目 次

・・・・・0・・・・・

はじめに

部長 浜田克紀

このたび npca に来て下さり、この部誌を取ってくれた皆さん、本当にありがとうございます。私は npca の現部長 katukky こと浜田克紀です。この一年間パソコンが壊れたりとあわただしかったですが、なんとかここまでくることができました。

この部のことを紹介いたしますと、旧校舎の四階でパソコンでプログラミングや音楽や CG の作成、サーバーの管理などの活動をしております。

部室にくる決まった時間などはなく、朝学校があいてから夜の 6:00 に学校がしまるまでのたいていいつでも活動しています。普段は部屋が散らかったりしていますが、まじめにやろうと思えばできるだけの環境は整っていると思います。現在稼動しているパソコンはサーバーを合わせて 4 台となっています。

あと、ホームページを運営しています。アドレスは `http://trlocon.ddo.jp/~npca/` になっています[1)]。

[1)] npca の部内サーバーが 8 月に起きた落雷から 2007 年 1 月ごろ復旧したので、部のホームページをレンタルサーバーからそちらへ移転させました。以前のアドレスを登録している方はブックマーク等の変更をお願いします。

・・・・1・・・・

CUI で始める動画変換

61 回生 中田亮

前書き

どうやら今年の部誌は量・質ともに不足気味らしくて、それを補う為に何故か私が部誌を書くという事態になっています。実は私は本来部員ではないのですが、部室のサーバや壊れたマシンを復旧したりしているうちに部員の一人として認識されてしまい、今年の部誌の量・質を補う為に何か書いてね〜と相成った訳で...。数年前はパソコン部員の技術もそれなりに高かったはずですが、技術が伝承されておらず、どうも質の良い物が書けていないようです。この部誌も昔の先輩方を引っ張りだして作っているようですね。この先パソコン部大丈夫なんでしょうか。しかし、そんなことを嘆いてもしょうがない。今年は「動画変換」について書く人がいないそうなので、私が書くことにしました。ちなみに、当初は初級編・中級編・上級編と 3 つに分けて、最初は GUI から入って少しづつ CUI に慣れれていく、という方式をとっていたのですが、逐一解説していくと流石に長くなりすぎるので上級編だけにしました (苦笑) そのためかなりマニアックな内容になっていることは否めません。それに、生半可な知識で書いているので見る人が見ればかなり恥ずかしい内容になっているかと思いますが...まぁ部誌の量が稼げるわけですからそれでもいいでしょう (苦笑)

エンコードの前に

さて、前年度の部誌で動画変換について書かれている物と言えば、「録画番組パソコンで！ AVI 版。」計画再浮上！、ですね。去年の部誌をなくした人は npca のホームページにあるのでそちらを読んでください。
ここでは動画の変換の基本方針として、「タダ」「軽い」「綺麗」「速い」「人気」の５つをあげています。しかし今回は少し違います。今回の変換の基本方針は画質です。いかにして少ないビットレートで高画質にするか。それが動画変換の永遠のテーマなわけで、今回はその限界に挑んでいこうと思います。去年の部誌では、適当なエンコーダとして「divx」「xvid」「wmv」「x264」の４つが紹介されていましたが、画質を取るなら x264 で決まりです。ちなみに、snow という、x264 よりもさらに先進的なエンコーダもあるようですが、再生互換性や再生負荷のことを考えるとちょっと...。互換性はともかく、再生負荷重すぎです。H.264 もさんざん重いと言われましたが、それよりも更に重いです。まぁ、使えない訳ではないので、さらっと説明しておきます。

また、画質を語る上で重要なのはエンコーダの性能だけではありません。ビデオフィルタも重要です。変換前にノイズリダクション、インターレースの解除等を行う必要があり、この性能にかかわってくるのがビデオフィルタです。ノイズリダクションやインターレースの解除をうまくできるか否かで、画質がかなり違ってきます。ここも

重要なポイントです。ビデオフィルタの機能を持ち合わせているソフトは、「avisynth」「mencoder」「aviutl」「ffmpeg」等があるのですが...。ここでちょっと言うことがあります。私は Macintosh 使いなので、Macintosh を前提として話を進めていきます。あぁ、この時点で更にマニアックになった気がしますが...。気にしないで進めましょう。まぁ、そんなにプラットフォームに依存することは書いてありませんから、Windows や Linux でも多少手順が違うだけで応用が効くかと思います。また、Mac OS 9 以前をお使いの方は残念ながらここに書いてあることは全く役に立ちません。Mac OS X にアップグレードすることをお勧めします。さて、aviutl に関しては、AVI 出力専用です。(プラグインを組み込めば AVI 以外にも出力できますが、実質 AVI 出力専用です) 以前は手軽に高画質が実現できるということで広く使われていたのですが、x264 が AVI(VFW) コンテナのサポートを廃止したので実質使えません。x264 開発ページに「no more vfw」の告知が出たときは結構騒がれたのですが、まぁいずれにせよ aviutl は Windows でしか使えないので対岸の火事といったところですね。avisynth は、aviutl ほど手軽に使えるわけではありませんが、優秀なビデオフィルタです。Windows を使っているのであればこれを使うべきなのでしょうが、今回は Macintosh 向けに書いているので解説しません。さて、Macintosh で一番使えるビデオフィルタは、mencoder です。もともと UNIX のソフトウェアなので CUI から利用することになるわけです。ffmpeg は GUI で利用できる動画変換ソフトで内部的に使われているようですが、ビデオフィルタは貧弱です。ということで、Macintosh でビデオフィルタを選ぶなら mencoder に決まりです。

エンコード方法の解説

さて、早速 mencoder をダウンロードしてきましょう。本来 UNIX のソフトウェアはソースコードだけ置いてあって、勝手に各自ビルドしていってください、といったノリのものが多いのですが、Mac OS X で動く mencoder のバイナリは有志がビルドしてくれたものがありますから始めのうちはそれを使いましょう。

`http://ffmpegx.com/download.html`

上記の URL は ffmpegX という GUI で動く動画変換のソフトですが、mencoder のバイナリが置いてありますのでそれを使いましょう。Windows、Linux のバイナリも探せばあるとは思いますがここでは紹介しません。

当然ですが CUI で動くツールなのでダフルクリックしても開きません。ターミナルから利用します。ここではターミナルの使い方については解説しません。ここに書くと長くなりすぎるので。必要なら自分で調べてください。ところで、mencoder だけ落としてきて、肝心の x264 は要らないのかと思われるかもしれませんが、落としてきた mencoder の中には x264 も含まれているので問題ありません。

さて、早速変換してみましょう。ホームフォルダ直下に mencoder と変換したいファイル (ここでは hoge.mpeg としています) を置いて、アプリケーションフォルダ内にあるターミナルを開いて以下のコマンドを実行すれば変換できます。ターミナルの扱い方に付いて説明しだすと長くなるのでここでは割愛します。別に Macintosh でなくとも同じコマンドで変換できるはずです。

```
./mencoder hoge.mpeg -nosound -ovc x264 -x264encopts bitrate=1024
:bframes=3 :b_adapt :weight_b :b_pyramid :keyint=240 :keyint_min=1
```

```
:scenecut=65 :qp_min=10 :qp_max=51 :qp_step=8 :qcomp=0.6 :ratetol=4
:deblock :deblock=0,0 :cqm=jvt :cabac :direct_pred=auto :nofast_pskip
:nodct_decimate :nointerlaced :noglobal_header :psnr :ssim :pass=1
:threads=2 :turbo=1 -passlogfile hoge.264.log
-vf pullup,softskip,pp=l5,crop=720:480:0:0,scale=640:480:::3,
hqdn3d=4:3:6,harddup -sws 9 -ofps 24000/1001 -of rawvideo -o hoge.264

./mencoder hoge.mpeg -nosound -ovc x264 -x264encopts bitrate=1024
:bframes=3 :b_adapt :weight_b :b_pyramid :keyint=240 :keyint_min=1
:scenecut=65 :qp_min=10 :qp_max=51 :qp_step=8 :qcomp=0.6 :ratetol=4
:deblock :deblock=0,0 :cqm=jvt :cabac :direct_pred=auto :nofast_pskip
:nodct_decimate :nointerlaced :noglobal_header :psnr :ssim :pass=2
:threads=16 :me=umh :me_range=32  :subq=7 :frameref=4 :mixed_refs
:8x8dct :partitions=all :trellis=2 :brdo :bime -passlogfile hoge.264.log
-vf pullup,softskip,pp=l5,crop=720:480:0:0,scale=640:480:::3,
hqdn3d=4:3:6,harddup -sws 9 -ofps 24000/1001 -of rawvideo -o hoge.264$
```

ageha さん[1)]の設定です。感謝！ (アニメ用設定ですので、実写では変えましょう。解説読んでください。)

な、長いですね...。流れ的には、まず高速な設定で変換し、そこでできた映像ファイルは捨ててログを取得、それを使ってもう一度低速だが高画質な設定でエンコードします。いわゆる 2 パスエンコードですね。これから ageha さんの設定に基づいて、オプションを一つずつ解説していきます。

`-nosound`
サウンドを出力しない。映像を生データで出力し、音声を別途変換して MP4 に詰めるのでここでは音声は変換しません。

`-ovc x264`
エンコーダの指定。x264 を使います。

`-x264encopts`
この後に続く長ったらしいオプションは x264 の設定項目です。x264 を mencoder を経由せず直接利用した場合は違った書式になるのですが、おおむね対応しているようです。

`-passlogfile`
ログファイルの場所を指定します。2 パスエンコードで使います。

`-vf`
ビデオフィルタのオプションをこのあとに指定します。

`-sws`
スケーリング (ビデオのリサイズ) の設定。0〜10 の間で指定します。数字が大きくなると、輪郭がなめらかにリサイズできますが、その分遅くなります。拡大する場合は結構差が出ますが、縮小・等倍の場合はあまり見た目変わりません。まぁでも、そこまで遅くなるわけでもないので 9 になっているようです。

[1)] http://agehatype0.blog50.fc2.com/

`-ofps`
出力 fps の指定。ビデオフィルタを経由し、インターレースを解除したので出力 24fps となります。正確には、23.976 ですが、より正確には $\frac{24000}{1001}$ です。ここでは分数で指定できるので $\frac{24000}{1001}$ になっています。
`-of rawvideo`
映像を生データで出力します。後で音声とともに MP4 に詰め直すため。
`-o hoge.264`
出力ファイル名の指定。

`-x264encopts` のオプションを解説します。ここを弄ることで画質が変わってきます。一部省略してます。

`bitrate=1024`
ビットレートを指定します。ここでは 1024 になっていますが、私は 768 でも十分だと思います。ですが、画質にこだわる場合は 1024 が良いのでしょう。
`subq=7`
1〜7 の間で指定します。subpel 精製品質の調整、らしいですが、そんなこと言われても訳分かりません。単純に、品質の度合いを表す数字と受け取って構わないようです。数字をあげると品質は良くなりますが遅くなります。とはいうものの、この値が低いと効かないオプションもあるので、最低でも 5 以上にした方が良いです。出来れば 7。
`frameref=1 frameref=4`
変換時に何枚前のフレームまでたどるか、という指標です。増やせば増やす程画質が良くなり速度が遅くなっていくのですが、5 あたりで効果が激減します。この値が多いと結構時間がかかるので、1 パス目は 1、2 パス目で 4、としているようです。アニメに効果的。
`keyint=240 keyint_min=1`
GOP の最大値、最小値を指定します。GOP とは Group Of Picture の略で、動画を圧縮する際には GOP 単位に区切って圧縮します。
GOP は、I フレーム、P フレーム、B フレームから構成されています。I フレームは、それだけでデコードが可能です。ようするに、jpeg や png の写真のように、それ単体で表示出来るようになっているフレームです。
P フレームは、デコードするのに直前の I フレームを必要とします。直前の I フレームとの「差分」だけを記録しているのです。その為、アニメの口パク等、ほとんど絵が変わらない場面では非常に圧縮効率が良いです。
B フレームは、更に P フレームの差分をとった物です。デコードするのに直前の I・P フレームを必要とします。使いすぎると画質が悪くなることもあります。よって、実際のデータのならび順としてはこんな感じになる訳です。IPPPBBIPPP-BIPBBBIPPPIP.... (B フレームはデータのならび順とデコード順が違うのですが、ここでは説明を割愛します)
ここで、IPPPBB までが 1 つの GOP、また次の IPPPB がまた一つの GOP です。I フレームをシーンチェンジの時において、そのあと少しずつ画面が変わっていく場面で P/B フレーム、またシーンチェンジが訪れたら I フレームを挿入、としてやって効率良く圧縮していく訳です。

GOP の最大値に設定された値になると、シーンチェンジが訪れていなくても強制的に I フレームを挿入します。つまり、画質的にはマイナスです。しかし、最初からではなく任意の場所から再生したいときは、I フレームからでないと頭出しが出来ませんからあまり GOP が長いと頭出しがしにくくなってしまいます。つまり、keyint の値は下げれば下げる程頭出しがしやすくなり、画質的にマイナス、ということです。設定する値は 240 フレーム=10 秒分で良いんじゃないかなと思います。頭出しなんか知ったことか！画質を追求する！という方はもの凄く大きくしてやれば望みうる最善画質になります。お勧めしませんが。
GOP の最小値は、この値よりも小さな値でシーンチェンジが訪れたとしても新しい GOP にしません。つまり、画質的にはマイナスです。(P フレームを挿入する訳ではなく、IDR フレーム等が絡んでくるのですが、ややこしくなるので割愛します。) ですが、実写では一瞬だけ映像を表示して次の瞬間には全く違う映像を表示する、なんてことはあまりないため、最小値を設定しておいた方がより効率よくエンコードできます。keyint_min=30 くらいでしょうか。まぁ、バイオハザードアポカリプスの終盤とかフラッシュバック的に連続して映像が表示されることもあるので、そこんとこどうかな〜とはおもいますが。アニメでは一秒未満でシーンチェンジすることも珍しくないので、keyint_min=0 もしくは 1 です。

scenecut=65

シーンチェンジを判定して I フレームを挿入するのですが、どれくらい映像が変わったら I フレームを挿入するか、という指標です。アニメではシーンチェンジが多いので高めに。特にアニメの OP では更に多くなりますのでこの値よりも更にもう少し高くしても良いでしょう。一方、実写では低めに。scenecut=40(デフォルト) くらい。落語など、ほとんど場面転換がない特殊な場合ではさらに低くします。

bframe=3

連続する B フレームの最大数。例えば bframe=3 だと、IPPBBB となるのが最大で IPBBBB とはなりません。bframe=0 にすると B フレームを使いません。画質を求めるなら B フレームは使うべきです。

b_adapt

B フレームの品質に影響するようです。とりあえず使っときましょう。当然 bframe=0 の場合は意味ありません。

weight_b

前述したように B フレームは使いすぎると画質が悪くなります。このオプションをつけると、B フレームに適さないフレームを発見し P フレームにします。bframe=1 ならオフ、2 以上ならオンで。

b_pyramid

B フレームは P フレームの差分をとったものですが、このオプションを使うと更に B フレームの差分をとります。その分圧縮効率はよくなります。bframe=2 以上でないと意味ありません。

deblock

デブロックフィルタ。これは必ず使うべきです。x264 のデブロックは大抵の場合、画質がよくなるのに対し速度はあまり低下しません。アニメに効果的。

deblock=0,0

デブロック強度の指定。-6〜6 の間で指定します。大抵の場合、0,0 でベストな

画質になりますので、素材に忠実にしたい、と思っていても下げるのはお勧めしません。あげるのは、オリジナルの段階で既にブロックノイズがある場合です。deblock=3,3 程度でしょう。

cabac

若干遅くなりますが、ビットレートが 10%〜15%節約できるので切らない方がいいです。画質には影響しません。

nofast_pskip

P フレームにおける速い段階でのスキップ検出。通常は画質低下等のペナルティ無しで速度が向上するのです...が、切ることで x264 特有の「闇階調でばたばた動くブロックノイズ」を抑えることが出来るので、ここでは切っているようです。実際に自分で試してみて決めることをお勧めします。

nodct_decimate

使うと若干ディテイルを除去するので、余ったビットレートを他に回せます。ここでは切られていますが、低ビットレートで変換する時やアニメでは使った方がいいかもしれません。ある程度ビットレートを見積もって素材に忠実にしたい場合はオフで。

pass=1 pass=2

パスの指定。1 が 1 パス、2 が 2 パス、という意味ではありません。1 が最初のパス、2 が最後のパス、3 がそれ以外のパスです。ですから、たとえば 5 パスでエンコードしたい場合は 13332 と変換していくわけです。とはいうものの、5 パスもやるともの凄く時間がかかりますし、x264 のマルチパスはあまり質がよろしくないようなので 3 パス程度にするくらいなら 2 パスにしておいた方が良いです。短編動画ではマルチパスが効果的です。短編故に時間もあまりかからないことですし。

threads=2 threads=16

複数 CPU で効率的に変換する為にスレッドを分割。最近、CoreDuo など、マルチコアの CPU が登場してきましたが (もっともそれより前からマルチコアの CPU はあったのですが) マルチコアの CPU で動画変換を行うと、(2 コアなら) 2 コアのうち 1 コアしか変換作業をせず、もう片方の CPU が宙ぶらりんな状態になります。それだと非常に勿体ないので、ここで指定された数値だけ作業を分担。複数の CPU に作業させます。分担させると圧縮効率に若干悪影響があるのでシングルコアの CPU を使っているならここのは threads=1 にしましょう。ageha さんのサイトでは、デュアルコアの場合で 1 パス目は threads=2、2 パス目は threads=16 とするのがもっとも良いという結論に達しています。最近はクアッドコア (4 コア) やオクトコア (8 コア) のワークステーション (もはやパソコンとは呼ばない) も登場してきました。きっとこれに動画変換をやらせたら気持ちが悪いくらい速くなるんでしょうねぇ...。

turbo=1

ターボモード。1 パスで有効。あまり画質に影響しない 1 パスのオプションを切って、エンコード速度を速くします。0〜2 で指定します。1 で良いでしょう。0 よりは 1 の方が、2 はちょっとオプション切りすぎな気がします。

nointerlaced

映像をインターレースされている物として扱います。mencoder でビデオフィルタでインターレースが解除されてから圧縮するので、当然オフ。

noglobal_header
ヘッダ情報を書き込む。PSP など一部のプレイヤーで必要になります。正直どっちでもいいです。
psnr ssim
それぞれ素材に対する忠実度を測る指標です。指定すると変換終了時に PSNR 値、SSIM 値を表示します。
me=umh
動き補償アルゴリズムの選択。次の中から指定します。
dia
ダイヤモンド。四角形サーチ。
hex
ヘキサゴン。六角形サーチ。
umh
マルチヘキサゴン。不等複数六角形サーチ。
esa
エクゾースティッド。徹底サーチ。esa は実用に堪えない程遅いですので使い物になりません。2 パス目で umh を使うのがいいでしょう。
me_range=32
me=umh,me=esa で動きを探索する半径を指定します。これもあまり大きくしすぎると遅くなります。重いと思うなら me_range=24 くらいに下げてもいいかもしれません。
mixed_refs
複数のフレームを参照することで動き補償の精度が向上します。オンにしておきましょう。
partitions=all
動き補償をする際、フレーム全体が同一方向に動くことは少なくてあちらこちらに散らばった動き方をします。そこで、マクロブロックという区画に仕切るのですが、その大きさを指定します。

p16x16,b16x16,i16x16	常に適用
p16x8,p8x16,p8x8	p8x8
p8x4,p4x8,p4x4	p4x4
b16x8,b8x16,b8x8	b8x8
i8x8	i8x8
i4x4	i4x4

たとえば、p16x16,b16x16,i16x16,i8x8,i4x4 の 5 つを使いたい場合は partirions=i8x8,i4x4 という風に指定します。all を指定すると全て使用。none を指定すると p16x16,b16x16,i16x16 以外は全て使いません。これも ageha さんのサイトで、何も考えず all を指定してよさそう、という結論になっています。
8x8dct
i8x8 を使うならオンにしておきましょう。

-vf オプションを解説します。インターレースの解除やノイズの除去に影響します。pp 系は基本的にインターレースをほぼ除去できます。

pp=lb
リニアブレンド。これを使うと完全にインターレースを除去できますが、全体がややボケて、動く物が二重化します。前後のフレームを半分ずつ混ぜ合わせた、ゴーストのようなフレームができます。
pp=l5
pp=lb の弱くした物。pp=lb と比べて副作用が少ないですが、場合によってはインターレースが残ることも。
pp=md
中央値補間。pp 系の中では単体で使う分でもっとも良い。デメリットはアニメなどの輪郭にジャギが出ること。
filmdint=fast=0/comb_thres=48
pp=md よりもクリアで、輪郭ジャギも出ません。デメリットは細かい部分が少しつぶれ、解除もれが出ること。アニメの口パクなどで良く残ります。filmdint 内でもオプションがあり、いろいろ挙動が変更できるのですが、ここでは解説しません。Macintosh では 2005 年夏頃から正常動作しなくなったようです。yadif=0 を使いましょう。
pullup,softskip
逆テレシネフィルタ。正確にはインタレ解除ではありません。インターレースがかかる前の状態に復元することを念頭においているので、アニメでは極めて美しく解除できます。欠点は、インターレースをかけた上から字幕ロゴ等が合成してあるとその部分が荒れること、始めからインターレースのかかった状態で撮影されているような番組では当然正しく解除できません。pullup 直後の softskip は必ず指定しましょう。
yadif=0
解除性能は filmdint とほぼ同等。ただ、解除もれを起こさない点で filmdint に勝ります。
mcdeint=0
新しく登場してきたインタレ解除フィルタです。インタレ解除フィルタのくせに動き補償まで備えています。yadif よりジャギが少なく、設定次第で万能ナイフにもなりそうですがまだ出てきたばかりで未実装の機能等があります。あと、猛烈に遅いです。
hqdn3d=4:3:6
デノイズフィルタ。前から順に、空間軸輝度、空間軸彩度、時間軸のデノイズ強度です。hqdn3d=4:3:6 がもっとも良い、という結論になっています。DVD など、ノイズが入っていない素材では削ってもいいでしょう。
crop=720:480:0:0
クロップ。映像の上下左右を裁ち落とします。上下の黒幕を裁ち落とす際に使います。最初のふたつで残す映像の幅と高さを、あとのふたつで切抜きの開始位置を指定します。注意したいのは、一般的な MPEG2 動画は 720 × 480 になっています。再生時に、640 × 480 に縮小しているのです。ですから、そこを留意した上で切りぬきましょう。
scale=640:480:::3 スケーリング。上の crop で裁ち落とした映像を指定したサイズにリサイズします。これを省くと、リサイズを行わない、つまり crop してい

ない場合は 720 × 480 になって横に延びた映像になるので必ず指定しましょう。
最後の 3 は、リサイズする際に輪郭の滑らかさなどに影響します。アニメなら 3、実写なら 4 が適切。

`harddup`

x264 で変換する際は必ず指定しましょう。でないと音ズレします。必ず `vf` オプションの最後に指定すること。

変換が終了すると、拡張子が 264 のファイルが出来あがります。これはこのままでは再生できません。MP4 に映像と音声を入れる必要があります。それには MP4Box を使用します。MP4Box のバイナリは以下のサイトで取得できます。今度は Windows や Linux のバイナリも入っています。

`http://www.tkn.tu-berlin.de/research/evalvid/`

MP4Box をホームフォルダ直下において以下のコマンドを実行します。

```
./MP4Box -fps 23.976025 -add hoge.264 -add 音声ファイル.mp4 -new hoge.mp4
```

音声ファイルは、AAC のものをつかいます。MPEG2 ファイルから音声を抽出するには MPEG Streamclip[2]を使います。抽出したあとは普通に iTunes 等で変換すれば良いです。動画ファイルから音声を抽出するには、拡張子は m4a になると思いますが、mp4 に変更しておきましょう。`-fps` は出力 fps です。小数で指定しないと音ズレします。

コマンドを実行し終わると、hoge.mp4 ファイルを生産します。これで完成です。お疲れさまでした。他のファイルは必要ないので削除しても構いません。なお、QuickTime では再生できない場合があります。VLC[3]や MPlayer[4]といった再生ソフトを使いましょう。ちなみに、mencoder は mplayer についてくるエンコーダです。mplayer をビルド (コンパイル) すると mencoder もついてきます。本来 CUI で動くものですが、GUI 版も配布されているので心配せずに。

QuickTime で再生したい場合には、avc1encoder というプラグインを組み込んでやれば良いのですが、B フレームを使っているとまた厄介な問題が出てきます。ageha さんも苦労しているようです。暫定的な対応策などが ageha さんのホームページに記載されていますので、それを参照してください。

後書き

どうだったでしょうか？ GUI で変換するよりもより高度でマニアックなオプションが使えたことかと思います。これを自分の好みで組み合わせることで、画質の限界に挑めるでしょう。本当は mencoder や x264 のビルド (コンパイル) 方法も解説したかったのですが、ちょっと部誌を書きはじめるのが遅すぎましたね。というか、今はかなり切羽詰まった状況で書いています。締め切り 3 日前とかいうレベルを超えてますね。あとで怒られるかも...。まぁそういうわけなので、多少表現がおかしかったとしても気にしないでください。これで mencoder や x264 に興味を持ってくれた方はは ageha さんのホームページを参照すると良いです。というか、これ自体がホームページの内容を読みやすくまとめたような形になっていますので。それではまた会う日まで。

---

[2] `http://www.squared5.com/`

[3] `http://www.videolan.org/vlc/`

[4] `http://www.mplayerhq.hu/design7/news.html`

・・・・2・・・・

CD,DVD を焼いてみよう

62 回生 平野湧一郎
63 回生 佐野亮佑

※基本的に Windows の 2000 以降を前提にしています。
※決してコンロで焼くわけではありません。

はじめに

「CD・DVD を焼く」という表現の理解があなたはできるでしょうか。上記の通り、コンロやガスバーナーで焼くわけではなく、データを CD や DVD に保存することを言います。CD や DVD を焼く主な理由は、フラッシュメモリなんかではガッツではなく容量が足りないような重いデータのバックアップを取るためです。無理やり PC の電源を落とし、目が覚めたら、体が縮……ではなく PC のデータが消えていたなんてのはわりとよくある話です。ある日突然 PC のデータが飛んで、アッー！なんてことにならないためにも、CD・DVD を焼く方法を覚えておきましょう。

CD の焼き方

CD は、700MB のものが普通です。ですので、少し重めのデータを保存するのに適しています。一口に CD といっても色々ありますが、データを焼くことができるものとしては主に CD には CD-R と CD-RW があります。CD-RW は一度書き込んだデータを消すことがありますが、値段がやや高めです。一度書いたデータを消さないつもりなら CD-R の方がいいでしょう。ですが、初心者の方は何度でも失敗できる CD-RW の方がいいかもしれません。

CD を焼くにはライティングソフトというものが必要です。代表的なものは、burnatonce や DeepBurner、CDRWIN などがあります。ご自分のパソコンに B'z Recorder Gold Basic というものが入っていれば、それを使っても焼けます。基本的にソフトをインストールし、指示に従ってボタンを押していけば焼けますが、例として burnatonce での焼き方を示しておきます。

1. burnatonce をダウンロードし、インストール
2. 起動し、「ディスクの作成」→「データ CD」を選択
3. 焼きたいフォルダ（ファイル）をドラッグや「フォルダ追加」(「ファイル追加」) で追加し、「完了」を選択
4. 「書き込み」で書き込めます。

プレイヤーで再生できる CD の焼き方　↑のようにしても、CD プレイヤーで再生できる CD を焼くことはできません。プレイヤーで再生できるようにするには、「音楽

CD」を作らなくてはなりません。burnatonce、DeepBurner などで焼いてもいいですが、XP なら Windows Media Player を使うのが手っ取り早いでしょう。「ドライブから書き込み」で焼けたと思います。

DVD の焼き方

DVD は、通常（片面）4.7GB 入ります。ですので、かなり重い動画やゲームも大丈夫です。DVD は CD よりも種類が多いですが、ここでは DVD-R での焼き方を説明します（他の種類のものも同じようなものです）。CD と同じく、ライティングソフトを使うことになります。B'z Recorder Gold Basic が入っていればそれを使うといいでしょう。ない場合は、個人的に DeepBurner を使うのがいいと思います。一応使い方を書いておきます。

1. DeepBurner をダウンロードし、インストール
2. 起動し、「データ CD/DVD の作成」→「マルチセッションを作成」を選択
3. 右側にスペースが開くので、焼きたいフォルダ（ファイル）をドラッグ
4. 「書き込み」→「書き込み」で書き込めます。

終わりに

フリーソフトをダウンロードできるアドレスを載せておきます。

- burnatonce
  `http://www.altech-ads.com/product/10001572.htm`
- DeepBurner
  `http://www.forest.impress.co.jp/lib/sys/hardcust/cddvdburn/deepburner.html`
- CDRWIN
  `http://www.altech-ads.com/product/10001573.htm`

ここまで読んでくださってありがとうございました。この文章が少しでも貴方のお役に立てば幸いです。

では、頑張って文化祭作品を仕上げてきますw

・・・・3・・・・

暗号入門じみたもの

60 回生 山本真吾

いつしか部誌を書かねばならない季節がやってきて、過ぎて行きつつある[1]。昨年予告したように、暗号にまつわる小話を書いてみようと思う。ほんの少しの暇と数学の勘をポケットに突っ込んで駆け出そう。夜明けまでまだ三時間ある。

アリスとボブには伝えたい言葉がある

とりあえず、状況をわかりやすく説明するために、登場人物をでっち上げておく。名前はアリスとボブ[2]。人間である、以外の設定は勝手に考えてくれて構わないが、本題とは何の関係もない。

アリスは、ボブに伝えたいことがあるとしよう。それは明日の天気に関する国家機密かもしれないし、コロネをどこから食べるか、みたいなつまらない話かもしれないが、そのあたりは重要ではない。とにかく他人に聞かれたくない内容なのであるが、同時に今すぐ伝えたい内容でもある。電話をすればいい？なにを言っているんだ、今この瞬間も彼らの秘密を暴こうと全世界の諜報機関が盗聴を試みているというのに、そんな無防備な真似ができるものか。

こういった非常に特殊な状況で利用されるのが、暗号通信である (冗談だ)。本稿の目的は、いくつかのよく知られた暗号アルゴリズムをちょっぴり数学風に解説することである。数学は苦手なので、厳密さに関しては目をつぶってもらいたいところだが、本当に目をつぶってしまっては読めない。ディレンマである。

アリスとボブは秘密の手紙を書く

さて、現代では暗号は個人の機密保持などにも利用されているが、本来暗号は軍事機密保持のために利用されるものであった。その歴史は古く、紀元前十九世紀くらいまでさかのぼることができるらしい。ヒエログリフで書かれた文章の中に、標準以外の文字を使って書かれたものがあるそうだ。

●● Caesar 暗号 ●●

古代に利用された暗号の中でとりわけ有名なのは古代ギリシャだかローマだかの指導者 Julius Caesar[3]が利用した「Caesar 暗号」だろう。彼は時に、アルファベットを 3 文字ずらして[4]手紙を書いた。こうすることで、万一手紙が敵の手に落ちても簡単に内容を知られるということはない。が、あらかじめ 3 文字ずらされていることを知ってい

[1] まったく申し訳ない。

[2] こういう場合に伝統的に使われる名前というものがあって、それが Alice, Bob, Chris, David. . . なのだ (諸説あるけれど)。頭文字によるのだろう。

[3] ジュリアス・シーザー。「ユリウス・カエサル」の方が一般的かもしれないが、「カエサル」ってかっこ悪いから嫌いだ。

[4] `a`→`D`, `b`→`E`, ..., `z`→`C` という具合に。

る味方には復号[5]することが可能だ。このとき、ずらされた文字数 3 は暗号鍵としての意味を持つ。

残念ながらこのタイプ[6]の暗号は解読が非常に容易だ。何しろ暗号鍵は 26 通りしか存在しないし、うち 1 通りは平文と同じになるから除外できるので実質 25 通りであるから、人間の手でその全てを調べることが十分可能で、しかも容易だ。

なお、3 文字ではなく 13 文字ずらしたものが、現在でも ROT13 などと呼ばれて使われることがある。もちろん、機密保持のためなどではなく、主にネタバレ防止などのためである。なぜ 13 なのかというと、アルファベットが 26 文字なので、もう一度暗号化すると元に戻るからである。暗号化と復号を同じ作業で行えるということだ。

●● 換字式暗号 ●●

Caesar 暗号では文字の置き換えに規則性があったが、もう少し複雑にして、文字をまったくばらばらに置き換えてしまう。このように文字を一文字または複数文字ごとに置き換えるタイプの暗号を総称して換字式暗号[7]と呼び、一文字ずつ置き換えるものを特に単一換字式暗号と呼ぶ[8]。一般の換字式暗号は Caesar 暗号よりも解読がやや困難であるが、まったく残念なことに現在ではやはり非常に容易である。

単一換字式暗号の場合について説明しよう。文字を置き換えるだけという性質から、平文中の同じ文字は暗号文中でも同じ文字に置き換えられる。ということは、それぞれの文字の出現頻度は一定に保たれるということだ。標準的な英文における各文字の出現頻度はすでに調べられているから、あとは暗号文中の文字の出現頻度と照らし合わせて見当をつけていけばいい[9]。このような手法を使うと素人でもすぐに解読できてしまう。

さらに、換字式暗号にはもうひとつ弱点が存在する。それは、鍵が複雑になるということだ。Caesar 暗号ではずらした文字数だけでよかったが、この場合はそういった単純な規則性が存在しないから、どのように文字を置き換えたのか示す必要がある。それはランダムに決められた文字の並びだったり本に書かれた文章の一節だったりしたわけだが、とにかくこういった特徴が鍵配送問題 (後述) をより困難にしてしまった。

アリスとボブは数学の女王と遊ぶ

●● 数学してみる ●●

さて、前節であげた Caesar 暗号を数式で表してみよう。数式で表すということはつまり暗号を数学的に取り扱うということだ。26 文字のアルファベットを 0 から 25 までの数字で表すことにする[10]。と、Caesar 暗号は次のように書き表せる。

$$C = (P + 3) \bmod 26$$
$$P = (C - 3) \bmod 26$$

---

[5] 暗号文を平文 (ひらぶん) へと戻す作業。ただし、鍵を知らない第三者が無理矢理行う場合は「解読」と呼ばれて区別されることが多い。

[6] Caesar が利用したのは 3 文字ずらすものだが、それ以外の文字数のものも Caesar 暗号と呼ばれる。

[7] かえじしきあんごう

[8] Caesar 暗号も単一換字式暗号である。

[9] たとえば暗号文中でもっとも出現頻度の高い文字は標準的な英文で最も出現頻度の高い文字である E を置き換えたものではないか、など。ただし、あえて E をまったく使わずに書かれた小説も存在するから、常に有効とは限らない。

[10] a→0, b→1, ..., z→25

P は平文、C は暗号文をあらわす。あと、謎の記号 mod は、剰余を意味する。たとえば $7 \bmod 3 = 1$ だし、$-2 \bmod 5 = 3$ だ[11]。つまりこれらの式は

- 暗号化するときは 3 を加えて 26 で割った余りをとる
- 復号するときは 3 を引いて 26 で割った余りをとる

ということを表している。もうちょっと数学してしまおう。暗号化のプロセスを関数 $f(x)$ だと考えると

$$f(x) = (x + 3) \bmod 26$$
$$f^{-1}(x) = (x - 3) \bmod 26$$

となる[12]。

●● mod と合同式 ●●

実は、mod にはもうひとつ使い方がある。どういう風に表現したものかわからないので実例を挙げておく。

$$3 \equiv 8 \pmod 5$$

意味は伝わるだろう。3 も 8 も mod5 した値が等しいということ[13]で、「3 と 8 は 5 を法として合同である」と読む。

法が等しい合同式に関する規則をいくつか並べておこう。なんだか本筋から外れ気味に見えるかもしれないが、そもそも本筋など気にしたためしがないし、あとで数学的解説をするときに必要になるので続ける。

加算と減算

$$3 \equiv 8 \pmod 5$$
$$1 \equiv 6 \pmod 5$$

合同式を複数並べて加算することができる。上記の例だと

$$4 \equiv 14 \pmod 5$$

となり、正しいことがわかるだろう。同様に減算も可能なのだが、いちいち式を書くのが面倒なので自分で確かめてほしい。あと、証明も割愛する。

乗算と累乗

さらに乗算も可能だ。

$$\begin{array}{rrrl} & 7 \equiv & 12 & \pmod 5 \\ \times & 4 \equiv & 9 & \pmod 5 \\ \hline & 28 \equiv & 108 & \pmod 5 \end{array}$$

[11] 二つ目の式はいささか直感的ではないかもしれないが、-2 に 5 を加算してみると納得できるだろう。

[12] $f^{-1}(x)$ は逆関数、つまり復号を表す。それにしても、-1 乗とは気が利いていると思う。

[13] 難しい言葉では「3 と 8 は 5 を法とする同じ剰余類に属している」と表現するらしい。

ということで、累乗も可能だ。

$$7^4 = 2401 \equiv 20736 = 12^4 \pmod 5$$

これは少しばかり便利な性質だ。たとえば、10^{100} を 7 で割った剰余を求める場合を考える[14]。これは 10^{100} 日後の曜日を知りたい場合に役に立つ (冗談)。

まず、

$$\begin{aligned} 10 &\equiv 3 \pmod 7 \\ 10^6 \equiv 3^6 &= 729 \pmod 7 \\ &\equiv 1 \pmod 7 \end{aligned}$$

を求めておく。6 という数字は右辺を 1 にすべく天啓に導かれて得た。あとはこれをさらに 16 乗して

$$\left(10^6\right)^{16} = 10^{96} \equiv 1^{16} = 1 \pmod 7$$

さらに足りない分を計算し、掛ける。

$$\begin{array}{rl} & 10^{96} \equiv 1 \pmod 7 \\ \times & 10^4 \equiv 4 \pmod 7 \\ \hline & 10^{100} \equiv 4 \pmod 7 \end{array}$$

ということで、今日が火曜日なら 10^{100} 日後は土曜日である[15]。やった、週末だ!

原始元　素数 7 があったとする。このとき、7 を法としたときの 3 の累乗を考えると、

$$\begin{aligned} 3^0 &\equiv 1 \\ 3^1 &\equiv 3 \\ 3^2 &\equiv 2 \\ 3^3 &\equiv 6 \\ 3^4 &\equiv 4 \\ 3^5 &\equiv 5 \\ 3^6 &\equiv 1 \pmod 7 \end{aligned}$$

となる。1 から 6 まで一巡しているのがわかるだろう。こういう場合、3 は $\mathbb{Z}_7$ の原始元である、という[16]。p が素数ならば、$\mathbb{Z}_p$ には必ず原始元が存在するらしい。ああそうだ、5 も $\mathbb{Z}_7$ の原始元だ。

逆数　たとえば、3 に 1/3 をかけると 1 になる。こういう場合、1/3 は 3 の逆数である、という。ふつう逆数といえば、分子と分母を逆転させたものであるが、さて、mod$p$ になると話は別になってくる。たとえば、

$$(3 \times 5) \bmod 7 = 1$$

---

[14] ちなみに、この値 10^{100} を googol と呼ぶ。検索サービス google の名はこれに由来するが、開発者がスペルを間違えたために微妙に異なっているらしい。本当だろうか。

[15] グレゴリオ暦が使用されていることを前提としているから、もしかするとあまり意味のない計算なのかもしれない。

[16] 本当はもう少し厳密に書きたかったのだけれど、ここには十分な余白がないし、決闘前夜なので時間がない。

となるから、5 は 3 の逆数となっている。

●● もっと数学してみる ●●

さて、Caesar が使ったのは 3 文字ずらすものだったが、それ以外の文字数のことも考えてやりたい。ということで、

$$f(x) = (x + K) \bmod 26$$
$$f^{-1}(x) = (x - K) \bmod 26$$

としよう。ここで K は鍵である[17)]。

ところで、このままでは暗号として心もとない。鍵が 26 種類しかないのなら、誰だって復号できる。そこで、複数の文字をまとめて暗号化することを考えよう。たとえば二文字ずつの場合、文字の組を aa, ab, ac, ..., zy, zz という順に並べて、0 から $26^2 - 1 = 675$ までの番号をつける。この新しい暗号は $f'(x) = (x + K) \bmod 676$ と表され、鍵 K は 0 から 676 までの値をとることができる。鍵の可能性が増えたということは第三者による復号が困難になったということである。

しかし残念なことに、このタイプの暗号はこの程度では簡単に解読されてしまう[18)]。かといって鍵の種類をもっと増やすと計算に時間がかかりすぎる[19)]。ということで、シーザー暗号は現在ではほとんど探偵ごっこか上述の ROT13 のような用途にしか使われていない。

アリスとボブは鍵を共有する

今度は問題を違った方向から考えてみよう。とりあえず、第三者にやすやすと解読されることのない暗号があったとする。と、どのように鍵を安全に伝えるかが重要な問題になってくる。これは鍵配送問題といって長い間解決されなかった問題であるが、戦後になって暗号が数学的に研究されるようになり、暗号界に「公開鍵暗号」という革命が起こった。

そんなわけで 1976 年に提案されたのが Diffie-Hellman 鍵共有[20)]である。これは、盗聴されている可能性のある経路を通して二人の人間が同じ情報を共有するためのプロトコルである。盗聴している第三者はその情報を共有することができないようになっている。つまり、その情報を暗号鍵として利用することができるということだ。すばらしい。

それなりに大きな素数 p と、その原始元 n を用意する。これらは公開して構わない。札束の裏に書いて東京タワーからばら撒いても構わない。

次に、アリスとボブはそれぞれ秘密の数 A_a, A_b をランダムに選ぶ ($0 \leq A_a, A_b \leq p - 2$)。選んだら、それぞれ

$$B_a = n^{A_a} \bmod p$$
$$B_b = n^{A_b} \bmod p$$

を計算し、相手に送信する。一方で A_a や A_b は秘密にしておく。

最後に、アリスは

$$K_A = B_b^{A_a} \bmod p$$

[17)] 本来は暗号鍵と復号鍵を区別して K_E, K_D としなければならないところだが、Caesar 暗号の場合どうせ同じなのでまとめて書いた。

[18)] コンピュータのある現在ならなおさらだ。

[19)] と、思う。

[20)] ディフィー・ヘルマン

ボブは

$$K_B = B_a^{A_b} \bmod p$$

を計算する。生真面目に計算するとわかると思うけれど、

$$\begin{aligned} K_A &\equiv B_b^{A_a} \\ &\equiv \left(n^{A_b}\right)^{A_a} \equiv n^{A_a A_b} \equiv \left(n^{A_a}\right)^{A_b} \\ &\equiv B_a^{A_b} \equiv K_B \pmod{p} \end{aligned}$$

となる。これでアリスとボブは同じ値を共有することができた。鍵配送問題の解決である。

本当にこれで問題ないのだろうか？たとえば B_a, B_b が盗聴されていたとしよう。n と p も既知だとする。これらの値から K_A あるいは K_B を求めることはできないのだろうか？

実は、できなくは無い。のだが、これは離散対数問題といって、現時点では効率的な解法が存在しない[21]。あるいは、$n^{A_a} = B_a$ とわかっていて、しかも $0 \leq A_a \leq p-2$ だから、総当りすれば正しい値にたどり着くことはできる。ただし、p は非常に大きな数[22]なので、そうやすやすと破られたりはしない。

アリスとボブは鍵を公開する

1977 年に Rivest, Shamir, Adleman によって発明された暗号が RSA 暗号で、現在でもいろいろな形で使われている。このあたりまで説明して終わりにしたい。我ながらよくがんばったものだ (気が早い)。

●● 鍵の公開と暗号化の手順 ●●

公開鍵暗号は、従来の暗号とはすこしプロトコルが異なっている。通信の手順を簡単に説明する。アリスがボブにメッセージを送ることを想像してほしい。

1. アリスは、どこからかボブの公開鍵を探してくる。
2. アリスは、送信するメッセージをボブの公開鍵で暗号化する。
3. メッセージを送信する。
4. 受信したボブは、自分の秘密鍵でメッセージを復号する。
5. 黒ヤギさんたら読まずに食べた。

最後の手順は不要である (真顔)。

これを見ると、公開鍵暗号には「公開鍵」と「秘密鍵」の二種類の鍵が必要なことがわかると思う。二種類の鍵は対になっていて、公開鍵は暗号化、秘密鍵は復号を行う[23]。

●● もう少し数学 ●●

数学が嫌いな読者が読んでいるならば、この節は飛ばしたほうがいいかもしれない。が、そこまで難しくはないと思う。

---

[21] 量子コンピュータが実現すれば解けるらしいが、よく知らない。

[22] それこそ googol とかでも足りないくらいに大きい。

[23] 実は、署名をする際などには逆のことが行われているのだが、混乱するといけないので黙っておく。

Euler の ϕ 関数　自然数 n より小さく、n と互いに素な自然数の個数を $\phi(n)$ とあらわす。素数 p, q に対して $\phi(pq) = (p-1)(q-1)$ となる。

Fermat の小定理　Fermat[24]といえば最終定理しか思いつかない人も多いだろうが、別に彼がそれしか仕事をしていないわけではない。そんな彼の定理の一つに Fermat の小定理がある。素数に関する定理で、以下のようなものだ。

$$a^{p-1} \equiv 1 \pmod{p} \quad (p \text{ は素数、} a \text{ と } p \text{ は互いに素})$$

現在では、この定理の対偶が素数判定に利用されている。あと、下で述べるけれど Euler が拡張したものが RSA 暗号では重要な意味を持つ。

Euler の定理　a, n が互いに素な自然数であるとき、

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

となる。n が素数ならば、$\phi(n) = n - 1$ からこれは Fermat の小定理と一致する (あたりまえだ)。

●●　鍵の算出と暗号化　●●

まず、それなりに大きな素数 p, q を選び、$n = pq$ と、$\phi(n) = n - (p + q) + 1$ を計算する。次に、$\phi(n)$ と互いに素で $\phi(n)$ より小さい自然数 e を選び[25]、$\bmod \phi(n)$ における e の逆数 d を計算する。このとき、

$$\begin{aligned} \text{公開鍵 } K_E &= \langle n, e \rangle \\ \text{秘密鍵 } K_D &= \langle n, d \rangle \\ f(x) &= x^e \bmod n \\ f^{-1}(x) &= x^d \bmod n \end{aligned}$$

となる。

●●　確認してみる　●●

なんだか妙にきれいにまとまっているが、実際のところ正しく機能するのだろうか？確かめることにする。目指す場所は、$f^{-1}(f(x)) \equiv x \pmod{n}$ だ。とりあえず代入してみる。

$$f^{-1}(f(x)) \equiv x^{de} \pmod{n}$$

$de \equiv 1 \pmod{\phi(n)}$ から、

$$\begin{aligned} &\equiv x^{(X\phi(n)+1)} \pmod{n} \\ &\equiv x^{X\phi(n)} x \pmod{n} \\ &\equiv \left(x^{\phi(n)}\right)^X x \pmod{n} \end{aligned}$$

n, x が互いに素ならば、Euler の定理から、

$$\begin{aligned} &\equiv 1^X x \pmod{n} \\ f^{-1}(f(x)) &\equiv x \pmod{n} \end{aligned}$$

---

[24] フェルマー。弁護士にしてフェルマー予想の主犯。　[25] あまり大きいと面倒なので 17 だとか 65537 がよく使われる

となる。n, x が互いに素でない場合にも、いろいろとこじつけて証明できるのだけど、面倒なので、割愛する。

●● 安全性 ●●

$\langle n, e\rangle$ は公開鍵で、$\langle n, d\rangle$ は秘密鍵だと書いた。ならば、公開されている e と n から d を計算できれば秘密鍵を入手したことになり、暗号は破れたことになる。その可能性を考慮してみる。

d は $\mod\phi(n)$ における e の逆数である。逆数を求める計算自体は Euclid の互除法を用いて簡単にできる[26)]。重要なのは、$\phi(n) = n - (p + q) + 1$ が計算できるかどうかだ。これはどうやら困難である。なぜなら、p, q が公開されていない以上、自分でそれを計算しなければならないのだが、それは何百桁という数 n を素因数分解しなければ得られないからである。そして、素因数分解に対して有効なアルゴリズムは今のところ存在しない[27)]。したがって、RSA 暗号は解読が困難であるとされている[28)]。

アリスとボブは締切に遅れる

時刻は午前五時半。本当はもう一日かけてゆっくり書きたかったのだけど、締切はもう過ぎているので、そんな悠長なことを言っているわけにもいきませんでした。そういいつつ悠長にあとがきなど書く僕です。

「暗号」というとなんだか縁の無いものにきこえがちですが、実際は生活の中にすっかり溶け込んでいます。たとえば Web サイトのアドレスが `https:`から始まったりしていたらそれは暗号化通信が行われているサインです (たぶん)。そういえば、何かのカードの暗号が破られたとかそういう話もあったような。

現在でも暗号は日々進歩していますが、とりあえず正しく利用すれば解読されない段階にはあると思います。量子コンピュータの実用化もまだしばらくかかりそうだし。しかし、それは暗号を正しく利用した場合の話であって、そうでない場合にはすぐに破られてしまいます。第二次世界大戦のドイツ軍も、強力な暗号を開発したにもかかわらず運用上のミスから解読されたりしています。それに、人間というのはダメな生き物なので、目の前に大金を積まれたらすぐに何でも話してしまいます。ソーシャルエンジニアリングとかいいます[29)]。

そういったことを防ぐには、何より暗号を利用する人間がそれを正しく理解することが必要だと思うのです。本稿がその助けになればいいな、などと書きますがそんな価値のあるものだとは思っていません。良くてきっかけ程度でしょう。それでも、より多くの人が関心を持つことは大切だと思うので、関心を持ってください (懇願)。

なんだかうまくまとまったような気がして上機嫌です。学校に行くまでにすこしは眠りたいのでここらで終わりにしたいと思います。ありがとうございました。

[26)] でなければ、鍵の生成に時間がかかりすぎる。

[27)] 量子コンピュータによる効率的なアルゴリズムは存在する。これは脅威となりうる。

[28)] しかし、RSA 問題が素因数分解とそっくり等価であることは示されていない。つまり、素因数分解以外の方法で解読できる可能性が無いと決まったわけではないということである。

[29)] そういえば暗号に対する攻撃についてほとんど触れていない。OB 特別寄稿の機会があればやろうかとも思う。

・・・・4・・・・

われらのパソコンを取り巻くものたち

61 回生 浜田克紀

前書き

電気会社の広告などをみてみると様々な周辺機器が発売されているのがわかりますが、最近はいろんなメーカーから多種多彩な商品が出てきているのでよくわからず困っている方も多いことでしょう。

ここでは、普通の家庭用パソコンでならまあ使う可能性があるものを紹介していきます。軽い気持ちで読んでくれるとありがたいです。

接続方式

とりあえず機器の説明の前に、つながれるパソコン側の話でもしましょう。いきなり機器の説明をしてもつなぐところの話をしておかないと本質的にちんぷんかんぷんでしょうから。

普通のデータ転送するための接続形式は USB、PC カードスロット、CD,DVD ドライブです。

●● USB(Universal Serial Bus) 接続 ●●

外部からデータを取り込んだり逆に電力を供給したりするときに普通使おうであろう形式。これには規格が二つあり、USB1.1 と USB2.0 です。USB1.1 は旧世代の規格で、そこまで多くないデータ転送ですむキーボード、マウス、プリンタなどの接続に今も使われています。昔 1.0 というのがあったそうですが、あまり見ないのと 1.1 と相違点がほとんど認められない (ちょっとアップグレードした感じ) なので、無視してもかまわないと思います。

転送速度は、Low Speed で 1.5MBps(一秒間に最大 1.5MB 送れるということ)、High Speed で 12MBps です。正直、今の音楽とかのデータ転送にはまったく向いていません。マウスやキーボード接続などに使ったりします。

USB2.0 というのは 2000 年にできた新規格で、480MBps とかいう 1.1 からは想像できないような値を出した規格です。これが今の主流です。

パソコンから USB に電力を送れるので、それを利用した面白いもの (クリスマスツリーとか) も最近売り出されているようですね。キーボードや画面などに USB 分岐がついている例も珍しくなくなってきています。

それ以後は新しい規格の案は出されていないので当分このままでしょう。使える OS は win95 以降 (安定して使えるのは 98 以降)、Mac OS8.1 以降となっています。

●● PC カードスロット ●●

これは PC カードと呼ばれるカード上のものをさすためにあります。USB とはパソコン内のマザーボードと呼ばれるところでの接続している位置が違います。昔は PC カードの外付け HDD などがありましたが、最近のフラッシュメモリの大容量化、さらに様々な機能をパソコン内部に取り付けるように代わって行ったためなどの理由により USB との差異がほとんどなくなってきています。現在使われている PC カードといえばメモリカードアダプタ、無線 RAN カードなどに限られてきています。PC カード自体が新めなのでまだ後続の企画が普及することはないでしょう。

●● PS/2 ●●

プレステ 2 ではありません。3 もありません。

主にキーボードやマウス、場合によってはゲームパッドなどをさすところです。最近は USB が一般化してきつつありますが、まだどちらも健在です。

機器の種類とか

データを持ち運びするためのものは CD,DVD、USB メモリ、外付け HDD、フロッピー[1)]です。

●● USB メモリ ●●

良く見かけると思いますが、USB メモリというのは USB 接続で、データの持ち運びに用いられるものです。非常にコンパクトですが、結構な量のデータを持ち運びできるという使い勝手の良いものです。今出ている最高のものが 8GB ですが、どんどん質が上がっていますので更に大容量のものがすぐに出るはずです。ですので 2 年前に買った 32MB の新品のものの値段よりも安く今 256MB の新品のものが手に入るということはざらです。(なんだか損した気分です。) ちなみに USB メモリの容量は 2 の累乗で表されています。

外付け HDD との違い
表面的な話

- コンパクトさ
  メモリスティックは 100g もなく、指よりも短いものがほとんどですが、外付け HDD はもっとずっと大きいものばかりです。それでも最近はまあまあコンパクトになってきていますが
- 容量の差
  これは圧倒的に外付け HDD のほうが大きいです。まあ図体だけ大きくて中身が無いというのも困りますからね。
- 転送速度
  メモリのほうが速いです。といっても外付け HDD も見るに耐えない速さというわけではないですが。

---

1) 現在はもうほとんど使われていない

- データ保存の仕組み
  HDD のほうは、簡単に言うとプラッタと呼ばれるアルミやガラスでできていて、片面だったり両面だったりする円盤にデータを記録するところがあり、それを回している間に磁気ヘッドという先のとがったもので読み書きしていくという仕組みです。読み書きをしていない普段の状態ではヘッドはプラッタの外側にあります。このプラッタの回転速度 (単位は rpm,5400,7200,10000 が一般的)、ヘッドが指示されたデータの位置まで行くまでの時間 (シークタイム)、ディスクのデータを読み書きする時間 (データ転送速度) などにより HDD の性能が決まってきます。
  ヘッドがずれて変なところに書き込んだりとかガラスなどでできたプラッタが割れたりなどというのは衝撃によって引き起こされるので、一般に HDD は衝撃に弱いということになるわけです。
  それに対しメモリは電荷を保持するという方法をとっている。わかりやすく言うとパソコンにつないで電力がメモリに供給されているときはメモリ内の電子は動けるが、そうでない場合は絶縁されていて、電子は漏れ出さない状態になっている。その状態を保持することによりデータ保存が可能となっているわけだ。
  当然 HDD よりも衝撃に強い。電気的なショックなどに弱いとか言われるかもしれないが、それは機械物ならすべてにいえることなので問題なし。4 階から例の 32MB のを落としても大丈夫だった。HDD なら間違いなく壊れる。
  結論として、メモリは外付け HDD よりはコストパフォーマンスはよくないが、速さと安定性とサイズ的にはとてもすぐれているということだ。
  ちなみに今話題の iPod はメモリ型のと HDD 型のとがあり、今の最新の世代は容量が大きい (30GB 以上) のは HDD で、それ以外はメモリとなっています。

●● キーボード ●●

文字を打つためのものです。言わなくても (略)

キーボードは PS/2 と USB の規格があります。それ以外はあったとしても存在すら知らないマイナーなものだと思うのでここでは書きません。

キーボードのキー配列には色々あり、上のほうに Internet や power キーなどが付いているもの、テンキー (普通のキーボードの右にある数字などの書いてあるキーたち) がないもの、逆にテンキーだけのものなどがあり、画一的なものはありませんが、少なくとも a〜z、1〜10、それらを囲んでいるキーたち (esc 半角 tab...) などのもの、それと F1〜F12 は普通は付いています。(テンキーキーボードは電卓などに使われる。)

さらにキーだけでなく形状なども様々で、ぐにゃぐにゃに曲がるもの、暗闇で光るもの、キー配列が斜めや普通とは違うものなどおもしろいものが多数あります。後、機械式のキーボードはクリック音はするが電子式は音はしない。なので意図的に音を出すようにする傾向が見られ、それが高じてクリック音が盛大にするキーボードなども今はあるそうだ。

●● マウス ●●

接続規格は上と同じです。

大きな変化をあげていくと、ボールマウス (すべて PS/2)、光学式マウス (混合)、レーザーマウス (混合) の順に進化してきました。

ボールマウスは、Windows95 などの時代に使われていたマウスの底面に仕込まれたボールが動きを感知し、その動き具合でカーソルを動かすというものです。ただ結構アナログ的 (？) なのでゲームなどをする際に動かしにくさに悩まされるはずです。

光学式マウスは、今でもよく使われているもので、マウスの底面にある発光機から光を出し、その光の量、方向、速度を受光機で感知してマウスポインターを動かすものです。ボールマウスよりもずっと使い勝手がいいですが、まだマウスを置く場所が不安定だと動きも不安定になるという問題も抱えています。まあマウスパッドの上で使用する際は気にならないと思いますが。最近のものは画像処理機能によりそんなところでもちゃんと光を感知してくれるものもあるようですね。

レーザーマウスは名前の通りレーザー光で動きを感知してカーソルを動かします。光学式と仕組みは一緒ですが、高性能のレーザー光を用いているので安定した読み取りが可能なので、自分も気に入ってます。

あと、近年はマウスの真ん中にくるくる上下に回せるボタンが付くようになりましたが、それはホイールです。これの付くマウスを左右 2 ボタンマウスと区別してホイールマウスといったりもします。これは画面のスクロールをするためのボタンで、実は押すこともできますが、最近は横にも動かせるものがでてきたりしました。ないと結構つらいです。

形状は底面が平べったい楕円形に二つ (ホイールあわせると 3 つ) が典型的ですが、ものすごく小さなマウスやマウスの上のボールを動かすもの、ジョイスティックなど多種多彩に自分の好みのマウスを使えるようになってきました。

ちなみに、Windows ユーザーには縁のないことですが、Mac はつい最近までワンボタンマウスできていました。(今もワンボタンで使えるらしいが)

どうでもいいことですが感度の単位はミッキーらしいです。ねずみだからですかね。あのディズニーのキャラなのにうるさく言われないのでしょうかねえ。

●● スピーカー、イヤホン ●●

今ではパソコンで音楽を聴くことは普通です。動画をみるにも音無しだときついですしね。そこで大体のパソコンにはスピーカーがついています。[2)]

つなぐ端子は普通は 6.3mm のステレオジャック、3.5mm ステレオミニジャックなどですが、パソコンで主に用いられるのはミニのほうです。

USB のものもごくたまに見かけますが、ほとんど変わりません。USB をほかの用途に回すためにも、イヤホン専用端子のほうがいいと思います。

最近のタイプには、カナル型、骨伝道型、ノイズキャンセリング機能つきなどがあります。

カナル型　かなるちゃんの声が聞けます。(誰?

嘘です、すみません。耳の中に短いゴムチューブを入れて、俗世間との空気が混ざり合うのを遮断しつつ音を伝導するシステムです。まあ普通の人が日常生活で使うなら雑音は大体消せるでしょう。

[2)] テレビのように画面と一体化しているものもあるが、原理やつなぐところなどはすべて同じである

骨伝導型　こつでんどうと読みます。

最近頭にイヤホンやヘッドホンをつけている変態をよく見かけると思いますが、それが骨伝導型のイヤホンです。頭骨からなんと内耳まで音を伝えていって直接純粋な音を聞けるというのが売りです。これから広まっていくでしょう。

ノイズキャンセリング機能　さっきからハード的に音を消してるけどデジタル的にノイズキャンセルできないのかと思う人もいるでしょう。この機能は音と逆周波数の音を出してノイズを消してしまうというものです。最近アメリカ軍が無線用に開発したものですが、実は会話は聞こえるのに雑音は除去されるという優れもの。音にこだわるが会話はしたいという方にはおすすめの一品です。(どこの宣伝ですか？)

●●　CD,DVD　●●

CD の主な規格としては、-R、-RW があり、
DVD の主な規格としては、+R,+RW,-RAM,-R,-RW があります。

また、このほかに最近はブルーレイ (blu-ray) ディスク (sony などが開発)、HD DVD(東芝などが開発) という 2 大規格が争っています。

容量は、CD は 700MB、DVD は片面一層 (普段使うもの) が 4.7GB、片面二層 (TSUTAYA などでおいているやつとか) は倍の 9.8GB となっていて、ブルーレイは片面で 25GB、両面で 50GB、HD DVD はまだ発売されていないものの片面 3 層で 45GB や 51GB などと格段に大容量化してきている。

規格の違いについては、-R は書き込みのみ、追加書き込み不可、-RW は読み書きができるが、HDD などのように上書き方式ではなくいったんフォーマット (厳密には違うが、簡単に言えば中身をすべて真っ白にすること) してから書き直す方式をとっている。大体 1000 回の書き直しが限度らしい。-RAM は上書きもできるもの。ただ、遅いので嫌う人もいる。書き直しはなんと 10 万回。一生ものですね[3)]

なぜ容量がこれほどまでに違うかは、そもそも焼くための光線の波長の違いによる。当然波長が短いほど密度が高くなるので多く書き込める。具体的にいうと、CD は赤外線レーザー、DVD は赤色レーザー、ブルーレイは名前のとおり青色レーザーを使って焼く。

ライティングについては CD,DVD を焼いてみようの記事で。

●●　フロッピー　●●

CD などの前身となった前世紀の遺物です。今読めるパソコンを持っているほうがびっくりです。容量も音楽一曲入れるのが難しいというひどいもの。

規格としては 1969 年の読み取り専用の 8 インチ[4)]型→最盛期は 5.25 インチ、3.5 インチのものが普及

対応 OS としては今の Vista ではもう無理だろうが、XP だとぎりぎり 5.25 インチが対応している。[5)]当然 Windows ができる前の DOS(黒画面) 時代からありました。

1969 年〜1990 年代まで栄えた記憶媒体の一つで、硬いカバーの中にやわらかい円盤状のもの (という) が入っている構造になっている。このにデータを書き込む使用になっているのだが、初期の頃はその円盤をつつむものがやわらかくて壊れやすかったそうです。

[3)] その前に物理的にがたがきそう

[4)] およそ 20cm

[5)] なぜか一番売れた 3.25 のほうは無理。

これまた今では考えられないことだが、ゲームなどのデータをたくさんのフロッピーにいれて売っていたそうだ。自分も見たことしかないですが。

最後に、ここに書いてあることは 5 年後くらいにはまったく役に立たない過去の知識となっているかもしれません。そういう世界だということを知っておいてもらえたら幸いです。

あとがきみたいなもの

どうも、npca の現部長 katukky(浜田 克紀) です。今回はおそらく OB 達のほうがよりよい文章を書かれていると思いますが、この文章を読んで役に立つことがあったら幸いです。

今年一年、部室によく来るメンバー達が数々の理由でいなくなったり、新たに中 3 の人たちが入ってきたりと部内の雰囲気がとても変わりました。プログラミングの技術以外に CG や midi などの技術を持った人間が入ってきたことは個人的にはとてもいいことだと思います。忘れもしない 9/21、突然自分が部長にさせられたり落雷によりサーバーが死亡したりと色々とあわただしい一年でしたが、充実はしていました。

最後に、この部誌を読んだ方の中に宇宙人、未来人、異世界人、超能力者がいたら、私のところに来なさい。以上

·····5·····

おわりに

前部長 山本真吾

文化祭が終わった。「まだ始まってもいない」という声が聴こえてきそうだが、僕のように受験生の身ゆえ文化祭で作品を展示しない部員にとっては、この部誌を書き終えた段階で文化祭は終わっているのである。文化祭当日は、死んだような目をした部員達とつまらぬ話に花を咲かせているだろう。あるいは眠っているかもしれない。

現在、npca は危機にある。部員のプログラミングスキルは低下の一途をたどっている。もちろん、プログラミングだけが npca の活動ではないのだから、それもまた時代の流れか、と思わないでもないのだが、どうもいまの部員には向上心というか、熱意が足りないと思う。今年の部誌が去年に比べて妙に薄いのはそのためだろうと勝手に想像している。いけない、ついつい説教じみたことを語ってしまった。歳のせいだろうか。

そんなわけで、この部誌を読んで npca という謎の集団に興味を抱いたならば、中学生の教室が密集している古い建物の四階、地学教室の横にある小部屋を訪れてほしい。熱意ある新入部員を心待ちにしている。最悪、未来人とか異世界人とかでもいいや、もう。