

株式会社 ITS MORE

2020年4月設立

ITS more

2020年8月3日 投稿者: SATOXITS

世の中の全てをファイルとして見る

開発：明日、いやもう今日ですが、大谷くんが先発だそうです。5時10分試合開始予定。

社長：おおそうですか。ではテレビを見る練習をしておかないと… リモコンはどこに行きましたかね。NHKですよね？

開発：BS1です。なんたって女子ゴルフと違ってメジャーですからねw

社長：NEWSWEBもとんと見てなかったです。ええ！照ノ富士復活優勝！？

開発：稀勢の里とのボロボロ対決を思い出しますね。

基盤：相撲やってるのも気づかなかったです。

経理：ところでNHKの衛星契約は受信料が月2000円のように。ケーブルテレビのほうでまとめて徴収されてると思いますが。

社長：おおっと、4LSU 相当ですかあ。そう考えるとかなりの額ですね。以前はずっとNHKつけっぱなしでしたから、元が取れてたとは思いますが…

基盤：リモコン発見。電源ぽちっと。そもそもBS1になってますね。準備OK。

* * *

開発：それで、自前のNFSサーバを作りたいと言っていた件ですが、ようやく見つけました。これです。

UNIX & LINUX

Home

Questions

Tags

Why is Linux NFS server implemented in the kernel as opposed to userspace?

Asked 7 years, 11 months ago Active 3 years, 2 months ago Viewed 15k times

<https://unix.stackexchange.com/questions/45899/why-is-linux-nfs-server-implemented-in-the-kernel-as-opposed-to-userspace>

社長：それだ。なんでなんですかね。

開発：nfsサーバをカーネルに取り込んだのは性能向上のためだったが、多くの問題を生んだ。今は反省している、みたいな流れですね。

9


Putting it in the kernel was an attempt to improve abysmal performance (mainly to PCNFS clients) and once that issue was solved noone looked at it much again.

There are a number of deficiencies with having NFS in the kernel, not least of which is that it doesn't play nicely with anything else touching the same filesystem (there are seriously nasty corruption risks) but back then (1993-4) we didn't realise that it would turn out to be an issue.

We were younger and more foolish, etc etc.

share improve this answer follow

answered Nov 26 '14 at 2:14

 stoatwblr

91 ● 1 ● 1

社長：うちはギンギンにアクセスするために使うのではないので関係ないですね。NFSクライアントから別サーバに中継するだけですし、基本、ユーザーローカルに動かす個人用のNFSサーバだし。

開発：それでこの記事の中で言及されてるのがまずFUSE。めちゃ遅いという認識は同じですね。一方、ユーザ空間で動くnfsdとしては、unfs3とGaneshaというのが紹介されています。

開発：で両方ともダウンロードして眺めたのですが、これはもうひと目見てunfs3で決まりです。めっちゃ簡単。名前のとおりNFSv3ですが、まあクライアントがそれで許してくれるなら問題ないでしょう。

開発：ビルドはこれだけ。あっというまです。

Building from source

You will need gcc, lex (flex), and yacc (bison) to compile UNFS3.

```
./bootstrap # (only when building from git)
./configure
make
```

開発：で、man ./unfsd.8 すると、ポート番号の指定は -m だそう。/etc/exports を変える時には -e。つまり、サーバの起動はこれで良いのではと。

```
MacMini% echo '/Users (rw)' > exports
MacMini% unfsd -d -m 9999 -e $PWD/exports
UNFS3 unfsd 0.9.22 (C) 2006, Pascal Schmidt <unfs3-server@ewetel.net>
/Users: ip 0.0.0.0 mask 0.0.0.0 options 4
```

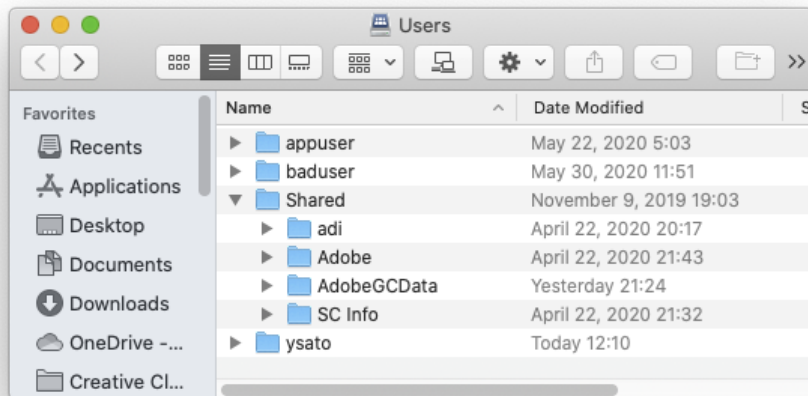
開発：一方でmountはこんな感じ。じゃん！

```
MacMini% mkdir $HOME/mynfs
MacMini% mount_nfs -o port=9999,vers=3 localhost:/Users $HOME/mynfs
MacMini% mount -t nfs
localhost:/Users on /Users/ysato/mynfs (nfs, nodev, nosuid, mounted by ysato)
MacMini%
```

基盤：通りましたね。

開発：でもって ls とかしてみる。

```
MacMini% ls -l /Users/ysato/mynfs
total 13
drwxrwxrwt  8 root   wheel  256 Nov  9  2019 Shared
drwxr-xr-x  11 appuser staff  352 May 22 05:03 appuser
drwxr-xr-x  11 baduser staff  352 May 30 11:51 baduser
drwxr-xr-x 148 ysato  staff 4736 Aug  3 12:10 ysato
MacMini% _
```



基盤：見えました。

開発：で書き込みの性能を見る。

```
MacMini% time openssl rand 1000000000 > 1GBa
openssl rand 1000000000 > 1GBa 0.64s user 1.51s system 84% cpu 2.543 total
MacMini% time openssl rand 1000000000 > $HOME/mynfs/ysato/1GBb
openssl rand 1000000000 > $HOME/mynfs/ysato/1GBb 0.62s user 2.07s system 40% cpu 6.738 total
MacMini% time openssl rand 1000000000 > 1GBa
openssl rand 1000000000 > 1GBa 0.63s user 1.03s system 82% cpu 2.017 total
MacMini% time openssl rand 1000000000 > $HOME/mynfs/ysato/1GBb
openssl rand 1000000000 > $HOME/mynfs/ysato/1GBb 0.60s user 1.79s system 67% cpu 3.542 total
MacMini% █
```

```

MacMini% time openssl rand 1000000000 > 1GBa
openssl rand 1000000000 > 1GBa 0.63s user 1.05s system 93% cpu 1.798 total
MacMini% time openssl rand 1000000000 > 1GBa
openssl rand 1000000000 > 1GBa 0.63s user 1.16s system 89% cpu 2.011 total
MacMini% time openssl rand 1000000000 > 1GBa
openssl rand 1000000000 > 1GBa 0.63s user 0.99s system 94% cpu 1.720 total
MacMini% █

MacMini% time openssl rand 1000000000 > $HOME/mynfs/ysato/1GBb
openssl rand 1000000000 > $HOME/mynfs/ysato/1GBb 0.60s user 1.88s system 69% cpu 3.577 total
MacMini% time openssl rand 1000000000 > $HOME/mynfs/ysato/1GBb
openssl rand 1000000000 > $HOME/mynfs/ysato/1GBb 0.60s user 1.84s system 69% cpu 3.506 total
MacMini% time openssl rand 1000000000 > $HOME/mynfs/ysato/1GBb
openssl rand 1000000000 > $HOME/mynfs/ysato/1GBb 0.60s user 1.84s system 73% cpu 3.336 total
MacMini% █

```

基盤：2倍位遅いですかね。約280MB/s、2.28Gbpsです。

社長：うちの使い方には全く問題なさそうです。

開発：ランダムアクセスした時にどうかという話がありますが、主な用途が付加価値的なデータ処理や別プロトコルのサーバへの中継だと考えれば、そちらで律速されますから、問題ないかなと思います。

基盤：アンチウィルスフィルタとか噛ませたら、ファイルI/Oが10MB/sくらいに落ちることもありますからね。あれは本当に不愉快というか使い物にならない。

開発：あとは安定性ですかね。というか、nfsが反応なくなると、I/O中のプロセスが固まって溜まりまくるといった伝統芸的な問題を何とかしたい。

社長：特にdfで様子を見ようとすると固まるあれですね。mount コマンドとかで状態を検査する機能があるとよいかと思いますが。

* * *

基盤：大谷くん、またダメでしたね。

社長：でも今回は1回はもちました。

開発：実戦のブランクも長かったですね。ピッチャーっていうのはそういうデリケートなポジションなんでしょう。

社長：それはそうと久しぶりにBS1を垂れ流して見たんですが、悪くないですね。

基盤：ですがあれを常時専用のテレビモニタに映しておくのは電気の無駄ですね。カメラ入力としてキャプチャしてデスクトップに小さく表示しておくのがよいのではないかと。

開発：HDMIをディスクに記録しながら中継したりUSBに出すような製品がありますよね。確か1万円くらい。おそらく、大して電気も食わない。

基盤：ひょっとしてQNAPの上位機種にあるカメラ機能がそのまま使えるとか。

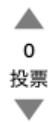
開発：上位版というと、うちのQちゃんより下位機種があるような響きですねw

経理：エアコン無しで、トイレと台所の照明は低消費電力LED化。次はテレビモニタのバックライトかと思います。あ、あとのオーディオアンプ。

基盤：ケーブルテレビを継続するなら、10年くらい前に廃止キャンペーンしてたこのセットトップボックスも交換したほうが良いかと思います。これもめっちゃ熱くなってるんで。

基盤：ああそれで、iMac用の増設メモリですが、正規版？がクソ高いので、アマゾンで安いのを見つけて発注しました。価格的にはMac用メモリ専門店の1/4くらい。16GBで6,000円強。こないだレノボに入れた8GBのとほぼ同等というか割安です。こなれてはいませんが、日本語も比較的まともだし、質疑にも答えています。

- Appleシステムの実運用環境で徹底的に品質検証を実施したので、Appleシステムと100%の互換性があることをお約束します。iMac、iMac 5k 2019、MacBook Pro、Mac Book、Mac Mini、Mac対応



質問:

[imac 5k 2019 に使用できますか?](#)

回答:

お客様
ご連絡ありがとうございます。
imac 5k 2019 にご利用いただけます。... [もっと見る](#)
投稿者: [v-color 出品者](#)、投稿日: 2020/05/04

社長：4倍という価格差。ハイリスク、ハイリターンな感じですねw

基盤：といたしますか、iMacをカスタマイズする時のオプションによると、うちのiMacで16GBから16GB増やすと+40,000円です。

メモリ

あなたにぴったりのメモリ容量は？

8GB 2,666MHz DDR4メモリ	- 20,000円 (税別)
16GB 2,666MHz DDR4メモリ	
32GB 2,666MHz DDR4メモリ	+ 40,000円 (税別)
64GB 2,666MHz DDR4メモリ	+ 80,000円 (税別)

開発：誰が買うんでしょうね？

基盤：お金持ちの奥さんとか。

社長：運悪く不良品に当たるかどうかですね。

開発：まあ、届いたら徹底的に初期テストをしましょう。

* * *

基盤：といたしますか、さっきからディスクがフリーズ気味なんですけど、unfs3のせいでは。

開発：そういえば、なんか固まったんでサーバ自体はkillしたんですが、umountとかdiskutil umount が効かなくなって…

社長：それは確か soft というマウントオプションを付けると良かったと思います。ファイルシステムとしてはちょっと嫌ですが、普通のアプリケーションプロトコルならタイムアウトで切れるとか普通だし、それに対応するようにプロトコルも実装も作られていますからね。

基盤：まあネットワークを意識したアプリじゃないと、ファイルI/Oが失敗した時の例外処理とか脱出なんて普通コーディングしないですからね。

社長：まー、カレントディレクトリがNFS上だったりするともう、どうにも回復もならなくなるわけです。zshあたりでそのへんなんとかしたいですね。

開発：うーん、だれかが掴んでてbusyのようで… Finderかな？プロセスをkill… あれー、すぐにまた生えてきちゃいますね。今朝も何度か強制unmountしてなんとかあったんですが。あの時はFUSEみたいなアイコンのがポップアップして答えたわけで… その後FUSEをコントロールパネルから抹消した程度なのですが… 呪い？

社長：もう小一時間がんばりました。再現性はあるでしょうから後で精査。リブートしましょう。

基盤：この、シャットダウンに抵抗するやつら、邪魔ですね。どうせ自動回復する機能があるんだから、おとなしく死ねば良いのに。OfficeとかFirefoxとか。ぷち。ぷち。ぷち。さあきれいにまりましたが… なんかくるくるしたままですね。

開発：電源リセットしましょう。

基盤：これやると、フィリップス君にレノボちゃんのDisplay Portのデスクトップがが顔を出すんですよねw で、再びHDMIを選んで MacMiniに戻す。

開発：きれいに立ち上がりました。というか、クラウド系のドライブのマウントがきれいさっぱりになりましたねw

社長：まあ、万が一MacMiniが立ち上がらなくなっても、もうiMacに環境があるし。最後の砦にレノボWin10も控えているわけです。BCPは完璧ですね。

経理：ベランダの打ち水は数分で蒸発してしまいます。

基盤：水道の蛇口がベランダにあると良いのですが… 設計ミスじゃないですかね。台所から引きましょうか。

社長：ていうか、うちのベランダってこんなに広がったんですね。野菜畑でもしましょうか。

* * *

開発：リポートしてからすごく快調で、diskutil なしの umount も普通に通るようになってしまいました。最初 / を export してマウントしたのが問題なのか、今はまだクラウドのドライブがほとんどマウントされてない関係なのか、よくわかりません。というか、unfsd 自体が出していると思っていた :/: の response が無いというようなエラーメッセージ、あーキャプチャし忘れました、MacOSのライブラリが出していたのかもしれない。strings unfsd しても出てこないんです。

開発：それはそうと、unfsd のバイナリはこんな感じ。

```
MacMini% ls -l unfsd
-rwxr-xr-x 1 ysato staff 143896 Aug  2 23:08 unfsd
MacMini% size unfsd
  TEXT   DATA   OBJC  others  dec      hex
77824   4997120  0      4295032832  4300107776  1004e7000
```

基盤：超コンパクト。

開発：ソースはこんな感じです。

```
MacMini% wc *.c
 496   1445  12299 attr.c
 974   2705  24004 daemon.c
 280    716   5370 error.c
 398   1478   9400 fd_cache.c
 473   1548  10771 fh.c
 347   1204   8372 fh_cache.c
 171    476   3377 locate.c
 379   2142  12382 md5.c
 289    823   7021 mount.c
1083   2963  28495 nfs.c
 116    341   2600 password.c
 215    688   5313 readdir.c
 293    813   5978 user.c
1062   3373  24959 winsupport.c
1656   3805  35920 xdr.c
 8232  24520 196261 total
```


開発：daemon.c はNFSプロトコルの出入り口で、実際の処理をやっているのがnfs.cです。

社長：つまり、うちが手をつっこむのは nfs.cだということですね。

開発：リードオンリーのファイルシステムなら、数個の関数しか関係しません。

```

/*
 * NFS service dispatch function
 * generated by rpcgen
 */
static void nfs3_program_3(struct svc_req *rqstp, register SVCXPRT * transp)
{
    union {
        GETATTR3args nfsproc3_getattr_3_arg;
        SETATTR3args nfsproc3_setattr_3_arg;
        LOOKUP3args nfsproc3_lookup_3_arg;
        ACCESS3args nfsproc3_access_3_arg;
        READLINK3args nfsproc3_readlink_3_arg;
        READ3args nfsproc3_read_3_arg;
        WRITE3args nfsproc3_write_3_arg;
        CREATE3args nfsproc3_create_3_arg;
        MKDIR3args nfsproc3_mkdir_3_arg;
        SYMLINK3args nfsproc3_symlink_3_arg;
        MKNOD3args nfsproc3_mknod_3_arg;
        REMOVE3args nfsproc3_remove_3_arg;
        RMDIR3args nfsproc3_rmdir_3_arg;
        RENAME3args nfsproc3_rename_3_arg;
        LINK3args nfsproc3_link_3_arg;
        REaddir3args nfsproc3_readdir_3_arg;
        REaddirplus3args nfsproc3_readdirplus_3_arg;
        FSSTAT3args nfsproc3_fsstat_3_arg;
        FSINFO3args nfsproc3_fsinfo_3_arg;
        PATHCONF3args nfsproc3_pathconf_3_arg;
        COMMIT3args nfsproc3_commit_3_arg;
    } argument;
}
"daemon.c" line 368 of 974 --37%-- col 2

```

開発：ちょっと getattr と lookup にprintfを突っ込んでみます。

```

MacMini% unfsd -d -m 9999 -e $PWD/exports
UNFS3 unfsd 0.9.22 (C) 2006, Pascal Schmidt <unfs3-server@ewetel.net>
/Users: ip 0.0.0.0 mask 0.0.0.0 options 4
1596427089 UNFS3 getattr(/Users) // (^-^)/ITS more
1596427089 UNFS3 lookup(/Users) // (^-^)/ITS more
1596427089 UNFS3 getattr(/Users) // (^-^)/ITS more
1596427089 UNFS3 getattr(/Users) // (^-^)/ITS more
1596427089 UNFS3 lookup(/Users) // (^-^)/ITS more
1596427089 UNFS3 lookup(/Users) // (^-^)/ITS more
1596427089 UNFS3 lookup(/Users) // (^-^)/ITS more
1596427089 UNFS3 lookup(/Users) // (^-^)/ITS more
1596427089 UNFS3 lookup(/Users) // (^-^)/ITS more
1596427089 UNFS3 lookup(/Users) // (^-^)/ITS more
1596427089 UNFS3 lookup(/Users) // (^-^)/ITS more
1596427089 UNFS3 getattr(/Users) // (^-^)/ITS more
1596427089 UNFS3 getattr(/Users) // (^-^)/ITS more
1596427089 UNFS3 getattr(/Users) // (^-^)/ITS more
1596427089 UNFS3 getattr(/Users) // (^-^)/ITS more
1596427110 UNFS3 lookup(/Users) // (^-^)/ITS more
1596427110 UNFS3 lookup(/Users/ysato) // (^-^)/ITS more
1596427110 UNFS3 lookup(/Users/ysato) // (^-^)/ITS more
□
unfs3-unfs3-0.9.22 — -zsh — 113x6
MacMini% mount_nfs -o port=9999,vers=3 localhost:/Users $HOME/my nfs
MacMini% mount -t nfs
localhost:/Users on /Users/ysato/my nfs (nfs, nodev, nosuid, mounted by ysato)
MacMini% ls -l /Users/ysato/my nfs/ysato/.zshrc
-rw-r--r-- 1 ysato staff 1242 Aug 2 03:34 /Users/ysato/my nfs/ysato/.zshrc
MacMini%

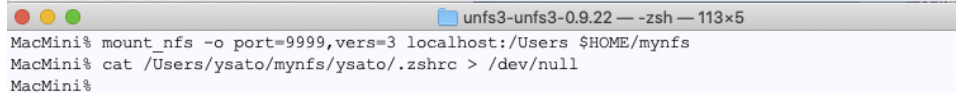
```

基盤：簡単過ぎる…

社長：たとえばファイルのメタ情報を仮想的なファイルとして読み書きできると良いですね。何かの拡張子的なものを付けて… たとえばファイル名が name なら `._name.attr` とか。

開発：あれ？open に対応するものは無いですね。とりあえず read…

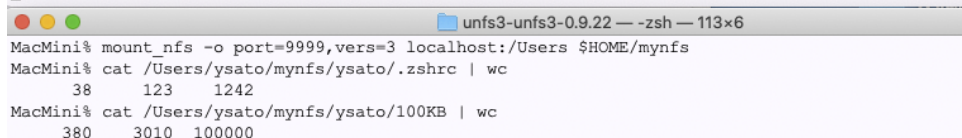
```
1596428157 UNFS3 lookup(/Users/ysato) // (^-^)/ITS more
1596428157 UNFS3 getattr(/Users/ysato/.zshrc) // (^-^)/ITS more
1596428157 UNFS3 read(/Users/ysato/.zshrc[8]+0,32768) // (^-^)/ITS more
□
```



```
MacMini% mount_nfs -o port=9999,vers=3 localhost:/Users $HOME/mynfs
MacMini% cat /Users/ysato/mynfs/ysato/.zshrc > /dev/null
MacMini%
```

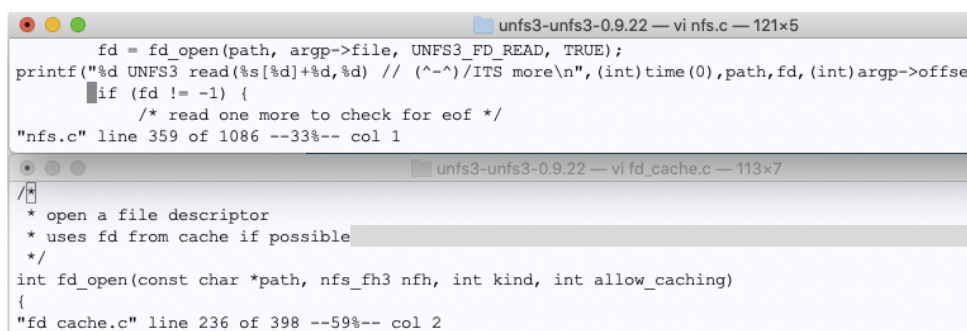
開発：1パケットに入らないのをread…

```
1596428344 UNFS3 lookup(/Users) // (^-^)/ITS more
1596428344 UNFS3 lookup(/Users/ysato) // (^-^)/ITS more
1596428344 UNFS3 getattr(/Users/ysato/.zshrc) // (^-^)/ITS more
1596428344 UNFS3 read(/Users/ysato/.zshrc[8]+0,32768) // (^-^)/ITS more
1596428354 UNFS3 getattr(/Users/ysato) // (^-^)/ITS more
1596428354 UNFS3 lookup(/Users/ysato) // (^-^)/ITS more
1596428354 UNFS3 getattr(/Users/ysato/100KB) // (^-^)/ITS more
1596428354 UNFS3 read(/Users/ysato/100KB[8]+0,32768) // (^-^)/ITS more
1596428354 UNFS3 read(/Users/ysato/100KB[8]+32768,32768) // (^-^)/ITS more
1596428354 UNFS3 read(/Users/ysato/100KB[8]+65536,32768) // (^-^)/ITS more
1596428354 UNFS3 read(/Users/ysato/100KB[8]+98304,32768) // (^-^)/ITS more
□
```



```
MacMini% mount_nfs -o port=9999,vers=3 localhost:/Users $HOME/mynfs
MacMini% cat /Users/ysato/mynfs/ysato/.zshrc | wc
38 123 1242
MacMini% cat /Users/ysato/mynfs/ysato/100KB | wc
380 3010 100000
```

開発：ふむふむ。ソースコードはこう。



```
fd = fd_open(path, argp->file, UNFS3_FD_READ, TRUE);
printf("%d UNFS3 read(%s[%d]+%d,%d) // (^-^)/ITS more\n", (int)time(0), path, fd, (int)argp->offse
    if (fd != -1) {
        /* read one more to check for eof */
    }
}
"nfs.c" line 359 of 1086 --33%-- col 1
```

```
/*
 * open a file descriptor
 * uses fd from cache if possible
 */
int fd_open(const char *path, nfs_fh3 nfh, int kind, int allow_caching)
{
    "fd_cache.c" line 236 of 398 --59%-- col 2
```

開発：なるほど、毎回openしてるんですね。ファイルディスクリプタがキャッシュに残っていたら再利用すると。

社長：要するにステートレスなんですね。このあたり、マルチプロセスでnfsdをするときにどう共有するのか。たぶん共有できないから、カーネルに入れたかったと。



開発：けっこうバッテリーを食いましたね。

基盤：可愛そうにQちゃんは、バックアップ無しの口に繋がれてました… 立ち上がりに数分かかるんですよ。

開発：なぜそのような迫害を。

基盤：最後に来た時に口が空いてなかったんですね。

QNAP：ぷうーーーーっ！

基盤：でも、元気に立ち上がりました。



社長：で、北の部屋の温度は？

基盤：30.0度でした。南より1度以上低いですね。

社長：風向きは？

経理：金鳥の煙は南から北へ向かって強く。

開発：うーん、とすると、北の部屋から南へ向けて強力なサーキュレータで送風する必要がありますね。

経理：単に北の部屋で仕事するという方法もあるかと思いますが。

社長：自然の風に任せるとすると、南のベランダに噴水付き庭園案は現実身が高まりません。昔ガーデニングで使ってた道具は全部廃棄しちゃったから、また揃え直さなくっちゃ♪

開発：近場でガーデニング系を置いてたホームセンターは全部消えちゃいましたからね。品揃えがちゃんとしている店までだと車で20分かかるのが大変です。

基盤：あまくぼにちっちゃい花屋さんみたいのを見かけましたけど。

* * *

社長：さて、では読むと実行して結果を返すようなファイルを作りましょう。

開発：とりあえず popen するって感じですかね。何か拡張子なり属性なり…

社長：属性の拡張は難しいでしょうね。とりあえず、ファイル名に'?'が付いてたら、'? 'より前を実行ファイル名にして、'? 'から後を引数にするみたいな。

開発：CGIですね。じゃパス名をstrchrして… あれー、read_svc() に渡ってこないですね…。なんかNFSクライアント側でディレクトリのキャッシュしているとか、そんな感じのようにも？

社長：ではとりあえず、拡張子的な目印では。

開発：たとえば .shx ならsystem()で実行して結果を返すと。うーむ、popen()は厳しいですね。何か named pipe 的なものでないか。とりあえず実行結果をファイルに書きちゃってそれを読む。うーむ、read-only filesystem? では/tmpに書いてやりましょう。Go!

基盤：おお出来ましたね。

開発：いやしかし、2回目からは前と同じ結果が返ってきてしまいますよ。どこかでキャッシュしてますね。ああ、ファイルのタイムスタンプ属性を見てるようですね。じゃあこれを、いつも現在時刻にしてしまう。これでどうかな？

```

MacMini% unfsd -d -m 9999 -e $PWD/exports
UNFS3 unfsd 0.9.22 (C) 2006, Pascal Schmidt <unfs3-server@ewetel.net>
/Users: ip 0.0.0.0 mask 0.0.0.0 options 4
1596440641 UNFS3 getattr(/Users) mtime=1575534091// (^-^)/ITS more
1596440641 UNFS3 fsinfo(/Users) // (^-^)/ITS more
1596440641 UNFS3 pathconf(/Users) // (^-^)/ITS more
1596440641 UNFS3 fsstat(/Users) // (^-^)/ITS more
1596440641 UNFS3 access(/Users) // (^-^)/ITS more
1596440641 UNFS3 getattr(/Users) mtime=1575534091// (^-^)/ITS more
1596440641 UNFS3 fsstat(/Users) // (^-^)/ITS more
1596440641 UNFS3 access(/Users) // (^-^)/ITS more
1596440641 UNFS3 getattr(/Users) mtime=1575534091// (^-^)/ITS more
1596440641 UNFS3 getattr(/Users) mtime=1575534091// (^-^)/ITS more
1596440641 UNFS3 access(/Users) // (^-^)/ITS more
1596440641 UNFS3 access(/Users) // (^-^)/ITS more
1596440641 UNFS3 getattr(/Users) mtime=1575534091// (^-^)/ITS more
1596440641 UNFS3 getattr(/Users) mtime=1575534091// (^-^)/ITS more
1596440641 UNFS3 getattr(/Users) mtime=1575534091// (^-^)/ITS more
1596440641 UNFS3 getattr(/Users) mtime=1575534091// (^-^)/ITS more
1596440641 UNFS3 getattr(/Users) mtime=1575534091// (^-^)/ITS more
1596440642 UNFS3 fsstat(/Users) // (^-^)/ITS more
1596440653 UNFS3 access(/Users/ysato) // (^-^)/ITS more
1596440653 UNFS3 access(/Users/ysato/xxx1) // (^-^)/ITS more
1596440653 UNFS3 access(/Users/ysato/xxx1/hello.shx) // (^-^)/ITS more
1596440653 UNFS3 getattr(/Users/ysato/xxx1/hello.shx) mtime=1596440617// (^-^)/ITS more
1596440653 UNFS3 getattr(/Users/ysato/xxx1/hello.shx) mtime=1596440617// (^-^)/ITS more
1596440653 UNFS3 read(/Users/ysato/xxx1/hello.shx > /tmp/unfs.shx.out) = [8] (^-^)/SatoxITS
1596440653 UNFS3 read(/Users/ysato/xxx1/hello.shx[8]+0,32768) // (^-^)/ITS more
1596440656 UNFS3 getattr(/Users/ysato/xxx1/hello.shx) mtime=1596440617// (^-^)/ITS more
1596440656 UNFS3 getattr(/Users/ysato/xxx1/hello.shx) mtime=1596440617// (^-^)/ITS more
1596440656 UNFS3 getattr(/Users/ysato/xxx1/hello.shx) mtime=1596440617// (^-^)/ITS more
1596440656 UNFS3 read(/Users/ysato/xxx1/hello.shx > /tmp/unfs.shx.out) = [8] (^-^)/SatoxITS
1596440656 UNFS3 read(/Users/ysato/xxx1/hello.shx[8]+0,32768) // (^-^)/ITS more
1596440658 UNFS3 getattr(/Users/ysato) mtime=1596440617// (^-^)/ITS more
1596440658 UNFS3 getattr(/Users/ysato/xxx1/hello.shx) mtime=1596440617// (^-^)/ITS more
1596440658 UNFS3 getattr(/Users/ysato/xxx1/hello.shx) mtime=1596440617// (^-^)/ITS more
1596440658 UNFS3 getattr(/Users/ysato/xxx1/hello.shx) mtime=1596440617// (^-^)/ITS more
1596440658 UNFS3 read(/Users/ysato/xxx1/hello.shx > /tmp/unfs.shx.out) = [8] (^-^)/SatoxITS
1596440658 UNFS3 read(/Users/ysato/xxx1/hello.shx[8]+0,32768) // (^-^)/ITS more

```

```

MacMini% mount_nfs -o port=9999,vers=3 localhost:/Users $HOME/my nfs
MacMini% cat $HOME/xxx1/hello.shx
#!/bin/sh
echo "Hello World! (" `date` ")\\n"
MacMini% cat /Users/ysato/my nfs/ysato/xxx1/hello.shx
Hello World! ( Mon Aug 3 16:44:13 JST 2020 )

MacMini% cat /Users/ysato/my nfs/ysato/xxx1/hello.shx
Hello World! ( Mon Aug 3 16:44:16 JST 2020 )

MacMini% cat /Users/ysato/my nfs/ysato/xxx1/hello.shx
Hello World! ( Mon Aug 3 16:44:18 JST 2020 )

```

基盤：やった！

社長：うーん、one giant leap ですね。

開発：明るい未来が見えた気がした。unfs3に感謝感激。

* * *

社長：思えば、私が初めてNFSに触ったのは、たぶん 1987年頃の Sun ででした。あの頃、NFSでこういう感じの遊びができれば面白いのにと考えたものですが、当時は商用の SunOSでしたし、たぶん無理でしたね。それが今や、こんなふうになんか出て来ちゃう。まさに夢のような世界ですね。

基盤：LinuxのNFSは 1993年、unfs3は 2003年に出来てたみたいですけどね。

開発：なにしろ90年代は、Unixワークステーションの全盛期でしたからねえ。1台何百万円はする。Linux？なにそれぞれのおもちゃ、みたいな感じでしたよ。下手に予算にも恵まれていた。

社長：やはり経済的にもミニマルなところから新しいものは生まれるんですね。ああそうだ、車検が上がったそうなので、取りに行ってきます。

経理：この額、この会社ではとても負担できないですね。

社長：まあ、車は半分趣味だから、尺度が違います。いわゆる社長のポケットマネーです。あっはっは・・・はは・・・

基盤：いずれ社長機密費とかできると良いですね。

— 2020-0803 SatoxITS

