

CyberRad 2003 ミニシアター2

チュートリアル(4)

手作りDICOMサーバの構築方法

藤田保健衛生大学 衛生学部

武藤 晃一

kmuto@fujita-hu.ac.jp

チュートリアルの目的

- DICOM画像サーバをユーザ自身で構築
 - いわゆる「参照系」として使えるものにする
 - 安価に構築しましょう
- では、なにが必要か？
 - ハードウェアは？
 - ソフトウェアは？
 - DICOMサーバソフトやその他のソフトはどうするの？
 - インストール方法は？
 - どうやって使う？

Do It Yourself するために必要なこと

- DICOMについては、何となく知っている
- “Linux” と聞いても尻込みしない
- GUIでなくCUIにも挑戦できる
- SQLという言葉聞いたことがある
- プログラミング言語にアレルギーは無いと思う
- インターネット、TCP/IPのことは多少わかる
- わからないことは自分で調べる覚悟がある
 - 重要です・・・

手作りするシステムの概要

- ハードウェア
 - PC-AT互換機
 - 今現在、一般的なスペックのパソコンでOK
 - メモリーは多めに、HDDはかなり多めに？！
- ソフトウェア
 - OS : Linux
 - オープンソースでフリーに入手可能
 - たいていのディストリビューションでOK
 - DICOMサーバ : CTN
 - フリーソフト
 - 「参照系」に使えるように少し改造
 - その他、必要となるソフトウェア
 - すべてオープンソースなフリーソフトウェアを利用する

CTNとは？

- Electronic Radiology Laboratory at the Mallinckrodt Institute of Radiology, Washington University in St. Louis にて DICOMの実装実験と普及を目的に開発され、anonymous-ftpでソースコードが公開されているソフトウェア群

Central Test Node

- CTNに含まれるImage Storage SCPの機能を持つソフト = archive_server

CTNの画像サーバがサポートするサービスクラス

SOP Class UID	SOP Class Name	Role
1.2.840.1008.1.1	Verification	SCP
1.2.840.1008.20.1	Storage Commitment Push Model SOP Class	SCP
1.2.840.1008.3.1.2.5.1	Detached Results Management SOP Class	SCP
1.2.840.1008.3.1.2.5.4	Detached Results Management Meta SOP Class	SCP
1.2.840.1008.3.1.2.6.1	Detached Interpretation Management SOP Class	SCP
1.2.840.1008.5.1.4.1.1.1	Computed Radiography Image Storage	SCU/SCP
1.2.840.1008.5.1.4.1.1.2	CT Image Storage	SCU/SCP
1.2.840.1008.5.1.4.1.1.3	Ultrasound Multi-frame Image Storage (Retired)	SCU/SCP
1.2.840.1008.5.1.4.1.1.4	MR Image Storage	SCU/SCP
1.2.840.1008.5.1.4.1.1.6	Ultrasound Image Storage (Retired)	SCU/SCP
1.2.840.1008.5.1.4.1.1.7	Secondary Capture Image Storage	SCU/SCP
1.2.840.1008.5.1.4.1.1.12.1	X-Ray Angiographic Image Storage	SCU/SCP
1.2.840.1008.5.1.4.1.1.12.2	X-Ray Radiofluoroscopic Image Storage	SCU/SCP
1.2.840.1008.5.1.4.1.1.12.3	X-Ray Angiographic Bi-Plane Image Storage	SCU/SCP
1.2.840.1008.5.1.4.1.1.20	Nuclear Medicine Image Storage	SCU/SCP
1.2.840.1008.5.1.4.1.1.481.1	RT Image Storage	SCU/SCP
1.2.840.1008.5.1.4.1.1.481.2	RT Dose Storage	SCU/SCP
1.2.840.1008.5.1.4.1.1.481.3	RT Structure Set Storage	SCU/SCP
1.2.840.1008.5.1.4.1.1.481.5	RT Plan Storage	SCU/SCP
1.2.840.1008.5.1.4.1.2.1.1	Patient Root Query/Retrieve Information Model - FIND	SCP
1.2.840.1008.5.1.4.1.2.1.2	Patient Root Query/Retrieve Information Model - MOVE	SCP
1.2.840.1008.5.1.4.1.2.1.3	Patient Root Query/Retrieve Information Model - GET	SCP
1.2.840.1008.5.1.4.1.2.2.1	Study Root Query/Retrieve Information Model - FIND	SCP
1.2.840.1008.5.1.4.1.2.2.2	Study Root Query/Retrieve Information Model - MOVE	SCP
1.2.840.1008.5.1.4.1.2.2.3	Study Root Query/Retrieve Information Model - GET	SCP
1.2.840.1008.5.1.4.1.2.3.1	Patient/Study Only Query/Retrieve Information Model - FIND	SCP
1.2.840.1008.5.1.4.1.2.3.2	Patient/Study Only Query/Retrieve Information Model - MOVE	SCP
1.2.840.1008.5.1.4.1.2.3.3	Patient/Study Only Query/Retrieve Information Model - GET	SCP

より具体的な構築方針

- RedHat Linux 上で CTN を make install
 - CTNの archive_server には jpeg 画像出力機能を付ける
 - データベースは PostgreSQL を使用
- 「参照系」システムは WWW を利用
 - Webサーバは apache
 - サーバサイドプログラミングには PHP4
 - クライアントは Windows の Internet Explorer
 - ユーザインターフェースに Javascript, DHTMLを使用

構築手順の概要

- PC-AT互換機をセットアップ
 - 自作してもよし、メーカー物を購入してもよし
- 必要なソフトウェアを入手
 - インターネットからダウンロードする
 - RedHat Linux = <http://www.redhat.com/>
 - PostgreSQL = <http://www.postgresql.org/>
 - apache = <http://www.apache.org/>
 - PHP4 = <http://www.php.net/>
 - CTN = <http://www.erl.wustl.edu/DICOM/>
<ftp://dicomctn.wustl.edu/pub/dicom/software/ctn/>

構築手順の概要

- 手作りを補助するためのpatchの入手
 - <http://www.fujita-hu.ac.jp/~kmuto/webdas/>
 - 演者が公開しています
 - 本システムを『WebDAS』と呼んでいます ☺
 - CTNにJPEG画像出力機能を付加するソースコード
 - 「参照系」に使用するPHPスクリプト書類
 - DICOM Viewer Java applet
 - <http://mars.elcom.nitech.ac.jp/dicom/>
 - 構築のための手順書など
- PC に RedHat Linux をインストール
- 入手した材料は全て /tmp に入れておく
- ソフトウェアの make install
 - ソースコードをコンパイルして所定の場所へインストール
- ソフトウェアの動作設定 (configuration)

構築の詳細 1

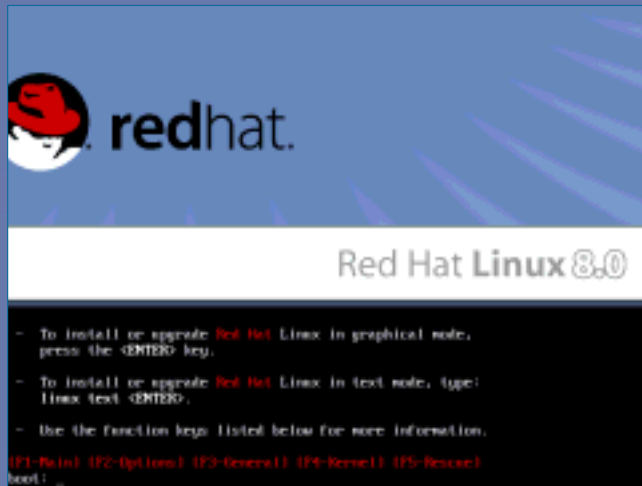
DICOM画像サーバの構築

DICOM画像サーバの構築

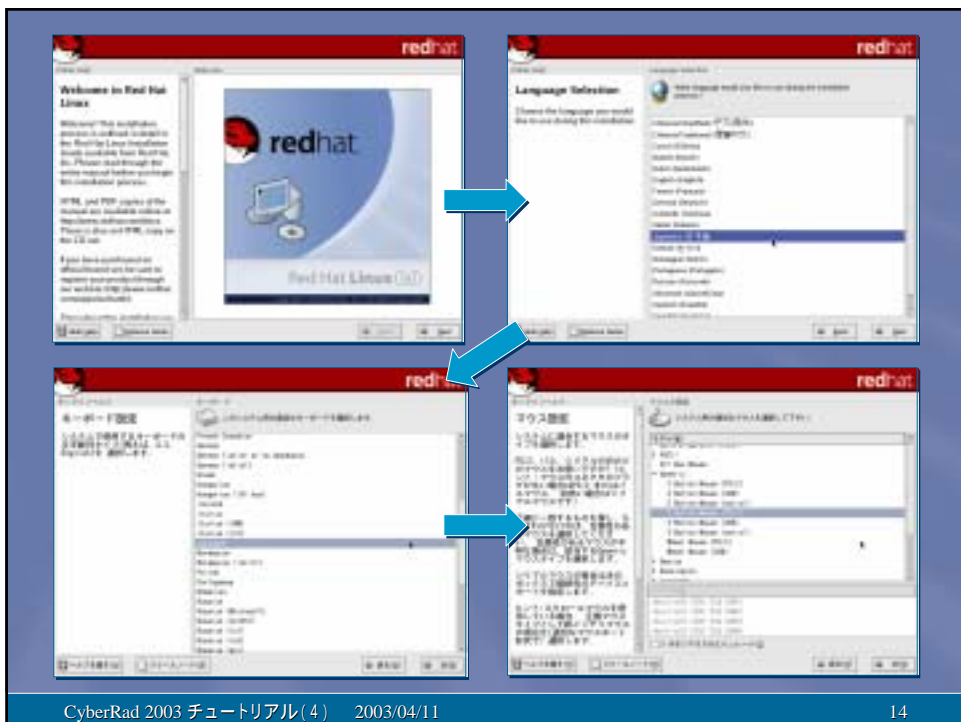
- RedHat Linux をPCにインストール
- Linux上に CTN をインストール

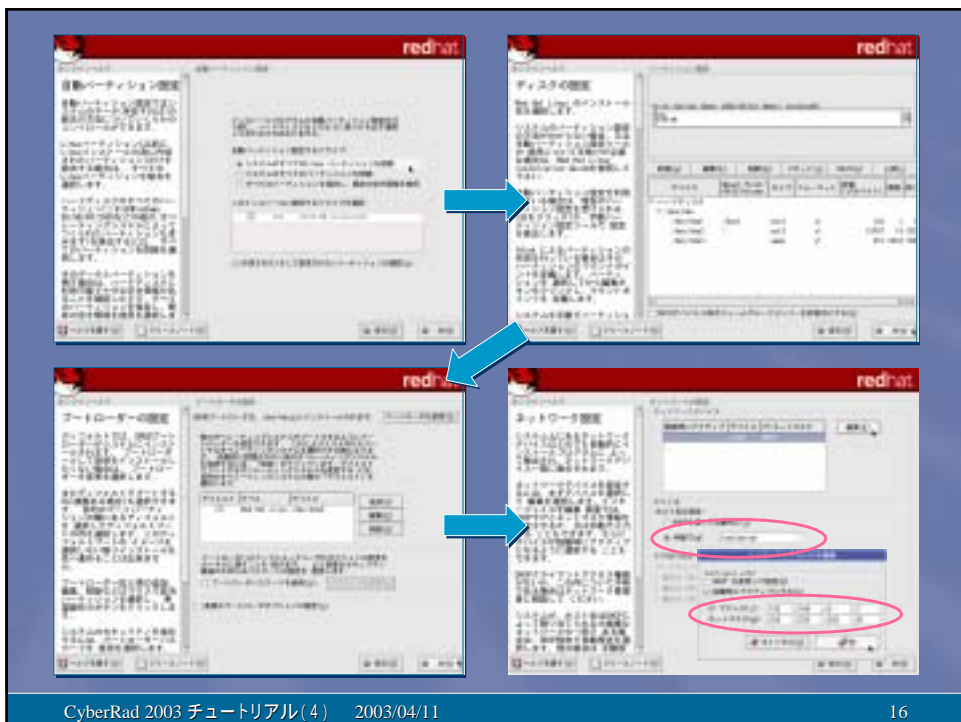
RedHatのインストール

- RedHat Linux 8J をPCにインストール
 - インストールはかなり簡単になってきました
 - RedHatディストリビューションに含まれる、PostgreSQL, apache, PHP4はインストールしない方針で
 - 後でそれぞれ自分で最新版をmake installします
 - 最新版 安全・安心 です
 - ネットワークはプライベートIPアドレスを固定で設定
 - 院内ネットワークでDHCPが動いていれば、それに対応することも可能です
 - 本システムは、あくまでも院内ネットワーク内のみで稼働させることを前提！



RedHat Linux 8JのCD-ROMを使って、CD-ROMドライブからPCを起動





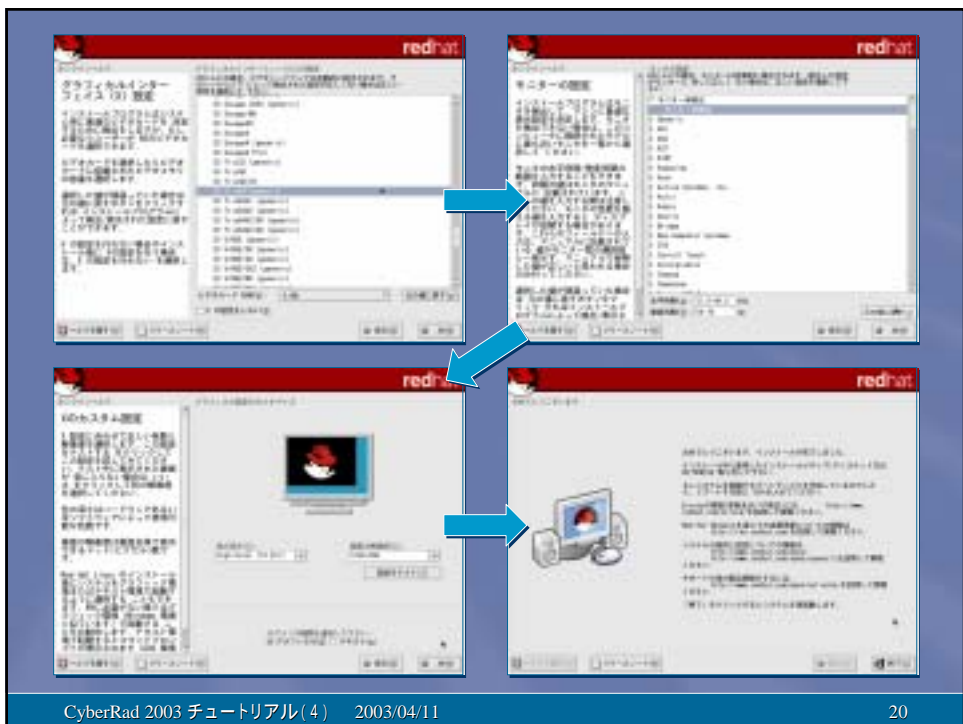
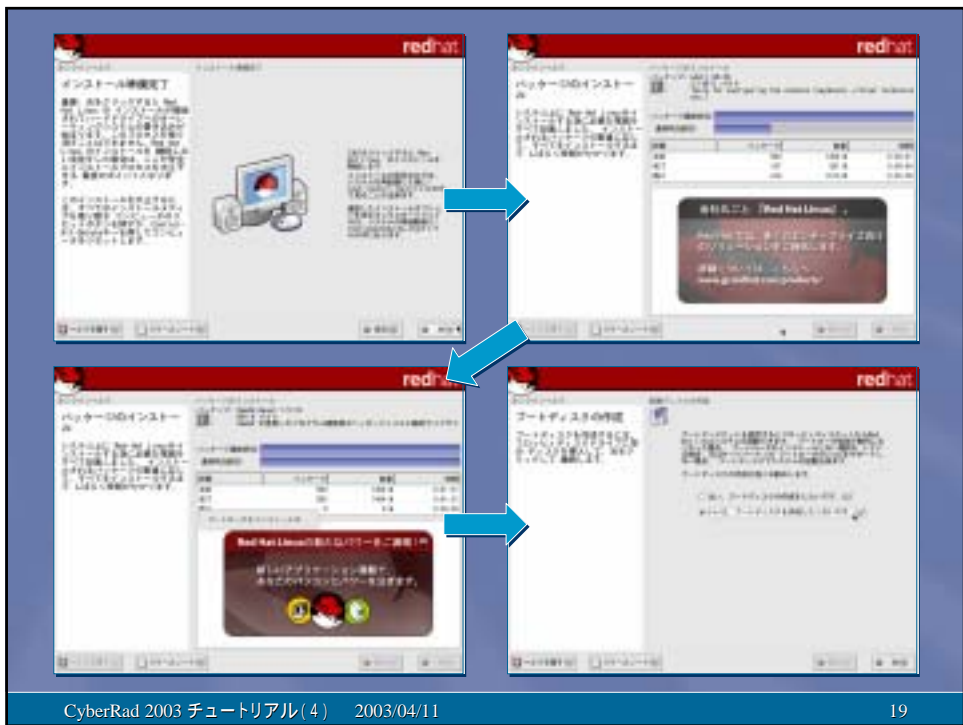
プライベートIPアドレス

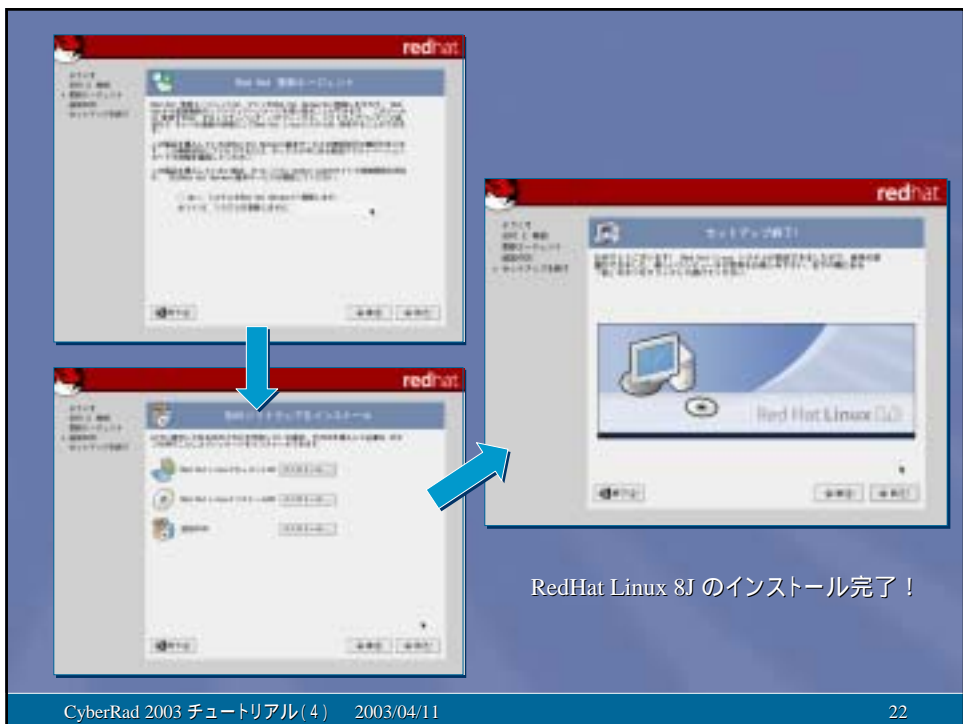
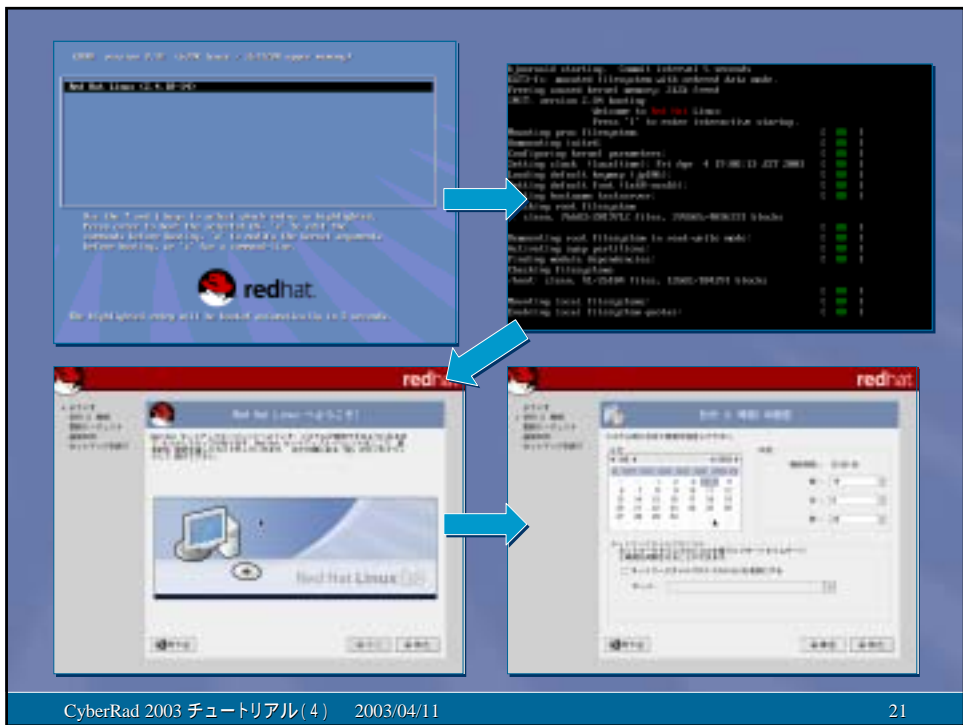
- インターネットに接続しない、LAN (Local Area Network) で使用するIPアドレス
 - RFC1918で決められている

IPアドレスの範囲	Net mask	Class / 最大ホスト数
10.0.0.0 ~ 10.255.255.255	255.0.0.0	A / 16,777,214 台
172.16.0.0 ~ 172.31.255.255	255.255.0.0	B / 65,534 台
192.168.0.0 ~ 192.168.255.255	255.255.255.0	C / 254 台

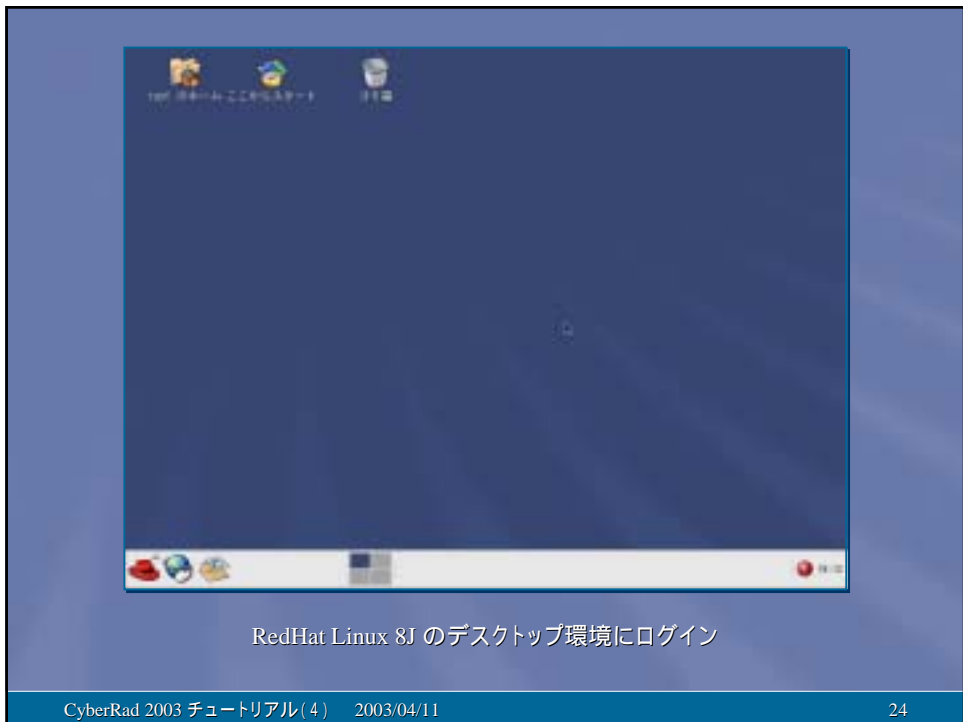
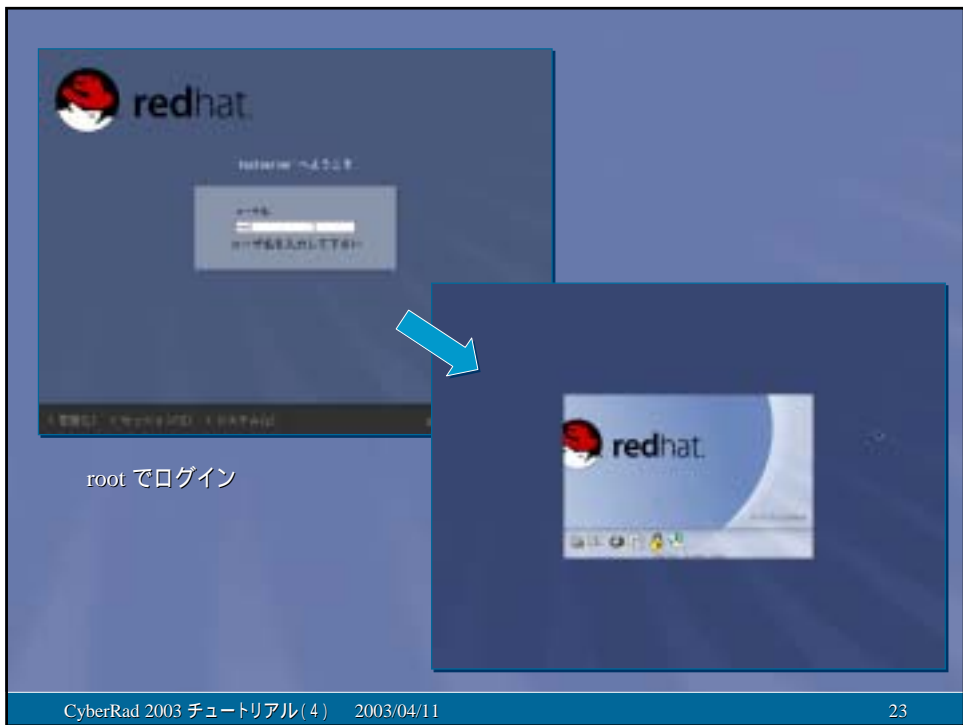
The image displays four Red Hat Linux configuration windows arranged in a 2x2 grid, connected by blue arrows indicating a sequence of steps. The windows are titled in Japanese and show various network configuration options:

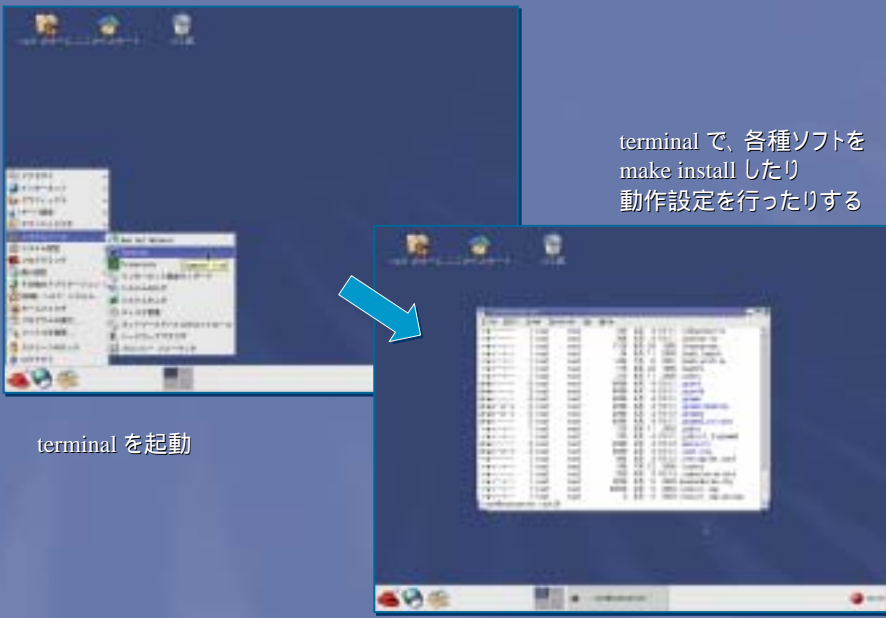
- Top-left:** "ファイアーウォール設定" (Firewall Settings) window, showing options for enabling the firewall and configuring SELinux.
- Top-right:** "ネットワークカードのインストール" (Network Card Installation) window, showing a list of available network cards and their drivers.
- Bottom-left:** "多ホスト環境の構成" (Multi-host Environment Configuration) window, featuring a world map and options for configuring a multi-host environment.
- Bottom-right:** "アカウントの設定" (Account Configuration) window, showing fields for setting up user accounts and passwords.





RedHat Linux 8J のインストール完了!





terminal で、各種ソフトを
make install したり
動作設定を行ったりする

terminal を起動

CyberRad 2003 チュートリアル(4) 2003/04/11 25

システムを実行するアカウントを作成

- CTNを実行するアカウント = 「ctn」
- PostgreSQLを動かすアカウント = 「postgres」

```
useradd -s /bin/tcsh ctn
passwd ctn
New UNIX password: ***** ctnユーザ用のパスワードを入力
Retype new UNIX password: ***** パスワードをもう一度入力
useradd postgres
passwd postgres
New UNIX password: ***** postgresユーザ用のパスワード入力
Retype new UNIX password: ***** パスワードをもう一度入力
```

PostgreSQLのインストール

- postgresql-7.2.3.tar.gz を展開し make install
- データベースの置き場所は /home/db
- /home/postgres/.bashrc に環境変数を登録

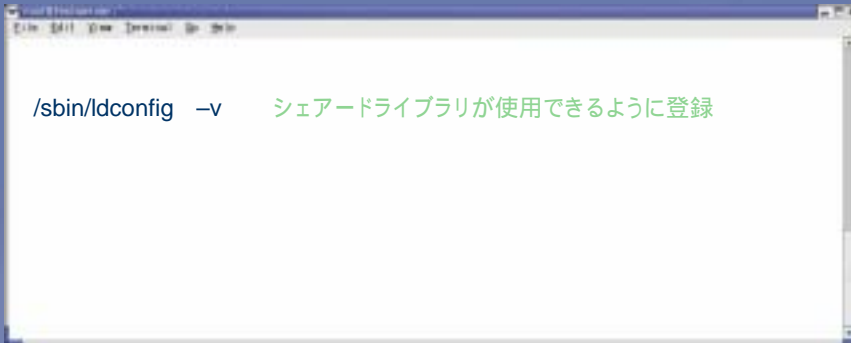
```
# pgsq env.
export POSTGRES_HOME=/usr/local/pgsql
export PGLIB=$POSTGRES_HOME/lib
export PGDATA=/home/db
#
export PATH=$PATH:$HOME/bin:$POSTGRES_HOME/bin
export MANPATH=$MANPATH:$POSTGRES_HOME/man
```

```
mkdir /usr/local/pgsql           PostgreSQLのインストール先
chown postgres:postgres /usr/local/pgsql
su postgres                       postgresユーザでインストール作業する
tar zxvf /tmp/postgresql-7.2.3.tar.gz   アーカイブを展開
cd postgresql-7.2.3
./configure --enable-multibyte=EUC_JP --enable-odbc
make
    :   たくさんのログが出力される
    :
make check
    :   たくさんのチェックログが出力される
    :
    All 79 tests passed.           ...と出力されれば全てOK

make install
```

PostgreSQLの動作設定1

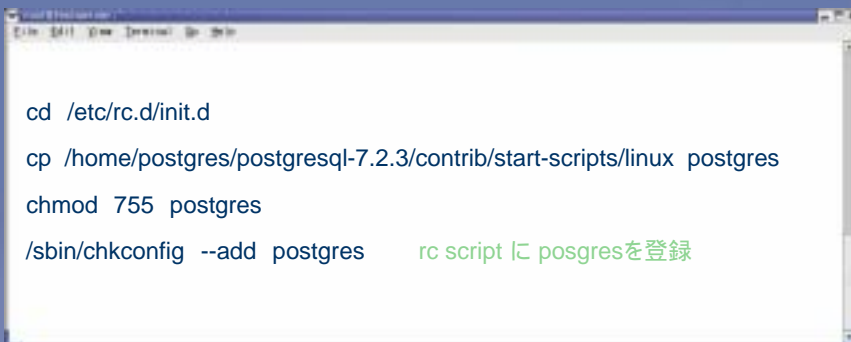
- シェアードライブラリの登録
 - /etc/ld.so.conf を編集
 - 『 /usr/local/pgsql/lib 』 を追加



```
File Edit View Settings Go Shell  
  
/sbin/ldconfig -v   シェアードライブラリが使用できるように登録
```

PostgreSQLの動作設定2

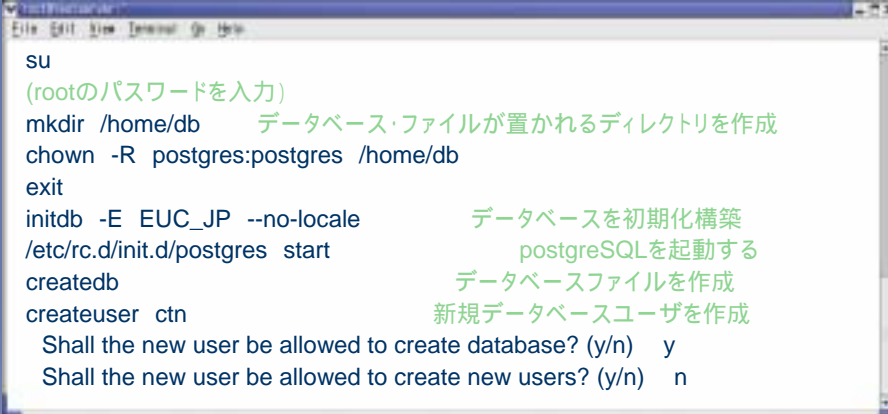
- postgresQLを起動するスクリプトを登録
 - /home/postgres/postgresql-7.2.3/contrib/start-scripts/linux を編集
 - 『 PGDATA="/home/db" 』 と変更



```
File Edit View Settings Go Shell  
  
cd /etc/rc.d/init.d  
cp /home/postgres/postgresql-7.2.3/contrib/start-scripts/linux postgres  
chmod 755 postgres  
/sbin/chkconfig --add postgres   rc script に postgresを登録
```

PostgreSQLの初期設定

- 一度、PC を reboot する
- ユーザ postgres でログイン



```
su
(rootのパスワードを入力)
mkdir /home/db      データベース・ファイルが置かれるディレクトリを作成
chown -R postgres:postgres /home/db
exit
initdb -E EUC_JP --no-locale      データベースを初期化構築
/etc/rc.d/init.d/postgres start   postgresQLを起動する
createdb                データベースファイルを作成
createuser ctn          新規データベースユーザを作成
Shall the new user be allowed to create database? (y/n)  y
Shall the new user be allowed to create new users? (y/n)  n
```

以上で postgresQL のインストールは
完了しました！

CTNの入手

- ctn-3.0.4.tar.Z を入手
 - <ftp://dicomctn.wustl.edu/pub/dicom/software>
 - 2003/4現在、最新版は ver.3.0.6のようです
- CTNへのpatchファイルを手入手
 - http://www.fujita-hu.ac.jp/~kmuto/webdas/linux/ctn-3.0.4-archive_server_jpeg.diff
 - Jpeg画像ファイル出力機能 (archive_server_jpeg)
 - JPEG変換には IJG の jpegライブラリを使用
<http://www.ijg.org/>
 - CTNのコンフィグレーション書類の補充、etc...

Jpeg画像出力機能

- archive_server の動作
 - DICOMデータを受け取るとメモリー内に保持
 - DICOMタグデータをデータベースにインサート
 - OSが提供するfile systemの所定のディレクトリに、DICOMデータをfileとして保存
- archive_server_jpeg の動作
 - OSが提供するfile systemの所定のディレクトリに、DICOMデータをfileとして保存
 - メモリー内のピクセルデータ(7fe0,0010)をコピーして、JPEGフォーマットに変換し、DICOMデータと同じ場所に、同じfile名で拡張子を“.jpg”に変えて保存
 - メモリー内のピクセルデータを縮小補完してからJPEGフォーマットに変換し、DICOMデータと同じ場所に、file名の最後に”_s”を付け、拡張子を“.jpg”に変えて保存

例)

- archive_server

/home/images/1_2_392_x_y_z_a_b/1_2_392_x_y_z_c_d/CT_340_1.dcm

- archive_server_jpeg

/home/images/1_2_392_x_y_z_a_b/1_2_392_x_y_z_c_d/CT_340_1.dcm

DICOM画像データ

/home/images/1_2_392_x_y_z_a_b/1_2_392_x_y_z_c_d/CT_340_1.jpg

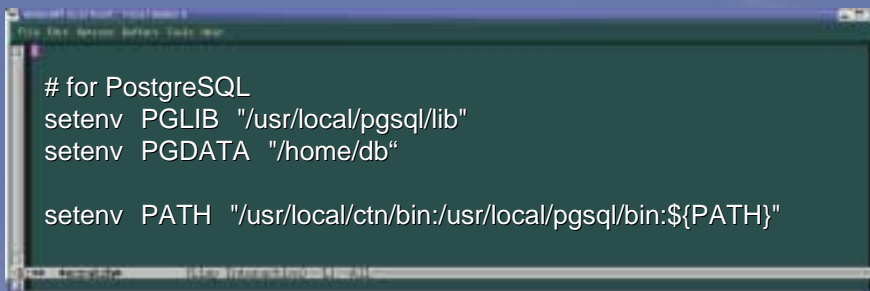
DICOM画像データと同じ解像度のjpeg画像データ

/home/images/1_2_392_x_y_z_a_b/1_2_392_x_y_z_c_d/CT_340_1_s.jpg

DICOM画像を縮小してからjpegへ変換したサムネイル画像データ

CTNのインストール

- ctn-3.0.4.tar.Z を展開
 - CTN は /usr/local/ctn にインストールする
- ユーザ ctn でログイン
- /home/ctn/.cshrc に環境変数を登録



```
# for PostgreSQL
setenv PGLIB "/usr/local/pgsql/lib"
setenv PGDATA "/home/db"

setenv PATH "/usr/local/ctn/bin:/usr/local/pgsql/bin:${PATH}"
```

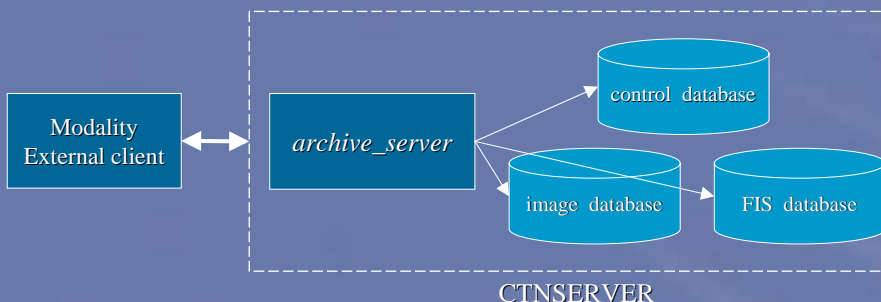
```

su
(rootのパスワードを入力)
mkdir /usr/local/ctn
chown ctn:ctn /usr/local/ctn
exit
mkdir ctn-3.0.4
cd ctn-3.0.4
tar zxvf /tmp/ctn-3.0.4.tar.Z      アーカイブを展開
patch -p1 -i /tmp/ctn-3.0.4-archive_server_jpeg.diff  patchをあてる
source ./environments/linux/linux.psqli.lesstif.noopt.env  makeの為の準備
cd facilities ; make links ; cd ..
cd libsrc ; make install ; cd ..
cd apps ; make install ; make gui-install ; cd ..

```

CTNの動作設定

- CTNが使用するデータベースを作る



- control database = CTNSERVERへのアクセスを制御
- Image database = Q/R, Storage サービスに利用
- FIS database = Storage Commitment 等に利用
 - archive_server の他に archive_agent, ris_gateway を併用する
 - 本チュートリアルでは FIS database は用いないで構築

```

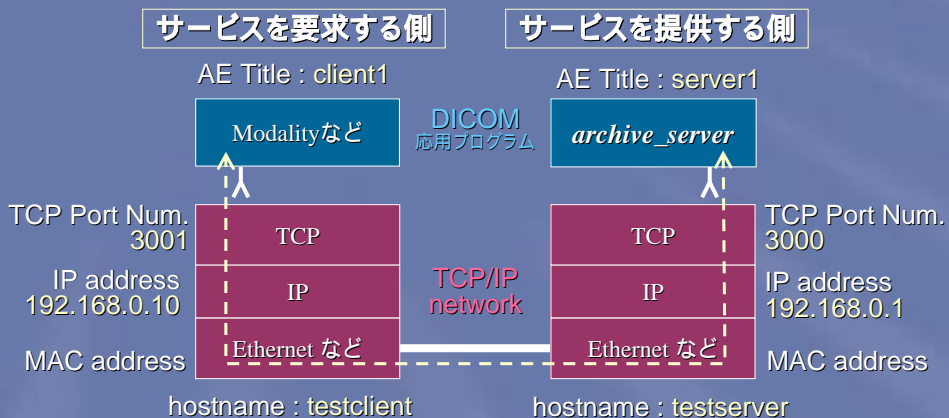
createdb                                postgresSQLのデータベースファイルを作成
cd ~/ctn-3.0.4/cfg_scripts/postgresql

./CreateDB CTNControl                  "CTNControl"という名前のcontrol database を作成
./CreateDB dicomImage                  "dicomImage"という名前のimage database の作成
./CreateTables Control CTNControl      データベースにテーブルを作成
./CreateTables DIM2 dicomImage         データベースにテーブルを作成
./CreateTables DIMView dicomImage     データベースにビューを作成
cd

```

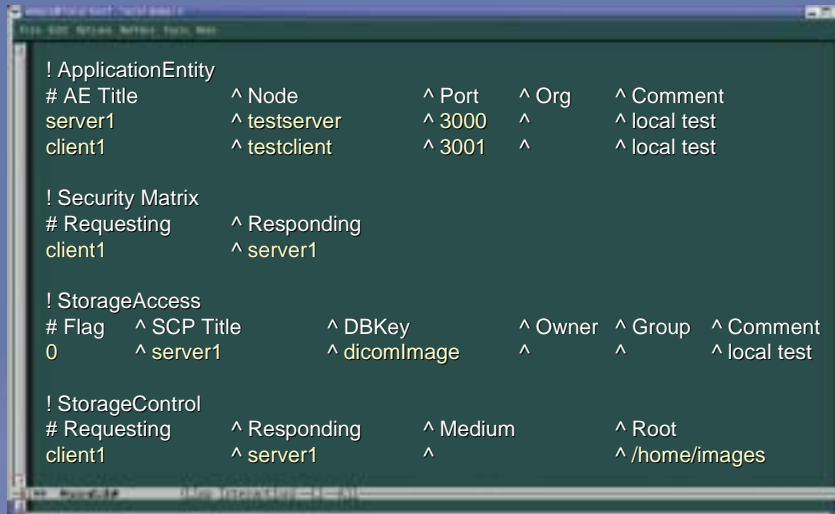
CTNControl データベースの調整

- CTNの画像サーバに接続可能な AE (Application Entity) を登録



CTNControlへ登録する情報を作成

- /home/ctn/test.ctn へ保存



```
! ApplicationEntity
# AE Title      ^ Node      ^ Port  ^ Org  ^ Comment
server1        ^ testserver ^ 3000  ^      ^ local test
client1        ^ testclient ^ 3001  ^      ^ local test

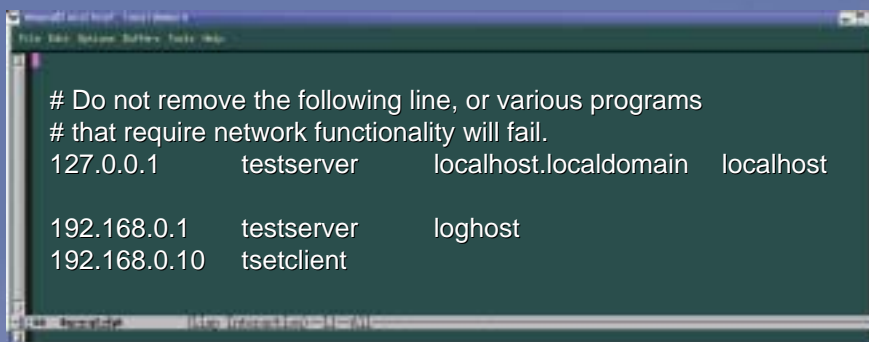
! Security Matrix
# Requesting    ^ Responding
client1        ^ server1

! StorageAccess
# Flag ^ SCP Title      ^ DBKey      ^ Owner ^ Group ^ Comment
0      ^ server1        ^ dicomImage ^      ^      ^ local test

! StorageControl
# Requesting    ^ Responding ^ Medium ^ Root
client1        ^ server1    ^      ^ /home/images
```

/etc/hosts に情報を登録

- testclient の名前解決 (hostnameとIPアドレスの一致) ができるようにする
- /etc/hosts を編集



```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1      testserver    localhost.localdomain localhost

192.168.0.1    testserver    loghost
192.168.0.10  tsetclient
```

CTNControl DBへ情報を登録

- DICOM画像データを保存するディレクトリは /home/images とする

```
su
(rootのパスワードを入力)
mkdir /home/images      画像データの保存先を作成
chown ctn:ctn /home/images
exit
load_control < /home/ctn/test.ctn      CTNControl へ登録
```

archive_server の起動

```
archive_server_jpeg

Usage: [-e] [-f db] [-g generic] [-i] [-j quality] [-k width] [-l logfile] [-n node] [-o max] [-q] [-r] [-t] [-v] [-z queue] port

-e Examine received images and do SOP validation
-f Use db as control database instead of default (CTNControl)
-g An override in the Security Matrix. If an application is configured to connect to <generic>, it has access to all applications on this server.
-i Ignore some problems in Association Request
-j Make Jpeg file and set compression quality.
-k Set width of jpeg thumbnail image. (default: 128 pixel)
-l Place log of association requests in <logfile>
-n Use <node> as name of server rather than hostname
-o Allow <max> simultaneous connections for an organization
-r Reduced capability. Turn off anything using FIS database
-q Quiet mode, don't dump a lot of messages to terminal
-t Use threaded version.
-v Place DUL and SRV facilities in verbose mode
-z Turn on queueing of status information

port TCP/IP port address

archive_server_jpeg -i -r -j 80 3000      jpeg出力も行う設定で起動
```

以上で CTN のインストールは
完了しました！

構築の詳細 2

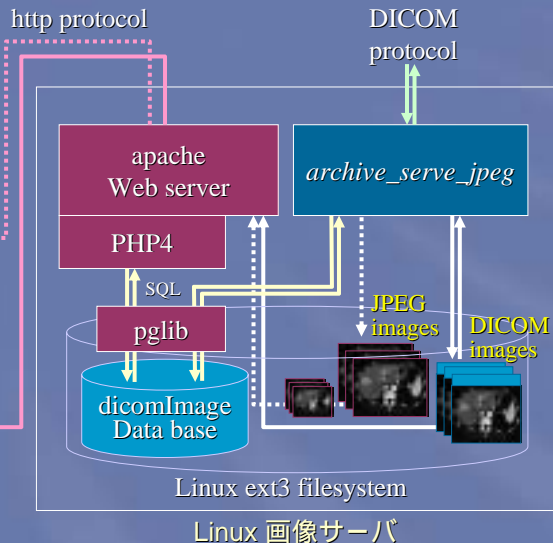
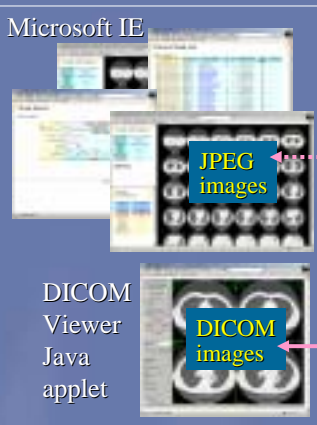
「参照系」の構築

「参照系」の構築

- WWW関連のソフトウェアをインストール
 - Webサーバ : apache
 - サーバサイド言語 : PHP4
 - PostgreSQLからデータを取り出し、動的にwebページを生成
- 「参照系」を実現するPHP書類のインストールとwebサーバの設定
 - WebブラウザでDICOM画像サーバ内の画像データ一覧や患者・検査の検索が行えるようなwebアプリケーションをPHPでプログラミング

「参照系」のシステム構成

画像データはURLで特定
httpで転送



apache のインストール

- apache_1.3.27.tar.gz を展開
 - apache は /usr/local/apache へインストールする
- root でログインする

```
tar zxvf /tmp/apache_1.3.27.tar.gz
cd apache_1.3.27
OPTIM="-O2" ./configure --enable-module=so
make
make install
```

apache の起動設定 1

- 『 httpd 』 というテキストファイルを作成する

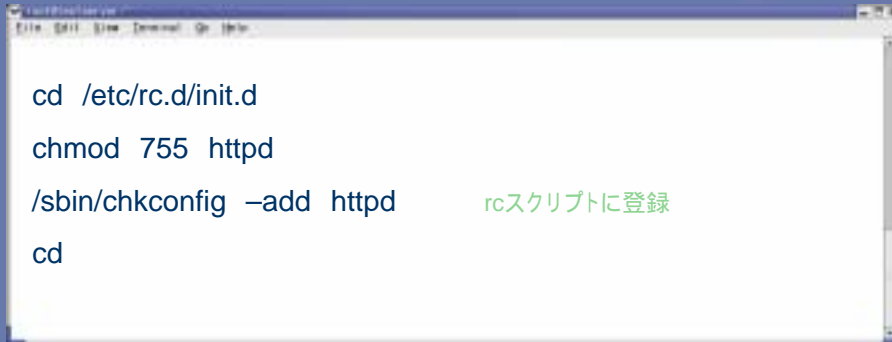
```
#!/bin/sh
# Startup script for the Apache Web Server
#
# chkconfig: 345 85 15
# description: Apache is a World Wide Web server.
# processname: httpd
# pidfile: /var/run/httpd.pid
# config: /usr/local/apache/conf/httpd.conf

# Source function library.
. /etc/rc.d/init.d/functions

case "$1" in
  start)
    echo -n "Starting httpd: "
    daemon /usr/local/apache/bin/httpd
    echo
    touch /var/lock/subsys/httpd
    ;;
  stop)
    echo -n "Shutting down httpd: "
    killproc httpd
    echo
    rm -f /var/lock/subsys/httpd
    rm -f /var/run/httpd.pid
    ;;
  status)
    status httpd
    ;;
  restart)
    $0 stop
    $0 start
    ;;
  reload)
    echo -n "Reloading httpd: "
    killproc httpd -HUP
    echo
    ;;
  *)
    echo "Usage: $0
{start|stop|restart|reload|status}"
    exit 1
esac
exit 0
```

apache の起動設定2

- apacheを起動するスクリプト『httpd』を登録



```
cd /etc/rc.d/init.d
chmod 755 httpd
/sbin/chkconfig --add httpd      rcスクリプトに登録
cd
```

apacheの動作設定

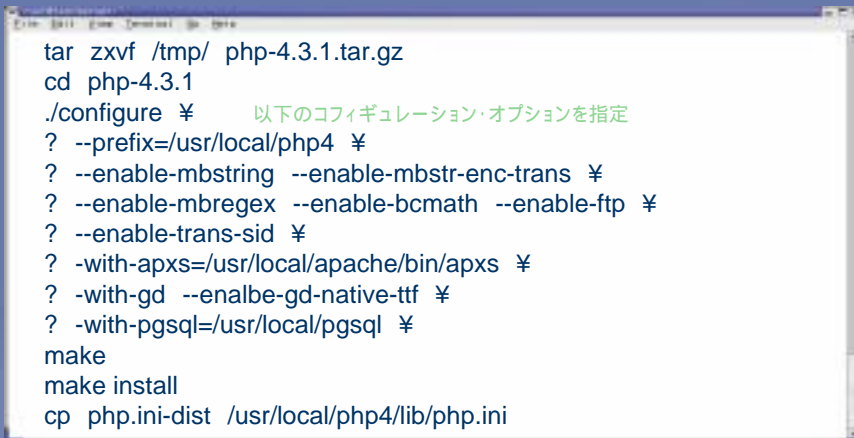
- apacheの動作設定ファイル『httpd.conf』を編集
 - /usr/local/apache/conf/http.conf
 - Webサーバで公開するディレクトリを設定
 - /home/webdas/
 - php書類やhtml書類を置く場所
 - http://hostname/ でアクセスできるように
 - /home/images/
 - DICOM画像、JPEG画像が保存される場所
 - » CTNControl データベースの StorageControl テーブルの Root カラムで指定したディレクトリ
 - http://hostname/images/ でアクセスできるように
 - PHP4のMIMEタイプを追加
 - PHPスクリプトがサーバ側で実行可能に設定

```
DocumentRoot "/home/webdas"    "http://hostname/" でアクセスできるように
...
<IfModule mod_mime.c>
AddType application/x-httpd-php .php    PHPのMIMEタイプを追加
...
# This should be change to whatever you set DocumentRoot to
<Directory "/home/webdas">
...
<IfModule mod_dir.c>                index.php をデフォルトに追加
DirectoryIndex index.html index.php index.cgi
</IfModule>
...
ErrorLog /home/ctn/httpd_logs/error_log    logは/home/ctn/httpd_logsへ保存
...
CustomLog /home/ctn/httpd_logs/access_log common
...
<IfModule mod_alias.c>                http://hostname/images/でアクセスできるように
Alias /images/ "/home/images/"
  <Directory "/home/images/">
    Options Indexes MultiViews        ディレクトリ参照を許可
    AllowOverride None
    Order allow,deny
    Allow from all
  </Directory>
</IfModule>
```

以上で apache のインストールは
完了しました！

PHP4のインストール

- php-4.3.1.tar.gz を展開
 - PHP4 は /usr/local/php4 へインストールする



```
tar zxvf /tmp/ php-4.3.1.tar.gz
cd php-4.3.1
./configure ¥      以下のコフィギュレーション・オプションを指定
? --prefix=/usr/local/php4 ¥
? --enable-mbstring --enable-mbstr-enc-trans ¥
? --enable-mbregex --enable-bcmath --enable-ftp ¥
? --enable-trans-sid ¥
? -with-apxs=/usr/local/apache/bin/apxs ¥
? -with-gd --enable-gd-native-ttf ¥
? -with-pgsql=/usr/local/pgsql ¥
make
make install
cp php.ini-dist /usr/local/php4/lib/php.ini
```

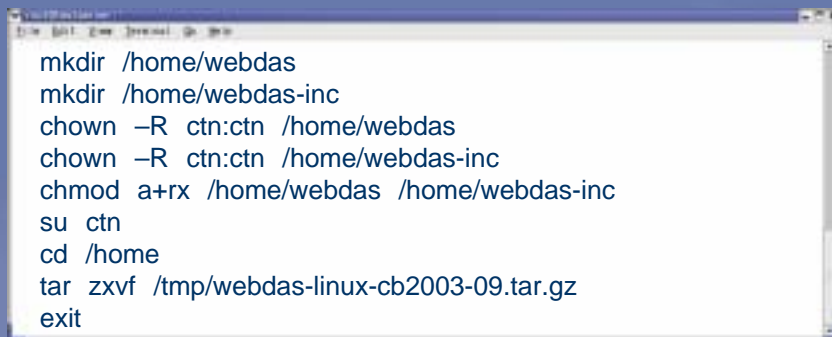
PHP4の初期設定

- PHP4の動作設定ファイル『php.ini』を編集
 - /usr/local/php4/lib/php.ini
 - 変更箇所
 - 『 Register_globals = On 』
 - セキュリティ上問題となる可能性がある！
しかし、「参照系」のwebアプリケーションを記述するPHPスクリプトは「On」を前提として作成してある
今後修正の予定です (^_^) ;
 - 『 include_path = “./:/home/webdas-inc” 』
 - Webアプリケーションの動作設定ファイルを、クライアントのwebブラウザからは参照できないようにするため

以上で PHP4 のインストールは
完了しました！

「参照系」のインストール

- WebDASの入手
 - <http://www.fujita-hu.ac.jp/~kmuto/webdas/linux/webdas-linux-cb2003-09.tar.gz>
 - /home/webdasおよび/home/webdas-incへ展開



```
mkdir /home/webdas
mkdir /home/webdas-inc
chown -R ctn:ctn /home/webdas
chown -R ctn:ctn /home/webdas-inc
chmod a+rx /home/webdas /home/webdas-inc
su ctn
cd /home
tar zxvf /tmp/webdas-linux-cb2003-09.tar.gz
exit
```

以上で WebDAS のインストールは
完了しました！

参照系の機能

どうやって検査毎の画像を表示するのか

DicomImage データベースの中身



画像データ検索

- 欲しい情報を取得するためのSQL文を作成
 - 患者ID「123456」をもつ検査を取得
 - SELECT * FROM StudyLevel WHERE PatParent = '123456'
 - クエリーの結果として、PatParentが「123456」であるStudyLevelテーブルの情報が返ってくる
 - 検査UID「1.2.392.x.y.z.a.b」を持つシリーズを取得
 - SELECT * FROM SeriesLevel WHERE StuParent = '1.2.392.x.y.z.a.b'
 - シリーズUID「1.2.392.x.y.z.c.d」を持つ画像の所在 (path) を取得
 - SELECT Path FROM ImageLevel, InstanceTable WHERE (SerParent = '1.2.392.x.y.z.c.d' AND ImageLevel.SOPInsUID = InstanceTable.ImageUID)

PHP4の役割

- PHP4はPostgreSQLへ接続し、SQLのクエリーを発行して、その結果の「表」データを配列変数に保持

```
- $objConn = pg_connect("user=ctn password=passwd
                        dbname=dicomImage");           // dicomImageデータベースへ接続
- $SQL = "SELECT * FROM ImageLevel, InstanceTable WHERE
        (SerParent = '1.2.392.x.y.z.c.d' AND ImageLevel.SOPInsUID =
        InstanceTable.ImageUID)";                   // 必要な情報を得るためのSQL文
- $objResult = pg_exec($objConn, $SQL); // クエリーを掛ける
- $rows = pg_numrows($objResult); // クエリー結果が何行あるか数える
- $val = pg_result($objResult, 1, "Path"); // 結果中の1行目の"Path"カラム

- 変数 $val には
  "/home/images/1_2_392_x_y_z_a_b/1_2_392_x_y_z_c_d/CT_340_1.dcm"
  のように、Linuxのファイルシステム上でのファイルの所在が格納される。
```

PHP4の役割

- クエリーの結果として得た「表」データを、webブラウザで見えるようにHTMLへと加工する

```
- 変数 $val には
  "/home/images/1_2_392_x_y_z_a_b/1_2_392_x_y_z_c_d/CT_340_1.dcm"
  のように、Linuxのファイルシステム上でのファイルの所在が格納される。
- $strPath = str_replace("/home/images", "", $val);
- $strPath = str_replace(".dcm", ".jpg", $strPath);
- $strPath = "http://hostname/images" . $strPath;
- echo "<IMG SRC=¥$strPath¥>";

- Webブラウザには
  「 <IMG SRC="http://hostname/images/1_2_392_x_y_z_a_b/
                                1_2_392_x_y_z_c_d/CT_340_1.jpg"> 」
  というHTMLタグとして送信され、ブラウザにJPEG画像が表示される
```


WebDASの構成

- /home/webdas/
 - menu.php http://hostname/ でアクセスされるディレクトリ topページ
 - patient-study.php 患者-検査一覧作成
 - refimage_topflame.php 1検査を表示するためのフレームを生成
 - study.php 検査情報を表示
 - series.php シリーズ情報を表示
 - image.php JPEGサムネイル画像、オリジナルサイズ画像表示
 - dicomview.php DICOM viewer Java applet 起動スクリプト
 - /dicomviewer/* DICOM viewer Java applet 本体
 - search_top.php 検査情報検索ページ
 - search_result.php 検索結果表示ページ
 - dcmdump.php DICOM ヘッダ情報表示
 - list.css Webページのスタイルシート
 - /img/* Webページで使用するアイコンなどの画像
 - /inc/* DHTML関連の Javascript 書類
- /home/webdas-inc/
 - webdassite.php データベースにアクセスするためのpasswd等を保持
 - common.php Webページの表示方法などを制御するinclude書類
 - common-mb.php 患者名のローマ字 ひらがな変換機能

患者-検査リスト

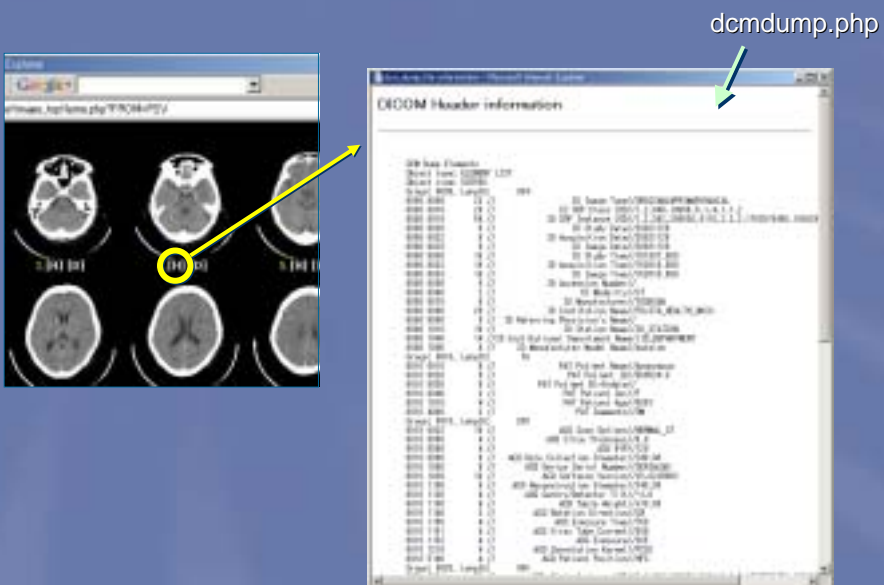
patient-study.php

Study Date	Patient ID	Patient Name	Age	Sex	Modality (MR)	# of studies	Comment
2002/1/28 18:33:37	00003-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:40:31	00003-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:20:30	00004-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:20:30	00004-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:11:00	00005-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:45	00006-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00006-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:29:04	00007-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:21:00	00007-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:21:00	00007-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00008-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00008-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00009-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00009-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00010-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00010-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00011-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00011-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00012-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00012-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00013-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00013-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00014-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00014-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00015-0	山田 太郎	60	男	ST (head)	10	
2002/1/28 18:00:30	00015-0	山田 太郎	60	男	ST (head)	10	

検査サムネイル表示

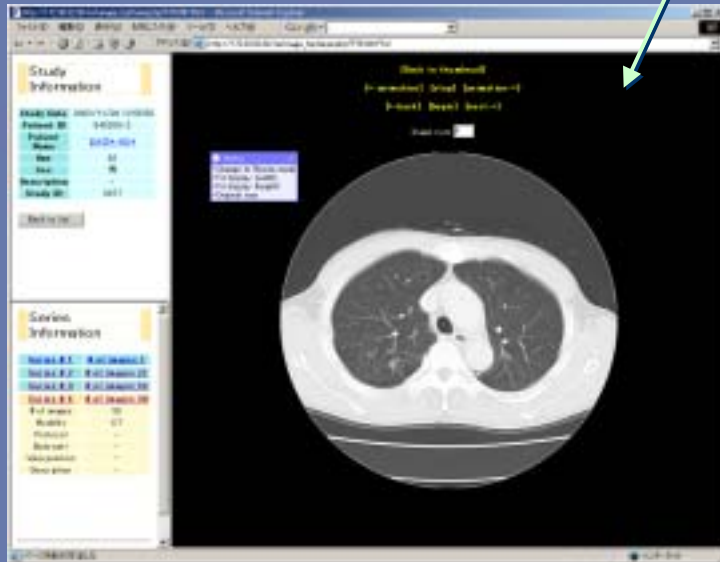


DICOM tag 情報表示

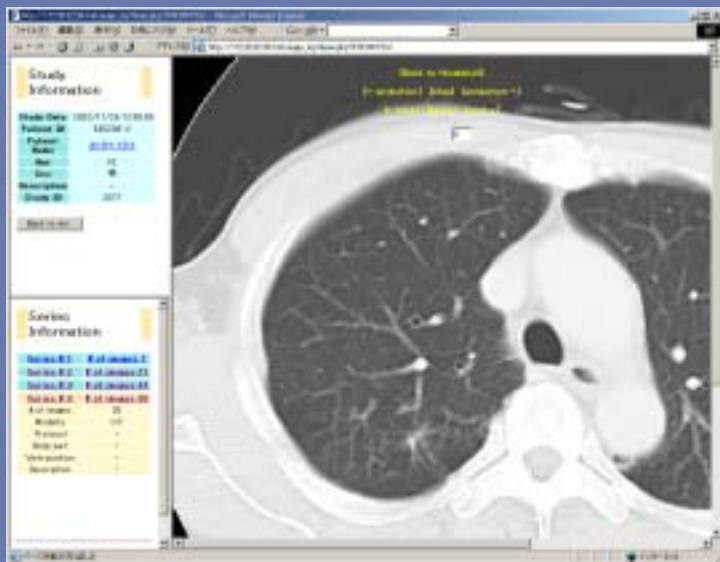


1 画像表示

image.php



画像のリサイズ(DHTMLの機能)



画像データの一覧をURLで保持

```
var images = new Array();
var image = new Image();
var image2 = new Image();
var image3 = new Image();
var image4 = new Image();
var image5 = new Image();
var image6 = new Image();
var image7 = new Image();
var image8 = new Image();
var image9 = new Image();
var image10 = new Image();
var image11 = new Image();
var image12 = new Image();
var image13 = new Image();
var image14 = new Image();
var image15 = new Image();
var image16 = new Image();
var image17 = new Image();
var image18 = new Image();
var image19 = new Image();
var image20 = new Image();
var image21 = new Image();
var image22 = new Image();
var image23 = new Image();
var image24 = new Image();
var image25 = new Image();
var image26 = new Image();
var image27 = new Image();
var image28 = new Image();
var image29 = new Image();
var image30 = new Image();
var image31 = new Image();
var image32 = new Image();
var image33 = new Image();
var image34 = new Image();
var image35 = new Image();
var image36 = new Image();
var image37 = new Image();
var image38 = new Image();
var image39 = new Image();
var image40 = new Image();
var image41 = new Image();
var image42 = new Image();
var image43 = new Image();
var image44 = new Image();
var image45 = new Image();
var image46 = new Image();
var image47 = new Image();
var image48 = new Image();
var image49 = new Image();
var image50 = new Image();
var image51 = new Image();
var image52 = new Image();
var image53 = new Image();
var image54 = new Image();
var image55 = new Image();
var image56 = new Image();
var image57 = new Image();
var image58 = new Image();
var image59 = new Image();
var image60 = new Image();
var image61 = new Image();
var image62 = new Image();
var image63 = new Image();
var image64 = new Image();
var image65 = new Image();
var image66 = new Image();
var image67 = new Image();
var image68 = new Image();
var image69 = new Image();
var image70 = new Image();
var image71 = new Image();
var image72 = new Image();
var image73 = new Image();
var image74 = new Image();
var image75 = new Image();
var image76 = new Image();
var image77 = new Image();
var image78 = new Image();
var image79 = new Image();
var image80 = new Image();
var image81 = new Image();
var image82 = new Image();
var image83 = new Image();
var image84 = new Image();
var image85 = new Image();
var image86 = new Image();
var image87 = new Image();
var image88 = new Image();
var image89 = new Image();
var image90 = new Image();
var image91 = new Image();
var image92 = new Image();
var image93 = new Image();
var image94 = new Image();
var image95 = new Image();
var image96 = new Image();
var image97 = new Image();
var image98 = new Image();
var image99 = new Image();
var image100 = new Image();
```

Javascriptで画像のURLを配列変数に保持することで、アニメーション等を実現できる

DICOM Viewer Java applet



最後に・・・

- どうやって情報を手に入れるか
 - PostgreSQL、apache、PHP4によるwebとデータベースの連携は、Internetでは良く使われる手法
 - Internet上にたくさん情報はあります！
- WWWで情報を得よう！
 - 解らないことは、まずGoogleで検索
<http://www.google.com/>
 - @IT IT関連技術の解説がたくさん！
<http://www.atmarkit.co.jp/>

ご清聴ありがとうございました。