

父と娘

「私の男」をこの前、待ち合わせ場所の本屋さんで買ったので読んでいます。今は、一章が終わって二章に入りました。ふすまの奥に何があるのかとても気になるのですが、それはまだ先のことみたいなので、一章で気になったことでも書いてみます。

わたしは男なので、父と娘という関係がうまく理解できません。ですが、わたしの妹を観察したり、たまに見かける女性の言葉を見かけて思うところがあります。

曰く、娘が一番最初に恋するのは父親だとか。また、父親にとって娘は絶対に手にできないのに大事に守り最後には奪われる存在だとか。

父親には理不尽で、娘は身勝手だとわたしは思います。

結局、理解するのには、父親になってみるしか無いのかもしれませんが。

二章では、美郎の親子関係が少し出てきて、少し感心しました。プリンセストヨトミには及びませんが、なかなか的確でした。

なぜか、口数が減ります。

bcpぎゃー

<boost/move/move.hpp>をbcpでコピーしたら大変な事になりました。中身を見ていると（見たくないけど）config, mpl, preprocessor, type_traitsが多いようです。Boost使うとこうゆうことになるんですね。
-> <http://ideone.com/37Dly>

Web

3.0の話があとで読むリストの中に入っていたので、読んだりビデオを見ました。流行語候補みたいですが、流行に疎いわたしにはよくわかりませんでした。
ビデオ -> <http://ieet.org/index.php/IEET/more/ray20110428>

gmailのタスク管理は、使いづらいでdeliciousにわざわざあとで読むタグをつくりました。deliciousへの投稿は、Taberarelooを使っているので、タグを上書きすればリストから外れるようにしてあります。何か別のサービスを使うべきかも知れません。

Boost.Moveのいい解説を読みました。boostのgitミラーを確認した限りドキュメントが無かったので、とてもありがたいです。
使い方はほとんどわかりました。BOOST_COPYABLE_AND_MOVABLEかBOOST_MOVABLE_BUT_NOT_COPYABLEでmoveの要件を満たし、BOOST_RV_REFでコンストラクタをつくります。
ただ、まだ原理をちゃんと把握するのはまだ難しそうです。
- (前編) <http://d.hatena.ne.jp/iorate/20110506/1304700716>
- (後編) <http://d.hatena.ne.jp/iorate/20110509/1304919658>

typoの報告をしたら直してもらいました。気がついたことがあったらすぐに報告できるように習慣付けたいです。

->

<https://github.com/mnmly/library/commit/d44a23d059e3326c4e7e8212692b68b94294067d>

「アンダルシア」

映画「アンダルシア」を見ました。予想していたよりもよかったです。これで、原作を買わない理由がなくなった気がします。

「天使の報酬」には少しガッカリしたのですが、今回は話がよくできていて面白かったです。伊藤英明がイタリア伊達男していましたし、黒木メイサさんは悪女してました。伏線の回収も綺麗で、すっきりしています。

ただ、タイトル割にアンダルシアに居た時間が短いのは詐欺だと思いました。それでも、他の町並みもアンダルシアに負けず劣らずよかったですので赦します。それと大臣は、他人の手柄を自分のものにするばかりでなく自分でも行動したほうがいいと思いました。

githubのwatchとfollowとdeliciousのsubscriptionを減らしました。どうでもいいのとか、多くて全部を見ることができないものを減らすとすっきりしますね。githubは更新頻度が高くてあまり関係のないプロジェクトを、deliciousはLuaとConcurrencyのタグを外しました。Luaに対する興味はそれほどなかったみたいです。

PSPのcmake toolchain

scriptにPSPのbuild.makの機能を取り込んでみたんですが、うまく動かないので少し困りました。それと、cmakeの変数名で大文字小文字を区別するようになったので、それに伴う変更も面倒でした。別のビルドシステムに浮気したいです。

Jenkinsを試しにVirtualBoxの仮想マシンに入れてみました。入れた見たときは動かなかったんですが、さっきブラウザからアクセスしてみたところ動いているようです。明日、試してみたいと思います。

Twitterの日本語ハッシュタグが使えるようになったらしいです。「魔法少女まどか☆マギカ」はハッシュタグにならないようですが、わたしは検索したいキーワードをそのまま検索したいので、そんなに関係ないです。話題のために使うそうですが、打ち込むのが面倒なので使ったことはありません。ハッシュタグはバカらしいが単純明快なアイデアだとどこかで聞いたことはありますが、未だにそれがよくわかりません。

液化石炭というものを初めて知りました。

- <http://www.newsweekjapan.jp/stories/world/2011/07/post-2186.php>

- <http://ja.wikipedia.org/wiki/%E7%9F%B3%E7%82%AD%E6%B6%B2%E5%8C%96>

石炭の埋蔵量（記憶している限り年数で）は石油よりも長かったなので、少し期待してます。

神様ドルズの二話を観ました。一話を見逃したのが惜しかったくらいよかったです。しかし！、一話はニコニコで配信されるようなので、問題ありません。

異国迷路のクロワーゼは、丁寧につくってある作品でした。ループ再生をしていたら、ガラスの割れる音がトラウマになりかけました。近藤さんがとってもイケメンだったので、不快な感じがしませんでした。登場人物全員がいい人すぎて、わたしが勝手につらくなるアニメです。

Jpcspが一ヶ月くらい前にアップデート

Jpcspが一ヶ月くらい前にアップデートされていたので、インストールしなしました。ただ、使う機会は特になさそうです。

父親のiPadから印刷するために、AirPrint Hactivatorというのを使ってみました。起動させてスイッチを入れれば簡単に使えるようです。
- <http://netputing.com/airprintactivator>

DDD2とアンダルシアを買いました。読むのが楽しみです。また、図書館から借りた元素に関する本を早く読み終えくなります。

Jenkinsにいくつかプラグインを入れてプロジェクトを設定しました。cmakeやgitなど、とりあえずビルドに必要なのが主です。最近、思うのですがブラウザから設定できる方が専用のネイティブアプリケーションよりも楽ですね。sshでログインして設定ファイルを書き換えるよりもずっと負担がありません。こういう方向ももう少し勉強するべきな気がします。

ユーロ安やドル安のニュースは少し不気味でした。どの国にもいいところはあるのですが、それは悪いところ無しには語れないのでしょうか。

中国の空っぽのマンションは、暴落した後に住んでみたいです。ああいった廃墟に憧れます。ただ、シルクロードが核の実験場だったと知ったばかりなので何があるのか少しだけ不安は残ります。

Undo/Redoが話題になっていたようです。diffを取るというのは想像通りでしたが、デザインパターンやアクティブプログラミングはよくわかりませんでした。一応、一定量(回数/容量)のdiffが溜まったらsnapshotを取るという組み合わせも想像してみたんですが、よくわかりません。

- <http://togetter.com/li/161523>

TODOリストが減らない時のストレスは、空回りを増やすのでやっぱりよくないですね。適度な量に絞るのも立派な自己管理なのでしょうね。ただ、多少きつめに設定しないと意義がないのでその辺りの加減はいつも難しいです。気がつくとき緩めすぎたり絞めすぎたり、うまくいきません。

Google

Docsの容量制限を使いきってしまって少し困っています。どうやら消しても空間が空くといったことはないようです。

人材育成のために日本の大学の資料がいっぱい公開されていました。ありがたいです。登録ができないのは少し残念ですが、RSSやPodcastが使えるので一通りみておきたいと思います。

- <http://stream.edubase.jp/>

頭痛

今日は、少し外に出たからなのか、外から帰ったあとは頭が痛かったです。

神様ドォルズの第一話を見ました。ニコニコ動画で見た空耳が忘れられなくなっています。My name is 豆腐♪は卑怯です。

続きが気になったので、本屋さんで見かけたら買ってみたいです。絵も立ち読みして悪くありませんでしたし、神様が出てくる話なので、難しい漢字があっても、その都度Wikiなどで確認するより楽でしょうし。ただ、懐の状態にも依りますので、熱が冷めない内に遭遇したいです。

前に買ったDDD

2を読み終わりました。あんな妹絶対欲しくないです。石杖所在さんには「御愁傷様」としか思えません。

少しだけ、どうしてあそこがあんなコトになっているのか伏線があったのですが、次の巻がいつ出るのかわからないのであまり期待しない方で気長に待つしかありませんね。日守秋星さんがやっと出てきました。ネタバレ覚悟で見てたWikipediaで知って以来ずっと楽しみなキャラクターだったのですが、思いの外軽い感じのお兄さんで拍子抜けしてます。それでも、反則的に強かったのはよかったです。所在さんの妹さんと闘う機会があれば是非、読みたいです。

輪るピングドラムの三話は見逃しました。代わりにBlood-

Cが録れていたの、三十分ほどずれたようです。ニコニコで見られたら見ます。

ピングドラムは少女革命ウテナをよく知らないの、どうしてあんなに期待されているのか未だによくわかりません。レンタル屋さんので探そうかと思えます。田舎のレンタルビデオ屋さんで見つかる見込みなどほとんどありませんが...

日曜日に父親がかりて来た白夜行の映画を見ました。時代設定にかなり力を注いでいるらしく堀北真希ファンの父親(50)は楽しめていたようです。

わたしは、原作を読んだことはあるんですが、原作読んでない人から見ると少しジェットコースターな気がしました。突然、人が死んでしまったり、色んな過程を飛ばしたり、最後の方に詰め込みすぎたり、など原作の厚さが影響したのでしょうか。海賊版マリオの話や知的財産権の話がなかったのは個人的に好きだったので、それを見られなくて残念でした。ただ、それは一般的につまらない話なので、まあ、いいです。自慰のしすぎはダメ、絶対です。

最後の方で、堀北真希さんの演じるキャラクターと結婚したダンスセレブ男性の妹さんが、レズ☆ハッピーになるというのは大分前に読んで忘れていた原作にもあった筋書きで、個人的にいいぞもっとやれでした。歪んだ性癖もテーマの一つみたいでしたが、その道の専門家の話や本をよく知らないの、わたしの感想はとりあえず「エッチければよし！」みたいです。

原作をもう一度、読みたくまりましたが、千ページもあるので家族の誰かが買ったら読み返そうかと思えます。東野圭吾さんの小説はメディアミックスされると別の味が出るので、刺激的です。

あんまりこうゆう感想だけ書いても何してるんだと思うので、少し情報系の話題についても書こうかと思えます。

Git関連でMac OS Xのターミナルの代替であるiTerm

2の開発がgithubに移ってました。emscriptenなど前々からよくあったんですが、こうゆうのは毎回、驚きます。Githubの成功もありますが、開発者が移行した理由は毎回興味深いです。まだ、googlecodeがGitに対応したそうです。InfoQの記事で知りました。Mercurialはよくわからないので嬉しいです。

Boost

1.47がリリースされていました。Chronoが個人的に大きいです。これで毎回OS依存の時間ライブラリを探さなくて済みますし、単位を毎回確認しなくて済みます。

Boost.Moveも収録されると勘違いしていたんですが@cpp_akiraさんの記事を見る限りまだ直しが必要みたいです。残念....

最近やっと使えるようになったBookmarkletを父親のiPadに入れて遊んでました。正直、今でもこれっぽっちのjavascriptでこんなことができるとはにわかに信じがたいのですが、便利ですね。わたしのChromeにはVozMeとInstapaperのを、父親のにはそれにxAutoPagerizeを入れました。

Instapaperは最近使い出したのですが、後で読むをスタックに貯める機能とAdBlock的な機能がいいです。読み込みの重いページがいいダイエットをしてくれますし、読み終えたものもあとで参照できます。なにより、ページが異常に軽いのも魅力です。かなり集中的にアクセスしても遅くなくなったりしません。後は、Chromeの拡張があれば万々歳です。

※
さっき調べたらInstapaperのChrome拡張がいくつかありました。「なに言ってるんだハゲ」で
す。
- <https://chrome.google.com/webstore/detail/gkopkcdfgadppcpanhoabhffmahehlbl>
- <https://chrome.google.com/webstore/detail/acgdjjilmhiofacmdnmmlndeokamkkcl>
がよさげです。後で試します。

アンダルシア 著：真保裕一 の感想

アンダルシアをちょっと前に読み終えたので更新稼ぎにその感想を書きます。話は巧くできていたんですけど、インパクトに欠けたり、伏線回収がレベル上みたいになって進みが悪かったです。終始、ワケありな女性に振り回されていたのは、納得もいきません。あれだけの秘密を隠し通すのは少し無理があります。それと、帯はちょっと詐欺です。フランス・スペイン・アンドラの警察とそこまで渡り合っていないですし、それなりに協力してるのでそんな言い方はないと思います。「アンダルシア」は一冊つかって書くよりも短編や幕間くらいの方があんまり期待しないで楽しめたのかもしれませんが。それから、天使の報酬で出てきたキャラクターがほとんど登場しなかったのは、そういうスタイルの人なのかもしれませんが、もったいなかったです。

どうやら、今回は映画が原作超えを達成した気がします。チリ毛のアンドラ警官は、映画で伊藤英明になってるみたいですね。本編で一番大きな事件が短期的なもので、最後に解決される伏線の方が大事だったところを踏襲していたのは原作を読んだ後に気がついたいいところでした。そして、戸田恵梨香をキャストしたのはわたしとくでGJです。

TwitPicの隠しAPI (って程でもない?)

色々いじってたらおもしろいことを発見した。

- <http://dev.twitpic.com/docs/thumbnails/>に書いてあるsizeには、もう二種類
- large - [twitpic.com/\(ID\)](http://twitpic.com/(ID))で表示される大きさ
- full - [twitpic.com/\(ID\)/full](http://twitpic.com/(ID)/full)で表示される大きさ

があるみたい。
これがわかれば、わざわざページを開かなくても画像のURLがわかる。

それと、chromeの拡張で使うときには、たぶん、
- <http://s3.amazonaws.com/twitpic/photos/>*
をmanifest.jsonで許可しないとイケない。

Kindleのショートカットキー

どこかにありそうだけど、自分用にメモメモ。

どこでも使えるショートカットキー

- Alt + Q~P: 数字をsymを使わずに入力。Q~Pは、1~9, 0に対応している。
- Alt + Del: テキストエリア内の全削除。

Homeで使えるショートカットキー

- アルファベットを入力 + 決定: そのアルファベット ☒ で始まる最初の題名か作者に移動。
- 検索したい単語を入力 + ☒: 入力した単語を検索。

読書中に使えるショートカットキー

- Alt + ☒: ハイライトした部分をtweetまたは共有する。
- Alt + B: ブックマークの削除。
- 矢印キーの↑/↓: カーソル移動。
- テキスト読み上げの開始位置もこれで指定できる。(後述のリンクより)
- タイプする: メモを残す。
- 決定キー: ハイライト。
- 矢印キーの→: 次の章へ進む。
- 矢印キーの←: 現在の章の最初に戻る。

PDFで使えるショートカットキー

- Shift + 矢印キー: 拡大した状態でページ内を移動。

ブラウザで使えるショートカットキー

- ☒: 再読み込み。
- Back:
- ズームした状態の場合、元の大きさに戻す。
- それ以外の場合、前のページに戻る。

オーディオ関係のショートカットキー

- Shift + Sym: テキスト読み上げ。
- Space: テキスト読み上げ中の再生/停止。
- Alt + Space: MP3を再生/停止。
- Alt + F: 現在再生しているMP3をスキップ。

以上、付属のマニュアルから。

どこでも使えるショートカットキー

- Alt + G: 画面の更新。
- Alt + Shift + G: スクリーンショットを撮る。
- Alt + Home: Kindle Storeに移動。

Homeで使えるショートカットキー

- Alt + Shift + M: ゲームを起動。
- G: マインスイーパーから五目並べに変更。
- M: 五目並べからマインスイーパーに変更。
- Alt + Shift + ':(ピリオド): シリアル番号とバーコードの表示。

ブラウザで使えるショートカットキー

- Alt + J: 右にスクロール。
- Alt + H: 左にスクロール。

画像Viewerもあるらしい。

※ モノクロなのでなにが楽しいやら…。(漫画ならいいけど)

- F: 全画面で表示する。
- Q: 拡大する。
- W: 縮小する。
- E: 元の大きさに戻す。
- C: 実際の大きさで表示する

- R: 画像の回転する。
- 矢印キー: 上下左右にスクロール。

メニュー

- メニューを表示すると時刻が表示される。
- Homeでメニューを開くとKindleの空き容量が表示される。

電源ボタン

- 押すだけ: スタンバイ状態に。
 - 7秒以上押す: 電源を切る。
 - 15秒以上推す: リセット。
- ※ 再起動は、「設定」のメニューからできる。

スクリーンセーバー

- ただの液晶じゃないので、ブラウン管みたいに必要らしい。
- 変える方法があるらしいけど、リスクがあったり。冫(´▽`) 冫ヤレヤレ

Jail Brake

- できるらしい。
- kiterm(<http://info.iet.unipi.it/~luigi/kindle/>)とかが入れられるらしい。
- もちろん、自己責任でどーぞって感じ。

以上は、以下のリンクから。

- <http://www.geoplanit.co.uk/?p=78>
- <http://www.geoplanit.co.uk/?p=80>
- <http://www.geek.com/articles/gadgets/kindle-3-hidden-features-include-2-games-image-viewer-screenshot-grab-20100930/>
- <http://blog.the-ebook-reader.com/2010/08/29/kindle-3-shortcuts-hot-keys-and-hidden-features/>
- <http://arstechnica.com/gadgets/guides/2009/10/weird-kindle-tricks-screensavers-screenshots-and-games.ars>

拡張が閉じられる時に、関数を呼ぶ。

```
- window.onunload = function() {};
```

をbackgroundページで、すればいいみたいです。
これは、ページを閉じるときにも応用できると思います。

TwitCastingの埋め込みとか。

ライブを埋め込むなら

- <http://twitcasting.tv/indexapivideo.php>

を使います。videoタグ必須だけど、Chromeなら問題ないです。

そのとき、posterをどのように取得したらいいのかわかりづらいわけですが、

- <http://twitcasting.tv/indexapithumbnail.php>

で取得できる画像をそのまま使えばなんとかなります。

最終的にタグは、

```
- '<video src="http://twitcasting.tv/:id/metastream.m3u8/?video=1" autoplay="true" controls="true" poster="http://twitcasting.tv/:id/thumbstream/liveshot"></video>'
```

-- :id - ユーザID

になります。width/heightを省略しても自動的にサイズが確定されるみたいなので、今回は指定していません。その他、オプションも必要に応じて指定してください。

録画の埋め込み方はよくわからないのですが、録画のposter用画像のurlは、

- <http://twitcasting.tv/:user-id/thumb/:record-id>

-- :user-id - ユーザID

-- :record-id - 録画ID。例えば、"<http://twitcasting.tv/:user-id/movie/5478435>"の場合、最後の数字の部分。

となってるみたいです。

Ustreamの埋め込みタグ。

http://developer.ustream.tv/data_api/docs

- UstreamのAPIは基本的にここに書いてある（みたい）。
- 誤字あるよ…。
- 書き忘れたらしきAPIあるよ。

Kindleのショートカットとかハックとか

- <http://blog.diannegorman.net/2010/09/kindle-3-keyboard-shortcuts-et-al/>
に書いてある内容もけっこうよかった。

サーチバーから、色々できるのは意外。

- @dict <keyword>
- @help - 使えるすべてのコマンドを列挙する。
- @print [no idea what this does]
- @store <keyword>
- @url <url> - URL無しの場合ブラウザを起動。
- @web <google search term >
- @wiki <keyword> - wikipediaと同じ。
- @wikipedia <keyword>

JSONのないブラウザ。

にわかには信じられないことですが、世の中にはJSONが使えないブラウザがあります。

わたしが遭遇したのは、Kindle 3.2.1のブラウザですが、(

<https://twitter.com/#!/beepcap/status/119702003936002048>

)の通り、w3mにも無いようです。Kindleでは、Term.ioを使おうとした時にJSON.stringifyの戻り値がおかしいことに気がついたのですが、jquery-json(<http://code.google.com/p/jquery-json/>)の\$.toJSONをJSON.stringifyの代わりに使ったら直りました。とりあえず、これでJSONが送れないのは解決です。

それでも、Kindleでいくつか問題があって、まだTerm.ioはうまく使えません。把握してるのは以下です。

- socketが時間を経ると閉じる。
- 標準出力が表示されない。
- consoleが全く使えない。
- (Kindleに限らず) 標準出力の内容がすぐに消える。

consoleの件は、一応の解決方法を見出しています。一番、簡単なのはalert()を使うことです。ただ、printfデバッグみたいで非常に原始的で使いづらくなります。もうひとつは、socketでつながっているのでデバッグ用のメソッドでメッセージをサーバに送る事です。ただし、この方法は何かサーバ側のコンソールに出力される標準出力と混じるので、現状あまりいい方法ではありません。標準出力を止める方法を探する必要があります。

ChromeのOmniBoxは可能性がありそう。

twi2url(<https://chrome.google.com/webstore/detail/dedkhngknipfmnikdajbppndcnoibpad>)にomniboxからのTweetを実装したら思いの外便利でした。OmniBoxは元々、検索用の機能みたいですが、URLバーからなんでもしたい人ならそれ以外の機能があってもいいのではないかと思います。

ただ、いくつか不満な点も見つけました。

- キーワードの補完が他の検索みたいに効かず、キーワードの全部を入れないといけないこと。

- OmniBoxの説明が” {拡張名} 検索” になってカスタマイズできないこと。

- 拡張一つにつき一つしかOmniBoxが与えられないこと。

検索用に設計されているせいか検索にしか使えないのは少し不便です。

それから、twi2urlにPINを使わないコールバック方式のOAuthを実装しました。前々から、PINはめんどくさいと思っていたので自己満足♪満足♪です。ただ、popupの"SignIn"を押すとWindowがいっぱい開かれるの症状の原因がよくわからないので、新しいバグをつくってしまいました。

参考にしたページは最初は認証用のメソッドを使わず、保護された情報を扱う時に使うメソッドを読んでいたのは少し難しかったです。(

<http://www.mobiledevelopersolutions.com/home/start/twominutetutorials/tmt5p1>)

Deliciousがリニューアルしてた。

5月頃から、YouTubeの創始者に拾われたニュースがあったみたいですけど、最近、Yahoo!から独立したみたいで、しばらく使えていませんでした。それから、Yahoo!じゃなくなったせいか、Yahoo

IDではログイン出来ず再度パスワードを作りなおしました。これ以上、Yahoo!に依存できないのは当然ですよ。ウァア-----。(´Д`°)。-----!!!!

それに加えてTaberarelooのブックマークタグ補完がその頃から不調なので、エラーが起きても放ったらかしにしていました。Issue(<https://github.com/to/tombloo/pull/31>)が立って、修正されていたので確認しなおしたら直っていました。素早い対応ありがとうございます。

そして、Google

Readerで気がついたんですが、Deliciousのsubscriptionsが使いなくなっていました。複数のタグを統合して、RSSを出力してくれる機能なのですが、残念です。まだ、移行作業中(<http://take-cheeze.tumblr.com/post/10846637194/what-happened-to-network-fans-and>)みたいなので気長に待つとします。

電腦コイル無料配信を見た。

- <http://www.tokuma.co.jp/coil/haishin.html>

でやってるのを聞きつけたので今日、見てました。このアニメを見ると、遊ぶ友達がそれほど電脳メガネ欲しくなりますね。

このアニメは、何話か暇つぶしに見てたんですが見返してみると大分、記憶していたものと違いました。すごい子どもっぽいアニメなはずなのに楽しめるのは、つくったひとの実力なのでしょうね。デジモンみたいにとっても楽しめました。

物語を見ていて思ったのですが、あの眼鏡は眼が悪い人でも使えるんでしょうかね？

眼鏡は本来そのためにあるのでとても不思議です。

それから、あと何年したらあの世界に追いつけるのか楽しみにもなりました。きっとわたしはもう大人になっているのかもしれませんが、自分がそこまでの道のりに少しでも貢献できたらと思っています。そして、その時になっても面白いアニメを見続けられればと願います。

Node.jsのドキュメントとか

英語

- <http://nodejs.org/docs/latest/api/all.html>

日本語

- http://nodejs.jp/nodejs.org_ja/api/all.html

英語の方が若干古いのは謎です。日本語の方は頻繁に更新されているらしく常に最新になっています。翻訳されている方に頭が下がりますね。

ターミナルで画面を綺麗にする方法。

まず、clearコマンド。でも、iTerm
2（これでしか実験してない）では画面が移動するだけで、バッファは綺麗になってない。なので、上の方にスクロールすると、まだ主力とかが残ってる。orz
iTerm 2（というか、OS
Xのターミナル系のアプリケーションでは）⌘+Kでバッファが綺麗サッパリする。iTerm
2では、右クリックでclear
bufferとして出てくる。ただ、この方法はターミナル限定なのでもっと一般的な方法は知りたい
。

Pomera

調べたら面白かったのでメモ。

辞書データの解析結果。

- <http://homepage3.nifty.com/TeaPot/pomera.htm>
- 辞書の単語数は、最大 2^{16} で、ちょっと不満。
- あと、変換ツールはCで書かれてて、苦手。

辞書ユーティリティ

- <http://sei.gee.jp/docs/program/hta/pomutil/index.html>
- Windows専用。
- 全然ブラウザアプリな感じじゃない…。(´・ω・`)

Atokの辞書形式

- http://www.hyperteika.com/ime/atok/atok_syosiki.html
- mozc(Google IME)のも似た感じだった。

ついでに、Google IMEに入れた辞書

- 顔文字 <http://matsucon.net/material/dic/>
- 和英・英和 <http://code.google.com/p/google-ime-user-dictionary-ja-en/>
- 強化辞書 <http://www.mediafire.com/?cy1mxpjd5l5h>

エスケープシーケンス

Term.ioやJS-Termをforkする過程で必要になったのをまとめとく。

一番充実したxterm公式のドキュメント

- <http://invisible-island.net/xterm/ctlseqs/ctlseqs.html>

別のjsによるterminalの実装。modifierを付加する時に参考になった。

- <http://www.java2s.com/Open-Source/Javascript/Ajax/shellinabox/demo/vt100.js.htm>

そこそこ詳しい。

- <http://bjh21.me.uk/all-escapes/all-escapes.txt>

- <http://rtfm.etla.org/xterm/ctlseq.html>

- <http://www.termsys.demon.co.uk/vtansi.htm>

- <http://www.ivarch.com/programs/quickref/termvt102.shtml>

ASCII関連。

- <http://ascii-table.com/ansi-escape-sequences.php>

- <http://ascii-table.com/ansi-escape-sequences-vt-100.php>

Wikipediaのページ。deleteキーのはけっこう参考になった。

- http://en.wikipedia.org/wiki/Delete_key

- http://en.wikipedia.org/wiki/ANSI_escape_code

VT100のドキュメント。VT100はde facto standardなので参考になる。

- <http://vt100.net/docs/vt100-ug/chapter3.html>

日本語

- <http://www.mm2d.net/c/c-06.shtml>

- <http://archive.linux.or.jp/JF/JFdocs/Bash-Prompt-HOWTO-5.html>

Chrome Sniffer

閲覧中のページのウェブアプリ名やJavaScriptライブラリを判定するChrome拡張。

- <https://chrome.google.com/webstore/detail/homgcnaoacgigpkljjekpignblkeae>
- <http://www.nqbao.com/chrome-sniffer>
- <https://github.com/nqbao/chromesniffer>

時々便利。wordpressの判定が特に役に立つ。

AirPrint Activator 2.0

<http://netputing.com/airprintactivator/airprint-activator-v2-0/>

- iPhoneからプリントしたくなったので、入れました。
- プリンタの共有をシステム環境設定でしないといけないのは地味にはまります。

Epsonの新しいプリンタは、AirPrintに対応してるそうです。家で使ってるプリンタ（EP-901A）にはファームウェアアップデートがなかったので、対応する予定は無さそうな気がします。

。それと、ちょっと前に気がついたんですけど、プリンタのIPアドレスをブラウザに打ち込むとHT TPでプリンタの操作ができるんですね。Webは偉大なのです。

母が年賀状の前に一度修理に出すと言っていたので、このプリンタは当分使うことになりと思われれます。次に、買い換える時は携帯からプリントできる機種にしたいです。それか、iOSとOS Xの統合を待つのもいいかもしれません。OS Xなら設定なしでプリントできますから。

コンピュータ開発史

コンピュータ開発史がけっこうよかったので長々とレビューを書いてしまいました。

<http://booklog.jp/users/takecheeze/archives/4320121384>

やっぱり、歴史をちゃんと学ぶのは大事です。そして一般常識のいい加減さに驚きました。歴史を知らないと人生損しますね。

フォン・ノイマンは、名がしれていた学者なだけで、でもそんなに関係ないなんて詐欺です。

Texttを使ってムラムラしたことなど

※ とりあえず下書き

View

- マークダウン(<http://daringfireball.net/projects/markdown/>)またはそれに類するものが使えたらいい
- URLのリンク化がほしい
- 項目の並べかえをしたい

Bookmarklet

- ただのBookmark機能だけみたい
- 選択した領域の引用がしたい
- <http://textt.net/b/clip> の仕様(パラメータとか)を公開して欲しい(できれば他のAPIも)
- iPhoneで <http://textt.net/b/clip> を縦状態で開くとテキストエリアが異常に大きい(こんな感じ <http://pic.twitter.com/nqyldfSZ>)
- <http://textt.net/b/clip> での投稿が終わったあとウィンドウを自動的に閉じるオプションがほしい

Kindle

- Kindleへの転送機能がほしい
- Kindleのメールアドレスを登録できるように

Dropbox

- iOSアプリのPlainTextみたいにDropboxと同期したい

RSS

- 修正内容のdiffがほしい

avahi

オープンソース版のBonjour。

パッケージ名は、avahi-utils。

IPアドレスを直接入力しなくてもホストにアクセスできるのでいいですね！

makeinfo

Ubuntuでmakeinfoが入ってるパッケージは、texinfoらしい。
magitを入れるときに困った。

svnコマンド

git-svnを入れてもsvnコマンドは入らないらしいので、
sudo apt-get install subversion
した。紛らわしい。全くambiguous。

window.open()

普通のページだとwindow.open()は新しいウィンドウをつくる関数らしいです。知りませんでした。

今まで、Chrome拡張で使ったことはあるのですが、そっちだと新しいタブを開いてくれたので、何も考えずに使ったら偉いことになりました。なので、わざわざてきとーな拡張のオプションを開いて、そこのコンソールからwindow.openして解決しました。ぬーん。めんどくさい仕様です。

zsh

Ubuntuでzshを使おうとしたら入ってなかったので

```
sudo apt-get install zsh
```

しました。またzsh-betaはchshで使えませんでした。デフォルトでzshが入ってるOS

Xマジzshです。

とりあえず、これで開発環境が整いました。これからポチポチissueをcloseしていければと思っています。

PanicのPromptで鍵を消す方法

Promptを使っていて鍵をいっぱい登録して消そうと思ってもできなかったので困りました。ただ使っている鍵が丁度なかったのも再インストールしたら全部消えてさっぱりしました。こういう時はすっきりしないのでApp Storeから問題を報告しました。役に立てばいいです。それから鍵を登録するのがめんどくさいのもうパスワードを保存する設定にしました。セキュリティ的にアレですが、疲れたので次善の策です。

Assignedの意味

githubとかのIssue

Trackerでよくassignって言葉を見るんですけど、「仕事の割り当て」って意味みたいです。
一人でやるときは必要なさそうな機能ですね。

どこにも載ってなかったの

iOSでBluetoothキーボードのショートカットキーの一覧がいっぱいあったんですけど、その中に載ってなかったのがひとつあったので、メモです。

CD/DVDの取り出しキー（三角の下に棒が一本入ってる奴）：iOSのキーボードを開く。

たまたま見つけたんですけど、地味に便利です。

PromptにBluetoothキーボードからの入力を受け付けないのがたまにあるので、手で入力する時に使えます。Shift +

カーソルキーが使えないのは地味につらいです。キーボード使う意味が半分くらいなくなります。

それと、Bluetoothキーボード用にエネループを買いました。単4のも充電器付きで欲しかったんですが、高かったのもち金では足りず、単3の充電器付きだけしかかえませんでした。

Jenkinsを使ってみた。

Jenkinsサーバを立てたのですが、自分だけで使っても悲しいのでネットに公開してみることにしました。

まず、でDNSを設定します。

そして、ルータのポート変換で、ポート8080をサーバに転送しました。

サーバは、ubuntu-

serverをVirtualBoxで稼働させています。ホストと同じハードディスクだと寿命を縮めるような話を聞いたので、ディスクイメージは外付けハードディスクに置きました。メモリもちょうど増設したので、1GBと個人的に太っ腹なことをしています。2GBに増やしてもいいかもしれませんが。

DNSは貧乏性なもので、

- <http://www.ieserver.net/>

をつかって、登録してみました。フリーは素晴らしいです。

使っている無線LANルータはWZR-HP-

AG300Hというのですが、少し不自由というか痒いところに手が届かない人でした。一応、つながったので許しますけど。ポート変換で、使ってるグローバルIPのポート8080をJenkinsが動いてるサーバのポート8080に転送して、そのサーバのIPをDMZに登録しました。なぜ、8080なのかというと、ポート80はルータの設定画面に必ず飛ぶからです。ついでに、sshサーバのパスワード認証を切って、sshポートも空けときました。

これらの設定を内部ネットワークから確認しようとしたんですが、内部ネットワークでは設定が働いていないらしく仕方なくiPhoneの3G回線から確認させられました。どこからでもできると想定していたので、少しガッカリしています。

ちなみに立てたサーバのURLは、"<http://takecheeze.orz.hm:8080/>"です。

セキュリティとかはよくわかりませんが、困ったら対策しようと思います。

Jenkinsをアップデートしようとした件で...

設定なしで、`apt-get install jenkins`したあとに、"<https://wiki.jenkins-ci.org/display/JENKINS/Installing+Jenkins+on+Ubuntu>

"を見つけて、この設定を適応したらJenkinsが起動しませんでした。なので、仕方なく`apt-key`で鍵を消して、`package`を消して元あったjenkinsを入れたら元に戻りました。(;´Д`)

「Jenkinsの管理」ページを見ると新しいのがあるらしいんですが、どうやら入れられない感じみたいです。

ショートカットキー

よくCtrl + 左右カーソルは、単語単位のスキップができるんですけど、OS Xの場合はいきすぎてしまいます。なので、ずっと不満だったんですが、iOSのキーボードショートカットをみたら、Alt + 左右カーソルがあったので、試しにOS Xで使ってみたら、こっちが単語単位のスキップでした。

OS Xは全社長の影響か、他とは少し違うのでたまに困ります。MacPortsみたいなプロジェクトがあるのは救いですけどね。

Jenkinsでgoogle-test

やっとわかったのでメモしときます。

基本はこのページです。

- <http://asr.s59.coreserver.jp/memo/dev/1017>

ただ少し大雑把なのでいくつか補足しときます。

- <https://wiki.jenkins-ci.org/display/JENKINS/xUnit+Plugin>

の最後の方にコメントされているように、google-testの出力形式(*)はxUnitと互換性があります。なので特別なプラグインを使わずともJenkinsはgoogle-testによるテスト結果を理解することが可能です。

まず、google-testでテスト結果をファイルに出力しないことには始まりません。

- http://d.hatena.ne.jp/E_Mattsan/20091012

は、それぞれのテスト実行ファイルが持つオプションでその中で今回は、"--gtest_output"が重要です。テスト結果をファイルに書き出すには使っているビルドシステムの設定を変えてこれらのオプションをテスト実行時に追加する必要があります。例えば、テストtest_0の結果をtest_0.xmlに出力したいのなら、

```
test_0 --gtest_output=xml:test_0.xml
```

のように実行するよう設定してください。

それができたら、ビルドファイルの変更をpushなりcommitしてそのプロジェクトのJenkinsジョブの設定で「ビルド後の処理」にある「JUnitテスト結果の集計」にチェックを入れて、テスト結果が出力されているファイルを設定します。ここではワイルドカードが使えるので、"result/test_*.xml"のようにすると便利です。また、複数のファイル指定も可能で、そのときはそれぞれをコンマ(,)でわけることができます。

っと、まあ、こんな感じです。こうゆうのを体験するとテスト書きたくない人の気持ちがよくわかります。

Instapaperをなめてました。

Instapaperをブラウザから使っていたんですが、ついこのあいだ有料版を買って、今まで「っぺ」と蔑んでいた有料版が、ブラウザ版を「っぺ」と蔑む理由になりました。使い勝手が全く違います。

まず、オフライン機能がついているのは大きいです。いくらiPhone 4Sが3G回線につながっていると言ってもネットワークを使うとどうしても遅くなることがあります。

そして、カユイところに手の届くネイティブアプリケーションならではの（？ iOSやHTML5の詳しい仕様は知らないです）機能もいっぱいありました。ブックマークレットの登録方法もいくぶんか親切です。

ただ、ブラウザ版でも満足してたのは事実で、Instagr.amのように特定の環境からしかアクセス出来ないなんてことはないのは、今でも満足しています。Web版は、こういったアプリケーションに用意する基本的な備えとよく聞くのですが、こういうのをちゃんと守っているのはGJです。

Instapaper（^ω^）ペロペロ。

PPで死にかけた。

昨日は、ビルドできないプロジェクトを脱却するべく、glsl_typeっという車輪の再発明もはなはだしいプロジェクトのビルドを通してました。glsl_typeとは、...簡単に言ってしまうとGLSLで使えるデータ型をC++でも使えるようにするヘッダオンリーなライブラリです。ソースコードは、githubにおいてあります。ちなみに、速度とかSIMDとかそういうのは、気にしたいのですが、どうにもわからないので放ってある自己満足なライブラリになっております。

それで、なぜビルドできていなかったかというと、Boost.PPの使いすぎでコンパイラのチェックすら通過していなかったのです。原因はいくつかあるのですが、まず、クラス定義をPPで生成しているおかげで、エラーの箇所が全くわからないので、エラーメッセージの文脈から大体の場所を推測するしかありません。そして、使っているコンパイラにも問題がありました。

clangはメタメタしたコードやPPピーしたコードのコンパイルエラーを直す時にあまり役に立ちません。親切が仇になっているというのでしょうか？

デフォルトで設定では、表示されるエラーの数は少ないですし、エラーひとつに何行ものご親切な解説がづらいです。それに比べてgccは、すっきり・最小限・デフォルトでエラー数の上限なしの大変不親切仕様ですが、PPピーやメタメタしたコードの時は、大量のエラーを俯瞰することが大事なので大変役に立ちます。（それから、システムの都合でC++11のライブラリが動かしやすいのはとてもいいですね！）

っと、まあそんな感じで、ビルドシステムを通さずにgccでコンパイルをした時の方が幸せでした。clangのエラーメッセージが天使に見えていた時期がわたしにもあったわけです。今はどちらかということお節介かもしれません。コンパイラにも渡したフラグについてもけっこうウルサイですし。それと、PPピーした時は列が4桁になるなんてこともあるので、右端で折り返す設定になっているときは、どこがエラー箇所なのかホントにわかりません。

最後に、エラーを取り除き終えた時には全く無意味なことをしていたと時間を浪費したと後悔しました。C++こわいです。

orz

やっとわかったこととか。

CMakeの`{EXECUTABLE_OUTPUT_PATH}`や`{LIBRARY_OUTPUT_PATH}`には、`{CMAKE_SOURCE_DIR}`系を使っちゃいけないことにやっと気が付きました。代わりに、`BINARY`がついているの変数を使ったほうが色々楽です。

理由はいくつかあるんですが、ビルドしたファイルを消しやすい、のが一番かもしれません。ソースコードは消したくないけど、ビルドしたファイル（ライブラリ、実行ファイル、etc.）は邪魔なので消したい時があるんですけど、Gitに頼りつつ消すのは正直めんどくさくて、ビルドディレクトリごと消せた方がはやいです。LLVMは、そういう事情でソースコードとビルドのディレクトリを分けないといけないんだと思います。ああゆう大きなプロジェクトではそういうことは手に負えないのでしょうか。（;´Д`）

それと、`CURRENT`が付いている変数もあまり使わない方がいいですね。ソースコードの参照の時は、いいんですが`OUTPUT_PATH`で使うとビルドしたファイルを消す時に、分散されて消すのに手間がかかるようになります。なので、`{CMAKE_BINARY_DIR}/bin`とか`{CMAKE_BINARY_DIR}/lib`に出力した方ががっと消せますし、Jenkinsで成果物を保存するとき、指定するファイルをワイルドカードで選択するのがずっと容易です。テストもその方が結果ファイルを集めるのが楽ですし、どこにあるのか迷わずに済みます。

っと、まあ、またビルドシステムをいじっていて特に進まなかった報告でした。

iOSをアップデートしました。

バッテリーの持ちが良くなると聞いて早速やってみました。
とりあえず、iTunesにつないで、アップデートの確認を押して、1GB未満のアップデートファイルをダウンロードしてインストールできました。本体の更新よりも、ファイルのダウンロードのほうが時間がかかった感じです。

感想は、・・・特に無いです。あんまり変わった気がしません。たぶん、バグフィックスとかソフトウェアのマイナーな改良しかやってないんでしょうね。0.0.1変わったくらいで、ガラッと変わったら逆に困ります。

たまたま、KindleにAppleのハードでAndroidを動かすプロジェクトのスライドが入っていたので読んだのですが、脱獄必須だったので動かす方向にはそこまで興味が沸かなかったです。ただ、ハードウェアそのもののリバースエンジニアリングは面白そうでした。

それと、そのついでに入っていたIceRobotのスライドもそこそこ真面目に読んだのですが、Appleで言うともしもOS

XとiOSを統合したらという仮定でのお話みたいで、将来性をとても感じました。折しも、今日Apache

Harmonyが終了する話を聞いたので、さらに将来性を感じます。ただ、GNUのライセンスの制約はあんまり好きじゃないので、その点では微妙かもです。

そういえば、ワールドカップ予選のタジキスタン戦を観ました。前半だけ観たのですが、事前に聞いてたグラウンドの酷さは、選手のプレイがよかったのか、よくわかりませんでした。タジキスタン側から何度か放たれたシュートはこわかったんですけど、結局守りきったそうです。

相手はカウンターしか攻撃手段がなさそうな感じだったので、一点くらい返せたのかな？と期待していたのですが、そういうことはなかったようです。日本代表強いですね。（イマサラ

HEROESとかハリー・ポッターとか

家にHEROESのDVD

Boxがあったので、最後のシーズンを見て最初のシーズンから見ました（漢文みたいな順序なので、ちゃんと書くと4→1→2→3って順番です）。勉強があるのに、そんなことしててもいいのかと聞かれると、「BGMだから大丈夫だもん（汗）」です。

「贖罪」は、元ダース・モールさんがかっこよかったです。それ以外は、特に印象に残ってない気がします。ヒロが突然治ったのは、(・ω・)I?な感じだったり、ラスボスが小物であり盛り上がり上がらなかったり、まあ、いろいろある気がしますけど、それはなかったことに…。なんというか、蛇足でした。いっそ、スピンオフにと思います。

シーズン1は、思ったよりも長かったです。テッド（主に放射線を出せる人）は能力だけ長寿でしたね。それから、伏線のはりかたがうまいなととても感心しました。

シーズン2は、マトリックス・リローデッド的な位置付けな気がします。剣聖武蔵編は、面白かったです。マシ・オカは和服を着てたのでお腹が気になりませんでした。わたしも見習って肥ったら和服にしようかと思っています。今回の危機については、バイオテロってなんか地味な印象なのであんまり危機感をお裾分けいただけなかったです。（両親が医療関係者だっという影響もあるかもしれません）

シーズン3は、前半がよかったです。ペトレリ家の壮大な家族喧嘩なだけな気がしますけど、父親のラスボス風格はとても(・∀・)1!!

ペトレリ父は最後にあっさりやられちゃうのが残念ですけど、最高の敵でした。それから、アダムのギャグにしかならなかったところも、別の方向で最高の敵でしたね。前回のラスボスがあんな雑魚に見えるなんて、いい演出です。ヒロとのやりとりもボスとは思えないほどギャグでした。

シーズン3の後半は、消化試合みたいで、観てて物足りなかったです。ペトレリ父に全てを持っていかれたといいますか、消化できる伏線がもうほとんどない失速感といいますか。ただ「反逆者」の意外な正体は、意表を突かれました。

最後に、サイラーはメイン・ヒロインではないかと思うくらい、特殊なキャラでした。ダーク・ヒーローさは、この作品で一番だと思います。最後まで殺されなかったスタッフは、偉大です。何度、死んでしまえと思ったことか。でも、シーズン3・4でのデレはとっても可愛かったです。「贖罪」でサイラーを更生させたのは、とても評価できます。

っと、まあ、そんな感じですか。こうしてちゃんと見てみるとスーパードラマTVで見てた時とは違えますね。特典映像はそんなに見てないので、また、時間を見つけたら見たいと思います。（もちろん、BGMとして！）

続いて、ハリー・ポッターです。昨日、Blu-ray Boxが来たのでYear 7だけ観ました。

内容ですが、原作は既読でしたが半分以上忘れてました（一度読んだことを忘れないような才能が望まれる限りです）。ただ、マッドアイの魔眼がどこにいったのかとか余計なことは少し覚えてます。

ヴォルデモート卿を見て思ったんですが、悪役にはもっと崇高な野望とか、嫌なヤツだけど人を惹きつける魅力とか、決してDQNだけじゃないところが欲しかったです。でないと、この人、魂を裂かれるような（実際に裂かれていますけど）思いをしてまで何をしたかったのか巨大な疑問が渦巻きます。

それくらいです。7年分、全部見返すのはちょっと怠いので、やめときます。

unique_ptr

Squirrelで最近、遊んでるんですけど、HSQUIRRELVNをunique_ptrで管理できないかと思いたち、実際にできたので、メモしときます。コードは↓な感じですが。想像以上に長くなりました、はい。STACK_SIZEは1024とか4096とか適当な定数です。深い意味はありません。

```
"std::unique_ptr<std::remove_pointer<HSQUIRRELVN>::type,  
std::function<void(HSQUIRRELVN)>> vm(sq_open(STACK_SIZE), sq_close);"
```

まず、unique_ptrでの肝はstd::functionです。最初は、関数ポインタ (sq_closeの) をそのまま渡したんですけど、コンパイルできなくて、ググったところどうも関数ポインタは受け付けられないので。なので、std::functionに出てきてもらって、見事解決！です。std::functionはよく見かけていたので、特に使い方に迷いませんでした。

次に、「typedefは時々お節介ですね」というお話です。別に、ヘッダをgrepして定義を見てもいいんですが（最初は実際に観ました。struct

SQVMみたいな名前だった気がするんですが、struct付けるの怠いです。C++なので別にいいんですけど）、それはそれで仕様が変わった時に困るので、type_traitsを使ってポインタのtypedefから型だけゲットしました。 "::type" を忘れたなんてしょーほてきなミスはしなかったわけでもしたわけでもしたかもしれないかもしれませんが、それはさておき、やっとtype_traitsさんの使い方がわかりました。ありがたやー。

で、まあ、これで終わればよかったんですが、HSQUIRRELVN以外に、HSQOBJECTというオブジェ（クト）があります。そして、どうせならこれも、C++（標準のスマートポインタ）で管理したいと思って少しだけ妄想してました。結論は、HSQOBJECTはポインタじゃないのでデータに載せるとかそういう方法を使わないと無理という判断で、やめにしています。時前で構造体を書くのはめんどくさいです。ヘッダにしたとしてもutilityなんて敗北宣言にも等しいところに置かれるのが関の山な気がします。

Jenkinsの再起動

「Jenkinsの管理」で「シャットダウンの準備」はあるんですが、再起動する方法がないので困ったのですが、プラグインのインストールなどをしていた時にとあるアイデアが閃いて使えたのでメモです。

ここでは、例として今日、開通させてもらったeasy-rpg.orgのJenkinsを例として使います。

"<http://easy-rpg.org:8080/>"にあるJenkinsを再起動させるには、`/updateCenter`すなわち

"<http://easy-rpg.org:8080/updateCenter/>"

にアクセスして「インストール完了後、ジョブがなければJenkinsを再起動する」をチェックすると再起動します。本来はプラグインのアップデートとかインストールに使うものみたいなんですが、そういう状態でなくとも素でアクセスできるようです。

近い未来で、脆弱性になりそうな気がしますけど、なかなか使えるなと思いました。

Screen使いになった話。

termをちゃんと設定すれば、emacsもちゃんと使えることがわかったので、ちょこちょこ試してから、

```
chsh -s /usr/bin/screen
```

でログインシェルをscreenにしました。サーバの接続を切っても、簡単に再開できると評判だったので、ちょっと遠いリモートからのsshの時に使う時にはこっちの方がいいかなと思ったので思い切ってやった感じです。メインマシンのは流石にちょっとこわいんですが、仮想マシンならこ

うゆうの気軽にできていいですね！
こ

うゆうのを毎回使うたびに思うんですが、\$TERMの設定で毎回必ずコケてる気がします。もっと自動的にやってほしいです。
tmuxというフォークがあるみたいですけど、screenとの互換性がもっと高ければ使いたいなと思いました。横分割も羨ましいんですが、screenの方がまた分がある気がします。

因みに現在の ~/.screenrc:

```
term xterm-256color
```

```
shell /bin/zsh
```

```
startup_message off
```

```
altscreen on
```

```
autodetach on
```

```
bindkey "^T" focus
```

```
<< EOF
```

EasyRPGでJenkins

ちょっと書きましたけど、Jenkinsをeasy-rpg.orgにインストールしてもらいました。URLは <http://easy-rpg.org:8080/> です。設定とかは、言い出しっぺのわたしがしたんですけど、パッケージのインストール（超絶めんどくさい）はサーバの管理人さんにしてもらいました。で、なんとかビルドできています。SCMのポーリングは、一時間おきにしてみました。一応、自分でやった経験が役に立っていますよ。

それと、easy-rpg.orgのサーバ構成みたいなのをちょっと聞いたんですけど、VPS羨ましいです。まずは、それです。\$4.5/monthらしいですけど、さくらVPSとかKDDIがやってるのよりも、リーズナブルな気がします。一概に比較はできないんですけど、CPUはけっこういいやつらしいので、速かったです。4コアなんてわたしのマシンではちょっと無理だと思います。

それから、自分の立てたサーバだと使うのに気がひけていたJenkinsのIRCボットも使ってみました。最初はあまりうまく動いていなかったんですが、設定をガチャガチャやったらなんかなりました。freenodeの#easyrpgにニックネームeasyrpg-jenkinsで常駐してます。"!jenkins help"で、使い方をみてみたんですが、想像していたよりも色々なことができるみたいです。まだ、よくわからないので、IRCボットにユーザ権限は与えていないんですが、また機会があればつかってみたいと思います。

さて、これで自分のところのJenkinsがあんまり要らなくなった感じです。プラグインHellしてるのでかなり消したいと思っています。他にもビルドしてるプロジェクトがあるので、しないと思いますけどね。

びふおー・あふたー

luabindのdebug.hppのアイデアはけっこう好きなんですけど、デバグが無いと使いづらかったの、なんか魔改造してみました。

びふおー：<https://github.com/luabind/luabind/blob/master/luabind/detail/debug.hpp>

あふたー：<https://github.com/take-cheeze/langbind/blob/master/nutbind/detail/debug.hpp>

Squirrel用のコードになってるのは割愛して、まず、ただのassertだった場所についてです。

元のコードは、assertがdebug.hppにそのまま書かれているので、debug.hppにあるassertの位置をアサーションが失敗した時に表示してきます。これでは色々とだめです。事件は会議室で起きてません。なので、stack_checker_typeは肥大化しますが、マクロで取得した行番号、ファイル名、関数名をstack_checker_typeに記録しています。因みに、これは特別なメモリアロケータを使う時によく使われているテクニックを応用です。こうゆう時に役立つんですね。

次は、エラーメッセージについてですけど、折角比較しているのでやっぱりテストみたいに期待した値と実際の値を表示したいです。また、スタックに積まれているオブジェクトの型も比較しています。原理はstd::vectorにstackの型(enum)を積んで、デストラクタが呼ばれてる時に比較してるだけです。書いてて思ったんですけど、こうゆうメッセージの整形ってけっこう大変ですね。(参考画像1：

http://twitter.com/#!/take_cheeze/status/139672460713930752/photo/1 , 参考画像2：

http://twitter.com/#!/take_cheeze/status/139672323841212416/photo/1)

今回は、あまり経験がなかったデバグ用のライブラリを書けて面白かったです。assert()だらけのソースコードはやっぱりよくないですね。

「饅頭怖くないもん」てきな

ニコニコで、「とらドラ！」の一挙放送があってタイムシフト予約したのを観ました。一応、観たことあるんですけど、改めて見ると前に観た時よりもよかったです。「生徒会長vs手乗りタイガー」は迫力があつたと思っていたんですけど、ニコニコのコメントで騒然となっているのを見るとやっぱり面白いですね。

そして今日、蒼穹のファフナーが5話くらい配信されていたんですけど、こっちもよかったです。ちなみに、こっちはリアルタイムで見えていません。「鬱系こわい、鬱系こわい、鬱系こわい、鬱系こわい、鬱系こわい、鬱系こわい、鬱系こわい、鬱系こわい」ってずっと言い続けたいくらいいいですね、鬱系。

それから、今日は声優さんが大幅に変わったルパンでした。日テレなのにフジテレビが出てきたり、石田彰が安定していたり、作画がジブリの人でけっこう動いたり、まあ、色々あるんですけど、いいですね！まだ、最後まで見てないんですが、じっくり観たいです。

最後に、「～なんて怖くないもん」って言うてる人は、きっとホントに怖いんでしょうね。そういう人をあまり追い詰めないように気を遣う必要があると思いました。基本的に愉快犯なわたしには難しそうですけど。

書き忘れたこと

「とらドラ！」でわたしが誰に似てるか考えたんですけど、川嶋亜美かなー？、ってというのが結論になりました。あんなに可愛くなかったり、頭も良くないし、察しも良くないんですが、一番わたしにじっくり来ると思います。そして、理想とする人は榊実乃梨です。あの明るさとか、理由はいくつかあるんですが大きいのは川嶋亜美が嫌悪してた理由に近いものだと思います。

Fate/Zeroが配信されてて観たんですけど、ウェイバーさんの見せ所がBパートにあったのは良かったです。原作でも好きな場面でした。小物なのに妙に謙虚なところとか想像してた通りで安心しました。それと、ランサー陣営はどうしてああも不運なのかと頭を抱えるばかりです。最後まで不遇ですけどね！

セイバーの不遇は本編のヒロインだから仕方が無いというか、第五次聖杯戦争を待てな感じでした。

輪るピングドラムは、BLOOD-

Cみたいで変に伸ばしてるのが少しツマラナイです。ピンドラは1クールくらいで、BLOOD-Cは一時間ずつの前・後編くらいでよかったのではないかと。一挙放送とかなら見たいかもしれません。

Jenkinsのslave.jar

JenkinsのノードをJNLPで使おうとしたんですけど、GUIしかうまく動かせなくて困っていたんですけど、CUIの動かし方（というかslave.jarの場所）がやっとわかった（このページのおかげ：<https://wiki.jenkins-ci.org/display/JENKINS/Distributed+builds>）ので、なんとか動かせました。ただ、なぜかビルドするノードが最後に接続したノードになって全部のノードでビルドする方法がよくわかりませんでした。とりあえず、`http://yourserver:port/jnlpJars/slave.jar`にあるslave.jarをwgetして、ノードの管理画面に書いてあるコマンドを打てばいいみたいです。

Jenkinsの安全な再起動。

このページに書いてあった。

<http://d.hatena.ne.jp/gnarl/20111122/1321938292>

「システム設定」

ここ最近、Jenkinsのシステム設定が使えなくて困っていたんですけど、さっき解決方法を見つけました。

- <http://jenkins.361315.n4.nabble.com/can-t-access-configure-page-after-upgrading-1-432-to-1-433-or-1-433-td3895116.html>

によると、"Jenkins Slave Setup

Plugin"を無効化すると元に戻るみたいです。たぶん、なんかのバグなんだろうね。こうゆうのをサンドボックスでうまく守ってるソフトはうまいなーと思います。JenkinsもDebian系のパッケージも新しくユーザつくって悪い事しないようにしてるのも似たような機構なんですけど、如何せん大雑把なせいとかカバーしきれないようです。

めでたしめでたし。

母をたずねて三千里

「アニメで読む世界史」という本を読んでいるんですけど、「母をたずねて三千里」の章で長年（実際はそれほど長くない）の疑問が解消しました。

アルゼンチンは、南米の国なので黄色人種がもっと多いはずだと思っていたんですけど、アルゼンチン人のサッカー選手メッシは思いっきり白人です。なのでメッシみると、いつももやもやしていました。

「母をたずねて三千里」の章で、19世紀のアルゼンチン（主人公の母親がいる国）について解説があって、端的に言ってしまうと昔はヨーロッパからアルゼンチンに行く移民の人が多かったみたいです。その時のアルゼンチンは世界的にも豊かで日本よりも豊かだったとか。それで、主人公の住んでいるイタリアはまとまりが無かったり、まあ大変だったらしいです。そんなわけで、移民が多いってことはそこで子どもを授かる人も多いわけで、ヨーロッパの血がどんどん混じたのだと想像します。

そんなわけで、また一つ使われない知識が増えてめでたし、です。

「ココア☆ソーダ☆クエン酸」

空耳はずっと嫌いだと思っていたんですけど、そうとしか聞こえなくなるともう開き直るしか無い気がします。「ここがそう楽園さ」が実際の歌詞みたいです。

蒼穹のファフナーで翔子が死んだのが、鬱系がこわい身としてはとてもよかったです！

声優さんの気の抜けた声はちょっと浮いていましたけど、自爆ENDがとても鬱でした。脱出できない兵器というのは――。――

ノイカガナモノカと思うわけですけど、鬱系なのでもっと絶望的に脱出できないというはいいシチュエーションだと思います。

sq_pushroottable()

SquirrelはLuaと違って引数が無くても第一パラメータにコンテキストを指定しないといけないみたいです。なので、sq_callでは引数の数は1以上にしないと、"wrong number of parameters"になって悲しい事になります。実際に長い間なっていました。(・へ・)

それから、Squirrelの標準ライブラリ (<http://www.squirrel-lang.org/doc/sqstdlib3.html>)にあるregister系の関数を使うときは、先にsq_pushroottable()しとかないとエラーになります。そして「sq_reseterror()って何？」レベルでエラーがそのままなので注意が必要です。エラーハンドリングは、やっぱり難しいので頭の良い人にもっとうまくやる方法を学びたいです。

。orz

数学

数学III・Cやってるんですけど、全くわかりません。ノート（実際はルーズリーフ）が真っ赤です。もうよくわかりません。というか、気がついたらA,B,I,IIの範囲もよくわかってない気がします。とりあえず、やってみてどれくらいまでなるのか考えないとまずいです。

「蒼穹のファフナー」観終わったぜ

やっとなんとか名残惜しくもというか、最終話がさっき終わりました。まず、「今期のアニメが全部ゴミに見える」というコメントには、ものすごく共感しています。一気見補正は実際にあるのですが、「魔法少女まどか☆マギカ」以上の名作で、今年前半に自分とはいったいなんだったのか考えてしまいました。まどマギよりも終わり方も展開もよかったです。前半は言われていたほど退屈ではありませんでした。そして、後半の文句をつけようがない出来。

SEEDみたいだと騙されてみてよかったです。

劇場版と外伝も観たくなりました。TSUTAYAに行ったら探そうかと思います。もう見るっきゃ無いです！（覚えていたらの話なんですけどね。（V`*ゞ）テヘッ）

因みに見るにあたって、あんまりいいことだとは思っていませんけど、Wikipediaで予習しました。この作品は予習しといてよかったですと思います。予習しとかないとよくわからなかったところがとてお多かったはずです。

「大人がかっこいいアニメ」というコメントがいっぱいあったんですが、ホントにその通りでとても新鮮でした。あんなにいい大人は希少種らしいです。「日本沈没」もその例外の一つで、あんな大災害が実際に起きた場合、本の通りにはいかないとか。新国連は、相対的に悪く見えるのは仕方ないというか。でも、事務総長のワインを飲んでるところとかああゆうのは、なんか赦せないです。

溝口さんは、もう神がかっていました。もう伝説です。

さて、やっとなんとかこれで勉強に集中できます。（;`Д`）

BLOOD

BLOOD+とかBLOOD-Cの原作のBlu-rayが来たので観ました。Amazon.comの方が安いのはちょっと複雑な気分です。もっと長いかと思っていたんですけど、45分とアニメ2話分くらいの長さでした。

注文した動機は、BLOOD+とBLOOD-Cはリアルタイムで見ている、原作を知らないのはやっぱりよくないなと思ったからです。2つの作品を語る時に常に神格化されているのは、やっぱり気になります。

一番、印象に残ったのは保健室の先生の顔でした。今でいう顔芸だと思うんですが、表情豊かで驚愕で固まった時の顔は最高です。

サヤは、兵器という感じであんまり可愛くはありませんでした。戦闘はとてつ躍動感があり、刀を持った時の戦闘シーンの輝きには感服します。因みに、わたしにはBLOOD+のサヤが一番好みです。

設定については、吸血鬼と刀と最後のベトナムに行く伏線がBLOOD+に戦闘力・兵器としてのサヤがBLOOD-

Cに受け継がれているようにみられます。Cの最終話の素晴らしい戦闘は原作の影響なんじゃないかな。BLOOD+の膨らませ方は、賛否両論でしょうけどうまいな一と思います。6:00から夕飯を食べながら見ていたときは原作を知らなくてもそこそこ楽しんでいました。

一人の時に何度か見返したいと思います。

Pixiv

iOSのPixivアプリはWebインターフェースと全く同じになった時に削除したんですが、この度、かつて無いもとの戻るメジャーアップデートがあったそうなので、インストールしました。元に戻ったといってもいくつか改良が加えてあったので、単純に元に戻ったというようではないです。ただ、少しUIのバグっぽいのがいくつかみられました。近いうちにアップデートがあればと思います。

JA

JA

～女子によるアグリカルチャー～を母が買ったので読んでみました。絵を主に描いてる方は知ってる漫画家さんだったんですが、もう一人の方は小川村出身だそうです。一応、同県でご近所さんとか。

えっと、まずは、参考画像 -> <http://instagr.am/p/W70iN/>

です。青年誌とかエロ同人とかじゃないです。念のため。長女は農業中毒で、限界まで農業をするのが快感になる不思議体質なんです。長女は体付きが最高だと思います。変ですけど。

この表紙を見たときは「萌え絵え...」（実際は少女漫画風でした）だったんですけど、内容はよかったです。農業についてはそんなに間違っていないと思います。というか、こういうのは萌え系の絵の専売特許じゃなかったです。少女漫画の方が歴史は長いと思います。

それはそうと、「ロゴが思いっきり農○だよね」と母と話していました。わたしの知る限り○協がこれを宣伝しているなんてことはないようです。していたら、「農○え...」なのでちょっと安心します。

ちょっと気になるのは両親は、お米をつくってるんですけど、家事とか力仕事とかをしてる描写がないところです。いくら次女が料理できるとはいえ、流石に母親も多少しているはずだと思います。それから、父親は影薄いです。まともに出たのは、主人公の友人が東京から来た時だけでした。

空の境界 未来福音

けっこう前に竹箒から出た同人誌が星海社から出版されたらしいので、買って読みました。今回は、未来予知に関するお話でした。外伝らしくあまりお話は動いてない感じなんですけど、黒桐幹也と両儀式の子どもは出てきました。周りの人間の影響かとてもおかしな子でした。それから、黒桐幹也くんの「痛覚残留」の時にあった給料未払いは、蒼崎橙子さんの実力でなんとかなったみたいです。橙子さんホントハイスペック。内容については、生活に支障がないなら測定タイプの未来視が欲しくなりました。ただ、小説の中の描写を見るかぎり、生活にはかなり支障がありそうなのでゲットできるとしても眼帯で封印したいです。

ギルティクラウンの友人の弟殺しとか

ギルティクラウンは、ニコニコ動画とかのコメントに影響されてないアニメではけっこう好きなんですけど、評価が低くてorzしてます。確かに、キャラクターデザインよりもアニメの作画がよかったです（特に主人公）、名前が読みづらかったり、樫いのりの声優さんと歌の音が全く違ったり、葬儀社の存在がおかしかったり、...、確かに散々ですね。

ノイタミナというよりは、クオリティの高い深夜アニメって感じだと思います。まどかと同じ枠の方が相応しいです。UN-

GOは、ノイタミナしてますね。オープニングはノイタミナ恒例のバンドですし。それから、2クールなのは嬉しい誤算です。てっきり1クールだと思っていました。

さて、本題です。「phase 09

捕食：prey」は今まで一番よかった回でした。たぶん、森鷗外の高瀬舟（青空文庫で読めるみたいです -> http://www.aozora.gr.jp/cards/000129/files/691_15352.html

)がモチーフのお話だと思います。治る見込みのない病気の弟を間接的に殺してしまう話です。

えっと、あらすじは、『

主人公の同級生の「谷尋」（やひろ）くんにはアポカリプスウイルスというものに侵されている弟「潤」（じゅん）くんが病気の進行のせいか殺処分にあいそうになります。そして、それを阻止するために谷尋くんは弟を逃げ出してその途中で主人公「集」（しゅう）くんに出会い、集くんの判断でテロ組織（一心、味方組織）「葬儀社」に保護してもらおうとGHQ（敵組織）から逃げまわるのでした。

逃げ回っている途中、集くんは谷尋くんか潤くんしか助けられない状況に陥り、潤くんが他人の本心が見えることを告白お兄さんの谷尋くんは、内心弟の潤くんを殺したがつていることを吐露します。そして、集くんが能力（つっこんじゃダメ >

<)で具現した谷尋くんの心の形「ハサミ」（鋏とは呼べないデザイン）は潤くんの命を断ち切るものだと教えて、それで潤くんを殺すように迫りました。

果たして、集くんは予め答えの出ている天秤に2つの命をかけ、弟を兄の鋏で殺します。

』といった感じでしょうか。潤くんの声優さんは「ゆるゆり」のクールな娘の中の人だそうです。

これを見て「国語の教科書で読んだことあるなー」と思って記憶を掘り返してわかったことは、森鷗外の「高瀬舟」にそっくりだということです。高瀬舟の登場人物に当てはめると、集くんは「庄兵衛」、谷尋くんは「喜助」、潤くんは「喜助の弟」といったところでしょう。（集くんは喜助の役目も一部負っていますが、そこは考えないことにします。）

「高瀬舟」をモデルにしたというのなら、ぱっとわからないけど言われるとわかるくらいにうまく中二設定で再現したのはうまいなーと思いました。脚本はそんなによくはないという評判とはなんだったのかと、フギャフギャ————m9(^Д^≡^Д^)^9m————!!!!!!してみたいです。

ギルティクラウン最後まで視聴継続決定ですね。(°▽°) — (°▽) — (°) — () — () — (°) — (▽) — (°▽) ———!!

契約プログラミング

assertについて調べていたら、契約プログラミングという言葉に行き着いて、それについても調べました。

わたし訳だと、デバッグビルドでは特定の値を強制するコードが実行され、リリースビルドではその検査をするコードが含まれないという方法みたいです。

C/C++ではassertがこれの役割を果たしています。assertはただのプリプロセッサマクロなんですけど、契約プログラミングをサポートする言語もあるそうです。

(wikipedia):

<http://ja.wikipedia.org/wiki/%E5%A5%91%E7%B4%84%E3%83%97%E3%83%AD%E3%82%B0%E3%83%A9%E3%83%9F%E3%83%B3%E3%82%B0>

D言語も契約プログラミングをサポートする言語の一つだそうで、事前条件 (precondition)、事後条件 (postcondition)、不変条件

(invariant)を完全にサポートしていました。中でも不変条件はメソッド一つ一つに検査する関数を挿入するのは骨が折れます。

(参考): <http://www.kmonos.net/alang/d/2.0/cppdbc.html>

事前条件と事後条件は、コンストラクタとデストラクタを使えば簡単に実装できると思います。

D風に書くと (名前とかだけ)

```
template<class T>
struct contract {
contract(T const& arg, std::function<void(T)> in, std::function<void(T)> out)
: arg_(arg), out_(out)
{ in(arg); }
~contract() { out_(arg); }
private:
T const& arg_;
std::function const out_;
}; // struct contract
```

といった感じでしょうか。Squirrelのスタックを監視するコードもこんな風に書きました。デストラクタは便利です。

Boost.ScopeExitを使えばもっとマシなコードになると思うんですけど、よく知らないので今回は使っていません。

そういえば、assert内では副作用のあるコードを入れられないのがけっこう不満だったんですが、BOOST_ASSERTのページ読んでいて、BOOST_VERIFYという副作用のあってもいいマクロを知りました。早速、使ってます。(

http://www.boost.org/doc/libs/1_48_0/libs/utility/assert.html)

テストを書くよりも契約プログラミングの方がわたしには書きやすいので、こっちの方をどんどん使っていこうかと思っています。

本とかドラマ

「家政婦のミタ」の最終回を観ました。最後までご都合主義がなかったのはとてもよかったと思います。あのドラマの脚本を書いている人は「女王の教室」を書いた人だそうです。どうりでというか、有能なのに人間性に欠陥があるっていう設定は、そのままなんですけど、女王の教室とこうも違うのは驚嘆に値します。

ただ、うららさんの御見合相手が実は、主人公家族の父親が勤めていた（途中で失業）会社の取締役の娘と付き合っていて政略結婚のために別れさせられていた、という裏設定は、単なる思いつきにも見えますし、最初から考えられていた伏線にも見えて、とても大胆だと思いました。「私が恋愛できない理由」の照明会社に勤める医者の娘と付き合っていた不幸男の別の答えと「プロポーズ大作戦」をサイドストーリーでやっちゃうのはなんていうか、すごいです。スパッとやっちゃってくれた方がすっきりします。

地元出身で本屋さんでかなり持ちあげられている作者さんの本を読んでいるんですけど、まあ、面白いです。もう、亡くなっている方なのは残念ですけど。

北林一光っていう人なんですけど、ちゃんと小説してます。ライトノベル作家だったらちょっと（`・`）やがです。阿智村出身のライトノベル作家は実在するそうで、村名がペンネームに入っています。

この人の本は二冊出版されているんですが、新しく出た方を読んでいます。その前にでた罨の話は、もう読了済みです。

まだ、主人公の家系について半分位語られたところなんですけど、なんかあんまり進んでません。他にやることあるとか、言い訳はいくつかあるんですが、他に面白い本があるから、というのが大きいです。専門書がたまっているのでそっちも消化したかったり、気分によって欲しい面白さが変わります。一気に読んじゃえばいいんですけど。

文章についてなんですけど、ライトノベルとかよりも会話は読みづらかな？って思います。その分、風景の描写は使ったこともましてや聞いたこともない言葉をいっぱい使って、非常にレベルが高いです。それと、性的な表現はいまいちでした。あんまり詳しくないんですけど、「おんなのよろこび」とかもっとエロい言葉な気がします。

関係ない話なんですけど、登場人物のあだ名が兄のあだ名と同じで複雑な気分になります。キラキラン☆ネームじゃないんですが、自分の名前じゃないので余計複雑です。しかも、その登場人物は兄と全く違う言動をします。（色々）安全な名前の付け方がないものかと思いますね。

ミルキィホームズ

ニコニコのミルキィホームズ一挙放送を観ました。
萌えるベン・トーみたいなギャグアニメでした。期待させといて裏切る。ダメダメやります。
生徒会の十代がもうすぐ発売されるらしいんですけど、読みたくなりました。
キャラクターのことだとミルキィホームズよりも生徒会長が一番好きです。聖母のところはもちろ
ろん、百合百合しい願望とか、エンディングとか、わたしの嗜好を完全に網羅しています。ポイン
ンポイントな人ならもうこの人ですね。
それから、頭がお花畑なコーデリアさんもよかったですね。人気の理由がわかります。頭がお花
畑でした。

screenrc

あんまり弄っていなかったんですが、イチイチemacsとかを起動するのがめんどくさくなったので ~/.screenrc を書き換えてみました。

```
# terminal and shell
term xterm-256color
defshell /bin/zsh
defscrollback 4096

# character encoding
defencodig utf-8
defkanji utf8
kanji utf8

startup_message off
altscreen on
autodetach on

bindkey "^T" focus

vbell on

# log
log on
deflog on
logfile "~/screen.log"

screen 0 "$HOME/Dropbox/config/activate_slave.sh"
chdir ~/Dropbox/src
screen 2
screen 3 ssh vm-host.local
screen 1 emacs

chdir
```

が今の ~/.screenrc です。Dropbox を VM とのファイル共有でつなげておくのとバックアップについては考えなくていいので楽できます。screen の 0 番は Jenkins のスレーブを起動するスクリプト、3 番目は vm のホストに ssh をはってもらいます。2 は開発用のシェルで、最後にある 1 は当然 emacs です。最後にしてあるのは、emacs から使うことが多いからです。設定ファイル晒しは、文章を稼げてらくできます。

sshが繋がらなかった原因がわかった。

VMのホストにsshが繋がらなくて困っていたんですが、原因がわかって解決できました。単純に権限の問題なんですけど、sshがらみのエラーを見る方法がわかったり、思いの外収穫がありました。

まず、一番参考になったページは <http://d.hatena.ne.jp/h-yano/20080225/1203952174> です。ファイルの権限が間違っていると、鍵の探索が正常に行われず症状なんですけど、この人はログの場所を書いておいてくださったので、それがとても助かりました。"/var/log/secure"なんですけど（OS X

Lionでは/var/log/secure.log）これをtailすれば、sshdで起きたエラーの出力がわかります。これのおかげでssh -vとはまた別の角度から、原因究明をできました。

結局、かなり前に変えたホームディレクトリの権限が原因でした。sshは、ディレクトリや鍵ファイルの権限が厳しいのが有名だったんですが、まさかホームディレクトリまでチェックしているとは思いません。（ただ、再帰的にやっているだけかもしれません。（；^ω^））

原因がわかれば、あとは簡単で"chmod 700

~/."すればいいだけです。ssh関連のファイルは600、ディレクトリは700に設定しておけば問題ないと思います。

で、今日、わかったことは本来そうあるものをオレオレルールで勝手に変えるのはよくないということです。じょーしきです！（`・▽・´）！！

感想とか

サイレント・ブラッドを読み終わりました。チキンレースで早い方で脱落しちゃった感じでした。もったいぶりすぎて逆にその価値を貶めてしまったというか。最後は超常現象でレースからおりて終わってしまいました。

翠さんは正直謎解きではあまり要らないというか、あんまり感情移入できないキャラクターにでした。山姥として生き残っていたのなら、もっと強烈だと思います。トンチもそんな感じで、ストーリーの進行上まあ必要なキャラクターに終わってめんどくさかったです。ゲームにもよくこうゆうキャラいます。

それから、オニサマにもっとスポットライトをあてても罰は当たらないはずです。深雪とか深雪とかにキャラをとて喰われてました。月姫の口アみたいです。

父親も「なぜ殺したし？」か「なぜ中途半端な殺し方したし？」な扱いでした。すごいすごい言われてた伝承のキャラをいざ使ってみるとステータスが微妙で特技も微妙だったので補欠にになってしまう時の気分です。

っと、まあ、気に入らなかつたところを列挙したんですが、バランスをとるためによかつたことでも書いてみます。

まず、「深雪とオババかわいー！」です。この二人が出てる時はとても癒されました。

そして、風景描写と山登りのこと（トンチを覗いて）はとても良かったです。語彙の量にはとても驚かされました。ああゆう、文章を書ける人はそうそういないでしょうね。

School

Daysも、一挙放送があつたのでタイムシフトで見たんですけど、えっちいゲームはああゆう風に間違えることができるのかとても感心しました。ああもわかりやすい悪い例を示すのは難しいそうですね。それから、言葉と世界をそのままの意味で考えてみると、また、少し面白いかったです。とても、紛らわしい名前だとは思いますが、意図してそういう仕掛けをできたらすごいと思います。

桂言葉にもっと身近で支えてくれる清浦刹那みたいな友人がいれば、惨殺エンドはなかつたかなーと思うんですが、出来事にもしもなんてないですよ。

それと、誰がクズなのかは

GUNSLINGER GIRL

3が来たので読みました。活字と違って、すぐに読み終わるのはいいです。

今回は、ピノッキオとトリエラの初戦でした。人外の運動能力相手にああも、闘えるっていうのはやっぱりかっこよく写ります。プライドというのはああゆう風にズタズタにするんですね。覚えまして。

ベン・トーの最終回が昨日ニコニコで配信されてました。

エンディングテーマで始まりオープニングテーマで終わるのは斬新でした。わたしの見たアニメでテーマの逆転はまだ見たことありません。

内容ですが最後までちゃんと走りきつたのはとても爽快でした。オルトロスの追放は、どうやってやったのか気になっていたんですが、とても残酷で意表を突かれました。姑息ですが、最も有効です。

そして、明確なラスボスがないのに物語が成立しているのは、なんとかのアンチテーゼとか色んな言い方があるんですが、単にラスボスも味方も空腹だったからなんでしょうね。面白いのは正義です。

松本にできた本屋さんに行ってきた。

家族から松本に大きい本屋さんが出来たと聞いたので、父といっしょに行って来ました。丸善書店というところですよ。書店よりも石油が先に出てくるのがとても謎い気がします。

現地に着いてみると、ビル丸ごとが書店でした。でっかいです。駐車場は近くのデパートのところにとめました。お会計の時に聞いたんですが、丸善書店で本を買って利用料金が安くなる駐車場は、ちょっと遠いところにありました。電車で来た方が楽しめる書店なようです。

書店には、地下一階、一階、二階といった構成になっていました。各階が、地下一階は専門書、一階は文房具や雑誌・登山情報・その他もろもろで、二階は漫画・ライトノベル・女性向け・絵本など一般向けといった風に分かれています。

今回の一番の目当ては、専門書でした。理由はいくつかあるんですが、近所の本屋さんで専門書は期待できない、Amazonでは本を立ち読みできないなど、需要があっただけです。結局。

内容ですが、たぶん、そんじょそこらでは置いていませんし、一度読んでみたいなーと思ったのもいっぱいでした。とりあえず、幸せ気分を補給できます。関数型言語の本もいっぱいありました。あんなにいっぱいあるのを見たことがありません。それと、アルゴリズムとかAmazonでしかお目にかかったことのない本もいっぱいでした。

それから、半導体に関する本がいっぱいあって「キャ———— (°▽°) —————!!」って叫びたくなりました。キモいのは承知です。悶えます。「CPUの創りかた」の扱いが大きかったのはとても意外でした。わたしはもう持っているので読み返そうかと思いました。

ただ、グラフィック関係の本をあんまり見かけなかった気がします。OpenCLはあるのにOpenGLが無いのはとてもおかしいです。ちゃんと見ていないだけかもしれないので次はちゃんと探してやります！

もう一つ、最近TwitterのTLでよく見かける「ゲームプログラマのためのC++」も見つかりませんでした。探し方が悪かったかもしれません。

最後に、今更気がついたんですが、蔵書検索システムがちゃんとあったのに、使わなかったのが今でも疑問です。それを使っておけば、探し方が悪かったなんていう必要なかった気がします。

丸善書店は、センター入試の帰りにまた行こうかと計画しています。何度行っても飽きない自信があります！

今年の振り返りとか。

あんまりすぎじゃないんですけど、書くこと無いので今年を振り返ってみます。

まず、プログラミングの話です。

C++は、闇の軍団予備軍くらいにはなれたと思いたいですね。わかったと思ったらわからなかったり。それと世間ではHaskellが人気らしいのでそっちに手を出さないと取り残されそうです。文法とかチュートリアルとか予習ばかりはしているんですけど、いかんせん触って見ないと意味ないと思います。なんかテキトウな本を一冊買ってがっつり使えるになるのが来年の目標の一つです。

CMakeが無駄によく使えるようになりました。頑張れば、ビルドシステム以外の目的に使いそうなくらいです。しませんけど。

Squirrelを今年の終わり数カ月がんばってました。ただし、あんまり形にはなってません。なので恥ずかしい限りな感じです。luabindのSquirrel版は頓挫して、まあ、アレです来年どうにかできたらいいですね！Lua 5.2が出たのでLuaに浮気したいですけど。

続いて、勉強の話です。もっとすればよかったに終始します。今年の終わりの方になってちゃんとやり出しました。こんなんじゃ何もしてないも同然なのがよくわかります。プログラミングする時間が減るんですけど、こっちの方がしないといけないことなので、増やしていきたいです。まだまだ、辛いことなので楽しくなるまでどんどんやっていきます。喉元すぎればなんとやらです。

それと、今年のアんまりいらなかった進歩について語るときましよう。

家事が上達しました。ええ。料理スキルや洗濯スキルや掃除スキルや後片付けスキル向上しています。

今年の大晦日では大活躍でした。裏方の気持ちがよくわかりました。

今年のアニメについてですが、魔法少女まどか☆マギカよかったです。でも、それ以上にファフナーに出会えてとても幸せな気分を味わえました。「虚淵てめー！」よりも「沖方てめー！」です。上には上がいました。

ま、そんな感じでした。わたしの趣味しか書いてないですね！

よいお年を！

お正月テンション

おかおめ。ことよろ。(死語)

ひらがな四文字で挨拶が済むなんて楽ですね。死語になっているのが残念です。便利なせいかきっと言葉自体の擦り切れるのがとてもはやいのでしょうか。因みに、字数を稼ぎたいわたしとしては迷惑です。長々と挨拶する方が煩わしいので、あえて使っているだけなので気にしないでください。

お正月らしいです。今回の年末年始は家の人口密度がとても上がりました。大晦日は紅白見てお蕎麦を食べて、終了でした。韓国人の人気のないグループが出てきた時以外、ずっと紅白です。妹はテレビオタクなので便利な解説役に徹していました。テレビオタクって便利ですね。

お正月は、毎年いく近くにある神社に初詣に行きました。近くには穂高神社とか諏訪大社とか比較的神秘的の多い地域らしいので有名所がいくつかあるそうですが、そこまで行く義理はないので毎年一番近くの神社に行きます。10円じゃ少ないと言われたので、110円をお賽銭として奉納しました。お祈りしている眼を閉じている時間というのは退屈ですね。

っと、まあそんな年末年始でした。そこまで派手じゃないです。そんな派手な日常も送っていません。

特筆するべきでも無いんですが、2012/01/01に寝坊しました。夜更しをしたのが原因です。なので、御節料理は年末に食べたもの以外特に食べていません。お雑煮は余っていたので食べました。おいしかったです。(ゲッフ)

一斉放送されたアニメとか

蒼穹のファフナー Right of

Leftを観ました。「沖方でめー！」なのはさておき、本編第一話で死んだ眼鏡の子がとても活躍しています。ティターンモデルかっこいいですね。宮野真守はガンダム00で名前をやっと覚えたんですが、これで知りたかったと切に願いました。STEINS;GATEとかの演技もいいと思うんですが、やっぱりシリアスな演技の方が締まります。この思いを更に強くすると杉田智和も、銀魂の前に知りたかったです。シリアスな役をやってても笑ってしまいます。

ただの定期配信なんですけど、Fate/Zeroを観忘れていてさっき気がついて観ました。分割2クールだそうで、次回は4月以降らしいです。今回も続きが気になる終わり方をしました。関係ないんですが、あんまり期待させる終わり方をすると、かなり時間を空けて次見ようと思った時のハードルとかどうしてもよさがめんどくさいです。関係ないんですが。

キャストがとうとう本気を出したそうです。だとするとバーサーカでミリタリー的なニヤニヤができる展開とかエク〜スキャリヴァ〜解放とかランサーの敗北とかはまだなんですね。個人的に思い出に残っているところは全部第二クールに残っているのもとても楽しみにしています。

また感想。

お正月で色々消化してます。

「生徒会の木陰」「生徒会の金蘭」を読みました。この前、丸善書店で買った本です。2年B組の話の完結編でした。全体的な感想は、モノゴトは終わらさないほうが良いと、ここ数年ずっと思っていたんですが、ちゃんと終わらすのも悪く無いかなと思いました。物書きの才能はないので、プログラムの話なんですが、やっぱりコードはちゃんと終わりまでかけたことは少ないです。明確な終わりが無いのは承知なんですが、満足と終わりのうまい平行線の交差点を見つけられるように、努力します。ああ、恥ずかしい。あんまり関係ないんですが、中目黒先輩を保志総一郎がやったらいいなー、と思いました。「おねがい☆ティーチャー」を観たり、キラ・ヤマトの外道っぷりを知ったです。はい。ひ弱でドSがぴったりフィットします。

化物語の一举放送も見終わりました。

アニプレックスがステルスマーケティングでゴニョゴニョってよく言うんですけど、まあ、面白いうちはいいじゃない？と思うくらい面白かったです。キャラクター同士の掛け合いとか、オープニングとか、放送当時ちゃんと見てたわけじゃないので改めてみてみるといい作品でした。

トップをねらえ！の無印も見終わりました。

「名作はいつでも名作」っていうのはホントでした。「(° ▽ °) o 彡 おっばい！おっばい！」だったんですが、パイロットの時間がどんどん地球人と乖離していく描写は残酷で、うるうるしました。光の速さについて描いた「ほしのこえ」くらいいいですね。2のダイバスターx4はきくだけで笑えこらえ切れなくなります。

一応、勉強もちゃんとしてます。教育テレビ見たり、問題集を読んだり（解いてはないです）、数学やったりです。原発の特集は、まあ、歴史的に大事そうなので今のうちにちゃんと勉強しとけば後々役に立ちそうなので前後編で三時間ですが見てます。まだ途中なんですが、ネットで言われているほど悪いって感じじゃないのがやっぱり残念ですね。キメラアント編でキルアが言っていたわかりやすい悪者を倒すみたいに罪悪感が残らなければと願ってしまいます。スマイリーキクチさんのいじめに加担したような気分です。

まとまったお金が入ったので。

サーバ機に使えるようなベアボーンPCを探してました。機械が嫌いじゃない質なので、こうゆう選定をしていると楽しいです。それと、ベアボーンという言葉をやっと知りました。まあ、このほう

がとりあえず、小さいこと条件にすると

- EeeBox (ASUS)
- Aspire Revo (Acer)
- ZBox (Zotac)

になりました。大きさは、CDケース並らしいのでテレビにつなげそうです。

CPUについては、D525かAPUが最高でした。D525でもいいんですが、周波数は少ないんですがAPUは使った事無いのでこっちのほうを選びたいです。それと、APUだとOpenGLとDirectXの新しいバージョンが使えるのでなおさら優先順位が上がります。IONというGPUがあるそうなんですが、OpenGL 1.5と言われるとう〜んというか。

それから、Bluetoothは欲しいです。有線とか邪魔臭くて困ります。

で、まあ、一通り見てみるとZotacのZBOXNANO-AD10-

PLUSがいいという結論に達しました。OSが付いてないんですが、値段的につりあってこれでいいかなーという感じです。Windowsほしいんですけど、あくまでオマケで欲しいという、めんどくさい理由なので今回はもういいです。USBの外付けドライブがあるのでLinuxの好きなディストリを入れてサーバにできれば実際はどうでもいいのかもしれませんが。

余談ですが、香港のZotacっていう会社は今回はじめて知りました。ASUSとかAcerみたいな部品メーカーみたいなんですけど、GPUの技術は他の2つよりも高いみたいで、OpenGL

4.0対応のPCを安く売ってるのは♪(°Д°)ガッツョブ!!だと思います。今まで、知らなかったのを後悔しているくらいです。

Zotac覚えましたし。

東のエデン

ニコニコで一挙放送があったので抜かりなく予約しておいたタイムシフトを見終わりました。東のエデンはわたしが深夜アニメを見始めた頃の作品で、この作品のおかげで今も深夜アニメを見続けている気がします。ただ、アニメが終わったあとはそれっきりで、映画化の話も聞いていたんですが、見てはいませんでした。なので、今回の劇場版込みの一挙放送は僥倖としかいいようがありません。ニューヨークのど真ん中でヤッター！したいです。

まずは、テレビ放送された方についてなんですが、まだ数年前の作品なのでけっこう覚えていました。名作ゲームみたいに前はわからなかったことがぱっとわかるのが楽しかったです。それと、数年前は今ほど物分りがよくなかったのも、さらに深く見れたと思います。放送当時の「わけがわからないけどワクワクした」感じは蘇らなかつたんですけど、それでも展開のおもしろさに何度もニヤニヤさせられました。わたし、キモイ。

劇場版Ⅰは、エヴァの「破」みたいな感じで、続きが気になりました。放映された当時見ていたら、どれだけ悶々したのか想像もできません。ノブレス携帯はそんなに役に立ってたいなかつたというのは、目的の達成にやむをえなかつたというか、正しい判断かなーと思います。そこそこ役に立っていましたし。それはそうと、ジュイスがどんどん萌えてきました。「なんだあの加(・V・)!!人工知能は!？」と何度も思いました。いっそ欲しいです。

劇場版Ⅱは、政治家風という一定の目処をつけたというか、悪くない終わり方でした。結局、滝沢は日本中に種を蒔いただけで、インキュベートする存在がまだいないといった終わり方なのは、まだ続きがつくれそうですね。結城くんと物部さんの最後は悲惨で二人はもっと別の終わり方をあげられなかつたのかなー、っと後味の悪さを噛み締めます。辻ぼっちゃんも、最後まで自分を押し通したというか、いつまでたっても富豪ポジションでした。この人のジュイスは調教されてんなーというの印象に残りました。

結局、セレソンは全員でなかつたので、サイドストーリーとかスピンオフでの活躍を期待したいです。それから、ジュイスの本体のデカさも二回目のゲームで変わったりしないかなーと思います。聖杯戦争みたいに第何次かでユニークな形になったらいいですね。

こうゆう一挙放送があるのはボックスが発売されるとか、新しくコンテンツを提供する場ができるとか、そうゆう時くらいしかないのがなんだか残念です。ただ、そもそも、それくらいの作品じゃないと見れないのかもしれませんが、はやく、自分の好きな作品のボックスを買えるような身になりたいです。たぶん、こんなこと思わなくて済みます。

とりあえず、東のエデンBlu-ray Boxほしいです。

午後のロードショー 「The NET 2.0」

ハッカー同士の熾烈な闘いかと思ったら、Enemy of the Stateの二番煎じでした。期待して損をしたというか。まあ、これは主人公が美少女なので、萌による二番煎じと大して変わらないのでそこまで文句を言えません。

「ラストxx分であなたは騙される」的な展開がなかったのは個人的に嬉しかったです。精神の分裂とかだった暁には、数時間鬱くなります。そこはほんとよかったです。

それから、主人公の美少女は異様に強かったのはちょっとずるいと思いました。普通、肉体的かスーパーハ-的に強いのかどっちかです。天は二物も三物も与えるのですね...

そして、インターポールの人がすごい間抜けというか、なんというか。最初から主人公を囮にするなりに、囮だということを告げておくべきだと思います。そうしておけば、最終的に主人公にネコババされることもありませんし、主人公をもっとうまく立ち回らせて負担もなかったはず。それに、告げるチャンスもかなりありました。最初に会った時、ストーキングしてる時、敵に襲撃された時、...etc。

最後に、女にやらせてみたかったんだよ！的な意気込みは(

・V・)イイ!!と思います。かなり、ミスリードしているのは気のせいです。

雑多な感想とか

クローズアップ現代の世界経済についての回をみたんですが、暗い気持ちになりました。「みんなが笑ってる〜♪」（日本）だった状況が今は、もう笑えなくなったそうです。「ガマ————m9(^Д^9m————!!!!!!」っていう気分でもないんですが、日本は更にその先にいって

るそうなので、表情がひきつるというか。それから、好きな作品でアメリカが弱体化しているのが、気になりました。すぐ挙がるのは「Darker Than Black」の一期と「とある魔術の禁書目録」なんですが、アメリカは次に滅びる大国っていう印象が強いです。日本をその大国の定義に含めるなら、順位が微妙になるのですが、自国なので例外ということにしときます。アメリカに渡った人たちは、理想郷をつくるために尽力したそうですが、どんどん地獄に変わっていくのはなんというか、哀しいです。まだ、250年も歴史がないと思うと千年以上歴史がある国って化け物に感じます。盛者必衰の理とはまだ残酷なのです。

Google IMEで「ばーじょん」変換するとGoogle IMEそのもののバージョンを変換できることに気が付きました。どうでもいいですよん！

NHKの「さかのぼり日本史」という番組で武士が起こした革命についての週を見終えました。時代は平安末期から、戦国前までなんですけど、改めてこうゆうのみると歴史の授業で教わったのと全く違ったので驚かされます。

数百年もかかってはいるんですが、世界を見渡しても国の一生で、軍人が国をちゃんと統治できたなんてことは、そうそうないので、それを成し遂げたご先祖様は偉大だと思いました。

それから、武士の価値観なんですが、何度もパラダイムシフトが起こった末の賜物だというのは全く知りませんでした。自然とそうなったと片付けてもいいんですが、撫民政策は世界中見渡してもそうそうできないと思います。（繰り返し、この単語が出てきたせいで覚えたわけじゃないです。）

で、少し関係ない話なんですが、日本史ってそこら辺だとダイジェストで飽きない程度に楽しめるんですが、時代が現代に近づくにつれて、だるくなるのは同属嫌悪からくるものなのか考えてしまいます。世界史の方は今かなり書き加えられている状況なので、楽しいです！歴史の生き証人という人もけっこういっぱい残るので、人類滅亡せずに落ち着いたら、出版社からいっぱいそういう本が出てくるのも楽しみにしています！

グラハム・ベル空白の12日間の謎

ブックログURL

- <http://booklog.jp/asin/4822284395>

図書館で見つけて面白そうだったので読み終わりました。内容はけっこうしっかりしていたので、買ってよかったと思います。

第一印象は、注意深く書かれているみたいなので、数学の「よって証明終了」的な読了感でした。世の中、いくらでもこんながあるので今更驚かないというか。

ネタバレな内容を言ってしまうと、ベルは電話のアイデアは思いついてはいたけど、その実現には程遠く、とある理由でイライシャ・グレイの発明を盗むことになってしまったといったところ。で、その理由なんですがベルのお嫁さんのパパ（弁護士）が当時電信を牛耳っていた企業の汚職が目にあまるものだったので、それに対抗するためにだとか。実際に可変抵抗という電話に必要な部分をつくったイライシャ・グレイは、たまたま電信を牛耳っていた企業側に立っていたというのは、きっと不幸な偶然だったのでしょう。

善悪の裁定はともかく弁護士の政治闘争は勝ったのは、かなり歴史を変えたのでそれはそれで意義のあったことだと思います。ベル研究所ではUnixが生まれましたし、敵対勢力は今ではかなり弱体化していました。

ただ、この本で書かれているイライシャ・グレイは決して申請に遅れていない事実をもっと広げるべきです。グラハム・ベル自身もそれを望んでいる手記もありますし、間違った歴史は正されなければなりません。グラハム・ベルは電話を取り巻く人の予想を裏切り広めた立役者として、イライシャ・グレイは電話を発明した立役者として、それぞれ正しく発信されることを望みます。

ここからちょっと傍道にそれた内容です。

この本を読んで思ったんですが、科学史をちゃんとやってみるのも悪くないなと思いました。自分の探求力に限界を感じてしまうと、未練をどうにかしたくなってるだけだと自覚はしてるんですが、そんな一本道でできる自信はないので、別の道でもがんばってみます。

ただ、アメリカの科学雑誌の編集者（Editor）は自身も超一流っていう話を思い出すと、「ひーひー」と後退ってしまいます。orz

screenでスクロールとか。

目的のページが見つかって、screenのシェルでマウスホイールでスクロールできるようになりました。

↓のリンクから

<http://superuser.com/questions/126606/how-to-get-mac-os-x-terminal-app-and-screen-vim-scrolling-to-play-nice>

"

```
# scroll
```

```
termcapinfo xterm* ti@:te@
```

"

を~/.screenrcに追加すれば使えました。

Dropboxのキャッシュをクリアする新しい方法

Macだったら

```
"rm ~/.dropbox.cache/*"
```

と打てばいいみたいです。 http://dropboxwiki.com/Forums_FAQ/Clear_Dropbox_Cache

には

```
"rm ~/.dropbox/cache/*"
```

と書いてありますが自分の環境で試してみたところ、`~/.dropbox/cache`には何もなかったのでそれっぽいディレクトリ名が補完候補にあったのでピンゴでした。それから、キャッシュを消す前にはDropboxを停止するのを忘れないようにしたいものだと思います。

MacPortsの設定とか

MacPortsで"port uninstall"を使うとたまに、アップグレードされたためにいくつかの違うバージョンが入っていて

"`port uninstall inactive`"
を付けても消せなくて、インストールされているバージョンを全部消さないといけないんですが、そんな時は、

"sudo port uninstall inactive"

を使うといいみたいです。(<http://apple.stackexchange.com/questions/10149/how-to-remove-unused-macports-packages>)

要は、activeでない(=使っていない)パッケージを全部消してくれるコマンドです。これをたまに実行しておけばディスクの節約にもなりますし、一々使っていないバージョンを指定して消す必要もなくなります。

それから、MacPortsのコンパイルを速くするためにいくつか新しいオプションを使ってみました。

<https://trac.macports.org/wiki/howto/ccache>

<http://d.hatena.ne.jp/hazy-moon/20090217/1234840827>

基本はccacheとコンパイルオプション"-pipe"です。

前者は、繰り返しコンパイルしている時に役に立ちます。/opt/local/etc/macports/macports.confにある"configureccache"をコメントアウトして"yes"すると使えるようになります。必要ならば、ccache_dirやccache_sizeを適宜変更するといいと思います。

後者は、メモリが豊富(10GBとか)にある環境で有効なオプションだと思います。コンパイラはいくつかフェーズがあるんですが、そのたびに一時ファイル(テンポラリファイル)が作られますがそれをファイルを介さずに全部メモリ上でやるよう要求するのが"-

pipe"オプションです。ccacheと同じようにmacports.confにある"configurepipe"をコメントアウトして"yes"にすると使えます。おそらくメモリが1GBとかだと少しパワー不足になるでしょう。512MBの仮想マシンで試してみたんですが、想像したよりも性能は出ませんでした。

それと、distccもあるみたいですが、これはネットワークがボトルネックになったり設定がめんどうだったりあんまりおすすりできません。Googleでこの問題点を克服したソフトがあるそうで、早くインフラそのものが公開されないかにやーマダァ?(・∀・)

)っ／＼☆チンチンってな風に待っています。

言い忘れましたがbuildmakejobsも少し盛ってみるといいかもしれません。あんまりいいCPUを使っていないのでよくわかりませんが。

Github Growlerを動かしてみた。

Growlを買ったので、GitHubの通知を受信したいと思いたち

- <https://github.com/DAddYE/githubwatcher>

- <https://github.com/ctshryock/GithubNotifier>

- <https://github.com/miyagawa/github-growler>

を試してみたんですが、「コマンドラインは嫌」「自分のアクティビティなんてどうでもいい」と行った理由でgithub-

growlerを使うことになったんですが、なぜかLionでは動かなかったのをちょっと検証してみました

。わたしの理解の範囲でわかった原因は端的に言ってしまうと、"use Cocoa::EventLoop;"が悪さをしてるみたいです。APIが変わったとかライブラリの変更か何かでCocoaのバックエンドは使えなくなっているといったところでしょうか？

で、そのコードをコメントアウトしたら動いたのでコミット、Pull Requestしてみました。

- <https://github.com/miyagawa/github-growler/pull/10>

それと、ビルドしたバイナリもあげておきました。

- <https://github.com/take-cheeze/github-growler/downloads>

因みに以下は、ビルドするときにした処理です。「Perl (><;)

わかんないんです！」っていう身にはこうゆうのはやっぱりよくわからないですね！

"

```
cpan -i .
```

```
perl Makefile.PL
```

```
make
```

```
sudo port install platypus perl
```

```
ln -s /opt/local/share/platypus /usr/local/share
```

```
curl -kL http://install.perlbrew.pl | bash
```

```
~/perl5/perlbrew/bin/perlbrew install-cpanm
```

```
PATH=$PATH:~/perl5/perlbrew/bin ./builder
```

```
"
```

Platypusのバグというか不具合というか

前の記事関連なのですが、Platypusがうまく動かなくてググったら、

- <http://trac.macports.org/ticket/32974>

を見つけてorz状態になりました。普通、こうゆうパスは実行ファイルに埋め込むものじゃないと思っていたんですが、そうでもないようです。バイナリ書き換えたり、Portfileをごにょごによしようかと少し悩んだのですがシンボリックリンクを使えば解消しそうだったので試したらうまくいきました。

なので、Platypusを使う前に

```
"ln -s /opt/local/share/platypus /usr/local/share"
```

を実行すればこの不具合はなんとかなります。こんな使い方もあるんですね。

cron-apt

関わっているプロジェクトのサーバ管理人さんがcron-aptを入れていると聞いてわたしも試してみました。

- <http://www.kotodama.jp/blog/index.php/2008/03/18/ubuntu-apt-cron-apt>を参考にしています。

とりあえず、dist-

upgradeを自動で、一時間おきにやってもらっています。IRC通知がないか調べてみたんですが、メール通知はあるんですがIRCはないみたいです。

Bazaarでござーる

LaunchpadがBazaar限定と聞いて、git-bzrやらbzs-gitを試しているんですが、どうもうまくいきません。git submoduleを使ってるのがよくなかったり、Recipeの書き方がよくわからなかったり…。それから、パッケージの依存関係を定義するのとかも、けっこう面倒です。ってなわけで、bzrは使わないほうが無難かにゃーというのが今のところのスタンスになっています。debパッケージをアップロードすれば、aptから使えるみたいなので、debパッケージをつくる方法さえわかればいいみたいです。因みに、LaunchpadではRecipeでrunコマンドが使えないので、旨味もありません。それから、依存パッケージもnmで'U'が付いているシンボルを使うと大体のやつがわかりそうでした。で、その後で、CMakeに含まれるパッケージングソフトであるCPackがけっこうイケてる子と知って、これを使えば、パッケージ依存関係さえ緩くやれば別にいいかなと思うわけです。install関数で指定したものはこれで自動的に、debパッケージ・Windowsインストーラ・Macインストーラ・RPMパッケージを作れます。全く幸せになれますね。

ゆるゆりとFate/Zero

ニコニコ斉放送があったので、タイムシフトで見えます。
ゆるゆりは、ガチレスが好きなわたしにはとても物足りなかったです。ただ、＼アッカ〜ン／には同情します。あかり、不憫な子。やっぱり、ガチレス成分は、ふぐり屋さんで補給するべきですね。

Fate/Zeroは、Fate/Stay
Nightを全クリしたくなりました。Fate/Zeroの原作は読んであるんですけど、やっぱりその原作を読まないとあんまり楽しめなくなっているようです。PS2の廉価版は二千円前後なので、落ち着いたら買ってエミュレータでやってみようかと思います。好きなヒロインは、間桐桜なので。

ゲームプログラマのためのC++

ちょっと前に話題になっていた本を読み終えました。

まず、中盤くらいから読み終わったら誰かにあげちゃいたくなっている自分がいました。なんといいですか、すごいすごい言われてた敵が蓋を開けてみたら初歩的な対策をしたらすぐにやられた感じです。ちょっとがっかりしました。

なので、この本はC++入門書の次くらいに読む本でした。間違っても、(わたしみたいに)「C++の【闇】もっと教えて下さい\(^o^)/」なノリで読んでもそんなに楽しくないです。誰か欲しい人いたら喜んであげちゃいます。

少し内容に突っ込むと、識別子とかは、クラス名や関数名はキャメルケースで、当然boostさんもあんまり出て来ません。STLコンテナの性質は一通り説明があって親切でした。わたしの場合はネットにあるようなリファレンスでも十分な人にはあんまりいらないですけど。STLのアルゴリズムにも一通り説明がありました。

デザインパターンは、よくあるあんまりよくないけど説明に適してそうな実装が親切に書いてあります。本一冊できるような分野(?)なので、ホントに理解したいならそうゆう本を一冊読んだほうがいい気はしました。シングルトンは昔少し調べたくらいなので、いくつかツッコめました。本の内容にツッコんでどうするのかはさておき、関数内でstaticに定義するタイプとか、初期化のタイミングとか。まあ、使えればいいものなので、どうでもいいことです。

それから、スマートポインタはもっと口が酸っぱくなる・耳にたこができるくらい取り上げて欲しかった思いがあります。new/deleteなんて普通は書かないはずで。

っと、まあそんな感じで読んでました。他にも色々あった気がしますが、ただのいいがかりレベルな気がするのでこの辺で。

Node.js楽しい

- <https://gist.github.com/1133326>

がいくつかの理由で、遅かったのでnode.jsで書きなおしてみました。

- <https://gist.github.com/1976029>

がその結果で、一応はよくなりました。まあ、それも当然で画像一個一個がダウンロードされるのを同期的に待っていたら時間はとてもかかります。なので、画像URLは一秒ごとに50個取得して、画像のダウンロードは250個ずつspawnで起動したプロセスでwgetしています。

試しに自分のTumblrアカウントで試してみたところ、一日位かかっていたダウンロードが、数時間くらいで終わりました。非同期の力すごいです。

あんまり関係ないんですが、setTimeout関数がうまくいかなかったんですが、匿名関数呼び出しを使ったらなぜかうまくいって少し釈然ともしませんでした。ただ、ちょっとした調べごとで下記のURLでも同じようなことがあったらしく、なんかの仕様みたいです。

- <http://dqn.sakusakutto.jp/2011/06/javascript-nodejs-sleep-sort.html>

LAST EXILE一挙放送

- <http://live.nicovideo.jp/watch/lv82519660>

でやってたやつタイムシフトをやっと見終わりました。12時間は色々ときつかったです。終始、しっかりした舞台設定と最先端技術に感心させられました。ほとんど粗の無い物語とか、良作というのを思い知った感じです。

で、まあ、今放送されている「銀翼のファム」とも当然比較したくなるわけですが、ファムの最後までを見ていない現在は、LAST

EXILEの方が楽しめました。00年代前半の作品であのCGはカッコいいですし、主人公の色恋沙汰がそれほどイライラせずに見れるのはとてもいいです。あらかさまな「もえー」とかもありませんし。

銀翼のファムでは、想像していたよりもCGが進化してなくて少しがっかりした覚えがあります。例えば、ヴァンシップに乗っている人のCGがひどかったり、前回では暗めの色の中で生えていたCGがファムでは、明るい色が多くてわたしでも少しおかしいなと思うことが多かったです。シナリオの進め方がわたしには合っていないらしく、所々抜けている気もします。もうちょっと漢と書いてオトコな登場人物がファムたちの加(・V・)イ!!日常の裏で戦争のシナリオを進めてくれていた方がわかりやすかったかもです。

それから、タイトルコールなんですが、なんであんなに色っぽいんですかね？ 声なんですか？ それともこの狂った感受性なんですかね？

最後に、「よういよ」が終始正義でかわいいのは卑怯だとは思いますが、思わずその空気に流されてしまう説得力があると感じました。あんまり裏付けがなくても自然なのはなんというか魔法です。

夜明けのブギーポップを読んだ。

ブギーポップシリーズを読んだことなかったんですが、それだけ借りるのも気は引けるような本を図書館で借りようと思ったら、シリーズの一つを見つけたので借りてみました。結局、目的の本よりも面白くて、シリーズの初めの本を買いました。

奈須きのこ先生の世界観が独特だったというか新鮮だと感じていたんですが、ブギーポップの方が古いし直接的な影響もあるので、その見方を徹底的に変えられてしまいました。歴史的な設定はあんまりなかったんですが、抑止力とかは根本的な部分は、ブギーポップのほうが先でした。

新しいと思っていたものが、何か別のものを全く別に焼き増ししたっていうのはよくあるんだなーと思いました。

「龍の起源」

ちょっと前に衝動買いした本をやっと読み終わりました。タイトルの通り龍の起源をたどる本です。

内容は大きく分けて、ユーラシア大陸の東西に存在する龍・日本の龍・その他、だったかと思えます。外国の龍についての話は面白かったんですが、日本の龍については河童とか天狗とか、まあ脱線気味でちょっとつまんなかったです。読める時にはそんなに感じなかったんですが、けっこうな量の文章もさかれていました。現代の創作について考察されたらそれはそれでドン引きだったり量が凄まじいことになりそうなので、話半分で読むにはちょうどよかったかもしれません。

この本では、龍の起源は基本的に2つで、メソポタミアと中国から発したようです。蛇を神様に祭りあげる文化は世界各地で生まれてて、それが龍に至るわけですけど、龍に至るまでの条件がなかなか厳しいので2つしかないのは驚きでした。で、その条件ですが、

- 大河の近くの国家

- コブラ（または、巨大な蛇）がない

だそうです。インダス文明とエジプト文明は、前者の条件は満たしているのですが後者はコブラがいるので、人々の想像力を閉口させてしまうので龍までには至らなかったとのこと。前者の条件についてですが、蛇は水と密接に結び付けれます（というのがセオリーだそうです）が、大河という規模になるとどうしても蛇だと小さくてイメージにそぐわないので、龍のような巨大な偶像が必要になったと書かれていました。人間の認識というのはまためんどくさいなと、再び思います。

余談なんですが、日本の龍は中国の方からきたそうです。最も、文明開化やらゲームのてけと一設定で西側のも大分ごっちゃになっている感はあります。それと、縄文時代中期から農耕がはじまっていたかもしれないというのは興味深かったです。确实というわけではなかったのですが、いくつかの証拠は面白かったですし、学校で教えてもらったことの更に先を知ることができるのはとても有意義だと思います。

日本の龍については、終始外国の文化だったそうです。理由はよくわからなかったんですが、皇室には龍はあまり入り込まず、中国の王様と差別化が図られているといった記述があります。そして、著者が龍は河童や天狗に変化したという仮説を提示しているわけですが、そこまでいくともはや別物なのではないかと思えます。それから蛇を信仰する文化も日本にありますが、龍と無関係ではないのですが上の2つの条件を考えると今回は別物と考えるべきです。

ユニコード戦記

ブログとかTwitterで、見かけて最近できた専門書がいっぱいある本屋さんで買ったきり積んでたんですが、今日やっと読み終わりました。本職の作家がかかれた本では無いので少し読みづらかったのですが、なかなか歯応えがあって苦にはならなかったです。

内容ですが、ノン・フィクションかと思って読んでたんですが、Unicodeみたいな標準の中身がどのように決められるのかといったことの方が大きかった気がします。それから、小林龍生さんの視点で語られてはいるんですが、毎度登場する「ヒデキ」こと樋浦秀樹さんが真の主人公じゃないかとかこよく言ってみたくなる内容でもありました。頑張れば、『アッ--!』な薄い本が書けそうでした。※冗談です。

島国島国言われる日本ですが先人たちの努力のおかげで大国に数えても差し支えないせいか、日本では深く考えずUnicodeは使えているようです。世界には、独自の文字コードすらなくて、Unicode入りを果たしたとしても諸事情で不完全な哀しいものとなったりするなんて事件もあると知ったときは、「このbonbonめ!」と自分を罵りたくなりました。

理想を手に入れた側がその理想論を押し付ける残酷さというか、もっとそういう歴史を知らないとなと思います。「強いられているんだ!」という言葉はどこまでも強烈です。

最終回ラッシュ

季節の変わり目なので、最終回ラッシュです。ドラマはほとんど終わっているんですが、アニメはまだいくつか残っているの、終わるのが寂しいながらその結末を楽しみにしています。

今のところ、一番インパクトがあったというか、稀に見るやり逃げというか、銀魂は強烈でした。あの構いなしな姿勢は学ぶべきかもしれません。それと、あそこまで受け入れがたい(・V・)イイコトイッタ!!はなかったです。不潔です。

ラストエグザイルは、散々でした。前作はきっと何かの奇跡だったのだと思いたいくらい変な終わり方でガッカリしました。主人公が要らない子というのは不幸を乗り越えて、何かの間違いだと思いたいです。新作の漫画が出ているそうなので、今度本屋さんに行ったら探そうかと思えます。アニメ関係者以外の方が書いたノベライズならもっと読んでみたいかもしれません。無印の一挙放送を観なかったら、こんなこと言っていないと思えますけど。

ギルティクラウンは、何が何だかできの悪い、名作ゲーム×アニメを見ているような気分でした。プレイヤーのために舞台を用意したけど、観客は全く楽しめなかったといえますか。絵とか音楽はとてもよかったです。オハナシダイジ。こちらはノベライズがでているそうなので、見つけたら買いたいです。本屋さんに行くたびに探している気がするのですが、見つかりません。ヽ(´ー´)ノ

ソ・ラ・ノ・ヲ・トの再放送も見たんですが、最終回だけ妙に物議を醸していたというのが、信じられませんでした。確かにキャラクターは「けいおん!」だったんですが、メカとか風景とか他にいいところがいっぱいあってよかったです。ほっこりするには調度良かったアニメだったと思えます。

ブラック★ロックシューターは、あんな暗い話をちゃんとハッピーに終わらせたのはよかったです。暗いまま終わったら鬱苦て不貞寝してました。

Anotherは放送されていなかったの、見ていなかったんですが、色んな評価を見ていると再放送を探したくなりました。

ミルキィホームズはなんにも変わらなかったの、主人公達がそろそろ本気を出すと面白いかなと次に期待です。ラードはラードでしたね!

「ちはやふる」はサザエさんタイムとか、深夜でやる理由がよくわからないくらい面白かったです。他人といっしょに見ても苦でないアニメは貴重でした。

他にも、妖狐×僕SS, アクエリオンEVOL, 戦姫絶唱のシンフォギア, 偽物語, モーレツ宇宙海賊, 輪廻のラグランジェ, 新テニスの王子様, etc.と終わってないまたは、最終回をまだみてないのがいくつかあるので、追って感想を書こうかと思えます。

余談ですが、これでも昔と比べてアニメは減ったそうなんだとか言われていますが、イマイチ信じられないです。まあ、こんなにアニメをつくってる国なんてそうそうないので、わかりやすく実感してみたいかもしれません。最近、背景とかを隣の国に外注しているそうなので、そういうところから対抗されるようになればと思えます。

リファラコントローラー

- <https://chrome.google.com/webstore/detail/fflbhcyjdhfppehpgehcodcacibgifae>

Google

ReaderとかでAmebloやFC2ブログの画像が表示できないことがよくあるんですが、この拡張入れたら直ってウハウハです。それと、twi2urlをちょっと再開していたりします。Herokuに移したいんですけど、色々めんどくさくなって結局自分のサーバに戻っているのは、やっぱり力不足を実感するような出来事です。

Subtree merge.

git submoduleは緊急避難としての手段としてはいいんですが、色々やりだすと不便で使いづらい微妙な機能です。なので、submoduleでは物足りなくなった時にどうすればいいのか？みたいなことをここ最近調べてたんですが、

- <http://stackoverflow.com/questions/1425892/how-do-you-merge-two-git-repositories>
で、知ったsubtree

mergeがよさそうです。分かれているリポジトリを特定のディレクトリに統合できる機能だとわたしは理解しています。そのまま、ファイルをコピーしてコミットするなんて手段も一応考えてはいたんですけど、統合元のリポジトリのコミットがちゃんと取り込まれるのはとてもいいです。

ただ、いくつか欠点があって、githubで変更履歴なんかを見る時、マージする時にファイルは移動するので、ファイルが移動すると移動元とゴニョゴニョ…（ノ。 ㇏。 (; ㇏）してくれないgithubでは、少し不便を強いられます。頑張れば見れないこともないですが、大変です。それと、マージ元のリポジトリのコミットが全部取り込まれるので、けっこうな数のコミットが加算されて、「うわっ」ってなるので心臓悪いと思います。貧乏人に大金を持たせてはならないのです。

最近みたアニメ

いくつか面白いのがあったので感想とか書いてみます。

まず、謎の彼女X。上級者向とか人を選ぶというか、言い方は色々あるんですが、面白くない人にはきつととても退屈なアニメだろうなと最初に思いました。ヒロインの声は本職の声優さんがやっているわけではないんですけど、慣れてしまえばとてもいいです。元の声がいいというか、逆に新鮮というか。個人的にストレートなストライクで、驚喜してしまいます。絵は、少し古臭いといえはいいのか、それほど萌えなかったんですけど、えっちいかったです。涎アニメなので涎の絵がいいので文句なんて無いんですけど。

次は、「這いよれ!

ニャル子さん」。ただの萌えるアニメかと思ったら、蓋を開けてみると、中身は形容詞がたいクトウルフ的な何かでした。Fate/Zeroと微妙にかぶっています。イカ娘ワクだと思ってみると色々浄化されていいです。

それから、CANAANです。

- <http://ch.nicovideo.jp/channel/canaan>

でひっそりと配信が始まっていました。うっかり見つけて、見だしたら内容にグイグイ引っ張られています。原作の428はやったことないので、とても楽しめるみたいです。そのうちやってみたいです、428。

っと、3つ取り上げてみました。他にもいっぱいあるんですけど、疲れたのでこのへんで

psplibrariesがマージされてた

けっこう前にリクエストしたPull Requestがマージされてました。

- <https://github.com/pspdev/psplibraries/pull/7>

色々やってたらめんどくさいことになって、もうマージされなくてもいいかなーとか思っていたのでよかったです。色んな修正が主なんですけど、angelscriptを追加したり、少し建設的なこともしてます。

ただ、rubyとboostはlibcのサポートが期待できないので、削除しました。boostはC++11がgcc-4.6.3でかなりサポートされているので、たぶんなんとかなります。

こういうのやってみて思うんですがやっぱりサポートって難しいです。別にホントにサポートしたわけではないんですが、エラーが出た場合うまく修正したりするのは色々必要で、自分にはあんまり向いていないと思ってしまう。場合によっては、他の人に「直して」リクエストを送ったりしないといけないのは、ちょっと苦痛でした。

ストライク

好きな絵師さんが挿し絵とカバーを書いているライトノベルを買ったら、そこそこよかったです。タイトルは「桜色の春をこえて」なのですが、たまたま行きつけの文房具屋さんで見つけて、絵師衝動買いでした。

内容は、女の子二人が同居してキャッキュウフフする話です。繰り返します。女の子がキャッキュウフフする話です。本筋は、それなのですが一応物語なので、わけありで、二人とも家庭環境でそれなりに問題を抱えていて、他人との関係に悩んでいる設定があります。

一人は優等生、もう一人は問題児だそうで、二人ともわけあって、両親と暮らしていません。物語は、優等生の娘が管理人さんたちのミスで家を出ないといけなくなって、たまたま隣に住んでいた「不良」の娘のところに居候することから始まります。

断っておくんですが、「不良」とよく形容される彼女ですが、損な役割でそうゆう役に当てはめられているだけで、けっこう普通です。ちょっとケバイとかそんなだけなようで、もう少し言葉を選んでもらった方が読む方としても安心できたかもしれません。「不良」という言葉けっこうな烙印なので。

悪い人が出てこないのは、不幸ぶっている彼女たちにはとてもいいのかなーと、ほっこりします。彼女たちの両親は見方によっては、善悪別れてしまうので数えないとすると、二人くらいしか悪人らしい悪人は出て来ませんでした。本当の意味で「不良」と転勤してしまった教師なのですが、両方とも出番は少ないんですが、波紋だけはとても大きかったです。そうゆう「影響」の描写はとても丁寧な気がします。

それから、いい大人がいっぱい出てくるのは、名作の証だと思います。実力の伴った口煩い教師、主人公達の母親とは対照的なご近所ママ、指導者として優秀であろう弓道部の顧問。みんなみんなかっこいいです。中でも「ガンジー」と呼ばれている柏崎先生は、作中で一番かっこいいので、最後までちゃんと見てあげるといいかと思います。

ただ、不良っ娘の趣味は作者の趣味っぽいので、もっと相応しい趣味を与えられなかったのか、悔やまれます。

node.jsで圧縮

node.jsでサーバ書いているんですけど、圧縮しないと性能的に満たされてなくなってきたので、どうにかできないかと調べたら、なんとかなりました。まず、zlibがnpmを使わずともnode.js組み込みで使えるようになっていたらしいです。で、そのドキュメントが

- <http://nodejs.org/api/zlib.html>

です。一通り使い方が書いてあるので、これに書いてある通りにやればだいたいうまくいきます。httpのクライアント・サーバ双方の使い方があるのはとても助かりました。

で、一つ成功すると次もしたくなるのが人間の欲というもので、node.jsでけっこう有名なモジュールrequestでも圧縮したデータを扱いたくなります。

- <https://github.com/mikeal/request>

いくつか調べてみたところ、これそのものをforkする必要はなかったです。（した後に気が付きましたorz）やることは、

- リクエストヘッダのAccept-Encodingに'gzip,deflate'をセットする。

- encodingを指定しない。（Bufferをコールバックに渡すようにする）

- 受け取った時にちゃんと展開する。

の3つです。ちょっとめんどくさいですが、ネットワークのオーバーヘッドが減ると思うと、大したものではないかと思えます。

それから、child_process.forkの通信でも圧縮したデータを転送したい場合があるんですけど、Bufferではやり取りができないみたいなので、base64でエンコードしてやりとりしてみました。base64でいいのか微妙なんですけど、他のencodingを見る限りこれが無難かと思って、容量が増えるかもしれないけど、base64にしています。よくわかりません....

ギルティクラウンのノベライズ(前編)

レクイエム・スコアの前半部分(2冊)を読み終わったので感想を書いておきます。二冊で1クール分の物語をカバーしているようです。

まず、小さなズレが多層に渡って重なった場合に生じるのがあのアニメかと苦笑いしました。ノベライズでそれなりに一貫性があるのを見ると、脚本家同士の認識の違いとか抑えるべきところを間違ったとか色々と想像してしまいます。例えるなら、食材に関して全く知らないベテランに高級食材を任せて、料理の出来が微妙だったといったところでしょうか。それぞれのコンビネーションが絶妙にマイッチングするというのは怖いですね。

アニメで語られなかった部分は主に

- ルーカサイト攻略の具体的な方法
- キャラの性格とか立ち位置
- その他の細かい設定

だと思えます。

一つ目は、物語としての山場なのに、すごい力ですごい台無しになっていたシーンとして個人的に印象的です。ネタバレ上等で結論を言うと、ダリルのヴォイドを使いました。ただし、「ヴォイドの融合」を使っているのりのヴォイドソードと融合、力の底上げでルーカサイトと張り合っています。たぶん、「ヴォイドの融合」は2クール目の中盤でガイがミサイルを撃ち落とすのに使ったアレです。誰かちゃんと解説して欲しかったシーンでした。

二つ目は、キャラクターがほとんど、人間として理解できないところに関するものです。アニメでは確かにクズだった登場人物がノベライズでは説明不足なところが少ないので、ある程度客観的にみてクズな人がちゃんと浮き彫りになりました。

三つ目は、匂わせるだけでもして欲しかったところです。また、重要な設定なのでより強調して欲しかったところもあったのでそれについても触れます。

- セガイさんがガッツ男に対してカウントしていたポイント
- スカイツリーが爆破された理由
- 四分儀の「流石です」(城戸研二がそれについてボヤク描写が個人的に思いついた)
- 葬儀社が腐ってもテロリストなところ
- いのりの歌とゲノム共鳴

DeflateRaw

- http://nodejs.org/api/all.html#all_zlib

で、DeflateとInflateにRawが付きのものがあるということがここ最近ずっと疑問だったんですけど、最近やっと使い道がわかりました。端的に言ってしまうとHTTPとかでデータをやりとりするときはRaw無しよりもこっちを使った方がいいみたいです。

きっかけは、

- <http://validator.w3.org/feed/>

というRSSフィードの妥当性を検査するサービスでした。Rawなしで送っていたらデータが壊れていると言われて、Chromeではちゃんと動いていたのにおかしいなーと、ドキュメントを見ていたらRaw付きならいいかもしれないと思って試したら見事うまくいきました。ドキュメントには、通常のdeflateではzlibヘッダを付加すると書いてあったのでたぶんそのデータが要らなかったのだと思います。

っとまあ、node.jsでサーバから圧縮したデータをdeflateして送るときは、DeflateRawの方がいいかもしれないといったお話でした。ドキュメントのサンプルだとRawは付いていないんですが、feed

validatorはwebの標準をまとめているところのものなので、たぶん、こっちの方が正しいと思います。たぶん。

bufferstream

Node.jsのBufferは便利なんですが、ストリームとして使えないのが玉に瑕だと思っていたんですが、わたくしごときがそう思っているなら、大体そういう問題は解決されているわけで、
- <https://github.com/dodo/node-bufferstream>
というのがありました。Read/Writeできるストリームらしくて、pipeもちゃんと使えるようです

ただ、使う時にNode.jsのストリームもまだちゃんと扱いきれていないせいか、pipeを設定した後にデータをend()で流さないといけないこと気がつくのに遅れて困りました。
で、まあ、ずっとBufferはpipeできると勘違いして書いていたコードがこれで解決しました。
めでたしめでたし

NEC無双

- <http://booklog.jp/item/1/4501324805>

「指紋認証技術」を読み終えました。第一印象は、NEC TUEEEEEEE!でしょうかね？たぶん、NECの方も執筆に参加しているのできっと自画自賛なんですけど、実績があるので自他共に認めるといった感じですね。USAが冷戦の影響で指紋にお金をかけられなかった事情とか、その他諸々あるんですけど、自力で世界中で使われて信頼性も高く国際標準化までできたのは、NECの実力だと思いました。

少し細かい話になると、指紋認識のためにハードウェアをわざわざ設計するのは、時代が違うな、と世代ギャップを感じます。今なら、よっぽど特殊な計算でも無い限り、ちょっとした並列計算のライブラリとかを使えば、事足りるのでしょうか、処理能力が足りなかった時代は、ハードウェアを設計するコスト < 計算能力だったのでしょうかね。

アルゴリズムとか、画像認識とかの話はちょっと触れる程度で、正直あんまりわからなかったです。実用化の歴史とかを読むだけなら十分なんですけど、数式の解説とかを入れると、数倍の厚さになると思うので導入としてはいいのかと思いました。機会があったら参考文献にあった論文を漁ってみようかとは思っています。それから、指紋認証開発キット(SDK)もあるらしいので、こっちもまた機会があったらと。

量子化学

「ものが壊れるわけ」という本を読み終えました。

- <http://booklog.jp/item/1/4309251846>

この本をタイトルのおりに読んでも、どうして壊れるのか？、というのは全くわからないんですけど、量子化学の本として読んだ場合けっこう面白いなと思います。タイトル詐欺かもしれないですけど、十分面白いのは評価できますね。

はじめの方にある歴史についての部分はとてもよく調べてあって、日本人なのに日本刀についてよく知らないなと痛感させられたり、人類の進歩について、よくわかりました。それから、繰り返し出てくるタイタニック号沈没の原因についても、本来ならしつこいと思ってしまうようなところを、視点を何度も変えてくれるおかげで、繰り返し見返すことができるのは、著者のセンスを感じます。

「衝撃的、ただただ衝撃的」では、著者がポンプ会社の弁護士に協力して、ポンプ会社の無罪を勝ち取った話なんですけど、潔白の証明というのがどれほど、めんどろなことが想像するのを放棄するくらいにはわかりました。幸いこの時は、相手側が完全に間違っていただけで、証明ができたんですが、とりあえず金がないので訴えてみたっていうのを相手する場合ってこうも疲れるんだなと、やったことも無いのにパスが発動するのはまったくこわいことです。

8章で、プルトニウムを使った原子力電池について、けっこう詳しく解説していたのは、とてもよかったです。宇宙関係でよくでてくる電池なんですけど、宇宙で使うのは当然らしくて詳しい解説はあんまりないので、これを機に多少マシな理解を得られました。要は、プルトニウムの核崩壊で生じるエネルギーを電力に変えて、この場合小型で長持ちな電池を実現できるということでしょう。で、本文では「そんな安全対策で大丈夫か？」と装備を聞いてきそうなるノリで検証するわけですが、うまく咀嚼しきれない部分なので結論は著者が見る限り、とある菌の嫌気呼吸によって生じる毒が原因の食中毒を気をつけた方がいいそうです。危険だと認識していると、危険だと認識されていないものよりも安全にできるいい例なのではないでしょうか？

それから、最後の方は著者自身の研究に☒いてで、量子化学に憧れている者としてはとても参考になりました。たぶん、きょーみない人が読んでもあんまり面白くないかもです。ただ、計算○○学に携わる人を二分する方法とか、ちょろちょろ面白い話はいくつかあります。

ヒイイイイイイイ

- <http://iwadon.tumblr.com/post/23163062534/github-growler-public-notes-textt>

なんか見覚えのある文字列だなー，と思ったらわたしが書いた記事でした。びっくりです。

ちょっとした補足なんですけど，Github

Growlerを日単位で（例えば，3日動かさっぱなしとか）起動したままにすると，重くなってきてしまいには，CPU利用率を圧迫し始めます。アクセスビリティモニタで，perlが重たいなーと

思っていたら，それはGithub

Growlerのもので，そういえばずっと起動したままなのを思い出し，再起動したら元の善良なプロセスに戻りました。たぶん，何かのリソースリークが起きているのだと推測されますが，例のごとくPerlはよく知りませんので，再起動して対策しています。

っと，まあ，たまにはGithub Growlerを再起動してあげてくださいというお話でした。

OAuthと時計

OAuthがうまくいかないと思ったら、
- <https://dev.twitter.com/discussions/1043>
に書かれている問題に遭遇したみたいです。自分の環境の時計が狂っていた場合、OAuthが使えなくなる症状が報告されています。OAuthでは、UNIX時間を使っているのが原因かもしれませんね。

プログラミングF#

友人に借りてたのを読み終えました。

- <http://booklog.jp/item/1/4873114667>

F#のことを一通り網羅していて、とてもよかったです。ただ、Windows環境を持っていないせいかあんまり読むモチベーションがうまくコントロールできなかったのはあんまり関係ない話です。

F#は関数型プログラミング言語と聞いていたんですが、あんまりそんな感じがしなかったというか、Haskellの記号ばかりでよくわからない(´・ω・`)っていう印象が強かったせいか、関数型プログラミングもそんなに難しくないなーと思えたのは一歩前進かなと思っています。F#自体、mutableなデータ構造とか、冗長だけど確実な書き方とか、そういうのがけっこう許容されているらしいので、優雅ではないけどそれなりに書けそうな感触をつかめたのはとてもいいです。

余談ですが、純粋関数型言語というのは副作用がないという記述をしばしばみかけることが多いですけど、実態は「副作用がよく管理されている」という風に映ったのは間違っていないと思っているのに自信がない実感です。でも、こういったことはプログラミングそのものとあんまり関係ない気がするので、追いつ追いつ確認した方が確実かもしれないので、ここはひとつ純粋関数型言語を使い始めてみるしかないのかもしれないです。

文法で少し思ったのは、ずっとカッコ(括弧)つける言語を触ってきたせいか、スペースだけで区切って関数を呼ぶのがとても綱渡りのように感じます。使いだせば「括弧とか邪魔」なんて言い出しそうなんですけど、どうも経験の差で気持ち悪いです。それから、関数型言語は色んな見慣れない記号がいっぱいでホントに別言語だなと思います。「君の知らない概念」というのがいっぱいなのでしょう。

F#は.NET系の言語なので、共通言語基盤(以下、CLI)を使っているわけですが、それについての記述は主人公がF#なので当然抑え目で、ちょっと欲求不満というか、もっとやって!と思いました。ただ、これはC#をちゃんとやった方が理解は速い気がします。こっちの方が、.NETでは主役なわけですから。

以上、読了後の感想でした。たぶん、開発環境を立てるのが大変なので友人が遊びだしたとか以外では、Haskellの方が使うのが楽なのでHaskellをメインに勉強を続けます。F#について読書できたのは、一つの言語だけの狭い視野にとらわれないいい経験になると思います。

ネットワーク通信技術入門

- <http://booklog.jp/item/1/4798111619>

図書館で見つけて、まあよさそうだったので借りてみました。まあ、よかったです。星にすると、★★★☆☆といったところでしょうか。

内容は広く浅くといった感じで、あんまり深く掘り下げないので、資格取るのにはよさそうだなーと思いました。ポアソン分布が出てきたりするんですけど、精々そこまででエラーの確率とか「そういうこまけーことはいんだよ」感じはあります。そういう細かいことを解説しだすと、本が一冊くらい出来上がるんですけど。

囲碁

昔、ヒカルの碁ブームで囲碁を覚えたクチなんですけど、気の長いおじいちゃんがするようなゲームなので、たまーには飽きての繰り返しをよくしています。で、最近、たまーにする周期が巡ってきてiOSアプリのSmart Goというのを買ってやっています。

250円の一番安いのを買ったんですが、19路盤が付いてなかったのは少し残念です。13路盤で基礎を固めるのも悪くないですけど。

とりあえず、Undoありでなんとか、白番を勝てるようになって来ました。最初は凡ミスとかで、置き石があっても負けることが多かったんですが、少しは考えるようになってくると、それなりに勝てるようになってきて楽しいです。特に、圧倒的な中押し勝ちで終わることができると、とても気分がいいです。

っと、まあ、空き時間を見つけながらどんどん打てたらいいと思います。目標は、賭け事よりも楽しいと思えるくらいです。

rss_twi2urlをHerokuに移した.

わたしのTwitterアカウントのTL,リスト,保存した検索から抽出したURLをrssに変換してる

- https://github.com/take-cheeze/rss_twi2url

を

- <http://floating-dusk-8988.herokuapp.com/>

に,移しました.前,失敗したんですが,いくつか解決方法を思いついてなんとか移行にこじつけることができました.とりあえず,自分の環境のCPU利用率を圧迫しないで済むようになって快適です.

因みに,OAuth認証していない場合,上のリンクへアクセスするとOAuth画面へリダイレクトするようにしたので,コンソールからURLをコピーする手間が減りました. Express (フレームワーク)だと302を送るのがすごい楽なんですけど,そういうのを使っていないとめんどくさいですね

そういえば,どのバージョンのnode.jsとnpmがHerokuで使えるのか調べたんですが,クリティカルなページがあって助かりました.

- <http://d.hatena.ne.jp/sasaplus1/20120503/1335980529>

以下のマニフェストに書いてあるバージョンをpackage.jsonのenginesで指定すれば,ちゃんと動きます.

- <http://heroku-buildpack-nodejs.s3.amazonaws.com/manifest.npm>

- <http://heroku-buildpack-nodejs.s3.amazonaws.com/manifest.nodejs>

node.jsそのものでのzlibサポートやchild_process.forkが新しい機能らしく,デフォルトのバージョンだと古くて使えないのは困ります.

それから,Herokuでの標準エラーとか標準出力を監視したい場合は,

- `heroku logs -t`

と打てばwatchしなくてもいいようです.

銀河へキックオフ！！

NHKで一挙放送をしていたので観ました。すごいよかったです。
わたしはもっと暗い話の方がずっと好きなんですけど、それを差し置いても素晴らしかったです

最初は、タイトルとかからイナズマイレブンを想像していたんですが、そんなことなく、「イナズマってなんですか？それおいしいんですか？」と次第に思うようになっていました。それっぽいダサイネーミングがところどころあるのは事実ですが、それでも子どもっぽいで済まされる程度でした。なんというか、ダサイなんて言った方がダサイ空気というのがあるんですね。

主人公は、立場上の補正がいくつかあるんですが、違和感があまりないので、うまく設定したなーと思いました。人望、視力、声量、平均的なフットボール力は劣るが主な特徴なんですが、バランスがいいです。順当に育てば、コーチから得点力を引いて、チームの要になれる感じなんでしょうね。声が大きいらしいんですが、役者さんが普段から既にけっこうな声を出している気がするので、あんまり声に威圧にされませんでした。

それから、ここまで主人公とヒロインが高めあっているのは久しぶりに見ました。最近のトレンドというか、わたしの嗜好というか、どちらかが圧倒している作品のほうが多い気がします。それか、ヤンデレという性能が高いのに精神的な脆さを露呈させる関係もあります。ともかく、主人公とヒロインがここまで対等だとなんだかとても眩しいのです。

ヒロインの娘の声も、とてもよかったです。関西弁（正確には大阪弁？）についてあんまりわからないんですが、他のアニメよりもアクセントが自然で、声もとても元気なのはとても気持ちよく感じられます。

で、まあ、今回は1～8話だけだったんですが、思ったよりも引きこまれてしまったので、9話からの視聴を開始しようかと思えます。

Chrome Secure Shell

ちょっと前に、Chromeではsshが使えると伝え聞いていたんですが、iTerm 2をわざわざ起動するのが最近めんどくさくなってきたので、思い切ってターミナルエミュレータもChromeで済ましてみることにしました。とりあえず、
- <https://chrome.google.com/webstore/detail/pnhechapfaindjhompbnflclldabbghjo>の通常版を使ってみたんですが、公開鍵が使えるだったので、入れたままにして開発版の方を試しました。

- <https://groups.google.com/a/chromium.org/forum/?fromgroups#!topic/chromium-hterm/4P21C89X6w4>

- <https://chrome.google.com/webstore/detail/okddffdblfhnmhodogpojmfkjmhinfp>にいけば手に入るんですけど、その前にchromium-htermグループに登録しないとイケません。そのせいで、入手できなくてそれっぽい記述を見つけてやっと気づきました。Google GroupでChrome拡張へのアクセス権を操作できるんですね。で、肝心の公開鍵なんですけど

-
<http://git.chromium.org/gitweb/?p=chromiumos/platform/assets.git;a=blob;f=chromeapps/hterm/doc/faq.txt>
に書いてあるように

- `term._command.showFileImporter();`

とjsコンソールで打ち込み、ボタンを表示させそれをクリックして鍵ファイルを開きます。Promptというアプリでも経験したんですけど、こういう場合は秘密鍵を渡すようです。

で、さっき上のテキストファイルを見て気がついたんですけど、けっこう色んな設定ができました。拡大率を制御することで小さくしてたフォントは、

- `term._prefs._set('font-size', size)`

で背景色と文字の色は

- `term._prefs._set('background-color', 'white')`

- `term._prefs._set('foreground-color', 'black')`

で変更できます。わたしは、黒背景だとよく見えないので、これができるのは助かります。それとカーソルの設定も色々できるらしいので、興味のある人は目的の設定方法を探すといいです。バックスペースの設定は、けっこう大事かもしれません。わたしは、Emacs側の設定をしているのでそんなに気になりませんがね。

何でこんなにjsコンソールを使わないといけないのかというと、「β版だから我慢して>

<」だそうです。その内に設定ページが作られるのかもしれませんが。

余談なんですけど、このアプリはhtermにsshをちゃんと使えるようにしたもののらしく労力はそれほどかかかっていないと思うんですけど、「NaClでここまでできるよー」を示すにはとてもいいのかもしれないね。htermというのはhtml-

terminalの略でjsで実装されたxtermクローンらしいです。ただ、hterm単体では、sshが使えないのでこのアプリではそこところをNaClで克服していました。

以上、iTerm 2は犠牲になったのだ、という話でした。

昨日の補足.

Chrome Secure

Shellの上級者向けのTipsとしてちゃんと出てくるんですが、ログインした後、ブックマークすればセッションを保存できるとか書いてあるんですが、現状、ホストとユーザ名くらいが限界みたいですね。まあ、そういう時のためにscreenとかtmuxとかを使うわけですけどね。

それから、term_prefs_で行う設定は、localStorageに保存されるので消える心配はないようです。いざとなかったら、localStorageをクリアすれば設定も綺麗になるというのもありがたいかもしれません。

それから、フォントの設定もChrome自身の最小フォントとかをいじりながら試した結果、わたしには6ポイントはどうしても使えないことがわかりました。一応、文字がそれっぽくわかりはするんですけど、ちょっと苦痛です。なので、今は7ポイントで使っています。画面に入る文字の量を考えるともっと小さくしたいのですが、こればかりは視力の限界ですね。行間をもうちょっと詰められたらいいのかもしれませんがね。

フロート系のアイス

フロートというのは、ジュース（炭酸とか）にアイスを浮かべたものなのですが、最近、これの名を冠したアイスの良さに気が付きました。わたしは、口の中が油とかでべったりするのがそれほど好きじゃないのでシャーベットの方が好きなんですけど、シャーベットを食べているとどうしても口の中が冷えやすいわけで、フロート系のアイスにはその緩和剤として、シャーベットよりも少ない量のアイスクリームが入っていて、口の中が快適なまま食べ終わることができるのはとてもいいです。

mruby

ARGSS (RGSSのオープンソース実装) をmrubyに移植しているんですが、これがまたちょっとめんどくさいです。例えば、rubyだと要らなかった第一引数に必ず必要になるVMインスタンス。これを全部直していくのはとてもめんどろな作業になります。sedさんに大活躍してもらったりしているんですけど、それでも、うまく置換できないのがあったりと、まあ、大幅な書き直しが必要なようです。

mrubyのAPIを触ってて思うんですけど、rubyよりもAPIが綺麗です。ちょっと驚きました。

それから、ちょっと困ったら

- <http://mattn.kaoriya.net/>

あたりを検索するといいいみたいです。公開された時にけっこうな活躍をされた方のブログなので、それっぽいコード片を探すならここです。

Another (上)

Anotherのアニメの話ちょくちょく聞いていたんですが、アニメを今から見るのはしんどいので少しためらっていましたが、その後原作の方は、けっこうしっかりとした話だと聞いたので文庫をちょうど見つけて買って昨日読み終わりました。

最初の方は、「へー。ホラーなんだ」みたいな気分で読んでいたんですけど、人が死んでくるとどんどん止まらなくなって、すぐに上巻の終わりにぶち当たってしまいました。「Anotherなら死んでいた」という言葉の意味がやっとわかってきました。ただ、人数としてはまだまだみたいなのでこれからその言葉の意味をより理解してくと思うととても楽しみです。

アニメでは赤沢さんが大活躍だそうですが、原作だとときとーに描写されて死んじゃうそうなのでちょっと残念です。幸い上巻ではまだあかざーさんは死んでいないので、まだ楽しみは残っています。

戦争

2chでの戦争に一つ区切りが付いたらしく、

- <http://2ch.net/warn.txt>

という結果に収束しました。悪質な2chまとめブログにノーが突き付けられたみたいですが、流石にメディアとしても品性を疑うような内容だったり、記事内容の傾向だと赦されないのですね。

それから、ここに至るまでの経緯の一部をまとめた記事も出回っていて面白かったです。

- <http://dic.nicovideo.jp/a/%E7%AC%AC%E4%B8%80%E6%AC%A1%E3%83%96%E3%83%AD%E3%82%B0%E9%80%A3%E6%88%A6%E4%BA%89>

Amazonは自分のマシンではそれほど使わないので、Cookieの話は関係なくて少し安心しました

今回の騒動の渦中にあるアニプレックスという会社なんですが、わたしも見ていて気に入っていたアニメにもいくつか関わっているようなので、その会社がしてきたことにとっても失望しました。アニメは素晴らしいんですけど、赦されないことってありますし、何よりもその素晴らしいアニメを貶めたのはもっと赦せません。

まあ、それはさておき。最後通牒を突き付けられたそれらのサイトなんですけど、その後のことは悪党の一味がどうなったかみたいで、けっこう面白いです。多くは、他の場所から匿名コメントを集めたりするんですが、それが大体規約違反だったり、墓穴をせっせと掘っているのが滑稽だったり、自分でコメントを書いてみたら、ひどすぎてどう接していいのかわからなかったりと散々でした。

ただ、一つ例外があって、

- <http://blog.livedoor.jp/insidears/>

という前まとめブログは、普通のニュースサイトになっていました。どれくらい回転したんでしょうね？

それと「ハム速、ついに自分のコメント欄をまとめ始める」というのも普通のブログですね。

- <http://logsoku.com/thread/hayabusa3.2ch.net/news/1338886979/>

なんていうか、コミケでダミーサークルで場所取りをしていたら同人にはまった、みたいな美談に類似した現象ですね。

余談ですが、これが公開された後、warn.txtを自分のドメインにも置くのが流行っていました。そのテキストファイルの中身は、大方Twitterレベルだったので、突発的な流行で終わりました。見たのはその日だけだと思います。

Growlが立ち上がらない時の対処方法その一

なんらかの影響で通知履歴が壊れている時があってGrowlが立ち上がらない時があるんですが、
そういう時には

```
- rm ~/Library/Application\ Support/Growl/notifications.history
```

をすれば何事もなかったかのように立ち上がるようになります。因みに、どうやってわかるかという、OS

Xのユーティリティからコンソールを立ち上げて、ログを検索して履歴が開けませんみたいなエラーメッセージがあればコレで対処できます。

OS Xのコンソールってターミナルも使えるので意外に便利ですね。

Herokuで障害

- <https://status.heroku.com/incidents/372>

にある通り、障害があってしばらくtwi2urlが使えませんでした。HTTPルーティングのエラーだとか（よくわからない（´・ω・`））

- <https://status.heroku.com/>

を見る限りこれほど大きな障害は珍しいようですね。でも、一時間ちょっとで対処できたのは、尊敬しちゃいます。とりあえず、直ってよかったよかった。

PDFをKindleで読む.

ちょっとタイトル詐欺なんですけど、KindleでPDFをそのまま読むのはつらいです。なので、別の形式に変換することになります。

で、変換形式にはいくつか候補があって

- mobi

- azw

があります。「epubの方が主流じゃないの？」とわたしは思うんですが、どうゆうわけかepubはKindleで直接読めません。

mobiは正直に言うてしまうとよくわかりません。とりあえず、Kindleが対応してるので使っています。運が良ければこの形式で配布されていることもあります。

- <http://calibre-ebook.com/>

のcalibreで変換できるのですが、pdfからの変換は正直微妙なのでおすすめできません。レイアウトが崩れるとか、けっこう散々です。たぶん、epubからならそれほどひどいことにはならないと思います。因みに、calibre自体はけっこうよくできたソフトなのでオススメです。

azwについては

- <http://d.hatena.ne.jp/otafone/20110627/1309170293>

- <http://d.hatena.ne.jp/Hoshiba/20100906/p1>

- <http://hondamarlbboro.blog112.fc2.com/blog-entry-107.html>

の記事を読むと大体わかるかと思います。メールに添付して件名を"convert"にして自分のKindleアカウントに送れば、綺麗に変換されたazw形式の文書がKindleにダウンロードされます。ただ、三つめの記事にあるように、宛先のホストは、"kindle.com"ではなく"free.kindle.com"にした方がいいみたいです。Wi-

Fiモデルの人はそんなに関係ないんですが、3Gモデルを使用している場合、手数料が発生するので念の為にわたしはアドレス帳の登録を変えました。

図書館戦争

中学生の時に原作の方は全部読んだんですが、アニメの一挙放送があったので観ました。最初から最後まで安定した出来で、最後の良化隊との戦闘は圧巻でした。さすがProduction I.G.といった感じで、観終わった後には腹丁度八分で続きの映画を観たくなりました。ちょっと惜しいなーと思ったのは、手塚兄の扱いでした。声優さんの声の印象もあるんですが、見せ所が全くないのはやっぱり惜しいです。テレビが輪番制で検閲の対象になるような報道をする話では大活躍なんですが、それがどこにあるのかを忘れたので、まあ、実際はどうでもいいかもしれません。メインキャラの出番が少ない話なので、人によってはあんまり楽しくないです。人から嫌われる実力者としては、けっこう好きな人なのでガンバレと応援だけしてます。それから別冊の展開を多少知っているとか、けっこうニヤニヤできますね。小牧さんの恋人とか、手塚と柴崎とか、元良化隊の副隊長とか。外伝は二冊までしか読んでなかったんですが、確認したら二冊だけだったみたいです。とりあえず、すべて読破していたのかと安心しました。図書館戦争は図書館にありそうなので、今読んでる本が終わったら、借りようかと思います。

ドラマ版未来日記が少し化けたかもしれない

未来日記の原作はちょっとWikiを読んだとかその程度で、アニメも一応一通りみただけなんですけど、個人的に「まあまあ」という評価を持っていたんですが、ドラマ版はそれのさらに「まあまあ」な評価でしたが、最近少しよくなったかなーと思い始めました。ドラマ版の世界で神様とかそういうのは実際にいなくて、すべて仮想世界の出来事であるというのは、思いの外活かしやすい設定なのかもしれないと、感じています。

原作で起こるゲーム(という名のsurvival game)はデウスの後継を選ぶものなんですが、やっぱり人間としては神様に打ち勝つ方がずっと楽しいとわたしは思っています。なので、一步踏み間違えれば、世界が終わる原作ではその展開は非常に限定的な形になり、ちゃんと後を継ぐ方が無難なんですけど、仮想空間の出来事なら割と神様殺して無秩序になってもどうとでもなるのはとても魅力的です。それから、「仮想空間で救われてもそれは仮想空間でのことで現実ではなにもできないだろ？」みたいなことを言われても仮想空間でがんばった分現実世界でもうまくやれるような展開を期待できるのはとてもワクワクします。

ただ、いくつか「解せぬ」というところもあって、やっぱり「未来日記の特性」というのが全く活かされていないのと、主人公の未来日記弱すぎるのはちょっと受け付けません。後者は、デウスがなぜ記憶を主人公の記憶を消さなかったのか？という疑問の答え次第では覆るのですが(場合によっては未来日記なしで主人公最強か主人公が最強の未来日記)、前者は原作のよさをぶっ潰した感じで無念です。未来日記の特性の比重が原作で低かった場合、あれほど続きません。それから、物語の根幹に関わるループ設定なんですけど、これももう少し活かすことができたかもしれないとifを考えてしまいます。アリガチではあるんですが、前回のループの記憶を使うともっと語りやすいですし、展開も垂れません。「デウスが誰なのか？」という命題についてももっとヒントを散りばめられるかもしれないです。

そして、もう一つ苦言を呈すなら、衣装というか全員の服が全体的にダサイです。イケメンの本郷奏多くんは、その割を全くくっっていないんですが、てんばーな主人公とか、ダサイコートのケージさんとか、どうしてこうなった、としか言えないです。夢をください。

Another (下)

この前、本屋さんに行ったので、参考書とかのついでに買いました。ちょっと不貞寝したいけど寝る気分じゃなかったのでも読み始めて、気がついたら読み終わっていました。賞味数時間だったんですが、ここまで引き込まれると、それなりに設定を知ってたとか抜きに本そのものの面白さを再認識させられます。

それはともかく、続きが読みたくなりました。著者の実力は疑いの余地が無いので、ターミネータ2くらいの出来を期待してしまいます。ただ、ホラーは鮮度が大事なのでもう一冊くらいが腹八分目だと思うので、それ以降はあんまりいらぬかもしれないです。

さて、内容ですが、前回「現象」をととても強く印象づけられたところから、見事に「現象」に打ち勝つまでの過程が丁寧でした。千曳さんも大活躍でしたし。ただ、いくつか「うーん」と思うようなところもあって、沼田妻の凶行がちょっと御都合主義かなと思いました。「現象」自体は、人が死にやすくなるように偏るだけなので、その兆候が全くなかったはずの沼田妻が凶行に及ぶには呪いの部類が必要です。もう少しそれっぽい理屈をでっち上げてもらった方が自然かなと思います。それから、千曳さんのメモが現象から見過ごされている理由とか、主人公の祖父の戯言とか、ヒロインの邪気眼とかも、わたし気になります。次回作の伏線として使えるので、まあいいですけど。

そういえば、漫画版も面白いし、あかざーさんが生き残ると聞いたので、ついでに買いました。それは小説よりも先に読んだんですが、評判通りでとてもよかったです。絵柄も最先端っぽくてとても気に入りました。水野さんもとてもいきいきしていたのも、とてもよかったです。はやく続きが読みたくなります。

Spidermanが世知辛い.

夜にSpider-Man

2を放送していたので録画して、さっき見たんですが、こんなにみるとヘビーな映画だっけ？とちょっとみて後悔しています。

まず、借金に追われているヒーローというのは、なんか嫌です。アメリカだと借金しているというのはそう珍しい状況じゃないと思うんですが、日本だと、カイジとかそっち方面でしか思いつきません。それか不幸体質か損な役回りとかもあるんですが、ヒーローでそれはないです。金融大国とは価値観が違うのかもしれませんが。

ただ、「時間がない」というのは、けっこう普遍的な悩みだったので、よかったです。よても共感できます。時間があれば、なんとかしてしまう主人公はイケメンですけど。

で、今回の敵なんですが、ロボットアームとか頭脳とか、補正はあるんですが、ただのお腹がでたおっちゃんでした。大活躍するんですが、こいつホントに科学者かよ？、というところがいくつかあって、発生する磁界の強さとか、制御チップの強度とか、実験施設の安全性とか、 $\text{IIIIIIIIIIII}(\text{°}\text{∇}\text{°})\text{IIIIIIIIIIIIIIIIII}$ という突っ込みどころがいくらなんです大杉です。それから、核融合炉もあんなに持続させるのは流石に無理なのではないかと思います。水素爆弾みたいに一瞬で、ニューヨークを消し炭にするか、プスプス不完全燃焼に終わるくらいしか想像できません。あと、トリチウムは原子炉襲ったほうが絶対速いです。

アクションについては申し分ないというか、わたしがそんなに興味を抱くことでもないので、程々によかったです。最後の方の電車を止める場所は、スパイダーマンが抑える必要なくね？、とか身も蓋もないことを思いましたけど、熱かったのででーでもいいかもしれません。それから、見ているとあんまり気が付かないんですが、スパイダーマンはニューヨークで一番映えるなと思いました。日本だとあんなに高い建物いっぱいありませんし。

最後にヒロインについてなんですが、物語を邪魔しないこと以外はあんまり見処がないなーと、ヒロイン力の高さに驚きました。あんまり可愛くないと思っていんたんですが、あんまり可愛いとヒーローが霞み易いですし動かしづらいですもんね。主人公以外の人と結婚しようとしたのはちょっと愛の試練としては大変ですね。花嫁脱走は絵になりますけど。

っと、まあ、そんな感じです。こうゆうのを真剣にみると世知辛いですね。寝る前にはもっと頭が空っぽでも観られる作品のほうが良いと改めてわかりました。

Tumblium

つい最近, Tumblr

Lifeが復活したばかりなのですが, たまたまTumbliumという代替アプリに出会って使ってみたら, Tumblr Lifeよりもよかったので, 鞍替えしました.

- <http://objc2.tumblr.com/>

よかった点を上げると,

- おそらくAPIをOAuthで使っているので, 10 per requestなTumblr Lifeに対し, 20 per requestで効率がいい.

- 重複したポスト (たぶんポスト元が同じ) や自分のポストをスキップできる.

- Likeの状態がちゃんとわかる.

で, ちょっと気に入らなかった点は

- 広告あり (一応, 課金して消せるので, なら払えよですね. そのうち払って消すかも)

-

Safariで開くがちょっと手間. メニューから呼び出さないといけないし, ロードが終わるまでできない.

-

画像のロードがもたつく. (ぼやけた状態で読み込まれたり, まだ読み込まれないなんてことが多い)

- Instapaperみたいに, Safariで直接リンクを開けない.

くらいですかね. Tumblr Life互換というわけでもないのに, Tumblr

Lifeがじっくりこなかった時に使ってみるといいかもしれません.

週末に読んだ本

あんまり書くことがないので読んだ本の感想でも書いておきます。

一冊目は、プルトニウムに関する本なんですけど、微妙でした。途中まで読んでやめまじや。訳本なんですけど、文章も読めませんし、意味も本当に大学につとめている人が書いているのか疑問になるくらいわかりづらかったです。もとは、原子力電池について調べたかったのですがどうもお門違いな本だったみたいでした。

二冊目は、「プラネタリウムを作りました。」という本です。プラネタリウムをつくった人の本なんですけど、 Σ (・□・;) な規模のプラネタリウムをポンポンつくっていく様は、驚愕のあまり外れた顎が元に戻りません。最初は子どもの工作だったものが超高校生級の作品になり、前人未到の地に至り、終いにはプラネタリウムを革新してしまいます。蒸気機関で有名なワットを現代で目撃している気分になりました。この人みたいな弛まぬ努力をしないと、と焦ります。

原子力事故

それほど他人事でもないと思いたいんですけど、まあ他人事なので原子力関係の本を流行に乗って読んでいるんですが、それについて最近ちょっと不満があります。読む側が流行に乗っているのなら当然、書く側も流行に乗っているわけで、けっこうな駄作も量産されているわけです。で、今後出てくる調査結果次第ではゴミになったり、既にゴミになっている本は既にいっぱいあるので、福島第一原発事故があったから書きだしたような本は読む気が起きません。

それで、そういう本をうまくフィルターするような選定基準が必要なんですけど、とりあえず考えてみたものが、「ねーよ」なものでした。「原子力への関心が今まで薄かったから需要がなかったから出版できなかったけど、事故による不幸中の幸いで出版できる見通しが付いたような本」なんていうものです。普通ないです。

Amazonで調べたんですけど、そんなのはパッと見で見つかりませんでした。図書館でもあたってみたんですけど、ビジネスマンとか隠居しているおじさんとかが好みそうな本ばかりで、期待できません。

それから、宇宙関係の番組で原子力電池について知っていたんですけど、事故を機にそれについての本が出版されていないか確認したんですけど、ありません。原子力電池というのはプルトニウムが核崩壊する時に生じる熱を電気に変換する代物らしいんです。無人探査機に搭載して何年も使える寿命の長い電池はそれはそれで魅力的だと思うのですが、どうも世の中ではそうでもないようでして、残念に思います。プルトニウム自体そう簡単に手に入りませんし、扱いが難しいらしいので、そういう知識を手にしてもあんまり役に立たない気がします。

GMailロック

KindleにPDFを転送していたら、アカウントがロックされました。主にCVPRの論文だったんですがいかんせん量が多かったがまずかったみたいです。Google側の判断基準はよくわかりませんが、どうもファイル共有・ストレージに使っていると疑われている気がします。短くて一時間、長くて一日だけ凍結されると説明には書いてあるので、まあ、気長に待ちます。ロックされたのはあくまでwebのフロントエンドらしくMailアプリからは普通に使えます。なのでそれほど不便でもないですが、webインターフェースのほうが都合がいいことも多いのではやく終わらないかなーと思っています。

凍結解除

お昼頃に確認したら、gmailが普通に使えるようになってました。とりあえず、一安心です。流石に、GBくらいファイルをやりとりするとgmailも怒るようです。そもそも添付ファイルの限度が25MBなので大容量向けのサービスを使うべきでした。といっても、Amazon側はメールでしか受け付けてくれないのでどうしようもないかもしれません。

ちょっどこわかったんですが、いつも送るくらいの量を復帰を確認したあとにKindleに送ってみましたが、問題はありませんでした。100MBもいかない量なので当然といえば当然なんですが、安心しました。

イギリスはおいしい

友人に「イギリスはおいしい」（林望著）を借りたので読んでいるんですが、まずいまずいと散々言われているという個人的な印象を撤回しないといけないようです。ビネガー入りポテトはとてもおいしそうですし、ビネガー味のポテトチップについて「わたし、気になります」。それから、薄い食パンも、今までずっと食パンの厚さについて不満を抱いていた身としては、とても正しいと思います。パンはご飯の代わりになりません。それぞれの良さがあるので、ローカライズしなくてもいいです。

百合男子 2巻

発売されたと聞いて買って来ました。相変わらず変な方向に暴走していてニヤニヤです。ただ、淫Qβはダメですよ。いくら、百合男子が百合カップルと利害が一致しても決して相容れないからといって、淫Qβのぬいぐるみは暗示としてはおそろしくなります。そういえば、主人公の師匠ともいべき人も出てきました。魚屋さんらしいです。冷蔵庫を百合媒体を隠す本棚に改造するなんて全く感心します。あの世界では日本の北端・南端、果ては海外から百合漫画を買うためだけに関東に来る人がいたり、その情熱に敬服させられます。それはそうと、その師匠は主人公のちょっとした行動でその百合百合ストとしての才能を見抜き百合漫画を与えていたらしいです。わたしもそんな百合百合ストの英才教育を受けたかったです。それから、「玲緒っぽいらじお」で百合男子がとりあげられていたんですが、本の裏側のカバーに「その花びらにくちづけを」のDVDパッケージがありました。オメデタ—————(V)。。—————！！です。

Kindleの読み上げ

Amazonの形式にPDFを変換しているくせに忘れていたんですが、mobiとazwはテキスト読み上げ(所謂 Text to Speech)に対応しています。それを思い出したように使いはじめてみたんですが、手動でページを捲るよりもずっと楽でした。

日本語のPDFや画像だけのPDFでは、当然使えないんですけど、英語なら読み上げがかなり優秀なので腕がもう何本か欲しいなんて思わずに済みます。それから、けっこう長いPDFでやるとつらいので、そういう場合は手動でガンガン進めていった方が捗るようです。

あんまり関係ないんですが、Kindleの再生速度 (fasterで) で読み上げようとしたら、見事に何もできませんでした。たぶん、人間のはっきり話せる速度以上だと思います。こういうことは機械に敵いませんね。

js書いてた。

ちょっとした息抜きに書いていたんですが思いの外楽しかったです。Typed Arrayを初めてちゃんと使ってみたんですが、DataViewが思いの外、万能だったり、予想外のことがいくつかありました。

因みに参考にしたのは、Khronosのドキュメントです。

- <http://www.khronos.org/registry/typedarray/specs/latest/>

基本的に、ArrayBufferをストリームとして扱えるようなインターフェースを書いたんですが、読み込みだけならかなりいいところいくなーと思いました。ただ、書き込みする場合は、他の言語で提供されるようなストリームは期待できないと感じました。それっぽく実装はできるかもしれないんですが、どこかで見た上限に達するたびにバッファ用のArrayBufferを二倍に伸長するとか、まあ、どちらにしろめんどくさいです。

それはそうと、

- <https://code.google.com/p/yume-hikki/>

- <https://github.com/rikkimongoose/yumehikki>

で夢日記のクローンをつくらってる人がいるのを知りました。今回js触ったのは、rpgツール2kの形式をjsでなんとかできないものかといった実験の一環で、そのクローンをつくらっている中の人からの質問にあったので試したものです。まだ、ストリームが実装できただけなので、もうちょっと進めてなんとか形になりそうなら、できたよー、とメール返そうかと思います。ダメだったらダメで、「スマヌ」メールになるかと。

opJapan-ACSが気持ち悪かった件

ここ数日AnonymousのIRCでログを撮っていたんですが、やめました。理由を挙げると

- 雰囲気気分が悪くなってくるから

- ダウンロード刑罰化と暗号化されたメディアのリッピング禁止がそれほど悪いとは思えないから

- IRCで話がなぜかループするから

- 積極的に発言する人たちの無知さに吐き気がするから

- Anonymousの中枢の方にいる人にした共同通信のインタビューから中枢との壁を感じたから

- 無用に警察と対立しようとする姿勢が理解できないから

- 掃除が目的なのに、なにがなんでもビラとか名刺とかゴミになりかねないものを配ろうとする矛盾

- 無関係な掲示板にコピペを貼る

くらいですかね。一番目は総括するとそうなるだけで全員がそうゆうわけではなく全体の傾向がそう耐えられなくなりました。自分の発言に帰ってくる発言のほとんどが、もう死んで方向性だけが固定された壺体みたいでひどく虚しくなります。

二番目は、サイバー攻撃のプロモーションとか、色々考えてみると法律としてはダメだけど、それほど政治的に悪い手とは思えないからです。銃刀法は容疑者をうまい具合に引き止める法律なんですけど、今回の法律は電子ネットワーク上でそれに近い振る舞いが可能だと感じています。デジタルなスパイへの対処もかなり有効なはず。まあ、もちろん銃刀法に警察の点数稼ぎに悪用される危険はありますけど。

三番目は、よくわかりません。ポットでもいるのか会話が狭い環でぐるぐるして気持ち悪かったです。

四番目は、DVDの方のCSSとか、WikiLeaksとAnonymousの区別とか、指摘するのが恥ずかしいようなところがいっぱいありました。無知での赦されると思っているような連中とあんまり関わりたくありません。

Anonymousの中枢の人については、伝えなかったこっちも悪いんですが、「浄化」という言葉は慎重にしてほしかったです。ポットっぽい無知で警察と対立したがってる会話がループしてる人たちはどうゆうわけか、「浄化」って使いたがりましたけど。そもそもテロリスト(=というイメージ。中東の人みたいな格好とか)と誤解されても仕方ないような格好でインタビューを受けるなんてウケます。

警察云々は、会話がループする人たちのことなんですが、許可とか、オフ会がデモかどうか？は、普通に警察に聞けばめんどくならない範囲で教えてくれるはず。それに面識があれば、もしもの時に多少融通がききます。警察は悪の組織じゃないですよ。

ビラとか名刺は、前時代的っていうか、Anonymousが存在できる理由を考れば(βノ・V・)ナけって感じがします。ARを利用したゲームを掃除するついでにやったり、まあ、ポットの人たちにはわからないと思いますけど。

最後は、ちょっと離れたらチャンネル#opJapan-

ACSが招待制になってた理由を調べたら、コピペされた掲示板がいくつか引っかけたのでうわーって引きました。

結局、大麻栽培したり密造酒つかった人の話が面白かった以外楽しくなかったです。

orz

PythonとRuby

なんとなくPython使ってたんですけど、なんか正規表現を使い始めてから、すごい使いにくいなと感じて引き返せそうなところだったのでRubyに書きなおしたらRubyがけっこうよかったです。Pythonの正規表現は糖衣構文が無い時点で、なんか期待しちゃいけない気はしてたんですが、まあ、そのとおりでした。モジュールをインポートしないといけないのは、絶対正規表現を差別してます。

そんな愚痴はさておき、インタプリタを起動してちょこちょこ試せる言語はまったくいいものですね。C++のビルド時間はまあ考えたくありません。たまに、速度に不満が出ることはありますが、一部をネイティブ化することもできるわけですしうまく棲み分けできるのはいいです。ただ、Rubyもシングルクォートとダブルクォートで微妙に仕様が違うのは勘弁です。できる限りシングルクォートで書く派なのでエスケープシーケンスがそのまま出力されて焦りました。それから、1.9になって文字コード周りがけっこう厳しくなったのでうっかりしていると、簡単に例外で死んじゃうのはこわいですね。

天にひびき6巻

発売日に本屋さんに行ったのに神様ドルズに気を取られて買えなかったのを昨日やっと買いました。

内容ですが前半（よりも多いかも）が21Cオケでのひびきの活躍で、後半（1/3くらい？）が久住くんのコンクールの話でした。

オケの皆さんの話はもっと掘り下げられそうだったりで、また活躍があるといいなと思います。入谷くん村野くんともう一人の絵被りの娘関連はけっこうありきたりなんですけど、よかったです。滅私っていってしまうとアレなんですけど、単なる時間の勘違いでそれにならなくて済んだのは、運がいいというか世間もそれほど鬼じゃないのだと思えて救われました。オーディション受かるといいですね。それから、南条くんは相変わらずモブの鑑でした。

後半のコンクールは、まあ、選ばれる側の感情をうまく引き出されてちょっと渋面になりそうですが、そこはハラハラさせるのがうまいということにしときます。ずっと出ているピアノの萩原さんは相変わらず卒なくこなしていて、あの安定性が羨ましいです。波多野さんは、久住くん誘えてよかったですね。いつもの、(´Д`´)。。。。。。(ノ`Д)ノした感じが演奏する時に凜として飛んでいっていますのはギャップ萌えます。

それから、話の間にある解説(?)をはじめてまともに読んだんですけど、随分世知辛い話が多くてorzでした。作中でも音楽事務所が突然潰れたんですが、道楽以外でこんなことできないなと思ってしまいます。たった一回の大ポカで消し飛ぶとは、つくづく人間って小さいですね。（遠い目）

今回の解説はスラスラ読めたのでそのうち、1~5巻のも読み返そうかなと思っています。