

### 3 発注側サーバ環境設定

以下に本システムにおける発注側サーバの環境設定手順を示す。

以下の手順は、基本的な UNIX コマンドを熟知しているものとして記述してある。

#### 3.1 前提条件

本章の環境設定を実施する前に、ハードウェアの設定、OS (Vine Linux 1.1) の設定及び、ネットワークの設定 (含むインターネットの接続設定) が実施されていること。

### 3.2 概要

主にモジュールをインストール及びセットアップすることにより、発注側サーバの環境を設定する。

(A) Apache Ver 1.3.9

WWWサーバモジュール

(B) OpenSSL Ver 0.9.4

WWWサーバ暗号化モジュール

(C) mod\_ssl Ver 2.4.2

(B)で使用されているライブラリモジュール

(D) Apache JServ Ver 1.0

WWWサーバでJava Servletプログラムを稼働させるためのモジュール

(E) JDK (Java Development Kit) Ver 1.1.7

Java言語モジュール

(F) JSJK (Java Servlet Development Kit) Ver 2.0

Java Servlet言語ライブラリモジュール

(G) Java Project X TR 2

Java用XML構造チェックモジュール

(H) cos.jar

Java Servlet言語ライブラリモジュール

### 3.3 環境設定手順

環境設定の前に4.1項『SWG7.3発注サーバ』のインストールを実施して下さい。以下にWWWサーバであるApacheの設定手順を示す。尚、ApacheにはDSO機能及びmod\_rewrite(URL Rewriting Engine)も同時に組み込む。

#### 3.3.1 Apacheのインストール

(1) 既存 Apache を停止する。

```
# /etc/rc.d/init.d/httpd stop
```

(2) インストールファイルを解凍する。(今回は/usr/local/src に解凍する例をしめす)

```
% cd /usr/local/src
```

```
% cp /copysrcpath/apache_1.3.9.tar.gz ./
```

( copysrcpath はコピー元のパス )

```
% gzip -cd apache_1.3.9.tar.gz | tar xvf -
```

( 解凍後 apache\_1.3.9 ディレクトリが作成される )

```
% cd apache_1.3.9
```

(3) Apache の構成を設定する。(今回は/usr/local/apache にインストールする例をしめす)

```
% ./configure --prefix=/usr/local/apache --enable-rule=SHARED_CORE --enable-module=so --enable-module=rewrite --with-perl=/usr/bin
```

注1：別のディレクトリにパスが通っていることがあるので./を必ず指定する

注2：--enable で始まる部分はDSO及びmod\_rewrite組み込みの為の指定

注3：今回は/usr/binの下にperlファイルが存在した

(4) モジュールを組み込む。

(3)でsrcディレクトリにConfiguration.apaciが作成されるのでこれをコピーしてモジュールを組み込む。

```
% cd src
```

```
% cp Configuration.apaci Configuration
```

( 5 ) 組み込んだモジュールをソースに反映させる。( src ディレクトリで実行する )

```
% ./Configure
```

注：別のディレクトリにパスが通っていることがあるので./を必ず指定する

( 6 ) コンパイルする。( apache\_1.3.9 ディレクトリで実行する )

```
% cd ..
```

( apache\_1.3.9 ディレクトリへ移動 )

```
% make clean
```

```
% make
```

( 7 ) インストールする。( root 権限で実施する )

```
( 8 ) % su
```

```
# make install
```

( 9 ) 動作テストを実施する。( root 権限で起動する )

( A ) 以下のコマンドで Apache を起動する。

```
# /usr/local/apache/bin/apachectl start
```

注 2 : /usr/local/apache は(2)で指定したディレクトリ

注 3 : bin ディレクトリは(2)でディレクトリ構成の確認をしたときの sbindir のパス

( B ) ブラウザで初期画面が表示されることを確認する。

任意のブラウザを起動する。

ブラウザ上の http コマンドラインより以下のコマンドを実行する。

http://mhi170.ctc.ne.jp ( 発注側サーバに設定した URL 名を入力する。 )

ブラウザ上に Apache の初期画面が表示されることを確認する。

確認できれば以下のコマンドで Apache を停止する。

```
# /usr/local/apache/bin/apachectl stop
```

### 3.3.2 OpenSSLのインストール

( 1 ) インストールファイルを解凍する。

```
( 今回は/usr/local/src/Apache-SSL に解凍する )  
% cd /usr/local/src/Apache-SSL  
% cp /copysrcpath/openssl-0.9.4.tar.gz ./  
( copysrcpath はコピー元のパス )  
% gzip -cd openssl-0.9.4.tar.gz | tar xvf -  
( 解凍後 openssl-0.9.4 ディレクトリが作成される )  
% cd openssl-0.9.4
```

( 2 ) openssl の構成を設定する。( 今回はデフォルトにインストールする )

```
% ./config
```

インストール先を指定する場合は以下の通り

```
% ./config --prefix=installpath --openssldir=meltpath
```

注 1 : 別のディレクトリにパスが通っていることがあるので./を必ず指定する

注 2 : *installpath*(インストール先パス)、*meltpath*((1)で解凍したパス)

( 3 ) コンパイルする。

```
% make  
( 注 ) 非常に時間がかかる  
% make test
```

( 4 ) インストールする。( root 権限で実施する )

```
% su  
# make install
```

### 3.3.3 mod\_sslのインストール

( 1 ) インストールファイルを解凍する。

```
( 今回は/usr/local/src/Apache-SSL に解凍する )  
% cd /usr/local/src/Apache-SSL  
% cp /copysrcpath/mod_ssl-2.4.2-1.3.9.tar.gz ./  
( copysrcpath はコピー元のパス )  
% gzip -cd mod_ssl-2.4.2-1.3.9.tar.gz | tar xvf -  
( 解凍後 mod_ssl-2.4.2-1.3.9 ディレクトリが作成される )  
% cd mod_ssl-2.4.2-1.3.9
```

( 2 ) apache の構成を設定する。

```
% ./configure --with-apache=../apache_1.3.9 --with-ssl=../openssl-0.9.4  
--prefix=/usr/local/apache --enable-shared=ssl  
注 1 : 別のディレクトリにパスが通っていることがあるので./を必ず指定する  
注 2 : --with-apache には apache のソースディレクトリを指定  
( 今回は../apache_1.3.9 )  
--with-ssl には openssl のソースディレクトリを指定  
( 今回は../openssl-0.9.4 )  
--prefix には apache インストールディレクトリを指定  
( 今回は/usr/local/apache )
```

( 3 ) apache の再コンパイルをする。( apache の build ディレクトリで行う )

```
% cd ../apache_1.3.9  
% make
```

( 4 ) インストールする。( root 権限で実施する )

```
% su  
# make install
```

### 3.3.4 JDKのインストール

(1) インストールファイルを解凍する。(今回は/usr/local/java に解凍する)

```
% cd /usr/local/java
```

```
% cp /copysrcpath/jdk_1.1.7-v3-glibc-x86.tar.gz ./
```

(*copysrcpath* はコピー元のパス)

```
% gzip -cd jdk_1.1.7-v3-glibc-x86.tar.gz | tar xvf -
```

(解凍後 jdk117\_v3 ディレクトリが作成される)

### 3.3.5 JSDKのインストール

(1) インストールファイルを解凍する。

(今回は/usr/local/java に解凍する)

```
% cd /usr/local/java
```

```
% cp /copysrcpath/jsdk20-solaris2-sparc.tar.Z ./
```

(*copysrcpath* はコピー元のパス)

```
% tar xfvz jsdk20-solaris2-sparc.tar.Z
```

(解凍後 JSDK2.0 ディレクトリが作成される)

### 3.3.6 Apache JServのインストール

(1) インストールファイルを解凍する。

(今回は/usr/local/java に解凍する)

```
% cd /usr/local/java
```

```
% cp /copysrcpath/Apache_JServ_1.0.tar.gz ./
```

(*copysrcpath* はコピー元のパス)

```
% gzip -cd Apache_JServ_1.0.tar.gz | tar xvf -
```

(解凍後 ApacheJServ-1.0 ディレクトリが作成される)

```
% cd ApacheJServ-1.0
```

(2) 英語版 Perl をインストールする。

(英語版でないとコンパイルが通らないため)

```
% rpm -e --nodeps perl
```

(日本語版のアンインストール)

注1 : 「`を削除できません。`」というメッセージが表示されるが問題はない。

```
% cd downloaddir
```

(*downloaddir* は英語版 Perl をダウンロードしたディレクトリ)

```
% rpm -i perl-5.004m7-1.i386.rpm
```

(英語版のインストール)

( 3 ) Apache JServ の構成を設定する。

```
% ./configure --with-apache-install=  
/usr/local/apache --prefix=/usr/local/java/Apache-JServ  
--with-jdk-home=/usr/local/java/jdk117_v3  
--with-jsdk=/usr/local/java/JSDK2.0/lib/jsdk.jar --enable-compressed  
-jar --enable-debugging
```

注 1 : 別のディレクトリにパスが通っていることがあるので ./ を必ず指定する

注 2 : --with-apache-install には apache のインストールディレクトリを指定  
(今回は /usr/local/apache)

--prefix には Apache Jserv のインストールディレクトリを指定  
(今回は /usr/local/java/Apache-Jserv)

--with-jdk-home には jdk の解凍ディレクトリを指定  
(今回は /usr/local/java/jdk117\_v3)

--with-jsdk には jsdk のクラスパスを指定  
(今回は /usr/local/java/JSDK2.0/lib/jsdk.jar)

( 4 ) コンパイルする。

```
% make
```

( 5 ) インストールする。( root 権限で実施する )

```
% su  
# make install
```

( 6 ) Perl を日本語版に戻す。

```
% rpm -e --nodeps perl
```

(英語版のアンインストール)

注 1 : 「        を削除できません。」というメッセージが表示されるが問題はない。

Vine Linux1.1CR の CD-ROM1 を入れる。

```
% mount /dev/cdrom /mnt/cdrom
```

```
% cd /mnt/cdrom/Vine/RPMS
```

```
% rpm -i perl-5.004_jp-3.i386.rpm
```

(日本語版のインストール)

```
% cd /
```

```
% umount /mnt/cdrom
```

### 3.3.7 Java ProjectX TR2のインストール

(1) インストールファイルを解凍する。

(今回は/usr/local/java に解凍する)

```
% cd /usr/local/java
```

```
% cp /copysrcpath/xml-tr2.zip ./
```

(*copysrcpath* はコピー元のパス)

```
% unzip xml-tr2.zip
```

(解凍後 xml-tr2 ディレクトリが作成される)

### 3.3.8 cos.jarのインストール

(1) インストールファイルを解凍する。

(今回は/usr/local/java に解凍します)

```
% cd /usr/local/java
```

```
% mkdir cos
```

```
% cd cos
```

```
% cp /copysrcpath/cos.zip ./
```

(copysrcpath はコピー元のパス)

```
% unzip cos.zip
```

### 3.3.9 コンフィグファイルの設定

( 1 ) Apache の httpd.conf ファイルを設定する。

以下に UNIX 標準エディタの vi にて編集する例を示す。

```
% cd /usr/local/apache/conf
```

```
% su
```

```
# vi httpd.conf
```

設定内容については添付資料 1 を参照

注 1 : 設定ファイルは httpd.conf と srm.conf と access.conf の 3 つあるがデフォルトでは httpd.conf に srm.conf と access.conf の内容が含まれているので httpd.conf のみを変更する。

( 2 ) Apache Jserv の httpd.conf ファイルを設定する。

以下に UNIX 標準エディタの vi にて編集する例を示す。

```
% cd /usr/local/java/ApacheJServ-1.0/conf
```

```
% su
```

```
# vi httpd.conf
```

設定内容については添付資料 2 を参照

( 3 ) Apache Jserv の jserv.properties ファイルを設定する。

以下に UNIX 標準エディタの vi にて編集する例を示す。

```
% cd /usr/local/java/ApacheJServ-1.0/conf
```

```
% su
```

```
# vi jserv.properties
```

設定内容については添付資料 3 を参照

( 4 ) Apache Jserv の zone.properties ファイルを設定する。

以下に UNIX 標準エディタの vi にて編集する例を示す。

```
% cd /usr/local/java/ApacheJServ-1.0/conf
% su
# cp zone.properties share_zone.properties
# vi share_zone.properties
# cp zone.properties secure_zone.properties
# vi secure_zone.properties
# cp zone.properties system_zone.properties
# vi system_zone.properties
```

設定内容については添付資料 4 を参照

注 1 : zone.properties ファイルは 3 ファイル作成する。(share 用、secure 用、system 用)

(それぞれの servlet を独立させ、セキュリティを確保するため)

### 3.3.10 サイト秘密キーの作成

( 1 ) キーを置くディレクトリを作成する。

( 今回は /usr/local/apache/etc/ssl/keys を作成する )

```
% su
```

```
# mkdir -p /usr/local/apache/etc/ssl/keys
```

```
# chmod 0700 /usr/local/apache/etc/ssl/keys
```

```
# cd /usr/local/apache/etc/ssl/keys
```

ファイルを 5 つ用意する。

( 適当なファイルを 5 つ /usr/local/apache/etc/ssl/keys へコピーする )

```
# cp file ./
```

秘密キーを作成する。( 今回は f1、f2、f3、f4、f5 の 5 ファイルを用意した )

```
# /usr/local/ssl/bin/openssl genrsa -des3 -rand f1:f2:f3:f4:f5 1024 > httpd.key
```

パスフレーズをきいてくるのでパスフレーズを入力する ( 確認のため 2 回 )

注 : /usr/local/ssl/bin は openssl をインストールしたディレクトリ

( 2 ) 作成されたキー httpd.key の権限を変更する。

```
# chmod 0400 httpd.key
```

### 3.3.1.1 サイト証明書の作成

( 1 ) 証明書を置くディレクトリを作成する。

( 今回は /usr/local/apache/etc/ssl/certs を作成する )

```
% su
# mkdir /usr/local/apache/etc/ssl/certs
# chmod 0700 /usr/local/apache/etc/ssl/certs
# cd /usr/local/apache/etc/ssl/certs
```

( 2 ) 証明書を作成する。

```
# /usr/local/ssl/bin/openssl req -new -x509 -days 365 -
key ../keys/httpd.key > httpd.pem
```

パスフレーズをきいてくるので秘密キーを作成したときのパスフレーズを入力する。

その他に国、都道府県、市、組織、部署、名前、メールアドレスを入力する。

注 : /usr/local/ssl/bin は openssl をインストールしたディレクトリ

注 : 『名前』は、サーバ名 (ドメイン名含む)

( 3 ) 作成された証明書 httpd.pem の権限を変更する。

```
# chmod 0400 httpd.pem
```

### 3.3.12 Apache 起動時のパスフレーズ入力の自動化

デフォルトでは Apache の起動時にパスフレーズを要求してくるのでパスフレーズをプログラムで入力するようにする。

(1) プログラムを作成する。(今回は/usr/local/apache/bin/pp-filter)

プログラム(シェル)の内容は以下の通り。

```
#!/usr/local/apache/bin/pp-filter
```

```
#!/bin/sh
```

```
echo パスフレーズ
```

注：このプログラムは root 権限でのみ実行できるようにしておく。

### 3.3.13 boot時の自動起動の設定

( 1 ) Vine Linux にインストールされていた Apache を削除する。

```
# cd /usr/sbin  
# rm httpd
```

( 2 ) 新しくインストールした Apache をコピーする。

```
# cp /usr/local/apache/bin/httpd ./  
注 : /usr/local/apache は apache インストールディレクトリ
```

### 3 . 3 . 1 4 ベーシック認証のパスワードファイルの作成

( 1 ) パスワードファイルを作成する。

```
#mkdir /usr/local/apache/auth
```

ディレクトリーの作成

```
% cd /usr/local/apache/auth
```

```
% su
```

```
# htpasswd -c authuser_system ユーザ名
```

ユーザのパスワードを聞いてくるのでパスワードを入力する。

authuser ファイルが作成される。

注 1 : -c はファイルを新規作成するのでユーザの追加のときは-c をとる。

### 3.3.15 Apacheで必要なディレクトリの作成

(1) /usr/local/apache/var/run ディレクトリを作成する。

```
% cd /usr/local/apache  
% mkdir var  
% cd var  
% mkdir run
```

(2) その他必要なディレクトリを作成する。

ディレクトリ構造については添付資料5を参照。

### 3.3.16 Jserv動作チェック

( 1 ) Apache を起動する。

```
% su
```

```
# /usr/local/apache/bin/apachectl start
```

エラーが出る場合はコンフィグファイル等の設定を確認する。

注 1 : Apache を起動すると自動で Apache-Jserv が起動する設定になっている。

( 2 ) Apache-Jserv が起動しているか確認する。

```
% lynx http://localhost/jserv/
```

Apache-JServ が稼動していれば、"ApacheJServ-1.0 Status" ページが表示される。

エラーが出る場合はコンフィグファイル等の設定を確認する。

( 3 ) Servlet が正常に動作するかを確認する。

ディレクトリ ( /usr/local/apache/logs ) の属性を、webuser が参照できるように変更する。

```
#cd /usr/local/apache/
```

```
#chmod 0755 logs
```

```
#chown root.webuser logs
```

Servlet 動作確認

```
% cd /usr/local/java/apacheJServ-1.0/example
```

```
% cp Hello.class /home/wg73/share/servlets/
```

ブラウザから、下記の URL を表示させる。

```
http://<hostname>/servlets/Hello
```

servlet が正常に動作している場合は

```
Example Apache JServ Servlet
Congratulations, Apache JServ is working!
```

の画面が表示される。