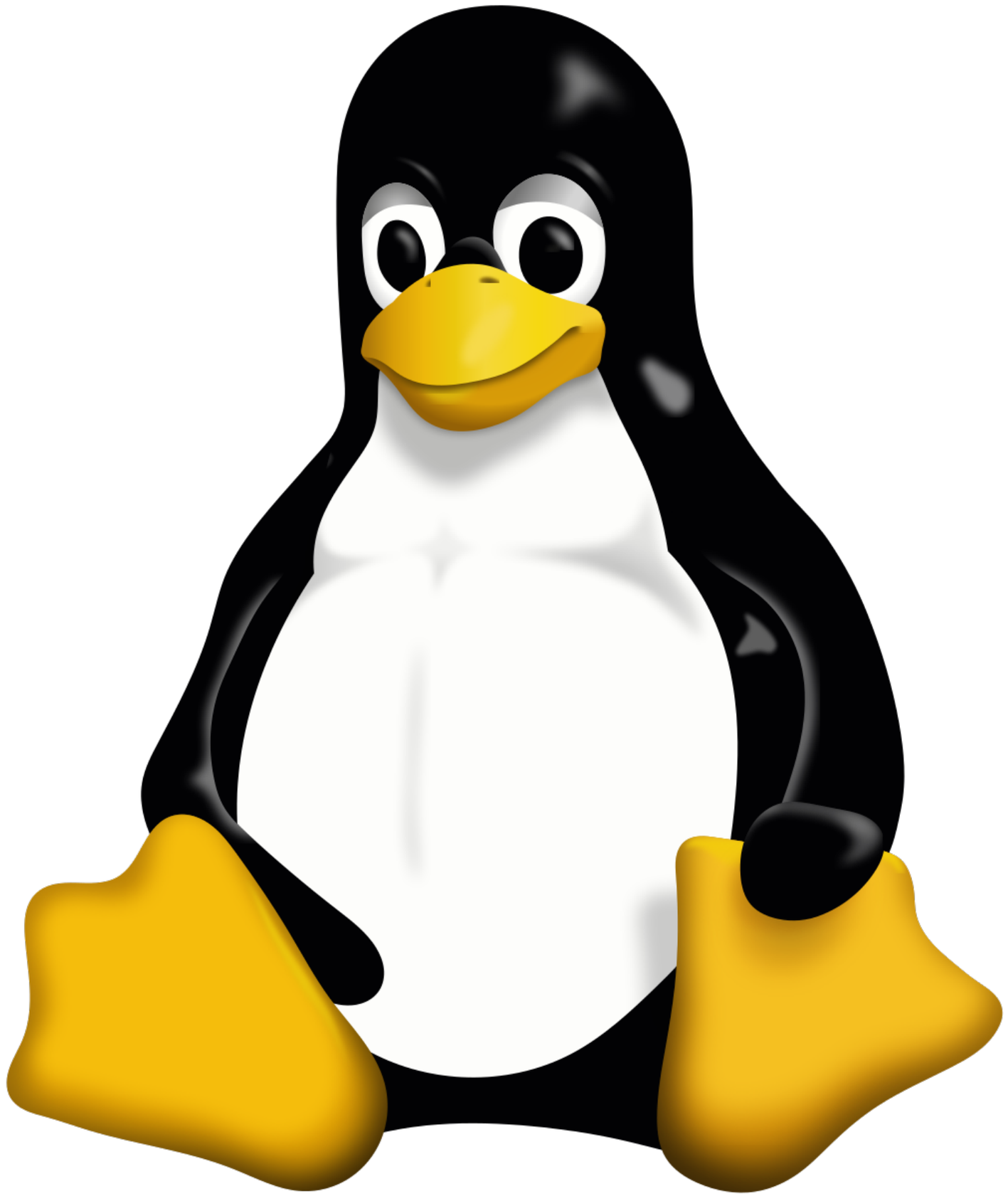
Никита Соболев

github.com/sobolevn



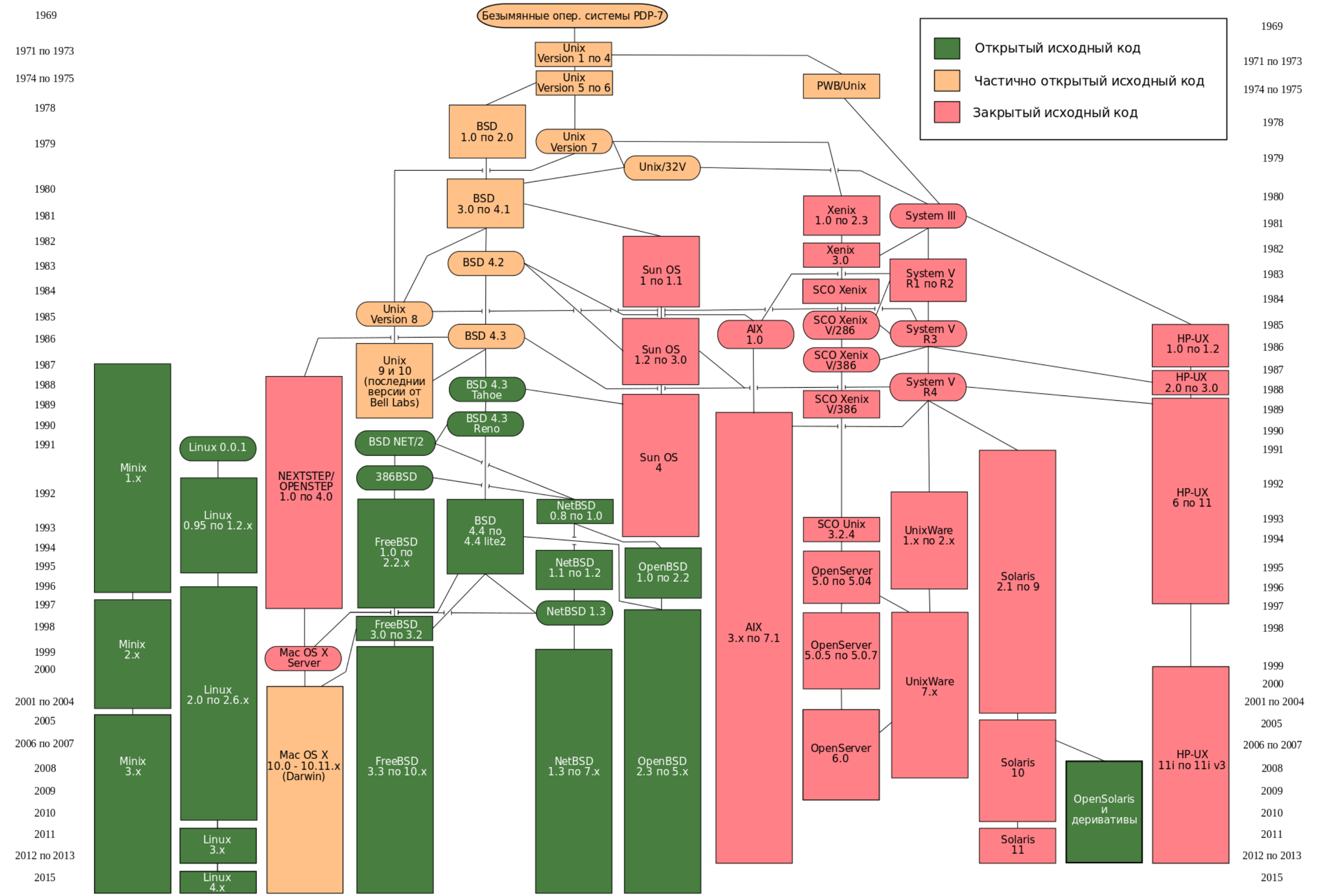
Linux

План занятия

- История и общая информация о Linux
- Работа с командой строкой
- Файлы и папки
- Пользователи и группы

Linux = представители семейства Unix-подобных операционных систем на общем ядре

Что такое "Unix-подобные"?


Открытый исходный код
Частично открытый исходный код
Закрытый исходный код
1969
1971 по 1973
1974 по 1975
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 по 2004
2005
2006 по 2007
2008
2009
2010
2011
2012 по 2013
2015
Безымянные опер. системы PDP-7
Unix Version 1 по 4
Unix Version 5 по 6
PWB/Unix
BSD 1.0 по 2.0
Unix Version 7
Unix/32V
BSD 3.0 по 4.1
Xenix 1.0 по 2.3
System III
Xenix 3.0
BSD 4.2
Sun OS 1 по 1.1
System V R1 по R2
SCO Xenix
Unix Version 8
SCO Xenix V/286
AIX 1.0
System V R3
BSD 4.3
Sun OS 1.2 по 3.0
HP-UX 1.0 по 1.2
SCO Xenix V/386
Unix 9 и 10 (последнии версии от Bell Labs)
BSD 4.3 Tahoe
System V R4
HP-UX 2.0 по 3.0
BSD 4.3 Reno
SCO Xenix V/386
BSD NET/2
Sun OS 4
Linux 0.0.1
386BSD
NEXTSTEP/ OPENSTEP 1.0 по 4.0
NetBSD 0.8 по 1.0
HP-UX 6 по 11
SCO Unix 3.2.4
UnixWare 1.x по 2.x
Minix 1.x
Linux 0.95 по 1.2.x
FreeBSD 1.0 по 2.2.x
BSD 4.4 по 4.4 lite2
OpenServer 5.0 по 5.04
Solaris 2.1 по 9
NetBSD 1.1 по 1.2
OpenBSD 1.0 по 2.2
AIX 3.x по 7.1
NetBSD 1.3
OpenServer 5.0.5 по 5.0.7
UnixWare 7.x
Minix 2.x
Linux 2.0 по 2.6.x
Mac OS X Server
FreeBSD 3.0 по 3.2
OpenServer 6.0
Solaris 10
OpenSolaris и деривативы
HP-UX 11i по 11i v3
Mac OS X 10.0 - 10.11.x (Darwin)
FreeBSD 3.3 по 10.x
NetBSD 1.3 по 7.x
OpenBSD 2.3 по 5.x
Minix 3.x
Linux 3.x
Solaris 11
Linux 4.x

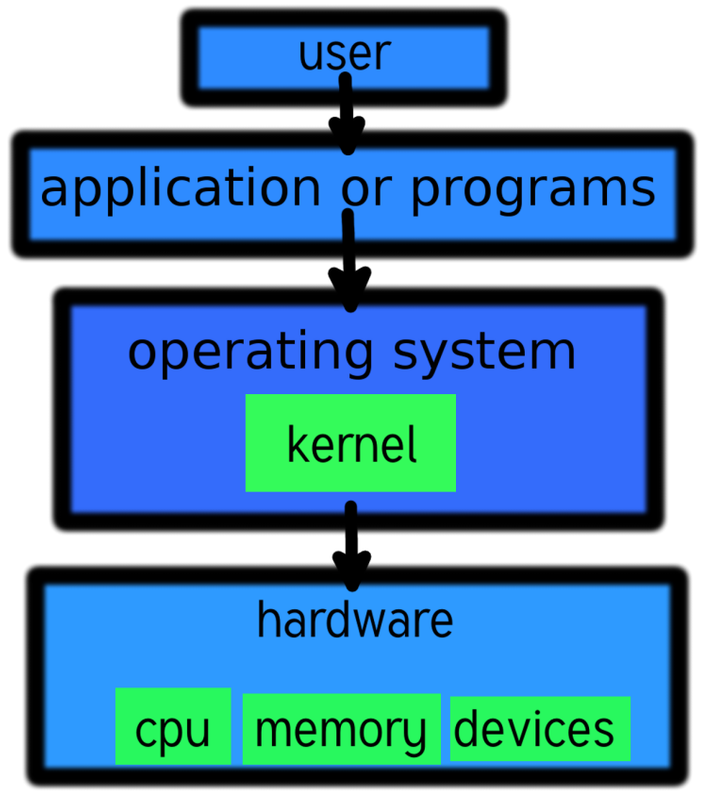
История Linux

- 1971 – Unix, Bell Labs: Кен Томпсон, Деннис Ритчи
- 1983 – GNU, Ричард Столлман
- 1991 – Linux, Линус Торвальдс

Unix way

Слои Linux

- Hardware (железо)
- Kernel (ядро)
- Userspace (рабочее окружение)



Linux = лучший в мире конструктор

Из чего состоит любой Linux?

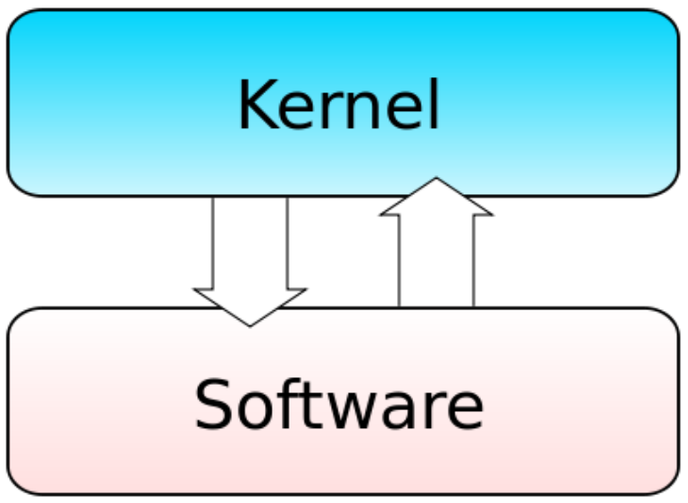
- Ядро (разных версий)
- Дистрибутив
- Среда рабочего стола (по необходимости)

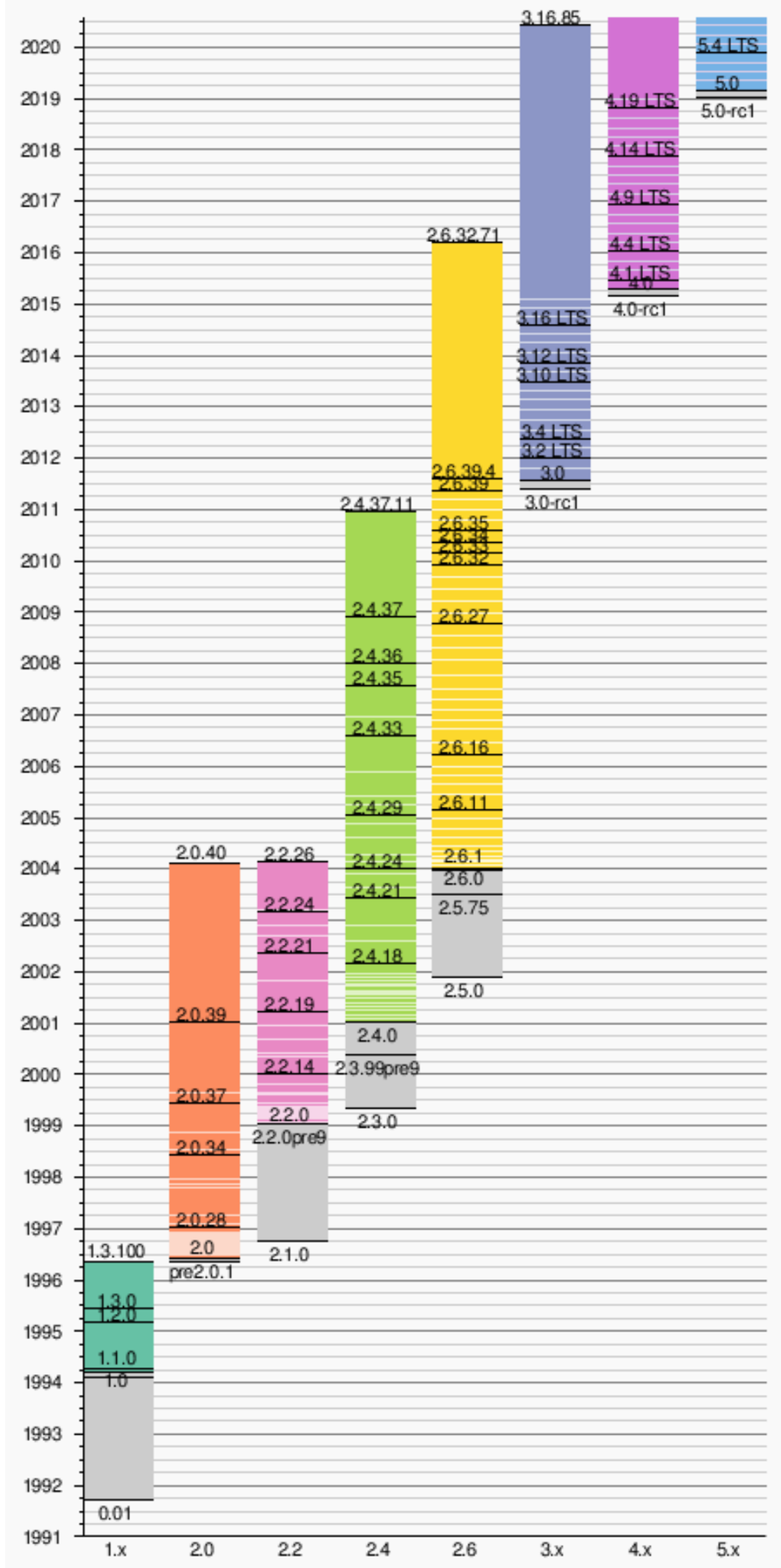
Из чего состоит любой Linux?

- **Ядро (разных версий)**
- Дистрибутив
- Среда рабочего стола (по необходимости)

Что такое "ядро"?

- Ядро позволяет софту и железу "общаться"
- В ядре спрятана основная сложность операционной системы
- Ядра бывают очень разными по возможностям, архитектуре и характеристикам




2020
2019
2018
2017
2016
2015
2014
2013
2012
2011
2010
2009
2008
2007
2006
2005
2004
2003
2002
2001
2000
1999
1998
1997
1996
1995
1994
1993
1992
1991
1.x
2.0
2.2
2.4
2.6
3.x
4.x
5.x
0.01
1.0
1.1.0
1.2.0
1.3.0
1.3.100
pre2.0.1
2.0
2.0.28
2.0.34
2.0.37
2.0.39
2.0.40
2.1.0
2.2.0pre9
2.2.0
2.2.14
2.2.19
2.2.21
2.2.24
2.2.26
2.3.0
2.3.99pre9
2.4.0
2.4.18
2.4.21
2.4.24
2.4.29
2.4.33
2.4.35
2.4.36
2.4.37
2.4.37.11
2.5.0
2.5.75
2.6.0
2.6.1
2.6.11
2.6.16
2.6.27
2.6.32
2.6.33
2.6.34
2.6.35
2.6.39
2.6.39.4
2.6.32.71
3.0-rc1
3.0
3.2 LTS
3.4 LTS
3.10 LTS
3.12 LTS
3.16 LTS
3.16.85
4.0-rc1
4.0
4.1 LTS
4.4 LTS
4.9 LTS
4.14 LTS
4.19 LTS
5.0-rc1
5.0
5.4 LTS

Из чего состоит любой Linux?

- Ядро (разных версий)
- **Дистрибутив**
- Среда рабочего стола (по необходимости)

Linux на сервере / Linux на десктопе









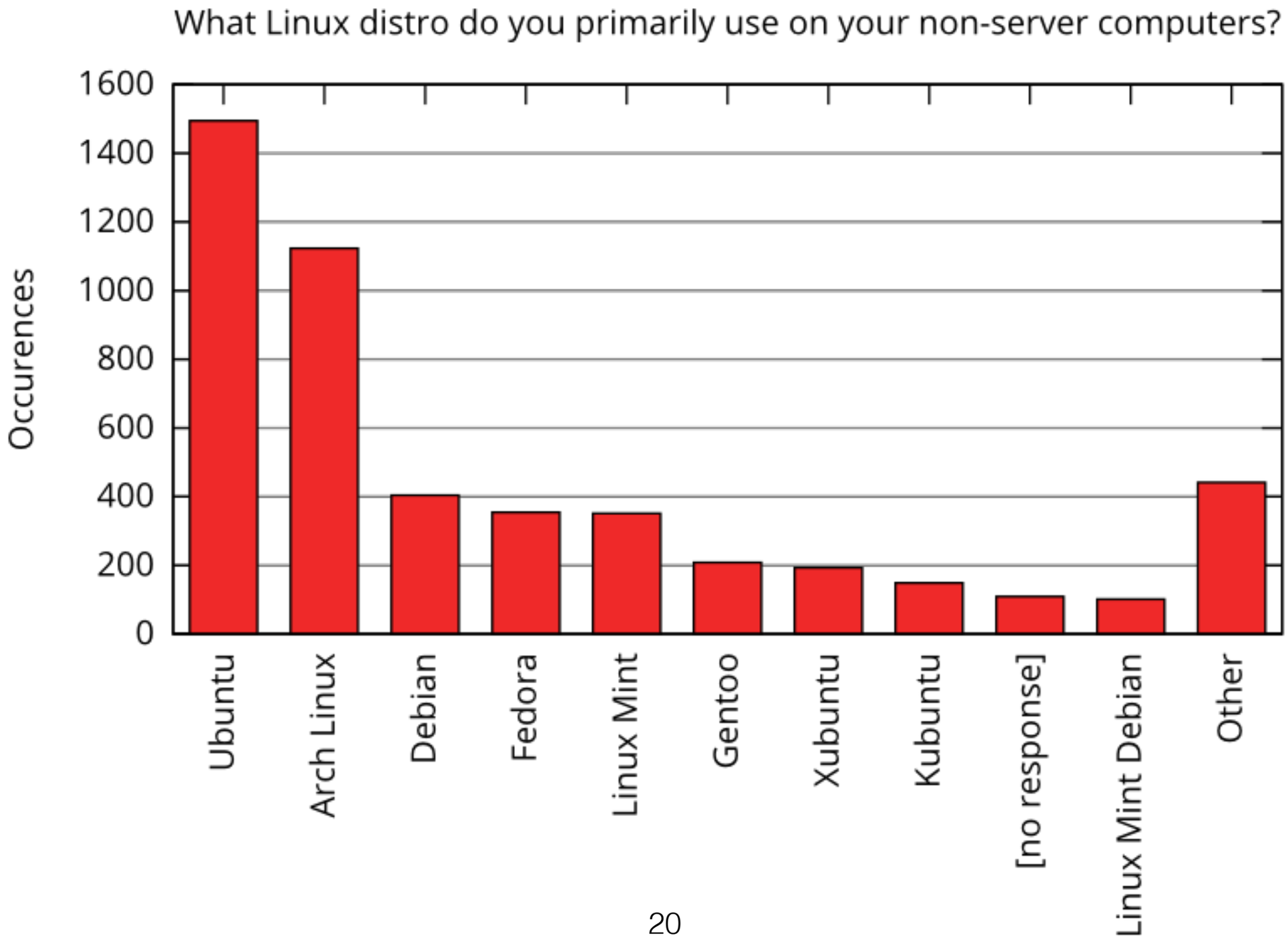






	Fedora	Red Hat	Oracle Linux	Ubuntu	SUSE
Pros	Free, provides the latest and greatest Linux features	Excellent management tools, great support options including self-help options for those on a budget	OS is free, lots of extras included with support subscriptions, affordable 24/7 support options	Basic OS is free, excellent Landscape management tools, scalability and long-term support	Custom installation options, long-term support versions available
Cons	Short support cycles and no support options beyond community support	Red Hat gets costly in a hurry, especially with any management tools added	Management tools are not Linux specific	Install could use a 'facelift', support can quickly become expensive with multiple servers	Support costs can get costly for larger installations, SUSE Manager is a bit more cumbersome than the competition
Best for	Those wanting a truly free Linux server and also access to the latest Linux technology	Those needing some comprehensive support options and the backing of a large commercial Linux provider	Those running other Oracle applications or looking for a robust Red Hat clone at a lower cost of ownership	Those needing a Linux server with long-term support and solid cloud credentials	Those needing a long-term support version of Linux and custom installation options

Дистрибутивы на десктопе



#101

WE DID IT, WE TIME TRAVELED!

BUT TO WHAT YEAR?

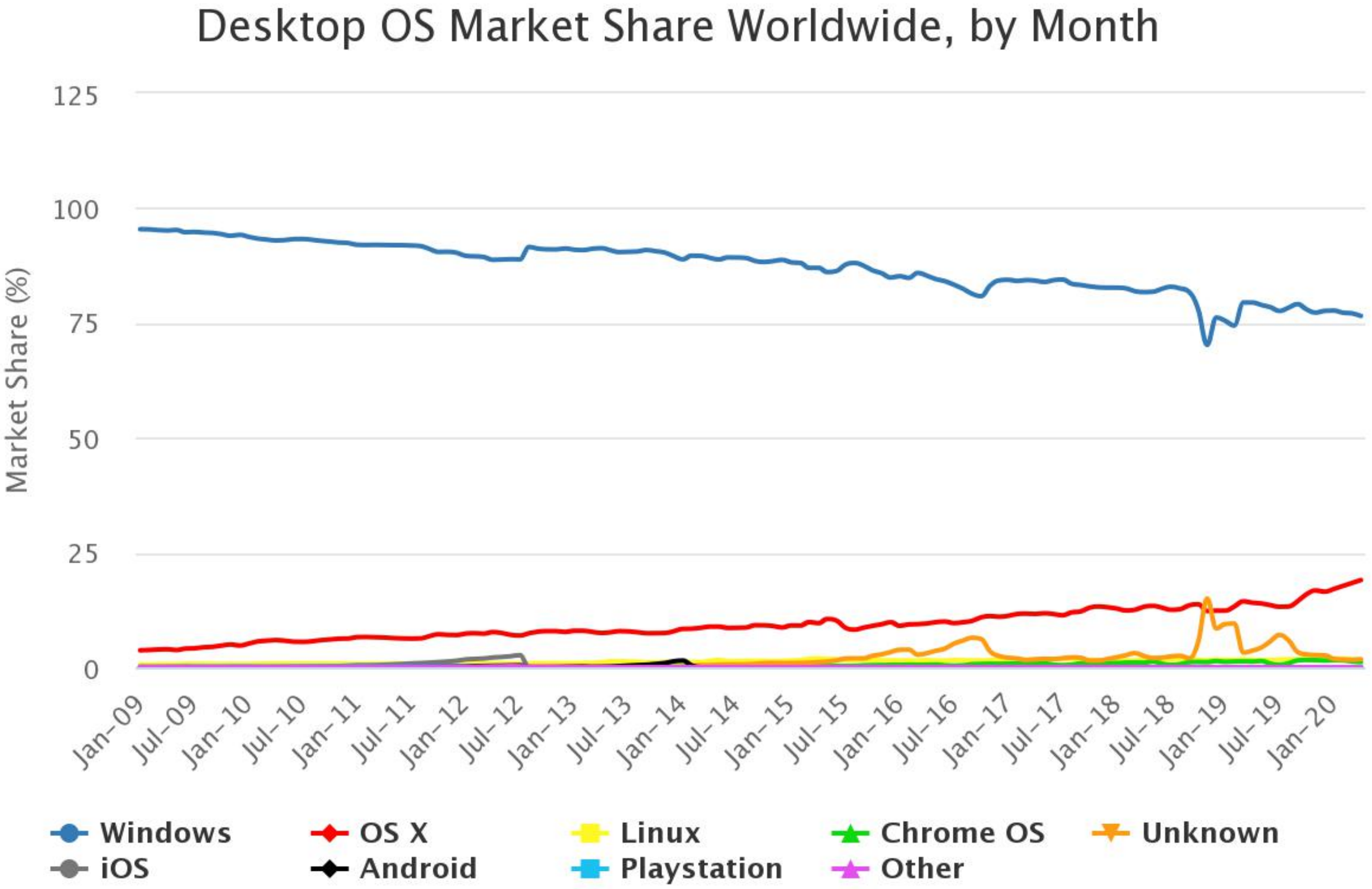
LET ME ASK THAT GUY OVER THERE

HEY! what year is it ?

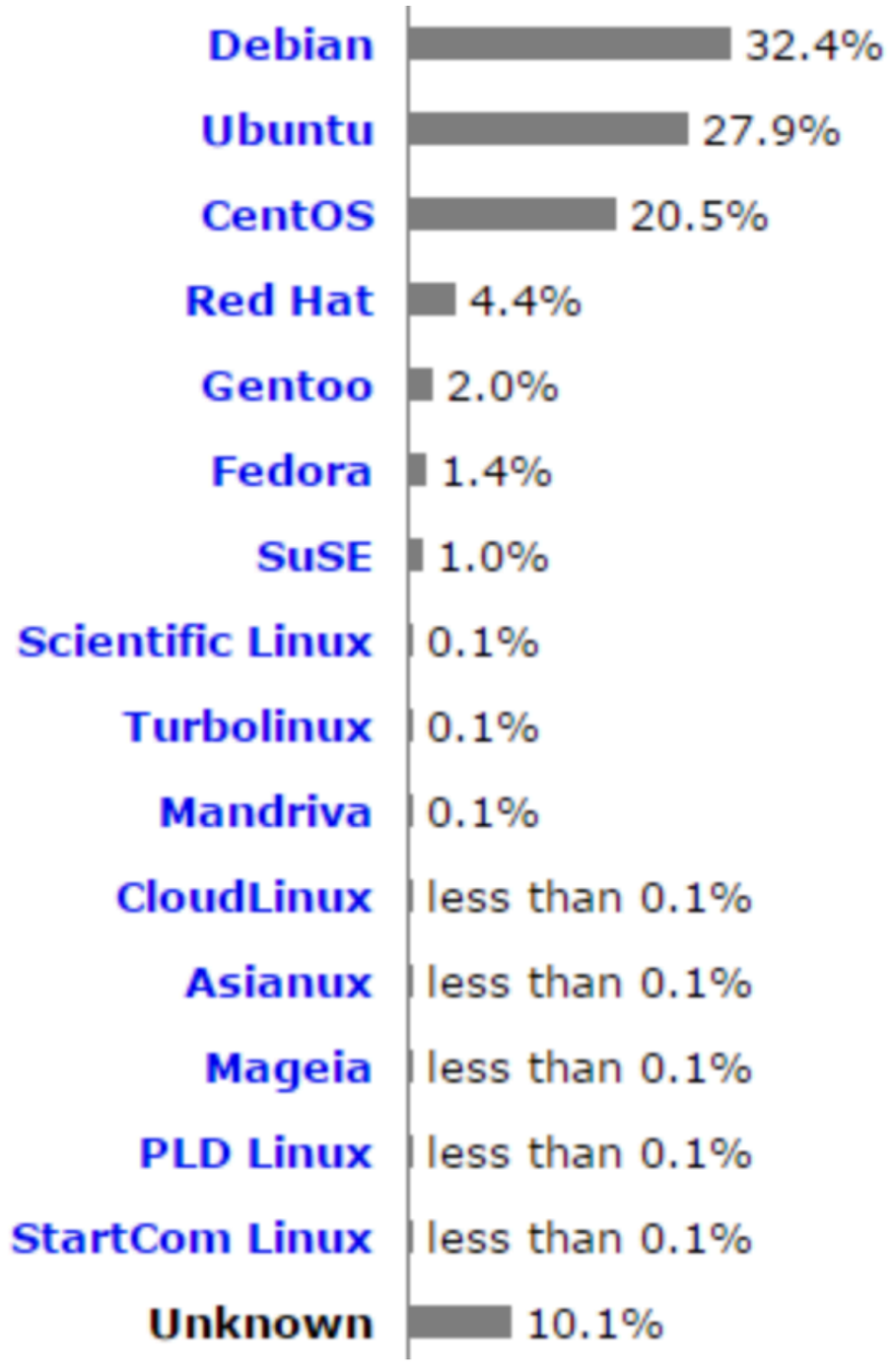
It's the year of the Linux desktop !

It could be any year from 1993

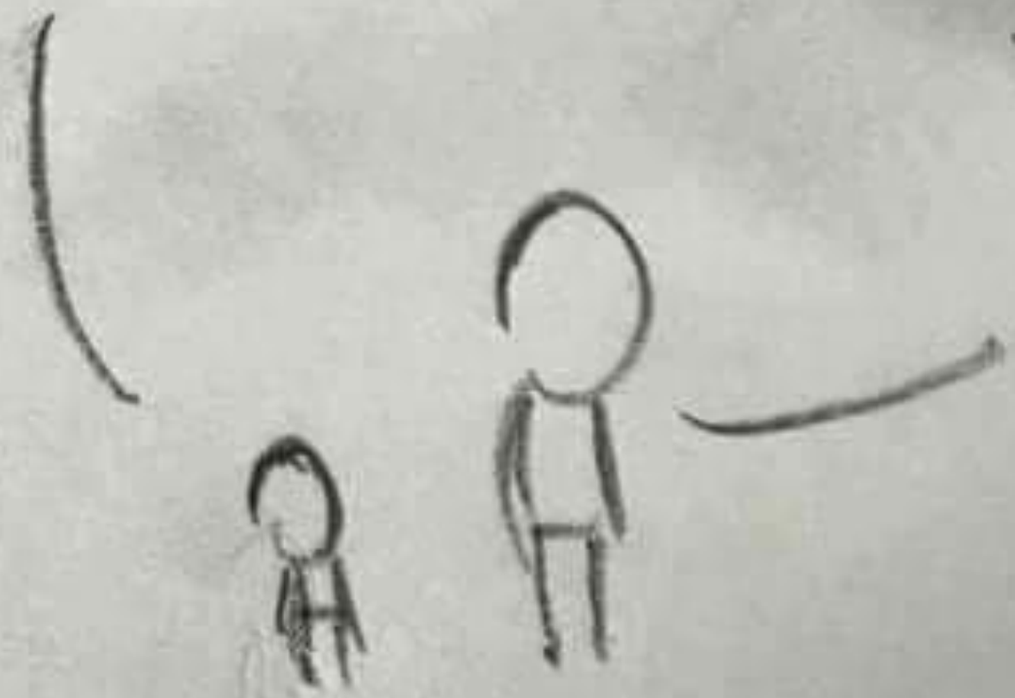
SRGRAFO


Desktop OS Market Share Worldwide, by Month
Market Share (%)
125
100
75
50
25
0
Jan-09
Jul-09
Jan-10
Jul-10
Jan-11
Jul-11
Jan-12
Jul-12
Jan-13
Jul-13
Jan-14
Jul-14
Jan-15
Jul-15
Jan-16
Jul-16
Jan-17
Jul-17
Jan-18
Jul-18
Jan-19
Jul-19
Jan-20
Windows
OS X
Linux
Chrome OS
Unknown
iOS
Android
Playstation
Other
© Dazeinfo / Data Source: StatCounter

Дистрибутивы на сервере



DADDY, WHAT ARE CLOUDS MADE OF?



LINUX SERVERS, MOSTLY

Как выбрать?

- Для десктопа: ~~Arch~~ Ubuntu
- Для сервера: вам скажет админ

Из чего состоит любой Linux?

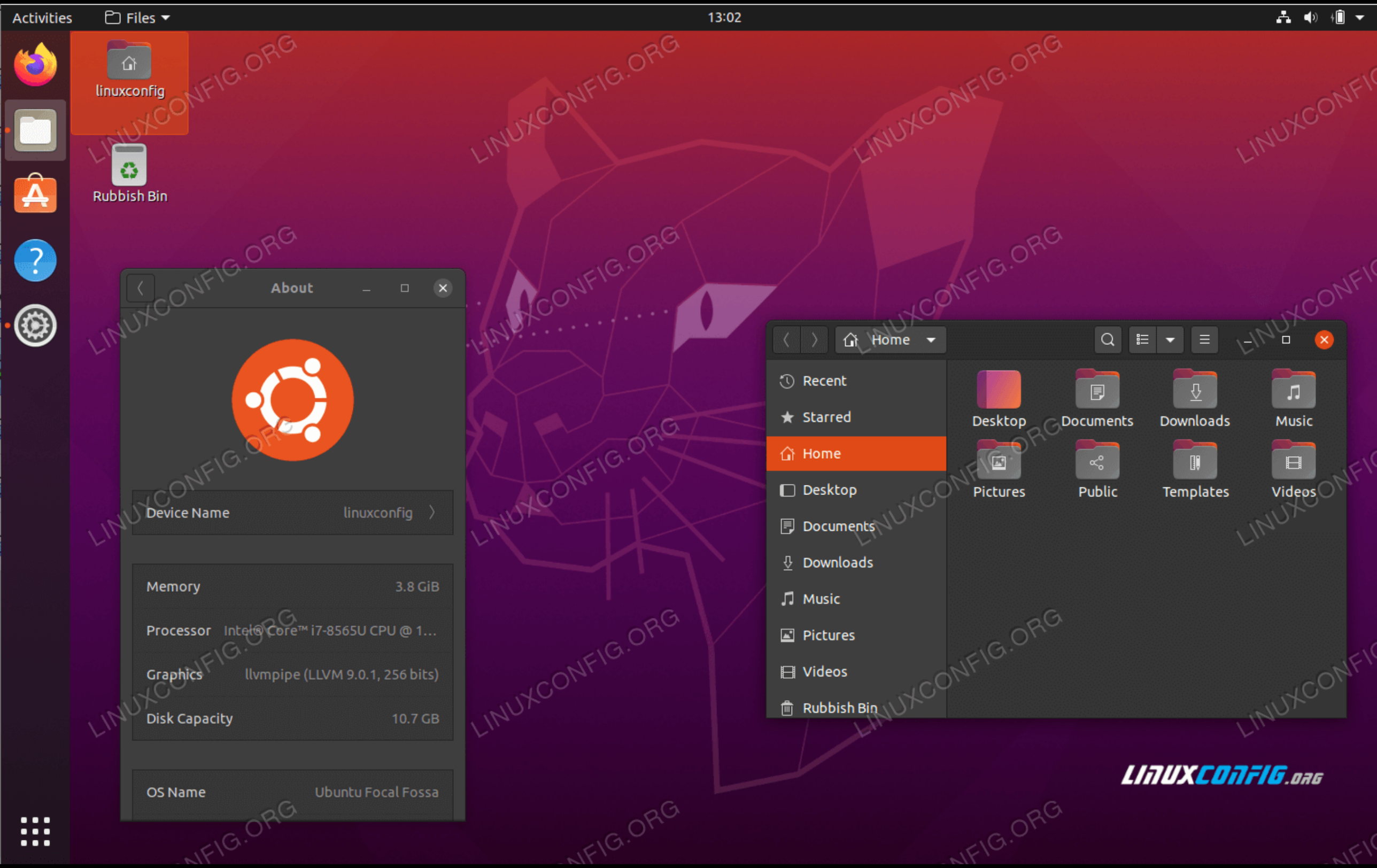
- Ядро (разных версий)
- Дистрибутив
- **Среда рабочего стола (по необходимости)**

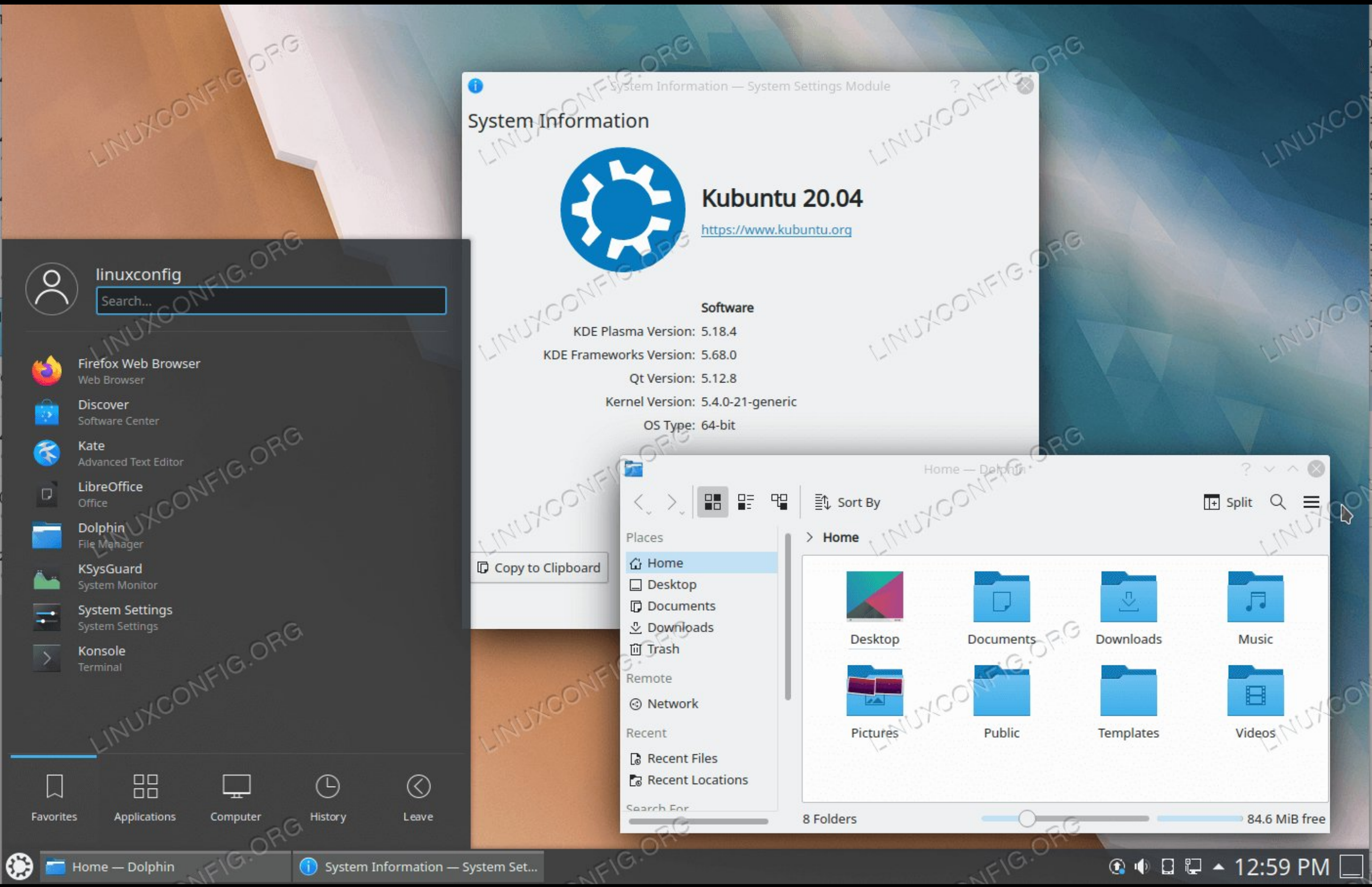
Какие бывают "среды рабочего стола"?

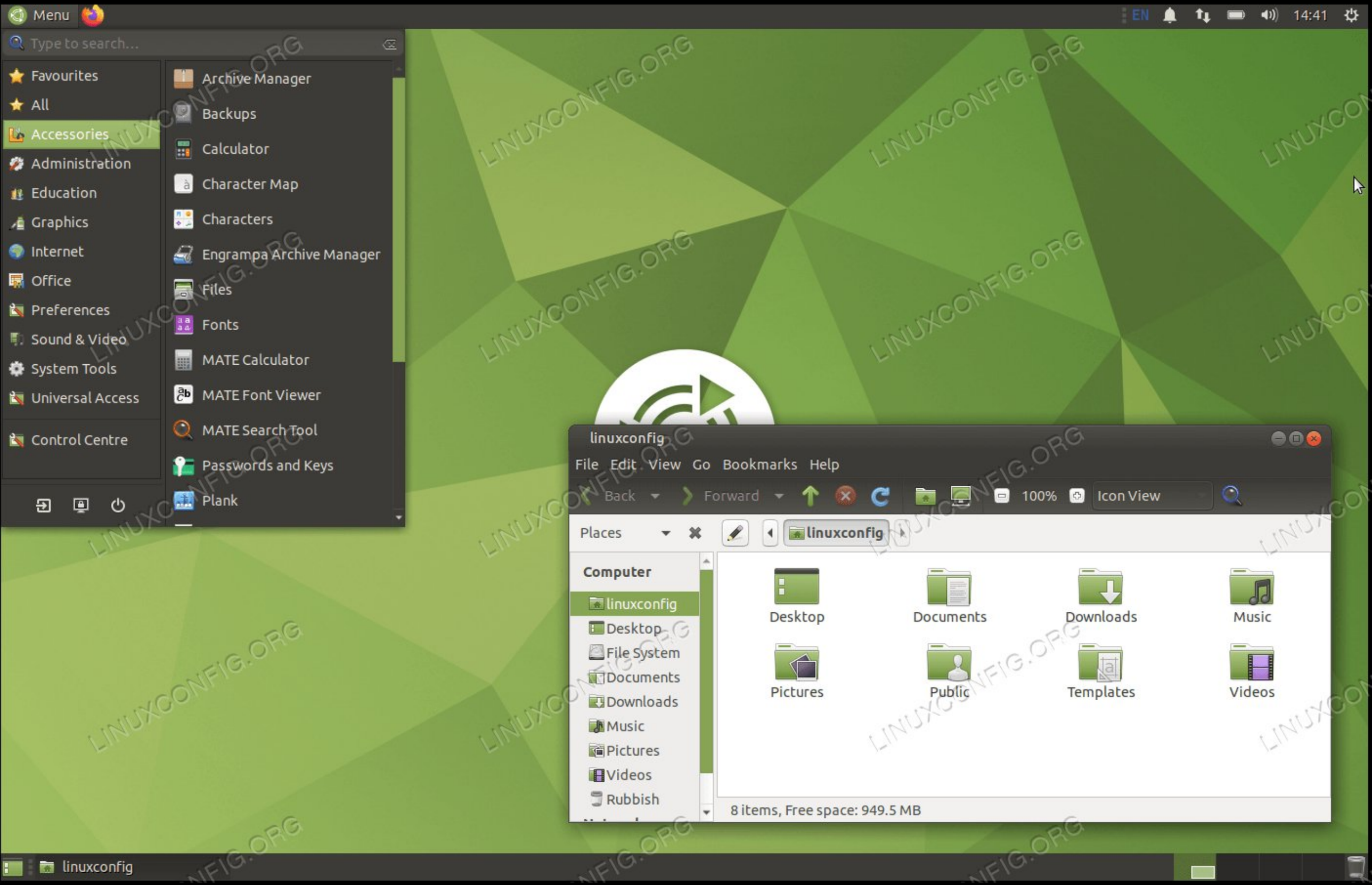
- Gnome
- KDE
- Mate
- XFCE
- Budgie
- Unity (умер, но нам его жаль)

Как выбрать?

Проще всего по красоте!


Activities
Files
13:02
linuxconfig
Rubbish Bin
About
Device Name
linuxconfig
Memory
3.8 GiB
Processor
Intel® Core™ i7-8565U CPU @ 1…
Graphics
llvmpipe (LLVM 9.0.1, 256 bits)
Disk Capacity
10.7 GB
OS Name
Ubuntu Focal Fossa
Home
Recent
Starred
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Rubbish Bin
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
LINUXCONFIG.ORG


System Information — System Settings Module
System Information
Kubuntu 20.04
https://www.kubuntu.org
Software
KDE Plasma Version: 5.18.4
KDE Frameworks Version: 5.68.0
Qt Version: 5.12.8
Kernel Version: 5.4.0-21-generic
OS Type: 64-bit
Copy to Clipboard
linuxconfig
Search...
Firefox Web Browser
Web Browser
Discover
Software Center
Kate
Advanced Text Editor
LibreOffice
Office
Dolphin
File Manager
KSysGuard
System Monitor
System Settings
System Settings
Konsole
Terminal
Favorites
Applications
Computer
History
Leave
Home — Dolphin
Sort By
Split
Places
Home
Desktop
Documents
Downloads
Trash
Remote
Network
Recent
Recent Files
Recent Locations
Home
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
8 Folders
84.6 MiB free
Home — Dolphin
System Information — System Set...
12:59 PM


Menu
EN
14:41
Type to search...
Favourites
All
Accessories
Administration
Education
Graphics
Internet
Office
Preferences
Sound & Video
System Tools
Universal Access
Control Centre
Archive Manager
Backups
Calculator
Character Map
Characters
Engrampa Archive Manager
Files
Fonts
MATE Calculator
MATE Font Viewer
MATE Search Tool
Passwords and Keys
Plank
linuxconfig
File Edit View Go Bookmarks Help
Back
Forward
100%
Icon View
Places
linuxconfig
Computer
linuxconfig
Desktop
File System
Documents
Downloads
Music
Pictures
Videos
Rubbish
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
8 items, Free space: 949.5 MB
linuxconfig

linuxconfig - File Manager About the Xfce Desktop Envi... 16 Apr, 13:46

Favorites
Recently Used
All Applications
Accessories
Education
Games
Graphics
Internet
Multimedia
Office
Settings
System

Web Browser
Mail Reader
File Manager
LibreOffice Writer
LibreOffice Calc
Pidgin Internet Messenger
Software
Terminal Emulator
Help

linuxconfig

About the Xfce Desktop Environment

About the Xfce Desktop Environment
Version 4.14, distributed by Xubuntu

About | Credits | Copyright

Xfce is a collection of programs that together provide a full-featured desktop environment. The following programs are part of the Xfce core:

Window Manager (xfwm4)
Handles the placement of windows on the screen.

Panel (xfce4-panel)
Program launchers, window buttons, applications menu, workspace switcher and more.

Desktop Manager (xfdesktop)
Sets the background colour or image with optional application menu or icons for minimised applications or launchers, devices and folders.

File Manager (thunar)
A modern file manager for the Unix/Linux desktop, aiming to be easy-to-use and fast.

Volume manager (thunar-volman)
Automatic management of removable drives and media for Thunar.

Session Manager (xfce4-session)
Restores your session on startup and allows you to shutdown the computer from Xfce.

arance,

Close

linuxconfig - File Manager

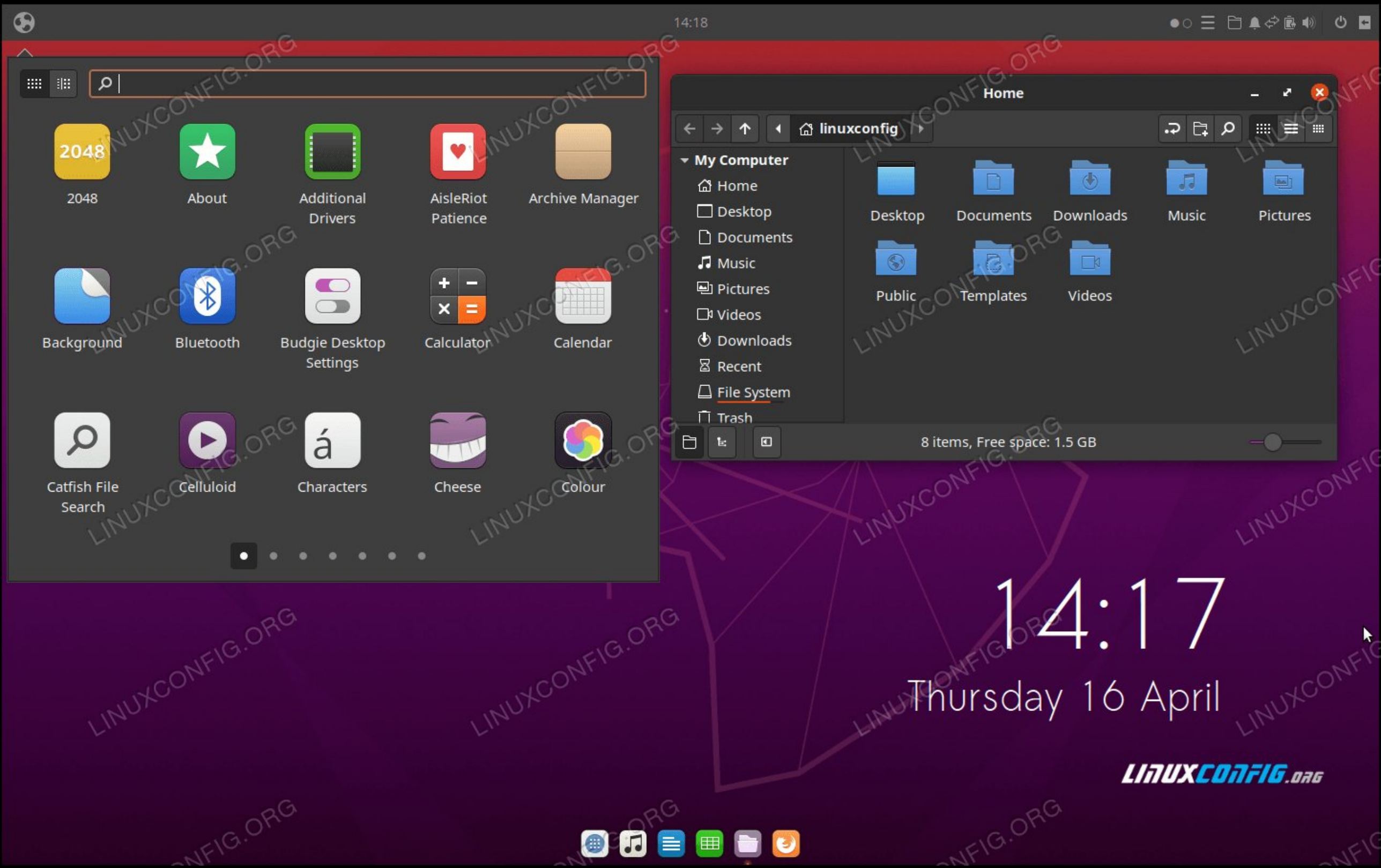
File Edit View Go Help

/home/linuxconfig/

DEVICES
File System
PLACES
linuxconfig
Desktop
Trash
Documents
Music
Pictures
Videos
Downloads
NETWORK
Browse Network

Desktop Documents Downloads Music Pictures
Public Templates Videos

8 items, Free space: 1.3 GiB


14:18
2048
About
Additional Drivers
AisleRiot Patience
Archive Manager
Background
Bluetooth
Budgie Desktop Settings
Calculator
Calendar
Catfish File Search
Celluloid
Characters
Cheese
Colour
Home
linuxconfig
My Computer
Home
Desktop
Documents
Music
Pictures
Videos
Downloads
Recent
File System
Trash
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
8 items, Free space: 1.5 GB
14:17
Thursday 16 April
LINUXCONFIG.ORG

EN 12:43 PM

Search applications

Filter results

Recently used

Unity Tweak Tool | Terminal | Help | Thunderbird Mail | Videos | Text Editor

Installed See 62 more results

Additional Drivers | Advanced Network Configuration | AisleRiot Solitaire | Archive Manager | Backups | Calculator | Calendar

Dash plugins

Applications Commands Files & Folders

Прежде всего – Linux используют внутри командной строки*

Linux = лучший в мире конструктор

Portable Operating System Interface (POSIX)

- Основные определения – список основных определений и соглашений, используемых в спецификациях
- Обоснование и принципы
- Системные интерфейсы (на языке C)
- Оболочка и утилиты – описание утилит и командной оболочки sh

Окружения командной строки

- sh – самая базовая версия
- bash – самая популярная версия
- zsh – самая удобная
- fish – самая хипстерская
- И другие

Хотите удобные конфиги?

github.com/sobolevn/dotfiles/blob/master/config/zshrc

Файловая система

Папки

pwd

```
/usr
» pwd
/usr

/usr
»
```

cd

```
/usr
» cd /tmp

/tmp
» pwd
/tmp

/tmp
»
```

cd

```
/usr
» cd /tmp

/tmp
» cd ..

/
» cd -
/tmp

/tmp
» 
```

mkdir

```
/tmp
» mkdir my_folder

/tmp
» cd my_folder

/tmp/my_folder
» pwd
/tmp/my_folder

/tmp/my_folder
»
```

Практика

Файл "folders.sh"

Файлы

ls

```
/tmp/my_folder
» ls

/tmp/my_folder
»
```

ls

```
/tmp/my_folder
» ls /usr
bin/  lib/  libexec/  local/  sbin/  share/  standalone/

/tmp/my_folder
» ls ..
CrashUpload-LiTr5/                dumps/      AlTest1.err  gameoverlayui.log
com.apple.launchd.AN5FdwTaSl/     my_folder/  AlTest1.out  gameoverlayui.log.last
com.apple.launchd.PerhthUB6L/     powerlog/   adobegc.log

/tmp/my_folder
» 
```

touch

```
/tmp/my_folder
» ls

/tmp/my_folder
» touch my-first-file.txt

/tmp/my_folder
» ls
my-first-file.txt

/tmp/my_folder
» 
```

mv

```
/tmp/my_folder
» mv my-first-file.txt new-name.txt

/tmp/my_folder
» ls
new-name.txt

/tmp/my_folder
»
```

mv

```
/tmp/my_folder
» mv new-name.txt ..

/tmp/my_folder
» ls

/tmp/my_folder
» mv ../new-name.txt .

/tmp/my_folder
» ls
new-name.txt

/tmp/my_folder
»
```

cp

```
/tmp/my_folder
» cp new-name.txt copy.txt

/tmp/my_folder
» ls
copy.txt  new-name.txt

/tmp/my_folder
»
```

rm

```
/tmp/my_folder
» ls
copy.txt  new-name.txt

/tmp/my_folder
» rm copy.txt

/tmp/my_folder
» ls
new-name.txt

/tmp/my_folder
»
```

-r

- mv -r – двигает папку
- cp -r – копирует папку
- rm -r – удаляет папку (!)

Будьте осторожны!
Не выполняйте!

- rm -r /
- rm -rf *
- rm -rf .
- rm -rf ..

Потоки ввода и вывода

Потоки

- stdin – стандартный поток для ввода информации
- stdout – стандартный поток для вывода, используется для вывода общей информации
- stderr – стандартный поток для ошибок, используется для вывода информации и данных ошибок

echo (stdout)

```
~
» echo 'Hello, linux!'
Hello, linux!

~
»
```

echo >&2 (stderr)

```
~
» echo 'Hello, error :(' >&2
Hello, error :(

~
»
```

echo >

```
/tmp/my_folder
» echo 'Nikita Sobolev' > my-name.txt

/tmp/my_folder
»
```

cat

```
/tmp/my_folder
» cat my-name.txt
Nikita Sobolev

/tmp/my_folder
»
```

echo >>

```
/tmp/my_folder
» echo 'Andreevich' >> my-name.txt

/tmp/my_folder
»
```

cat

```
/tmp/my_folder
» cat my-name.txt
Nikita Sobolev
Andreevich

/tmp/my_folder
»
```

Какие команды нам доступны?

echo $PATH

```
~
» echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin

~
»
```

Волшебные директории: Filesystem Hierarchy Standard

- /bin - основные утилиты необходимые в работе
- /usr - пользовательские утилиты
- /tmp - временные файлы
- /etc - файлы конфигурации системы
- /home - файлы пользователя

ls /bin

```
~
» ls /bin
'['*       cp*      df*      hostname*    link*    mv*      rm*       stty*    unlink*
 bash*     csh*     echo*    kill*        ln*      pax*     rmdir*    sync*    wait4path*
 cat*      date*    ed*      ksh*         ls*      ps*      sh*       tcsh*    zsh*
 chmod*    dd*      expr*    launchctl*   mkdir*   pwd*     sleep*    test*

~
»
```

Практика

Файл "files.sh"

Пользователи и группы

Система пользователей и групп

- Пользователь
- Группа пользователей

Зачем она нужна?

- Каждому пользователю и группе можно выделять свои разрешения: что ему можно, что ему нельзя

- Каждому пользователю можно выдавать "свои" места в файловой системе, где будут жить его данные

- Данная функция является основой системы безопасности и краеугольным камнем всей системы

Создание пользователей и групп

- groupadd newgroup – создаст новую группу newgroup

- useradd -G newgroup -p password newuser – создаст нового пользователя newuser с паролем password в группе newgroup

groups

```
/tmp/my_folder
» groups
staff everyone localaccounts _appserverusr admin _appserveradm _lpadmin com.apple.shar
epoint.group.1 _appstore _lpoperator _developer _analyticsusers com.apple.access_ftp c
om.apple.access_screensharing com.apple.access_ssh

/tmp/my_folder
»
```

users

```
/tmp/my_folder
» users
sobolev

/tmp/my_folder
»
```

id

```
/tmp/my_folder
» id sobolev
uid=501(sobolev) gid=20(staff) groups=20(staff),12(everyone),61(localaccounts),79(_app
serverusr),80(admin),81(_appserveradm),98(_lpadmin),701(com.apple.sharepoint.group.1),
33(_appstore),100(_lpoperator),204(_developer),250(_analyticsusers),395(com.apple.acce
ss_ftp),398(com.apple.access_screensharing),399(com.apple.access_ssh)

/tmp/my_folder
»
```

Разрешения

(или "запрещения")

ls -l

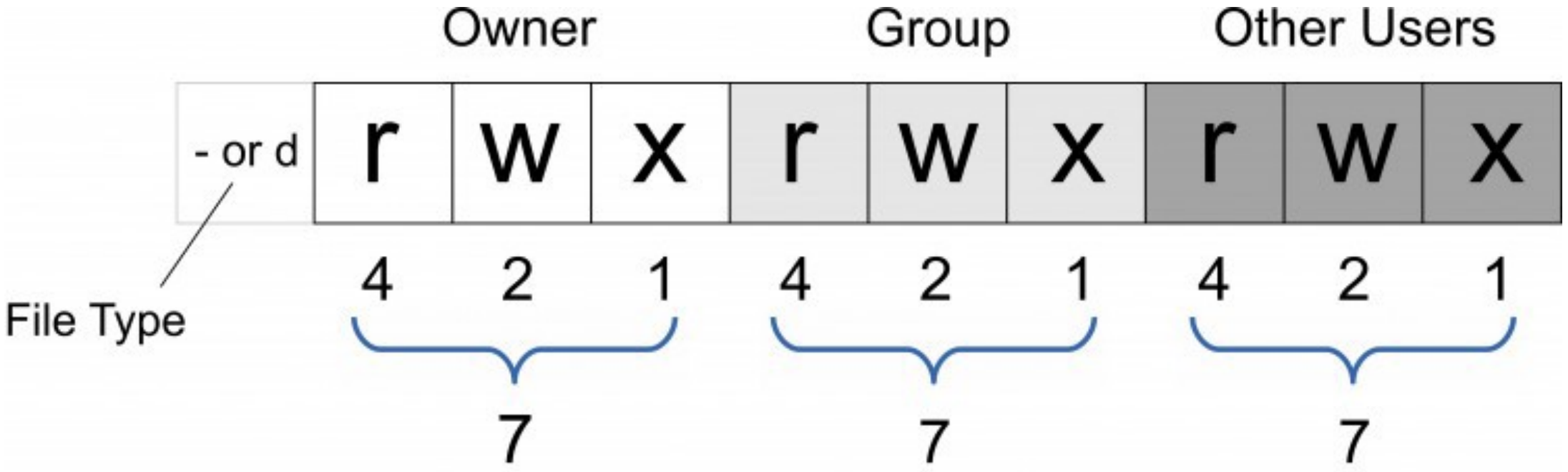
```
/tmp/my_folder
» ls -l
total 4.0K
-rw-r--r-- 1 sobolev wheel 26 Sep 29 19:27 my-name.txt
-rw-r--r-- 1 sobolev wheel  0 Sep 29 18:52 new-name.txt

/tmp/my_folder
»
```

ls -l

```
/tmp/my_folder
» ls -l
total 4.0K
-rw-r--r-- 1 sobolev wheel 26 Sep 29 19:27 my-name.txt
-rw-r--r-- 1 sobolev wheel  0 Sep 29 18:52 new-name.txt

/tmp/my_folder
»
```

ls -l

```
/tmp/my_folder
» ls -l
total 4.0K
-rw-r--r-- 1 sobolev wheel 26 Sep 29 19:27 my-name.txt
-rw-r--r-- 1 sobolev wheel  0 Sep 29 18:52 new-name.txt

/tmp/my_folder
»
```


Owner
Group
Other Users
- or d
r
w
x
r
w
x
r
w
x
4
2
1
4
2
1
4
2
1
File Type
7
7
7

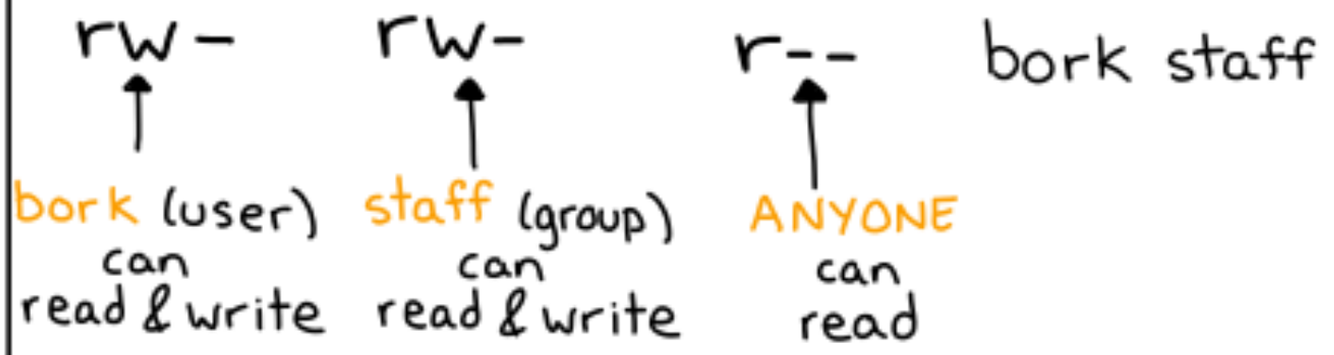
Julia Evans
@b0rk

unix permissions

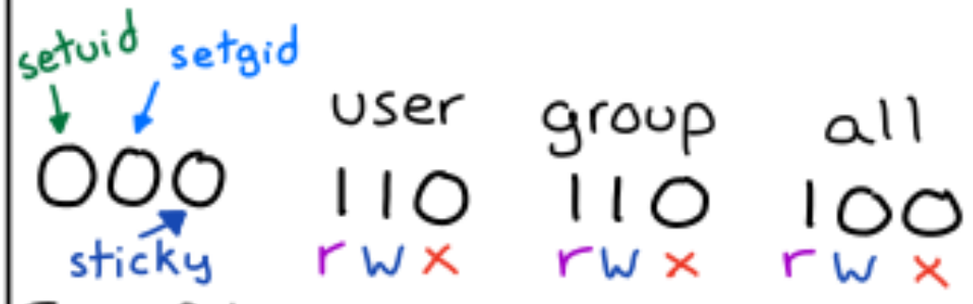
There are 3 things you can do to a file

read write execute

`ls -l file.txt` shows you permissions. Here's how to interpret the output:



File permissions are 12 bits



For files:
- r = can read
- w = can write
- x = can execute

For directories, it's approximately:
- r = can list files
- w = can create files
- x = can cd into & access files

110 in binary is 6

So rw- r-- r--
= 110 100 100
= 6 4 4

`chmod 644 file.txt`
means change the permissions to:

rw- r-- r--

Simple!

`setuid` affects executables

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

rws r-x r-x root root

this means ping always runs as root

`setgid` does 3 different unrelated things for executables, directories, and regular files.

- rw- r-- r-- = 644

- - – обозначает файл, d – обозначал бы папку
- rw- – обозначает, что владелец (sobolev) может читать и писать = 6
- r-- – обозначает, что группа (wheel) может читать = 4
- r-- – обозначает, что другие пользователи могут читать = 4

chmod

```
/tmp/my_folder
» chmod 400 my-name.txt

/tmp/my_folder
» echo 'New name' > my-name.txt
zsh: permission denied: my-name.txt

/tmp/my_folder                                                    1 ⚠
» ls -l
total 4.0K
-r-------- 1 sobolev wheel 26 Sep 29 19:27 my-name.txt
-rw-r--r-- 1 sobolev wheel  0 Sep 29 18:52 new-name.txt

/tmp/my_folder
»
```

chmod

```
/tmp/my_folder
» chmod 644 my-name.txt

/tmp/my_folder
» echo 'New name' > my-name.txt

/tmp/my_folder
» cat my-name.txt
New name

/tmp/my_folder
» ls -l
total 4.0K
-rw-r--r-- 1 sobolev wheel 9 Sep 29 20:37 my-name.txt
-rw-r--r-- 1 sobolev wheel 0 Sep 29 18:52 new-name.txt

/tmp/my_folder
»
```

ls -l

```
/tmp/my_folder
» ls -l
total 4.0K
-rw-r--r-- 1 sobolev wheel 26 Sep 29 19:27 my-name.txt
-rw-r--r-- 1 sobolev wheel  0 Sep 29 18:52 new-name.txt

/tmp/my_folder
»
```

chown

```
/tmp/my_folder
» chown sobolev:everyone my-name.txt

/tmp/my_folder
» ls -l
total 4.0K
-rw-r--r-- 1 sobolev everyone 9 Sep 29 20:37 my-name.txt
-rw-r--r-- 1 sobolev wheel    0 Sep 29 18:52 new-name.txt

/tmp/my_folder
»
```

Практика

Файл "permissions.sh"

Самая важная команда!

man что-угодно

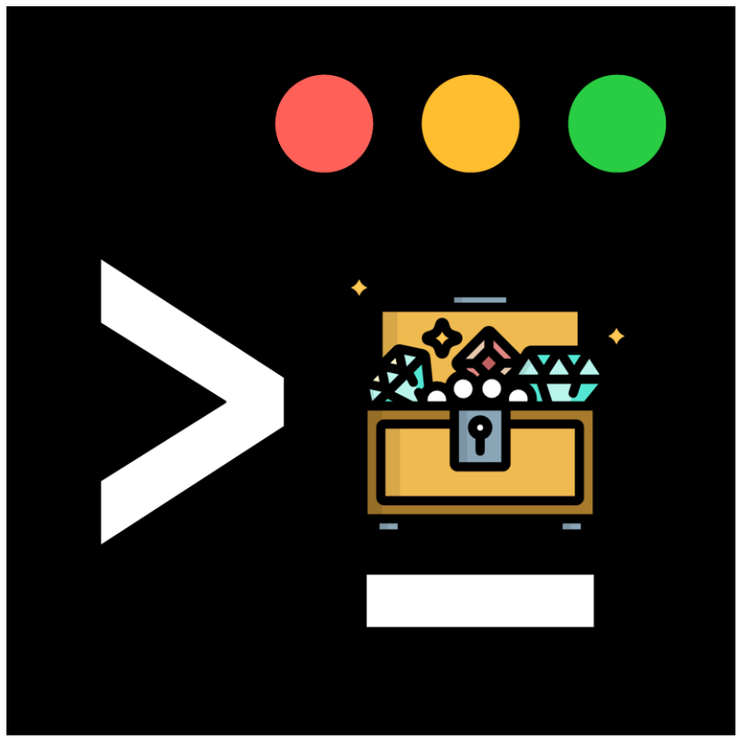
```
CHOWN(8)                  BSD System Manager's Manual                 CHOWN(8)

NAME
     chown -- change file owner and group

SYNOPSIS
     chown [-fhv] [-R [-H | -L | -P]] owner[:group] file ...
     chown [-fhv] [-R [-H | -L | -P]] :group file ...

DESCRIPTION
     The chown utility changes the user ID and/or the group ID of the speci-
     fied files.  Symbolic links named by arguments are silently left
     unchanged unless -h is used.

     The options are as follows:

     -f      Don't report any failure to change file owner or group, nor mod-
             ify the exit status to reflect such failures.

     -H      If the -R option is specified, symbolic links on the command
             line are followed.  (Symbolic links encountered in the tree tra-
             versal are not followed.)

     -h      If the file is a symbolic link, change the user ID and/or the
lines 1-25
```

Краткий обзор пройденного

- История и общая информация о Linux
- Работа с командой строкой
- Файлы и папки
- stdin, stdout, stderr
- Пользователи и группы

Выводы

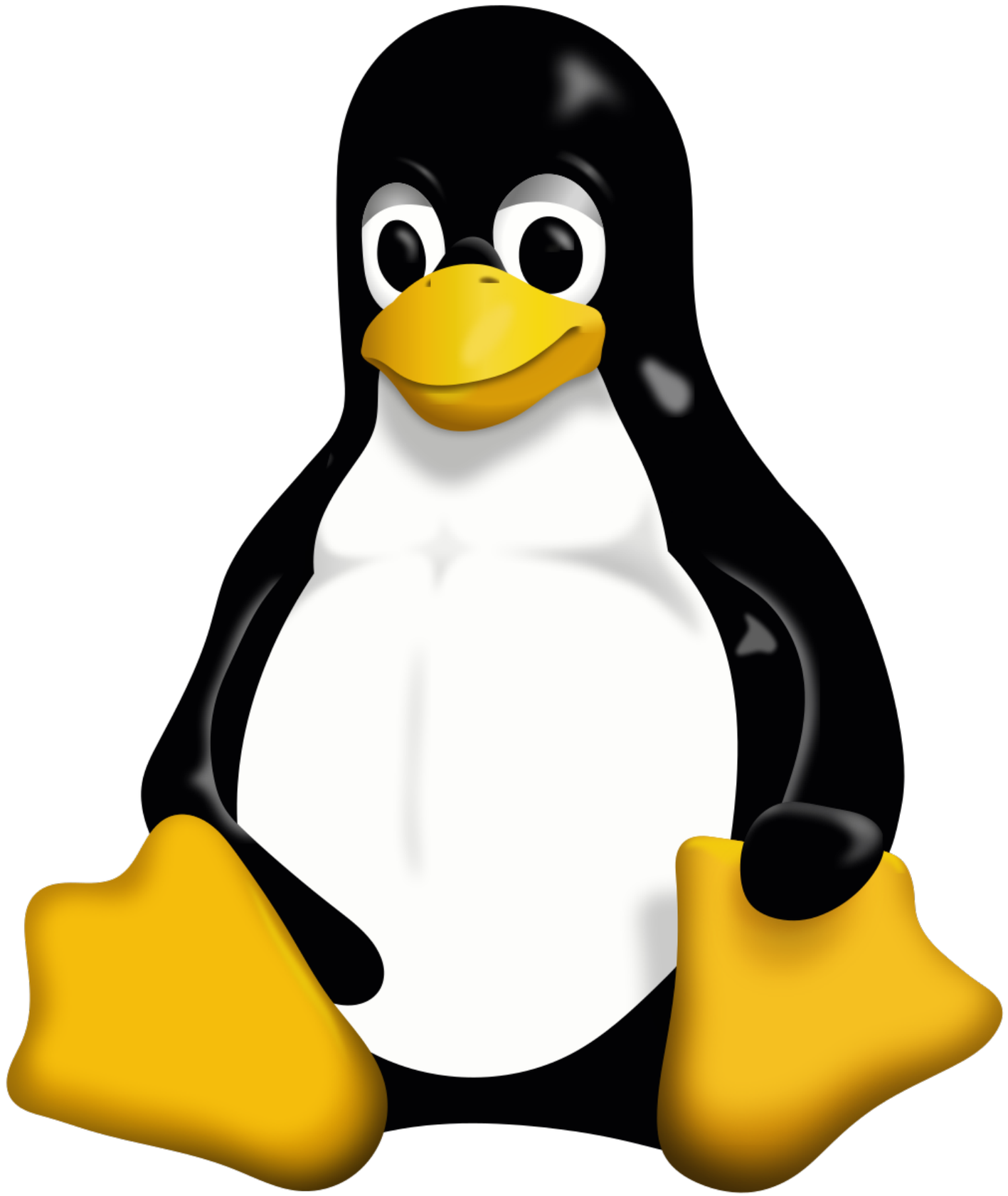
- Linux = основная операционная система для работы серверов
- Linux = конструктор, он может быть любым
- Пользоваться Linux обычно приходится в консоли (сначала неудобно, но пути обратно нет)
- Перед пользователями Linux открывается целый мир крутых штук для разработки и администрирования



github.com/sobolevn/dotfiles

Никита Соболев

github.com/sobolevn



Краткий обзор первого занятия

- История и общая информация о Linux
- Работа с командой строкой
- Файлы и папки
- stdin, stdout, stderr
- Пользователи и группы

sh

План занятия

- Переменные
- Условия и циклы
- Функции
- Sub-shells и pipes
- Полезные команды

Переменные

ls

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» ls
README.md code       course.md ex.py     helpers   homework  lectures

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

COLUMNS=1 ls

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» ls
README.md code       course.md ex.py     helpers   homework  lectures

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» COLUMNS=1 ls
README.md
code
course.md
ex.py
helpers
homework
lectures

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Формы обращения

- $VAR – может быть опасным
- "$VAR" – правильный вариант, если ничего нет после VAR
- "${VAR}" – правильный вариант, если VAR внутри другой строки
- '$VAR' – просто строка

Практика

Файл "vars.sh"

Условия

if

```sh
#!/usr/bin/env sh

MY_VAR=1

if [ "$MY_VAR" -eq 1 ]; then
  echo 'MY_VAR is 1!'
else
  echo 'MY_VAR is not 1 :('
fi
```

if

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» MY_VAR=1

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» if [ "$MY_VAR" -eq 1 ]; then
  echo 'MY_VAR is 1!'
else
  echo 'MY_VAR is not 1 :('
fi
MY_VAR is 1!

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Условия

Сравнения

- **s1 = s2** – идентичны ли два значения?
- **s1 != s2** – не идентичны?
- **s1 -eq s2** – алгебраическое равенство
- **s1 -ne s2** – алгебраическое неравество

Булевые операции

- **e1 -a e2** – булево "И"
- **e1 -o e2** – булево "ИЛИ"
- **!** – отрицание

Специальные условия

- **[-f** – существует ли файл
- **[-d** – существует ли директория
- **[-n** – не ли пустая строка
- **[-z** – пустая ли строка
- ...

[– просто программа

```
TEST(1)                   BSD General Commands Manual                   TEST(1)

NAME
     test, [ -- condition evaluation utility

SYNOPSIS
     test expression
     [ expression ]

DESCRIPTION
     The test utility evaluates the expression and, if it evaluates to true,
     returns a zero (true) exit status; otherwise it returns 1 (false).  If
     there is no expression, test also returns 1 (false).

     All operators and flags are separate arguments to the test utility.

     The following primaries are used to construct expression:

     -b file        True if file exists and is a block special file.

     -c file        True if file exists and is a character special file.

     -d file        True if file exists and is a directory.

lines 1-25
```

[

```
MY_VAR=1

[ "$MY_VAR" -eq 1 ] && echo 'MY_VAR is 1' || echo 'MY_VAR is not 1'
```

[

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» MY_VAR=1

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» [ "$MY_VAR" -eq 1 ] && echo 'MY_VAR is 1!' || echo 'MY_VAR is not 1 :('

MY_VAR is 1!

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

test

```
1 #!/usr/bin/env sh
2
3 MY_VAR=1
4
5 test "$MY_VAR" -eq 1 \
6   && echo 'MY_VAR is 1!' \
7   || echo 'MY_VAR is not 1 :('
```

test

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» MY_VAR=1

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» test "$MY_VAR" -eq 1 \
  && echo 'MY_VAR is 1!' \
  || echo 'MY_VAR is not 1 :('

MY_VAR is 1!

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

test

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» MY_VAR=2; test "$MY_VAR" -eq 1 \
  && echo 'MY_VAR is 1!' \
  || echo 'MY_VAR is not 1 :('

MY_VAR is not 1 :(

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Практика

Файл "ifs.sh"

shebang

```
1 #!/usr/bin/env sh
2
```

Практика

Файл "shebang.sh"

Циклы

while

```
#!/usr/bin/env sh

SUM_RESULT=0

while [ "$SUM_RESULT" -le 5 ]; do
  echo "Welcome $SUM_RESULT times"
  SUM_RESULT=$(( SUM_RESULT + 1 ))
done
```

while

```
~/Documents ... x5_school_common/code/code4  master ✗
» SUM_RESULT=0

while [ "$SUM_RESULT" -le 5 ]; do
  echo "Welcome $SUM_RESULT times"
  SUM_RESULT=$(( SUM_RESULT + 1 ))
done
Welcome 0 times
Welcome 1 times
Welcome 2 times
Welcome 3 times
Welcome 4 times
Welcome 5 times

~/Documents ... x5_school_common/code/code4  master ✗
»
```

for

```
#!/usr/bin/env sh

MY_ARRAY=(1 2 3)

for item in ${MY_ARRAY[*]}; do
  echo "item: $item"
done
```

for

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» MY_ARRAY=(1 2 3)

for item in ${MY_ARRAY[*]}; do
  echo "item: $item"
done
item: 1
item: 2
item: 3

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Практика

Файл "loops.sh"

Функции

```
1 my_ls() {  # declaring
2   echo 'Listing files'
3   ls
4   echo 'Done!'
5 }
6
7 my_ls  # calling
```

calling my_ls

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» my_ls() {
  echo 'Listing files'
  ls
  echo 'Done!'
}

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» my_ls
Listing files
README.md code      course.md ex.py     ex.sh     helpers   homework  lectures
Done!

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Практика

Файл "functions/declaration.sh"

Локальные переменные

```sh
#!/usr/bin/env sh

my_function() {
  local variable=1
  OTHER_VARIABLE=2
}

my_function

echo "OTHER_VARIABLE is $OTHER_VARIABLE"
echo "variable is: $variable"
```

Локальные переменные

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» sh code/code4/functions/locals.sh
OTHER_VARIABLE is 2
variable is:

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Практика

Файл "functions/locals.sh"

Аргументы функции

```sh
#!/usr/bin/env sh

sum_two_numbers() {
  local first
  local second

  first="$1"
  second="$2"

  echo "$((first + second))"
}

sum_two_numbers 2 3
```

Аргументы

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» sum_two_numbers() {
  local first
  local second

  first="$1"
  second="$2"

  echo "$((first + second))"
}

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» sum_two_numbers 2 3
5

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Значения по-умолчанию

```
1 #!/usr/bin/env sh
2
3 greet() {
4   local user
5   user="${1:-root}"
6
7   echo "Hello, $user"
8 }
9
10 greet sobolevn
11 greet
```

Значения по-умолчанию

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» greet() {
  local user
  user="${1:-root}"

  echo "Hello, $user"
}

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» greet sobolevn
Hello, sobolevn

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» greet
Hello, root

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

"Все" аргументы

```sh
#!/usr/bin/env sh

echo_all_words() {
  echo "Words: $*"
}

echo_all_words 'cat' 'dog' 'piggy'
```

Все аргументы

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» echo_all_words() {
  echo "Words: $*"
}

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» echo_all_words 'cat' 'dog' 'piggy'

Words: cat dog piggy

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Все аргументы

- "$*" – все аргументы как строка
- "$@" – все аргументы как массив

Syntax	Effective result
$*	$1 $2 $3 … ${N}
$@	$1 $2 $3 … ${N}
"$*"	"$1c$2c$3c…c${N}"
"$@"	"$1" "$2" "$3" … "${N}"

Практика

Файл "functions/arguments.sh"

Возвращаемое значение функций

Типы возвращаемых значений

- Статус код операции: 0 / 1 / другое
- Значение

Статус код: "$?"

Статус код

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» cat ex.sh
echo 'I do exist'

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» echo "$?"
0

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Статус код

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» cat missing.txt
cat: missing.txt: No such file or directory

~/Documents/Work/consulting/X5/x5_school_common  master ✗                1 ⚠
» echo "$?"
1

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

return

```sh
#!/usr/bin/env sh

can_fail() {
  if [ "$1" -eq 5 ]; then
    echo 'error'
    return 1
  fi

  echo 'correct'
  return 0
}

can_fail 1
echo "status code is: $?"

can_fail 5
echo "status code is: $?"
```

return

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
can_fail() {
  if [ "$1" -eq 5 ]; then
    echo 'error'
    return 1
  fi

  echo 'correct'
  return 0
}

can_fail 1
echo "status code is: $?"

can_fail 5
echo "status code is: $?"
correct
status code is: 0
error
status code is: 1

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Значение через $()

subshell

```
1 #!/usr/bin/env sh
2
3 can_fail() {
4   if [ "$1" -eq 5 ]; then
5     echo 'error'
6   else
7     echo 'correct'
8   fi
9 }
10
11 first=$(can_fail 1)
12 echo "returns: $first code: $?"
13
14 second=$(can_fail 5)
15 echo "returns: $second code: $?"
```

$()

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» can_fail() {
  if [ "$1" -eq 5 ]; then
    echo 'error'
  else
    echo 'correct'
  fi
}

first=$(can_fail 1)
echo "returns: $first code: $?"

second=$(can_fail 5)
echo "returns: $second code: $?"

returns: correct code: 0
returns: error code: 0

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Практика

Файл "functions/returns.sh"

Теперь мы знакомы со всеми основными штуками

Julia Evans
@b0rk

brackets cheat sheet

shell scripts have a lot of brackets

`(cd ~/music; pwd)`

a group of commands.
runs in a subshell.

`{cd ~/music; pwd}`

a group of commands.
runs in the same process.

`VAR=$(cat file.txt)`

run a command & assign output (similar to ``)

`x=(1 2 3)`

creates an array

`x=$((2+2))`

`$(())` does arithmetic

`if [ ... ]`

`/usr/bin/[` is a program that evaluates statements

`${var}`

another way to reference the variable `$var`

`a{.png,.svg}`

this expands to `a.png a.svg`
it's called "brace expansion"

`if [[ ... ]]`

`[[` is bash syntax. it's more powerful than `[`

`${var//search/replace}`

`${}` supports search/replace & lots more fancy things

Файлы скриптов

Простой запуск

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» cat ex.sh
```

```
       File: ex.sh
──────┼──────────────────────────────────────────────────────────────────────
   1  │ #!/usr/bin/env sh
   2  │
   3  │ to_be_called() {
   4  │   echo 'done!'
   5  │ }
```

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» sh ex.sh

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» to_be_called
zsh: command not found: to_be_called

~/Documents/Work/consulting/X5/x5_school_common  master ✗                127 ⚠
»
```

source

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» cat ex.sh
```

```
       File: ex.sh
──────┼──────────────────────────────────────────────────────────────────────
   1  │ #!/usr/bin/env sh
   2  │
   3  │ to_be_called() {
   4  │   echo 'done!'
   5  │ }
──────┴──────────────────────────────────────────────────────────────────────
```

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» source ex.sh

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» to_be_called
done!

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

export переменных

Позволяет использовать переменные при source

```
1 #!/usr/bin/env sh
2
3 MY_VARIABLE='value'
4 export $MY_VARIABLE
```

export

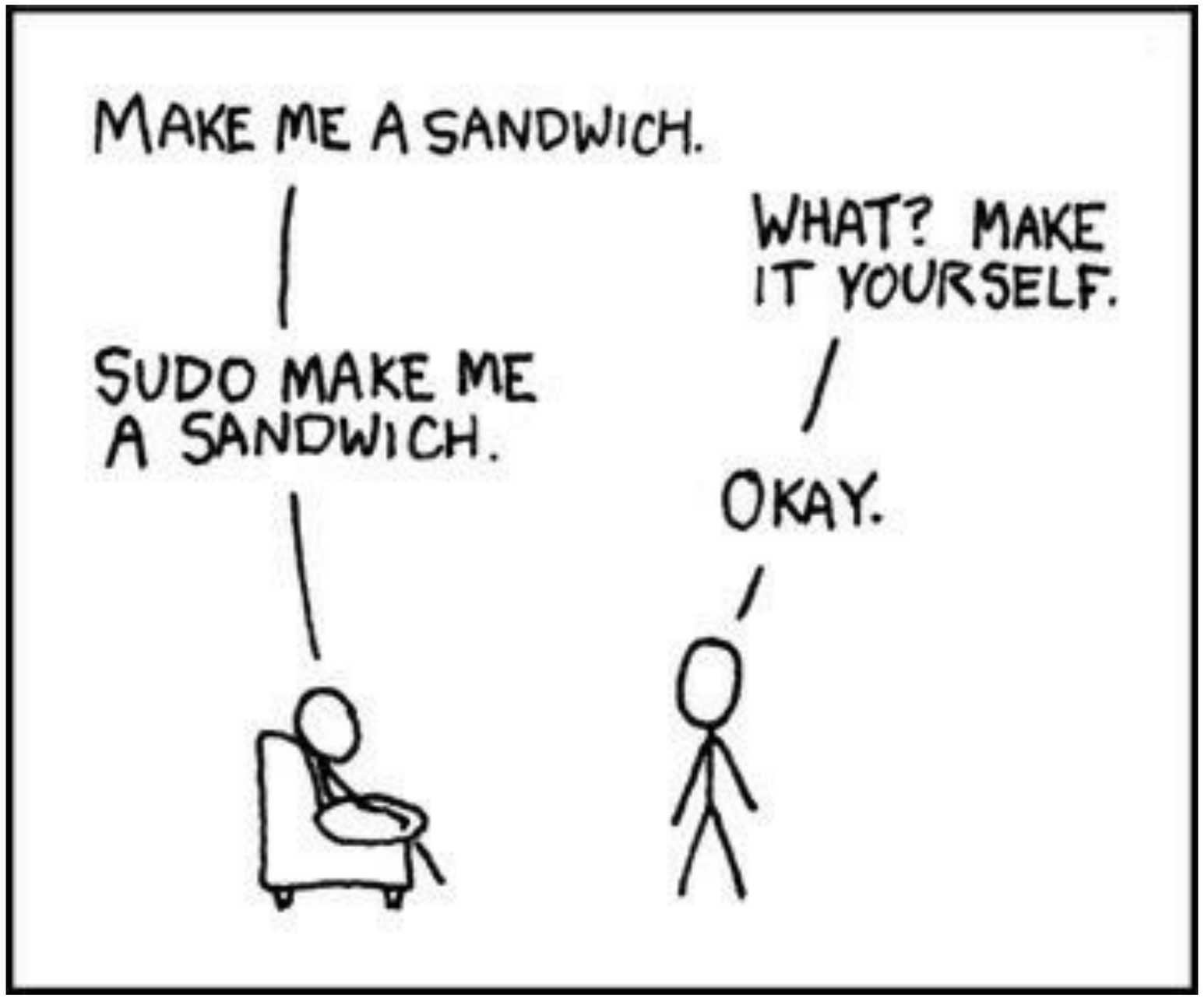
```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» bat ex.sh
```

```
       File: ex.sh
   1   #!/usr/bin/env sh
   2
   3   MY_VARIABLE='value'
   4   export $MY_VARIABLE
```

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» source ex.sh

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» echo "$MY_VARIABLE"
value

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Полезности

sudo

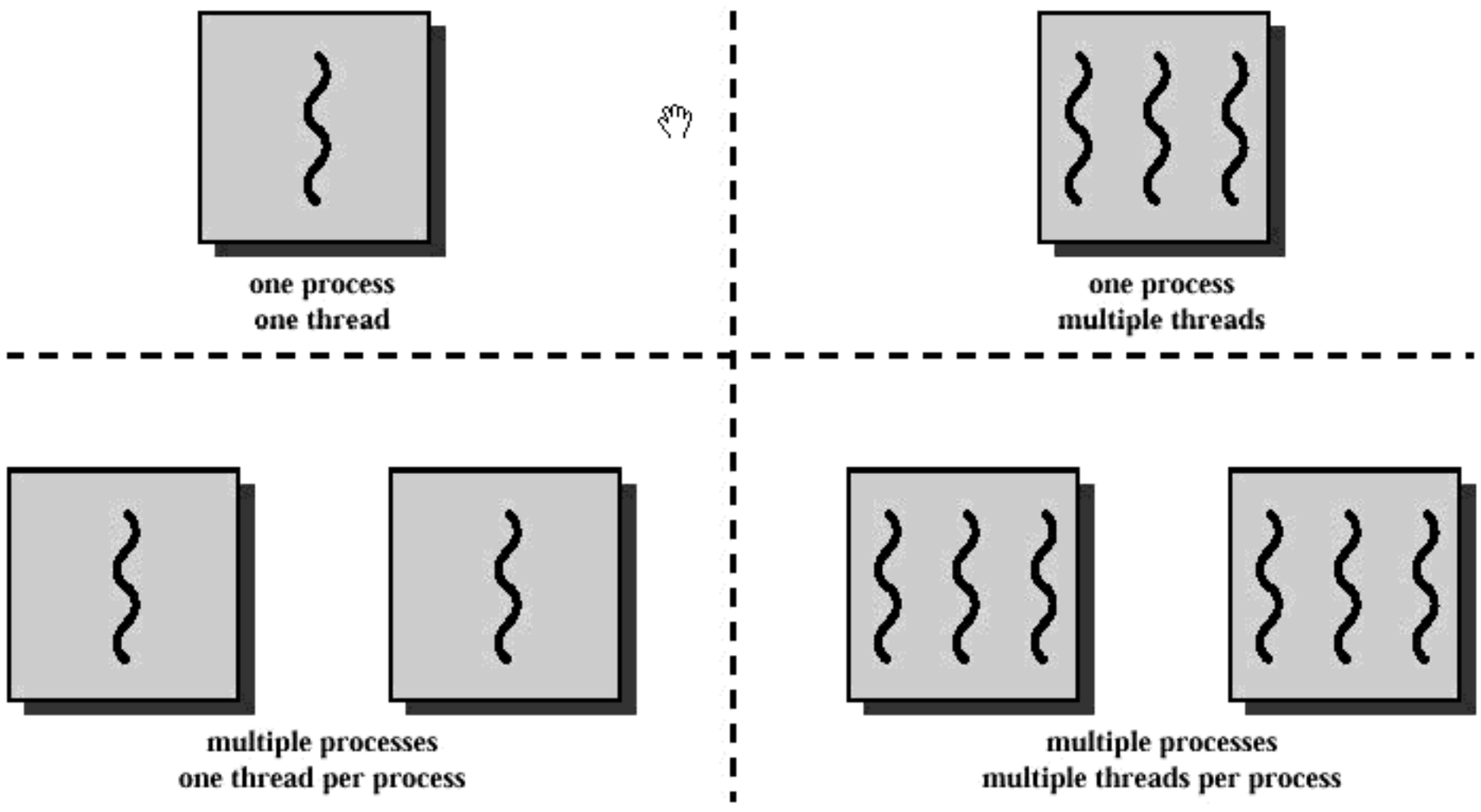

MAKE ME A SANDWICH.
SUDO MAKE ME A SANDWICH.
WHAT? MAKE IT YOURSELF.
OKAY.

sudo

```
SUDO(8)                    System Manager's Manual                    SUDO(8)



NAME
       sudo, sudoedit - execute a command as another user

SYNOPSIS
       sudo -h | -K | -k | -V
       sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
       sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
            [command]
       sudo [-AbEHnPS] [-C num] [-g group] [-h host] [-p prompt] [-u user]
            [VAR=value] [-i | -s] [command]
       sudoedit [-AknS] [-C num] [-g group] [-h host] [-p prompt] [-u user]
                file ...

DESCRIPTION
       sudo allows a permitted user to execute a command as the superuser or
       another user, as specified by the security policy.  The invoking
       user's real (not effective) user ID is used to determine the user name
       with which to query the security policy.

       sudo supports a plugin architecture for security policies and
       input/output logging.  Third parties can develop and distribute their
lines 1-25
```

Работа с процессами

Процессы и потоки



ps

```
PS(1)                    BSD General Commands Manual                    PS(1)

NAME
     ps -- process status

SYNOPSIS
     ps [-AaCcEefhjlMmrSTvwXx] [-O fmt | -o fmt] [-G gid[,gid...]]
        [-g grp[,grp...]] [-u uid[,uid...]] [-p pid[,pid...]]
        [-t tty[,tty...]] [-U user[,user...]]
     ps [-L]

DESCRIPTION
     The ps utility displays a header line, followed by lines containing
     information about all of your processes that have controlling terminals.

     A different set of processes can be selected for display by using any
     combination of the -a, -G, -g, -p, -T, -t, -U, and -u options.  If more
     than one of these options are given, then ps will select all processes
     which are matched by at least one of the given options.

     For the processes which have been selected for display, ps will usually
     display one line per process.  The -M option may result in multiple out-
     put lines (one line per thread) for some processes.  By default all of
     these output lines are sorted first by controlling terminal, then by
lines 1-25
```

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» ps aux
USER               PID  %CPU %MEM      VSZ    RSS   TT  STAT STARTED      TIME COMMAND
_windowserver      226   4.7  0.8  7628592  64308   ??  Ss   Wed10PM 142:50.68 /Syste
sobolev          31528   1.0  1.0  5425944  84656   ??  S    12:15PM   0:49.53 /Appli
sobolev          31521   0.9  0.9  5802404  74664   ??  R    12:15PM   1:20.26 /Appli
sobolev          31523   0.8  0.4  4718476  32904   ??  S    12:15PM   0:26.11 /Appli
sobolev          51369   0.1  1.2  5268120 104132   ??  S    12:39PM   2:30.74 /Appli
root             71343   0.0  0.1  4307460   9624   ??  Ss    2:43PM   0:00.05 automo
sobolev          71013   0.0  0.4  4382728  31436   ??  Ss    2:42PM   0:00.38 /Syste
sobolev          71011   0.0  0.4  4892216  32276   ??  S     2:42PM   0:01.44 /Syste
sobolev          67736   0.0  3.7  7695104 309364   ??  S     2:28PM   0:11.45 /Appli
sobolev          66511   0.0  0.2  4324440  16324   ??  S     2:17PM   0:00.16 /Syste
sobolev          65579   0.0  0.2  4354400  15900   ??  S     2:08PM   0:00.77 /Syste
sobolev          58817   0.0  0.1  4322564  12444   ??  S     1:19PM   0:00.60 /Syste
sobolev          54806   0.0  0.1  4322564  12460   ??  S    12:55PM   0:01.25 /Syste
sobolev          53420   0.0  0.9  5381824  77844   ??  S    12:50PM   0:07.47 /Appli
sobolev          53419   0.0  0.1  4298332   4772 s001  Ss+  12:50PM   0:01.37 /usr/l
sobolev          52927   0.0  0.3  4364528  22160   ??  S    12:43PM   0:19.55 /Syste
sobolev          52926   0.0  0.2  4322564  12636   ??  S    12:43PM   0:01.47 /Syste
sobolev          52357   0.0  0.1  4822624   5172   ??  Ss   12:42PM   0:00.07 /Syste
sobolev          51384   0.0  1.4  8890964 115748   ??  S    12:39PM   0:11.27 /Appli
sobolev          51383   0.0  0.4  4548484  32372   ??  S    12:39PM   0:02.28 /Appli
sobolev          51380   0.0  0.1  4333124   6120   ??  Ss   12:39PM   0:00.10 /Syste
sobolev          51378   0.0  0.4  4703244  33440   ??  S    12:39PM   0:01.17 /Appli
sobolev          51371   0.0  0.2  4460800  16256   ??  S    12:39PM   0:00.08 /Appli
```

kill

```
KILL(1)                    BSD General Commands Manual                    KILL(1)

NAME
     kill -- terminate or signal a process

SYNOPSIS
     kill [-s signal_name] pid ...
     kill -l [exit_status]
     kill -signal_name pid ...
     kill -signal_number pid ...

DESCRIPTION
     The kill utility sends a signal to the processes specified by the pid
     operands.

     Only the super-user may send signals to other users' processes.

     The options are as follows:

     -s signal_name
             A symbolic signal name specifying the signal to be sent instead
             of the default TERM.

     -l [exit_status]
lines 1-25
```

killall

```
KILLALL(1)                BSD General Commands Manual                KILLALL(1)

NAME
     killall -- kill processes by name

SYNOPSIS
     killall [-delmsvz] [-help] [-u user] [-t tty] [-c procname] [-SIGNAL]
             [procname ...]

DESCRIPTION
     The killall utility kills processes selected by name, as opposed to the
     selection by pid as done by kill(1).  By default, it will send a TERM
     signal to all processes with a real UID identical to the caller of
     killall that match the name procname.  The super-user is allowed to kill
     any process.

     The options are as follows:

           -v          Be more verbose about what will be done.

           -e          Use the effective user ID instead of the (default)
                       real user ID for matching processes specified with the
                       -u option.

lines 1-25
```

Пример

```
1 #!/usr/bin/env sh
2
3 kill -9 31154
4 killall Chrome
```

htop

```
  1  [||||||||||||||||||||          57.6%]   Tasks: 358, 1213 thr; 4 running
  2  [|||                            6.0%]   Load average: 3.93 4.40 3.13
  3  [||||||||||||||||||            56.7%]   Uptime: 3 days, 13:11:39
  4  [|||                            6.6%]
  Mem[||||||||||||||||||||||3.81G/8.00G]
  Swp[||||                       114M/1.00G]

    PID USER      PRI  NI  VIRT   RES S CPU% MEM%   TIME+  Command
  12391 sobolev    17   0 4272M  136M ? 99.3  1.7  0:57.35 /Users/sobolev/Documents/gith
  31521 sobolev    24   0 5709M 77632 ?  0.7  0.9  3:39.73 /Applications/Hyper.app/Conte
  12403 sobolev    17   0 5264M 96292 ?  1.4  1.1  0:02.16 /Applications/Hyper.app/Conte
  30579 sobolev    16   0 8474M  429M ?  3.3  5.2  3h38:08 /Applications/Firefox.app/Con
    395 sobolev    17   0 4918M 84464 ?  0.0  1.0  1:54.80 /System/Library/CoreServices/
  31523 sobolev    17   0 4602M 57680 ?  0.6  0.7  1:10.90 /Applications/Hyper.app/Conte
  91047 sobolev    16   0 7436M  188M ?  0.6  2.3  0:59.92 /Applications/Firefox.app/Con
  91051 sobolev    24   0 7416M  148M ?  0.0  1.8  0:07.39 /Applications/Firefox.app/Con
  13719 sobolev    24   0 4196M  2452 R  0.1  0.0  0:00.02 htop
  90935 sobolev    16   0 7512M  131M ?  0.1  1.6  0:47.51 /Applications/Firefox.app/Con
    445 sobolev    32   0 5829M 10068 ?  0.0  0.1  1:40.35 /Applications/Flux.app/Conten
  92145 sobolev    24   0 22.7G  223M ?  0.1  2.7  0:20.30 /Applications/Visual Studio C
  98551 sobolev    24   0 18.6G  161M ?  0.0  2.0  0:08.88 /Applications/Visual Studio C
  94292 sobolev    24   0 7399M  131M ?  0.1  1.6  0:11.26 /Applications/Firefox.app/Con
  77401 sobolev    24   0 7563M  167M ?  0.0  2.0  1:15.02 /Applications/Firefox.app/Con
  91592 sobolev    24   0 7435M  115M ?  0.1  1.4  0:26.80 /Applications/Firefox.app/Con
  F1Help  F2Setup  F3Search F4Filter F5Tree  F6SortByF7Nice - F8Nice + F9Kill   F10Quit
```

Поиск по тексту

grep

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» cat code/code4/items.txt
```

```
       │ File: code/code4/items.txt
───────┼────────────────────────────────────────────────────────────────────────
   1   │ Samsung
   2   │ Apple
   3   │ Microsoft
   4   │ Google
```

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» grep Goog code/code4/items.txt
Google

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Удобные замены

- https://github.com/ggreer/the_silver_searcher
- https://github.com/BurntSushi/ripgrep

Поиск файла

find

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» find . -name items.txt
./code/code4/items.txt

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» find . -name 'items.*'
./code/code4/items.txt

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» find . -name 'items.*' -type f
./code/code4/items.txt

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Удобные замены

- https://github.com/sharkdp/fd

- https://github.com/junegunn/fzf

Pipes

Пайпы позволяют соединять команды воедино через |

find

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» ps aux | grep zsh
sobolev          31527   1.6  0.1  4298944   6280 s000  Ss   12:15PM   0:24.62 /usr/lo
cal/bin/zsh --login
sobolev          74204   0.0  0.0  4277256    716 s000  R+    3:01PM   0:00.00 grep --
color=auto --exclude-dir=.bzr --exclude-dir=CVS --exclude-dir=.git --exclude-dir=.hg -
-exclude-dir=.svn zsh

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

Алгоритм

- Сначала делаем `ps aux`

- По его результатам делаем `grep`

xargs – полезная штука для композиции пайпов

xargs

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» ls
README.md code       course.md ex.py     ex.sh      helpers   homework  lectures

~/Documents/Work/consulting/X5/x5_school_common  master ✗
» ls | sort -fd | xargs echo "sorted: $1"
sorted:  code course.md ex.py ex.sh helpers homework lectures README.md

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

xargs

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» find . -name '*.sh' -type f | sort -n | xargs wc
       2       11       86 ./code/code1/config.sh
      29       70      421 ./code/code1/first_example/git.sh
      76       90      523 ./code/code3/files.sh
      32       30      186 ./code/code3/folders.sh
      56       73      479 ./code/code3/permissions.sh
      76       71      570 ./code/code4/functions/arguments.sh
       9       16      105 ./code/code4/functions/declaration.sh
      51       74      614 ./code/code4/functions/locals.sh
      43       71      454 ./code/code4/functions/returns.sh
      95      172      926 ./code/code4/ifs.sh
      60       71      484 ./code/code4/loops.sh
      10       37      183 ./code/code4/pipes.sh
      28       34      243 ./code/code4/vars.sh
       3       12       65 ./ex.sh
     570      832     5339 total

~/Documents/Work/consulting/X5/x5_school_common  master ✗
»
```

xargs

```sh
#!/usr/bin/env sh

find . -name "*.sh" -print0 | xargs -0 rm -rf
```

Практика

Файл "pipes.sh"

set

gnu.org/software/bash/manual/html_node/The-Set-Builtin.html

set

- **-o errexit** или **-e** – падай на первой ошибке
- **-x** – покажи команды, которые выполняются
- **-o nounset** – падай на необъявленных переменных
- **-o pipefail** – падай на проблемах в pipes

ln

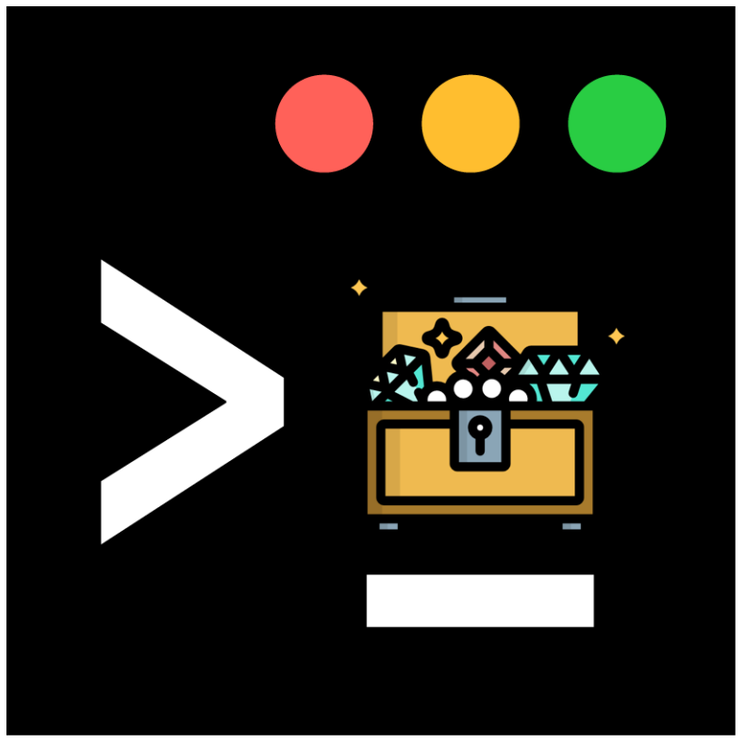
```
LN(1)                    BSD General Commands Manual                    LN(1)

NAME
     link, ln -- make links

SYNOPSIS
     ln [-Ffhinsv] source_file [target_file]
     ln [-Ffhinsv] source_file ... target_dir
     link source_file target_file

DESCRIPTION
     The ln utility creates a new directory entry (linked file) which has the
     same modes as the original file.  It is useful for maintaining multiple
     copies of a file in many places at once without using up storage for the
     ``copies''; instead, a link ``points'' to the original copy.  There are
     two types of links; hard links and symbolic links.  How a link
     ``points'' to a file is one of the differences between a hard and sym-
     bolic link.

     The options are as follows:

     -F    If the target file already exists and is a directory, then remove
           it so that the link may occur.  The -F option should be used with
           either -f or -i options.  If none is specified, -f is implied.
lines 1-25
```

shellcheck

shellcheck.net

shellcheck

```
~/Documents/Work/consulting/X5/x5_school_common  master ✗
» shellcheck ex.sh

In ex.sh line 3:
new_var=1
^-----^ SC2034: new_var appears unused. Verify use (or export if used externally).


In ex.sh line 4:
MY_ARRAY=(1 2 3)  # not POSIX
          ^-----^ SC2039: In POSIX sh, arrays are undefined.


In ex.sh line 8:
for item in ${MY_ARRAY[*]}; do
            ^-------------^ SC2039: In POSIX sh, array references are undefined.

For more information:
  https://www.shellcheck.net/wiki/SC2034 -- new_var appears unused. Verify us...
  https://www.shellcheck.net/wiki/SC2039 -- In POSIX sh, array references are...

~/Documents/Work/consulting/X5/x5_school_common  master ✗                    1 ⚠
»
```



t.me/
opensource_findings

Краткий обзор пройденного

- Переменные
- Условия и циклы
- Функции
- Sub-shells и pipes
- Полезные команды

Выводы

- shell – вполне себе язык программирования
- На нем нужно и можно писать инфраструктурный код (и даже делать большие проекты)
- В Linux есть множество полезных примитивов для работы
- Множество готовых команд: только бери и делай!