株式会社 ITS MORE
2020年4月設立

Digitally signed by 佐藤 豊
Date: 2020.07.13 04:33:14 +09'00'

2020年7月12日 投稿者: SATOXITS

タイムスリッパMac版

社長：やはり、MacOSXでもやりたいですね。

開発：焦点は、LD_PRELOAD と同じ機能が MacOSXにあるかなのですが、どうやら DYLD_INSERT_LIBRARIES というのがそれのようです。あと、LD_LIBRARY_PATH でなくて DYLD_LIBRARY_PATH だということは覚えています。

開発：では、さっそくやってみましょう…

```
MacMini% ./godate
2020-07-12 21:34:05.614626 +0900 JST m=+0.000061854
MacMini%
MacMini% DYLD_INSERT_LIBRARIES=./libgasket.so.1 \
DYLD_LIBRARY_PATH=. \
./godate
1970-01-01 09:22:33 +0900 JST m=+0.000072819
```

基盤：できた！

開発：イキナリできてしまった・・・

社長：まじですか？しかもGoで。

開発：いや正直、そもそも実現性の目星がつくまでに数時間はかかると思っていたのですが、、5分でできてしまいました。あと、DYLD_INSERT_LIBRARIESはこういうふうに書くと、DYLD_LIBRARY_PATHも要らないようです。

社長：めでたいので飲みに行きたい…

＊　＊　＊

```
MacMini% cd /Library/Developer/CommandLineTools/usr/bin
MacMini% file ld
ld: Mach-O 64-bit executable x86_64
MacMini% size ld
__TEXT   __DATA   __OBJC  others  dec     hex
1581056 131072  0       4295974912      4297687040      100298000
MacMini% ls -l ld
-rwxr-xr-x  1 root  wheel  2678608 Apr  9 03:24 ld
```

開発：まあ、ライブラリを静的にリンクしているか動的にリンクしてるかでしょうね。dyldinfo…

```
MacMini% dyldinfo -dylibs /usr/bin/ld
attributes     dependent dylibs
                /usr/lib/libxcselect.dylib
                /usr/lib/libSystem.B.dylib
MacMini% dyldinfo -dylibs /Library/Developer/CommandLineTools/usr/bin/ld
attributes     dependent dylibs
                /usr/lib/libxar.1.dylib
                @rpath/libtapi.dylib
                @rpath/libswiftDemangle.dylib
                /usr/lib/libc++.1.dylib
                /usr/lib/libSystem.B.dylib
```

基盤：なるほど…　というか、/usr/bin/ld のほうってなんにも入ってないですね。起動用のスクリプトですか？みたいな。

```
MacMini% nm /usr/bin/ld
0000000100002010 d __dyld_private
0000000100000000 T __mh_execute_header
0000000100000f77 T _main
0000000100002008 S _shim_marker
                 U _xcselect_invoke_xcrun
                 U dyld_stub_binder
MacMini%
MacMini% nm /Library/Developer/CommandLineTools/usr/bin/ld | wc
   12190   36291 1025488
```

開発：一昨日はタイムスリップの実現性について絶望のフチをさまよいながらやたらと色々やってたんで、何やってたんだかですが、そういえば dtruss だか dtrace だか使えという結論だったように思います。dtrace …　複雑過ぎる…　dtruss … sudo が必要なのか。sudo dtruss godate。おお、出た。

基盤：しかし、gettimeofdayの呼び出しは記録されていないですね。

開発：どうもよくわからないですね。うちの欲しいのは、特定の動的ライブラリの特定のシンボルの参照をトレースするということで…　てかそれは実行しなくても静的にわかるか。結局ld.so相当のものが MacOSX にあるのかという話しになる。ああ思い出した、それが dyld じゃないかというところで、path になかったから SEE ALSO の dyldinfo に行ったという経緯です。あと、otool といのが objdump の後継でみたいな話しとか…

社長：objdump って昔使ってましたね。今もあるんですか？

開発：objdump…　ありますね。ですが、Mach-O x86-64 がどうも、Invalid/Unsupported object file format. とか。

社長：一休みしましょう。

基盤：一休するって、子供とてまりをついたりするやつですかね。

開発：そういえば空気を吹いてでボールを浮かす健康器具、だいぶ前に届いたんですがまだ開けてませんでした。どこにいっちゃいましたかね。

社長：子供の頃よく遊びましたよね。あれって思うに、タバコ代わりにならないですかね？

＊　＊　＊

開発：それで前読んで何がいいたいのかわからなかった man dyld をもう一度読んだんですが、要するに環境変数 DYLD_PRINT_LIBRARIES と DYLD_PRINT_BINDINGS を定義してやれば、動的リンクをトレースできる、ということでした。

DYLD_PRINT_LIBRARIES
When this is set, the dynamic linker writes to file descriptor 2 (normally standard error) the filenames of the libraries the program is using. This is useful to make sure that the use of DYLD_LIBRARY_PATH is getting what you want.

DYLD_PRINT_BINDINGS
Causes dyld to print a line each time a symbolic name is bound.

基盤：標準エラー出力に出すって書いてないんですが、書いた人のムラ気ですかね。

開発：で、これを定義して実行してやると、ばっちりどのライブラリがロードされてどうバインドされたかが見えるのでした。

開発：そして残念なこと。/bin/date とか /bin/ls は、dyld で制御されている実行形式ではないようなのです。かといって中身がぜんぶ静的に詰まっているわけでもなく、nm でみると参照は全部 U。dyld_stub_binder というのを呼んで、自分で動的リンクをしているようなのです。

開発：Apple は dyld_stub_binder のソースを公開しています。30step 程度の小さなアセンブラのコードで、86版とARM版が入っています。

社長：うーん、ARM の LDR とか BL 命令がナツカシス。

基盤：というか、DYLDの ARM 版て…　Macのプログラムが ARMで動くってことですか。

開発：そういえば…　だとすると衝撃的ですね。

＊　＊　＊

開発：xclock dylib をタイムスリップさせようとしてちょっと行き詰まったので、気分を変えて Liunux の cal でやってみました。

```
ns1$ xdate
==== instantTimeSlipper (2020, ITS more)
==== /usr/bin/cal Today
      July 2020
Su Mo Tu We Th Fr Sa
           1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

==== /usr/bin/cal after TimeSlip to 1970
    January 1970
Su Mo Tu We Th Fr Sa
              1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

社長：ブラボー！

基盤：まあ cal 1 1970 と大差ないでけどね。

開発：ところがどっこい。それだと「今日」を示すハイライトが出ないわけですよ。

```
ns1$ cal 1 1970
    January 1970
Su Mo Tu We Th Fr Sa
              1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

開発：まあ、cal 1 1 1970 では出るわけですけど。

```
ns1$ cal 1 1 1970
    January 1970
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

社長：まあこれはあくまで動的リンクの使用例のデモ用ですからね。

社長：ていうか今私は、Google IME で分節を広げたり縮めたりするのが Control-O、Control-I だというのに気づいてびっくりしています。これって onew と同じなんです。素晴らしい。

開発：それで、instantTimeSlipperのコマンド型式なんですが。コマンド名は its、時刻設定する引数の型式は date に上位互換、が良いかなと思います。

開発：実装上は、差し替えたgettimeofdayやclock_gettimeで、現時刻からその分のオフセットを足した値を返すのが良いと思います。

開発：課題は、自分で差し替えてしまったオリジナル関数をどう参照するかという点です。

社長：DeleGate でVCのランタイムを差し替えた時には、open を差し替えて _open を呼ぶ、みたいにしてましたね。ソケットのハンドルが、普通のファイルのハンドルと別世界なので、普通のハンドルと同じように見せてストリームI/Oをできるようにするためでした。

開発：それで、標準ライブラリをnmして気づいたのですが、libc もそうなっているようなんです。それで今、クロスリファレンスをみるコマンドってなんだっけと思い出そうとしているのですが…

社長：昔は性能を測定するのに使ってた prof コマンドの機能としてあったような…　というかあれはソースからクロスリファレンスを作るやつでしたね。

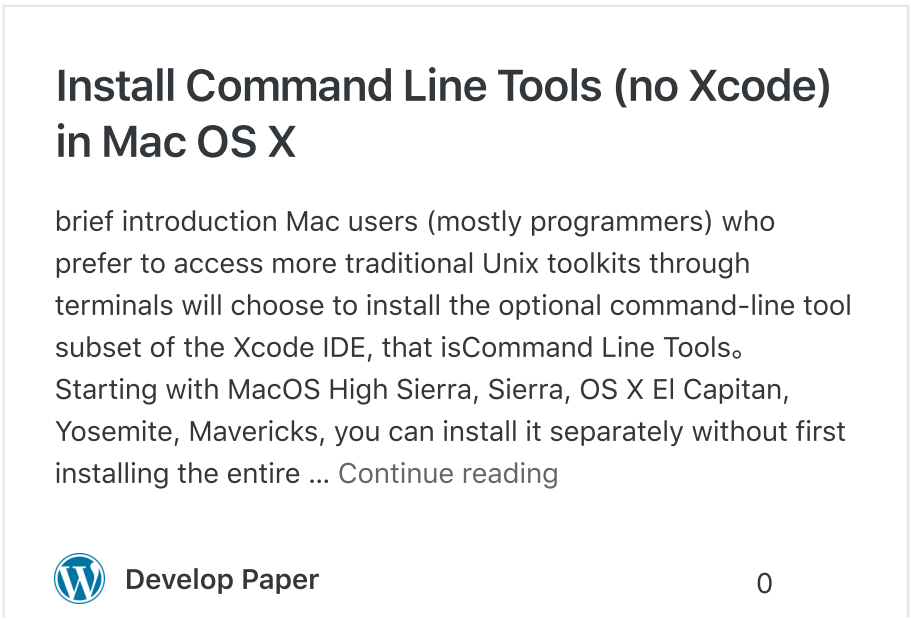
開発：この、知りたいのはこの __ をシンボル名に前置した版はなんなのかという事ですが…

基盤：洗濯おわりました。結論として、この血痕を落とすのは絶望的です。ズボンには膝に穴もあいちゃったし、捨てるのが吉かと。

社長：というべきところですが、少し昨日の飲み疲れが残っていまして。このまま続けましょう。何か他のアプリというかコマンドで。

開発：まず ls あたりかと思うんですが、MacOSXにはstraceというのが無いんですね。そういえば確かtruss系だったようにも思うんですが…　というか一昨日、XCodeのコマンドのbinがどこかにあるのを知って試しもしたのですが、メモルのを忘れました…　全端末と全ブラウザの表示テキストを全文検索できると良いのですが…　あきらめて検索します。

基盤：あれ？このページじゃないですか？Vivaldiのguestユーザで開いてます。



(2019-02-01) https://developpaper.com/install-command-line-tools-no-xcode-in-mac-os-x/

基盤：/Library/Developer/CommandLineTools/usr/bin ですね。

開発：思い出しました。その dyldinfo というのが必要だったんです。ldd みたいなやつ。

基盤：でも他はほとんど /usr/bin にもありますね。ただ、サイズが異様に違うんですが。

```
MacMini% file /usr/bin/ld
/usr/bin/ld: Mach-O 64-bit executable x86_64
MacMini% size /usr/bin/ld
__TEXT   __DATA   __OBJC  others  dec     hex
4096     4096     0       4294991872      4295000064      100008000
MacMini% ls -l /usr/bin/ld
-rwxr-xr-x  1 root  wheel  31472 May 28 09:24 /usr/bin/ld
```

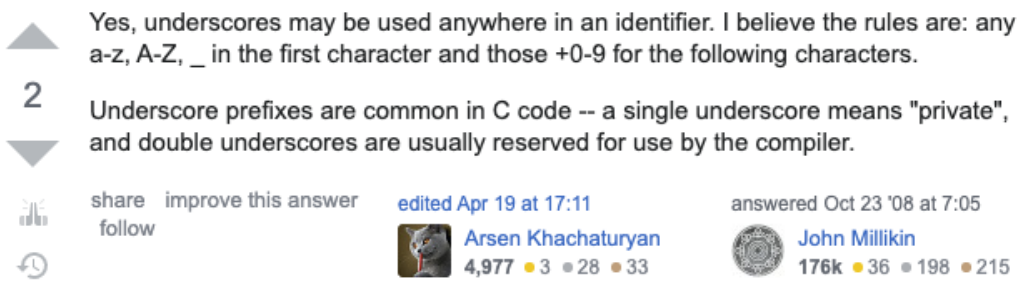
社長：しかたがないですね。というか同じズボンの一代目も、階段からころげおちた時に膝に穴があいて終わりました。あのときの膝の出血はひどくて、かさぶたがとれるまでに1ヶ月以上かかりました。

基盤：けっかん商品？

開発：それで、underscore symbol とかで検索すると、シンボル名の前につける _ には色んな意味がある。我々の育った頃にはCかアセンブラか、的な。その後は、非常に様々な定義というか仕様があるようです。

What are the rules about using an underscore in a C++ identifier?
[Stackoverflow, 2008]

開発：非常に簡単なサマリーとしては、_ はプライベート、__ はコンパイラが使用する、というものです。



開発：あまりここに深入りしても仕方がないので、我々としては、Linux版ではユーザは標準関数名 func() で呼ぶ、我々は func をラップして、 __func() を呼ぶ、とりあえずこれで行けると想定して進めたいと思います。

社長：そうしましょう。

＊　＊　＊

開発：Linuxについてはおおむねこれで行けることがわかりました。MacOSXですが、/bin/date などは dyld では制御できない。一方、そのアセンブリファイル（dyld_stub_binder.s）を見てみると、要するにこれを呼んでるだけなんです。

```
// call dyld::fastBindLazySymbol(loadercache, lazyinfo)
   bl      __Z21_dyld_fast_stub_entryPvl
```

開発：で、この関数の仕様がわからないのですが、動的リンクをやってるんだと思います。なので、この関数に、こちらで用意した動的ライブラリを見てもらうにはどうしたらよいか、ということになります。

開発：それと、このあたりを調べていたら、lldb というのが、MacOSXにおける dbx 相当品であることがわかりました。ただ残念ながらセキュリティ上の制約から、/bin/date とかをトレースすることはできないようです。

開発：それと、Appleが公開しているソースプログラムの中に dyld.cpp があること。これも何かに使えそうです。

開発：というようなことで、かなり状況は見えてきた気はします。まあ、gettimeofday は単に糸口だったので、そこそこのところで切り上げて、本質に切り込んで行きたいと思います。

社長：そうですね。ただ、せっかくなのでタイムスリッパはなんかの形で公開しましょう。今日はこのへんで。

— 2020-0712 SatoxITS