

# 県庁で大活躍(予定)！

## OpenCOBOL★Perlで 汎用機ダウンサイジングに挑んだワケ

村部 淳也



<http://www.lancard.com/>

# アジェンダ

---

- 会社紹介／自己紹介
- なぜなにOpenCOBOL
- 経緯とか技術とか
- ソリューションの方向性
- 現在／未来

# 会社紹介

---

- 有限会社ランカードコム
- 所在地
  - 長崎県長崎市扇町33-40
  - 東京都新宿区四ッ谷1-10-2-408
- 業務内容
  - 長崎県にてOSSを活用した業務提案
- 主要顧客
  - 長崎県庁、他自治体など

# 自己紹介

---

- 村部 淳也
  - アクセス
    - Twitter:[@murave](#)
    - Mail:[murabe@lancard.com](mailto:murabe@lancard.com)



- 所属コミュニティ
  - Rubyビジネス・コモンズ (スタッフ)
  - 他、いろんなところに出没

# 質問

---

# 質問

---

- OpenCOBOLを使っている方？

# 質問

---

- OpenCOBOLを使っている方？
- 汎用機メーカーの方？

# 質問

---

- OpenCOBOLを使っている方？
- 汎用機メーカーの方？
- MicroFocus COBOL等のCOBOLコンパイラを扱っている方？



なぜなにOpenCOBOL

なぜ汎用機から  
ダウンサイジング  
したいのか？

汎用機の氣に存る  
ありこれ

---

# 汎用機の気になると あれこれ

---

- ・ 高い維持費

# 汎用機の気になる あれこれ

---

- ・ 高い維持費
- ・ 特定業者への依存

# 汎用機の気になる あれこれ

---

- ・ 高い維持費
- ・ 特定業者への依存
- ・ 拡張性の低さ

# 汎用機の気になる あれこれ

- ・ 高い維持費
- ・ 特定業者への依存
- ・ 拡張性の低さ

➔ コストパフォーマンス  
と自由度の低さ

リフトウェアはOSSに  
ハードウェアはPCに



ソフトウェアをOSSに  
ハードウェアはPCに

---

- ・コストダウン

# ソフトウェアをOSSに ハードウェアはPCに

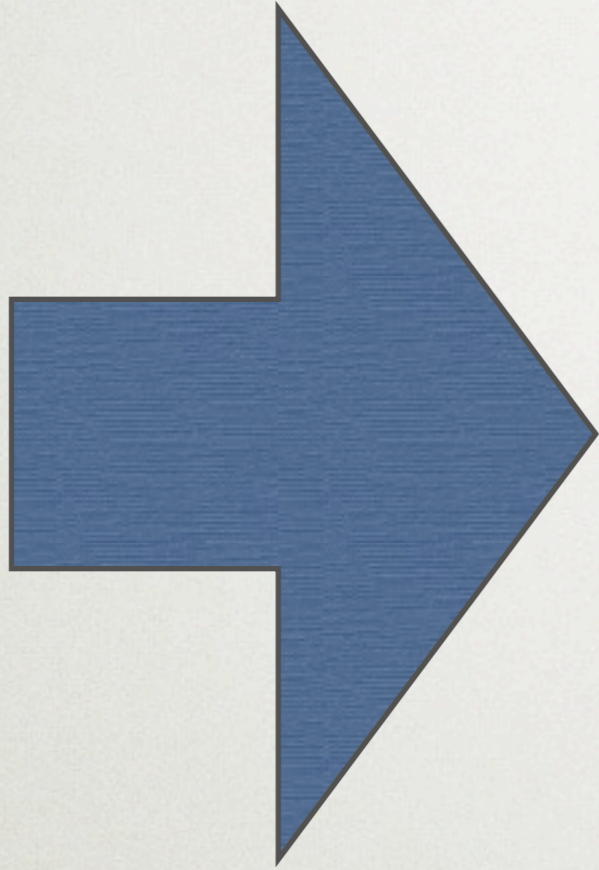
---

- ・コストダウン
- ・高い自由度・拡張性

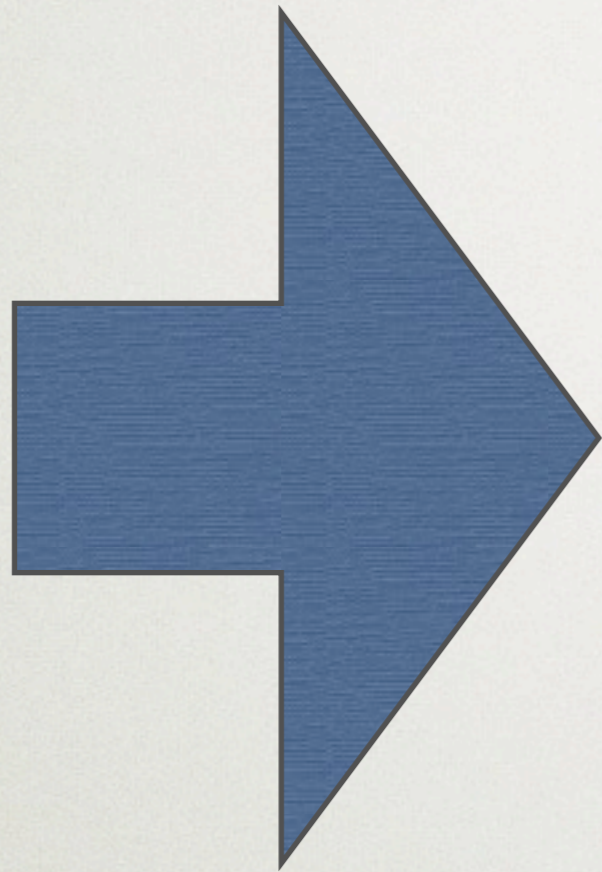
# ソフトウェアをOSSに ハードウェアはPCに

---

- ・コストダウン
- ・高い自由度・拡張性
- ・特定業者へ依存しない



解  
決!



解決!

志が叶ったから実現できる？

例えば、こんな  
中々面白い話

# よくある風景

---

# よくある風景

---

- ・ 業務を変えたいのでシステムもそれに合わせて少し変えたい



# よくある風景

---

- 業務を変えたいのでシステムもそれに合わせて少し変えたい
- システムの拡張性の低さが問題に

# よくある風景

---

- ・ 業務を変えたいのでシステムもそれに合わせて少し変えたい
- ・ システムの拡張性の低さが問題に

「システムをつくり直そう！」

新しい言語で！

---

# 新しい言語で！

---

- ・ 「せっかくだからオレは新しい言語を  
選ぶぜ！」

# 新しい言語で！

---

- ・ 「せっかくだからオレは新しい言語を選ぶぜ！」
- ・ 別の言語で既存システムと同じ機能を再実装

# 新しい言語で！

---

- ・ 「せっかくだからオレは新しい言語を選ぶぜ！」
- ・ 別の言語で既存システムと同じ機能を再実装
  - ・ 不具合混入

# 新しい言語で！

---

- ・ 「せっかくだからオレは新しい言語を選ぶぜ！」
- ・ 別の言語で既存システムと同じ機能を再実装
  - ・ 不具合混入
  - ・ 膨れ上がる工数

# 業務知識はどこに？

---



# 業務知識はどこに？

---

- ・ 昔は業務の全てを知っている人がいた

# 業務知識はどこに？

---

- ・ 昔は業務の全てを知っている人がいた
- ・ 今はシステムを使う事自体が業務になっている

# 業務知識はどこに？

---

- 昔は業務の全てを知っている人がいた
- 今はシステムを使う事自体が業務になっている
- 運用者が業務のロジックを知っていると  
いうのは幻想

# 業務知識はどこに？

---

- 昔は業務の全てを知っている人がいた
- 今はシステムを使う事自体が業務になっている
- 運用者が業務のロジックを知ってるというのは幻想
- ノウハウはシステムの中にある

# 蓄積されたノウハウ

---

# 蓄積されたノウハウ

---

- ・ 既存システムに蓄積された業務ロジックを解析する事になる

# 蓄積されたノウハウ

---

- ・ 既存システムに蓄積された業務ロジックを解析する事になる
- ・ 既存システムのソースコードから仕様書を起こし直してみたり

# 蓄積されたノウハウ

---

- ・ 既存システムに蓄積された業務ロジックを解析する事になる
- ・ 既存システムのソースコードから仕様書を起こし直してみたり
  - ・ 解析ミス、誤った認識



# 蓄積されたノウハウ

---

- ・ 既存システムに蓄積された業務ロジックを解析する事になる
- ・ 既存システムのソースコードから仕様書を起こし直してみたり
  - ・ 解析ミス、誤った認識
  - ・ 進まぬ構築、膨れ上がる工数

事りかむる目的

---

# すりかわる目的

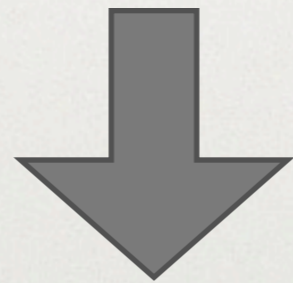
---

「業務のためにシステムをつ  
くり直したい！」

# すりかわる目的

---

「業務のためにシステムをつくり直したい！」



「どうにかしてシステム構築を終わらせないと…」

# 本来の目的

---

# 本来の目的

---

業務改革・改善

# 本来の目的

---

## 業務改革・改善

システムをつくり直す事が  
目的ではなかったのでは？

必要なのは  
力ーザンな環境



必要なのは  
オープンな環境

---

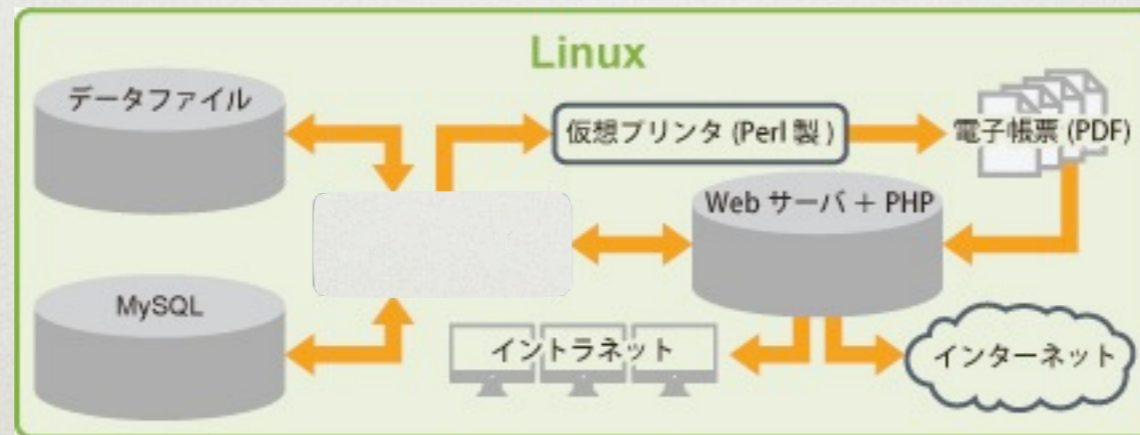
既存のコードをOSSで

動かせれば最高！

# オープンな環境といえは

## 「LAMP」

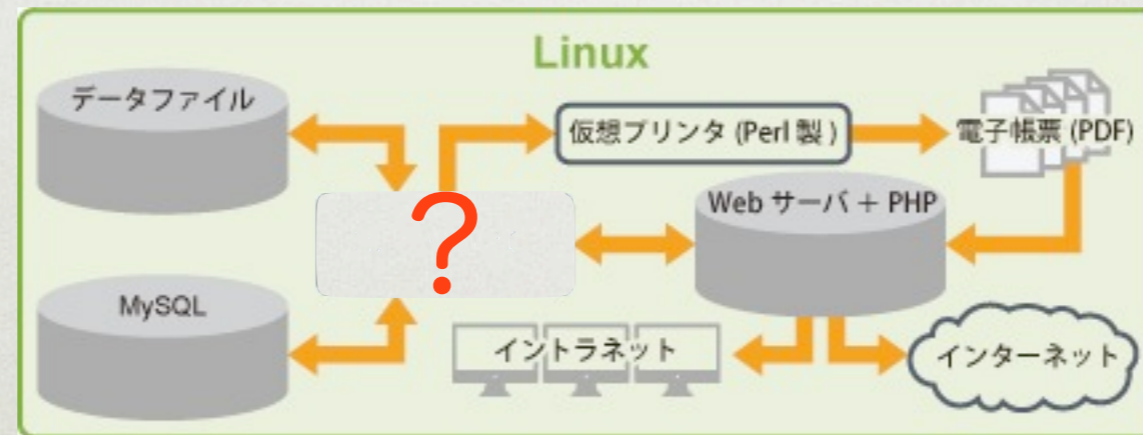
Linux+Apache+MySQL+[Perl,PHP]



# オープンな環境といえは

## 「LAMP」

Linux+Apache+MySQL+[Perl,PHP]



「なにかが足りない」

# 汎用機といえは**COBOL**

---

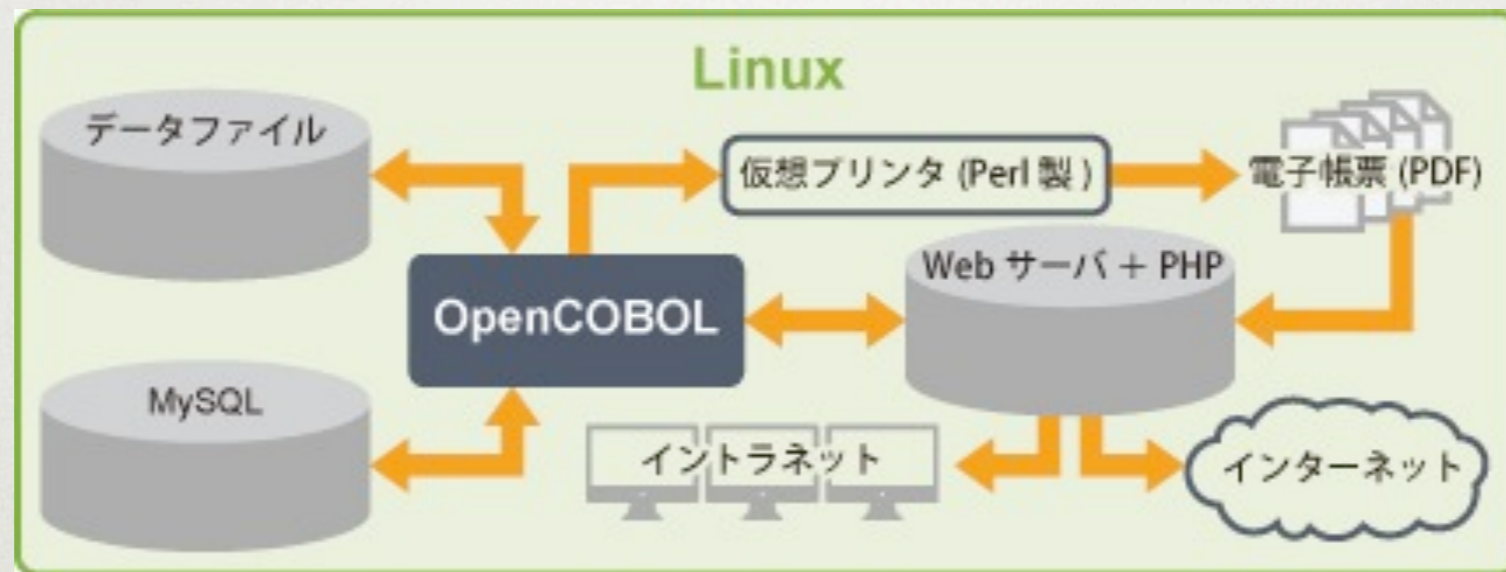
**OSSのCOBOL**があれば

**オープン**な環境で

**既存システムのリソースを**

**生かせる！**

# OpenCOBOL !



- 日医標準レセプトORCAで使われているOSSのCOBOL
  - ORCAについては <http://www.orca.med.or.jp/> 参照
  - ORCAでの稼働実績は十分（1万以上稼働中）  
<http://www.orca.med.or.jp/orca/nintei/kadou.rhtml>

↑ 4

OL!

経緯とか技術とか

# 主な登場人物

---



お客様



弊社M氏



# 事のはじまり

---



「COBOLで動いてるシステムを他の言語でつくり直すのって馬鹿らしい」

「かといって現状のままメーカー丸投げとかありえない」

「OSSのCOBOLってないの？」

# あったのです

---



「日医標準レセプトORCAで使われているOpenCOBOLがありますよ」

「ORCAでの稼働実績は十分ですね」

「オープンソースなのでコンパイラを改造できますし、イケそうです」



「ダウンサイジングやってよ」

---

かくして

汎用機との戦いが  
はじまった

# 今更ですがOpenCOBOLとは

---

- COBOLからC言語へ変換してコンパイル
- 元開発者は 西田 圭介氏
- 現開発者はRoger While氏
- 現安定バージョンは OpenCOBOL 1.0
- OpenCOBOL 1.1ではncursesを使用した画面制御機能の拡充とデバッグ機能の強化等
- ライセンスはGPLv2。OpenCOBOL 2.0ではGPLv3へ移行？



# 今更ですがOpenCOBOLとは

- COBOLからC言語へ変換してコンパイル
- 元開発者は 西田 圭介氏
- 現開発者はRoger While氏
- 現安定バージョンは OpenCOBOL 1.0
- OpenCOBOL 1.1ではncursesを使用した画面制御機能の拡充とデバッグ機能の強化等
- ライセンスはGPLv2。OpenCOBOL 2.0ではGPLv3へ移行？

OSSコンソーシアム  
OSS COBOL WG  
OpenCOBOLセミナー  
懇親会にて弊社峰松と



# 海外での実績

---

COBOL-IT 社 ( <http://www.cobol-it.com> )

汎用機ダウンサイジング専門

- 実績

- フランス財務省

**MicroFocus** → OpenCOBOL

- ロイヤル・アホールド (英国：世界3位の小売業者)

**IBM COBOL** → OpenCOBOL

- FlyBuys (オーストラリア：カード、買物ポイント付与)

**Fujitsu COBOL** → OpenCOBOL

# 長崎県での実装環境

---

- 既存環境：

- 国産メーカー汎用機
- 専用ページプリンタ
- 専用端末エミュレータ

- 移行先：

- Linuxサーバ数台（業務種類毎に1台ずつ）

# 長崎県での作業対象

---

- 職員総合、財務会計、予算編成の3システム
- COBOLプログラム
  - 合計で約500万行、約1万5千本
- JCL（ジョブコントロール言語）
  - 合計で約100万行、約6500本
- 帳票レイアウト
  - 職員総合システムで約400本
  - 他システムは未知数（同数程度、計1200本？）





「COBOLソース、少し整形しないとコンパイル通らないな」





「COBOLソース、少し整形しないとコンパイル通らないな」



「ソースコードを自動変換するプログラム作った！」

# 既存COBOLソースを自動変換

- 自動変換のメリット
  - 業務ロジックを変えるわけではないので、ほとんど機械的に置き換えが出来る
  - 自動変換でカバーしていない無いパターンは都度パターン追加
    - 移植ノウハウの自動化



「実はCOBOLだけじゃなくってJCLっていうバッチ処理用の言語があるんだ」





「実はCOBOLだけじゃなくってJCLっていうバッチ処理用の言語があるんだ」



「JCLをほぼそのままPerlで動作させる仕組みを作った！」

# JCLの移植

---

- 特定汎用機メーカーのJCLのOSS実装は無い  
→動作環境を新規開発するしかない

でも新たに処理系作りたくない

- お客様の要望により大幅な改造は無理
  - 文がセミコロンで終端していて、
  - LABEL : で定義されたラベルにGOTOで飛べて
  - 強力な文字列置換機能を持ち
  - マルチスレッドで動作して
  - 大文字のキーワードが使えて
  - ヒアドキュメント機能があって
  - プログラムが割と多くて
  - 動作が安定しており実績があって仕様も枯れた言語

ということでPerl

# JCLの移植例 (JCL)

## JCLでのバッチ処理の記述例

- ▶ ¥JOR OUTID=SPR OUTDEV=PR/M050/JSTD CLASS=A ;
  
- ▶ ¥COMM 月次パラメタSTORE;
- ▶ ¥INPUT ACCEPT1,TYPE=DATA,LIST=YES;
- ▶ AIDF0270 1 4210918 例月(1) +△+支給日
- ▶ AIDF0271 4210822 4210918 償還完了者リスト抽出期間(自~至)
- ▶ \*\*\*\*\* \* 前回支給日の翌日~今回支給日
- ▶ ¥ENDINPUT;
  
- ▶ ¥CREATESEQ
- ▶ INFILE=(ACCEPT1 FILESTAT=SYSIN )
- ▶ OUTFILE=(AID.F089K RECSIZE=80 BLOCKSZ=11440 RECFORM=FB BPB=4);
  
- ▶ ¥COMM 引去金ファイル作成;
- ▶ AIDM2510:¥STEP AIDM2510 FILE=AIA.IL1 DUMP=DATA;
- ▶ ¥ASSIGN AIDF0010 AID.F101 SHARE=ALL HOLDMODE=NO;

# JCLの移植例 (perl)

perlでのバッチ処理の記述例

```
▶ #!/usr/bin/perl
▶ use strict;
▶ use warnings;
▶ use JCL;

▶ JOR "OUTID=SPR OUTDEV=PR/M050/JSTD CLASS=A ";

▶ COMM "月次パラメタSTORE";
▶ INPUT "ACCEPT1, TYPE=DATA, LIST=YES", <<_EOT;
▶ AIDF0270 1 4210918          例月(1) + Δ + 支給日
▶ AIDF0271 4210822 4210918    償還完了者リスト抽出期間(自~至)
▶ *****                  * 前回支給日の翌日~今回支給日
▶ _EOT
▶ ENDINPUT;
▶ CREATESEQ
▶ "INFILE=(ACCEPT1 FILESTAT=SYSIN )",
▶ "OUTFILE=(AID.F089K RECSIZE=80 BLOCKSZ=11440 RECFORM=FB BPB=4)";

▶ COMM "引去金ファイル作成";
▶ AIDM2510: STEP "AIDM2510 FILE=AIA.IL1 DUMP=DATA";
▶ ASSIGN "AIDF0010 AID.F101 SHARE=ALL HOLDMODE=NO";
```





「帳票をどうしたものか」





「帳票をどうしたものか」



「使ってたレイアウトファ  
イルでPDFを生成できる  
仮想プリンタ作った！」

# 帳票資産の活用

- COBOLプログラムと連動した帳票の移行
  - 改行、改ページ等の処理についてのルールがCOBOLプログラムに埋まっている
    - 例：連続した同じデータは1ページに纏める等
      - なるべくそのまま移植したい
  - 帳票レイアウトデータは罫線位置などデザインの情報が記載されている
    - 線の開始位置、長さ、太さ等

# 帳票資産の活用

- 汎用機プリンタエミュレーションの利点
  - プログラム内のアルゴリズムをそのまま活用
    - 改行、改ページのロジック等
  - 帳票レイアウトの再デザインが不要
    - 現在の運用を変えないまま移行可能

# 移植作業

- COBOLソースの変換とコンパイル

```
$ acos2oc.pl COBFILE.txt
```

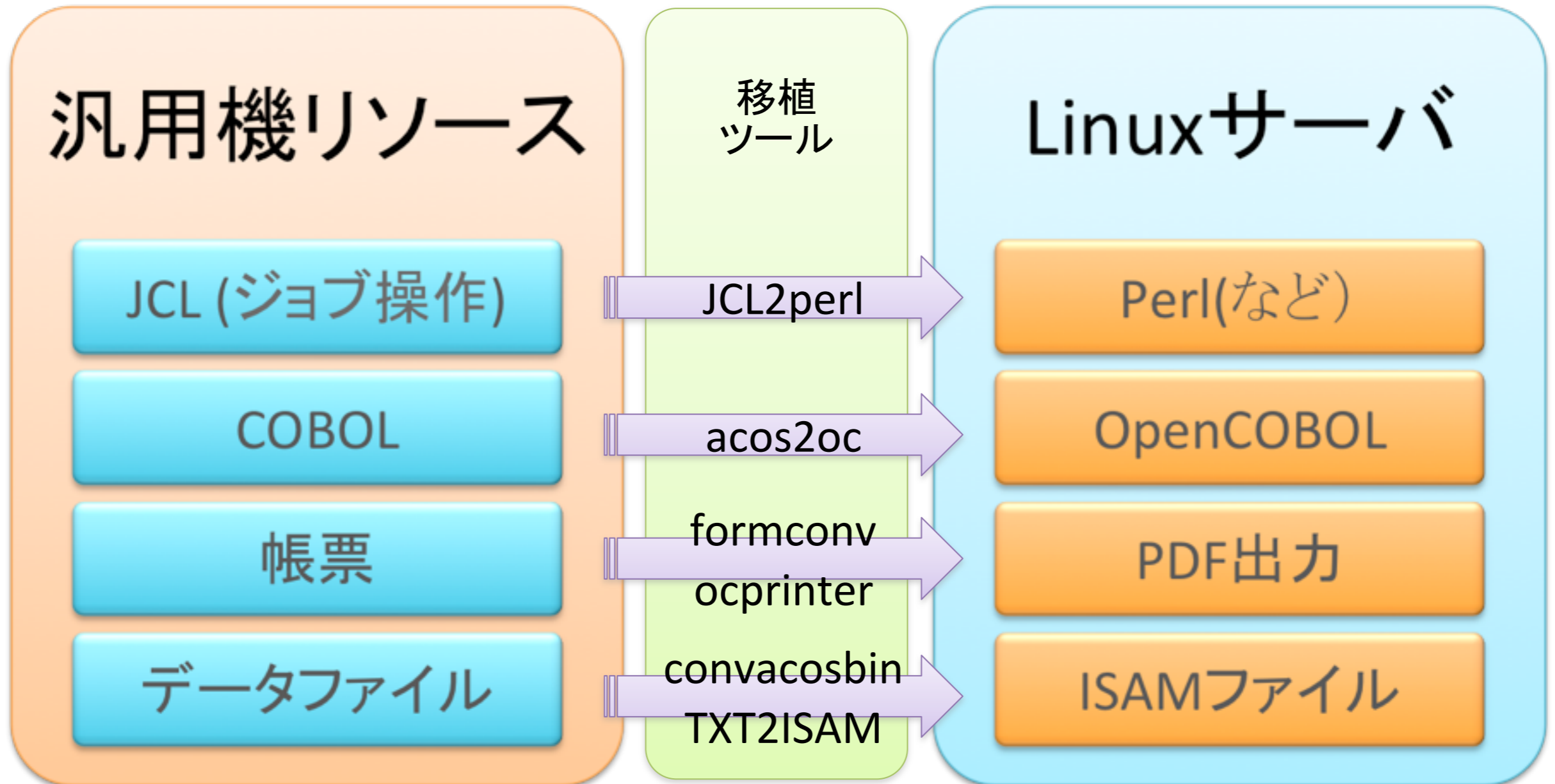
- \* JCLの変換

```
$ JCL2perl.pl JCLFILE.txt > JCLFILE.jcl
```

- \* 帳票レイアウトファイルのコピー

```
$ cp FORMFILE.txt form/FORMFILE.form
```

# 移行イメージ



# 移植ツールについて

- JCL2perl
  - バッチジョブ用JCL言語をPerlスクリプトへ変換するトランスレータ
- acos2oc
  - 汎用機COBOLプログラムをOpenCOBOL用に自動書き換えと依存性を考慮したコンパイルを行う。長崎県庁にて汎用機ダウンサイジング用にカスタマイズしたOpenCOBOLと組み合わせることで90%以上のソースはそのままコンパイル可能。
- formconv
  - 汎用機用帳票レイアウトファイル内の機種依存文字変換ツール
- ocprinter
  - 汎用機用帳票レイアウトファイルを直接解釈しフォーム描画を行う。またOpenCOBOLが生成した印刷用中間データをレンダリングしてPDFを生成する仮想プリンタ
- convacosbin
  - 汎用機用ファイルのEBCDICデータとLinux側のSJISデータの文字コード変換を行う。データファイルのレイアウトを考慮しバイナリ、数値、符号付き数値、パックスド10進、漢字の個別レイアウト指定可能
- TXT2ISAM
  - 固定長データをISAM形式データファイルへ変換する。このISAM形式データファイルはOpenCOBOL、C言語、Perl、JCLから操作可能



「COBOLからMySQLに  
アクセスしたい」







「COBOLからMySQLに  
アクセスしたい」



「Perlを介してアクセス  
可能にした！」

# OpenCOBOL+Perl+MySQL

---

- COBOLは固定長のバイト列を扱う
- Perlで固定長のバイト列作ってやれば  
いい？
- MySQLとのやり取りをperlに任せれば  
OK?

# OpenCOBOL+Perl+MySQL

---

- OpenCOBOL側記述例

```
*      OPEN          INPUT          IN4-F.  
CALL  "cob_perl_require" USING "COB_PERL.pm"  
CALL  "cob_perl_call"  USING "mydb_open"  
      "IN4-F" "AIACF050".  
CALL  "cob_perl_results" USING  
      COBPERL-STATUS LN010-FSTAT1.
```

～略～

```
CALL "cob_perl_close".
```

# OpenCOBOL+Perl+MySQL

---

- Perl側記述例

```
sub mydb_open {
    my ($fh, $dbname) = shift;
    eval {
        require $dbname.".pm";
    };
    my $db;
    eval "my $db = new $dbname()";
    $COBPERL_DB::dbnames{$fh} = $db;
    $db->{'dbname'} = $dbname;
    $db->{'status'} = "00";
    $db->{'fstatus'} = "00";
    $db->Open;
    ($db->{'fstatus'}, $db->{'status'});
}
```

# Perl、魔法の宝石

---

- Perlを介してもアクセス速度は十分！
- COBOL・ストレージ間に層を増やすことで様々な拡張が可能に！
- COBOLからPerlを呼び出すモジュールは様々な応用が可能。ソリューションに強力な拡張性を付与！

# 例えばバーコード生成

- OpenCOBOL

```
CALL "cob_perl_require" USING "barcode.pl".  
CALL "cob_perl_call" USING "barcode"  
    "sample-barcode.png" "1234567890".
```

- Perl

```
#!/usr/bin/perl  
use GD;  
use Barcode::Code128;  
  
sub barcode {  
    my ($filename, $value) = @_;  
    my $barcode= new Barcode::Code128;  
    $barcode->text(FNC1, $value);  
    open(PNG, ">$filename") or die "Can't write $filename: $!\n";  
    binmode(PNG);  
  
    print PNG $barcode->png("CODE 128");  
    close(PNG);  
}  
1;
```





「PHPからCOBOLの□  
ジック呼びたい」





「PHPからCOBOLの□  
ジック呼びたい」



「PHPのモジュール作っ  
た！」



# php+OpenCOBOL

---

- PHP側記述例

```
if(!extension_loaded('opencobol')) {  
    dl('opencobol.so' . PHP_SHLIB_SUFFIX);  
}  
$err    = opencobol_init();  
$handle = opencobol_load("/tmp/TEST0001.so");  
$err    = opencobol_call(  
    $handle,  
    "TEST0001",  
    array(  
        "WK-KETA" => "9876543",  
        "WK-A"    => "ABCDEFGHIJKLMNOPQRSTUVWXYZ" ));
```



「メーカーCOBOLではOK  
だったのにOpenCOBOL  
でコンパイルが通らない」





「メーカーCOBOLではOK  
だったのにOpenCOBOL  
でコンパイルが通らない」



「通るようにコンパイラを  
改造した！」

# メーカー独自拡張への対応

- 非標準なためそのままでは対応出来ない  
→ OpenCOBOLを機能拡張

例：印刷関連など

1カ所の機能追加でかなりの効果がある



「コンパイルは通ったけど  
メーカーCOBOLと少し動  
きが違う」





「コンパイルは通ったけど  
メーカーCOBOLと少し動  
きが違う」



「同じになるようにコンパ  
イラを改造した！」

# 動作の違い・問題点の例

---

– データ型、演算精度の違い

- COMP-1 浮動小数点、丸め誤差

– 文字の取り扱い

- 全角文字列、SPACEの解釈

– 文字罫線

- 外字領域にマップされた文字罫線

– 外字

- 地名、人名、特殊記号



「機能がたりない」







「機能がたりない」



「機能追加した！

しまくった！」

# OpenCOBOLの実際

---

- 現在のOpenCOBOLの機能はかなり不足している
    - PIC N項目(NATIONAL項目)が PIC X項目の2倍のbyte数確保してくれる程度のL10N
    - 半角カナ変数名、漢字変数名には未対応
    - 印刷出力も単純なラインプリンタのみ対応
    - 通貨記号に「¥」が使えませんでした
    - 海外では業務利用例を見かけるが、国内ではORCAでの利用を除けば学習用の位置づけ
- 改造しないと業務には使えない

# 長崎県版OpenCOBOL

- CHARACTER TYPE、COLUMNでの印刷文字種、位置指定機能
  - DATE IS NATIVE指定時に日付を和暦にて返す機能
  - COB\_DATE環境変数設定時にACCEPT～FROM DATE等で設定値を返す機能
  - SPECIAL-NAMESで定義した外部値の取り込みと吐き出し
  - RETURN-CODEとは独立したSYSTEM-STATUS値
  - DUMMYファイル機能(/dev/nullのような機能、ISAMでも利用可)
  - SEQ形式（固定長）ファイルのレコード長、キー長チェック機能
  - PIC XからPIC NへのMOVE時に半角→全角の自動変換
  - PIC NへのMOVE時に全角空白での補完
  - 固定長ではなく改行コードをレコード区切りとする可変長入力データ形式対応
  - SETでのファイル状態のコピー
  - PIC NにてALL SPACEを指定した場合に全角空白でのフィル
  - COPY～REPLACINGでの部分一致置換
  - 半角カナ変数名の使用
  - データハンドラの外部化によるmysql他データベースとの連携
  - 印刷出力時の行について絶対位置指定 ・ 印刷出力時に行復帰コードの自動挿入
  - ファイル名の代わりに URI渡しが可能
  - 2次キーの取り扱いについて柔軟な対応
  - STRING ～ DELIMITED BY SPACE 等 STRING文の日本語対応
- ・・・などなど

# OSS的開発手法を導入

---

- プロジェクトはredmineで管理
- ソース、バイナリはsvnで配布
- 長崎県職員と複数の業者メンバーが参加
- 開発環境が全てOSSなので参加各社の社内でコンパイル、実行、帳票デザインが可能になった
- 全体的な開発コストが低下

# ソリューションの 方向性

# 私たちの方針

---

- ・コア側でなるべく対応
- ・ソースコードやレイアウトファイルは出来る限り再利用

# 自動変換!!



ビジネスロジック
汎用機 COBOL プログラム 汎用機 JCL バッチ
帳票
汎用機用帳票レイアウト
データ
データファイル

ユ一ヨコヨクノ

ビジネスロジック
OpenCOBOL <sup>®</sup> プログラム Perl バッチ
<small>※ OpenCOBOL とは OSS の COBOL コンパイラであり、ソースコードも公開されており自由に使用できます。 日本医師会のレセプトソフト ORCA 等での稼働実績もあり安心です。 Windows、MacOSX 等 Linux 以外でも動作します。</small>
帳票
電子帳票 (PDF)
データ
データ ファイル + リレーショナル データベース

コトノノ + ユ一ヨクノ

# 得之礼百效果

---



# 得られる効果

---

- ・ 全体の工数を劇的に削減
- ・ ソースコード手作業移植  
による不具合増加を防止

# Open COBOLだから

---

- ・ 「汎用機のコストパフォーマンスと自由度の低さ」をPC+OSSへダウンサイジングすることで解決できる
- ・ 既存資産を活用できる
  - ・ 同じモノを再実装するのは無駄だし危険

現在／未來

# 長崎県は並行運用中

- 既存業務、他業務とのデータ連携
  - DBと汎用機系とのデータ相互変換
- データ突き合わせ
  - 移行元と同じ入力データから同じ出力データが得られるか？
- 処理速度
  - バッチが予定時刻までに処理完了するか？

# 移植作業の実際

- 汎用機COBOLから、OpenCOBOLへの移植
  - 自動化しているためコンパイルを通す段階  
まではほぼノータイム
  - 例：200ソースの変換とコンパイルまで2時間
- JCLの移植
  - 自動化しているため移植自体の工数は少ない

# 移植作業の実際

- データ移行

- レイアウトを考慮しながらの変換作業

- FILLER等の未使用と思われた箇所にデータが埋められている場合がある

- 基本的に手作業

- 帳票の移植

- メーカー製の帳票マクロプロセッサ（NL2等）の機能をどれだけ使っているか？

# 移植作業の実際

- 移植作業工数(イメージ)

- COBOL 5%

- JCL 5%

- データ 50%

- 帳票 40%

- 検証作業に一番時間が必要

# 長崎県以外への適用

- 関西方面某県において県JAの会計システム
  - 元環境：N社オフコン+Oracle
    - 80画面、250プログラム 30万ステップ
  - 新環境：OpenCOBOL-1.1 +Oracle
    - 画面はOpenCOBOL画面制御によるキャラクターベースのメニュー（元プログラムを活用）
    - PostgreSQLを検討、組み込みSQL用ツールも作成もしたが工期の都合で置き換えなかった。



# とりまく環境

- ユーザ企業による活動が活発化
  - OSSコンソーシアム OSS COBOL ワーキンググループの結成など
- OpenCOBOLを使用したCOBOL案件の求人
  - 国内での**本格利用**が増えている

# 将来こうなる (?)

---

- ファイル名をURI渡し出来るように改造
  - misam://directory/file (実装済)
  - mseq://directory/file (実装済)
  - mysql://servername/db?table= (実装済)
  - hdfs://resource (実装中)
  - 等 各種データ形式を直接操作
- C言語以外のソースへの変換 (Perl, Scala, Java 等)
- クラウド環境での動作、等々

# ありがとうございました ご清聴

汎用機ダウンサイジング

お問い合わせ ☎ 095-840-0021 LANCARD.COM

汎用機ダウンサイジングとは 資料・記事 具体的な構成例 FAQ

ダウンサイジングで悩んでいませんか?  
その悩みに  
お応えします。

汎用機の維持費が  
つらい……

ダウンサイジングで  
システム全部作り  
直しになっちゃうの?

パソコンの常識が  
通用しない……

ダウンサイジングを  
検討中だけど、どうし  
たら良いんだろう?

パソコンで動作するLinuxにダウンサイジング!

コンピュータの性能向上やコストダウンの恩恵をダイレクトに享受できるパソコンはダウンサイジング先として最適です。トラブルが発生した場合でも、パソコンの知識で対応することが可能です。

既存資産をリサイクル!

既存システムには業務ノウハウが詰まっています。それを捨ててイチから作り直すなんて無駄以外のなにものでもありません。変換して活用しましょう!  
本ソリューションでは脅威の自動変換率で既存リソースを最大限に活用することができます。

自動変換!!

Linux

データファイル MySQL 仮想プリンタ (pdf製) 電子帳票 (PDF) Webサーバ+DB インターネット

OpenCOBOL

ビジネスロジック

OpenCOBOLプログラム Perl バッチ

ビジネスロジック

汎用機 COBOL プログラム 汎用機 JCL バッチ 帳票 汎用機帳票レイアウト データ データファイル

帳票

電子帳票 (PDF)

データ

データ + リレショナルファイル + データベース

お問い合わせ

初回コンサルティング無料です!  
お気軽にお問い合わせ、ご相談ください。

TEL : 095-840-0021 / FAX : 095-843-4648

お問い合わせフォーム

© OPENCOBOL.COM

Copyright © 2010 LANCARD.COM Inc. All Rights Reserved.

- Web少し整備しました。
- 検索ワードは「lancard 汎用機ダウンサイジング」あたりで。
- 問い合わせフォームがあります。ご利用ください。

<http://www.lancard.com/it-model/>

LANCARD.COM