



ビデオ教材検索システム「Whale」  
インストールマニュアル  
ver.170428

## 目次

1. はじめに.....	1
2. システム要件.....	1
3. インストールに必要なもの.....	1
4. インストール手順.....	2
4.1. Whale 本体の設置.....	2
4.2. 各種パッケージのインストール.....	3
4.3. 設定.....	8
4.3.1. Whale の設定.....	8
4.3.2. Core の設定.....	8
4.3.3. Web の設定.....	8
4.4. その他のソフトウェアの設定.....	9
4.4.1. PHP の設定.....	9
4.4.2. MySQL の設定.....	9
4.4.3. Perl の設定.....	10
4.4.4. Cron の設定.....	11
5. アンインストール.....	12
6. Q&A.....	12
7. 付録.....	13
7.1. Whale ツリー構造.....	13
7.2. ipadic の utf-8 変換スクリプト (ipadic_conv_utf8.sh).....	17
7.3. データベース自動生成スクリプト (whale_database.sql).....	17
7.4. ffmpeg-php インストールのエラー修正.....	18

## 1. はじめに

本ドキュメントは、授業ビデオ検索システム「Whale」を CentOS 上でのインストールの手順を示したものです。

## 2. システム要件

以下にあげるシステム要件は、開発時の動作環境です。あくまでも例なので、その他の Linux ディストリビューションの環境でも知識があれば動作可能です。

OS	CentOS 6.8
CPU	デュアルコア以上のものが望ましい
メモリ	2GB 以上

## 3. インストールに必要なもの

- (1) Linux サーバ
- (2) Whale 本体 (Core+Web)
- (3) Apache (HTTP サーバ)
- (4) PHP(5.6.29 以上推奨)
- (5) MySQL (データベース)
- (6) Julius (大語彙連続音声認識システム)
- (7) Darts (Chasen を動かすのに必要なライブラリ)
- (8) Chasen (形態素解析システム)
- (9) ipadic (Chasen 用の辞書)
- (10) rpmforge (ffmpeg のインストールに必要なリポジトリ)
- (11) ffmpeg (動画と音声の変換ソフト)
- (12) ffmpeg-php (PHP で ffmpeg を使う)
- (13) Kaldi (英語の音声認識システム)
- (14) sox (音声のフィルタ)

#### 4. インストール手順

本章では Whale のインストールの手順について説明します。なお、本システムは Linux ディストリビューションのひとつである CentOS でインストールをします。

今回は /whale ディレクトリ以下に Whale (ver170428) をインストールすると仮定して進めていくことにします。

##### 4.1. Whale 本体の設置

本体を解凍し、公開したいディレクトリ上に設置します。また、ビデオのアップロード用のディレクトリを作成し、Core と Web にそのリンクを用意します。

まず whale\_170428.tar.gz をブラウザまたは wget 等のコマンドを用いて、/home/(username)ディレクトリにダウンロードしてください。

以下のコマンドで解凍し、/whale に移動させます。これで配置は完了です。

```
$ tar zxvf whale_170428.tar.gz  
$ mv whale_170428 /whale
```

## 4.2. 各種パッケージのインストール

Whale の動作に必要な各種パッケージをインストールします。すでにインストールされているパッケージがある場合はそのパッケージのインストールを飛ばしてください。

自動インストールスクリプト `install_CentOS.sh` を用意してあるので、どうしてもインストールがわからない方は以下のコマンドを実行して各種パッケージをインストールください。スクリプトの内容は本節に記載してあるコマンドを順に実行するものとなっています。

```
$ cd /whale/util/  
$ chmod +x install_CentOS.sh  
$ ./install_CentOS.sh
```

### (3) Apache (HTTP サーバ)

HTTP サーバである Apache をインストールします。

```
$ yum install -y httpd
```

### (4) PHP

PHP 本体をインストールします。

バージョンは 5.6.29 以上で動作確認。

```
$ yum install -y php
```

### (5) MySQL(データベース)

MySQL データベースとライブラリをインストールします。MySQL インストール時にパスワードを設定する画面が表示されますので、任意のパスワードに設定してください。

```
$ yum install -y mysql mysql-server  
$ yum install -y php-mysql
```

### (6) Julius(大語彙連続音声認識システム)

Julius をインストールします。apt-get でもインストールできますが、本システムでは使用することができませんので、以下の手順に従ってインストールしてください。

ブラウザ等から `julius-4.2.2.tar.gz` のアーカイブファイルをダウンロードして home ディレクトリにおいてください。

\*どうしてもコマンドラインしか使用できない方は `wget` を利用してください。ただし、バージョンにより URL が変わるのでご注意ください。

```
$ wget http://ij.dl.sourceforge.jp/julius/56549/julius-4.2.2.tar.gz
```

ホームディレクトリに配置した場合は、まずは以下のコマンドで展開します。

```
$ tar xzf julius-4.2.2.tar.gz  
$ cd julius-4.2.2
```

本システムで使用するために Julius のパラメータを変更してからコンパイル及びインストールを

行います。sed コマンドで置換を行っていますが、うまくいかない場合はテキストエディタで直接変更してください。スペースの数に注意。

```
$sed -e 's/#define MAXSEQNUM          150/#define MAXSEQNUM          10000/'  
libsent/include/sent/speech.h > new1.h  
$sed -e 's/#define MAXSPEECHLEN  320000/#define MAXSPEECHLEN  32000000/' ./new1.h  
> new2.h  
$ mv new2.h libsent/include/sent/speech.h  
$ ./configure  
$ make  
$ make install
```

(7) Darts (Chasen を動かすのに必要なライブラリ)

Darts をインストールします。Chasen より必ず先にインストールしてください。ブラウザ等から darts-0.32.tar.gz のアーカイブファイルをダウンロードして home ディレクトリにおいてください。Darts のインストールの途中でコンパイルするためのコンパイラをインストールします。

\*どうしてもコマンドラインしか使用できない方は wget を利用してください。ただし、バージョンにより URL が変わるのでご注意ください。

```
$ wget http://chasen.org/~taku/software/darts/src/darts-0.32.tar.gz
```

ホームディレクトリに配置した場合は、以下のコマンドによりインストールします。

```
$ tar xzf darts-0.32.tar.gz  
$ cd darts-0.32  
$ yum install -y gcc-c++  
$ ./configure  
$ make  
$ make check  
$ make install
```

(8) Chasen (形態素解析システム)

Chasen をインストールします。ブラウザ等から chasen-2.4.5.tar.gz のアーカイブファイルをダウンロードして home ディレクトリにおいてください。

\*どうしてもコマンドラインしか使用できない方は wget を利用してください。ただし、バージョンにより URL が変わるのでご注意ください。

```
$ wget http://ijl.dl.sourceforge.jp/chasen-legacy/56305/chasen-2.4.5.tar.gz
```

ホームディレクトリに配置した場合は、以下のコマンドによりインストールします。

```
$ tar xzf chasen-2.4.5.tar.gz  
$ cd chasen-2.4.5  
$ ./configure  
$ make  
$ make install
```

(9) ipadic (Chasen 用の辞書) ,

ipadic をインストールします。ブラウザ等から ipadic-2.7.0.tar.gz の archive ファイルをダウンロードして home ディレクトリにおいてください。

\*どうしてもコマンドラインしか使用できない方は wget を利用してください。ただし、バージョンにより URL が変わるので、ご注意ください。

```
$ wget http://jaist.dl.sourceforge.jp/ipadic/24435/ipadic-2.7.0.tar.gz
```

本辞書は文字コードが Whale のシステムと対応していないため文字コードを変換したのちにインストールを行います。なお変換スクリプトは whale/util にありますのでご利用ください。変換スクリプトファイルの作成方法は、付録をご参照ください。

```
$ tar xzf ipadic-2.7.0.tar.gz
$ cd ipadic-2.7.0
$ ./configure
$ make
$ make install
$ cp /whale/util/ipadic_conv_utf8.sh ./
$ chmod +x ipadic_conv_utf8.sh
$ ./ipadic_conv_utf8.sh
$ `chasen-config --mkchadic`/makemat -i w
$ `chasen-config --mkchadic`/makeda -i w chadic *.dic
$ make install
$ cd /usr/local/etc/
$ sh -c "iconv -f euc-jp -t utf-8 chasenrc > temp"
$ mv temp chasenrc
```

(10) rpmforge (ffmpeg のインストールに必要なリポジトリ)

OS のバージョンや CPU の bit 数によってインストールするリポジトリに違いがあるのでそのパラメータを取得してダウンロードする。手動でダウンロードする場合は確認してください。また、毎回このリポジトリにアクセスすると yum の読み込みが遅くなるので自動に適用されないようにしたほうがよい。

```
if ( `cat /etc/redhat-release | grep -c "3¥."` ) then
set var = 3;
else if ( `cat /etc/redhat-release | grep -c "4¥."` ) then
set var = 4;
else if ( `cat /etc/redhat-release | grep -c "5¥."` ) then
set var = 5;
else if ( `cat /etc/redhat-release | grep -c "6¥."` ) then
set var = 6;
else if ( `cat /etc/redhat-release | grep -c "7¥."` ) then
```

```
set var = 7;
else
echo "OS 対応外です"
exit
endif

if ( `uname -a | grep -c "x86_64"` ) then
set bit = "x86_64";
else if ( `uname -a | grep -c "amd64"` ) then
set bit = "x86_64";
else if ( `uname -a | grep -c "i386"` ) then
  if ( $var > 5 ) then
    set bit = "i686";
  else
    set bit = "i386";
  endif
else
echo "バージョン外です";
exit
endif

set url = "http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el${var}.rf.${bit}.rpm";

wget $url;

set trans = "rpmforge-release-0.5.3-1.el${var}.rf.${bit}.rpm";

rpm -ivh $trans
yum -y update rpmforge-release
sed -e "s/enabled = 1/enabled = 0/" /etc/yum.repos.d/rpmforge.repo > new.repo
mv new.repo /etc/yum.repos.d/rpmforge.repo
```

(11) ffmpeg (動画と音声の変換ソフト)

ffmpeg をインストールします。

```
$ yum install -y --enablerepo=rpmforge ffmpeg
```

(12)ffmpeg-php (PHP で ffmpeg を動かす)

実際には必要ないと思われるが念のためインストールする。frame のエラーが発生したら



ffmpeg\_frame\_error.sh を、movie のエラーが発生したら ffmpeg\_movie\_error.sh を実行すればインストールできます。

```
$ yum -y install php-devel re2c
$ yum -y install --enablerepo=rpmforge ffmpeg-libpostproc ffmpeg-devel
$ wget http://downloads.sourceforge.net/project/ffmpeg-php/ffmpeg-php/0.6.0/ffmpeg-php-0.6.0.tbz2
$ tar jxf ffmpeg-php-0.6.0.tbz2
$ cd ffmpeg-php-0.6.0
$ phpize
$ ./configure
$ make
$ make install
$ sed -e "5 a ; Enable ffmpeg extension module ¥n extension=ffmpeg.so" /etc/php.ini > new.ini
$ mv new.ini /etc/php.ini
```

#### (13)sox のインストール

英語認識に使う sox をインストールします。主に wav ファイルにハイパスフィルタをかけるのに使います。

```
yum install -y sox
```

#### (14)kaldi

英語認識に使います。とても説明が長くなるので省略します。  
必要に応じて、/whale/core/test3.sh 内を書き換えてください。

### 4.3. 設定

Whale の設定を行います。Whale 全体の設定ファイルの変更と core および web ディレクトリにパーミッションやシンボリックリンク等の設定を行います。なお、ここからは whale の配置を /whale としていることを前提としていますので、そのほかの場所に配置した場合は、コマンドのファイルパス等を適切に変えて進めてください。

#### 4.3.1. Whale の設定

Whale の設定を行います。

whale.conf の設定（場所は /whale/whale.conf）

設定名	設定例	説明
coredir	/whale/core	core ディレクトリのパス
webdir	/whale/web	web ディレクトリのパス
videodir	/whale/video	video ディレクトリのパス
workdirname	workspace	音声認識結果保存ディレクトリ名
Db	whale	システムのデータベース名
dbuser	whale	データベースのユーザ名
dbpass	whaleadmin	データベースのパスワード

#### 4.3.2. Core の設定

Whale の core ディレクトリの各ファイルやディレクトリにパーミッションを設定します。**Core と Web の設定は自動設定スクリプト config\_new.sh を用意してありますのでご利用ください。**

```
$ cd /whale/util/  
$ chmod +x config_new.sh  
$ ./config_new.sh
```

```
$ cd /whale/core  
chmod +x *.pl  
chmod +x wavcut  
chmod +x wav_timedata  
chmod +x td  
chmod 777 workspace
```

#### 4.3.3. Web の設定

Whale の web ディレクトリ内の各ファイルやディレクトリにパーミッションを設定します。また、core ディレクトリの workspace ディレクトリのシンボリックリンク等を web に作成します。**4.3.2 で自動設定スクリプトを利用した場合は、以下の設定もされていますので飛ばしてください。**

```
$ cd /whale/web  
cd /whale/web  
chmod +x *.pl
```

```
chmod +x *.csh
chmod +x hb4/hb4
chmod +x hb4_tfidf_norm/hb4_tfidf_norm
chmod +x tf_idf_hb4_search/tf_idf_hb4_search
chmod +x scene/*
chmod +x add_vinfo_ba_all/add_vinfo_ba_all
chmod +x kensaku/kensaku30_linear.csh
chmod +x src/*
chmod +x src/tf_idf_hb4_kensaku_NTCIR10db_linear/kensaku
chmod 777 temporary
ln -s /whale/video/ video
chmod 777 video
ln -s /whale/core/workspace/ workspace
cd /whale
chown -R apache .
```

#### 4.4. その他のソフトウェアの設定

その他のソフトウェアに関して設定を行います。バージョンによって、設定項目がなくなったり変更されたりしている場合があります。これらの設定は、自動設定スクリプトには含まれてはいませんので手動で設定を行ってください。

##### 4.4.1. PHP の設定

php.ini の設定（場所は/etc/php.ini）

該当する以下の部分を修正してください。ファイルのバックアップを取ってから設定することをお勧めします。また、以下の設定は最低限の値となっていますので、使用する環境に合わせて数値を各自設定してください

修正前	修正後
max_execution_time = 30	max_execution_time = 300
memory_limit = 128MB	memory_limit = 2048MB
post_max_size = 8MB	post_max_size = 2048MB
upload_max_filesize = 2M	upload_max_filesize = 2048M

##### 4.4.2. MySQL の設定

```
$ mysql -p
```

ここから MySQL のコマンド操作となります。以下のコマンドを記述して新しいユーザ（今回は whale とする）を追加してください。またパスワード（今回は whaleadmin とする）も設定してください。CentOS7 使用していて、うまくいかない場合には、

```
yum install mariadb mariadb-server mariadb-devel
/etc/my.cnf.d/server.cnf の[mysqld]の後に
```

```
character-set-server=utf8
skip-character-set-client-handshak
```

を追加

/etc/my.cnf.d/client.cnf の[client-mariadb]の後に

```
default-character-set = utf8
```

を追加

```
systemctl enable mariadb.service
systemctl start mariadb.service
```

```
mysql_secure_installation
```

を上記コマンドの前に実行すると環境が整います。

```
mysql> grant all ON *.* to whale@"localhost" identified by "whleadmin";
mysql> flush privileges;
mysql> quit
```

追加したユーザでログインしなおします。

```
$ mysql -u whale -p
```

ログインできればユーザ登録が成功しています。次に MySQL のコマンドから whale 専用のデータベースの作成とテーブルを追加します。スクリプトにすべて記述してあるので、スクリプトを読み込んでください。また、初期の whale 管理ユーザが自動的に追加されます。(user:whale, pass:whleadmin)

```
mysql> source /whale/util/whale_database.sql
mysql> quit
```

#### 4.4.3. Perl の設定

システムの動作に必要なモジュールを追加します。

```
$ perl -MCPAN -e shell
```

初めて起動した場合は設定画面が出ますが、最初の項目で自動設定にするか聞かれるので、適当に Enter を打っていれば設定が終わるはずです。設定後 CPAN のコマンドラインになるので、モジュールをインストールします。CPAN のコマンドラインにならなかった場合は、上記のコマンドをもう一度入力して CPAN コマンドラインを起動させてください。

CentOS7 使用していて、うまくいかない場合には、

```
yum install libyaml-devel
yum install openssl-devel zlib-devel readline-devel
yum install perl-Time-HiRes
yum -y install perl-CPAN
```

を上記コマンドの前に実行すると環境が整います。

```
cpan> install Config::Simple
cpan> exit
```

#### 4.4.4. Cron の設定

Cron と呼ばれるスクリプトを自動実行させるソフトウェアに Whale を設定します。以下のコマンドを入力して設定ファイルを開いてください。

```
$ crontab -e
```

何らかのエディタで開かれたはずですが、この設定ファイルに以下のコマンドを追記してください。

```
00 4 * * * perl /whale/core/cron_whale.pl > /dev/null 2>&1
```

上記の設定では、毎朝 4 時に Whale が自動的に未解析のビデオファイルを解析するように設定されます。時間を変えたい場合は cron の設定方法に従って設定を書き換えてください。

#### 4.4.5. httpd の設定

CentOS7 使用していて、うまくいかない場合には、

```
yum install -y httpd
```

```
systemctl start httpd.service
```

```
systemctl enable httpd.service
```

を試してください。

設定ファイル例(httpd.conf)を適宜、参考にしてください。

```
mv -i /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.ori
```

```
cp /whale/httpd.conf /etc/httpd/conf/httpd.conf
```

```
systemctl restart httpd.service
```

それでもうまくいかない場合には、firewalld も必要に応じて disable にしてみます。

```
systemctl stop firewalld
```

```
systemctl disable firewalld.service
```

さらに SELinux の無効化も試してみてください。

```
setenforce 0
```

恒久的に無効化するためには/etc/sysconfig/selinux ファイルの SELINUX の値を変更します。値を変更後、サーバ再起動により反映されます。

```
SELINUX=disabled
```

## 5. アンインストール

Whale のアンインストールを行う場合は、以下の手順に沿って操作してください。

1. /whale ディレクトリを削除
2. MySQL の whale データベースおよび whale ユーザを削除
3. Whale システム用に cron に登録した記述を削除
4. Whale インストール時にインストールした各種パッケージをアンインストール（不要な場合）

## 6. Q&A

Q1 : Whale のインストールに失敗する。
A1 : 次の 2 点の問題が考えられます。①インストールの手順がおかしい可能性があります。②Whale 以外にインストールしたソフトウェアのバージョンが適切ではない可能性があります。
Q2 : Ubuntu 以外のディストリビューションで使いたい。
A2 : Ubuntu 独自で動作するものは使ってはいないはずなので、おそらくほかのディストリビューションで運用することも可能です。しかし、本ドキュメントでは Ubuntu にインストールする方法しか説明してありませんので、そのディストリビューションにあったインストール方法にコマンドを置き換えて、インストールを行ってください。
Q3 : Whale って何？くじらですか？
A3 : くじらです。本システムの作成者がなんとなく思いついた名前です。
Q4 : ./configure 時に no acceptable cc found in \$PATH と出る
yum install -y gcc を行ってください
Q5 : perl -MCPAN -e shell がうまく実行できない
yum install perl-CPAN を実行してください

## 7. 付録

細かいスクリプトの内容はここに記載しておきますので、何らかの形でうまくいかない方は参照してください。自分で作成する場合は適当なエディタを起動させて、以下のコードを記述して保存してください。

### 7.1. Whale ツリー構造

インストールに成功したらおそらくこのようなツリー構造になります。何か動作がおかしい場合などに御参考ください。

```
whale
|-- core
|   |-- all.pl
|   |-- chasen_jiritsu.pl
|   |-- chasen_stopword.txt
|   |-- check.txt
|   |-- cron_whale.pl
|   |-- csrc
|   |   |-- util
|   |   |   |-- td
|   |   |   |-- td.c
|   |   |-- vad
|   |   |   |-- readme.txt
|   |   |   |-- time.dat
|   |   |   |-- vad
|   |   |   |-- vad.c
|   |   |-- wavcut_new2
|   |   |   |-- wavcut
|   |   |   |-- wavcut.c
|   |-- cut
|   |-- cut_scene.pl
|   |-- cut_triphone.pl
|   |-- dat
|   |   |-- asr
|   |   |-- asr.log
|   |-- database.pl
|   |-- database2.pl
|   |-- en2jp.php
|   |-- endian.pl
|   |-- engASR.pl
|   |-- get_sen_en.pl
|   |-- get_sentence.pl
|   |-- get_triphone.pl
|   |-- hear_file
|   |-- julius.pl
|   |-- julius_config
|   |   |-- CSJ-APS
|   |   |   |-- hmmdefs
|   |   |   |-- hmmdefs.gz
|   |   |-- kyoto-gt1_19k.dic
|   |   |-- kyoto_kouen.bingram
|   |   |-- logicalTri.add
|   |-- src
|   |   |-- blockr.pl
|   |   |-- get_kana_NTCIR9.pl
```





```
| | |-- upload.css  
| | |-- video.css  
|-- datelist.php  
|-- eva_inter-new.csh  
|-- eva_inter.csh  
|-- get_scene-new.pl  
|-- get_scene.pl  
|-- hb4  
| | |-- 00readme  
| | |-- compile  
| | |-- hash.c  
| | |-- hash.h  
| | |-- hb4  
| | |-- hb4.cpp  
| | |-- r  
| | | |-- hb4.out_col  
| | | |-- hb4.out_row  
| | | |-- hb4.outf  
| | | |-- hb4.term  
|-- hb4_tfidf_norm  
| | |-- 00readme  
| | |-- compile  
| | |-- hb4_tfidf_norm  
| | |-- hb4_tfidf_norm.cpp  
| | |-- in3.txt  
| | |-- r  
| | | |-- hb4.out_col  
| | | |-- hb4.out_row  
| | | |-- hb4.outf  
| | | |-- hb4.term  
| | | |-- hb4.tfidf  
| | | |-- hb4.tfidf_norm  
| | | |-- tfidf.mat  
| | | |-- tfidf.txt  
| | | |-- tfidf_norm.xlsx  
|-- index.php  
|-- js  
| | |-- index.js  
| | |-- jqm_custom.js  
| | |-- jquery-1.7.1.min.js  
| | |-- jquery.cookie.js  
| | |-- jquery.mobile-1.1.1.js  
| | |-- jquery.mobile-1.1.1.min.js  
| | |-- upload.js  
| | |-- video.js  
|-- kensaku  
| | |-- kensaku30_linear.csh  
|-- kensaku.pl  
|-- kujira.ico  
|-- kujira_apple.png  
|-- login.php  
|-- mysql_manager.php  
|-- rank-new.pl  
|-- rank.pl  
|-- register.php  
|-- registered.php  
|-- scene
```

```
| | |-- all30_1best.csh  
| | |-- all30_1best.csh.off  
| | |-- all30_1best_linear-new.csh  
| | `-- all30_1best_linear.csh  
|-- scene_search-new.php  
|-- scene_search.php  
|-- scene_search.pl  
|-- scene_table.pl  
|-- search.php  
|-- search.pl  
|-- src  
| |-- add_vinfo_ba_all -> ../add_vinfo_ba_all/add_vinfo_ba_all  
| |-- convse_stdio_no.pl  
| `-- tf_idf_hb4_kensaku_NTCIR10db_linear  
| | |-- Readme  
| | |-- cdp.c  
| | |-- cdp.h  
| | |-- get_token.c  
| | |-- get_token.h  
| | |-- kana.txt  
| | |-- kana2rom.pl  
| | |-- kensaku  
| | |-- key2kana  
| | |-- key2n.txt  
| | |-- key2tri  
| | |-- main.c  
| | |-- memo_kensaku.txt  
| | |-- net.txt  
| | |-- rom.txt  
| | |-- rom2tri  
| | |-- search_db.c  
| | |-- search_db.h  
| | |-- sh  
| | | |-- kana2rom.pl  
| | | `-- rom2tri  
| | |-- sub_hash.c  
| | |-- sub_hash.h  
| | |-- tf_idf.c  
| | |-- tf_idf.h  
| | |-- topic.tri  
| | `-- tri.txt  
|-- table_block.pl  
|-- temporary  
|-- tf_idf_hb4_search  
| |-- cdp.c  
| |-- cdp.h  
| |-- get_token.c  
| |-- get_token.h  
| |-- kana.txt  
| |-- key2n.txt  
| |-- key2tri  
| |-- main.c  
| |-- net.txt  
| |-- rom.txt  
| |-- sh  
| | |-- kana2rom.pl  
| | `-- rom2tri
```



```
create table teacher(  
id integer not null primary key auto_increment,  
name text,  
identifier text  
);  
  
create table user(  
id integer not null primary key auto_increment,  
name text not null,  
pass text,  
admin int not null  
);  
  
create table video(  
id integer not null primary key auto_increment,  
name text ,  
class_id integer,  
date text,  
recognition integer not null,  
english integer not null  
);
```

・ 7.4. ffmpeg-php インストールのエラー修正

ffmpeg\_frame\_error.pl

```
#!/bin/csh -f  
cd /usr/local/src/ffmpeg-php-0.6.0  
sed -e "s/RGBA32/RGB32/" ffmpeg_frame.c > new.c  
mv new.c ffmpeg_frame.c  
make  
make install
```

ffmpeg\_movie\_error.pl

```
#!/bin/csh -f  
cd /usr/local/src/ffmpeg-php-0.6.0  
sed -e "s/list_entry ¥*le;/zend_rsrc_list_entry ¥*le;/" ffmpeg_movie.c > new.1  
sed -e "s/list_entry new_le;/zend_rsrc_list_entry new_le;/" new.1 > new.2  
sed -e "sizeof(list_entry)/sizeof(zend_rsrc_list_entry)/" new.2 > new.3  
mv new.3 ffmpeg_movie.c  
make  
make install
```