

オンライン大学における演習系科目の運営

中谷 祐介 後藤 幸功
サイバー大学 IT 総合学部

Online IDE for Supporting Programming Courses at Online University

Yusuke NAKAYA Yukinori GOTO
Faculty of IT and Business, Cyber University

概要: 大学に一切通学する必要がないオンライン大学では、プログラミング等の演習系科目もオンラインの環境で運営する必要がある。本稿では、オンライン大学における演習系科目の運営について報告する。本稿で対象とする演習系科目は、C 言語, Java, PHP 等のプログラミング科目と、UNIX のオペレーションやサーバの構築・設定を行う科目である。これらの科目におけるオンラインでの演習環境、運営方法について報告する。

キーワード: e ラーニング, プログラミング, IDE

1. はじめに

サイバー大学 (以下, 本学) は, すべての授業をインターネットを通じた e ラーニングにより学生に提供する大学である [1]-[4]。本稿では, このような大学を「オンライン大学」と呼ぶこととする。本学では, 学生に対し, ビジネスと IT エンジニアリングの両方の基礎知識を修得するカリキュラムを提供している。IT エンジニアリングに関する科目には, プログラミング演習や UNIX のオペレーションとサーバの運用に関する演習科目がある。これら演習科目を受講するとき, 学生は各自の PC に演習に必要なソフトウェアをインストールしなければならない。しかし, 学生の PC 操作のスキルや PC の仕様, OS の種類が異なるため, ソフトウェアのインストールを全員同じように行うことは難しい。そのため教員は, 学生の演習環境を構築するための指導に時間が費やされてしまう問題がある。そこで, 学生がプログラミング関連の演習科目を受講する際の演習環境を大学が提供することで, 学生が受講に集中しやすい環境を提供することが望まれる。すべての学生の受講環境として統一している環境はウェブブラウザであることを考慮して, ウェブブラウザ上で動くプログラミング環境と UNIX のターミナル環境

を提供するサービスを学生に提供することを考える。これらの環境を提供するサービスとして, **codigm** 社が開発した「Goorm IDE」(以下, **goorm**) がある。本学ではこの **goorm** を 2017 年度から導入し, プログラミングに関する科目と, UNIX のオペレーションとサーバ運用に関する科目において演習環境として利用した。本稿では, **goorm** の概要を述べ, **goorm** を利用する科目の内容および運営と, そこから派生する運営上の課題について述べる。

2. サイバー大学における授業運営

ここでは, 本学における授業運営について述べる。本学では, 独自の e ラーニングシステムを構築している。学生は, ウェブブラウザを使用してこのシステムにアクセスし, 科目の受講, 課題の提出, 期末試験等, 大学における活動をすべてこのシステム上で行う。本学の科目では, すべての授業はインターネットを通じてコンテンツとして配信される。コンテンツには以下の 2 種類がある:

- Video on Demand (VoD) 形式
- Web Based Training (WBT) 形式

VoD 形式のコンテンツは, 授業内容のスライドと教

員の映像により構成される。教員がスライドの内容をもとに授業を展開する、映像が中心のコンテンツである。

WBT 形式のコンテンツは、ウェブページの形式で構成される、文字情報を中心として映像、音声を組み合わせたコンテンツである。学生は、テキストを読みながら、映像や音声による補足情報を参照して受講を進める。

各科目は、これらの 2 種類のコンテンツを組み合わせ構成している。

各科目では、全 15 回の授業を毎週 1 回ずつ配信する。学生は、各回の授業コンテンツを視聴した後、各回に設置されている課題を完了することでその回の出席が認められる。課題には、小テスト、レポート、ディベートがあり、各回においていずれかの課題が必ず課される。このように、学生は授業を受けるというインプットと課題に取り組むというアウトプットの双方向の学習を行う。学期末には期末試験を実施し、学期を通しての科目の理解度を判定する。

また、学生からの質問に対しては、授業内容に関する質問などを受け付ける専用の掲示板をシステム上に用意している。学生がその掲示板に質問を投稿すると、教員、補助教員がそれに回答し指導を行う。

3. カリキュラムと要求仕様

ここでは、本学のプログラミングに関連する科目のカリキュラムと、そこで求められる演習環境に対する要求仕様について述べる。

3.1 カリキュラムの内容

プログラム関連の科目では、以下に挙げる 6 つの学習目標を達成することができるように授業を構成している：

- ① プログラミングスキルを習得する
- ② コンパイラ言語とインタープリタ言語の違いを理解する
- ③ 統合開発環境を使用してソフトウェア開発ができる
- ④ UNIX のオペレーションができる
- ⑤ ディレクトリの概念を理解したファイル管理ができる

- ⑥ ウェブサーバや DB サーバの設定ができる
- ⑦ ウェブアプリケーションを用いたサービスを構築できる

学習目標①のプログラミングスキルの習得については、以下の言語を学習する：

- C 言語
- Java
- JavaScript
- PHP
- Perl
- Bash による Shell スクリプト

学習目標②については、各言語を習得するための科目において、講義と演習によってその違いを学習する。

学習目標③～⑥については、UNIX のオペレーションと UNIX 上のサーバ設定を行うことで、ウェブアプリケーションの構築ができるまでの内容を一貫した授業として行う。

これらの授業を学生に提供するうえで、各科目で異なる演習環境を提供することは、学生の負担が高くなり好ましくない。そこで、プログラミング関連科目の演習環境として、これらの学習目標を達成できる統一した演習環境のシステムが必要となる。

3.2 演習環境の要求仕様

2.1 で述べたように、プログラミング言語の学習と、UNIX のオペレーションまでを含めた演習環境の要求仕様を検討する。

まず、1. で述べたように、学生の PC 環境はそれぞれ異なるため、ウェブブラウザ上で動く演習環境が必要となる。次に、プログラミング言語を学習するために、各種言語が事前に準備された環境が必要となる。そして、プログラムを記述するためのテキストエディタも必要となる。さらに、UNIX のオペレーションを行うために、UNIX におけるターミナルの環境が使用できなければならない。これらを考慮すると、UNIX の環境をウェブブラウザ上で 1 人 1 台提供できることが求められる。最後に学習目標達成のために、ソフトウェアの統合開発環境も必要である。

これらをまとめると、以下の機能を提供するウェブブラウザ上で提供できるサービスが必要となる：

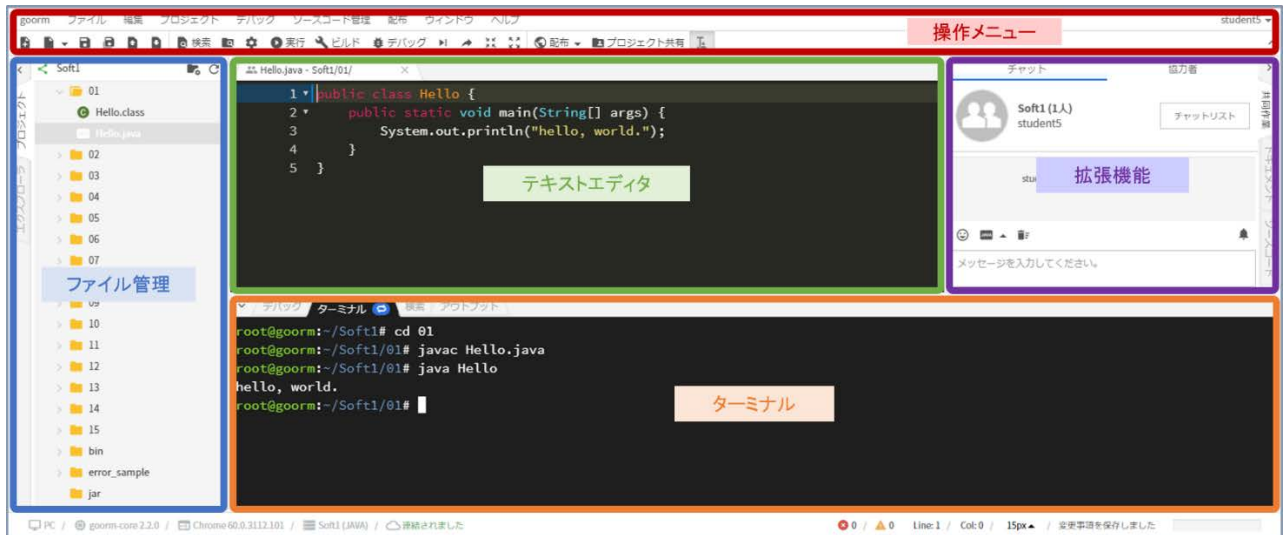


図 1 演習環境「goorm」の画面構成

- UNIX 環境を個々に利用できる
- ウェブブラウザ上で UNIX のファイルを編集できる（統合開発環境と共有）
- ウェブブラウザ上で UNIX のターミナルを使用できる
- UNIX のファイルシステムと連動した統合開発環境がある

これら 4 つを提供するサービスとして、本学では goorm を導入し、これを用いてプログラミング関連の科目を提供している。

4. 演習環境「goorm」

ここでは、本学が導入している goorm について、その概要と機能を述べる。

4.1 goorm の概要

goorm は、Linux の環境をベースとしたオンラインで利用することができる統合開発環境 (IDE) である。IDE としての機能だけでなく、ユーザ間でファイルを共有する機能や、ユーザ間でチャットを行う機能を備えている。

学生が goorm を利用するためには、ウェブブラウザ上で大学が指定するウェブサイトアクセスし、学生ごとに配布されるログイン情報を使用して goorm のシステムにログインを行う。ログインが完了すると、学生が履修する演習科目の一覧が表示される。学生が演習を行う科目を選択すると、その科目の演習環境が起動する。起動後の演習環境の画面を図 1 に示す。

4.2 goorm の機能

ここでは、図 1 をもとに、goorm の機能を解説する。

4.2.1 操作メニュー

「操作メニュー」の部分では、演習環境の操作を行うためのメニューが提供される。演習環境の設定や、記述したソースファイルのコンパイル、実行の操作など、さまざまな操作を行うためのメニューが準備されている。goorm 上で作成したファイルをユーザのコンピュータ上にダウンロードする操作や、ユーザのコンピュータ上で作成したファイルを goorm 上にアップロードする操作も、ここから実行することができる。

4.2.2 ファイル管理

「ファイル管理」の部分では、ユーザの演習用ディレクトリの内容がグラフィカルに表示される。ディレクトリ、ファイルの作成、削除や、ファイルの移動などをマウス操作により実行することができる。

4.2.3 テキストエディタ

「テキストエディタ」の部分では、テキストエディタの機能が提供される。一般的なテキストエディタと同様に、プログラミング言語に応じたキーワードの強調表示や、括弧の対応関係の表示など、プログラミングの初心者をサポートする機能も備えている。

4.2.4 ターミナル

「ターミナル」の部分では、UNIX 環境のターミナルの機能が提供される。UNIX のオペレーションを行うために利用するほかに、コマンド操作によるソースファイルのコンパイル、実行を行うこともできる。ま

た、複数のターミナル画面が必要な場合は、エディタを表示する部分にターミナル画面を表示することも可能であり、タブの切り替えによってエディタとターミナルの表示を変更することができる。

4.2.5 拡張機能

「拡張機能」の部分では、IDE としての機能以外の拡張機能が提供される。例えば、ファイル編集の履歴を表示する機能がある。学生があるファイルを編集すると、その段階でのファイルの内容が履歴としてシステム上に保存される。教員は、学生から質問があったときに、その履歴を参照して、学生に指導を行うことができる。また、チャット機能もある。教員と学生との間でチャットを行うことで、ファイルの内容を確認しながらリアルタイムに指導を行うことができる。

5. プログラミング関連科目での運営

ここでは、goorm を利用した演習系科目のうち、プログラミングに関連する科目として、Java のプログラミングを学ぶ科目の運営について述べる。

5.1 科目内容と運営

この科目では、プログラミングの経験がない学生も考慮して、Java によるプログラミングを基礎から学ぶ科目の構成となっている。この科目の各回の構成を表 1 に示す。

表 1 Java プログラミング科目の構成

授業回	授業内容	課題
第 1 回～第 3 回	講義	小テスト
第 4 回	演習	レポート
第 5 回～第 7 回	講義	小テスト
第 8 回	演習	レポート
第 9 回～第 11 回	講義	小テスト
第 12 回	演習	レポート
第 13 回～第 15 回	講義	小テスト
期末試験	試験	—

この科目は、表 1 に示すように、12 回の講義回と 3 回の演習回で構成している。学期の前半(第 8 回まで)において、Java によるプログラミングの基礎を学び、後半(第 9 回以降)では、オブジェクト指向プログラ

ミングの基礎を学ぶ。講義回は 4 章のコンテンツで構成している。第 1 章では VoD 形式により各回に学ぶ概要を講義する。第 2 章～第 4 章では、WBT 形式によりコンテンツを提供している。学生は、コンテンツ内のテキストを読み進めると同時に、映像や音声による解説を視聴し学習を進める。各講義回の終わりには「小テスト」を実施し、その回における理解度を確認する。演習回は 2 章の WBT 形式のコンテンツで構成している。第 1 章で過去の復習を行い、第 2 章で Java によるプログラミングに関する演習問題を出題する。学生は演習問題に取り組み、結果を「レポート」として提出する。講義、演習内容についての学生からの質問については、先に述べた掲示板や電子メールで受け付ける。学期末には「期末試験」を実施し、全体を通しての理解度を確認する。

この科目内において、Java によるプログラミングの演習は goorm 上で行う。goorm では学生 1 人に対して 1 台の UNIX 環境が与えられる。学生は、自身の環境においてディレクトリ、ファイルの管理を行う。プログラミングの際は、goorm (図 1) 上の「ファイル管理」の部分でディレクトリ、ファイル管理を行いながら、「テキストエディタ」の部分でソースファイルの作成、編集を行う。作成したソースファイルは、「ターミナル」の部分において、コマンド入力によりコンパイル、実行を行う。

5.2 従来との比較

goorm を導入したことによる、この科目の運営について、導入前と導入後の比較を行う。

まず、演習環境の構築についてである。この科目では、goorm を導入する前は、学生が各自の PC 上に、Java によるプログラミングの環境を構築していた。具体的には、テキストエディタと Java SE Development Kit (JDK) をインストールし、パスの設定などを行っていた。すでに述べたように、学生の PC の OS 環境は統一されていないため、ダウンロードするファイル、インストール手順、設定の手順などが学生によって異なり、演習環境を構築するまでの指導に時間を要していた。goorm の導入により、学生は各自で演習環境を構築する必要がなくなり、ウェブブラウザ上でシステムへのログインの操作を行うことで、すべての学生に共通の

演習環境を提供できるようになった。これにより、学生は学期の初めから授業内容に集中できるようになった。担当教員も演習環境の構築のための指導時間が減り、授業運営に注力できるようになった。

次に、プログラミングの指導についてである。goormの導入前は、プログラミングに関する質問がある場合、学生はその都度、自身が記述したソースファイルを質問用の掲示板や電子メールに添付した上で、質問文を記述していた。担当教員は学生からの質問を受けると、添付されたソースファイルを確認し、質問に対する回答を行っていた。goormの導入後は、担当教員が学生の演習環境を閲覧することができる。そのため、学生はソースファイルを添付することなく、質問だけを行えるようになった。担当教員は、学生の演習環境を閲覧することで、該当のファイルだけでなく関連するファイルなども参照することができ、指導しやすくなった。また、学生が作成したファイルの履歴が保存されているため、学生の学習過程を閲覧しながら適切な指導を行えるようになった。

6. ウェブアプリケーション関連科目での運営

ここでは、goormを利用した演習系科目のうち、UNIXのオペレーションとサーバの運用に関する科目としてウェブアプリケーションを学ぶ科目の運営について述べる。

6.1 授業内容と運営

ウェブアプリケーション関連の科目では、ウェブアプリケーションのためのプログラミング学習のほかに、UNIXのターミナル上でのファイル操作と基本コマンドを学習する。プログラミングの学習方法については、5.1で述べたプログラミング科目の運営とほぼ変わらないため、ここでは説明を省略する。

UNIXのファイル操作やオペレーションについては、15回の授業のうち、8回を使用して行う。この8回分の授業内容では、学生は4.2.4で述べたgoormのターミナル機能を使用して演習を行う。

学生は1人に1台のOSを使用する。実態としては1学生に対してdockerの1コンテナが割り当てられた環境で実習を行う。そのため、学生は一般IDでは

なく管理者権限であるrootのIDを用いて演習を行う。

UNIXの基本コマンドは、大きく2つに分類して学習する。1つはlsやcdなどのファイルおよびディレクトリ操作に関するコマンドである。もう1つはmanやserviceなどのアプリケーションやシステムの操作に関連するものである。ここで、ファイル操作に関する演習では、dockerのコンテナ環境で問題はない。しかし、システムに関するコマンド、例えばpsコマンドではOS全体のプロセスを確認することはできない。演習ではこの点に注意して授業を行うが、これについては、その相違を説明することで回避でき、演習内容に影響はない。

6.2 従来との比較

次に、goormを導入したことで、従来演習との比較を行う。

一つ目の比較は、ターミナルの操作に関する比較である。従来演習では、複数のターミナルを使用するとき、ターミナルソフトウェアを立ち上げ、複数のウィンドウを操作しなければならなかったため、ウィンドウ操作に慣れない学生はどのウィンドウで何をしているのか、またどのウィンドウが画面上のどこにあるかわからない状態であった。しかし、goormではターミナル画面がエディタと同じところに表示され、複数のターミナルを表示しても、タブによる切り替えができるようになったため、操作がしやすくなり、ウィンドウの行方不明という状態がなくなった。

その一方で、エディタとターミナルが同じところに表示される。これにより初学者にはエディタとターミナルの区別がつかず、ターミナル上で入力するコマンドをエディタ上で入力したり、その反対のことはしたりするため「授業通りに動かない」という質問が従来よりも増えた。

二つ目はユーザ権限に関する比較である。従来授業では、学生は1つのOSに対し一般ユーザでアクセスして学習していた。そのため、OSの環境設定ファイルを修正したり、自分でパッケージソフトウェアをインストールしたりすることができなかった。しかしgoormの演習環境では、1学生に1つのコンテナが割り当てられrootの権限で操作できる。そのため、従来授業では行えなかったOSの環境設定ファイルの修

正や、パッケージソフトウェアのインストールができるようになり、授業内容の幅が広がった。その一方で、環境設定ファイルを間違えて変更することで、システム全体が正常に動かなくなることもある。この問題が発生することを想定して、現状では、必要に応じて学生自身が初期設定されたコンテナを複数複製したり、削除したりすることを可能としている。実際の運用では、学生が複数のコンテナを不用意に複製しないようにするため、複製方法について指導せず、必要に応じて教員が複製し、学生に新しいコンテナを提供するようにしている。

7. 評価と考察

goorm を導入することでソフトウェア関連の科目とウェブアプリケーション関連の科目において、学生が各自の PC にアプリケーションをインストールできないという問題は解決できた。また、1 学生に 1 つの OS を管理者権限で使用できるようになったため、ウェブアプリケーション関連の科目では、これまで指導できなかったシステム関連の内容を指導することができるようになった。

goorm を導入したことで、学生に対する学習効果や学生の学習意欲、学習継続性が向上したかどうかはまだ評価できていない。これは、サイバー大学では入試に学力試験を導入していないため学生の学力に大きなばらつきがあり、毎学期受講生の学力が一定していないためである。そのため、数学期にわたり受講状況や成績を集計する必要がある。

8. 今後の課題

goorm を使用した科目の学生の受講データはまだ 1 学期分しかない。今後は、数学期にわたり受講データを収集し、従来の受講状況や成績と比較し goorm を導入したことで、学生への効果が上がったかどうかを調査する。goorm では学生ごとにコンテナの利用時間と CPU の使用時間を収集できるため、詳細な演習環境の利用状況を把握することができる。

これらの情報を使用して、利用状況と学生の成績との関連、学生の利用状況による受講状況の把握などを行い、授業改善に利用することを検討することが課題

となる。

9. まとめ

本稿では、オンライン大学であるサイバー大学における授業の主な運用方法であるビデオを視聴する VoD 方式の授業とテキストを中心とした WBT 方式の授業形式について紹介した。そして、とくに技術系の演習科目であるプログラミングおよびウェブアプリケーション関連の科目についての紹介、そこで使用する演習環境 goorm について紹介した。goorm はウェブ上で提供される統合開発環境であり、また docker のコンテナ技術を用いて 1 人に 1 台の UNIX-OS を仮想的に提供するシステムである。goorm の導入により従来までの演習授業で課題となっていたいくつかの問題点の解決と、授業内容の拡張の可能性を示した。今後の課題として、goorm の利用状況などを収集し、学生の成績との相関や受講継続率との相関について調査する。また、これらの調査結果をもとに、授業の改善を行うことを示した。

謝辞

本学への演習環境の提供および演習環境の細やかな修正にご対応頂いた、goorm の開発元の codigm 社と日本での代理店である株式会社 PIC に感謝する。

参考文献

- (1) 後藤 幸功, 松田 健, 松本 早野香: “e-ラーニングによるリメディアル授業の実践と課題”, 情報処理学会第 74 回全国大会講演論文集, pp.4-549 – 4.550 (2012).
- (2) 後藤 幸功, 中谷 祐介: “オンライン大学における出席時間に関する考察”, 2016 年電子情報通信学会全国大会講演論文集, D-15-33 (2016)
- (3) 後藤 幸功, 中谷 祐介: “オンライン大学のプログラミング演習科目における受講状況と成績の関係に関する考察”, 情報処理学会第 78 回全国大会講演論文集, pp.4-525 – 4-526 (2016)
- (4) 中谷 祐介, 後藤 幸功: “オンライン大学におけるプログラミング教育の取り組み”, 2016 年電子情報通信学会全国大会論文集, D-15-34 (2016)