

BYOD に対応したソフトウェア開発演習環境の提供

中鉢 欣秀

アプリケーションの開発を学ぶプログラミング演習を実施するためには学習者が利用する演習用の PC にプログラミング言語やライブラリ、関連するツール類等の必要な演習環境を準備する必要がある。本提案では BYOD 環境を前提とし、演習に必要なツール類一式を仮想環境である Vagrant を用いて提供してプログラミング演習で使用した実績について報告する。プログラミングの学習用途に仮想環境を用いたことの有用性や効果について考察し、今後の改善についても述べる。

When we teach application software development with programming, it is necessary to prepare the adequate programming environment including some programming languages, libraries, related tools and so on. This proposal is the way how to prepare those environment to the students with virtualization technologies using Vagrant for the BYOD condition. We consider the effect of using virtual environment to teach programming. Future work will be also mentioned.

1 はじめに

プログラミング演習においては、学習者が利用するためのプログラミング言語やライブラリ、関連するツールなどのソフトウェアを提供する必要がある。利用するもの一式が全て揃っていることに加え、各ソフトウェアのバージョンもしっかりと揃えて置く必要がある。これらが少しでも欠けると学生の演習に支障をきたし、講師は本来の目的とは外れた難解なトラブルシューティングに追われることになる。

大学におけるプログラミング演習において、学習者が利用する PC の環境を提供する方法にはいくつか考えられる。

例えば、PC 教室を利用できる場合は予め学生が使用する全てのマシンに必要なソフトウェア一式を含む

イメージを作成し、提供する方法が一般的である。この方法の場合、演習で必要になる全てのツール類を細かなバージョンの差異を見きだめて事前に確定させ、授業の開始前にイメージを準備する作業が必要である。もし、イメージの作成で間違いがあった場合、サイド作成して配布するための時間がかかり、授業の進行を遅らせることにもなる。

あるいは、演習用のツール類をサーバ上に構築しておき、学生にアカウントを提供し、SSH によりリモートログインを行わせる方法もある。この場合、サーバに全てのアカウントを作成したり、パスワードや認証用の公開鍵/非公開鍵などを適切に準備しておく必要がある。

また、演習用に一台のサーバを利用する方法では、ポート番号がかち合うため、学生が任意にサーバ（アプリケーションやデータベース等のデーモン）を立ち上げることができない。学生ごとにポート番号を変えることができれば可能ではあるが、手間が掛かる。サーバをどこに設置するか、あるいは、セキュリティ上の問題も懸念される。

以上見てきたとおり、プログラミング演習を実施する際に、学習者に共通の開発環境を提供する方法に

* Providing software development lecture for BYOD situation

This is an unrefereed paper. Copyrights belong to the Author(s).

Yoshihide Chubachi, 産業技術大学院大学情報アーキテクチャ専攻, Master Program of Information Systems Architecture, Advanced Institute of Industrial Technology.

は複数考えられる。本発表では、仮想環境の技術である Vagrant [2] を利用し学生のプログラミング演習環境を提供する手法について提案し、その有用性について議論する。近年事例が増えている BYOD (Bring Your Own Device) への適用例を紹介し、考察する。

2 提案手法

2.1 BYOD でのプログラミング演習

近年、学生も一人一台自分の PC を持つようになってきた。もちろん、全ての学生が PC を所持しているわけではないが、IT を学ぶ学生であれば所有している可能性は高いだろう。

プログラミング演習でも、PC 教室の PC ではなく自分の PC を利用したいという希望する者もいる。また、授業の課題を自宅でやりたいという要望があることも多く、学生が所有する PC を前提とした演習環境を構築することは学生のとってもメリットも有る。

また、本学の enPiT [4] で実施している夏合宿のように、宿泊施設の会議室などでアドホックにプログラミング演習をすることも有る。

自分のデバイスを自分で持ってくる、BYOD でのプログラミング演習を実施するには次の点を考慮すべきである。

1. Windows と Mac による OS の差異
2. 必要なソフトウェアが正しくインストールされていること

1. についての解となるのが仮想化技術の導入である。仮想化技術は Windows と Mac (あるいは Linux) の OS の差異を吸収するアブストラクションレイヤとして機能する。これを利用し、例えば Linux の一種である Ubuntu を仮想環境において動作させれば、ホスト OS の差異を問わず OS を共通にできる。

2. については、仮想環境にある OS の状態 (ソフトウェアのインストールや各種の設定) をパッケージとしてインポート/エクスポートする機能で対応する。

2.2 Vagrant を用いた仮想環境の提供

Vagrant [2] は HashiCorp 社が提供する仮想環境のツールである。インターネット上のクラウドに様々な OS のパッケージが登録されており、自由にダウン

```
Vagrant.configure("2") do |config|
  config.vm.box = "bento/ubuntu-18.04"
  config.vm.provision "shell",
    path: "provisioner.sh"
end
```

図 1 Vagrantfile ファイル

ロードして実行できる。また、既存の OS のパッケージを利用し、利用したいツール類を追加し、各種の設定を施した上で新しいパッケージとして配布することも可能だ。

Vagrant を利用するためには VirtualBox や VMware, AWS といった仮想化基盤を提供するプロバイダーも用意しておく必要がある。本提案では Vagrant を VirtualBox [3] と組み合わせて利用している。

2.3 プログラミング演習用 Vagrant 環境の構築

プログラミング演習に用いる Vagrant 環境を構築するために、Vagrant を用いて学生に配布するイメージをパッケージにする。Vagrant では配布するパッケージを box と呼ぶ。

図 1 に配布用 box を作成するための Vagrant の設定ファイルを示す。Bento [1] が提供する ubuntu 18.04 を土台となる box として利用している。

Vagrant では、土台となる box に任意のソフトウェアを追加でインストールしたり、設定を施す作業をプロビジョニングと呼ぶ。これを行うために作成した shell のスクリプトファイルを図 2 に示す^{†1}。

以上の設定を準備し、Vagrant でプロビジョニングを実行するとその結果をパッケージファイルとして出力できる。

なお、これらのファイルを MIT ライセンスのもとで公開している^{†2}。

^{†1} Bento プロジェクトは Chef での利用を想定しているが、この使い方であれば冪等性を担保する必要がないため shell を使う方が実行時間が短い。

^{†2} https://github.com/yhubachi/vagrant_enpit

```

#!/bin/sh

# update apt repository
apt-get update

# tools
apt-get install -y emacs

# postgres
apt-get install -y postgresql
systemctl enable postgresql.service

# create a database for vagrant
/etc/init.d/postgresql start
sudo -u postgres createuser -d vagrant
sudo -u postgres createdb vagrant
/etc/init.d/postgresql stop

# install heroku tool belt
snap install --classic heroku

# ruby
apt-get install -y autoconf bison build-essential \
  libssl-dev libyaml-dev libreadline6-dev zlib1g-dev \
  libncurses5-dev libffi-dev libgdbm5 libgdbm-dev \
  libpq-dev

sudo su - vagrant <<'EOF'
  cd ~
  # rbenv
  git clone https://github.com/rbenv/rbenv.git ~/.rbenv
  echo 'export PATH=~/.rbenv/bin:$PATH' >> ~/.bashrc
  echo 'eval "$$(rbenv init -)"' >> ~/.bashrc
  # ruby-build
  mkdir -p ~/.rbenv/plugins
  git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/plugins/ruby-build
  # install ruby
  ~/.rbenv/bin/rbenv install 2.5.5
  ~/.rbenv/bin/rbenv global 2.5.5
  ~/.rbenv/shims/gem install rails
EOF

```

図 2 provisioner.sh ファイル

2.4 Vagrant 環境の配布

前項で作成したパッケージは HashiCorp 社が提供する Vagrant Cloud を用いて公開することができる。パッケージングしたファイルをこのクラウドにアップロードすると、だれでも自由にダウンロードできるようになる^{†3}。

これをダウンロードし、演習のための開発環境として利用するためには、次の作業が必要となる。

1. VirtualBox のダウンロードとインストール
 2. Vagrant のダウンロードとインストール
 3. 利用するための Vagrantfile の作成
 4. box のダウンロードと仮想環境の立ち上げ
1. と 2. は配布元のサイトから容易に実施できる。

^{†3} <https://app.vagrantup.com/ychubachi/boxes/enpit>

また、3. を行うためには Vagrant を用い、次の通りコマンドを実行する。

```
vagrant init ychubachi/enpit
vagrant up
```

これを行うだけで、配布した Vagrant による仮想環境が立ち上がり、SSH 接続によりログインできる。

3 提案手法の評価

この仮想環境は 2013 年度の enPiT プログラムから継続して利用しており、2017 年度から本学の修士課程の科目でも利用を始めた [6][5]。enPiT では本学の学生の他、BizApp 分野/BisSysD 分野の参加校・連携校の学生^{†4} も利用した。利用した学生はプログラムの経験者からある程度の経験者まで、本学の社会人学生も履修したことから年齢層も様々である。演習は各自が所有する PC を用い、BYOD で行った。

提案手法を用いた演習は、Web アプリケーションのチーム開発を通して最先端のアジャイル開発手法を学ぶためのプログラミング演習である。Web アプリケーションの開発フレームワークには Ruby on Rails を用い、データベースは PostgreSQL、ソースコードのバージョン管理ツールには Git を用いた。

開発した Web アプリケーションは、クラウド環境である Heroku でを公開し、ソースコードの共有には GitHub、継続的統合 (CI) のツールとして Travis CI を利用した。受講者はこれらのツールを使いこなし、アジャイル開発の手法を体得する。

このような演習を実施するために利用したツールは前掲の図 2 に示すスクリプトにより、予めパッケージ化し、学生に提供した。

4 考察

4.1 本提案の長所

本研究で開発した Vagrant の仮想環境パッケージの長所は、全ての受講者が完全に共通の開発環境を利用できるという点にある。仮に、演習で用いた全てのツールを受講者自身にインストールさせたとすると、

どんなに手順書を整備してもヒューマンエラーは防ぐことができないし、受講者のスキルによってはインストール自体につまずく恐れもある。

本環境の導入には VirtualBox と Vagrant とという 2 つのソフトウェアのインストールが必要であるが、ここ 7 年間の経験はトラブルが生じた受講者はいなかった。また、作成したパッケージの配布も、一般的なネット回線で 10~15 分程度でダウンロードできるサイズであり、大きなトラブルにはならなかった。

4.2 本提案の短所

一方、Windows 環境の受講者は、仮想化技術を制御する Hyper-V と呼ばれる Windows の設定に起因するトラブルが発生した。また、ユーザ名を日本語でつけている場合にも問題が生じた。

現在配布している環境は CUI での利用を想定している。CUI を使わせることの意義はあるとは考えるが、先に進んだ学生が GUI を備えた IDE や高機能のエディタを利用しようとするとき使いづらいという問題もあった。

5 おわりに

本発表ではアプリケーション開発演習のために仮想環境を用いる方法について提案した。受講者は各自の PC で全く共通の開発環境を円滑に利用することができ、BYOD によるアドホックな演習環境にも対応する。今後、仮想化技術の進化を踏まえ、本手法を改善させていきたい。

参考文献

- [1] HashiCorp: bento - Vagrant Cloud. <https://app.vagrantup.com/bento> (accessed on 2019-08-05).
- [2] HashiCorp: Vagrant. <http://www.vagrantup.com/> (accessed on 2019-08-05).
- [3] Oracle: Oracle VM VirtualBox. <http://virtualbox.org/> (accessed on 2019-08-05).
- [4] 産業技術大学院大学: enPiT BizApp 産業技術大学院大学. <http://enpit.aiit.ac.jp/> (accessed on 2019-08-05).
- [5] 小山 裕司中鉢 欣秀: アジャイル開発技術及び協調作業スキルセット学習のための体系的な教育プログラム, 産業技術大学院大学紀要, No. 10(2017), pp. 37-41.
- [6] 小山 裕司中鉢 欣秀: アジャイル人材育成プログラムの専門職大学院修士課程への導入, 9 2018, pp. 425-426.

^{†4} 主に、琉球大学、筑波大学、はこだて未来大学