etc/vsftpd.chroot_listファイルにはchrootさせないユーザを記述できるようになります。

/etc/vsftpd/vsftpd.conf

```
…
chroot_local_user=YES # ローカルユーザをすべてchrootする
chroot_list_file=/etc/vsftpd.chroot_list # ここに書かれたユーザはchrootしない
…
```

/etc/vsftpd.chroot_list

```
adminuser # 以下のユーザはchrootしない
servermanager
…
```

NTP

- NTPサーバについて
  - NTPとは時刻同期のためのプロトコル。ネットワークに接続されたコンピュータの内蔵時計の時刻を同期させます。
  - NTPクライアントがNTPサーバに現在時刻の問合せを行い、NTPサーバは自身の時刻を返答します。
  - DNSのように階層構造をとっています。Stratumと呼ばれます。
  - 最上位層(Stratum1)はGPSや原子時計を利用して正確な時刻を維持しています。
  - 一般的な運用方法としては、各システム内にメイン、サブの二台のNTPサーバを置き、システム内の他のコンピュータはこのNTPサーバに問い合わせます。NTPサーバは外部の最も近い上位のNTPサーバに問い合わせます。
  - 非常に高い精度を必要とする場合には、3台以上の上位のNTPサーバを指定します。

NTP

- NTPサーバの設定

/etc/ntp.conf

```
...
# -- CLIENT NETWORK -------
restrict 192.168.1.0 mask 255.255.255.0 notrust nomodify notrap # 許可するNTPクライアント
...
# --- OUR TIMESERVERS -----
restrict 192.168.1.2 mask 255.255.255.255 nomodify notrap noquery # NTPサーバからは受け付けない
...
server clock.redhat.com # 上位のNTPサーバ
server clock2.redhat.com # 上位のNTPサーバ
server clock3.redhat.com # 上位のNTPサーバ
...
# authenticate yes # 認証をしない
…
# keys          /etc/ntp/keys # 認証をしないので鍵も使用しない
```

設定が終了したらntpdを起動します。

```
# service ntpd start
```

サーバ再起動時にntpdを実行するには、以下のコマンドを実行します。

```
# chkconfig ntpd on
```

NTP

NTPサーバの同期確認

```
# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*clock1.redhat.c .CDMA.           1 u  393 1024  377  173.850    1.359   0.285
+clock2.redhat.c .CDMA.           1 u  991 1024  377  178.030   -0.129   0.083
+clock3.redhat.c clock1.redhat.c  2 u  183 1024  377  130.129   -1.949   0.056
```

フィールド	意味
remote	上位のNTPサーバのホスト名またはIPアドレス
refid	remoteが参照している上位のNTPサーバのホスト名またはIPアドレス
st	remoteの階層
when	上位NTPサーバに以前問い合わせてからの経過秒
poll	問い合わせの間隔(秒)
reach	ステータス(8進数)
delay	上位NTPサーバとの時間差(ミリ秒)
offset	計算されたoffset値(ミリ秒)
jitter	さらに計算された分散値(ミリ秒)

ログ

- ログについて
  - アプリケーションのログ、システムのログなど。
  - ログは重要なもの。エラーの解析、アクセス分析、不正行為の調査、経営戦略など様々な用途に活用できます。
- ログローテーション
  - ログを放置しておくとどんどん増えていくので定期的にローテーションしてバックアップをとり古いものは消します。
  - Red Hat系Linuxではデフォルトでlogrotateがインストールされていて、一週間毎にローテーションし、4週間分残します。
  - 必要に応じてローテーション間隔を変更します。
  - 独自のスクリプトを作成すればログファイルの名前を自由につけたりログローテションのタイミングを自由に設定したりできます。

ログ

- logrotateの設定

以下はすべてデフォルト値です。このままでも問題ありません。

/etc/logrotate.conf

```
# see "man logrotate" for details
# rotate log files weekly
weekly ← 周期の指定

# keep 4 weeks worth of backlogs
rotate 4 ← 世代の指定

# create new (empty) log files after rotating old ones
create ← ローテーション後に新規にファイルを作成する

# uncomment this if you want your log files compressed
#compress ← ローテーション後に生成されるファイルをgzip形式で圧縮する

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d ← 別の設定をとりこむ指定

# no packages own wtmp -- we'll rotate them here
/var/log/wtmp { ← 実際のログファイルとそのログ固有の設定
    monthly
    create 0664 root utmp
    rotate 1
}
```

ログ

- logrotateの停止

ログをすべて残したい場合や独自のログローテーションをさせたい場合には、logrotateによるログローテーションを止めます。

logrotateはcronにより毎日実行されますので、これを削除します。または、cronで実行されないように別のディレクトリに移動します。

```
# rm /etc/cron.daily/logrotate
```

ログローテーション

- 日毎の日付入りログローテーション

例）Apacheのログを毎日ローテーションさせ、古いログを前日の日付入りで保存していくためのシェルスクリプトです。

access_log_rotation.sh

```
#!/bin/sh
#
# Apache access log rotation
# LOGDIR=/var/log/httpd/ # log directory
LOGDIROLD=/home/svrmgr/log/httpd/ # old log directory
LOG=access_log # log file format
LOGOLD=access_log_ # old log file format
LOGEX=.log # log file extension
CALCPROG=/home/svrmgr/util/calc_date.pl # the program to calculate past date
if [ ! -r $LOGDIR -o ! -r $LOGDIROLD ]
then
  echo "Log directory does not exist"
  exit
fi
if [ ! -f $CALCPROG ]
then
  echo "Date calculate program does not exist"
  exit
fi
YESTERDAY=`$CALCPROG -1` # get the date of yesterday
（続く）
```

ログローテーション

```
（続き）
cd $LOGDIR
if [ ! -f $LOGDIROLD$LOGOLD$YESTERDAY$LOGEX ]
then
  cp $LOG $LOGDIROLD$LOGOLD$YESTERDAY$LOGEX # rename log
  chown svrmgr:svrmgr $LOGDIROLD$LOGOLD$YESTERDAY$LOGEX # change user
  cp /dev/null $LOG # init log
  chmod 644 $LOG # change mode
fi
kill -HUP `cat /var/run/httpd.pid`
exit
```

cronに登録します。（毎日0時1分にログローテーション）

```
# crontab -e
（以下、編集内容）
# rotate apache log
1 0 * * * /home/svrmgr/scripts/access_log_rotation.sh > /dev/null 2>&1
```

以下のように過去のアクセスログが保存されます。

```
# ls -l /home/svrmgr/log/httpd/
-rw-r--r-- 1 svrmgr svrmgr 1557760 12月 2 00:01 access_log_20051201.log
-rw-r--r-- 1 svrmgr svrmgr 1873220 12月 3 00:01 access_log_20051202.log
-rw-r--r-- 1 svrmgr svrmgr 2024303 12月 4 00:01 access_log_20051203.log
…
```

バックアップ

- バックアップ方法について
  - ファイルシステムの一部を丸ごとリモートにコピーするには、tarコマンドが適しています。
  - 複数のマシン間でファイルシステムを同期するには、rsyncコマンドが適しています。
  - rsyncではファイルやディレクトリの差分だけを扱うことができるため、効率よく同期を取ることができます。
  - rsyncはTCP873番ポートを使用して独自プロトコルで通信を行いますが、sshやrshなどのセッション上で動作させることも可能です。
  - rsync独自プロトコルでの通信させるためにはあらかじめ設定ファイルの作成が必要ですが、ssh経由で動作させれば設定ファイルも必要ありません。
  - ブートセクタのバックアップにはddコマンドを利用します。

バックアップ

- tarでバックアップ

```
# tar zcvf home.tar.gz /home
# scp home.tar.gz backup:
```

- SSH経由のtarでバックアップ

```
# tar zcvf - /home | ssh backup "cat > www-home.tar.gz"
```

- ディレクトリ丸ごとコピー

```
# tar zcf - /usr/local/apache/htdocs | ssh www2 “cd /usr/local/apache; mv htdocs htdocs.bak; tar zpvxf -”
```

※tarコマンドに-zスイッチをつけて転送データの圧縮を行っていますが、必ずしもパフォーマンスが向上するわけではありません。

- サイズの大きいアーカイブのコピー

```
# ssh www2 “cd /usr/local/backup; tar zpvxf -” < big-archive.tar.gz
```

アーカイブのサイズが非常に大きくリモート側のファイルシステムにいったんアーカイブのコピーを置くスペースがない場合の方法です。

バックアップ

- rsyncでバックアップ

sshを使ってファイル転送を行うには、次のようにrsyncのコマンドを指定します

```
(1) rsync --rsh=ssh [オプション] 送信元　[ユーザ@]ホスト:送信先
(2) rsync --rsh=ssh [オプション] 送信元　rsync://[ユーザ@]ホスト[:ポート]/DEST
(3) rsync --rsh=ssh [オプション] [ユーザ@]ホスト:送信元　送信先
(4) rsync --rsh=ssh [オプション] rsync://[ユーザ@]ホスト[:ポート]/送信元　[DEST]
```

```
# rsync --rsh=ssh -av --delete /home/user/data/ backup:/home/user/data
```

--deleteスイッチをつけると削除まで含めて正確なコピーを行います。

※sshでログインするためのパスワードの入力を求められますが、cronなどから自動処理を行うためには、ssh接続認証用の鍵を用意し、そのパスフレーズを空にしておく必要があります。

アップデート

- アップデートについて
  - アップデータファイルが各OSの供給元よりリリースされます。
  - アップデートには通常3種類あります。
    - セキュリティアップデート
    - バグフィックス
    - エンハンスメント
  - アップデートによっては既存のサーバの動作に影響を及ぼす場合もありますので注意してください。

  - Red Hat Enterprise Linuxではup2dateコマンドを利用してRed Hat Network (RHN)に接続しアップデートします。
  - Red Hat Network (RHN)に接続するためには、サブスクリプションの購入および登録と、サーバのRHNへの登録が必要です。
  - RHNのサイト： http://rhn.redhat.com/
  - Fedora Coreではyumというツールが標準でついていてこれを利用します。

up2dateによるアップデート

サーバのRHNへの登録

```
# rpm --import /usr/share/rhn/RPM-GPG-KEY ← Red Hat GPGキーのインストールが必要
# rhn_register
```

ウィンドウが表示されるので、指示に従って登録します。
登録後はRHNサイトにログインしてサーバ登録を確認してください。

アップデート

```
# yum update <software>
# yum update ← 全パッケージをアップデート
```

自動アップデート

chkconfigで自動アップデートを有効にできます。何かデーモンが常駐するわけではなく、実際はロックファイルができてcronで実行されます。

```
# chkconfig yum on
```

yumによるアップデート

- インストール

```
# yum install <software>
```

- アンインストール

```
# yum remove <software>
```

※アンインストールしようとするソフトウェアを必要とするソフトウェアも一緒にアンイントールします。

- アップデート

```
# yum update <software>
# yum update ← 全パッケージをアップデート
```

- 自動アップデート

chkconfigで自動アップデートを有効にできます。何かデーモンが常駐するわけではなく、実際はロックファイルができてcronで実行されます。

```
# chkconfig yum on
```

パッケージ管理

- パッケージ管理について
  - Red Hat系では、RPM (Red Hat Package Manager)が使用されます。
  - パッケージはそれぞれのOS、OSバージョン、CPU環境ごとに細かく用意されています。
  - パッケージには必要なファイルがすべて含まれており、ユーザはコマンド一つで簡単にインストールすることができます。
  - パッケージによっては他のパッケージとの依存性がある場合がありますので注意が必要です。
  - パッケージを使用するとインストールされるディレクトリが決まってしまいますので、自由にインストールしたい場合には、ソースからのコンパイルをお勧めします。（時には技術力アップのためにも。）

パッケージ管理

- ## インストール

新規インストールの場合、または旧バージョンのパッケージを残してアップデートをしたい場合に行います。

```
# rpm –ivh <package>
```

- ## アップデート

旧バージョンのパッケージに上書きされます。

```
# rpm –Uvh <package>
```

- ## アンインストール

他のパッケージとの依存関係があると削除できない場合があります。

```
# rpm –e <package>
```

- ## パッケージのインストール状況確認

```
# rpm –q <package>
# rpm –qa ← すべてのパッケージの状況を確認
```

- ## パッケージの内容確認

```
# rpm –qi <package> ← パッケージの詳細内容を確認
# rpm –qf <package> ← インストールされるファイルのリストを確認
```

監視

- 監視について
  - サーバやネットワーク機器が正常に動作しているかどうか常にチェックします。
  - 運用者は各種コマンドを使用してサーバの状態をチェックします。
  - 24時間365日すべての項目の監視は人手ではできませんので、通常は自動で行います。
  - 異常が発生したらその情報を監視マネージャのコンピュータに集め、警報装置などでアラームを鳴らして運用者に知らせます。
  - サーバの監視内容には、以下のようなものがあります。
    - ✓ ハードウェア監視
    - ✓ ping監視
    - ✓ サービス監視
    - ✓ プロセス監視
    - ✓ リソース監視

サーバ状態チェック

- 各種サーバ状態チェックのコマンド

システム起動時の情報確認

```
$ dmesg
```

システムの稼働時間と平均負荷の確認

```
$ uptime
 22:34:18 up 95 days, 15:09,  1 user,  load average: 0.00, 0.01, 0.00
```

システム全般のアクティビィの確認

プロセス情報、メモリ情報、スワップ情報、I/O情報、システム情報、CPU情報を確認できます。

```
# vmstat
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs us sy id wa
 0  0    868  15960  54960 267304    0    0     1     1    4     0  1  0 99  0
```

CPUとデバイスの確認

CPU情報とデバイスのI/O情報を確認できます。

```
# iostat
Linux 2.6.11-1.1369_FC4 (localhost.localdomain)        2006年05月28日
CPU平均:  %user   %nice    %sys %iowait   %idle
           0.59    0.00    0.37    0.03   99.01
デバイス:         tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
hda              0.45         1.61        12.46   13335622  103077168
```

サーバ状態チェック

- プロセスチェック

全プロセスのCPU使用量の確認

```
$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   1744   568 ?        S    Feb21   0:11 init [5]
root         2  0.0  0.0      0     0 ?        SN   Feb21   0:00 [ksoftirqd/0]
root         3  0.0  0.0      0     0 ?        S    Feb21   0:00 [watchdog/0]
```

CPUを多く使用しているプロセスをリアルタイムで定期的に確認

```
$ top
top - 00:38:26 up 95 days, 17:13,  1 user,  load average: 0.00, 0.00, 0.00
Tasks:  85 total,   1 running,  84 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.6% us,  0.3% sy,  0.0% ni, 99.0% id,  0.0% wa,  0.0% hi,  0.0% si
Mem:    645324k total,   627380k used,    17944k free,    54980k buffers
Swap:  1572856k total,      868k used,  1571988k free,   265384k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
    1 root      16   0  1744  568  492 S  0.0  0.1   0:11.65 init
    2 root      34  19     0    0    0 S  0.0  0.0   0:00.12 ksoftirqd/0
    3 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
```

サーバ状態チェック

■ ディスクチェック

マウントしているディスク容量を表示

```
$ df
Filesystem           1K-ブロック    使用    使用可  使用% マウント位置
/dev/mapper/VolGroup00-LogVol00
                     75861016  22990484  48954756  32% /
/dev/hda1              101086     11785     84082  13% /boot
/dev/shm               322660         0    322660   0% /dev/shm
```

指定したディレクトリのディスク容量を表示

```
# du
```

ディスクのパーティションを確認

```
# fdisk -l

Disk /dev/hda: 81.9 GB, 81964302336 bytes
255 heads, 63 sectors/track, 9964 cylinders
Units = シリンダ数 of 16065 * 512 = 8225280 bytes

 デバイス Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1          13      104391   83  Linux
/dev/hda2              14        9964    79931407+  8e  Linux LVM
```

サーバ状態チェック

- ネットワークチェック

ARPテーブルの確認

```
# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
brc-14v.home             ether   00:90:CC:81:F7:A8   C                     eth0
# arp -a
brc-14v.home (192.168.1.1) at 00:90:CC:81:F7:A8 [ether] on eth0
```

ルーティングテーブルの確認

```
$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eth0
0.0.0.0         192.168.1.1     0.0.0.0         UG        0 0          0 eth0
```

プロトコルごとのトラフィック状況の確認

```
$ netstat -s
…
```

ネットワークインターフェースごとのトラフィック状況の確認

```
$ netstat -i
…
```

サーバ状態チェック

- ネットワークチェック

ネットワークソケット状況の確認

```
# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:ftp                   *:*                     LISTEN
tcp        0      0 *:http                  *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
…
tcp        0      0 backup.local.yohan.cc:ssh   ::ffff:125.100.241.194:3716 ESTABLISHED
…
udp        0      0 *:32768                 *:*
…
```

TCP/IPのパケットを監視

```
# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
01:09:02.369004 IP 125.100.241.194.3951 > backup.local.yohan.cc.http: S 407068493:407068493(0) win
        64240 <mss 1460,nop,nop,sackOK>
01:09:02.391896 IP backup.local.yohan.cc.http > 125.100.241.194.3951: S 113697931:113697931(0) ack
        407068494 win 5840 <mss 1460,nop,nop,sackOK>
01:09:02.385059 IP 125.100.241.194.3951 > backup.local.yohan.cc.http: . ack 1 win 64240
…
```

サーバ状態チェック

- ネットワークチェック

相手のマシンにパケットが届くかどうか確認

```
$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=0.447 ms
…
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.386/2.087/4.119/1.691 ms, pipe 2
```

相手のマシンにパケットが届くまでの経路を確認

```
$ traceroute www.yahoo.co.jp
traceroute: Warning: www.yahoo.co.jp has multiple addresses; using 202.93.91.141
traceroute to www.yahoo.co.jp (202.93.91.141), 30 hops max, 38 byte packets
 1  brc-14v.home (192.168.1.1)  0.532 ms  0.424 ms  7.971 ms
 2  125.100.241.193 (125.100.241.193)  10.057 ms  10.201 ms  10.737 ms
…
```

自動監視

- 自動監視について
  - エンタープライズ向けサーバ製品の場合、OSに依存せずにハードウェアを監視するための特別な機構があることが多いです。
  - サーバやNW機器の生死の確認は通常pingで行います。
  - サーバ上で動いているサービスの確認は、各サービスが利用しているTCPポートのSYN ACKによって行います。
  - サービスの正常性の確認およびレスポンス速度を確認するために、定期的にサーバにアクセスしコンテンツを取得する場合もあります。
  - プロセス監視、リソース監視については、各コマンドの出力結果を抽出処理することで監視します。
  - 各プロセスのエラーを検知するためにログ（主に/var/log/messages）を監視します。

システムログ

- syslogについて
  - 重要なログを記録してエラーを検知するとともに、後でのトラブル調査に備えます。
  - 多くのサービスプログラムは、ログの出力にシステムロガーsyslogを使用します。
  - swatchなどのツールでログを自動的にチェックできます。
  - -rオプションを使用して、/etc/syslog.confで送り先を指定すれば、リモートコンピュータにログを書き出すことができます。ネットワーク上のすべてのコンピュータのログを集中管理できます。
  - ログメッセージは平文のまま送信されるので、悪意のあるユーザーによるログの改ざんなどの危険性が伴います。

/etc/sysconfig/syslog

```
SYSLOGD_OPTIONS="-m 0"
KLOGD_OPTIONS="-x"
```

SYSLOGD_OPTIONSの指定内容
-m 0：“MARK”されたメッセージをログしない。
-r：リモートからのsyslogdの使用を許可する。

システムログ

- syslogdの設定

各行で指定されたファシリティとレベル以上のログ対応するファイルに記録されます。
レベルがnoneの場合にはログは一切記録されません。
コメントを除いた本文の1行目は、mail, news, authority, cronのファシリティについては、1行目の設定が適用されないことを示しています。

/etc/syslog.conf

```
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;news.none;authpriv.none;cron.none          /var/log/messages
# The authpriv file has restricted access.
authpriv.*                                                  /var/log/secure
# Log all the mail messages in one place.
mail.*                                                      /var/log/maillog
# Log cron stuff
cron.*                                                      /var/log/cron
# Everybody gets emergency messages
*.emerg                                                     *
# Save news errors of level crit and higher in a special file.
uucp,news.crit                                              /var/log/spooler
# Save boot messages also to boot.log
local7.*                                                    /var/log/boot.log
```

システムログ

- ファシリティ

syslogで出力されるログをどのカテゴリで出力するかを指定する。

ファシリティ	対象	代表的なプログラム
auth (security)	認証時	login, su, getty
authpriv	認証時	―
cron	Cron	cron, at
darmon	デーモン	ftpd, named, pppd
kern	カーネル	カーネル
local0-local7	ローカル	―
lpr	プリンタ	lpd
mail	メール	Sendmail, ipop3d
news	ニュース	innd
syslog	syslog	syslogd
user	ユーザ	ユーザプロセス
uucp	uucp	uucp

- レベル

ログの重要度や緊急度を指定する。

レベル	対象
none	出力なし
emerg (panic)	システムが不能に陥っているような問題
alert	ただちに修正されるべき問題
crit	ハードウェアエラーのように致命的な問題
err (error)	一般的なエラー
warning (warn)	警告
notice	通知
info	情報
debug	デバッグ情報

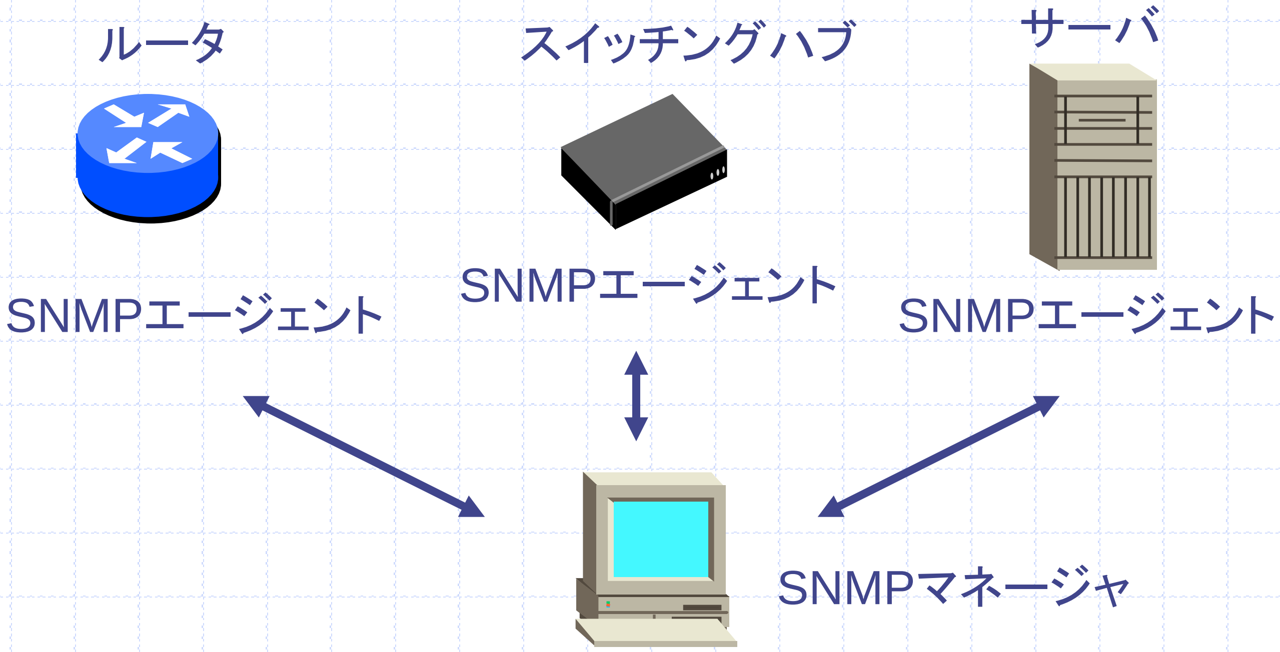
- アクション

動作を指定する。

出力先	アクションの例
ファイルおよびデバイス	/var/log/messages, /dev/console
名前付きパイプ	|<パイプ名>
ホスト	@loghost
ユーザ	root, user1, user2 *

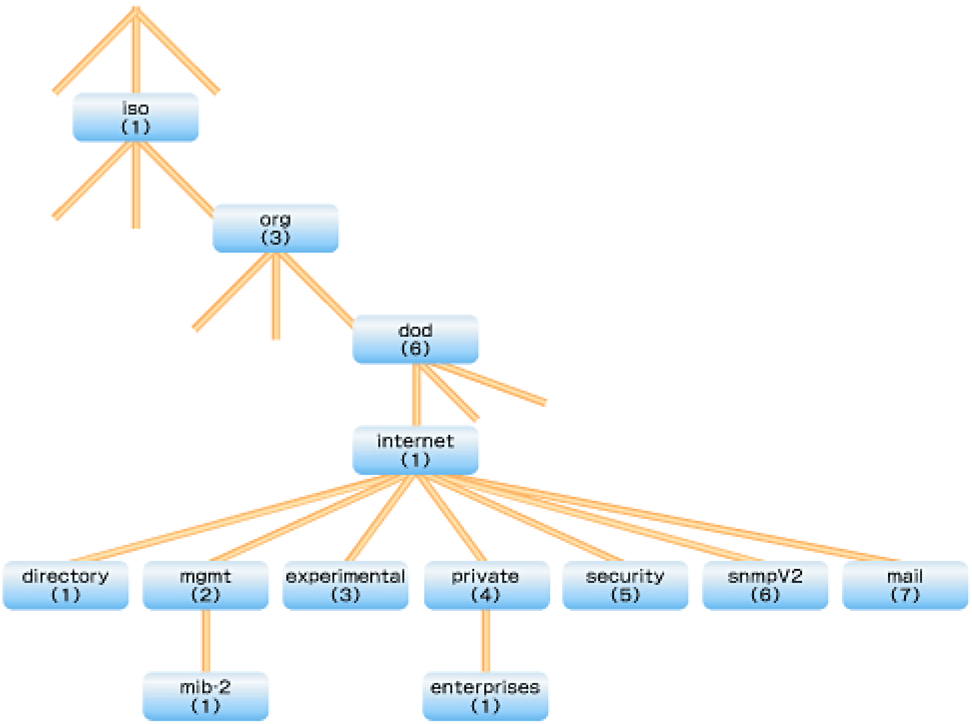
SNMP

- SNMPについて
  - ネットワークやサーバの状況を監視するためのプロトコル。
  - 管理対象の機器にはSNMPエージェントというサービスが稼動していて、SNMPマネージャで管理情報を取得します。
  - Net-SNMPがよく使用されます。SNMPエージェントの本体のnet-snmpパッケージとマネージャにあたるプログラム群より構成されるnet-snmp-utilsパッケージがあります。



SNMP

- MIB
  - 管理情報は、MIB (Management Information Base)と呼ばれ、オブジェクトから構成されます。各オブジェクトにはOIDまたはObject IDと呼ばれる番号がつけられています。
  - 標準的なMIBとしてMIB2がRFC1213で定義されています。



OID: .1.3.6.1.2.1

主なMIBのカテゴリ

カテゴリ	概要
System	ホストやルータの名称などのシステム情報
interfaces	ネットワークインタフェースに関する情報
at	ARPなどのアドレス変換情報
Ip	IPに関する情報
icmp	ICMPに関する情報
tcp	TCPに関する情報
udp	UDPに関する情報
egp	EGPに関する情報
snmp	SNMPに関する情報
private	製品特有の情報

SNMP

- SNMPの通信
  - SNMPコマンドにより、マネージャはエージェントからの情報の取得およびエージェントへの操作を行います。
  - TRAPコマンドは、エージェントでイベントが発生した場合にマネージャへ知らせるコマンドで、機器の障害検知によく使用されます。（CPUエラー、CPU使用率超過、ディスク容量不足など）
  - **コミュニティと**呼ばれるパスワードを使用します。情報の取得のみのreadコミュニティは標準では“public”という文字列を使用し、情報の設定が可能なread/writeコミュニティは標準では“private”という文字列を使用します。

SNMPコマンド

SNMPコマンド	**処理内容**
GET-REQUEST	指定したMIB変数の値を取得
GET-NEXT-REQUEST	正確な名前を指定せずにMIB変数の値を取得 （要素がいくつあるかわからないテーブル型の情報を取得する場合に使用）
SET-REQUEST	指定した変数に値を設定
GET-RESPONSE	SET-REQUESTに対する応答
TRAP	イベントが発生したことの通知

SNMP

- snmpdの設定

/etc/snmp/snmpd.conf 初期値

```
# First, map the community name "public" into a "security name"
#       sec.name  source          community
com2sec notConfigUser  default       public
# Second, map the security name into a group name:
#       groupName      securityModel securityName
group   notConfigGroup v1           notConfigUser
group   notConfigGroup v2c           notConfigUser
# Third, create a view for us to let the group have rights to:
# Make at least  snmpwalk -v 1 localhost -c public system fast again.
#       name           incl/excl     subtree         mask(optional)
view    systemview    included   .1.3.6.1.2.1.1
view    systemview    included   .1.3.6.1.2.1.25.1.1
# Finally, grant the group read-only access to the systemview view.
#       group          context sec.model sec.level prefix read   write  notif
access  notConfigGroup ""      any       noauth    exact  systemview none none
…
# It is also possible to set the sysContact and sysLocation system
# variables through the snmpd.conf file:
syslocation Unknown (edit /etc/snmp/snmpd.conf)
syscontact Root <root@localhost> (configure /etc/snmp/snmp.local.conf)

# Added for support of bcm5820 cards.
pass .1.3.6.1.4.1.4413.4.1 /usr/bin/ucd5820stat
```

SNMP

- snmpdの設定

/etc/snmp/snmpd.conf 設定例

```
com2sec notConfigUser  default       public
com2sec LocalGroup     127.0.0.1     public            ---(1)
…
group   notConfigGroup v1           notConfigUser
group   notConfigGroup v2c          notConfigUser
group   lovalSecurity  v1c     localGroup              ---(2)
group   lovalSecurity  v2c     localGroup
group   lovalSecurity  usm     localGroup

view    systemview   included   .1.3.6.1.2.1.1         ---(3)
view    all          included   .1 80

access  notConfigGroup ""     any      noauth    exact  systemview none none
access  localSecurity  ""     any      noauth    exact  all none none              ---(4)
…
syslocation Second at Rack #1          ---(5)
syscontact <admin@sensaba.net>         ---(6)

# Added for support of bcm5820 cards.
pass .1.3.6.1.4.1.4413.4.1 /usr/bin/ucd5820stat        ---(7)
```

SNMP

- snmpdの設定

(1)～(4)は、SNMPマネージャからのアクセス制御に関する設定
(5)～(6)は、SNMPエージェントが管理する情報のうち、あらかじめ設定しておく必要のある情報
(7)は、SNMPマネージャからの該当OIDに対する呼び出しの処理を外部コマンドに依頼するという設定ですが、標準的には特に役割がありません。

(1) マネージャアドレスとコミュニティアドレスからグループへのマッピング
Com2sec <Group Name> <Manager IP> <Community>
指定したIPアドレスとコミュニティからアクセスしたきたマネージャに対応するセキュリティグループを設定します。

(2) グループに対するセキュリティモデルの設定
group <Group Name> <Model> <Security>
セキュリティグループに対して、適用するセキュリティモデルを設定します。

(3) Viewの設定
view <View Name> <View Type> <SubTree> <Mask>
viewとは参照可能なMIBサブツリーの規定です。Maskは省略可能です。

(4) グループに対するアクセス権の設定
access <Group Name> <Context> <Model> <Level> <Prefix> <Read> <Write> <Notify>
グループセキュリティ、モデルセキュリテイI、Viewを関連づけて、一連のセキュリティを定義します。

(5) システムの場所に対する設定
syslocation <LocationSTRING>
system.sysLocation.0で参照可能なシステムの配置場所に関する情報の設定です。

(6) システム管理者のアドレス
syslocation <Contact Info>
system.sysLocation.0で参照可能なシステム管理者に関する情報の設定です。

SNMP

- ucdavisパラメータの設定
  - NET-SNMP特有の機能であるucdavisパラメータを設定すると、さらに多くの情報を取り出すことができます。

プロセスの状態監視
proc <Name> <Max> <Min>
<Name>で指定したプロセスの状態を監視します。prTable.prEntry.prErrorFlag(.1.3.6.1.4.1.2021.2.1.100.p)では、該当プロセスが<Min>個以上存在し<Max>個以下であることを管理することができます。

```
proc sendmail 20 1
```

disk
disk <Path> <Minimal Space>
<Path>で指定したディレクトリにマウントさえているディスクパーティションの状態を監視します。dskTable.dskEntry.dskErrorFlag (.1.3.6.1.4.1.2021.9.1.100.d)では、ディスク容量が正常範囲内であることを管理することができます。

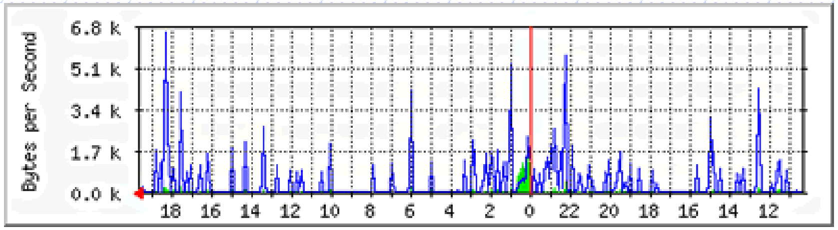
```
disk /var 10%
```

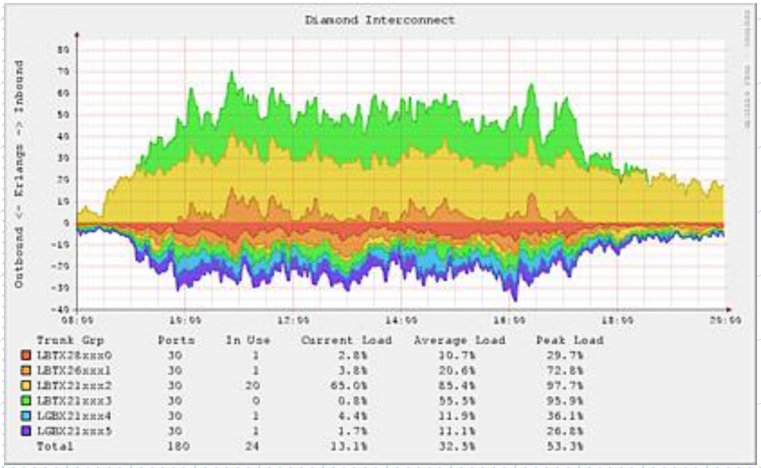
load
load <Max1> <Max5> <Max15>
システムのロードアベレージを管理するための設定です。<Max1>、<Max5>、<Max15>は、それぞれuptimeコマンドで出力される1分平均、5分平均、15分平均のロードアベレージと比較されます。laTable.laErrorFlag (.1.3.6.1.4.1.2021.10.1.100.x)では、ロードアベレージがこの値の範囲内であることを管理することができます。

```
load 15 12 10
```

グラフ化ツール

- MRTGとRRDtool
  - MRTG（Multi Router Traffic Grapher）は、SNMPエージェントから取得したデータを加工してグラフ化するツールです。
  - MRTGはHTML形式のページを作成するため、Apacheなど HTTPデーモンが動作しているサーバで利用すれば、X Window Systemに頼ることなくWebブラウザ経由でグラフの閲覧が可能です。
  - RRDtool (Round Robin Database tool)を使用するともっと複雑なデータ加工とグラフィカルなグラフ化が可能です。
  - RRDtoolでは、データベースのサイズは増加せず常に一定量となります。





サーバ設定確認

- サーバの構成および設定の確認
  - サーバの設定が完了したら、サーバの構成情報や設定情報を一通り確認します。
  - サーバ台数が多い場合にはコマンドを打つのが大変ですので、一つのシェルにまとめてしまうと便利です。
  - SSHのログインから始まって一連のコマンドを自動化するアプリを作成すると、より便利です。

サーバ設定確認

■ ハードウェア情報

```
# cat /proc/cpuinfo  ← CPU
# cat /proc/meminfo  ← メモリ
# cat /proc/partitions  ← ディスクのパーティション
# df -m  ← ファイルシステム
# cat /etc/fstab ← ファイルシステムの静的な情報
```

■ ネットワーク情報

```
# hostname
# ifconfig -a
# route
# cat /etc/sysconfig/network
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
# cat /etc/sysconfig/network-scripts/ifcfg-eth1
# cat /etc/sysctl.conf
# netstat -a
```

サーバ設定確認

■ アカウント情報

```
# cat /etc/passwd
# cat /etc/shadow
# cat /etc/group
```

■ OS情報およびシステム情報

```
# cat /etc/redhat-release
# uname -a
# chkconfig --list
# ls /etc/rc.d/rc3.d/
# ls /etc/rc.d/rc5.d/
# cat /etc/inittab
# cat /etc/hosts
# cat /etc/resolv.conf
# cat /etc/nsswitch.conf
# cat /etc/ntp.conf
# /usr/sbin/ntpq -p
# ls /etc/init.d/
# cat /etc/xinetd.conf
```

サーバ設定確認

セキュリティ設定情報

```
# cat /etc/hosts.allow
# cat /etc/hosts.deny
# iptables --list
# cat /etc/sysconfig/iptables-config
```

cron、ログ情報

```
# crontab -l
# ls /etc/cron.daily/
# ls /etc/cron.hourly/
# ls /etc/cron.monthly/
# ls /etc/cron.weekly/
# cat /etc/cron.daily/logrotate
# ls /etc/logrotate.d
# ls -l /var/log
# ls -l /var/log/httpd
# ls -l /var/run
```

Linuxの長所と短所

- Linuxの長所
  - RPM等のパッケージが充実しているので管理がしやすいです。
  - 企業、個人で幅広く使用されているので、ドキュメントが充実しています。
  - 最新の商用アプリの多くがSolarisやBSDよりもまずLinuxに対応しています。
  - OS自体はフリーです。Linux対応のハードウェアの種類も多く低価格で導入が容易にできます。
- Linuxの短所
  - カーネルを含めてアップデートが頻繁にリリースされるため運用が煩雑になります。
  - 高信頼性を要求される基幹向けサーバとしては実績が少ないです。（重要な箇所にオープンソースで本当に大丈夫かという先入観がまだ広くあります。）
  - OS自体はフリーでも、サポートをつけると意外と高くつきます。

Security

May 28th ,2006

Security　目次

- サーバセキュリティ
  - 不要サービス停止
  - ファイアウォール
  - SSH鍵認証
  - ・・・
- WEBセキュリティ
  - SSL
  - ログイン認証
  - メタ文字無効化
  - ・・・
- メールセキュリティ
  - 送信ドメイン認証
  - POP before SMTP
  - アンチウィルス
  - ・・・

サーバセキュリティ

- サーバのセキュリティ項目
  - サーバ分析
  - 不要サービス停止
  - ファイアウォール
  - SSHの鍵認証
  - 改ざん検知
  - rootkitの検出

WEBセキュリティ

- WEBのセキュリティ項目
  - SSL
  - ログイン認証
  - メタ文字無効化
  - Apacheのセキュリティ設定
  - WEBアプリケーションのセキュリティ

メールセキュリティ

- メールのセキュリティ項目
  - 送信ドメイン認証
  - POP before SMTP
  - APOP
  - SMTP Auth
  - POP3 over SSL / IMAP4 over SSL
  - SMTP over TLS
  - アンチウィルス
  - アンチスパム