LinuxAcademy

Linuxサーバ構築手順書

OSのインストールから基本サーバ構築まで
（DNS・SSH・Web・メール・パケットフィルタリング）

K.E
2014/04/19

目次

（0）はじめに

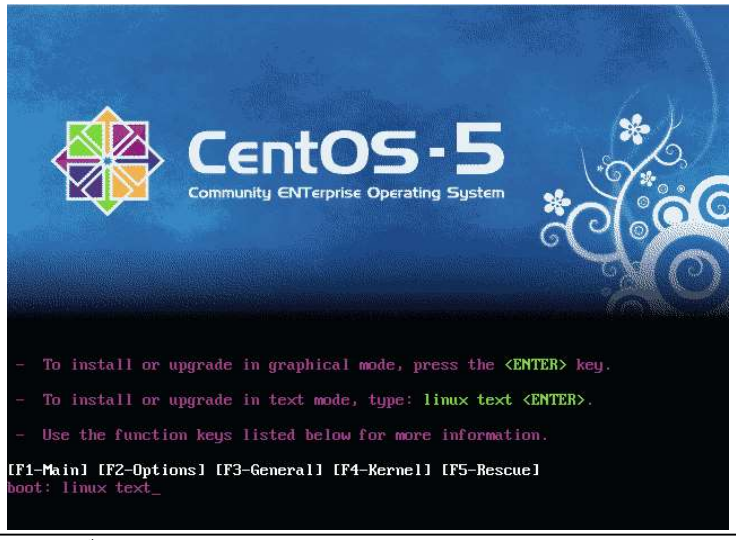
　今回、リナックスアカデミーにおける、「サーバ構築演習」という講座内で、OSのインストールからサーバ構築を通し、最後にセキュリティ設定を一通り行うことになった。
　LinuxベーシックからLinuxマスターコースで学んだことをアウトプットしながら、サーバの構築を行い、その構築の手順書を作成することになった。
　今回は、Linuxのインストールから始め、DNS、SSH、Web、メールサーバの構築を行い、最後にパケットフィルタリングによるセキュリティ設定も行った。

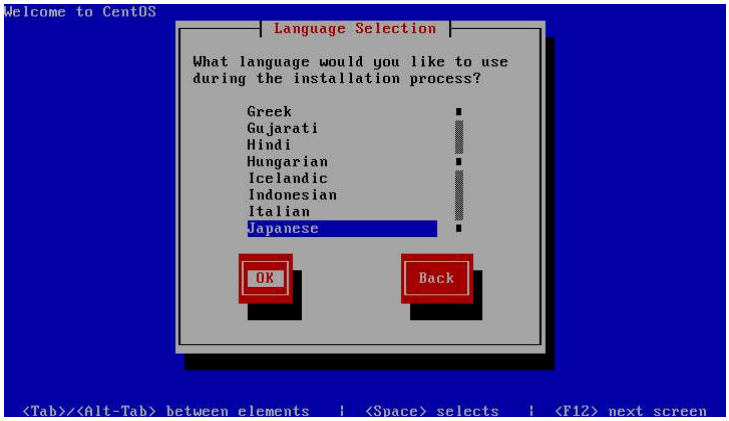
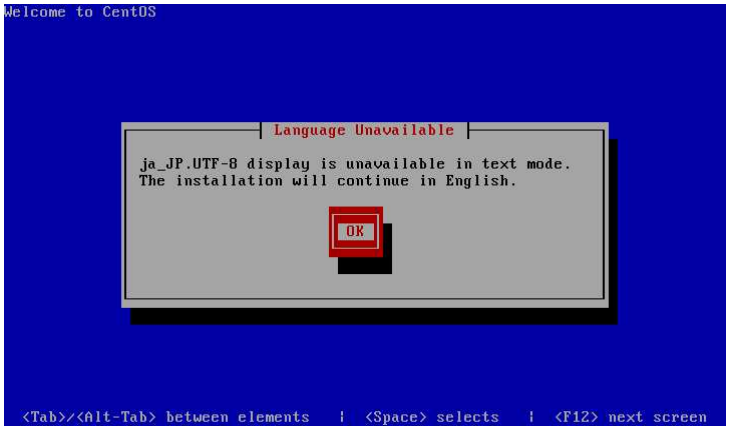
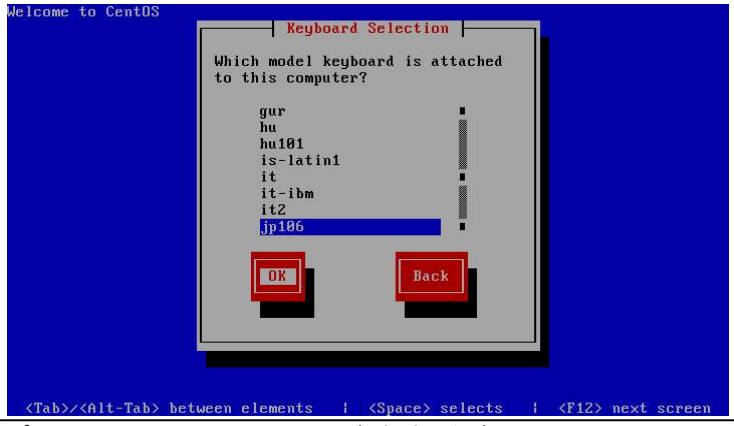
前提として、以下の知識・技術があることが望ましい。
・Windows操作経験者（熟練レベルでなくてよい）
・簡単なvi操作
・基本コマンド（cd, cp, ls, pwd, mv, cat, tail, chmod, dig, nslookup, pingなど）
・DNSサーバ、Webサーバ、メールサーバなどの基本的知識

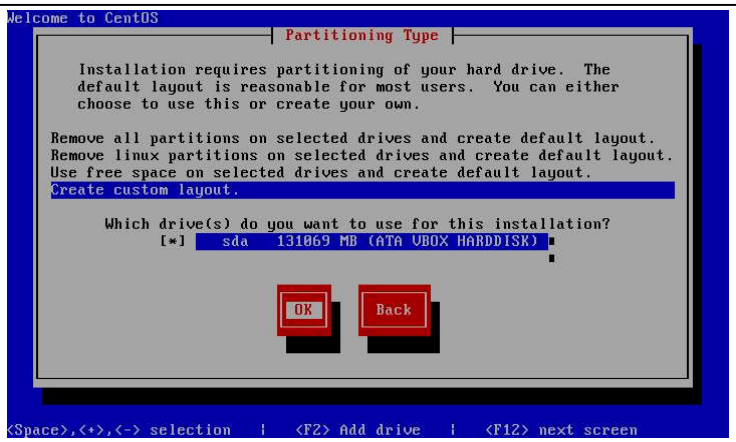
また、サーバ構築演習において、サーバ構築を行うに当たって、LinuxベーシックコースやLinuxマスターコースで使用したテキスト以外にも以下の文献も利用した。
参考文献「改訂新版 28日で即戦力！ サーバ技術者養成講座」

（1）Linuxインストール手順

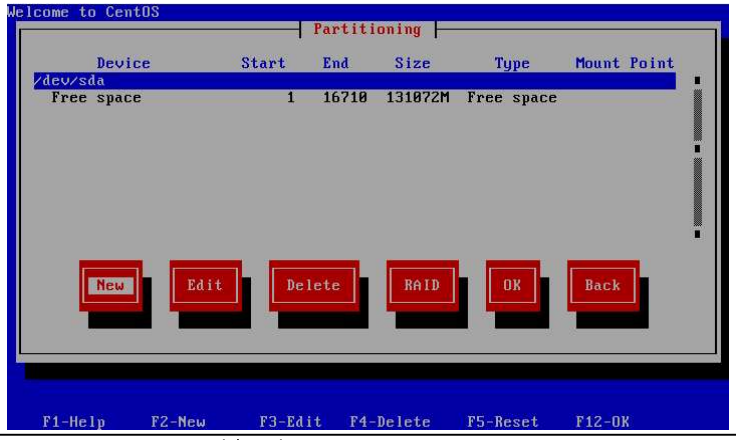
サーバ構築の前提として、OSのインストールを行いながら、パッケージの選択、インストール後の各種システム設定を行う。
まず、現在立ち上げているパーソナルコンピュータ（以下、PC）にCentOS 5.3がインストールされているDVDをDVDドライブへ入れてください。その後、再起動を行うと以下の画面が立ち上がります。

番号	設定する項目	設定手順
1	CentOSインストール起動画面 	左下に「boot:」とあるので、そこに「linux text」と入力し、[Enter]を押す。
2	CDメディアテストメッセージ 	「Skip」を選択する。 ※[Tab]キーまたは[←][↓][↑][→]のカーソルキーで選択し、[Enter]で決定する。
3	CentOS画面 	そのまま「OK」を選択する。

4	言語選択 	「Japanese（日本語）」を選択し、「OK」を押す。
5	日本語はサポートされていない旨の警告 	特になにもせず「OK」を押す。 ※インストールは「linux text」モードで設定のため、設定時のみは日本語は使用できないという意味。
6	キーボードの種類の選択 	「jp106」（日本語）を選択し、「OK」を押す。
7	パーティションレイアウト設定方法の選択画面	今回は自身でカスタムしてパーティションを構成するので「Create custom layout.」を選択し、「OK」を押す。



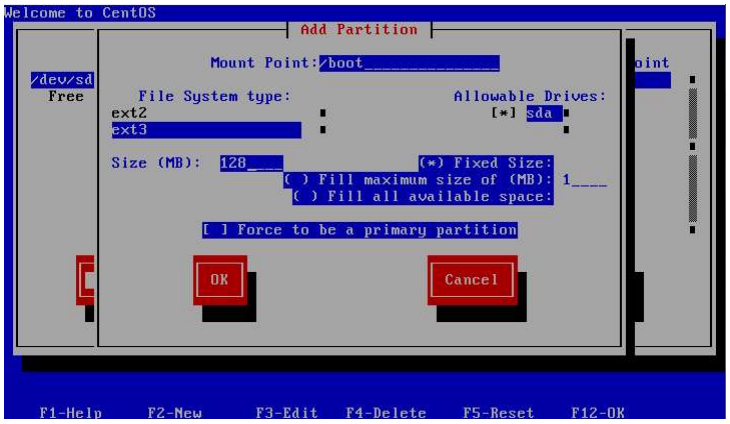
8 現在のハードディスクの領域一覧が表示される。



もし、Mount Pointに何か表示されている場合は、カーソルキーを使ってフォーカスを移動して、[Tab]キーを使ってさらに[Delete]を使って既存のパーティションを削除する。

パーティションをすべて削除したら、左記のように「new」を選択する。

9 ファイルシステム割り当て

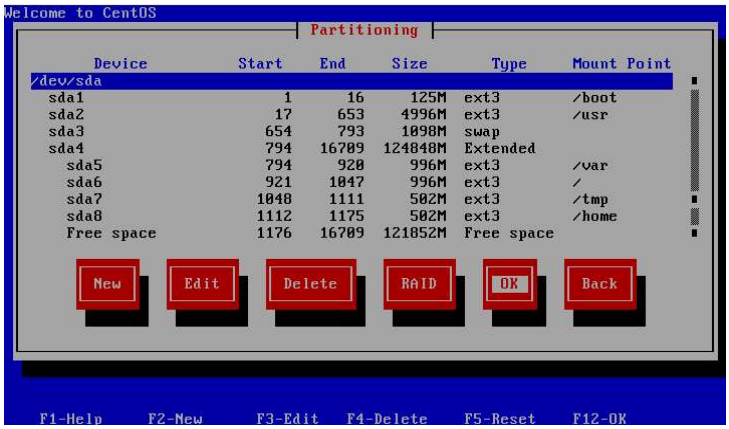
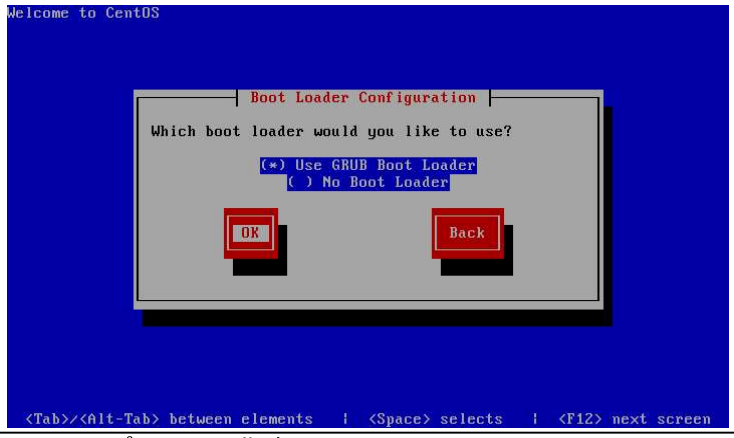
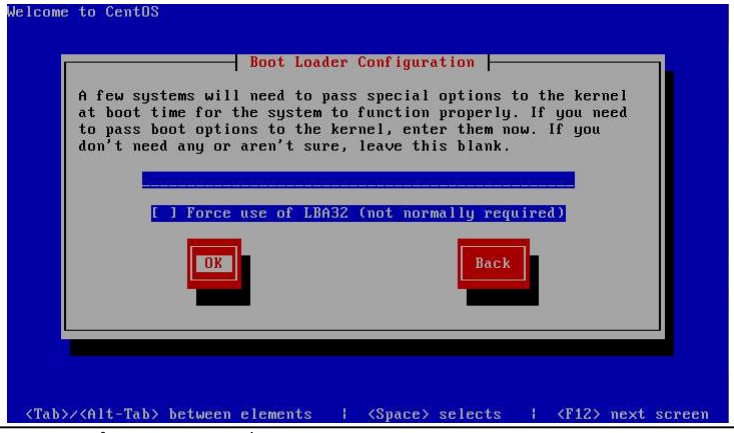


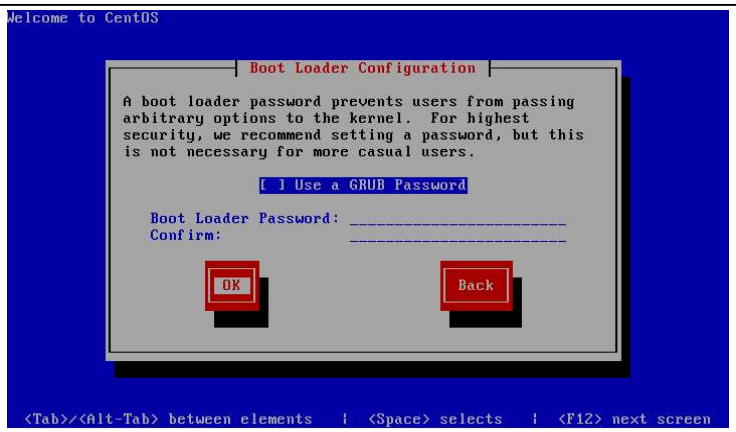
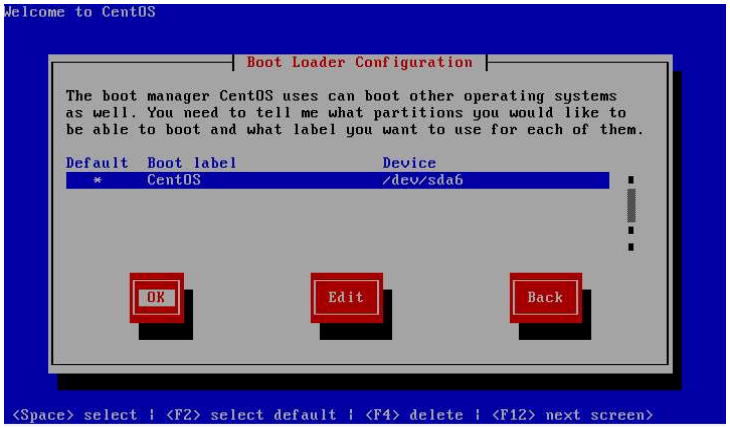
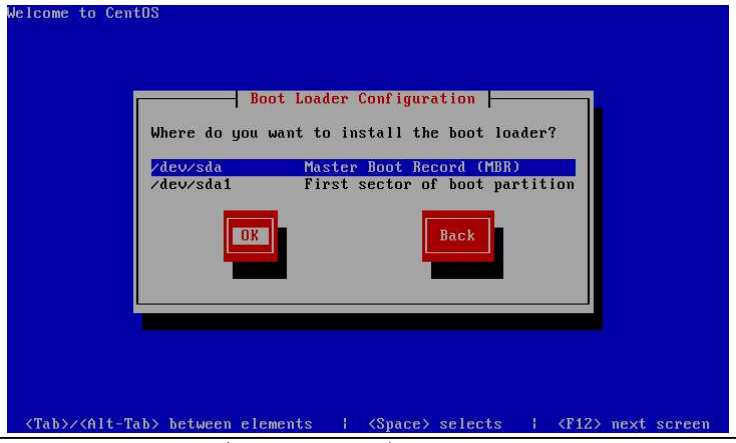
以下の表を参考に、左記のように、Mount Pointに「/boot」、File System Typeに「ext3」、Sizeに「128(MB)」を設定する。

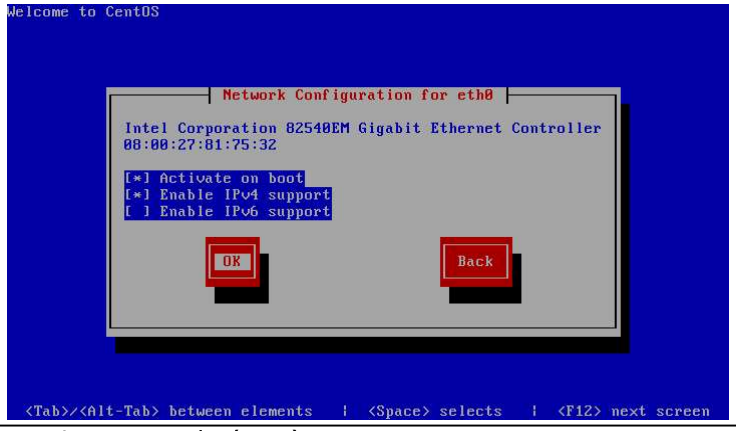
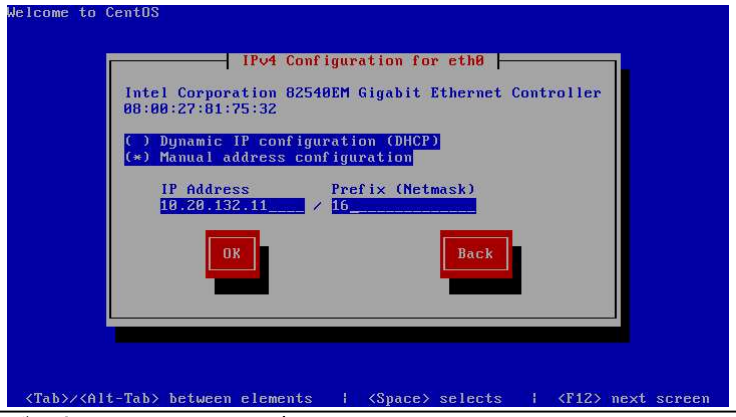
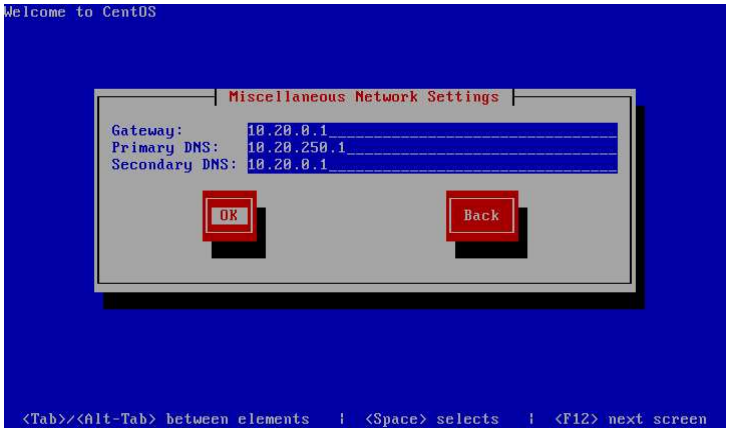
そのほかのパーティションは以下の表の通りに設定してください。

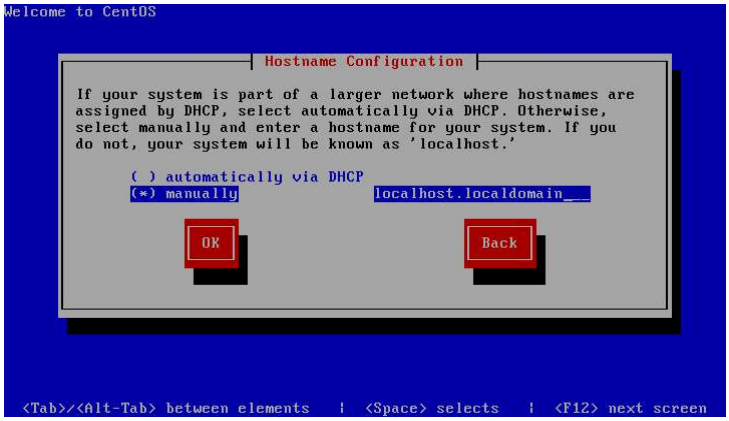
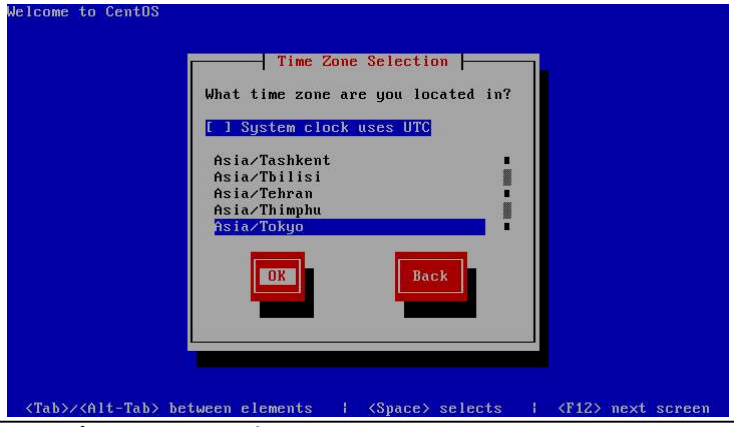
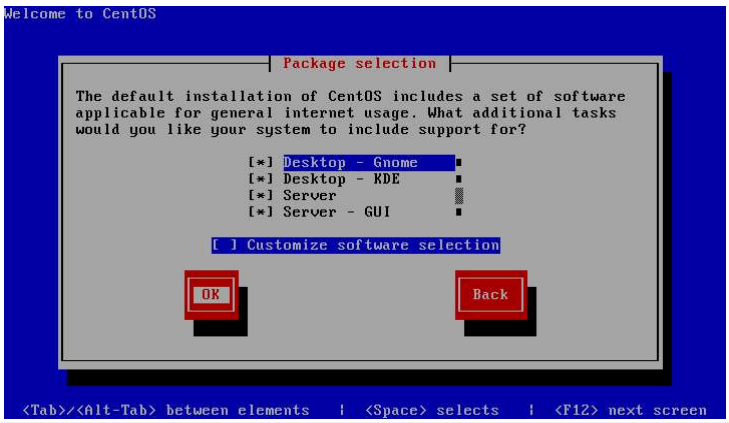
Mount Point	File System type	Size（MB）
/boot（これは設定済み）	ext3	128
/	ext3	1,000
/usr	ext3	5,000
/home	ext3	500
/var	ext3	1,000
/tmp	ext3	500
（何も入力しない）	swap	1,100

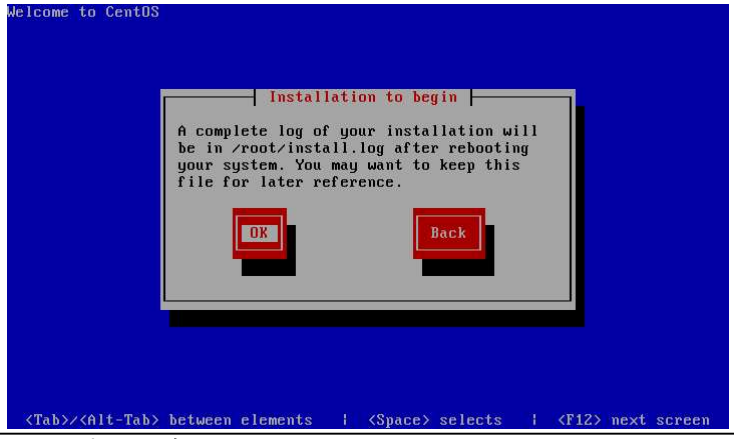
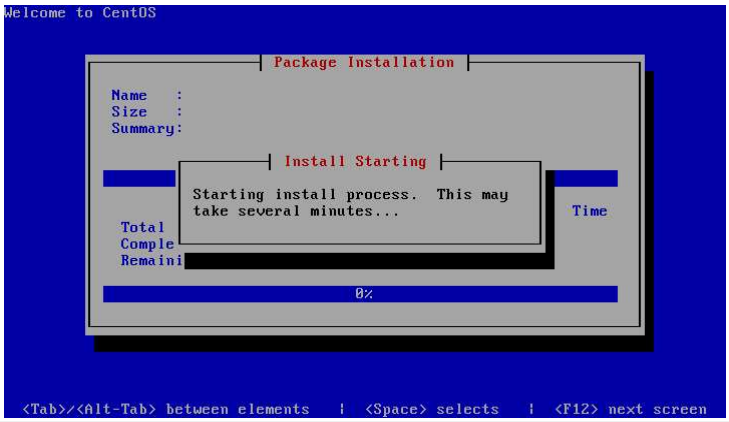
10 割り当てたファイルシステム一覧 左記のように設定されていれ

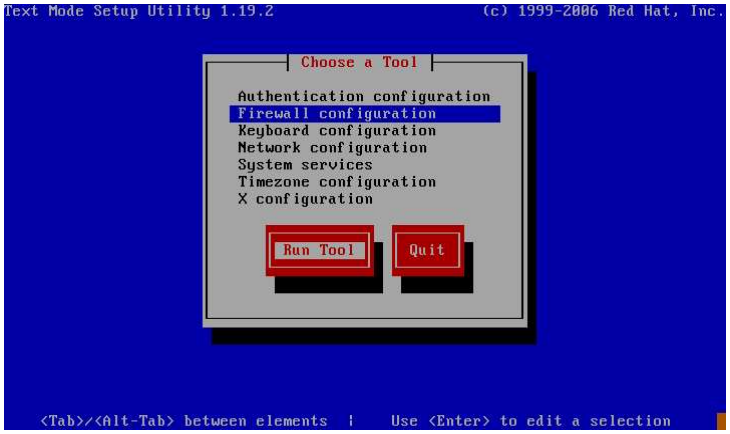
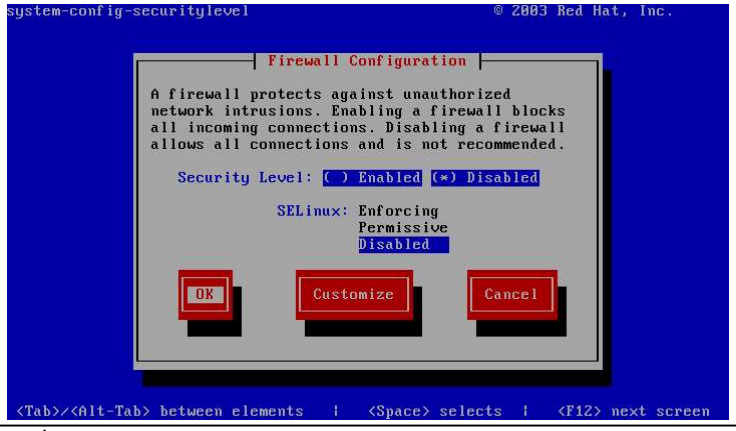
		ば「OK」を押す。
11	ブートローダプログラム（GRUB）のインストール 	今回は何もせず、「OK」を押す。
12	GRUBオプションの指定 	オプションも指定できますが、これも何もせず「OK」を押す。
13	GRUBパスワード設定	GRUBのパスワードを設定できるが、これも何も入力せず「OK」を押す。

		
14	GRUBのインストール 	これも何もせず「OK」を押す。
15	GRUBインストール先 	そのまま「OK」を押す。
16	ネットワークの設定についての確認画面	「Yes」を選択。

		
17	NIC（Network Interface Card）の詳細設定 	左記の様に「Activate on boot」と「Enable IPv4 support」について[*]として「OK」を押す。 （「Enable IPv6 support」の*ははずす）
18	IPアドレスの設定（eth0） 	DHCPは設定せず、Manual address configurationに（*）をつけ、 IPアドレス：10.20.132.11 ネットマスク：16 （ネットマスクは16とすると、255.255.0.0となる）
19	ゲートウェイとDNS設定 	左記のように、ゲートウェイとDNSを設定する。 ゲートウェイ：10.20.0.1 Primary DNS：10.20.250.1 Secondary DNS：10.20.0.1

20	ホスト名の設定 	Automatically via DHCPの*をはずし、manuallyに（*）として選択する。 ホスト名は任意（この時点で、h011.s132.la.netとしてもよいし、後ほど設定も可能）
21	タームゾーンの設定 	特に問題なければ、System clock uses UTCのチェックをはずし、「Asia/Tokyo」を指定する。
22	Rootパスワードの設定 	rootのパスワードを設定する。（忘れないようにメモを取りましょう）
23	パッケージを選定 	以下の4点にチェックを入れて、「OK」を押す。 [*]Desktop-Gnome [*]Desktop-KDE [*]Server [*]Server-GUI

24	インストール開始画面 	インストールに必要な質問はこれで終わりです。 問題なければ「OK」を押してインストール開始してください。 ※インストールには時間がかかります。
25	インストール中 	インストールには時間がかかります。
26	インストール完了 	なお、インストールが終わると再起動「Reboot」を求められるので、「Reboot」を押す。
27	各種設定画面	Firewall configurationを選択し、「Run Tool」を押す。

		
28	Firewall・SELinuxの設定 	Security Levelを「Disable」に、SELinuxを「Disable」に設定し、「OK」を押す。
29	再度メニュー 	「Quit」を押す。

これで、一通りのインストールは終了となります。

インストール後の設定

インストールしたら、以下の設定を行います。
（1）ログインの実施
rootでログインをし、パスワードを入力します。

```
CentOS release 5.10 (Final)
Kernel 2.6.18-371.6.1.el5 on an i686

h011 login:root
Password:
```

※パスワードはマスクされているため、画面に表示されませんが、入力はされています。

（2）startxコマンドでGUIモードにする。（これは任意）
startxコマンドを投入し、X Window Systemを立ち上げ、GUI画面にしておきます。GUIにしておくと後々設定が容易となります。

```
[root@h011 ~]# startx
```

すると、GUI画面になります。

（3）GNOME端末を立ち上げる。（[アプリケーション]->[アクセサリ]->[GNOME 端末]）
すると、以下の端末が立ち上がる。（GNOME端末の背景色や画像は各自で設定できます。）

（4）updateをしておく。
パッケージは最新のものではないため、アップデートを行っておきます。（時間がかかります）

```
[root@h011 ~]# yum -y update
```

※アップデートは時間がかかるため、別にGNOME端末を立ち上げて、（5）以降の作業を実施しても構いません。

（5）ユーザ（student）の追加
useraddコマンドで、studentユーザを追加しておきます。さらにパスワードも設定します。
（パスワードは設定確認のため2回求められます。）

```
[root@h011 ~]# useradd student
[root@h011 ~]# passwd student
```

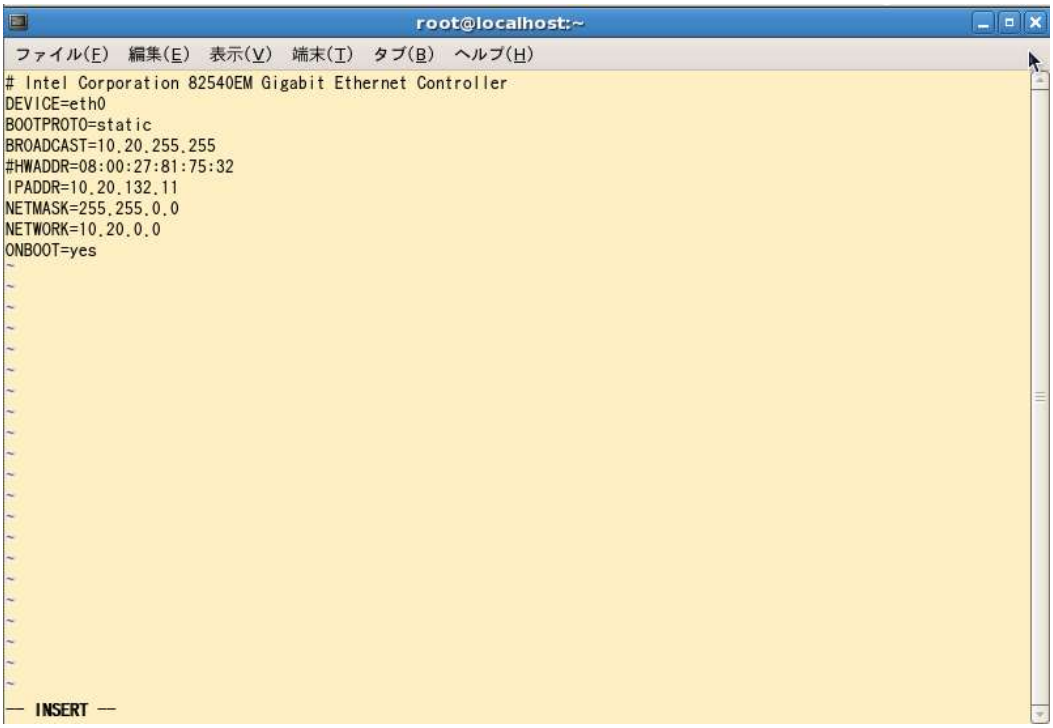
（6）viをvimに設定する。
テキストエディタであるviをvimに設定変更しておく。

```
[root@h011 ~]# alias vi='vim'
```

```
[root@h011 ~]# alias
alias cp='cp -i'
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias mv='mv -i'
alias rm='rm -i'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
```

（7）MACアドレスの消去
以下のコマンドを投入し、HWADDRの行を消去する。（または下記のように先頭に#をつけてもよい）

```
[student@h011 ~]$ vi /etc/sysconfig/network-scripts/ifcfg-eth0
```



（2）DNSサーバ構築

DNSサーバ（ネームサーバ）は、ホストの識別子を利用者が理解しやすい形式（ドメイン名）とコンピュータが理解しやすい形式（IPアドレス）を対応付けるサーバである。

まず、bind-chrootは削除する。（使用しないため）

```
[root@h011 etc]# rpm -e bind-chroot
```

設定ファイルを開き、下記の情報を記載する。（viでnamed.confを新規作成）

```
[root@localhost ~]# vi /etc/named.conf
```

viで下記の情報を記載する。（なお、下記で青文字の記載は任意）

```
// Sample named.conf for LinuxAcademy
//
options
{
        directory "/var/named/named";          // the default
};

// Root Server キャッシュ
zone "." IN {
        type hint;
        file "named.root";
};

// localhost に対する正引きゾーン
zone "localhost" IN {
        type master;
        file "localdomain.zone";
};

// localhost の逆引きゾーン
zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "localhost.zone";
};

// 正引きサンプル
// IP が 10.20.xx.yy の場合、t0xx0yy.la.net ドメインを構築
zone "t132011.la.net" IN {
        type master;
        file "t132011.zone";
};
```

サンプルファイル/usr/share/doc/bind-9.3.6/sample/var/named/ を/var/named直下にコピーする。

```
[root@h011 named]# cp -r /usr/share/doc/bind-9.3.6/sample/var/named/ /var/named
```

次に、viで/var/named/直下に/var/named/t132011.zoneのゾーンファイルを作成する。

```
[root@localhost ~]# vi /var/named/t132011.zone
```

※赤文字のところは今日の日付を記載。

```
$TTL    86400
t132011.la.net.         IN SOA ns.t132011.la.net. root.t132011.la.net. (
                        2014040501      ; serial (d. adams)
                        3H              ; refresh
                        15M             ; retry
                        1W              ; expiry
                        1D )            ; minimum
t132011.la.net.         IN      NS      ns.t132011.la.net.
t132011.la.net.         IN      MX      10      smtp.t132011.la.net.
ns.t132011.la.net.      IN      A       10.20.132.11
smtp.t132011.la.net.    IN      A       10.20.132.11
h132.t132011.la.net.    IN      A       10.20.132.11
www.t132011.la.net.     IN      CNAME   h011.t132011.la.net.
```

次に作成したゾーンファイルの現在の所有者・グループを確認。

```
[root@h011 named]# ls -l /var/named/named/t132011.zone
```

```
-rw-r--r-- 1 root root 745  4月  5 10:53 /var/named/named/t132011.zone
```

所有者・グループがrootとなっているので、これをnamedに変更する。

```
[root@h011 named]# chown named:named /var/named/named/t132011.zone
```

```
-rw-r--r-- 1 named named745  4月  5 10:53 /var/named/t132011.zone
```

次に、/etc/resolv.confファイル（リゾルバファイル）に下記情報を追記。

```
[root@h011 ~]# vi /etc/resolv.conf
```

```
[root@h011 ~]# more /etc/resolv.conf
domain s132.la.net
nameserver 10.20.250.1
nameserver 10.20.0.1
nameserver 10.20.132.11
```

これにより、自身の各種サーバ（Webサーバ・メールサーバ等）において、名前解決をする際に、どのDNSサーバを見ればよいか指定できる。

下記コマンドを実行し、以下の通りNOERRORと出力すれば成功。

```
[root@h011 named]# dig h011.t132011.la.net
```

```
; <<>> DiG 9.3.6-P1-RedHat-9.3.6-20.P1.el5_8.6 <<>> h011.t132011.la.net
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40598
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;h011.t132011.la.net.           IN      A

;; ANSWER SECTION:
h011.t132011.la.net.    86400   IN      A       10.20.132.11
```

```
;; AUTHORITY SECTION:
t132011.la.net.         86400   IN      NS      ns.t132011.la.net.

;; ADDITIONAL SECTION:
ns.t132011.la.net.      86400   IN      A       10.20.132.11

;; Query time: 0 msec
;; SERVER: 10.20.132.11#53(10.20.132.11)
;; WHEN: Sat Apr  5 14:52:08 2014
;; MSG SIZE  rcvd: 86
```

上記コマンドで、赤文字のところをwww.t132011.la.netやsmtp.t132011.la.netと変更してもNOERRORとなるかを確認してみてください。

上記確認後、リゾルバへ下記情報を追記。

```
[root@h011 named]# /etc/resolv.conf
```

```
domain h011.s132.la.net
nameserver 10.20.132.11
nameserver 10.20.250.1
nameserver 10.20.0.1
```

最後に、次回以降起動時からnamedを起動するように設定しておく。

```
[root@h011 slaves]# chkconfig --list named
named           0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@h011 slaves]# chkconfig --level 2345 named on
[root@h011 slaves]# chkconfig --list named
named           0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

（3）SSHサーバ構築

サーバ上でSSHサーバを構築する。この設定を行うことで、データセンタなどのサーバが設置してあるところに対して、遠隔で管理できます。

まず、sshdが起動していることを確認します。

```
[root@localhost ~]# /etc/init.d/sshd status
openssh-daemon (pid  2711) を実行中...
```

（※なお、下記のコマンドの方法で確認しても問題ない）

```
[root@localhost ~]# service sshd status
openssh-daemon (pid  2711) を実行中...
```

次回以降から、自動で起動されるよう、chkconfigコマンドで設定しておく。

```
[root@localhost ~]# chkconfig --list sshd
sshd            0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@h011 slaves]# chkconfig --level 2345 named on
[root@h011 slaves]# chkconfig --list sshd
sshd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

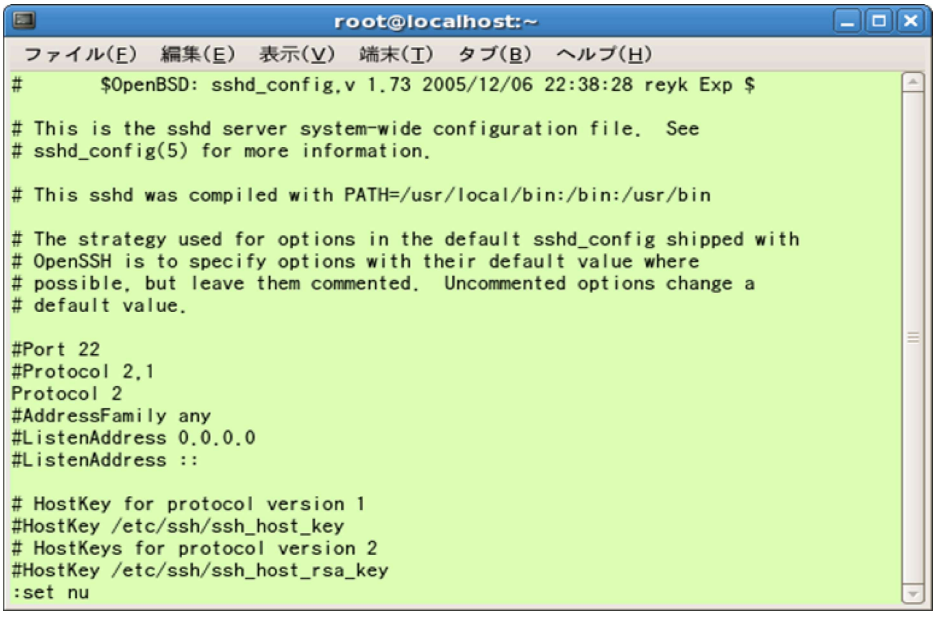
次に、認証テストのために、ユーザアカウントを作成しておきます。ユーザ名は自分の名前で構いません。

```
[root@localhost ~]# useradd enomoto
[root@localhost ~]# passwd enomoto
Changing password for user enomoto.
New UNIX password:
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

SSHの設定を行うため、設定ファイル（/etc/ssh/sshd_config）を開く。

```
[root@localhost ~]# vi /etc/ssh/sshd_config
```

※設定ファイルをviで開き、下記のように、「:set number」と入力すると、設定ファイルの各行に行番号が挿入される。また、コマンドモードで「:44」と入力するとその行に移動できる。




```
38 #LoginGraceTime 2m
39 #PermitRootLogin yes
40 #StrictModes yes
41 #MaxAuthTries 6
42
43 #RSAAuthentication yes
```

```
44 PubkeyAuthentication yes
45 #AuthorizedKeysFile     .ssh/authorized_keys
46
47 # For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
48 #RhostsRSAAuthentication no
49 # similar for protocol version 2
50 #HostbasedAuthentication no
```

.sshへ移動し、公開鍵id_dsa.pubが生成されていることを確認。

```
[root@localhost ~]# /etc/init.d/sshd restart
sshd を停止中:                                [  OK  ]
sshd を起動中:                                [  OK  ]
```

自ユーザとしてログイン（この例ではユーザはenomoto）し、自分の公開鍵を生成する。

```
[enomoto@localhost ~]$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/enomoto/.ssh/id_dsa):
Created directory '/home/enomoto/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/enomoto/.ssh/id_dsa.
Your public key has been saved in /home/enomoto/.ssh/id_dsa.pub.
The key fingerprint is:
3b:a5:af:a9:46:92:8f:35:14:e0:45:5d:1e:fa:6a:43 enomoto@localhost.localdomain
```

.sshへ移動し、公開鍵id_dsa.pubが生成されていることを確認。

```
[enomoto@localhost ~]$ ls -a
.  ..  .bash_logout  .bash_profile  .bashrc  .kde  .mozilla  .ssh  .xauthDswzo9
[enomoto@localhost ~]$ cd .ssh
[enomoto@localhost .ssh]$ ls
id_dsa  id_dsa.pub
```

自身の秘密鍵id_dsaを使用してログインできるよう、自身の公開鍵を.sshディレクトリ以下にauthorized_keysファイルに追記する。

```
[enomoto@localhost .ssh]$ cat ./id_dsa.pub >> ~/.ssh/authorized_keys
```

グループ内の他のユーザに変更されないように、アクセス権を600（rw-------）にする。

```
[enomoto@localhost .ssh]$ chmod 600 authorized_keys
[enomoto@localhost .ssh]$ ls -al
合計 5
drwx------ 2 enomoto enomoto 1024  3月 29 14:37 .
drwx------ 5 enomoto enomoto 1024  3月 29 14:30 ..
-rw------- 1 enomoto enomoto  619  3月 29 14:37 authorized_keys
-rw------- 1 enomoto enomoto  744  3月 29 14:30 id_dsa
-rw-r--r-- 1 enomoto enomoto  619  3月 29 14:30 id_dsa.pub
```

sftpでstudentとしてログインする。そして、自身のホームディレクトリにid_dsa.pubを転送する。

```
[enomoto@localhost .ssh]$ sftp student@localhost
Connecting to localhost...
The authenticity of host 'localhost (127.0.0.1)' can't be established.
RSA key fingerprint is d5:b4:1f:d2:17:34:35:30:4d:40:7a:5e:a8:05:b7:94.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
```

```
student@localhost's password:
sftp> put id_dsa.pub
Uploading id_dsa.pub to /home/student/id_dsa.pub
id_dsa.pub                                    100%  619     0.6KB/s   00:00
```

studentユーザへ公開鍵の転送が終わったら、sftpからはquitコマンドで抜ける。

```
sftp> quit
```

次にsshコマンドを用いて、studentユーザとしてUNIXパスワードとしてログインを行い、studentユーザに公開鍵登録を行う。

```
[enomoto@localhost ~]$ ssh student@localhost
student@localhost's password:
Last login: Sat Mar 29 14:22:47 2014 from localhost.localdomain
```

先ほど自分のユーザからsftpした自分の公開鍵がstudentホームディレクトリに転送されていることも確認。

```
[student@localhost ~]$ ls
Desktop  SSH_Manual.odt  Xen_install.odt  id_dsa.pub  構築演習_install.odt
```

.sshディレクトリを作成し、.sshディレクトリへ移動。

```
[student@localhost ~]$ mkdir .ssh
[student@localhost ~]$ cd .ssh
[student@localhost .ssh]$ ls -a
.  ..
```

転送してきた自分の公開鍵をauthorized_keysに登録（追記）する。

```
[student@localhost .ssh]$ cat ../id_dsa.pub >> authorized_keys
[student@localhost .ssh]$ ls -al
合計 4
drwxrwxr-x  2 student student 1024  3月 29 14:50 .
drwx------ 24 student student 1024  3月 29 14:46 ..
-rw-rw-r--  1 student student  619  3月 29 14:50 authorized_keys
```

.sshのパーミッションを700へ、authorized_keysのパーミッションを600へ変更する。

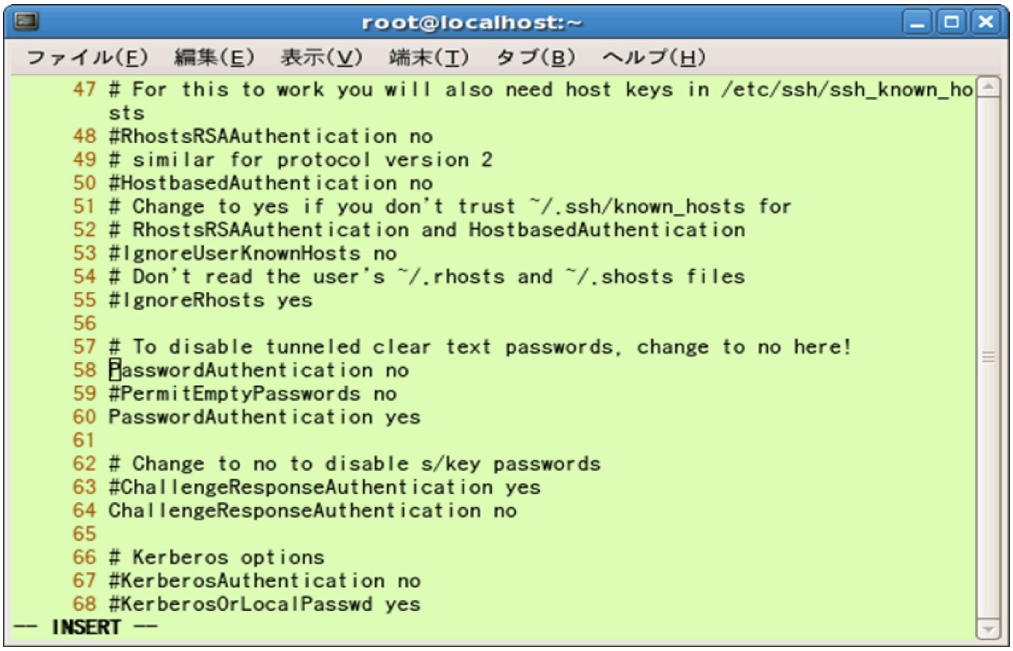
```
[student@localhost .ssh]$ pwd
/home/student/.ssh
[student@localhost .ssh]$ chmod 700 .
[student@localhost .ssh]$ cd
[student@localhost ~]$ ls -al
        ※※※ファイル多数のため省略※※※
drwx------  2 student student  1024  3月 29 14:50 .ssh
[student@localhost .ssh]$ chmod 600 authorized_keys
[student@localhost .ssh]$ ls -l .
合計 1
-rw------- 1 student student 619  3月 29 14:50 authorized_keys
```

一旦ログアウトし、再度studentユーザへssh接続する。その際に自身の秘密鍵のパスフレーズが問われる。その際にプロンプトが帰ってくれば成功、再度パスワードを求められたら、登録鍵のファイル名やパーミッションを確認してください。

```
[enomoto@localhost ~]$ ssh student@localhost
Enter passphrase for key '/home/enomoto/.ssh/id_dsa':
Last login: Sat Mar 29 14:43:55 2014 from localhost.localdomain
[student@localhost ~]$
```

UNIXパスワード認証はセキュリティ上安全でないため、/etc/ssh/sshd_config内の設定で「#PasswordAuthentication = yes」の行（58行目）で、#を外し、yesからnoに変更する。

```
[root@localhost ~]# vi /etc/ssh/sshd_config
```



最後に、sshdを実行し、設定を反映させる。

```
[root@localhost ~]# service sshd restart
```

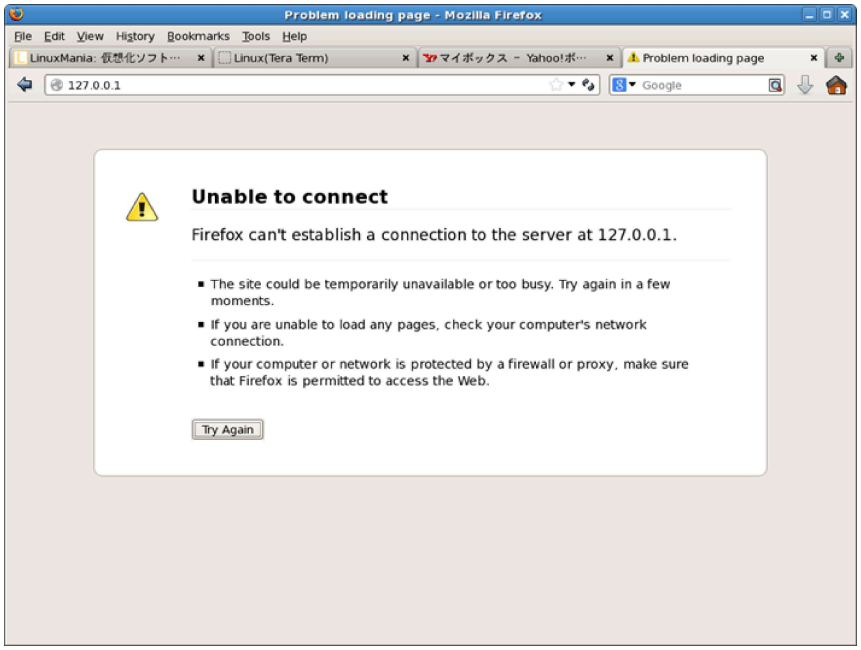
再度、自ユーザでstudentへログインし、その際にパスフレーズ（passphrase）が問われれば成功です。

```
[enomoto@h011 ~]$ ssh student@localhost
Enter passphrase for key '/home/enomoto/.ssh/id_dsa':
Last login: Sat Apr  5 12:38:13 2014 from h011.s132.la.net
```

（4）Webサーバの構築手順

WebサーバはHTML文書や画像などの情報を蓄積し、FireFoxやInternetExplolerなどのWebブラウザからの要求に応じて、インターネットを通じてこれらの情報を発信します。ここではWebサーバの構築の手順を示す。

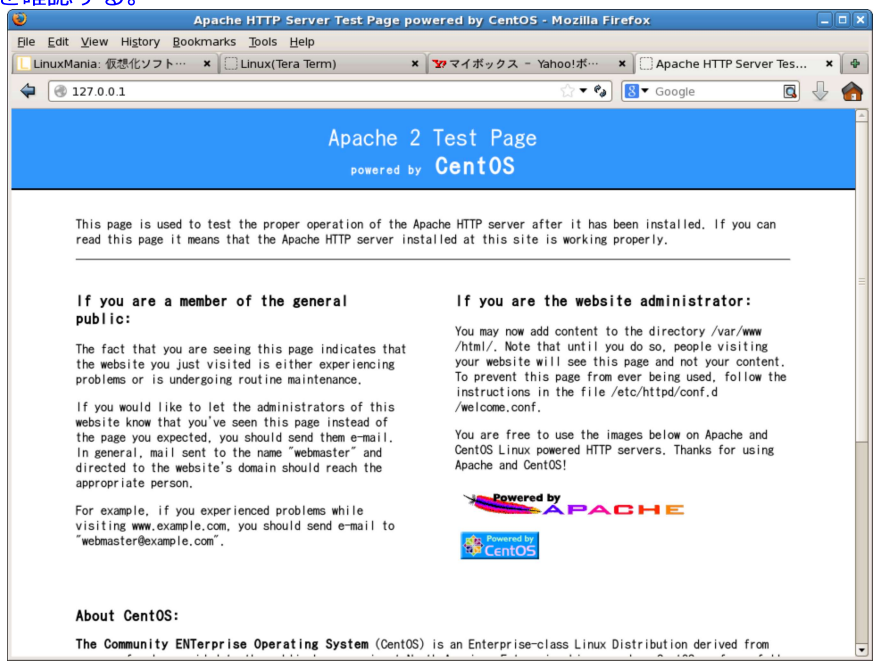
Webサーバのアプリケーションである、Apacheの起動スクリプトは/etc/init.d/httpdである。
まず、起動しない状態で、firefoxを立ち上げ、URL欄に「www.t132011.la.net」を入力してみると、「Unable to connect」と表示されることを確認。



下記のコマンドを投入し、Apacheを起動させる。（なお、root権限のみ実行可能）

```
[root@localhost ~]# /etc/init.d/httpd start
httpd を起動中:                                [  OK  ]
```

この状態で、再度URL欄に「www.t132011.la.net」を入力すると、テストページが起動され、Apacheが正常に稼働していることを確認する。



次回以降、httpdが自動起動されるよう、httpdを立ち上げておく。

```
[root@localhost ~]# chkconfig --list httpd
httpd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@localhost ~]# chkconfig --level 345 httpd on
[root@localhost ~]# chkconfig --list httpd
httpd           0:off   1:off   2:off   3:on    4:on    5:on    6:off
```

プロセスがrootが1個、apacheが8個立ち上がっていることを確認。

```
[root@localhost ~]# ps aux | grep httpd
root     31396  0.0  2.0  22716  9536 ?        Ss   15:28   0:00 /usr/sbin/httpd
apache   31399  0.0  1.1  22848  5624 ?        S    15:28   0:00 /usr/sbin/httpd
apache   31400  0.0  1.1  22848  5576 ?        S    15:28   0:00 /usr/sbin/httpd
apache   31401  0.0  1.1  22848  5576 ?        S    15:28   0:00 /usr/sbin/httpd
apache   31402  0.0  1.1  22848  5576 ?        S    15:28   0:00 /usr/sbin/httpd
apache   31403  0.0  1.0  22848  4928 ?        S    15:28   0:00 /usr/sbin/httpd
apache   31404  0.0  1.0  22848  4928 ?        S    15:28   0:00 /usr/sbin/httpd
apache   31405  0.0  1.0  22848  4928 ?        S    15:28   0:00 /usr/sbin/httpd
apache   31406  0.0  1.0  22848  4928 ?        S    15:28   0:00 /usr/sbin/httpd
root     31972  0.0  0.1   5120   800 pts/2    S+   15:36   0:00 grep httpd
```

index.html（HTMLファイル）を作成する。※必ずファイル名はindex.htmlとしてください。

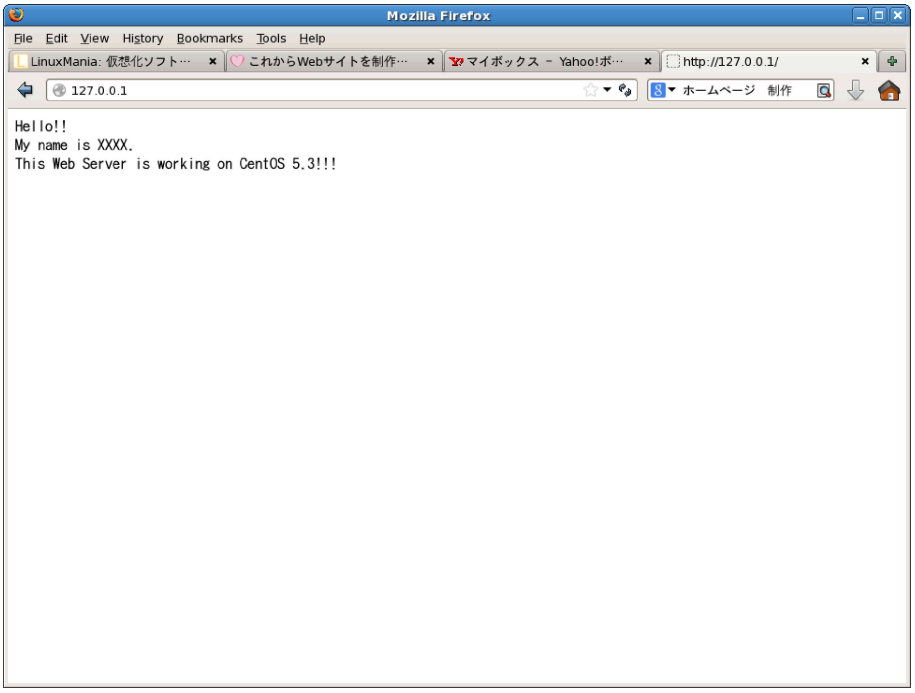
```
[root@localhost ~]# cd /var/www/html/
[root@localhost html]# vi index.html
```

viを開いたら、下記のとおり、Webページを作成する。

```
<html>
<body>
Hello!!<br>
My name is XXXX.<br>
This Web Server is working on CentOS 5.3!!!<br>
</body>
</html>
```

※なお、<html>や
といったものはWebページを制作する際に利用する「HTMLタグ」と呼ばれるものである。Webページを作成する際には最低でも
<html>
～
</html>
が必要である。
HTMLは主にWebページの設計部を制作し、それに対してCSSと呼ばれるものもあり、これはWebページの装飾部を制作する。
HTMLタグの詳細についてはホームページ制作の書籍を参照されたい。

再度、firefoxへアクセスして、URL欄にwww.t132011.la.netと入力し、表示されるページが変更されていることを確認する。



ローカルユーザの公開するWebページ設定

/var/www/html上にindex.htmlを作成したが、このディレクトリは一般ユーザは書込み権限が与えられていない。そのため、このままでは一般ユーザはHTMLファイルを/var/www/html上に置くことができない。

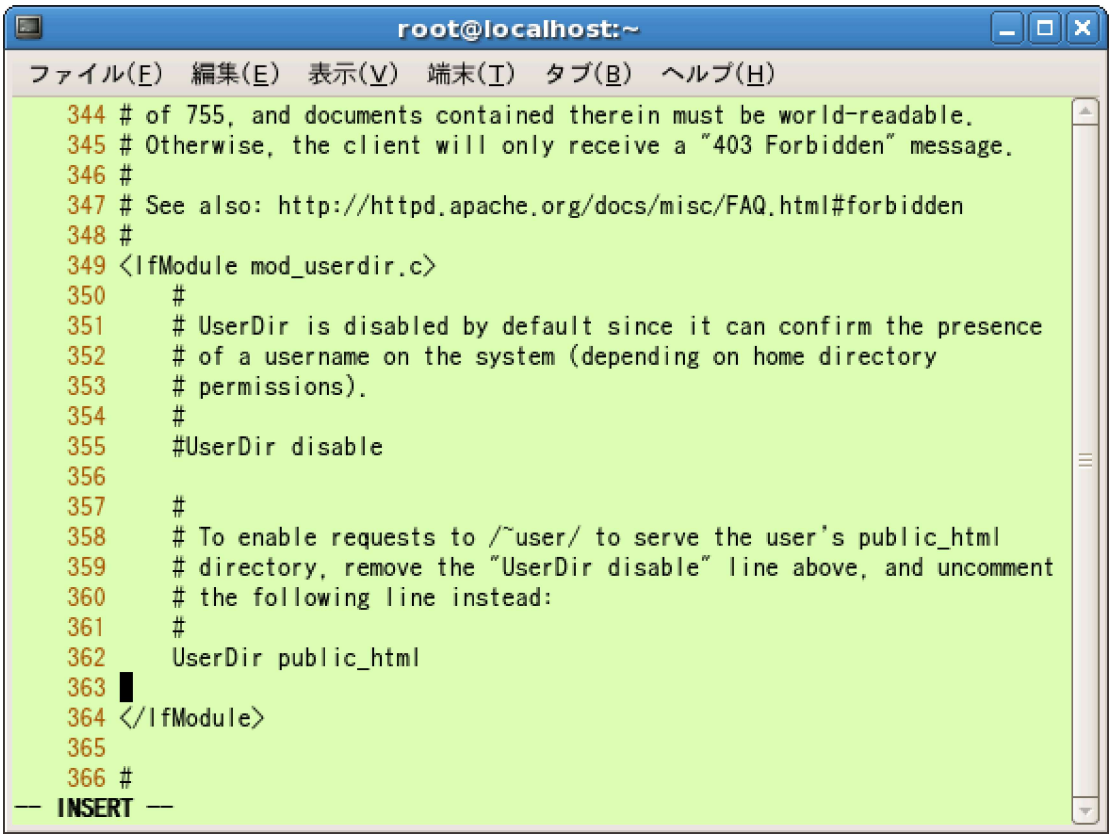
そこで、一般ユーザでもWebページが公開できるよう、ホームユーザディレクトリというものを利用する。

まず、root権限で/etc/http/conf/httpd.confを設定する。
これは、ローカルユーザでもWebページを制作できるように設定するものである。

viで/etc/http/conf/httpd.confを開き、355行目へ移動する。

```
[root@localhost ~]# vi /etc/httpd/conf/httpd.conf
```

次に下記の図のように、
（1）UserDir disableに対して#でコメントアウトする。（355行目）
（2）UserDir public_htmlに対して#を外して有効にする。（362行目）



設定後、設定内容を反映のため、httpdを再起動する。

```
[root@localhost ~]# /etc/init.d/httpd restart
httpd を停止中:                                [ OK ]
httpd を起動中:                                [ OK ]
```

ローカルユーザでログインし、ホームディレクトリ上にpublic_htmlディレクトリを作成する。さらに
public_htmlに移動する。

```
[student@localhost ~]$ mkdir public_html
[student@localhost ~]$ ls
Desktop         id_dsa.pub   web_Manual.odt
Xen_install.odt  public_html  構築演習_install.odt
```

```
[student@localhost ~]$ cd public_html/
```

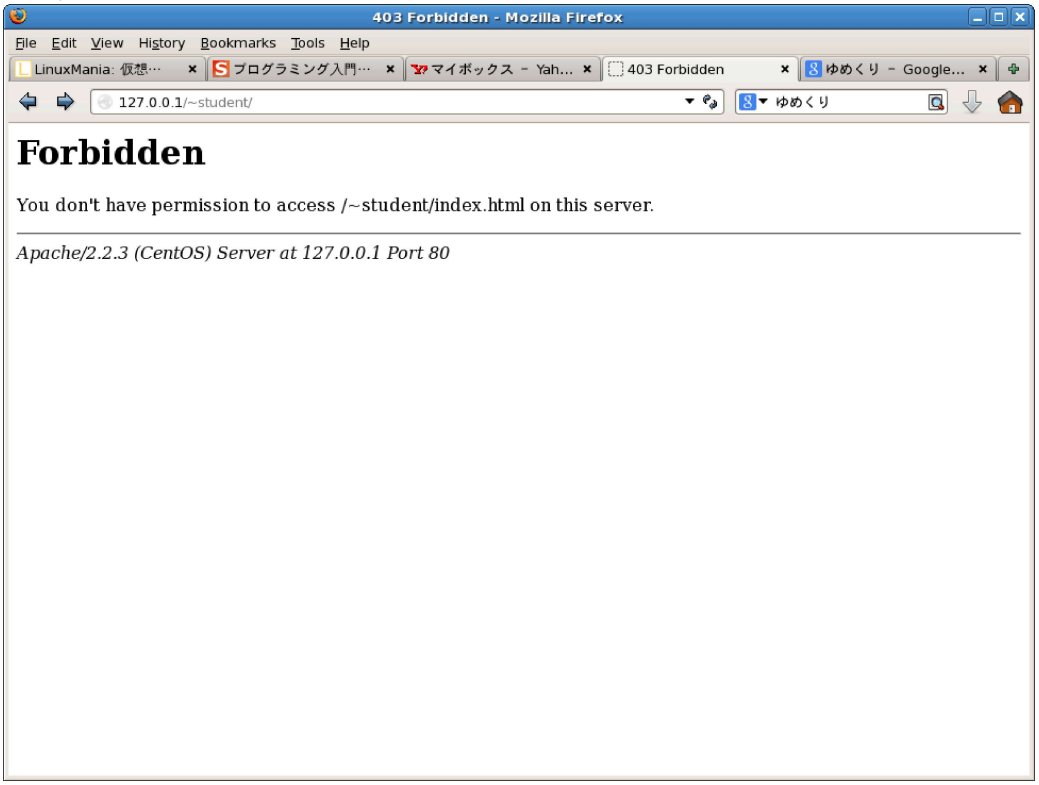
index.html（HTMLファイル）を作成する。※必ずファイル名はindex.htmlとしてください。

```
[student@localhost public_html]$ vi index.html
```

viを開いたら、下記のとおり、Webページを作成する。

```
<html>
<body bgcolor="#ccccff">
This page is local user's page!<br>
This file is /home/student/public_html/index.html<br>
</body>
</html>
```

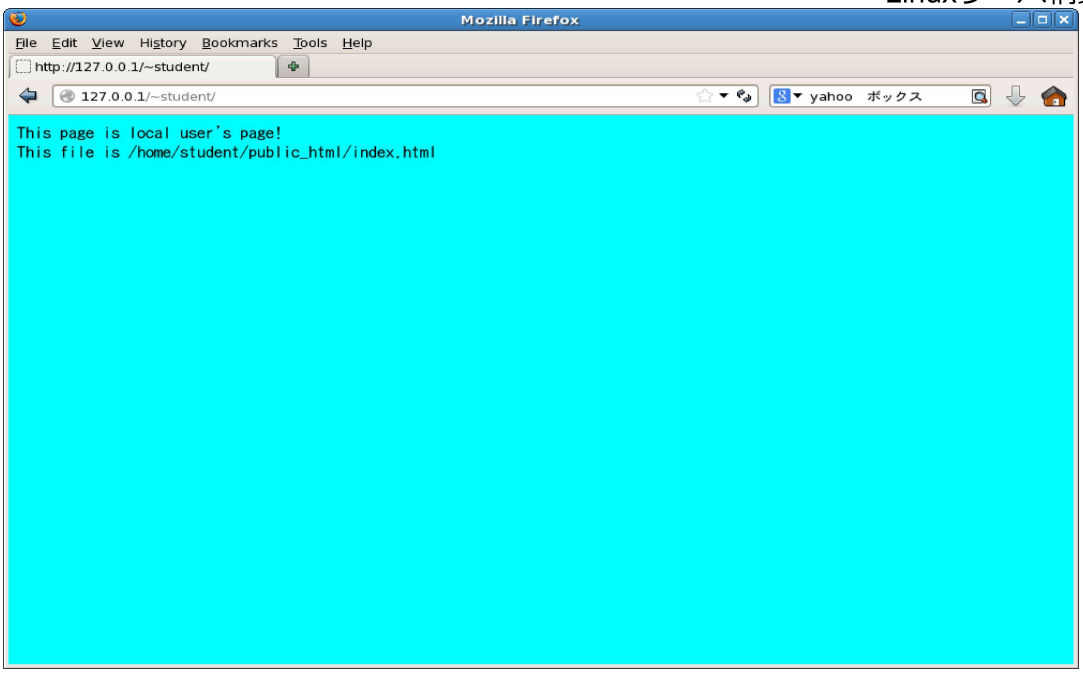
firefoxへアクセスして、URL欄にwww.t132011.la.net/~studentと入力し、表示されるページがforbiddenと表示されていることを確認する。



これは、/home/studentディレクトリ自体が第三者に対して実行権を許可していないため、このような表示となるので、/home/studentのパーミッションを701に変更する。

```
[student@localhost ~]$ chmod 701 /home/student
```

すると、先ほどのfirefoxでWebページに更新をかけると、下記のようにWebページが一般ユーザでも表示できる。



基本認証
ここでは、ユーザ名とパスワードを求めるWebページを表示する設定を行う。

まず、/var/www/html直下にstaffというサブディレクトリを作成する。

```
[root@h011 ~]# cd /var/www/html/
[root@h011 html]# ls
index.html
[root@h011 html]# mkdir staff
[root@h011 html]# ls
index.html  staff
```

作成したstaffファイルに、staffというユーザ名で認証をかける際には以下の設定を行う。

viで/etc/httpd/conf/httpd.confの設定ファイルを開く。

```
[root@h011 ~]# vi /etc/httpd/conf/httpd.conf
```

以下のディレクティブを記載する。

```
<Directory "/var/www/html/staff">
    AuthType Basic
    AuthName "STAFF Only!!"
    AuthUserFile /var/www/html/staff/.htpasswd
    Require user staff
</Directory>
```

設定したhttpd.confファイルを反映する。

```
[root@h011 html]# /etc/init.d/httpd restart
httpd を停止中:                                [  OK  ]
httpd を起動中:                                [  OK  ]
```

次にパスワードファイルを作成する。（ここではパスワードをstaffとしています）

```
[root@h011 html]# htpasswd -c /var/www/html/.htpasswd staff
New password:
Re-type new password:
Adding password for user staff
```

-cオプションを付けると、パスワードファイルが新たに作成され、指定したユーザ（staff）のユーザ名とパスワードが追加される。

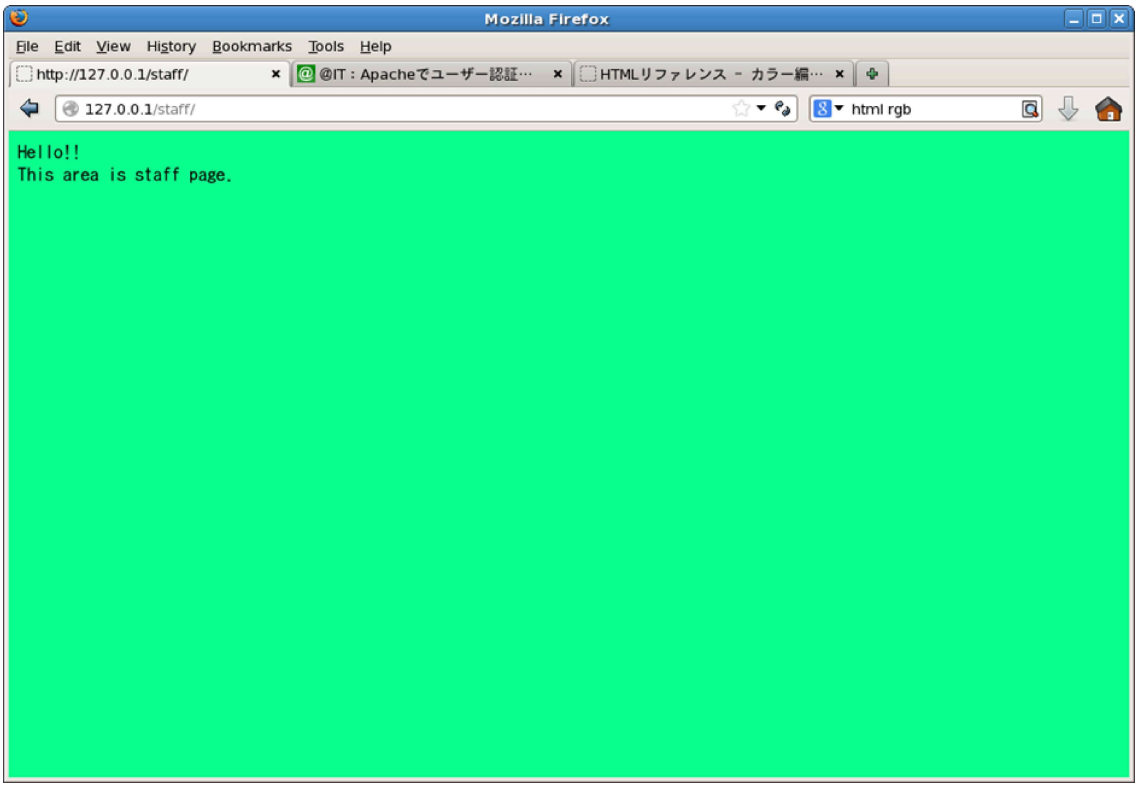
/var/www/html/staff直下に、index.html（HTMLファイル）を作成する。※必ずファイル名はindex.htmlとしてください。

```
[root@localhost ~]# cd /var/www/html/staff
[root@localhost html]# vi index.html
```

viを開いたら、下記のとおり、Webページを作成する。

```
<html>
<body bgcolor=”CCFF99”>
Hello!!<br>
This area is staff page.<br>
</body>
</html>
```

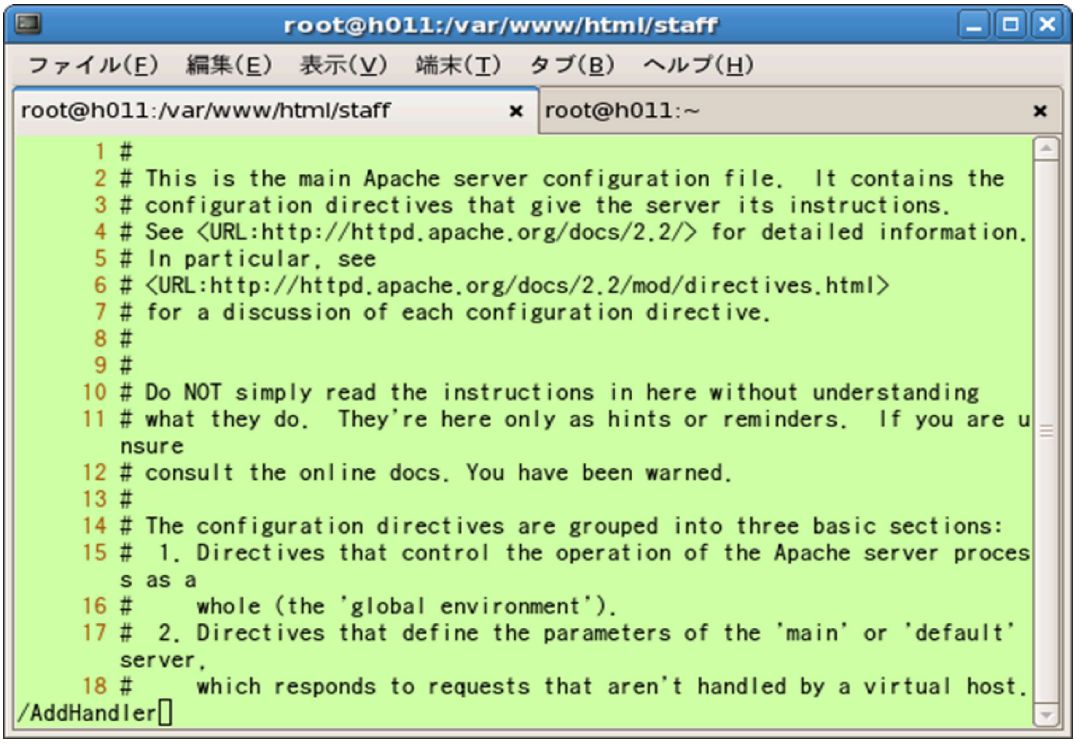
再度、firefoxへアクセスして、URL欄にwww.t132011.la.net/staffと入力すると、ユーザ名とパスワードを聞かれ、それぞれにstaffと入力するとアクセスできる。



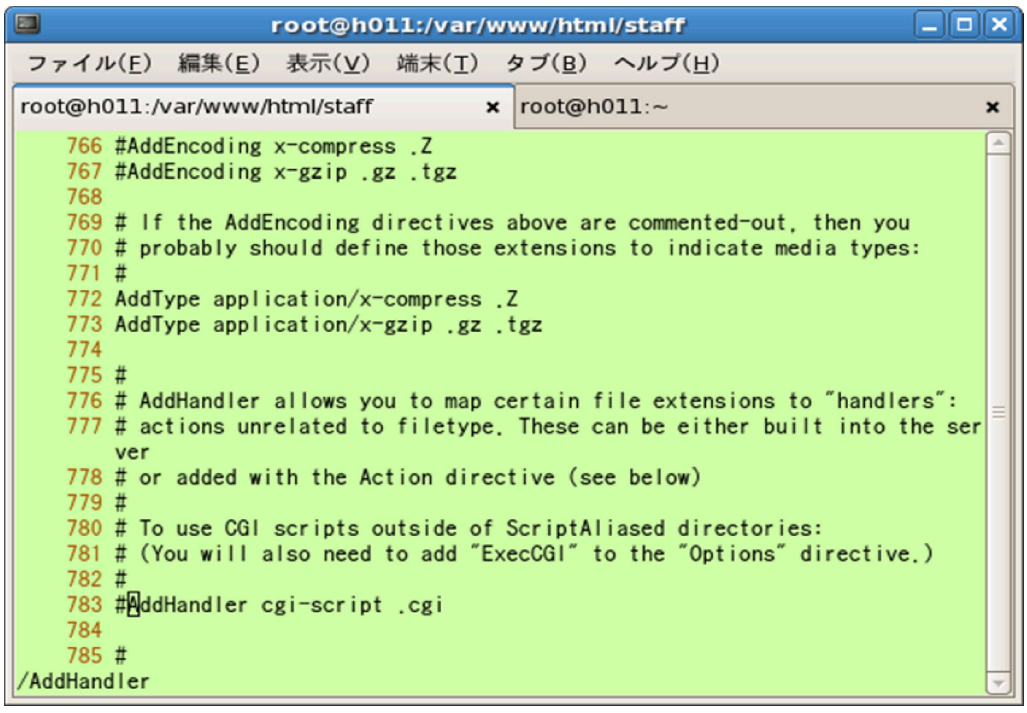
CGIスクリプトの配置

いままでに作成したWebページは静的であった。これから作成するWebページは動的なページを実現するもの

であり、CGIを用いて作成していく。今回はScriptAlias方法でCGIを用いる。
viにて/etc/httpd/conf/httpd.confファイルを開く。その際に、コマンドモードで「/AddHandler」と記載すると、「AddHandler」という文字を検索できる。さらにキーボード「N」で次の検索。



「N」を何回か押下して、下記の行（この例では783行目）の#を削除する。



これにより、cgi-binというディレクトリ内へのファイル要求があると、CGIが実行させる。

次に、var/www/cgi-bin ディレクトリへ移動。

```
[root@h011 staff]# cd /var/www/cgi-bin
```

次にその直下でsample.cgiを作成する。

```
[root@h011 cgi-bin]# vi sample.cgi
```

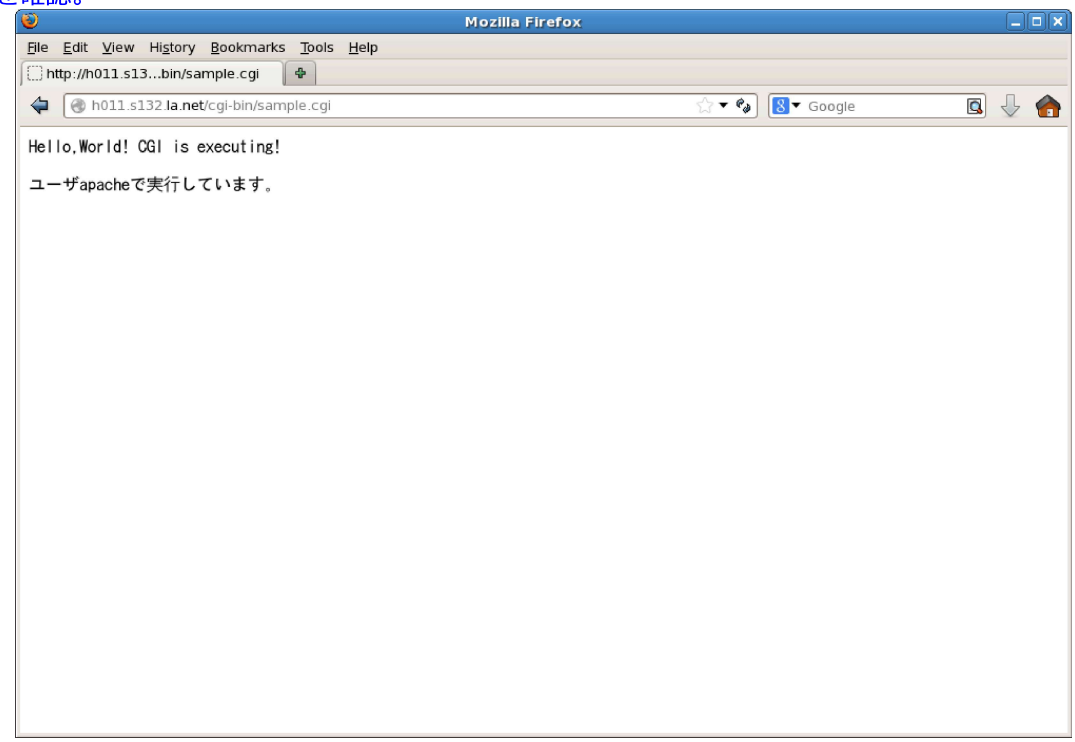
viを開いたら、下記のとおり、Webページを作成する。

```
!/bin/bash
echo "Content-Type: text/html"
echo ""
echo "<html><body><p>Hello,World! CGI is executing!</p>"
exec_user=`whoami`
echo "<p>ユーザ$exec_userで実行しています。</p></body></html>"
```

次にsample.cgiのパーミッション775に変更する。

```
[root@h011 cgi-bin]# chmod 775 sample.cgi
[root@h011 cgi-bin]# ls -l
合計 4
-rwxrwxr-x 1 root root 202  4月  5 14:13 sample.cgi
```

再度、firefoxへアクセスして、URL欄にwww.t132011.la.net/cgi-bin/sample.cgiと入力すると、CGIが実行されていることを確認。



（5）メールサーバ構築

メールサーバ構築にはSMTP（転送）としてpostfixが、POP（受信）としてdovecotが利用されている。それらの設定を行い、メール送受信を行えるようにする。

SMTPサーバの構築

まず、postfixがインストールされているかをrpmコマンドで確認。

```
[root@h011 ~]# rpm -qi postfix
パッケージ postfix はインストールされていません。
```

今回は利用しないが、メールサーバアプリで有名なsendmailはインストールされている。

```
[root@h011 ~]# rpm -qi sendmail
Name        : sendmail                     Relocations: (not relocatable)
Version     : 8.13.8                         Vendor: CentOS
Release     : 8.1.el5_7                    Build Date: 2011年08月12日 02時32分29秒
Install Date: 2014年04月05日 11時01分44秒      Build Host: builder10.centos.org
            ※※※省略※※※
Sendmail を再設定する必要がある場合は sendmail-cf パッケージも
インストールする必要があります。Sendmail に関するドキュメントが
必要な場合は sendmail-doc パッケージをインストールしてください。
```

※sendmailはセキュリティホールがときどき見つかる。そのため、現在ではあまり利用されないアプリケーションである。

まず、postfixをインストールする。なお、-yオプションはインストール中の質問に全てyesで答えるという意味である。

```
[root@h011 ~]# yum -y install postfix
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
            ※※※省略※※※
Complete!
```

postfixがインストールされているかをrpmコマンドで確認。

```
[root@h011 ~]# rpm -qi postfix
Name        : postfix                      Relocations: (not relocatable)
Version     : 2.3.3                          Vendor: CentOS
Release     : 6.el5                        Build Date: 2013年01月09日 13時42分22秒
Install Date: 2014年04月05日 17時05分29秒      Build Host: builder10.centos.org
Group       : System Environment/Daemons    Source RPM: postfix-2.3.3-6.el5.src.rpm
Size        : 9249122                        License: IBM Public License
Signature   : DSA/SHA1, 2013年01月10日 03時55分48秒, Key ID a8a447dce8562897
URL         : http://www.postfix.org
Summary     : メール転送エージェント
Description :
Postfix is a Mail Transport Agent (MTA), supporting LDAP, SMTP AUTH (SASL),
TLS
```

次に、デフォルトでインストールされている、sendmailとpostfixの入れ替えを行うためのツール（system-switch-mail）をインストールする。

```
[root@h011 ~]# yum -y install system-switch-mail
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
            ※※※省略※※※
Complete!
```

sendmailとPostfixを入れ替えるために下記コマンドを実行。

```
[root@h011 ~]# system-switch-mail
```

すると、以下の状態になる。Postfixを選択する。

OKを選ぶと下記のように反映される。

さらにOKを押すと終了する。これによりpostfixが利用できる。

次に、/etc/postfix/main.cfという、設定ファイルをlessやviコマンドで全体像を掴むようにします。

```
[root@h011 ~]# vi /etc/postfix/main.cf
```

次に、main.cfで設定されているステータスをpostconf -nで確認します。

```
[root@h011 ~]# postconf -n
alias_database = hash:/etc/aliases
alias_maps = hash:/etc/aliases
```

```
command_directory = /usr/sbin
config_directory = /etc/postfix
daemon_directory = /usr/libexec/postfix
debug_peer_level = 2
html_directory = no
inet_interfaces = localhost
mail_owner = postfix
mailq_path = /usr/bin/mailq.postfix
manpage_directory = /usr/share/man
mydestination = $myhostname, localhost.$mydomain, localhost
newaliases_path = /usr/bin/newaliases.postfix
queue_directory = /var/spool/postfix
readme_directory = /usr/share/doc/postfix-2.3.3/README_FILES
sample_directory = /usr/share/doc/postfix-2.3.3/samples
sendmail_path = /usr/sbin/sendmail.postfix
setgid_group = postdrop
unknown_local_recipient_reject_code = 550
```

次に、設定ファイルにて、以下の項目を設定します。（#が付いているものは削除、なお、先に設定したDNSサーバが正しく設定されているものとします。）

myhostname = mail.t132011.la.net
（@以下のホスト名を設定する）

mydomain = t132011.la.net
（@以下のネットワーク名を設定する）

myorigin = $mydomain
（例えば送り先をstudentだけとしても、自動でstudent@h011.s132.la.netとしてくれる。）

inet_interfaces = all
（inet_interfacesをallにすることで、すべてのI/Fから受付可能。通常はIPアドレスを指定。）

mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
（自分宛にきたメールかをmydestinationのホスト部を見てローカル配送するかどうかを設定する。）

mynetworks = 10.20.132.11,127.0.0.1,10.20.0.0/16
（ 中継が必要なメール送信要求を受け付けるホスト（もしくはNW）を指定する。不正リレーを防ぐ。）

なお、$myhostnameなど、$の付いている変数は、myhostnameで指定した値がそこに入るという意味である。

変更箇所をpostconf -nコマンドで確認する。

```
[root@h011 ~]# postconf -n
alias_database = hash:/etc/aliases
alias_maps = hash:/etc/aliases
command_directory = /usr/sbin
config_directory = /etc/postfix
daemon_directory = /usr/libexec/postfix
debug_peer_level = 2
html_directory = no
inet_interfaces = all
mail_owner = postfix
mailq_path = /usr/bin/mailq.postfix
manpage_directory = /usr/share/man
mydestination = $myhostname
mydomain = t132011.la.net
myhostname = smtp.s132.la.net
mynetworks = 10.20.132.11, 127.0.0.1
```

```
myorigin = $myhostname,$mydomain
newaliases_path = /usr/bin/newaliases.postfix
queue_directory = /var/spool/postfix
readme_directory = /usr/share/doc/postfix-2.3.3/README_FILES
sample_directory = /usr/share/doc/postfix-2.3.3/samples
sendmail_path = /usr/sbin/sendmail.postfix
setgid_group = postdrop
unknown_local_recipient_reject_code = 550
```

設定したら、/etc/init.d/postfix startでpostfixを起動させる。

```
[root@h011 ~]# /etc/init.d/postfix start
postfix を起動中:                                          [  OK  ]
```

次に、enomoto宛にメールを送る。
下記コマンド投入後、Subjectに何か件名を入れ、その後に、メール本文（下記ではtest）を入れる。本文を入れ、最後１行に．（ドット）を入れると、Cc:と表示されるので、そこには何も入れず、[Enter]を押すと送信できる。

```
[root@h011 ~]# mail enomoto@t132011.la.net
Subject: This is a test mail(No.1)
これはテストメールです。
.
Cc:
```

最後にメールが送られたかを確認する。各ユーザのメールは/var/mail/studentに格納されている。

```
[root@h011 ~]# cat /var/mail/enomoto
From root@t132011.la.net  Sat Apr 12 14:57:58 2014
Return-Path: <root@t132011.la.net>
X-Original-To: enomoto@t132011.la.net
Delivered-To: enomoto@t132011.la.net
Received: by mail.t132011.la.net (Postfix, from userid 0)
        id 80FF22EC1C; Sat, 12 Apr 2014 14:57:58 +0900 (JST)
To: enomoto@t132011.la.net
Subject: This is a test mail(No.1)
Message-Id: <20140412055758.80FF22EC1C@mail.t132011.la.net>
Date: Sat, 12 Apr 2014 14:57:58 +0900 (JST)
From: root@t132011.la.net (root)

これはテストメールです。
```

※練習として、何通かメール送付してみましょう。なお、インターネットがつながる環境であれば外部へも送信が可能ですが、誤送信などセキュリティ事故には気を付けましょう。
（なお、外部からの受信はPOPを設定していないため、まだできません）

転送処理の設定

ここでは転送処理について設定を行うが、転送処理には、システム全体としての転送処理と、ユーザごとで行う転送処理の2つがある。
（１）システム全体としての転送処理
viを用いて、/etc/aliasesを開き「# Basic system aliases -- these MUST be present. 」の項目を参照する。

```
[root@h011 ~]# /etc/aliases
```

```
※※※省略※※※
# Basic system aliases -- these MUST be present.
mailer-daemon:  postmaster
postmaster:     root

# General redirections for pseudo accounts.
bin:            root
※※※省略※※※
```

※この中で、mailer-daemon: postmasterとpostmaster: rootとあるが、この名前でのユーザは存在しない。（実際に/etc/passwdを調べてみるとそのユーザがないことがわかる）

これらは、メール送受信時に、メールサーバ内で動くプログラムのログ通知やエラー通知などのログを最終的にrootユーザのみが受け取れるようになる設定となっている。
これを、studentユーザが受信できるように設定するには以下のように設定する。

```
[root@h011 ~]# /etc/aliases
※※※省略※※※
# Basic system aliases -- these MUST be present.
mailer-daemon:  postmaster
postmaster:     student

# General redirections for pseudo accounts.
bin:            root
※※※省略※※※
```

この状態で、newaliasesコマンドを投入する。

```
[root@h011 ~]# newaliases
```

※postfix自体、設定した/etc/aliasesは参照しないため、設定しただけでは結果が反映されない。バイナリ化されている/etc/aliases.dbを参照するので、/etc/aliasesを設定したら、必ずnewaliasesコマンドで/etc/aliases.dbに反映させる。

この状態で、newaliasesコマンドを投入する。

```
[root@h011 ~]# mail root@t132011.la.net
Subject: 管理者権限->student
これは管理者rootからstudentへ自動転送されています。
.
Cc:
```

すると、メールがstudent宛にも送られていることがわかる。

```
[root@h011 ~]# cat /var/mail/student
From root@t132011.la.net  Sat Apr 12 11:23:27 2014
Return-Path: <root@t132011.la.net>
X-Original-To: root@t132011.la.net
Delivered-To: root@t132011.la.net
Received: by mail.t132011.la.net (Postfix, from userid 0)
        id BF4FA2EC1C; Sat, 12 Apr 2014 11:23:27 +0900 (JST)
To: root@t132011.la.net
Subject: 管理者権限->student
Message-Id: <20140412022327.BF4FA2EC1C@mail.t132011.la.net>
Date: Sat, 12 Apr 2014 11:23:27 +0900 (JST)
From: root@t132011.la.net (root)
X-UID: 5
Status: RO

これは管理者rootからstudentへ自動転送されています。
```

POPサーバ構築

メール受信のサーバ構築についての手順を記載する。POPサーバではdovecotというアプリケーションを使用する。
まず、dovecotがインストールされているかを確認する。されていなければ、yum -y install dovecotコマンドでインストールする。（下記の状態はインストールされている。）

```
[root@h011 ~]# rpm -qi dovecot
Name        : dovecot                      Relocations: (not relocatable)
Version     : 1.0.7                          Vendor: CentOS
Release     : 8.el5_9.1                      Build Date: 2013年06月26日 00時01分54秒
Install Date: 2014年04月05日 11時01分57秒      Build Host: builder17.centos.org
Group       : System Environment/Daemons    Source RPM: dovecot-1.0.7-8.el5_9.1.src.rpm
Size        : 3735717                        License: LGPL
Signature   : DSA/SHA1, 2013年06月26日 11時24分21秒, Key ID a8a447dce8562897
URL         : http://www.dovecot.org/
Summary     : Dovecot Secure imap サーバー
Description :
Dovecot は Linux/UNIX 系システム用の IMAP サーバで、セキュリティを重視して
書かれています。また、小規模 POP3 サーバも含まれています。maildir または
mbox 形式のいずれかでメールをサポートします。
```

次に、vi で/etc/dovecotの設定ファイルを開き、20行目にあった#を外し、imapとimapsを削除する。

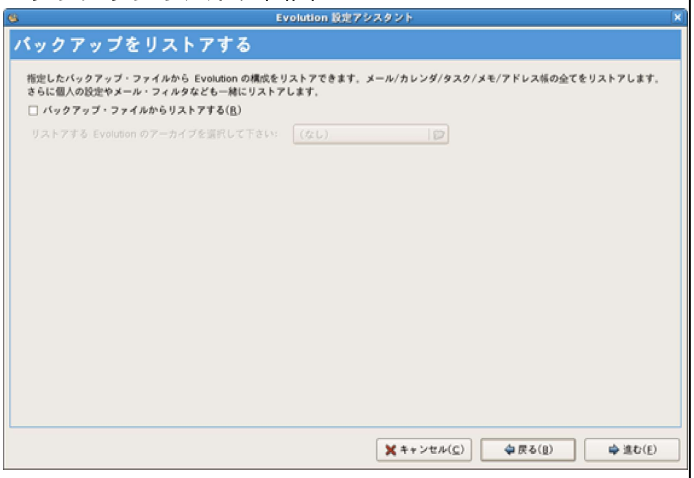
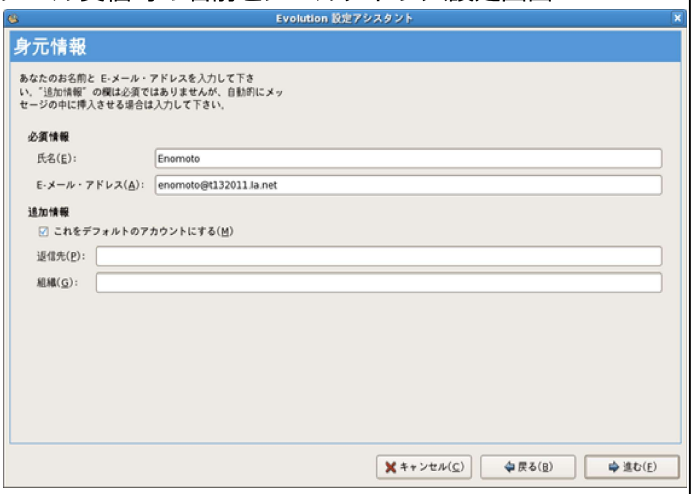
```
[root@h011 ~]# vi /etc/dovecot.conf
            ※※※省略※※※
    18 # Protocols we want to be serving: imap imaps pop3 pop3s
    19 # If you only want to use dovecot-auth, you can set this to "none".
    20 #protocols = imap imaps pop3 pop3s
    21 protocols = pop3 pop3s
            ※※※省略※※※
```

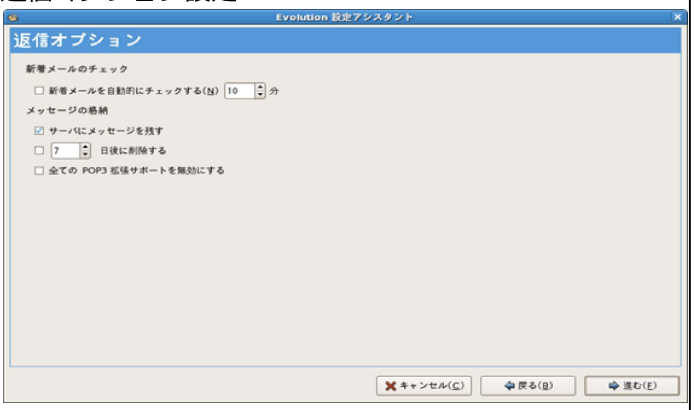
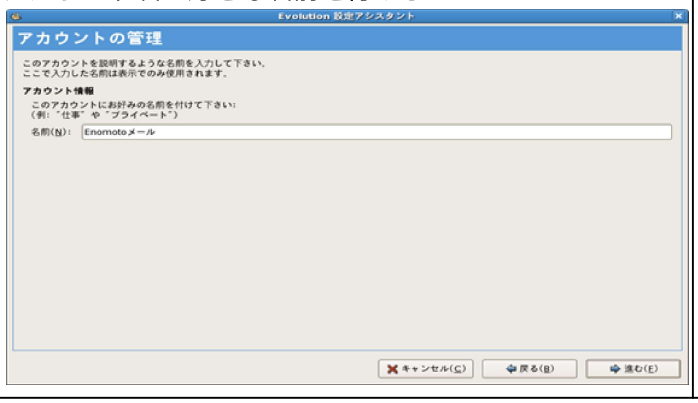
設定したら、/etc/init.d/dovecot startでdovecotを起動させる。

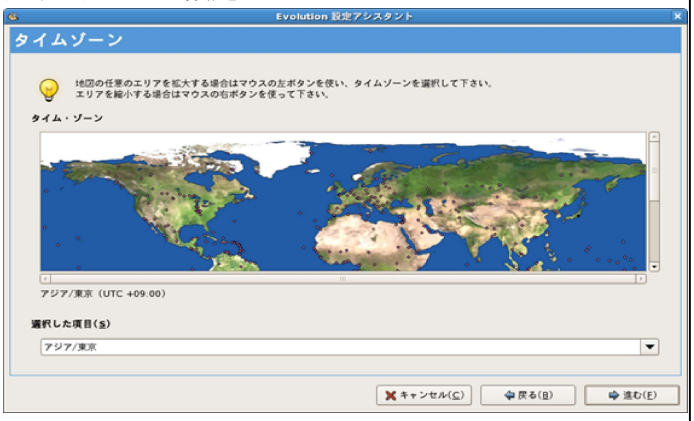
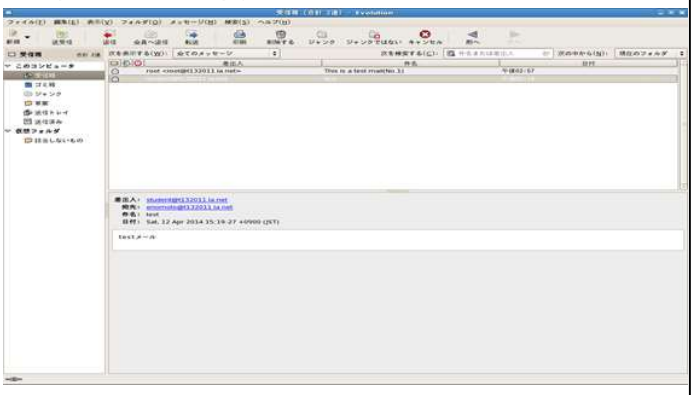
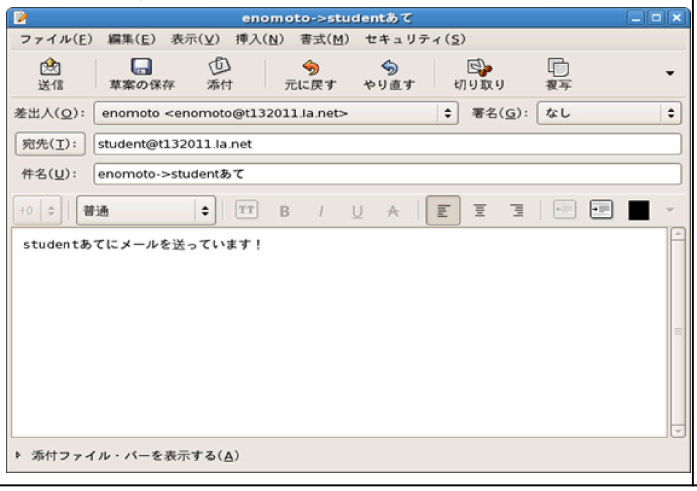
```
[root@h011 ~]# /etc/init.d/dovecot start
Dovecot Imap を起動中:
```

次に、デスクトップ上記メニューより、[アプリケーション]->[インターネット]->[電子メール]の順で、Evolutionを立ち上げ、以下の手順にしたがいメーラの設定を行う。

番号	設定する項目	設定訂順

1	メーラ設定案内 	ここでは「進む」を押す。
2	バックアップリストア画面 	ここでも何も設定せず「進む」を押す。
3	メール受信時の名前とメールアドレス設定画面 	・「氏名」欄には任意の名前（ここではEnomotoとする） ・メールアドレス欄には enomoto@t132011.la.net を入力。 なお、enomotoはあらかじめuseraddコマンドで作成したユーザで、作成されていなければ、 # useradd enomoto のコマンドで作成してください。 作成したら「進む」を押す。
4	受信サーバの設定	「サーバ」欄に h011.s132.la.net 「ユーザ名」欄に enomoto

		を入力し、「進む」を選択。
5	返信オプション設定 	「サーバにメッセージを残す」にチェックを入れる。 チェックしたら「進む」を選択。
6	SMTPサーバの設定 	「サーバ」欄に smtp.t132011.la.net を入力。 チェックしたら「進む」を選択。
7	アカウント名に好きな名前を付ける 	メール受信時など、アカウント名に好きな名前を付けることができる。 ここでは「Enomotoメール」としている。 設定したら「進む」を選択。

8	タイムゾーンの設定 	選択した項目が「アジア/東京」であることを確認し「進む」を選択。
9	メーラが立ち上がる。 	この状態で、左上の「送受信」ボタンを押す。 ※パスワードを求められたら、入力してください。
10	メールを送付 	enomotoユーザからstudentユーザあてにメールを送る。 Studentユーザのメールアドレスは student@t132011.la.net とする。 作成したら、送信する。

送信したら、cat コマンドで、/var/mail/studentを開き、受信を確認。

```
[root@h011 ~]# cat /var/mail/student
        ※※※省略※※※
From enomoto@t132011.la.net  Sat Apr 12 15:48:24 2014
Return-Path: <enomoto@t132011.la.net>
X-Original-To: student@t132011.la.net
Delivered-To: student@t132011.la.net
Received: from [10.20.132.11] (unknown [10.20.132.11])
        by mail.t132011.la.net (Postfix) with ESMTP id F04142EC1C
        for <student@t132011.la.net>; Sat, 12 Apr 2014 15:48:23 +0900 (JST)
Subject: =?UTF-8?Q?enomoto-=3Estudent=E3=81=82=E3=81=A6?=
From: enomoto <enomoto@t132011.la.net>
To: student@t132011.la.net
Content-Type: text/plain; charset=UTF-8
Date: Sat, 12 Apr 2014 15:48:23 +0900
Message-Id: <1397285303.17028.1.camel@h011.s132.la.net>
Mime-Version: 1.0
X-Mailer: Evolution 2.12.3 (2.12.3-19.el5)
Content-Transfer-Encoding: 8bit

studentあてにメールを送っています！
```

（6）パケットフィルタリング

ネットワーク上ではパケットという単位でデータが流れており、そのパケットには送信先、送信元、ポート番号やプロトコルの情報が付いている。

今回はそのパケットの情報の条件を満たしているかどうかにより、パケットを破棄するかどうかの設定を行う。
例えば、今まで設定してきた、Webサーバやメールサーバを利用するにはウェルノウンポートの番号があり、その番号をもとに、破棄するかしないかを設定できる。
使用しないポートは必ず閉じておくように設定しておくべきである。これは、ポートスキャンという、悪意を持った第三者がポートが開いているかを確認し、それにより、アクセス可能かどうかを調べ、攻撃対象になるため、必ずセキュリティ対策が必要である。

まず、現在設定されている、ポリシと個々のルールのリストを表示させる。その方法には、iptables -Lと入力する。

```
[root@h011 ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

上記の結果より、3つの経路があることがわかる。
それぞれ、INPUT、FORWARD、OUTPUTの3つがあり、チェインと呼ばれる。それぞれのチェインに対してパケットフィルタリングを行っていく。

一般的に、3つの経路に対して、
・INPUTは基本的に全て拒否（必要のあるポートのみ開放）
・FORWARDは基本的に全て拒否（必要のあるポートのみ開放）
・OUTPUTは基本的に全て許可

その他、以下の仕様（1）～（3）を満たす、パケットフィルタリングの設定を行う。

（1）INPUTチェインの設定
　　・フィルタフィングポリシは「DROP」

・サービスに関するパケットのルール（以下の表参照）

プロトコル	宛先アドレス	送信元ポート番号	送信ポートアドレス
Any	ローカルホスト	-	-
ICMP	自分のホスト	-	-
TCP	自分のホスト	Any	80
TCP	自分のホスト	Any	53
UDP	自分のホスト	Any	53
TCP	自分のホスト	Any	25
TCP	自分のホスト	Any	110
TCP	自分のホスト	Any	22
TCP	自分のホスト	80	Any
TCP	自分のホスト	53	Any
UDP	自分のホスト	53	Any
TCP	自分のホスト	25	Any
TCP	自分のホスト	110	Any
TCP	自分のホスト	22	Any

（2）FORWARDチェインの設定
・フィルタフィングポリシは「DROP」

（3）OUTPUTチェインの設定
・フィルタフィングポリシは「ACCEPT」
（基本的にはデフォルトで設定されている）

（1）INPUTチェインの設定
まず、INPUTチェインについて、ポリシをDROP（破棄）に設定し、一旦INPUTは全て拒否とする。

```
[root@h011 ~]# iptables -P INPUT DROP
```

INPUTのポリシがDROPであることを確認する。

```
[root@h011 ~]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

このままでは、何も受信できなくなってしまうため、前表の仕様に従い、個々のポートに設定をしていく。

```
[root@h011 ~]# iptables -A INPUT -d 127.0.0.1 -j ACCEPT
[root@h011 ~]# iptables -A INPUT -p icmp -d 10.20.132.11 -j ACCEPT
[root@h011 ~]# iptables -A INPUT -p tcp -d 10.20.132.11 --dport 80 -j ACCEPT
[root@h011 ~]# iptables -A INPUT -p tcp -d 10.20.132.11 --dport 53 -j ACCEPT
[root@h011 ~]# iptables -A INPUT -p udp -d 10.20.132.11 --dport 53 -j ACCEPT
[root@h011 ~]# iptables -A INPUT -p tcp -d 10.20.132.11 --dport 25 -j ACCEPT
[root@h011 ~]# iptables -A INPUT -p tcp -d 10.20.132.11 --dport 110 -j ACCEPT
[root@h011 ~]# iptables -A INPUT -p tcp -d 10.20.132.11 --dport 22 -j ACCEPT
[root@h011 ~]# iptables -A INPUT -p tcp -d 10.20.132.11 --sport 80 -j ACCEPT
```

```
[root@h011 ~]# iptables -A INPUT -p tcp -d 10.20.132.11 --sport 53 -j ACCEPT
[root@h011 ~]# iptables -A INPUT -p udp -d 10.20.132.11 --sport 53 -j ACCEPT
[root@h011 ~]# iptables -A INPUT -p tcp -d 10.20.132.11 --sport 25 -j ACCEPT
[root@h011 ~]# iptables -A INPUT -p tcp -d 10.20.132.11 --sport 110 -j ACCEPT
[root@h011 ~]# iptables -A INPUT -p tcp -d 10.20.132.11 --sport 22 -j ACCEPT
```

これにより、INPUTチェインの必要部分のみのポート開放が設定された。

（2）FORWARDチェインの設定

FORWARDチェインについて、ポリシをDROP（破棄）に設定し、FORWARDチェインは全て拒否とする。

```
[root@h011 ~]# iptables -P FORWARD DROP
```

（3）OUTPUTチェインの設定

OUTPUTチェインについて、ポリシをACCEPTに設定し、OUTPUTチェインは全て許可とする。

```
[root@h011 ~]# iptables -P OUTPUT ACCEPT
```

最後に、（1）～（3）の設定が以下のように反映されているか確認。

```
[root@h011 ~]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             h011.s132.la.net
ACCEPT     icmp --  anywhere             10.20.132.11
ACCEPT     tcp  --  anywhere             10.20.132.11        tcp dpt:http
ACCEPT     tcp  --  anywhere             10.20.132.11        tcp dpt:domain
ACCEPT     udp  --  anywhere             10.20.132.11        udp dpt:domain
ACCEPT     tcp  --  anywhere             10.20.132.11        tcp dpt:smtp
ACCEPT     tcp  --  anywhere             10.20.132.11        tcp dpt:pop3
ACCEPT     tcp  --  anywhere             10.20.132.11        tcp dpt:ssh
ACCEPT     tcp  --  anywhere             10.20.132.11        tcp spt:http
ACCEPT     tcp  --  anywhere             10.20.132.11        tcp spt:domain
ACCEPT     udp  --  anywhere             10.20.132.11        udp spt:domain
ACCEPT     tcp  --  anywhere             10.20.132.11        tcp spt:smtp
ACCEPT     tcp  --  anywhere             10.20.132.11        tcp spt:pop3
ACCEPT     tcp  --  anywhere             10.20.132.11        tcp spt:ssh

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```