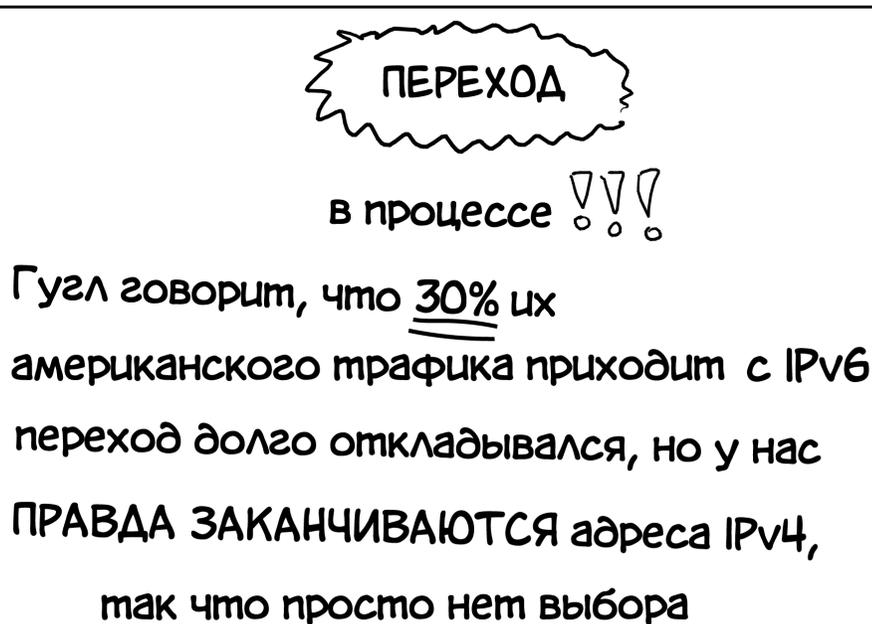
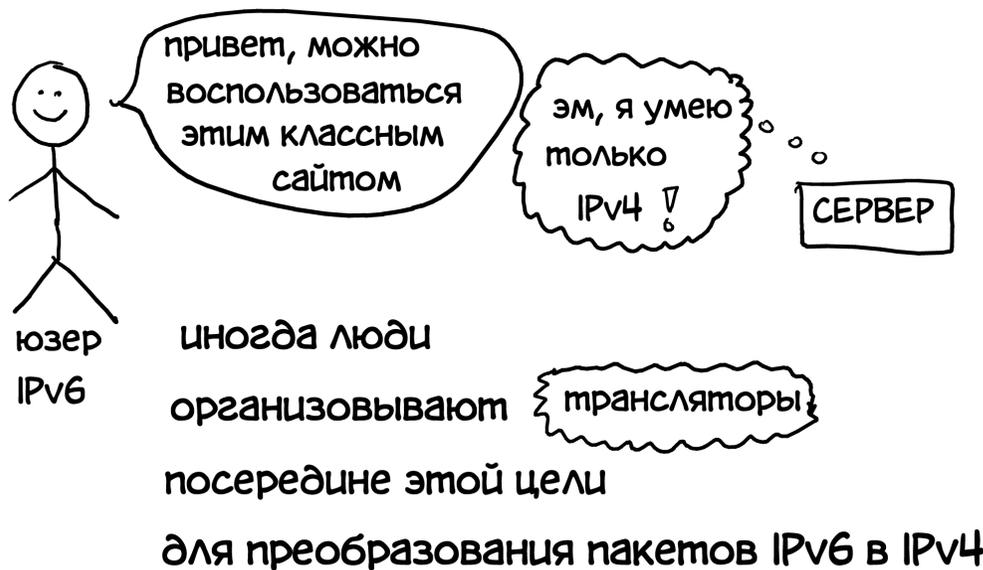
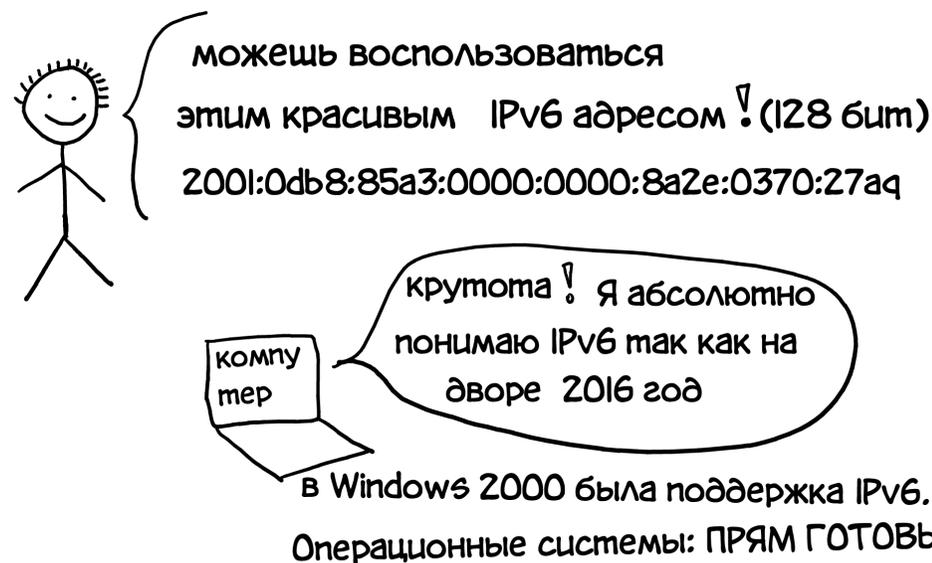
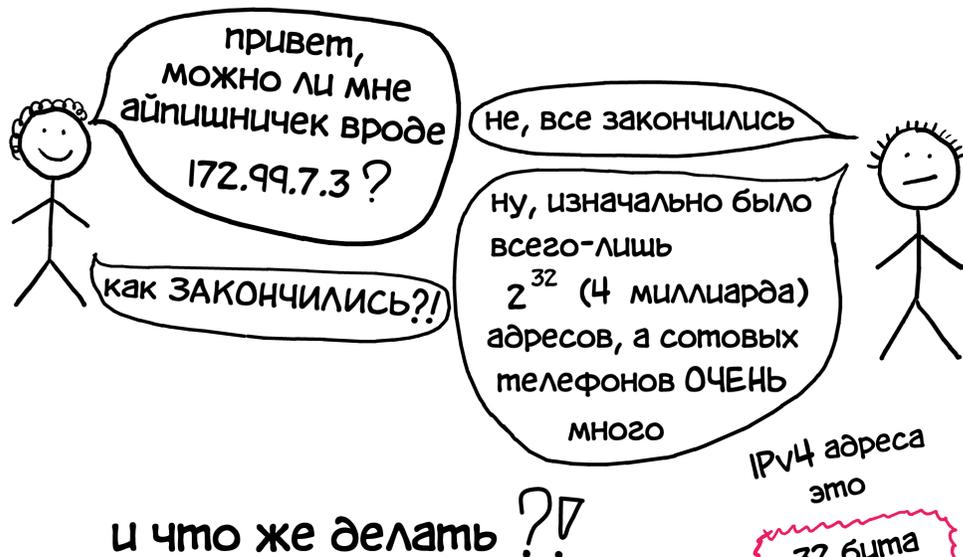


# IPv6

JULIA EVANS  
@b0rk

drawings.jvns.ca



# распределение памяти

Julia Evans  
@b0rk

В любой момент вашей программе доступно ограниченное количество памяти

587 MB

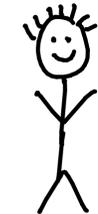
■ занято

□ свободно



тут было занято, но освободилось

и она может попросить ОС выделить больше



google  
chrome

теперь у меня есть 1,8 Gb!  
офигеть!

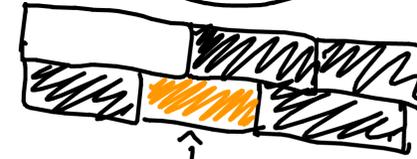
ваш распределитель пытается заполнить неиспользуемое пространство когда поступает запрос на память



можно мне 512 байт памяти?

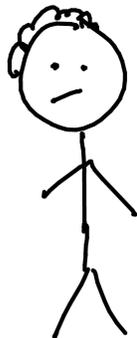
ДА

malloc



тут ваша новая память

вы можете изобрести собственную стратегию распределения памяти



алгоритм malloc в libc не очень, поэтому я сделаю всё сам

особенно актуально, если вы хорошо понимаете свою модель доступа к памяти

это, типа, нормально, если вам **ОЧЕНЬ** важна производительность

альтернативы malloc в libc

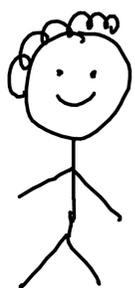
jemalloc

tcmalloc  
Google

# мужские страницы офигенны

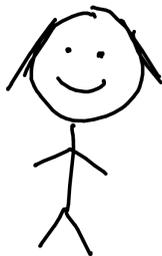
JULIA EVANS  
@b0rk

страницы man=класс (иногда. качество варьируется)



оказывается, я могу увидеть документацию для программ (типа grep) если я выполню **man grep**!

но это ещё не всё! страницы man есть для множества других штуквин!



man страницы объединены в 8 разделов

- ①
- ②
- ③
- ④
- ⑤
- ⑥
- ⑦
- ⑧

/usr/share/man/man 5

находится 5 раздел на моем компьютере

## ① программы

\$ man grep  
\$ man ls

## ② системные вызовы

\$ man sendfile

## ③ функции C

\$ man 3 printf  
\$ man fopen

## ④ устройства

\$ man null  
для /dev/null docs

## ⑤ файловые форматы

\$ man sudores  
для /etc/sudores  
круто → \$ man proc

## ⑥ games (не очень полезно)

но man si хорош, если у вас есть si

## ⑦ разное

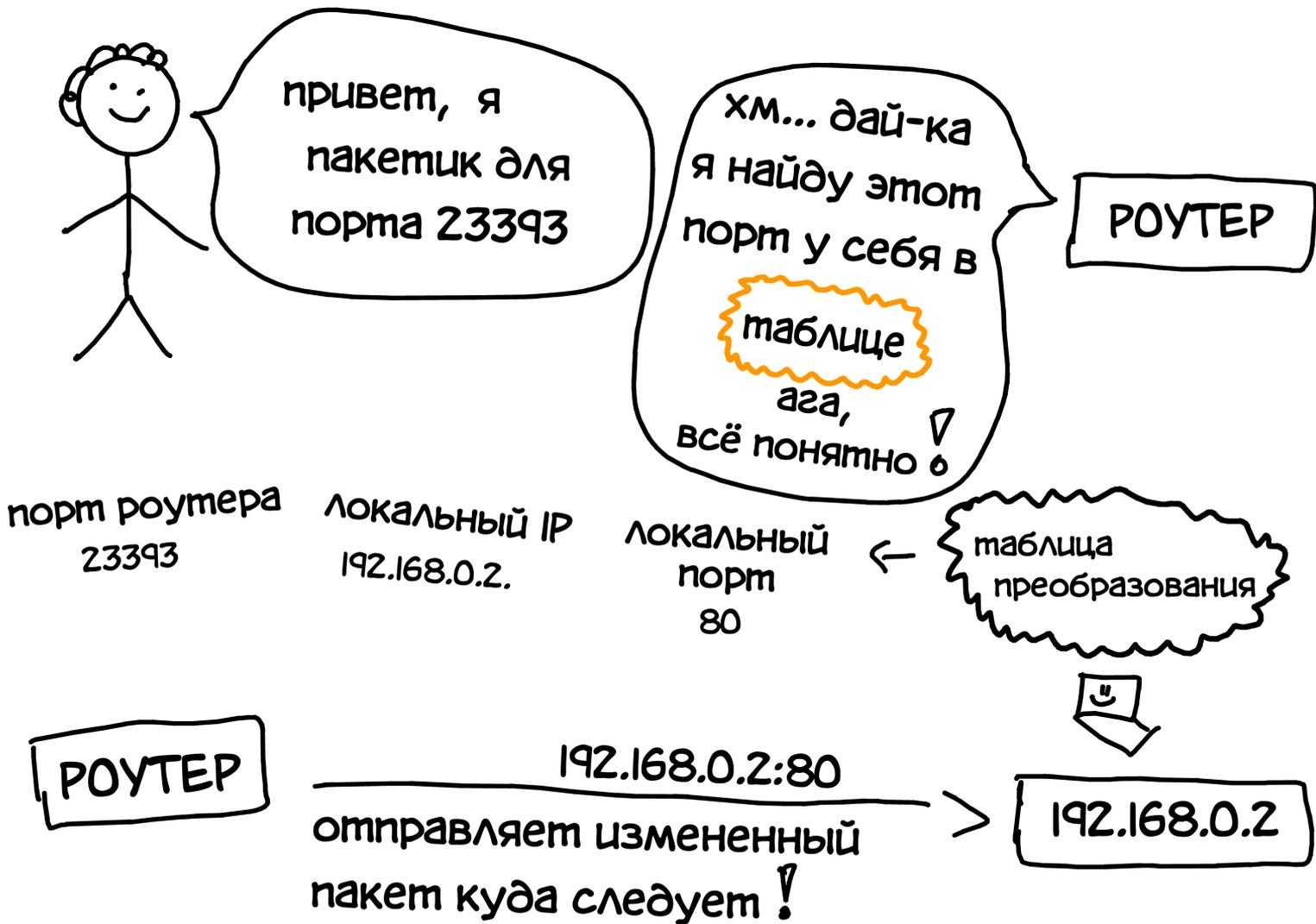
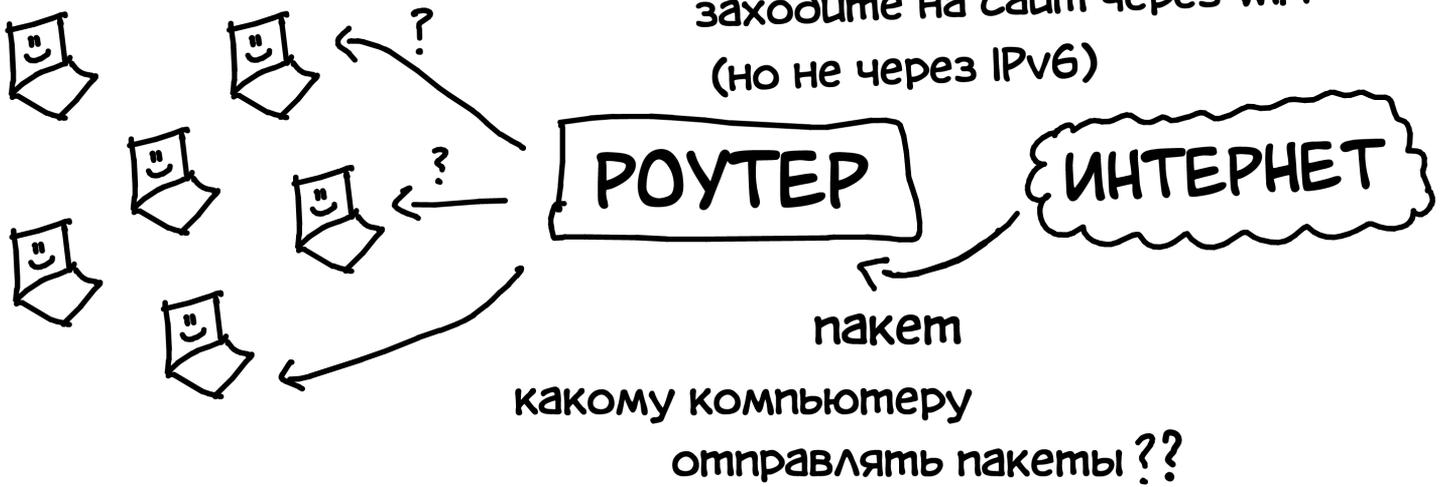
\$ man 7 pipe  
\$ man 7 symlink  
(вот эти клёвые!)

## ⑧ программы сисадмина

\$ man apt  
\$ man chroot

# ПРЕОБРАЗОВАНИЕ СЕТЕВЫХ АДРЕСОВ

происходит каждый раз,  
когда вы, например,  
заходите на сайт через wifi  
(но не через IPv6)



# права в unix

3 типа вещей, которые  
можно сделать с файлом

↓  
**r**ead (читать)  
↓  
**w**rite (записывать)  
↓  
**x**ecute (исполнять)

```
$ ls -l awesome.png
```

```
rw-r-- bork staff
```

↑  
bork может  
делать вот  
это  
(**юзер**)

↑  
staff может  
делать это  
(**группа**)

←  
**КТО УГОДНО**  
может делать  
это

```
$ ls -l /bin/ping
```

```
rwsr-xr-x root root
```

↑  
**setuid** флаг

Это значит что ping всегда  
выполняется из под root (его  
владелец), независимо от того,  
кто выполнил команду.

что за дела с этими  
755?

7	значит	rwX
6	→	rw-
5	→	r-x
4	→	r--

оно бинарно?

5 → 101 → r-x

755 это  
rwX r-x r-x

разные  
странные  
права

setgid

sticky bit

Но тут  
закончилось  
место

# конвейеры

JULIA EVANS  
@b0rk

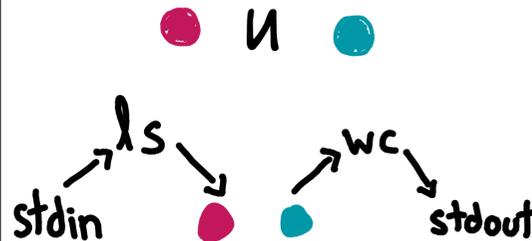
drawings.jvns.ca

иногда требуется  
направить вывод одного  
процесса на ввод другого

```
$ ls | wc -l
```

53  
53 файла!

Конвейер - это пара  
магических файловых  
дескрипторов



когда `ls` выполняет  
`write (●, "hi")`  
`wc` может его  
прочитать

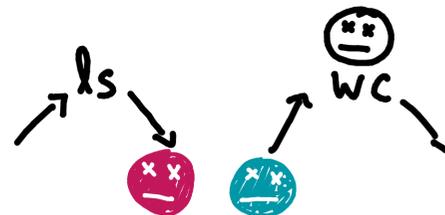
```
read (● )  
→ "hi"
```

## буфер конвейера

ls: сейчас я запишу  
базилцион байт  
в ●

эм, нет, если у меня  
заполнен буфер,  
то тебе придется  
подождать ●

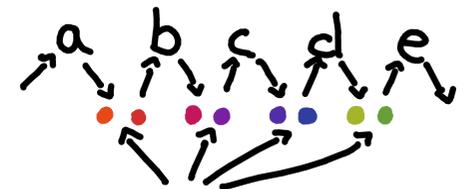
что если целевой  
процесс помрет?



ls отправляется сигнал  
SIGPIPE если ●  
закроется (LS) обычно  
умирает

в конвейер можно  
запихнуть  
ОЧЕНЬ МНОГО  
всякого разного

```
$ a | b | c | d | e
```



конвейерные пары

@bork

Julia Evans

# из чего состоит процесс

? !

**ПЕРЕМЕННЫЕ ОКРУЖЕНИЯ**

вроде PATH или LD\_LIBRARY\_PATH

**РАБОЧИЙ КАТАЛОГ**

/home/bork

**PID**

39762

**БИНАРНИК**

(код ассемблера)

**ЮЗЕР**

и который его выполнил

**ГРУППА**

**ПАМЯТЬ**

**ОТКРЫТЫЕ ФАЙЛЫ**

включая сетевые соединения

**РЕГИСТРЫ**

**ОБРАБОТЧИКИ СИГНАЛОВ**

знаете что происходит, если нажать ctrl+c ?

**РОДИТЕЛЬСКИЙ ПРОЦЕСС**

29931

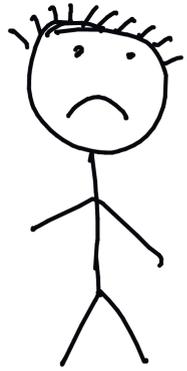
а иногда и

**NAMESPACES and CGROUPS**

**ДОЧЕРНИЕ ПРОЦЕССЫ**

JULIA EVANS  
@b0rk

# ♥ TSP ♥

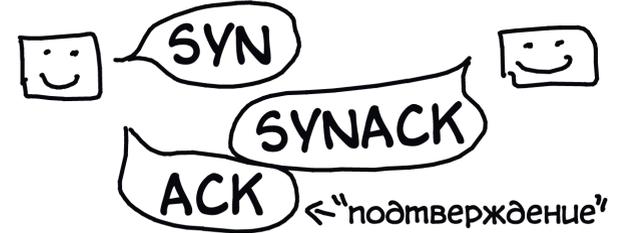


хочу отправить своему другу картинку котика, но у нас плохое сетевое соединение!  
Пакеты теряются, что же МНЕ ДЕЛАТЬ?!

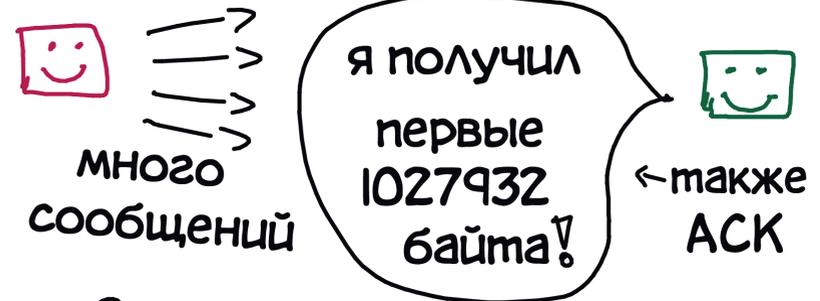
! ☆  
TSP придет на помощь  
! ☆

## как работает TSP

① открываем соединение



② следим за тем, что было успешно отправлено



③ попробуем снова, если нужно



④ закрываем соединение  
(еще тут есть контрольные суммы)

TSP - сетевой протокол, позволяет надежно передавать данные! Каждый раз, открывая страничку, вы делаете это по TSP

Хабрахабр

← доставляется по  
♥ TSP ♥

Перевела Команда FirstVDS.ru