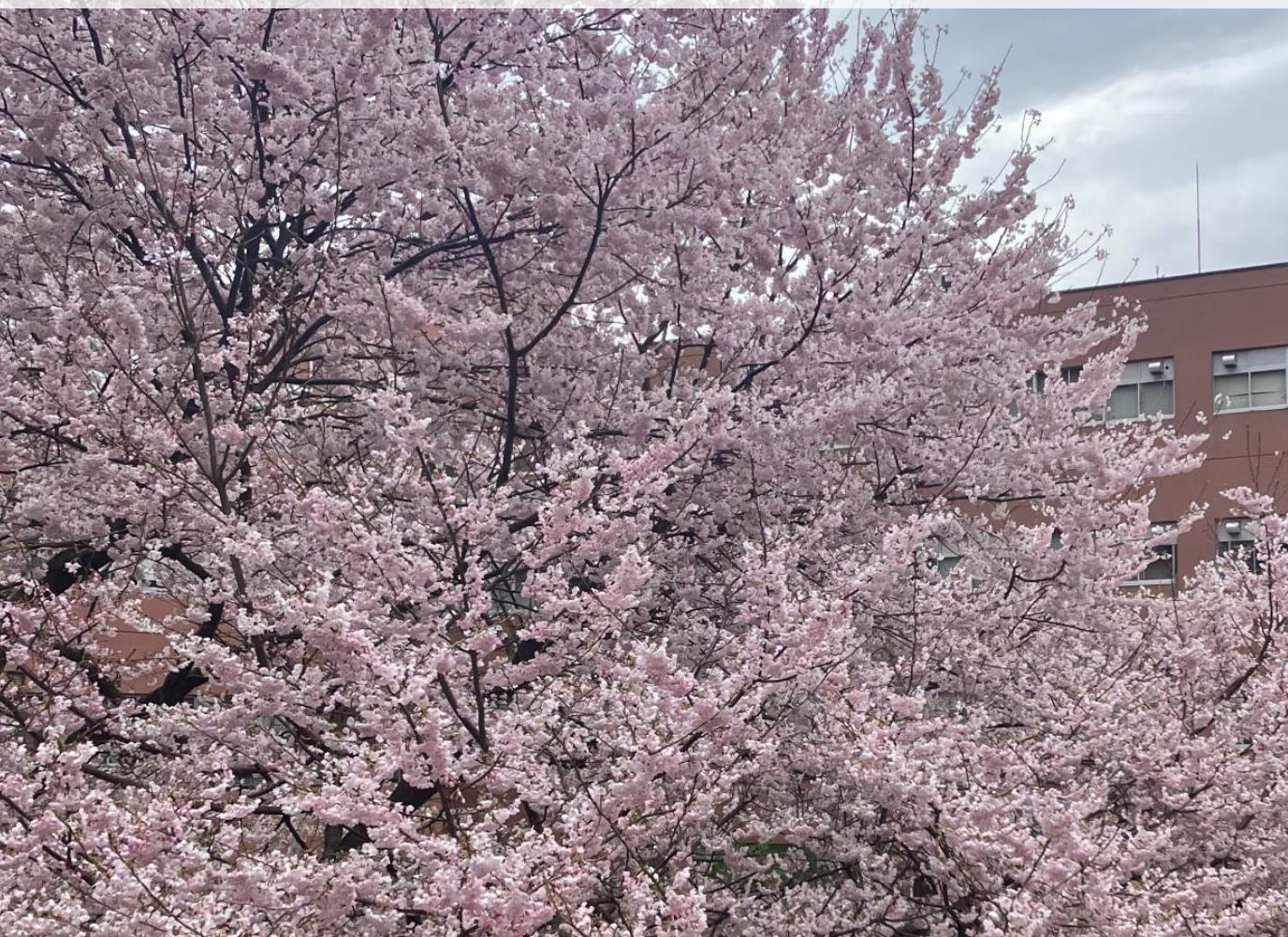


WORD

From College of Information Science



2023 入学祝い号

目次

Heaven's Gate 入門	mopisec	1
トキポナ入門の入門	Azumabashi	5
つくばで大病を患つたら	puripuri2100	12
OpenSCAD で 3D モデルをコードっぽく書く	yuseiito	27
ポスト西友時代を生きる	なかやばし	38
怪しい団体の勧誘に巻き込まれた話	$Q=0JyqLCq2c4)NmX-B8V$	41
Linux 利用のすすめ	motorailgun	44
情報技術関連課外活動のすゝめ	Crow314	47

Wordian になろう

新入生の皆様、ご入学おめでとうございます。

WORD は「情報科学」から「カレーの作り方」までを手広くカバーする総合的な学類誌です。年に数回発行しており、主に第三エリア 3A、3C 棟などで無料で配布しています。学内で見かけたら、ぜひ最新刊を手に取ってご覧ください。バッナンバーも、ウェブサイト (<https://www.word-ac.net>) 上にアップロードされています。

そして、そんな WORD を発行しているのが、情報科学類公認組織である我々「WORD 編集部」です。メンバー (wordian と呼んでいます) は、情報科学類生に限らず、数学類生、比較文化学類生、大学院生なども在籍しています。

編集部の拘束時間には、年に数回の赤入れや製本作業等発行に伴う作業があります。コロナ禍以前は週一回「編集会議」が行われていたようですが、今年度どうなるかは未定です。参考までに、昨年度は赤入れ以外にそういった会議は行われませんでした。いずれにしても、日常的に活動する必要はありませんので、他サークルの掛け持ちの障壁にはなりませんが、それでも先輩や同期と繋がりをもち、大学生活を充実させる環境として活用いただけること請け合いです。

あなたも編集部員として、WORD に記事を書きませんか？ 少しでも WORD に興味を持った方は是非、情報科学類学生ラウンジ隣の編集部室（3C212）へいつでも見学に来てください。時間を問わずに開いています（人がいない時間帯もありますので、事前に連絡いただけとより確実に案内ができます）。

質問がある方は、word@coins.tsukuba.ac.jp か、Twitter アカウントをお持ちの方は@word_tsukuba までお気軽にお問い合わせください。

文責 編集長 伊藤祐聖

Heaven's Gate 入門

文 編集部 mopisec

1 はじめに

皆さんこんにちは。この記事では 64 ビット版の Windowsにおいて 32 ビットアプリケーションから 64 ビットコードを実行する Heaven's Gate というテクニックの基本について解説するとともに、Heaven's Gate を使用しているアプリケーションをリバースエンジニアリングする方法を紹介していきます。

2 Heaven's Gate とは

64 ビット版の Windows には 32 ビットアプリケーションを実行するために、WOW64^{*1} というエミュレーションレイヤーサブシステムが実装されています。これを利用して、64 ビット版の Windows では 32 ビットアプリケーションが WOW64 上で互換モードで実行されるようになっています。しかし、互換モードでは 64 ビット版の Windows のネイティブ API を実行することができないため、ネイティブ API が 32 ビットアプリケーションから呼び出された場合、パラメータの変換等を経た上で 64 ビットモードで処理を実行し、処理が終わり次第互換モードに復帰するように実装されています。このように 32 ビットアプリケーションの実行中に互換モードから 64 ビットモードに切り替えて、64 ビットコードを実行するテクニックを Heaven's Gate と呼びます^{*2}。

このテクニックに関する技術的な詳細については、George Nicolaou 氏の書かれた “Knockin' on Heaven's Gate – Dynamic Processor Mode Switching” という記事^{*3}をご確認ください。簡潔にまとめると、コードセグメントセレクタ 0x0033 を指定して FAR JUMP / FAR CALL / RETF 等を実行することで互換モードから 64 ビットモードへ切り替えることができるというテクニックとなっています^{*4}。

3 マルウェアによる悪用

現在の Windows が 32 ビットアプリケーションに対する互換性を維持する上で大きな役割を担っている Heaven's Gate ですが、マルウェアが解析妨害のために本テクニックを悪用している事例が確認されています。これは多くの逆アセンブラーは 32 ビットアプリケーションを読み込んだ場合、もちろん 32 ビットコードとして逆アセンブルするので、64 ビット

*1 Windows-on-Windows64。WoW64 や Wow64 と表記される場合もあります。

*2 Heaven's Gate はネイティブ API が呼び出されたタイミング以外でも使用することができます。

*3 <http://rce.co/knockin-on-heavens-gate-dynamic-processor-mode-switching/>

*4 コードセグメントセレクタ 0x0023 を指定することで 64 ビットモードから互換モードに切り替えることもできます。

コードが紛れていた場合、正しい逆アセンブル結果が得られない為です。

以下の画像は dadas190 氏が GitHub 上で公開している Heaven's Gate のサンプルプログラム^{*5}を IDA Pro^{*6} のバージョン 8.2sp1 を用いて逆アセンブルした結果です。RETF 命令以降が正しく逆アセンブルできていないことが読み取れます^{*7}。

```
.data:00403000 loc_403000:          ; DATA XREF: _X64Call+62↑o
.idata:00403000      push    33h ; '3'
.idata:00403002      push    11223344h
.idata:00403007      retf   ; retfによるHeaven's Gate
.idata:00403008      ; -----
.idata:00403008      push    esi   ; ここから64ビットコード(正しく逆アセンブルできていない)
.idata:00403009      push    ebx
.idata:0040300A      dec     eax
.idata:0040300B      mov    ebx, esp
.idata:0040300D      dec     eax
.idata:0040300E      and    ebx, 0Fh
```

図 1 Heaven's Gate が使用されている 32 ビットアプリケーションの逆アセンブル結果

4 静的解析

前節にて解説した通り、多くの逆アセンブラーは 32 ビットアプリケーション内の 64 ビットコードを正しく逆アセンブルすることができません。では、Heaven's Gate が使用されているアプリケーションはどのように静的解析^{*8}すれば良いのでしょうか。最も簡単な方法は 64 ビットコードと思われる領域を実行ファイルから抽出して、別途 64 ビットコードとして逆アセンブルするというものです。

実際に前節と同様のサンプルプログラムから 64 ビットコードを抽出して静的解析できる状態にしてみましょう。

まずは 64 ビットコードの開始地点と終了地点を把握しましょう。開始地点のアドレスは 64 ビットモードに切り替えているコードの直後であるため、0x403008 です。また、互換モードに切り替えるための RETFQ 命令 (0xCB) が存在するため、終了地点のアドレスは 0x403078 です。よって、0x403000 から 0x403078 までの範囲が 64 ビットコードであると考えられます。

次に 64 ビットコードを抽出する必要があります。IDA Pro を使用している場合、逆アセンブル画面 (IDA View) 上で 64 ビットコードの領域を選択して、“Edit” - “Export data” (Shift+E) を選択して raw bytes としてファイルに出力しましょう。お使いの逆アセンブラーに同等の機能がない場合、開始地点および終了地点のオフセットをもとに 64 ビットコードを抽出するプログラムを作成しても良いでしょう。

*5 <https://github.com/dadas190/Heavens-Gate-2.0/>

*6 <https://hex-rays.com/ida-pro/>

*7 EAX を不自然にデクリメントしている命令の有無などをもとに、64 ビットコードかそうでないか推測することができます。

*8 ソフトウェアリバースエンジニアリングにおいて逆アセンブラーや逆コンパイラによって生成されたコードをもとにアプリケーションの処理内容を読み解く解析手法のこと。

```

.data:00403070 6A 23          push  23h ; '#'
.data:00403072 68 44 33 22 11    push  11223344h
.data:00403077 48          dec   eax
.data:00403078 CB          retf
.data:0040307E

; retfq <- 32ビットモードに切り替えている
; (64ビットコードはここまで)

```

図 2 互換モードに切り替えているコード

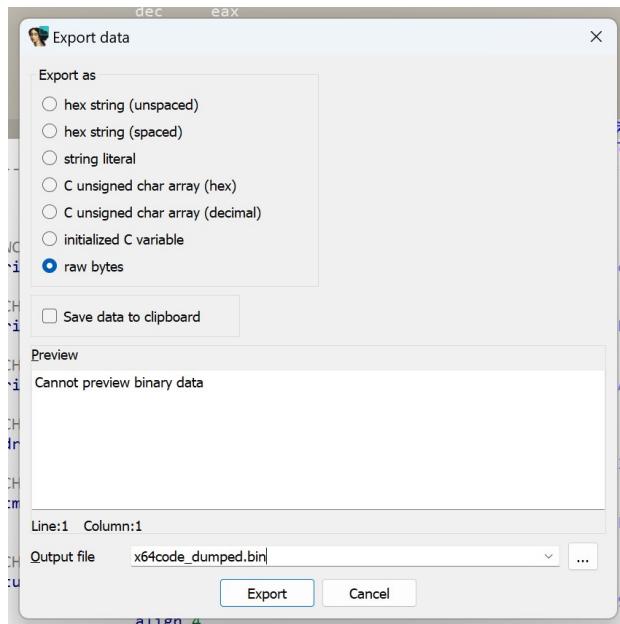


図 3 64 ビットコードを抽出する

あとは出力されたファイルをバイナリファイルとして逆アセンブラーに読み込み、64 ビットコードとして逆アセンブルするよう設定すれば、静的解析を進めることができます*9。

```

seg000:0000000000000000 ; Segment type: Pure code
seg000:0000000000000000 seg000 segment byte public 'CODE' use64
seg000:0000000000000000 assume cs:seg000
seg000:0000000000000000 assume es:nothing, ss:nothing, ds:nothing, fs:nothing, gs:nothing
seg000:0000000000000000 push rsi
seg000:0000000000000001 push rbx
seg000:0000000000000002 mov rbx, rsp
seg000:0000000000000005 and rbx, 0Fh
seg000:0000000000000009 add rbx, 38h ; '8'
seg000:000000000000000D sub rsp, rbx
seg000:0000000000000010 mov rcx, 39h ; '9'
seg000:0000000000000017 mov rsi, 11223344h

```

図 4 64 ビットコードとして逆アセンブルした結果

*9 シェルコードを解析する際と同様のステップを踏むことで静的解析が可能です。

5 おわりに

今回の記事では Heaven's Gate というテクニックの基本および静的解析の方法について紹介してきましたが、いかがでしたでしょうか？ こういったテクニックに付けられる名前にはカッコイイものが多いのですが、その中でも Heaven's Gate は筆者イチオシだったので紹介しました。マルウェア解析やリバースエンジニアリングをする上では、こういったテクニックについてどれだけ知っているかどうかが効率を大きく左右します。興味を持たれた方は、他にも面白いテクニックがたくさんあるので、ご自身で色々調べてみてください。

トキポナ入門の入門

文 編集部 Azumabashi

1 はじめに

1.1 *toki!*

toki! mi jan Asumapasi. mi ken toki kepeken toki pona. sina toki ala toki kepeken toki pona? toki pona li toki pona. sina ken kama toki kepeken toki pona!

この文章は、トキポナという人工言語で書かれた文章です。試しに日本語に訳してみましょう。

こんにちは。Azumabashiです。私はトキポナを話せます。あなたはトキポナを使って話せますか？トキポナは良い言語です。あなたもトキポナを使って話せるようになるでしょう。

この記事では、トキポナに入門してみます。といっても文法のほんの初歩的なところだけをやります。

なお、プログラミング言語も広義の人工言語と言えるかもしれません。しかし、ふつう「人工言語」といえば自然言語に似た言語のことを指すので、以下ではプログラミング言語については考えません。このため、本稿で「言語」と言った場合には自然言語を指します。

また、トキポナの単語を表すためには、*lipu pu* にならってイタリック体にします。すなわち、*a* はトキポナの単語で、*a* はトキポナの単語でない単語です。

適宜練習問題をつけています。練習問題の解答は本稿の末尾に用意されています。

1.2 なぜトキポナ？

人工言語で最も有名なものと言えば、エスペラントでしょう。そのほかにもロジバンなどもあります。では、なぜトキポナなのでしょうか。

トキポナには、学習するにあたって次のような利点があります。

- 基礎的な単語が 120 語くらいしかない。
- 文法が比較的単純であり、性や格といった概念がない。

まとめると、トキポナは簡単な言語である、ということになります。では、なぜいま簡単な言語を学習するべきなのでしょうか。

そもそも自然言語の表現能力は、人間が使うには豊かすぎるのではないかでしょう。せっかく自然言語に豊かな表現能力があっても、それを使う人間が賢くなれば意味がないど

ころか、有害ですらあります。某青い鳥の SNS で行われている水掛け論・罵倒・中傷・煽りなどなど、有害であるという事例を挙げればきりがありません^{*1}。それでは人間性や道徳といったものを涵養すればこういったことはなくなるのでしょうか。仮にこう考へても、そもそも人間性や道徳を涵養する方法とは何かが全く自明ではありません。そもそもそういった方法があるのかどうかもわかりません。あつたとしても結局は「気をつけようね」に帰着されそうな気もします。「気をつけようね」で済むなら型システムやテスト駆動開発といった手法はいらないはずですが、実際はそうではないことを考へると、どうもこれではうまくいかなさそうです。

それでは、人間性や道徳に頼らずにまともなコミュニケーションを取るためにはどうすればよいでしょうか。どうやら使う言語ごと乗り換える必要がありそうです。負荷なく乗り換えるためには、学習コストができるだけ低くなければなりません。言い換えれば、乗り換える先の言語は簡単な言語でなければなりません。ただし、簡単にするとといっても、いたずらに表現能力を奪ってしまってはよくありません。ある程度の事柄は表現できてくれなければ困ります。トキポナは、ある程度の表現能力を備えつつ、既存の言語に比べて学習コストが低く抑えられています。すなわち、我々はまともにコミュニケーションを取るためにトキポナを学習しなければならないのです。

1.3 「トキポナ語」という表記について

自然言語のアノロジーで、「トキポナ語」と言いたくなるかもしれません。しかし、トキポナとは *toki pona* であり、*toki* に言語という意味が含まれるので、「トキポナ語」という言い方は厳密には不正確です。「トキポナ」と言いましょう。

2 発音とアルファベット

語学書というのはふつう発音の項から始まりますが、外国語の発音を習うのはしばしば困難を伴います。英語ですら l と r を正しく発音できないのですから、それ以上に発音規則が複雑な言語（フランス語など）で苦労するのは当然といえば当然です。

トキポナにはこのような心配はありません。なぜなら、1 文字を除いてローマ字読みすればよいからです。その 1 文字というのが「j」です。*j* はヤ行で読みます。例えば、*jan* という語の読み方をカタカナで示すと「ヤン」になります。同様に、*lipu* は「リプ」です。簡単ですね。

なお、トキポナで使えるアルファベットは次のものに限られています。英語よりも少ないです。

- 母音 : a, e, i, o, u
- 子音 : j, k, l, m, n, p, s, t, w

これらの単語に収まらない単語は、適当に近い発音のものに置き換えます。例えば、*Nihon*

^{*1}これらは某青い鳥の SNS に特有の現象という訳ではないですが、わかりやすさのためにこのようにしています。

は *Nijon* になります。 *h* がないので発音の近い *j* の音になる、という訳です。

練習問題 1 本稿冒頭に掲載したトキポナの文章を発音せよ。

3 最初の一語：*mu*

それでは早速トキポナの文法を見ていきましょう……としたいところですが、その前にトキポナの単語の特徴を浮き彫りにしてみましょう。情報関係のエンジニアの鳴き声は「にゃーん」であることが知られています。この「にゃーん」はトキポナでは何と言うのかというと、「*mu*」です。簡単ですね。これから「にゃーん」と言いたいときには *mu* と言いましょう。文にしたいときは「*mu.*」とピリオドを打てばよいです。

早速ですが例題です。

例題 1 次の文をトキポナ訳せよ。

1. わんわん。 (犬の鳴き声)
2. がおー。 (トラの鳴き声)
3. かーかー。 (カラスの鳴き声)

そんなの知らないよ、という方はトキポナ力が足りません。解答は次のようにになります。

1. *mu.*
2. *mu.*
3. *mu.*

「にゃーん」の訳語として *mu* を紹介しました。確かにこれは正しいのですが、このように書くとミスリーディングになります*2。正しくは *mu* は動物の鳴き声全般を指します。つまり猫が鳴こうがオオカミが吠えようが全部 *mu* です。このような感じで、トキポナの単語の意味は極めて大雑把であるところに特徴があります。

ちなみに、トキポナでは固有名詞でもない限り、文頭でも単語の頭を大文字にしません。したがって、「*mu.*」で正しい文章になります。

4 修飾方法

言葉を修飾するためにはどうすればよいでしょうか。例えば、「小さな動物」を表現したいとします。「小さな」は *lili*、「動物」は *soweli* ですので、*soweli lili* となります。形容詞が複数連なる場合も同様です。「小さな黒い動物」は、「黒い」が *pimeja* なので、*soweli lili pimeja* です。このとき、*lili* および *pimeja* はいずれも *soweli* を修飾しているものと解釈されます。逆に言えば、修飾語を被修飾語の後ろにつらつらと連ねればよいです。

修飾の順番は *pi* で変更できますが、いい例が思いつかなかったので省略します。

*2 「A は B だ」という言明から「非 A は B でない」すなわち「A と B は同値だ」という印象を受けがち。

5 「A は B である」の表現

それでは早速「A は B である」の形の文章を作ってみましょう。構文は単純で、「A *li* B」です。ただし、A が *mi* 「私」または *sina* 「あなた」の場合は *li* が省略され、「A B」になります。

例題を見てみましょう。

例題 2 次の文をトキポナ訳せよ。

1. 人間は最悪だ.
2. 子猫はかわいい.
3. わたしはねこです.

順番に解答を見ていきましょう。

1. 人間は *jan*, 「最悪だ」は「非常に悪い」ということなので, *ike* 「悪い」と *mute* 「とても」で *ike mute* です。したがって, 「*jan li ike mute.*」が正解です。
2. 「子猫」は「小さな動物」なので, *soweli lili* です。「かわいい」は「良い」ということなので, *pona* 「良い」です。したがって, 「*soweli lili li pona.*」が正解です。
3. 「私」は *mi*, 「猫」は *soweli* なので, 「*mi soweli.*」が正解です。 *mi* が「主語」なので, *li* が省略されることに注意してください。

「子猫」 = 「小動物」 = 「*soweli lili*」, というような言い換えに注意してください。こんな感じでトキポナの文章は構成されています。簡単ですね。

練習問題 2 次の文章をトキポナ訳せよ。

1. 私は伊藤と申します.
2. 日本語は悪い.
3. 酒は良い.
4. 死は救済である.

[hint] *telo* 「液体」, *nasa* 「狂う」, *moli* 「死」

6 「A が V をする」の表現

次に、「A が V をする」の表現方法を見ていきます。基本的な構文は、「A *li* V *e* B」で「A が B に V する」です。例によって A が *mi*, *sina* の場合は *li* が省略されます。B は英語で言うところの目的語です。目的語を伴わない場合は *e* B が省略されます。

例題 3 次の文章をトキポナ訳せよ。

1. 猫は果物を食べる.
2. 彼らは本（複数形）をもっている.

3. 私はフランス語を話す.

解答です.

1. 「猫」は *soweli*, 「果物」は *kili*, 「食べる」は *moku* ですから, 「*soweli li moku kili.*」が正解です.
2. 「彼ら」は *ona*, 「本」は *lipu*, 「持っている」は *jo* ですから, 「*ona li jo lipu.*」が正解です. トキポナでは单複の概念がないので, これで正解です.
3. やや引っ掛けです. 「～語を話す」は *toki kepeken toki ~*というコロケーションで確立しています. また, 「フランス語」は *toki Kanse* ですから, 答えは「*mi toki kepeken toki Kanse.*」です.

練習問題 3 次の文章をトキポナ訳せよ.

1. 人々は歩く.
2. 私はトキポナを学習している.
3. 彼は本を読む.

[hint] *tawa* 「動く」 *kama* 「～になる」 *sona* 「知っている」 *lukin* 「(何かを) 見る」

7 時制

せっかくなので, トキポナで書かれた文章の和訳をやってみましょう.

例題 4 次の文章を和訳せよ.

- *sina moli.*

解答例のほんの一部を以下に示します.

- あなたは死ぬ.
- あなたは死んだ.
- あなたは死んでいる.

実は, トキポナには時制の概念がありません. 現在でも未来でも過去でも同じ文章になります. したがって, 英語ほど訳語が厳密に確定する訳ではありません.

練習問題 4 練習問題 2 と練習問題 3 で訳した文章を, 日本語に「再翻訳」せよ.

8 更なる学習のために

8.1 もっとトキポナの文法をやろう

ここまで知識で, 様々な文章をトキポナで表現できたことになります. しかし, まだまだトキポナの文法でやるべきことが残っています. 例えば, 次の例題を考えてみます.

例題 5 次の文章をトキポナ訳せよ.

1. 誰が狩りをしていますか? ——田中さんです.
2. あなたは英語を話しますか? ——いいえ.
3. 私は寝たいです.

こういった文章をトキポナで表現するためには、トキポナの文法をもう少しやる必要があります。

おすすめなのは、*pu*^{*3} をやることです。単語の意味はやや古いらしいのですが、ひととおりの文法を学習できます。また練習問題とその解答もついています。日本語版はないので英語版を読むしかないですが、言語学などの専門的な語彙は使われていないので読みやすいと思います。

有志がまとめた文法もあります。例えば *jan Nanko* のものがあります^{*4}。

8.2 トキポナ縛り

lipu Tuwita のユーザの一部では *toki kepeken toki pona* を突き詰めすぎて「トキポナだけで話す」といったこともやられているらしいです。俗にいう「トキポナ縛り」です。トキポナの学習にいいらしいですが、私はやったことないです。求む挑戦者。

8.3 トキポナと形式言語

ちなみにですが、トキポナは正則言語（正規言語）でないことが知られています。この証明は *jan Nima* によってまとめられています^{*5}。形式言語については深く触れないでの、「オートマトンと形式言語」を取ろう！

9 おわりに

トキポナで話すことはよいことであるということがおわかりいただけたでしょうか。トキポナはいい言語です。皆がトキポナで話さなければなりません。まずは平砂宿舎の *soweli* にトキポナを話させることにしましょう。

toki pona li pona. soweli li toki kepeken toki pona. jan li toki kepeken toki pona. sina wile kama sona e toki pona anu seme? lon! o kama pu!

練習問題 5 上記のトキポナの文章を日本語に訳せ。ただし、本稿で使われていない文法事項が用いられているので、適宜 *pu*などを参照すること。

*3 *ku* (official toki pona dictionary) の冒頭の記事の名称は「Notes on *lipu pu*」なので、本そのものとしては *lipu pu* が正しいような気もしますが、日本国内での通称に合わせます。

*4 <https://twitter.com/notolytos/status/1409484535151042568>

*5 書き途中みたいですが。 <https://github.com/nymwa/dogeza>

付録: 練習問題の解答例

あくまでも「解答例」であり、これら以外の解答もあり得ます。語学書には解答例がついていないものもあるのですが、*pu*に合わせてつけることにしました。

練習問題 1 省略。

練習問題 2 1. *mi jan Ito.*

2. *toki Nijon li ike.*

3. *telo nasa li pona.* 酒は「人を狂わせる液体」なので、*telo nasa*です。

4. *moli li pona mute.* 「救済である」 = 「非常によい」と解釈します。*mute*はなくとも構いません。

練習問題 3 1. *jan li tawa.*

2. *mi kama sona e toki pona.* 「学習している」は「知っている状態になる」ということなので、*kama sona*です。

3. *ona li lukin e lipu.* 「読む」は*lukin*です。

練習問題 4 省略。

練習問題 5 読者の皆さんのが*pu*を買って本格的にトキポナに入門することを期待して、解答例を省略します。

つくばで大病を患つたら

文 編集部 puripuri2100

1 はじめに

こんにちは。昨年、大学生活開始早々にして手術が必要な緊急入院を体験したpuripuri2100です。

つい数日前に医者から「もう病院こなくて大丈夫だね」という太鼓判を押してもらい長い闘病生活に終止符を打つことができたため、今回の経験をもとにした知見を共有したいと思います。

2 前提の整理

病気は多種多様であり一人の人間の経験のみをもとに全てを判断するべきではありません。私がどのような状況であったかを記載しますので、このような条件で得られた知見であるということを念頭に個々の事例に合わせて上手く使ってくれることを期待します。

2.1 かかった病気

気胸という病気にかかりました。これは肺に穴があき、中の空気が胸腔内に漏れだすという病気です。

酷く悪化すれば心臓に負荷をかけ、最悪死に至ります。

主な自覚症状としては

- 胸の痛み
- 呼吸時の苦しさ
- 背中・喉などへの鈍痛

などになります。

通常は自宅等で安静にすることで穴が自然に塞がり、胸腔内の空気も粘膜などから吸収されて元通りとなります。しかし、自然治癒が遅い場合などには病院でドレーンと呼ばれる管を胸腔内に差し込んで空気を抜くことで心肺への圧力を低減させる治療を行います。それでも治癒しない場合には手術を行って穴を物理的にふさぎます。

痩せ型で身長が高めの若い男性が発症しやすい傾向があり、再発率も高いです。こわいですね。

2.2 学年・時期・期間

2022年（学部1年時）の5月中旬に発症し、最終的な退院は7月上旬でした。

1回退院した後に再発して再度入院しているためややこしいですが、概ね以下のよう経

過でした。

5月中旬 朝に発症し、同日夕方に救急搬送されてそのまま緊急入院。

5月中旬 手術

5月下旬 入院から1週間で退院

6月下旬 再発し、同じく夕方に救急搬送・緊急入院

6月下旬 手術

7月上旬 入院から2週間半退院

～3月上旬 通院しながら経過観察

特筆するべきこととして、1回目の入院は春Aの期末期間に、2回目の入院は春Bの期末期間に被っていることがあります。

2.3 住所・搬送先

筑波大学付近のアパートで1人暮らしをしています。実家は関東圏です。

搬送先は筑波大学附属病院でした。

3 救急搬送されるまでに必要なこと

3.1 通報のタイミング

昼間に違和感に気が付けていたら

昼間に身体の違和感に気が付いていた場合、まずは近隣の病院を受診することを検討しましょう。特に筑波大学保健管理センター^{*1}は身近で気軽に受診できる施設ですので、積極的に利用しましょう。

違和感に気が付いたときにはもう遅かったら

違和感に気が付いたときには既に病院が閉まっている時間帯である場合や、そもそもとして動くことがもうできないような状況になっていたりすることがあるでしょう。その時には迷わず119番に通報しましょう。通報の際に必要な情報は電話先の人が上手く誘導して聞いてくれるので、焦らず質問に答えていきましょう。

本当に通報してよいか不安な場合には救急安心センター事業（7119）に電話をかけ、相談しましょう。私も1回目の発症時には7119に電話をして相談し、「今すぐ119に通報しなさい」と言われてから通報しています。

3.2 通報から搬送までの間

搬送されることを親しい人に言っておくと良いでしょう。後々楽です。もし余裕があるなら体調に違和感を感じたときに言えると良いです。後述する準備をしてくれる人が一緒にいるときには非常に良いです。

*1 <https://www.hokekan.tsukuba.ac.jp/>

通報から搬送までの間には搬送される準備をしておく必要があります。

搬送されるときに最低限必要なものは以下のものです。

- ・健康保険証
- ・お薬手帳と今ある薬
- ・筑波大学の学生証
- ・現金 2 万円ほど
- ・携帯電話
- ・充電器等
- ・もしもあるなら各種病院の受診カード

健康保険証・お薬手帳・学生証の 3 つを渡すことで、救急隊・病院は治療の際に必要な情報を得ることができます。また、あなたの状態を各所に連絡する際に携帯電話は必要です。特に、入院の際に保証人が必要であることからも頼れる人に対して連絡を取ることができる状態を常に維持しておきましょう。

連絡と言えば、このときにクラス担任に対して救急搬送されることをメールすることも重要です。そのまま入院することになったとしても、クラス担任を通じて各講義の担当教員に連絡を出してもらうことができる他、様々な配慮をスムーズに受けられたりお得な情報を得られたりするからです。文面としては

○○先生

お世話になっております。情報科学類○クラスの○○です。

この度、急な体調不良により救急搬送されることになってしましました。搬送先や入院の有無、病名などがわかり次第追って連絡したいと思います。

情報科学類○年○クラス（学籍番号）（本名）

程度の簡単なもので良いと思います。

入院を覚悟している場合、入院の準備も必要です。入院のために必要なものとしては、前述のものに加えて以下の物があると思います。

- ・暇つぶし道具
- ・下着と靴下の替え
- ・マスクのストック
- ・洗顔セット
- ・タオル
- ・クリアファイル
- ・割り箸
- ・手提げ袋
- ・症状的に許されるなら軽食

入院中はかなり暇なので、暇つぶし道具を持っていくと良いでしょう。私はパソコンと本を大量に持ち込みました。トラックボールマウスを持っていくと、狭い机環境でも快適な操作性を手に入れることが出来て非常に良かったです。キーボードとしてはHHKBがコンパクトで持ち運びしやすかったです。この入院中で「並行プログラミング入門（オンライン）」・「自然言語処理入門（オンライン）」・「著作権法入門（有斐閣）」などの積読を消化することができて非常に良かったです。



図1 病室でパソコンの用意をしている様子

下着・洗顔セット・タオル・マスクなどは相当必須です。上着に関しては安いレンタルサービスを利用することができるため用意しないでも問題ありませんが、下着については替えが必要になります。コインランドリーがあるので大量にはいりませんが、無いのは良くないと思います。病棟にはシャワーがあり、それをほぼ毎日利用することができます。ただし、タオルや洗剤の貸し出しあまり渋られるので持っていくに越したことはないでしょう。症状によってはシャワーを利用するることはできませんが、看護師に要望を出すことで洗髪・洗顔は利用することができます。この際にも洗剤やタオルがあると非常に便利です。

入院中の食事には箸等が付属してこないため、割りばしを大量に持っていくことは必須です。使い捨ての小さいプラスチックスプーンがあると、たまに出てくるヨーグルトなどに対応でき良いでしょう。食事に関するですが、夜に搬送されると翌日の朝まで食事が無いため夜食として軽食を持っていくと良いと思われます。

入院中には様々な書類が渡されます。これらの中には退院後も必要なものがあたりするので、まとめておくためのクリアファイルや手提げ袋があると良いでしょう。手提げ袋



図 2 病室で本を読んでいる様子



図3 病室でベッドの角度を弄って遊んでいる様子

としては入学時に貰うことができる ITF バッグが便利でした*2。

筑波大学附属病院は今のところ COVID-19 対策で面会禁止になっていますが、差し入れ自体は看護師を通じてできるようになっているので、もし入院中に必要なものが出てきたりした場合は外部の知り合いの助けを借りることができます。看護師を通じて病院内でものを買うこともできますので、搬送の際に全てを用意しようと特別頑張る必要はありません*3。安静にする方が大事です。

4 入院になつたら

成人年齢が 18 歳に引き下げられたことで、大学 1 年生であっても入院の際に必要な同意書などにサインをすることができます。良かったですね。医師の説明を聞いて治療の方針にどこまで同意するのか自分で判断してください。必要な手続きが終わったら病棟に運ばれて様々なことが行われます。もし、その過程で自分の置かれている状況がわからなくなつたら遠慮なくそこにいる医師・看護師に聞きましょう。

入院の際には身元保証人を 1 人付ける必要があります*4、その人には病院に来て書類にサインをしてもらう必要があります。本人とは独立で生計を立てている成人であれば良いため、親や友人に頼むと良いでしょう。入院してから数日経っても良いとはいえ、実際に病院に来てもらう必要があるため、身元保証人を頼みたい人には搬送前にあらかじめ連絡しておくとスムーズになります。親と不仲であり金の絡む保証人になってくれる友人が居ない、などの場合には病院に相談しましょう。かなり柔軟な対応をしてくれると思います。

入院となると講義を受けられない・課題ができないなどの困った状態になります。そのため、履修している講義の担当教員に対して

- 救急搬送されたこと
- 緊急入院していること
- 講義が当分受けられないこと
- 課題が出せないこと
- (期末期間に被っていたら) テストが受けられないこと

を伝え、対処してもらいましょう。体力的な余裕があれば各講義の教員にメールを 1 通ずつ送れると思ってしまいますが、これは数時間にも及ぶ大変な作業であるため、通常はやらない方が良いと思います。私は全部 1 通ずつ書いて気絶しかけました。クラス担任に履修している講義の一覧を送り、クラス担任から事情を送ってもらいましょう。教員の中には学生からのメールは無視するのに教員からのメールには一瞬で反応する方が居ます*5。教員パワーは利用できるときに利用しましょう。概ね

- 課題提出の遅延を認めてくれる

*2 担当医からは「たまに筑波大生が持ってくるこの変なバッグ、なんなの?」と疑問の声を頂きましたが

*3 でも健康保険証・携帯電話・金は頑張って手元に持って来ましょう

*4 入院費用回収の担保の他にも、緊急時の連絡先などの役割があります

*5 本当に居ました

- テストを追試してくれる
- テストを代替課題してくれる

などの配慮をしてもらえます。たまに「遅延 1 週間だけ許してあげるけどそれ過ぎたら残念だけど落单だね」などとのたまう教員も居ます。必修・選択必修科目でそのような扱いを受けたときはクラス担任などに全力で抗議しましょう。自由科目の場合は泣き寝入りする方が楽かもしれません。私は 2 科目でそのような扱いを受けて入院中に課題を必死にやりましたが、結局得单 1・落单 1 で良い結果ではありませんでした。

5 入院中の過ごし方

5.1 飲食

飲み物は各病棟にある自販機でペットボトルを購入するのが基本になります。自力でたどりつけない場合には看護師に頼んで買ってきてもらいましょう。このときのために現金があると便利です。

歩けるようになったら大学病院の 1 階に併設されているコンビニまで買いに行く、などができます。ここでは 2L ペットボトル飲料が売っていたり、ジュースが売っていたりします。また、お菓子も大量に調達できて非常に嬉しいです。後述している屋上にある売店ではパンが売っており、美味しいです。夕方には半額になっていたりするため、これを狙っても良いかもしれません。

病院食は基本的に美味しいです⁶。メニューも豊富で保温されていて美味しさが保たれています。季節のイベントの日には特別メニューが出たりします。

5.2 暇つぶし

症状が良くなると許されることが増えてくると思います。歩くのが許されている場合には、ぜひけやき棟屋上のテラスを散歩してみてください。その手前にあるラウンジでパンを買って食べるのも良いでしょう。歩くのが許されてからずっと、パンを食べて中島みゆきの曲を聞きながらテラスを周遊していた記憶があります。ベンチがあり景色も良いため、歩くのに飽きて休憩している間ものんびりすることができます。

歩くのが許されていない場合は本を読んだりスマホを弄ったりパソコンを叩いていたりしましょう。病棟の休憩スペースには外部ネットワークに接続できる LAN ケーブルが提供されているため、LAN ケーブルをさすことができるパソコンを持っている方はそれを使ってネットサーフィンを楽しんでも良いでしょう。

6 手術になったら

手術中には方針の変更や輸血などの新たに同意が必要な処置が行われることがあります。このとき、患者に対して同意の確認をすることはとても難しい⁷ため、代わりに同意の判断

*6毎回の感想を記録するのが入院中の楽しみになっていました

*7例えば全身麻酔中は明らかに不可能です



図4 病院食（朝食）



図5 病院食（昼食）



図 6 病院食（夕食）



図 7 病院食（七夕バージョン）



図8 テラスの様子1



図9 テラスの様子2



図 10 テラスからの眺め

をしてくれる付添人を用意する必要があります。これは入院時の保証人と同一人物でなくとも問題はないはずです。両親やその時に時間が空いている友人などに頼みましょう。手術が終わっても容態が安定したことが確認されるまで上記の付添人は帰ることが難しいため、概ね手術終了予定時刻から 2 時間程度追加で待機することを見越しておくと良いと思います。

手術に必要なものは病院内で調達することができます。剃刀の刃や手術痕を抑えるバンドなどを購入した記憶があります。これらのリストアップや購入の付添・手伝いは看護師の人が行ってくれるため、支払い手段の用意さえできれば特に考えることもありません。

手術直前は絶食させられることが多々あります^{*8}。食べられないこと自体は良いのですが、OS-1 を朝食代わりに渡されたときにはその不味さに非常に困惑しました。飲み物についても制限がかかることがあるため、指示にしっかりと従いましょう。指示に従わないと最悪手術が中止・延期されます。

手術が無事に終わった後も、回復までが大変です。痛み止めで常時麻酔を点滴等で投与されて意識が常に朦朧としたり、全身麻酔直後の誤嚥防止で半日近く飲食が禁止されたりします。特に手術後に追加で半日飲食が禁止されたことが個人的に入院生活で 1 番辛かったです。看護師に頼むことで口の中を潤すことはできる^{*9}のですが、飲むことはできないので喉の奥が常に乾燥し続けていてストレスがとてもかかりました。

^{*8}特に全身麻酔のときには誤嚥防止で胃を空にする必要があるらしいです

^{*9}口に水を含みそのあとコップに吐き出すことだけができます



図 11 手術当日の朝食であるところの OS-1

7 たいいん！

無事に症状が落ち着き通院治療・自宅療養で対処できるようになると退院できます。退院の数日前くらいに医師から「検査などで〇〇という条件を満たしたら退院できますね。概ねあと〇日くらいでしょう。」というようなことを言われるようになります*10。

退院時には全ての荷物を一気に持つて帰らなければならず大変です。荷物を預けられる家族・知り合いが来られる場合には、看護師を通じて差し入れの逆をやっても良いかもしれません。1人で全てを持って帰る場合でも、病院が台車を貸してくれるので院内の移動はそれでとても楽をすることができます。また、病院の玄関のすぐ先にタクシーがそこそ停まっているのでそれを利用して身体に負担をかけずに帰宅することができます。

退院時には書類手続きや薬の説明などが行われます。これらには概ね2時間ほどかかります。そのため、朝1番で退院しようとしても結局は正午過ぎに病院を出ることになります。退院後の予定を入れる際には気を付けておきましょう。

退院の付近では入院費の支払いが待っています。場合によっては驚くような値段になっていると思いますが、高額医療費支給制度などの救済措置が国などから出されています。病院もそれらの制度の説明を事前にしてくれる所以、遠慮なく利用しましょう。制度を利用する際には所得額が参考されることがあるため、扶養者や自身の所得を証明する書類を事前に用意しなければならない可能性を頭に入れておきましょう。どのような書類が必要であるのか、必要な書類を発行するまでにかかる日数はどれほどか、といった状況は様々ですので必ず確認しましょう。

*10もちろん言われた日数は確実ではなく、容態によっては大幅に前後します

8 退院後

退院して一息をついたら無理のない範囲で祝いましょう。めでたいので。WORD 編集部の人たちとホールケーキをラウンジで喰らうなどをしていました。



図 12 退院祝いのケーキの様子

退院した後にも追加で通院をして経過観察をする必要があることもあります。また、日常生活の中で制約が掛かったりすることもあります^{*11}。必ず医師の指示に従い、なんらかの気づきがあった場合には伝えるようにしましょう。退院直後に容態が悪化した時のためのホットラインなどを退院時に貰うことができるので、必要そうになったらドシドシ活用していきましょう。

退院をするとようやく大学の講義を進めることができるようにになりますが、一気に再開するととても大変です^{*12}。期限を再延長してもらうなどの配慮を担当教員に求めていきましょう。このとき、クラス担任に相談しながら連名でメールを送ると話がとてもスムーズになります。配慮の根拠として診断書を求められることがあるため、発行しておきましょう。退院時に申し込んでも後から申し込んでも大丈夫ですが、1枚持っておくと便利です^{*13}。

9 おわりに

この記事が大いに活躍するような場面が来ないことを祈っています。

*11例えは、私は当分の間自転車に乗ることを禁止されました

*12大変さのあまり精神的に辛くなり、新たに精神科にかかる羽目になりました。精神科の検索・予約はクラス担任経由で大学の支援制度を利用して行いました。便利ですね。

*13発行には1枚につき3300円と、2週間程度の期間がかかります

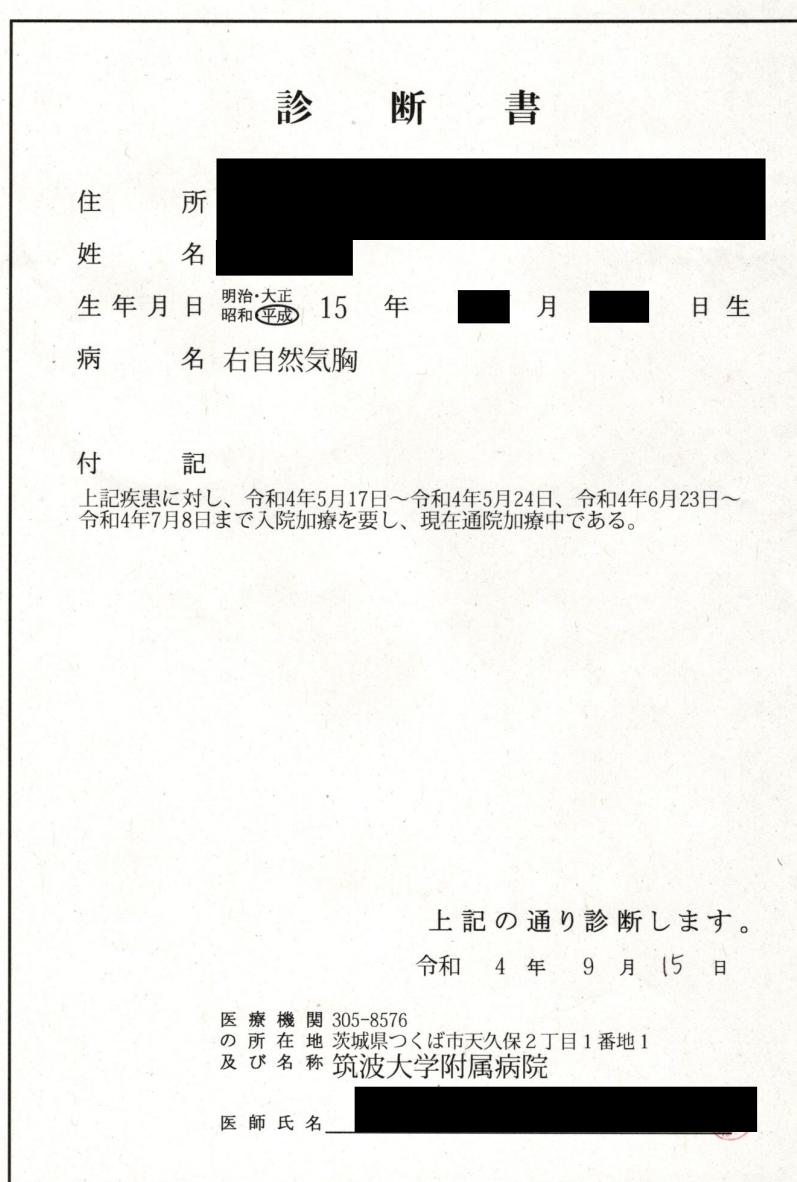


図 13 診断書の実例（氏名・住所・生年月日などを消す加工を行っている）

OpenSCAD で 3D モデルをコードっぽく書く

文 編集部 yuseiito

1 はじめに

新入生および編入生のみなさま、ご入学おめでとうございます。WORD 編集部の yuseiito です。人間とメシに行くのが好きです。^{*1}

入学後、アパートや宿舎での生活を充実させるために、ちょっとしたものを自作できると便利なことがしばしばあります。そんなときに、3D プリンタを持っていると有用です。

ところが、いくら良い 3D プリンタがあっても、3D プリンタで実世界に物体を出力するには、まずその元となるデータ（3D モデル）が必要です。^{*2} 一般に、3D モデルを作るためには 3D CAD が利用されます。代表的な 3D CAD としては、Fusion 360^{*3} や SOLIDWORKS^{*4} などがあります。^{*5 *6}

これらはグラフィカルなインターフェースを持つツールで、3D モデルを作る時には画面に表示された 3DCG に対してマウスやキーボードで操作を加えます。直感的で、操作性も高いと言えますが、現代的なソフトウェアのプログラミングに慣れ親しんだ人にとっては、以下のように若干不便な点が残ります。

- 宣言的記述ができない。
- バージョン管理ができないか、できても独自手法によって行われる。

そもそもこれらのソフトウェアの内部表現はプロプライエタリな形式なので、一度作成したツールを他のソフトウェアに移行したり、作成したコードを複数のプログラムを用いて検証したりすることは難しいのが現実です。

そんな問題意識に対する一つの解決策として、OpenSCAD という少し変わった 3D モデリングツールで遊んでみたので、本稿ではそれを紹介したいと思います。

^{*1} ぜひ誘ってください。よくしゃべる関西弁のオタクなので、coins ラウンジ等でお近くにいるときは簡単に識別可能だと思います。

^{*2} ゲームや XR といった CG の世界でも「3D モデル」の語が使用され、FBX などのデータ形式が存在します。しかし、実世界に物体を出力するためにはその内外の境界が明らかであることが重要なため、通常 STL をはじめとするポリゴンモデルやソリッドモデルの表現を利用します。このように、同じ「3D モデル」の語でも、分野によって微妙に指示示すものが異なり、そのデータ形式によって表現できる情報が異なっています。これらは相互に変換できないことが多い点にご注意ください。

^{*3} <https://www.autodesk.com/products/fusion-360/overview>

^{*4} <https://www.solidworks.com/>

^{*5} これらのツールは高機能で、実際には純粋な CAD ではなく CAM や CAE の機能も併存しています。

^{*6} Fusion360 をはじめとする Autodesk 諸製品は大学生に対して学生向けライセンスを発行しています。非常に有用ですので是非とも試してみてください。

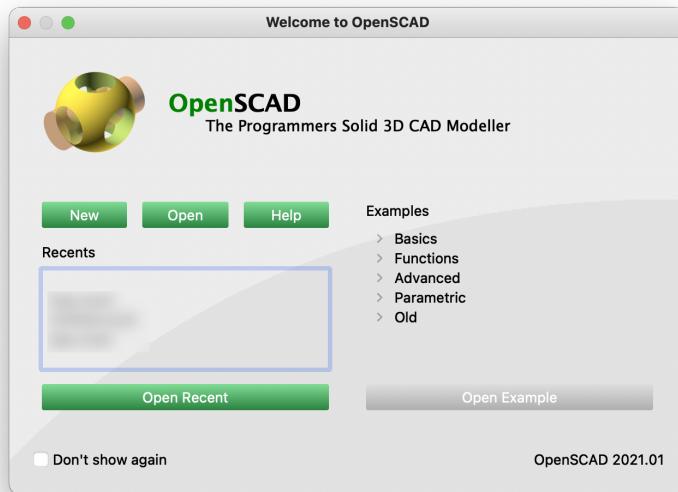


図 1 OpenSCAD の起動画面

2 OpenSCAD とは

OpenSCAD とは、3D モデルをプログラムのようにテキストベースで記述することができる 3D CAD です。C++ で書かれたオープンソースのソフトウェアで、独自の記法を用いて 3D モデルを記述します。独自の記法といっても、標準的なプログラミング言語の作法にかなり似ているので、C や Python といったプログラミング言語を多少触ったことのある方なら簡単に使いこなせることと思います。（具体的な文法については後述します。）

<https://openscad.org/> からダウンロードすることができますので、インストールしてみてください。

Mac OS や Windows, Linux といった主要な OS に対応しているようです。^{*7} 筆者のように Mac OS を利用されている場合は、Homebrew を利用してインストールすることもできます。

3 OpenSCAD の基本的な使い方

では、OpenSCAD を起動してみましょう。

起動画面（図 1）で「New」をクリックして、新しいファイルを作成します。画面の左側にテキストエディタがあり、右側に 3D モデルがプレビューされます。

それでは左側に、以下のようなコードを書いて $\text{⌘} + \text{S}$ ^{*8} で保存してみましょう。

```
1 | cube(10);
```

^{*7}筆者による動作確認は Mac OS でのみ行いました。

^{*8}おそらく Windows や Linux では Ctrl+S だと思います。

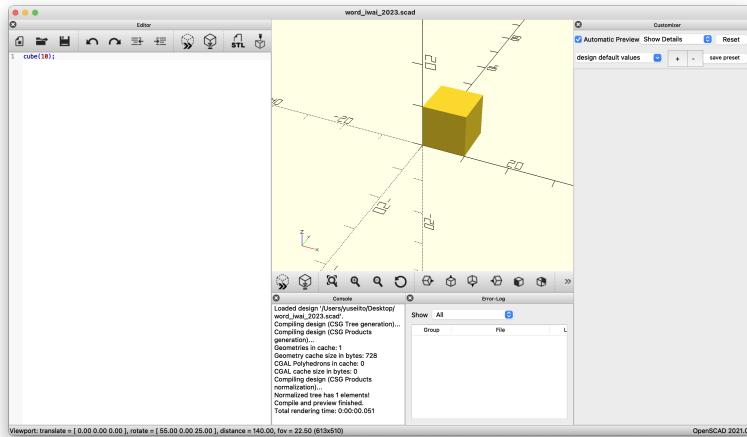


図 2 立方体をモデリング

右側のプレビューが更新され、一辺が 10mm の立方体が表示されるはずです（図 2）。

OpenSCAD では、このように所望の立体を構成する基本的な形状を組み合わせてプログラム上に記述することで、3D モデルを作成します。

オブジェクトの位置を変更するには、`translate()` を使います。

たとえば、以下のようにすれば、立方体の原点が $(x, y, z) = (0, 0, 0)$ から $(0, 0, 5)$ に移動します。

```
1 translate([0, 0, 5]) cube(10);
```

さらに、`union()` を使うと、複数のオブジェクトを結合することができます（図 3）。

```
1 union() {
2   cube(10);
3   translate([0, 0, 5]) cube(10);
4 }
```

逆に、`difference()` を使うと、オブジェクトの差分を取り（先のオブジェクトから後のオブジェクトの部分を除く）ことができます（図 4）。*9

```
1 difference() {
2   cube(10);
3   translate([0, 0, 5]) cube(10);
4 }
```

*9 どうやらプレビュー表示の CG の境界判定が微妙で、除かれる範囲の境界と除く面の境界が一致するときに、きちんと除ききれていないような表示になることがあるようです。境界面の表示がおかしい時には、コードの `translate` の引数を `[0,0,6]` のように変えることでよりはっきりとした表示にすることができます。

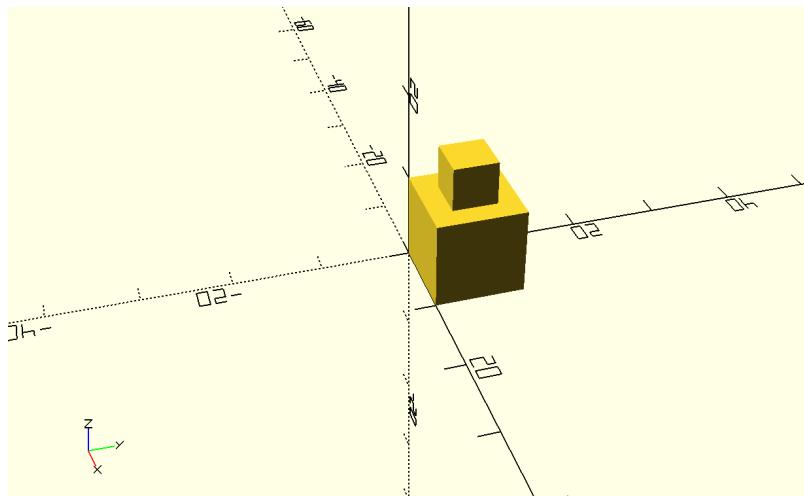


図 3 立方体を積み重ねる

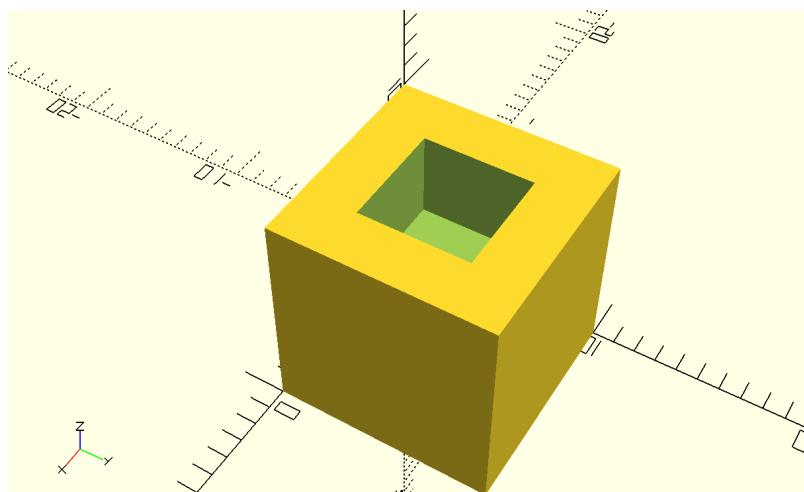


図 4 立方体から立方体を除く

`union` や `difference` 以外に、`intersection`（積集合）や `minkowski`（ミンコフスキーアンド）などの関数もありますので、ぜひ試してみてください。^{*10}

^{*10}<https://openscad.org/documentation.html>

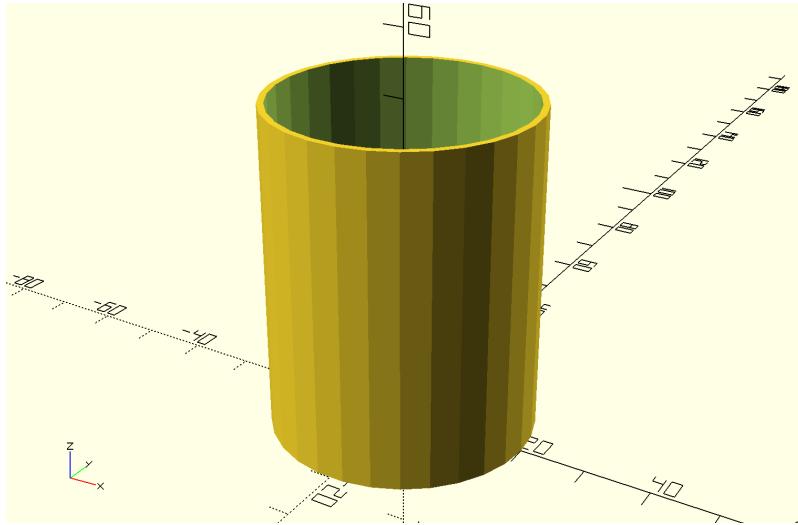


図 5 最も単純なカップ

4 実践編：カップを作る

それではここから、実際に少し複雑な形状を OpenSCAD で書いてみます。題材は、カップです。

まず、最も原始的な形状は、以下のように表現できます（図 5）。これは、高さ 50mm、底面の半径 20mm の円柱から、肉厚が 1mm になるようにより小さい円柱を引いたものです。

```

1 difference(){
2     cylinder(50,20,20);
3     translate([0,0,1])
4     cylinder(50,19,19);
5 }
```

しかしこれだと、口に当たる部分が直角になっており、鋭利で危険です。そこで、口に当たる部分に丸めが効くように、`minkowski` を使ってみましょう（図 6）。

```

1 difference(){
2     minkowski(){
3         cylinder(50,20,20);
4         sphere(1);
5     }
6     translate([0,0,1])
7     cylinder(50,19,19);
8 }
```

同様に、内側の底面の部分にも丸め処理が必要です。

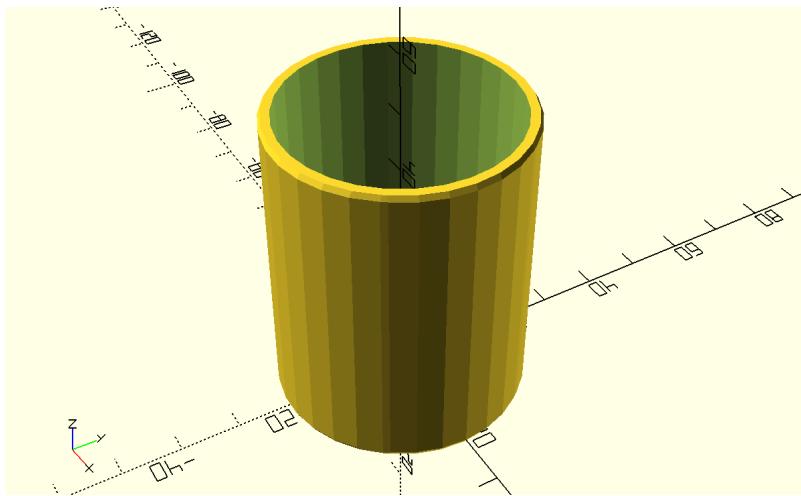


図 6 外側に面取りを適用したコップ

```
1 difference(){
2   minkowski(){
3     cylinder(50,20,20);
4     sphere(1);
5   }
6   translate([0,0,1])
7   minkowski(){
8     cylinder(50,19,19);
9     sphere(1);
10  }
11 }
```

最後に、取手をつけてみます。

```
1 union(){
2   difference(){
3     minkowski(){
4       cylinder(50,20,20);
5       sphere(1);
6     }
7     translate([0,0,1])
8     minkowski(){
9       cylinder(50,19,19);
10      sphere(1);
11    }
12 }
```

```

11      }
12    }
13    translate([0,20,22])
14    rotate([0,90,0])
15    difference() {
16      cylinder(5, r = 20);
17      translate([0,0,-0.5])
18      cylinder(6, r = 20 - 5);
19      translate([-20,-20,-0.5])
20      cube([40,20,6]);
21    }
22 }

```

これで、コップの完成です（図 7）。*11

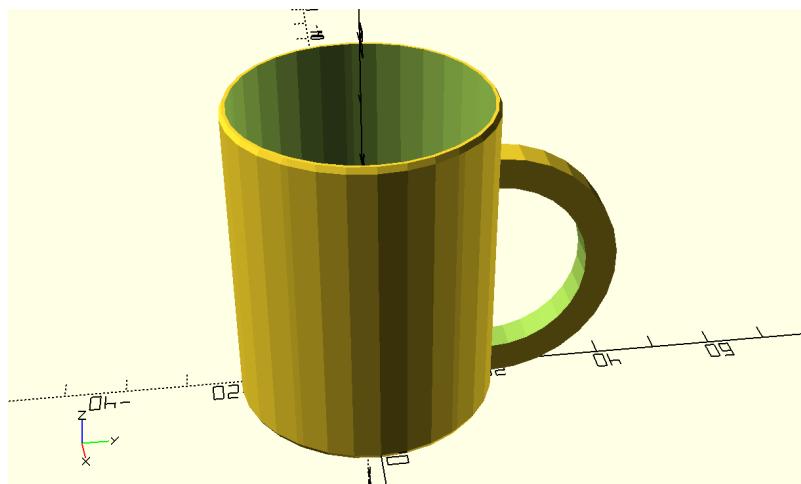


図 7 コップ

*11 これは嘘で、本当は取手の方にも丸めを適用するなど細かい処理が色々あるのですが、新しい操作はもうこれ以上出てきません。記事にするにはコードが不必要に長くなるだけなので、この辺りで区切りをつけます。

5 コップのリファクタリング

前章では、コップのモデリングを順を追ってご説明しましたが、ここでコップのコードを改めて眺めてみると、多くの定数が直接指定されていることが分かります。ここで、例えばコップの直径を変えたり、取手の半径を変えたりしたい場合には、それに合わせてそれぞれの定数を変更するために人力で線形代数のようなことをせねばならないことがお分かりいただけるかと思います。これでは保守性などあったものではありません。

そこでここからは、コップのコードをリファクタリングしていきます。そもそも、ほとんどの定数は、コップや取手の半径をはじめとする基本的なパラメータを元にした計算によって求めることが可能です。OpenSCAD では、一般的なプログラミング言語同様、変数や基本的な代数演算子が利用できます^{*12} から、これらを使って置き換えることができます。変更後のコードは以下のようになります。

なお、// から始まる部分はコメントです。ここに設計意図などを書いておけるのも、コードで立体を記述することのメリットと言えるでしょう。

```
1 rCup=20; // Cup Radius
2 hCup=50; // Cup Height
3 tCup=3; // Cup Thickness
4 rHandle=20; // Handle Radius
5 tHandle=5; // Handle Thickness
6 wHandle=5; // Handle width
7 rRound=1; // Round Radius
8 digOffset=0.5; // Dig Offset
9
10 union(){
11     difference(){
12         minkowski(){
13             cylinder(hCup,rCup,rCup);
14             sphere(rRound);
15         }
16         translate([0,0,tCup])
17         minkowski(){
18             cylinder(hCup,rCup-tCup,rCup-tCup);
19             sphere(rRound);
20         }
21     }
22
23     translate([0,rCup,hCup/2])
```

*12 https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/The_OpenSCAD_Language#Variables

```

24     rotate([0,90,0])
25     difference() {
26         cylinder(wHandle, r = rHandle);
27         translate([0,0,-digOffset])
28         cylinder(wHandle+digOffset*2, r = rHandle - tHandle
29             );
30         translate([-rHandle,-rHandle,-digOffset])
31         cube([rHandle*2,rHandle,tHandle+digOffset*2]);
32     }
33 }
```

さらに、コードの中での「コップ本体」と「取手」というような部品ごとの区切りを明らかにするため、モジュールにまとめてみます。こうなっていれば、取手を別の形に変えたり、同じ取手を別ものに取り付けたりすることができ、再利用性がさらに高まります。

```

1 module CylinderCupBody(r,h,t,rounds){
2     difference(){
3         minkowski(){
4             cylinder(h,r,r);
5             sphere(rounds);
6         }
7
8         translate([0,0,t])
9         minkowski(){
10            cylinder(h,r-t,r-t);
11            sphere(rounds);
12        }
13    }
14 }
15
16 module CupHandle(r,w,t){
17     digOffset=0.5;
18     difference() {
19         cylinder(w, r = r);
20         translate([0,0,-digOffset])
21         cylinder(w+digOffset*2, r = r - t);
22         translate([-r,-r,-digOffset])
23         cube([r*2,r,t+digOffset*2]);
24     }
25 }
```

```

25 }
26
27 rCup=20; // Cup Radius
28 hCup=50; // Cup Height
29 tCup=3; // Cup Thickness
30 rHandle=20; // Handle Radius
31 tHandle=5; // Handle Thickness
32 wHandle=5; // Handle width
33 rRound=1;
34
35 union(){
36     CylinderCupBody(rCup,hCup,tCup,rRound);
37     translate([0,rCup,hCup/2])
38     rotate([0,90,0])
39     CupHandle(rHandle,wHandle,tHandle);
40 }
```

再利用の一例として、取手を両側に取り付けたコップは、以下のように書けます（図 8）。

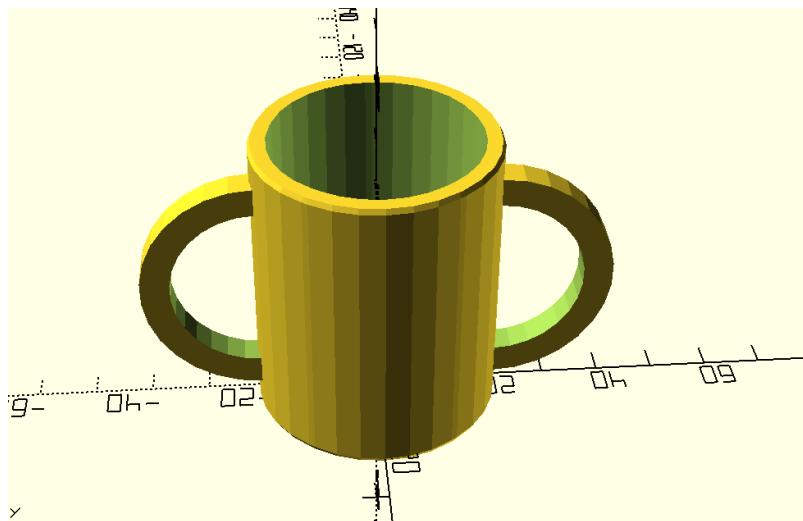


図 8 両側に取手を付けたコップ

```

1 // (前略)
2 union()
3 {
4     CylinderCupBody(rCup, hCup, tCup, rRound);
5     mirror([ 0, 1, 0 ])
```

```

6  {
7      translate([ 0, rCup, hCup / 2 ]) rotate([ 0, 90, 0 ])
8          CupHandle(rHandle, wHandle, tHandle);
9      }
10     translate([ 0, rCup, hCup / 2 ]) rotate([ 0, 90, 0 ])
11         CupHandle(rHandle, wHandle, tHandle);
12 }

```

OpenSDAD では、STL ファイルを出力することができるので、ここから 3D プリンタで印刷することも可能です。今回はそこまでは扱いませんが、自分で書いたコードが 3D プリンタで出てくると結構感動するので、試されることをお勧めします。

6 おわりに

このように、OpenSCAD を使うと、コードで立体を記述することができます。そのことにより、以下のようなメリットが享受できます。

- プレーンテキストなので、再利用や解析、バージョン管理が容易。
- 単純な形状なら数行で済むため、GUI ベースのものよりも簡単に記述可能。
- 形状を `module` として再利用できる。
- コメントを書くことで、設計意図を「図面」(=コード) に記録できる。
- 変数や代数演算が利用でき、複雑な形状も記述できる。
- 好みのテキストエディタで編集できる。^{*13}

また、これは私の妄想でしかありませんが、このようなオープンなフォーマットを利用することで、将来的には OpenSCAD とは別のテストツールのようなものも書けるのではないかと考えています。

例えば干渉チェックなどを行うテストケースを記述して CI するなんて世界が実現すれば、筆者のような CI だいすきお兄(姉)さんなら恍惚してしまうこと間違いありません。

OpenSCAD によって皆様が 3DCAD に振り回される不毛な時間がなくなり、より便利な有体物がこの世にデザインされることを願ってやみません。^{*14}

^{*13}筆者は Neovim を使って書いています。

^{*14}その際にはぜひ Public なリポジトリでオープンソースにしていただけると嬉しいです。プレーンテキストで簡単に共有できますからね！ :)

ポスト西友時代を生きる

文 編集部 なかやばし

西友つくば竹園店が3/31に閉店した。最近はロピアに行きがちだったものの、それでも西友のプライベートブランドは、ロピアに並ぶ品物より安く、西友の冷凍唐揚げ、冷凍うどん、そばには夕食の献立が決まらない時、かなり世話になった。さらに、西友は駅前・春日地区では珍しく、日用雑貨も食品も扱うスーパーだったので、食品の買い物ついでに洗剤を買うこともできた。

つくば竹園店が閉店すると、最寄りはどこになるのであろうか？ 西友つくば竹園店が閉店してからの最寄りの西友は、ひたち野うしく駅東口にある西友ひたち野うしく店になる。どれどれ、どれくらいの距離があるんだろうか、と openstreetmap.org へアクセスし、春日宿舎から西友ひたち野うしく店までの経路検索をかけてみる。すると、道のりは片道で13kmもあり、これを週1で行くにも月1で行くにもかなりしんどい。

というわけで、筑波大学周辺に住む筑波大の学生にとって、ポストコロナ時代へのゆるやかな移行と共に、今年度からはポスト西友時代も到来することとなる。どのスーパーで、抜けてしまった西友の穴を埋めるのか？ 金銭的にも時間的にも苦学生である我々にとって、これは重大な問題である。

この記事では、西友の穴を埋められるスーパーを紹介していこうと思う。

1 おことわり

この記事は、西友ロスのガワを被ったスーパー品評記事である。

また、私は春日1丁目住みであるため、ここで取り上げるスーパーは、私の生活圏となる主に駅前や天久保1、2丁目のスーパーとなる。なので、桜や春日3、4丁目に住む人はあまり参考にはならないかもしれない。というのも、桜にはカスミ、ウエルシアやカワチがあるし、春日3、4丁目は駅方面へ向かうよりも、最早トライアルのほうが近い。

2 100円ローソン

松見通り沿い、天久保2丁目に立地する、スーパーなのかコンビニなのかよくわからぬ代物である。24時間営業なので、春日からならば夜間徘徊に最適である。ただ、ローソンとはいえローソンにあらず、コンビニとは明らかに雰囲気が違う。焼き芋が店前に陳列されており、店に入ると直ぐに果物が出迎えてくれる。また、ゆとりがなく所狭しと商品が並ぶのもコンビニとしては異質である。

安いものと高いものが混ざっており、基本的にはプライベートブランドを買うとよい。プライベートブランドの卵は、6個入りで、10個入りを使い余す一人暮らしにとってはとてもありがたい。6個入りの卵は、新興ベッドタウンのつくばでは珍しい。加えて、ロピアで

は 10 個入りに比べ高いので西友に頼っていたのだが^{*1}、なんと 100 円ローソンでは良心的な価格で売ってくれている。また、冷食のチャーハンは一食分で 130 円程度と安く、味としても申し分ない。ただ、パンはいまいちであることが多い。

3 ロピア

駅周辺にある商業施設の 1 つ、トナリエ CREO 1 階にある。

基本的には安いが、一部高い。それでも正直誤差と言ってしまってもよい気がする。酒の品揃いはよく、見たことのないワインやウイスキーなどがそこそこの数並ぶ。生鮮食品はかなりの量で安い値段で売られているため、毎日自炊しない限り、友人と分ける、先に切って冷凍保存などをしないと腐ってしまうので注意が必要。また、日用雑貨は扱われていないので、ウエルシアなどへ別途行く必要がある。

閉店が早く、20:00 には閉まってしまうのが社会不適合者にとって厳しいこと、現金のみでしか決済できないことが玉にキズ^{*2}。

4 スギ薬局 つくば天久保店

ビッグエコーの北東、ファミリーマート天久保 1 丁目店の南西に位置し、吾妻中学校の南側に面する道を東大通りに向けて進むとある。最近オープンした。閉店は 22:00 で、ロピアより長く営業していてこちらも助かる。開店セールなのか、アイスコーヒー、食パンや菓子が安い。開店セール明けの今後の動向に注目。

5 トライアル

学園の森にあるディスカウントストア。たまに起る、夜中の徘徊への欲望を満たすには春日 1 丁目から丁度いい距離にある。24 時間営業なのがありがたい。食品から雑貨、食器、衣類まで、なんでも揃うので、トナリエのダイソーや DECO HOME が閉店した時にとても重宝する。

6 ウエルシア

基本的に、価格としてはロピアに負ける。が、春日・駅前には日用雑貨を扱うスーパーがなくなるため、日用雑貨を買う際には重宝する。T ポイントユーザにとっては、月曜日に買い物をするとポイント 2 倍になるため、計画的に利用すると得である。ここでは、春日から近く、個人的に利用する 2 店舗を紹介する。

6.1 つくば学園吾妻店

吾妻 3 丁目にある。EXPO センター北東、または筑波学院大東の方向にある、ちょっとした飲み屋街と住宅街が混在した場所にある。品揃えは少ない印象。だが、常に「濃厚 Choco ブラウニー」のまとめ買いが安いので重宝している。

^{*1}なんと 1 個単価ではなく、単純に価格が高いのである！

^{*2}しかし、宿舎には両替機がないため、ATM で下ろしてきた 1000 円札が 100 円玉に崩せる点、宿舎民としてうれしい

6.2 つくば研究学園店

品揃えはそこそこ多い。24時間営業なのもうれしい。酒、洗剤、シャンプー、食材、菓子が揃っていて、バランスがよい。調剤もやってくれるので、通院した際には便利そう。

6.3 つくば学園中央店

西友のあるデイズタウン南西にある。品揃えは少ない印象。店舗としては小さめで商品が所狭しと並ぶ。

7 おわりに

ここまで、様々な食品雑貨店を見てきた。

新入生の皆さんも、西友を使っていた22年度以前の入学者も、この記事を参考により苦学生生活を送ってほしい。

ちなみに私は、当初ロピアときどき西友と使い分けていた。しかし、西友の閉店を知つてから、基本的にはロピアへ行くが、菓子やコーヒー、食パンはスギ薬局、夜中の散歩では100円ローソン、たまに気が向いたら行くトライアルとウエルシア、という形で各店舗を使い分けるようになった。

なんとも締まらない感じで終わろうとしているが、赤入れまであと1時間を切っているので許してほしい。

怪しい団体の勧誘に巻き込まれた話

文 Q=0JyqLCq2c4)NmX-B8V

1 はじめに

この記事はクイズの体裁になっている。空欄が **1** のように設けられているので、この空欄にあてはまる適切な語・数値を補充したり、選択肢から適当なものを選ぶことで記事が完成する。ただし、異なる空欄に同じ語が入る場合もある。また、同じ番号のついた空白には、同じ語句が入る。2回目以降にその空白が出現する場合は、**1** のように細い空白で示す。要するに共通テストと同じ体裁である。正解は筆者 (Q=0JyqLCq2c4)NmX-B8V)のみぞ知るので、自信のある解答ができた方は Q=0JyqLCq2c4)NmX-B8V を探すように。

2 事業

1 月 **2** 日、筆者は諸般の事情により **3** でぶらぶらしていた。すると、ある人物に声をかけられた。この人物を X としよう。少し話してみると、X は地元の人で、X と筆者は趣味が同じらしい。**4** の交換を持ち掛けられたので、交換することにした。

5 月 **6** 日、筆者の居住地である **7** 周辺で **8** というお誘いをもらう。筆者は快諾し、**9** 月 **10** 日、場所は **11** となった。ただし、**3** から見ると **11** はアクセスがやや悪く (**7** 周辺には **3** からのアクセスがより良い場所が存在する), 不自然だなあと思っていた。

9 月 **10** 日を迎えた。その日の朝、「もしかして X の目的は **12** なんじゃないか」と若干の疑いが出てきたが、ひとまず **11** に向かうこととした。**11** で X と合流した筆者は、当初の目的を達成するために **11** にある **13** に向かった。この日の **11** の **13** は混雑しており (**9** 月 **10** 日は **14** 曜日だったので仕方ない), 滞在時間が **15** 分に制限されていた。

当初の目的を済ませたとき、X がこう切り出した。「筆者に試してもらいたいことがある。」この時点では X との共通の趣味の話題が始まることを予期していたが、どうもそうではないらしい。曰く、筆者が幸せになるためには **16** とよい **17** が必要らしい。**16**、なんと **18** であろうか。この時点で筆者の **12** ではないかという予感は 99% ほど確信に変わる。そして **16** と **17** を手に入れる、すなわち **19** を確立するためには、**20** に従って **21** とよく、それを試せる場所が **7** にある **22** という場所にあるという。**20** である。**12** ではないかという予感は見事的中することになった。さらに、**21** を試している人は **23** 人もいるという。しかも **22** で **21** ことには **24** という。実際 X も **21** ことで **25** であるという。**24** ! そんなうまい話があるだろうか、いやない。ふと X を見ると、**26** という状態だった。

文章にして簡潔にまとめると手短なように思われるかもしれないが、かなり論法がまわりくどく、かつまくしててくるような口調であったため、かなりの時間をしていたことに注意しなければならない。ただし白状すると、筆者は恐怖感よりも **27** という感覚を抱いていた。早く逃れたい。「**28**」。ふと時計を見ると、例の **15** 分制限をもう越えているではないか！ 幸いにして **15** 分制限を越えていたがために **13** を後にすることに X も賛成した。なお **13** で **8** のために筆者と X それぞれが要した費用は、各自で負担した。

13 を後にして **11** の駅まで戻るときにも、ひたすら「**24** から **22** に行かないか、**23** までの交通費も出す」という話が続いていた。**29** という四字熟語があるが、最早筆者は文字通り **29** という感じである。そのまま駅に行き、**30** 線の改札を入ったが、X もついてきてしまった。最終的に **30** 線のホームまで続いていたが、**21** を試している人は「**23** 人もいる」という発言を思い出し、「**31**」なので（意訳）と言い残して電車に乗ることで、無事に（？）逃げることができた。このときに X は「**32**」のようなことを言っていたかもしれないが、よく覚えていないしよく聞き取れもしなかった。

ちなみに、このことを筆者が内輪のグループに報告したときの文面が、以下である。

【悲報】 **33** と言われて行ったら **12**

それに対する内輪の反応は、「**34**」「**35** シリーズだ」というものであった。

さて、X は一体何者だったのだろうか。X と会った場所が **11** であること、**20** というキーワードおよび **22** に何かの場所があるということ、**12** をしているという状況から、筆者は **36** と推察する。当初は **37** かと思っていたが、**37** はよくよく考えるとこれらの条件を満たさない。ただし、最後まで X は特定の団体名を名乗らなかったため、**36** ののかは X のみぞ知るところである。

その日は、筆者はやけっぱちで **38** 行き **39** で **40** をしていた（もっとも、その後調べたところによれば、X がもし **36** であれば **38** 行ったのは悪手だったかもしれない）。**12** されるくらいなら **41**、というのが偽らざる筆者の本音である。

3 教訓

筆者が以上のような経験に遭遇してしまった原因として適切なものを、次の①から④のうちからすべて選べ。**42**

- ① 見知らぬ人と **4** を交換したから。
- ② **4** を使っていたから。
- ③ 自宅通学だからといって油断していたから。
- ④ 性善説に従って動いていたから。

もし **12** に巻き込まれてしまったときの対応として適切なものを、次の①から④のうちからすべて選べ。 **43**

- ① 素直に相手に従う。
- ② きっぱりと拒絶する。
- ③ 相手の所属団体と推察される団体と対立している団体の構成員のふりをする。
- ④ 既に相手の所属団体の構成員であるふりをする。

実際は、 **43** のように行動することはしばしば難しいため、Xのような **12** が目的の人物と深く接触しないことが重要になる。このための方法として適切なものを、次の①から④のうちからすべて選べ。 **44**

- ① 外出しない。
- ② インターネット上で提供されるサービスを使わない。
- ③ **4** を使わない。
- ④ 人に話しかけられても無視する。

4 大学公式の情報

- 2022年7月15日付「【注意喚起】カルト勧誘について」<https://ssc.sec.tsukuba.ac.jp/archives/17221>
- 2022年9月20日付「【注意喚起】不審な団体等の勧誘に注意」<https://ssc.sec.tsukuba.ac.jp/archives/19605>
- 「セーフティライフ（2023年度版）」<https://www.tsukuba.ac.jp/campuslife/support-support/safeliving.pdf>

5 おわりに

幸いにして筆者は（少なくとも執筆時までは）逃げ切ることができているが、一般の場合はそうとは限らないので、反面教師にしてほしい。

- 「セーフティライフ」に必ず目を通す。
- つくばにいてもいなくても、怪しい団体の勧誘を受けることがある。
- 一般の場合、人間に対する態度は性悪説に基づくべきである（性善説に基づいてよいのは特殊な場合に限られる）。
- 何かあったら、すぐに信頼できる第三者に相談する。

もっと思想の強いことを述べようと思えば述べられるのだが、本記事の趣旨からは脱線するため、この程度にしておく。

Linux 利用のすすめ

文 motorailgun

1 はじめに

はじめて。coins21 に 3 編予定の motorailgun です。記事執筆時点では別の大学に学籍があります。

今回は新入生歓迎号に新入生が記事を書いていたらちょっとおもしろいんじゃないかというノリで、Linux の利用のすすめについて紹介させていただこうかと思います。

2 Linux を利用する利点と欠点

2.1 Linux とは？

まず、Linux とはなんぞや、と思った方も多いかもしれません。一言で Linux と言ったとき、狭義には Linux カーネルを指すこともありますが、一般には Linux カーネルに加えその他必要なソフトウェア等を同梱した Linux ディストリビューションを指すことが多いです。大まかに言えば、Windows や macOS、ChromeOS のようなオペレーティングシステムのことだと考えて差し支えありません。

2.2 利点

さて、Windows や macOS の代わりに Linux 使用する場合、どのような利点があるのでしょうか。

もちろん様々な意見はあると思いますが、私は多くの人が同意してくれるであろう点として、「自由」があると考えています。

皆さんはご自身の使っている OS に不満を抱いたことはあるでしょうか。例えばこれから寝ようとしているときにパソコンをシャットダウンしたら、Windows Update で部屋が青色に照らされたまま長く待たされたり、最新の OS への更新を促す通知がうざったく留まったり……。

このような挙動は、しばしば無効にできなかったり、あるいは特殊な方法で無効にする必要があります。一方で Linux の場合、あるソフトウェアの挙動が嫌であれば似たような働きをする別のソフトウェアを使うとか、もしくはソフトウェアそのものに手を入れて挙動を変更してしまうことが容易に可能です。ただし多くの場合は、そこまでしなくとも設定等で変更が可能な場合が多いです。

基本的に Linux は、OS のほとんどの部分の状態や挙動を自分でコントロールすることができます。不満があれば改良することができます*1。

*1 これは、Linux に使用されているソフトウェアの多くが FLOSS=自由ソフトウェアとして公開されているた

また、自動化を念頭にした機能がデフォルトで多く含まれているというのもあるでしょう。Linux はサーバー用途で使われることが多いため、人の手が介さない作業を実行するための準備が整っています。例えば、あるフォルダのファイルを一日一回バックアップしたい……といった時は、cron を使えば^{*2} コマンド一つで設定することができます。cloud-init や Ansible、あるいは単純なスクリプトで PC のセットアップを自動化することすら可能です。

2.3 欠点

もちろん、欠点も存在します。

その一つは、商用ソフトウェアやサービスが動かない、もしくはサポートされない場合があるということです。例えば多くの人が利用しているであろう Microsoft Office や Adobe ソフトウェアはそのままでは動きません^{*3}。仮にサポートしていたとしても、一部のディストリビューションのみ^{*4} といったケースもあります。

一方で、少なくとも近年においては、多少の改善を見せてきていると言えるかもしれません。例えばゲームに関しては、Valve 社の貢献や DXVK の登場によって、かなりの割合のゲームが Linux 上で動作するようになってきました。ユーザーからの報告をまとめたサイトである ProtonDB^{*5} によると、Steam 上の Top 1000 のゲームのうち 86% が silver ほとんど問題なく動作する以上であるとされています。

また、Linux 上では動かないソフトウェアの代わりに動くものを探すという手もあります。Maya に対する Blender、Illustrator に対する Krita、Photoshop に対する GIMP、Microsoft Office に対する LibreOffice……のように、近年では驚くほどのクオリティのソフトウェアが OSS として公開されています。

サポートされないというのは、ソフトウェアだけでなくハードウェアにも当てはまります。特に、標準規格に則っていない、メーカー独自のソフトウェアやドライバが必要なハードウェアや、極端に新しいものは単純に PC に挿しても動かないといったことが多いです。私も中学生くらいのときには、iPod Shuffle がうまく使えなかったり、Wacom のペンタブレットが繋いだだけでは動かずに苦労したものでした。

逆に、標準規格に則っているものは動く公算が高いです。Web カメラであれば USB Video Class、音声デバイスであれば USB Audio Class 等の標準規格が挙げられます。

3 ディストリビューションの紹介

Linux ディストリビューションには、様々な特徴を持ったものが多種多様に存在しています。ここでは、その中から主に 2 つ紹介します。

めです

^{*2} 私としては cron ではなく systemd timer を使うことを推奨しています

^{*3} Adobe Photoshop に関しては Linux 上で動かすためのリポジトリをいくつか観測しています

^{*4} RHEL や Ubuntu のみ、など

^{*5} <https://www.protondb.com/>

3.1 Ubuntu

Canonical と Ubuntu コミュニティが主体となって開発されているディストリビューションです。体感ではありますが、新規の Linux ユーザーのうちかなりの割合の人がここから始めている印象を受けます。

最初から GUI が整備されていたり、日本語コミュニティの存在によって日本語化が丁寧であったりと、元 Windows ユーザーでも非常に使いやすい印象があります。また日本語情報が多く、トラブル発生時に先達を参考にしやすいという利点もあります。

一方で、ソフトウェアのバージョンが若干古く抑えられがちだったり、近年は Snap の同梱・一部パッケージの Snap 化がやや懸念されます。

3.2 Arch Linux

Arch Linux コミュニティによって開発されているディストリビューションです。これまた体感ではありますが、Linux をしばらく使い続けた人間が次にはまる沼のように感じています。

上流のソフトウェアに極力変更を加えずにパッケージ化していること、バージョン番号がなく、ローリングリリースと言って常に最新の状態に保つことが前提なのが特徴です。また、標準でインストールされるソフトウェアの数が少なく、それ以外をユーザーの選択に任せていること、ユーザーの作成したパッケージを配布する^{*6}Arch User Repository の存在が特徴です。

非常にカスタマイズ性の高い一方で、多くのソフトウェアがリリースのすぐ後に降ってくるため、不具合に遭遇する確率がやや高いという欠点もあります。ただし、ディストリビューション側が面倒を見る部分が少ないとあって、その不具合を解消しやすいという側面もあります。

3.3 その他のディストリビューション

この他にも、アプリケーションを別個の VM に隔離する Qubes OS、構成や設定を宣言的に管理する NixOS、ルーターでの利用に特化した OpenWRT など、様々なディストリビューションが存在しています。

4 おわりに

ここまで Linux と Linux ディストリビューションの紹介をざっくりとしてみましたが、どうだったでしょうか。私の自業自得ですが、時間もなく説明しきれなかった部分も多いので、興味を持ってくださった方はぜひ、インターネット上の情報を調べたり、自分で Linux に触れたりしてくださると幸いです。

^{*6}正確には「パッケージをビルドするためのファイル」を配布する

情報技術関連課外活動のすゝめ

文 Crow314

1 はじめに

新入生・移行生の皆さん、入学・移行おめでとうございます。情報科学類へようこそ！編集部の Crow314 です。

筑波大学では、有志学生による情報技術に関する活動が盛んです。これらの活動に参加することで、ただ授業を受けているだけでは得られない技術的な知識や経験が得られるこことでしょう。

本記事では、筑波大学において有志学生によって行われる情報技術に関する活動を紹介します^{*1}。なお、ここでは大学から何らかの形で公認された活動のみを紹介します。

2 WORD 編集部

まず、皆さんが今手に持っている^{*2} この雑誌についてご存知でしょうか。この雑誌は情報科学類の学類誌「WORD」といいます。WORD は「情報科学」から「カレーの作り方」までを手広くカバーする総合的な学類誌です。過去の記事は、公式ホームページ (<https://www.word-ac.net>) から閲覧できます。WORD の歴史^{*3} に興味があれば、WORD51 号「学類誌今昔物語」^{*4} をご一読ください。

WORD 編集部は WORD の発行や WORD に掲載する記事の執筆を行う、情報科学類公認の団体です。WORD 編集部には筑波大学でも屈指の技術を持つ編集部員が多数在籍しています。編集部員の専門分野は、アルゴリズム、システムプログラム、ネットワーク、セキュリティなど多岐にわたります。このため、WORD 編集部は WORD の発行に携わるだけでなく、技術者コミュニティとしての側面を有しています。

WORD 編集部は情報科学類に属する団体ですが、WORD 編集部には情報科学類生だけでなく、地球学類、物理学類、情報メディア創成学類、知識情報・図書館学類など、様々な学類の学生が所属しています。また、OB との交流が多いことも特徴のひとつです。

WORD 編集部室は編集部員であれば 24 時間利用可能です。編集部室は記事の執筆や製本といった作業だけでなく、編集部員同士の交流の場としても使用されます。

WORD 編集部については、word@coins.tsukuba.ac.jpへのメール送信、Twitter アカウント [@word_tsukuba](https://twitter.com/word_tsukuba)への DM 等により問い合わせが可能です。

*1 各活動の勧誘広告を兼ねていたりいなかつたりするという噂も……

*2 電子版で読んでいる場合は手に持っていない可能性もありますが。

*3 過去には情報科学類の広報誌と呼ばれたこともあるとかないとか。

*4 <https://www.word-ac.net/post/2022/0605-word51/>

3 全代会 情報処理推進特別委員会

全代会とは全学学類・専門学群・総合学域群代表者会議の略で、筑波大学公式の学生組織です。全代会は中学校・高等学校での生徒会、高等専門学校での学生会に類するもので、大学と協同して学生の生活環境の改善などに取り組んでいます。全代会は、主としてクラス代表者会議から選出された学類の代表^{*5}により構成されますが、学類からの選出を経ずとも「専門委員」として活動に携わることができます^{*6}。

情報処理推進特別委員会は2023年1月18日に設立された全代会の特別委員会です。情報処理推進特別委員会の英名称はInformation-technology Promotion Committeeであり、その頭文字を取ってIPCという略称で呼ばれます。

IPCの活動目的は、「本学の情報基盤の整備及び情報関連サービスの提供並びにこれらの活動の推進」とされています^{*7}。また、大学公式と共同のアプリケーション開発など、「大学の(比較的)公的な立場でしかできない活動をすること」も目的の一つです^{*8}。

IPCでは次のような活動を実施しています。

- 全代会の情報インフラ運用
- 新歓Web^{*9}の開発・運用
- 情報処理に関する行事の企画
- 情報技術者コミュニティ UNTIL. の運営
- アプリケーション開発・運用

IPCで企画した行事として、NTT東日本の「とう道」の見学会や情報技術に関するLT^{*10}会の運営が挙げられます。

全代会及びIPCについては、zdk@stb.tsukuba.ac.jpへのメール送信、Twitterアカウント@public_zdk又は@until_tsukubaへのDM等により問い合わせが可能です。

4 UNTIL.

UNTIL.は先述の全代会IPCが運営する、筑波大学に関する情報技術者コミュニティです。UNTIL.はUNiversity of Tsukuba Information-technology Lab.の略です^{*11}。

UNTIL.は、筑波大学の在学生、OB、その他筑波大学向けにサービスを提供する者などで構成されます。UNTIL.の目的は、技術者の交流、技術的な知見の共有、技術者とアプリケーション開発チームを繋ぐ、といったものです。UNTIL.の特徴として、アプリケーショ

*5新入生オリエンテーションなどで「クラス代表」に選出され、更に「クラス代表者会議」において「議長」あるいは「副議長」に選出されるという流れを経ます。

*6ただし、専門委員は本会議での議決権は有しません。

*7全学学類・専門学群・総合学域群代表者会議 令和4年度第7回本会議 議案書より。

*8UNTIL. LT 0x01 運営挨拶より。

*9<https://www.stb.tsukuba.ac.jp/~shinkan-web/>

*10LTはLightning Talkの略で、5分程度の短いスライド発表のこと。

*11UNTIL.のLはLabではなくLoreの略という説や、正規表現でL.*と表されるもの全てという説もあります。

ン開発やインフラ運用などが目的ではなく、交流することやコミュニティを形成すること自体が目的という点が挙げられます。このため、コミュニティに所属するにあたっての敷居が、他の組織より比較的低いといえます。

UNTIL. では LT 会 UNTIL. LT の開催や、構成員主体でのアプリケーションの協同開発といった活動が行われています。UNTIL. LT は主として UNTIL. 構成員向けの行事ですが、UNTIL. 構成員以外が参加することも可能です。UNTIL. LT の運営は全代会 IPC が担っています。

UNTIL. については、Twitter アカウント @until_tsukuba への DM 等により問い合わせが可能です。また、<https://until-tsukuba.github.io/> からも情報を確認できます。

5 学実委 情報メディアシステム局

学実委とは学園祭実行委員会の略で、筑波大学の学園祭である雙峰祭を企画・運営する、全代会の下部組織です。

情報メディアシステム局は、学園祭実行委員会の部局の 1 つです。略称では jsys と表記され、情シスと呼称されます。

jsys では、雙峰祭の運営に関する次の活動を実施しています*12。

- システムの開発及び情報基盤の維持・管理
- 各種 Web サイトの制作・管理
- 各種映像の制作
- 後夜祭・対面ステージ企画の生中継
- オンラインステージ企画の収録・生配信
- 生中継・動画配信に伴う著作権の管理

学実委及び jsys へは、info@sohosai.com へのメール送信、Twitter アカウント @sohosai への DM 等により問い合わせが可能です。

6 さいごに

興味のある活動は見つかったでしょうか。見つかったようであれば、是非各団体に問い合わせ、見学や質問などを行ってみてください。そして、各活動に携わってみてください。

本記事で紹介した活動は、学内で行われる活動のほんの一部です。ここで紹介したもの以外にも、筑波大学生向けアプリケーションの開発やコンテストへの出場など、様々な活動が存在します。

このような課外活動を通じて、皆さんの大学生活が実りあるものとなることを祈っています。

*12 第 48 回筑波大学学園祭「雙峰祭」学園祭実行計画書より。

情報科学類誌

WORD

From College of Information Science

入学祝い号 2023

発行者	情報科学類長
編集長	伊藤祐聖
	筑波大学情報学群
	情報科学類 WORD 編集部
制作・編集	(第三エリアC棟212号室)

2023年4月7日 初版第1刷発行

(256部)