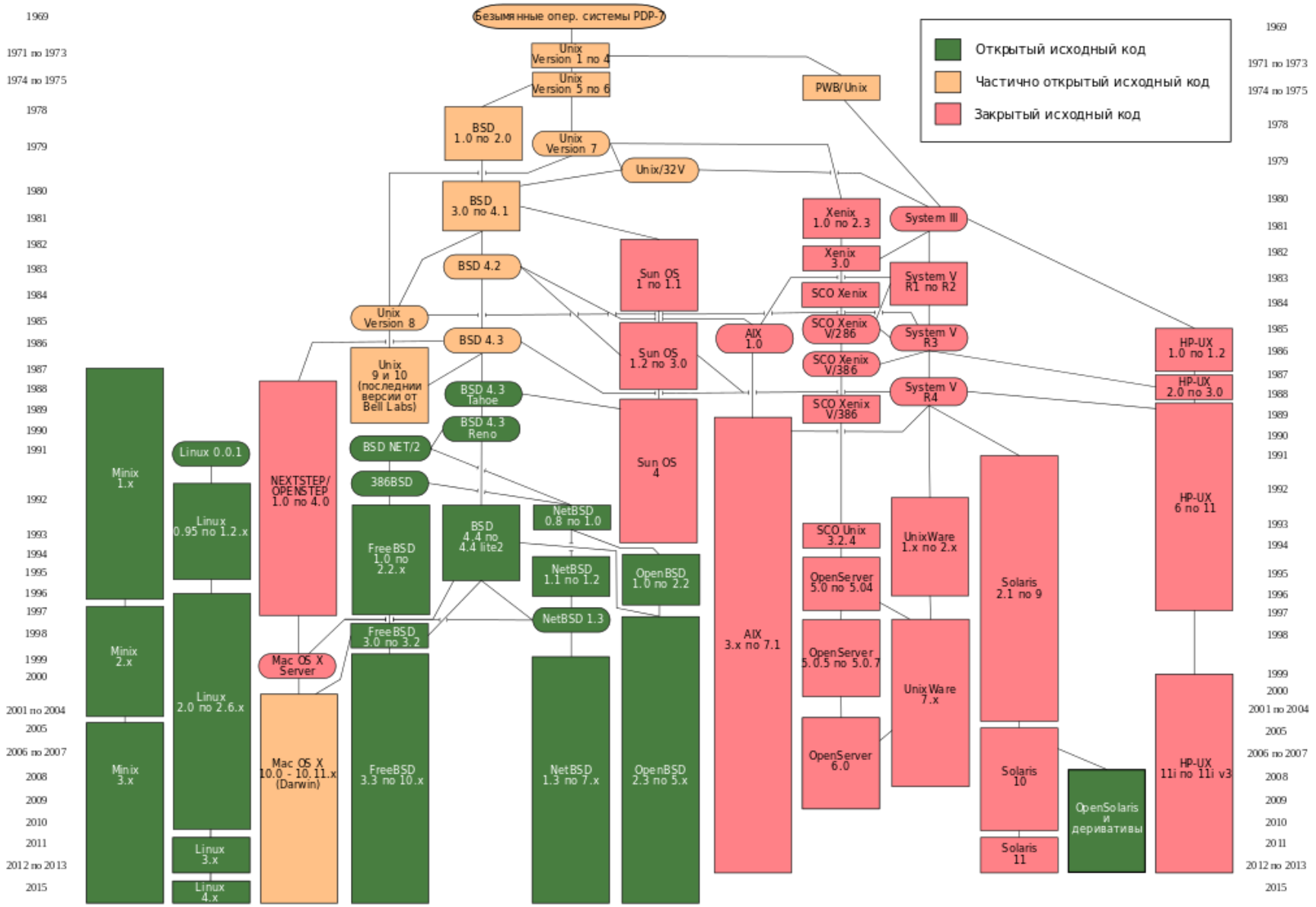
Основы Unix

Часть 1. Архитектура. Файлы. Пользователи. Полномочия.

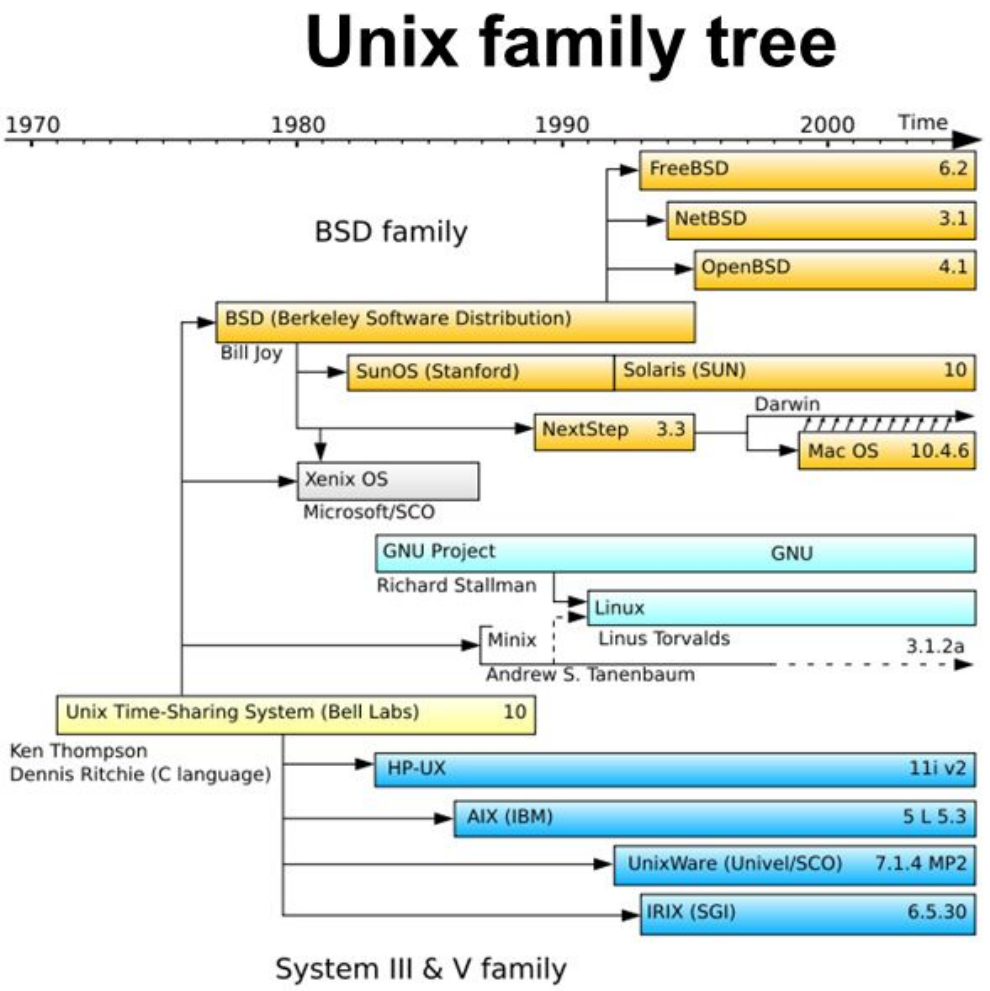
Unix

- #2 для десктоп приложений (MacOS, Linux)
- #1 для серверов в Интернете (~90%)
- #1 для вычислительных кластеров
- #1 для Enterprise OS (Linux, Solaris)
- #1 для встроенного оборудования (incl. TV)
- #1 для мобильных устройств (Android, iOS)

Семейство Unix (1)



Семейство Unix (2)



Семейство Unix (3)

- GNU/Linux (open-source initiative to bring a Unix-like environment to both the desktop and server space, Linus Torvalds)
- MacOS X (Apple's desktop operating system)
- BSD (branch from the earliest Unix specs, following the design principles of the Berkeley Software Distribution)
  - FreeBSD, NetBSD, OpenBSD
- AIX (series of Unix-based operating environments developed by IBM for its server)
- Solaris (proprietary server operating system based on Unix and developed by Sun Microsystems)
  - OpenSolaris (open-source variant of Solaris)
- HP-UX (series of Unix-based operating environments developed by HP for its servers)
- Minix (Unix-like open-source project by Andrew Tanenbaum)

Семейство GNU/Linux

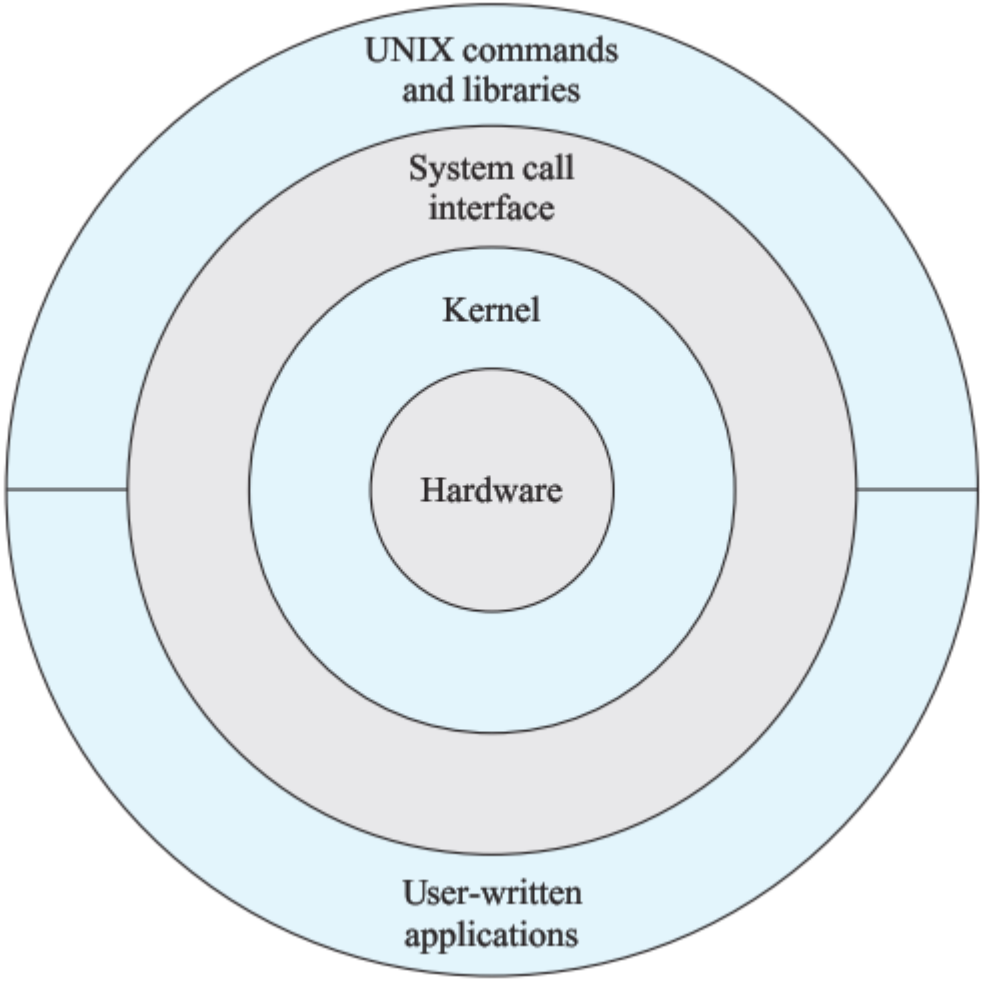
Наиболее популярные дистрибутивы и их ответвления:

- Debian (stable distro, mostly for servers)
  - Ubuntu (user-friendly desktop and server distro, based on Debian)
    - Linux Mint (user-friendly desktop distro, based on Ubuntu)
    - Elementary OS (user-friendly distro, based on Ubuntu)
- Slackware (oldest Linux distro, good for low-level Linux education)
  - openSUSE (more user-friendly distro, based on Slackware)
- Arch Linux (minimalistic Linux distro)
- RedHat (enterprise distro with support)
  - CentOS (based on enterprise RedHat distro with support, used for servers)
- Fedora (user-friendly distro, supported by RedHat)

Все дистрибутивы базируются на ядре Linux и, как правило, имеют одинаковый минимальный набор базовых утилит. Отличия: программы, драйвера устройств, менеджер пакетов, стартовые скрипты, поддержка разных графических окружений, наличие открытого исходного кода, подходы к поддержке ОС.

https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg

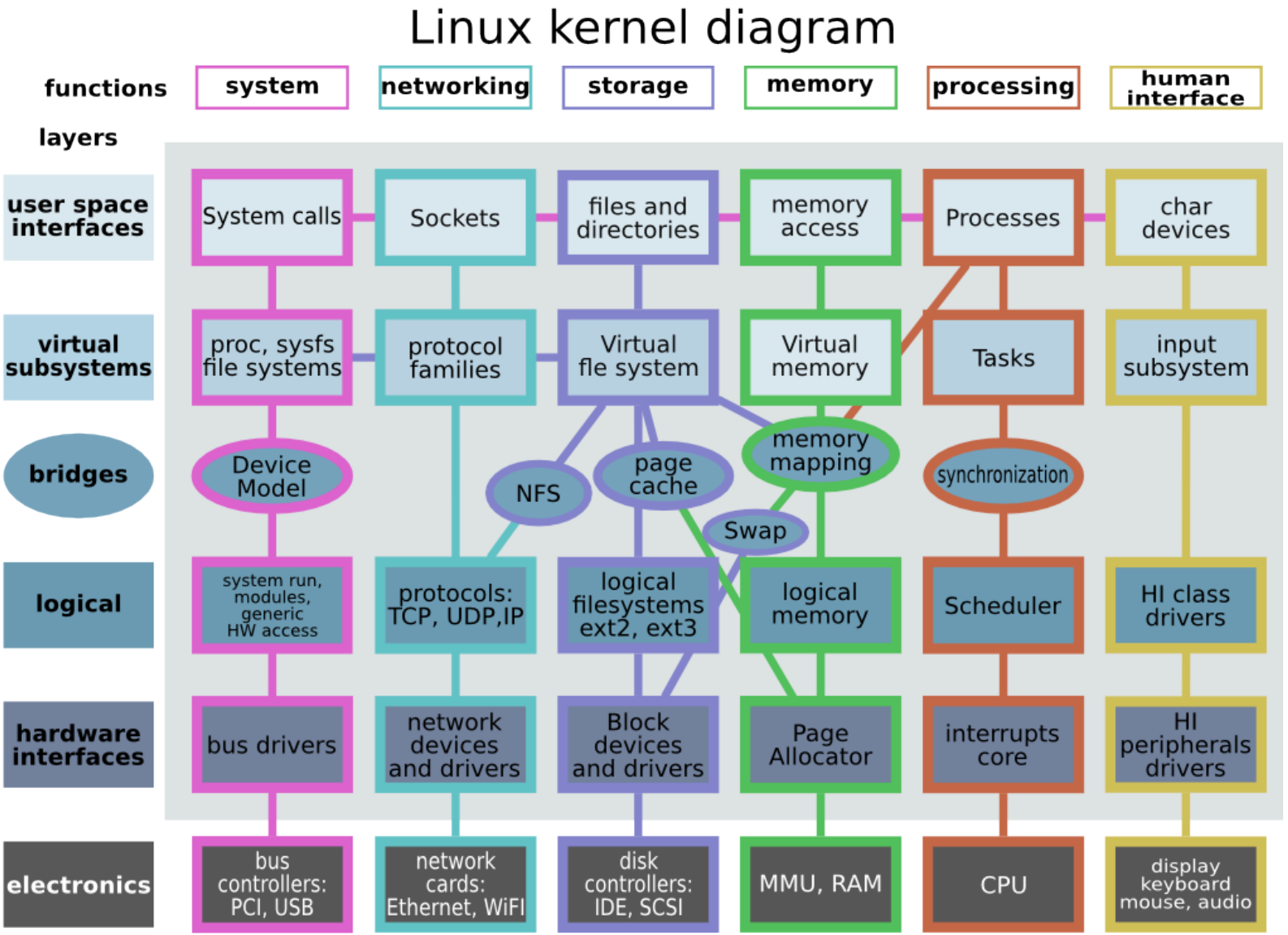
Архитектура Unix



Особенности архитектуры Unix

- Изначально многопользовательская ОС
- Графический модуль отделён от ядра
- Драйвера устройств находятся непосредственно в ядре для улучшения производительности
- По-умолчанию идёт с поддержкой удалённого доступа
- Так как драйвера устройств находятся внутри ядра в качестве модулей, и есть фиксированный API ядра (system calls), то приложения легко переносить с одной аппаратной платформы на другую

Ядро Linux


Linux kernel diagram
functions
system
networking
storage
memory
processing
human interface
layers
user space interfaces
virtual subsystems
bridges
logical
hardware interfaces
electronics
System calls
Sockets
files and directories
memory access
Processes
char devices
proc, sysfs file systems
protocol families
Virtual fle system
Virtual memory
Tasks
input subsystem
Device Model
NFS
page cache
memory mapping
synchronization
Swap
system run, modules, generic HW access
protocols: TCP, UDP,IP
logical filesystems ext2, ext3
logical memory
Scheduler
HI class drivers
bus drivers
network devices and drivers
Block devices and drivers
Page Allocator
interrupts core
HI peripherals drivers
bus controllers: PCI, USB
network cards: Ethernet, WiFI
disk controllers: IDE, SCSI
MMU, RAM
CPU
display keyboard mouse, audio
© 2007-2009 Constantine Shulyupin http://www.MakeLinux.net/kernel/diagram

System call interface

The system call is the fundamental interface between an application and the Linux kernel.

http://man7.org/linux/man-pages/man2/syscalls.2.html

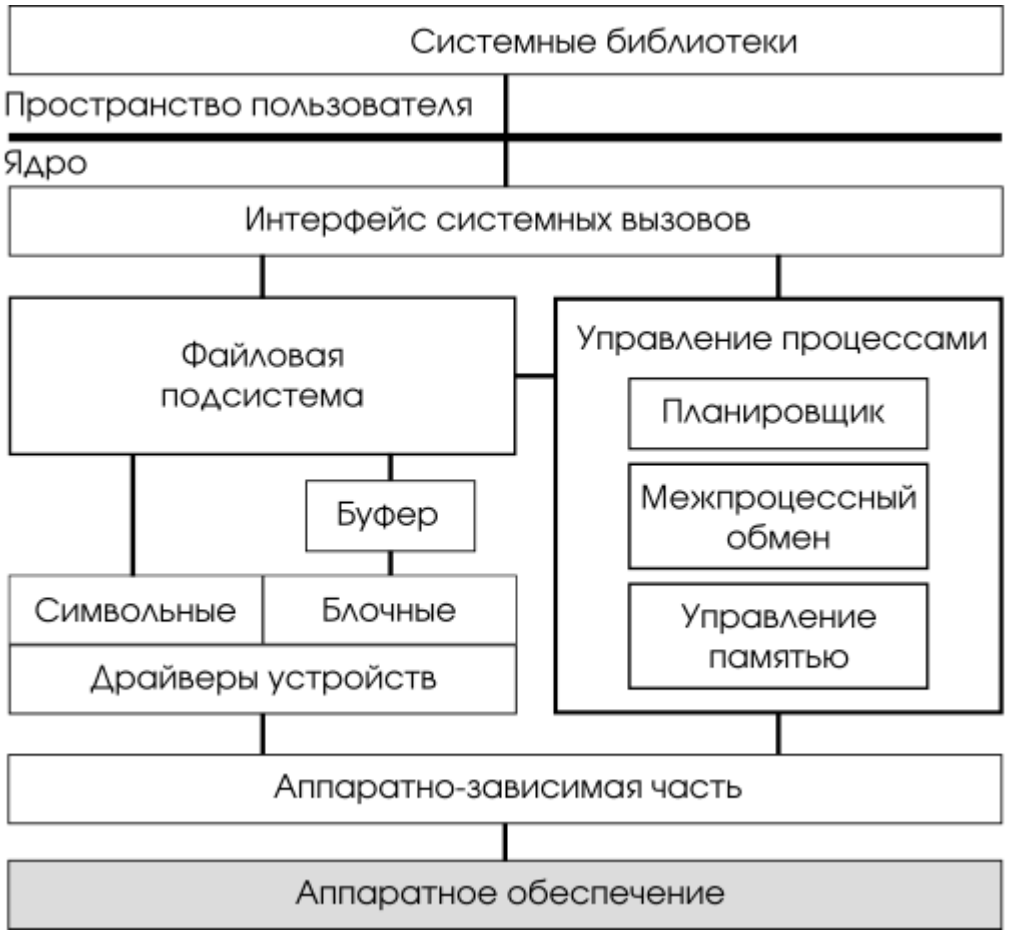
Example system calls:

- open
- read
- write
- close
- wait
- exec
- ...

POSIX

The Portable Operating System Interface (POSIX)[1] is a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems. POSIX defines the application programming interface (API), along with command line shells and utility interfaces, for software compatibility with variants of Unix and other operating systems.

Файлы и процессы

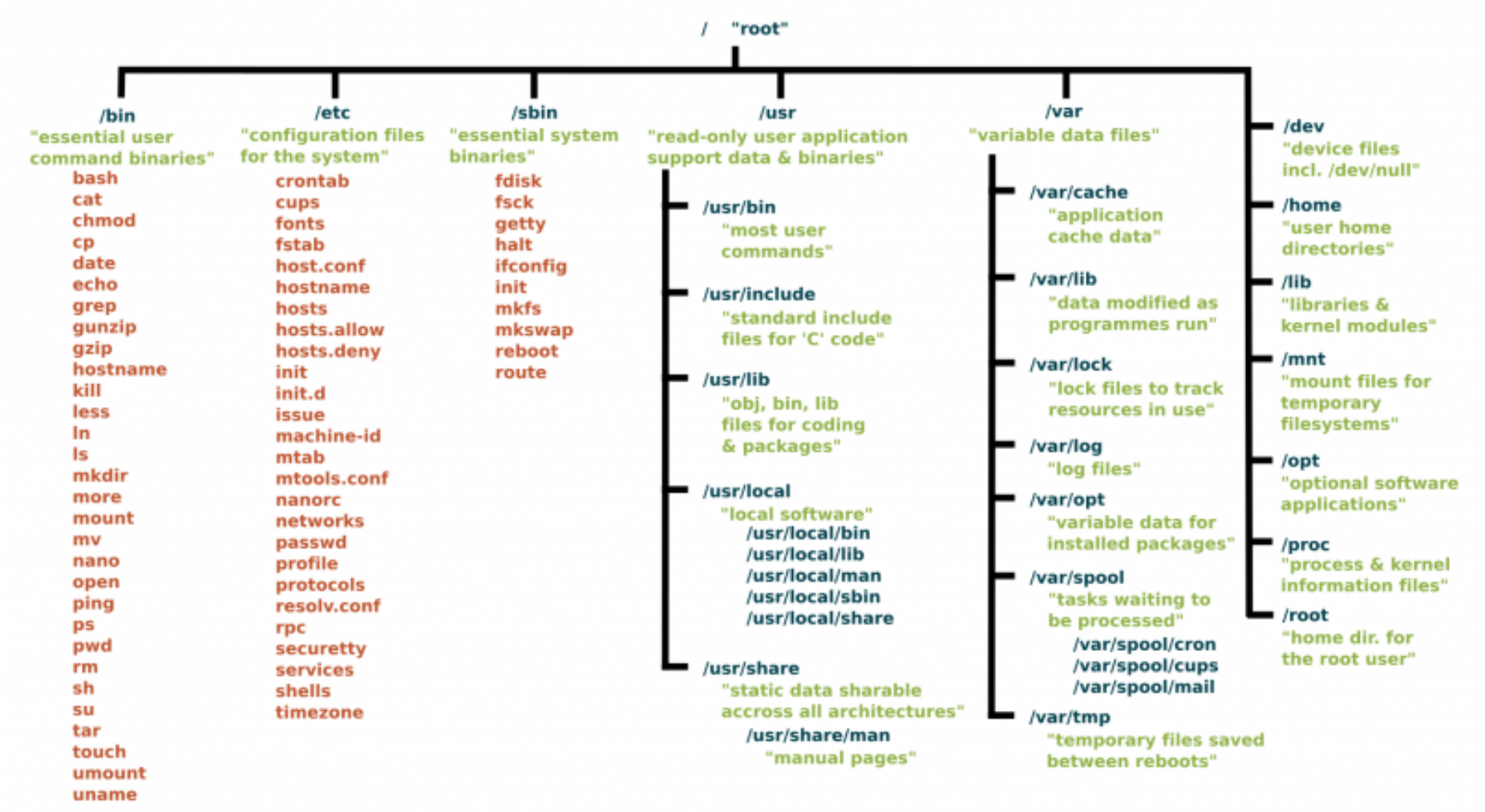


Источник: http://heap.altlinux.org/modules/unix_base_admin.dralex/ch01s02.html

Файловая система

- Древовидная структура
- Корневая точка монтирования /
- Имена файлов и каталогов регистрозависимые
- Специальные файлы устройств /dev
- Символическая ссылка (псевдоним)
- Именованные каналы и сокеты (взаимодействие между процессами)

Типовая структура файловой системы



* Может немного отличаться в зависимости от дистрибутива.

man

Чтобы посмотреть справочную страницу команды:

$ man <command>

Например, для команды копирования:

$ man cp

Или сохранить справочную страницу в файл (для печати):

$ man cp > cp_man.txt

Операции с каталогами

$ cd <dir>
$ cd ..
$ cd ~
$ ls
$ ls <dir>
$ mkdir <dir>
$ cp -r <dir1> <dir2>
$ mv <dir1> <dir2>
$ rm -r <dir>

Операции с файлами

$ touch <file>
$ less <file>
$ more <file>
$ cp <file> <file.backup>
$ rm <file>
$ mv <file> <newfile>
$ echo "Hello!" >> <file>
$ cat <file>
$ tac <file>

Пользователи и группы

$ su
$ su - <user>

$ useradd <user>
$ groupadd <group>

$ usermod -aG <ftp> <user>
$ groupmod -n <group_m> <group>

$ userdel <user>
$ groupdel <group>

$ passwd [<user>]

Владение

```
$ ls -o
drwxrwxr-x  2 user 4096 дек 17 11:48 dir
-rw-rw-r-- 1 user 12 дек 17 11:23 file.txt

$ chown user:user <file>
$ chown -R user:user <dir>
```

Полномочия

$ ls -o
drwxrwxr-x 2 user 4096 дек 17 11:48 dir
-rw-rw-r-- 1 user 12 дек 17 11:23 file.txt

Read, write, execute for owner, group and other

Например:
-r--r--r-- только на чтение для всех
-rwxr--r-- владелец может читать, писать и исполнять; остальные только читать

$ chmod u+x, g-w, o+w <file>
$ chmod u-w, g+x, o+x <file>
$ chmod u=rwx, g=r, o=r <file>

$ chmod 644 <file>
$ chmod 755 <file>

Полномочия

Octal	Binary	File Mode
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwx

Жёсткая ссылка

- Жёсткая ссылка - это псевдоним
- Жёсткая ссылка указывает на ту же область памяти, на которую указывал исходный файл
- Нельзя сделать жёсткую ссылку на каталог
- Нельзя сделать жёсткую ссылку на файл за пределами текущей ф-ой/системы
- Файл физически удаляется, когда удаляется последний псевдоним

$ ln <file> <file2>

При копировании, происходит полное дублирование. Новый файл будет указывать уже в другую область памяти.

$ cp <file2> <file3>

При отображении в каталоге, можно увидеть, сколько имён ссылаются на указанную область памяти (в данном случае есть две ссылки на область памяти: file и file2).

$ ls -l

-rw-rw-r-- 2 user user 0 дек 24 09:31 file
-rw-rw-r-- 2 user user 0 дек 24 09:31 file2

Символьная ссылка

- Символьная ссылка - это "ярлык". Можно работать с ярлыком точно также как с файлом/каталогом
- Можно делать символьную ссылку на каталог
- Можно делать ссылки на файлы из других файловых систем
- Файл/каталог, на который ссылаемся, может не существовать в данный момент
- Если удалить файл, то ссылка не удалится, но испортится. Если удалить ссылку, файл не удалится

$ ln -s <file> <file2>

При обычном копировании скопируется файл, а не ссылка (см. man cp для других вариантов):

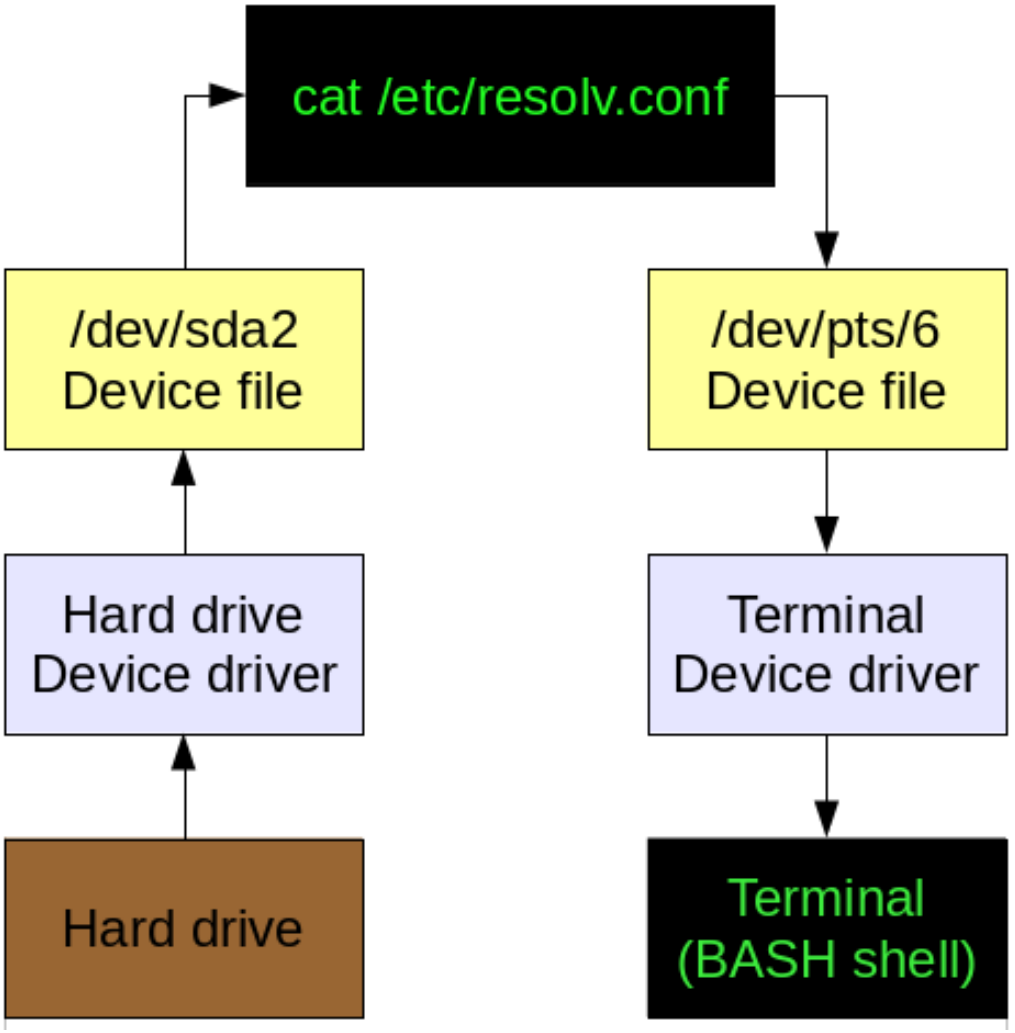
$ cp <file2> <file3>

При отображении в каталоге ссылка отображается с префиксом l. Кроме того отображается полный путь до файла, на который ссылается:

$ ls -l

-rw-rw-r-- 1 user user 0 дек 24 09:31 file
lrwxrwxrwx 1 user user 4 дек 24 09:40 file2 -> file

Файлы устройств



In Unix-like operating systems, a device file or special file is an interface to a device driver that appears in a file system as if it were an ordinary file. These special files allow an application program to interact with a device by using its device driver via standard input/output system calls. Using standard system calls simplifies many programming tasks, and leads to consistent user-space I/O mechanisms regardless of device features and functions.

Файлы устройств

$ ls -l /dev

```
brw-rw----  1 root disk       8,   0 дек 19 10:50 sda
crw--w----  1 root tty        4,   1 дек 19 10:50 tty1
crw-rw-rw-  1 root root       1,   3 дек 19 10:50 null
```

Character devices (c)

Character special files or character devices provide unbuffered, direct access to the hardware device

Block devices (b)

Block special files or block devices provide buffered access to hardware devices, and provide some abstraction from their specifics.

Pseudo-devices

Device nodes on Unix-like systems do not necessarily have to correspond to physical devices.

Пример работы с файлами устройств

Отправить строку в терминал
$ echo "Hello" > /dev/tty1

Отправить на печать
$ cat file > /dev/usb/lp0

Создать файл из псевдослучайных чисел
$ cat /dev/urandom > file

Прочитать что в ОЗУ (если есть полномочия)
$ dd if=/dev/mem bs=2048 count=100

Отчистить содержимое файла
$ cat /dev/null > access.log

Полезные ссылки

Архитектура:
https://www.opennet.ru/docs/RUS/unix/

Ядро:
https://github.com/torvalds/linux
https://www.kernel.org/doc/html/latest/index.html
http://www.makelinux.net/kernel_map/

Файловая система:

https://en.wikipedia.org/wiki/Unix_filesystem#Conventional_directory_layout
https://www.linux.com/blog/learn/intro-to-linux/2018/4/linux-filesystem-explained

Полномочия:
https://ru.wikipedia.org/wiki/Chmod

Сборка ОС на базе ядра Linux:
https://buildroot.org/