

2024

秋号

百萬石

Hyakumangoku Vol.63



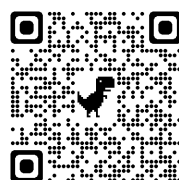
部長挨拶

gae

gae@mma.club.uec.ac.jp

「百萬石」をお手に取っていただき、ありがとうございます。MMA 部長の gae です。弊誌 (以下、部誌) は、電気通信大学の公認サークル MMA が新歓期 (春号) と弊学が開催する調布祭時期 (秋号) の年 2 回発行・頒布しているものです。MMA とは、"Microcomputer Making Association" の頭文字に由来しており、1975 年に創立されたサークルです。現在ではコンピュータ・パソコンに関連することを中心に、ネットワークやセキュリティなどの分野でも活動しています。

この部誌には、MMA の部員が行なっている活動を記事にしたものが掲載されています。内容は多岐に渡り、高度なプログラミングの話から旅行記など、MMA の活動の自由さが感じられるものとなっております。「秋号」は調布祭に頒布されるものです。これまでの歴代の部誌がすべて pdf で読めるようになっている、部員が作成したサイトもあります。先輩方の歴史とともに楽しみたいだけだと思います。



今回は秋号ということで、高校生も多く読むでしょう。そこで、電通大志望の高校生向けに少し話しましょう。電通大は、実は国公立なんですね。高校生の皆さん知ってました？なので、一般受験では共通テストと 2 次試験を受ける必要があります。そして、2025 年からはそのどちらにも情報 (2 次は選択) が追加されます。さすが電通大ですね。たしか 2 次試験で情報を選択できることを初めて発表したのは電通大だったと思います。

さあ、そんな中で MMA です。電気通信大学の中で、さらにプログラミングやサーバーを扱いたいという計算機だいすきといった人たちももちろんいますが、毎年初心者の方も数多く入部しています。むしろほとんどの新入部員が初心者です。プログラミングは大学から始めても全然大丈夫です。特に MMA に入っていれば、周りにプログラミングを教えてくれる先輩がたくさんいます。同級生同士でも教え合うことができます。

活動は**計算機 (コンピュータ)**を使っていれば、**活動内容は完全に自由です**。部内でもさまざまな活動が行われているため、誰が何をしているか完全に把握している人はいないでしょう。それぞれが好きに自分のやりたいことをやっています。私も電通大、そして MMA に入ってからプログラミングを始めた 1 人です。そんな人がいま 2 年連続で部長をできているのだから MMA は勉強にとってもいい環境が揃っており、それを十分に活かすことができれば、プログラミングをしっかりと学び楽しむことができるでしょう。2 年前まで何もやったことがなかった私が保証します。

MMA の宣伝はこの辺にして、MMA の近況です。9/20～9/22 の3日間、静岡に合宿に行ってきました。「MMA の合宿なんて、どうせプログラミング三昧なんでしょ？」と思ったそのあなた。全く違います!!!!MMA の合宿は**花火合宿**です。線香花火でも手持ち花火でもなく、打ち上げ花火です。2年生2人に花火の免許を取ってもらい、関係各所に連絡をしてやっと打ち上げられました。2年連続で大きな事故もなく、無事に終わることができました。下の写真はその様子です。(2023年の写真です。今年は打ち上げるときに風が強くてカメラのピントが合わなかった... ごめんなさい...)



最後になりましたが、MMA は**自分のやりたいことを実現できる場**です。充実した環境、助けになる先輩・同級生たち、こんな場所で活動する部員たちの部誌を読んでみてください。難しいものもあるとは思いますが、きっと面白いものもあります。ぜひご一読ください。

2024 年 11 月吉日



百萬石 Vol.63

Hyakumangoku Vol.63

2024 Autumn

MMA

目次

第1章	歌広場ダイヤモンド会員への道	Udon	1
1.1	はじめに		1
1.2	料金比較		1
1.3	歌広場の良いところ		3
1.4	歌広場の会員制度		3
1.5	ダイヤモンド会員を維持した方法		5
1.6	おわりに		6
1.7	おまけ		7
	参考文献		7
第2章	打ち上げ花火が失敗しそうになった	namor	8
2.1	はじめに		8
2.2	打ち上げ花火		8
2.3	おまけ		10
	参考文献		11
第3章	部内用マイクラサーバを立てた話	hydrogen	12
3.1	はじめに		12
3.2	動機		12
3.3	サーバの立て方		12
3.4	運用		15
3.5	おわりに		15
第4章	チーム分け自動化しよう	koshihkari	16
4.1	はじめに		16
4.2	要件		16
4.3	遺伝的アルゴリズム		17
4.4	試してみた		18
4.5	おわりに		18

4.6	おまけ	19
第 5 章	サーバーへの攻撃を可視化	daiki 20
5.1	はじめに	20
5.2	前提	20
5.3	elasticsearch, kibana, logstash を用いた可視化	20
5.4	webGL を用いた可視化	21
5.5	今後の展望	22
5.6	おわりに	22
第 6 章	SE 初心者がシステム大幅更新した話	shiragi 24
6.1	はじめに	24
6.2	OS サポート切れまで 1 ヶ月しかないじゃないか！	24
6.3	フリーダムすぎたサーバーたち	25
6.4	リモートワークってやりづらい...	26
6.5	VLAN？なにそれおいしいの？	26
6.6	レガシーとエラーの戦い	27
6.7	まだまだ終わらない更新作業	31
6.8	おわりに	31
第 7 章	ロシア語を学ぶ	nasu 33
7.1	はじめに	33
7.2	なぜロシア語なのか	33
7.3	ロシア語って難しいの？	34
7.4	ロシア語能力検定 4 級に挑む	36
7.5	まとめと今後の展望	37

第 1 章

歌広場ダイヤモンド会員への道

Udon

1.1 はじめに

お元気ですか、Udon と申します。

みなさん、カラオケには行っていますか。大学生といたらカラオケって感じしますよね。我が MMA でも部会後などにカラオケに行くことが多いです。

電通大の周りにはいくつかカラオケ店があります。有名なチェーンでいうと、ビッグエコーとかまねきねことかがあるわけです。

ですが、今回は歌広場をおすすめしたいと思います。それとともに、歌広場の会員制度についても紹介したいと思います。

1.2 料金比較

ではまず、料金を比較しましょう。電通大周辺のカラオケチェーン店で料金を比較してみました。全て税込、歌広場はすべてソフトドリンク飲み放題プランの料金です。

表 1.1 料金比較 1

	平日朝うた (9:00-12:00)	週末朝うた (7:00-12:00)
歌広場 (学割)	×	×
歌広場 (一般)	×	×
まねきねこ (会員、複数人)	55	11
まねきねこ (会員、1 人)	55	55
ビッグエコー (会員、複数人)	×	×
ビッグエコー (会員、1 人)	×	×
ビッグエコー (学割)	×	×

表 1.2 料金比較 2

	平日昼 (18:00 まで) 30 分	平日昼フリータイム
歌広場 (学割)	190	1130
歌広場 (一般)	250	1450
まねきねこ (会員、複数人)	165	990
まねきねこ (会員、1 人)	275	1540
ビッグエコー (会員、複数人)	180	990
ビッグエコー (会員、1 人)	240	1320
ビッグエコー (学割)	250	1380

表 1.3 料金比較 3

	週末昼 (18:00 まで) 30 分	週末昼フリータイム
歌広場 (学割)	250	1500
歌広場 (一般)	320	1980
まねきねこ (会員、複数人)	275	1540
まねきねこ (会員、1 人)	385	2090
ビッグエコー (会員、複数人)	310	1980
ビッグエコー (会員、1 人)	370	2300
ビッグエコー (学割)	290	1890

表 1.4 料金比較 4

	平日夜 (18:00 から) 30 分	平日夜フリータイム
歌広場 (学割)	350	550
歌広場 (一般)	450	1800
まねきねこ (会員、複数人)	363	1925/770(まふ)
まねきねこ (会員、1 人)	473	2475
ビッグエコー (会員、複数人)	430	2150
ビッグエコー (会員、1 人)	480	2300
ビッグエコー (学割)	450	2250

表 1.5 料金比較 5

	週末夜 (18:00 から) 30 分	週末夜フリータイム
歌広場 (学割)	400	1320
歌広場 (一般)	580	2500
まねきねこ (会員、複数人)	495	2750/1300(まふ)
まねきねこ (会員、1 人)	605	3300
ビッグエコー (会員、複数人)	520	2600
ビッグエコー (会員、1 人)	570	3420
ビッグエコー (学割)	500	2500

ここで注意なんですけど、歌広場とビッグエコーの学割以外では料金にドリンク代が含まれていません。ドリンク代が別途かかるので、実際にはもう少し高くなります。+250 円くらいと考えたらいいかもです。

そう考えると、歌広場が総合的に一番安いって感じです。

1.3 歌広場の良いところ

なんといっても「料金にドリンク代が含まれている」ということでしょう。他のカラオケ店だと、ドリンク代が別途かかることが多いです。しかも、ワンドリンク制なのに割と高い、ということもあります。

歌広場では、料金にドリンク代が含まれているほか、デフォルトでソフトドリンク飲み放題なので、好きなだけ飲むことができます。ドリンク代のコスパは非常に良いといえます。

また、ドリンクバーということなので、好きなドリンクを選べるほか、歌ってる時に店員さんが飲み物を持ってきて気まづくなることもないです (ドリンクに限った話。食べ物頼んだ場合は運んできてくれる)。

また、一人カラオケが問題なくできることも大きいです。まねきねことかビッグエコーだと一人カラオケで料金が上がったりするんですね (世知辛い)。歌広場は一人カラオケでも料金が変わらないので、気軽に行くことができます。というかそうじゃないとダイヤモンド会員を維持するのは厳しいです (そこまで高頻度に付き合ってくれる人がいない)。

ただ、まねきねこ違って WiFi がない、などといったことはあります。

1.4 歌広場の会員制度

歌広場には会員制度があります。アプリ会員 (年会費無料) になるだけで料金が 5%OFF になるわけですが、それに加えてシーズンごとの来店回数に応じてランクが上がるシステムとなっています。シーズン終了時、どれくらい来店したかによって次のシーズンのランクが決まります。

2024 年 11 月現在、以下のようにランクアップするシステムになっています。

1. レギュラー会員：登録時から、5%OFF

2. ブロンズ会員：6 回以上、10%OFF
3. シルバー会員：12 回以上、15%OFF
4. ゴールド会員：20 回以上、20%OFF
5. プラチナ会員：25 回以上、25%OFF
6. ダイヤモンド会員：30 回以上、30%OFF

これに従って、上記の表でいうところの料金が下がっていく、ということになります。

ここで来店回数についてなんですが、1 日に 2 回以上来店しても、たとえ違う店だったとしても 2 回分カウントはされないみたいです。よって、「来店回数」というよりは「来店日数」と考えたほうがいいです。

注目したいのは、ダイヤモンド会員になると、料金が 30%OFF になるということです。これは非常に魅力的に感じます。

まねきねこなども会員制度があり、ダイヤモンドまで行くと 30%OFF になるんですが、やはり元の料金に差があるので、同じ割引率なら歌広場のほうが安い、となります。

ここで注意なんですが、学割と会員ランクによる割引は併用することができません。どちらか一方を選ぶことになります。

[1] をもとに計算したところ、夜のウタ学フリータイムを除けば「金・土・祝前日の 18 時以降」以外の時間帯では、プラチナ会員以上の割引料金 (25%OFF) を用いないと学割料金よりも安くならないことがわかりました。

なので、プラチナ会員になるまでは、学割料金を使ったほうが良いです。幸い、学割料金を使ったとしても来店回数は問題なくカウントされます。

また、夜のフリータイムを用いる場合なら、学割料金のほうがダイヤモンド会員料金よりも安くなります。というか夜は 1 時間以上の場合フリータイムの方が安くなります。

大学生で、あんまりカラオケに行かないっていう人なら、学割料金のままでも大して損はしないので、学割だけ使っていてもそんなに損はないと思います。ただ、カラオケが好きで頻繁に行くなら、ダイヤモンド会員まで頑張るとよりお得に歌うことができる、といった感じだと思います。



図 1.1 会員証

1.5 ダイヤモンド会員を維持した方法

2024/07/01	カラオケルーム歌広場 調布天神通り店	▼
2024/07/03	カラオケルーム歌広場 調布天神通り店	▼
2024/07/05	カラオケルーム歌広場 調布天神通り店	▼
2024/07/08	カラオケルーム歌広場 立川南口駅前店	▼
2024/07/09	カラオケルーム歌広場 調布天神通り店	▼
2024/07/10	カラオケルーム歌広場 立川南口駅前店	▼
2024/07/12	カラオケルーム歌広場 調布天神通り店	▼
2024/07/14	カラオケルーム歌広場 立川南口駅前店	▼
2024/07/15	カラオケルーム歌広場 調布天神通り店	▼
2024/07/19	カラオケルーム歌広場 調布天神通り店	▼
2024/07/23	カラオケルーム歌広場 調布天神通り店	▼
2024/07/29	カラオケルーム歌広場 調布天神通り店	▼
2024/07/31	カラオケルーム歌広場 立川南口駅前店	▼
2024/08/02	カラオケルーム歌広場 立川南口駅前店	▼
2024/08/05	カラオケルーム歌広場 調布天神通り店	▼
2024/08/06	カラオケルーム歌広場 立川南口駅前店	▼
2024/08/08	カラオケルーム歌広場 立川南口駅前店	▼
2024/08/11	カラオケルーム歌広場 調布天神通り店	▼
2024/08/13	カラオケルーム歌広場 池袋西武口店	▼
2024/08/14	カラオケルーム歌広場 立川南口駅前店	▼
2024/08/15	カラオケルーム歌広場 調布天神通り店	▼
2024/08/19	カラオケルーム歌広場 立川南口駅前店	▼
2024/08/20	カラオケルーム歌広場 立川南口駅前店	▼
2024/08/22	カラオケルーム歌広場 立川南口駅前店	▼
2024/08/24	カラオケルーム歌広場 立川南口駅前店	▼
2024/08/26	カラオケルーム歌広場 調布天神通り店	▼
2024/09/11	カラオケルーム歌広場 調布天神通り店	▼
2024/09/12	カラオケルーム歌広場 調布天神通り店	▼
2024/09/18	カラオケルーム歌広場 調布天神通り店	▼
2024/09/25	カラオケルーム歌広場 仙台クリスロード店	▼
2024/09/29	カラオケルーム歌広場 調布天神通り店	▼

図 1.2 7～9 月

ダイヤモンド会員になるには、1 シーズンで 30 回以上来店する必要があります。1 シーズンは 3 か月なので、だいたい 3 日に 1 回来店する必要があります。また、1 シーズン達成したら永続というわけではなく、次のシーズンも同じように 30 回来店する必要があります。次のシーズンから見た「次のシーズンのランク」は、そのシーズンの来店回数によって決まるので、0 回来店だと一気にレギュラー会員に落ちます。

そう聞くと結構ハードルが高い気がします。というわけで、参考までに自分がダイヤモンド会員になり、それを維持した方法を紹介します。なんか取り立てて言うべきことでもない気がしますけどまあいいです。

とにかく、「たくさん行くためにはそれなりのきっかけが必要」といった感じです。

1.5.1 30 分だけでも行く

歌広場は 30 分だけ歌うということができます。なので、空き時間があったら 30 分だけでも行くようにしていました。だいたい 6 曲くらい歌えます。

空きコマがある場合はそこを利用していました。また、時間に余裕があるなら 1 時間くらい歌ってました。

1.5.2 一人カラオケをする

一人のほうが基本的には予定が柔軟に組めます。というか 30 分だと複数人の場合ほとんど歌う時間がないので、短時間だけ行くときはたいてい一人でした。これは一人カラオケが追加料金なしでできる歌広場で有効な手段だと思います。ぼっちに優しい！

1.5.3 夜フリータイムを使う

歌広場の夜のフリータイムは最長で 11 時間入ることができます。その間歌い放題、飲み放題、アイスクリーム食べ放題なので、学割料金 (ウタ学) を使うとすさまじくコスパがいいです。

複数人で来ても時間をあまり気にすることがなく、ゆっくり歌うことができます。また、空き時間でゆっくりできます。

また、別にカラオケルームでカラオケばかりしなければならないわけでもないのです。一人で来た場合は自習室として使ったりもしていました。たまに隣の部屋から歌ってるのが聞こえますが、それを除けば結構コスパも環境も良いのではないのでしょうか (充電もできるし)。

1.5.4 生活の一部にする

毎シーズンで 3 日に 1 度行かなきゃならんので、結局これを意識することが一番大切になってきます。カラオケが大好き！生活！！って人はダイヤモンド会員を目指すとういと思います。

また、採点を入れておくと、来て歌うたびに自分の点数が上がっていった楽しくなるので、おすすめです。

1.6 おわりに

繰り返しになりますが、電通大周辺なら歌広場が一番コスパが良いと思います。

人数関係なく、頻繁にカラオケに行く！という人はここでダイヤモンド会員を目指してみるのもいいと思います！たまにしかカラオケに行かない、という人でもとりあえずここに行っておくと安く歌えるかなと思います。

今更なんですが、この記事は別に企業案件とかではないです。あと、副部長なのに技術系のこと書かないのかよ！と思うでしょう。ごめんなさい (一応何個かネタはあった)。以前 hydrogen くんが「MMA

はカラオケサークル」と発言していたので、それを受けてこの記事を書きました。MMA の雰囲気を感じてもらえたら嬉しいな〜という感じです。

ではまた次の部誌でお会いしましょう。次はさすがに技術的なことを書きます。

1.7 おまけ

各部員が歌っている曲の傾向をまとめてみました。

- gae : J-POP(菅田将暉とか優里とか) - Udon : アニソン (ガンダムが主)、特撮ソング、東方アレンジ、ずとまよ- namor : J-POP(嵐とかミセスとか) - hydrogen : アニソン、ゲームの曲- kitatai : King Gnu(めちゃくちゃ上手い) - will : ボカロ曲、洋楽- akitoshi : 昭和の歌謡曲

また、今年の5月からカラオケ記録をつけてるんですが、10月終了時点で以下のような感じです。

	平均人数	時間合計(分)	価格合計(円)	曲数合計	1分当たりの金額(円/分)	1曲当たりの金額(円/曲)
全体	1.722222222	8215	31500	865	3.834449178	36.41618497
2024/05		1530	6770	213	4.424836601	31.78403756
2024/06		2040	5500	172	2.696078431	31.97674419
2024/07		1425	5700	173	4	32.94797688
2024/08		1230	6600	121	5.365853659	54.54545455
2024/09	2.4	920	2195	40	2.385869565	54.875
2024/10	1.5	1040	4560	140	4.384615385	32.57142857

図 1.3 記録

参考文献

[1] <https://www.utahiro.com/store/storelist/storeinfo/tabid/64/pdid/0131/Default.aspx>

[2] <https://www.karaokemanekineko.jp/asset/cms/price-pdf/765.pdf>

[3] https://big-echo.jp/shop_info/%E3%83%93%E3%83%83%E3%82%B0%E3%82%A8%E3%82%B3%E3%83%BC%E8%AA%BF%E5%B8%83%E6%9D%B1%E5%8F%A3%E9%A7%85%E5%89%8D%E5%BA%97

[4] <https://cstar.karaoke-foryou.com/%e6%96%99%e9%87%91%e3%82%b7%e3%82%b9%e3%83%86%e3%83%a0/>

第 2 章

打ち上げ花火が失敗しそうになった

namor

2.1 はじめに

今年も無事に合宿に終えることが出来ました。今年度合宿担当だった ndk くん, お疲れ様でした! 個人的にはパノラマパークからの景色がはっきり見え, とても良かったです。

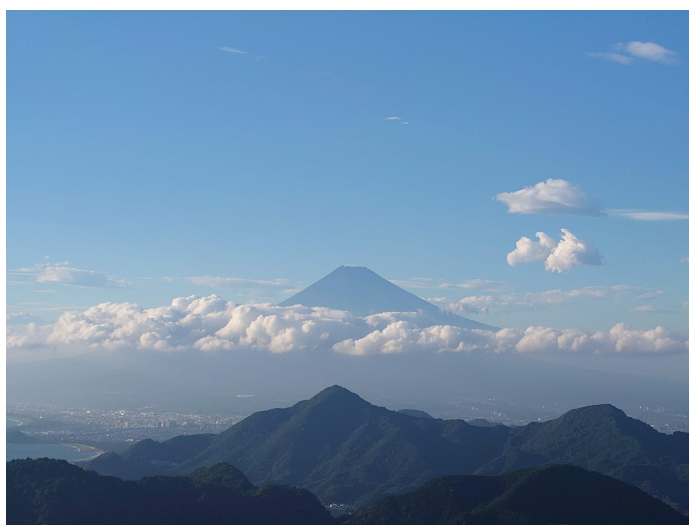


図 2.1 パノラマパークからの様子

2.2 打ち上げ花火

合宿最終日の夜に恒例の打ち上げ花火を行いました。そこでちょっとしたトラブルがあったので残したいと思います。

4 発目の花火を打ち上げる際に, 打ち上げボタンを押しても花火が打ち上がらないという事態が発生しました。この場合, ロングヒューズには引火していると打ち上げ火薬や花火玉に火が入っている

可能性があります。いつ爆発するかわからないことから、まずは時間をおいて様子を見ました。しばらく観察して点火の気配がないことを確認し、OBの今田さん協力のもと消火器を持っていきロングヒューズ引火しているかどうかを見ることにしました。あやしがりて寄りて見ると、電気導火線は燃えていましたがロングヒューズに点火している様子はありませんでした。

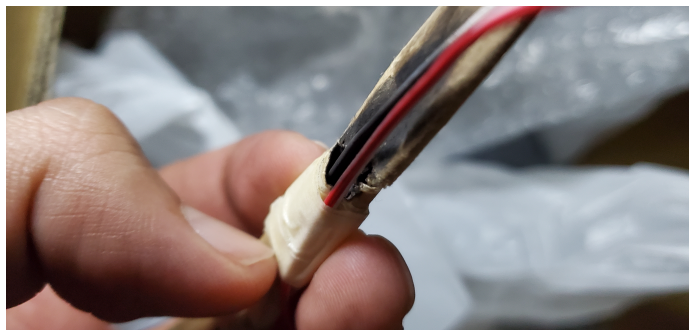


図 2.2 燃えた電気導火線

原因は、電気導火線をロングヒューズに接続する際に切り込みを入れてヒューズの火薬に押し込むようにしていたのですが、その切り込みが浅かったために火薬に接触していなかったことでした。その後、燃えた電気導火線を予備の新しい電気導火線に交換し、5発目の花火を打ち上げることができました。予備の電気導火線は必須でした。



図 2.3 実際の様子

2.3 おまけ

ビジュアル栄える展示物を作るべく花火のシュミレーターを作ってみました。ボタンを押すと花火が打ち上がる映像が流れるものです。以下にその構成と作成したものを紹介します。

2.3.1 使用したもの

- Touch Designer
- ESP32
- 赤いボタン

Touch Designer は、ノードを繋げることでプログラムを作成することができるソフトウェアです。これで音声やその他入力に対してリアルタイムにイケイケな映像を作成することができます。実際に作成したものは以下の通りです。



図 2.4 Touch Designer のノード

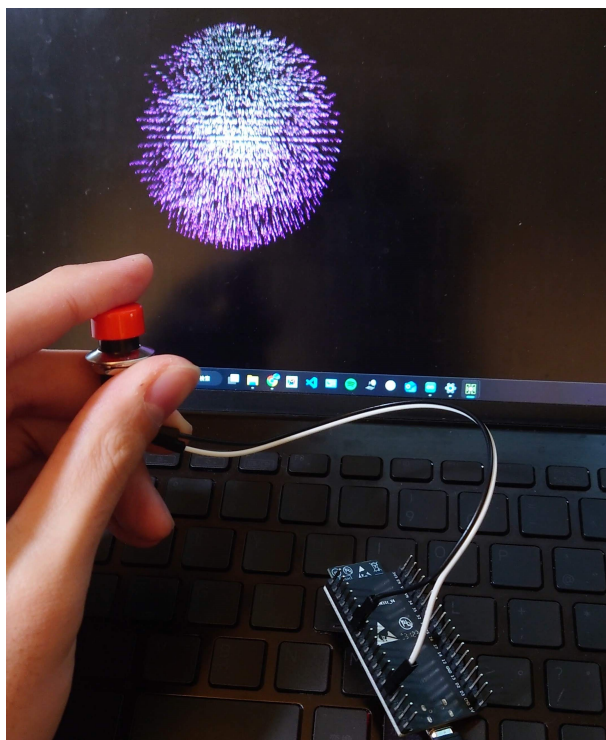


図 2.5 打ち上げ花火シュミレーター

??を参考に打ち上げ時のエフェクトと音声を追加しました。ESP32 から Bluetooth でキーボードの入力を送り、Touch Designer で特定のキーが押されたときに花火を打ち上げるようにしました。構成自体はとてもシンプルで簡単に映える展示が出来ました笑。ボタンとマイコンをしまう箱を作った完成です(部誌執筆時に間に合わず)。

参考文献

- [1] ぎまパパチャンネル.[TouchDesigner Particle Fireworks[パーティクルで花火]]. YouTube. 2024/11/07 <https://youtu.be/GH73XIP9so8?si=hdJaHwT6JTG7nUOT>

第 3 章

部内用マイクラサーバを立てた話

hydrogen

3.1 はじめに

こんにちは、3 年にもなってようやく部誌を書き始めた hydrogen といいます。もっぱら水素と呼ばれています。今回は、部内でマイクラサーバを立てた時のあれこれや、その後の運用について書いていきます。

3.2 動機

Minecraft 自体は Java 版の 1.7 の時代からやっていたが、サーバを立てるのは初めてのことであった。以前から、身内だけで遊ぶためのサーバを立てたいと思っていたものの、自分の懐が芳しくなく、管理できるような暇もなかったのではなかなか実現できなかった。しかし、3 年にもなりある程度資金に余裕が生じたことと、後輩とも遊びたいという気持ちがあったので、テスト終了後にサーバを立てることにした。

3.3 サーバの立て方

3.3.1 VPS の選定

まず、Minecraft サーバはその性質上 VPS のような自由度の高いサーバで運用する必要があるため、各社の VPS を比較検討した。その結果、初回料金で比較するとある程度の性能で一番安いからという理由で ConoHa Game の 8GB プランを使うことに決定。今のところ、性能面で不満はない。

3.3.2 サーバの構築

サーバの構築は、ConoHa Game のマイクラサーバ構築のページに従って行った。ただ、ConoHa が提供しているサーバー管理ツールは導入せず、SSH で直接操作することにした。サーバーラッパーとして、Paper を採用し `server.properties` を次の通りに設定した。

```
...  
difficulty=hard
```

```
enable-command-block=true
...
enforce-secure-profile=false
...
gamemode=survival
...
view-distance=24
white-list=false
```

3.3.3 統合版でもログイン可能にする

GeyserMC というプラグインを導入することで、統合版からでもサーバにアクセスできるようになる。ただ、このプラグイン単体では Java 版のアカウントが必要になるので、統合版アカウントのみでログインできるようにするため、Floodgate というプラグインを導入した。

ただ、このままでは統合版ユーザーがネザー岩盤上で何もできなくなってしまうので、config.yml を編集して許可するようにする必要がある。

```
...
allow-bedrock-nether-building: true
...
```

3.3.4 Dynmap の導入

次にマッププラグインとして Dynmap を導入した。これの導入に非常に手間取ったので、導入方法を以下に示す。

Dynmap の jar ファイルを plugins フォルダに入れたのち、ufw でポートを開放し、SSL に対応させるため apache2 をインストールする。

```
$ sudo ufw allow http
$ sudo ufw allow https
$ sudo apt install apache2
```

Let'sEncrypt を使って SSL 証明書を取得するため、certbot をインストールし証明書を取得する。

```
$ sudo apt install certbot python3-certbot-apache
$ sudo certbot --apache
```

これだけでは、証明書が自動更新されないなので、cron で毎月の頭に自動更新するように設定する。

```
0 * 1 * * certbot renew && systemctl restart apache2
```

次に、apache2 の設定ファイルを編集する。

```
<VirtualHost *:80>

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    ErrorLog ${APACHE_LOG_DIR}/error.log
```

```

CustomLog ${APACHE_LOG_DIR}/access.log combined

RewriteEngine on
RewriteCond %{SERVER_NAME} =<SERVER_NAME>
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=
    permanent]

ProxyPass /dynmap/ http://localhost:8123/
ProxyPassReverse /dynmap/ http://localhost:8123/
</VirtualHost>

```

```

<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{REQUEST_URI} /dynmap$
RewriteRule /* %{REQUEST_URI}/ [R=301,L]
</IfModule>

<IfModule mod_ssl.c>
<VirtualHost *:443>

    DocumentRoot /var/www/html

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    ServerName <SERVER_NAME>
    SSLCertificateFile /etc/letsencrypt/live/<SERVER_NAME>/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/<SERVER_NAME>/privkey.
        pem
    Include /etc/letsencrypt/options-ssl-apache.conf

    ProxyRequests Off
    ProxyPass /dynmap http://localhost:8123
    ProxyPassReverse /dynmap http://localhost:8123
</VirtualHost>
</IfModule>

```

これで、`https://<SERVER_NAME>/dynmap` にアクセスすることで SSL 対応の Dynmap が見れるようになる。

ただ、余計なマップも表示されてしまうので、次のコマンドを実行して不要なマップを生成しないようにしたうえで、別のマップを追加する。

```

/dmap pause all
/dmap purgemap MAIN:cave
/dmap addmap MAIN:surface2
/dmap pause none
/dmap fullrender MAIN

```

3.4 運用

3.4.1 サーバー始動後 3 日目、荒らしあらわる

こうして、サーバーを立てた後 3 日目に荒らしが発生した。その時は、偶然にも僕がログインしていたため家が燃やされる程度の被害で済んだものの、これを機にいくつかの対策を講じることとした。

まず、GeoLite2 というデータベースを使って IP アドレスから国を特定し、ホワイトリストに載っていないプレイヤーの日本国外からの接続を禁止するプラグインを作成した*1。

このプラグインは不審な接続を検知すると接続を拒否し、その旨を Discord に通知する。これにより、荒らしが発生することがほぼなくなった。



図 3.1 実際の動作

3.5 おわりに

今回は、部内でマイクラサーバを立てた話を書いてみた。このサーバーは今でも稼働しており、部員ならだれでも遊ぶことができる。平均ログイン人数は日に 1 から 2 人程度だが、たまに 5 人以上がログインすることもある。来年の夏までは確実に運用を続ける予定である。

*1 ソースコードは <https://github.com/hydrogend/excludeaccess> にて公開している

第 4 章

チーム分け自動化しよう

koshihkari

4.1 はじめに

こんにちは、部室では滅多に見ない koshihkari(こしひかり) と申します。部誌を書くのは 2 回目になります。今回はスプラトゥーンの大会のチーム分けを自動化した話をしようと思います。スプラトゥーンでは主に 4 人 vs4 人で戦うということがわかっていればゲームについてよく知らなくても大丈夫です。今年はスプラトゥーンの大会を運営する機会が多くあり、その中でもチーム分けに時間がかかることがあったので自動化したいと思い、取り組みました。

4.2 要件

大会では参加者全員に楽しんでもらえるようにできるだけ各チームの強さを均等にすることが重要です。スプラトゥーンでは X パワーと呼ばれるプレイヤーの強さを表す数値があり、主にこの数値を用いてチーム分けをします。また、これに加えてチームの各プレイヤーのポジションも重要です。あるチームに同じポジションのプレイヤーが集まってしまうとポジションに偏りのないチームより弱くなってしまうし、何より楽しめなくなってしまう。そのためポジションについても考慮できるものを作る必要があります。これは全ての組み合わせを調べることで解決しそうですが、大会の参加者は大抵 16 人～24 人でありこれより多い時もあります。無駄が多いですが特に工夫などせず考えると参加者数の階乗だけの計算量になってしまうのでこれも解決しなければなりません。また 1 つのチーム分けの候補だけではなく複数の候補を見比べて良いと思ったものを採用する形にしたいとも思っていたので、これも実現できるようにします。長くなってしまったので簡単にまとめておきます。

- チームごとの X パワーの差が小さくなるようにする
- プレイヤーのポジションを考慮する
- ある程度速く動作させる
- 複数の組み合わせの候補を提示する

これを満たせそうなアルゴリズムを考えたり調べたりした結果遺伝的アルゴリズムが良さそうだった

のでこれで作ることにしました。

4.3 遺伝的アルゴリズム

遺伝的アルゴリズムはみなさん知っていますでしょうか。調べたら詳しく解説してくれているサイトが多くあるので細かくは書きませんが簡単に説明すると、データを遺伝子に見立てて選択や交叉、突然変異を用いて解を探索していくアルゴリズムのことを指します。良い個体を選択して交叉することで最適解へと近づけていきます。

4.3.1 個体の評価

今回はプレイヤーを遺伝子、その列を個体と考え前から 4 人ずつを 1 チームとして扱いました。良い個体を選択するには個体の良さを評価しなくてははいけません。ポジションよりもより影響の大きい X パワーを軸に評価を行いました。

```
def calc_power_score(self, team):
    average = self.players.calc_average_power()
    team_average = self.calc_team_average_power(team)
    return abs(average - team_average) * self.POWERPOINT
```

このような感じで参加者全体の平均パワーとチームの平均パワーとの差の絶対値に設定した定数を掛けた値を減点していく形にしました。ポジションに関しては影響の大きい後衛被りのみを設定した値だけ減点という形にしました。ここで、次に説明する選択を行うために 10000 点からの減点としました。

4.3.2 選択

今回はルーレット選択と呼ばれる選択方法を採用しました。各個体の評価値を基に良い個体を選ばれる確率が高くなるようにして二つの個体を選び、交叉を行います。

```
def rourate_select(self, generation):
    scores = [individual.get_score() for individual in generation]
    weights = [score / sum(scores) for score in scores]
    selected = random.choices(generation, weights=weights, k=2)
    return selected.copy()
```

評価値を基に重み付けをする際に評価値が負の値になってしまうと面倒なので 10000 点からの減点方式にしました。

4.3.3 交叉

交叉では選択した個体を掛け合わせてよりよくなるようにします。交叉のアルゴリズムの実装は結構悩みました。今回は選択された二つの個体でそれぞれ各プレイヤーの XP におけるチーム内順位を求め、個体間で順位が異なるプレイヤーだけをランダムに並び替えたものを新たな個体としました。言葉ではわかりにくいと思うので図も載せておきます。A,...,H の 8 人のプレイヤーはこの順に X パワーが高い

です。A,H,D,E の 4 名はチーム内の順位が二つの個体の中で同じであるので変更を加えず、他のプレイヤーの位置をランダムに並び替えます。



図 4.1 交叉の仕組み

4.3.4 突然変異

突然変異は似たような個体が多く残ってしまうことを防ぐために行います。あまり考えずに指定した確率でプレイヤーをランダムで 2 人選んでスワップするようにしました。

```
def mutation(self, generation):
    # 突然変異
    for individual in generation:
        # 定められた確率でランダムな位置で入れ替えを行う
        if random.random() < self.mutation_probability:
            i, j = random.choices(range(self.players.get_player_count()), k=2)
            individual.swap_genom(i, j)
```

4.4 試してみた

実はまだこれを作成してからチーム分けが必要な大会の運営を行っていないので活かせてはいないのですが、過去大会のデータを用いるとかなり良い感じにチーム分けができていたと思うので作ってよかったと思いました。しかしまだ少し処理に時間がかかったり、交叉で全くポジションについて考慮できていないことなど改善点も見つかったのでよりよくしていきたいと思います。本音を漏らすとグラフにして結果をわかりやすく伝えたかったのですが面倒なのでやめました。

4.5 おわりに

他にもこんなアルゴリズムとか良さそうだななどあれば教えていただけると嬉しいです。最後まで読んでいただきありがとうございました。

4.6 おまけ

実は交叉では何もせずに突然変異の確率を 1 とするとめっちゃくちゃ高速に割といい感じのチーム分けを行ってくれました。交叉で結構悩んだのでちょっと悲しかったです。。。

第 5 章

サーバーへの攻撃を可視化

daiki

5.1 はじめに

大変です!MMA のサーバーが攻撃を受けています! どのくらい攻撃されているかを可視化しましょう!

5.2 前提

MMA には数多くのサーバーがありますが、そのなかでも moon というサーバーの 22 番ポートはかなり攻撃を受けています。MMA にはすでに、攻撃元の ip アドレスから緯度経度を割り出し、その ip から攻撃回数などと併せて json に出力するシステムがありました。今回はこのデータをもとに、マップ形式でどこからどれだけ攻撃されたかを可視化しました。

5.3 elasticsearch, kibana, logstash を用いた可視化

検索エンジンである elasticsearch, データの編集や可視化に用いる kibana, それらにデータを転送する logstash を用いて可視化します。

まず、elastic 公式から elasticsearch, kibana, logstash のアーカイブをインストールします。elasticsearch と kibana の初期設定を終わめます。

今回は部内だけの用途なので、それぞれの config でセキュリティオプションを disable にします。(https だといろいろ面倒だった)

logstash の config で、json ファイルを読んで elasticsearch に送るように記述します。この config の要求する json フォーマットがかなり厳しく、コロンの後に空白があると駄目、ファイルの最後に改行がないと駄目、などかなり悩まされました。

```
input {
  file {
    path => "/path/to/json/your.json"
    start_position => "beginning"
    sincedb_path => "/dev/null"
    codec => "json"
  }
}
```

```
}

filter {
}

output {
  elasticsearch {
    hosts => ["http://elasticsearch:9200"]
    index => "ipdata"
    document_id => "%{ip}"
  }
  stdout { codec => rubydebug }
}
```

データが転送できていることを確認したら、kibana で map を作り、埋め込み用の iframe を出力させて、web サイトに張り付けて完成。

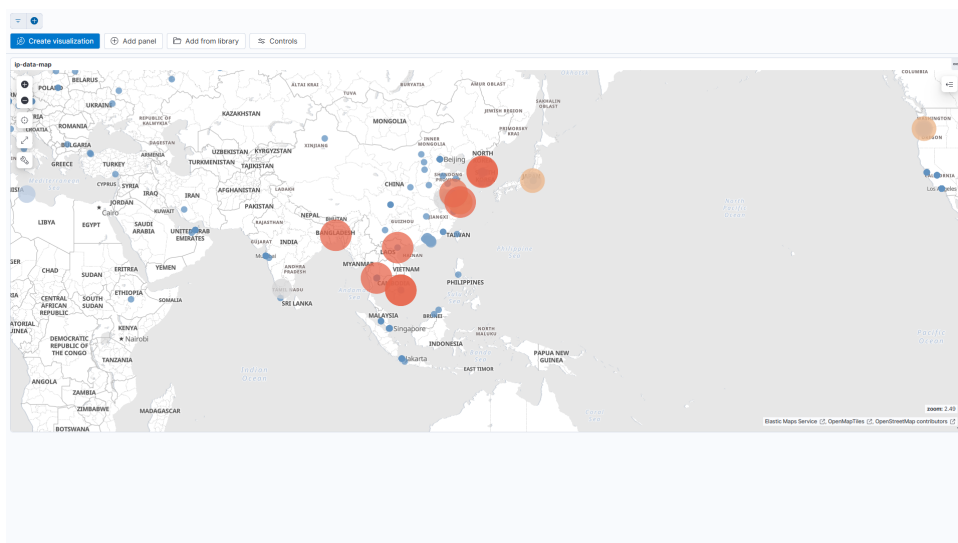


図 5.1 kibana-map

ここでの elasticsearch, kibana, logstash は systemd で daemon 化してバックグラウンドで実行しています。

5.4 webGL を用いた可視化

kibana map での可視化はすっきりしていて良い感じでした。しかし何かが足りない、それは何か。そう浪漫です。浪漫を追い求めるために、webGL(three.js) を用いて可視化してみます。

three.js は 3D 表現に長ける javascript ライブラリで、簡単に 3D 表現が実装できるらしいです。今回はインターネットの海に浮いていた example を拝借しました。

Google Data Arts チームによる WebGL GLobe というプラットフォームを利用します。3D の地球

上の任意の位置に magnitude を持った棒グラフを立てることができるようになっています。データとして json 形式で latitude, longitude, magnitude を並べたリストを与えます。

```
[  
  [latitude, longitude, magnitude, latitude, longitude, mag...]  
]
```

python で http サーバーを立て、GET リクエストがきたら攻撃データの json を WebGL Globe 用フォーマットに整形して返すコードを書きます。先ほどの kibana map を張り付けたサイトをホストしているサーバーから、python http サーバーに GET リクエストして、webGL Globe にデータを与えると...

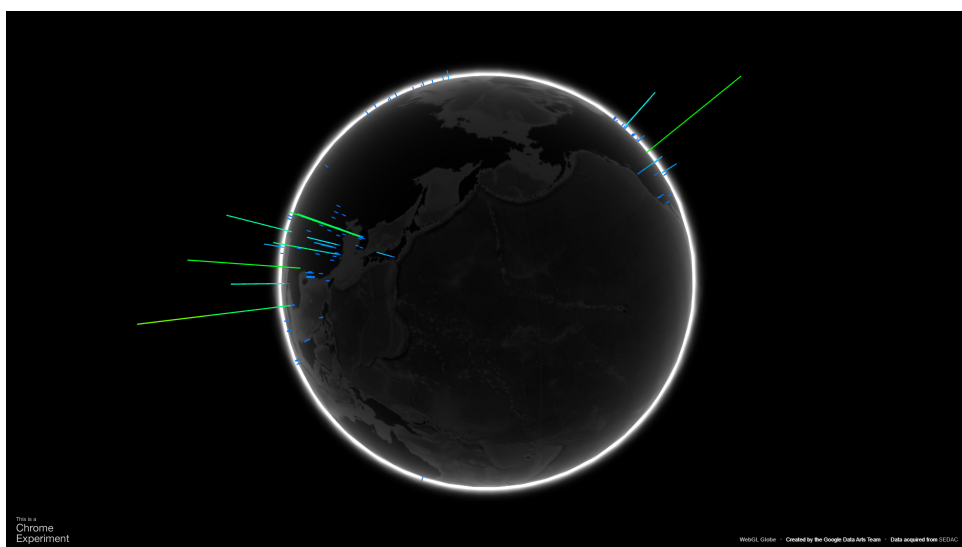


図 5.2 webgl-globe-map

良い感じにかっこよくなりました。

5.5 今後の展望

やはり攻撃地点から東京までの軌跡を描いてよりかっこよくしたいですね。kibana を使っているのでグラフ出力などもできそうです。また、メンテナンス性の向上のために systemd で行っているところを docker コンテナに置き換えていくつもりです。

5.6 おわりに

書いてみるとやったことは少ないですが、ここまでくるのにかなり苦労しました。作成に当たって、部長の gae, 部員の shiragi には仮想サーバーの sudo 権限を頂いたり proxmox の権限を頂いたりと大変お世話になりました。多大なる感謝をここに記します。

ご精読ありがとうございました。

第 6 章

SE 初心者がシステム大幅更新した話

shiragi

6.1 はじめに

はじめましての方ははじめまして。shiragi と申します。

最近はこの MMA のシステム管理をさせてもらっています新入部員です。(23 年度入学、24 年度入部です。)

入部した当初は「CTF やるぞ!」と意気込んでいたのですが、あれ、どうしてシステム管理やっているんだろう...?

そのわけは 2024 年 6 月に遡ります。

6.2 OS サポート切れまで 1 ヶ月しかないじゃないか!

MMA は、部員が共有で使えるサーバーが数多く設置されています。

例えば、ホームディレクトリに雑に置いたりするためのサーバー、wiki が動いているサーバー、Gitlab が動いているサーバーなどがあります。

私が入部した当初、そのサーバーのほとんどは、家庭用と何ら変わらない自作 PC で動いてました。かれこれ 6 年以上動き続けているそうで、だいぶホコリまみれでした。

ずっと動いていると、中身の OS も放置しているとどんどん古くなっていきます。そのときはなんと、ほとんどが Debian buster でした。Debian buster のサポート期限は 6 月末です。1 か月しかありません。

「別にサポート切れたって動くんだしいいじゃん」

ダメです。

サポート切れの OS がセキュリティ的に危ないのもそうですが、情報セキュリティ対策基準によれば、

第 111 条

4 部局運用責任者及び部局システム管理者は、脆弱性対策の状況の定期的な確認により、脆弱性対策が講じられていない状態が確認された場合並びにサーバ装置、端末及び通信回線装置上で

利用するソフトウェアに関連する脆弱性情報を入手した場合には、セキュリティパッチの適用又はソフトウェアのバージョンアップ等による情報システムへの影響を考慮した上で、ソフトウェアに関する脆弱性対策計画を策定し、措置を講ずること。

国立大学法人電気通信大学情報セキュリティ対策基準 (https://www.cc.uec.ac.jp/rule/情報セキュリティ対策基準-20240909_修正.pdf) より引用

サークルの部員は別に「部局運用責任者」ではありませんが、通念上「部局運用責任者から委任されている」と解釈されます。

つまり、ちゃんと対策せず、インシデントとか起こしてしまうと、サークル活動の存続にも影響を及ぼします。だから、対策しなければまずいのです。

身の回りのセキュリティが整っていないのに、「CTFでハッキングだ！」なんてのんきなこと言ってもらえません。

そのため、私はこのサーバーたちのシステム更新をすることにしました。

6.3 フリーダムすぎたサーバーたち

Debian の OS アップグレード自体は、ドキュメント通りに従えば、おおむねうまくいきます。(ほんとに?)

しかし、「そもそも PC 自体が老朽化しており、新しいマシンを導入する必要がある」

と言う人がいたため、新しいマシンに OS をクリーンインストールし、旧サーバー群からサービスを 1 つずつ手作業で移植することになりました。

ちなみにその人というのは「konoe_akitoshi」さんです。システムの特に物理層～ネットワーク層に強い関心を持ち、今回のシステム更新において、ebay やヤフオク等で機材を手配したり、私に様々な指導を賜った方です。

そんな新しいマシンはなんと、Dell 社の PowerEdge R630 !!

自作 PC から業務で使われる本格派のラックサーバーに昇格です。

いままで、複数のパソコンで動いていた様々なサービスをこの 1 台に統合し、しかも Proxmox の力で仮想化することでより便利に扱えるようにしようという考えから決まりました。(考えたのも決めたのも konoe_akitoshi さんです。)

ではその「様々なサービス」ですが、これがほんとに**様々**でした。

MMA は技術系サークルであり、かつフリーダムな運営方針をとっていることから、サーバーたちの環境も見事にフリーダムです。

使えず放置された RADIUS サーバー、何かが足りないネットワークブート、在処不明の DNS サーバーに、学習コストの高さゆえ見捨てられた Kubernetes...

私の仕事はまず、本当に必要なサービスがどれかを見極めるところからでした...

6.4 リモートワークってやりづらい...

必要なサービスを見極めるとはとっても、私は新入部員です。部員がいつもなにをして、どのサービスを使っているのか何も知らない状態でした。

そのため、私はまず、部室に率先して在室し、いろいろな人に質問したり、自分で触ったり、先人たちが整備くださったドキュメントを見たりしてサービスの選定を進めました。

しかし、このやり方には限界があります。それは、同じく部室によく在室する人からしか話を聞き出せないことです。

MMA には、対面参加と幽霊部員だけでなく、Slack を使ったオンラインチャットで生存している部員も大勢います。それはその人のスタイルだったり、研究などで多忙だったりと様々な理由故です。

対面参加している部員は限られる中、手あたり次第先輩や同級生に使いたいサービスを質問したり、不明な点について詳しくそうなオンライン上の部員が誰なのかを尋ねたりして、かなり時間をかけて進めました。

ただ感じたのは、**オンライン上の部員に質問するのがダルい！**

返事がなかなか来ず遅延したり、来ても結局わからなかったりと、コミュニケーションがとにかく不便でした。

「対面で質問できればいろいろ情報をポンポン聞き出せるのに...」と思いながらも、仕方なくしていた次第であります。

米 Amazon が出社を強制したのも納得です。

6.5 VLAN？なにそれおいしいの？

無事サービス選定も完了し、サーバー構築開始だ！

といきたかったのですが、サーバーの構築の前にとある機材の設定が必要でした。

そう、スイッチングハブです。自作 PC とおなじく、旧来から使われていた L2 スイッチは、もう物が古く、交換が必要でした。(konoe_akitoshi さん曰く)

しかもそのスイッチングハブは、もちろん業務用のもので、VLAN 等の細かい設定ができます。そして、対外接続と内部サーバー、Wi-Fi クライアントでネットワークを分けるためには設定が必須でした。

ところが困ったことがあります。私、なにもわからなかったのです。

うちの Wi-Fi ルーターはいじったことはあっても、Cisco コマンドなんてこれっぽちもわかりませんでした。

かく言う konoe_akitoshi さんも、コマンド周りは Chat-GPT 頼り...

そのため、私は VLAN 等の基礎知識と Cisco コマンドをほぼ強制的に習得する羽目になりました。

なんだかんだ完全に理解するのは一週間はかかりました。VLAN はすぐに理解できたのですが、インターフェースの扱いや trunk ポートで Untagged な通信をするやり方など、細かいところも学ばなければいけず...

なんだか久しぶりにパソコンのスキルが上がったな感がありました。

6.6 レガシーとエラーの戦い

スイッチングハブのリプレースが終わったら、とうとうサービスの移植作業です。

相当な規模と数のサービスがあるため、いったんすべてストップさせて作業することなんて当然できません。

幸いにも、旧サーバーと新サーバーは物理的に分かれているため、旧サーバーを動かしたまま新サーバーに順次構築していくという方式で作業できました。

中でも大物は、部員がホームディレクトリに雑にファイルを置いた”nest”サーバーと、
<https://wiki.mma.club.uec.ac.jp/> が動いている”wiki”サーバー、
<https://gitlab.mma.club.uec.ac.jp/> が動いている”gitlab”サーバー、
部室用鍵ドア管理システム”kagisys”のバックエンドのサーバーでした。

”nest”はとにかく容量が多い！（総容量 1.5TB）

”wiki”は使われている技術が古すぎる！

”gitlab”はドキュメント見てもエラーでうまくいかない！

”kagisys”も使われている技術が古すぎる！

システムエンジニアの宿命といいますか、何事もなく作業が終わるなんて奇跡はないんですね...

6.6.1 ”nest”との戦い

”nest”とは、MMA にとってのメインのサーバーであり、ホームディレクトリに雑にファイルを置いて作業するためのサーバーです。

そのため、ホームディレクトリを丸ごと移行すればとりあえずいっちょ上がりなのですが、その移行のために使った最初のコマンドがこちらです。

```
（移行先） $ scp -p （移行元）:/home /
```

これを実行して「時間かかるから一晩放置しとくか！」とすやすや寝て翌目覚めると

「まだ終わってないんだ... うん？容量オーバー！！？？」

事前に”df”コマンドで十分容量があることを確認したはずなのになぜこんなことに...

なぜなら、コピーするディレクトリの中にシンボリックリンクがあったからです。

”scp” コマンドは、シンボリックリンクはリンク本体ではなく、リンク先のディレクトリをコピーします。

しかしこれでは不都合なことがありました。そう、無限ループです。

シンボリックリンク先が親ディレクトリのものがたくさんあり、ext4 のパス長制限ではじかれているにもかかわらず、ループでコピーされたファイルたちでストレージが埋め尽くされてしまいました。

このことを経験して、私は次のコマンドを覚えました。

```
(移行先) $ rsync -av (移行元):/home /
```

このコマンドを使うことで、シンボリックリンクだけでなく、UID, GID, アクセス権限をそのままに、ホームディレクトリをコピーすることができました。(実はハードリンクはコピーされず無視されるのですが、今のところ苦情はないので OK ということで...)

ただディレクトリをコピーするだけの作業でしたが、私の知識不足のためにかなり時間がかかりました。

6.6.2 "wiki" との戦い

MMA で利用しています Wiki エンジンは、”MoinMoin” というものです。

”MoinMoin” の歴史は古く、なんでも 2006 年ぐらいからあるのだとか。

趣があって結構と言うわけにはいかず、古い技術故、移行作業も手探りで大変でした。

”MoinMoin” は、Python 2.7 で動いています。現行の Python のバージョンは 3.13.0 です。
(2024/11/07 現在)

Python 2.7 はすでにサポートが終了しています。Ubuntu 24.04 では、apt コマンドで入れることすらできません。

Python 2.7 自体は公式から tar ボールをダウンロードして自前でビルド&インストールできますが、”MoinMoin” を動かすためには、Apache2 の mod_wsgi (WSGI : Web Server Gateway Interface) で Python を動かせるようにしなければなりません。

しかし、そのために必要な libapache2-mod-wsgi も、Python 本体と同様に、以下のコマンドで入れることはできません。

```
$ apt install libapache2-mod-wsgi
```

そのため、私は必要なファイルを手動で移植する対応を取りました。

具体的には以下のファイルです。

```
$ ldd /usr/lib/apache2/modules/mod_wsgi.so
```

(略)

```
libpython2.7.so.1.0 => /usr/lib/x86_64-linux-gnu/libpython2.7.so.1.0
```

(略)

移植できればあとは、MoinMoin のプログラムとデータリソースを先ほども登場した `rsync` コマンドでコピーするだけです。

(自前でインストールした Python は `apt` でインストールされるものとファイルパスが違うために後々厄介になった話はやめておきましょう。)

これで最新の Ubuntu 24.04 で古の”MoinMoin”を動かすことができました。

6.6.3 ”gitlab”との戦い

Gitlab も部内でよく利用されるサービスです。ありがたいことに、導入もサーバー移行も公式ドキュメントにすべて方法が記載されています。

それに従うだけで万事 OK !

となるわけがないのがシステム管理の習わしです。悲しい。

いろいろつまづき箇所がありましたが、最も厄介だったのが、バックアップ用ファイルの作成作業です。

作業自体は

```
$ sudo gitlab-backup create
```

のコマンドを実行するだけです。

しかし、何度実行してもエラーが出力され、うまくいきませんでした。

原因は、Gitlab 上のあるリポジトリが壊れていたためにバックアップ用データが作れなかったというエラーです。

ご丁寧にも、エラー出力時にどのリポジトリが壊れているか教えてくれたため助かったのですが、あまり部室にいらっしゃらない先輩のリポジトリでした。

勝手に私が削除するのは当然駄目なので、きちんと Slack で連絡をし、リポジトリの削除のお願いをしました。

しかし、先輩が忙しかったこともあり、結局 3 日ほど対応に時間を費やしてしまいました。(一応別の作業も並行しながら行っていたためスケジュールの影響は抑えられました。)

削除したことで無事バックアップ用コマンドも動作するようになり、あとは難なく(?) 移行作業をすることができました。

(そのあとに `certbot` が勝手に入ってインストーラーが止まったり、旧サーバーのバージョンが古いためにアップデートを繰り返して時間を費やしたりとありましたが...)

6.6.4 "kagisys" との戦い

MMA はサークル棟の居室を一部屋借りて作業しています。その部屋の鍵は物理鍵ですが、数に限りある中部員同士で共有するのも面倒なため、Arduino と Linux マシンを用いて電子錠システムを導入しています。それが"kagisys"です。

Kagisys 自体は歴史がかなり長く、2006 年から存在するという記録があります。現在は 2 代目にあたる Kagisys2 が 2017 年から稼働しています。

Arduino 側はそのままですが、Linux マシン側は今回のシステム大幅更新の対象になったため、移行作業が必要になりました。

Kagisys は MMA 部員のみの独自開発で、Linux 側は Ruby 2.7 で稼働しています。現行のバージョンは 3.3.6 です。そうです。先述の"wiki"と同じく、古いバージョンだったためにインストール作業が大変でした。

Python と同じく古い Ruby を最新の Ubuntu の apt でインストールすることはできません。

ただ、Kagisys のプログラムファイル自体は一つのディレクトリに完結していたため、Docker コンテナ化して対応することにしました。

Docker コンテナ化すれば、コマンド一発でインストール&稼働できるだけでなく、移植作業も大変容易になります。

しかし、この Docker コンテナ化も一筋縄ではいきませんでした。

Kagisys は、Docker 公式の Ruby イメージだけではライブラリが足りず、追加するために bundler コマンドを Dockerfile に書く必要があります。

しかし、先人達を書かれたドキュメントのコマンドをそのまま書くだけではどうもうまくいきません。

仕方がないので、Google 先生や Chat-GPT にお世話になりながら試行錯誤していました。

特に syslog ドライバ周りを動作させるのが鬼門でした。ライブラリをインストールしたり、設定ファイルを手動で配置したりといろいろ試しましたが、

正解は docker-compose.yml に

```
volumes:
- type: bind
  source: /dev/log
  target: /dev/log
```

を追記するだけでした。拍子抜けです。

また、「なんとかエラー駆除が終わって動いたぞ!」と思ってしばらく動かしていたら、今度は無限に再起動する問題が発生しました。

docker-compose.yml に restart: always を書いているため、何らかの問題が生じてプログラムが強制終了していることはわかります。

しかし、

```
$ docker compose down
$ docker compose up -d
```

をすると、再起動せずちゃんと起動し、再現できません。ログにも記録がなかったため、原因究明に時間を要しました。

原因としては、pid ファイルによる多重起動防止のプログラムが悪さしていました。

そのため、Dockerfile ファイルを

```
CMD rm -f /tmp/pidfile && ./kagid start
```

と書き換えて起動直前に pid ファイルを消すことで解消できました。

6.7 まだまだ終わらない更新作業

この部誌を執筆している 2024 年 11 月 7 日現在、まだシステム更新作業が完了していません。はじめたのが 6 月初頭なので、すでに 5 ヶ月も経過しています。

なぜなら、私も konoe_akitoshi さんも、サークル活動のほかに、アルバイトや自動車教習等、やることが多く、忙しいからです。

しかし、現状は冗長性、信頼性の観点から万全ではありません。そのため、隙間時間を絞りだしながら、また、部員の活動に著しく影響が出ないよう時期を見計らいながら、少しずつ完成度が上がるよう努めています。

最近では、Proxmox バックアップサーバーの構築が完了し、万が一メインのサーバーがぶっ壊れても安心できるようになりました。

そしてハードウェアルータが導入され、それを設定するという私のタスクが増えました (^^;)

6.8 おわりに

振り返ってみれば、私はシステム管理については完全に初心者のはずでしたが、ここまで私を育てる MMA の潜在的教育力は本当にすさまじいです。

もちろん、それは部員の皆様が、システム管理作業にご理解いただき、また様々ご協力をいただいているおかげです。この場を借りまして、心より感謝申し上げます。

さて、本職の SE の方から見たら非常に初歩的な内容ばかりでしたが、ここまで読んでいただきありがとうございました。

この記事を読んで、システム管理の大変さを少しでもご理解いただければ幸いです。

また、電気通信大学に興味のある受験生の皆さん！ MMA に入ればこういったサーバー群を思う存分

に触れますよ！

初心者の方でも大丈夫です！やる気さえあれば sudo を振り回していろいろできます！

現に初心者の私がこれほど盛大に振り回していますので！

では、次の「百萬石」で、完成したサーバー群とともにお会いしましょう。

(はやく CTF やりたいな...)

第 7 章

ロシア語を学ぶ

nasu

7.1 はじめに

こんにちは、nasu です。先輩が「部内の活動と関係ないことでもなんでも書いていいよ！」と部誌の寄稿を呼びかけていたので、プログラミングやらその周辺は悉く初心者で書けることが無いなどと思っていたのですが書いてみることにしました。今回これを書くにあたってどのくらい関係ないことを書いて良いのか分からず不安だったので、まずは過去の事例を探してみました。すると、特に最近は「〇〇作ってみた」「〇〇の自動化」みたいなものが多いように見えて、本当に関係ない話をしてる人があまりいないことに気がきました。じゃあ本当に関係ないことを書いてやろうじゃないか。一応遡れば、マリオ 64 スター 120 枚の話とかもあったので、私も時オカ 64 の話とか FFVII(Play Station) の話とかしてみようかなとも思ったんですが、遊ぶ時間は無いし例えば RTA が出来るほど上手いわけではないのでボツになりました。そんなわけで書いたのがこちらです。

7.2 なぜロシア語なのか

電通大では第二外国語で中国語、韓国・朝鮮語、ドイツ語、フランス語、ロシア語の 5 つから選んで学べます。この中から選ぶにあたり、私は「最も理解できない言語」を選ぼうと決めました。日本人に一番なじみのない言語を読めたら、なんかもう最強じゃね？って思ったんですね。

中国語の表記は漢字ですから文字において日本語と共通点があります。韓国・朝鮮語には隣国の中国由来の言葉がいくつかあり、発音が日本語にそっくりな言葉もいくつかあります。ドイツ語やフランス語は語族こそ違えど、ヨーロッパ内で多く交流があった場所の言語であり、ラテンアルファベットを使うため、綴りからそこそこ意味が分かりそうだと思います。ロシア語はというと、文字は読めないし、なじみのある言葉も「イクラ」「ノルマ」「マトリョーシカ」などその程度しか知らない。そうだ、ロシア語をやろう。

言語に詳しいわけではなかったのですが最終的にはオリエンテーションで決めましたが、電通大に入る前からこんなことを考えており、中韓独仏露から選ぶならロシア語に限るだろうとは何となく思っていました。

というわけで現在ロシア語を学んでいます。基準が「最も理解できない言語」なので、仮にアラビア語が選択肢にあったらアラビア語を学んでいたかもしれません。

7.3 ロシア語って難しいの？

7.3.1 現在の学習進度とそれまでの道のり

まずは私のロシア語レベルですが、大体入門～初級レベルといった程度だと思います。CEFR(ヨーロッパ言語習得程度の指標)で言うと A1 程度で、まだまだ赤子です。必修第二外国語で使っている教科書(『初級ロシア語 20 課』, 白水社)は一通り終えました。後から知ったのですが、電通大の露語の授業では教科書は一年生では全部終わらず、二年生で終わるそうです。え。

実はロシア語自体には高校生の頃から興味があったので YouTube で学習を試みたことがありましたが、アルファベットの読み方を覚えてただけで(すぐ忘れました)、結局モチベーションも続かず断念していました。学習を本格的に始めたのは大学に入ってからで、授業に必要な教科書を買ったことでモチベーションが爆上がりしました。前期の間はなんだかんだ忙しくて授業に合わせて勉強する程度で精いっぱいだったのですが、夏休みに入ったら頑張ろうと決めていました。

夏休みに入って、とりあえず教科書を終わらせよう！というモチベーションのもと、第 6 課までしか終わっていなかった教科書を夏休みのうちに第 20 課まで全て終わらせて練習問題も全問解きました。偉すぎる。しかし、それは決してスムーズに進みませんでした。私自身飽き性なので、同じことを繰り返すのが好きではなく、1 つ終わったら次、1 つ終わったら次、と進めていました。授業では 2,3 週間に 1 課進むペースだったのですが、1 日に 2,3 課進める勢いでガツガツやっていたので、かなりストレスになりました。

手前でやったことを覚えていないと次に進んでも意味が無いので、7 課からの本文、単語、文法事項をすべてルーズリーフにまとめ、付属の音声も全て聞いてディクテーションしました。でも、単語は 1 課に 30 語くらい(3 課で 90 語!)出てきて覚えきれないし、特に動詞は全くなじみがないのでゴリ押ししかありません。ストイックに取り組んだおかげか内容の吸収は良かったのですが、精神的には良くなかったのか、8 月半ばに始めてから 3 日で飽きてしまい、再開したのは 9 月半ばでした。続きからまた 3 日ほど勉強してなんとか教科書を終え、ロシア語の入門レベルを卒業(?)できました。合計して 1 週間ちょいしか教科書やってないやんけー。

7.3.2 ロシア語をここまで学んできた感覚

教科書が終わった時点での感覚では、しっかり勉強すれば怖がるほど難しい言語ではないなと思います。暗記が必要な部分は頑張らなければいけませんが、それさえ乗り越えれば普通だと思います(まだ乗り越えていませんが)。しかしながら、簡単というわけでは決してなく、むしろ慣れていない初学者にはあまりにも複雑で難しく見えて、まるで常願寺川を遡上している気分になります。

7.3.3 おすすめの参考書

暗記のためにはやはり繰り返しの学習に勝るものは無いと身に染みて感じたのですが、良くも悪くもロシア語学習の書籍は入門書が大半を占めており、問題集や中級・上級書は片手で数え切れる程度しかありません。ですので、習った文法について実際に問題演習をしながら確実に進めたいという人にとっては少し学習しづらい言語だと言えるでしょう。

そこで、問題演習をしたい方には、無責任ながら私は買っていないのですが、東京外国語大学が出している教科書『大学のロシア語 I』をおすすめします。大きいので電車で読んだりとは出来ませんが、教科書準拠の問題集『大学のロシア語 II』が別冊として売られており、併用することでしっかりと定着させながら学習が進められそうです。

私は先述の教科書『初級ロシア語 20 課 (桑野隆:著, 白水社)』を既にもっていたので、別の『自習ロシア語問題集 (中村健之介:著, 白水社)』を購入しました。これも内容は素晴らしい良書でおすすめしたいのですが、字が小さかったり、他の本に準拠しているわけではないので注のある語とない語が自分の知識と一致しなかったりと、仕方ないと思いつつやや使いづらいなと感じます。今まで外国語学習に限らず学校などで教科書に準拠したワークを使ってきた方は多いと思いますが、その何と素晴らしいことか、やっと気づきました。

他にも『ニューエクスプレスプラス ロシア語 (黒田龍之助:著, 白水社)』も良いです。これも私は持っていないのですが、ネットで色々調べていると評判が良いのでおすすめしておきます。余談ですが、このシリーズは非常に多くの言語を扱っており、中にはアイヌ語やバスク語などのマイナー言語もあります。この本が気に入ったら他の言語でもこのシリーズで勉強したくなるんだろうなあ、と思います。^{*1}

あとは定番ですが『NHK まいにちロシア語 (NHK 出版)』も良書でしょう。ラジオ放送に合わせて毎月 1 冊ずつ半年かけて学ぶ全 6 冊構成になっていて、定期的に新作の話が追加されています。初級編と応用編が一冊に入っていますので、一通り入門レベルが終わってからでも使い続けられる長持ちな本だと言えます。

また、問題演習に限って言えば、検定試験の過去問を購入して取り組むのも良いと思います。入門書は一通り理解しておく必要がありますが、級によってそのレベルが明確でし、そのレベルの文法のほとんどが凝縮されていますから、知識の確認や苦手発見に最適です。

ここで紹介した入門書には全て音声 CD が付いています (問題集には付いていません)。ロシア語は発音 (特に子音) が外国人にとって非常に難しい言語ですから、まずは CD で本物の発音を聞いて少しずつ慣れるとか、聴取の練習に使うなど、初学者には欠かせません。

^{*1} 生協の書籍を扱っている方がこれを読んでおられましたら、ニューエクスプレスプラスシリーズだけでも置いてくれると我々言語好きが救われますのでよろしくお願いします。

7.4 ロシア語能力検定 4 級に挑む

教科書を終えたので早速検定試験に挑戦してみようと思い、10/26(土)に検定があるとのことでしたので勢いで申し込みました(9月上旬)。今回挑戦したロシア語能力検定 4 級は一番簡単な級です。教科書をやっただけで受かるレベルなのかよく調べないまま申し込んだのですが、9月末に過去問を買って解いてみたところレベル感は教科書程度だったので、頑張れば受かるかなーと思っていました。しかし、ここで壁が現れます。ロシア語能力検定 4 級は「文法」「露文和訳」「和文露訳」「朗読」の 4 項目から構成されているのですが、和文露訳だけが全然できないのです。ロシア語の検定はやや特殊で、すべての項目で 6 割以上得点しないと合格できません。過去問を買ったのは 9 月末でしたから夏休みがすぐに終わってしまい、いよいよ勉強時間がありません。試験までの 4 週間、毎週 1 回分やるだけの過去問があったので最低限それだけでもと思い欠かさず解きましたが、やはり和文露訳だけが伸びず、そのまま試験当日になってしまいました。

出願時に東京での受験を希望したので、会場は世田谷区経堂の東京ロシア語学院でした。受験者が何人くらいいたか忘れましたが、私が受験した教室には 60 人ほどいたようです。試験は昼すぎ(14 時)からでした。電車内や、会場に着いてからも単語帳を眺めて最後の足掻きをしていましたが、試験なんて当日になってしまったらもうどうしようもないのが常です。直前まで単語を詰めたらすぐに切り替えて、試験開始です。

4 級の試験時間は筆記の「文法」「露文和訳」「和文露訳」で 90 分、10 分の休憩があって朗読が 10 分(2 分説明、5 分黙読、3 分録音)です。文法は過去問の手応えからとりあえず解答すれば 6 割は取れると思っていたので、露分和訳・和文露訳に多く時間を割こうと決めていました。実際にこれが功を奏し、露文和訳・和文露訳ともに過去イチ良い手応えでした。幸運にも知っている単語や表現が多く訳しやすかった上、時間があったので落ち着いて取り組めたのが良かったと思います。

ここまで読んで気になるのが合否についてだと思いますが、発表は 12 月 16 日なので残念ながらまだ分かりません。和文露訳の部分点がたくさん入って合格していたら良いなと思いつつ、意外と辛口採点だったりしないかとか文法で油断して落としてたりしないかという不安もあり、これを書きながらもウズウズしています。受かっていたら次回(5 月)かその次(10 月)に 3 級を受けてみようと思います。

余談ですが、日本で知られているロシア語の検定は 2 種類あり、今回受験した、いわゆる英検のような「ロシア語能力検定試験」(4 級～1 級)の他に、ロシア語版 TOEFL のような「ロシア語検定試験(TPK И)」(入門、基礎、第 1～第 4 レベル)があります。前者はロシア語能力検定委員会が主催して日本でのみ実施されており設問も日本語ですが、後者は日本対外文化協会と東海大学が主催しロシア語検定試験実行委員会が実施するロシア連邦教育科学省認定の試験で、設問から全てロシア語で書かれているそうです。外国人のロシアでの留学や学位認定には、TPK Иの指定された級の合格証明が必要だそうで、私個人としては留学する気はありませんが、いつか受けてみたいです。こちらも各セクションで 6 割以上の得点が合格条件らしいです。ロシア語さん厳しい～。

7.5 まとめと今後の展望

ロシア語は変化が多い言語です。英語なんかとは比べものになりません。しかしその分、日本語のように主語無しで話したり (日本語と違い主語は明確です!), ある程度語順を崩して話せるメリットもあります。また、世界で9番目に話者が多い言語ですので、覚えると話せる相手がネイティブで1.5億人、非ネイティブを含めると2.5億人くらい増えます (2024年春時点)。言語を学ぶと必ずその国の文化にも触れることになりますし、語彙の成り立ちや文法などにも面白い発見が絶対にあります。ぜひロシア語の学習を通してロシアと周辺諸国の美しい側面に興味を持っていただきたいですし、ロシア語に限らずとも、それぞれの好きな言語の良さや面白さを感じてほしいなと思います。

どの言語にも言えることですが、最も基礎的で日常的な語彙ほどよく使われるので、変化が不規則になりやすく、例外が多いです。0から1にするのは最も大変な過程ですが、ここを乗り越えて覚えてしまえば土台になりますから、次を覚える手助けの道具になります。なので、どの言語を学ぶ時も「そういうものだ」と割り切って根気よく取り組んでみてください。その先でしか味わえない面白さがあります。

さて、今後の展望ですが、先ほど述べた通りロシア語能力検定の結果が12月16日に発表されるので、今はそれを楽しみに待ちながら勉強したいと思います。気が付いたら言語大好きマンになってしまったので、将来的には国連公用語 (英語、中国語、フランス語、スペイン語、ロシア語、アラビア語) を全て話せるようになりたいな—と思っています。ロシア語の次はスペイン語かなあ。とりあえず今はロシア語の日常会話ができる程度 (検定の2級程度/CEFR B1~B2) を目指して精進します。

ここまで読んでくださりありがとうございました。Большое Спасибо!!

百萬石 2024 -秋- © 電気通信大学 MMA

2024 年 11 月 22 日 初版第一刷発行 【本書の無断転載を禁ず】

著 者 gae, Udon, namor, hydrogen, koshihkari, daiki, shiragi, nasu
表 紙 Udon
編集者 Udon
発行者 電気通信大学 MMA
発行所 電気通信大学 MMA 部室
〒182-8585 東京都調布市調布ヶ丘 1-5-1 サークル会館 208 号室
<https://www.mma.club.uec.ac.jp/>
印刷所 電気通信大学 MMA 部室
製本所 電気通信大学 MMA 部室



歯めた歯科
内国産1年産3000%以上
ンプラン
17:08
10120-221
なり 西八王子

電 気 通 信 大 学



立川駅

立川駅北口