

シェルスクリプトを中心とすることで 高い普遍性を持たせた情報技術教育プログラムの設計と実装

森 祥寛^{1,a)} 松浦 智之^{2,b)} チャワナット ナカサン^{†1,c)} 大野 浩之^{1,d)}

受付日 2022年8月15日, 採録日 2023年1月19日

概要: 現代社会において, 情報技術教育は, コンピュータリテラシーから始まり, データサイエンス, 機械学習などさまざまな講義として実施されている. しかし多くの教育機関でほとんど行われていないのが, コンピュータ上でのシェルやコマンドの使い方やシェルスクリプトなどの活用方法である. そこで, われわれは, 2015年より POSIX 中心主義を踏まえた「シェルスクリプト」を土台にした教育プログラムの設計と実装の研究を開始した. 本稿では, この教育プログラムの設計について論じ, その実装である講義について報告する.

キーワード: シェルスクリプト, POSIX, 情報科教育, 教育実践報告

Design and Implementation of Information Technology Education Centered with High Universality on Shell scripts

YOSHIHIRO MORI^{1,a)} TOMOYUKI MATSUURA^{2,b)} CHAWANAT NAKASAN^{†1,c)} HIROYUKI OHNO^{1,d)}

Received: August 15, 2022, Accepted: January 19, 2023

Abstract: In modern society, information technology education is provided as various lectures such as computer literacy, data science, and machine learning. However, what is seldom done in many educational institutions is how to use shell commands and how to utilize shell scripts on computers. Therefore, our research group started research on the design and implementation of an educational program based on “shell script” based on POSIX centralism in 2015. In this paper, we discuss the design of this educational program and report on the lecture which is the implementation of the program.

Keywords: shell script, POSIX, information science education, educational practice report

1. はじめに

われわれは, 2015年度から USP 研究所と金沢大学総合メディア基盤センター^{*1}の大野, 森らとの間の共同研究「IoT時代に資するユニケージ [1] 開発手法の普及啓発に関する研究プロジェクト [2]」を出発点として, 「シェルス

クリプト」を土台にした教育プログラムの設計について研究を開始した. そしてその実装として, さまざまな講義などを実践してきた.

われわれは, 講義を通じて, 履修者にシェルスクリプトを用いたプログラミングによって, 電子工作から大規模データ処理に至るさまざまな情報技術に触れる機会を提供している. 履修者の多くはプログラミングなどに関する専門教育を受けていない大学生であるが, 彼らに対して, 情報技術に対する理解を深めるとともに, 最も根幹をなす基盤知識と技術を学ばせている. 現代社会では, 新しい技術やハードウェア, プログラミング言語, アルゴリズムなどが次々に登場している. その中で, 流行り廃りに影響されることのない (あるいは影響されることの少ない) 「情報技術」を学ばせること, そして, そのための教育プログラ

¹ 金沢大学学術メディア創成センター
Emerging Media Initiative, Kanazawa University, Kanazawa, Ishikawa 920-1192, Japan.

² ユニバーサル・シェル・プログラミング研究所
Universal Shell Programming Laboratory Ltd., Minato, Tokyo 105-0003, Japan.

^{†1} 現在, カセサート大学
Presently with Kasetsart University, 50 Ngamwongwan Rd, Lat Yao, Chatuchak, Bangkok 10900, Thailand.

a) mori4416@staff.kanazawa-u.ac.jp

b) t-matsuura@usp-lab.com

c) chawanat.n@gmail.com

d) hohno@staff.kanazawa-u.ac.jp

^{*1} 同所は 2021 年 4 月に改組し, 現在は学術メディア創成センター

ムを設計し、社会実装することが、われわれの研究目的である。

これにより、各種ソフトウェア・ハードウェアプロダクトの使い方といった表面的な学習に終始せず、むしろコンピュータの原理やアルゴリズム、技術仕様・規格といった普遍的な領域について深く理解した人材を育成し、彼らが情報産業界で長年にわたって活躍できること、そして彼らが効率的かつ可読性・保守性・持続性の高いコードを生産して社会基盤を構築できるようにすることを目指す。またこの取り組みは、情報技術教材そのものの普遍性を高めることにも効果的であり、教育現場における教材開発や教員教育の負担軽減もねらいの1つである。

本稿では、この目的達成のためにわれわれが実践した教育プログラムの設計と実装について論ずる。

2. 研究背景

2.1 背景

近年、情報技術はさまざまな分野で活用され、それらの技術を持つ人材が必要とされている。2010年代始めにはICTなどが、2010年代半ばからはデータサイエンスやDX、Society5.0などの言葉で表される情報技術を元にした社会の変革が求められてきた。2010年代初頭頃から電子工作を中心としたメイカーによる新たなものづくりの潮流は、新たな文化であるとしてクリス・アンダーソンによって名付けられた「メイカームーブメント [3]」が広がった。メイカームーブメントの広がりには、これからの情報技術が、ソフトウェアとハードウェアの両方の取り扱いが必要となることを示している。2020年代に入り、工業化社会 (Society3.0) から情報化社会 (Society4.0) への移行は完了し、社会全体にどのようなデータをどのように存在させるのか、そして存在するさまざまなデータをどのように活用するかが重要となっている。そのため教育機関では、情報技術に親和性を持った、さまざまな情報技術を持つ人材を育成する取り組みが必要となった。

これに応える形で、日本の初等中等教育では「教育の情報化 [4]」が、高等教育では「ICT活用教育の充実」などが継続的に実施されている。その中で情報技術に関しては、情報活用能力の育成が求められている。情報活用能力の議論は、1990年代の後半から続けられているが、2020年代の現在では、この能力は初等中等教育に限るものではなく、高等教育や社会全般においても必要なものとなっている。

日本の高等教育のICT活用教育の充実は、2000年代半ばから、各高等教育機関やその連携などによる組織的な取り組みが進められていった。金沢大学においては、学類^{*2}

入学時にノートパソコン [5] を準備させPCを必携としていることも、この中の取り組みの1つである。金沢大学の学生が、講義にパソコンを持ってくることが可能なことも、本研究実施の背景となっている。

2.2 シェルスクリプトを積極採用する理由

ユニケージ開発手法は、業務システム内製化の要求から、情報技術教育を十分に受けていない者であってもソフトウェア開発・改修・保守が担えるようにするための次のような原則 [6] を守りながら発展してきた。

- 高度な知識や複雑な操作技術が要求されるライブラリ・ミドルウェア等は利用しない。
- データには対しては可読性や一覧性を重視し、テキスト形式で処理し、ファイル・ディレクトリで格納する。
- システムを構成する各プログラムやサーバには単純な役割を与え、かつ疎結合させるように設計する。

このようなユニケージ開発手法の目的や原則は情報技術教育においても有効であると考え、われわれはこの考え方を取り入れた講義「計算科学特論」(4.1 2015年度参照)を実施することとした。

ユニケージ開発手法の原則は、UNIX哲学 [7] が示す「定理」の多くと考え方が一致しており、原則を守るうえで都合が良いため、開発プラットフォームとしてUNIX系OSを採用している。また、開発の主体となる言語についても同定理に従ってシェルスクリプトを採用している。

シェルスクリプトは、UNIX系OSを操作するUNIXシェルのコマンド文から構成されており、コンピュータ操作の自動化、アプリケーションの実行、さらにはアプリケーション同士の連携なども行える。また、テキストデータ加工 (部分抽出、文字列検索・置換、挿入、削除など) やファイル操作 (作成、名前変更、複製、移動、一覧表示、削除など) のためのコマンドがOSインストール直後から多数用意されており、後から自作してそれを実行することもできる。

これにより、シェルスクリプトは、互換性の高さや、可読性の良さ (専用ソフトウェアが不要なため)、処理の軽さ (低性能なコンピュータでも扱いやすい)、長寿命 (専用ソフトウェアの製品寿命に依存しない)、耐障害性 (専用ソフトウェアの調子に依存しない) というメリットを得られる。さらに、テキストファイル含む、ファイルを中心としたデータ管理が可能となる。特にシェルスクリプトが動くUNIX系OSでは、データの出入口もハードウェア (デバイス) も便宜上ファイルに見せかけて扱うため、デバイスからの、あるいはデバイスへのデータ転送が、ファイルの読み書きと同じ要領で実現できる。

このようなメリットにもかかわらず、本研究を開始した2015年から2022年の現在に至るまで、シェルスクリプト

^{*2} 金沢大学では学域学類制度を取っており、学類は学部に対応する。

はコンピュータを扱ううえで必要な技術であっても、シェル上でコマンドをシェルスクリプトに仕上げる方法を中心に教授するような講義はほぼ存在していなかった。これは国立大学協会に所属している国立大学法人のシラバスとして公開されている 2022 年度の講義内容からも明らかである。公開されているシラバスを確認すると、シェルスクリプトを学ばせることをあらかじめシラバスに記入している講義が、シラバスを全文検索できた大学 62 校のうち 37% に存在した^{*3}。高い割合に見えるが、確認できた講義数はわれわれの講義を含めて 45 講義であり、それらの講義で教授する内容のほとんどが、高水準言語を用いたプログラミングやアルゴリズムなどに関するもので、シェルやコマンド、シェルスクリプトについては、それらの講義の中で数回程度教えると書かれていることがほとんどであった。

一方で、MIT では以下のような講義「The Missing Semester of Your CS Education [9]」が行われており、そこでは「コンピュータに代表される計算システムは、手作業を自動化するようにつくられているが、多くの学習者は、繰り返し作業を手作業で行ったり、IoT 機器の操作やデータ処理などでコマンドラインインタフェースのツール、特にシェルを十分に活用できていない。」と述べ、行った講義を YouTube で公開している。この講義資料はボランティアのサポートを受けて各国語に翻訳されている。

2.3 POSIX 中心主義とシェルスクリプト

われわれは、POSIX が長期間にわたって変更されないことから 25 年後も使用可能なプログラムを作成する方法論についても導入を図った。

一般論として、ソフトウェアは以下の 3 点が満たされていることが望ましい。

- (1) 簡単かつ素早く開発できる
- (2) 完成後も長期にわたって必要な性能を維持できる
- (3) 開発後 10 年あるいは 20 年という長い時間が経過しても修正が必要になったら即座に修正できる

しかし、実際には時間の経過とともに、開発したソフトウェアを取り巻く OS やミドルウェアやアプリケーションの仕様や実装が変化するため、(2) や (3) の実現は容易ではない。そこで POSIX 環境下でシェルスクリプトを中心に使用し、プレーンテキスト形式のデータ処理を行うことで (1) から (3) を満たすことに成功したのが、著者の松浦が提唱した「POSIX 中心主義」に基づくプログラミングである [10]。これによって松浦は、多くの高水準言語で作られたプログラムやコードなどが、アップデートによって動かなくなってしまうことに対して、長期間使用可能なコードのプログラミング方法を提示した。

この POSIX 中心主義は、「POSIX 準拠」「交換可能性の担保」「W3C 勧告準拠」の三要素から成り立っており、これを踏まえることで、25 年後も使用可能なプログラムを作成できるとしている。ここで POSIX 準拠とは、プログラム作成時の基本指針として POSIX (IEEE Std 1003) 規格 [11] に極力準拠することである。特に問題がない限りはこの文書で記されている文法やコマンド、オプション、正規表現、保証される出力結果のみに依存したプログラムを書く。これは単独の POSIX 環境下の UNIX 系 OS 上で動かすプログラムや、クライアントサーバ構成をとるアプリケーションにおけるサーバ側プログラムが対象となっている。次に交換可能性担保とは、POSIX の範囲では実装が難しい処理を実現するための指針である。交換可能性とは、今利用している依存ソフトウェア（仮に A とする）と同等機能を有する別の実装（仮に B とする）が存在し、何らかの事情により A が使えなくなったときでも、B に交換することで A を利用していたソフトウェアを継続して使える性質として定義し、これを担保しうる可能性のことである。POSIX 中心主義では、交換可能性の担保を条件に POSIX の範囲外のコマンドの使用を認めている。最後に W3C 勧告準拠とは、クライアントサーバ構成をとるクライアント側アプリケーション開発のための指針である。2022 年現在、この構成においてクライアント側に用いられる主な技術は Web (HTML/CSS/JavaScript などの規格の集合) であり、POSIX 準拠の指針が適用できない。しかし、Web にも、下位互換を重視しながら、ブラウザ間の互換性が確保できるように多くのブラウザ開発団体が準拠している W3C 勧告があり、クライアント端末向けに Web プログラムを作成する場合は W3C 勧告への準拠に努めるという指針が設けられた。

われわれの教育プログラムでは、シェルスクリプトを中心に据えて設計と実装をしている。その意義は、履修者がさまざまな場面でシェルスクリプトを使用できるようになることである。このシェルスクリプトは POSIX 中心主義のもとで作られることを前提とした。POSIX 中心主義に従って作成したシェルスクリプトは、25 年後であっても使用可能となる。さらに POSIX に準拠した環境であれば、OS などを問わず、POSIX の範囲内で同じように使用可能となる。同時に、後述 (3.5.2 参照) のシェルスクリプト言語論 1 のような、シェルスクリプト自体を教えるための講義の場合や、シェルスクリプトを用いて大規模データ処理を演習する講義の場合などでは、POSIX 中心主義から外れる場合もある。

3. シェルスクリプトを中心とした教育プログラムの設計

3.1 教育プログラム設計目的と概要

われわれは、シェルスクリプトを積極採用することで、

^{*3} https://github.com/hohno-46466/rZone-POSIX_Education_Syllabus_Survey に調査結果ファイルを掲載

UNIX 哲学に基づく普遍的なシステム開発やプログラミングが可能な情報技術を身につけることが可能であるとして、以下の教育プログラムを設計した。

- (1) シェルやコマンドを使って IoT 機器などを扱うためのシェルスクリプトについて学ぶ教育プログラム
- (2) 大規模データを処理するためのシェルスクリプトについて学ぶ教育プログラム。なお、このプログラムは、社会におけるデータサイエンスの必要性の高まりに応じることと、共同研究者である USP 研究所が学生向けに構築したコマンド群 PT4A^{*4} (詳細は後述) の普及啓発も視野に入れている
- (3) プログラミング言語としてのシェルスクリプトについて学ぶ教育プログラム

すべての教育プログラムでは、POSIX 中心主義のもとでシェルスクリプトを使用できるようになることを目的としており、これによって、流行り廃りに影響されることのない (あるいは影響されることの少ない) 情報技術を身につけさせる。学習者は、コンピュータを使ってあらゆる種類の電子的なデータや情報を作成や処理などを行うために、POSIX 中心主義を前提としたシェルスクリプトの使い方を学ぶことになる。(1)では、メイカームーブメントの流れから、情報技術で取り扱うべき範囲としてソフトウェアとハードウェアの両方に対応するために、シェルスクリプトを用いて IoT 機器を始めとするセンサやアクチュエータなどを、ときにはインターネットでデータの送受信を行いながら扱う方法を学ぶ。(2)では、データサイエンスで必ず必要となるインターネットを通じて大量に入手できるさまざまなデータ類の取り扱い方法、特に表計算ソフトウェアでは処理しきれない量のデータを処理するために、大規模データ処理の方法を学ぶ。そしてこの(1)(2)を円滑に学ぶための前段として、(3)でシェルスクリプトを使ったプログラミング方法を学ぶ。

3.2 教育プログラム全講義共通の内容

われわれの教育プログラムで提示する全講義 (3.3 より提示) は、情報技術を身につけさせるための、いわゆる、情報教育としては、最も基礎的な知識と技術の学びとして想定している。このためわれわれの教育プログラムは、何らかの教育課程には組み込まれておらず、講義も教養科目 (金沢大学では共通科目) として実施しており、受講開始時点では、履修者にはパソコンに関する高度な知識や技能ではなく、使用可能なパソコンの準備とその使用能力を求めている程度である。

履修者が講義で学ぶシェルスクリプトを入力するターミナルは、WindowsOS や macOS など動いているさまざまなソフトウェアやアプリケーションと違い、グラフィカ

ルインタフェース (以下、GUI と言う) ではない。このためマウスを使った操作は行えず、入出力には、コマンドラインインタフェース (以下、CLI と言う) と呼ばれる文字が表示されるだけの画面に、キーボードから直接コマンドを入力する。履修者の多くは、CLI を使用したことがなく、コマンド入力や入力したコマンドの修正などの作業において、最初にマウス操作によるカーソル移動を試みてしまい、結果として正しい操作が行えず、CLI そのものへの不満を募らせることがある。

すべての講義では、最初に POSIX に準拠した UNIX 系 OS が使用可能な環境として、このターミナルを履修者が準備したパソコンに用意させる。パソコンが、WindowsOS であれば WSL (Windows Subsystem for Linux) をインストールさせ、macOS もしくは Linux であればそれ自体が POSIX 環境であるので最初から利用可能なターミナルを使用させることとした。

macOS のターミナルを使用する場合は、Homebrew [12] をインストールさせ、さまざまなコマンド群のインストールが可能になるようにした。これは POSIX 中心主義における交換可能性の担保を実現するために必要とされるもので、Homebrew のようなパッケージ管理ソフトによって、POSIX 文書にないソフトウェアやコマンドであっても導入できる環境を整えさせる。また、履修者は、われわれの提示している講義内容のためだけにパソコンを使用するのではない。さまざまな活動に併せて、さまざまな形でソフトウェアやコマンドを導入することがあるため、新しいソフトウェアやコマンドをインストールする方法を学ばせるためにも必要となる。

一方、WSL 上では Ubuntu を導入するのでコマンド管理には Ubuntu/Debian 環境で推奨されている apt コマンドを採用し、apt-get コマンドは非推奨とした。これは、講義や演習において、2 種類の微妙に異なるコマンドの混用を許すと、履修者に余計な誤解や混乱が生じることをこれまでの経験で得ていたからである。シェルには sh, csh, bash, zsh などがあるが、対話的利用では bash を、シェルスクリプトでは sh を用いることを原則とし、大規模データ処理などで後述の PT4A を用いる場合には bash を使用している。対話的利用とは、CLI の画面にコマンドを打ち込み、逐次その結果の出力を見ながら作業をする利用方法である。この場合に要請されるのは、作業効率向上である。sh では、ヒストリー機能が無かったり、入力間違いを訂正する機能が無いなど作業効率が悪いこと、対話的利用における操作自体はその一連のコマンド操作を記録し、後世に残す対象とはしないため、POSIX 中心主義の制約を課す対象とはせず、bash を使用する。また PT4A のコマンド群では、シェルスクリプトとして使用する際に、仕様として bash が求められているため、bash を使用する。bash を使用する場合は、必要に応じて最近の

^{*4} Personal Tukupai for Academic

macOS で採用されている zsh への読み替え方法を提供することとした。以上の対応により、履修者は、自身のノートパソコンで POSIX 環境下の CLI 端末を使用でき、パソコンの OS の種類によらず、シェル上の操作やシェルスクリプトプログラミングなどの作業を同じように実施可能になった。

講義では、その後、CLI による操作方法、コマンドライン入力とは何か、コマンド入力の方法、記号の読み方と入力方法などの説明を行う。コマンドライン入力は、UNIX コマンドによる処理がファイルの処理であることから説明する。これは前述の UNIX 哲学に基づく処理であること、その中のシェルスクリプトのコマンド群は C 言語などによるプログラミングの結果の活用にはほかならないこと、シェルスクリプトのコマンドを使って、データ処理を行う場合、その多くは、データファイル内に書かれたプレーンテキストを扱うことになり、直観的な作業が可能になることを説明する。そして、1 コマンドごとの結果を確認しながら、コマンドとコマンドを「| (パイプ)」でつなぐことで、段階を追った作業が可能となり、最終的にある特定のファイル処理を行うスクリプトが完成する。これは扱うデータの発生源や種類、規模が異なっても、同じような作業となる。ここまで行うことで、環境構築が完了する。

なお、これら環境設定やコマンド入力などに必要となる知識や技術については、できるだけ講義開始前までに作業や習得が完了していることが望ましいため、「ものグラミング」や「大規模データ処理」に関連した講義受講者のための参考情報^{*5}という Google スライドを作成し、単独でオンライン学習が可能な教材を準備して、講義開始までに事前に必要な作業と知識の習得をさせる。これらの作業は、履修者によってその進捗や作業内容に差ができるため、別途、作業進捗を書くための共有スライドやスプレッドシートを Google ドライブ上に用意し、履修者が相互に進捗確認しながら作業ができるようにしている。このスライドには、プログラミングで使用する記号の読み方や入力方法なども記載しており、学習できるようにしている。一方で、WSL のインストールについては、マイクロソフト社のアップデートによって、インストール方法がターミナルへのコマンド入力だけでできるようになり、作業内容がスライド 1 枚分のチャートシートの提示で収まるようになった。これらの作業によって、履修者の所持しているパソコンをできるだけ同じ状態で受講や演習をさせることが可能となり、講義開始時には補足的な説明を行うだけで良いようにしている。

教材への Google スライドやスプレッドシートなどを Google ドライブ上に用意する活用方法は、学習管理システムを介した教材提示を容易にするとともに、教材単独で

のアクセスも容易にする。学習ログなどを解析する学習分析の観点からは問題もあるが、履修者の所属が、一大学にとどまらない大学コンソーシアム石川いしかわシテカレッジ [13] などで講義を行う際には、教材提示方法として有効である。また、一部教材への編集権を履修者にも与えることによって、講義中の課題到達確認なども容易に行うことができる。本来は、学習管理システムを用いることが望まれるだろうが、履修者の属性などを勘案したとき、この方法が妥当であると判断した。

ここからは、教育プログラムごとの内容と講義設計について説明する。

3.3 IoT 機器などを扱うためのシェルスクリプト

3.3.1 IoT 機器などの取り扱いとデータ処理

メイカームーブメントは、電子回路を利用した電子工作を中心としたものづくりへの挑戦をしやすい環境を作った。一方で、これまでマイコンボードを用いた電子工作を完成させるために必須だったプログラミングが、インターネットへの接続機能などの付加により複雑さが増しており、この知識や技術の習得へのコストが挑戦への躊躇要因となっている。これを解決するために提供する方法が後述の「ものグラミング」であり、われわれは POSIX 環境下での「ものグラミング」によるプログラミング技術の習得する教育プログラムを作成した。

この教育プログラムでは、ものグラミングの考え方による IoT 機器などの取り扱い方法を学び、それらを扱うために必要なプログラミング技術の習得と、機器類との間で授受されるデータ処理の方法を学ぶことが目的である。この目的のために、履修者は、「ものグラミング」における IoT 機器などの取り扱い方法として、センサやアクチュエータなどを搭載した電子デバイスを POSIX 機 (POSIX に準拠した UNIX 系 OS を搭載したパソコンやマイコンなど) に接続することから始める。この教育プログラムでは、接続するセンサやアクチュエータなどは、主に Arduino [14] で制御される。履修者はそれらを使用するために Arduino IDE を使ってスケッチを書き、データの取得やアクチュエータの操作をする。取得するデータやアクチュエータの操作にはプレーンテキストを使用する。これによって、人間にも理解しやすいデータ送受信が実現できる。履修者は、さらにセンサで測定したデータを、MQTT [15] (mosquitto コマンド [16] を使用する) を使ってクラウドに向けて送信する方法を学び、センサが得たデータを Twitter でつぶやくことまでをシェル上でコマンドを使って行う。逆に、Twitter でのつぶやきの内容に応じて、Arduino を制御して LED を点灯させたり、アクチュエータを動かしたりする。履修者は、1 行のコマンドライン (ワンライナープログラム) や簡単なシェルスクリプトを書くことで、IoT 機器や、センサ、アクチュエータ

*5 <https://bit.ly/3ddj5hk> (2022 年 11 月 8 日アクセス確認)

などを操作できるようになることを目標とする。このため講義ではさまざまな IoT 機器やセンサ、アクチュエータなどを使用する。

3.3.2 ものグラミング

「ものグラミング^{*6}」は、もともとは趣味で電子工作を楽しむものづくり愛好家に、自分の作品とクラウド上のサービスを連携させるプログラムを「簡単」に「手早く」かつ「安全」に構成してもらうことを念頭に著者の大野が提案したプログラミング方法である [17], [18]。大野は、2022 年現在、ものグラミングについて、電子デバイスの選択や POSIX 機との接続方法や取り扱いによって、ものグラミング 1 から始まる 5 種類のものグラミングを提案している。われわれの教育プログラムでは、ものグラミング 1 とものグラミング 2 の範囲で IoT 機器などを扱うための講義を構築しており、以下ではグラミング 1 と 2 について述べる。

「ものグラミング 1」は、Raspberry [19] のような小さいが本格的な POSIX に準拠した UNIX 系 OS が動作し、さまざまなセンサやアクチュエータなどと接続が可能となったマイコンが誕生した過程で生まれた。生まれた当初の「ものグラミング 1」は、POSIX 機単体でセンサからデータを受け取った直後からの処理を前述の POSIX 中心主義に則ったシェルスクリプトで行う作業方法であった。この方法によって、IoT 機器などを取り扱う場合でも、高い互換性と長い持続性の確保を可能とした。

この考えを発展させて、ものづくりのためのプログラミングにおける適材適所と選択と集中を徹底させたのが「ものグラミング 2」である。「ものグラミング 2」では、センサやアクチュエータを操作する部分は、Arduino UNO のようなシンプルな組み込みマイコンに担当させ、それをセンサやアクチュエータの数だけ用意する。そのうえで、センサを担当するマイコンはセンサからの値をシリアルポート（主に USB 接続による）を介して POSIX 環境を使用できる Raspberry Pi やパソコンなどにデータを送るだけの作業を行う。アクチュエータであればその逆の流れとなる。これを端的に表したものが図 1 である。

このとき、この構成を実現するために POSIX 機上で実行するワンライナーは以下になる。

```
左側の矢印  cu -l /dev/tty... | command_A | ...
右側の矢印  ... | command_X | cu -s /dev/tty...
```

ここで利用している cu コマンドは 1980 年台から POSIX 機に用意されている、シリアルポート介して他の

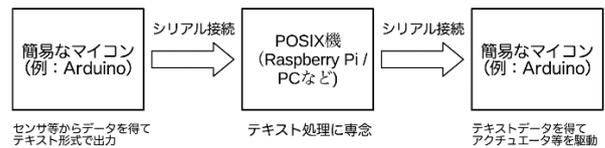


図 1 ものグラミング 2 におけるシリアルポートを介したデータの流れ

Fig. 1 The flow of data through serial port in Monogramming2.

システムに遠隔アクセスするターミナルエミュレータであるが、「ものグラミング 2」では標準入出力 (STDIN と STDOUT) をシリアルポートを接続するコマンドとして利用している。

どの「ものグラミング」の場合でも、このマイコン外とのデータやり取りにおいてはデータはプレーンテキスト形式で扱い、データの処理が必要な場合は POSIX 機 (Raspberry Pi やパソコンなど) 側で、POSIX 中心主義に則って行うこととし、センサやアクチュエータを担当するマイコンでは行わない。これを POSIX 中心主義の観点から見ると、センサやアクチュエータを取り扱う電子デバイスがどのように動いているかを POSIX 機側は知る必要が無く、POSIX 機に接続した組み込みマイコンは POSIX 機から見るとプレーンテキストを送受信してくるだけのデバイスに過ぎないからである。ものグラミングは、電子デバイスを POSIX 中心主義に基づいて取り扱えるようにするために電子デバイス側に課している条件設定であるともいえる。

これによって、扱うセンサやアクチュエータの数が増えても、容易に機能の切り分けができるようになる。これはセンサやアクチュエータだけの話ではなく、それを扱う人員にもあてはまり、グループで作業するときなど、自分が得意な分野を担当する形で役割分担をし、作業に貢献できるようにする。

3.3.3 講義：クラウド時代の「ものグラミング」概論

全 16 回からなる 2 単位の演習型の講義で、大学コンソーシアム石川いしかわシティカレッジで開講するため履修者の所属の幅が広がっている。講義の目的は、以下の 3 点に集約できる。

- センサやアクチュエータなどの簡単な操作方法の習得
- 「ものグラミング」の考え方を踏まえた 1 行のシェルスクリプトを用いてセンサから取得したデータをクラウドへの送信したり、クラウドに送信されたデータをパソコンやスマートホンで受信したりする流れを MQTT を使って実現する方法の習得
- 同様に、MQTT を使ってパソコンやスマートホンからクラウドを介してアクチュエータや表示器を制御する方法の習得

また、履修者の到達目標として

- ネットなどでも安価で購入可能な、さまざまな電子デバイスなどを動かすためのプログラミング環境を、自

^{*6} 「ものグラミング」の名称は、IoT 機器などの「もの」を扱うプログラミングであること、プログラムの多くが 1 行のコマンド形式で表記されるため「1 つ」を意味するギリシア語系の接頭辞「Mono-」で表わすことの 2 つからきている。

分のノートパソコン上に構築する。

- 構築したプログラミング環境で、「ものグラミング」の考え方に則したさまざまなプログラミングができる。
- 前述のプログラムを、徐々に発展させて、電子デバイスそのものをクラウド上の大量のデータやサービスと連携させ、クラウドと手元の小さな電子デバイス間で相互に影響を与える方法を身につける。
- 併せてインターネットを活用するさいに注意すべきセキュリティなどを説明できる。

を定めた。

履修者には、センサなどの機器を貸し出し、貸し出した機器を図2のように履修者のパソコンに接続して操作することから始める。

使用する機器については、「ものグラミング」の発展とともに表1のように変遷した。

最初期のRaspberry Pi以降は、「ものグラミング2」を前提に、Arduinoを中心とした機器を使用している。使用している機器の選択理由は、Arduinoに接続するセンサやアクチュエータが使いやすいこと、ドキュメントが整っていることである。センサやアクチュエータなどの操作を学ぶにあたって、機械的構造を知ることは重要である。しかしそのすべてを1つの講義で解説することは難しい。そこで、既存の良くできたドキュメントを講義でも参考にすることは有意義であると考え、講義で使用する機器を選定した。

使用するセンサやアクチュエータはArduinoで制御されるため、履修者のパソコンにArduino IDE [23]をインストールさせ、スケッチを書き、Arduinoでセンサやアクチュエータを制御させる方法を学ぶ。センサが取得したデータは、プレーンテキスト形式でシリアル接続をしたPOSIX機（履修者のノートパソコンなど）に送られる。履修者は、送られてきたデータをワンライナーまたはシェルスクリプトで制御する方法を学ぶ。この過程でmosquittoコマンドを使ったクラウドを介したデータの送受

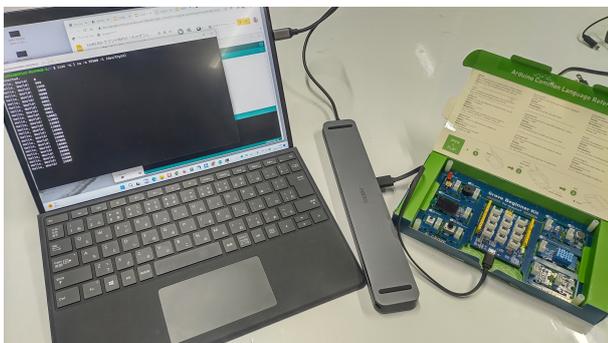


図2 センサをパソコンと接続し、実際に操作している様子

Fig. 2 The appearance of connecting a sensor to a computer and operating it.

信、その際にタイムスタンプや送信者名などをデータに付加する処理についても学ぶ。また、履修者には、kotori-otoko [24]を使って、コマンドラインからTwitterを操作する環境を与え、センサのデータをMQTTを介してTwitterに送って投稿やダイレクトメッセージとしたり、Twitter上のつぶやきを抽出してMQTTを介してアクチュエータの動作に反映させる手法を学ばせる。さらにこれら一連の操作をシェルスクリプトにまとめ、IoT機器などを取り扱う方法として習得させる。

3.3.4 集中講義：シェルスクリプトを用いた「ものグラミング」演習

金沢大学共通教育で実施する4日間全16回分の講義からなる2単位の演習型の集中講義である。集中講義のため、4日間で全16回の講義を実施する。実施目的、履修者の到達目標および講義内容は、クラウド時代の「ものグラミング」概論と同じであるが、集中講義という講義形態であるため、一定の期間中に集中的に解説と演習を行うことができる点が異なる。特に演習はできるまで行うことが可能となる。

3.3.5 講義：シェルスクリプトを用いた「ものグラミング」演習

金沢大学共通教育で実施する全8回からなる1単位の演習型の講義である。クラウド時代の「ものグラミング」概論の講義から、要素を抜き出して、8回の講義に組み直した。実施目的は、クラウド時代の「ものグラミング」概論で示した3つの目的と同じである。また、履修者の到達目標として

- ネットなどでも安価で購入可能な、さまざまな電子デバイスなどを動かすためのプログラミング環境を、自分のノートパソコン上に構築する。
- 構築したプログラミング環境で、「ものグラミング」の考え方に則したさまざまなプログラミングができる。
- 前述のプログラムを、徐々に発展させて、電子デバイスそのものをクラウド上の大量のデータやサービスと連携させ、クラウドと手元の小さな電子デバイス間で相互に影響を与える方法を身につける。
- 併せてインターネットを活用するさいに注意すべきセ

表1 使用したIoT機器など

Table 1 The used IoT devices, etc.

開講年度	使用したIoT機器など
2016～2017年度 (ものグラミング1)	Arduino, ESP-WROOM-02, Raspberry Pi
2018～2020年度 (ものグラミング2)	Arduino, エレキット SU-1203[20], SU-1204[21]
2021～2022年度 (ものグラミング2)	Seed studio Grove Beginner Kit for Arduino[22]

セキュリティなどを説明できる。

- POSIX 環境におけるシェルスクリプトについて、新しい視点から説明できる。
- 「Win/Mac/UNIX すべてで 25 年後も動く普遍的なプログラム」を書く方法を会得する。
- シェルスクリプトを日頃の問題解決に適用できるようになる。

と定めた点が異なっている。16 回の講義を 8 回にしたために、クラウド時代の「ものグラミング」概論に比べると講義内の演習時間が少ないが、「ものグラミング」に必要な要素はすべて伝えられる。

3.4 大規模データの処理とシェルスクリプト

3.4.1 データ処理と PT4A (Personal Tukubai for Academic)

ある特定のファイル処理を行うスクリプトでは、それを構成する各々のコマンドは、行える操作が限定されているため、行いたい作業を簡単には記述できなかつたり、記述できても処理が完了するまでに時間がかかることがある。そこで用意されたのが、USP 研究所が開発したエンタープライズ向けコマンド群 `uspTukubai` で、PT4A [25] はこれらをパーソナル環境で動作させるためのパッケージである。PT4A はアカデミック分野に対して、無償提供（利用者登録が必要）されていて、Windows10・11 の WSL, macOS, GNU/Linux (Ubuntu, CentOS) で利用可能となっている。PT4A で提供されているコマンドは、そのまま POSIX コマンド (`wc` など) と同じ機能を持つコマンドや、`awk`, `sed` コマンドを使用することで同じ処理は可能となるコマンドもある。しかし、使用する OS やハードウェアで微細な調整がなされていて処理速度が速いこと、`awk` や `sed` コマンドを使用するのに比べて簡単にスクリプトが書けることが、PT4A の優位点となる。

われわれの教育プログラムでは、この PT4A などを使用することで、シェルスクリプトで大規模なデータの処理を容易に行う方法を学ぶことが目的である。この目的のために、履修者は、処理する大規模データが保存された USB メモリ（後述）を POSIX 環境で処理できるようにするためのマウント作業から始める。マウントされることで、大規模データにアクセスが可能となるので、これから処理すべき大規模データがどのようなものかを示す。同時にデータは、シェルコマンドを使って処理することが可能となるので、基本的なシェルコマンドを使わせて、簡単な処理をさせる。PT4A の使い方としては、USB メモリ内にいくつかのサンプルスクリプトがあらかじめ保存されているため、講義内でサンプルスクリプトで行われる処理内容を解説するとともに、サンプルスクリプトの使い方を示し、履修者に処理をさせる。履修者は、サンプルスクリプトに手を加えて、アメダスデータから日本各地の気象情報

を分析するか、Twitter のつぶやきデータを分析するかして、レポートにまとめることを目標とする。

なお、われわれの教育プログラムにおける大規模データ処理では、最終的なレポートの作例として表計算ソフトウェアによるグラフを用いる方法を提示した。ただし、教育内容が特定の表計算ソフトウェアの使い方に偏ってしまえば、当初の目的である普遍性の高い情報教育とはいえなくなる。そこで、表計算ソフトウェアに依存する部分では、CSV などの汎用テキスト形式データ取り込み機能、棒グラフ・折れ線グラフといった基本的なグラフ作成機能だけに留める。

もし統計計算などが必要になる場合には、各表計算ソフトウェアが持つ組込み関数を使うのではなく、シェルスクリプトや UNIX コマンド上で計算プログラムを実装し、計算とテキスト形式へのデータ変換を済ませた状態で表計算ソフトウェアに与えるように指導する。こうすることで、教育内容の普遍性を維持できるとともに、履修者は計算アルゴリズムそのものについての理解を深める機会を得られる。

3.4.2 大規模データの準備

大規模データ処理を容易に行う方法を学ぶために、まずは処理を行うための大規模データを準備した。準備にあたって、われわれは、データサイエンス教育などを実施する場合、大規模データとして求められるのは定型データ（構造化データ）と非定型データ（非構造化データ）の 2 種類であり、その規模は現行の Excel で取り扱い可能なレコード数の上限 (1,048,576 行) を大きく超える規模と考えた。そして、シェルスクリプトを使って処理するためには、データファイルはプレーンテキストベース (TXT ファイルや CSV ファイルなど) で作られている必要がある。また、講義で使用するためには、履修者にデータを渡す必要があるが、その大規模データとしてのデータファイルサイズを勘案すると USB メモリに保存し、授業期間中貸し出すこととなる。一定程度の数量が購入可能で、貸し出し可能な USB メモリの容量としては 256 GB (2021 年末頃で約 3,000 円/本) の機種となり、ここからデータサイズの上限が決まる。そこでわれわれの教育プログラムでは、定型データとして気象庁アメダスデータ [26] を、非定型データとして Twitter のつぶやきを特定のキーワードで集めたデータを用いることとした。以下が定型データ、非定型データについての詳細である。

定型データ：気象庁アメダスデータ

気象庁では、アメダス^{*7}と呼ばれるシステムを運用しており、そのための観測施設を日本の各地 1,300 ヶ所に有している。ここで取得されたデータは、気象庁の Web

^{*7} AMeDAS : Automated Meteorological Data Acquisition System : 自動気象データ収集システム

ページから公開されており、誰でも取得可能である [27]. またこれらのデータは、気象庁の外郭団体である「一般社団法人気象業務センター [28]」からの一括購入も可能である. われわれの教育プログラムで使用する大規模データとしては、購入可能な 2008 年から 2021 年までの 1 分ごとの 1,320 地点のデータを、共同研究者である USP 研究所が購入した. これは圧縮されたバイナリデータで保存されており、1 年間で約 6 億 8000 万レコード、全部で約 90 億レコード分、1,420 GB 分のデータであった. これらのデータをテキストデータに変換し、2020 年までのデータは、10 分ごとのデータになるように間引き、2021 年のデータについては、1 分ごとのデータとした. 併せて、記録されているデータの項目も整理し、「観測所、時刻、雨量、気温、風速、風向、日照時間」という形で、1 年ごとにファイルを分割したデータセットを準備した. この結果、合計でレコード数 1,263,388,904、およそ 63.6 GB のファイルが用意できた. このうち、2018 年、2019 年、2021 年のデータについては、履修者が授業で演習に使用するパソコンの性能を考慮し、月ごとにデータを分割したファイルも用意した.

非定型データ：Twitter データ

Twitter でつぶやかれたつぶやきは、著者の 1 人である松浦によって作成されたコマンド群 *kotoriotoko* を用いて収集した. *kotoriotoko* では、特定のキーワードで検索をかけてつぶやきを収集することができ [29], われわれの教育プログラムでは、「オリンピック/パラリンピック」を中心にキーワードを設定したつぶやきを収集した. 収集期間は 2019 年 7 月 14 日から 2021 年 9 月 26 日までで、全 187,831,519 ツイート、およそ 82.8 GB の非定型データとなった. なお、非定型部分はつぶやきの文章部分であるが、つぶやかれた日時などについては定型化されている. このためデータ全体としては半構造化データの形をとっている.

表 2 シェルスクリプトを用いた「大規模データ処理」演習の講義の流れ

Table 2 Lecture outline on big-data processing exercises using shellScripts.

講義回	講義概要
第 1 回	ガイダンスと履修者のパソコン環境整備
第 2 回	CLI 入力と 簡単なシェルスクリプトを用いた演習
第 3 回・第 4 回	USB メモリ内の大規模データの内容紹介と PT4A の使い方
第 5 回・第 6 回	大規模データ処理の実践と分析例の提示
第 7 回・第 8 回	シェルスクリプトを用いて 大規模データ処理を演習

3.4.3 講義：シェルスクリプトを用いた「大規模データ処理」演習

金沢大学共通教育で実施する全 8 回からなる 1 単位の演習型の講義である. この講義では、履修者にノートパソコンを準備してもらい、そのパソコンを使ってシェルスクリプトによる大規模データ処理の演習をさせる. また、履修者の到達目標として

- 自分の使用している携帯型パソコン上でシェルスクリプトを使えるようになる.
- シェルスクリプトを用いて、大規模データ処理ができるようになる.
- 併せて、処理する大規模データを取得する方法も学び、必要に応じて取得できるようになる.
- シェルスクリプトを日頃の問題解決に適用できるようになる.

と定めた. 講義内容は表 2 のとおりである.

講義では用意した大規模データを、250 GB の USB メモリに保存して履修者に貸与する. これは使用するデータが大規模であり、ダウンロードし自らのノートパソコンに保存させることは難しいと判断したからである. シェルスクリプトを使用する環境として Windows10/11 の場合は WSL 上に Ubuntu を導入してターミナルを起動し、macOS/Linux の場合は OS に搭載されているターミナルを使用する. 講義第 1 回、第 2 回で、POSIX に準拠した UNIX 系 OS が使用可能な環境を構築、PT4A のインストール、WSL における USB メモリのマウント/アンマウント方法を行う. 第 3 回、第 4 回で、USB メモリ内の大規模データの内容の紹介と、実際のデータの閲覧方法を通じて、基本的なコマンド操作方法を解説、実践する. コマンド操作法の解説では、教員が解説をしながら、履修者の目の前でスクリプトを構築していき、順次、データ処理方法を教授する. さらに、シェルはインタプリタであるので、シェルの操作結果を逐一見せながら操作を継続し、スクリプトとして完成させる過程を見せることで、他の高水準言語より分かりやすい形でデータ処理が行えることを示

```

左側の文字列が時刻、右側の数字がその時刻に集まったツイート数です。
count 1 | というのは、「1列目から1列目までの文字列が一致する行の行数を数えよ」という意味です。今回の例もまた、開始列と終了列が同一なので不自然に見えますが、複数の列に渡って同一な行数を数えたい場合に役に立ちます。
さて、今数えた結果についても、後で利用できるように、1列目にある数字以外の文字を排除（fsedコマンドの使用）をして、さらにファイルに保存しましょう。
$ find . -name '*.txt' | xargs cat | self 1 | fsed -e 's/[0-9]//1' >Tokyo2020tw_sec-n.txt
$ find . -name '*.txt' | xargs cat | self 1.1.16 | count 1 | fsed -e 's/[0-9]//1' >Tokyo2020tw_min-n.txt
〇数えたファイルを細切れにする。
1単位のツイートを数えた場合ならファイルの行数はただか数百行です。
一方、秒単位で数えた場合、365日×12か月×30日×24時間×60分×60秒、かなりのオーダーの行数ですから、そのまま計算ソフトに渡そうものなら、表計算ソフトはフリーズするか強制終了するでしょう。そもそも、そのような大量の行のデータでグラフを描いてもまともには読めません。
もし秒単位にツイート数を数えたデータからグラフを書くのであれば、現実的にはどこか1日（0時から24時まで）のグラフにすることになるでしょう。そこで、数え上げたデータの中から、特定の期間を抽出する方法を解説します。
秒単位に数えたファイルTokyo2020tw_sec-n.txtが既にあり、ここから2020年7月24日のツイートのツイート数だけを抽出したいとしましょう。その場合は次のようなコマンドでできます。
$ awk '$1==20200724000000 && $1<20200725000000' Tokyo2020tw_sec-n.txt
（結果が出るまで時間がかかるかもしれませんが）
awkとはAWKコマンドの無難な省いて高速化したTukubaのコマンドです。
awkコマンドの速い方を知りたい方は検索をお願いします。

```

図 3 非構造化データを処理するためのチュートリアルの一部
Fig. 3 A section of the tutorial for processing unstructured data.

す。第5回以降は、USBメモリ内に添付した、サンプルスクリプトを用いたデータ分析例（図3）の解説と、実際に分析をした結果を提示する。この講義を通じて、履修者は自分のパソコンを使って、処理時間はかかっても簡単に大規模データ処理ができることを知り、サンプルスクリプトの活用などの演習によって、シェルスクリプトを用いた大規模データ処理手法を習得する。

3.4.4 集中講義：シェルスクリプトを用いた「ものグラミングと大規模データ処理」演習

金沢大学共通教育で実施する全16回からなる2単位の演習型の集中講義である。集中講義のため、4日間で全16回の講義を実施する。実施目的、履修者の到達目標および内容は、シェルスクリプトを用いた「ものグラミング」演習およびシェルスクリプトを用いた「大規模データ処理」の内容を合わせた内容になる。集中講義であるため、短期間に集中的に解説と演習を実施できる。特に演習は講義時間中に履修者にできるまで行わせることも可能である。

3.5 プログラミング言語としてのシェルスクリプト

3.5.1 POSIX 中心主義プログラミングの優位性

POSIXに準拠したシェルやコマンド群をできる限り用いて、プログラミングを行う「POSIX中心主義」を踏まえれば、25年後であっても使用可能なプログラムを作ることが可能となる。そこでわれわれの教育プログラムでは、シェルやコマンド群の活用方法と、そこからシェルスクリプトを組んでいく基本的な手法を学ぶこと、さらにその際にPOSIX中心主義を踏まえたプログラミング方法を学ぶことを目的とする。

3.5.2 講義：シェルスクリプト言語論1

全16回からなる2単位の演習型の講義で、大学コンソーシアム石川いしかわシティカレッジで開講するため履修者の所属の幅が広がっている。そこでわれわれは学生や社会人聴講生が、履修しやすいように開講時間をシティカレッジで最も遅い、19:10から20:40までの時限に設定し、web会議システムを使用したライブ配信によるオンライン講義とした。これによって社会人などの聴講をやすくし、県外からの履修者にも広く門戸を開いた。この講義では、プログラミング言語としてのシェルスクリプトの基礎を学び、実際に使用するための方法を学び、さらに自分でコマンドを作成するための方法を学ぶ。そこで、履修者の到達目標として

- POSIX環境におけるシェルスクリプトの背後にある「思想」を説明できる。
- POSIX環境におけるシェルスクリプトを自分のパソコンで使用する環境構築ができる。
- POSIX環境におけるシェルスクリプトとはどのようなものか、その仕組みと得手不得手を説明できる。
- 「Win/Mac/UNIXすべてで25年後も動く普遍的なプ

表3 シェルスクリプト言語論1の講義の流れ

Table 3 Lecture outline on ShellScript language theory 1.

講義回	講義概要
第1パート 第1回 第2回 第3回	ガイダンスとシェルスクリプトについて シェルスクリプトは何のために使うのか シェルスクリプトと実社会の繋がりと、 今後の展望
第4回	POSIX環境の整備
第2パート 第5回 第6回 第7回	CLIの初歩 コマンド操作の基礎 リダイレクションとパイプ
第3パート 第8回 第9回	自分のコマンドを作る コマンドは、何のために作るのか?
第4パート 第10回 第11回 第12回	テキストエディタ・正規表現 シェルスクリプトから使う 「リトルランゲージ」 シェルスクリプトに仕立てる方法
第5パート 第13~16回	自分なりのコマンド作成演習

ログラム」を書くことができるようになる。

と定めた。講義回と講義概要は、表3のとおりである。

この講義は、全16回の講義を5つのパートに分けた。

第1パートでは、履修者は、シェルを用いたプログラミングで実際に何ができるのか、そのメリットとデメリットなどについて学習する。同時に履修者が扱っているパソコンにPOSIXに準拠したUNIX系OSが使用可能な環境の構築をする。さらに、シェルスクリプトで何ができるようになるかについての理解を深めるため、実際にシェルスクリプトを使ってビジネスをしている方に講演してもらい、ビジネス現場におけるシェルスクリプトの利用について解説する。これによって履修者に、シェルスクリプトを身につけることによって何ができるようになるかを明確にイメージさせる。

第2パートでは、履修者は、第1パートで整えたPOSIX環境を使って、具体的なシェルスクリプトの使い方とシェルスクリプトで実際に何ができるのかを演習とともに学習する。GUIではないCLIについての説明から始め、履修者はシェルの入力とシェルスクリプトの構築、実行方法を学ぶ。そのために17個のシェルの基本コマンドについて解説し、実行したいことを実行するために、逐次、コマンドを伸ばしてスクリプトを組む手法について演習を行う。履修者は、回を追うごとにコマンドラインが伸びていくことで、シェルスクリプトの仕組みと意義、意味を知ることができる。また新しいコマンドの導入方法についても解説し、その例として、aptコマンドを使ってイン

ストール可能な `sl` コマンドや `calc` コマンドをインストールさせ、さらに `apt` コマンドではインストールできない、Twitter のつぶやきなどをコマンドラインから行える `ko-toriotoko` のインストールを実施した。

第3パートでは、履修者は、シェルのコマンドを作成する方法について学ぶ。さまざまなコマンドを使用してシェルスクリプトを組み上げる作業を行う中で、使用しているコマンドを自分で作成できることを知り、自分でコマンドを作成する際の方法や注意点を知ることで、シェルスクリプトによるプログラミングをより効果的に行えるためである。

第4パートでは、履修者は、ターミナルで使用可能なテキストエディタの種類と使い方について学ぶ。またシェルスクリプトを構成するコマンドの中から、重要なものについて、その機能と使い方を具体的な演習とともに学ぶ。特に `sed` コマンドや `awk` の使い方をていねいに説明することで、履修者のできるようになることの幅を広げる。

第5パートでは、履修者に、これまで学習してきた内容を用いて、自分なりのコマンドを作成するという課題を与えた。コマンド作成作業では、いきなりコマンドを作成するのではなく、コマンドのマニュアル (`man` ファイル) を作成させ、作成したいコマンドの内容について議論をした。その後実際にコマンドを作成し、そのコマンドについて評価をした。

表4 シェルスクリプト言語論2の講義の流れ
Table 4 Lecture outline on ShellScript language theory 2.

講義回	講義概要
基礎編	
第1回	ガイダンス, POSIX とは何か
第2回	演習環境の整備
第3回	シェルスクリプトとは何か, 簡単な演習
第4回	POSIX 中心主義
第5回	POSIX 準拠とは (コンピューターの標準語と方言)
第6回	シェルスクリプトの落とし穴
実践編	
第7回	時刻の加減算はどうやるのか
第8回	ストリーミング型プログラミング
第9回	データベース管理ソフトに頼らぬデータ管理
第10回	Web ブラウザーの生のやり取りとは?
応用編	
第11~12回	デバッグとバージョン管理
第13~14回	実習: アプリ作成
まとめ	
第15~16回	まとめと総括

3.5.3 講義：シェルスクリプト言語論2

全16回からなる2単位の演習型の講義で、大学コンソーシアム石川いしかわシティカレッジで開講するため履修者の所属の幅が広がっている。この講義は、シェルスクリプト言語論1を開講するまでシェルスクリプト言語論という講義名で実施してきた、プログラミング言語としてのシェルスクリプトの応用的な使い方、特にPOSIX中心主義に基づくプログラミングを学ぶ、履修者の到達目標としては、

- POSIX 環境におけるシェルスクリプトについて、新しい視点から説明できる。
- 「Win/Mac/UNIX すべてで25年後も動く普遍的なプログラム」を書く方法を会得する。
- シェルスクリプトを日頃の問題解決に適用できるようになる。

と定めた。講義回と講義概要は、表4のとおりである。

シェルスクリプト言語論2では、全16回の講義を4つのパートに分けた。基礎編では、履修者は、POSIXとシェルスクリプトについての基本的な知識と、POSIX中心主義とそのプログラミング方法について学んでいく。実例編では、履修者は、POSIX中心主義に準じた具体的なシェルスクリプトによる簡単なプログラミング演習をする。応用編では、履修者は、アプリケーション作成に向けてデバックとバージョン管理方法について学び、あらかじめ準備した郵便番号から住所を検索するアプリケーションを写経(学習者にあらかじめ図4のようなプログラミング例などが印刷された教材^{*8}を配布し、学習者はそれを見ながらプログラムを作成していく作業。学習者にコピー&ペーストは行わせない)を通じて作成する演習を行う。最後のまとめでは、履修者は、これまでのまとめと演習などで作成した内容についての総括を行う。

この講義では、POSIXに準拠したUNIX系OSが使用可能な環境としては、教員側でPOSIXサーバを用意し

```

Prog.2 : 事業所住所早見表 (mkdcj_jigyosyo.sh) その6/7
126 # --- 郵便番号別ファイル作成 -----
127 if type unisp >/dev/null; then #
128 unisp -p $tmpf_sipcavsp #
129 #
130 elif type gunisp >/dev/null; then #
131 gunisp <$tmpf_sipcavsp 2>/dev/null || { #
132 error_exit 7 "No Zip archive extractor found (unisp)" #
133 } #
134 #
135 else #
136 error_exit 8 "No Zip archive extractor found (unisp or gunisp)" #
137 #
138 # 日本郵便 郵便番号-住所 csvデータ (shiff_710) #
139 if type iconv >/dev/null 2>&1; then #
140 iconv -c -f CP932 -t UTF-8 #
141 elif type nkf >/dev/null 2>&1; then #
142 nkf -o=iso #
143 #
144 else #
145 error_exit 9 "No KANJI converters found (iconv or nkf)" #
146 #
147 # 日本郵便 郵便番号-住所 csvデータ (UTF-8変換済) #
148 # : : 行番号 : 列番号 : csvデータセルデータ #
149 # : : 行番号 : 列番号 (郵便番号, 4-郵便局番, 5=市区町村, 6-7=町) : データ #
150 awk 'BEGIN{is="#"; ps="generated"; cm="at"; ts="timestamp_web"; } #

```

図4 あらかじめ準備した郵便番号から住所を検索するアプリケーションの一部

Fig. 4 A section of the application for searching addresses from a pre-prepared postal code.

*8 <https://richlab.org/j/5dc2> (2022年11月8日アクセス確認)

て、履修者に対してアカウントを発行し、SSHにてログインさせる形を取る。そのため演習環境を整備も、SSHで教員側のPOSIXサーバにログインするための環境構築となる。なお、講義中で、履修者に対してPOSIXに準拠したUNIX系OSが使用可能な環境設定についての解説は行う。これによって、履修者のパソコンにはターミナルエミュレータをインストールするだけで済むようになった。

ターミナルエミュレータは、当初、TeraTermを用いていたが、2021年度からはコードエディタであるVisual Studio Code (以下、VSCode) をインストールさせ、VSCode内でSSHなどの拡張機能をインストールさせることでターミナルの代わりとしている。VSCodeは、オープンソースで開発され、国際化が行われており、多種多様なプラグインが揃っているという特徴があり、著者らの調査の範囲では、英語圏の多くの大学でも積極的に採用されているといった特徴を有し、今後長期間にわたって国内外で利用されていくと考えられる。これがVSCodeを採用した理由である。

この環境構築方法は、クラウド時代において、インターネットを介してサーバを操作することが前提となるであろう社会においては、ローカル環境である学生のPC環境を整える作業を最小限にしながら、一般的なPOSIX環境下のサーバを扱うことを学べるという点で、優位な方法と考えた。また、履修者のパソコンへのインストール作業などを最小限にすることも、講義の主旨以前の内容で履修者が躓いてしまう可能性を減少させることができる。

同時に履修者の所有しているパソコンの性能が極端に低い、あるいは古い場合を除き、パソコンの性能によらず同等の学習ができる点もメリットである。併せて、POSIXサーバを教員側が管理できるため、履修者の進捗状況の把握がしやすく、必要に応じてサポートもしやすい。特にこの講義では、写経を有効な学習方法と位置づけており、その進捗管理やデバックサポートなどがしやすい。

従来は履修者から演習上の質問を受けた場合に履修者のパソコンの画面を覗き込まなければならず、場合によっては履修者のパソコンを教員が操作するために教員の指がキーボードのキートップやポインティングデバイスに直接触れることもあったが、この方法では教員のパソコンからログインして診断・指導できる。これにより、対面講義であっても感染症対策としてのソーシャルディスタンスを確保できるようになった。また、Web会議システムを用いたオンライン講義においても、参加形態によらず同等の教育を受けることができる。これは、今後、対面以外の講義形態が定着していくと考えたときの重要な要素であり、国境を越えた国際的な教育展開時にも必要となる。

一方でデメリットも存在する。履修者は講義期間中に学習のためにPOSIXサーバにアクセスしているため、講義期間が終了すればサーバ自体にアクセスできなくなる。この講義では、ある程度の期間、アクセスを許可しているが、それは永遠に続くわけではない。従って、履修者は、講義終了時に、講義中に行ったさまざまな作業による成果物などを手元にコピーしておく必要がある。また、履修者数が増えたときに機能不全に陥ってしまう。履修者が10名、20名であれば、同様に講義を実施できても、100名、200名、さらに1,000名以上になれば、それぞれのスケールで対応方法を検討しなくてはならない。

4. これまでの教育実践と報告

4.1 国内での実装

前節で論じたわれわれの教育プログラムは、実際にシェルスクリプトを活用した講義を実施する中で、その内容の改善を図っている。ここからは、2015年度後期から2022年度まで開講してきた講義について、時系列に沿って、年度ごとに行われた前節講義内容の補足と改善点を述べる。

まず、講義の種類と開講時期は図5のようになる。なお、履修者の到達目標については、毎年見直しをかけてい

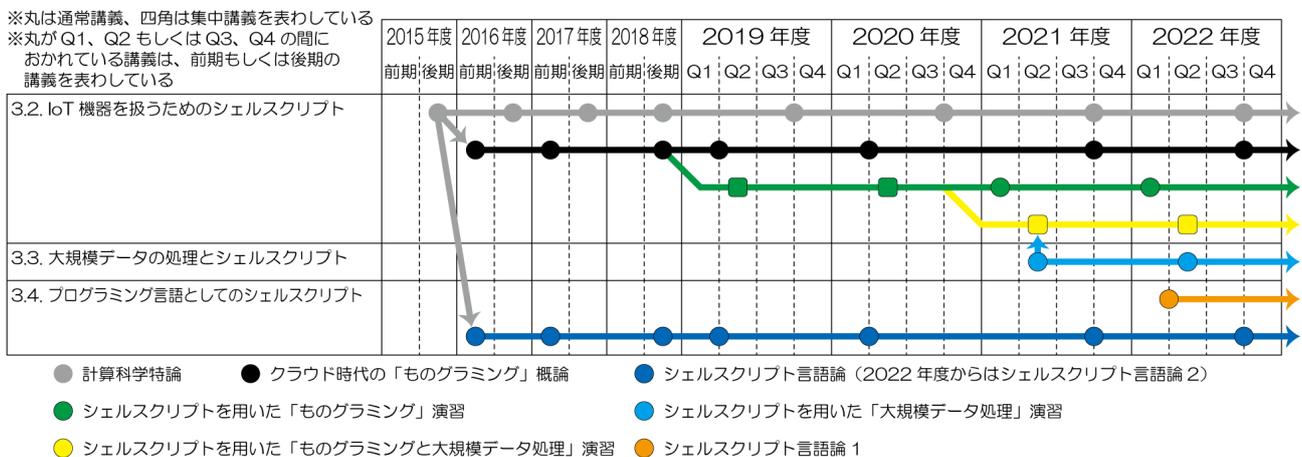


図5 これまでの教育実践による講義の流れ

Fig. 5 The flow of lectures based on previous educational practices.

るが、特に大きな変更は行われていない。

2015 年度

後期に、大野が担当している講義「計算科学特論」（金沢大学数物科学類 3 年生向け）で、われわれの教育プログラムの試行をしたのが、この研究の出発点である [30]。この講義で、われわれの研究グループとして初めて、Raspberry Pi を教材として用い、UNIX とコマンドラインインタフェース、UNIX コマンドで操作するハードウェア、UNIX コマンドで操作するネットワーク環境、UNIX コマンドとシェルスクリプトなどについて、学生が所持しているノートパソコンを使った演習を踏まえて実施した。なお、計算科学特論は、2022 年現在まで、内容を更新しながら継続して実施している。

2016 年度

前期より大学コンソーシアム石川いしかわシティカレッジにて、「クラウド時代の『ものグラミング』概論」と「シェルスクリプト言語論」を開講した。この 2 つの講義が、われわれの教育プログラムによる教育の本格的な開始である。

講義を開始時点で Windows10 がリリース（2015 年 7 月 29 日）されていたが、多くの履修者は、まだ Windows7 を使用していた。そのため POSIX に準拠した UNIX 系 OS が使用可能な環境として、履修者には Cygwin をインストールさせた。なお、履修者の使用するパソコンの OS が Windows10 に切り替わることを前提に、2016 年 8 月 2 日の Windows10 アップデートに伴いリリースされた WSL（Windows Subsystem for Linux）のベータ版を、履修者の UNIX 系 OS が使用可能な環境に使用することを検討開始した。

2017 年度

前期に大学コンソーシアム石川いしかわシティカレッジで「クラウド時代の『ものグラミング』概論」と「シェルスクリプト言語論」を開講した。

この段階では、POSIX に準拠した UNIX 系 OS が使用可能な環境として、WSL は選択肢の 1 つという位置づけであった。履修者は自身の環境に応じて、Cygwin か WSL をインストールした。

2018 年度

後期に大学コンソーシアム石川いしかわシティカレッジで「クラウド時代の『ものグラミング』概論」と「シェルスクリプト言語論」を開講した。この年度から「ものグラミング」は、「ものグラミング 2」に発展した形で講義「クラウド時代の『ものグラミング』概論」に盛り込まれ、使用する機器が表 1 のように Arduino とエレキットとなった。

この段階では、POSIX に準拠した UNIX 系 OS が使用可能な環境として、WSL を WindowsOS を使用している履修者のほぼ全員が使用できるようになった。

2019 年度

前期に大学コンソーシアム石川いしかわシティカレッジで「クラウド時代の『ものグラミング』概論」と「シェルスクリプト言語論」を開講した。金沢大学の学生に対して、金沢大学共通教育で、夏休み期間中の集中講義「シェルスクリプトを用いた『ものグラミング』演習」を開講した。集中講義として開講したのは、受講形態の違いによる講義内容および実施方法を検討するためである。

2020 年度

前期に大学コンソーシアム石川いしかわシティカレッジで「クラウド時代の『ものグラミング』概論」と「シェルスクリプト言語論」を開講した。本講義も、2020 年 5 月 14 日に開始日を変更した上で、Web 会議システムを用いたオンラインによるライブ配信型講義として実施した。なお、2020 年 7 月末には講義実施方法指定の指示緩和がされ、オンラインに加えて対面講義も行うハイフレックス形式で講義を実施した。金沢大学共通教育で、夏休み期間中の集中講義「シェルスクリプトを用いた『ものグラミング』演習」をハイフレックス形式で開講した。オンラインで参加する学生には、あらかじめ貸与機器などを郵送した。

2021 年度

金沢大学共通教育で、Q1^{*9}に「シェルスクリプトを用いた『ものグラミング』演習」を Q2 に「シェルスクリプトを用いた『大規模データ処理』演習」を開講した。夏休み期間中の集中講義「シェルスクリプトを用いた『ものグラミングと大規模データ処理』演習」を開講した。なお、金沢大学では、学生に対してデータサイエンス教育を行うために、データサイエンス特別プログラムを実施している [31]。われわれは、この特別プログラムに協力し、これら 3 つの講義を提供した。後期に大学コンソーシアム石川いしかわシティカレッジで「クラウド時代の『ものグラミング』概論」と「シェルスクリプト言語論」を開講した。

この段階では、POSIX に準拠した UNIX 系 OS が使用可能な環境として使用している WSL の機能が整えられ、インストール作業などが容易になった。

2022 年度

前期に大学コンソーシアム石川いしかわシティカレッジで「シェルスクリプト言語論 1」を開講した。後期に「クラウド時代の『ものグラミング』概論」と「シェルスクリプト言語論 2」を開講した。シェルスクリプト言語論 2 は、これまで実施してきたシェルスクリプト言語論の講義内容のまま改称した講義である。金沢大学共通教育で、Q1 に「シェルスクリプトを用いた『ものグラミング』演習」

^{*9} 金沢大学では、大学コンソーシアム石川シティカレッジ提供科目などを除いて、講義実施区分としてクォータ制（全 8 回 1 単位講義）を採用している。Q1 は年度開始から数えて 1 番目のクォータ、Q2 は 2 番目のクォータの意味である。

習」を Q2 に「シェルスクリプトを用いた「大規模データ処理」演習」を開講した。夏休み期間中の集中講義「シェルスクリプトを用いた『ものグラミングと大規模データ処理』演習」を開講した。

4.2 国内での 2022 年度前期における運用結果

2022 年度前期開講の講義「シェルスクリプト言語論 1」の履修者は全 14 名であった。履修者の内、学生 7 名（内金沢大学学生は 2 名、5 名は石川県内別大学生）、社会人聴講生 7 名で、学生と社会人という異なる属性で半分に分かれた。この結果、講義中の講師からの質疑などへの回答が、学生からは理論的な視点から、社会人からは実務的な視点からのものが多くなった。講義では、こういった視点の違いを利用して、履修者への「あなたにとってのシェルスクリプトとは？」といった問いかけに対して、質疑応答の内容の幅が広がり、履修者を講義内容についてより深い理解に導くことが可能となった。

また履修者からの「コマンドの作り方を学びたい」という意見を受けて、コマンドそのものを作成する講義を、全体の講義内容を調整しつつ 2 回実施した。この講義では、当初、全 16 回の講義を 4 つのパートに分けて実施する計画であったが、この質問を受けて 5 つのパートに分けることになった。このコマンド作成を学びたいという意見は、講義における最終課題にも影響を与え、履修者には最終課題として「自分で何らかのコマンドを作成する」を課した。講義では、最終課題のほかにも、表 5 にあるような課題を課し、適宜進捗状況を確認していった。

コマンド作成課題の後、学生の履修者から「私にとって初めてコマンドを作るのは想像より難しく、複数人の受講者にとって講義 3 回分のサポートのみでは間に合っていないように感じました」という意見が出された。また、講義内容全体に対して「講義スライドでは参考にするるとよいページなどが示されていましたが、特にオプションや引数を扱えるようにするのは難しく感じました。そのため、講

表 5 シェルスクリプト言語論 1 の課題例

Table 5 Examples of assignments for ShellScript Language Theory 1.

	課題例
基礎編	プログラムとプログラミング言語と OS の関係とは、シェルスクリプトは何のためにあるのか、テキストファイルとは何か、シェルスクリプトは実際にどのような現場で用いられているのか、「ユニケー手法」が、紹介した各事例に、効果をもたらした本質的な要因は何か、「グルーテクノロジー」とは何か。
実践編	知っているコマンドをできるだけ挙げる。実際に使ったことのあるコマンドを挙げる。正規表現の練習問題

義内でオプションや引数を扱ってみる演習を行いサポートをすれば、スムーズにコマンド作りに移れるのかなと思いました」という意見もあり、シェルスクリプトやコマンドを初めて用いる学習者には、まずコマンドのオプションや引数の扱い方の演習がより必要であることが分かった。

2022 年度夏休み期間中に開講した集中講義「シェルスクリプトを用いた「ものグラミングと大規模データ処理」演習」の履修者は全 19 名であった。集中講義であるため、学生からは「はじめは授業で何について話されているのか、よく分からなかったのですが、3 日目くらいから wsL が何かが何となく分かってきて、最終的には wsL で基本的なコマンド (less, ls, sl, data) などを使えるようになりました」というような反応が返ってきた。一方で「内容があまりに豊富で 4 日間かつめで語りきれるような内容ではなかったので、集中講義として受けてしまったのはやや後悔した。受けるなら普通の授業で受けたいところ（学生提出の原文ママ）」という反応もあり、講義と講義の間に 1 週間という内容を咀嚼する期間がある通常の講義との違いが如実にでてきた。

4.3 海外での実装と運用

「ものグラミング」手法の普及啓発を意図した教育活動はすでに海外でも実施している。一方、シェルスクリプトを用いた大規模データ処理に関する教育活動はこれから企画して展開する段階にある。「ものグラミング」の海外展開はまだ限定的ではあるものの以下に述べるようにインド、カナダでは大学学部生を対象とした講義やワークショップを実施したり、卒業論文や修士論文の研究活動の中に「ものグラミング」教育を組み込んだりしてきた。また、ポルトガルでも今後の展開についての議論を行った。以下では時系列ではなく、地域別にこれまでの経緯と今後の展開のうち実現が確実あるいは検討が進んでいる事項について述べる。

4.3.1 インド

インドでは 2018 年にハイデラバード州のハイデラバード工科大学において、「ものグラミング 1」をベースにしたワークショップを開催した。当時、同大学でも Raspberry Pi の利活用に注目が集まっており、学生にも Raspberry Pi は人気が出始めていた。当時はまだ「ものグラミング 2」手法が確立していない時期だったため、コマンドラインやシェルスクリプトを活用して Raspberry Pi 単体でデジタル入出力を行う方法（すなわち「ものグラミング 1」）を実践して見せた。その際、前述の UNIX 哲学に通じる物事はシンプルであることが重要であること (Keep it simple) を説く「KISS の法則」を紹介し、われわれのアプローチの重要性を強調した。

今後は、2023 年度後半に現地にて対面型の「ものグラミング 2」ベースのワークショップや大規模データ処理演

習を実施する方向で準備を進めている。

4.3.2 カナダ

カナダ最東端にあり大西洋に突き出す形のノバスコシア州の州都ハリファクスには同州最大となる州立ダルハウジー大学がある。著者の一人（大野）はこの大学のコンピュータ工学部に2018年にサバティカル研修として5ヵ月ほど滞在した。滞在中、訪問先の研究室で「ものグラミング1」手法を学部および大学院生に伝えるとともに、まだ完成はしていなかった「ものグラミング2」手法についても議論しその内容を発展させた。2018年9月に帰国した後も、同大学の同学部には2018年に1度、2019年に2度、2020年に1度訪問して「ものグラミングワークショップ」や関連レクチャーを開催している。その後は、新型コロナウイルス感染症の世界規模での蔓延により、2020年後半以降は現地での対面方式でのワークショップ[32]は開催できていない。

今後は、2023年以降に対面型のワークショップを再開する方向で、カナダ側と調整を行っている。さらに、シェルスクリプトを用いた大規模データ処理演習についても、250 GB USBメモリ中のデータはそのままにし、解説文書とスクリプトのコメント部分を英訳した「英語版USBメモリ」の作成を終え、英語圏での対面型の演習の準備を整えているので、オンライン型ならただちに、対面型の場合は2023年度中に実施できる状態にある。

4.3.3 ポルトガル

ポルトガルのリスボンにはUSP研究所のヨーロッパ拠点がある。同拠点から紹介を受け、同拠点の技術者とともに2018年にリスボン大学工学部を訪問し、「ものグラミング1」について技術的説明を行い今後の連携について話し合いを行った。リスボンでの「ものグラミング」の展開はこれからであるが、USP研究所ヨーロッパ拠点ではすでにPT4Aのビジネス版であるTukubaiの利用が進んでいるので、「ものグラミング」だけでなくPT4Aを使った大規模データ処理教育についてはリスボン大学工学部でワークショップ形式で展開できる。

今後は、新型コロナウイルス蔓延が終息する前提で2023年度後半に、前述のインドと連携してワークショップや演習を展開する方針を堅持する。

4.3.4 タイ

本稿の著者の一人のナカサンはタイから来日して国内で修士課程と博士課程を終えて学位を取得し、その後金沢大学に就職してわれわれの研究グループに合流した経緯を持つ。その後2022年5月にタイに帰国して現職（カセサート大学）に就いた。このためカセサート大学のコンピュータ工学部には「ものグラミング」やシェルスクリプトを用いた大規模データ処理演習を受け入れる人的環境が整っている。また、2022年10月には同大学から著者らが勤務する金沢大学に、今後の大学間の広範な連携を模索するため

の目的で訪問団が訪れ、その際に著者らの連携も話題になった。その際、われわれは大規模データ処理演習用に用意した250 GB USBメモリを手渡し、今後の具体的な連携について協議した。

今後は、2022年度中に大規模データ処理演習のパイロット授業をオンラインで実施し、タイの新学期が始まる2023年6月以降に現地で大規模データ処理演習と実機を用いる「ものグラミング」ワークショップを実施する。

5. 教育プログラムの講義の実践結果と検討

情報技術教育には、コンピューターリテラシーから始まり、さまざまなプログラミング言語、アルゴリズム、データサイエンス、機械学習、深層学習など、比較的容易なものから高度なものまでたくさんの講義がある。しかし多くの教育機関でほとんど行われていないのが、われわれが実施しているシェルスクリプトを中心に据えた学びであり、講義である。これらは独学もしくは研究室、ゼミなどで先輩からの指導という形で学ぶことが多い。このような状況の中、われわれの教育プログラムの実践によって、通常では情報技術教育のミッシングリンクとも呼べるものを埋めることができる。この観点は、MIT有志によって作られた「The Missing Semester of Your CS Education」が高い評価を得て、さまざまな言語によって翻訳されていることから明らかである。そしてわれわれの教育プログラムでは、The Missing Semester of Your CS Educationでも指摘されているパソコンを始めとした計算システムを効率的、効果的に扱うことが可能となり、時間の浪費だけでなく、データ損失やタスクが完了できないなどの問題も未然に防ぐことができる。The Missing Semester of Your CS Educationでは、エディタやバージョン管理、セキュリティと暗号までであるが、われわれの教育プログラムでは、大規模データ処理まで含んでいる。この演習をこなすことで、シェルスクリプトの使用によるデータ処理を身につけることができ、定型データ、非定型データを問わずに、さまざまな解析を行えるようになる。これは講義の最終課題として提出されたレポートからも読み取ることができ、もっとも優秀なレポートでは、オリンピックへの肯定的な意見のつぶやきと、日本の内閣支持率との関係性を論じたものがあつた。ほかにも、金沢市と富山市の平均気温の差や、金沢市と佐賀市の不快指数の比較などが論じられていた。必ずしも履修者全員が、しっかりと分析をし、立派なレポートの作成ができているわけではないが、最低限、提供したデータとサンプルスクリプトを用いて計算をし、その結果をまとめることはできていた。

また、POSIX中心主義によるプログラミング方法は、時間経過によらず使用可能なプログラムが作れるだけでなく、25年後に、どのような計算機であってもPOSIX環境下であればプログラミングを行える技術を身につけること

となる。そして、IoT 機器などとの組み合わせでは、データそのものの生成から、ネットワークを介したデータの送受信、データの整形と整理が、1 行のスクリプトで可能となる。Society5.0 において、「ものグラミング」の考え方は、社会のいろいろな場所に設置、配置され稼働される IoT 機器などの取り扱い方法のソフトウェア面からの有力な提案の 1 つといえる。

一方で、2022 年段階においては、教育プログラムの設計と実装（と感想の評価）までしかできておらず、定量的な評価にはいたっていない。2015 年度からの講義では、履修者が一桁台であり、定量的な評価が難しかったこともあるが、履修者も徐々に増え、2019 年度に開講した集中講義などでは 20 名以上の履修者がでてきた。その後のコロナ禍で、教室の収容人数が大幅に削減されたこと、講義形式に制限が加えられたことなどもあり、評価が難しくなっている。なお、金沢大学には、オンラインシステムを用いた授業評価アンケートも存在するが、2021 年度までは履修者による回答が任意であったため回答数が極端に少なく評価ができなかった。2022 年度に入ると、授業評価のシステムが変更され、履修者のほぼ全員が回答をする形になったので、その結果の一部を抜粋したものを図 6 として掲載した。この図 6 は、学習目標の達成度を把握するための図である。この図では、縦軸が履修者総数 (N) に対する回答者数の割合であり、横軸が達成度の評価で、これは履修者が -50 から 50 までの点数をつけて自己評価する。このため中心が 0 点となり、右に行くほど評価は高い。これを見ると、学習目標の達成度として、履修者は一定程度の達成をしていると自己評価していることが分かる。しかし、あくまでも生の自己評価のデータを示したも

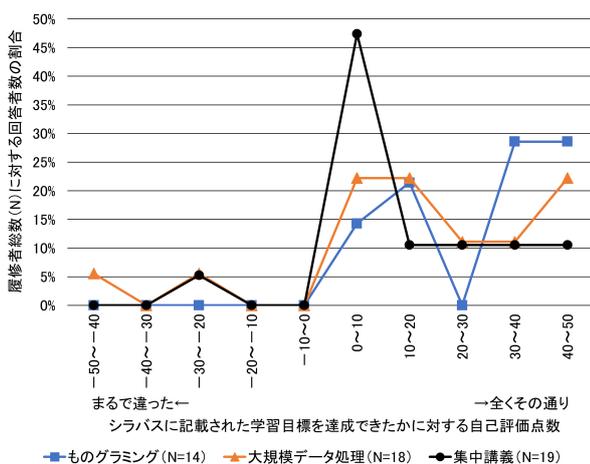


図 6 2022 年度金沢大学における授業評価アンケート結果の一部で学習目標達成度を聞いたもの。(図内の N は履修者総数)

Fig. 6 A part of the 2022 Kanazawa University class evaluation survey results asking about the achievement of learning objectives. (N represents the number of respondents in the figure.)

のであることから、これ以上の評価は難しい。

2022 年度に入り、コロナ禍に端を発する状況も落ち着いてきたこともあり、今後は定量的な評価を実施するため、その評価基準などの設計と実装を行う。評価基準の設計と実装においては、2022 年度から新しい学習指導要領が高等学校に適用されることに伴い、必修となった教科情報の科目「情報 1」を踏まえ、それを学び、さらに 2025 年に実施される大学入学共通テストの「情報」を受けて入学してくる学生たちを基準とすることが必要である。これは 2025 年度以降の高等教育初年次 (大学 1 年生向け) の情報教育の在り方が、既存の教育からの変更が求められるのは「情報」が大学入試科目となったことから明らかである。情報 1 の教育内容が情報分野に関して非常に充実しており、特にプログラミングやアルゴリズム教育では、従来のビジュアルベーシックからの脱却が図られ、その内容も優れている。これらが大学の入学試験科目としても必要とされるため、高等学校での情報教育の在り方自体が変わるとわれわれは評価している。そのため、評価基準の設計についても 2025 年度入学者基準で準備している。

6. まとめと展望

ここまで、われわれが設計した教育プログラムの内容と、それに基づく講義について論じた。この教育プログラムは、単にデータサイエンス教育の 1 つに「シェルスクリプト」を取り込むというだけでなく、プログラミングやデータ処理などのさまざまな場面で、その有効性や効率性、さらに普遍性を、シェルスクリプトを用いることで担保できること、教授することができる。また、この教育プログラムの核の 1 つであるものグラミングを踏まえることで、柳戸によるカルマンフィルターを使った倒立振り子 [33] や、松浦による時間同期コマンドの作成 [34] に至っている。今後、さらに金沢大学大学院博士課程前期の全学生選択必修講義「スマート創成科学」の一部でも、この教育プログラムを実施することになっており、放送大学石川学習センターでも対面講義「シェルスクリプトデータ処理演習」の実施をするなど、国内での教育展開を図っている。

謝辞 本研究は、金沢大学学術メディア創成センターとユニバーサル・シェル・プログラミング研究所の産学連携共同研究として推進された。関係各位のご厚意ご高配に、深く感謝する。

参考文献

- [1] 有限会社ユニバーサル・シェル・プログラミング研究所: ユニケージ開発手法, 入手先 <<https://www.usp-lab.com/methodology.html>> (参照 2022-08-14).
- [2] 森 祥寛, 大野浩之, 北口善明, 中村和敬, 松浦智之, 當仲寛哲: IoT 時代に資する「ものグラミング」教育の開発と実践 (安心・安全な社会に貢献する教育システ

- ム/プログラミング・情報技術教育に関する実践・支援システム/一般), JSiSE research report, Vol.31, No.7, pp.113-118 (2017).
- [3] Anderson, C.: Makers: The New Industrial Revolution, Crown Business (2014).
- [4] 文部科学省: 教育の情報化の推進, 入手先 <https://www.mext.go.jp/a_menu/shotou/zyouhou/index.htm> (参照 2022-08-14).
- [5] 森 祥寛, 佐藤正英, 大野浩之, 笠原禎也, 井町智彦, 高田良宏, 東 昭孝, 二木 恵, Nakasan Chawanat: 金沢大学における携帯型パソコン必携化に関する 12 年間の取組, 学術情報処理研究, No.23, pp.29-42 (2019).
- [6] 當仲寛哲 他: ユニケーゼ原論 USP 研究所 (2010).
- [7] Gancarz Mike: UNIX という考え方: その設計思想と哲学, 株式会社 オーム社 (2001).
- [8] IEEE: POSIX certification, 入手先 <<http://get.posixcertified.ieee.org/>> (参照 2022-08-14).
- [9] Athalye, A., and Gjengset, J., and Gonzalez, O. J. J.: The missing semester of your CS education (日本語版)", 入手先 <<https://missing-semester-jp.github.io/>> (参照 2022-08-14).
- [10] 松浦智之, 大野浩之, 當仲寛哲: ソフトウェアの高い互換性と長い持続性を目指す POSIX 中心主義プログラミング, デジタルプラクティス, Vol.8, No.4, pp.352-360 (2017).
- [11] The Open Group: What is POSIX?, 入手先 <<https://collaboration.opengroup.org/external/pasc.org/plato/>>
- [12] Max, H.: Homebrew, 入手先 <https://brew.sh/index_ja> (参照 2022-08-14).
- [13] 大学コンソーシアム石川: いしかわシティカレッジについて, 入手先 <<https://www.ucon-i.jp/newsite/city-college/index.html>> (参照 2022-08-14).
- [14] Arduino: Arduino, 入手先 <<https://www.arduino.cc/>> (参照 2022-08-14).
- [15] OASIS: MQTT - the standard for IoT messaging, 入手先 <<https://mqtt.org/>> (参照 2022-08-14).
- [16] Eclipse Foundation: Eclipse Mosquitto, 入手先 <<https://mosquitto.org/>> (参照 2022-08-14).
- [17] 大野浩之, 森 祥寛, 北口善明, 中村和敬, 松浦智之, 石山雅三, 當仲寛哲: ものづくりのための「ものグラミング」と実践的教育環境の構築, 情報処理学会マルチメディア, 分散協調とモバイルシンポジウム 2016 論文集, 1335-1340.
- [18] 大野浩之, 松浦智之, 森 祥寛, 「ものグラミング」手法のプログラミング教育への応用 (2022 年度第 4 回研究会スキル開発とその支援技術/一般), JSiSE research report (2022), 37, 4, 26-32.
- [19] Raspberry Pi Foundation: Raspberry Pi, 入手先 <<https://www.raspberrypi.org/>> (参照 2022-08-14).
- [20] 株式会社イーケイジャパン: センサー活用入門 [SU-1203], 入手先 <<https://www.elekit.co.jp/product/SU-1203>> (参照 2022-08-14).
- [21] 株式会社イーケイジャパン: モーター・リレー・プザー制御入門 [SU-1204], 入手先 <<https://www.elekit.co.jp/product/SU-1204>> (参照 2022-08-14).
- [22] Seeed Technology Co., Ltd.: Grove Beginner Kit for Arduino, 入手先 <<https://jp.seeedstudio.com/Grove-Beginner-Kit-for-Arduino-p-4549.html>> (参照 2022-08-14).
- [23] Arduino Software: Arduino IDE, 入手先 <<https://www.arduino.cc/en/software>> (参照 2022-08-14).
- [24] 秘密結社シェルショッカー日本支部: 恐怖! 小鳥男, 入手先 <<https://github.com/ShellShoccar-jpn/kotoriotoko>> (参照 2022-08-14).
- [25] 有限会社ユニバーサル・シェル・プログラミング研究所: PT4A: Personal Tukubai for Academic", 入手先 <<https://www.usp-lab.com/pt4atop.html>> (参照 2022-08-14).
- [26] 気象庁, Japan Meteorological Agency: 地域気象観測システム (アメダス), 入手先 <<https://www.jma.go.jp/jma/kishou/known/amedas/kaisetsu.html>> (参照 2022-08-14).
- [27] 気象庁, Japan Meteorological Agency: 過去の気象データ, 入手先 <<https://www.data.jma.go.jp/gmd/risk/obsdl/index.php>> (参照 2022-08-14).
- [28] 一般財団法人気象業務支援センター: 気象業務支援センター Japan Meteorological Business Support Center, 入手先 <<http://www.jmbsec.or.jp/jp/index.html>> (参照 2022-08-14).
- [29] 松浦智之, 當仲寛哲, 大野浩之: 大量ツイートの収集・分析を個人で手軽に実現可能にする方法の提案, デジタルプラクティス, No.11, pp.173-190 (2020).
- [30] 松浦智之, 中村和敬, 石山雅三: IoT 時代に資する「ものグラミング」教育のための授業開発と実践 (2015 年度第 6 回研究会 21 世紀型能力と情報教育/一般), JSiSE research report, Vol.30, No.6, pp.91-98 (2016).
- [31] 金沢大学: 金沢大学データサイエンス特別プログラム, 入手先 <<https://note.w3.kanazawa-u.ac.jp/news/239>> (参照 2022-08-14).
- [32] Dalhousie University: Facing an "Internet of Things" future, 入手先 <https://www.dal.ca/faculty/computerscience/news-events/news/2019/03/12/facing_an_internet_of_things_future.html> (参照 2022-08-14).
- [33] 柳戸新一, 松浦智之, 鈴木裕信, 大野浩之: シェルスクリプトを用いた UNIX 哲学に基づくリアルタイム制御, ソフトウェアシンポジウム 2021 論文集, pp.73-81 (2021).
- [34] 松浦智之, 柳戸新一, 鈴木裕信, 大野浩之: UNIX 機における IoT 機器制御のためのタイミング管理, ソフトウェアシンポジウム 2021 論文集, pp.82-91 (2021).



森 祥寛 (正会員)

金沢大学総合メディア基盤センター助教。博士 (理学), 大型計算機を用いた数値計算でクォークの閉じこめ機構について研究。2006 年から ICT を教育に活用する手法の研究に従事し, 金沢大学のポータルを作成。教育システム情報学会 会員。



松浦 智之 (正会員)

(有) USP 研究所 技術研究員, (株) 松浦リッチ研究所別働隊 代表取締役隊長。経年劣化に強いソフトウェアやデータ保存形式についての研究を主として推進し, その成果をソフトウェア開発・保守で実践し, また教育・出版等で普及・啓発を図っている。



チャワナット ナカサン (正会員)

金沢大学学術メディア創成センター助教を経て2022年6月よりカセサート大学専任講師.



大野 浩之 (正会員)

金沢大学学術メディア創成センター教授. 博士 (理学). 東京工業大学大学院講師, 郵政省通信総合研究所 (現 NICT), 内閣官房併任を経て2006年より現職. 非常時情報通信, 情報セキュリティ, ものづくり支援などの研究に従事.