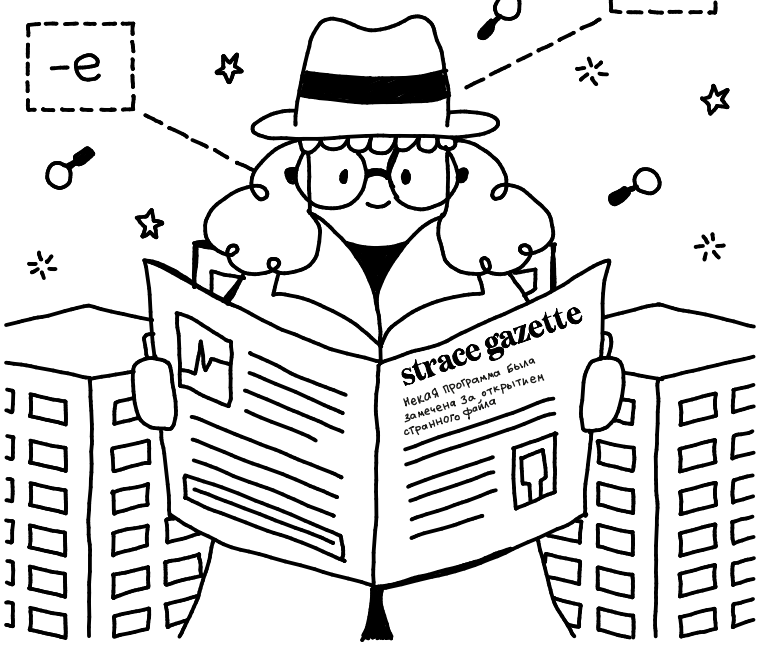
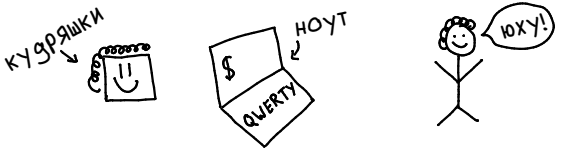
Шпионим за вашими программами с помощью strace

автор Джулия Эванс

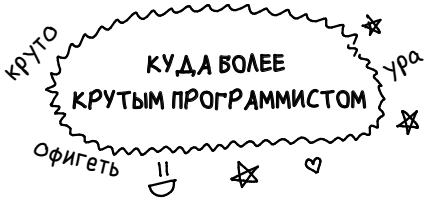


Кто делает это?

Привет! Меня зовут Джулия, и я выгляжу примерно так:



В прошлом году до меня дошло, что, понимая, что находится под капотом твоей операционной системы чуть глубже, ты становишься



Это было так занимательно, что я решила поделиться этим со всеми.
И вот делюсь с тобой!

У меня есть ещё такие комиксы, ищи у меня

в блоге: jvns.ca
в твиттере: @b0rk
в почте: julia@jvns.ca

♡ небольшой манифест ♡

операционные системы

этот комикс покажет тебе, что:

- твой компьютер – только твой
- твоя операционная система – только твоя
- опенсорсная лицензия позволяет тебе читать и менять код!
- LINUX - это РЕАЛЬНО КРУТО!

—> —> —> —> —> —> —> —> —> —> —> —> —> —> —> йи-иииха —> —> —> —> —> —> —>

ПОГНАЛИ УЧИТЬСЯ

—> —> —> —> —> —> —> —> —> —> —> —> —> —> это реально по фану —> —> —>

Что такое Strace???

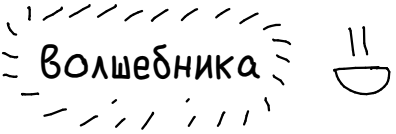
произносится как ess-trace

(на Макось надо юзать dtrace и dtruss)

strace - это программа для Linux, которая позволяет смотреть за тем, что делает та или иная программа без использования:

- дебаггера
- исходного кода
- Даже знания самого языка программирования не нужно (?!!?!Блин, да как это возможно? это вообще законно?)

в общем, Strace превращает тебя в настоящего



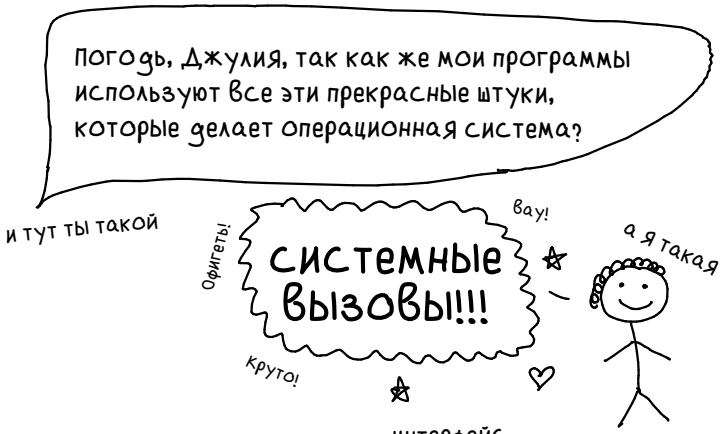
Чтобы понять, как это работает, давай немного поговорим об

операционных системах

Почему ты должен ♡ свою операционку

вот лишь несколько вещей, которые она для тебя делает:

- Понимает, как работает жёсткий диск и как файловая система на нём преобразует байты в файлы, чтобы ты мог открыть свой чёртов файл
- Запускает код каждый раз, когда ты нажимаешь на клавишу, чтобы ты мог печатать
- Реализует сетевые протоколы типа TCP/IP, чтобы ты мог просматривать ~~веб-страницы~~ фото котиков в Интернете
- Отслеживает состояние ВСЕЙ памяти, используемой каждым отдельным процессом.
- Ну и просто знает всё о железе в твоём компьютере и его работе, позволяя тебе сосредоточиться на написании программ! ♡



Системные вызовы – это ~~API~~ интерфейс твоей операционной системы

Хочешь открыть файл? Используй для этого `open`, а потом `read` или `write`

Отправляешь данные по сети? Сначала запусти `connect`, чтобы установить соединение, а потом используй `send` и `recv`, чтобы отправлять и получать фотки котиков.

Любая программа на твоём компьютере постоянно использует системные вызовы, чтобы управлять памятью, создавать файлы, устанавливать сетевые соединения и многое другое.

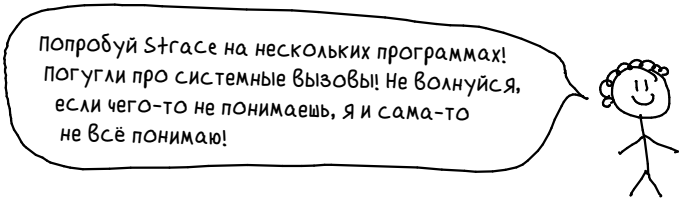
первое знакомство с strace

после всего сказанного выше про операционные системы и системные вызовы ты можешь подумать, что пользоваться strace трудно.

На самом деле начать просто! Если у тебя стоит Linux, то я настаиваю, чтобы ты попробовал ПРЯМО СЕЙЧАС.

запусти: `strace ls` время волшебства!

в результате команда выдаст на экран большое количество данных. На первый взгляд это может смутить. Но не пугайся, я объясню главное на следующей странице ☝



аннотация к Strace

Когда ты запустишь Strace, ты увидишь тысячи строк кода примерно такого вида:

```
$ strace ls /home/bork/blah
execve("/bin/ls", ["ls", "/home/bork/blah"], [/* 62 vars
*/]) = 0
brk(NULL)                                = 0xb67000
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
open("/proc/filesystems", O_RDONLY)      = 3
 ... omitted ...
open("/home/bork/blah", O_RDONLY|O_NONBLOCK|O_DIRECTORY) =
3
fstat(3, {st_mode=S_IFDIR|0775, st_size=168, ...}) = 0
getdents(3, /* 3 entries */, 32768)      = 80
getdents(3, /* 0 entries */, 32768)      = 0
close(3)                          = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136,
5), ...})                                = 0
write(1, "awesome_file\n", 13)           = 13
close(1)                          = 0
close(2)                          = 0
exit_group(0)                            = ?
```

Исследования показали, что здесь понадобится объяснение

(на самом деле я просто спросила своих друзей, всё ли им понятно, и ответ был: «НЕ-А»)

Давай разберёмся, как интерпретировать результат Strace

```
11999 execve("/bin/ls",["ls","/home/bork/blah"]) = 0
```

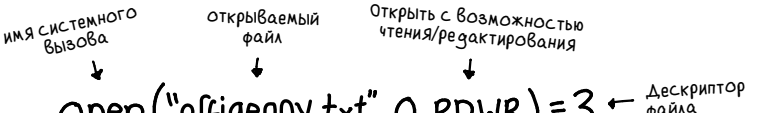
① ② ③ ④

① PID (выводится, если запустить strace -f)

② Имя системного вызова (execve запускает программу)

③ Аргументы системного вызова (в данном случае это программа, которую нужно запустить, и аргументы, с помощью которых её нужно запустить)

④ Ответ, который вернула команда



Здесь 3 - номер файлового дескриптора.
Linux сам отслеживает открытые файлы по номерам!
Ты можешь посмотреть все файловые дескрипторы для процесса ID 42 и на какой файл они указывают с помощью команды:

```
ls -l /proc/42/fd
```

fd — файловый дескриптор!

Если тебе что-то непонятно в данных, выведенных с помощью Strace:

- это нормально! Системных вызовов очень много.
- Ознакомься с мануалом по данному системному вызову!

```
man 2 open
```

- Помни, что даже простое понимание read + write + open + execve уже может привести тебя к успеху ♥

Мои любимые системные вызовы

open

У тебя когда-нибудь было такое, что ты не был уверен, какие конфиги использует та или иная программа?
ЭТО БОЛЬШЕ НИКОГДА НЕ ПОВТОРИТСЯ
Отложи доку и смело вводи:

```
strace -f -e open mplayer Yura_Shatunov.mp3
```

write

Программы пишут логи.
Если твоя программа записывает очень важную информацию, но ты не знаешь, какую именно и куда, то тебе может помочь

```
strace -e write
```

`read` - это тоже крутая штука.

Иногда программа посылает сетевые запросы другому компьютеру, и мне нужно знать, КАКОМУ ИМЕННО.

```
strace -e connect
```

Показывает каждый IP адрес, к которому обращается программа.

Знаешь, что на самом деле весело? Шпионить за сетевой активностью! Если ты пользуешься службой HTTP, занимаешься отладкой и твой мозг уже кипит... Возможно, настал момент, когда нужно взглянуть на данные, которые ДЕЙСТВИТЕЛЬНО НА САМОМ ДЕЛЕ передаются по сети...

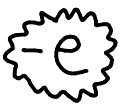
это твои бро ♡

В мой первый рабочий день скрипт на Ruby, который должен был запускать несколько ssh-команд, не работал. Ой-вей!
И что теперь? Копаться в коде, чтобы найти проблему? Чур меня!

```
strace -f -e execve ./script.rb
```

Указал на проблемную ssh-команду, и мы всё исправили!

Мои ♡ параметры командной строки в strace



в шоке от всех этих непонятных системных вызовов? Попробуй это:

```
strace -e open
```

Увидишь только вызовы open в данный момент. Так-то лучше, да? ♡

-f значит следовать (от follow)

твоя программа запускает подпроцессы? (многие так делают)

`-f` покажет, что делают подпроцессы. А еще можно вообще всегда использовать -f! Я вот всегда так делаю

p значит PID (номер процесса)

«БОЖЕЧКИ МОИ! Я ЗАПУСТИЛ ПРОГРАММУ ШЕСТЬ ЧАСОВ НАЗАД, А ТЕПЕРЬ МНЕ НУЖНО ЗАПУСТИТЬ STRACE ДЛЯ НЕЁ»

Не волнуйся! Тебе просто нужно найти PID (process identification number, например, 747) и запустить это:

```
strace -p 747
```

Помни: если процесс запущен от root'а, то у тебя должны быть полномочия суперпользователя, потому что безопасность

-s значит строки (strings)

Иногда я смотрю на полученные данные от команды gecvfgom, и они выглядят так: gecvgom (6, "A potom monstr kaaak...")
О нет, саспенс

`strace -s 800` покажет первые 800 символов каждой строки. Я использую этот параметр вообще всегда ★

-o значит выходные данные (output)

Давай начистоту, что бы кто ни говорил, Strace всегда выдаёт тонну данных. Используй это:

```
strace -o sliskom_mnogo_hreni.txt
```

Разберёшься с ними позже!

Не знаешь, к какому файлу обращается файловый дескриптор «3»? `-y` параметр в более новых версиях Strace. вместо номеров он покажет имена файлов!

подведём итоги

Хочешь проследить за ssh-сеансом?

```
strace -f -o ssh.txt ssh julia.box.com
```

Увидеть, какие файлы открывает процесс Dropbox sync? (с PID: 230)

```
strace -f -p 230 -e open
```

Поздравляю! Теперь ты

А если серьёзно, то, конечно, ещё много чего можно узнать об операционных системах, на других, более высоких уровнях колдунства. Однако я считаю, что даже простой Strace, сам по себе – удивительно полезный инструмент.

А какой он классный! Однажды я ехала в поезде из Нью-Йорка в Монреаль. Поезд шёл 12 часов, а у меня даже книги не было с собой. Только ноутбук без доступа к Интернету. Тогда я просто начала запускать Strace в программах на своём ноутбуке, и у меня получилось понять, как работает "killall", не читая исходный код или что-либо ещё.

А ещё он всегда помогает мне с отладкой ♡

★ Счастливого использования Strace! ★

Ссылки и FAQ

Я написала уже семь постов про Strace. У меня что-то типа наваждения. Найти их ты можешь здесь:

`jvns.ca/categories/strace`

(Не)часто задаваемые вопросы:

В: Есть ли Strace для OS X?
О: Нет, но попробуй dtruss/dtrace!

В: Можно ли применить Strace к самому Strace?
О: Канеш! Ты увидишь, что Strace использует системный вызов ptrace, на котором основана вся магия!

В: Нужно ли запускать Strace применительно к рабочей базе данных?
О: Нененененене. Это очень сильно её замедлит.

В: Есть ли возможность отслеживать системные вызовы так, чтобы это не замедляло работу программ?
О: На более новых версиях Linux иногда можно использовать `perf trace`.

♡

Понравилось?
Найти больше можно здесь!
http://jvns.ca/zines

Переведено командой FirstVDS

Другие переводы комиксов Джулии тут
https://firstvds.ru/blog/julia_evans