2020年7月9日 投稿者: SATOXITS

レイヤー間に打つ楔（くさび）

開発：なにか真面目なことがやりたいですね。

基盤：一方そのころ遊びに飽きた浦島は…

社長：おもしろまじめが当社のモットーです。

開発：フロントエンド側はエンドユーザの一人として面白いわけですが、やはりOS側に手を突っ込む面白さは別です。おれの指先一本で大怪獣もでんぐり返るぞ的な事が出来てしまう。

社長：フロントエンド側ではコンテクストメニューとIMEあたりに開拓の余地があるように見えます。それでユーザから見た景色が変わる。バックエンド側は、少くとも当社が入り込む余地があまり無いような。

開発：これは逆に、全く同じ景色に見えるのに、全く別のものを見ているという事が実現できる面白さかなと思います。

社長：まあインタフェースとインプリメンテーションてやつですね。プロトコルであり、アーキテクチャであり。

開発：私らの学生の時はPDPからVAXへくらいの頃で、いろんな計算機の方式が研究されてた時代。バローズみたいな面白い計算機も生き残っていた。CISCどころではなくオブジェクト指向の計算機のようなものもあり。そんな時に目にした「コンピュータアーキテクチャとはインタフェースである」というフレーズが非常に印象に残っています。ソフトウェアから見たら実際命令セットというインタフェースなわけですが。ですが、そのインタフェースを実用のものにできる実現方式というものが重要なんでは無いかと。

社長：ともあれ、インタフェースさえ同じであればその先の実体は何でも良いというのが根本的な考え方ですね。通信プロトコルであり、APIであり、システムコールであり。い

かに良い切り口で切るかが重要。

開発：そこで、あまり切り口が生々しいというか枯れていないところに突っ込んでしまうと、互換性ガーという非生産的な戦いに人生を奪われてしまう。

社長：ですが、10年前はまだ生々しさが残っていましたが、今現在はかなり安定したように思われます。

開発：それで、どこの切り口に切り込むかと考えると、ネットワーク的には垂直斬りでゲートウェイ。典型的にはプロキシ。一方でコンピュータ内では水平斬りだと思うわけです。

社長：水平断面に楔を打って遊びを噛ませて面白いことを注入したい。

基盤：グルコサミンですか。

開発：使える道具はやはり、動的ライブラリです。強く記憶に残っているのが、SOCKSにおけるSOCKSifyです。あれは socket ライブラリを差し替えるものだった思いますが、再コンパイルを必要としていたことを考えると、あの当時はまだ動的ライブラリは一般化してなかった。

社長：いえ、shared library は SunOSの頃から。 でも私は、標準的なライブラリも静的にリンクして実行形式にしてた事が多かった。そうしないと、OSがアップグレードする毎に使えなくなってしまうとか、ポータビリティを損なう副作用が強かったですね。いわゆるワークステーションの時代。

基盤：おっと、whois.com からトランスファー完了 Congratulations! のメールが来てますね。18:02。このドメインを昨日AWSオレゴンに作ったライトセール機に向けましょう。

開発：そう言えばあそこから xso の its-more.jp に ping したらRTT 100ms以上かかりましたね。すごくファーイースト。

社長：我社も世界展開のためには北米に拠点を置くのは必須です。特にpointillismみたなアプリでは。

基盤：さらに500円奮発してヨーロッパにも拠点とか。

開発：ですが今や、OS側もアプリケーション側も、システムコールや標準ライブラリのインタフェースというか仕様においては、かなり落ち着いた状態にあるんだと思います。もはや枯れたというか。

開発：4月に新しい基盤方式を考えて試作した時には、この水平斬りをやって外に引っ張り出して遠くに投げたわけですが、やはりアプリをコンパイルし直さないといけないのではないか、すると適用性が損なわれる。特にメインのターゲットとして考えたWindowsではそうなのでは無いかと思い、立ち止まってしまいました。ですが、原理的にはWindowsだって標準のDLLを差し替えればよいはず。ただデフォルトのを差し替えるのは危険すぎるから、ユーザごと、アプリごと、オンデマンドに差し替えたい。

開発：それで検索したら、Windowsでも、UNIX的なお手軽さでは無いですが、アプリが使用する動的ライブラリを選択する手段があることがわかりました。

基盤：セキュリティ上やばくないですかね。

開発：そのへんは、動的リンクをする際に認証なりする道具はあるんだろうと思いますし、なければ作れば良い。それで、これは動的ライブラリ差し替え作戦を少し進めてみようと思ったわけです。

社長：差し替えもしくはラッピングですね。

開発：実装上の問題は、たとえば libc.so みたいな巨大なライブラリの塊全体を入れ替えたいわけでは無く、ライブラリの一部分だけ差し替えたりラッピングしたい、それを簡便に実装できるだろうか？ということです。

社長：そのへんはあんまり簡便さを求めなくても良いのでは。

開発：性能上も、実際に全ての機能が噛ませたインタフェースを通るとすると、損なわれる危険性があります。関係ないものは直結したい。

開発：ひとつの理想は、localize.so とか personalize.so 的な、もしあれば見るよみたいなオプショナルなライブラリを全部のアプリが見てくれることでは無いかと思います。.ほにゃららrcみたいな。

社長：もし差し替え作戦がうまくいくなら、各アプリにそういう細工をちょっと入れてもらうという手もあるかと思いますが。

開発：あと、組み込み系のCではよく、 WEAK という属性を使って関数単位での差し替えをやるのですが、ひょっとしてそれがうまく使えるないだろうかとか。

社長：なにかひとつやってみますか。

開発：それで、一昨日ロボット追跡をやった時に、10進整数表記の時刻を date +format コマンド的にやりたいと思ったんですが、やり方がわからなかったので単に strftime() を

呼ぶプログラムを作りました。こういうのは、自分で作れば数分なのに、探すと数十分かかった上に、無かった。という話になる。どこでも同じようにやりたいことを、郷に従うと良くないという話です。

開発：で、もし date コマンドをそのまま使って、time() とか　gettimeofday() を差し替えればよいのではないかと思ったわけです。例えば定数値を返すgettimeofday()を定義して date コマンドに食わせる。現時刻以外のフォーマット出力ができる。その手を使って、システムの実際の時間とは違う時間でアプリを動かしたら面白いとか、けっこう用途が広いかもしれないと。

社長：システムコールをRPCで実装するとかはありがちだと思いますが。Unix のシステムコールをWindowsのWin32にマッピングしちゃったりとかも。でも、環境変数で関数を定義したりCGIで実装できたりしたら面白いですね。

基盤：性能的にはすごいことになりそうですが。

開発：最近のネットワークドライブとか、そういう感じですよ。それにうんざりしたところが、うちの原点の一つになっているとも言えます。

＊　＊　＊

社長：うーん、思いついた。gasketっていう名前はどうかな。隙間を適当に埋めるやつ。etだし。Google の GAS を連想させてしまうのは良くないかもですが…

基盤：ちょっとガス欠みたいな。

開発：えーと語源…　ロングマン現代英英辞典から…語源「**gasket** (1600-1700) Probably from French garcette, from Old French, “little girl”」。たぶんということですが。

基盤：へー。

社長：へー。どういう紆余曲折ですかね。意味深な感じも。

社長：そもそもガジェットとかウィジェットとかほにゃられっととか、et てプチ系だとは思いますが、明確に理解したことはありませんでした。

開発：えーと ette 語源…

> フランス語を語源から覚える！接頭辞と接尾辞【-ette編】
> 接尾辞-etteの意味はズバリ、**小型化**と**女性化**です。…　接尾辞-etteを単語の後ろに付けると、**その単語が元々持っている意味を小さくする**ことができます。…
> （例）シガレット、ピンセット、アントワネット、タブレット、オムレツ、…

社長：思い出しました。第2外国語で教わりました（笑）

社長：えーと、whois.com では gasket.live が $3.88、.space が 0.88。xso では .live が380円、.space が50円。xso 強いですね。10年だと1年あたり2500円程度。whois.comでは$23程度。接戦です。

経理：まちがって10年でポチらないで下さい。

開発：gasket.live て生体シートみたいな感じですかね。

基盤：whois.com は webgasket.com $9.88でどうですかとか勝手に提案してきてますねｗ　確か nso でもそう。

社長：うーん、名前の件はちょっとペンディング。

開発：sket / 助っ人でどうですかね。

社長：閑話休題。

＊　＊　＊

開発：ではまず原型 。

```
#include <stdio.h>
#include <sys/time.h>
//int gettimeofday(struct timeval *tv, struct timezone *tz);
#include <time.h>
//size_t strftime(char *s, size_t max, const char *format,
//  const struct tm *tm);
//struct tm *localtime(const time_t *timep);

int main(int ac, char *av[]){
    char atime[128];
    struct timeval tv;
    struct timezone tz;
    time_t utime;
    struct tm tm;
    char *fmt = "%Y%m%d-%H%M%S";

    gettimeofday(&tv,&tz);
```

```
    utime = tv.tv_sec;
    tm = *localtime(&utime);
    strftime(atime,sizeof(atime), fmt,&tm);
    printf("%s\n",atime);
}
```

基盤：これは Go では無いようですが。

開発：こういう時は原点に戻るのです。実行するとこうなる。

```
linux% a.out
20200709-204825
```

開発：次に自前の gettimeofday()をでっち上げて上書きする。

```
int gettimeofday(struct timeval *tv, struct timezone *tz){
    tv->tv_sec = 12*60 + 34;
    tv->tv_usec = 0;
    return 0;
}
```

```
linux% a.out
19700101-091234
```

開発：わたしは unixtime の 1970年1月1日を見るたびに、ベル研Unixとケン・トンプソンのことを思い出して目頭が熱くなるのです。Linux？なにそれというか。

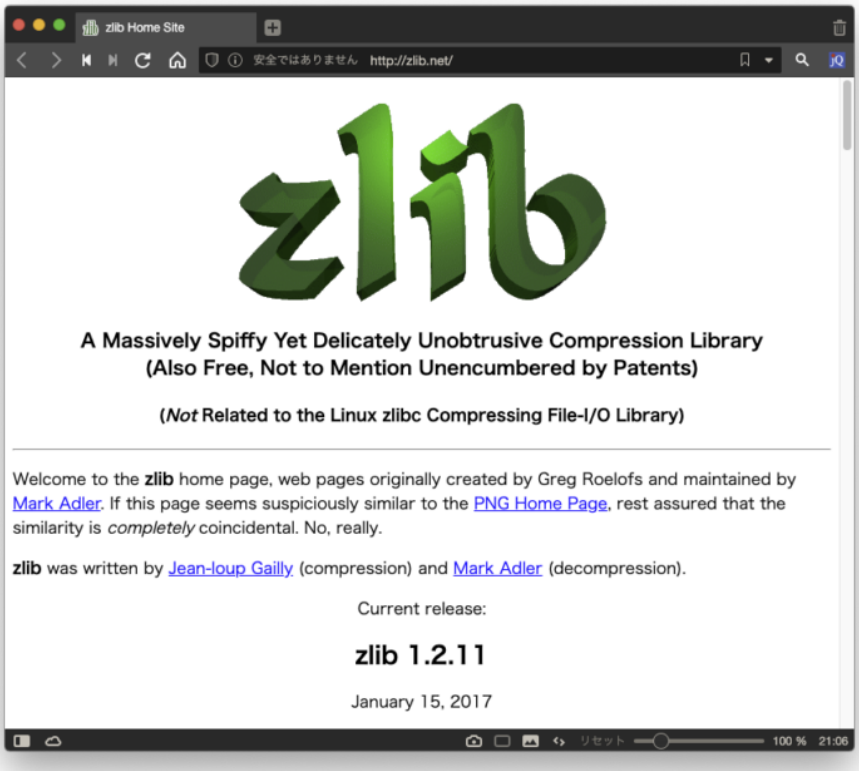
社長：ビル・ジョイがフェラーリ買って喜んでた写真も思い出しますね。

開発：あとはこの、なんちゃって gettimeofday を動的ライブラリで差し替えるだけです。が、やりかたを全く忘れてしまいました。DeleGateって、自分はshared library になるんでしたっけ？

社長：記憶にございません。

開発：シンプルなところで、昔使ってた zlib に学びますか。…

基盤：zlib …



開発：うっ。懐かしさで涙が…

社長：20年前からあなたの容貌は全く変わっていなかった。

基盤：残念ながらこのロゴのpngは動的に生成されるようで、Last-Modified はわからないですね。

開発：MacOS Catalina では 1.2.11、Amazon Linux には 1.2.8 が入っております。では、ソースをダウンロードして make。ああ、./configure を最初にやれと。で make。

```
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/adler32.o adler32.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/crc32.o crc32.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/deflate.o deflate.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/infback.o infback.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/inffast.o inffast.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/inflate.o inflate.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/inftrees.o inftrees.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/trees.o trees.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/zutil.o zutil.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/compress.o compress.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/uncompr.o uncompr.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/gzclose.o gzclose.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/gzlib.o gzlib.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/gzread.o gzread.c
gcc -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN  -DPIC -c -o objs/gzwrite.o gzwrite.c
gcc -shared -Wl,-soname,libz.so.1,--version-script,zlib.map -O3 -fPIC -D_LARGEFILE64_SOURCE=1 -DH
AVE_HIDDEN -o libz.so.1.2.11 adler32.lo crc32.lo deflate.lo infback.lo inffast.lo inflate.lo inft
rees.lo trees.lo zutil.lo compress.lo uncompr.lo gzclose.lo gzlib.lo gzread.lo gzwrite.lo  -lc
rm -f libz.so libz.so.1
ln -s libz.so.1.2.11 libz.so
ln -s libz.so.1.2.11 libz.so.1
```

社長：思い出しました。-fPIC -DPIC でコンパイルして、-shared でライブラリにする。みたいな。

開発：では意味はわかりませんが真似して…

```
linux% gcc -shared -fPIC gettimeofday.o -o libgasket.so

linux% file libgasket.so
libgasket.so: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked,
BuildID[sha1]=f9e2d127a6bdaf34193c0d313673785694aac575, not stripped
```

基盤：shared library ができましたね。

開発：でこれをリンクして実行してみる。

```
linux% cc main.c libgasket.so
linux% a.out
a.out: error while loading shared libraries: libgasket.so: cannot open shared object file: No
such file or directory
```

開発：なるほど。ではこれで。

```
linux% export LD_LIBRARY_PATH=.
linux% a.out
19700101-091234
```

基盤：大成功。

社長：one giant leap for our company !!

開発：しかし持つべきはよき先達ですね。zlib ありがとう。

社長：DeleGateでは動的ライブラリとして、Zlib と OpenSSL、あとlibPAMににお世話になりました。良い巡り合いだったと思います。

開発：一応 strace で確認…

```
linux% strace a.out
execve("./a.out", ["a.out"], [/* 41 vars */]) = 0
brk(0)                                  = 0x9d1000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_A
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No
open("./tls/x86_64/libgasket.so", O_RDONLY|O_CLOEXEC) =
open("./tls/libgasket.so", O_RDONLY|O_CLOEXEC) = -1 ENOE
open("./x86_64/libgasket.so", O_RDONLY|O_CLOEXEC) = -1 E
open("./libgasket.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0`
fstat(3, {st_mode=S_IFREG|0775, st_size=6120, ...}) = 0
getcwd("/home/ysato/gasket", 128)       = 19
mmap(NULL, 2099544, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_
mprotect(0x7f5a2d5ac000, 2093056, PROT_NONE) = 0
mmap(0x7f5a2d7ab000, 4096, PROT_READ|PROT_WRITE, MAP_PRI
f5a2d7ab000
close(3)                                = 0
```

基盤：ええーっ、なんか不思議なところを探しまわてってますね。

開発：基本的に、参照側のファイルと同じディレクトリを探してほしいものですが。でも、Unix的にはそのパス名を逆探知する方法はないかも知れないですね。

基盤：ld.so のマニュアルによると、LD_LIBRARY_PATH は使えない場合が多そうです。

```
   o  Using the environment variable LD_LIBRARY_PATH, unless the execut
```

社長：まあでも、原理的にはこれで良いでしょう。

基盤：date コマンドにこれを食わせられないですかね。

開発：strace date … まあ libc.so しか使ってないですね。cp -p libgasket.so libc.so.6。strace date …

```
ns1$ strace date
strace: ./libc.so.6: no version information available (required by strace)
strace: ./libc.so.6: no version information available (required by strace)
strace: ./libc.so.6: no version information available (required by strace)
strace: ./libc.so.6: no version information available (required by strace)
strace: ./libc.so.6: no version information available (required by strace)
strace: ./libc.so.6: no version information available (required by strace)
strace: ./libc.so.6: no version information available (required by strace)
strace: ./libc.so.6: no version information available (required by ./libc.so.6)
strace: relocation error: ./libc.so.6: symbol __cxa_finalize, version GLIBC_2.2.5
not defined in file libc.so.6 with link time reference
```

開発：そして全てのコマンドが動かなくなったとｗ。LD_LIBRARY_PATH をアンセット。

社長：よしよし。何か見えてきましたね。今日は念願のドメイン名移管も成功したし、お祝いに行きましょう。

— 2020-0709 SatoxITS