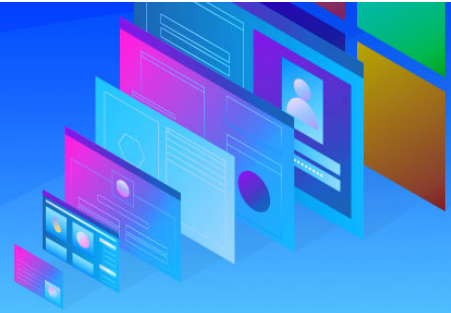


[КАК СТАТЬ АВТОРОМ](#)[Зарплаты айтишников](#)[Почему мы бросаем IT-курсы?](#)

Лучшее предложение на рынке  
VPS хостинг с Windows от **523** ₽/месяц

Лицензия на ОС включена в стоимость



Реклама ООО «ИТ Финанс», едд: LUYKAPRS

**2478.63**

Рейтинг

**RUVDS.com**VDS/VPS-хостинг. Скидка 15% по коду **HABR15**[Подписаться](#)**ru\_vds**

21 ноя 2020 в 16:15

## Кунг-фу стиля Linux: наблюдение за файловой системой



6 мин



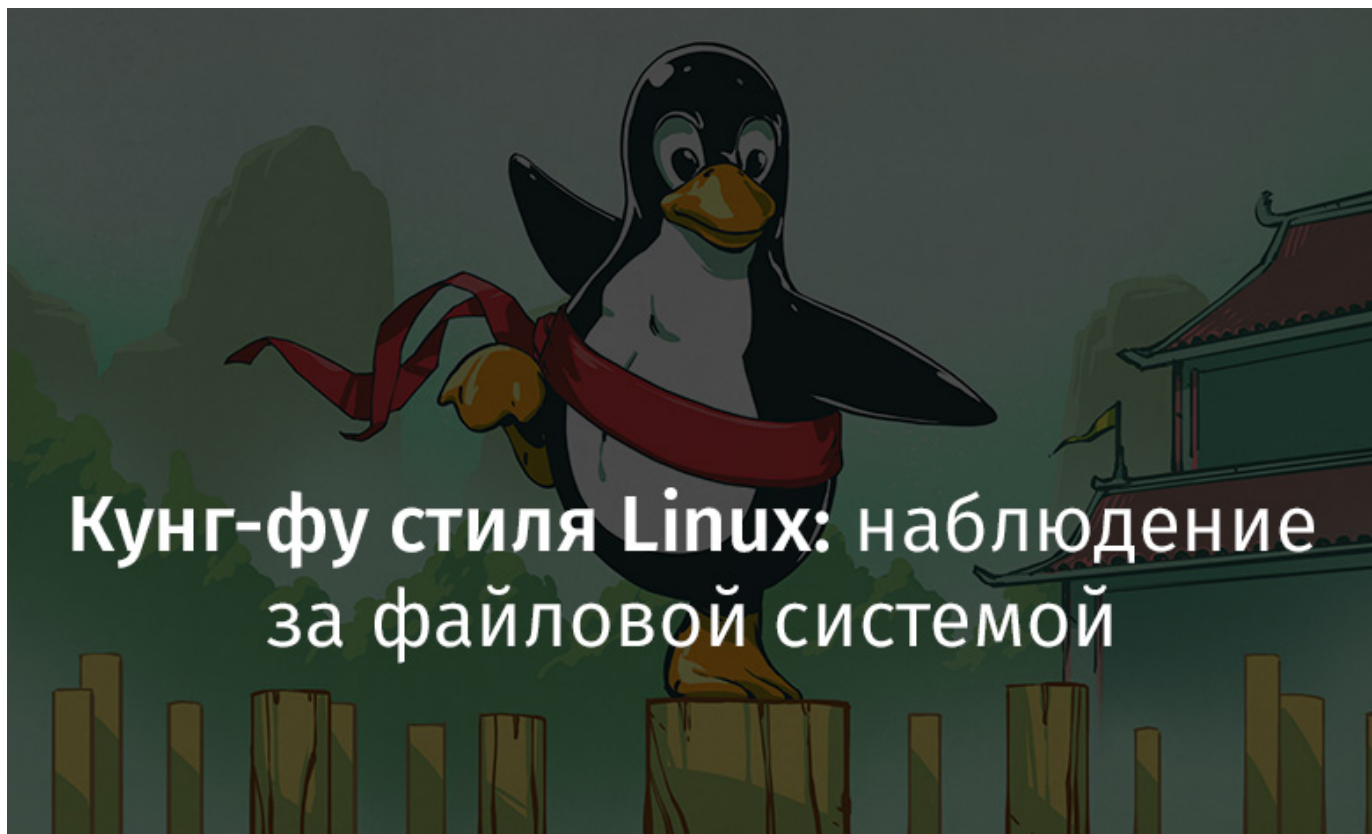
19K

Блог компании RUVDS.com, Настройка Linux\*, Системное администрирование\*

[Перевод](#)

Автор оригинала: Al Williams

Пользователи UNIX-подобных ОС, чтобы достичь желаемого результата, привыкли организовывать совместную работу различных программ, рассчитанных на решение какой-то одной задачи. Эти команды, например, помещают в файл Bash-скрипта, а потом, по своей инициативе, вызывают этот скрипт из командной строки. А что если нужно, чтобы система сама реагировала бы на какие-то изменения, действуя без вмешательства пользователя? Например, нужно организовать наблюдение за директорией и автоматически вызывать какую-то программу тогда, когда в эту директорию будет скопирован файл с использованием FTP. При этом не хотелось бы сидеть за компьютером во время передачи этого файла и ждать завершения операции.



► [Все переводы серии](#)

Самый простой, но не очень-то красивый способ это сделать заключается в периодическом сканировании директории. Вот весьма примитивный скрипт, который реализует такой

◆ +39

📖 120



💬 8

```
#!/bin/bash
while true
do
  for I in `ls`
  do cat $I; rm $I
  done
  sleep 10
done
```

В этом примере я вывожу содержимое файла в консоль и удаляю его. Но в реальной жизни с подобным файлом можно сделать что-нибудь куда более интересное. Это – пример того, как не стоит писать скрипты. Ведь скрипт всё время выполняется. К тому же, такое решение задачи нельзя назвать красивым или остроумным. (Если вы полагаете, что в данном примере мне стоило бы использовать конструкцию `for I in *` – попробуйте это сделать в пустой директории и вы поймёте причину использования команды `ls`.)

## Как организовать наблюдение за файловой системой в Linux?

### Более симпатичное решение задачи

Если говорить честно, то для решения нашей задачи хорошо было бы сделать нечто, выглядящее лучше вышеприведённого примера. В современных ядрах (начиная с 2.6.13) есть механизм уведомлений о событиях файловой системы, представленный интерфейсом `inotify`. Соответствующие вызовы можно использовать программно, применяя заголовочный файл `sys/inotify.h`. Существует и набор инструментов командной строки, который можно установить, обычно представленный пакетом `inotify-tools`.

Один из инструментов этого пакета, `inotifywait`, позволяет улучшить код нашего примера. Скажем, переделать код можно так:

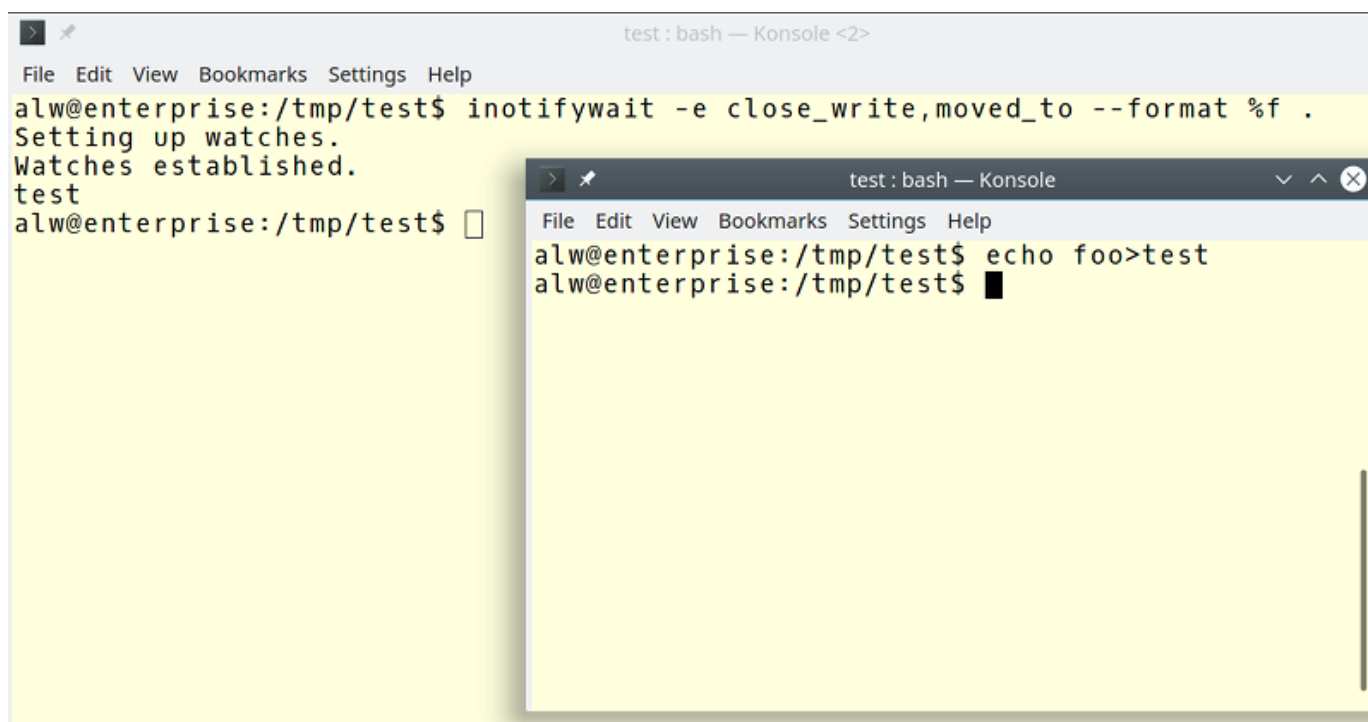
```
#!/bin/bash
while true
do
    if FN=`inotifywait -e close_write,moved_to --format %f .`
    then
        cat $FN
        rm $FN
    fi
done
```

Я думаю, что этот вариант лучше предыдущего. Скрипт вызывается не через заданные промежутки времени, а только тогда, когда в файловой системе что-то произошло. Я полагаю, что любая нормальная программа, воздействующая на файлы, находящиеся в директории, либо открывает файлы для записи и закрывает их после завершения операции, либо перемещает файлы в директорию. Наша программа отреагирует на оба эти события, а благодаря `%f` команда сообщит имя файла. Есть, конечно, и другие события, за которыми можно наблюдать.

Если вас интересует вопрос о том, почему необходимо наблюдать за событием перемещения файла, подумайте о том, как работает большинство текстовых редакторов и программ для загрузки файлов по сети. Обычно новому файлу не дают постоянного имени до тех пор, пока не завершится его обработка. Например, Chrome будет загружать файл с именем `test.txt`, дав ему временное имя `test.txt.crdownload` (или имя, подобное этому). Файл будет переименован (перемещён) в файл `test.txt` только после завершения загрузки.

Если вы хотите попробовать команду `inotifywait` без написания скрипта, чтобы

своими глазами увидеть то, как она работает, откройте пару окон терминала и воспроизведите в этих окнах то, что показано ниже.



```
test : bash — Konsole <2>
File Edit View Bookmarks Settings Help
alw@enterprise:/tmp/test$ inotifywait -e close_write,moved_to --format %f .
Setting up watches.
Watches established.
test
alw@enterprise:/tmp/test$ █

test : bash — Konsole
File Edit View Bookmarks Settings Help
alw@enterprise:/tmp/test$ echo foo>test
alw@enterprise:/tmp/test$ █
```

### Исследование команды `inotifywait`

В окне, расположенном на заднем плане, выполните команду `inotifywait`. Не забудьте о точке в конце команды, которая указывает на то, что наблюдать надо за файлами в текущей директории. Затем, в другом окне терминала, создайте в той же самой директории файл. В окне первого терминала будет выведено имя файла, после чего `inotifywait` завершит работу. В скрипте используется тот же самый механизм, с его помощью скрипт записывает имя файла в переменную `FN`, выполняет некие действия и перезапускает `inotifywait`. Кстати, можно сделать так, чтобы утилита `inotifywait`, после срабатывания, не завершала бы работу, но это усложнит скрипт. Это, правда, устранил проблему, которая может возникнуть в том случае, если имя файла меняется в ходе попытки его обработать средствами скрипта.

Ещё один инструмент для наблюдения за файловой системой, `inotifywatch`, тоже реагирует на изменения файлов, но он осуществляет мониторинг файловой системы в течение некоторого времени, а потом выдаёт сводку по зафиксированным изменениям. Если вы полагаете, что это именно то, что вам нужно, почитайте справку по `inotifywatch`.

### Новый скрипт

Хотя мы и улучшили наш скрипт, он всё ещё далёк от идеала. Вполне возможно то, что нужно организовать наблюдение за множеством директорий. Вряд ли кому-то захочется

воспроизводить вышеописанную последовательность действий в применении к каждой интересующей его директории.

Для решения нашей задачи можно воспользоваться и ещё одной программой, называемой `incron` (я почти уверен в том, что вам эта программа пригодится, и что её стоит установить). Эта утилита похожа на `cron`, но вместо того, чтобы выполнять какие-то действия, ориентируясь на события, связанные с временем, она ориентируется на события, связанные с файлами. После её установки вам, вероятно, если вы планируете ей пользоваться, стоит отредактировать файлы `/etc/incron.allow` и `/etc/incron.deny`. Особенно – если вы будете применять её, работая как обычный пользователь.

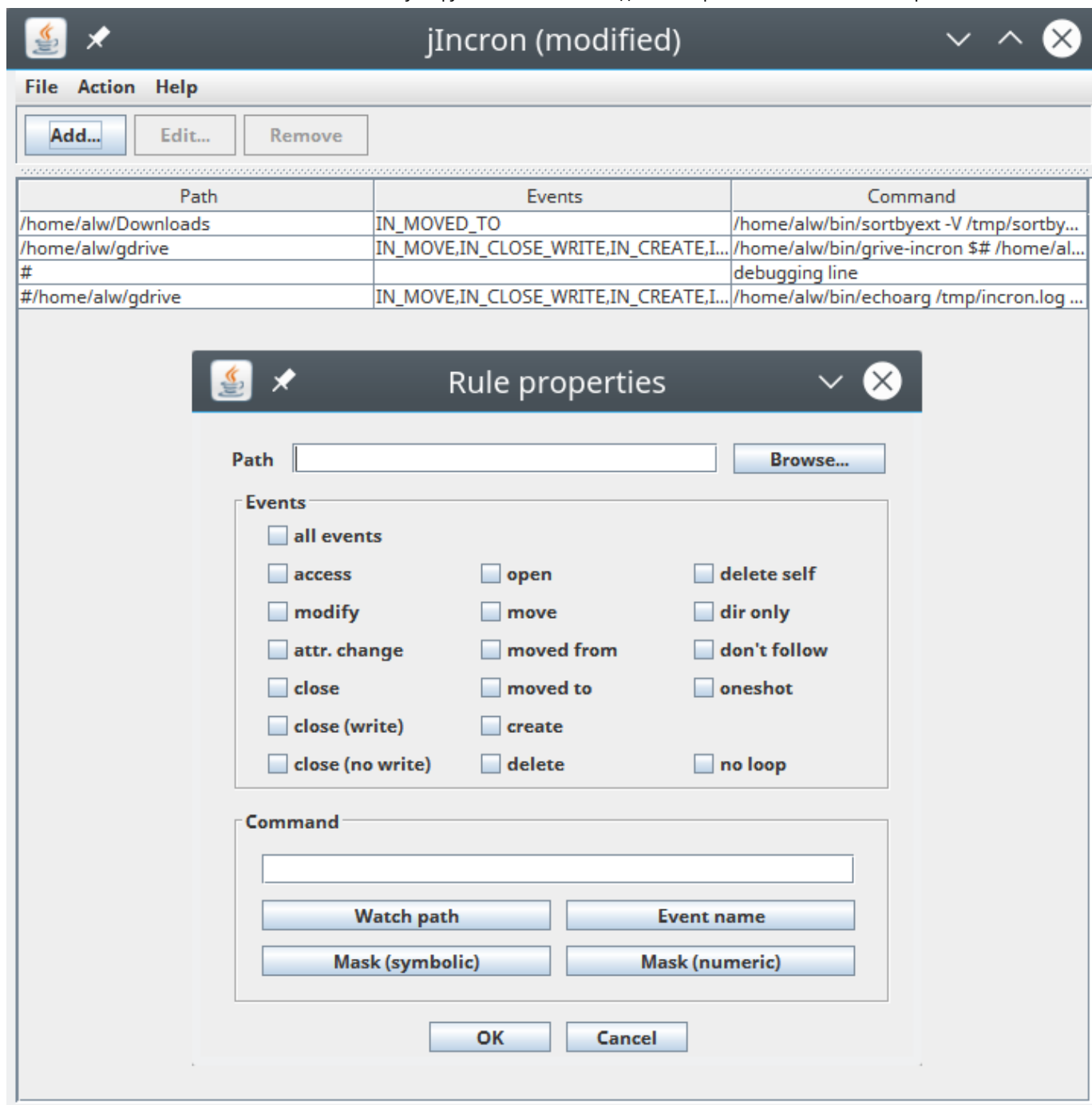
Предположим, нам надо выполнять некий скрипт в том случае, если в директории `hexfiles` появится какой-нибудь файл. Для редактирования таблицы событий, за которыми наблюдает `incron`, можно воспользоваться командой `incrontab -e`. Данные, вносимые в эту таблицу, представляют собой описания задач и по-особенному форматируются. В частности, тут для разделения столбцов используются не знаки табуляции, а пробелы. Вот строка, которая позволяет решить вышеописанную задачу:

```
/home/alw/Downloads/hexfiles IN_CLOSE_WRITE,IN_MOVED_TO /home/alw/bin/program_cpu $
```

Конструкция `$/#$` в конце этой строки позволяет получить полный путь к файлу, на событие, произошедшее с которым, отреагировала программа. Тут, кроме того, можно получить время возникновения события – либо в виде текста (`$%`), либо в виде числа (`$&`). С помощью `incron` можно наблюдать за обычными событиями. Утилита, кроме того, поддерживает различные опции. Например, можно не разыменовывать символические ссылки. Подробности об `incron` ищите в справке к утилите.

## Настройка `incron` с использованием графического интерфейса

Я – не большой любитель Linux-инструментов с графическим интерфейсом, но я знаю, что многим они нравятся. Если вы – из их числа – вот `incrontab`-редактор, написанный на Java. Нельзя сказать, что этот проект отличается подробной документацией, но его использование упрощает то, что в него можно импортировать файл `incrontab`. Если он у вас есть – ищите его по пути `/var/spool/incron/your_user_id`. Если посмотреть на окно редактора, показанное ниже, можно заметить, что он позволяет упростить создание `incron`-таблицы.



### Графический интерфейс для incron

Обычно системные файлы можно найти в `/etc/incron.d`. Пути расположения файлов можно задавать в файле `/etc/incron.conf`. Поэтому, если вам нужно поменять место расположения файла таблицы, но вы не знаете о том, где его искать, загляните в этот файл.

## Итоги

Использование `incron` выглядит как очень аккуратное решение задачи мониторинга файловой системы. Системная программа берёт на себя заботу о наблюдении за событиями файловой системы, а скрипт запускается лишь тогда, когда это нужно. Кроме того, используя эту программу, можно легко узнать о том, за чем именно организовано

наблюдение. В результате `inotify` – это инструмент, с помощью которого можно сделать куда больше, чем наблюдение за текущей папкой в открытом окне терминала.

Планируете ли вы пользоваться `inotify`?

► [Все переводы серии](#)

# Habrahabr10

Промокод для скидки 10% на виртуальные серверы RUVDS



## ЗАКАЖИТЕ СЕРВЕР И ПОЛУЧИТЕ `//ispmanger` В ПОДАРОК

БЕСПЛАТНО  
3 месяца на любом тарифе

 RUVDS

**Теги:** Linux, системное администрирование

**Хабы:** Блог компании RUVDS.com, Настройка Linux, Системное администрирование

## Редакторский дайджест

Присылаем лучшие статьи раз в месяц

Электронная почта



RUVDS.com

VDS/VPS-хостинг. Скидка 15% по коду **HABR15**

Telegram ВКонтакте Twitter

520 242.6



Карма   Рейтинг

@ru\_vds

Пользователь

Подписаться



## Комментарии 8



vaniaser  
21 ноя 2020 в 22:27

Название инкрон как бы намекает на использование крона, это обертка для крона?  
Почему просто не использовать крон?

♦ -2   Ответить



onix74  
21 ноя 2020 в 23:07

По-моему, в статье есть ответ. сгон – это то, что по расписанию (по времени).  
insgon – это то, что по событию.

♦ 0   Ответить



KodyWiremane  
2 дек 2020 в 20:53

А ещё у сгон'а есть брат апасгон – для ситуаций, когда компьютер может выключаться не по расписанию, а задачи всё-таки надо запускать более-менее регулярно. Просто оставляю это здесь, вдруг кто-то не знает.

♦ 0   Ответить



astenix  
22 ноя 2020 в 03:30

У вас в примере

```
inotifywait -e close_write,moved_to --format %f .
```

параметр «-e» был превращён парсером в «-е».

Соответственно и внезапный ответ

```
Couldn't watch -e: No such file or directory
```

♦ 0   Ответить





**Karpion**

22 ноя 2020 в 03:56

Если вы полагаете, что в данном примере мне стоило бы использовать конструкцию `for I in *` – попробуйте это сделать в пустой директории и вы поймёте причину использования команды `ls`.

Попробуйте создать в директории файл с пробелом в имени – и Вы поймёте, как Вы ошиблись.

+5 Ответить

**ZyXI**

22 ноя 2020 в 04:26

Кроме того, т.к. в начале скрипта написано `#!/bin/bash`, то в этом скрипте можно просто использовать `shopt -s nullglob` и никаких проблем с пустыми каталогами не будет.

+3 Ответить

**Oxyd**

29 ноя 2020 в 04:59

Я просто оставляю это здесь...

```
man systemd.path
```

0 Ответить

**kt97679**

30 ноя 2020 в 18:54

Цитата из мана:

Internally, path units use the `inotify(7)` API to monitor file systems. Due to that, it suffers by the same limitations as `inotify`, and for example cannot be used to monitor files or directories changed by other machines on remote NFS file systems.

0 Ответить



Зарегистрируйтесь на Хабре, чтобы оставить комментарий

## Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ

**RationalAnswer**

9 часов назад

## Самые дикие налоговые проблемы, в которые встряли уехавшие из России айтишники в 2024

10 мин

16K

[Обзор](#)

+53

45

132

**tguev**

10 часов назад

## Zen of Python: история, реализация и пасхалки

Простой

6 мин

2.4K

+35

48

2

**anton-artemenko**

5 часов назад

## Что изменилось за последние 25 лет в медицине и почему вас часто лечат по старинке

9 мин

2.6K

+34

14

19

**mark\_ablov**

9 часов назад

## Щупаем первый 8-битный процессор от Intel – 8008

Простой

8 мин

3.6K

[Кейс](#)

+28

20

0

**rust\_salikhov**

9 часов назад

## Оптимизатор монтажей прокатного стана-2000

13 мин

874

+28

5

1

**krivotester**

8 часов назад

## Как я превратил обычный бизнес в IT компанию, читая книги

**Простой**

14 мин



2.1K

Тutorial

**+27**

31



6

**galqiwi**

7 часов назад

## Как развивалась технология экстремального сжатия LLM: от QuIP до AQLM с PV-tuning



4 мин



1.9K

**+23**

17



4

**melnik909**

7 часов назад

## Магия CSS на практике: советы по вёрстке от гика. Часть 2

**Средний**

8 мин



1.1K

Tutorial

**+19**

28



2

**inetstar**

3 часа назад

## Слияние словарей в PyTorch: зачем нужно и подводные камни

**Средний**

10 мин



401

Tutorial

**+18**

8



3

**ManegSPb**

6 часов назад

## Приватный Kubernetes за 50 минут



13 мин



1.6K

Показать еще

ВАКАНСИИ КОМПАНИИ «RUVDS.COM»

Ведущий системный администратор  
от 200 000 до 250 000 ₽ · RUVDS.com · Москва · Можно удаленно

Больше вакансий на Хабр Карьере

ИНФОРМАЦИЯ

Ваш аккаунт	Разделы	Информация	Услуги
Войти	Статьи	Устройство сайта	Корпоративный блог
Регистрация	Новости	Для авторов	Медийная реклама
	Хабы	Для компаний	Нативные проекты
	Компании	Документы	Образовательные
	Авторы	Соглашение	программы
	Песочница	Конфиденциальность	Стартапам



Настройка языка

Техническая поддержка

© 2006–2024, Habr

ruvds.com

VPS (CPU 1x2ГГц, RAM 512Мб, SSD 10 Gb) – 190 рублей в месяц  
ruvds.com

VPS Windows от 523 рублей в месяц. Бесплатный тестовый период 3 дня.

ruvds.com

VDS в Цюрихе. Дата-центр TIER III – швейцарское качество по низкой цене.

ruvds.com

Антивирусная защита виртуального сервера. Легкий агент для VPS.

ruvds.com

VPS в Лондоне. Дата-центр TIER III – английская точность за рубли.

ruvds.com

VPS с видеокартой на мощных серверах 3,4ГГц

ruvds.com

## ПРИЛОЖЕНИЯ

---



### RUVDS Client

Приложение для мониторинга и управления виртуальными серверами RU VDS с мобильных устройств.

Android iOS

## ВИДЖЕТ

**RU VDS**  
Лучшее предложение  
на рынке

**VPS с WINDOWS**  
от 523 ₽/месяц

Лицензия на ОС включена в стоимость

Реклама. ООО «МТ ФИНАНС». erid: LjN8KDmrk

## БЛОГ НА ХАБРЕ

---

3 часа назад

Слияние словарей в PyTorch: зачем нужно и подводные камни

 401  3

---

7 часов назад

Магия CSS на практике: советы по вёрстке от гика. Часть 2

 1.1K  2

---

вчера в 20:00

Профессиональные игроки в покер знают оптимальную стратегию, но не всегда её используют

 7.6K  4

---

вчера в 16:00

Война YouTube против блокировщиков рекламы

 34K  298

---

21 июл в 20:00

Памяти Кевина Митника – хакера, ломавшего ФБР, АНБ и Кремниевую долину. Часть 9: туман Сиэтла, саспенс и чёрный вертолёт

 7.6K  7