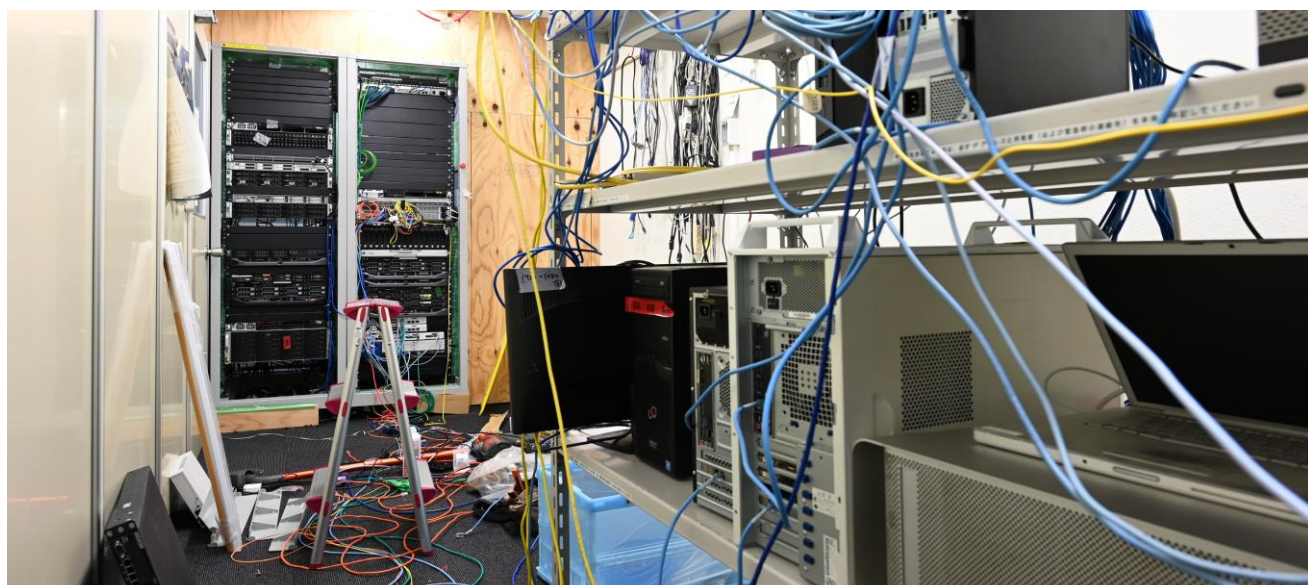




国立大学法人 筑波大学 情報学群 情報科学類

産学間連携推進室



2024 年度 活動成果報告書

目次

情報科学類 産学間連携推進室の概要（細島涼雅）	1
持続可能な AC 部屋のネットワーク・システム構成に関する考察と実装（間瀬太陽）	5
子音情報のみを用いた日本語文字入力手法の提案と実装（細島涼雅）	16
Zig 言語の普及に関する活動（細島涼雅）	21
人為的ミス・いたずらの削減を主眼に置いた飲食店における注文システムの開発（中野あおい）	23
超軽量な量子通信エミュレータの実装（松田心杏）	27
法律文書の自動解析（北野尚樹）	32
モバイル回線を組み合わせた頑健な管理用サーバの構築（服部真吾）	45
つくば WIDE Project 公開ミラーサービス（太田弘樹）	59
居住空間での業務用電子計算機の運用に関する研究（太田弘樹）	61
WIDE Project つくば NOC 運用報告（太田弘樹・菊田一真・服部真吾・間瀬太陽・中野あおい）	65
中学生向け産学官連携 ICT 教育イベント開催の総括（根本晃輔・服部真吾）	68
情報機器資源リユースプロジェクト活動報告（服部真吾）	101

情報科学類 産学間連携推進室の概要

筑波大学 情報科学類 産学間連携推進室

学生代表 202311951 細島涼雅 [s2311951@u.tsukuba.ac.jp]

1 概要

筑波大学 情報学群 情報科学類 産学間連携推進室（以下、当研究室という）は、主に情報科学類生による研究開発活動の支援を目的として学生主体の研究室であり、工学系学系 F 棟 3F230 室の割り当てを受け活動を行っている。

通常、情報科学類生は 4 年次における研究室配属まで、学内に研究拠点を持つことが無い。そこで当研究室では 3 年次以下で研究活動を行いたい学生に対し、学内での研究開発拠点を提供することで活動を支援し、そのような学生が互いに切磋琢磨し合える環境を作ることを目的に各種活動を行っている。また、研究開発した成果を外部にアウトプットする機会も提供し、この研究室から大学発ベンチャー起業や、優秀な技術者が排出される流れを創造することも目的に活動している。

当研究室は 2014 年度より情報学群情報科学類に移管され、同室の活動に置いて正式な支援及び予算の提供を情報科学類より受けている。これは、現在の多岐に亘る活動に加え、今後のより柔軟な活動を目指す上で大きな礎になっている。

2 目標

2024 年度において当研究室は、今までと引き続き以下の項目を目標として各種活動を行った。

- 個人または複数人でプロジェクトを立案遂行し、問題点を認識し、その問題を解決する一連の流れを経験することによって総合的な問題解決能力および開発能力の向上を図る。
- 前年度の成果をもとに研究開発活動を継続的に行い、より長期間にわたるプロジェクトの遂行を通じて、より大規模なプロジェクトを遂行する能力の向上を図る。

3 目標に対する評価

2024 年度において当研究室の活動について、上述した目標に対する評価を述べる。

当研究室においては、様々な活動を行うにあたり、その基礎となる考え方として「問題解決型開発」を取り入れている。これは、既存の技術などについて問題を提起し、その問題を解決するためにソフトウェア等の実装を行うというものである。各活動において、それぞれ問題点を提起しその問題を考察し解決するための実装を行い、問題の解決に挑んでいる。

このように、一連のプロセスに沿って開発できる統合的な開発能力を育成することに重点を置いた点については、今後長期間活動を行っていくにあたり必須であると考えられ、評価できると思われる。

4 組織構成について

本章においては、当研究室の組織構成について記載する。

4.1 略歴

産学間連携推進室は2007年5月頃から活動を開始した。当時は、活動拠点として旧第三エリア3B棟3B408号室を利用していた。当時の活動内容は、主にメンバー個人が各々興味のある分野について、個人の裁量にて何かしらの活動を行うというものであり、組織としてまとまって活動を行うことはなく、活動拠点を個人が個人の目的に利用するという形態が取られていた。

また、厚生労働省が行っている特定健康診査に関するデータ入力ソフトウェアの開発をオーダーメイド創薬株式会社（現、株式会社 LOTUS）およびソフトイサ株式会社との産学連携活動の一環として行った。このソフトウェアは現在、全国各地の診療所等で利用されている。

その後、第三エリア3B棟の改修工事に伴い活動を一時的に休止した。その期間に現在の活動拠点である第三エリア工学系学系F棟3F230号室に利用開始準備を進め、新たな体制での組織運営を行うための基盤の整備を行った。3F230号室が利用可能になった後、各種インフラ整備および運営体制の整理が行われた。

更に2011年から毎年春に研究開発成果の報告会を学内にて行った。成果報告会はプロジェクトが目指す目標や問題点、それに対する解決方法などに関して実際に行った実装の開設などを中心に行われた。また、2023年には情報科学類の主催する夏のオープンキャンパスにてスライド発表を行い、来訪した高校生らに対し研究の紹介等を行っている。

4.2 人員構成

当研究室は、学年主体の組織である。現在では以下の学生が所属しており、互いに様々な分野で活動を行っている。

- 大学院 システム情報工学研究科 博士後期課程：1人
- 大学院 システム情報工学研究科 情報理工学位プログラム 博士前期課程：2人
- 情報科学類 4年：1人
- 物理学類 3年：1人
- 情報科学類 3年：2人
- 情報科学類 2年：1人

5 他組織との連携

5.1 WIDE Project

産学間連携推進室は 2007 年 11 月に筑波大学で初めて WIDE Project の活動拠点となった。その一環として、筑波大学学術情報メディアセンター内に構築した NOC (Net-work Operation Center) においてネットワーク分野の様々な研究開発活動を行っている。

5.2 ソフトイーサ株式会社

産学間連携推進室は、筑波大学発ベンチャー企業であるソフトイーサ株式会社から様々な支援を受けており、研究開発活動に必要な計算機リソース等をはじめ様々な機材の貸与を受けている。また、実験用途として IPc6 ネットワークのコネクティビティを提供するなど、相互に支援を行っている。

さらに、PacketiX NET 公開実験への協力、モバイル回線を用いた管理用予備回線の構築の方法を共同で模索するなどの活動もしている。

5.3 輝日株式会社

産学間連携推進室のメンバーが立ち上げた筑波大学発ベンチャー企業である。ネットワーク機器の提供や技術的なアドバイスなどの支援を受けているほか、共同でネットワーク関連の研究および実証実験、教育活動などを行っている。

6 活動概要

当研究室における活動の基準は、各個人の興味にある。所属するメンバーがやりたい分野について様々な研究開発活動を自由に行うことでさらなる技術力向上を目指している。また、複数のメンバーが協力して分野をまたがった研究開発を行うことで、様々な知見を得て成果につなげることも目的としている。

具体的なプロジェクトとしては以下のものがある。

6.1 産学官連携プロジェクト

- 中学生向け産学官連携 ICT 教育イベント「インデペンデンス・サーバー・デイ」(つくば市総合教育研究所・OPEN プロジェクト)
- 「テキストベースのプログラミングに挑戦してみよう！」(つくば市総合教育研究所・ソフトイーサ株式会社)

6.2 産学連携プロジェクト

- PN トンネルソフトウェアの活用に関する研究および公開実験（ソフトイーサ株式会社）
- 固定 IP アドレス割り当て型サービス（インターリンク株式会社・ソフトイーサ株式会社）
- モバイルデータ回線のボンディング技術（輝日株式会社）
- セキュア公衆無線 LAN に関する研究及び実証実験（輝日株式会社）
- 小・中学生向けプログラミング講座開催支援（輝日株式会社）

6.3 他組織連携プロジェクト

- WIDE プロジェクト筑波大学 NOC の運営（WIDE プロジェクト）
- 国内向けパブリックミラーサーバの運営（WIDE プロジェクト）
- 情報機器資源のリユース・リサイクルに関する支援（筑波大学システム情報系技術室）

6.4 学生活動プロジェクト

- 居住空間での業務用電子計算機の運用に関する研究
- モバイル回線を組み合わせた頑健な管理用サーバの構築
- 法律文書の自動解析
- 超軽量な量子通信エミュレータの実装
- 人為的ミス・いたずらの削減を主眼に置いた飲食店における注文システムの開発
- 子音情報のみを用いた日本語文字入力手法の提案と実装
- 持続可能な AC 部屋のネットワーク・システム構成に関する考察と実装

持続可能な AC 部屋のネットワーク・システム構成に関する 考察と実装

情報学群 情報科学類 間瀬 太陽 (s2410334\@u.tsukuba.ac.jp)

協力: 産学間連携推進室 太田弘樹、服部真吾、中野あおい、細島 涼雅、金子 尚樹、松田 心杏、根本 晃輔

2025 年 04 月 3 日

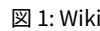
産学間連携推進室では3F230の居室、いわゆるAC部屋の一角にあるサーバールームにおいて、独自のネットワークを引き込み、WIDE Project つくば NOC¹の運用に代表されるサーバーなどの設置を行ったり、生活線として使用するために運用している。AC 部屋ではそういった活動は非常に歴史が長く、サーバールーム内には様々な問題点がある。本文ではそれに対して既存の設備をできるだけ維持しながら、改善を図る方法を考察し、また実装し、現時点での結果などを報告する。

1 使いやすく・高セキュリティな仮想化基盤・共用サービス基盤の構築

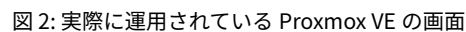
現在 AC 部屋の主な内部向けサービスとして、構成員向けの Wiki サイト 図 1 などが挙げられる。これらは一時的に WIDE 側設備に収容されており、AC 部屋ネットワークの内側から閲覧できる。これは、外部からも閲覧したい場合などがあるため困っている。また、DNS 名が割り振られているが、内部の DNS リゾルバに名前解決を向けないといけず、同一 LAN でも端末によってはセキュリティ機能によって見れなかったりなどがある。また、著者などが常用している昨今の便利なメッシュ型 VPN サービスの Tailscale などを用いていると、Tailscale 内の Magic-DNS 機能のために、PC 内の DNS リゾルバ宛先が書き換えられ、DNS サーバーを指定してもうまく行かないなどがある。

構築のインフラを探していると、AC 部屋の構成員が使える仮想化基盤というのはかつてあったようであるが、現在は停止をしており、ネットワーク疎通性もない状態だった。そのため、この機会に再構築をし、それと同時に構成員が内部向けにあると便利なサービスを気軽に、セキュアに立ち上げられるようなものを構築したいと考えた。

¹Network Operation Center



仮想化基盤の構築で用いるソフトウェアは、VMWare 社の Broadcom による買収によって近頃一般にも注目を集めている Proxmox VE²を用いることとした。VMWare に類似しており、仮想化基盤として必要十分な機能をシンプルに提供していると考えためである。個人的にはベタすぎて面白くないと考えるが、ここで必要以上に複雑、もしくはマイナーなシステムを導入すると、引き継ぎの失敗やサポート終了によるリプレースを余儀なくされる可能性があると考えたためである。



そのため、ここでは単一ホストのようなものを作り、そこにそれぞれ必要に応じて構成員がコンテナを追加していく形を取ることとした。単純なサービスのホストには必要十分であるし、そうすることによって、何が動いているのかが把握が簡単になり、誤って外部に公開されないようにファイヤーウォール

6

ルを構成するなどの最低限のセキュリティ設定が入れられる等の大きなメリットがある。また必要に応じて運用側が用意したコンテナとの連携等も可能になる。

コンテナの追加に関して、希望する人に対してホスト OS のアカウントを追加して、各自 CLI で立てる、という形を取るのが通常であると考える。が、それらがどのような構成ファイルで動いているのかが見れなくなり、またコンテナの一覧性でも難が有り、前述のような問題が起こると考える。そのため、今回は Portainer³と呼ばれる Web-UI によるコンテナマネジメントソフトウェアを用いて、Docker を制御する形を取る。これらにより、Web-UI によって構成ファイルを用いたコンテナの追加・コンテナの管理等が可能になる。

Portainer 上⁴には Traefik⁵という OSS の近代的な HTTP リバースプロキシ⁶が稼働しており、コンテナのシステム等と連携しコンテナにラベルをつけるだけで外部公開と共に HTTPS 化などが出来るようになっている。

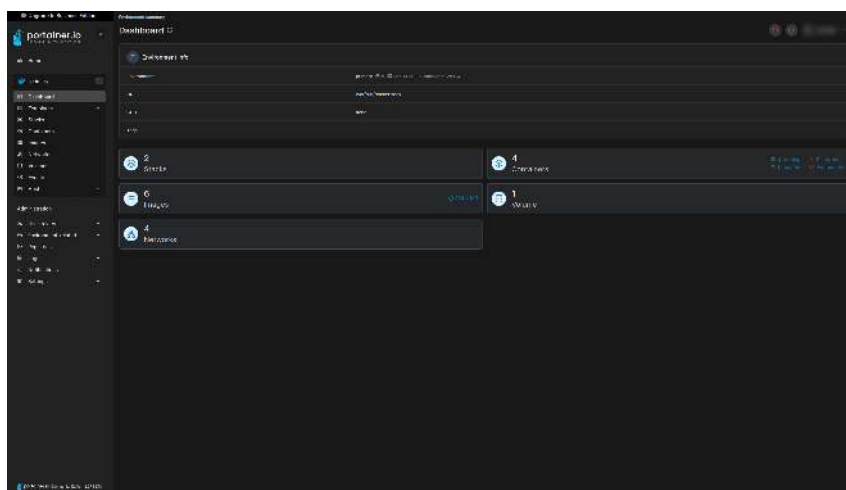


図 3: 実際に稼働している Portainer のスクリーンショット。

内部向け Web コンテンツに対する認証についても、筑波大学の統一認証のような SSO のようなものを気軽に使用できるようにしたい。が、IdP⁷を完全にセルフホストで運用するのはアカウント追加の負担やダウン時リスクなどを考え、GitHub と連携をし認証することとした。

具体的には、OAuth2-Proxy という OSS を用いて実装する。GitHub の OAuth2 App として連携し、加入している Organization を取得し AC 部屋の現役構成員が全員加入している ac-room-tsukuba であれば認可し、その後セッションが継続する限り通常のリバースプロキシとして動作する、という機構である。このようにすればこちらは IdP を持つことなく、信頼性がある認証が行えると考えた。

³<https://www.portainer.io/>

⁴実際には Docker 上にある。

⁵<https://traefik.io/traefik/>

⁶実際には TCP・UDP のプロキシ機能等もあるが、ここでは HTTP のリバースプロキシとして用いている。

⁷Identity Provider

実装したものを図4と図5に示す。認証が求められる所に対してそれを Portainer で設定しているのを前提として、ユーザーが認証されていない場合、図4にリダイレクトされ認証後に表示されるといいう形である。

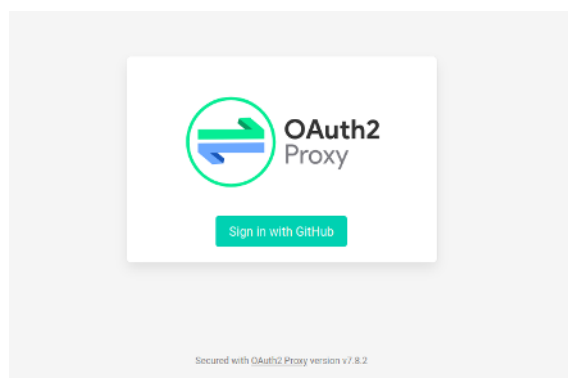


図 4: 認証画面

```
services:
  oauth2-proxy:
    image: quay.io/oauth2-proxy/oauth2-proxy:v7.8.2
    container_name: oauth2-proxy
    environment:
      OAUTH2_PROXY_PROVIDER: "github"
      OAUTH2_PROXY_CLIENT_ID: "${OAUTH2_PROXY_CLIENT_ID}"
      OAUTH2_PROXY_CLIENT_SECRET: "${OAUTH2_PROXY_CLIENT_SECRET}"
      OAUTH2_PROXY_COOKIE_SECRET: "${OAUTH2_PROXY_COOKIE_SECRET}"
      OAUTH2_PROXY_COOKIE_DOMAINS: ".tsukuba-ac.net"
      OAUTH2_PROXY_WHITELIST_DOMAINS: ".tsukuba-ac.net"
      OAUTH2_PROXY_EMAIL_DOMAINS: "a"
      OAUTH2_PROXY_GITHUB_ORG: "ac-room-tsukuba-rightly"
      OAUTH2_PROXY_REDIRECT_URL: "https://ac-auth.tsukuba-ac.net/oauth2/callback"
      OAUTH2_PROXY_HTTP_ADDRESS: "0.0.0.0:4180"
      OAUTH2_PROXY_REVERSE_PROXY: "true"
      OAUTH2_PROXY_SHOW_DEBUG_ON_ERROR: "true"
    ports:
      - 4180:4180
    labels:
      traefik.enable: "true"
      traefik.http.routers.oauth2.entrypoints: "websecure"
      traefik.http.routers.oauth2.rule: "Host(`${ac-auth.tsukuba-ac.net}`) && PathPrefix(`/oauth2/`)"
      traefik.http.routers.oauth2.tls.certresolver: "leresolver"
      traefik.http.services.oauth2.loadbalancer.server.port: "4180"
      traefik.http.middlewares.oauth2.verify.forwardAuth.address: "http://oauth2-proxy:4180/oauth2/auth"
      traefik.http.middlewares.oauth2.verify.forwardAuth.trustForwardHeader: true
      traefik.http.middlewares.oauth2.chain.middlewares: "oauth-auth"
      traefik.http.middlewares.oauth2-auth.errors.service: "oauth2"
      traefik.http.middlewares.oauth2-auth.errors.status: "401"
      traefik.http.middlewares.oauth2-auth.errors.query: "/oauth2/start?rd={uri}"
    networks:
      - portainer-traefik

networks:
  portainer-traefik:
    external: true
```

図 5: OAuth2-Proxy の設定(試験中のもの)

OAuth2・今回導入した各種ソフトへのある程度の理解が必要だが、これだけの行数で簡易な GitHub を IdP とする認証が実装できた。これらは既存のコンテナとして立ち上がっており、単純に HTTP アプリケーション全体に認証を掛けたいだけあれば、コンテナに一行ラベルの設定を投入すれば既存のコンテナがそれを認識し、認証のフローを導入する。

例示として、図6を挙げる。これは traefik/whoami というリクエストされたヘッダを返すという単純なアプリケーションのデプロイ設定・外部公開・認証の設定をしている。

```
services:
  whoami:
    image: traefik/whoami
    container_name: whoami
    labels:
      traefik.enable: "true"
      traefik.http.routers.whoami.entrypoints: "websecure"
      traefik.http.routers.whoami.rule: "Host(`${ac-secwhoami.tsukuba-ac.net}`)"
      traefik.http.routers.whoami.tls.certresolver: "leresolver"
      traefik.http.services.whoami.middlewares: "oauth"
      traefik.http.services.whoami.loadbalancer.server.port: "80"
    networks:
      - portainer-traefik

networks:
  portainer-traefik:
    external: true
```

図 6: 投入した設定

類似のことをする SaaS として、Cloudflare 社が提供する Cloudflare ZeroTrust などが挙げられる。サーバーPCでデーモンを起動させ、それを経由して外部へのアクセス性をもたせ、任意の認証も付けられるというものである。検討したが、無料版はユーザー数が 50 人までという制限もあり、また、そもそも

そこで用いるドメインの権威を Cloudflare の権威サーバーに向けないといけないというものであり、現実的ではなく、Cloudflare 社に完全に依存してしまうことになってしまうため、除外した。

1.2 将来

これらはまだ実証実験の段階で、今後更に維持しやすいシステム等について考えながら、ドキュメント化もしつつ運用していきたいと考える。

1 つ思ったこととしては、IdP として GitHub を使うということは、万が一 GitHub がダウンした際に何らかにアクセスできなくなってしまうというリスクがあることである。一つ思いついたのは、通常時は GitHub を IdP として用い、認証時にオンプレの IdP にユーザが存在しない場合は、そこにユーザーを作成しておき、ダウン時・オフライン時などは明示的にオンプレの IdP を用い、逆にオンプレの IdP は、ダウンしていても基本的に影響がないようにする というのである。これについてすでに実装が有るかななどを調べていきたい。

また、これは元の話とはかけ離れてしまうが、WIDE Project から Proxmox VE のダウンロードサーバーが極端に遅いように思えた。Traceroute などを用いて調査すると、日本に一番近い公式のダウンロードサーバはどうやらシンガポールにあるようである。Proxmox VE のアップデートなどを提供する apt のレポジトリサーバーも同様である。

探した所、中国圏などではミラーをしている所を発見できた。が、日本圏ではミラー等が無いようである。そもそも Proxmox VE には公式ミラーという概念が無いので、問題が無いかを精査して、問題なく出来そうであればミラーをしたいと考える。



図 7: AC 部屋(WIDE Project ネットワーク)よりダウンロードの様子 1MB/s 以下ぐらいしか出ない

```
masetaiyo@masepc3 ~ main [1] @ sudo traceroute download.proxmox.com
traceroute to download.proxmox.com (117.120.5.24), 30 hops max, 60 byte packets
 1  192.168.7.253 (192.168.7.253)  2.277 ms  2.119 ms  2.031 ms
 2  ve-29.cisco1.tsukuba.wide.ad.jp (203.178.132.65)  3.099 ms  3.017 ms  2.932 ms
 3  ve-2045.juniper1.notemachi.wide.ad.jp (203.178.136.85)  4.971 ms  4.865 ms  4.779 ms
 4  101.203.88.32 (101.203.88.32)  4.698 ms  5.740 ms  4.533 ms
 5  ae1-9.rt.eqx.sin.sg.retn.net (87.245.233.101)  78.157 ms ae5-8.rt.eqx.sin.sg.retn.net
(87.245.234.154)  85.737 ms ae1-9.rt.eqx.sin.sg.retn.net (87.245.233.101)  77.997 ms
 6  203.174.80.58 (203.174.80.58)  81.432 ms  82.067 ms  81.926 ms
 7  * * *
 8  * * *
 9  sg2.cdn.proxmox.com (117.120.5.24)  75.009 ms  74.982 ms  77.249 ms
```

2 ファシリティの改善

図8と図9に改善を図る前の画像を挙げる。これらの問題点は主に以下のとおりである。

- 19 インチラック
 - (図9 右上)運用終了している WIDE Project を収容していたルーターがラックを専有している。
 - ホットアイル・コールドアイルの分離のためラックの空き部分にブランクパネル等を設置するよう努めているが、ブランクパネルの在庫がないのでビニール等で塞がれていたり、そもそも塞がれてなかったりする。
 - 全体的に配線がカオス状態。
- デスクトップ PC ラック
 - 使われていない不要物が置かれている。
 - 100V AC 電気の配線がいわゆるタコ足にタコ足を重ねている状態。
 - 全体的に配線がカオス状態。

これらの問題点を一気に解決するため、AC 部屋では2日のダウンタイムを設け、これらの改善が行われた。

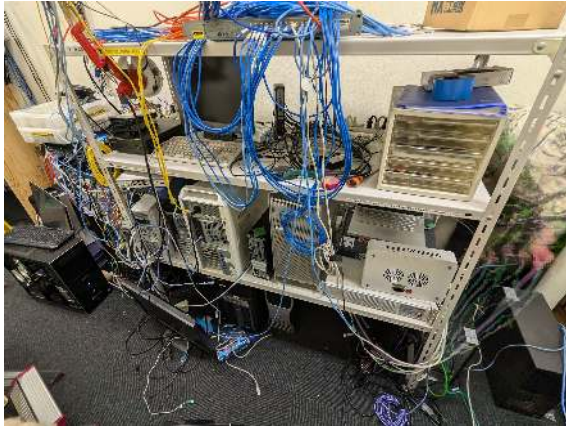


図 8: デスクトップ PC ラック

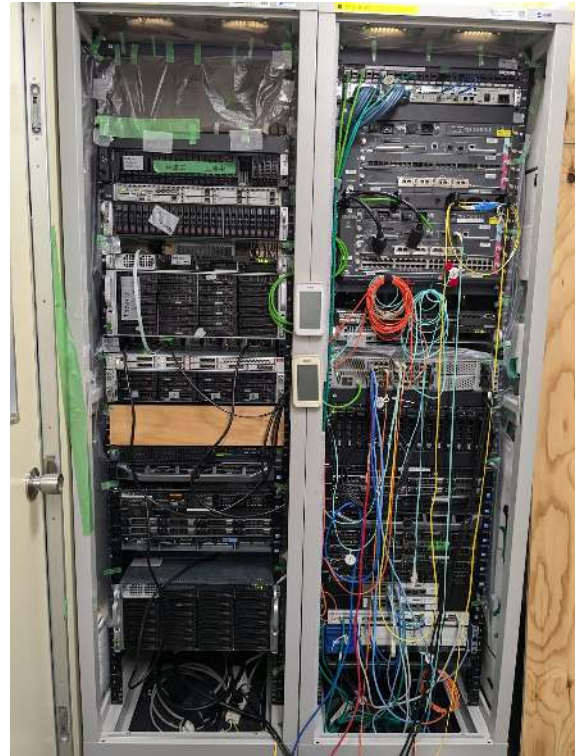


図 9: 19 インチラック

2.1 デスクトップ PC 周り : 電源周りの改善・物の整理

電源周りの問題として、図 8 では見えないのだが、AC 100V コンセントの配線が OA タップによる分配に分配を重ねられている。また、通常の家にあるような壁コンセントから出しているため、上記の棚移動に伴い、どちらの面でも作業が出来るようになり、過失により抜いてしまう可能性が高くなったのが挙げられる。

解決策として、OA フロア(フリーアクセスフロア)に使われるような床用コンセントの製品図 10 が AC 部屋にあったのでそれを応用するというものにした。これにより確実な電源が構築できた。改善の余地があるとすれば、これらの製品群が OA フロア向けに販売されている特性上、配線長が長く、このような短距離の配線だと余長が多く出てしまう点が挙げられる (図 11 下部で散見される)



図 10: フリーアクセスフロア用（OAフロア） | Panasonic : https://www2.panasonic.biz/jp/densetsu/haisen/switch_concent/floor_outlet/building/oa_floor/ より引用



図 11: AC 部屋サーバールームに設置されたジョイントボックス

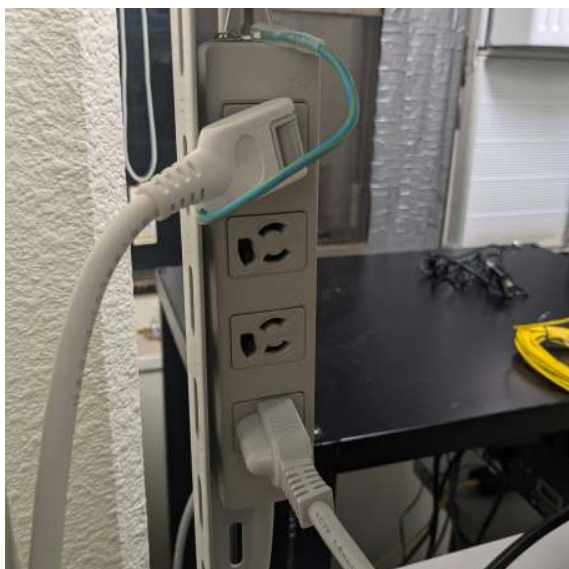


図 12: ジョイントボックス用 OA タップ



図 13: エンドユーザー向け OA タップ

これら以外にも上に乗っていた不要物を撤去し、純粋なデスクトップ PC ラックとして使うことにし、収容能力の拡大ができた。棚自体をリプレースする計画も立っており、棚を奥に置くことで、設置をして徐々に移行ができるようにした。

ネットワークの部分でも図 14 のような 19 インチラックとデスクトップ PC ラック間の渡り配線が多く、また基幹となる配線も含まれるため、危ないというのは感じていた。そのため、構成上不要な配線を削減させるのは前提として、これらを解消するためにも当該ラック両方に L2 スイッチを設置し、それらの一部ポートに対してその間のみで使うアクセス VLAN を設定し、それらを多重させたトランクポートを設定することで、仮想的なパッチパネルのように活用するといった改善も試みている。

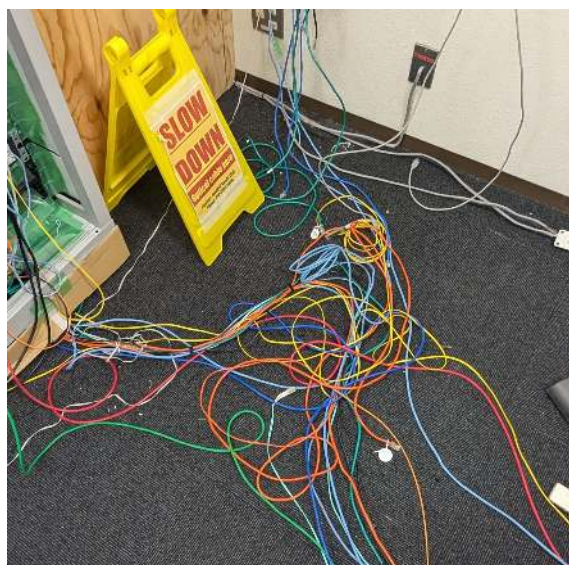


図 14: 渡り配線

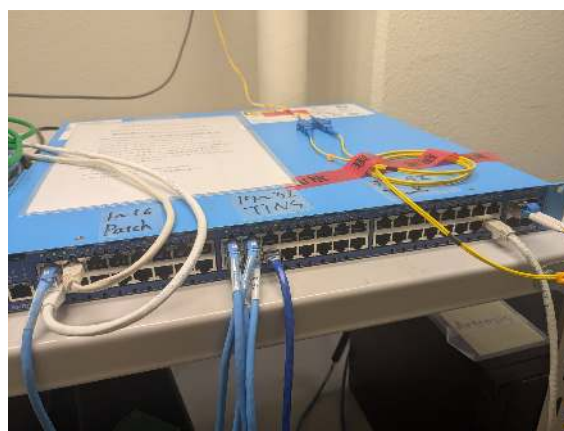


図 15: L2 スイッチを活用したパッチパネル？

2.2 19 インチラック：ルーター抜去・ブランクパネルの設置

4U ある Cisco WS-C6503-E ルータ 2 台を抜去した所や、元から空いていたところについて、今回大量に購入したブランクパネルを設置した。ここで少し興味深かったのは、1 個約 2,000 円する FS 社製の 1U ブランクパネルと今回購入した音楽用機材に使われる 1 個約 600 円 CLASSIC PRO(サウンドハウス社)の 1U ブランクパネルの隙間の違いである。値段の違い・どこに用いられる方向性の違いなのか、如実に出ている。図 18 図 19 を参照されたい。だが、CLASSIC PRO のものに安価なまま U 数によるブランクパネルのバリエーションがあるため、それらを購入できればそこまで問題にならないと思われる。



図 16: FS 社製 1U ブランクパネル



図 17: CLASSIC PRO 1U ブランクパネル



図 18: FS の物同士であれば隙間はほぼ空いていない。



図 19: (画像真ん中 3U が当該) 少し隙間が空いている。

2.3 改善後

図 21 と 図 20 に示す。執筆現在も進行中であり、完全なものにはなっていない為、一部まだ見にくいところがある。

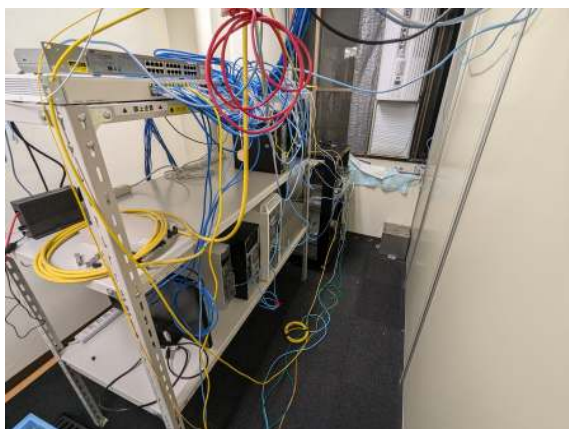


図 20: アフター : AC 部屋 PC ラック



図 21: アフター : AC 部屋 19 インチラック

2.4 将来

これらはまだ続いている作業であり、まだ完全には出来ていないものも多数ある。そのため、今後はこれらを完全に行っていく作業をしていきたい。

子音情報のみを用いた 日本語文字入力手法の提案と実装

情報科学類 3 年 細島 涼雅
学籍番号 202311951

第 1 章 はじめに

日本語入力システムは、日常的に使用される重要な技術である。現在の主流な日本語入力方式には、ローマ字入力やかな入力があるが、これらは母音と子音の組み合わせを基にした入力方法であり、特定の入力方式に依存している。本研究では、新しい日本語入力方式として「子音情報のみを用いた入力システム」を提案する。

本システムは、ユーザーが子音のみを入力すると、システムが適切な単語を補完することで、日本語入力を支援する。例えば、「nhng」と入力すると、「日本語」「二本木」などの候補を生成し、文脈に応じた最適な単語を選択する。本研究では、この入力方式を実現するために、KenLM を用いた n-gram モデルと、形態素解析辞書 ipadic を活用する。

本システムは、VR 環境における文字入力や、フリック入力が困難な状況においても応用が可能である。

現在多くの HMD に標準で搭載されている文字方法はコントローラで仮想的なキーをポインティングすることにより入力を行うような直接ポインティングによるインタラクションである。主流の直接ポインティングによる入力手法では、物理的コントローラのアバタとなる VR 空間の仮想コントローラの先から Ray を飛ばし、Ray がオブジェクトと衝突した地点をポインタの位置として利用する。また仮想的なキーボードは一般的な QWERTY レイアウトか五十音のレイアウトが一般的に利用される。



図 1: 現在主流の入力インターフェース（PC Watch 2023 年 3 月 4 日の記事より引用）

しかし、このインターフェースの課題点として「キーが多く、またキー自体が小さいため、ポイントするのが難しく入力ミスが発生しやすい」、「空中で腕を動かして操作する必要があるため疲労が大きい」といったものが挙げられる。さらに、VR 空間上では空間把握が難しいため、文字入力においては様々な支援が求められる。そのため、子音のみを用いた簡易な入力方式が有効であると考えられる。

また、手指の障害や一時的な制約（手袋の着用など）によりフリック操作が難しいユーザーにとっても、本方式は利便性の向上に寄与する可能性がある。

第 2 章 先行研究

田中らは、子音のみを入力することで入力キーの操作回数を劇的に減らせることを示した。一方で、入力が曖昧になるため候補数が増え、日本語入力における精度の課題が生じることを指摘している。[1]

この研究では、候補数を絞り込むために、HMM（Hidden Markov Model）を用いた言語モデルと、PPM（Prediction by Partial Matching）を用いた学習モデルを融合させて実装している。しかし、この研究が発表されたのは 2001 年であり、その後の技術の進展を考慮すると改善の余地があると考えられる。特に、計算資源の向上や大規模データを活用した統計的言語モデルの発展を踏まえれば、より精度の高い手法が可能である。

また、本研究の調査では、2001 年以降に同様の子音入力に関する研究は見つからなかった。したがって、本研究は田中らの研究を基盤としつつ、より現代的なアプローチを採用することで、日本語入力システムの新たな可能性を探ることを目的とする。

本研究では、HMM と PPM ではなく、KenLM を用いた n-gram 言語モデルと ipadic 辞書を活用することで、語の候補を効率的に絞り込み、入力のアム昧性を軽減する手法を採用する。先行研究との比較を通じて、提案手法がより実用的であることを示す。

第 3 章 実装

本章では、提案する日本語入力システムの実装について述べる。実装するシステムは、ユーザから入力された子音情報から、あり得る単語を列挙し、入力されている文字列をもとに選択される確率の高い順にユーザに提示するというものである。文脈情報を考慮するために、直前の単語と組み合わせた確率計算を行っている。

具体的な本システムのデータ処理フローは以下ようになる。

1. ユーザが子音列（例: “nhng”）を入力する。
2. 辞書データをもとに可能な単語のリストを生成する（例: “日本語”, “二本木”）。
3. KenLM により各単語の確率を計算し、最適な候補をランキングする。
4. 確率の高い順に単語をユーザに提示する。

本実装により、オフライン環境でも高速に動作する日本語入力システムを実現した。

3.1 n-gram 言語モデルの学習

KenLM を使用して、日本語の大規模コーパスから n-gram 言語モデルを学習した。今回は Wikipedia 日本語コーパスを用いて学習を行った。また、形態素解析エンジンとして MeCab を使用し、形態素解析と分かち書きを行った。

3-gram モデルを用いて、入力された単語列に対する単語候補のスコアリングを行った。

3.2 子音情報の抽出

ipadic 辞書を用いて単語の子音情報を抽出した。辞書にある「読み」の情報を利用し、子音の情報のみに変換した。

第 4 章 現在の進捗

実装の簡略化及び、簡単な実現可能性を評価するために十分であるため、今回実装したのは、簡単な CLI ツールとして作成した。

図 2 に、システムが動作している様子を示す。


```

入力列: <s>
子音列を入力 (終了するには Ctrl+C) : tkbdagk

候補:
1. ('筑波大学', -7.846527099609375)
0. キャンセル

番号を選択 (0 でキャンセル) : 1
確定: 筑波大学

入力列: 筑波大学
子音列を入力 (終了するには Ctrl+C) : h

候補:
1. ('は', -9.289896011352539)
2. ('は', -9.289896011352539)
3. ('は', -9.289896011352539)
4. ('へ', -10.477800369262695)
5. ('へ', -10.477800369262695)
0. キャンセル

番号を選択 (0 でキャンセル) : 1
確定: は

入力列: 筑波大学は
子音列を入力 (終了するには Ctrl+C) : tkb

候補:
1. ('部', -15.11402702331543)
2. ('つくば', -15.602736473083496)
3. ('筑波', -16.049102783203125)
4. ('乳首', -16.07625961303711)
5. ('焚火', -16.337257385253906)
0. キャンセル

番号を選択 (0 でキャンセル) : 2
確定: つくば

入力列: 筑波大学はつくば
子音列を入力 (終了するには Ctrl+C) : n

候補:
1. ('の', -13.51097583770752)
2. ('の', -13.51097583770752)
3. ('の', -13.51097583770752)
4. ('の', -13.51097583770752)
5. ('の', -13.51097583770752)
0. キャンセル

番号を選択 (0 でキャンセル) : 1
確定: の

入力列: 筑波大学はつくばの
子音列を入力 (終了するには Ctrl+C) : sagadagk

候補:
1. ('総合大学', -15.854410171508789)
2. ('総合大学', -16.93230438232422)
0. キャンセル

番号を選択 (0 でキャンセル) : 1
確定: 総合大学

入力列: 筑波大学はつくばの総合大学
子音列を入力 (終了するには Ctrl+C) : ds

候補:
1. ('です', -14.236690521240234)
2. ('出し', -15.225807189941406)

```

図 2: 作成したシステムで、「筑波大学はつくばの総合大学です」という文を入力している様子

ある程度の文章であれば、入力することができることがわかった。しかし、使いやすい UI の提供や、ユーザを用いた実験などは行えていない。

第 5 章 今後の展望

前章で挙げた使いやすい UI の提供や、ユーザを用いた実験の他に今後の展望として以下が挙げられる。

5.1 ユーザ適応型モデルの導入

個々のユーザーの入力傾向を学習し、パーソナライズされた予測を行うことで、より精度の高い入力補助を実現する。特に、今回作成しているシステムは、性質上辞書の内容を非常に強く受ける。そのため、ユーザ適応型モデルの導入は使いやすさを向上させるために必要であると考えられる。

5.2 リアルタイム適応

入力中のコンテキストを考慮し、リアルタイムで候補の再ランキングを行うことで、より自然な日本語入力を可能にする。

5.3 LLM や Transformer の活用

より高度な自然言語処理技術を活用し、従来の n-gram モデルよりも精度の高い予測を行う可能性を探りたい。

特に、現在 LLM の台頭により、非常に高い精度で単語の予測や、文章生成が可能となっている。

また、日本語入力への応用として、ニューラルかな漢字変換システム Zenzai は、かな漢字変換に Transformer Decoder ベースの言語モデルを適用した手法を提案し、高い変換精度を達成している。[2]

参考文献

[1]. 田中 久美子, 犬塚 祐介, 武市 正人 : 携帯電話における日本語入力 - 子音だけで 日本語が入力できるか, 情報処理学会論文誌 Vol.43 No.10, 2002

[2]. 三輪 敬太, 高橋直希 : ニューラルかな漢字変換システム Zenzai, 言語処理学会第 31 回年次大会

Zig 言語の普及に関する活動

情報科学類 3 年 細島 涼雅

学籍番号 202311951

第 1 章 はじめに

Zig 言語は、高いパフォーマンスとシンプルな設計を特徴としながらも、まだ普及が進んでいない言語のひとつだ。新しい言語を広めるには、実際に手を動かして試せる環境が重要だが、一般的な開発用途ではセットアップの手間や学習コストが障壁となることが少なくない。

そこで私は、「競技プログラミングを通じて Zig を普及させる」というアプローチに注目した。競技プログラミングでは、オンラインジャッジによって即座にプログラムの正誤を確認できるため、初学者でも気軽に Zig を試すことができる。また、競技向けのライブラリが充実すれば、実用的な場面での利用も広がる可能性があると考えられる。

この目的のもと、国内最大の競技プログラミングコンテストを行うサービスである AtCoder での Zig 言語サポートの強化、およびそれに付随するライブラリの開発に取り組み、次期言語アップデートに取り込まれることに至った。

第 2 章 活動の概要

2.1 AtCoder における Zig 環境のアップデート

AtCoder では Zig 0.10.1 が提供されていたが、Zig の開発は活発に進んでおり、最新バージョン（当時）である 0.14.0 との間に大きな仕様変更が存在した。

以前はライブラリの依存関係を外部のパッケージマネージャを用いて管理していたが、それを Zig 0.11.0 以降から Zig に付随するようになったものに変更することを提案するなどをした。

2.2 ac-library の Zig 版の開発

競技プログラミングにおいては、高速なデータ構造やアルゴリズムを提供するライブラリが不可欠である。AtCoder では C++ 向けに ac-library (AtCoder Library) が提供されている

[1]が、Zig には同等の標準ライブラリが存在しない。そこで、ac-library の Zig 版を作成し、競技プログラミングに適した形で提供した。

C++版 ac-library を参考にしつつ Zig 向けに最適化、競技プログラミング以外の場面でも汎用的に使えるような設計と実装を行った。

ゼロオーバーヘッドな抽象化を活かし、パフォーマンスを維持し、Zig の言語特性である comptime を活用し、テンプレートのような柔軟な構造を実現した。

また、オリジナルの実装では明示されていないエラーの種類を整理し、Zig のエラーハンドリング機構を活用した設計を導入し、意図しない計算エラーや無効な操作をコンパイル時または実行時に適切に検出できるようにした。

これらの取り組みにより、AtCoder 上での Zig の利用環境が向上し、競技プログラミングを通じた Zig の普及が促進されることを期待している。

第 3 章 今後の展望

本活動では、AtCoder における Zig 環境のアップデートと ac-library の Zig 版の開発を通じて、競技プログラミングを Zig の普及手段とする取り組みを行った。しかし、Zig のさらなる普及のためには、競技プログラミングに限らず、より広い分野への展開が重要である。

継続した AtCoder における Zig の環境のアップデートと、ac-library の Zig 版のメンテナンスを行っていきたいと考えている。特に、Zig の開発は活発であり、今後も仕様変更や最適化が進むことが予想されるため、AtCoder の Zig 環境を最新の状態に保つことは、言語の競技プログラミングにおける利便性を維持する上で重要である。

さらに、競技プログラミング以外の分野でも Zig の普及を促進することを目指す。Zig は汎用プログラミング言語であり、様々な分野に応用することができる。その応用についてのチュートリアルを作成することで、多くの人が Zig に興味を持つことができ普及に貢献できるのではないかと考えている。

Zig は、シンプルな構文、強力な comptime、明示的なエラーハンドリングなどの特徴を持ち、システムプログラミングや競技プログラミングの分野で高い可能性を秘めている。今後は、これらの特徴を活かした活動を継続し、Zig の普及に貢献していきたい。

参考文献

[1]. AtCoder. 「AtCoder Library (ac-library)」. GitHub, 2025 年 3 月 31 日取得, <https://github.com/atcoder/ac-library>

人為的ミス・いたずらの削減を主眼に置いた 飲食店における注文システムの開発

情報科学類 3 年 202312794 中野あおい

1 はじめに

近年の飲食店においては、ホール側・客のテーブルにタブレット端末を配置し、客がそこから注文を行うシステムが広く採用されている。しかしそのような方式には、むやみに端末を操作してシステムの混乱を試みるようないたずらのリスクや、通信の不具合により注文が届かなかったり、会計が一致しなくなってしまうたりするような基本的な問題が存在する。また、端末の設定がうまくいかず不正に会計が行われる等人為的ミスが原因のトラブルが起こる可能性がある。

そのほかにも、このようなシステムは SIer のようなシステム開発企業が開発していることがほとんどで、知見の共有等がなされにくい傾向にあり、さらに飲食店ごとのカスタマイズに高額な料金が発生する可能性があるなど、さまざまな飲食店のニーズに応じた形となっていないように思われる SIer 等営利企業の既存のクローズドなシステムは導入初期に高額な費用が必要で、さらに保守費用も必要になるため、注文システムの導入に費用をかけづらい場では採用されにくく、中小規模・個人経営の飲食店や大学の学園祭の模擬店等では、注文をシステム化することを望んでもハードルが高く手を出すことができない。

そこで、人為的ミス・いたずらの削減を主眼に置いたオープンな飲食店向けのシステムを開発することで、これらの問題を解消し、また飲食店の注文システムといった一般的なシステムの開発市場を広げていきたい。また、オープンソースで公開することにより、開発者以外が手を加えることができ、知識があれば欲しい機能を自分で追加することができる。高校・大学の学園祭の模擬店等で注文をシステム化したいが一から実装するのは難しい、のようなことがあっても、オープンソースで公開されたコードがあれば、システム化のハードルが低くなることが予想される。

2 実装方法

Web アプリケーションとしての実装を想定する。客席に置かれた端末から注文する形式（客席端末版）や、客席端末を用意するのが難しい場合に向けて、客が自身のスマートフォンで QR コードを読み取って注文することができる形式（スマートフォン版）にも対応する。

飲食店のレジで情報を集約する機器を親機と呼び、客側の端末や店舗のキッチンに置く端末などを子機と呼ぶ。本研究では親機、客側の子機、キッチン側の子機の実装と各機器が共有するデータベースの実装を行う。

端末で不正な操作が行われたときに親機に通知されるシステムや、一定の時間おきに親機と子機のデータの突合せを行うシステムなどを開発するアプリに組み込む。バックエンドの言語は GO 言語を使用し、データベースは PostgreSQL、フロントエンドでは React, TypeScript を使用する予定である。

3 先行研究

現在主に飲食店で使われているセルフオーダーのシステムは、店内に設置された券売機式の端末で注文する方式や、客がスマホのアプリから注文する方式、客席に置かれたタブレット型の端末を用いて注文する方式、客が QR コードを読み取って Web アプリなどから注文する方式などがある。

これらを用いることで店は人件費を削減し、更に売り上げや注文履歴の管理もしやすくなる。

現在客席端末としての使用を予定されるソフトウェアで、広く利用されているものの中にオープンソースのものは見つからなかった。また無料で端末にインストールすることができる、POS と連携する注文アプリの中には、かつて筆者が実際に飲食店で使用したアプリに、アプリが途中で中断されるとデータが破損し、正常に会計が行われないものがあった。これは、誤操作や通信トラブルへの対策の観点で十分でない点において改善の余地があると考ええる。

客が自身が所有するスマホで注文する形式の Web アプリケーションは、オープンソースのもの [1] が存在する。これはいたずら防止などの策を提供できていないように感じるため、改善の余地があると考ええる。よって、これを一部参考にしつつ改善を加え、スマートフォン版の作成、客席端末版への応用を考える。

4 進捗

まず、フロントエンド側の開発を進め、客席端末版のシステムの見た目の部分を整えた。React+TypeScript を用いて実装を行った。

以下の図 4.1、図 4.2 が今回作成した画面である。ジャンルのボタンを押すことによって商品の一覧のセットが切り替わる。(図 4.1 は「ジャンル 1」、図 4.2 は「ジャンル 2」を表示している。)

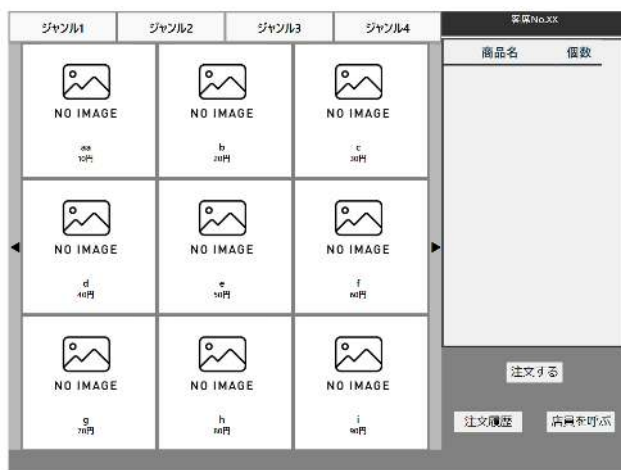


図 4.1: タブレット端末版画面

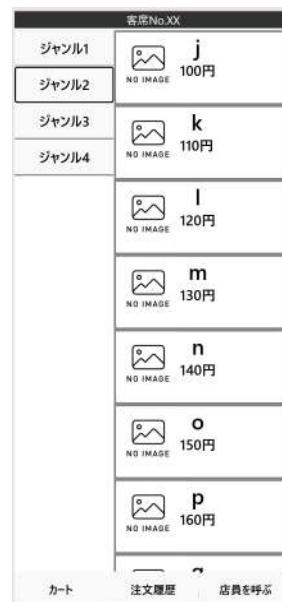


図 4.2: スマートフォン端末版画面

また、一覧の商品を選択すると図 4.3、図 4.4 のようにポップアップが表示され、個数を選択し「決定」ボタンを押すと、リストにアイテムが追加される。

タブレット端末版は画面右のリスト (図 4.5)、スマートフォン端末版は「カート」ボタンを押すことでリスト (図 4.6) を確認でき、それぞれ注文ボタンを押すと、リストの内容がバックエンド側に値として渡される、という操作になっている。

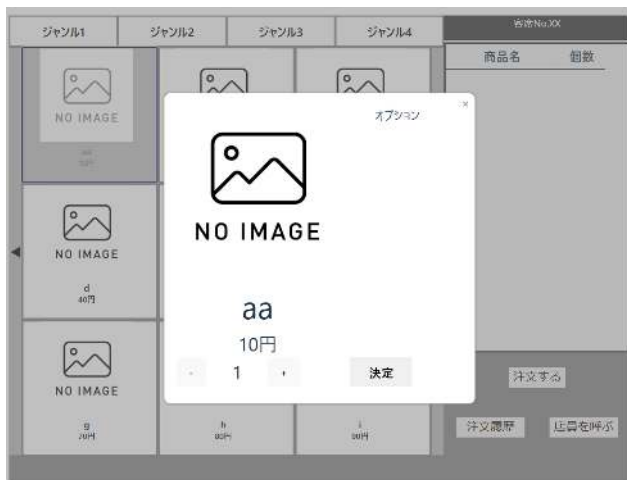


図 4.3: タブレット端末版ポップアップ

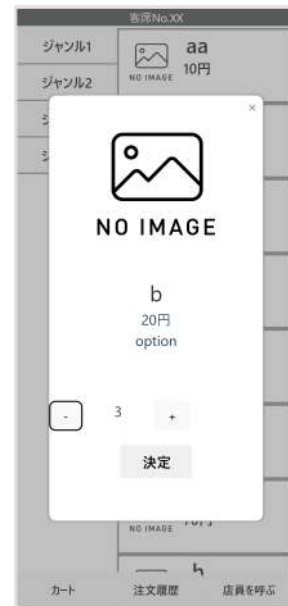


図 4.4: スマートフォン端末版ポップアップ

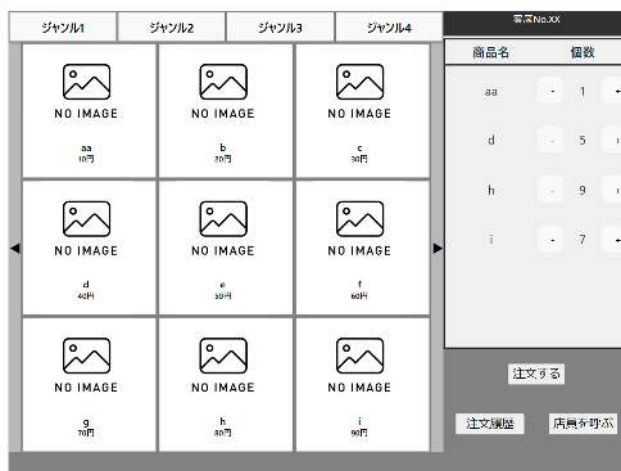


図 4.5: タブレット端末版注文リスト

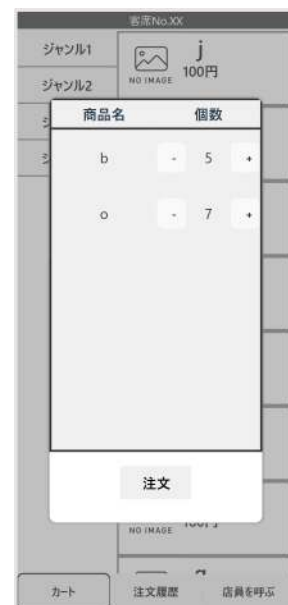


図 4.6: スマートフォン端末版注文リスト

これらの実装によって、客席端末のフロントエンドの最低限の機能の実装ができたと考える。
また、ローカルにデータベースサーバを建て、PostgreSQL を使った学習を進めた。

5 今後の展望

今後は、親機や店舗のキッチン側の端末に表示する画面の実装や、学習を進めバックエンド、データベースの実装などを考えている。

現在はローカルに建てて使用しているデータベースサーバを、産学官連携推進室等にサーバマシンにおいて

外部からもアクセス可能な状態にし、バックエンドの実装を進め、バックエンドを通した端末とデータベース間のデータの受け渡しを可能にする実装に取り組みたい。

また、現在自身の経験の下に作成した客席端末の UI や今後作成する親機等の端末の UI を、複数人から評価してもらうなどによって改良し、ユーザフレンドリーで人為的ミスの起きにくいような UI にしたいと考える。

6 おわりに

本研究に取り組んだところ、セルフオーダーシステムの実装は複雑で、SIer などの企業が開発をすべて請け負う理由をよく理解できた。しかし今後もセルフオーダーシステムの需要は増えると予測され、オープンソースのソフトウェアの需要も存在すると考える。本研究の主眼であるソフトウェアの機能では防ぐことが難しいヒューマンエラーや「人為的ミス・いたずら」の防止などといった問題へのアプローチだけでなく OWASP Top 10 のような一般的な脆弱性に対するアプリケーションの構築を行いたいと考える。

参考文献

- [1] オープンソースのセルフオーダーシステム. <https://github.com/kawax/self-ordering>.

超軽量な量子通信エミュレータの実装

202211669 理工学群物理学類3年 松田 心杏

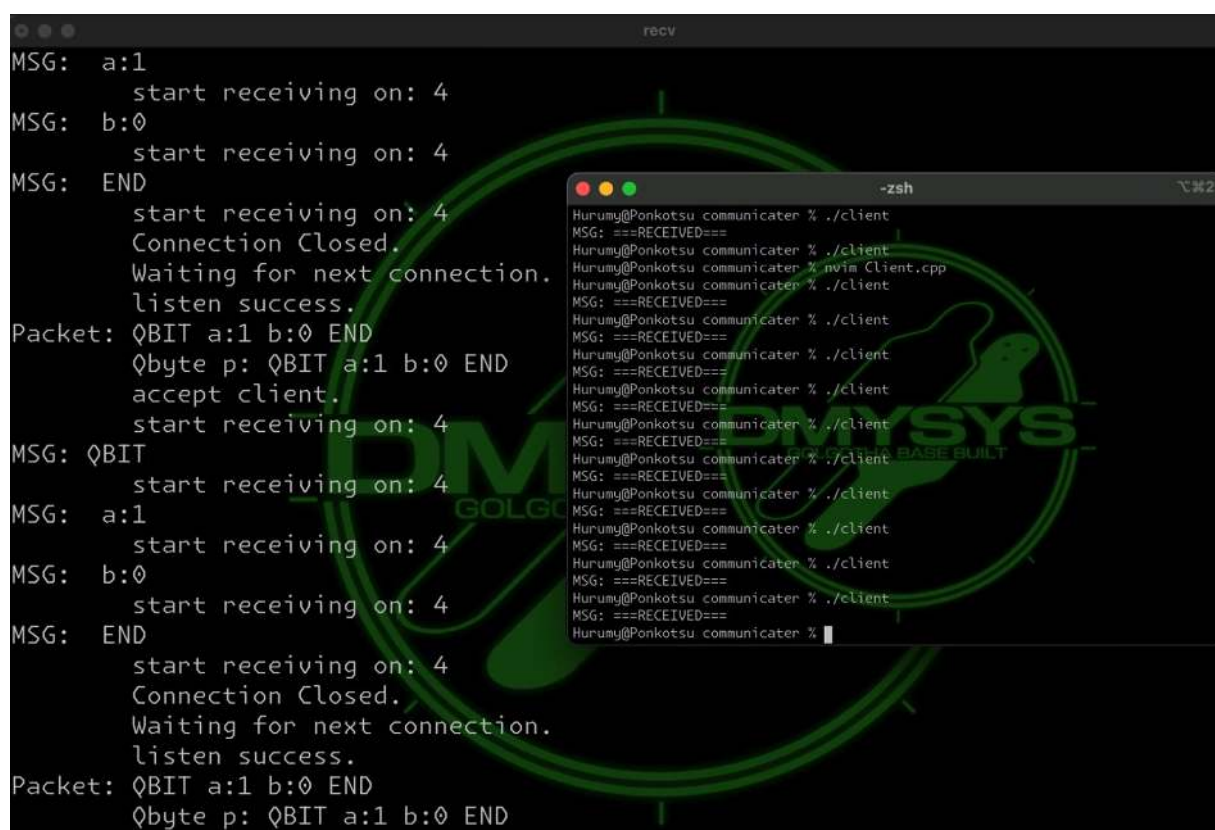
概要

量子力学的性質を利用して通信内容の暗号化を行うエミュレータを作成しました。プロトコルのみの実装です。

非常に容量が小さく、軽快に動作し、UIもUNIX Commandのそれを踏襲し極めてシンプルに作られているのが特徴です。(もちろん、メモリリークありません)

C++により実装されており、GitHubから誰でも使うことができます。

<https://github.com/Hurumy/stoq-qtos>



```
MSG: a:1
      start receiving on: 4
MSG: b:0
      start receiving on: 4
MSG: END
      start receiving on: 4
      Connection Closed.
      Waiting for next connection.
      listen success.
Packet: QBIT a:1 b:0 END
Qbyte p: QBIT a:1 b:0 END
      accept client.
      start receiving on: 4
MSG: QBIT
      start receiving on: 4
MSG: a:1
      start receiving on: 4
MSG: b:0
      start receiving on: 4
MSG: END
      start receiving on: 4
      Connection Closed.
      Waiting for next connection.
      listen success.
Packet: QBIT a:1 b:0 END
Qbyte p: QBIT a:1 b:0 END
```

```
Hurumy@Ponkotsu communicator % ./client
MSG: ===RECEIVED===
Hurumy@Ponkotsu communicator % ./client
Hurumy@Ponkotsu communicator % nvim Client.cpp
Hurumy@Ponkotsu communicator % ./client
MSG: ===RECEIVED===
Hurumy@Ponkotsu communicator % ./client
MSG: ===RECEIVED===
Hurumy@Ponkotsu communicator % ./client
MSG: ===RECEIVED===
Hurumy@Ponkotsu communicator % ./client
MSG: ===RECEIVED===
Hurumy@Ponkotsu communicator % ./client
MSG: ===RECEIVED===
Hurumy@Ponkotsu communicator % ./client
MSG: ===RECEIVED===
Hurumy@Ponkotsu communicator % ./client
MSG: ===RECEIVED===
Hurumy@Ponkotsu communicator % ./client
MSG: ===RECEIVED===
Hurumy@Ponkotsu communicator %
```

背景

量子分野といっても、量子コンピュータをはじめ、量子暗号、量子通信等様々な分野があります。そして、その全てが学際的な分野になっています。

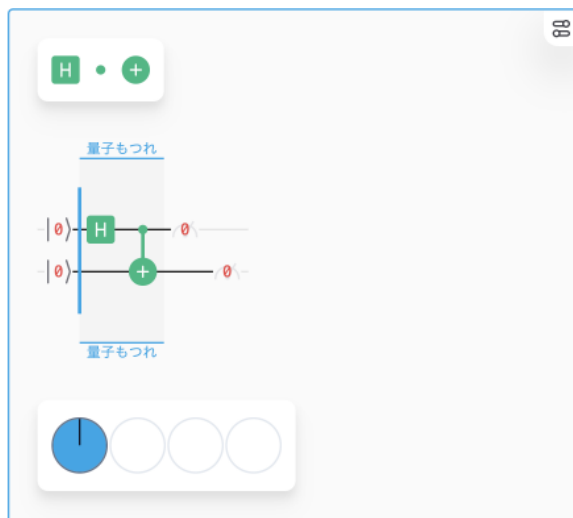
しかしながら、工学の人であれば量子コンピュータについてある程度知っているか？といえ、決してそうではないというような事情があります。それは数学の人でも物理の人でも同じで、各学問の中で量子分野はあまり認知を得られていない、というような状態になっている、と思っています。これはさらに言えば、ましてや一般人、という状態をも導いています。とにかく何をされていて何が問題になっているのか、当事者以外にはほとんど知られていないのです。

一方で、宇宙やDNAといったトピックは、子供向けの科学読本や雑誌で盛んに扱われ、世間的にもよく認識されています。これを踏まえて、わたしは子どもが気楽に使えるようなソフトウェアを開発し、宇宙分野のように親しんでもらうことが認知を高める近道になるのではないかと考えました。

当然、(おそらく)同じような問題意識に基づいて開発されたソフトウェアがすでに複数存在します。

ex1. Qni

QniはTIS Inc.が作成した量子コンピュータエミュレータです[1][2]。ポップでかわいいUIと、非常に親切かつ日本語のドキュメントが存在するのが特徴的です。



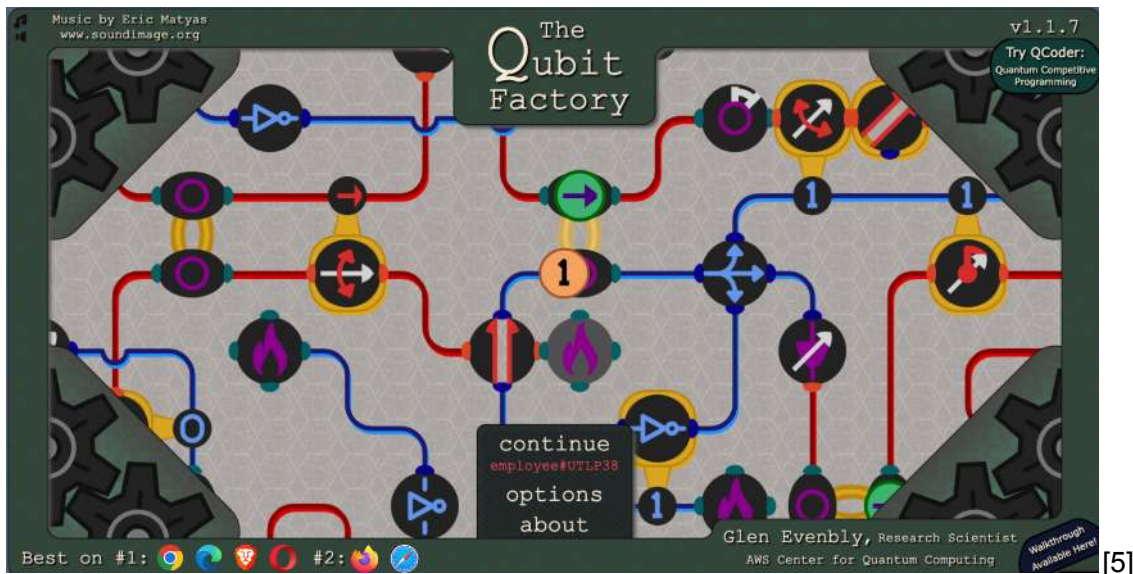
[3]

しかし、ドキュメントをじっくり読み込んで理解しながら遊ぶのが前提ようになっており、どちらかといえば量子回路を理解する教科書としてドキュメントが存在し、アプリ自体はその付録のようになっていると思います。

量子的性質を活かした有効なアルゴリズムを考えるのは情報学の最前線で行われているような研究であり、なんとなく動かしてみても思いつくようなものだとは考えにくいです。実際、現在でも量子的性質を活かしたアルゴリズムは60個程度しか発見されていません[4]。そのため、チュートリアルが終わっても特にやることはない、というような状態に陥ってしまいます。

ex2. The Qubit Factory

The Qubit FactoryはAWS量子計算センター所属のGlen Evenblyさんが開発した、量子回路をモチーフとしたパズルゲームです。[5]



プレイしてみるとわかると思いますが、パズルゲームとしての出来は素晴らしいですが、これは量子回路の知識があってもかなり難しいです。そして量子回路の知識がないと楽しめないと思います。敷居がかなり高いです。

さて、これらの試みを踏まえ、実際に自分自身で遊んでみて、何よりもネックなのが「パソコンなどとは異なり、自分で新しいものをつくる敷居があまりにも高い」ことだと感じました。

例えばScratchなどはプログラミングという作業を抽象化した上で作る楽しさを提供することでユーザーに学ぶモチベーションを与えています。しかし量子分野で「作る楽しみ」を提供するのは非常に難しいということがわかりました。

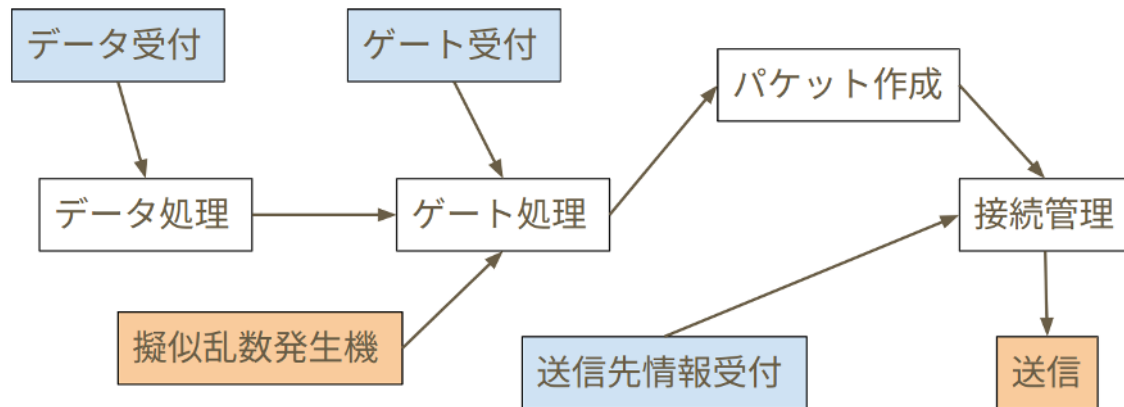
そこで、別の楽しさを提供しつつ、実際に学んで価値のあるような量子的アイデアを伝えられるような作品を作る必要があります。そこでわたしが目をつけたのが「争う楽しみ」です。

世の中には争いそのものを楽しむゲームは様々あります。例えばMinecraftのPvPサーバーなどでは敵の拠点の位置などの情報も重要な情報になる[6]ため、暗号通信の需要も存在するのではないかと考えました。ここに量子通信というソリューションを提供することで子どもに気軽に・気づかないうちに実際に触れてもらう、というアイデアを思いつきました。

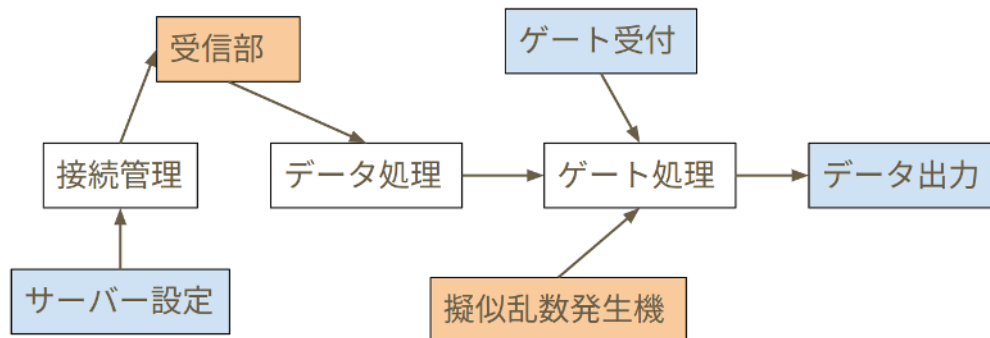
実装

主要部分の実装はすぐにできたのですが、どのような形で公開すべきかを大いに迷い、今のようなUNIXコマンドを踏襲した形式に落ち着きました。

送信部の設計:



受信部の設計:



以前はWebサーバーなども一緒に実装していたのですが、公開のしかたを改めるにつれてまるっとカットされました(結構大変だったのにカットしてしまい悔しいので供養)

なお、利用可能な量子ゲートは4種類しか実装していません。これはゲート数を増やすよりもむしろH,X,Y,Zゲートをまずシンプルな変換として理解してもらうことを重視しているからです。実際にセキュアに通信するためには、各文字、できれば各ビットに対して別々のゲートを適用することが重要です。これはUNIXコマンドを踏襲したUIにすることで容易に達成されました。

今後の展望

今後はこの作品を単にライブラリというところに留めず、とりあえず実際に自分自身でMinecraftのModを作成することを目標にしています。そのためにまずはMinecraftのマルチサーバーを建ててみるべく、実験用ということでTINSのグローバルIPを一ついただきました。(ありがとうございます！)

また、このソフトウェアは2024年度SecHack365で発表した[7]ものですが、メンターの中に量子系の人が全くおらず、一切アドバイスを受けることができませんでした。
そのため、自分自身本当に意味のあるものを作れているのか、根本的な勘違いをしていないか、大変怯えています。現在、量子系の先生にこの作品を見せてアドバイスを受けるため、いろいろな人に訪ねて回っているところです。

また、このソフトウェアにとどまらず、わたしが物理系のバックグラウンドと情報工学の技術の両方を持つことを活かし、よりそこでしかできないようなアイデアを以て有意義なものを作っていきたいと考えております。

参考文献

- [1]: GitHub Inc., GitHub, 2024.Aug.30
<https://github.com/qniapp/qni> (参照 2025.Mar.31)
- [2]: TIS Inc., Qni, 2024.Aug.30
<https://qniapp.net/> (参照 2025.Mar.31)
- [3]: TIS Inc., Qni Tutorial, 2024.Aug.30
<https://qniapp.github.io/qni/> (参照 2025.Mar.31)
- [4]: Stephen P. Jordan, Quantum Algorithm Zoo, 2025.Mar.21
<https://quantumalgorithmzoo.org/> (参照 2025.Mar.31)
- [5]: Glen Evenbly, QubitFactory, 2024.Nov.26
<https://www.qubitfactory.io/> (参照 2025.Mar.31)
- [6]: miraheze.org, 2b2t Wiki, 2025.Jan.24
https://2b2t.miraheze.org/wiki/Asgard_I (参照 2025.Mar.31)
- [7]: National Institute of Information and Communications Technology, SecHack365 2024年度 作品一覧, 2025.Mar.31
<https://sechack365.nict.go.jp/achievement/2024/> (参照 2025.Mar.31)

法律文書の自動解析

筑波大学 情報学群 情報科学類 産学間連携推進室
情報科学類 202212117 北野尚樹 [s2212117@u.tsukuba.ac.jp]

1 はじめに

法律の世界は、六法のような基本的な法律があり、それらが埋められない隙間をより詳細な規則を定める法令が埋め、それでも埋められない隙間を判例が埋めるような構造となっている。

法的な問題の解決を行う時、問題となっている行動が法的にどう分割されるのか、そしてそれぞれがどの法令や判例に対応しどのような判定になるのかをまとめ、最終的な結論を導き出す必要がある。問題の分割や結論を出す部分は依然として人が担当しなければならないところではあるが、細かい法令や判例を読みながら分割された細かい問題に対する判断をしていく過程をソフトウェアで支援することは可能である。細かい手続きを定めた法令は実務上とても重要でありながらその条文は長く複雑であることが多く、ソフトウェアによって自動で適切な注釈を付けることができれば、法務業務の負担軽減に繋がり社会的な意義があると考えている。

本プロジェクトでは、法令・条例・判例などの法律文書に対して読み替え規定文や略称定義文、他の法令などへの参照表現について抽出することを目指している。法令を解析して構造化する研究は様々あるが、本研究では特定の項目についてのみ解析を行い、必要な情報を抽出することに主眼を置いた。これにより、多く存在する例外への対処をしやすくなり、解析の対象としない項目については無視をすることでスムーズな検証が行えるようになることを期待している。

2 前年度までの成果の概要

2.1 読み替え規定文の解析

法律文書には準用と読み替えという概念が存在する。これは、一つの条文を使いまわしつつ必要最小限の修正を加えて、設置の目的などが似ている複数の規定を定義する手法である。使いまわす手法が準用と呼ばれ、準用の際に加える必要最小限の修正が読み替えである。図1のように、まず規定Aとそれを記述する条項Aが存在する。規定Aに似た新しい規定A'ができたとき、条項Aを読み替え規定で修正しながら規定A'の記述を行う。これが準用である。

準用の具体例を以下に挙げる。医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律（昭和三十五年法律第百四十五号）の第七条第四項では薬局の管理について以下のように定めている。

薬局の管理者（第一項の規定により薬局を実地に管理する薬局開設者を含む。次条第一項及び第三項において同じ。）は、その薬局以外の場所で業として薬局の管理その他薬事に関する実務に従事する者であつてはならない。ただし、その薬局の所在地

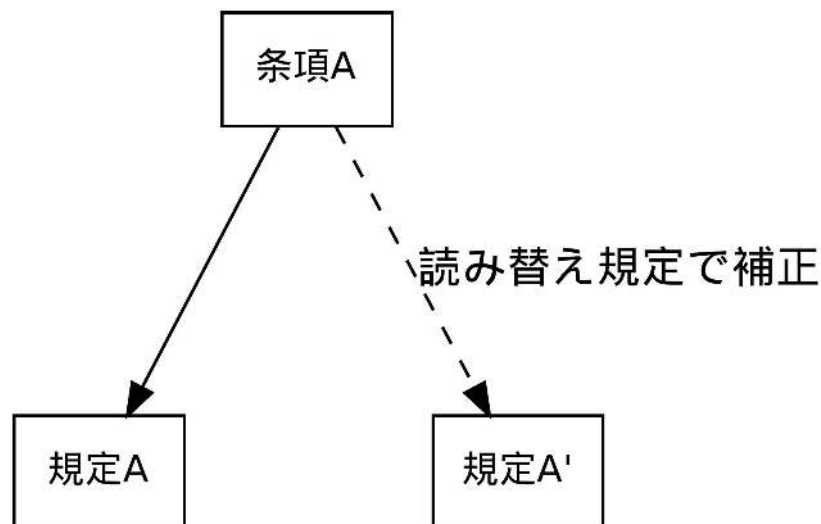


図 1：準用と読み替えの概念図

の都道府県知事の許可を受けたときは、この限りでない。

そして同法律の第八条第一項では薬局の管理者の義務について以下のように定めている。

薬局の管理者は、保健衛生上支障を生ずるおそれがないように、その薬局に勤務する薬剤師その他の従業者を監督し、その薬局の構造設備及び医薬品その他の物品を管理し、その他その薬局の業務につき、必要な注意をしなければならない。

そして同法律の第十七条第八項では医薬品等総括製造販売責任者について以下のように定めている。

医薬品製造管理者については、第七条第四項及び第八条第一項の規定を準用する。この場合において、第七条第四項中「その薬局の所在地の都道府県知事」とあるのは、「厚生労働大臣」と読み替えるものとする。

薬局の管理について規定している第七条第四項と薬局の管理者の義務について規定している第八条第一項を医薬品製造管理者について準用することで、語句が少し違うだけの条文を再度書くコストを減らしている。この条文を読む人は「医薬品製造管理者について準用する」という文言をもとに、第七条第四項と第八条第一項内に登場する「薬局の管理者」という語を「医薬品製造管理者」に置換し、第十七条第八項の条文が意図する文章を考えて読むこととなる。ただし、薬局に許可を出すのが都道府県知事であるのに対して、医薬品製造所に対して許可を出すのが厚生労働大臣であるという制度上の差異があり、これは非自明であるため条文中で読み替えの指示を行っている。

この読み替え規定文を解析し、読み替えられる語と読み替える語のペアをパターン

ベースで抽出する手法を提案した。提案手法は読み替え規定文中に含まれる鍵括弧とその鍵括弧の直後の語に着目するものである。2022 年 12 月 4 日時点で公開されていた法令データに対して提案手法で解析を行ったところ、読み替え規定文 11672 個中、11428 個は抽出に成功し残りの 244 個は失敗した。

このように単語の出現するパターンのみで読み替え規定文のうち 97% 以上の読み替え規定文を解析することができた。今後の課題として解析に失敗した条文を調べたところ、大きく分けて

- 文中の括弧の対応が壊れているもの
- 表形式で与えているもの
- 読み替える語等の指定に鍵括弧をしていないもの

の 3 つに分けられることがわかった。今後はこれらの括弧の対応がとれていない条文や表形式で与えられている条文の自動検出と表の解析機能を実装することが目標である。また、後述する法令名の自動抽出機能と組み合わせることで、読み替え規定文適用後の条文の自動生成を行えるようになりたいと考えている。

2.2 判例の自動収集およびテキストデータ化

法律分野の文書の中では法令だけではなく判例も重要である。法令が埋められない穴を判例によって埋めていき、次の類似事例の時の判断に活かされるからである。そのため、判例の解析も法令と同様に行う必要がある。当時は判例のテキストがオープンデータとして公開されているものが無かった、これを作成した。

裁判所が公開している裁判例検索という Web サービス^{*1}をスクレイピングし、判例文の PDF ファイルと判例に紐付けられている事件番号や法廷名と紐付けて保存できるようなソフトウェアを作成した。

作成したデータは年代ごとにわけ、GitHub で CC0 ライセンスのもと公開している^{*2}。しかし、PDF ファイルからのテキスト抽出がうまく行かないことがあり、今後もテキスト抽出の精度を上げる必要がある。

また、この収集および解析の過程で図2のように年月日にマイナスの値が出現している詳細ページが7つ存在することを確認した。発見した翌日である 2023 年 1 月 16 日に最高裁判所に報告を行ったところその数日後に全てのページが図3のように修正された。

2.3 法律文書解析用ライブラリの整備

法律文書を解析したり収集したりして生成したデータを統一的なインターフェースで提供できるようにするためにライブラリの整備を行った。

日本政府が法令のデータを公開している e-Gov 法令検索^{*3}では、公開データの形式を

^{*1} https://www.courts.go.jp/app/hanrei_jp/search1

^{*2} https://github.com/japanese-law-analysis/data_set

^{*3} <https://laws.e-gov.go.jp/>

原審裁判所名	仙台高等裁判所
原審事件番号	
原審裁判年月日	<u>昭和-954年3月25日</u>

図 2：昭和 -954 年という表記（下線筆者）

原審裁判所名	仙台高等裁判所
原審事件番号	
原審裁判年月日	<u>昭和46年3月25日</u>

図 3：昭和 -954 年という表記が修正されたあとのページ（下線筆者）

「法令標準 XML スキーマ」^{*4}で定めている。これに対応し、XML 文字列から構造を生成するパーサーや構造から XML 文字列への変換を行う関数を実装した。また、地方自治体の条例などの解析にも対応するために、素のテキストからデータ構造への変換の関数も実装した。これらをライブラリとして Rust のパッケージマネージャに登録し^{*5}、OSS として誰でも自由に使えるようにした（図4）。

法令標準 XML スキーマだけではカバーすることができない解析結果を表す型や法律などのメタデータを表す型などを定義したり、解析ツールを作成する上であると便利な関数を定義したりしたライブラリを作成し、OSS として誰でも自由に使えるように公開した^{*6}。これにより、解析で得られた大量のデータをこのライブラリを通じて他のユーザも簡単に扱うことができるようになり、解析結果を組み合わせるソフトウェアを作りやすくなることが期待される。

^{*4} https://elaws.e-gov.go.jp/file/XMLSchemaForJapaneseLaw_v3.xsd

^{*5} https://crates.io/crates/japanese_law_xml_schema

^{*6} https://github.com/japanese-law-analysis/jplaw_tools



図 4：法令標準 XML スキーマライブラリのページ

3 今年度の成果

3.1 パターンベースの定義規定・略称規定の解析

3.1.1 背景

法律文中には、その法律文中でのみ使用される用語を定義したり、長い用語に対して略称を定める文が存在する [5, 6]。具体的な例としては、例えば次の文では「細胞加工物の原材料となる人または動物の細胞」という正式名称に「細胞」という略称が対応している。

この法律において「細胞」とは、細胞加工物の原材料となる人または動物の細胞をいう。

また、次の文では「地方公共団体金融機構」という正式名称に「機構」という略称が対応し、「当該募集に応じて当該機構債券の引受けの申込みをした者に対して割り当てる機構債券」という正式名称に「募集機構債券」という略称が対応している。

地方公共団体金融機構（以下「機構」という。）は、その発行する機構債券を引き受ける者の募集をしようとするときは、その都度、募集機構債券（当該募集に応じて当該機構債券の引受けの申込みをした者に対して割り当てる機構債券をいう。以下同じ。）について次に掲げる事項を定めなければならない。

これらは定義規定・略称規定と呼ばれており、文書全体で高頻度で利用されるため、読者が読む際に混乱や負担を生じさせる原因の一つとなっていると指摘されている [1, 2]。そこで、こうした定義規定・略称規定を法律文書中から自動的に特定し、それらに注釈情報を付与できれば法律文書の可読性向上や読者の理解促進に大きく寄与できることが期待される。しかし、法律文書中の定義規定・略称規定は書式が多様であり、既存の手法では高精度な抽出が実現できていなかったため、今回はより高精度で網羅的な正式名称と略称の抽出手法の開発に取り組むこととした。

3.1.2 関連研究

中村ら [3, 4] は、法令文中の定義規定・略称規定に焦点を当て、正規表現やパターン処理により正式名称と略称の抽出する手法を提案した。具体的には、定義規定・略称規定文の代表的な書式を定義し、それに対応する正規表現を開発することで自動抽出を実現した。しかし、整理したパターンが限定されており、抽出漏れの発生が課題となっている。

また、節2.1での読み替え規定文の解析ではパターンの構築手法は、定義規定・略称規定の解析にも応用可能な面があるが、読み替え規則と定義規定・略称規定の文型も文法的な目的も大きく異なるため、単純に直接適用することは難しく。新たなアプローチが求められる。

3.1.3 提案手法

定義規定・略称規定を観察し、次の5つの文型に分類した。

- トイウ形：○○（以下「△△」という。）
- トハ形：この法律において「△△」とは、○○をいう。
- ヲイウ形：△△（○○をいう。）
- ヲイウ - 規定補足形：△△（□□に規定する△△をいう。）
- ヲイウ - トウ圧縮形：△△等（△△又は□□をいう。）

そして、この文型に対応するように次のようなパターン処理で正式名称と略称を抽出するようにした。

1. ヲイウ形とその派生形・トハ形・トイウ形に該当する法令文を抽出
2. 括弧書き中のテキストと位置を記録し除去
3. トハ形の場合、法令文中の「とは、」と「をいう。」・「をいい、」の間を正式名称と、「とは、」の直前の鍵括弧内を略称と定義
4. ヲイウ - トウ圧縮形の場合、括弧の直前の「等」の直前で括弧書きに含まれる最長のものを略称と、弧書きの頭から「をいう。」・「をいい、」までを正式名称と定義
5. ヲイウ - 規定補足形の場合、括弧書き内の「に規定する」と「をいう。」・「をいい、」の間を略称と、括弧書きの頭から「をいう。」・「をいい、」までを正式名称と定義
6. ヲイウ形の場合、括弧書きから「ひらがなの直後の読点」もしくは文頭までを略称と、括弧書きの頭から「をいう。」や「をいい、」までを正式名称と定義
7. トイウ形の場合、括弧書きから「ひらがなの直後の読点」もしくは文頭までを正式名称と、括弧文中の鉤括弧内を略称と定義
8. 記録していた括弧書きを元の箇所に復元

9. 以上の 2-7 までの操作を括弧書き中に再帰的に適用
実装は Rust で行い、GitHub で公開している*7。

3.1.4 実験

施行中の法令からランダムで抽出した 401 個の定義規定・略称規定に対し、手動で正式名称と略称のアノテーション処理を行ったものを評価用のデータとして用いた。これに対して既存手法である中村らの手法と今回の提案手法を適用した。結果を、正式名称と略称それぞれの正誤の組み合わせにおける頻度割合をあらわしたものが表1と表2になった。

表 1：既存手法の処理結果

	略称正答	略称誤答
正式名称正答	22%	6%
正式名称誤答	7%	64%

表 2：提案手法の処理結果

	略称正答	略称誤答
正式名称正答	63%	13%
正式名称誤答	22%	2%

また、近年発展が著しい LLM による解析とも比較することとした。GPT-4o mini と o1-mini の二つの言語モデルに対し、整理した 5 つの文型をもとにした few-shot プロンプトを与えて解析させた。処理結果について、正式名称と略称それぞれの正誤の組み合わせにおける頻度割合をあらわしたものが表3と表4である。

表 3：GPT-4o mini の処理結果

	略称正答	略称誤答
正式名称正答	17%	1%
正式名称誤答	51%	31%

表 4：o1-mini の処理結果

	略称正答	略称誤答
正式名称正答	36%	1%
正式名称誤答	30%	33%

LLM による解析結果と提案手法での解析結果について、正式名称のテキスト長ごとに正答率を比較してみると、LLM による正答率は表5に示した通りとなっている。

3.1.5 考察

実験結果より、提案手法は正式名称と略称の両方を正しく抽出する能力が既存手法と LLM による解析よりも高く、正式名称と略称のどちらかだけでも抽出できる割合も 98(63 + 13 + 22)% となっており、これも既存手法と LLM による解析よりも高い値となっている。

*7 https://github.com/japanese-law-analysis/pattern_based_formal_and_abbr_extraction

表 5：正式名称のテキスト長別の各手法の正答率

	0-20 字	20-40 字	40-60 字	60-80 字	80-100 字	100 字以上
GPT-4o mini の正答率	29%	13%	5%	5%	0%	0%
o1-mini の正答率	56%	27%	28%	19%	18%	5%
既存手法の正答率	27%	18%	20%	30%	18%	25%
提案手法の正答率	68%	65%	46%	61%	40%	50%

また、LLM による解析では正式名称のテキスト長が長くなると正答率が下がるという傾向があり、このことよりテキスト長が長くなりやすい法令文の解析に対して提案手法が頑健であると言える。

3.1.6 言語処理学会第 31 回年次大会での報告

本研究は産学間連携推進室の卒業生である東京科学大の西山大輝さんとの共著として言語処理学会第 31 回年次大会で「法令文の可読性向上のための定義規定・略称規定における文型定義及びパターンベースの正式名称・略称抽出手法」というタイトルで発表した [7]。

3.1.7 まとめ

定義規定・略称規定について文型を整理し、対応するパターンベースの解析手法を開発したことにより、既存手法や LLM を利用した手法よりも精度の良い正式名称と略称の抽出を行うことができるようになった。

一方で、パターン化できない文型の存在や、意味情報が必要な並列表現の解析が必要となるケースの存在などがあり、これらに対応できるような処理方法を検討する必要がある他、今回抽出した手法を実際にユーザーに提示するソフトウェアの作成などを検討する必要がある。

3.2 法令検索 Chrome 拡張アプリ "lawscape" の開発

3.2.1 背景

文書に関連する法令を検索するには、従来は人の手で持っている知識をもとに e-gov 法令検索で法令名やキーワード検索を行う必要がある。しかしこれは検索を行う人の知識に強く依存しており、多くの人にとって難しいタスクとなっている。また、一つの法令にたどり着いたとしてもそれに関連する法令をさらにたどっていく作業も同様の課題を抱えている。そのため、今回は法令検索を自動で行うアプリを作成した。

3.2.2 関連事例

法令の検索という観点からは、有料サービスである D1-Law^{*8}や Legalscape^{*9}などがすでに存在している。これらのサービスは法令以外にも判例や解説書についても検索することができるという利点がある一方、専用の検索ページで専門知識を持つ人が使うことを前提としたものとなっており、一般の人が気軽に検索するためのツールとなっていない。

そこで、今回一般の人が気軽に使えるための検索ツールを作成することとした。

3.2.3 作成したもの

Web ブラウザ上に表示されたテキストを選択して検索を行うと、そのテキストに関連した法令や判例を表示する Chrome 拡張アプリを作成した。この Chrome 拡張アプリを使うと図5と図6のように、ブラウザ上のテキストで法令検索を行うことができるようになる。

この Chrome 拡張アプリは"lawscape"という名称で開発し、GitHub 上で MIT ライセンスのもと公開している（図7）。

3.2.4 実装

Chrome 拡張アプリそのものは JavaScript を用いて実装した。このアプリでは次の手順に従って検索と結果の表示が行われる。

1. ユーザーが選択したテキストをアプリが受け取る
2. 事前に用意した検索用 API サーバーにテキストや検索用のパラメータ値を送り、検索を行う
3. 検索結果の JSON データを整形して表示する。

検索用 API サーバーは節2.3で整備した法律文書用ライブラリを使って加工したデータを meilisearch^{*11}検索エンジンに登録し、外部に対して Web API として公開するサーバーである。このサーバーでは次の手順で検索を行う。

1. 検索エンジンに対して検索クエリを発行し、結果を受け取る
2. 結果の法令間の参照状況を解析し、結果を返す

これにより、単なる名称や内容のベクトルの類似度だけではなく、関連する法令同士をまとめてユーザーに提示することができるようになり、より利便性が向上するようになる。

3.2.5 デジタル庁法令ハッカソンでの発表

本アプリはデジタル庁が 2025 年 2 月から 3 月にかけて開催した “「法令」×「デジタ

^{*8} <https://www.daiichihoki.co.jp/d1-law/>

^{*9} <https://www.legalscape.jp/>

^{*11} <https://www.meilisearch.com/>



図 5：Web ブラウザ上の「電子申請」という単語を選択したときにあらわれる「関連する法令文書を検索」という選択画面



図 6：「電子申請」というテキストの検索結果の表示

ル」ハッカソン”^{*12}において発表した（図8）。

3.2.6 まとめ

^{*12} <https://www.digital.go.jp/news/9fb5ef8e-c631-4974-96d9-0b145304c553>

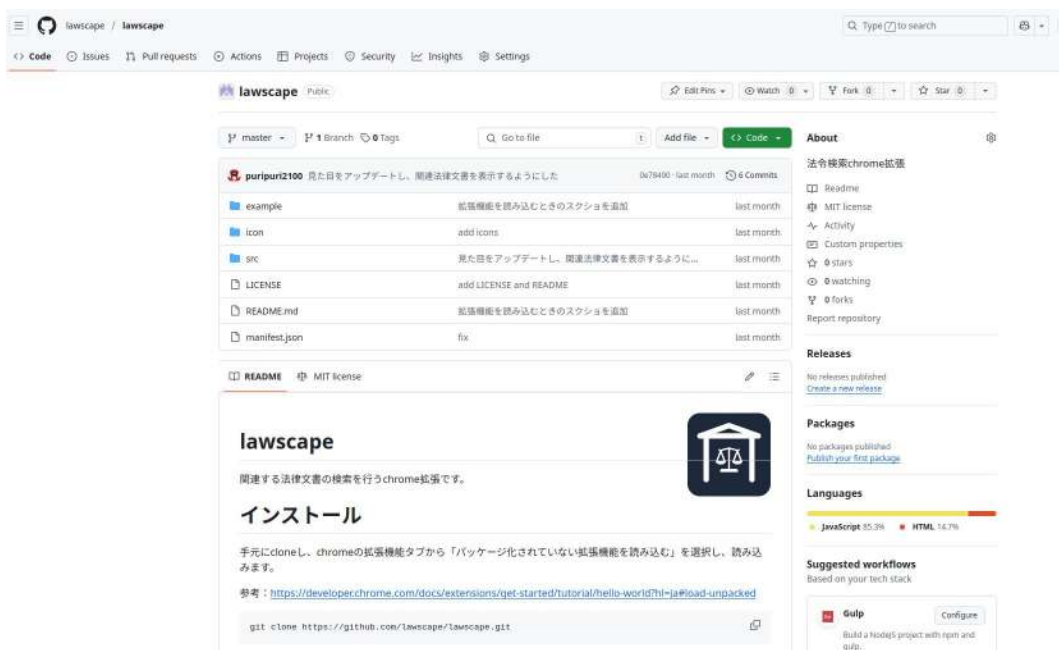


図 7：lawscape のリポジトリ *10

作品名：lawscape チーム名：法令API楽しみ～部

作品の概要

- 法制執務者や法務に従事する者が関連する法律文書を検索し、目的の文書にたどり着きやすくする機能を提供する作品。入力した単語の類似度判定により関連法令を探しブラウザ上で表示することができる。

社会に対して生み出す価値

- 法令などの法律文書は増え続けており、法制執務者などが取り扱わなければならないトピックも増えており、適切な前例や文献にたどり着くコストが高まっている。
- 本サービスにより検索コストを大きく低減することができるようになり、人が創造性を発揮する作業に集中できるようになる。

新規性

- 本作品は、検索エンジンのチューニング可能な点やブラウザ（Chrome）拡張という形での検索のしやすさという点で革新的な作品。
- 単なるキーワード検索ではなく、機械学習モデルを用い、法令XMLデータ中のテキストと検索ワードの類似度判定を行う。

作品の画面イメージ

17

図 8：lawscape の “「法令」×「デジタル」ハッカソン” での発表資料 [8]

検索結果や解析結果を実際にユーザーに提示し使ってもらうためのインターフェースの作成を行った。本アプリの作成を通じて、法律文書からの抽出した情報を人が認知し扱いやすい形に加工することの難しさを感じた。また、法令ハッカソンへの参加を通し、現場の行政職員が必要としている機能や UI などの情報を調査することができた。

4 おわりに

本年度では、昨年度までの取り組みについて第 57 回情報科学若手の会で発表を行った

り(図9)、法令文中の定義規定・略称規定から正式名称と略称を抽出するためのパターンベースの開発を行い自然言語処理学会第31回年次大会で報告したり、Webブラウザ上に表示されたテキストに関連する法令を検索することができるChrome拡張アプリを作成しデジタル庁法令ハッカソンで発表したりした。

ショート発表 3:「係り受け解析を用いた法律文書中の略称規定の解析についての報告」 北野 尚樹 様

法律文書では条件を丁寧に指定したためにとても長くなってしまった単語が多く存在している。これらの長い単語を毎回使うのは煩雑で読みにくくなるため、略称を定めて略称の方を使用することが多い。例えば税理士法第四十八条の六では次のような定義で「社員等」という略称を定め使用している。>当該税理士法人の社員又は使用人である税理士(以下この条及び第四十八条の二十四項において「社員等」という。) 略称を使った条文を読む際に略称の定義を参照して詳しい条件を確認したいという需要が存在する。これに応えるために法律文書中の略称規定を解析する手法を考案した。複雑で長い規定文に対応するためには単純なパターンマッチでは不可能であったため、係り受け解析の結果を使った解析方法を考え、実装した。実際の法令に対して実装したプログラムを適用した結果について報告する。

図9：第57回情報科学若手の会での発表概要^{*13}

また、本年度の活動を通じて行政職員やリーガルテックの会社の方々や研究者の方々など、関連する分野に取り組んでいる方々との意見交換や今後の連携について話し合う機会を得ることができた。これらの機会を活かし、今後も研究を発展させていきたいと考えている。

5 謝辞

本プロジェクトを進めていくうえで産学間連携推進室のメンバーや卒業生の西山大輝さん、山本幹雄先生、川見唯史弁護士、森田岳人弁護士、中川由賀先生(中京大学)、中山幸二先生(明治大学)、伊澤文平弁護士等の皆様よりアドバイスを頂きました。また、産学間連携推進室の担当をしてくださっている新城靖先生や弁護士や法学の研究者の方々を紹介していただいた伊藤誠先生にもお世話になりました。この場をお借りしまして、深く感謝を申し上げます。

参考文献

- [1] Eric Martinez, Francis Mollica, and Edward Gibson. Poor writing, not specialized concepts, drives processing difficulty in legal language. *Cognition*, Vol. 224, p. 105070, 2022.
- [2] 竹井直樹, 柴田文明. 保険約款と保険商品のわかりやすさの向上について. *損害保険研究*, Vol.72, No.2, pp. 109-127, 2010.
- [3] 中村誠, 小川泰弘, 外山勝彦. 法令文中において括弧書きで定義されている法令用語とその語釈文の抽出. *言語処理学会第19回年次大会発表論文集*. 2013, pp.670-673
- [4] Makoto Nakamura, Ryusei Kobayashi, Yasuhiro Ogawa, and Katsuhiko Toyama. A pattern-based approach to hyponymy relation acquisition for the agricultural thesaurus. In *Proceedings of International Symposium on Agricultural Ontology Service 2012 (AOS2012)*, 2012.
- [5] 石毛正純. 法制執務詳解 新版 III. 株式会社ぎょうせい, 2022.
- [6] 磯崎陽輔. 分かりやすい法律・条例の書き方. 株式会社ぎょうせい, 2011.
- [7] 北野尚樹, 西山大輝. 法令文の可読性向上のための定義規定・略称規定における文型定義及びパターンベースの正式名称・略称抽出手法. *言語処理学会第31回年*

^{*13} <https://wakate.org/2024/11/18/57threport/>

次大会発表論文集. 2025, pp.2243-2248

- [8] デジタル庁. “「法令」×「デジタル」ハッカソン作品一覧”. https://www.digital.go.jp/assets/contents/node/information/field_ref_resources/9fb5ef8e-c631-4974-96d9-0b145304c553/fdb9e06d/%E4%BD%9C%E5%93%81%E4%B8%80%E8%A6%A7%EF%BC%88PDF%E4%BC%8F28.4MB%E4%BC%89.pdf, 2025 年 3 月 31 日閲覧.

モバイル回線を組み合わせた頑健な管理用サーバの構築

情報メディア創成学類 服部 真吾 (s2111559@u.tsukuba.ac.jp)

2025 年 3 月 31 日

概要

本報告は、産学間連携推進室における研究活動の成果として、ネットワーク障害発生時においても遠隔管理を可能にする「管理用トラフィック専用モバイルバックアップ回線」の構築と実装について述べるものである。研究室やデータセンターなどの拠点では、ネットワーク障害時に物理的なアクセスが困難な状況が発生することがあり、迅速な復旧が難しいという課題がある。本研究では、Raspberry Pi を用いた低コストかつ柔軟性の高いソリューションを提案する。本システムは、中央のコントローラサーバに対して各ホストが SSH リバーストンネルを確立し、管理者はコントローラサーバを介して各ホストに簡単にアクセスできる仕組みとなっている。特に、主回線障害時には管理用トラフィックのみをモバイル回線に切り替え、SSH リバーストンネルを活用することで、効率的な遠隔管理を実現した。この手法は、一般のトラフィックとは独立した管理経路を確保することで、ネットワーク復旧時間の短縮と運用コストの削減に貢献するものである。

1 はじめに

ネットワークを基盤とする研究環境や組織において、回線障害は避けられない問題である。特に主要な対外接続が失われた場合、拠点に物理的に訪問して対応せざるを得ないケースが多い。設置箇所が近傍であればこの対応も容易だが、データセンターなどの施設では入館手続きや移動に時間を要し、復旧が大幅に遅れる原因となる。

産学間連携推進室は、研究室配属前の学生に対して研究環境を提供することを目的とした研究室であり、様々なネットワークセグメントやトラフィックが混在している。この環境では、新たなネットワーク技術の検証や実験が日常的に行われており、その過程で意図せず対外接続が失われる可能性がある。例えば、ルーティングプロトコルの設定変更、VXLAN 構成の実験、ファイアウォールルールのテストなどにより、ネットワークが一時的に不安定になることがある。

本研究の目的は、実験や不慮の事故により対外接続が失われた場合でも、遠隔から状況調査や復旧作業を行える仕組みを構築することであり、平時と非常時の両方において安定的に利用できる管理用システムの実現を目指す。特に注目したのは、全てのトラフィックではなく「管理用トラフィックのみ」をモバイル回線に切り替えるアプローチである。これにより、低コストかつ効率的なバックアップ経路を確保し、ネットワーク管理の安定性と柔軟性を向上させることを目指した。

2 課題の所在

研究室やデータセンターのようなネットワーク環境における管理機能のバックアップに関して、複数の課題が存在する。まず第一に、経済的な課題がある。既存の専用機器を用いたバックアップソリューションは一般に高価であり、限られた予算内で運用される研究室環境においては導入障壁が高い。また、こうした専用機器は柔軟性に欠け、研究用途における様々な実験的構成に対応しきれないことが多い。

第二に、トラフィック管理に関する技術的な課題がある。障害発生時に全てのトラフィックをモバイル回線へ迂回させる方式は、モバイル回線の帯域制限や通信量制限を考慮すると現実的ではない。特に研究環境では、大容量データの転送や継続的な実験通信が発生するため、限られたモバイル回線の容量をすぐに使い果たしてしまう恐れがある。また、全てのトラフィックをモバイル回線で処理しようとする、暗号化やトンネリングのオーバーヘッドにより通信効率が低下し、システム全体のパフォーマンスに悪影響を及ぼす可能性がある。

最後に、運用管理における複雑さの課題がある。冗長性を備えたシステムは一般に構成が複雑になる傾向があり、日常のメンテナンスや障害対応が煩雑になりがちである。特に学生が主体となる研究室環境では、運用知識の継承や手順の標準化が難しく、引き継ぎ時に情報が失われるリスクが高い。また、複雑な冗長構成は設定ミスを引き起こしやすく、かえって安定性を損なう原因となることもある。

これらの課題を総合的に解決するためには、(1) 低コストで実現可能、(2) 必要最小限のトラフィックのみを扱う、(3) シンプルで保守性の高い、という条件を満たすソリューションが求められる。特に学生が主体となって運用する産学間連携推進室のような環境では、これらの条件を同時に満たすことが重要である。

3 関連研究

3.1 モバイル回線を用いたバックアップネットワークと OOBM の事例

近年、企業や大学のネットワークインフラにおいて、4G/5G などのモバイル回線をバックアップ経路やアウト・オブ・バンド管理 (OOBM) の手段として活用する事例が増加している。これは主に、主回線が停止した場合でも業務継続性を確保するためのソリューションとして注目されている。

小売業界での実装例として、多店舗展開する企業が LTE ルーターを各店舗に設置し、主回線障害時には自動的にモバイル回線へフェイルオーバーする仕組みを導入している事例 [1] がある。Opengear 社が提供するモバイル回線対応コンソールサーバー [2] などは、遠隔地に設置された重要インフラ機器に対して安全な OOBM アクセスを提供し、主回線停止時や新拠点立ち上げ時でも WAN 接続を維持する用途で広く利用されている。

学術研究環境や小規模拠点においても、実用的かつ低コストでモバイル OOBM を実現する取り組みが見られる。技術系ブログなどでは、Raspberry Pi とモバイル Wi-Fi ルーターを組み合わせ、インターネット経由で遠隔地からネットワーク機器のシリアルコンソールにアクセスできる OOBM 装置の自作例が報告されている [3]。従来、ネットワーク機器のコンソール管理には専用のアナログ電話回線とモデムの組み合わせが使用されてきたが、コスト面の問題から現代では敬遠される傾向にある。こうした背景から、安価な Raspberry Pi と 4G 回線を活用することで、回線障害や設定ミスにより SSH ログインが不可能になった場合でも、モバイル回線経由でコンソール操作が可能な復旧経路を確保する手法が注目を集めている。

3.2 安価なハードウェアを用いた実装例の調査

低コストでモバイルバックアップ回線や OOBM を実現する試みとして、Raspberry Pi ベースの実装例が報告されている。典型的な構成では、Raspberry Pi にモバイルバックアップ回線を接続することで回線の冗長化を行い、この上で USB シリアル変換ケーブルを接続してネットワーク機器のコンソールポート等と接続する。このセットアップにより、主回線が利用不能になったとしても、Raspberry Pi はインターネットにアクセスすることができるため、外部からの操作が可能となる。この際に、USB 接続 LTE モデム [4] を使用し、Raspberry Pi の回線を冗長化する例報が報告 [5] されている。このモデムは Raspberry Pi に USB 接続すると標準の USB ネットワークカード (NIC) として認識され、単体でルーティング機能も提供する。この構成では、Raspberry Pi に直接 LTE 通信機能を持たせられるため、別途モバイル Wi-Fi ルーターを用意する必要がなく、さらに構成をシンプルにできる利点がある。

ハードウェアコストの面では、Raspberry Pi 本体 (約 5,000 円)、モバイル Wi-Fi ルーターもしくは USB 接続 LTE モデム (約 5,000~10,000 円)、SIM 通信料 (月額 1,000~3,000 円程度) の組み合わせで、市販の産業用 OOBM ソリューション (数十万円) と比較して大幅なコスト削減が可能である。また、こうした自作アプローチでは柔軟なカスタマイズも容易であり、特定の用途に最適化された構成を実現できる利点もある。

自動化とメンテナンス性の向上も重要な検討事項である。実用的な実装例 [3] では、Raspberry Pi 起動時に自動的に SSH リバーストンネルを確立し、万一の接続断時にも一定間隔で再接続を試みるスクリプトが実装されている。また、トンネルの健全性を定期的に確認し、「ゾンビ状態」(プロセスは生きているが実際には通信できない状態) を検出・修復する仕組みも採用されている。こうした工夫により、低コストハードウェアでありながら実用的な信頼性を持つ OOBM ソリューションが実現されている。

学術研究機関においても、九州大学などで Raspberry Pi と LTE 通信モデムを組み合わせた遠隔監視・データロガー装置の開発事例が報告されている。これらの研究では、商用製品と比較して大幅なコスト削減を実現しつつも、実用的な性能と信頼性を持つシステムの構築が示されている [6]。

このように、Raspberry Pi や Pixela 製 LTE モデムなどの安価なハードウェアコンポーネントは、小規模拠点やラボにおけるバックアップ回線や OOBM 機能の実現ハードルを大きく下げている。

る。これにより、従来は専門機器の導入が必要だった遠隔管理や予備回線の確保が、低コストな自作ソリューションでも実現可能となっている点が注目される。

3.3 管理用トラフィックのみモバイル回線へ切り替える設計思想

バックアップ回線として限られた帯域のモバイル回線を効果的に活用するには、すべてのトラフィックではなく管理用トラフィックのみをモバイル回線に切り替える設計が多くの利点を持つ。

まず、セキュリティの観点から見ると、管理用ネットワークを本番ネットワークから物理的・論理的に分離し、モバイル回線という独立した経路で接続することで、社内 LAN がウイルス感染や攻撃を受けた場合でも管理ネットワークへの影響を最小限に抑えられる。さらに、OOBM インターフェース経由でのみ管理通信を行うことで、管理トラフィックがインターネットに直接露出せず、閉域網として構成できる利点もある。

次に、帯域利用効率の面でも大きな優位性が認められる。一般に、ネットワーク管理に関わる通信量は全体トラフィックと比較して極めて小さい。そのため、管理用トラフィックのみをモバイル回線に迂回させることで、モバイル回線の帯域制限（月あたり数 GB 程度）の範囲内で十分に運用可能である。対照的に、大容量のユーザーデータ（動画会議やファイル転送など）をモバイル回線に迂回させた場合、すぐに帯域不足に陥り、高額な追加料金が発生するリスクがある。管理トラフィックに限定することで、低速・低容量のモバイル回線であっても十分な安定性と経済性を両立できる。

第三に、実際の障害対応能力の向上も重要な利点である。管理用トラフィック専用のバックアップ経路を確保することで、万一メインネットワークが完全にダウンした場合でも、管理者は LTE 経由でルータやスイッチのコンソールインターフェースや管理 GUI にアクセスできる。これにより、本番トラフィックは一時的に停止した状態であっても、復旧作業を遠隔から即座に開始できる。結果として、物理的な訪問を必要とせず、ダウンタイムの大幅な短縮が可能となる。

産業用途では、Opengear 社などが提供する製品が「事業継続性の確保」という観点からこの設計思想を採用している。また一部の企業では、通常は光回線を使い、障害検知時のみ管理用にモバイルルーターを自動起動する仕組みを導入し、「使わなかった時ゼロ円 Wi-Fi」[7] などのサービスと組み合わせることで、待機時のコストをゼロに抑えつつ、いざという時の管理アクセスを確保する工夫も見られる。

3.4 VPN と SSH リバーストンネルの比較分析

モバイル回線を介して遠隔ネットワークに安全に接続する方法として、VPN 接続（特に SoftEther VPN, L2TP や IPsec による L2VPN）と SSH リバーストンネルという二つの主要なアプローチが考えられる。これらは通信量、セキュリティ、構成の複雑さ、運用コストなどの観点で異なる特性を持つ。

通信量と帯域使用効率の観点では、VPN プロトコルは一般に暗号化やカプセル化に伴う追加の

プロトコルオーバーヘッドが発生し、データ使用量が増加する傾向がある。例えば L2VPN 実装の一例である L2TPv3 では、Ethernet フレームに対してさまざまなヘッダが付加されるため、その分トラフィック量が増える。また、OpenVPN のような汎用 VPN ではデータオーバーヘッドがさらに大きくなる。これに対して SSH リバーストンネルは、基本的に単一の SSH 接続 (TCP) 上で必要なポートのみを転送する方式となるため、余分なヘッダ情報は少なく抑えられる。特にネットワーク全体をブリッジ接続する L2VPN では、ARP ブロードキャストなどの不要なパケットまで転送される可能性があるのに対し、SSH トンネルであれば管理用ポート (SSH/22 番ポートやコンソールポートなど) に限定して転送することが可能である。結果として、モバイル回線上的データ使用量を最小限に抑えられる利点がある。

セキュリティ面では、VPN と SSH はともに強力な暗号化技術に対応しており、適切に構成すれば通信内容の機密性は十分に確保できる。L2VPN が閉域網サービスとして提供される場合、インターネットを経由せず高度に隔離された通信経路となるため盗聴リスクが低く、セキュリティ上の優位性がある。一方、SSH リバーストンネルはファイアウォールや NAT 越えの手段として有用だが、サーバーからクライアント側への接続を許可する設計となるため、誤設定時のリスクも存在する。例えば、リバーストンネルを確立した先のサーバーが侵害された場合、そこを経由してクライアント内部への侵入経路となる可能性がある。このリスクに対しては、公開鍵認証の厳格な運用やポート転送の制限によるアクセス制御が必要となる。

構成の容易さと柔軟性においては、SSH リバーストンネルに優位性がある。SSH リバーストンネルは一般的な SSH 接続とポート転送の知識があれば、特別なハードウェアやソフトウェアを追加導入せずに短時間で構築できる。実際、前述の Raspberry Pi を用いた OOBM 装置の事例でも、`ssh -R` コマンドを使用した簡素な構成でリモートアクセスを実現している。対照的に、VPN 環境の構築には VPN サーバーの用意、クライアント設定の配布、場合によっては認証局の準備などが必要となり、初期設定の難易度が比較的高い。特に L2VPN で拠点間ブリッジを構成する場合、両端の機器でタップデバイス設定やブリッジ設定を正確に行う必要があり、高度なネットワーク知識が求められる。このように、導入の容易さという観点では SSH トンネル方式が優れていると言える。ただし、VPN には一度接続が確立すれば複数機器への透過的なアクセスが可能になるという利点もある。

最後に、モバイル回線利用時の負担とコスト面での比較も重要である。モバイル通信は一般に従量課金や月間データ量制限があるため、常時接続型の VPN では不要な通信コストが発生しやすい。VPN を常時有効にしておくと、トンネル維持のための制御パケットが継続的に送受信され、データ使用量が少しずつ増加する。対照的に、SSH リバーストンネルは必要な管理ポートのみを開放し、通常使用しないトラフィックはモバイル回線に載せない運用が可能のため、データ通信量の節約につながる。さらに、リバーストンネルはクライアント側からサーバーへの接続を開始するモデルであるため、モバイル側では NAT 越え設定が不要で、使用しない時間帯は切断しておけばコストゼロとなる利点がある。

これらの比較から、研究室規模のネットワークにおける管理用バックアップ回線として、SSH リバーストンネルは低コストかつ効率的なソリューションとなり得ることが示唆される。特にモバイ

ル回線の細い帯域と従量制という制約下では、必要最小限の通信だけに絞れる SSH トンネル方式がコスト効率と運用効率の両面で有利と言える。

4 提案手法

本研究では、モバイル回線を利用した SSH リバーストンネルによるバックアップ経路を自動的に制御する構成を提案する。提案手法では、拠点側のホスト（以降、`hostA`）がグローバル IP を持つ中継用コントローラマシン（`controller.example.com`, IP アドレス: 203.0.113.1）に対して SSH リバーストンネルを確立する。

4.1 リバーストンネリングの技術的背景

SSH リバーストンネリングは、NAT やファイアウォールの内側に位置するホストに対して、外部からのアクセスを可能にする技術である。通常の SSH 接続では、SSH クライアントがサーバに接続するが、リバーストンネリングでは、アクセスされる側のホストが SSH クライアントとして動作し、グローバル IP を持つサーバにトンネルを確立する。

この技術は特に以下のような環境で有用である：

- 動的 IP アドレスが割り当てられる環境
- NAT 環境下の機器へのリモートアクセス
- ファイアウォールによって外部からの接続が制限されている環境
- 複数拠点のリモート管理を統一的なインターフェースで行いたい場合

本研究では、この SSH リバーストンネリングにフェイルオーバー機能を組み込むことで、主回線障害時にもシームレスなアクセスを実現する構成を目指している。

4.2 システム全体構成

提案システムの全体構成を図 1 に示す。本システムは以下の要素から構成される：

- **拠点側ホスト (HostA)**：有線 (eth0) とモバイル回線 (eth1) の二重インターフェースを持つ
- **コントローラマシン (Controller)**：グローバル IP アドレスを持ち、トンネル終端として機能
- **管理者 (Operator)**：コントローラマシンを介して拠点側ホストにアクセス

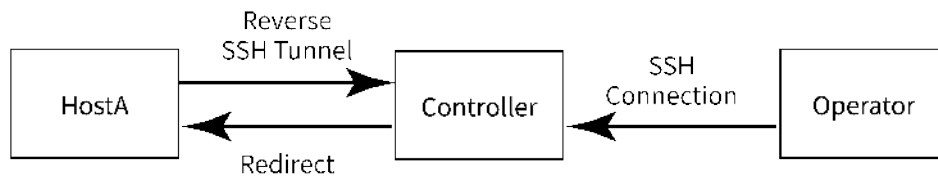


図1 提案システムの全体構成

4.3 ルーティングによる経路制御

平常時は、hostAの有線インターフェース（eth0）を用いてリバーストンネルを確立するが、主回線が断絶した場合には、eth1（モバイル回線）を用いて再接続を行う。このフェイルオーバーは、ルーティングテーブルの優先度（metric）設定によって実現される。

具体的には、netplan による以下のような設定を適用することで、コントローラマシン宛（203.0.113.1）のトラフィックに対して経路制御を行う：

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      dhcp4-overrides:
        use-routes: true
      routes:
        - to: 0.0.0.0/0
          via: 192.168.10.1
          metric: 100
        - to: 203.0.113.1/32
          via: 192.168.10.1
          metric: 90 # 優先度を高く設定
    eth1:
      dhcp4-overrides:
        use-routes: false
        use-dns: false
      routes:
        - to: 203.0.113.1/32
          via: 192.168.2.1 # eth1 のゲートウェイ
```

```
metric: 100 # eth0 より優先度を低く
```

この設定では、以下のような経路制御が行われる：

- eth0 経由のコントローラ宛ルートには metric 値 90（高優先度）を設定
- eth1 経由のコントローラ宛ルートには metric 値 100（低優先度）を設定
- 通常のインターネットトラフィック (0.0.0.0/0) は eth0 経由でのみルーティング
- eth1 はコントローラマシンへのアクセスのみに限定（帯域節約のため）

これにより、主回線（eth0）が利用可能な場合は優先的に使用され、断絶時には自動的にモバイル回線（eth1）へと切り替わる仕組みとなっている。

4.4 障害検知システムの実装

ネットワークインターフェースの状態監視には、systemd サービスとして実装されたシェルスクリプトを使用している。このスクリプトは、前述のコントローラ以外の外部ホスト（198.51.100.1）への ping 監視を行い、ネットワーク状態の変化を検出する。

監視スクリプトの主要部分を以下に示す：

```
#!/bin/bash

LOG_FILE="/var/log/network-monitor.log"
INTERFACE="eth0"
CHECK_HOST="198.51.100.1" # 外部接続性確認用
CHECK_INTERVAL=1 # 1 秒ごとにチェック

PREV_STATE="unknown"

while true; do
    if ! ip link show $INTERFACE | grep -q "state UP"; then
        CURRENT_STATE="down"
    else
        if ping -c 1 -W 2 $CHECK_HOST >/dev/null 2>&1; then
            CURRENT_STATE="up"
        else
            CURRENT_STATE="no_connectivity"
        fi
    fi
fi
```

```

if [ "$PREV_STATE" != "up" ] && [ "$CURRENT_STATE" = "up" ]; then
    log "$INTERFACE recovered from down/no_connectivity state to up"
    log "Will restart SSH tunnel service in 60 seconds"
    (
        systemctl stop ssh-tunnel.service
        sleep 60
        log "Executing delayed restart of SSH tunnel service"
        systemctl restart ssh-tunnel.service
    ) &
elif [ "$PREV_STATE" = "up" ] && [ "$CURRENT_STATE" != "up" ]; then
    log "$INTERFACE changed from up to $CURRENT_STATE"
    log "Will restart SSH tunnel service in 60 seconds"
    (
        sleep 60
        log "Executing delayed restart of SSH tunnel service"
        systemctl restart ssh-tunnel.service
    ) &
fi

PREV_STATE=$CURRENT_STATE
sleep $CHECK_INTERVAL
done

```

このスクリプトでは、ネットワークインターフェースの物理的な状態（ip link コマンドによる確認）だけでなく、実際の外部接続性（ping による確認）も検証している点が特徴である。これにより、インターフェースは物理的に UP 状態だが実際の通信ができないような「半断線」状態も検出可能である。

障害または復旧を検出した場合、スクリプトは 60 秒の遅延後に SSH トンネルサービスを再起動する。この遅延は、次節で説明するポート衝突問題への対策として機能している。

4.5 SSH リバーストンネルサービス

SSH リバーストンネルは systemd サービスとして実装されている。サービス定義ファイルを以下に示す：

```

# /etc/systemd/system/ssh-tunnel.service
[Unit]
Description=SSH Reverse Tunnel

```



```
After=network.target
```

```
[Service]
```

```
User=netadmin
```

```
Group=netadmin
```

```
Environment="HOME=/home/netadmin"
```

```
ExecStart=/usr/bin/ssh -N -R 2222:localhost:22 \  
relayuser@203.0.113.1 -i /home/netadmin/.ssh/id_rsa
```

```
Restart=always
```

```
RestartSec=30
```

```
[Install]
```

```
WantedBy=multi-user.target
```

このサービス設定では、以下の特徴がある：

- -N オプションで対話的シェルを起動しない（純粋なトンネル接続）
- -R 2222:localhost:22 により、コントローラの 2222 ポートから hostA の 22 ポートへのトンネルを確立
- サービス障害時は 30 秒後に自動再起動（Restart=always, RestartSec=30）

4.6 接続タイミング制御と同時接続問題の解決

ネットワーク切断時の挙動として、hostA の接続が突然切れた場合でも、コントローラ側には一定時間プロセスが残存する。この状態で別インターフェースからリバーストンネルを確立しようとすると、ポート番号の衝突が発生する問題がある。

この問題に対処するため、リバーストンネル用ポートの解放タイムアウトを 45 秒に設定し、ネットワーク疎通性喪失からリバーストンネル再確立までの待機時間を 60 秒と定めた。これらの時間設定により、新旧の SSH セッションの衝突を避け、古いセッションが確実にタイムアウトした後、にのみ新しい接続を確立するよう制御している。

この方式により、新旧のプロセスによるポート競合を回避し、安定した自動切替を実現している。ただし、この手法はタイムアウト値に依存する暫定的なものであり、今後の改良が必要である。

4.7 コントローラ側の転送制御とユーザ構成

リバーストンネリングの実現には、コントローラ側で複数のユーザアカウントを適切に構成する必要がある。まず、relayuser と呼ばれる特別なユーザをコントローラに作成する。このユーザは、各拠点のホスト（hostA など）がコントローラに対してリバーストンネルを確立する際に使用

される。hostA は自身の systemd サービスとして、このユーザアカウントを利用してコントローラへの接続を維持する。また、各拠点のホストには netadmin と呼ばれるユーザを作成する。このユーザは、コントローラ等からログインして操作する際に利用される。

さらに、コントローラには各管理対象ホストに対応した個別のユーザ（例：hosta, hostb など）を作成する。管理者がこれらのユーザとしてコントローラにログインすると、対応する拠点ホストの netadmin ユーザへ自動的にリダイレクトされる仕組みとなっている。この方法により、管理者は hosta@controller.example.com のように単一のアドレスと適切なユーザ名の組み合わせだけで、目的のホストに透過的にアクセスできる。

この転送制御の実装には、各ユーザの authorized_keys ファイル内の command=オプションを活用し、以下のような設定を行っている：

```
command="ssh -p 2222 -o StrictHostKeyChecking=no -i HostA に接続する際の公開鍵 \
netadmin@localhost",no-agent-forwarding 管理者の公開鍵
```

この設定により、管理者がコントローラに接続すると自動的に対象ホストへリダイレクトされる仕組みが実現される。また、StrictHostKeyChecking=no オプションによって、ネットワーク切替時に発生する可能性のある鍵変更警告が抑制され、切替プロセスの透過性が向上している。さらに、no-agent-forwarding 設定を適用することで、認証情報の不正転送リスクを低減し、セキュリティ面での堅牢性を確保している。

通常であれば、複数拠点の管理では各ホストに直接接続するための様々な IP アドレスや接続情報を管理する必要があるが、本構成では単一のコントローラへの接続とユーザ名の指定のみで、目的のホストへアクセスが可能となっている。

5 システム評価

本研究で構築したシステムの有効性を検証するため、複数のシナリオに基づいた動作確認試験を実施した。評価項目としては、主に（1）通常時の接続性、（2）障害検知機能、（3）接続切替の自動化、（4）回線復旧時の挙動、の4点に着目した。

5.1 基本機能の動作検証

まず基本的な接続性について、管理者がコントローラサーバを経由して各ホストに接続できるかを検証した。‘ssh hosta@controller.example.com -i 公開鍵’の形式でコマンドを実行することで、controller を中継点として hosta に対する SSH 接続が正常に確立されることを確認した。この接続は、内部的には SSH リバーストンネルを介しているが、管理者からはあたかも直接接続しているかのように操作可能であり、操作性の面での実用性が確認された。

5.2 障害検知と接続切替の検証

次に、主回線（eth0）障害発生時の自動検知と切替機能を検証した。実験では、意図的に主回線を切断し、システムの挙動を観察した。その結果、ping による監視機能が正常に働き、回線切断を即座に検出できることが確認された。特に、監視プロセスはネットワーク状態の変化を迅速に捉え、切断イベントが正確に記録された。

さらに、接続切替のタイミングについては、次のような挙動が確認された。主回線切断後、既存の SSH 接続は 45 秒のタイムアウト時間経過後に自動的に終了し、使用していたポートが適切に解放された。このタイムアウト値は、実運用環境での一時的なネットワーク不安定に対する耐性と、障害時の迅速な切替のバランスを考慮して設定されたものである。

5.3 自動復旧プロセスの検証

主回線切断から 60 秒後には、systemd によるサービスの自動再起動が実行され、バックアップインターフェース（eth1）を介した SSH リバーストンネルの再確立が行われた。この時間差設定により、ポートの衝突を避けつつ、合理的な時間内での接続復旧が実現された。本検証では、バックアップ回線を経由した管理通信が正常に機能し、主回線障害時においても遠隔からの管理操作が可能であることが確認された。

5.4 回線復旧時の挙動検証

最後に、主回線復旧時の動作を検証した。主回線（eth0）が復活した際、システムは自動的にこれを検出し、バックアップインターフェース（eth1）上で確立されていたリバーストンネルを適切に閉じ、主回線を介した新たなトンネルの再確立に成功した。この切り戻し処理においても、接続の連続性が保たれ、管理者側での操作変更を必要としない透過的な経路切替が実現された。

一連の評価試験を通じて、本システムが想定通りの耐障害性と自動復旧能力を持つことが確認された。特に、主回線とバックアップ回線の切替が自動化されており、管理者が接続方法を変更することなく一貫した操作性を維持できる点は、実用的な OOBM ソリューションとして有効であると判断される。

5.5 提案手法の利点

本構成の主要な利点として、まず標準技術の活用が挙げられる。SSH および systemd の標準的な機能のみを使用しており、特別な VPN 機器や商用製品を必要としないため、既存の知識やツールを活用した実装が可能である。これに関連して、安価なハードウェア（Raspberry Pi + モバイル回線）で実現可能であるという低コスト性も重要な特徴である。研究室環境や小規模組織でも、予算的な制約なく導入できる点は実用上の大きなメリットと言える。

システム拡張の容易さも本手法の強みである。新たな拠点の追加は、基本的にコントローラにユーザを追加するのみで完了し、複雑なネットワーク設定変更を必要としない。この特性により、運用中のシステムに対しても柔軟に拠点を追加できる。さらに、統一的なアクセス方法を提供する点も利点である。主回線の状態にかかわらず、管理者は常に同一の接続方法（ユーザ名@controller.example.org）でターゲットホストにアクセスでき、バックグラウンドでの経路切替を意識する必要がない。

セキュリティ面では、標準的な SSH 鍵認証を使用し、不要な接続の制限が可能であるため、堅牢な認証基盤を再実装することなく、既存のセキュリティプラクティスを活用できる。最後に、ユーザエクスペリエンスの観点からは、透過的な接続の実現が挙げられる。ユーザから見ると、複雑なトンネリングや経路制御の仕組みを意識することなく、直接ホストに接続しているように操作できる点が、管理作業の効率化に寄与している。

特に、分散拠点の管理においては、各拠点のネットワーク構成や接続状態を意識することなく、統一的なインターフェースを通じてアクセスできる点が大きな利点である。

5.6 今後の課題

現在の実装には、いくつかの技術的課題が残されている。まず、ネットワーク切替時に発生するポート衝突問題については、現状ではタイムアウト値による暫定的な回避策を採用しているが、より堅牢な解決策が必要である。例えば、動的ポート割当やコントローラ側でのセッション管理機能の実装などが考えられる。また、複数の管理者が同時に一つの拠点にアクセスする場合の制御方法についても検討の余地がある。加えて、提案システムでは新規拠点の追加が容易であると述べたが、実際に多数の拠点やユーザを追加した際のスケーラビリティについては十分な検証が行われていない。特に、コントローラ上での多数のリバーストンネルの同時管理や、多様なネットワーク環境に設置された拠点の挙動の差異については、さらなる実証的評価が必要である。

性能面における課題としては、モバイル回線使用時のレイテンシや帯域幅の実測評価が十分に行われていない点が挙げられる。特に、高負荷時や回線状況が不安定な環境での挙動については、さらなる検証が必要である。また、システムの中核であるコントローラは現状では単一障害点となっており、その冗長化構成の検討も重要な課題である。コントローラ障害時にも接続性を維持するための、マルチコントローラ方式や自動フェイルオーバー機能の実装が望まれる。

運用管理面においては、切替イベントの自動通知システムの実装や、各拠点の接続状態をダッシュボード等で可視化する機能の追加が望まれる。現状では、ネットワーク切替が発生した際の通知機能が十分でなく、障害対応の迅速化のためには、管理者への自動アラート機能などの実装が有効であろう。

さらに長期運用における課題として、安定性評価や様々なネットワーク障害パターンに対する挙動検証がある。実際の運用環境では、想定外のネットワーク状況変化や障害が発生する可能性があり、それらに対する堅牢性の検証が重要である。

6 謝辞

本研究は、ソフトイーサ株式会社様との共同研究である。

参考文献

- [1] ヤマハ株式会社. Wan 回線障害時に備える（モバイルバックアップ）：コマンド設定. 最終アクセス日: 2025 年 3 月 31 日.
- [2] Opendgear. Acm7000 リモートサイト・ゲートウェイ. 最終アクセス日: 2025 年 3 月 31 日.
- [3] padoo. Raspberry pi zero w でワイヤレスシリアルコンソールサーバー. 最終アクセス日: 2025 年 3 月 31 日.
- [4] 株式会社インターネットイニシアティブ. Pix-mt100/110 + raspberry pi. 最終アクセス日: 2025 年 3 月 31 日.
- [5] ファーエンドテクノロジー株式会社. モバイルインターネットをバックアップ回線にしてみました！最終アクセス日: 2025 年 3 月 31 日.
- [6] 啓二松島. Raspberry pi を用いた lte 通信対応データロガーの製作. Technical report, 九州大学応用力学研究所, 7 2021. 最終アクセス日: 2025 年 3 月 31 日.
- [7] 株式会社モバイル・プランニング. 「使わなかった時ゼロエン wifi」いざというときに役立つ、バックアップ回線用「+f fs040u」取扱い開始！最終アクセス日: 2025 年 3 月 31 日.

つくば WIDE Project 公開ミラーサービス

太田弘樹 s2220573@s.tsukuba.ac.jp

2025-03-31

つくば WIDE Project 公開ミラーサービスについて

つくば WIDE Project 公開ミラーサービス (以下ミラーサービス) は現在産学間連携推進室に場所をお借りし運用している WIDE Project つくば NOC によって広く一般に提供しているサービスの一つである。このミラーサービスでは主に日本国内向けに Linux ディストリビューション (CentOS, Ubuntu, Raspberry Pi OS など) やソフトウェア (GCC, Vim など) の配信を行っている。1 日あたり、おおむね 140 万件のリクエストがあり、WIDE Project つくば NOC において、このミラーサービスは大きなトラフィックの発生を伴うプロジェクトであり、様々なオープンソースプロジェクトの支援を行うとともに、つくば NOC に定常的に通信が発生する環境を提供している。

機材構成

- MB/Chassis: Dell PowerEdge R530
- CPU: Intel® Xeon® CPU E5-2630 v3 @ 2.40GHz
 - Thread(s) per core: 2
 - Core(s) per socket: 8
 - Socket(s): 2
- RAM: 384GB
- Storage
 - RAID: Broadcom / LSI MegaRAID SAS-3 3108
 - DISK: 6TB HDD x8 RAID6 約 33TB および 2TB HDD * 15 RAID6 約 24TB
 - * 内約 33TB 程度使用中

リクエストの変化

本ミラーサービスでは、これまで CentOS 7 のリポジトリミラーを提供していたが、この CentOS 7 は 2024 年 6 月 30 日にメンテナンス期限を迎え、公式サポートが終了した。

これに伴い、公式ミラーから CentOS 7 関連のリポジトリが抹消され、我々を含む各ミラーからも、同リポジトリの提供が終了した。

しかし、これを利用していたコンピュータからは、依然としてソフトウェアの更新を確認するリクエストがひっきりなしに飛んできている。

計測すると一日あたり、約 18000 ホスト (IP アドレス) からのリクエストが行われているが、これすなわち現在においてもインターネット上にサポートのない OS を利用しているホストが約 18000 存在しているということになる。

これらの情報は、更新取得失敗時に他のサーバーを試すような挙動になっているものであれば、他のミラーサービスにおいても同様の情報を取得可能である。しかし、各ミラーサービスにおいてこれらの情報を悪用しないことは保証されておらず、各ミラーサービス運営者の良心に任せられているということになる。

ライフサイクルの終焉というものはなかなか想像しにくいものではあるが、こういったミラーの仕組みなどをうまく活用して、脆弱なシステムをなるべく生み出さないような仕組みが必要かもしれないと感じた。

ミラーサービスの技術

ミラーサービスの同期は多くの場合一般的なファイル同期用のアプリケーションで行われており、同期の都度、すべてのファイルの更新状態を確認し、変更のあったファイルを同期している。しかし、ミラーサービスの同期の特徴として、大半のファイルは変更がなく、ごく一部のファイルのみ変更追加されるという点がある。そこで、今後はミラーサービス同期に特化したファイル同期用のアプリケーションの拡張などを研究していきたいと考えている。

今後について

近年、データ配布の方法として CDN などの利用も増え、また、非営利プロジェクトへの理解のある CDN プロバイダなどもあり、これらのミラーサービスの存在を感じることも少なくなっているようである。しかし、依然として本サービスのようなミラーを使ったソフトウェアの配布方法も多く、利用の減少は見られない。現在約 33TB のデータをホストしている本ミラーサーバであるが、これからもサービスの安定提供に努めていくだけでなく、ソフトウェアの配布技術に関連する研究なども行っていきたい。

居住空間での業務用電子計算機の運用に関する研究

太田弘樹 s2220573@s.tsukuba.ac.jp

2025-03-31

本研究の目的

現在、産学間連携推進室では居住空間と隔離されない形で計算機区画が設置されており、当該区画には各員の研究に必要な無数の計算資源や通信装置が設置されている。しかし、この機材が居住空間に影響を及ぼし、室員の研究活動の妨げとなってしまうていた。この研究ではそういった問題を解決しようとするとともに、計算機室を用意できないが業務用電子計算機を利用したいという組織において発生しうる問題の列挙や対処方法の検討を行う。

業務用電子計算機の特徴

本稿で述べる業務用電子計算機は、一般の人が日常や業務において卓上で利用するような電子計算機ではなく、専用の架台に設置され、平時は近傍に人がいない場所に設置されるような計算機を指す。

これらの装置は性能や空間効率、冷却能力などを念頭において設計されており、一般向けの装置であれば考慮すべき騒音や発熱はほぼ考慮されておらず、隔離された専用の部屋で空調などを管理された状態で使用されること、つまり利用環境側で対応することが想定されている。

これまで

昨年までのところで、産学官連携推進室ではホームセンターで一般に入手可能なものを用いて、機器騒音の改善を行った。

これに伴い、騒音のみならず空調についてもある程度の改善が見られたが、依然として機器排気の吸い戻しが発生しており、機器周辺のみ気温の高い状況が継続していた。

改善手法

効率を重視する近年のデータセンターでは、様々な手法によってそれを達成している。空調に関して、従来は機器の設置された空間はオープンスペースで、部屋全体を冷却する方法が取られていた。しかし、この方法で

は空調機を出た冷気は一部、機器へその冷熱を伝えることなく、空調機へ吸い戻されつこととなる。私はヒートポンプの専門家ではないので、詳細な説明は省くが、大まかに説明すると外気温と空調機の吸気温度の差が小さければ小さいほど、効率が良いということになる。

つまり、外気温が 25°C だったとして、大量に吸い戻しをしているサーバーの吸気温度を十分に低く、例えば 20°C に保つために、室温を 10°C 程度まで多大に下げなければならないとする。空調機としては、温度差が 15°C 程度になり、それなりに寒い部屋を更に冷やさなければならない、空調機から見える室温と外気温の差が大きくなってしまい効率が悪い。

しかし排気吸気が完全に分離されている場合、機器排気が 25°C 程度あるとして、外気温 25°C、機器の吸い戻しはなく空調機器にすべて吸気される場合、空調機の吸い込み温度も 25°C となり、外気との温度差はほぼなく、効率も良くなる。

そのため近年のデータセンターでは、主に機器の排気側と吸気側を明確に分け、吸気側には空調機からの冷却された空気を、空調機の吸気には機器排気を吸い込ませることで効率を改善している。

そこで、産学間連携推進室においても、機器の排気と吸気をできる限り分離し、空調効率の改善を行った。

限界

産学間連携推進室のサーバールームには、残念ながら空調機が設置されておらず、居住空間の空調機を利用してサーバールームの冷却も行っている状況であり、機器の排気を完全に空調機の吸気へ導くことはできない。

そこで、居住空間も含めて、熱の回り込みができる限り少なくなる設計とする必要がある。

実際の構成

これまでのサーバールームは以下のような構成になっており、図中 Srv が排気を吸い戻してしまい、高温になっていた。ひどいときは吸気側温度で 40°C を超えることもあった。空調機は図中 Living Space 側にしか存在せず、ドアを開け、設定温度を 18°C に設定し過度に冷却することで、どうにか機器を冷却しようとしていたが、機器は吸い戻しであまり冷えないうえ、部屋は寒すぎるし、機器の騒音が居住空間側に多大に漏れてきており、人間の作業空間としてあまりに不快な状況となっていた。

```
+-----+
|               |
|  +-----+    |
|  |Srv |      |
+---+---+-(DOOR)-----+
|               |
|               |
|  Living Space  |
|               |
```

```

|           |
+-----+

```

そこで昨年のところで、衝立や換気ルートを設置を行い

```

+-----+
|       |
| +----+
| |Srv |
+ (VENT) -+----- (VENT) +
|       |
|       |
| Living Space
|       |
|       |
+-----+

```

上図のような状況になっていたが、今年は更にラック内での吸い戻しを防ぐために、養生テープによる目張りを行い、その効果を測定した。



図1

今回はダイヤテックス社製パイオランテープ Y-09-GR を使用したが、気密のある程度あるテープであれば何を使用しても問題ないであろう。

結果

本対策前は、昨年の改善があってもなお 30℃ 程度あった吸気温度が、約 23℃(空調機の設定温度、ほぼ室温)に改善し、それに伴い冷却に関する騒音も減少したように感じる。排気側気温は 28℃ 程度あるが、これはある程度パワーのある送風機で居住空間側に十分に混和させており、居住空間側の影響はほぼ感じられない。

まとめと今後

今回は産学間連携推進室の居住性向上のために計算機区画の空調効率改善を安価にできる方法について実験を行った。

結果、機器の冷却効率を安価に向上することができ、またそれに伴って機器の騒音を低減する事もできた。

しかし依然として、昨年の改善での問題点同様、可燃物を多大に利用しており、商用環境において利用が難しい点はまだ課題が残る。ただし、可燃物使用に制約のない環境や、個人宅での業務用計算機の習熟などの使用用途であれば、今回のような手法は適用可能であり、有効であると考ええる。

これまでの実験等によって、産学間連携推進室の三大居住環境問題であった騒音・寒さ・乾燥のうち、騒音を低減させ、寒さはほぼ問題ない状態になった。

しかし、常にサーバーが空気を温め、空調機がこれを冷却し続けるという構造は、コンプレッサー方式の除湿機そのものであり、部屋は常に非常に乾燥している。湿度計が常に計測下限を示している状況は、夏場は良いが、冬場は喉が張り付いてしまう感覚を覚えるようで、居室としてはいささか辛い。

今後は引き続き騒音対策を続けるとともに、乾燥についても対策を行っていきたいと考えている。

WIDE Project つくば NOC 運用報告

太田弘樹 s2220573@s.tsukuba.ac.jp

菊田一真 s2420597@u.tsukuba.ac.jp

服部真吾 s2111559@u.tsukuba.ac.jp

間瀬太陽 s2410334@u.tsukuba.ac.jp

中野あおい s2312794@u.tsukuba.ac.jp

2025-04-03

つくば NOC とは

つくば NOC は筑波大学術情報メディアセンター支援の元、情報科学類産学間連携推進室内サーバー室にて運用している NOC (Network Operation Center、ネットワークを構成する機材等を収容する施設) である。

本 NOC では通常の大学ネットワークから独立した別系統のネットワークを独自に運用しており、大学ネットワークでの運用や実施が難しいプロジェクトや実験 (不特定多数との間で大量のトラフィックを発生させるもの、大学外の一般に向けたサービスの運営、ハニーポットを用いたものなど) を実施している。また、この過程においてネットワーク運用に関する知見や技術の習得を目的としている。特につくば WIDE Project はこのつくば NOC を使用した主要な研究活動である。

つくば WIDE に関して

このつくば WIDE は 2007 年 11 月から産学間連携推進室に場所をお借りし、運用している WIDE Project の拠点の 1 つである。WIDE Project は複数の大学など教育機関や企業の研究所などにより構成されており、大規模ネットワークの運用・研究のほか、安定した高品質なネットワークを一般の WIDE BB 利用者や研究者等に提供することを目的とした活動を行っている。つくば WIDE では、この WIDE Project の一部として、つくば地域において WIDE バックボーン (東京大手町) と接続された NOC を運用している。2014 年には CTF シリーズを開催することで新たに情報セキュリティ分野での活動が行われた。つくば WIDE では現在、以下のようなプロジェクトを行っている。

パブリックミラーサービスの提供

主に日本国内向けに Linux OS (CentOS, Debian, fedora など) や各種ソフトウェア (Apache, GCC, Vim など) の配信を行うミラーサーバーである。

30TB 超のストレージレイを構成し、イメージなどの提供を行っている。おおむね 1 日あたり 140 万件の

リクエストがある。

つくば WIDE において、このプロジェクトは最も大きなトラフィックの発生を伴うプロジェクトである。

学内向け中規模ネットワークの運用

産学間連携推進室のほか、学術情報メディアセンター、情報科学類 WORD 編集部、新城研究室にまたがる学内中規模ネットワークを運用している。このネットワークは学内ネットワークとは独立した実験環境が必要な場合や NOC ではない各組織の部屋内でサーバーを運用したい場合などに使用している。また、このネットワークの運用を通してネットワーク運用に関する技術や新しい知見の習得を目指している。

構成

対外線接続

つくば NOC と WIDE バックボーンとの間はレイヤ 2 で接続している。接続は筑波大学学術情報メディアセンター内のアクセススイッチと WIDE プロジェクトバックボーンの間では SINET の L2 接続サービスを、学術情報メディアセンター内と産学間連携推進室内を結ぶ通信は学内ネットワークの光ファイバーをお借りして接続している。

なお、学内ネットワークの借用と SINET の L2 接続サービスの利用は学術情報メディアセンターの多大な支援により実現しているものである。

経路

アドレスレンジ

IP アドレスは以下の範囲のアドレスを使用している。

- IPv4 アドレス: 203.178.132.64/26
- IPv6 アドレス: 2001:200:0:7C00::/56, 2001:200:1C8::/48

経路交換

上記の対外線に関する記載の通り、筑波大学及び SINET とは直接的にルーティングを行える箇所は存在せず、経路交換は行っていない。経路交換は WIDE バックボーンとの間でのみ行っている。ルーティングプロトコルは IPv4 接続においては OSPF を、IPv6 接続においては OSPFv3 を用いている。

新規メンバー

2024 年度のところで、つくば WIDE NOC メンバーに間瀬さんと中野さんが新しく加入した。

近年、Web 系などのフロントエンジニアばかりが着目されているように感じるが、インターネットを支えるバックエンド・インフラに興味を持ってくれたことは喜ばしいことである。

機材管理や、その他ノウハウの引き継ぎを継続して行うとともに、今後も若者へのリーチ活動にも力を注いでいかなければならない。

つくば NOC の今後

WIDE Project や SINET との直接接続

これはまだ検討段階ではあるが、つくば WIDE として AS を運用したりし、筑波大学を経由した WIDE Project との接続ではなく、一組織として独自に接続する検討をしている。

現在の接続が筑波大学経由である以上、筑波大学自体の運営や各研究機関どうしをつなぐ通信が重要であり、我々の行う実験等でこれらを圧迫してしまうことはこちらとしても避けたいと考えている。

そこで筑波大学とは物理的にも完全に独立したネットワークで WIDE Project と接続することでこういった懸念を払拭することができると考えている。また、筑波大学全体の帯域幅に影響されることなく通信を行うことができるため、より高速なネットワーク (100Gbps クラス) の運用に関する知見や技術の習得が行えると考えている。

筑波大学学内との直接通信

筑波大学学内での研究活動でもつくば NOC のパブリックミラーサービスを利用しているという声を時々聞く。しかし、現状、つくば NOC で行っているその他のサービスも含めて、通信先は学内にあるにも関わらず、筑波大学学内からの通信はつくばから出て、大手町のデータセンターを一度経由し、つくばへ戻ってきている。これは双方の他所行きの通信を圧迫するだけではなく、遅延の増加も招き非常に非効率である。そこで筑波大学と経路交換を行い、学内で完結可能な通信は学内で完結させたいと考えている。これにより、当パブリックミラーで提供しているソフトウェアを利用している研究ではより高速に環境構築やソフトウェア更新などができるようになり、通信速度に研究活動が律速される機会が減少し、研究支援につながると考えている。

中学生向け産学官連携 ICT 教育イベント開催の総括

～ サーバ構築実習イベント「インデペンデンス・サーバー・デイ」の継続的な開催 ～

2025 年 4 月 13 日 筑波大学情報科学類 産学間連携推進室 活動成果報告会

ICT 教育イベントプロジェクト代表 根本 晃輔、 副代表 服部 真吾

協力：産学間連携推進室 関口亜聖、太田弘樹、北野尚樹、細島涼雅、間瀬太陽

概要

産学間連携推進室においては、2016 年度より茨城県つくば市総合教育研究所¹、および筑波大学発ベンチャー企業数社²等と協力し、茨城県つくば市の中学生を対象とした実践的な産学官連携 ICT 教育イベント「インデペンデンス・サーバー・デイ」を運営している。本イベントは、各受講生に固有の物理サーバ（Raspberry Pi に代表されるシングルボードコンピュータ 1 台）とグローバル固定 IPv4 アドレス 1 つを配布し、受講生にネットワーク・サーバの動作原理、構築や運用の方法、運用上重要なセキュリティ対策等を学んでもらうものである。受講生は本イベントの開催期間中だけでなく、イベントの後も割り当てられたサーバを継続して自由に運用することができる。本イベントの目的は、高度情報化社会に顕著な「クラウド化」の傾向の中で、特に情報技術・IT インフラ技術に強い興味を持つ生徒らに対し、自主学習の難しい「物理的なサーバの動作や運用」の学習機会を提供し、更に高度な情報技術の学習に向けての橋渡しを狙うことにある。本報告においては、2024 年に開催した「インデペンデンス・サーバー・デイ 9」の開催報告、2016 年以降の継続的な開催の中での形式の変遷や考察、また 2025 年に開催予定の「インデペンデンス・サーバー・デイ 10」に向けての取り組みについて述べる。

1. 序論

2020 年度の小学校教育における「プログラミング教育」の必修化を契機に、近年、若年層へのプログラミング教育に対する注目・関心が急速に高まりつつある。

¹ つくば市教育局内の組織で、教育における情報技術の導入・運用を専門的に扱う組織。物理的な施設はつくば市大形 1333-1（旧大形小学校）に所在する。

² 2025 年 4 月時点での協力組織は、ソフトイーサ株式会社（代表取締役 登 大遊 氏・筑波大学情報科学類 2003 年度 AC 入学）、輝日株式会社（代表取締役 佐藤 大哲 氏・筑波大学情報科学類 2016 年度入学）、太平電算（代表 根本 晃輔・筑波大学情報科学類 2013 年度入学）である。本イベントの開催に当たり、各組織にはネットワーク資源・各種機材・運営学生への謝金等の協力を頂いている。

特に 2010 年代後半以降においては、官・民の別を問わず、多様な難易度の若年層向けプログラミング・情報技術教育イベントが開催されている。「セキュリティ・キャンプ」の知名度が向上し、応募者が年々増加傾向にあること、小中高生を対象とする「未踏ジュニア」が 2016 年にスタートしたことなどは特に記憶に新しい。

1－1．高度な技術の学習機会の必要性

さて、官民の主体を問わず開催されている「プログラミング」に関連するイベントについて、簡易なビジュアルプログラミング環境である Scratch³や、ビジュアルプログラミングを介して物理的なパーツを動作させオリジナルのロボットなどを設計することのできる LEGO MINDSTORMS⁴、小中学生に高い人気を誇るサンドボックスゲーム環境を基盤として簡易なプログラミングに習熟できる Minecraft などの環境を用いて簡易的なプログラミングを学ぶようなものについては、さまざまな企業・団体により、日本各地で同様のものが開催されている。公立小中学校における「プログラミング教育」の運用、または部活動における「プログラミング」の取り組みの拡大もあり、小中学生の児童・生徒らにとって、簡易なプログラミングを学習するための土壌が成立しつつある。

しかし、これらの「世間一般における取り組み」のほとんどは、あくまで特定のプラットフォームの範囲内で Scratch / Python などの環境に閉じた・ごく簡易なプログラミングのみを取り扱うものであったり、運営主体が異なっても特定の教材・教具を用いたワンパターンなものばかりであったり、果ては「プログラミング未経験者可」として講師の求人がなされているものがあつたりと、イベント・講座内で技術の学習が閉じており、情報技術全体の学習への発展性のないものと言えるだろう。それらより高度な内容に当たる「テキストベースのプログラミング」や「サーバ・ネットワークの構築実習」「オリジナルのサービスの設計や構築・保守」を取り扱う、若年層を対象とし、なおかつ気軽に参加できるようなイベントは、2025 年 4 月現在においても、従前同様、ほとんど存在しない⁵ 取り組みであると言える。

³ 一般にはキャラクター（画像）の動きをプログラミングするものであるが、発展させてゲームを制作したり、動画作品のようなものを制作したりすることが可能である。

⁴ 2000 年代初頭から同名のシリーズが展開されているが、現在販売されているものは当時より高性能・高機能なシステムとなっている。また、一部の教育機関が実際に教育の場に取り入れているなど、活用事例も豊富になってきている。

⁵ 既存の「社会人向けプログラミングスクール」が小中学生向けに門戸を開くケースもあるが、24 回の授業で 70-80 万円近い高額な代金が発生すること、また講師の技術力に疑義があること、該当のスクールについて IT 技術者からの悪評が目立つことなどから、現実的な選択肢ではないといえる。

また、学校教育などの場における「プログラミング」の内容をほとんど理解してしまっている児童・生徒らにとっては、学校や各種のイベントが取り扱う内容より高度なことに取り組んでみようとしても、それを牽引してくれる、あるいは困った時に相談できる「IT 関連の正しい知識・技術を持った大人」を周囲に見つけることが難しく、結果として高度な知識・技術に触れる機会を失っているという現状が挙げられる。

学校・地域の特性如何では、学校に掲示されたポスター等から高度な技術イベントの存在を知るようなことさえも難しいケースも多い。国・関係機関が運営するようなイベントのポスターでさえ、学校には渡るが実際に掲示すらされない、というケースも散見される。このような背景から学校の支援・環境づくりに期待ができない場合、保護者に IT 関連の知識があり、各家庭において教育ができることが最善であるが、IT 技術に習熟した保護者の全体数はさほど多くはないと推測される。

中学生・高校生といった若年層がサーバ構築やネットワークに関する専門的な技術を学ぶための教育イベントの例としては、サーバ・ネットワーク等が強く関連する「情報セキュリティ」の高度な学習を主題とし、毎年 8 月に独立行政法人情報処理推進機構（IPA）によって開催されている「セキュリティ・キャンプ⁶」が存在する。しかし、セキュリティ・キャンプは全国の 22 歳以下の学生・生徒を対象としたものであり、年齢に基づく対象層は大学生・一部の大学院生までと幅広いことから、参加者全体に占める中学生の割合は非常に低い。また、参加に当たっては応募課題の回答内容による書類審査の過程も存在するため、中学生・高校生にとってはやや敷居の高い環境となっている面が否めない。

なお、「セキュリティ・キャンプ」に積極的に参加している中学生・高校生は存在するものの、特に近年において、その絶対数は少ないものである。例えば 2018 年の開催回においては全体の参加者 85 名中、高等専門学校の参加者が 7 名、高等学校の参加者が 4 名、中学校の参加者が 1 名、中等教育学校の参加者が 1 名であった。

「セキュリティ・キャンプ」は、情報セキュリティの第一線に立つ技術者がそれぞれの専門分野の講義を行う、極めて高度な内容を扱うイベントとして知られている。情報技術に興味・関心がある中学生・高校生にとっては、先に述べたように、いきなり「セキュリティ・キャンプ」等の高度なイベントに応募することのハードルは相応に高いものとなるであろう。また、応募課題に対して回答するための知識・技術を身に着けるための場が身近

⁶ 情報処理推進機構「セキュリティ・キャンプ」>「セキュリティ・キャンプ事業紹介」

<https://www.ipa.go.jp/jinzai/security-camp/about.html>

にある中学生・高校生は決して多くない。当人がおかれた環境に強く左右されない、より広範な若手の情報技術者の育成という観点を考慮すれば、最初から「セキュリティ・キャンプ」に参加できるような知識・技術を持ち合わせているような生徒でなくとも、将来的にこのような先進的なイベントへの参加を見据えた上で、受講生が自らの知識・技術を高めていけるような場が求められていることは、論を俟たない。

このように、児童・生徒らにとって実践的な学習機会が不足している、より詳細に言えば「学校教育における情報技術の学習に満足していないが、セキュリティ・キャンプのような高度なイベントに参加するための知識・技術を有しているレベルには至っていない」児童・生徒らが「より高度な学習機会にアクセスするための、学校教育より高度で実践的な知識・技術を得る」ための場がないという状況は、将来にわたって各種の技術を維持・活用するための高度な IT 人材の育成という観点から、大きな損失を招きうるのではないかと考えられる。

1－2．クラウド化の傾向、物理的なネットワーク技術の学習機会の減少

近年の情報化社会に顕著な「クラウド化」の傾向は、情報技術を専門としない一般的なユーザにも、容易かつ安価に大規模なコンピュータ・ネットワーク・サーバを取り扱うことを可能にした。古くは 1990 年代最末期～2000 年代初頭頃から活用されているような「レンタルサーバ⁷」がその例となるが、2010 年代に入って以降、AWS や Google Cloud Platform などの「クラウドコンピューティングサービス⁸」が急速に知名度を獲得し、商用の大規模なサービスに用いられている例、またオンプレミスで構成されたシステムがクラウド環境に移行した例も数多く存在する。

確かにこれらのサービスを用いることで、何らかの Web サイト、あるいは Web アプリケーションを手っ取り早く公開したいユーザにおいては、本来は大変複雑な知識・技術や物理リソースを必要とするような「ネットワーク・サーバ」に関する構築の操作を最低限とし、その複雑かつ本質的な部分に関する作業をレンタルサーバ・クラウド事業者に転嫁することによって、Web サイト、あるいはアプリケーションの開発そのものに自らの開発リソースを集中させることができるだろう。

⁷ 一例として、さくらインターネット株式会社による「さくらのレンタルサーバ」は、年額 1,500 円ほどから利用できるなど、低価格で利用しやすいことで有名である。商業用途に利用できるエディションも提供されている。

⁸ これらのクラウドコンピューティングサービスは、利用したい時に利用したいだけの資源を都度課金にて利用できるという手軽さから、個人用途から商業用途まで幅広く用いられている。

しかし視点を変えれば、これらの技術を用いることで、ユーザはネットワーク・サーバ環境そのもののハードウェア・ソフトウェア的な動作について詳細に理解することなく、環境を利用するために必要な表層的な知識・操作方法のみを習得すれば良い状況が生まれてしまうと言えよう。したがって、これらの技術の発展は、それぞれの物理的な環境を動作させるための複雑な実装にユーザが直接触れる機会を減少させるものと成り得るのである。いわゆる「レンタルサーバ」の時代から連綿と続く「クラウドコンピューティングサービス」に関連する諸技術の発展以降、情報技術に強い興味や関心を持つ子供らにとって、物理的なサーバの動作や、インターネットとの直接の繋がりを知覚する機会は次第に失われつつある。

1－3. ICT 教育イベント「インデペンデンス・サーバー・デイ」の運営

このような背景を踏まえ、産学間連携推進室 ICT 教育プロジェクトでは、2016 年以降、グローバル IP アドレスと物理サーバを用いたサーバ構築・運用技術を学ぶ産学官連携 ICT 教育イベント「インデペンデンス・サーバー・デイ」を毎年継続的に運営している。本イベントは、つくば市総合教育研究所における通年の ICT 教育イベント「キッズプロジェクトプログラミング講座」の一環として実施されているものであり、「インデペンデンス・サーバー・デイ」は小中学生の夏休み期間に開催されるイベント「つくばプログラミングフェスタ」内の 1 講座として開催されているものである。

本イベントにおいては、将来の優れた IT 人材となりうる子供らに、サーバの基本的な構築方法や動作原理、セキュリティなど各種の先進的な問題について学習する環境を提供することを目標としている。

以降の章においては、本イベントの通算 9 回目の開催に当たり、2024 年 8 月に開催した「インデペンデンス・サーバー・デイ 9」に関する報告、2016 年以降の各回イベントにおける実施形式・講習内容の変遷や考察、また 2025 年に開催が予定されている「インデペンデンス・サーバー・デイ 10」に向けての取り組みについて述べる。

2. 「インデペンデンス・サーバー・デイ 9」の開催報告

2－1. 2024 年度開催「インデペンデンス・サーバー・デイ 9」の概要

産学官連携 ICT 教育イベント「インデペンデンス・サーバー・デイ 9」は、2023 年 8 月 21 日・22 日の 2 日間、両日とも 9:30 - 16:00 につくば市総合教育研究所において

「つくばプログラミングフェスタ 2024」の 1 講座として開催され、つくば市の中学生 10 名が参加した。なお、本年度は後に述べる事情から、例年通りの状況であれば応募者数が 40～50 名となることが想定されるところ、定員 10 名に対し、応募者数は 10 名ちょうどであった。

本イベントは 2018 年以降、事前のポスター・パンフレットなどでの広報に注力している。特に 2022-2023 年度においてはつくば市の全中学生の人数の 1% に当たる 50-60 名ほどの応募を記録したことから、本活動が広く周知されている様子を窺うことができ、広報によって特に大きな成果を得られていることは間違いないことである。例年の状況として、申し込み開始当日に申し込みを行う受講希望者、(自己申告によるが) 既に自宅でサーバを構築・運用した経験のある受講希望者が応募者に多くみられるなど、極めて意欲的な受講生が多いように思われる。

しかし 2024 年度の本イベントにおいては、つくば市内の公立学校において推進されていた配布物の「ペーパーレス」施策により、この広報活動に極めて大きな障害が生じることとなった。

当初、本イベントの主催組織であるつくば市総合教育研究所が紙媒体リーフレットの全生徒への配布に難色を示しており、結果として例年通りの「全員に紙媒体のリーフレットを配布する」形をとることが不可能となってしまったのである。結果として、本イベントの告知に関する情報の主要な流通経路は「学校からの連絡を保護者が電子的に確認できる Web サービス」上で、保護者の端末にしか配信されず、なおかつその配信形態もプッシュ通知などを伴うものではなく「多数のイベント情報が羅列されている中に 1 つ加わる程度」という形となったことから、応募者数に基づく広報効果は 2018 年度以降・2023 年度まで(リーフレットの全生徒への配布期間)と比較し 7-8 割減という、実に顕著な激減に至ったのである。

本イベントでは、従前より行ってきたリーフレットの個人配布により、各学校に在籍する全ての生徒に対してプッシュ型の直接的な訴求が可能であった。しかし本年度に限っては「電子的なシステム上で、保護者が本イベントを目ざとく発見し、なおかつ本イベントの趣旨を理解するか、本イベントを信頼することにより、その時点で初めて自らの子女に本イベントの存在を通知し、併せて本イベントへの参加を勧めるようになる」というプロセスが介入した形となってしまったのである。その結果として、本イベントの存在が対象者である生徒らに十分に広報されなかったことは、残念でならない。

後に広報の効果についてつくば市総合教育研究所と協議し、各学校 100 部ずつ（個人への配布ではなく、リーフレットを設置し生徒に持ってもらうことを前提としつつ、配布形式は各学校に委任）のリーフレット設置、またポスターの作成・配布を行うことはできたものの、応募期限の迫る中での対応となったこと、また個人配布による強力な広報効果を得られなかったことから、2024 年度の実参加者は例年比で激減し、個別配布のリーフレットが存在しなかった 2017 年度・第 2 回開催時の水準（定員 10 名・応募 10 名）まで落ち込むこととなったのである。

本イベントは、その目標として「高度な情報技術を学習しようとする若年層の人材が、その置かれた環境に左右されることなく、自ら学び進めることの難しい情報技術に触れることのできる場を築くこと」を掲げている。しかし、本年度に限っては上記の事由により、生徒が自由に選択することのできない「環境」がフィルタとなって情報流通を妨げ⁹、結果として情報技術に対する潜在的な興味・関心を有している生徒らの多くに本イベントの開催情報さえも行き渡らなかった、と推測される状況が生じてしまったように思われる。なお、本年度の状況はつくば市総合教育研究所とも共有しており、次年度以降は紙媒体での・各個人への配布を再開できる見込みであることを補足する。

「インデペンデンス・サーバー・デイ 9」の開催告知に利用したリーフレットは、次頁掲載の通りである。本リーフレットは、産学間連携推進室所属の学生のデザインによって、また協力組織の資金負担によって製作され、前述の通り、つくば市内の公立中学校各校に 100 部ずつ配布・設置された。なお、同様のデザインであるポスターも、つくば市内の全ての公立中学校に掲示された。

⁹ この表現は、つくば市内の公立学校に在籍する全中学生に本イベントの告知を行わんとする運営組織から見た際の結果的な事象を表現したものであり、情報流通を阻害することとなった「環境」における故意の有無は問わない。すなわち、このようなイベントの存在に無関心であり、そもそも情報流通が機能不全を生じているケースも想定されれば、「受験勉強をしなければならないのに、パソコンとはけしからん」「サーバを構築して、逮捕されるようなことになったら大変だ」などの明確な意図をもって情報流通を阻害しているケースも、また等しく想定されるものである。最近では「データ・センター」を建設しようとする、到底現実には発生しえないような理由を掲げた反対運動が見られる傾向があるが、それと似たような事象が少なからず発生しているのではないかと、という懸念は拭えない。とはいえ、つくば市に限ってみれば「インデペンデンス・サーバー・デイ」の受講を子女に勧めてくれるような、また場合によっては全時間帯の見学をも希望するような熱心な保護者の方が、少なくとも一定数いらっしゃるのである。このことは、実にありがたいことである。



INDEPENDENCE SERVERS DAY 9

- RENEWAL -

インデペンデンス・サーバー・デイ・9

～つくば市中学生対象 サーバー・固定IPアドレスふれあいイベント～

2024.08.21^(水)/22^(木) 09:30～16:00
両日参加必須

1人1つ固定グローバルIPアドレスとサーバーを配布

自分専用の固定グローバルIPアドレスとサーバーを持ち、インターネットに専用線でこれらを直結し、世界に向かって自由にサービスを公開したいと思ったことはありませんか。このイベントでは、参加者の皆様に1台ずつ、サーバー(Raspberry Pi)を配布し、筑波大学のコンピュータ系の学生による講習のもと、自由にサーバーを構築していただきます。また、固定グローバルIPアドレスを1人1個配布し、イベント終了後も、各自のサーバーを設置しておける電源およびインターネット専用線接続環境をつくば市総合教育研究所内に用意します。「レンタルサーバー」や「クラウド」、「NATの内側」では味わえない、真のサーバーおよびインターネット直結環境を用いて、高度でセキュアなサーバー構築の体験をしましょう。

講座内容

- サイバーセキュリティ専門家による安全のための講義
- インターネットサーバー構築に関する講義および実習
- 専用設備内に設置されたサーバーの継続運用

講師：筑波大学の情報科学類および大学院情報理工学学位プログラム等関係者数名
※ ご自分のPCをお持ちの方は是非ご持参ください。

開催情報

対象：つくば市内の公立中学校に在学している生徒
・ 中学校の中学生
・ 義務教育学校の後期課程生
・ 中等教育学校の中学生

定員：10名(超過時は抽選となります)
料金：無料
場所：つくば市総合教育研究所 2F (予定)

詳しくは <https://tsukuba.open.ad.jp> **お申込み** <https://forms.office.com/r/Hv68EufV6B>

主催：つくば市総合教育研究所 共催：筑波大学 情報学群 情報科学類 協賛：筑波大学 OPEN プロジェクト ソフトイーサ株式会社 輝日株式会社 太平電算

図1：「インデペンデンス・サーバー・デイ・9」のリーフレット

2-2. 「インDEPENDENS・サーバー・デイ 9」の主催・後援・協力組織

本イベントの主催組織は、つくば市総合教育研究所である。後援組織である筑波大学情報科学類には、講座において利用する機材の貸与や費用の支援を受けているほか、産学間連携推進室の学生が本イベントの直接的な運営に関与している。協力組織である筑波大学 OPEN プロジェクトおよびソフトイーサ株式会社・輝日株式会社には、同様に機材の貸与や費用の援助、グローバル IP アドレスなどの提供を受けている。

本イベントの実施に当たっては、ソフトイーサ株式会社によってつくば市総合教育研究所の実習室内に光ファイバ線（フレッツ光）の引き込みが行われ、筑波大学 OPEN プロジェクトの技術によって、フレッツ光を経由して OPEN プロジェクトのネットワークを利用できる状態としている。このようにして構築されたネットワークは、本イベントの開催当初より現在に至るまで、「インDEPENDENS・サーバー・デイ」において受講生が構築したサーバを運用するために継続して活用されている。

また、本イベントは産学間連携推進室内の「情報機器資源リユースプロジェクト」を中継し、筑波大学システム情報系技術室より、受講生が遠隔操作の実習・調べ物などで一時的に利用するための、情報機器資源リユースプロジェクトによってリユースされた Windows ノート PC 8 台（図 2 にその一部を示す）、ディスプレイ 13 台の貸与を受けている。



図 2：情報機器資源リユースプロジェクト・技術室経由で貸与を受けた Windows ノート PC

更に、イベント当日に受講生の実習の補佐をするスタッフとして、開催当初より筑波大学情報科学類の学類生有志、情報メディア創成学類の学類生有志、情報理工学位プログラム（旧・コンピュータサイエンス専攻）大学院生有志の協力を受けている。

また、本年度は会期前である 2024 年 7 月に「日本ネットワーク・オペレーターズ・グループ JANOG54 Meeting」内「若者へのネットワーク技術継承 BoF」において「インDEPENDENCE・サーバー・デイ」の取り組みを広報する機会が存在した。当該の BoF に関連する熱心な議論を経て、本年度は上述の筑波大学関係者・協力企業各社のスタッフに加え、当該講演の共同講演者である 株式会社ミラパス 小島様、当該講演への参加者である BGP NAP 小坂様、また 慶應義塾大学 SFC 学部生である岡田様の協力を受けることができた。

[Day2] 2024年7月4日(木)		
開催時間	開催場所	タイトル
10:00～11:30	会議室 201+202	TS-OPS(時刻同期) BoF
10:00～11:30	会議室 205	若者へのネットワーク技術継承BoF
10:30～11:30	会議室 206	地域NOG BoF
11:40～12:20	会議室 206	無線LAN認証連携BoF

図 3：JANOG54 BoF 一覧より「若者へのネットワーク技術継承 BoF」

何より当該 BoF へのお誘いを頂いた株式会社ミラパス 小島様をはじめ、関係の皆様方にはこの場を借りて、厚く御礼を申し上げる次第である。

2－3．対象とする受講生について

本イベントの対象は、つくば市内に在学する中学生で、2024 年 8 月 21 日および 8 月 22 日 両日 9:30～16:00 の時間帯の全てを通して対面形式のイベントに参加できる、以下のスキルと意欲を持つ生徒とした。

- パソコン操作（キーボード、インターネット検索）ができること。
- 「TCP/IP」、「ポート番号」などの用語の意味・概念が少しわかること。
- インターネットサーバー構築に興味があること。

2-4. 講習内容について

本イベントでは、参加者のそれぞれに小型の SBC である Raspberry Pi 3B を 1 台ずつ配布し、Raspberry Pi 上の Raspberry Pi OS (debian ベース、旧称 Raspbian) で動作する Apache Web サーバを用いた基本的なサーバ構築手法と、HTML・CSS を用いた簡易な Web サイトの作成方法に関する講義・実習を行った。講習の様子を収めた写真を、下記図 4 に示す。



図 4:「インDEPENDENCE・サーバー・デイ・9」開催の様子

下記に、講座において取り扱った内容の概略を箇条書きで示す。

- サイバーセキュリティ専門家による、公開サーバのセキュリティの講義・諸注意
- コンピュータの動作原理および OS の基本的な概念、および Linux の基礎についての講義と、ターミナルを用いたコマンドによるディレクトリ・ファイル操作の実習
- グローバル IP アドレスとホスト名の設定を実習形式で行ったのち、ping、traceroute、nslookup などのネットワークコマンドの操作方法に関する実習
- Raspberry Pi OS への Apache のインストールと基本的な設定、およびサーバ上への HTML コンテンツの配置
- HTML と CSS の記述による Web サイト制作の実習
- 実際にサーバをラックに設置したのち、サーバへの SSH・SCP による遠隔接続の実習、および遠隔操作方法に関する説明
- 自由な質疑応答コーナー「なぜなにコーナー」

また、本年度は上記の講座の流れに加え、下記に示すさまざまな体験コーナー・説明コーナーを設け、休み時間やイベントの終了後などに、受講生らが積極的な体験を伴って情報技術に触れていくことのできる取り組みを行った。いずれのコーナーも多くの受講生が実際に体験を行っており、受講生の低レイヤ・ネットワーク技術に対する強い興味・関心が感じられた。

- LAN ケーブル成端体験コーナー（株式会社ミラパス 小島氏による）
- 光ケーブル成端体験コーナー（ソフトイーサ株式会社による）
- 1U サーバ分解体験コーナー（産学間連携推進室 服部氏・関口氏による）
- フレッツ光関連機材 見学コーナー（輝日株式会社による）

これらの取り組みのうち「1U サーバ分解体験コーナー」は従前より行ってきたものであるが、他の取り組みは今年度に初めて行ったものである。以降において、その様子を示す写真をいくつか掲載する。



図 5：LAN ケーブル自作（成端）体験コーナーの様子

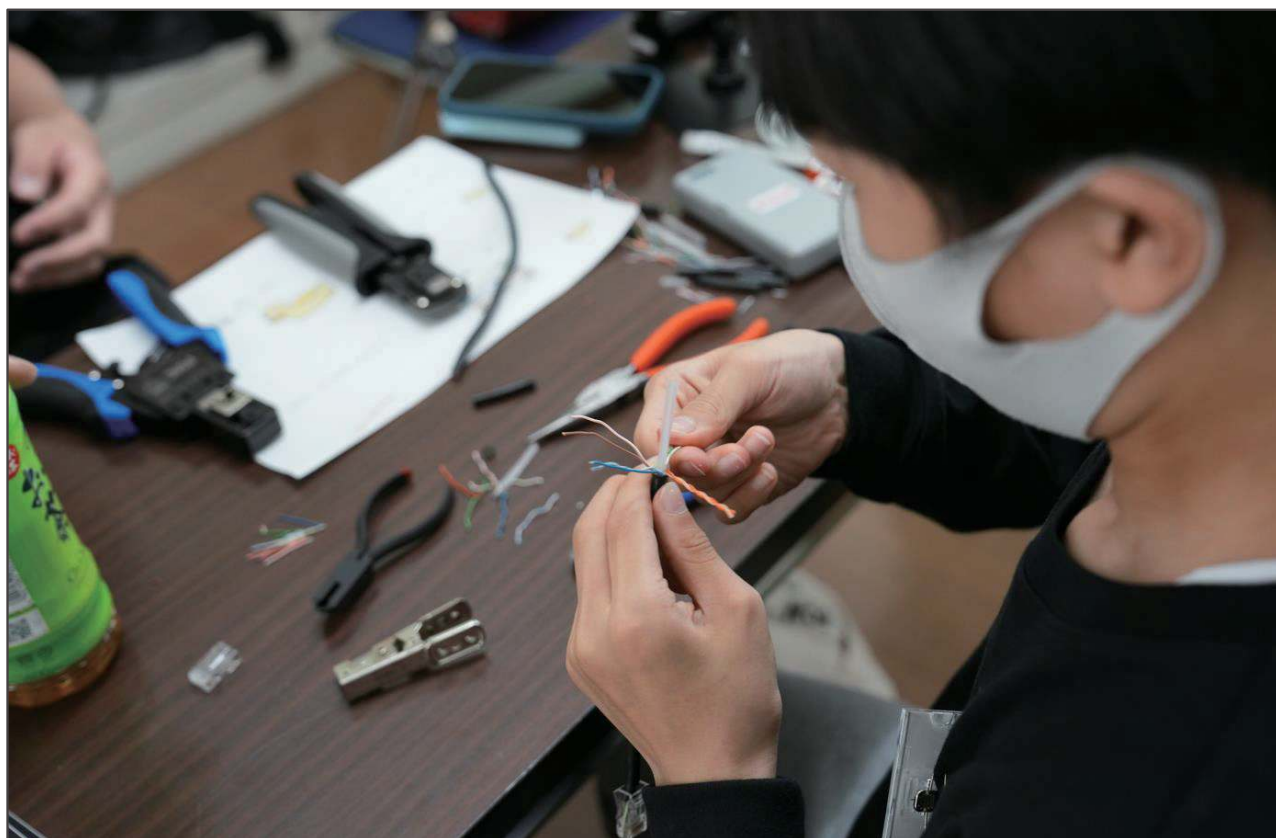


図 6：LAN ケーブル自作（成端）体験コーナーで LAN ケーブルの成端に挑戦する受講生



図 7：光ケーブル自作（成端）体験コーナーにおける、
可視光を用いた光ファイバケーブルの構造の提示・光信号の伝送の実演



図 8：毎年恒例となった、1U サーバ分解体験コーナー
 （分解する機材の提供は、情報機器リユースプロジェクト・システム情報系技術室による）

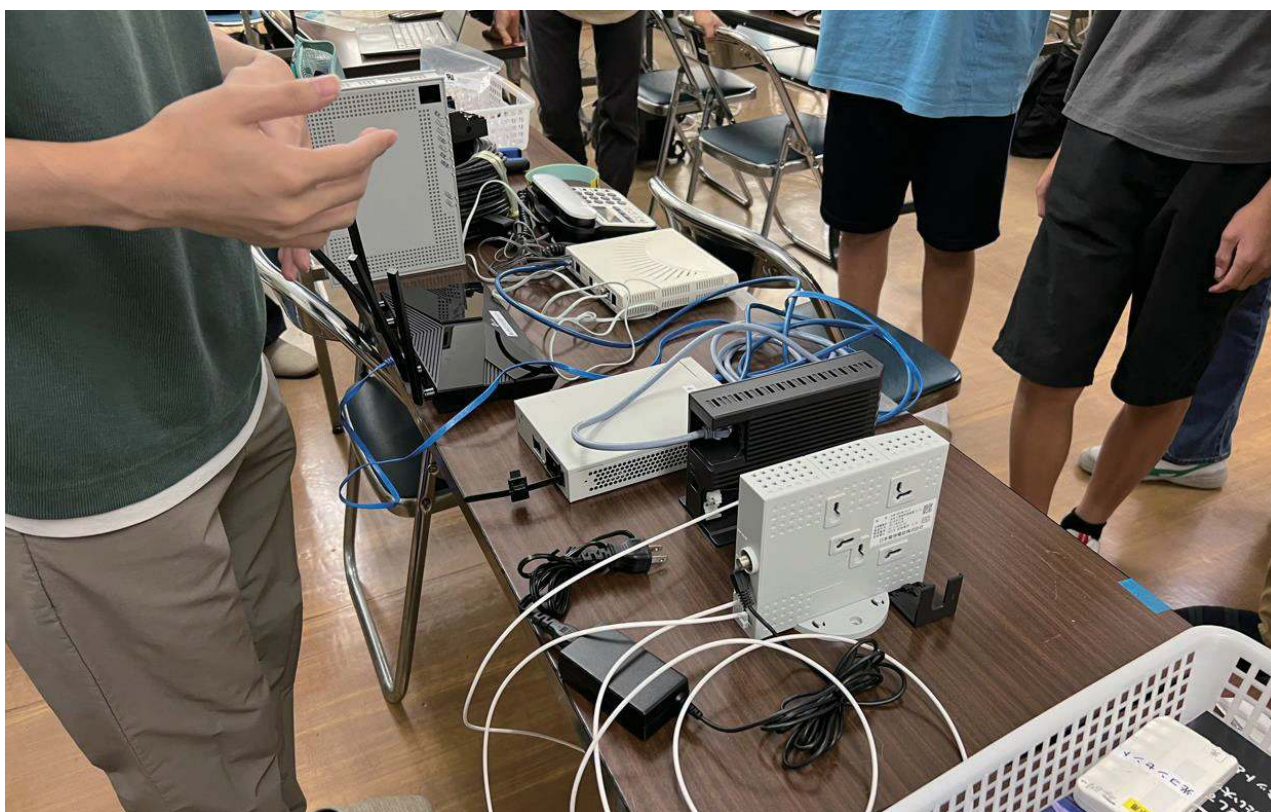


図 9：フレッツ光関連機材見学コーナー

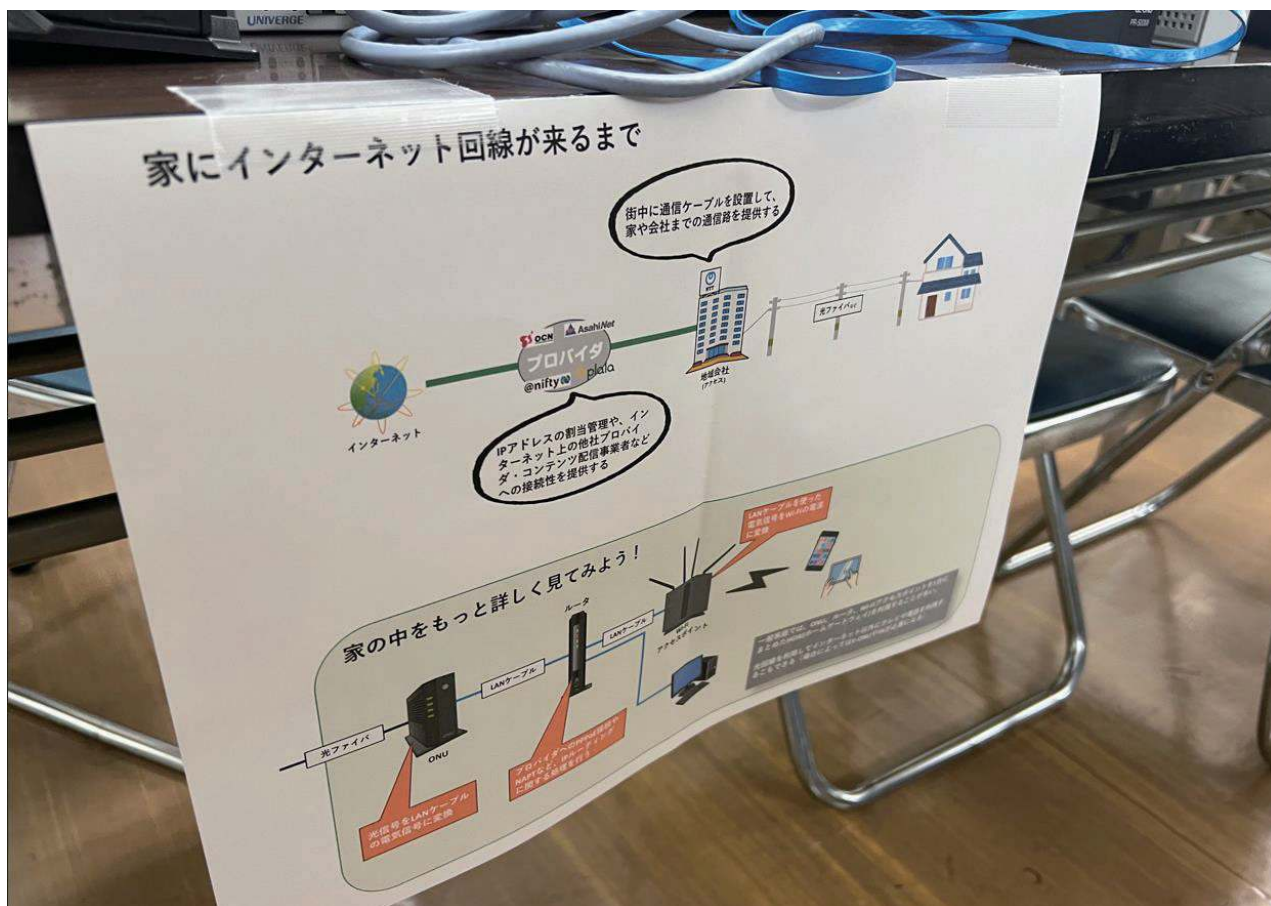


図 10：フレッツ光関連機材見学コーナー掲示の「家にインターネット回線が来るまで」

なお、当日の講習においては、それぞれについてオリジナルのスライドを作成し学習用の教材とした。「コンピュータの動作原理および OS の基本的な概念、および Linux の基礎についての講義と、ターミナルを用いたコマンドによるディレクトリ・ファイル操作の実習」は服部氏が、「グローバル IP アドレスとホスト名の設定を実習形式で行ったのち、ping、traceroute、nslookup などのネットワークコマンドの操作方法に関する実習」は太田氏が、「HTML と CSS の記述による Web サイト制作の実習」は北野氏・OB 西山氏が、それぞれ教材作成・講義の実施を担当した。また、それぞれの分野の専門書を受講生らに配布し、事後の学習に役立ててもらえるように配慮した。

イベントで受講生らが利用した Raspberry Pi はイベント以後もつくば市総合教育研究所内に設置し、受講生が遠隔ログインを行って引き続き実習を行えるようにした（図 11 にその写真を示す）。また、単純なサーバの構築手法を解説するだけでなく、近年の中学生の関心の強い話題を用いてサーバ・ネットワークについて受講生により強い関心を持ってもらうため、図 12 のようなスライドを用いて「なぜなにコーナー」と称する自由な質疑応答コーナーを設け、参加者の中学生から寄せられたさまざまな疑問・質問に講師陣が回答した。

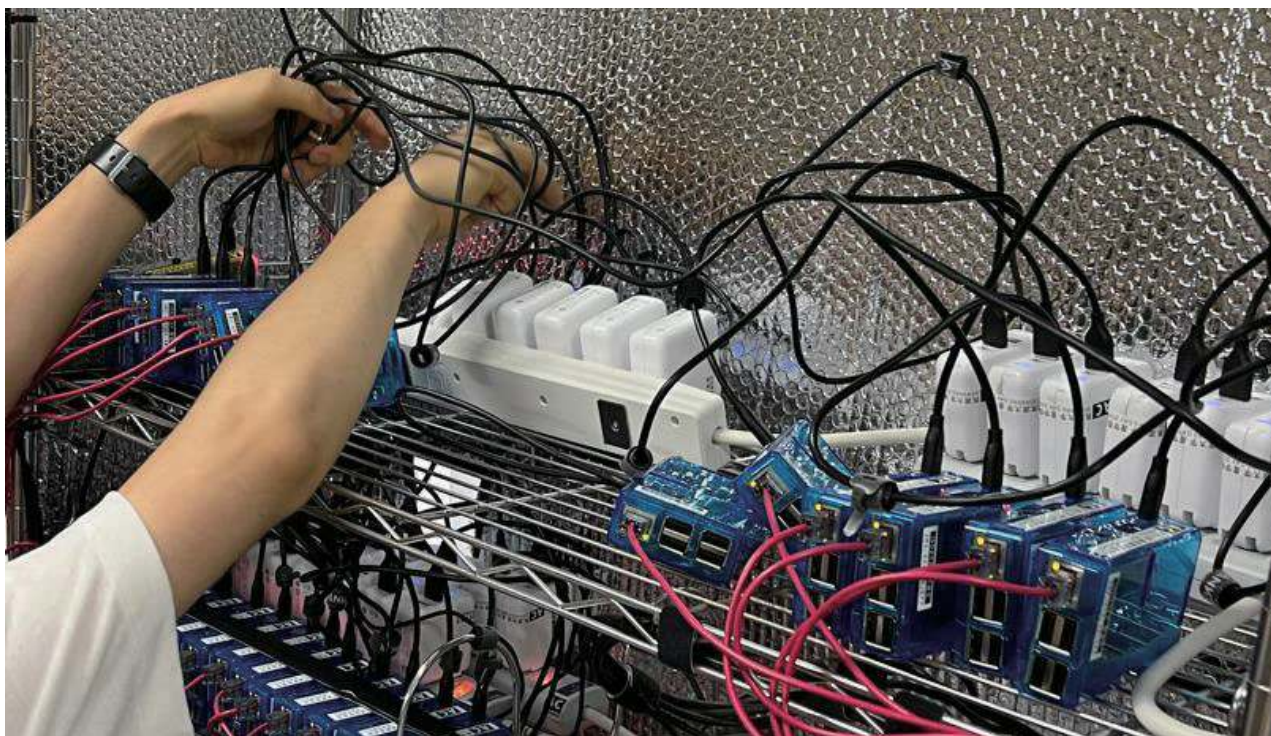
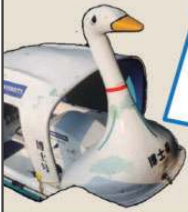


図 11 : 「インデペンデンス・サーバー・デイ・9」において受講生が設置したサーバ

Q4.

AO 入試（筑波大学 AC 入試）で 提出したものの内容を教えてください。





**情報技術を活用して、視覚障害者が快適かつ安全に生活
できる方法の提案**をしました。【服部】

部活のいろいろなことを管理するシステムを作った話、
ライブラリにコントリビューションした話、**ゲームとその
ためのコントローラ**を作った話などを書きました。【細島】

チャットBotを作ったり/**サーバー・ネットワーク**をやっ
ているぜー/**セキュリティ・キャンプ**などなどなどに**参加
したぜー**ということを書きました。【間瀬】

【このスライドは 2024 年 8 月 21 日 18:00 時点の情報を元にしてます / 各 FAQ の内容は講師個人の感想や経験に基づくもので、再現性は保証されません】

16

図 12 : 「なぜなにコーナー」における質疑応答の例

また、本イベントはサーバの継続的な運用の実習の場としても活用できるようにするため、運用上の問題や技術的な質問などに回答するための連絡の場である Slack を設けるなど、受講生らの興味・関心を長期的に満たすことのできるような運営を行っている。

図 13 に、Slack において実施された受講生とのやり取りの例を示す。

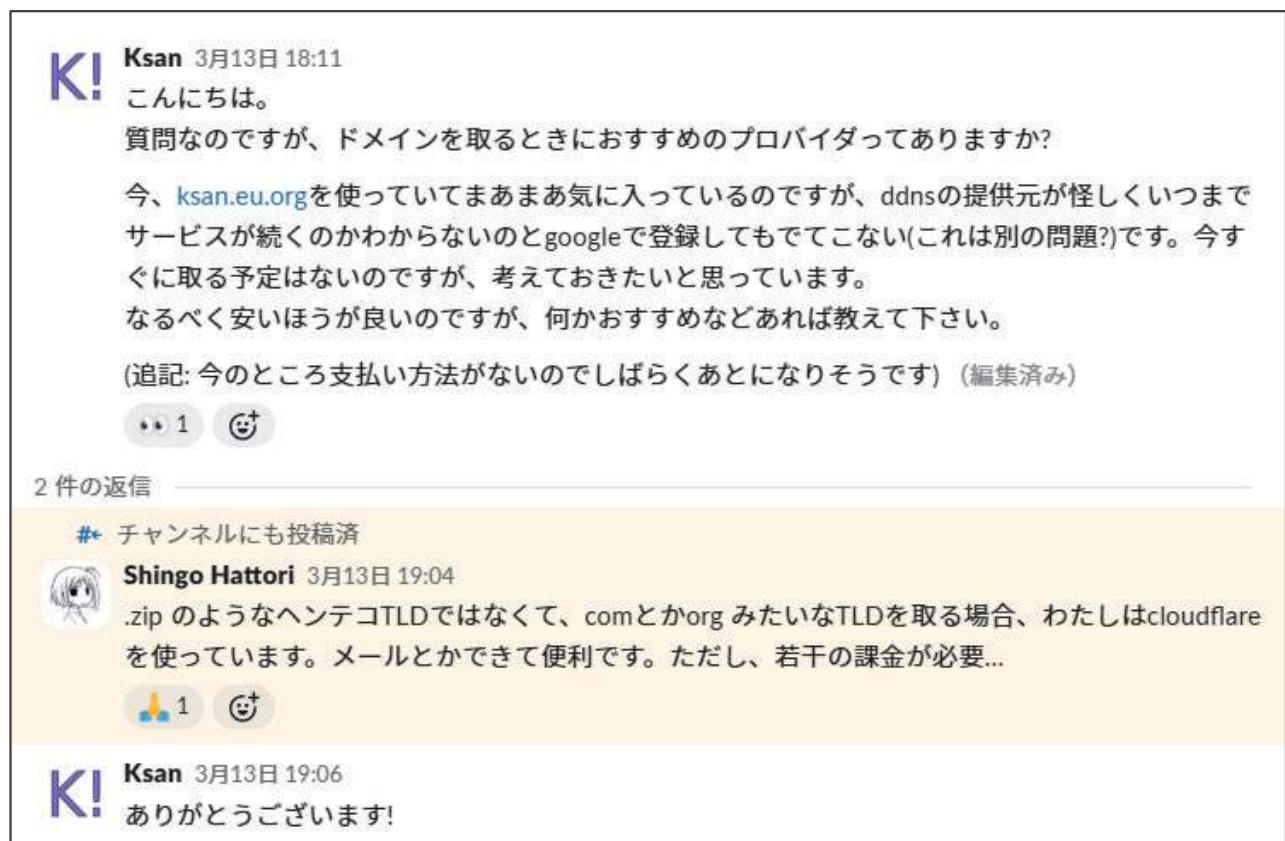


図 13：Slack における受講生との継続的なやり取り

2-5. イベント実施を踏まえて

本イベントにおいては、各回の受講生に対して「講座の難易度」や「扱った問題の難易度」「講座への感想」などの項目によるアンケート調査を実施している。「インデペンデンス・サーバー・デイ・9」においても、本イベントは中学校におけるコンピュータ（技術家庭科）の授業を易しいもの・物足りないものと感じる受講生らにとって、学校の授業では扱わないような幅広い事項を 2 日で扱うというややペースの速い・ごく実践的な実習¹⁰であり、ある程度高い難易度でありながらも、楽しみながら実践的な内容を理解してもらえるイベントとなったと考えられる。

¹⁰ 筑波大学情報科学類において開設されている授業である「コンピュータリテラシ」を基準として考えれば、大学生が数週間かけて学習するような内容を 2 日でこなす内容となっていると言えるだろう。また、一般的な IT 企業の新人研修（情報系の強力なバックグラウンドを持たない新人に対する技術研修）として考えるのであれば、少なくとも 2-3 週間以上をかけて、少しずつ慣らすように扱っていかなければならないような内容である、とも言えるだろう。

3. 「インデペンデンス・サーバー・デイ」の各回実施における変遷

本イベントは、2016 年 8 月からの計 9 回の実施の中で、その講義内容・実施形態について応募者層の変化、受講生（あるいは応募者）の反応、学習機材の調達市況等に基づき、さまざまな変化が生じている。まずは「インデペンデンス・サーバー・デイ」の初回の開催の概要について示した後、2023 年度の開催に至るまでの変化の概要について、下記の通り示す。

3－1. 2016 年：「インデペンデンス・サーバー・デイ」初回の実施

2016 年 8 月 23 日に開催された「インデペンデンス・サーバー・デイ」には、定員 15 名に対し 5 名の受講生が参加した。当時は現在のようなリーフレットによる広報をほとんど行っていなかったこと、また他に類を見ないイベントであったことから、定員を大きく割り込んでの開催となった。

当該回におけるコンピュータやネットワークの概念に関する講義、サーバ構築の実習については、標準で利用するエディタが 2016 年の実施回のみ vi であったこと、また 2016 年当時は講座全体が 1 日での開催であったことを除き、2024 年とほとんど変わらない内容の構成にて行った。サーバ上に展開するコンテンツについては、運営側が用意した PHP による Twitter Bot を展開する、といったものであり、その難易度に比して当日の講義・実習に十分な時間を用意できなかったことから、受講生の知識・技術のレベルを実習ベースで引き上げることへの課題が残る実施となった。

なお、当初は受講生が興味を持ってもらえると推定される題材として選定した「Twitter Bot」は、参加者内での Twitter の利用率が極めて低かったこともあり、もとより受講生の興味・関心を集められるものでは無かったとも考えられる。

3－2. 2017 年：「インデペンデンス・サーバー・デイ」2 回目の実施

2017 年 8 月 2 日に開催された「インデペンデンス・サーバー・デイ 2017」には、定員 10 名に対し 10 名の応募があり、応募した 10 名全員が受講生として参加した。当該年より本イベントはつくば市総合教育研究所の「キッズプロジェクトプログラミング講座」に組み込まれ、同講座が夏休みに開催するイベントである「つくばプログラミングフェスタ」において、プログラミング・情報技術を扱ういくつかの他の講座と同日の開催となった。なお、現在では日程の都合上、「つくばプログラミングフェスタ」の他の講座と本イベントは別日での開催となっている。

前年度の開催からの変更点は、大きく分けて 2 点である。

そのうちの 1 点目は、講習中に受講生が利用する（講習において利用を推奨する）エディタを vi から nano に変更したことである。vi は有用なテキストエディタであり、講師・スタッフの多くが操作に習熟しているが、vi に初めて触れる受講生にとって、これを短時間で使いこなせるようになることは難しいものであった。一方で nano は vi より操作を行いやすく、また実習環境・他の多くの Linux ディストリビューション環境においても同様に操作を行うことができるため、受講生にとっては格段にテキストの記述が行いやすくなったと言えるだろう。これにより受講生のコンソール環境におけるテキスト編集の簡易性が増し、各種の設定ファイルの編集等をよりスムーズに進めることが可能となった。

2 点目は、サーバ上に展開するコンテンツを HTML および CSS から構成される Web ページとし、運営側で用意したサンプルファイルを受講生が各自のサーバ上に展開し、それを用いて基本的な HTML のタグや CSS による装飾を学ぶという形をとったことである。しかし、2017 年度に運営側で用意した Web ページのサンプルは一般的な企業のコーポレートサイトに近い、やや複雑な構成のものであったため、そのサンプルを受講生が自在に改変したり、受講生が自身のコンテンツを設置したりする例はほとんど見られなかった。

このように、実習に用いるソフトウェア、および実習を通して取り扱う題材については、前年の実施回より簡易かつ基本的な技術を用いたものとなったものの、運営側で用意したサンプルが複雑なものであったため、受講生にとってはそのサンプルを活用した学習が難しかったものと考えられる。この反省は、2018 年度の開催に大きく活かされることとなった。

3-3. 2018 年：「インデペンデンス・サーバー・デイ」3 回目の実施

2018 年 8 月 1 日に開催された「インデペンデンス・サーバー・デイ 3」には、当初の定員 10 名に対し多数の応募があり、抽選の結果として 11 名の受講生が参加した。なお、当該年も本イベントはつくば市総合教育研究所の「つくばプログラミングフェスタ」内において、他の講座と同日での開催となった。本イベントは他の多彩な、なおかつ集客力が大きいと思われる講座と排他的な申込形態をとるイベントであったが、前述のリーフレットの各個人宛の配布による広報の効果がみられたこともあり、本イベントのみの応募者数は定員の 4 倍程度と、前年に比して格段の盛況が見られた。

前年度の開催からの変更点は、大きく分けて 3 点である。

そのうちの 1 点目は、受講生用サーバとして利用する機材を Raspberry Pi 3B から Raspberry Pi 3B+ に変更したことである。この変更は、2018 年 6 月に日本国内で Raspberry Pi 3B+ が発売され、受講生向けに最新の、より良い環境を提供したいというモチベーションが生じたことによる。これにより、確かに受講生が利用するサーバの処理性能はある程度向上したものの、実際に受講生が実習を行う上で体感できるような変化、あるいは Raspberry Pi 3B+ でなければ行えなかったような実習・取り組みの内容の変化は生じなかった。一方で、Raspberry Pi 3B+ は Raspberry Pi 3 よりも発熱量が多く、サーバーラック内に密集させたことによる過熱、また過熱による動作の停止が見られた。このことから、Raspberry Pi 3B+ を利用したのは当該年度のみであり、Raspberry Pi 4 が発売されて以降の 2021 年に至るまで、引き続き Raspberry Pi 3B が利用されることとなった。

2 点目は、HTML および CSS のサンプルとなる Web ページを非常に簡易なものとし、HTML のタグ構造を読み取りやすく、また編集しやすくしたことである。これにより、受講生がサンプルページを改変する形での学習に取り組むやすい環境を整えることができたと言える。しかし、2018 年時点ではイベント開催は 1 日のみであり、イベント当日の実習時間を十分に確保できなかったことから、HTML および CSS を用いた Web ページの実習については、その多くが受講生の自主学習の範疇に組み入れられることとなった。この反省を活かし、2019 年度以降に導入される 2 日間開催の形態が検討された。

3 点目は、受講生がイベント終了後も本イベントのスタッフに気軽に技術的な相談を行えるよう、更なる発展的な学びの場としての環境を設けるべく、従前より用意していたメールでの運営への連絡・質問に加え、グループチャットシステムである Slack を用いてスタッフ・受講生の交流の場を開設したことである。しかし、受講生らの家庭の方針などからグループチャットシステムに加入するという事への敷居が高かったこと、Slack というツールの非技術者層への認知度が低かったこともあり、結果的に Slack グループに加入した受講生は参加者 11 名中 3 名であった。

3-4. 2019 年：「インデペンデンス・サーバー・デイ」4 回目の実施

2019 年 8 月 22 日・23 日に開催された「インデペンデンス・サーバー・デイ 4」は、前年度に講座の時間が不足したことを受け、2019 年 8 月 9 日に先んじて開催された「つくばプログラミングフェスタ」とは独立した 2 日間の連続的な日程にて実施する運びとなった。開催日程が異なることから「つくばプログラミングフェスタ」の他講座に申し込んだ受講生でも参加できる形態とし、講座当初の定員 10 名に対し 3 倍程度の申込があり、抽選の結果として 12 名の受講生が参加した。

前年度の開催からの変更点は、大きく分けて 2 点である。

1 つ目は前述の通り開催日程を 2 日間とし、受講生が HTML や CSS を用いて各自のコンテンツを制作する時間を 2 日目に集中して確保したことである。当日の運営スタッフが受講生の学習を適度に支援したことにより、受講生 12 名のうち 11 名がサンプルとなる Web ページを改変し、あるいは独自の Web ページを制作することによって、サーバ上に独自のコンテンツを展開することができた。

2 つ目は、受講生が自主学習において利用するために配布している市販の技術書の難易度を落とし、より平易なものとしたことである。具体的には、従来は HTML と CSS を用いた Web ページの「デザインの技法」について扱う書籍を用いていたところ、当該年度においては HTML のタグの説明や CSS の指定の説明などの基礎的な内容を中心に扱う書籍に変更したものである。これにより、先に述べた HTML / CSS に関連する事項の実習時間の確保と併せ、受講生が独自のコンテンツを制作・構築することを大きく支援することができたと言える。

3-5. 2020 年：「インデペンデンス・サーバー・デイ」5 回目の実施

2020 年 8 月 17 日に開催された「インデペンデンス・サーバー・デイ・5」は、同時期に流行した新型コロナウイルス感染拡大の影響を受け、従前の対面形式での開催とは異なり、Zoom によるオンラインでの開催となった。また、新型コロナウイルス感染拡大の影響により中学生の夏休み期間が短縮されたことから、前年度実施された 2 日間での開催は行われず、2016-2018 年と同様に 1 日開催の形式となった。中学生にとっては学業、私事など多彩なイベントが短い夏休みに集中したこともあってか、2020 年度の申し込み数は前年より少ない 20 名弱であったが、定員 5 名に対し倍率ベースで 3-4 倍程度の申し込みがあり、最終的に 6 名が参加したことから、抽選の倍率そのものは前年度と大きく変わらない状況であった。

2020 年度の「インデペンデンス・サーバー・デイ・5」においては、オンラインでの開催であったことから、Raspberry Pi の実機に触れてのネットワークの設定など、本講座の根幹を成す内容の一部を講師・スタッフが事前に行うこととなった。また 1 日開催であったことから、講座内で扱うことのできる内容も前年度より絞られることとなり、学習効果は前年の「インデペンデンス・サーバー・デイ」と比較し大きく低下してしまったのではないかとと思われる。

1 日開催という状況自体は 2018 年までの実施形態に戻った、と考えればそれまでであるが、2018 年までのイベントにおいて発生していた時間の不足は、本年においても発生することになってしまった。

3－6．2021 年：「インデペンデンス・サーバー・デイ」6 回目の実施

2021 年 7 月 29 日・30 日に開催された「インデペンデンス・サーバー・デイ 6」は、前年度の反省を生かし、新型コロナウイルス感染症対策を十分に実施した上で、Raspberry Pi を利用した物理的なサーバのセットアップ過程のみ、全受講生を半数ずつに分けての対面開催を行い、HTML や CSS による Web コンテンツの制作過程については全受講生が同時にオンラインで受講するというハイブリッド形式を採用して開催された。2019 年同様の「完全な 2 日間開催」とは至らなかったが、本講座における「物理的なサーバのセットアップ」という重要な部分を、対面で、受講生自らがサーバに触れる形で行うことができたことは、混迷する世相の中で、大変に喜ばしいことである。

なお、講習の内容は概ね 2019 年に準じる形となった。2019 年に引き続き「なぜなにコーナー」という自由な質疑応答コーナーを導入したことにより、受講生からの技術的な興味・関心に基づく積極的な質問が見られるようになった。また、短時間ではあるが「サーバ分解コーナー」という、実際にラックマウントサーバの中身を眺めることのできるコーナーを設けたのも 2021 年からの変更点である。このコーナーは、2024 年の開催に至るまで長らく引き継がれている。

3－7．2022 年：「インデペンデンス・サーバー・デイ」7 回目の実施

2022 年 8 月 22 日・23 日に開催された「インデペンデンス・サーバー・デイ・7」は、2019 年以来の完全対面形式、および 2 日間全日での開催となった。ただし新型コロナウイルス感染拡大の影響から、受講生数は 2019 年同様とはならず、募集定員は 8 名に絞られた。なお、応募倍率は 7 倍近くと過去最高を記録し、実際の参加者は 2 名の欠席により 6 名となった。

2022 年においては、半導体の価格高騰・急激な円安の進行等により、Raspberry Pi を含めたシングルボードコンピュータが極度の品薄に見舞われ、その調達が著しく困難となった。次頁図 14 に、2022 年 7 月 22 日時点で Amazon.co.jp において記録した Raspberry Pi 3B の販売価格を、図 15 に同日時点での Raspberry Pi 3B の価格推移グラフを掲載する。

そのため、2022 年の「インデペンデンス・サーバー・デイ・7」の開催に当たっては、Raspberry Pi の代替となるシングルボードコンピュータを選定するべく多様なシングルボードコンピュータ製品の比較・検討（使い勝手、性能、2022 年当時の調達可能性の検証等）を行い、結果として 2022 年度のみ ASUS Tinker Board S R 2.0 を採用することとなった。



図 14：Raspberry Pi 3 ボードの価格（2022 年 7 月 22 日に amazon.co.jp にて記録）



図 15：Raspberry Pi 3B の価格推移（2022 年 7 月 22 日に記録）

なお、2022 年の「インデペンデンス・サーバー・デイ・7」は 2019 年の開催とほぼ同様の開催様式であったことから、2021 年に追加された「サーバ解体コーナー」を除いては、2019 年の開催当時より大きな変更点は生じていない。ただし、技術面においては ASUS Tinker Board S R 2.0 について、従前採用していた Raspberry Pi と概ね同様の扱いが出来るようにするため、Tinker Board S R 2.0 に導入する OS、また投入する設定等について、講師・スタッフを交えてさまざまな議論・挑戦がなされた。

3-8. 2023 年：「インデペンデンス・サーバー・デイ」8 回目の実施

2023 年 8 月 23 日・24 日に開催された「インデペンデンス・サーバー・デイ・8」は、2019 年以来の完全対面形式、および 2 日間全日、また受講生数を 2019 年段階の形式に戻しての開催となった。応募倍率は 5 倍近くと高い値を記録し、実際の参加者は 11 名となった。

2023 年においては、協力企業が Raspberry Pi 3B の安価・多数の調達に成功したことを受けて、利用するシングルボードコンピュータを再び Raspberry Pi 3B に戻す形がとられた。講習の内容は概ね 2019 年に準じ、2019 年以降に導入された「なぜなにコーナー」、2021 年以降に導入された「サーバ解体実習コーナー（図 16）」等も維持することで、最も大規模に・充実した内容の講習を行うことができたものと思われる。



図 16：実際に触れる形で再開した「1U サーバ解体コーナー」

また、実際に IT 企業における技術者育成に関わっている IT 技術者の監修を受けたことにより、講座全体の品質を向上させることができたほか、実践的な情報技術学習の場の構築に強い興味・関心を有する技術者に実際に「インデペンデンス・サーバー・デイ」のサポートスタッフとして参加して頂くことにより、この取り組みを広報し、また実際の企業での実践的な情報技術学習の取り組みに活用してもらえるような足掛かりとすることができた。

3-9. 2024 年：「インDEPENDENS・サーバー・デイ」9 回目の実施

2024 年 8 月 21 日・22 日に開催された「インDEPENDENS・サーバー・デイ・9」は、2023 年同様の完全対面形式、および 2 日間全日、10 名規模での開催となった。前述の状況から生徒らに対する直接的な広報が不十分となってしまったため、応募倍率は定員通りの 1 倍となり、応募者・実際の参加者は 10 名であった。

しかし、応募者の人数が従前比で減少した状況以外の点では、LAN ケーブル・光ケーブルの自作（成端）を行える体験型コーナーの設置、「インターネット回線が自宅に来るまで」といった極めて身近なテーマの見学型解説の実施、JANOG BoF を通した活動の共有と若年層向けの取り組みの活発化を目指した呼びかけ、そして同 BoF 等を通しての筑波大学外からの多くの協力が得られたことなど、受講生にとって有益な、また運営にとっても有益な、さまざまな取り組みを行うことのできた会期であったように思われる。また何より、「インDEPENDENS・サーバー・デイ」の講義資料の一部を印刷・製本業者による製本を行った形態で配布することができ、国立国会図書館への納本にも至ったことは、9 年来の大きな成果といえよう。

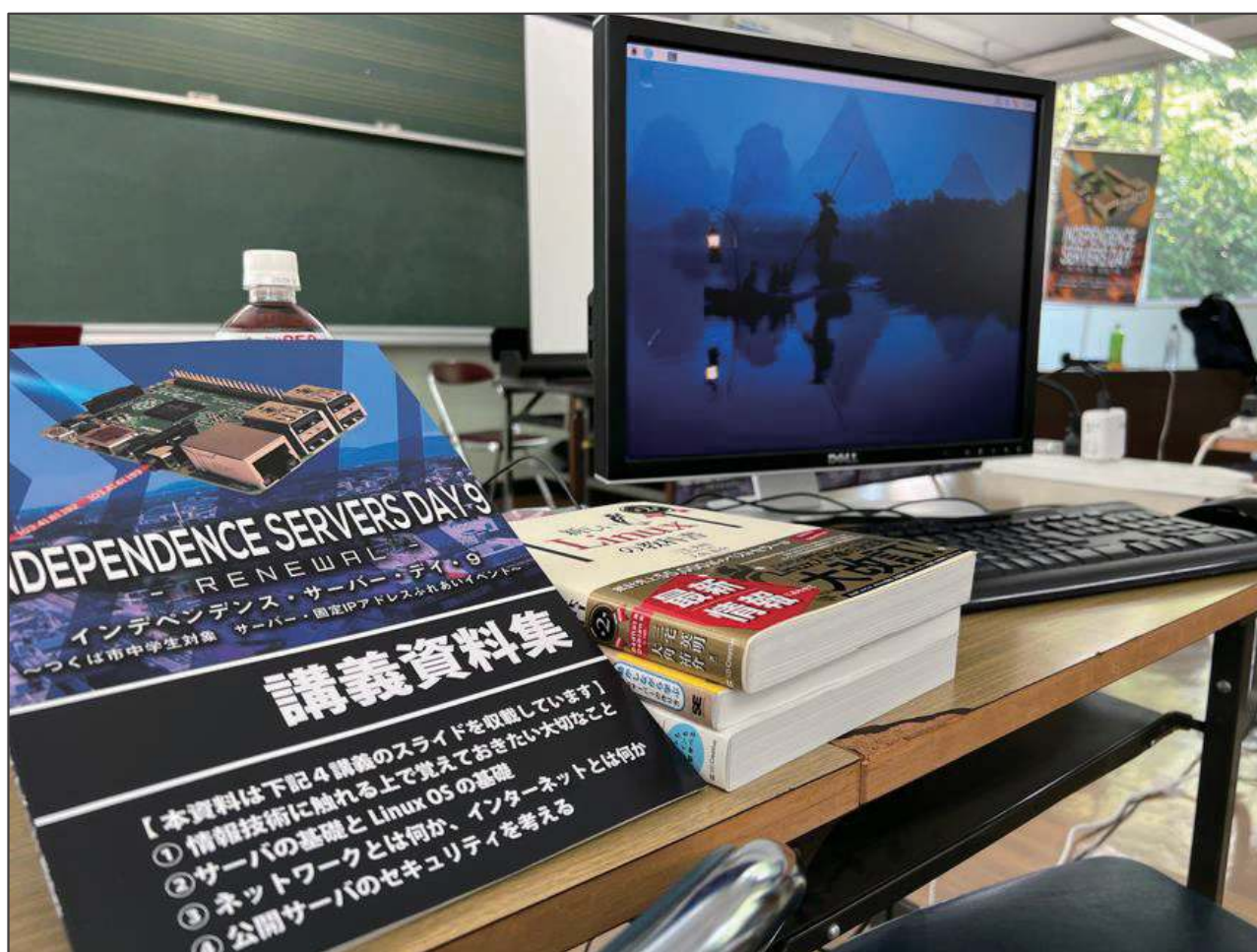


図 17：「インDEPENDENS・サーバー・デイ 9」講義資料集と各種資料の様子



図 18：国立国会図書館における「インデペンデンス・サーバー・デイ 9」講義資料の収蔵

3-10. 「インデペンデンス・サーバー・デイ」全体の変遷を通して

ここまで述べたように、「インデペンデンス・サーバー・デイ」は、2016 年の初回の実施から 2024 年まで 9 回の実施の間において、年ごとに前年度の反省を活かし、講義・実習内容を変化させている。全体の変遷の中での受講生を対象としたアンケート結果、および技術的な質問・サーバの利用状況の調査の結果やそれらの変化を照合することにより、講義・実習内容の変化が受講生にいくつかの変化をもたらしていることが読み取れる。その内容について、次の通り示す。

まずは 2019 年度の開催において、開催日程を 2 日間としたことにより、それまでの期間においては低かった「サーバ上への独自のコンテンツの展開」までを行うことのできている受講生、ひいてはイベント期間後にサーバを独自に活用している受講生の割合が急激に増加したことが挙げられる。2018 年までの実施回においては、7～8 割程度の受講生がサンプルファイルをサーバ上に展開したのみで学習を終えてしまい、また受講生に配布した書籍の難易度も高いものであったことから、受講生が独自にコンテンツを制作するための土壌が整っていなかった可能性が高いものと考えられる。

また、2016 年の初回実施時には PHP を利用した Twitter Bot を制作する、という中学生にとって難解な題材を取り扱っていたところ、2017 年から HTML および CSS を利用した Web ページの制作に題材を変更し、また 2019 年においては受講生に配布した書籍の難易度を従来より引き下げ、より基礎的な書籍を配布するように変更したことにより、受講生が新しいコンテンツを制作するためのイメージ作りや、そのコンテンツを制作するための技術習得に向かっていきやすい環境を整えることができたことが結果として表出しているように思われる。

次に、2018 年度以降に設けた受講生・スタッフ共用の Slack グループチャット、また 2019 年度以降に設けた「なぜなにコーナー」において、いずれも前年より多くの質問が寄せられていることが挙げられる。このようにコミュニティを設けることにより、受講生らにとっては情報技術に関する多様な知識・技術の習得の場を獲得でき、講座の内容をベースとした自主学習を行う際だけでなく、自主的に多様な技術を学習していく際のサポートを受ける環境を整備することができるのである。なお、2017 年までの間ではメールでの質問サポートのみを実施したものの、メールベースでの技術的サポートは受講生にとって難しかったようであり、実際に寄せられた質問の件数は 2016 年および 2017 年の 2 年間で 1 件のみであった。このことから、受講生のメールによる質問サポートを利用するモチベーションは相応に低かったことが読み取れる。

また、2019 年以降の開催においては技術的な質問の場として「どのような内容でも質問ができる」という趣旨の質疑応答コーナーを設けたことにより、メールベース・Slack ベースでの質問件数の倍近くの質問が寄せられたことも変化の 1 つに挙げられる。この質疑応答コーナーは、イベントの 1 日目のアンケート内に「何らかの質問を記入した場合は翌日の講習にて解説を行う」という趣旨の自由記述欄を設け、2 日目のイベントの最後にスライドを用いたプレゼンテーション形式にて質疑応答を行うものである。導入当初の 2019 年は 12 名の受講生から 8 件の質問が、最新の 2023 年は 7 名の受講生から 13 件の質問が寄せられ、その質疑応答を通して受講生に「講座の内容に限らない」情報技術に関する知識・技術を身に付けてもらう契機とすることができたように思われる。また、過去の世代の受講生からの質問や「何らかの資格に合格した」「Web サイトを作成した」などの報告も定期的にみられるなど、受講生が盛んに Slack を活用しようとしている様子もみられている。

そして 2024 年度は、資料の製本という受講生・スタッフ双方にとって大きな意義のある取り組みを達成したほか、大きな試みとして体験型のコーナーを増やし、受講生が休み時間にさまざまな「ネットワーク技術における低レイヤの体験」を行える環境を整備した。果たして、この環境は極めて有効に活用され、今後の「インデペンデンス・サーバー・デイ」においても同

様のコーナーを設けることの意義が示されたほか、「インデペンデンス・サーバー・デイ」級の規模でなくとも、手軽に情報通信技術に触れてもらえるようなイベント・出張講座の開催に向けた足掛かりを築くことができたものとも思われる。

4. 2025 年度の「インデペンデンス・サーバー・デイ」

2025 年度も「インデペンデンス・サーバー・デイ」は実施の方向にて調整を進めている。無事に実施となった場合、2025 年度は通算 10 回目の開催となる。

成果発表会当日：2025 年 4 月 13 日時点においては、その詳細が完全に確定している状態ではないが、2025 年の開催規模については、2024 年と同様のもの（定員 10 名・応募状況により 12 名まで受け入れ・2 日間開催）となる予定である。本イベントはつくば市をはじめ、関係組織からの期待も大きく、十分な広報を行った環境の下では極めて高い応募者数の水準を維持していることから、2024 年までの「インデペンデンス・サーバー・デイ」の開催によって得られた知見：たとえば「なぜなにコーナー」の効率的な運用、スタッフ・受講生間の Slack（に限らないが）コミュニティでの関係性の醸成、サーバ・ハードディスクなどの解体コーナーをはじめとした「体験コーナー」の運用等を引き続き実施・発展させることを目標としていきたいと考えている。

また、講座の本筋はもちろんのこと、「インデペンデンス・サーバー・デイ 9」における各種の体験コーナーのような、講座から派生したさまざまな「興味を惹く」活動にも目を向けてもらえるよう、引き続き取り組みを進めていく所存である。

5. 総括と今後の課題

「インデペンデンス・サーバー・デイ」は、2025 年度以降も継続的に開催し、2016 年度～2024 年度と同様、産学間連携推進室のメンバーが各種講義・体験コーナーの企画、講習資料の作成や当日の運営に関与する予定である。「インデペンデンス・サーバー・デイ」の総括、および今後の本イベントの継続開催や、他種の発展的な ICT 教育イベントを行う上での課題について、次の通り記す。

5-1. 「インデペンデンス・サーバー・デイ」の総括

「インデペンデンス・サーバー・デイ」は、過去 9 年に亘る開催実績、およびその中で積み上げられてきた多様なノウハウの下、昨今の情報化社会において、地方自治体と地元の大学・

ベンチャー企業、そして「インデペンデンス・サーバー・デイ 9」では若年層の技術者の育成に強い興味・関心を有する学外の技術者までもが関与するという、産学官連携の理想的な形の下、高い技術・知識を持つ意欲的な中学生に対し、より高度かつ実践的な技術・知識を学べる有意義な「情報技術者教育」の場を提供することができたと考えられる。先述のように、直近回の開催においては広報面での大きな問題が生じたものの、その中でも定員を充足するだけの応募があったこと、また総じて受講者の満足度が高かったことから、大きな成功を収めたと言える。

一方で、現在の本イベントは 2 日の中にさまざまな要素を詰め込む形の開催であることから、内容の取捨選択や講義速度の調節について、引き続き緻密な計画が必要となることは間違いないことである。本イベントは先に述べた通り、今年度以降も継続して開催する予定であるが、引き続き講座で取り扱う情報量の精査・検討を行い、「受講生の理解」を第一に据えた適切な進度で実習を進められるような改善を行っていく予定である。

また、本イベントの受講希望者数、および受講生へのアンケートにおいて、前年度に引き続き、「つくば市の中学生」の実践的な IT 技術を取り扱うイベントへの参加の意欲が非常に高いことが分かった。アンケートの回答内容や、イベントの Web サイトへのインターネットコミュニティ上の反応などから、サーバ構築・運用について取り扱う本イベントの定期的な開催、また地域や年齢層などの規模を拡大しての開催だけでなく、以前産学間連携推進室が運営していた中学生向けイベントである「テキストベースのプログラミングに挑戦してみよう！」に代わり得るような一般的なプログラミング言語を用いたプログラミングの実習・アプリケーションの制作や、OS やネットワークの詳細な事項について掘り下げて取り扱うような多種の実践的なイベントを開催することが、将来活躍するであろう高度な IT 人材の育成に強く求められているのではないかと考えられる。

5-2. 他種の講座の開設について

「インデペンデンス・サーバー・デイ」、またかつて開催していた「テキストベースのプログラミングに挑戦してみよう！」の両講座における受講生向けのアンケートを通じ、情報技術に高い関心を持つ中学生から、引き続き多様な実践的技術を取り扱う講座の開催を望む声が挙がっていることが分かっている。

また、著者は中学生向けに筑波大学が実施している「夏休み自由研究お助け隊」内の講座「Arduino を使ってプログラムしてみよう」におけるメイン講師・教材作成担当を 8 年間務めているほか、「つくば SKIP アカデミー」において同系統の講座である Arduino プログラ

ミング講座を運営した経験もあること、また 2017 年頃より一般企業における技術研修を業務として行っていることなどから、高度な意欲を持つ中学生向けの ICT 教育に関する知見が豊富に蓄積されている状態にある。

2025 年度においては「インデペンデンス・サーバー・デイ」の継続開催を軸におきつつ、つくば市総合教育研究所をはじめとする各関係組織・組織内外の協力者と連携しながら、より幅広い参加者を募ることのできる他種の講座の開催・「インデペンデンス・サーバー・デイ」の内容の洗練・マテリアルの整備という 2 つの目標を設定し、本活動をより発展的なものとしていくことを検討している。

新規に開設する講座の具体的な候補としては、下記のようなものが挙げられる。

● 「情報機器資源リユースプロジェクト」によって回収された廃棄 PC を活用し、PC の分解や組み立ての実習を行い、併せて PC を構成するさまざまな部品について解説する短時間・単発の講座を開催する。短時間であることから気軽に参加できることが想定されるほか、低予算で繰り返し実施でき、親子での参加等も可能であるため、実施までのハードルが極めて低い。また、現在産学間連携推進室が学内向け（新入生向け）に行っているイベントの知見の応用が可能である。

● 「openfab 創房」において設置されている機材を用いた、デジタルなものづくりを体験できる講座を開催する。システム情報系技術室の支援を得られる状況にあるほか、「夏休み自由研究お助け隊」での内容に近い実習形式となるため、こちらも実施までのハードルが低く、なおかつ一般的な中学生の興味を惹ける内容ともなっている。また、筑波大学の設備を活用する講座となることから、大学自体の取り組みの広報にも繋げることが可能である。

● 簡易に取り扱えるマイコンボードである Arduino を活用し、ブレッドボードを用いた簡易な電子工作と組み合わせ、触って遊べるおもちゃ・ゲームを制作する。ある程度限られた予算・受講期間であっても、早押しゲームや記憶ゲームなど、受講生の記憶に残るような楽しい作品を制作することが可能である。性質上、先述の「夏休み自由研究お助け隊」と近い内容となることが想定される¹¹。

¹¹ 「夏休み自由研究お助け隊」では 2.5 時間の枠内で、Arduino を用いた LED の点灯から、スイッチと LED、ブザーを用いた記憶ゲームの制作までを取り扱っている。なお、一部に筑波大学情報科学類開設科目である「情報科学基礎実験」の内容と共通している部分があり、この点でも中学生にとっては応用的な内容を扱っていると捉えることが可能である。

● Java 言語や C# 言語などを活用し、Scratch で実装するようなキャラクターの描画を伴うプログラムを記述する。Scratch を既に体験したことのある生徒に受講してもらうことにより、Scratch など習得したプログラミングの手法が、実際の開発に用いられるプログラミング言語でも活用できることを学習してもらうことを目標とする。また、場合によっては、産学間連携推進室関係者が開発したオープンソースのゲームエンジン Altseed を活用し、簡易なゲームに拡張することも検討する。

● Raspberry Pi の GPIO ピンを活用し、各種のセンサなどの部品と連携したデバイスを制作する。Raspberry Pi の性質上、インデペンデンス・サーバー・デイで取り扱っているような単体でのサーバの構築が可能であるため、例えば「アクセスすると部屋の温度が実測値で表示されるサーバ」など、複数の要素を組み合わせた成果物を仕上げる事が可能である。ただし、内容の都合上 1-2 日での実施が難しいことが想定されるため、この内容での開催に当たっては開催日程やその対象についてよく検討する必要があるものと思われる。

5-3. 講座開催に伴う知見の共有について

「インデペンデンス・サーバー・デイ」の開催を通して得られた知見については、産学間連携推進室のイベント運営者有志により「プログラミング教育講座の舞台裏」および「教育者のための情報技術」という自費出版書籍に取りまとめを行い、定期的に日本最大の同人誌即売会「コミックマーケット」にて刊行している。その様子を図 19 に示す。



図 19：「プログラミング教育講座の舞台裏」の頒布の様子

「プログラミング教育講座の舞台裏」は、2019 年 8 月に創刊し、2025 年 4 月現在時点では Vol.06 まで、および「教育者のための情報技術」というスピンオフ書籍も Vol.02 までが出版されている。Vol.01 から Vol.06、およびスピンオフ書籍 Vol.02 までを含めた現在までの累計発行部数は、現時点で 1000 部弱に上る。「コミックマーケット」会場ではコンピュータ技術者、教育者、また情報技術教育に携わる方々によって多く購入されているようである。

6. 学会発表等に関する情報

2018 年春時点までの「インデペンデンス・サーバー・デイ」に関する総括は、2018 年 3 月 13 日 - 2018 年 3 月 15 日に早稲田大学にて開催された 情報処理学会 第 80 回全国大会にて、学生セッション 6ZF-06「サーバ構築と運用の実習を通じた中学生対象の実践的な ICT 教育イベントの試み」として口頭発表を実施した。本イベントの性質上、学会のセッションにおいてはかなり異色の口頭発表となったものの、このようなイベントの開催を高く評価する声も多く、有意義な発表であったと考えられる。

2020 年春時点までの「インデペンデンス・サーバー・デイ」に関する総括は、2020 年 3 月 14 日にオンラインにて開催された 情報処理学会「コンピュータと教育研究会」第 154 回研究発表会にて、学生セッション[3]「サーバ構築・長期運用の実習を主題とする中学生向け ICT 教育イベントの継続的实践とその成果」として口頭発表を実施し、学生奨励賞を受賞した。情報技術教育に関する研究会において、本イベントの概況、およびその成果を発表できたことは、社会的にも大きな意義のあることであると思われる。

2020 年夏以降の「インデペンデンス・サーバー・デイ」に関しては、2020 年～2022 年までの新型コロナウイルス感染拡大の影響下におけるイベント開催形態の変遷、また 2023 年度以降における本イベントの発展・拡大等も踏まえた内容として、今後の学会発表に向けた準備を行っている。

7. 謝辞

「インデペンデンス・サーバー・デイ 9」の開催に当たっては、関係各所の多くの皆様のご協力・ご支援を頂きました。「インデペンデンス・サーバー・デイ」を継続して主催して頂いております つくば市総合教育研究所 の皆様、ご後援を頂いた 筑波大学情報科学類 関係教職員の皆様、ご協力を頂いた 筑波大学 OPEN プロジェクトのご担当者の皆様、ソフトイーサ株式会社の皆様、輝日株式会社の皆様に、心より御礼を申し上げます。

また、本講座に当日のスタッフとしてご協力いただいた、産学間連携推進室のメンバーの皆様、筑波大学大学院・筑波大学の有志の皆様、そして株式会社ミラパス 小島様、BGP NAP 小坂様、慶応義塾大学 SFC 岡田様に、厚く感謝を申し上げます。

情報機器資源リユースプロジェクト 活動報告

～ 「まだ使える」機器の有効な活用を目指して ～

2025 年 4 月 13 日 筑波大学情報科学類 産学間連携推進室 活動成果報告会
情報機器資源リユースプロジェクト 第 7 代統括作業総責任者 服部 真吾
協力：産学間連携推進室 根本晃輔、関口亜聖、太田弘樹、北野尚樹、松田心杏、
細島涼雅、中野あおい、間瀬太陽

1. 序論

1-1. 本プロジェクトの背景

筑波大学 システム情報系エリアにおいては例年、毎年 2 回（6 月および 12 月）の頻度で、図 1 のような各種の大型物品¹の一斉廃棄が行われている。その中でも高度な情報機器資源であるコンピュータについては、図 2 のように、毎回の物品廃棄で約 250 台～300 台が廃棄されている。



図 1：システム情報系エリア物品一斉廃棄の様子

¹ 什器類、家電類、実験機器類、情報機器類を指す



図 2：2021 年 6 月に廃棄されたコンピュータ類

これらの廃棄されたコンピュータについては、確かに Intel Pentium4 世代や Intel Core 2 世代、あるいは第 6 世代以前の Core i シリーズの CPU を搭載したコンピュータなど、既に研究の場での現用が難しいもの²が殆どである。しかし一部には、たとえば第 7-8 世代以降の Intel Core i 世代のコンピュータなど、パーツを交換・増強すれば現用に耐えうる性能でありながら、単純な経年・入替のため廃棄されたものや、修理・交換可能な部品の一部のみが故障したために廃棄されている比較的新しいコンピュータも存在している³。それらは部品を適切に交換・修理したり、性能向上を図れるパーツを追加したりすることで、ある程度の期間、再度利活用することが可能になる。

このように、故障したコンピュータを積極的に買い換え、故障したコンピュータを廃棄することのできる研究室がある一方で、筑波大学における一部の研究科・専攻においては、高価な実験機材を購入するためにコンピュータに十分な予算を設けることができず、やむ

² 既に製造から 12 年以上、Core 2 世代では約 20 年が経過しており、近年のソフトウェアの要求性能に満たない性能の機器が多いこと、および長期の使用による修復困難な故障などが発生しているものが多いことから、現用が難しいものと考えられる

³ 中にはマザーボードの致命的な故障など重大なものも含まれているが、OS レベルの障害や一部の交換可能なパーツの故障などが発生した際に、その原因を詳細に特定することなく廃棄されているものも少なくないと推測される

を得ず Intel Core 2 Duo ～ 初代 Core i シリーズ 世代などの古いコンピュータを限られた台数で利用している研究室が存在している。また、研究室における急激な人数の増加に対し、予算が限られているなどの理由によりコンピュータの新規導入が難しいという状況や、留学生のために短期的に利用できるコンピュータが必要になるという状況にある事例もみられるほか、2020～2021 年頃には在宅での勉学・研究を行うために貸出用の PC が必要となった研究室なども存在した。

筑波大学の物品廃棄において廃棄される物品のうち、コンピュータ以外の一般の備品については、教職員が閲覧可能なシステム上で次の利用者を募集するなどのリユースの仕組みが構築され、実際に運用されている。

しかし、性能や経年の状況が外観から判断できず、限られた情報からでは詳細な状況を判断しづらいコンピュータについては、その仕組みの枠の中で取り扱うことが難しい状況にある。更に、コンピュータを廃棄・譲渡する際にはデータの厳重な消去が必要になるが、その方法については教職員間でも徹底がなされておらず、そのような徹底が不十分な状況下で安易な廃棄・譲渡を行うことは、時として情報漏洩などの重大な事故を招きうるものとなる。

1－2．本プロジェクトの目的

このような諸課題を解決するための情報機器資源のリユースの試み、および付随的な目的として情報記録媒体の確実な廃棄の補助・物品一斉廃棄の支援を行う本プロジェクトは、システム情報系技術室の要請の下、2015 年 6 月に根本（産学間連携推進室所属）らにより発足した。そして、2025 年 4 月現在に至るまで、規模を随時拡大しながら運用が続けられている。

本プロジェクトは、筑波大学システム情報系技術室の指揮・管理の下で、物品の一斉廃棄の際に廃棄されるコンピュータを適切に修理・再構成し、システム情報系技術室によって上述のようにコンピュータを必要としている研究室や各種組織などに分配することで、情報機器資源のリユースを行うことを目的としている。またリユースの過程の中に、廃棄されるべき情報記録媒体の確実な破壊・廃棄、および再利用される情報記録媒体におけるデータの確実な削除を組み込むことで、物品廃棄に際し発生しうる情報漏洩などのインシデントを未然に防ぐことも付随的な目的としている。

2. プロジェクトの概要

2-1. 活動の概略

「情報機器資源リユースプロジェクト」は、システム情報系技術室の要請・協力の下、情報科学類産学間連携推進室内に設置されているプロジェクトである。本プロジェクトにおいて、産学間連携推進室のメンバーはコンピュータの修理・再構成などの技術的部分を主に扱っているため、本稿ではその担当範囲における活動について述べる。

本プロジェクトは2015年6月（発足時）～2024年12月度の全19回の物品廃棄において、デスクトップパソコン・ディスプレイ・ノートパソコンを併せ、総計850台以上のリユースを行った。集計が完了している2021年6月度までの内訳を表1に示す。

表1：リユースの内訳。2020/06 は新型コロナウイルス感染拡大の影響により中止。

Data	Desktop	Laptop	Display	Sum
15/06	15	10	15	40
15/12	15	0	15	30
16/06	23	0	20	43
16/12	16	2	34	52
17/06	32	21	39	92
17/12	12	2	5	19
18/06	30	20	10	60
18/12	28	15	9	52
19/06	35	5	12	52
19/12	17	2	11	30
20/06	-	-	-	-
20/12	25	5	14	44
21/06	16	5	17	38

本プロジェクトは現在、2025年6月の物品一斉廃棄においても同様の作業を進めており、2025年7～8月中には再利用可能となったデスクトップパソコン・ディスプレイ・ノートパソコンをシステム情報系技術室に引き渡す予定である。

2-2. 工程と運用

本プロジェクトにおけるコンピュータの修理・再構成の工程においては、部品が欠損・故障しているコンピュータから性能の高い再利用可能な部品（CPU・メモリ・グラフィックカード等）を取り出し、状態の比較的良好なコンピュータにそれらの部品を移植することで、修理・性能向上を行う形をとっている。また、液晶ディスプレイなどの表示機器についても、動作確認を行った上で、ある程度の解像度があるものや経年していないものについては形式の取りまとめを行い、機器のリストと合わせ技術室に引き渡す形をとっている⁴。

現在の本組織の運用形態は、産学間連携推進室メンバーの 9 名および情報科学類・情報メディア創成学類・大学院生の有志学生 4 名からなる本プロジェクトのように少人数な組織であっても、十分に運用可能なものである。なお、活動の様子を記録した写真を図 3・図 4 に示す。図 3 はディスプレイ等各種機器の状態確認、図 4 は専用に構築した機器を用いてハードディスクのデータ消去を行っている様子である。



図 3：ディスプレイ等各機器の動作確認を実施している様子

⁴ なお、2017 年 12 月以降の表中の「ディスプレイ」の数値は高解像度のものや、形式の取りまとめを行っているものに限っている。高解像度でないもの、あるいは若干の経年があるものの利用が可能であるディスプレイについてもリユースを行っているが、作業の簡易化のために取りまとめを行っていないため、この表中には記載していない。



図4：ハードディスク処理作業の様子

3. リユースの事例

本プロジェクトの活動実績を示す上での参考として、実際にこれまでのプロジェクトの進行の中で行ってきたコンピュータの修理・再構成の例の一部を示す。

3-1. ノートパソコンの液晶画面の入れ替え

搬出された同型のノートパソコン 2 台のうち、1 台は液晶ディスプレイ部分が破損していたものの筐体をはじめとする他部分には損傷がなく、もう 1 台はタッチパッド・DVDドライブ部をはじめとする筐体部に大きな歪みがあったものの、液晶ディスプレイ部分には損傷がない状態であった。

そこで、後者のノートパソコンから液晶ディスプレイ部を取り外し、前者のノートパソコンに移植することで液晶ディスプレイの交換作業を行い、無事に利用できる状態に回復させた。なお、後者のパソコンに搭載されていた取り外し可能な他のパーツのうち、メモリとハードディスクについては取り外しを行い、他のノートパソコンへの移植を行った。

3－2．エアダスターや無水エタノールなどを用いた分解清掃

搬出されたデスクトップパソコン・ノートパソコンのいずれについても、基板上・ファン内部などの各部位に蓄積した埃などについては、エアダスターや無水エタノールを用いた清掃を行っている。また、ノートパソコンについては消毒用エタノールを用いた清掃を行い、次の利用者が快適に利用できるような環境を整えている。

特に、デスクトップ Mac 環境などの特殊な環境の動作を維持するためのパーツは絶対数が少なく、供給の少ない部品を清浄しながら利用する必要がある。中には煙草の脂と思われるものが付着しているものもあったが、この分解清掃を行うことで外見・動作とも問題のない状況に仕上げる事が出来た。

3－3．パーツ世代・電力量などを加味した各種パーツ等の増設や高性能化

搬出されたデスクトップパソコンについては、例えば CPU をより上位のものにアップグレードできるものや、更に大容量のメモリを搭載できるものなどが存在する。それらのパソコンについては、アップグレード可能なパーツがあれば、電力量や想定用途などのさまざまな要素を加味した上で、再利用の際に支障が出ない範囲でのアップグレードを施している。

4．今後の展望と課題

本プロジェクトにおける今後の活動の展望、および現状の課題に関して以下の通り記す。

4－1．継続的な活動に向けて

本プロジェクトは 2025 年度の物品廃棄においてもコンピュータの修理・再構成を実施しており、今後も活動を継続していく予定である。特に 2022-2024 年度の活動においては、特殊工具を含めた資材の大幅な拡充等を行ったことにより、前年度に引き続き作業効率や作業速度を向上させること、および作業手順のマニュアル化を進めることで作業の属人化・負担の集中を防ぎ、今後も継続して活動ができるプロジェクト運営を行うこと、の 2 点を重点におく形で活動を進めることができた。本プロジェクトは創設より 5 代に亘り代替わりを続けながら活動を継続しているが、大学組織に特有の「メンバーの入れ替わり」があっても持続可能な、なおかつ人々に必要とされる組織を構築していく上では、本年度の成果には非常に大きな意義があるといえる。

4－2．他エリアへの拡大に向けて

筑波大学においては、システム情報エリアに限らず、さまざまな学域・エリアにおいて物品の一斉廃棄と、それに伴うコンピュータ類の搬出が行われている。それらのコンピュータ類にも、システム情報系物品廃棄において搬出されているコンピュータ類と同様、まだ十分に利用可能な製品や、修理および再構成を行えば数年間は利用可能な製品が含まれており、これらをシステム情報系の物品廃棄における本プロジェクトと同様の形でリユースすることができれば、当該エリアにおける情報機器資源の効率的な運用や、比較的高性能なコンピュータを活用した研究の推進に役立つことが考えられる。またもちろん、情報技術に明るくない学域・エリアにおいては、本プロジェクトの副次的な効果である HDD/SSD などの記録媒体の安全な処理を推進していくことで、より安全に情報記録媒体の廃棄を進めていくこともできるだろう。

エリアを跨いだ活動には現状下で困難が多いと思われるが、このような活動をより広範な形で推進していくことにより、大学内における経年した旧型機器の入れ替えの推進による研究活動の効率化や、昨今の大学・各種研究機関において非常に大きな問題となっている予算の逼迫の解決、廃棄機材からの情報漏洩への対策に対する一つのアプローチとなるのではないかと思われる。

4－3．知見の一般共有

本プロジェクトの副次的な効果として、コンピュータの組み立てやトラブルシューティング、およびそれらに関連する知見が豊富に得られることが挙げられる。例えば、以下のようない知見を得ることができる。

- PC パーツの世代の見分け方や、同一世代内での性能の素早い見分け方⁵
- PC ケース内の整線や、カスタマイズの行いやすいパーツの配置方法
- CPU やメモリのトレイなど、余剰パーツを収納するための特殊な部材の入手先⁶

このような一般的な知見について、プロジェクト内部のメンバーだけでなく、Web サイトなどを活用して外部にも積極的に公開していくことは、本プロジェクトそのものの広

⁵ 中には外見上非常に見分けづらいパーツが存在するが、型式などが記載されている部分を探せばすぐに世代を特定することのできるパーツ（メモリなど）も存在する。

⁶ たとえば、CPU トレイは国内に取扱いがなく、需要がニッチなため ebay などによる海外通販を利用するしかない。しかし、その海外通販もあまり知られていない。

報・発信の観点からも望ましいことである。このような情報の共有は 2024 年度までにあまり力を入れて実施できなかったことでもあるため、2025 年の活動においては特に積極的に推進されるべきことであると考えられる。

それだけでなく、例えば、コンピュータの組み立てを行ったことがないが、それに興味のある小～中学生を対象に、コンピュータのパーツの役割を説明しながら実際に組み立てていく講座を開設するなど、ワークショップ形式で情報発信を実施していくことも選択肢の 1 つとなるだろう⁷。

4－4．廃棄機材の他プロジェクトにおける活用

現在試験的に実施している事柄として、情報記録媒体について安全な処理を施した状態の廃棄機材を「インデペンデンス・サーバー・デイ」における「サーバ解体コーナー」、産学間連携推進室の主催する情報科学類生向け PC 分解講座・ネットワーク講座等に活用する取り組みが挙げられる。このような取り組みは、物理的にコンピュータという機器の内部に触れる機会が減ってきている近年の生徒ら・学生らにとって、またとない学習機会となるであろう。

4－5．今後の活動に向けての課題

現在、この活動は特に PC ハードウェアに関する知識を有する学生によって成り立っている。しかし、昨今のノート PC の高性能化やタブレット型 PC の出現などから、デスクトップ PC を組み立てたことのある学生の割合は年々減少しつつあるように思われる。しかし、研究の場で用いられるコンピュータは、依然としてデスクトップ PC が殆どであろう。

そのような状況下においても滞りなくこの活動を継続させるために、デスクトップ PC の組み立てなどについて興味のある学生へのアプローチを中心に、持続可能なプロジェクト運営に向けての方策の構築に、今後十分に時間をかけて取り組まなければならないと考えられる。2024 年度も 1 名の新入生・1 名の学生を本プロジェクトに迎え入れることができたため、 신입生に対する技術指導を中心として、今後の活動の持続可能性を高められるような取り組みを進めていくことが求められるものと思われる。

⁷ 「夏休み自由研究お助け隊」や「インデペンデンス・サーバー・デイ」などとの協調実施も視野に入らう。

5. 謝辞

本プロジェクトを進行するにあたり、システム情報系技術室 雨谷 恵 様をはじめとする、筑波大学内のさまざまな方々、プロジェクトメンバーの皆様、プロジェクト OB/OG の皆様のご支援・ご協力を頂いております。この場をお借りしまして、深く御礼を申し上げます。

2024 年度 筑波大学 情報学群 情報科学類

産学間連携推進室 活動成果報告書

発行日 2025 年 4 月 13 日 初版発行

発行者 細島涼雅（情報科学類 産学間連携推進室 学生代表）

編集者 北野尚樹

製作者 情報科学類 産学間連携推進室所属学生