Teach Yourself KMC

# 独習KMC vol.β

京大マイコンクラブ　著

KMC
Kyoto University
Micro Computer Club

# About KMC

**KMC 第34代副会長 crys**

KMCこと京大マイコンクラブ (Kyoto university Microcomputer Club) は, 京都大学の学生を中心に活動するコンピュータサークルです. まだパーソナルコンピュータという用語も普及していない1977年に設立され, 34年目となる今なおその名を引き継いでいます.

活動は, ゲーム製作や競技プログラミングをはじめ, DeskTopMusicやCG, ハード関連では基板からCPUの作成など, コンピュータに関するあらゆることを対象に, 個人やグループ単位でプロジェクトを立てたり勉強会を行ったりしています. そのほか, IRCnetサーバを提供したり, コミックマーケットに出展したりと, 外部への活動も行っています.

部員は, 工学部, 特に情報学科の者が多いですが, 理学部その他の学部から, また学外からも, コンピュータに関心のある者が集まっています. 部員の興味は人それぞれで, サーバ管理からゲーム製作まで広く関心を持つ者もいれば, イラスト一筋で活動する者もいます. プロジェクトは強制参加ではないので, 自分の好みに合ったものに参加し, 取り組んでいます.

インターネット上にKMCのwebページを公開しており, そちらでKMCについての概要や歴史などをご覧になることができますが, このKMC部誌「独習KMC」は, 我々の最近の活動をよりよく知っていただこうということで, 発案, 製作しました. この一冊が, KMCへの関心の種となれば幸いです.

京大マイコンクラブ(KMC)Web Site : http://www.kmc.gr.jp/

# 独習KMC発刊にあたって

**Tetsu=TaLow 未だにVer.4**

今を去ること四半世紀前, 時はまだ昭和. 一人の小柄な少年? が高い志に目を輝かせ, 京大正門をくぐった. 新入生向け健康診断の長い行列に並んだ彼には, 各部活からの勧誘の声が浴びせられ, チラシは両腕に溢れんばかりになった. しつこい勧誘攻勢に切れた彼は叫んだ.

「勧誘はいらん! ボクはKMCに入るんや! 」

その声を聞きつけた大柄な男が素早く駆けつけて, チラシを彼に手渡した. 「KMCです! 」（声のやたら大きなこの人こそ, KMC中興の祖, Ko!氏こと第9代会長児島彰氏である）.

そんなこんなで彼はKMCに入部した. KMCは当時既にその名を業界に轟かせており, 高校生にもよく知られた存在だった. インターネットはおろか, パソコン通信ですら商用化されていなかった当時は, 「マイコン少年」達の主な情報源はパソコン雑誌と口コミ. KMCの初期メンバーが作り出した独自LAN技術PLANETは, その解説が月刊アスキーの連載になっていたこともあり, よく知られていた. 彼はその連載でLANの概念を初めて知ったのだ. 憧れて入ったKMCには, 他の分野でも多くの才能が揃っていた. 彼は周囲に触発されながら, 多くのことを学ぶ. CもUNIXもPrologもScheme もスキャンラインレンダリングもチョムスキー階層も, みな彼はKMCで知り, 学んだ. 彼は当時の情報工学科の学生だったはずだが, 大学で学ぶのと同等以上のことをKMCの中で学んだのである.

京大の学風を表す言葉の一つに「自学自習」がある. この大学は100人の凡才を産もうとも1人の天才が産まれればよいとする考えで, 徹底した放任主義を貫いてきた. しかし, この自学自習による天才育成が達成されるにはそれなりの「場」が必要だ. 知的好奇心を, 知識欲を, 自己顕示欲を, 果ては若き煩悩をリビドーを, 共鳴するがごとく増幅するためのシゲキ的な場が. 例え全ての人間が1Tbpsで繋がれる時代が来ようとも, 人の情熱をネット越しに伝えるのは容易ではあるまい. サイバーなコミュニケーションの先端を走るハズのICTな趣味においても, 部活という形態が大きな意味を保ち続けるのは, 「世の中にはこんなことまで出来るヤツが実在する」ことを実感し「俺はこんなことができるのだ! 」とジマンするために, 同じ空間の空気を吸うことが必要不可欠だ

からに違いない. 本誌「独習KMC」は, そんな場を今も維持し続けるヤツらの, 熱かったりユルかったりする想いから溢れ出たジマンの固まりで出来ている. 是非, 暖かい気持ちでご笑覧頂きたい. 人によっては生暖かかったりするかも知れないが.

さて, 件の彼はその後, 京大を博士課程まで勤め上げた後に教官職に就いた. 一度他大で修行した後にまた京大に戻り, 最年長現役部員兼顧問として今もKMCに在籍し, その役得で本誌の巻頭を頂いている. 多忙のため奴らと「場」を共有することはなかなか難しくなりつつあるが, 当時の情熱と煩悩はしっかり燃やし続けている. 18歳児の魂43歳まで. 当時, トラジマビキニの宇宙人娘に心底入れ込んだ男は, 今はKMC後輩が原作の, あの鬼娘と同一の音で飛ぶヒロインが出てくる深夜アニメを毎週正座して視ているのだ. 「ハクアたんまだかなー」とかつぶやきながら.

# 目次

**ゲームの話** 2
- zap 3
- cliclu 5
- Honey Freers 7
- Cross Wings 9
- CRAUZEL 10

**フリースペース** 13
- 惑わされるという幸福 14
- マザーボードの買い方ちっく 15
- C++0xの新機能constant expressionの紹介 18
- あなただけの脳内骨格標本 19
- 第二部室電子錠について 22
- irc.kmc.jpについて 29

**今年のKMC** 31

**部員へのアンケート** 37

**勉強会特集** 43
- 実践Common Lisp読書会 44
- パーフェクト線形代数教室 45
- PRML読書会 47
- ICPC勉強会 48
- 組み合わせ最適化読書会 49
- ゆっくりお絵描き 50
- DTM勉強会 51
- C#でゲームを作ろう2010 52
- リアルCPUを創ろうVer.2.0 53

**あとがき** 55

# ゲームの話

p3 zap
p5 cliclu
p7 Honey Freers
p9 Cross Wings
p10 CRAUZEL

Kyoto University Micro Computer Club
KUMCC 2010



http://kmc.jp

# zap

@seikichi

## はじめに

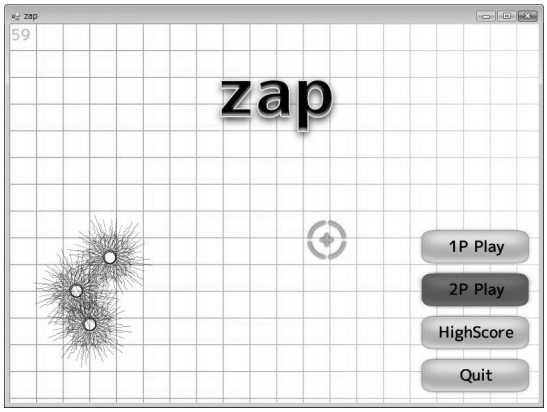
前回(C78)と今回(C79)のコミックマーケットで頒布したzapの開発経緯等について語ります.

## プロジェクト開始

zapはKMCの新入生プロジェクトとして始まりました．新入生プロジェクトというのは，入部したての新入生を上回生が指導しながら進めていくプロジェクトです．私も1回生の頃はnojima先輩にお世話になりました．KMCでは新入生プロジェクトが毎年いくつか立てられ，zapのそのうちの1つでした．

新入生プロジェクトを受け持つことになっても，具体的にどういうゲームを作るかは考えてなかったのですが，花見の際に先輩からWiiザッパー使って何かようぜ，とアイデアを貰ったためガンシューティングっぽい何かを作ってみることにしました．

## 開発




新入生のgutti君にはハイスコアや名前入力のシーン，気球やヨット等の的のコードを書いてもらいました．途中を試験期間が挟んだため，結局締切前に修羅場したのは反省しています．反省したはずなのにcliclu で修羅場しているのは気のせいですよ．
音楽はenba君，イラストはrai君に一部をお願いしました．ちなみに私はコードを書くのが主な仕事でした．イ

ラストや音楽がサッパリでも，手伝ってくれる仲間がいるというのはありがたいことです.

## 開発環境

C#とSlimDXを使用しました．当初はManagedDirectXを使用する予定だったのですが，同じクラスの友人で今年KMCに入部したtacker君が一晩で書き換えてくれたので乗り換えました．本当にやるとは思わなかった．まさにジェバンニ.

Wiiリモコンとの接続にはWiiRemoteというライブラリを使用しました．一昨年に同じライブラリを使用したゲームがあったので，サンプルには困らなかったように思います.

プロジェクトのバージョン管理にはGit，プロジェクト管理にはRedmineを用いました．以前はプロジェクトの管理を内部のPukiWikiで行っていたのですが，昨年の学園祭の頃からRedmineが普及し始めました．またここ最近KMCでもGit派が増えてきましたが，「これからの時代はBazaar」と主張する新入生や，まだまだ一定数いるSVN派のことを考えると，宗教戦争のネタが1つ増えただけのような気もします[1].

## 終わりに

KMCではプログラミング未経験者も大歓迎ですよ! よ! (宣伝)

1 流石にCVSのプロジェクトは消えた.

# cliclu

@seikichi

## はじめに

C79で頒布するKMCDに収録されている，cliclu というゲームの開発経緯について語りたいと思います．

## プロジェクト開始

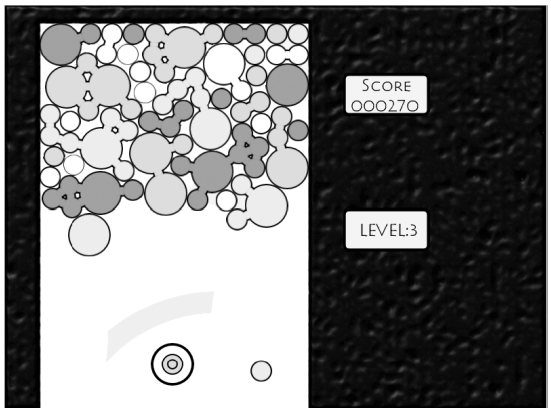
事の発端は10月の終わりに遡ります．例年だと学園祭に向けたプロジェクトが進められている時期なのですが，今年は部員の人数の割に動いているプロジェクトが少なく，会長が「部室でブレインストーミングをしよう」と提案しました．

部室のホワイトボードに「ブロック崩し」という案があったのと，前々から物理エンジンで何かゲームを作ってみたかったので，「Box2Dでブロック崩し的な何かを作るか」と思ってプロジェクトを引き受けることになりました．ちなみにプロジェクト名はphysでした．physicsから取ったもので，「短い方が打つの楽だな」と思って名付けたのを覚えています．本当適当ですね．

## 開発! 開発! 開発!

そんなこんなで始まったプロジェクトですが，一番困ったのはゲームのルール部分でした．

途中まではブロック崩しの面影があったのですが，どうもゲームとして面白く無く，また物理エンジンを上手く使えていない内容でした．そこで部室で他の人に意見を求め，全て取り入れてみたところ，気が付いたらブロック崩しとは異なるよく分からない何かが出来上がっていました．ちなみに私は「さ○がめ」を意識したのですが，他の部員からは「**それは無い**」「**むしろぷ○ぷよ**」などの突っ込みを受けてしまいました．似てると思うんですけどねぇ……．




↑むしろぷ○ぷよらしい

本当は新入生にいくらか仕事を振りたかったのですが，ルールがあやふやなまま開発が進んでいたのと，締切まで時間が無かったため，結果的に音楽以外はほぼ私1人で開発してしまいました．来年はもう少し早くプロジェクトを開始したいですね．

### 開発環境

忘れないうちに開発環境の話をしておきます．今回はC#とXNA Game Studio 4.0，物理エンジンのライブラリとしてはBox2D.XNAを利用しました．XNAには今まで触れたことが無かったのですが，2Dのゲームを作るだけなので，慣れるのに時間は掛からなかったように思います．ドキュメントも整備されていて使いやすいので，これからC#でゲームを作ってみたいという方にはXNAをお薦めします．

### 終わりに

音楽以外はほぼ私1人で開発したと書きましたが，ゲームのアイデアやバランス調整は他の部員の協力無しには成り立たなかったと思います．卒論やレポート，他のプロジェクトで忙しい中，アドバイスをくれた部員の皆様に感謝です．

最後になりますが，プロジェクトは計画的にね!

# Honey Freers

**とびうお**

## 序

私が２０歳になるまでの数週間を華々しく飾った戦術シミュレーションゲーム, おちうお(ゲーム名: Honey Freers)の開発記です.

## 開発の始まり

KMCでは毎年夏ごろから京大の学園祭, 11月祭に向けてゲームを開発しています. 今年は10月になっても１つもプロジェクトが立ち上がらなかったため, 会長主導のブレストで案をだすことになりました. 僕は「Wiiモートでキャラを触りながら育てるゲーム」というのを提案しましたが, 子供たちに見せられないよ! と却下されました. ブレスト最中に「黙れ」と言われたのは初めての経験でした.

結局僕がもらったのは, 「敵にさらわれていたお姫様を城から救出する」というストーリーの戦術シミュレーションゲームの案でした. このプロジェクトには「**おちうお」(単位落とすんだよねとびうお)**という名前がつけられ, プロジェクトが始動しました.

## 重圧と失敗

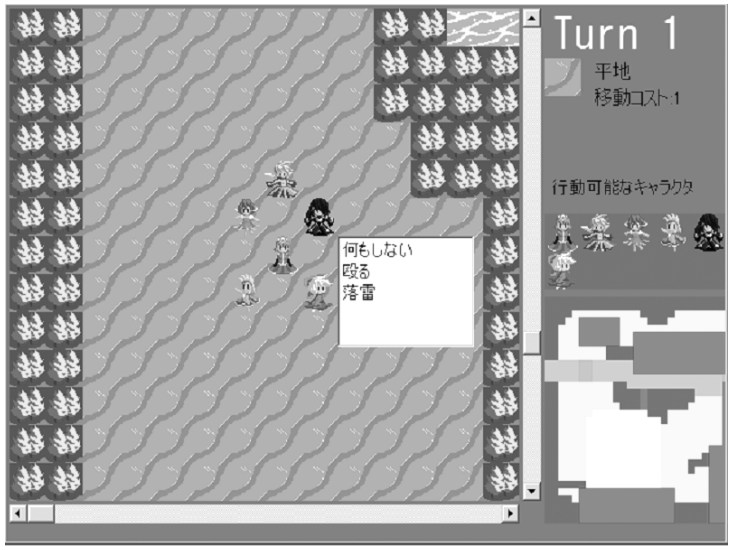
Gitのレポジトリを作成し, 画面にキャラクターが並ぶぐらいまで作った後僕は２週間ぐらいのんびりしていました. するとなんということでしょう, レポジトリの中にキャラクターの画像ファイルがたくさん追加されていました. 自分で後輩に頼んでおいて驚くもなにもないはずなのですが, これで僕は本当にプロジェクトを落とすわけにはいかなくなりました.

開発を進めているうちに, 僕が何も考えずに作ったコードには様々な問題が発見されました. **２つの変数から所持されるインスタンス, コンストラクタ内でまわるゲームルー**

**プ, 製作者 (とびうお) でさえ理解していない状態遷移**などから, このままでは開発が進められないことが判明しました. nojima先輩から「今何かできているようにみえても, それは幻想なので」という宣告もあり, NF３週間前にしておちうおはフルスクラッチを決定しました. その後２日間 (実働２０時間) で元と同じ見た目にまで復活したのは, ずっとつきっきりで僕を操作してくれた偉大なるdis先輩のおかげだと思っています.

### 結論とか

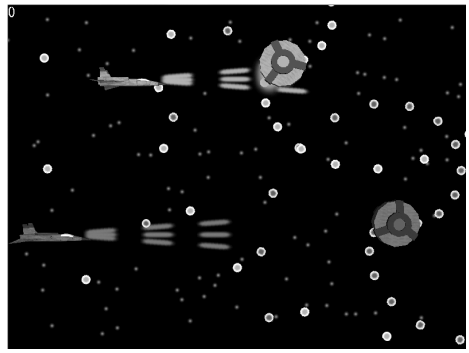
おちうおはNFの締切りの日ではとりあえず動く形になりました. まだ完成じゃないと先輩方に言われつつ, NF向けのマニュアルを書きながら20歳の誕生日を迎えたことはきっと一生忘れないことでしょう.




今回ここまで僕が苦しんだ一番の理由は, 自分の能力をあまりに超えたものを短期間で書こうとしたことです. 工期と機能の量はゆとりをもって, という教訓を次に活かしたいと思います.

# Cross Wings

**tsutcho, hideya, enba**



Cross Wingsプロジェクトリーダーの tsutcho です. 今回はじめて3D のゲームに挑戦いたしました. 3D にすること自体は難しくなかったのですが, 問題はメインエンジン制作とモデリングの両方を自力で行わなければならないということでした. 基本的には 2D なら絵師とプログラマーがそれぞれいて分業するのですが, モデリングが出来る人間はごく少なかったのです(そして他の人も別のプロジェクトで忙しかった). 結局 1 週間に 3 度の徹夜などを敢行した結果, なんとか完成したわけです.

Cross Wings のプログラムを担当したhideya です. プログラム担当とはいっても私がやった事は主に敵の配置といった部分で, 「ゲームとして動作させる」ためのプログラムはtsutcho 氏に書いていただきました. 私はゲーム制作を殆どやった事が無く, プログラムもまだ初心者であるため, 頭に浮かんだ敵の動きをプログラムできないなどといった事態に非常に悩まされました. とはいっても, 以前からやりたいと思っていたゲーム制作での悩みは甘美でもあります. 一つ苦難を乗り越えるたびに, 自分が強くなったと実感できる. 実に素晴らしい事です. さて, 〆切まであと約 1 週間. まだまだゲームは完成していません……おそらくやって来る人生二度目の修羅場, 私は無事切り抜けられるのでしょうか……

——12 月14日 教室内にて, 講義の内容が耳を通り抜けていくのを感じつつ

# CRAUZEL

## (グラフィック, シナリオ, サブプログラマの視点から)

文責：deskull(松永 闇斎)

http://www.kmc.gr.jp/proj/crauzel/

事は自分が新入生として入った2005年から始まりまして, 当時修士であったtic氏を筆頭に, ガチでSLGを作ろうというプロジェクトでありました.

小2で従姉の姐さんからKOEIのスーパー三國志 II をもらって以来SLG脳に冒され, 天下統一 II やら信長の野望(天翔紀〜嵐世紀), ネクタリス, ギレンの野望シリーズに脳髄をさらにかき回された私めは二も三もなく飛びつきました. 大学入る以前からPC98やWindows 用某スクリプト言語で戦略級SLGを作ろうとして, 幾度と無く玉砕してきた経緯もありましたので, KMCという協力者豊かな環境で, いよいよ完成品まで作れると思ってこの上なく高揚としたものです.

ところが, 当然というか全てはうまくいかないもので, 当初1年かけてと思ったものがズルズルと総計3年以上かかってしまいました. 一部初期メンバーは卒業などで退場し, 後輩の中でもプログラムに長けた人達に徹底的にデバッグをしてもらったりしてやっとです. 各人それぞれに問題はあったと思いますが, 作品に対するエゴの曝け出し具合でタスクを滞られせた点に関して言えば, まぎれもなく私めが第一級罪人でございます.

とりあえず, 己の主要な役割の一つだったシナリオ担当(後グラフィックデザイン兼業)の立場からグダグダの変移を振り返りますと, まず『 Baal-Zebul 』というシナリオ案で一つ進めておりました. 大学入学前に地元高校生の同志達の間で進めて頓挫した企画を再構築したものです. 人間の元英雄が栄光の時代から眼をつけられていた異邦の

魔神に魂を売り，魔界の一将校として戦乱の渦中に飛び込むという内容でした．ファルコムが韓国からローカライズした『西風の狂詩曲』の影響を受けまくったために，登場人物の名前をラテン系ヨーロッパの史実上の人物からひっぱり出しまくり，今思うと「うわぁ」な調子です．こいつは話として極めて大きくなりすぎたのと，キャラクターデザイン担当者と連絡が取れない件など，様々なことから廃案となり，自分がキャラデザを兼ねて別案を考えることにしました．

次に考えたシナリオが『Necromaster』と言いまして，タイトル通り屍霊使いが主人公のこれまた陰鬱さをむしろ売りにした内容でした．これも高校時代からの原案でして，当初はマッドサイエンティスト的な主人公が，捩れた性格と背徳的な行為から，少しずつ人間性を取り戻していくような話にしていく予定でした．しかし，彼を取り巻くであろう環境を考えていくにつれて，二転三転．主人公の初期のキャラデザはこれまた某新世界の神まんまな具合でして，当時の流行の影響をもろ受けしすぎる底の浅さがまだ見て取れました．

一方，この頃にはKMCの素晴らしい先輩方が開発に携わったローグライクバリアント『変愚蛮怒』の元ネタ(指輪物語，真世界アンバーシリーズ，クトゥルフ etc)の啓蒙もありまして，どこかCRPGの表層的な部分に追従するばかりの調子から，真面目にファンタジーのノウハウを得る機会もあり，間違いなく物書きとしての質は上昇しつつありました．ええ，完成がまともにできてないんじゃアレですが．

当初, 屍だけが友達の寂しい主人公の周りに, 主人公も当人達自身をも互いに引き立て合うような愉快なキチガ^H^H^H仲間達をおけるようになり, また屍霊術のような邪悪な技をも容認する奇怪な文化像を構築するなど, 一挙に世界が広がったものです.

で, 結局また話が広がりすぎて収拾をつけるのに苦労しました. というより収拾できてるとは言い切れません. 当初考えていた登場人物の視点からのエピソードはごっそり削り, 架空の歴史の中で起きたいくつかの戦いを淡々と列挙していく形になりました. タイトルも『CRAUZEL』になって現在に至っております. ここまでの過程で削られた産物もいずれ何らかの形で日の目が見れればいいのですが.

システムもシステムで非常にグダグダが続きましたといいますか. 一応, システムとシナリオの連動は必要と言う形でかろうじて容認されていたのですが, 本来システム製作の権限持っていたyasufuku 氏に対して越権行為もはなはだしく, あれこれちょっかい出した点は否めません.

yasufuku氏は同じSLG好きでも, ルールからステージ構成のセオリーまで好みが色々と真逆だったもので, それをお互い意識して話を進めるのにも一苦労でした. 最終的には皆と相談しながらさらに混乱しつつも, 安定して手ごたえのあるバランスになったはずではあるので, ご迷惑をかけた皆様には平謝りしつつも, よかったとは思っています.

個人的には続編であるとかリメイクであるとか, そういうのをまだ諦めておりません. ハイ・ファンタジーとしての観点も, ウォーSLGとしての観点もこれから実現していきたいものがちらほらあるので, いい加減片付けていきたいところです.

フリースペース


p14 惑わされるという幸福
p15 マザーボードの買い方ちっく
p18 C++0x の新機能
constant expression の紹介
p19 あなただけの脳内骨格標本
p22 第二部室電子錠について
p29 irc.kmc.jp について

# 惑わされるという幸福

**hideya**

皆さんは, パズルをどのような存在だと考えているのでしょうか? 多くの方は, 頭の体操, 暇つぶし, 毎日のささやかな楽しみ, と思っているでしょう. それらは決して間違ってはいません. しかし, 決してそれがパズルの全てではありません.

パズルには, 制作者がいます (中にはプログラムによる自動生成で作られた問題もありますが, それらは粗製乱造された紛い物であると私は考えます). そして, 彼らが作った問題＝作品には, 彼らの魂がこもっています.

時として, 見た目の美しさに. 時として, 解き味[1]に. 時として, 難易度に……問題はまさしく, 制作者の写し鏡であると言えるでしょう.

あなたが問題から個性を感じたら, それは制作者の個性に他なりません. 問題を解くという事は, 制作者と向き合って対話している事と同値なのです! ですから, これからパズルを解く時はパズルの個性を感じ取ろうとしてください.

初めはなかなか難しいでしょう. せっかくその気で解いたパズルが, 何の面白味も無い作品であることもあるでしょう. (そんな作品は, どうぞ鼻で笑って下さい) ですが, もし個性を感じ取ることが出来たなら……あなたはこれまでに無い, 新たなる世界が広がっていく感覚を覚える事でしょう. それはあなたの人生における大いなる楽しみとなる筈です!

さぁ, あなたもパズルの雄大な世界に第一歩を踏み出しましょう!

ほら, 「ソルバー[2]作るか」なんて言ってないで……

1 解き味……解いていく過程の展開の仕方のこと.

2 ソルバー……ここでは単に, 様々な問題を解くプログラムのこと.

# マザーボードの買い方ちっく

Chick

やはりGeekたるもの自分の使うPCくらいは自作したいもの．ですがCPUやグラフィックボードなどに比べてマザーボードというのは一体どのモデルを買えばいいのかよくわからないという人が案外多いのではないでしょうか．実際にPCを自作する際には予算との兼ね合いもあってか，マザーボードは軽視されがちですが，実は大変重要なパーツなのです．ほとんどの主要なパーツはマザーボードの上に載ることになるため，マザーボードの選び方によってどんなPCに仕上がるか決まるといっても過言ではありません．そこで今回はマザーボードの選び方について，若干の個人的こだわりを含めつつ解説しようかと思います．

まずはマザーボードにはどのようなものがあるかについてですが，用途によっていくつかの種類に分けられます．省スペース型，エントリー向け，メインストリーム，ゲーマー用，オーバークロッカー用，そしてサーバー用です．一般にこの順に従って値段が高くなり，基板自体も大きくなる傾向にあります (例外も多数ありますが) ．

まずは省スペース型についてですが，これにはMini-ITXやMini-DTX といったフォームファクタ (マザーボードの基板の大きさやケースに固定するネジ穴の位置などに関する規格) が採用されることが多く，PCIスロットが1〜2本しかないなど拡張性に乏しい代わりに基板が小さく，場所を取らないことが特徴です．Intel Atomなどといった省電力型のCPUを搭載し，高機能NASや省電力サーバーとして使う人が多いです．一方拡張性や冷却性に乏しいため，CPUやグラフィックボードの搭載に関して制約が多く，メインマシンとして使うには性能が不足しがちなのでお薦めできません．また，搭載するケースも小さいものを使うことになるため，組み立て作業が少し面倒になります．汎用マシンとして使うよりは，用途を絞った使い方をするマシンに向いたものだと言えるでしょう．

次にエントリー向けですが，これにはMicro-ATXというフォームファクタが使われ，GPUをオンボードで搭載したものが多くみられます (Intel 6シリーズのチップセットの場合，GPUコア自体はCPUに内蔵されますが，GPUの機能を使うにはマザーボードでの対応が必要になります) ．また，ストレージ周りについてもRAID機能を搭載しないなど，上位のものに対して機能を絞った点が多いです．しかしPCゲームやオーバークロックなどをしないという前提であれば，グラフィックボードを別に搭載する必要がないために予算を削減でき，その分CPUを高性能なものにすることができるため，このようなマザーボードが適していると

いえるでしょう．基板の大きさも比較的小さいので，HTPCなどのような用途にも向いています．

次にメインストリームです．これにはATXという最も一般的なフォームファクタが採用され，拡張性と機能性に富んでいます．最近では簡易的ではありますがオーバークロック機能を搭載していることが多く，またRAID機能も充実しており，迷ったらこれを買っておけば間違いはないでしょう．初めて自作をする場合にもお薦めです．ただし，グラフィックボードを2枚以上搭載する際には制約があることが多いです．そのような場合にはゲーマー用のマザーボードがお薦めです．

ゲーマー用マザーボードでは2枚以上のグラフィックボードをPCI-Express x16の帯域で動作させることが可能であり，製品によっては3枚以上の搭載もできるようになっています．また，オーバークロック機能も充実しており，CPU電圧やメモリアクセスタイミングなども細かく設定できます．オーディオチップが高級なものであったり，専用のサウンドカードが付属していることもあり，3Dゲームをプレイする際にはより高品位なサウンドを楽しむこともできます．ハイエンドなチップセットを搭載しており，さらに冷却性の向上のために豪華なヒートシンクがついていることが多いです(それゆえCPUクーラーなどとの干渉が起こることもありますが)．あなたが普通の神経の持ち主であればゲーマー用のマザーボードを買えば，機能面で不満を感じることはほとんどないでしょう．

しかし，それでも満足できず，挙句は**CPUに液体窒素を注いでしまうような人**にはやはりオーバークロックマザーしかありません．上記のゲーマー用マザーボードの機能に加え，さらに詳細なオーバークロック設定のできるBIOSや，水冷にも対応したチップセットクーラー，基板上に電源スイッチやCMOSクリアのボタンなどがあり，BIOSのPOSTコードが表示されるようになっているなど，至れり尽くせりです．製品によっては基板上のスイッチや外部に接続したデバイスからのオーバークロック設定もできるようになっているなど，オーバークロックに関してはマザーボードだけで全てが事足りてしまうかのようです．またその用途から，ハードウェアとしての堅牢性も非常に高いです．ただし，これら2つのカテゴリに属するような製品は3万円〜4万円と非常に高価です．財布や口座の残高と相談しましょう．

最後にサーバー用途ですが，これはこれまでに紹介したどのマザーボードとも異なり，XeonやOpteronといったサーバー・ワークステーション用のCPUを搭載し，場合によっては複数個のCPUを搭載します．普通にPCのを自作する際に使うことはほとんどないでしょう．マザーボードを選ぶ際には，以上で紹介したものの中から自分に用途にあったものをまず選び，その中から具体的な製品を選ぶと良いでしょう．その際，搭載するCPUに対応したマザーボードであるかを一番に確かめておいてください．

では実際にどのようにしてマザーボードを選ぶかについてですが，日本人的な感性に従えば，最初に気になるのはどのメーカーの製品が良いかということでしょう．しかしマザーボードに限らずPCのパーツに関しては共通していますが，どのメーカーが良いとかいうのは一概には言えません．ですが初めて自作する場合には，ASUSやGIGABYTE，MSIといった比較的国内でのシェアの高いメーカーを選ぶとサポートが受けやすいかと思います．ただ，これらのメーカーでなくとも自分の気に入ったマザーボードがあれば，そちらを選んだほうが良いでしょう．

その上でまず注意しておくべき点はATX24ピンコネクタの位置です．これはかなり大きいコネクタで，それに対応して電源ケーブルも非常に太いため，マザーボードの真ん中にこれが位置している場合はケーブルがかなり邪魔になります．そのためこのコネクタは基板の端に位置していることが多いのですが，廉価なマザーボードでは基板の真ん中，CPUにかなり近いところに位置していることがあります．これでは電源が下置きの場合にはグラフィックボードが邪魔をしてケーブルが届かなかったり，CPUクーラーが設置できなかったりします．なによりメンテナンスが非常にやり辛いです．ですからマザーボードを選ぶ際には最初に24ピンコネクタがどこにあるかを確認しておきましょう．

次に注目すべきはおそらくPCIスロットの配置についてでしょう．PCIスロットの配置は後で移動させることもできないため，不用意に選んでしまうと後で後悔することになります．特に2スロット以上を占有するような大きなグラフィックボードを搭載する場合，ただでさえ2スロット埋まってしまう上に冷却性を考慮するとそのグラフィックボードの真下のスロットは事実上なにも搭載することができなくなります．そのため，あらかじめPCIスロットにどのようなカードを搭載するか考えておき，それに合ったスロット配置をもつマザーボードを選ぶようにしておくと良いです．

これを考慮すれば製品の候補としてはおのずと2〜3個くらいに絞られてくるでしょう．この上で決め手となるような点があるかというと，実はもうあまり重要な注目点は残っていません．現在では基板上のコンデンサもほぼ全ての製品で固体コンデンサになっており (Intel を除く)，あまり差別化はできていません．

SATAコネクタ，冷却ファン接続用のピンや，USBピンヘッダが自分にとって使いやすい位置にあるかどうか，電源フェーズ数が十分であるかどうかを確認する程度でしょう．あとは各マザーボードが売りにしているような機能を見比べ，価格との兼ね合いで決めてしまえば良いです．

ただ，ここで私が1つだけ推しておきたい点があります．それはPOSTコードの表示機能です．これはPCが何らかの不具合で起動しなかった際，どこでコケたのかを教えてくれる機能で，オーバークロックをするかどうかに関わらず大変便利な機能です (オーバークロックをしな

い限りはPCが起動しないという事態に陥ること自体あまり多くありませんが). 残念ながらこの機能を搭載するマザーボードは価格が高い傾向にありますが, これがないとエラーの原因を特定することが困難であり，有効な解決策をとることができなくなります.

ですからもし最後にどのマザーボードを買うか迷った場合には，POSTコードの表示機能がついたマザーボードを買うと幸せになれるかもしれません．マザーボードは直接PCのパフォーマンスに関わらないパーツですが，各製品に他にない個性のある素晴らしいパーツです．これにこだわれば自作の世界はさらに広がります．ぜひとも自分にあった最高のマザーボードを見つけてみてください.

# C++0xの新機能 constant expression の紹介

**chir**

今現在 C++ の新しい規格 C++0x の策定が進んでいます. また，C++0x の新機能の一部は既に一部のコンパイラで実装され始めています. 今回は，GCC4.6 から実装された constant expression という機能を紹介したいと思います.

この新機能によって, 新たに constexpr という予約語が作成され, 関数の戻り値や変数にたいしてこのキーワードを用いることができるようになります. このキーワードを戻り値に使用された関数は, コンパイル時に実行され, 関数の戻り値がコンパイル時定数になります. 関数の戻り値がコンパイル時定数になることによって, 今まではコンパイルすることができなかった様々なコードがコンパイルできるようになります.

例えば,

```
constexpr int array_size
(int x)
{
    return x * x;
}
int array[array_size(5)];
```

というようなコードをコンパイルして実行することができるようになります.

また, numeric_limits テンプレートなどが constexpr を用いて書き直されることになるので,

```
const int z =
numeric_limits<int>::max();
```

というコードもコンパイルできるようになります.

この他にも C++0x には様々な新機能が存在します. 多くの機能はすでに一部のコンパイラで使用可能になっているので, 是非いろいろ試してみてください.

# あなただけの脳内骨格標本

**Moko**

なにやら怪しげなタイトルになってしまいましたね.

こんにちは, KMCで絵ばっかり描いているMokoです. 最近, 夜中に突発的にお絵描き勉強会が始まるようになりました. 夜中は前頭葉が麻痺して理性がふっとぶせいか, とても有意義な議論が行えました. ここでは, その場で言語化できたお絵描きのコツのようなものを書こうと思います.

## よく観察して, 骨格を見極める

絵の上達方法には諸説ありますが, 最たる近道は「とにかく描くこと」だと思います. そして, 「対象物をこれでもかとさまざまな角度から, 舐め回すように観察する」ことだとも思います.

例えば人の絵を上手に描きたいのなら, アニメや漫画の絵を参考にしてもよいのですが, 最もよいのは身近な友人を捕まえて描かせてもらうことです. その際に, 顔のパーツを三次元的に捉えるようにしてみましょう.

私は実は, 部員の顔のパーツをこっそりと穴が開くほど眺めており, 脳内でデータベース化しています. この前それを言ったらひどい (というか変態) と言われ

ましたが, 気にしてはいけません. バレないように目で盗撮しましょう. 最初は時間がかかりますが, 慣れればさっと観察できるようになります. そして, 家に帰ったら, あの人の顔はこんなだったかな……と, そらで立体的に思い出せるくらい頑張ってみます. まさにこれは恋です. いいえ, 嘘です.

とにかく顔パーツの記憶ができたら, 次は身体もしっかり記憶するようにしてみます. そしてひたすら描きます. そうするとあら不思議, なんだか顔や身体の「基本構造」が見えてくるのです.

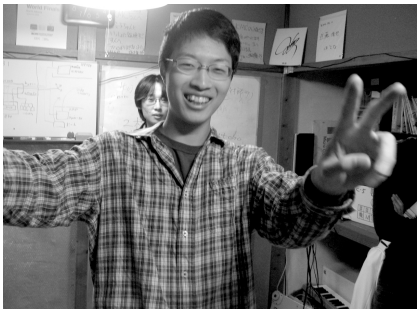
そして基本構造が見えれば, 簡単にディフォルメができます.

たとえば, ある部員の写真を例にして, 基本構造をつかんでみましょう.
かっこよく線形代数の証明問題を解く彼をとりあげてみました. 下から撮った写真なので足が長く見えます.

まず, よく観察して模写をすると次の絵のようになります.
キリッとしていますね. さらにこれをよく睨んで, 服などの余計な要素を取り払っていくと, 2番目の絵のようになります.

頭頂から眼窩までが2xとすると, 眼窩から頬骨のあたりまでがx, また, 頬骨から顎のあたりまでが同様にxの長さになることや, 肘と身体の中心線の交点がみぞおちに来ること, 手はだらりとおろすと太ももの中ほどに来ることなどが分かってきます.

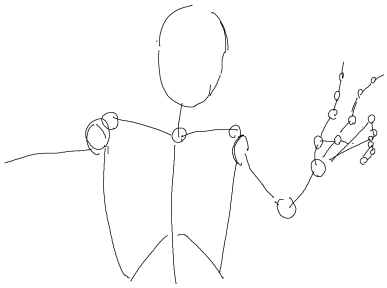
この調子で, もうひとつ例を挙げてみましょう. 次の写真の彼は満面の笑みをたたえていますね. 後ろに「0は自然数だ」と訴えてやまない人がいますが, まずは手前の彼をよく観察します.



いい笑顔を模写しました. もっとよく睨みましょう.

すると, さっきのように色々見えてきます. 首から来た筋肉が鎖骨につながり, それが水平に伸びて肩から腕を吊るしていることや, 肋骨がみぞおちの少し下あたりに来ることがわかります. また, 首は後頭部から伸びているようです.

これをもっとじっと見てみると, その中身が見えてきます.
つまり, 骨です. どばーん! 人間はみなこのような構造なわけですね. 可愛いあの子も, ニクいあいつも, みいんな肉と骨と皮からできているのです.



よく絵を描くときには棒人間を描くとわかりやすいと言いますが, 実は逆で, 観察して構造を理解して, 初めてきちんとした棒人間が描けます. 棒人間は観察の末に得られる, とても難しいものなのです. ともあれ, これであなただけの脳内骨格標本ができました. あとは, これをグリグリ動かして, 好きなポーズを取らせれば, 最近よくKMCのゲームや看板に登場するキャラクター, 「みこたん」もこの通り[1].

かなり端折って説明しましたが, 感じはおわかりいただけたでしょうか? これを読んだあなたも, 是非明日からは隣の友達を熱く見つめてみてください. きっと何かが見えてくるはずです.

1 ComicStudioの細い線で描いてたらかすれ気味になりました.

# 第二[1]部室電子錠 (box2-key)について

walkure@kmc.gr.jp

## 1.電子錠について

KMC部室は2007年6月よりFeliCaをキーとする電子錠が稼働しています. 仕掛けとしては, 以下の二つです.

- FeliCa製造時に書き込まれているunique ID(以下IDm)を読み出し, 登録されたIDmならば鍵を解錠.
- 扉が閉まっている際に解錠されていれば施錠する(オートロック[2]機能).

外観写真 左側にPaSoRiがある

1　歴史的経緯による名前なので, 今は第二部室しかありません.

2　扉が歪んでて, 完全に閉まりにくい事から現実はセミオートロックですが.

また, 解錠時のログをIRC及び内部Webページへ出力することで, 誰が来たのかネットワーク上から随時確認することが出来ます. 尤も, 「誰がいつ解錠したか」しか取れないので, 部室の様子を外から知るためにWebカメラを設置してあります.

## 2.導入に至る経過

KMC部室は当然ながら昔から鍵がありました. 確か鍵を得る条件は半年程度アクティブな部員であるとかそんな感じだったはず[3]. 部室に常駐する人がいなくなった[4]ことにより, 鍵を持たない部員(特に新入部員)が部室へ任意に行くことが出来ないという問題[5]が発生しました. そこで,

- 何時誰が開けたか分かる
- 任意時点でrevoke出来る

等の利点を持つ電子錠を導入することにして, 新入部員へ早く解錠権限を渡せるようにすることになりました. 鍵ハードウェアとしては, リーダー及びカードが安価容易に入手できる[6]事から, FeliCaを採用[7]しました. FeliCaである事を要求するので, Suica 等のIC乗車券, 京大の学生証, おサイフケータイなどFeliCa一般に対して対応しています.

## 3.錠前

当初は既存の鍵に怪しげなメカ[8]を仕掛けてサムターンを回して解錠を制御する予定でした. しかし, 準備段階での様々な検討および事前調査により最終的には錠前ユニットを電気錠ユニットへ交換することにしました. 主な理由としては以下三つのはず.

3 自分が物理鍵を渡される権限を得る前に電子錠を実装したので覚えてない.

4 昔は事実上住んでるに等しいような人がいたそうで, 鍵問題は顕在化し得なかった.

5 それによる定着率の低下, という話があったらしい.

6 この時点(2007年)で京大の学生証IC化という話は既に出ていた.

7 と言うか, 他に選択肢が(主にカードリーダという点で)なかった.

8 モーターでぐりんと回すようなそんな感じで,

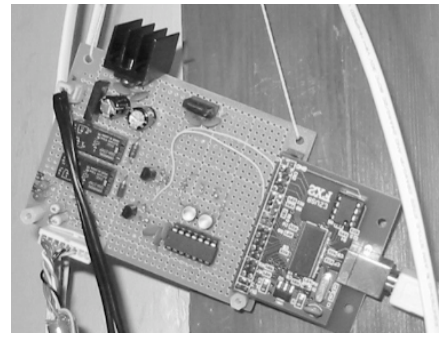
- 怪しげなモノを付けると壊れたときに修復が面倒[9]
- 大家さんに鍵交換の許可が取れた
- 通販で買えることが分かった(3万円程度[10])

当初は業者に依頼する予定でしたが, ぐぐって見つけた店に行って話をしたところ「錠ユニットだけつけて」という発想を理解して貰えず断られてしまったため, 結局自力で交換をしました. 扉が古くてユニットがそのまま使えなかった為にユニット交換は扉の加工を伴ったので, もう少し見てくれを綺麗にしたいところ.

## 4.鍵制御部

USBインタフェイスを持つマイクロコントローラEZ-USB FX2[11]を使いました. 基本的にハードウェアはシンプルに鍵を駆動するのみとして, マイクロコントローラ内のプログラムで全部処理する[12]構成としてあります. コントローラはDC3.3V動作, 一方鍵ソレノイドはDC24V駆動なので, 繋ぐための回路を適当に組みました.

基本的にはコントローラの出力をトランジスタ経由でリレーをドライブして24Vをスイッチング. また, 解錠施錠センス及び扉開放センスはマイクロスイッチなので3.3Vを流してLPFを咬ませ, 74AC14[13]で受けてチャタリングを潰しています. 一番のハマり所はコントローラ出力のpull down忘れ[14]. 電気的に鍵とコントローラをフォトカプラ入れて絶縁するつもりが端折ったので, なんとかしたいところ. あとは見てくれがやっぱり残念なので, なんとかしたい.



駆動回路 右側にEZUSBがぶらさがっている

9 工作できる人がその時にいなければ修復不能のがらくたになる.

10 怪しげなメカより高いか安いかに関して検討した記憶はない.

11 速度的にはかなりオーバースペック. AVR/PICはライタが無くて断念. FT245等は当時存在せず.

12 当初はハードウェアでソレノイド駆動タイマを実装する予定でしたが.

13 74HC14は買いに行った寺町マルツで売り切れてた.

14 TRMの記述を読み落としてた. 先輩に指摘されるまで気付かず.

## 5.プログラム(コントローラ側)

EZ-USB FX2はCPUとしてIntel 8051を積んでいて, 8051から見えるレジスタにUSBデータが降ってくる[15]仕掛けになっています. 開発は8051用の環境を流用できます. コンパイラはsdccを使いました[16]. ファームのホストからコントローラへの転送はfx2_programmerを使っています.

基本的なコードは

- 鍵操作としてソレノイド100ms程度のインパルス電流を流すタイマ
- 解錠かつドアが閉まっていれば施錠するループ
- USBから解錠指令が来たら解錠
- ソレノイドを駆動したら, 6sec程は電圧を流さないタイマ

という四段構成です. このロジックもう少しなんとかしたいところ. 現時点では

- ちゃんと施錠されたか確認する
- 10秒ぐらい待てよ

など, 運用上のルールで残念な所をカバーしています.

- Fenrir's BLog: EZUSB FX2LP Keilからsdccへ
  http://fenrir.naruoka.org/archives/000469.html
- fx2_programmer
  http://volodya-project.sourceforge.net/fx2_programmer.php

## 6.プログラム(コンピュータ側)

USBでEZ-USB及びカードリーダPaSoRiと繋がるLinux ( Debian/ lenny )マシンをホストとしました.

15 高速通信しようとすると何か色々あるようですが.

16 ぐぐったりTRM読んだりして頑張ったんだけど, 数年前の事とか記憶が……. ちなみにsdccはaptitudeで入ります.

PaSoRiとの通信はlibpasori[17]を用い, EZ-USBとの通信はusbdevfs[18]を使っています. ちなみに, リーダへのpollingでIDmが取れます. 基本的には libpasori をそのまま使っています.

ソフトウェアは, 以下三つのユニットから構成されています.

- リーダにアクセスしてIDm乃至カードがないエラーを返すバイナリ
- USBで解錠指令を送るバイナリ
- 以上のバイナリを必要に応じて叩くperl script

perl scriptは上二つのバイナリを随時叩いて, 以下の動作を行っています.

- 取得したIDmの照査
- IRC及びファイルへのログ吐き出し

このような構成にしたので, 解錠時のホスト側動作を容易に拡張変更する事が出来ました. また, IDmをperl scriptが読み込むのでシンプルに文字列を扱うコードを実装できました.

## 7.セキュリティ

鍵を考える上でセキュリティに関して考えるのは必須です. 合い鍵をみんなに渡してやっていた従前の方式よりセキュリティが劣っては困るので考察をしたはずですが, 当時のログがない[19]など, 情報がないので今もう一度考えてみます.

IDm使って認証する実装は, 外にテンキーを設置してみんなが個々に割り振られた解錠番号を打ち込んで解錠するのとほぼ同値と考えられます. よって, 番号を割り振られた人以外の賊が知れば解錠できます. 具体的にはFeliCaを盗まれる, カードを複製されるの二つですが, 前者は普通の鍵でも起こりうる事態で, 気付いた時点で revoke できるぶん電子錠の方が有利です. また, 後者に関してはまず対象カードのIDmを得る為にカードを得るかホストへ侵入する事が必要になりますが, 合い鍵を作られないように

17 http://libpasori.sourceforge.jp/

18 最初はlibusb使ってたけど何かに躓躇いて諦めた.

19 当時わたしは掛け持ち先の別サークルに出ずっぱりで例会に出れてなかった.

するのと同等の注意で前者はカバー, 後者に至っては鍵のセキュリティ範疇から外れるので, 物理鍵に劣るとは考えにくい.

また, PaSoRiへつながるUSBケーブルが露出している[20]ので, ケーブルぶった切ってPC繋いでUSB機器をエミュレーションすれば解錠は原理的に可能です. 部室内に現在ある機器の価格及び価値を考えて, そこまでする賊がいるのかどうか.

故障に関してですが, 鍵ユニットを丸ごと交換したので, 鍵ユニットが壊れたら鍵ユニットを交換, また駆動系が故障して修復不能なら電気錠の物理鍵を複製して従前に戻す方向にしました.

## 8.問題点

電子錠システムが稼働して3年が経過しました. 大きなトラブルとして, 2008年度春合宿の際に電気錠と制御部を繋ぐケーブルが外れて部室へ入れなくなると言う件がありました. 押し込んで解決しましたが, 抜本的な解決にはケースに入れる必要がやっぱりあります. また, ソフトウェアが頭悪かったり, マイコンが鍵動作の際[21]にリスタートしているようなログをホストが吐く[22]など, 不安な点がないとはいえない. いつかなんとかしたい.

## 9.先例

制作時点でぐぐって見つかる先達は電通大の例であり, 彼らが作成したという事実と, 作成したライブラリ[23]はかなりの助けになりました. 現在では, 電通大以外に於いても作成した例が見つかり, やっぱりみんな考えることは一緒なんだという感慨. 彼らとの違いは

- 鍵ユニットを丸ごと交換(扉をガリガリ工作した)

20 当初は窓内側にPaSoRi置こうとしたんですが, 窓ガラスが鉄線入りでカードへ給電出来なかったので諦めた.

21 電圧がdropしている?

22 デバイスが抜け, 再挿入されたのと同等のログを吐く.

23 libpasoriのこと.

という一点に尽き, 大学の部室を借りているサークルとの違いとも言える. 最も, 大学の部室扉ぐらいだと加工せずともユニット交換可能だったかもしれないし, 京大なら頑張れば電気錠ユニット設置ぐらいはしてくれた[24]かもしれない.

- 百萬石 2008年春号 / 電気通信大学MMA
  http://w3.uec.ac.jp:8081/club/mma/hyakumangoku/hyakuman08a.pdf
- 非接触ICカードを利用した電子錠の制作 / 芝浦工業大学文化会無線研究部
  http://www.shibaura-hac.net/kenkyu/2006/takano1.pdf

## 10.あとがき

前述したように, トラブルは皆無ではありません. また, ドキュメントの端々から滲み出る残念さなど, 何とかしたい点はあります. しかし, 修理が必要な故障が未だ起きていない事は, それなりにうまくできたんじゃないかな, という気持ちになりはします.

このようなミッションを完遂できたのは, 「自力で出来るのなら自力でやっちゃう」というKMC文化のおかげなんじゃないでしょうか. この前, 部室トイレのリフォーム[25]を後輩中心にやりましたが, それもDIYサークルとしての, KMCのカラーが結構出てたような気がします.

24　西部構内の部室棟立て替えに関して, 事前に要求した団体のドアは電池駆動のナンバーロックになった, 等.

25　壁と床を張り替え, 便器が暖房便座に進化した.

# irc.kmc.jpについて

walkure@kmc.gr.jp

KMCが提供するIRCnetサーバ“irc.kmc.jp”について, 簡単な説明をします.

## 0.IRCって?

IRC (Internet Relay Chat) とは, 所謂チャットのシステムです. 最近ではSkype等に押されつつある気がしますが, 世界的に根強いファンがまだ数多く存在します. 大学サークルに於いて会話・雑談用に使われている例も散見され, KMCもIRCを使って地理的障壁を乗り越えコミュニケーションを取っているサークルの一つです. 歴史が深い簡素なプロトコルなので, コンピュータのOSを問わず, 軽快にチャットを楽しむことが出来ます[1].

ユーザはクライアントソフトを使ってサーバに繋ぎ, 会話をします. チャンネルという談話室みたいな空間を任意に作ることが出来, 複数人で会話をする際に便利[2] です. 複数のサーバがツリー構造のネットワークを構成して広大な世界を構築できる[3]のが特徴ですが, サーバ同士の接続が切断されれば当然世界が分かれてしまうという弱点[4]もあります.

サーバ群のネットワークは世界に数多く存在します. 有名なネットワークとしてはIRCnet, EFnet, freenode等が挙げられます. これらは国際的なネットワークですが, 国内に閉じたネットワークも当然に存在します. 日本ローカルではFriend Chat, ReichaNet, 2ch@IRCなどが有名です.

KMCでは, 国際的なIRCサーバネットワークであるIRCnetに接続しています.

---

1 最近でもUstream.tvのコメント機能実装に使われています.

2 もちろん, 個々人に対して話しかけることも出来ます.

3 Relayの由来.

4 グラフ構造なら, 複数の経路を持てるのであまり顕在化しないのですが.

## 1.IRCnetについて

IRCnetは世界的なIRCサーバネットワークの一つで, 日本からはWIDEがサーバを提供していた[5]ため, 日本に於いてもメジャーなネットワークの一つです. このネットワークには「ユーザは自国に設置されたサーバを使う」というポリシーがあって, 国外からの接続を基本的に拒絶します. しかし, 日本に於いては日本語処理の問題[6]などから, チャンネルに日本のサーバ経由でしか入れない設定[7]をすることが一般化してしまっています. これにより, 海外からいつものチャンネルで会話をしようとすると苦労を強いられることになります.

KMCでは, IRCnetに於いて様々なチャンネルを作って相談したり, 雑談したりしています.

## 2. 経緯と現状

KMCがIRCnetにサーバを提供することになったのは, WIDEのサーバ提供終了アナウンスに伴って部員の一部から「やろうぜ」, という声が挙がったことによります.

KMCとして意志決定をしたのは2010年の春で, それから関係各所への交渉や調整を始めました. 7/1にIRCnetサーバのひとつirc.media.kyoto-u.ac.jpと接続しています.

現在は, 部員に対して限定的に公開して設定や動作の確認を行っています. この部誌を発行する時点で, サーバを一般公開するタイミングは未定となっています. また, IRCnet が世界的なネットワークであることから世界的に攻撃を受ける可能性があるので, サーバは日本国内の一部プロバイダ以外からは到達不能な設定で公開する予定です.

5 2010年6月末で終了.

6 多くのユーザが日本語エンコーディングにISO-2022-JPを使っていて, IRC的に使えない文字とバッティングする.

7 チャンネルマスク「:*.jp」のこと. チャンネル名を%で始める一部クライアントは暗黙のうちにこの設定をしています.

# 今年のＫＭＣ

2010年のKMCのイベント
by tsutcho

# 今年のKMC

KMC第33代会長 tsutcho

このコーナーではKMCの今年（2010年4 月から12月まで）の活動を紹介していこうと思います. 勉強会, プロジェクトについては別の項にまとめてあるのでそちらを参照してください. ここはイベントの紹介がメインです.

## 4月 一般活動……新歓

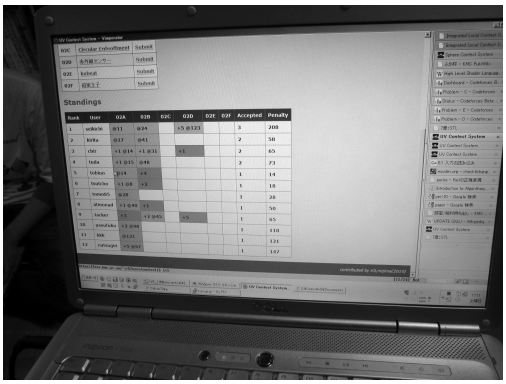
### 花見

川原で花見をしました. 団子をダンボール箱いっぱい買ってきたりピザを買ってきたりと楽しみました. 花を見る人はいないかと思いきや(木登りという手段で)桜を楽しむ人もいました. KMCらしく(?)ノートパソコンを取り出して小池スタイルを始める人の姿も……？

## 5月 一般活動……続・新歓

### ラピッドゲームコーディング祭

新入生に 1 日でゲーム作りを体験してもらいました. シューティング, アクション, フレンド○―クの某ゲームの再現などなど, 1日で様々な作品が生まれました. 写真はこの日がこどもの日だったということで部員が持ってきた兜を装備している様子です.



### ふか杯

部内でプログラミングコンテストを行いました. コンテストに提出されたコードが適しているか否かを自動的に判定するプログラムも部員が製作しました. 初心者でも解ける問題から誰にも解かせないつもりの問題まで, いろいろな問題が用意されました. 同様のイベントは9月にも開催されました.

### 新歓コンパ

新入生歓迎のコンパをしました. 部員がたくさん入ったのと店の選択にちょっと失敗したのとで凄まじい人口密度になりました. ちなみに自己紹介だけで1 時間以上の時間をかけました. みんなキャラが立っています. 流石京大.

## 6月 一般活動……主に勉強. 夏コミに向けて

### OB会

OB の方々のお話を聞いたり宴会をしたりしました. Rubyについて語ってもらったり，小さな飛行船を飛ばしていただいたりしました. KMCは今年で 34 周年. OB の中には「神のみぞ知るセカイ」作者の若木民喜氏や, 大学教員, Google に勤める人などすごい人達がいます.

# 8月 一般活動……夏コミ直前追い込み

## C78

今夏のコミケにも出展しました. コミケの何たるかについてはここで語る必要はないでしょう. 例年より多くのお客様にお越しいただきました. この時にはまだシールや部誌の販売は行っていませんでした.

# 9月 一般活動……各々勉強. NFの準備もしているべきだった……

## コーディング合宿

3 日間ひたすら琵琶湖のほとりの合宿所に引きこもってコーディングをしました. 無言で作業を行い, 参加者内での連絡までIRC(チャットの一種)でするという異様な光景でした. 3Dのゲームやパズルを解くプログラム, 果てはプログラミング言語などそれぞれ思い思いのものを製作しました.

## 夏合宿

ドイツのオクトーバーフェストに行きました. 今年で記念すべき200 周年を迎えるビール祭りです. ジョッキ(巨大)になみなみと注がれたビールを皆で飲んだりしたみたいです. 参加者は全員20 歳以上です.

### KMC野球

文化系サークルであり, その中でも特にひ弱な人達の多いKMCで何を思ったか野球をしました. 流石に18人は集まらなかったので三角ベースです. 無事死人もけが人もなく行うことができました.

## 10月 一般活動……NF 準備

### おでん

寒くなってきたのでおでんを食べました. 大鍋と炊飯器 3 台を用い, 大量の素材を買い込んで振る舞いました. 大量とは具体的に言うと米 16合, 卵26 個, 大根3 本などです. 左の写真はあくまでも一部です.

## 11月 一般活動……NF 追い込み

### NF

京都大学の学園祭であるNF(November Festival)に出展しました.
教室でゲームを遊んでもらったり販売したりしました. C79 で頒布した CD の中身はここで公開したゲームのバージョンアップ版です.

その他, 写真が残っていないイベントたちを紹介します.

### 4月……カレー

新入生歓迎のためカレーとポテトサラダを作って皆で食べました. 大鍋いっぱいのカレーと大量のポテト, サラダが一瞬で平らげられました.

### 7 月……ICPC 国内予選

世界的プログラミング大会の予選です. 京都大学から出場する6チームのうち4チームにKMCのメンバーが所属していました. うち1 チームが地区予選へとコマを進めました.

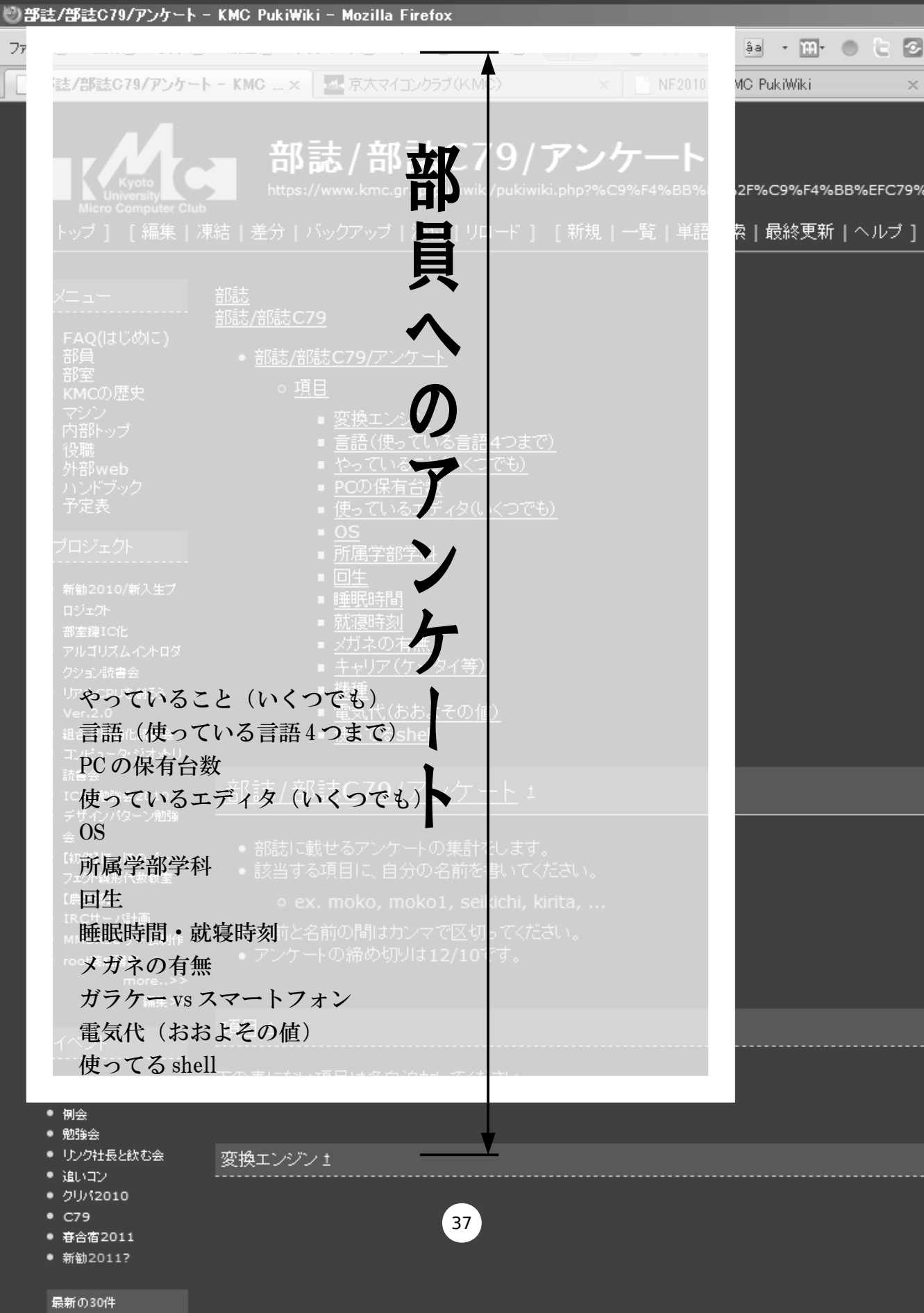
### 12 月……追い出しコンパ

卒業する人を送るコンパです. 原稿執筆時点ではまだ行われていません.

### 最後に

なにか書こうと思ってこの項を作ったものの書く体力が残ってないので一言だけ.

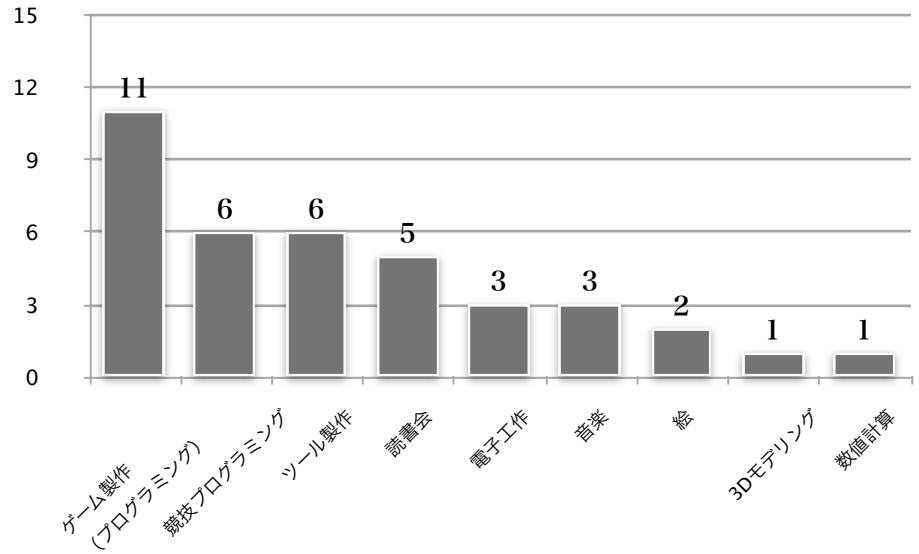
# KMC楽しいよ！


部員へのアンケート
やっていること（いくつでも）
言語（使っている言語 4 つまで）
PC の保有台数
使っているエディタ（いくつでも）
OS
所属学部学科
回生
睡眠時間・就寝時刻
メガネの有無
ガラケー vs スマートフォン
電気代（おおよその値）
使ってる shell

# 部員へのアンケート

kirita

## やっていること (いくつでも)

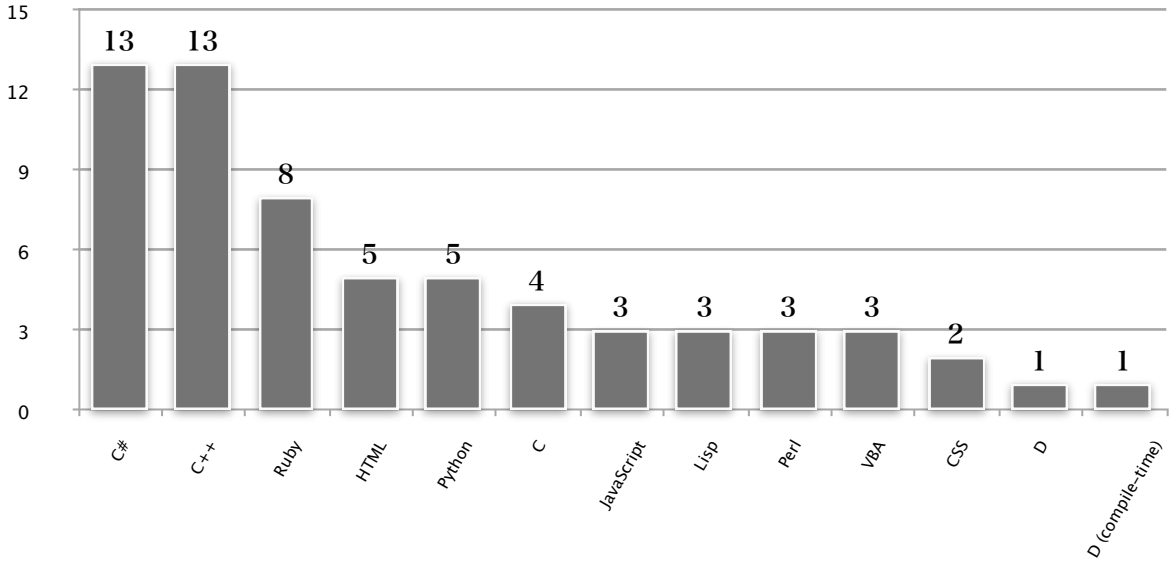



ゲームを作っている人がやはり多いようで, KMCの主流といえるでしょうか. また競技プログラミングも盛んです. コンピュータに関連した技術的なことのほかに音楽, 絵などの活動も行われています. これらは主にゲームの素材として需要があります.

## 言語 (使っている言語4つまで)

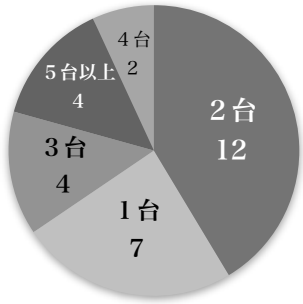
KMCにおいてゲーム開発で主流となっているC#, 競技プログラミングなどあらゆる分野で用いられているC++が同数で1位という結果になりました. スクリプト言語としてはRubyが最も人気だったようです. Ruby関係の著作もある業界の大物がKMC出身だということも関係あるのでしょうか??
同じくスクリプト言語としてPython, Perlが続きます. Perlユーザーは減りつつあるよ

うです. プログラミング言語とは少し異なりますがHTML・CSSを使いこなす人もいます. おなじみC言語は電子工作などで使う人などを除いて常用する人は減っているようです. Javaが全くいなかったのは意外ですね. 全員から回答を得られたわけではないですが, 少数派であるとはいえるでしょう.



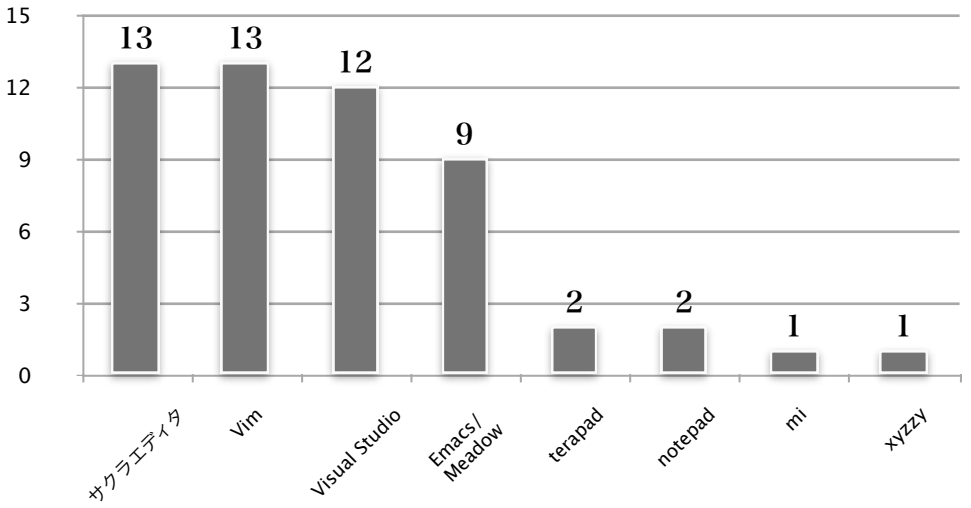

### PCの保有台数

ノート1台, またはそれに加えデスクトップ1台を所有する人が大半を占めていることがわかりました. 自宅サーバーを所有する人がそれ以上の台数を所持しているようです. 中には5台以上も所有するマニアックな人もいるようです.

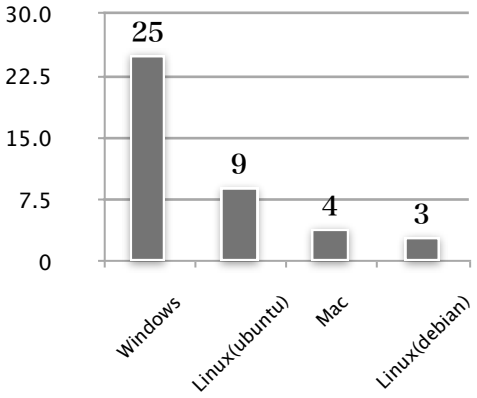



### 使っているエディタ (複数回答可)

Vim > EmacsということでこのアンケートではVimに軍配が上がったようです. 統合開発環境についてはVisual Studioしかありませんがが Javaを使っている人がいないためEclipseも使われていないということでしょうか. ちなみにこの文章は xyzzy で作成されております.



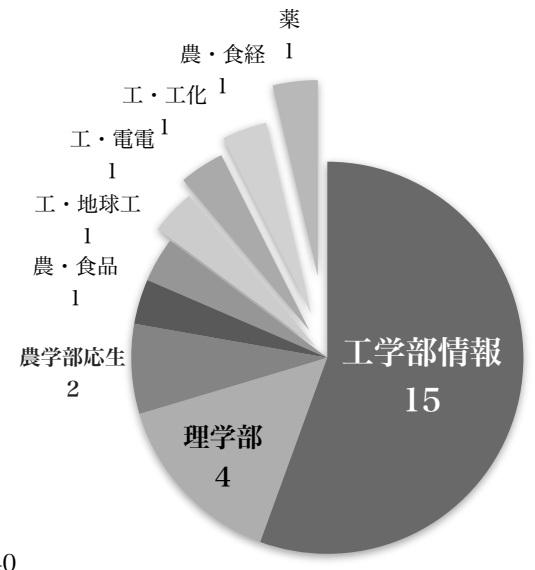

**OS (複数回答可)**



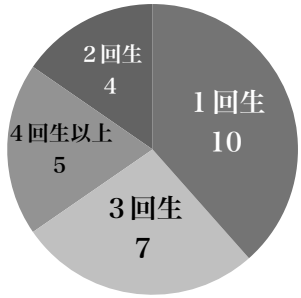

そうはいっても Windows が主流ということのようです. Ubuntuも人気がありますね. Macはやや少ないでしょうか……. Debian はリモートで使う人が多くを占めます.

**所属学部学科**

サークルの性質上情報学科が大半をしめています. そしてほぼ理系です. さらにいうとほとんど男です. KMCでは学部学科回生性別を問わず広く新入部員を募集しております!

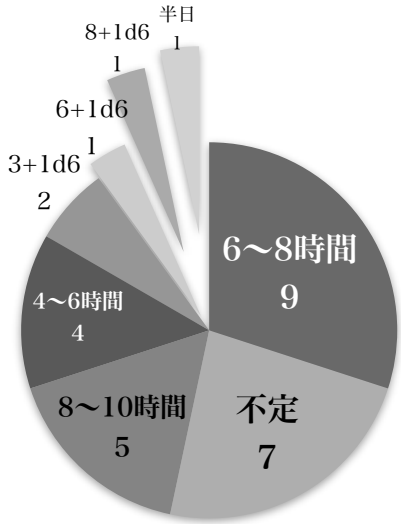


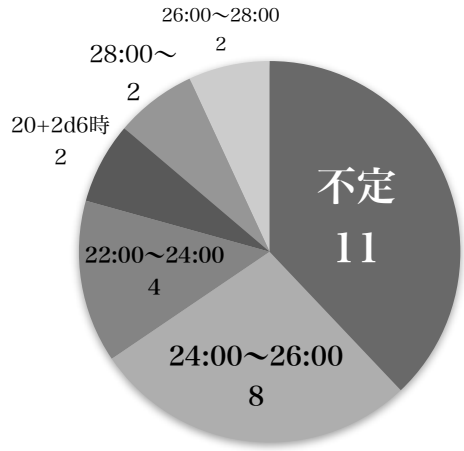
## 回生




今年度 (2010年) はたくさんの新入生が入ったため1回生が多くなってます (新入生といっても1回生には限らないのですが) . 私も新入生です.

### 睡眠時間・ 就寝時刻

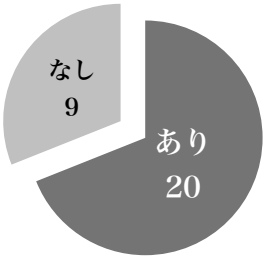



どちらも不定という選択肢が目につきます. 日によってまちまちという答えもあって部員の生活サイクルが心配されます. 標準的な大学生がどのような生活を送っているのかは知りませんが, 多少夜は遅くとも睡眠はしっかりととりたいものですね.

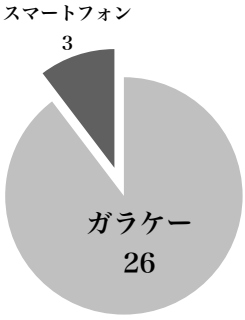
ちなみに, 1d6とか2d6という表記ですが, これは「n面のダイスをm回振ったときに出た目の数の合計」という意味です. つまり, 1d6なら, 「6面ダイスを1回振ったときに出た目の合計」ということになります. KMCでは比較的よく使われる表現で, ゲーム好きの人が多いことのあらわれと言えますね. にしてもダイスで睡眠時間を決めてしまうなんて…….

### メガネの有無




日本人の眼鏡率は大体2人に1人といわれている (らしい) ことを考えるとこの2対1という比率はそれに比べればやや多いようです.

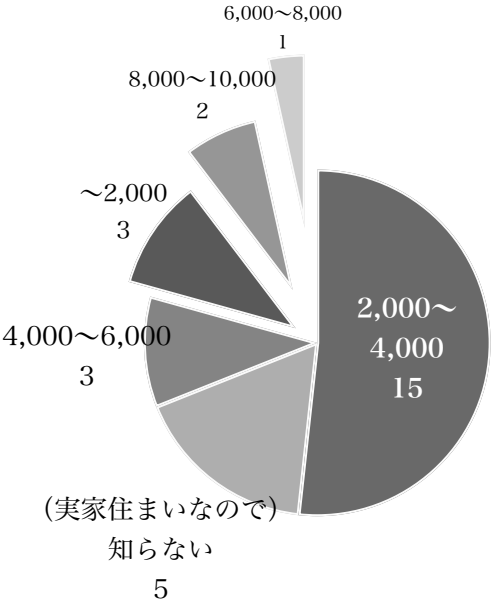
### ガラケーvsスマートフォン




ガラケー圧勝ですね……大抵の部員はノートパソコンを持ち歩いていることが多いので携帯端末にそこまでの機能を求めないということなのでしょうか? それにしてもスマートフォンが少ない.

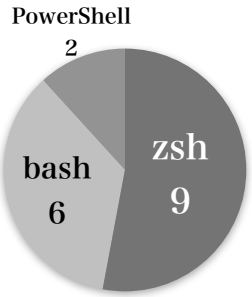
### 電気代 (おおよその値)

常時パソコンをつけているイメージのあるサークル (偏見) ですので電気代も非常識なことになるかと思ったのですがそんなことはありませんでした. ただ中には下宿にサーバを所持していて, 夏場などになると1万円を超える電気代がかかった, ということも聞きます. くれぐれもご注意を……




### 使ってるshell

そもそもこの項目は回答率が少なかったです. 私もあまり分かりません. zshが強力ということで人気のようです.



# 勉強会特集

p44 実践 Common Lisp 読書会
p45 パーフェクト
線形代数教室
p47 PRML 読書会
p48 ICPC 勉強会
p49 組み合わせ最適化読書会
p50 ゆっくりお絵描き
P51 DTM 勉強会
p52 C# でゲームを作ろう 2010
p53 リアル CPU を創ろう
Ver.2.0

# 実践Common Lisp読書会

yukiharu

### 参考書の紹介

タイトル通り使用している参考書は『実践Common Lisp』 (著 PeterSeibel. 共訳 佐野匡俊, 水丸淳, 園城雅之, 金子祐介. オーム社. 2008.) です.

この本では, 第1章「なぜLispなのか? 」に始まり, 前半に Common Lisp の基本的な仕様の紹介, 後半にそれを定着し活用する為のより実践的なコードを書く訓練をすることで, 読者がこれから長く (あるいは今までよりも更に親密に) Lisp系言語と付き合っていくのに必須となる一通りの知識を得られるように構成されています.

ともすれば抽象的な議論に終始しがちなきらいのある他の多くのLisp解説書に対して, 「実用的な何かをするソフトウェアを書くのに言語をどう使えばいいか」という問いを投げかけつつ, 例として挙げられたコードがその解への道筋を示すスタイルが初学者の理解を助けます.

### 読書会の内容

読書会は輪講形式で進められます. 少人数ながらも, 積極的な質問と活発な議論によって参加者一人一人が目的意識を持った活動することが可能です.

ほぼ半年をかけて培った基礎力をもとに, 今秋のNFに前後して各々がテーマを持ってその実現に努め, 来期以降は参考書後半の中核をなすストリーミングサーバの作成を目標としています.

Lisp系言語は現在のプログラミング言語界隈においては, 主流派とは言えない位置にあります. しかしながらその言語としての能力と拡張性 (これはLispの誇るマクロによるところが大きいですが) は他のどのような言語にも劣らず, またこれからもそうありつづけるでしょう.

Lispを学ぶことは決して狭い世界だけで通用する理解を得ることではなく, むしろプログラミングにおける良いセンスと共通理解を学ぶ機会であるとの信念を持ちながら, 当読書会は粛々と活動を続けています.

# パーフェクト線形代数教室

koji

## 紹介と名前の由来

この講座では今まで線形代数を真面目に勉強してこなかった人や一回生向けの講座で, 線形代数を最初からやるというものです.

参加者は, 農学部３人, その他の部員若干名からなります.

講座担当は基本的に私で, ホワイトボードを使って解説をしますが, たまたま居合わせた上回生にTA (TeacherにArekore言う人) に回ってもらうこともあります.

教科書は使用せず, 大学の線形代数再履授業のレジュメを使っています.

酒に酔った私が, 部室で一緒にいた人たちに線形代数のことをしゃべったところから話が発展してこの講座ができました.

講座の名前は一部の方面で有名な某歌のタイトルから拝借したようですが私にはよく分かりません.

## 講座の内容

始めに実数体Rに対して$R^n$という集合を定義し, この元同士の足し算や, Rとの掛け算を定義し, $R^n$数ベクトル空間をつくります. そこから, 部分ベクトル空間, 一次結合, 一次独立, 基底など線形代数を語る上で外せない概念をやります.

こうして数ベクトル空間に慣れ親しんだ後に, 体の定義と抽象的なベクトル空間の定義に移ります.

抽象的なベクトル空間とは, 元同士の足し算や, 体との掛け算が定義され, いくつかの演算法則がなりたつような特別な集合のことです.

話は変わり写像の定義に移ります.

写像とは, 集合Sから集合Tへの対応で, Sの元xを一つ決めるとTの元yがただ一つ定まるという性質を満たすものです. 写像にはfなどと名前をつけて, $f : S \to T$と書いたりします. 写像の例はたくさんありまして, 微分積分で出てくる関数というものはたいてい写像です.

次に単射, 全射の定義をします. 単射や全射は特別な性質を持った写像のことです.

単射とは, Sの任意の異なった二つの元aとbを写像で移した先のTの元$f(a)$と$f(b)$はやはり異なっている, つまり, 違う元は違う元に移る, という性質をもった写像です.

全射とは, Tの任意の元yに対して, Sの元xが存在して, $y=f(x)$となるような写像のことです.

単射であり全射であるような写像は全単射とよばれます. すべての写像が単射に

なったり全射になったりすることはもちろんありません.

単射や全射という写像は, 線形代数に限らずいろいろな数学の分野でよく出てくる重要なものです. ここから, ベクトル空間と写像の話が合流します.

写像とは集合から集合への対応関係でしたが, 線形代数ではベクトル空間からベクトル空間への写像を考えます. さらにベクトル空間からベクトル空間への写像で, 線形になっている写像を線形写像といいます.

この線形写像が線形代数ではベクトル空間と同じぐらい重要な役割を果たすわけです.

今までの内容は上記の通りです. 一般的な線形代数の授業と似たような感じです. ちなみに, 適度に練習問題をはさんで参加者に解いてもらったりしています.

### これからの予定

線形写像をやると次は行列表示です.

線形写像はベクトル空間からベクトル空間への写像のことでしたが, この写像, 線形性という性質を持つが故にベクトル空間の基底を取ることによって, 行列で書くことができるのです. これが行列表示で, この行列を表現行列といいます.

さて, ベクトル空間の基底の取り方は無数にありますが, どの基底をとっても線形写像としての意味は変わりません.

そして基底の取り方によって表現行列は変わります.

となれば, 基底を適当に取り替えて, 表現行列をできるだけ簡単なもの (たとえば対角成分以外は全部0) にしたくなってきます.

これは行列の標準化とよばれます. 標準化には対角化や, ジョルダン標準化などがあります. 行列の標準化と一緒に行列式の理論もやる予定です.

以後はこのように進んでいきます.

# PRML読書会

**@seikichi**

唐突ですがPRMLはご存知でしょうか．えっ？磁気記録の方式[1]？違います，どこの顧問ですかあなたは．

PRML[2] とは「パターン認識と機械学習 - ベイズ理論による統計的予測」[3]の略称です．KMCでは今年度からPRML読書会が始まりました．

この本はパターン認識や機械学習のアルゴリズムや考えをベイズ理論の観点から解説した教科書です．値段が少し高め[4]なのがネックですが，内容は基礎から丁寧に，解説は数式だけでなくカラーの図も載っており，充実した内容となっています．必要な前提知識は学部1年の線型代数と微積分ぐらいかと思います．

私はこの分野に対して事前知識が無い状態で読み始めたのですが，豊富な解説のおかげで何とか理解して読み終えることができました．

内容はKMCの日頃の活動とは余り縁が無さそうですが，この分野に興味を持った部員がある程度集まったため読書会がスタートしました．参加者は現在6人で，各々が1章を担当して読み進めています．

東京で開催されている読書会に比べると参加人数も少なく，各人の負担もかなり高めですが，人数が少ない分活発な議論ができていると思っています．

私としてはそのうち成果物に活かせればいいなと思って参加していますが，まだまだ道は険しそうです．

1 Partial Response Maximum Likelihood

2 Pattern Recognition and Machine Learning

3 ISBN-13: 978-4431100133，978-4431100317

4 上下巻で14300円

# ICPC勉強会

@ir5

## ICPC (International Collegiate Programming Contest) について

皆様はICPCというものをご存知でしょうか．ICPCは1976年ごろから開催されはじめた世界規模のプログラミングの大会で，2009年度の大会では1,931もの大学と約22,000もの人が世界各地で開催された地区予選に参加しました．そのうち103チームが2010年2月に中国で開催された世界大会に進出し，世界一を競いました．[1]

日本国内においては，まず7月ごろに国内予選が開催されます．国内予選で好成績を残すと次は11月ごろに開催されるアジア地区予選に参加します．アジア地区予選でも好成績を残すと最終的に世界大会へ出場する権利を獲得できる，といった風になっています．

大会は国内予選では3時間程度，地区予選や世界大会規模になると5時間にも及びます．また大会は3人で1チームとなって挑みます．3人で1台のコンピュータしか使うことができないため，チームワークを生かすことが重要になってきます．

他の選手よりも良い成績を残し，最初の関門である国内予選や，世界大会への足掛けとなる地区予選を突破するためには日ごろの練習が不可欠となります．

## 勉強会について

勉強会は問題を4〜6つほど選んでそれを90分かけて解いて，後でその検討をする，という構成で行われています．解いている最中はみんな割と黙々とキーボードを打ってる感じなので，勉強会というよりは練習会と言ったほうが近いかもしれません．

問題を引用とする元としてはUVa Online Judgeという，この界隈ではかなり古くからあるオンラインジャッジシステムを使っていました（が，最近サーバーが異常に重いのでそろそろ変えようかという話も……?）．

問題セットは最初の方は僕が選んでいましたが，最近は参加者から選ぶ人を回しています．

もともとは新入生向けのつもりで立ち上げた勉強会だったのですが，入部した頃と比べると実力を付けていっている部員も何人かいて，この勉強会のおかげなのかはあまり定かではないですが役に立っていれば幸いかと思います．これからもプログラミングコンテストに参加する人たちにとって有益な場であり続けることを願います．

1 引用元：http://icpc2010.honiden.nii.ac.jp/outline

# 組み合わせ最適化読書会

@ir5

組み合わせ最適化 (Combinatorial Optimization: Theory and Algorithms) というとても分厚いコンピュータサイエンスの本があります．グラフ理論の基本的な定義〜踏み込んだ内容や，線形計画法などについて書かれたアルゴリズムの集大成のような本で，開くと中に数式がいっぱい書かれていて，見ているだけで疲れそうになります．

そんな少しハードコアな本でも，1人だと読むのが大変でも有志が何人かで集まって読めば進められるのでは? となって始まったのがこの読書会です．たしか2009年の学祭が終わった頃 (11月終盤) にこの本を開いて眺めていたような気がします．

本書は1章の序盤の軽い話題から始まって，2章でグラフ理論の基礎，3〜5章で線形計画法，6章から12章くらいまで本格的なグラフ理論のアルゴリズムに入り，15章あたりからNP問題になります．全部で23章で構成されています．

で，さすがに全部を読むことは厳しいので，最初の基礎的なところは一通りやって，応用的なところで面白そうなところをやっていくという方針で進めました．

勉強会の進め方としては，担当者をあらかじめ決めておき，担当者が当日にスライドで発表する，という形式で行いました．内容の濃い章になるとスライドが何十枚にも重なって貫禄がありました．

### 2章

グラフ理論の初歩の章なのですが，普通のグラフ理論なら比較的多く使用されているはずの図などがほとんど無く，ひたすら抽象的な数式が並べられていて理解するのにとても苦戦した記憶があります．グラフ理論は言葉の定義がかなり豊富にあって定義を追いかけるだけでも大変でした．

### 6,7,8,9章

最小全域木問題，最短パス，最大流，最小費用流など基本的でありつつも重要でかつそんなに簡単じゃないところをやりました．ここまでを2〜3ヶ月くらいでやっていたみたいです．

### 10章，11章

マッチング，重みつきマッチングをやりました．なんとなくNP完全なのかなぁと思っていたマッチング問題が実は$N^3$程度のオーダーで解けるということをこの本で初めて知って驚いていた記憶があります．2ヶ月くらいでやったようです．

今ではやっていたことを忘れてしまって花をつぶしていく程度の記憶しかなくて残念なのでいつかまた読み直して今度は

実装までしてしまいたいところ (しかしそれはいつになるのやら……) .

### 13章

マトロイドという謎の概念をほんのさわりだけやりました．やっていることの抽象度が高いことと，高い難易度の割に具体的な応用例が見えないことからこのへんからだんだん飽きてきてしまった感じがします.

### 3章

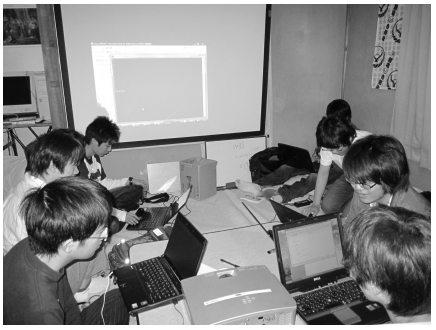
線形計画の章です．面倒だったので飛ばしていたのですが，組み合わせ最適化問題を考えていくと線形計画はどうしても避けることができないようだったのでやることにしました．現在進行中．いつ終わるのでしょうか.

かれこれ1年近く続いていますが，基本的なアルゴリズムはある程度抑えられて良かったのではないかと思います.

# C#でゲームを作ろう2010

**担当者:tsutcho**

C#という言語でゲームを作ることを目標にした勉強会です. 全くプログラミング経験のない人向けにプログラミングの初歩の初歩から勉強しました.



後半は具体的にシューティング, アクション, パズルなど具体的なゲームの作り方を学びました. C78, C79 において頒布した (する) CD に, 勉強会参加者が作成したミニゲームが“Micro Games”として収録されています.

最終的に各々が製作したゲームは以下のとおりです. 勉強会参加者の中には他のプロジェクトで忙しくゲームを作れなかった人もいるのでこれが参加者全員というわけではありません.

tsutcho……CULUQULU (パズルゲーム)

crys……Fairy Puzz (パズルゲーム)

hideya……迷路

inde……Let’s Avoiding! (避けゲー)

プログラミング経験0から約4ヶ月でここまで作れるようになりました. これからも精進していきたいですね.

# リアルCPUを創ろうVer.2.0

**報告者　電気電子工学科2回生 possum**

## Ⅰ. 目的

TD4（とりあえず動作する4bitCPU）の製作を行う. 製作を通じて, 電気電子回路の基礎, 半田付け等の技術を学ぶ.

## Ⅱ. 概要

本プロジェクトは, 担当者 tahi が多忙のため未完に終わったプロジェクト「リアルCPUを創ろう」の後継であり, 新担当者となったpossumがVer.2.0と銘打って新入生プロジェクトの一つに加えたものである.

TD4の仕様を表1に示す. この機能を持ったCPUを汎用IC10個で実現する. この程度ならば, 基盤上に人手で配線するのに十分小規模である.

詳細は, 教科書『CPUの創りかた』（参考文献[1]）を参照のこと.

## Ⅲ. 方法

プロジェクト担当者を含める5人程度で, 回路図から部品配置・配線の設計, 実装, 動作確認を分担して行う.

製作は, 大きく分けてクロック・リセット部, ROM周辺, ALU周辺の順に行う.

**表1: TD4の仕様**

| | |
|---|---|
| 汎用レジスタ | 4bit×2 |
| アドレス空間 | 4bit (16byte) |
| プログラムカウンタ | 4bit |
| フラグレジスタ | 1bit |
| 算術演算 | 4bit加算のみ |
| 動作クロック | 3MHz程度※1 |

※1: 実際に動かす際には, 手動か1Hzか10Hzとしている.

## Ⅳ. 使用器具

配線工具一式, アナログテスタ, 配線デバッグと作業ゲーに対するやる気.

## Ⅴ. 注意

高温になる器具を扱うので火傷, 火災に注意すること. copy & paste, undo は使えない.

## Ⅵ. 実験経過

4月末から週2〜3時間程度のペースで製作しており, 12月17日現在で第14回を迎えた. 現在ROMのセレクタ周辺を製作中である. 詳細についてはKMC活動報告ブログ（参考文献[2]）を参照のこと.

## Ⅶ. 考察

ROMに実施回数14回中9回を既に割いており, あと2回程度続くと考えられる. 残りは中枢部分だが, 年度内には完成したい.

## 参考文献

[1] 渡辺郁: CPUの創りかた, (毎日コミュニケーションズ, 2003)

[2] KMC活動報告ブログ, http://d.hatena.ne.jp/kmc-log/

# あとがき

**hidesys(編集補佐)**

こんにちは. hidesysです. 今回独習KMC Vol.$\beta$において目次と各企画の扉ページを担当させていただきました. あとがきというとこれだけを読んで部誌を買うかどうか決める人もいるわけで, 密室のトリックとか最後のどんでん返しとかバラしちゃいけないんだと思いつつ. むつかしいな.

ところで, 素晴らしく面白いゲームやあなたの生活をもっと豊かにするWebサービスの裏側の人達を想像したことはありますか? KMCは「コンピュータを使って何かをするサークル」. だから, 部員にはプログラマーはもちろん音屋や絵描き果ては競技プログラミング参加者までいろんな種類の人々が集まってます. あなたが利用する製品やサービスの裏側にはこんな人達が居て, そしてそういった人達にも普段の生活があるわけです. この部誌は, 部員がそれぞれの日常を好き勝手に記述した総体みたいな体裁になってます. だからこれを読むと, 京都にあるよく分からないコンピュータ弄りのサークルのことをちょっとだけ知ることができて, 親しみが持てるかもしれません.

とかく, 「KMCに入ってみたい! 」な人から「儂も昔はコードを書いていてのう」な人まで, コンピュータに興味のある様々な人々に楽しく読んでもらえる内容になりました.

この部誌を作るに当たって, 企画立案のMokoさんや編集担当のMokoさんそして予定管理のMokoさん, その他大勢の部員の協力と尽力がありました. お疲れ様でした.

**Moko (編集統括)**

KMC3回生のMokoです. 十ウン年ぶりの部誌の復活に際して, 旗振りやページ構成などを担当しました. もともとは, OBである若木民喜さんが現役だったころ, まだ存在していた会誌の存在を知ったことが, 今回の復刊のきっかけでした (当時は挿絵を担当していた, というpostがTwitterで流れてきたのです) .

しかし, 学園祭が終わってから思い立ったもので, なにせ時間がありませんでした. 当初はTeXでテンプレートを作り, 記事を担当した部員ひとりひとりがレイアウトを組む

ことで, 編集作業を極力抑えようと考えていましたが, 時間のない中で触ったことのないTeXにちゃれんぢしたため挫折してしまいました. 上回生のdis氏がついてくれていながら情けない話ではありますが, 是非次回はTeX習得に挑戦したいものです. 結局は, 今回は部員にプレーンテキストで文章を書いてもらい, 私がMacのPagesというドキュメント作成ソフトでレイアウトを行う方法を取りました.

時間のない中, 理屈抜きで名指しで記事執筆の仕事を投げまくったのですが, 返ってきたものは巻頭言を筆頭に, 実にユニークな文章ばかりでした. みんな自分の趣味全開に走っていて, 魅力がほとばしっています. 誰一人として内容が被ったりしないところが素晴らしい.
文章の魅力は, その巧拙ももちろん要素のひとつではありますが, 何よりも訴えることがあるかどうかが大切です. 訴えることがなければ, いくら美辞麗句を並べたところでそれは空虚なのです. KMC部員は特に「見てくれなんて最後だ, どうでもいい, デザインめんどい」と言う人が多く, 絵描き兼デザイナーとしては時々悲しくなります. しかし, それだけ訴えることがきちんと用意されていると言う事でもあります.

私は絵描きなので, 時々夜中に部室でお絵描きをおっぱじめます. 最近は, 絵の練習をしている人が多く, 私が描きはじめるとなんとなくそんな人達が集まり, 思わず熱中して数時間あまりアドバイスをしつつ描き続けることがあります.
絵を描いていて一番嫌いなのが, 「相手が絵をちゃんと見てくれず褒め殺しにしてくることがある」ことなのですが, ここではそんなことはありません. おかしい時はおかしいと言ってくれます. それに, 「ここがちゃんと描けてねーぞ! 」と気軽に指摘できます. 「ヘタなんて言ったら嫌われるかな」なんて思う必要もないのです. みんな真面目です. そういう真摯な空気がKMCには漂っていて, 脳みそが奮い立つのを感じます. かといって, 真面目一辺倒でもない. こういう空気でないと私は窒息して生きていけないのかも知れません. Tetsu=TaLow 最年長現役部員が巻頭言で書かれている通り, KMCはいまだに学びの場であり, 自己顕示欲の場であり, さまざまなものをくすぐり続けている場なのです.

この部誌は, 今後も発行し, 定期発行に持って行きたいと思っています. 幸い, すでに次回の編集長をしてみたいと言っている人もいます. また, なかなか部室やコミケに来られないOBの方にも毎年のKMCの様子がわかるよう, メーリングリストで部誌を配布する予定です. お楽しみに.

2010.12.23

# これだけは譲れない一言

**書いた人：たくさん**

最後に、個性豊かな部員たちの「これだけは譲れない一言」でしめくくろうと思います.

Moko：ボンちゃんは正義
tsutcho :ドクターペッパーを関西でも売って下さい
tuda：0は自然数です(ｷﾘｯ
possum：美少女文庫はラノベです.
tetsu：2次元の嫁は別腹です.
tobiuo :とびうおだよー
seikichi：seikichiは本名じゃないと何度言ったら……
hideya：俺の娘は絶対可愛いと思うんだ, あぁ娘が欲しい娘娘俺の娘
crys：デンリュウのおなかと俺の単位はいつもふかふか
walkure：Kanonのヒロインは名雪.
hidesys：男の娘は絶対可愛いと思うんだ, あぁ娘が欲しい娘娘男の娘
Deskull：リアル猫分 (not 猫耳娘) が足らん
yukiharu：そういうこともあるでしょうが, そうでないこともあるでしょう.
ir5：肩が凝っていますしアブソルの柔らかい足裏で踏まれたい.
koji：Lispで侵略してやるでゲソ!
アトモナド：センター試験はBASICを捨てて簡潔で直感的な関数型言語を採用すべき.
kirita：柚子レモン (柑橘類)

# 独習KMC Vol.β

**発行：京大マイコンクラブ**

**発行日：2010年12月31日**

編集統括：Moko

編集補佐：hidesys

記事執筆：Tetsu=TaLow, seikichi, koji, ir5, Moko, yukiharu, chir, hidesys, walkure, deskull, hideya, tsutcho, enba, kirita, Chick, tobiuo, possum, crys (順不同)

校正・監修・ツッコミ：All KMC members. **Thank you very much!**

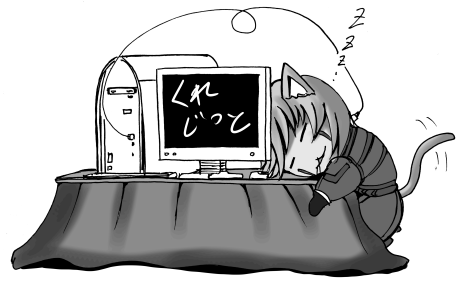
表紙イラスト：crys

挿絵：deskull, Moko

メールアドレス：info@kmc.gr.jp
Web URL　　：http://www.kmc.gr.jp/

落丁・乱丁は, 在庫がある限り交換いたします.

上記のメールアドレスまで, ご連絡ください.



# 独習KMC vol. β

■利用分類

部誌

■製品分類

KMC

■レベル

初級　中級　上級

20101226

20101231

Kyoto University Micro Computer Club