

# OpenStack を用いた Web サーバ設定演習環境の構築

中崎 満晶<sup>1,a)</sup> 越智 徹<sup>2,b)</sup> 中西 通雄<sup>3,c)</sup>

**概要:** 情報系学部では、Web サーバや DNS サーバの設定演習の授業を置いているところがある。演習には 1 人 1 台の PC を用いるため複数台の PC を用意する必要があり、実ネットワーク環境の準備といった種々のコストがかかるのに加え、演習中に学習者への指導に目が行き届かない。なかでも、設定ファイルを編集して動作を確認する内容は、演習として比較的複雑である。本研究では、Web サーバ設定演習のための仮想化環境を構築した。PC 実機を複数台用いる代わりに、OpenStack を用いて演習環境を構築している。さらに、Web サーバソフトウェアの設定演習において教員が学習者を円滑に支援するためのシステムを作成した。ブラウザの画面、コマンド操作、設定ファイルの変更履歴およびログが見やすく表示された画面を利用することで、教員が学習者の PC のところに行って状況を見る手間を減らすことができる。

**キーワード:** サーバ設定演習, OpenStack

## Construction of OpenStack virtual machine environment for Web server setting exercise

MITSUAKI NAKASAKI<sup>1,a)</sup> TORU OCHI<sup>2,b)</sup> MICHIO NAKANISHI<sup>3,c)</sup>

**Abstract:** Some of the undergraduate schools of Computer Science have a class of server settings exercise of Web and/or DNS server(s). Usually each student uses one PC in the exercise class. So, computer lab staff have to prepare many PCs and dedicated network configuration. In addition to its time and effort, teachers cannot assist every student carefully, because students modify a couple of setting files. To solve these problems, we constructed a virtual machine environment using OpenStack for Web server setting exercise, which will help teachers to assist students effectively. Teachers can understand students' status using the operation history screen which chronology lists up browser snapshot, command history, file modification history and logs. Furthermore, teachers can operate student's virtual PC remotely without going to his seat.

**Keywords:** Server setting exercise, OpenStack

## 1. はじめに

### 1.1 研究概要

大阪工業大学情報科学部では、専門教育の一環として「サーバ構築管理」という選択科目が開講されており、20 人ほどが受講している。1 人 1 台の PC を用いて OS のインストールから DNS サーバ、Web サーバおよびメールサーバの構築を行う。なかでも、整備されて流れ作業的になっている OS のインストールと比較すると、Apache HTTP Server(以下、単に Apache と書く)や SMTP Server などの設定は、設定ファイルを編集して動作確認をする必要があ

<sup>1</sup> 大阪工業大学大学院情報科学研究科情報科学専攻  
Graduate School of Information Science and Technology, Osaka Institute of Technology

<sup>2</sup> 大阪工業大学大学情報センター  
Computing Center, Osaka Institute of Technology

<sup>3</sup> 大阪工業大学大学情報科学部コンピュータ科学科  
Department of Computer Science, Faculty of Information Science and Technology, Osaka Institute of Technology

a) m14a07@st.oit.ac.jp

b) ochi@center.oit.ac.jp

c) naka@is.oit.ac.jp

るため、演習としては比較的複雑である。

このような実機を用いた演習環境では受講者数分の PC を用意する必要があり、ネットワーク環境の準備といった種々のコストもかかる。この問題に対して、プライベートクラウドを用いた演習環境の提案がなされている [1]。

本研究では Web サーバソフトウェア (Apache) の設定演習に限定し、仮想のサーバ設定演習環境を構築した。学習者は演習室や BYOD の PC 端末を用いてブラウザ上で演習できる。また、教員が学習者を円滑に支援するためのシステムを作成した。Web サーバの設定演習において学習者を支援するためには、(1) 不具合状況の把握、(2) 原因の特定、(3) 修正のプロセスが必要であり、(1) には学習者のブラウザの画面、(2) にはログ、コマンド履歴および設定ファイルの変更履歴が手掛かりとなる。しかし、ログ以外の手掛かりは自動的に残らない。そこで、著者らが開発を行ってきたサーバ設定演習環境 [4] の機能を拡充して、学習者のブラウザの画面、コマンド履歴、ファイルの変更履歴およびログ (以下、まとめて操作履歴とする) を、操作履歴の更新があった順に時系列で表示し、課題状況と合わせて教員画面で確認できるようにした。

## 1.2 類似研究との比較

先行例としてネットワーク機器の設定ファイルを採点するシステムがある [5]。このシステムでは設定内容の「曖昧さを考慮した」採点ができる。また、ネットワーク機器の設定ファイルについて模範解答と比較、採点を行う機能を作成するにあたって、テキスト形式の設定ファイルという性質上、同様にテキスト形式であればどのような設定ファイルでも比較、採点ができるようなアルゴリズムを実装している。そのため、Apache の設定ファイルなども同様に採点できる。しかし、設定が意味的に間違っている場合には実際に実行して動作を確認するまで、このシステムでは事前にチェックできない。また、動作確認時のエラーは標準エラー出力やエラーログに出力される。

本研究で作成したシステムでは、動作確認時の標準出力、標準エラー出力およびログを取得しているため、意味的に間違った設定の特定に必要なエラーメッセージを確認できる。

また、北澤ら [6] による IP ネットワーク構築演習の演習履歴の管理を可能とした先行例もある。この例では学習者がブラウザから演習を行うことができる。また、授業時間外や在宅時の自習における学習者の学習状況を把握するために、ネットワークの構築状況を自動採点した結果と学習者の演習履歴および演習手順を GUI で閲覧できる。

本研究と比較して、ブラウザから演習を行う点と操作履歴を時系列で表示している点は共通している。しかし、操作履歴閲覧画面ではコマンドの履歴のみが表示されているのに対し、本研究では各コマンドの実行結果も表示して

いる。

少し視点を変えてプログラミング演習について考えて見ると、この授業でも教員による学習者の支援が必要となる場合が多い。この支援を効率的に行うために進捗を把握するシステムはいくつか運用されている [2][3]、

事例 [2] ではプログラミング演習において学習者の進捗を把握するために、オンラインにエディタでのアルファベットの文字入力や削除などの操作を履歴としてデータベースに保存している。本研究でも、ファイルの変更履歴をデータベース (DBMS) で管理することも検討したが、差分を容易に作成できるようにするため git を用いてファイルの変更履歴を管理するようにした。

事例 [3] では掲示板を作成する Web プログラミング演習において、Web サービスのログや Selenium を用いて動作確認を行うことで学習者の進捗状況を把握している。しかし、掲示板の動作確認のたびに学習者の掲示板に同じ記事を投稿すると、学習者にも迷惑がかかり、Web サービスのログを汚してしまう。本研究では、Web サービスのログを汚さないために、学習者の環境を複製して動作確認を行うようにした。

## 2. 本システムについて

### 2.1 演習環境

本演習環境は、CentOS をベースとして、Apache、OpenStack および Moodle といったソフトウェアを用いて構築した。

OpenStack はクラウド基盤を構築するオープンソースのミドルウェアである。IaaS (Infrastructure as a Service) やストレージサービスを提供するための仮想サーバやストレージ、ネットワークの管理機能などを提供する。複数の実計算機で運用することも可能だが、本研究では 1 台のサーバでの処理能力を調査するために、あえて 1 台のサーバで実験運用を行った。

Moodle は PHP で書かれたオープンソースの LMS (Learning Management System) である。本研究では Moodle のユーザ管理機能を利用してユーザ管理をしている。また、OpenStack と連携したプラグインを作成し、学習者がブラウザから仮想サーバでのサーバ設定演習を行えるようにした。さらに Moodle のソースコードを編集し、ユーザの作成および削除時に OpenStack 側でもユーザごとの仮想サーバの起動・削除が連携して行われるようにした。

本演習環境を図 1 に示す。2 台の実機サーバがあり、Moodle がインストールされたサーバは演習環境用、Docker がインストールされたサーバは課題状況把握用である。学習者にはそれぞれ Apache の設定用と動作確認用の 2 台の仮想サーバを提供し、ブラウザ経由で演習を行ってもらう。2 台の仮想サーバは OpenStack で作成したプライベートネットワークに接続されているが、NAT で外部と通信が

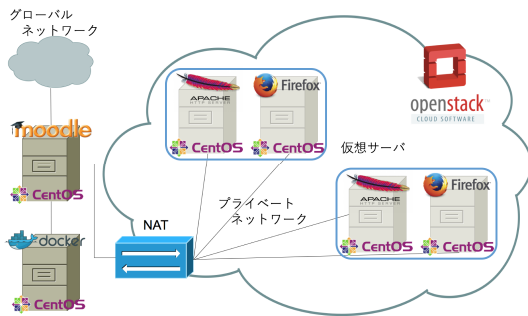


図 1 演習環境

表 1 サーバ環境 (演習環境用)

種類	詳細
CPU	Intel(R) Core(TM) i7-4770K CPU
メモリ	32GB
ハードディスク	HDD 2TB 7200rpm
OS	CentOS Linux release 7.1.1503 (Core)
ミドルウェア	Apache / 2.4.6
	MySQL / 5.5.40-MariaDB-wsrep
	PHP / 5.4.16
	OpenStack / Juno
	Moodle / 2.9

表 2 サーバ環境 (課題状況把握用)

種類	詳細
CPU	Intel(R) Core(TM) i7-6700K CPU
メモリ	32GB
ハードディスク	M.2 SSD 240GB
OS	CentOS release 6.7 (Final)
アプリケーション	Apache / 2.2.15
	MySQL / 5.6.27
	PHP / 7.0.2
	Docker / 1.7.1

可能である。また、グローバル IP アドレスを割り当てることで外部からアクセスも可能である。

## 2.2 サーバ環境

演習環境および課題状況の把握に使用したサーバの環境は表 1, 表 2 に示した通りである。

## 2.3 Apache における設定内容の分類

Apache を用いた Web サーバの設定を、それぞれ一行のみ設定ファイルを変更する内容と 2 行以上変更する内容に分類し、エラーの原因を特定する際に、(A) 設定ファイル、(B) コマンド履歴、(C) ログの 3 つのうち、どれを必要とするかを表 3, 表 4 に示す。

## 2.4 システムの仕様

学習者および教員はブラウザから Moodle にログインして操作を行う。ログインしたユーザ名によって学習者用画面あるいは教員用画面が表示される。今回は Moodle をイ

表 3 1 行のみ変更する内容

		手掛かりの組み合わせ		
		(A)	(B)	(C)
設定の種類	サービスの起動		○	
	DirectoryIndex の設定	○		○
	ポート番号の変更		○	
	DocumentRoot の変更		○	
	IP 制限	○		○
	ヘッダの設定	○		
	モジュールの読み込み		○	

表 4 2 行以上変更する内容

		手掛かりの組み合わせ		
		(A)	(B)	(C)
設定の種類	CGI の設定	○		○
	Basic 認証, Digest 認証	○	○	
	SSL の設定	○	○	
	Virtual Host の設定	○	○	

ンストールした時に自動で作成されるユーザ「admin」を教員用アカウントとし、それ以外を学習者アカウントとしている。

学習者用画面を図 2 に示す。左側の黒い部分は VNC コンソールで、学習者は VNC コンソールから Web サーバ設定演習を行い、右側の VNC ブラウザから設定の動作確認を行う。

教員用画面を図 3 に示す。この画面では、学習者の学籍番号 (図 3 では test1, test2) と Web サーバの IP アドレスの右側に、学習者の VNC コンソールと学習者の操作履歴を確認できるボタンがあり、ボタンの右側に学習者の課題状況が表示されている。課題状況は一定時間ごとに更新され、チェックして合格になった状態を「緑色」で、不合格の状態を「赤色」で表している。課題状況の確認時に学習者の環境にアクセスするとログを汚してしまうため、学習者の設定ファイルおよび Apache の動作確認に必要なフォルダ・ファイルをコピーし、Docker のコンテナで Web サーバを立てて動作確認を自動で行っている。今回は DirectoryIndex の設定のみをチェックできる仕組みを作成している。

Console ボタンを押すと学習者と同じ VNC コンソールが表示される。そのため教員は学習者の操作をリアルタイムに見ることができる。逆に教員がコンソールを操作すれば、その操作を学習者がリアルタイムに見ることができる。教員が学習者の画面を操作する必要があった場合、このモーダルに表示されている VNC 画面から操作することができる。

Diff ボタンを押すと操作履歴を確認するための画面が表示される (図 4)。画面には学習者のブラウザの画面のス

表 5 稼働しているサービス

サービス名	役割
format	inotifywait コマンドで script コマンドが出力したコマンドの履歴の変更を検知した時、操作履歴を HTML ファイルとして整形して nginx のドキュメントルートに設置する
gitinit	起動時に/etc/httpd および /var/log/httpd 配下を git の管理下に置くように初期化する
httpdlog	inotifywait コマンドで/var/log/httpd 配下のファイルの変更を検知した時、その変更をコミットする
inotifywait	inotifywait コマンドで/etc/httpd 配下のファイルの変更を検知した時、その変更をコミットする
keyfix	Windows キーなどの登録されていないキーが押された時に、キーコードに関するエラーが出るためその対策
script	script コマンドでコマンドの操作履歴を/var/logに出力する

ナップショットが横に時系列に並んでおり、いずれかのスナップショットをクリックするとその左側に前回のブラウザでの動作確認後の操作履歴が表示される。

操作履歴の画面を図 5 に示す。この画面ではコマンドの操作履歴に加え、Apache ログと設定ファイルの変更履歴の差分をハイライト表示している。

教員は課題状況から、課題ごとに「赤色」の表示の学習者を特定し、その学習者の操作履歴を確認して間違っている箇所を教えることで学習者の支援を行う。学習者から質問された場合も同様である。

## 2.5 コンソール用イメージファイル

コンソール用イメージファイルの CentOS は CUI で動作する。あらかじめ Apache がインストールしてあるので、Apache を起動すれば Web サービスが立ち上がる。

操作履歴を取得する方法として OpenStack がインストールされているサーバから各学習者の仮想サーバの操作履歴を取得して処理するのではなく、学習者のイメージファイルに操作履歴を取得する仕組みを埋め込み、それぞれの仮想サーバで処理するようにしている。こうすることで操作履歴の取得にかかるオーバーヘッドを少なくしている。

取得した操作履歴は HTML ファイルとして整形し、学習者の仮想サーバを Web サーバとして立ち上げることで、HTTP 経由で確認できるようにした。Apache は演習で使用するため、操作履歴を提供するサーバには nginx を用いた。

操作履歴を取得するために今回作成した複数のサービスが動作している。それぞれのサービスの役割を表 5 に示す。役割の説明におけるコマンドやパスは次の通りである。inotifywait コマンドはファイルの変更を監視し、script コマンドは端末の操作を記録するコマンドである。

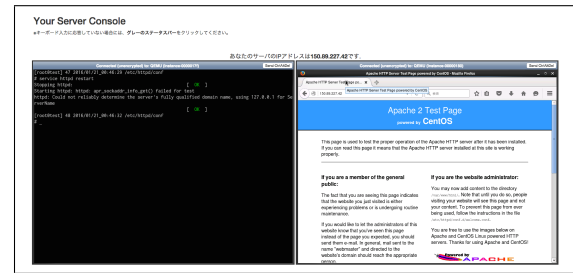


図 2 学習者用画面



図 3 教員用画面

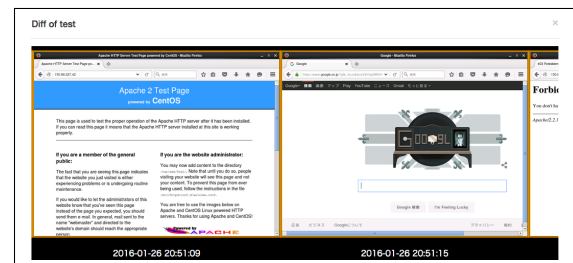


図 4 スナップショット一覧

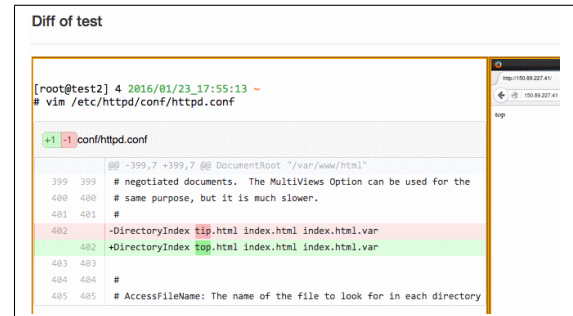


図 5 操作履歴

## 2.6 ブラウザ用イメージファイル

ブラウザ用スナップショットは GUI で動作する。スナップショットを起動すると自動ログインし、Firefox が立ち上がるようになっていく。

また、Firefox でページのロードが完了した際に、画面全体がキャプチャーされて画像として保存されるようにした。Apache で Web サーバを立てて、PC の画面全体をキャプチャーするエンドポイントをあらかじめ用意し、Firefox のプラグインでロード完了イベントを検知してプラグインからエンドポイントを叩くことで実現している。

## 3. 評価

### 3.1 評価方法と結果

本学情報科学部の 3, 4 年生あわせて 10 人に被験者とし

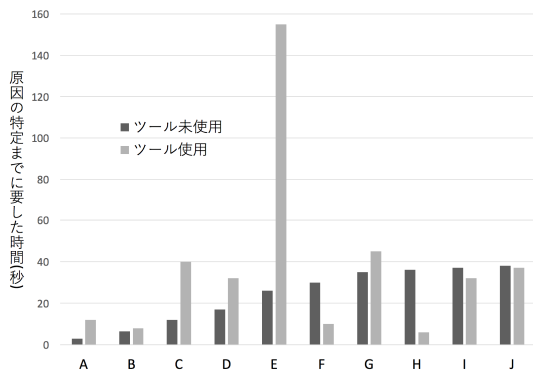


図 6 エラーの原因特定に要した時間

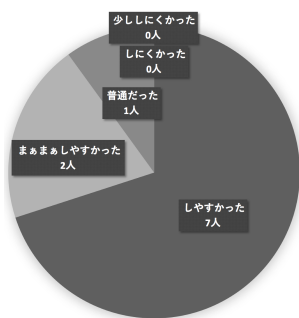


図 7 原因の特定しやすさ

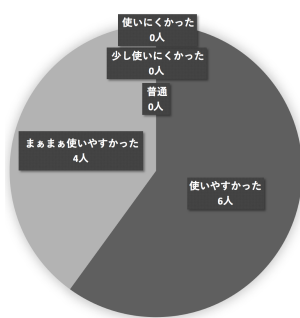


図 8 使いやすさ

て協力してもらって教員用機能の評価を行った。本研究の操作履歴画面(図 4, 図 5)を使用した場合と、操作履歴の画面を使用せずに学習者用画面(図 2)のみを使用した場合のそれぞれを、教員の立場で体験してもらった。学習者が行う演習内容は 2.3 節において同じカテゴリの DirectoryIndex の設定と IP 制限を行った。どちらもエラーが出力されない typo を著者らが仕込んでおいた。教員役の被験者が設定ファイルの typo 誤りを見つけるまでに要した時間の計測とアンケートによる調査を行った。結果を図 6 に示す。10 人を A から J で表している。

アンケート結果では、「操作履歴画面を使ったほうが原因を特定やすかった」が 9 人(図 7)、「操作履歴画面が使いやすかった」が 7 人(図 8)、本システムの良かった点(表 6)では、「変更点が強調されていてわかりやすい」という意見が多く、悪かった点(表 7)では「演習の作業内容が多い場合に対応しきれないように思う」、「フレーム内の操作履歴の再読み込みがめんどろ」などの意見があった。

### 3.2 スケーラビリティの検証

Moodle から手動で学習者を順次追加し、本システムの負荷状況を dstat コマンドを用いて計測した。15 人の学習者(30 台の仮想サーバ)を追加し終えた直後の状態は、メモリ使用量が約 20GB であった。実メモリは 32GB あるため余裕があったが、仮想サーバ起動中は CPU 使用状況のうち、待ち状態の割合が平均 30% ほどであった。そこで、一番最後に起動した学習者の仮想サーバの状態を確認する

表 6 本システムの良かった点(自由記述)

回答番号	回答
1	ファイルを変更したところがわかる
2	一覧で画面が見れる
3	学習者の編集したポイントがわかりやすく従来の方法よりもミスの発見が早く済みそう
4	教える側の人からしたら色で変更点がわかりやすかった
5	手を加えた点が一目で分かるので授業の補助システムとして役立つそうだなと感じた。教員補佐の負担を減らせそう
6	学習者の編集部分が一目で分かった
7	学習者のどこを修正したとかの履歴を見られる点見易い。サーバーがちゃんとできたかどうか確認するためのアクセスした際の Web 画面が見られる点が便利
9	どこを変更したかが一目で分かった。また、追加された文字だけ強長されていて見やすい
10	ブラウザの画面を更新するごとに設定ファイルの変更部分を確認できるので間違いをすぐに発見できる点
11	変更した箇所が色分けされて表示されるのがわかり易かった

表 7 本システムの(自由記述)

回答番号	回答
1	モーダルのサイズが見にくい
2	フレームの再読み込み
3	授業の作業内容が多い場合に対応しきれないのでないかと少し感じた
4	トラックパッドが使いにくい
5	詳しいものではないので Diff とかが何を示しているのか分からなかった

と、サーバ設定用の仮想サーバは起動が完了しており、動作確認用の仮想サーバは起動中であった。なかには起動に 30 分かかるものもあり、非常に処理が遅い状態であった。

仮想サーバの起動に用いているイメージのスナップショットのサイズは、サーバ設定用が 2.0GB、動作確認用が 5.1GB である。また、サーバ設定用は CUI、動作確認用は GUI で動作するため、GUI の方が CUI より起動時の処理が重いことや、イメージファイルのサイズが大きいために、起動に時間がかかっていると考えられる。ファイルサイズの大きさゆえに、アクセスに時間を要していることから、HDD を SSD に換装することで起動時の I/O 待ちを改善できると思われる。

また、学習者を 15 人にした時に、仮想サーバが起動後に動作確認を行った。サーバ設定用の方は通常通り動作したが、動作確認用のサーバはキー入力の応答が遅く、使用するには少し厳しい状態であった。学習者数を減らし、8 人にした状態であればどちらの仮想サーバもストレスなく操作できる状態だった。

本システム環境の場合、8人程度であれば一度に演習を行うことができることがわかった。

## 4. 考察と今後の課題

### 4.1 考察

被験者10人中6人は、操作履歴画面を利用した場合に、原因の特定に要する時間が増加している(図6)。設定ファイルを1行のみ書き換える内容に対し、操作履歴画面に表示している操作履歴の情報が多すぎることが原因と考えられる。

9人の被験者が、操作履歴画面を使ったほうがエラーの原因と特定しやすいと感じた理由として、操作履歴の差分表示におけるハイライトが効果的であったと考えられる。

被験者Aと被験者Dはサーバ管理の経験があったが、経験のない被験者のうちの半分が原因の特定に要した時間を短縮できているため、本研究の操作履歴画面はサーバ管理に関する経験がなくても利用できる可能性を示した。

### 4.2 今後の課題

2.3節であげた設定内容が正しく設定されていても、動作確認で正しく動作しないことがある。エラーがファイアウォールやSELinux、権限と所有者に起因する場合である。本研究では、環境や権限などによるエラーを特定するための情報を取得していない。権限やエラーに関係する可能性がある手掛かりを取得することが今後の課題として挙げられる。

アンケート結果にもあったように作業内容が多いと操作履歴が多くなるため、横スクロール量の増加に伴う利便性の低下や情報が多すぎるといった問題が発生する。必要な情報だけをまとめて見やすく表示し、それ以外の情報は必要になった際に操作することで確認できるようにすれば対処できると考えられる。

課題状況把握サーバではDirectoryIndexの設定のみのチェックスクリプトを作成したが、他の課題についても同様に作成する必要がある。

### 4.3 導入について

本演習環境の導入に関して、OpenStackのインストール方法に関してのマニュアルを作成した。しかし、Moodleのインストールおよびプラグインの導入方法、課題状況把握サーバの構築についてのマニュアルは作成中である。OpenStackのインストールだけでも内容が複雑でインストールに時間がかかる。

そこで、クラウド環境は外部のサービスであるAmazon Web Service(以下、AWS)を利用できないかを調査した。AWSのEC2と呼ばれるサービスでは有料で仮想サーバを提供しており、スナップショットの作成やネットワークの作成といったOpenStackと同等のことが可能である。加

えてAPIが用意されており、仮想サーバの起動・停止・再起動・削除・情報の取得が可能であるため、本演習環境をAWSで運用することも可能と考えられる。AWSで演習環境のスナップショットを作成しておけば、スナップショットから仮想サーバを起動して必要な設定をするだけで利用可能になるため、EC2の使用料金はかかるが導入コストおよび保守コストを削減できる。

## 5. 結論

本研究はWebサーバ設定演習において教員画面から学習者の課題状況と操作履歴を確認できるようにすることで、学習者の支援を円滑に行える可能性を示した。

本システムはWebサーバ設定演習のみに限定していたが、監視するフォルダを変えるだけでファイルの変更履歴を取得できるため、nginxやBINDといったソフトウェアの設定やCGIの演習においても活用できると考えられる。

ファイルやフォルダの権限・所有者の情報の取得、操作履歴画面に表示する情報の整理、課題状況把握用の各課題のチェックスクリプトの作成、および性能向上のための工夫が今後の課題である。

## 参考文献

- [1] 鎌田 元樹, 榊田 秀夫: “OpenStackを利用したサーバ設定演習システムの提案”, 情報処理学会研究報告, Vol.2013-IOT-23, No.3, pp. 1-6, (2013.9)
- [2] 井垣 宏, 齊藤 俊, 井上 亮文, 中村 亮太, 楠本 真二: “プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PV の提案”, 情報処理学会論文誌, Vol.54, No.1, pp.330-339, (2013.1)
- [3] 安留 誠吾: “Webプログラミング演習における学習進捗把握”, 教育システム情報学会 第39回全国大会, H3-3, (2014.9)
- [4] 中崎 満晶, 越智 徹, 中西 通雄: “OpenStackを利用したサーバ設定演習環境の構築および Moodle との連携”, 教育システム情報学会学生発表会, (2014.3)
- [5] 島野 顕継, 内田 光一, 河辺 幹也, 福田 匡志: “ネットワーク機器を用いた演習における授業支援システムの開発”, CAUA 会誌, Vol.14, pp.70-75 (2014.3)
- [6] 北澤 友基, 井口 信和: “演習履歴の管理を可能とする IP ネットワーク構築演習システムの開発”, 情報処理学会第76回全国大会, 1ZF-3, (2014.3)