

UNIX

Introduction to OS

The concept of word "UNIX"

1. Operating system, late 60's in Bell Labs.
2. Operating System Family: Linux, Mac OS, FreeBSD, Solaris...

Linux

- Open source
- Development started at 1991, Linus Torvalds
- Developers: enthusiasts, big companies (73 thousands person-years)
- OS of UNIX family

Spread across platforms

Smartphones - 88% (Android)

Servers - 60%

Supercomputers - 97%

Embedded Systems - 50%

Netbooks - 35%

PC - 1 to 5 % (+ 12.5% macOS)

Standardization

- Despite the variety of versions of UNIX, the basis of the entire family is the fundamentally the same architecture and a number of standard interfaces (in UNIX everything is standardized - from the location of system directories and files to the interface of system calls and the list of drivers for base devices)
- POSIX

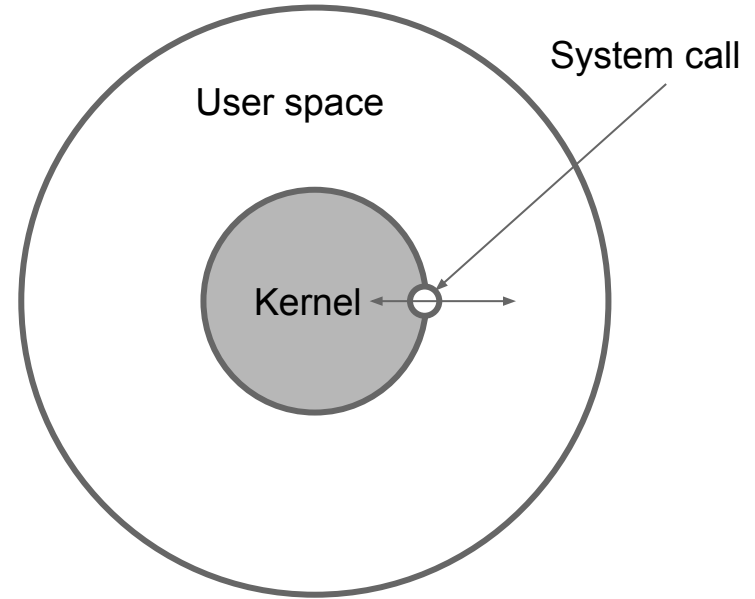
High level architecture

Kernel type: monolithic

Access levels:

User space

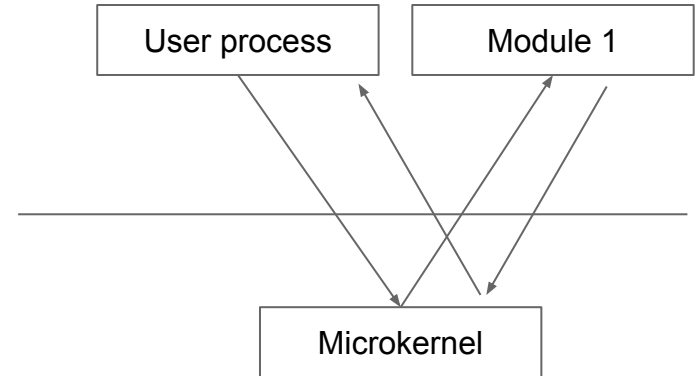
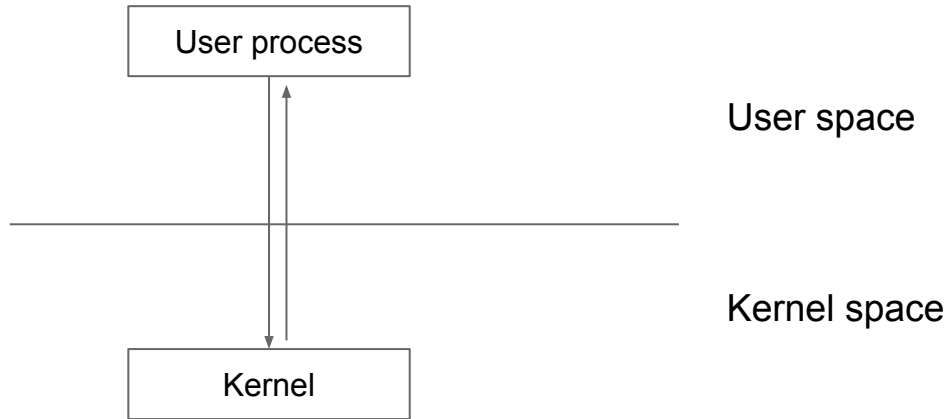
Kernel space



Monolithic VS Microkernels

Monolithic kernel is a single large process running entirely in a single address space

In **microkernels**, the kernel is broken down into separate processes, known as servers



UNIX Kernel

System Libraries

User space

Kernel space

system calls interface

Files subsystem

Processes management

scheduler

memory management

interprocess communication

hardware-dependent part

hardware

Main components of the OS

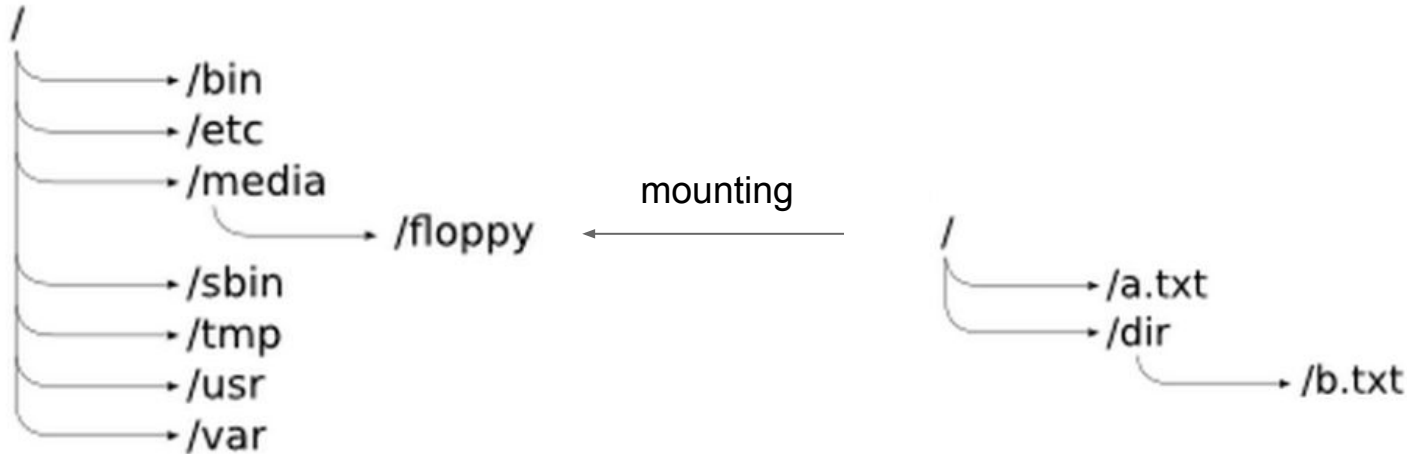
Process Management

File system:

- Organization of catalogs
- Organization of data on device

Mounting File Systems

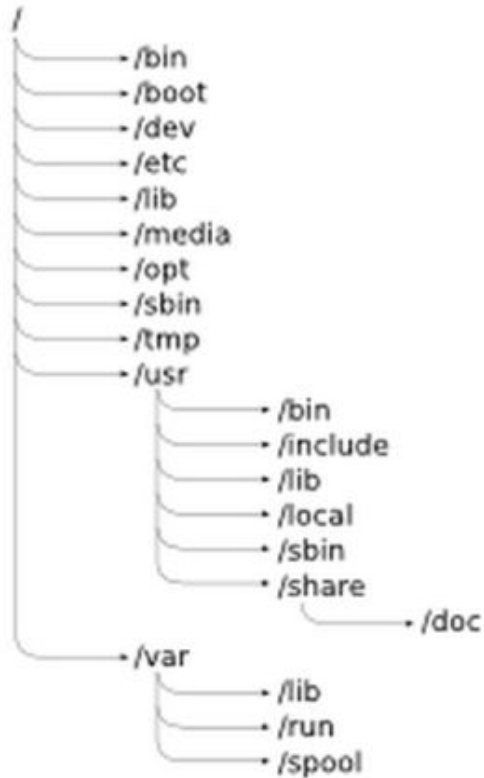
In UNIX file systems join a single tree with a common root



filename in the source file system: **/dir/b.txt**

filename after mount: **/media/floppy/dir/b.txt**

File system standard



Basic Catalogs

root directory /	The root directory / is the backbone of any UNIX FS. All other directories and files are located within the tree, generated by the root directory, regardless of their physical location.
/bin	In this directory are often used commands and utilities of the public system. This includes all the basic commands available even if only the root file system was mounted. Examples of such utilities are: ls , cp , sh , and so on .
/boot	The directory contains everything necessary for the process of loading the operating system: the loader program, the kernel image of the operating system, and so on.
/dev	The directory contains special device files that allow access to peripheral devices. The presence of such a directory does not mean that special device files can not be created elsewhere, it is simply convenient to have one directory for all files of this type.

Basic Catalogs

/etc	This directory contains system configuration files
/home (optional)	The directory contains home directories of users. Its presence in the root directory is not necessary, and the content depends on the features of a particular UNIX-like operating system.
/lib	A directory for static and dynamic libraries needed to run programs located in the /bin and /sbin directories.
/media	A standard directory for temporarily mounting file systems-for example, flash drives, DVD-ROMs, and so on.
/opt	Catalog for additional software installed in the system. Typically, this directory installs programs that are not included in the main distribution.

Basic Catalogs

/root (optional)	The home directory of the root user. Its presence in the root directory is not necessary.
/sbin	This directory contains commands and utilities for the system administrator. Examples of such commands are: route , halt , init , etc. For similar purposes, the /usr/sbin and /usr/local/sbin directories are used.
/usr	<p>This directory repeats the structure of the root directory - contains the directories /usr/bin, /usr/lib, /usr/sbin, serving for similar purposes.</p> <p>The /usr/include directory contains the C header files for all libraries in the system.</p>
/var, /tmp	Используются для хранения временных данных системных (/var) и пользовательских (/tmp) процессов. Каталог /var обычно содержит часто изменяемые системные файлы, например в каталоге /var/log размещаются системные журналы.

Command line

Command interpreter

Command shell UNIX:

- sh (shell), bash, dash, zsh
- csh (C shell) - C-like syntax (not POSIX)
- ash (Almquist shell) - for embedded systems

Command Types

- External (date, pwd, cd, ls)
- Internal (echo)
- Alias

View/Set Environment Variables

Look inside:

```
echo $PWD
```

Installation:

```
export A=1
```

```
export PATH=$PATH:/opt/tools
```

Standard environment variables

DISPLAY	The variable is used by the graphics subsystem X11 and indicates the address of the X server and the number of the screen used
EDITOR	If some utility requires editing the file, then instead of writing and using the built-in editor, you can transfer this file to edit the program, the path to which is stored in the variable EDITOR
HOME	The variable contains the name of the home directory of the current user.
LANG и LC_...	Variables that specify the interface language and other program localization parameters.
PATH	The environment variable contains a list of directories, separated by the symbol ":". This list is viewed every time the command is run - it searches for executable files with the corresponding command name.

Standard environment variables

SHELL	The name of the current shell program.
TERM	The type of terminal that is currently used. This variable is analyzed by programs in order to vary its interface depending on the capabilities of the terminal.
USER	The name of the current user.
_ (underscore)	Exactly the last command executed in the shell.

Basic Commands

Manuals

Calling up the reference guide:

man [section number] [team name]

1. user utils and other tools
2. system calls
3. library functions
4. external devices (and their representation in the system)
5. formats and tables (file types, protocols, etc.)
6. games and all sorts of "unnecessary" utilities
7. "rest", i.e. what does not fit into other sections
8. commands and tools of the system administrator

man 3 printf

Search by manuals

whatis [name]

whatis passwd

passwd (1) - update a user's authentication tokens(s)

passwd (5) - password file

passwd (8) - manual page for passwd wrapper version 1.0.5

Commands for working with FS

pwd	The pwd (print working directory) command lets you find out the name of the current directory
cd	Used to change the current directory <ul style="list-style-type: none">• cd .. — to return to the parent directory (climb one directory up through the tree);• cd . — go to the current directory;• cd / — go to the root directory;
ls	Provides a variety of information about files and directories (an analog of the DIR command for DOS-systems)
mkdir	Creating a Catalog

Commands for working with FS

rmkdir	Deletes the directory whose name is specified as a parameter. The directory to be deleted must be empty.
touch	Changes the time of the last access to the file. Creates a new file if it does not exist.
rm	Deleting files
cp	Copying files
mv	Moving/renameing a file
ln	Creating hard and symbolic links

Commands for working with text

echo	Outputs the arguments to the standard output device, separating them with spaces and ending them with a newline character (\n)
cat	Reads the specified files as arguments and outputs their contents to the standard output device
less	Used to view files that are larger than the screen size
head	Displays the first few lines of the file. By default, the first 10 rows are displayed
tail	Displays the last few lines of the file. By default, the last 10 rows are displayed

Commands for working with text

wc	It is intended for counting the number of lines, characters and words in the specified files
grep	Can search for a string in files, and can work as a filter
nano	Text editor

Commands for obtaining system information

date	Date output
cal	Display the calendar
ps	Displays a list of all running processes
top	Output of dynamically changing information about processes and used resources of the system
free	Provides information about memory usage
df	Displays information about all mounted file systems

Commands for obtaining system information

who	Displays the list of users currently working in the system
uname	Outputting the name of a running UNIX system

Commands for processes management

kill	Used to send a signal to a process
exec	Used to replace the shell process with another process
&	The "&" symbol is used to start programs in the background
<i>bg, fg u jobs</i>	Commands are used to work with tasks - processes that are started from the command interpreter

Conveyor

Redirecting the output of one command to the input of another:

```
cat /etc/passwd | grep root
```

Conveyor

archive.tar.gz



Archive - several
files combined into
one



Compressed archive

Example

Manual termination of processes

```
tail -f
```

```
ps -ef | grep tail
```

```
kill -9 1234
```

Redirecting IO

0 - Input Stream

1 - Output Stream

2 - Error Stream

```
cat /etc/passwd 1> ./temp.txt
```

```
cat /etc/passwd > ./temp.txt
```

```
cat /etc/passwd >> ./temp.txt
```

Creating scripts

```
#!/bin/bash
```

```
echo "Hello from bash script"
```

```
#!/bin/python
```

```
print "Hello from python script"
```

- + you should give the script the right to execute (`chmod +x scriptfile`)

Sample script

```
#!/usr/bin/env bash

for i in `seq 1 10`;
do
    current_time=`/usr/bin/time -f "%e" sleep $i`
    printf "%10s %s sec\n" $i $current_time;
done
```

time - without specifying the full path, the internal bash command will be executed, not the program

Security

Security Management

Rights are divided into three types:

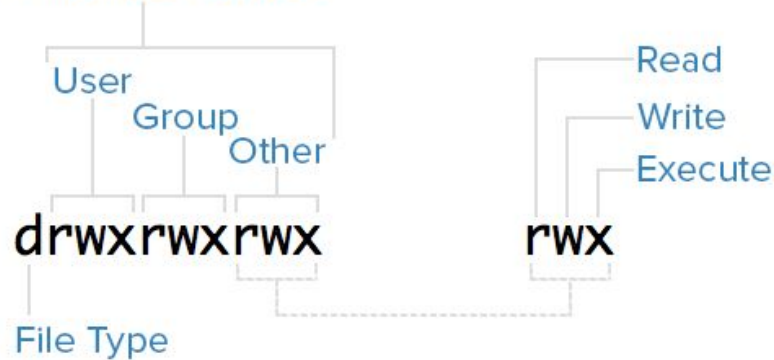
- Owner
- Group
- Others

Security Management

ls -l



Permissions Classes



r - read rights
w - write rights
x - execution rights

Values for access rights

Permission	Files	Folders
read (r)	Viewing the contents of a file	Viewing file names inside a directory
write (w)	Modify and delete a file	Delete the directory. Change the contents of the directory (create, delete, rename files).
execute (x)	Executing the file (you also need read permissions)	Access to files inside the directory (almost always should be present)

Changing file permissions

chmod $\begin{bmatrix} u \\ g \\ o \end{bmatrix} \begin{bmatrix} + \\ - \end{bmatrix} \begin{bmatrix} r \\ w \\ x \end{bmatrix}$

chmod u+rw,go-rwx sample.txt

Changing file permissions

7 5 5
├──┬──┬──
11 101 101
├──┬──┬──
1 r-x r-x
rwx

chmod 755 sample.txt

Change of owner/group

chown owner [:group] file

chown user1:group1 sample.txt

chown user2 sample.txt

chown :group2 sample.txt

Special Rights

sticky bit (Deleting a file/directory only by the owner)

chmod +t filename rwx r-- r-T

setuid/setgid bit (Running the file with owner/group rights)

chmod u+s filename rw**S** r-- r--

chmod g+s filename rwx r-**S** r--

The UNIX software environment

Compilation

```
#include <stdio.h>
```

```
int main () {  
    printf("Hello world\n");  
  
    return 0;  
}
```

```
gcc simple.c -o simple
```