

Sapid インストールマニュアル

-RPM-

愛知県立大学情報科学部 山本研究室
近藤 司 山田 誠也

2013 年 12 月 27 日

1 はじめに

本マニュアルは Sapid を RPM でインストールする方法を説明する。
CentOS には、32bit と 64bit の 2 つあるため、その両方でインストールを行った。
以下の表がインストールした時の環境である。

ソフトウェア	バージョン	
	CentOS 6.5 (32bit)	CentOS 6.5 (64bit)
OS	CentOS 6.5 (32bit)	CentOS 6.5 (64bit)
gcc	4.4.7-4	4.4.7-4
g++	4.4.7-4	4.4.7-4
flex	2.5.35	2.5.35
bison	2.4.1	2.4.1
java	1.7.0.45	1.7.0.45
ant	1.7.1	1.7.1
tk	8.5.7-5	8.5.7-5
imake	1.02-11	1.02-11
libXt	1.1.3-1	1.1.3-1
libxml2	2.7.6-14	2.7.6-14

詳細な CentOS のバージョンは以下のコマンドで確認できる。

CentOS 32bit

```
# uname -a
Linux centos6.5 2.6.32-431.el6.i686
#1 SMP Fri Nov 22 00:26:36 UTC 2013 i686 i686 i386 GNU/Linux
```

CentOS 64bit

```
# uname -a
Linux centOS 2.6.32-431.el6.x86_64
#1 SMP Fri Nov 22 03:15:09 UTC 2013 x86_64 x86_64 x86_64 GNU/Linux
```

各環境をインストールするに当たって、root アカウントになる必要がある。root アカウントになる方法は、以下のコマンドをする必要がある。

root になるコマンド

```
# su -
```

今後も root 権限が必要なところが多々あるので、テストまでの作業はすべて root アカウントで行ってください。

1.1 yum のアップデート

今回は、yum というパッケージ管理システムを使ってインストールを行う。初めに、yum のアップデートを行わなければならないので、以下のコマンドでアップデートを行う。

yum のアップデート

```
# yum update
```

アップデートが終わったら実際に各環境をインストールしていく。

2 各環境のインストール方法

2.1 CentOS 32bit で環境構築

gcc のインストール

```
# yum -y install gcc
```

ここで、コマンドの-y はオプションで、すべてが yes になります。また、install を search に変換すると指定したソフトがあるかどうか探してくれます。

g++ のインストール

```
# yum -y install gcc-c++
```

flex のインストール

```
# yum -y install flex
```

bison のインストール

```
# yum -y install bison
```

java のインストール

```
# yum -y install java-1.7.0-openjdk-devel.i686
```

ant のインストール

```
# yum -y install ant.i686
```

tk のインストール

```
# yum -y install tk-devel.i686
```

imake のインストール

```
# yum -y install imake.i686
```

libxt-dev のインストール

```
# yum -y install libXt-devel.i686
```

libxml2-dev のインストール

```
# yum -y install libxml2-devel.i686
```

2.2 CentOS 64bit で環境構築

gcc のインストール

```
# yum -y install gcc
```

g++ のインストール

```
# yum -y install gcc-c++
```

flex のインストール

```
# yum -y install flex
```

bison のインストール

```
# yum -y install bison
```

java のインストール

```
# yum -y install java-1.7.0-openjdk-devel.x86_64
```

ant のインストール

```
# yum -y install ant.x86_64
```

tk のインストール

```
# yum -y install tk-devel.x86_64
```

imake のインストール

```
# yum -y install imake.x86_64
```

libxt-dev のインストール

```
# yum -y install libXt-devel.x86_64
```

libxml2-dev のインストール

```
# yum -y install libxml2-devel.x86_64
```

2.3 プロキシ設定方法

余談だが、もしプロキシを使っている環境でインストールを行っている場合インストールする前に以下のコマンドを端末上に入力する。

プロキシ設定

```
# export http_proxy="http://(プロキシサーバ):(ポート番号)"
```

3 Sapid のインストール

Sapid のソースを以下のサイトから、ダウンロードしてくる。

Sapid ソース

```
http://www.sapid.org/FTP-CURRENT/
```

ダウンロードするのは、最新版をインストールするようにする。今回インストールしたバージョンは、上記にあるように Sapid-6.100 です。

その後、ダウンロードが完了したら以下のコマンドで、Sapid を解凍をする。

Sapid の解凍

```
# tar -zxvf Sapid-6.100.tar.gz
```

3.1 JAVA_HOME の環境設定

~/.bashrc に以下のコードを追加する。

JAVA_HOME の設定

```
JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.45
export JAVA_HOME

PATH=$JAVA_HOME/bin:$PATH:$HOME/bin
export PATH
```

その後、以下のコマンドを打ち込み、ファイルの更新をする。

.bashrc の更新

```
# source ~/.bashrc
```

3.2 インストール

解凍した Sapid のフォルダに移動して、以下のコマンドを打ち込む。

————— MAKEFILE の作成 —————

```
# xmkmf -a 2>&1 | tee LOG-XMKMF
```

————— make コンパイル —————

```
# make 2>&1 | tee LOG-MAKE
```

————— インストール —————

```
# make install 2>&1 | tee LOG-INSTALL
```

これで、エラーが出なければインストールは成功しています。

3.3 Sapid のパス設定

テストを行う前に、Sapid のパスと環境を設定する必要があるため、`~/.bashrc` に以下のコードを追加し、環境設定を行う。

————— Sapid のパス設定 —————

```
if [ -f /usr/local/Sapid/lib/SetUp.sh ]; then
    . /usr/local/Sapid/lib/SetUp.sh
fi
```

その後、以下のコマンドを打ち込み、ファイルの更新をする。

————— .bashrc の更新 —————

```
# source ~/.bashrc
```

インストールした Sapid は、`~/usr/local/Sapid/`にあります。

4 Sapid の実行テスト

4.1 dhrystone-2.1(32bit)

32bit 環境では、以下のコマンドを実行しテストを行う。

————— コマンド —————

```
# cd /usr/local/Sapid/sample/test/dhrystone-2.1
# mkSapid -a 2>&1 | tee LOG_MKSAPID
# make test-all 2>&1 | tee LOG_DHRYSTONE
```

以下のメッセージでテストの成功が確認できる。

————— 成功結果 —————

```
Sapid: Test: dhry: Passed.
```

4.2 dhrystone-2.1(64bit)

64bit 環境では、dhrystone-2.1 のテストを 32bit 環境と同じコマンドで行うと、エラーが起こる。そのため、以下のコマンドでテストを行う。

————— コマンド —————

```
# cd /usr/local/Sapid/sample/test/dhrystone-2.1
# mkSapid -a 2>&1 | tee LOG_MKSAPID_D
# make DEFINE=-D__x86_64__ test-all 2>&1 | tee ~/LOG-TESTALL-DHRYSTONE
```

32bit 環境と同じく、以下のメッセージでテストの成功が確認できる。

————— 成功結果 —————

```
Sapid: Test: dhry: Passed.
```

4.3 java-zip

Java のテストは、32bit 環境 64bit 環境どちらも同じコマンドで動作します。そのため、以下のコマンドを実行しテストを行う。

————— コマンド —————

```
# cd /usr/local/Sapid/sample/test/java-zip
# mkSapid -a 2>&1 | tee LOG_MKSAPID
# make test-all 2>&1 | tee LOG_JAVA-ZIP
```

以下のメッセージでテストの成功が確認できる。

————— 成功結果 —————

```
Sapid: Test: java-zip: Passed.
```