

WebCamを作ってみた

作者：Hiroki @ 3J

1 事の発端

2010年初め、X680x0同好会の部室に設置してあったサーバーのclaclaがお亡くなりになったことが確認された。過去には多くの仕事をしていたようだが、すでに私が入学したころにはWebカメラを用いて5分ごとに部室を盗撮監視するWebCamのサービスを行っているだけであった。

最後の部室サーバーが死亡したことを機に、そのままサーバーを抹消することも可能であったろう。

しかし、WebCamは多くの利点を持っている。盗撮防犯と言う部室の平和を見守るという意味もあれば、研究室の先輩方が部室の内部を確認し部員を見つけたら部室に向かう等の、サークルにおいて重要なコミュニケーションを促進する効果も確認されている。Web管や鯖管にとってみれば、その挙動やサービスがどれほど可愛らしいものであるかは言うまでもない。

そこでWebCam復活のため、ファイル共有サーバー・プリンタサーバーを立てることを主な名目に掲げてサーバーを作り、WebCamを新たに作ることにした。個人的にはWebCamメイン。これは、その復活までの軌跡をまとめた記事である。

2 確認

2.1 claclaの死亡状態の確認

claclaはなぜ死んだのか。その原因を探るべく再起動をしたところ、以下のような出力を確認した。

```
job control turned off
(initramfs)
```

(initramfs)の行では入力も可能である。軽く検索したところジョブ制御(Ctrl + C等)が使えない特殊シェルとのことである。早い話、ping等で実行回数を指定しないと、延々とpingを打ち続けるらしい。つまり彼女は命令に対してひたすら一途と言うことである。

dmesgでエラー内容を見たところIO系統のエラーを多数確認した。

その後、USB<=>IDE変換ケーブルを用いてHDDにアクセスしようとしたところアクセス拒否。さらにOSをインストールしようとするフォーマットの時点でエラーが発生し、書き込めなかった。つまり、HDDが壊れてしまったらしい。



図1. 命令に一途な彼女

最近物覚えが悪くてねえ



図2. そろそろ寿命なHDD

2.2 スペックの確認

HDD が死んだだけであれば、HDD を交換するだけでサーバーとして復活する。そこでスペックを調べた。

パーツ名	スペック
CPU	850MHz
Memory	1GB

若干貧弱でその他ハードも古いため、新しいLinux を GUI で動かそうと思うと少し辛いだろう。そもそもサーバーに GUI は必要ないのだが、CUI だけだと日本語の入力が行えないなどの致命的な欠陥が存在する。(回避方法はいくつかある。)

これでは後に管理する人も大変かもしれない上に、簡単な誤字を直すために別マシンからファイル転送と言うのもあまり良い管理体制ではないので、新たにサーバーを作ることにした。

3 WebCam の復活計画

残念ながら WebCam のプログラムはバックアップしていないため、WebCam はロストテクノロジーになるかと思えた。(後日部室マシンから発見されたのは黙っておこう。)しかし私は過去に製作者である先輩から内部処理を直々に教わっていた。「無いなら作れ」の精神の元、WebCam を 0 から作ることにした。

仕組みは単純で、内部では cron でシェルを動かし、撮影を行う。その後 Perl の Image::Magick 等のモジュールを用いて画像に対して日付を入れ、後は CGI で一覧を出力するだけである。

一番の問題はコマンド 1 つで撮影をする部分である。ここに関するコマンドやプログラムはいくつも存在するのだが、まず Web カメラのドライバを入れなければならない。そのために型番などからドライバを探したりする必要があるのだが非常に面倒である。

そこで、UVC (USB Video Class) という、Web カメラの世界共通規格に注目する。この技術により、UVC 対応 Web カメラと UVC ドライバを搭載した OS ならば、挿すだけで Web カメラを認識することが可能である。これを使えば、コマンドはすでに用意された `fswebcam` を用いればよいだけなので、導入に対する敷居が一気に下がった。

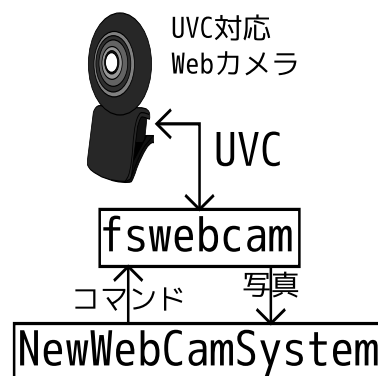


図 3. 作るのは 1 番下だけ

Linux においては Linux Kernel 2.6.26 以上のカーネルを採用したディストリビューションでは標準採用されている。CentOS は早く Kernel バージョンを上げるべき。

これを考慮すると、2010 年春の時点で対応している Linux がいくつか絞られ、Debian や Fedora、Ubuntu などが残った。この中でも clacla の中に入っていた、パッケージ数が多い、安定している等という理由で、Debian を採用することにした。後あんまり Ubuntu 好きじゃない。

4 買い出しと自作

新たにサーバーを作ることでも決定したので、秋葉原に買い出しに出かけた。今回の自作で重要視したのはサーバー機にするため、必要な部分にはお金をかけることにした。

具体的には HDD を高耐久性のものを採用したり、ネットワークをマザーボードののに頼ると貧弱なので NIC を専用買った。うるさいと嫌なので静音のために電源も少し余裕のあるものにした。発熱量を抑えファン回転を少なくするために低消費電力の CPU を選んだ。

主な購入パーツとスペックは次の通り。

パーツ	型番	スペックとか
CPU	AMD AthlonII x2 235e	消費電力 45W, 2.7GHz, AM3 ソケット
メモリ	UMAX DDR3-1333	1GB * 2, DDR3
マザーボード	GIGABYTE GA-880GM-USB3	DDR3, AM3 ソケット対応
HDD	Seagate ST3500514NS	500GB, サーバー向け高耐久
光学ドライブ	バルク品	OS インストールできればいいよね
電源	TAO SilentCool 630W	630W
ケース	GIGABYTE GZ-M1	ミドルタワーより少し背が低い
NIC	玄人志向 GbE-PCIe	1000Base-T/100Base-TX/10Base-T
Web カメラ	CMS-V27SETBK	UVC 対応, 130 万画素, 本体切り離し可能

自作に関しては部室で行った。パソコンの内部について知ってもらおうと、新入生などが参加できるように組み立てを行った。

その際、メモリがしっかりと刺さっていなかったためにビープ音が連続して流れ、参加者一同に不安な空気をもたらした。そのことから新しいサーバーの名前は Beep 音から **beebee** となった。不吉な名前でもこの先大丈夫かと不安になる方もいると思うが、X68 の部室には自作講座中に煙をモクモクと吐いた **mokmok** がいる。mokmok はデスクトップマシンでは現在最も古いにも関わらず現役で活動している。一方普通の名前のパソコンが次々と壊れているので、これはこれで縁起の良い名前かもしれない。

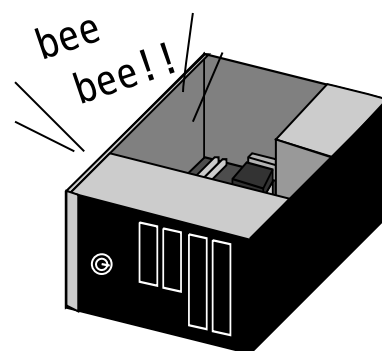


図 4.beebee

後日 OS として Fedora 13 を仮インストールし、UVC の機能を確認することとした。一度挿すだけで認識し、`lsusb` や `dmesg` 等でも正常に UVC ドライバが動いていることなどを確認した。

5 WebCam の作成

5.1 fswebcam

WebCam を作る際に基盤となるコマンドが `fswebcam` である。このコマンドをただ実行すると Web カメラでの撮影が行える。しかしそのまま撮影すると少し白っぽい画像となった。

原因を調べるべく `cheese` という動画を見る GUI のアプリケーションを起動し、様子を見た。すると起動直後は少し白く、その後調整してくっきりとした色になったことから、開始直後の数フレームは飛ばす必要がある。

それを踏まえてコマンドの引数が決定した。

```
fswebcam -d /dev/video0 -S 20 -r 1600x1200 --png 0 temp_0.png
```

コマンドの引数

-d デバイスを指定する。省略すると/dev/video0 が選択される。
-S 飛ばすフレーム数で今回は 20 にした。省略すると 0 になる。
-r サイズ指定。これを省略すると中途半端なサイズ (384x288) が指定され
 うまく撮影できずタイムアウトする。
--png 出力する画像形式を PNG にする。0 の意味は良く分からない。
temp_0.png 出力ファイル名。

これで色のくっきりした画像 temp_0.png が生成できる。

5.2 ディレクトリ構成

今回はディレクトリの構造を以下のようにする。

```
+ log_[Web カメラの番号]
  + [年 4 桁][月 2 桁][日 2 桁]
    + pic
      + 大きな画像達
    + thumb
      + サムネイル達
```

この [年 4 桁][月 2 桁][日 2 桁](20100531 など) というディレクトリを残すログの数だけ保
存し、それを読み込む。基本的にサムネイル一覧を表示し、それをクリックすると日付や撮
影時間がつけられた画像を大きく表示することができるようにする。

大きな画像とサムネイル名を一致させることで、ディレクトリ名を置換するだけでもう片
方にもアクセスできるようにする。

このディレクトリの自動生成やサムネイルの作成などは全て Perl に任せる。

5.3 PerlScript

次にサムネイル等を生成する。

今回は Perl の画像処理を行う Image::Magick モジュールを採用する。(開発環境には GD し
か入らず、Debian には Image::Magick しか入らなかったのがこれで頑張るしかない。) 処理
の手順は以下ようになる。

- Web カメラが撮影した temp_0.png を読み込む。
- ファイル名の生成とディレクトリが存在しない場合にはディレクトリを作る。
- Image::Magick で画像を大きな画像サイズにリサイズしてコピーする。
- Image::Magick でコピーした画像をリサイズする。

- 大きな画像をファイルに出力する。
- サムネイルをファイルに出力する。

ページがサムネイル一覧を作るので、サムネイルが存在するときに大きな画像が存在している必要がある。そのために大きな画像をより早く生成するようにしている。

基本的に上のことしかしてないので比較的簡単な(モジュール任せの)処理となる。

5.4 ShellScript

ここで Web カメラで撮影し、その画像を適切に加工するシェルスクリプトを作る。

まずは撮影するコマンドに関してだが、これはシェルスクリプトにして簡単に実行できるようにする。ここでファイル名を `cap_0.sh` とする。(cap_[Webカメラの番号].sh)

```
01 | #!/bin/sh
02 | fswebcam -d /dev/video0 -S 20 -r 1600x1200 --png 0 temp_0.png
03 |
04 | perl photograph.pl 0
```

この後 `chmod 0705 cap_0.sh` を実行して、`./cap_0.sh` というコマンドだけで実行できるようにする。次のコマンドの `perl photograph.pl` は Perl の `photograph.pl` を実行する。この Perl-Script は画像を適切に加工して、サムネイルとタイムスタンプのある大きな画像を生成する。最後の引数 0 はビデオ 0 番に指定したもの (`/dev/video0` とは限らない。設定による) を使うという意味になる。

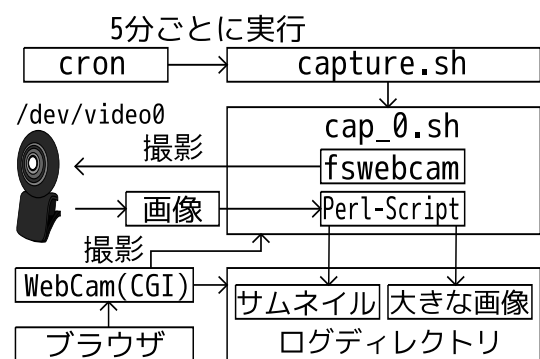


図 4. 全体像

ここでスケジューリングに用いる cron について触れておく。

cron は Linux のスケジュールを管理するプログラムで、ユーザーが独自に時間と実行ユーザーとコマンドを登録することで、指定時間に実行ユーザー権限でコマンドを実行する。

ただし、ここでシェルは実行した場所でパスが決定してしまうので、必ず `cd` で WebCam のプログラムがある場所まで移動しなければ正常に動作しない。

それを考慮した上で、先ほど作った `cap_0.sh` を実行するシェルを作る。プログラムを置いたディレクトリを `/home/x68/public_html/webcam/` とし、作るシェルは `capture.sh` とする。

```
01 | #!/bin/sh
02 |
03 | #カレントディレクトリ移動。
04 | cd /home/x68/public_html/webcam/
05 |
06 | #撮影と画像加工をする。
07 | ./cap_0.sh
```

これを cron に登録すれば自動的に実行される。その前に chmod を使って実行権限を書き変えておく必要がある。

5.5 cron

最後に cron に登録する。

vi 等のエディタで /etc/crontab を編集する。そして末尾に以下の文を追加する。WebCam を実行するユーザーを x68 とする。

```
*/5 * * * * x68 /home/x68/public_html/webcam/capture.sh
```

cron は [分 時 日 月 曜日 実行ユーザー コマンド] でコマンドを定期的に実行する。* は毎回と言う意味で分のところが* なら毎分で、日のところならば毎日という意味になる。*/5 は 5 で割り切れる時間なので 5 分ごととなる。

これで 5 分ごとに撮影する。

6 Perl-CGI や JavaScript

CGI に関しては一覧表示するだけなので特に特筆しない。

トップページに関しては JavaScript を用いてボタンや更新を担当させた。その理由に関しては、撮影後最新の情報に更新すると、GET や POST の形式だと再度撮影を行ってしまう。それを避けるために、トップページは最新の情報に更新してもそれによる誤撮影ができない JavaScript(Ajax) によるデータ送信を実装した。それに伴い、更新も JavaScript(Ajax と DOM) を用いて動的に書き換えた。

7 まとめ

以上の作業を経て、無事に WebCam が復活した。一番大きいのはやはり Linux にも挿すだけで Web カメラを認識する UVC が搭載されたことだろう。適切な Web カメラ (と言っても最近発売の Web カメラの多くは対応している) を買えば容易に似たようなシステムを構築することができる。

現在、世界中にはいろいろな規格と言うものが存在し、それぞれが自分勝手に機能を拡張するので、プログラマーを泣かせることが良くある。世界共通規格を設定することで、いくつかの環境で簡単に同じ挙動で動作するため、開発が非常に楽だった。

早くいろいろな規格を世界共通にしてほしいと願うばかりである。

8 今後の展開

WebCam の今後に関してだが、最近 UVC に対応した赤外線 Web カメラなるものが手ごろな価格で出回っている。これを部室の反対側に設置すれば、暗い内部の様子も撮影できる。なかなか面白そうなものなので、いつかそれを追加することが夢であり、WebCam のシステム自体も複数の Web カメラの撮影に対応できるように作っている。(log_[Web カメラの番号] とか cap_[Web カメラの番号].sh とか)

他にも、本会誌内「つくってみた」で作られた Web カメラ in ボールも提供されたが、これに関して fswebcam コマンドを使っての撮影を行うことはできない。しかし、撮影コマンドは fswebcam を使う必要は必ずしもないので、近々設定して撮影をさせたいと考えている。いつか、部室に数多の監視カメラが設置される日も近いかもしれない。

9 参考資料とか

- CPAN (GD、Image::Magick 関連。)
- man fswebcam (fswebcam の引数関連。)

10 その他

調布祭当日に販売している X68DVD のデジタル会誌。もしくは X680x0 同好会の公式サイト (<http://www.x68uec.org/>) 内のその他の作品 > 2010 年度 (vol.16) にこの記事のデジタル版がアップされるとともに、スクリプト一式を ZIP で固めて置いています。欲しい方はこちらからダウンロードしてください。

11 WebCam

WebCam の雄姿。

部室の全てを、私は見ている !!

