

# JavaScript академија

---

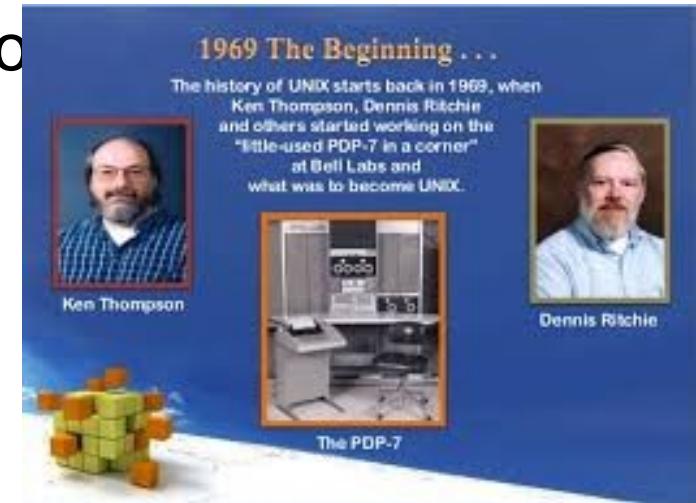
Основи на GNU/Linux оперативен систем за развој на  
веб апликација

# Содржина

- Историја на UNIX и GNU/Linux
- Работа преку GUI
- Работа преку CLI/Bash
  - Filesystem структура
  - Вовед во Bash и env
  - Навигација низ директориуми
  - Креирање манипулација на директориуми и датотеки
  - Привилегии на корисници (директориуми и датотеки)
  - Pico/Nano текстуални уредувачи
  - Креирање и распакување на компресирани датотеки
- Инсталација на софтвер (apt-get)
  - Додавање на PPA складишта (repositories)
  - Инсталација
  - Деинсталација
  - Update и upgrade на системот
- Користење на SSH
  - Конектирање на оддалечен опслужувач
  - Генерирање на ssh клучеви со ssh-keygen
- Менаџирање на сервиси

# Unix

- Потеклото на Linux датира од пред 50 години, кога компјутерските системи биле скапи и гломазни, секој со свој посебен оперативен систем.
- Тим од развиваачи во лабораториите на Bell Labs, (Ken Thompson, Dennis Ritchie, Rudd Canaday, и Doug McIlroy) започнува да работи на решение на проблемите со компатибилноста.
- Тие го развиваат Unix оперативниот систем, кој
  - Едноставен
  - Напишан во С
  - Со код што може да се искористи повторно



# Unix

- Последното е особено важно, бидејќи на тој начин Unix може да се пренесува на било која компјутерска платформа, при што само јадрото бара адаптација напишана во асемблер.
  - С е токму напишан за оваа намена.
- UNIX првично беше користен само во многу големи околини со мејнфрејм и мини-компјутери, но до крајот на 80-тите години, многу луѓе имаат домашни компјутери.
- До тоа време, нема многу UNIX верзии за персонални компјутери, повеќето луѓе користат MS DOS или Windows.

# Што е GNU / Linux?

- Linux е оперативен систем кој првично започал во 1991 година како личен проект на Линус Торвалдс, студент на Универзитетот во Хелсинки.
- Kernel верзијата 1.0 е објавена во 1994 година и денес најновата стабилна верзија е 5.8.12
- Linux е развиен под GNU GPL и изворниот код е достапен за секого.
  - Linux и различните дистрибуции не се секогаш бесплатни – но достапен е изворниот код.
- GNU / Linux може да се користи за многу намени како што се мрежно поврзување, развој на софтвер и како платформа за крајниот корисник.

# Што е GNU / Linux?

- **Linux + GNU Utilities**
- Linux јадро креирано од Линус Торвалдс
- Множество алатки напишани од Ричард Сталман од MIT, основачот на GNU и креатор на Free Software Foundation
- Основната идеја била да се создаде free Unix



# Што е GNU / Linux?

“Small programs that do one thing well”

**Shells:** BASH, TCSH, alias, watch, clear, history, chsh, echo, set, setenv

**Систем информации:** w, whoami, which, free, echo, date, cal, df, free, info

**Информации за команди:** man, info

**File System:** ls, mkdir, cd, pwd, mv, ln, touch, cat, file, find, diff, cmp, /net/<hostname>/<path>, mount, du, df, chmod, find

**Филтри:** grep, egrep, more, less, head, tail

**Мрежа:** ping, ifconfig, ssh, telnet, nslookup

**Едитори:** vim, gvim, emacs -nw, emacs

# Linux дистрибуции

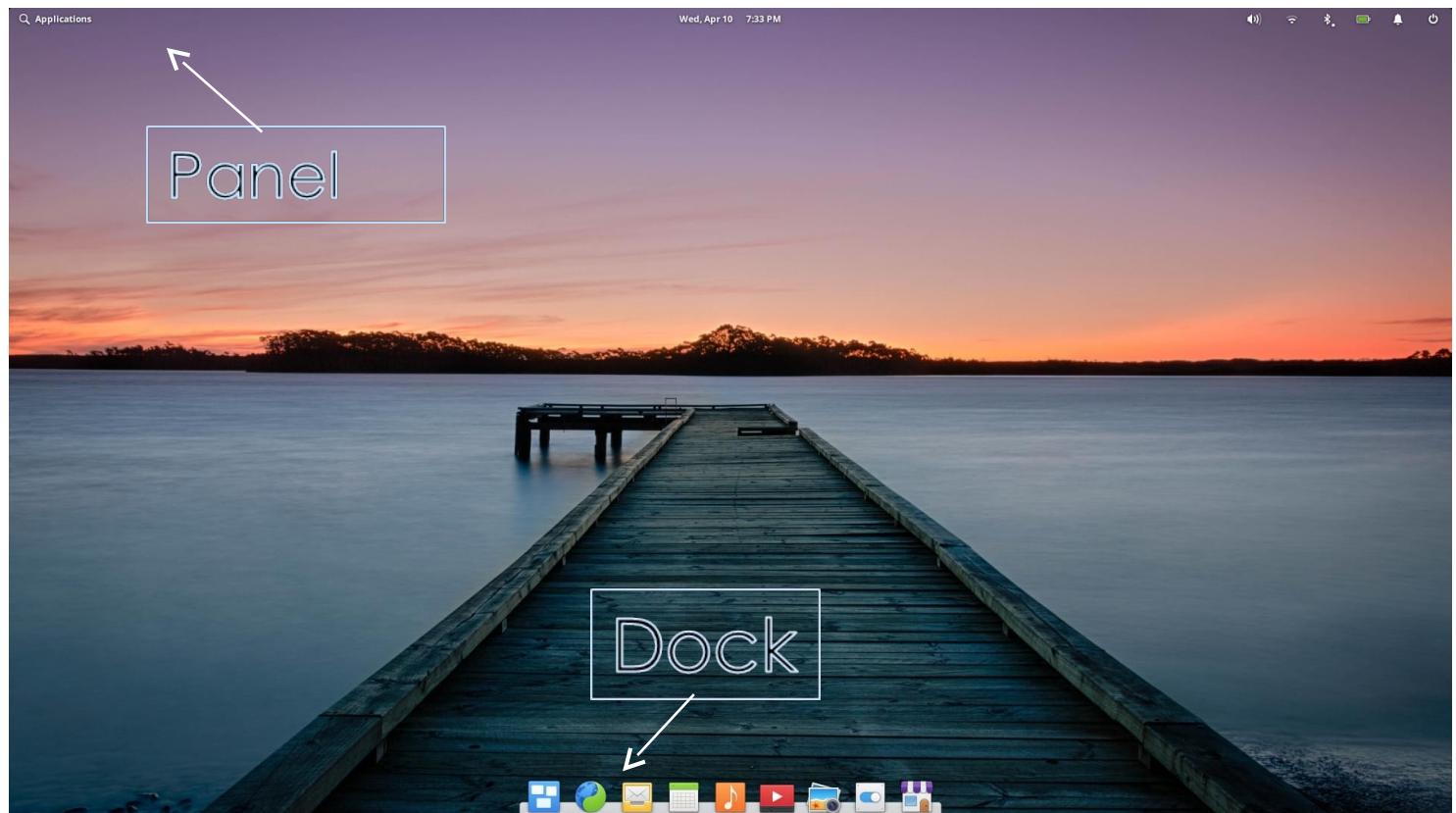
- Дистрибуција на Linux (distribution или distro) е оперативен систем направен од колекција на софтвер, кој се заснова на кернелот на Linux
- Linux има голем број на дистрибуции



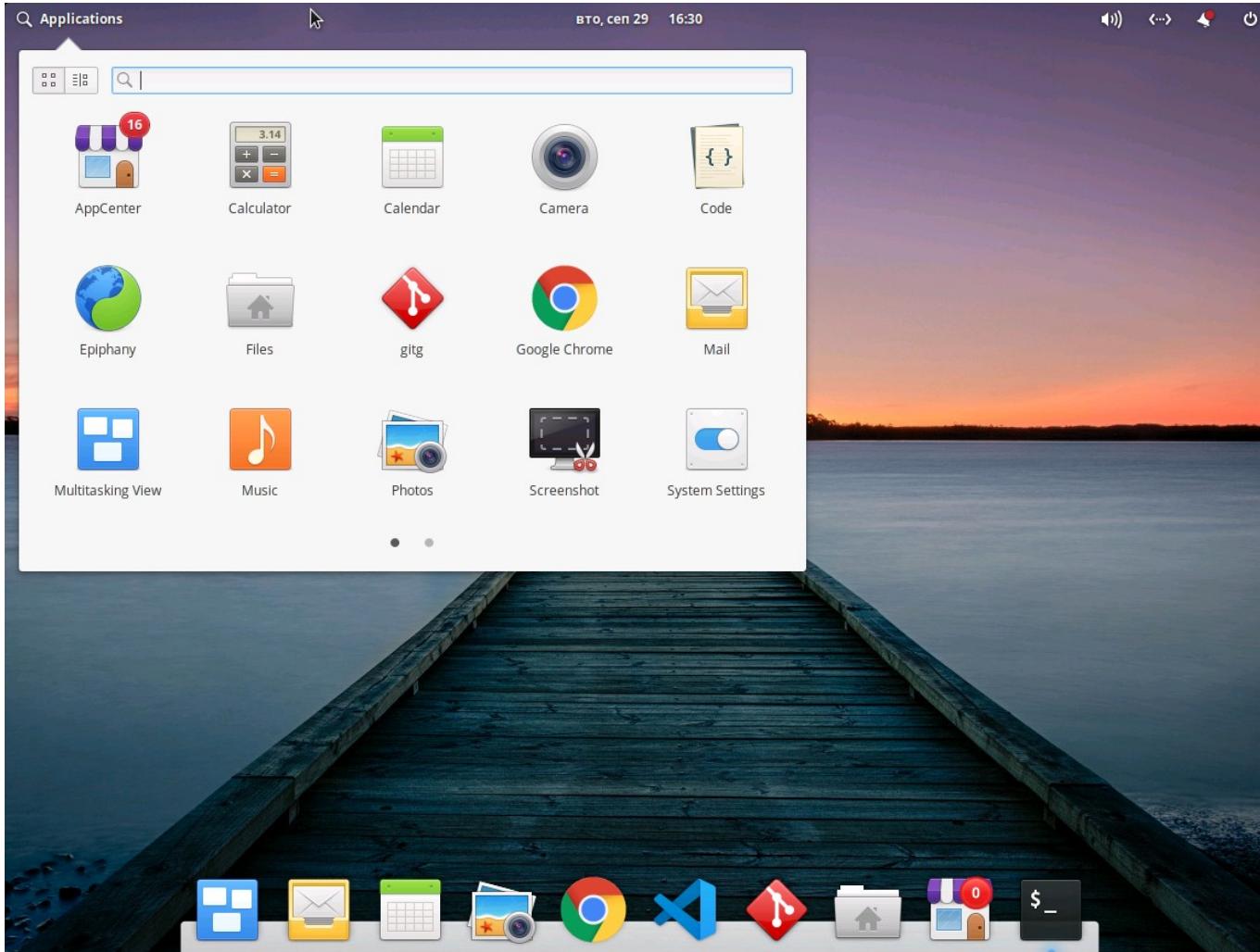
# Elementary OS

Elementary OS е дистрибуција на Linux базирана на Ubuntu LTS. Се промовира како замена за macOS и Windows и има pay-what-you-want. Оперативниот систем, работната околина и придружните апликации се развиени и одржувани од Elementary, Inc.

Elementary OS desktop



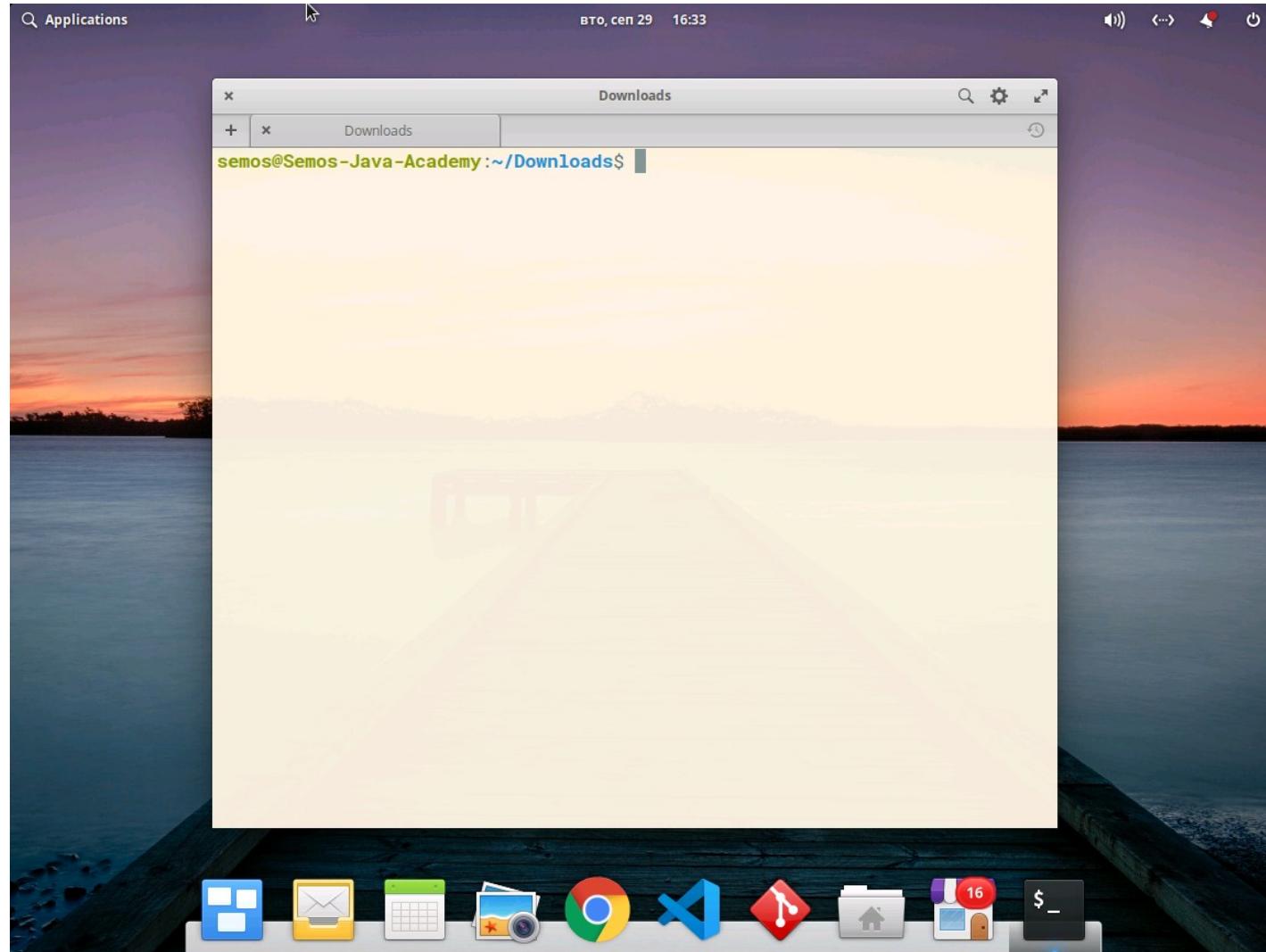
# Applications



# CLI/bash

- Командниот интерпретер (**shell**) е кориснички интерфејс кој е базиран на командна линија.
- Командниот интерпретер е истовремено интерпретер на кориснички команди и скрипт интерпретер.
- Кога корисник ќе се најави интерактивно на системот, автоматски се стартува shell.

# Terminal



# Команди

- Кај командниот интерпретер комуникацијата се одвива преку командна линија.
- Командите најчесто се извршни програми кои интерпретерот ги пронаоѓа и ги извршува.
- Тие обично ја имаат следната структура:

**команда опции аргументи**

- Пример:

**ls -l -a Tekstovi**

Алатки за системски информации:

**clear** – брише екран

**uname** – прикажува информација за системот. Најчесто користена опција е "-a"

**last** – листа низа од најавени корисници

**tty** – прикажува на кој терминал си моментално најавен.

**date** – прикажува системско време и датум

**cal** - календар

**w, who** –прикажува информации за најавени корисници

Помошни информации за команди:

**man**

# Видови на командни интерпретери

- **sh (Thompson shell)**. Најстар интерпретер, кој го напишал Ken Thompson од Bell Labs (1971-1975)
- **bash (Bourne-Again shell)**. Тој не е единствениот команден интерпретер (**shell**) кај Linux и UNIX оперативните системи. Тој е наследник на Bourne Shell (sh), кој го напишал Steve Bourne од Bell Labs. Но, bash е достапен кај сите Linux варијанти и е затоа погоден за пишување на скрипти кои се преносливи на различни системи.
- Друг популарен shell е **C Shell (csh)**, создаден на UCB, чие име потекнува од фактот што тој има синтакса слична на C програмскиот јазик.
- **TC Shell (tcsh)** е продолжение на C shell.
- **Korn Shell** напишан од David Korn од Bell Labs, вклучува особини и од Bourne shell и од C shell.
- Еден од посовремените интерпретери е **Z shell (zsh)**. Тој содржи комбинација од алатки од csh и bash.

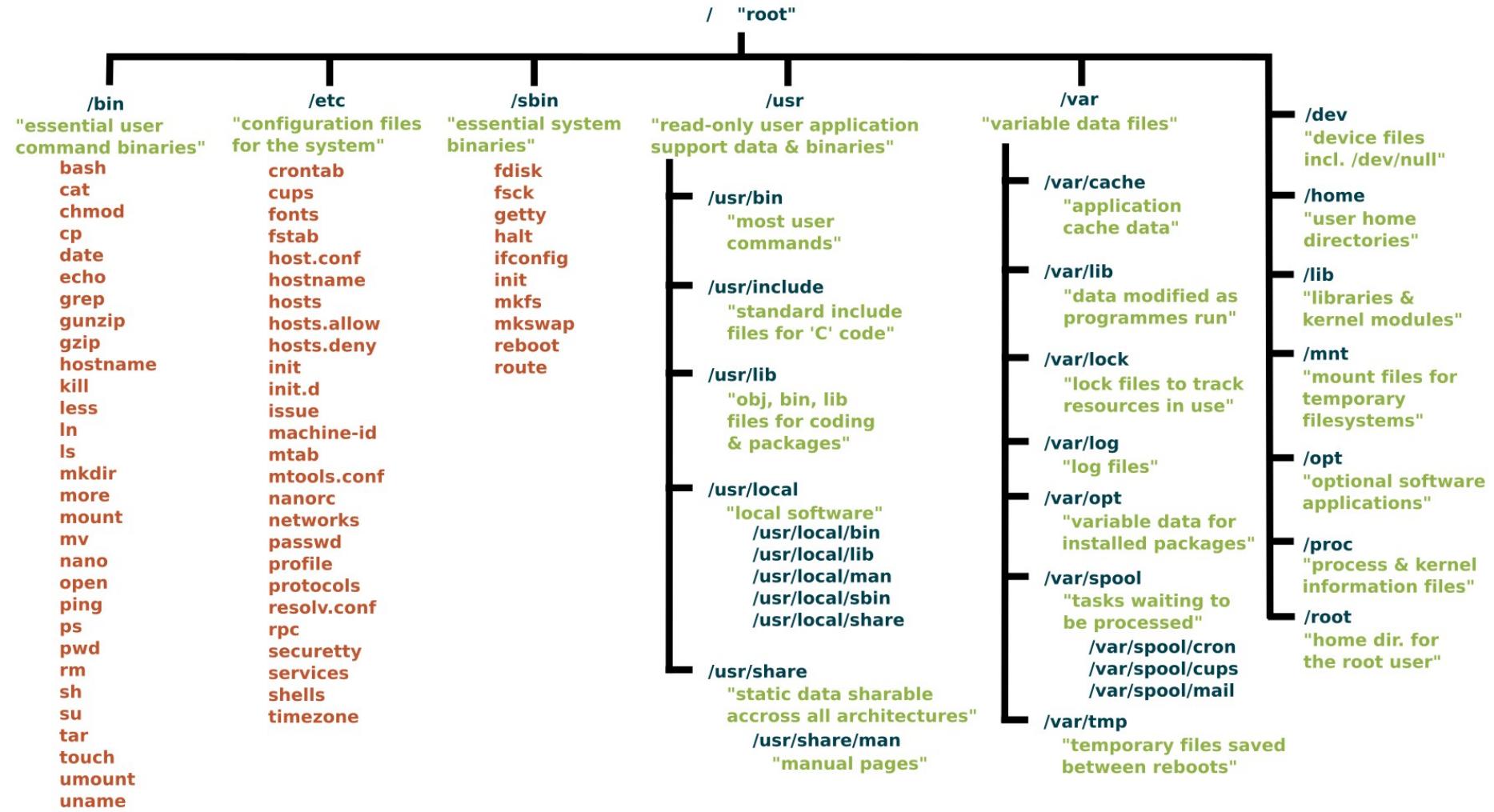
# Видови на командни интерпретери

	Bourne	C	TC	Korn	BASH
историја на команди	не	да	да	да	да
сионими	не	да	да	да	да
командни процедури	да	да	да	да	да
дополнување на имиња на датотеки	не	да	да	да	да
уредување на командната линија	не	не	да	да	да
контрола на задачи	не	да	да	да	да

**\$echo \$SHELL**

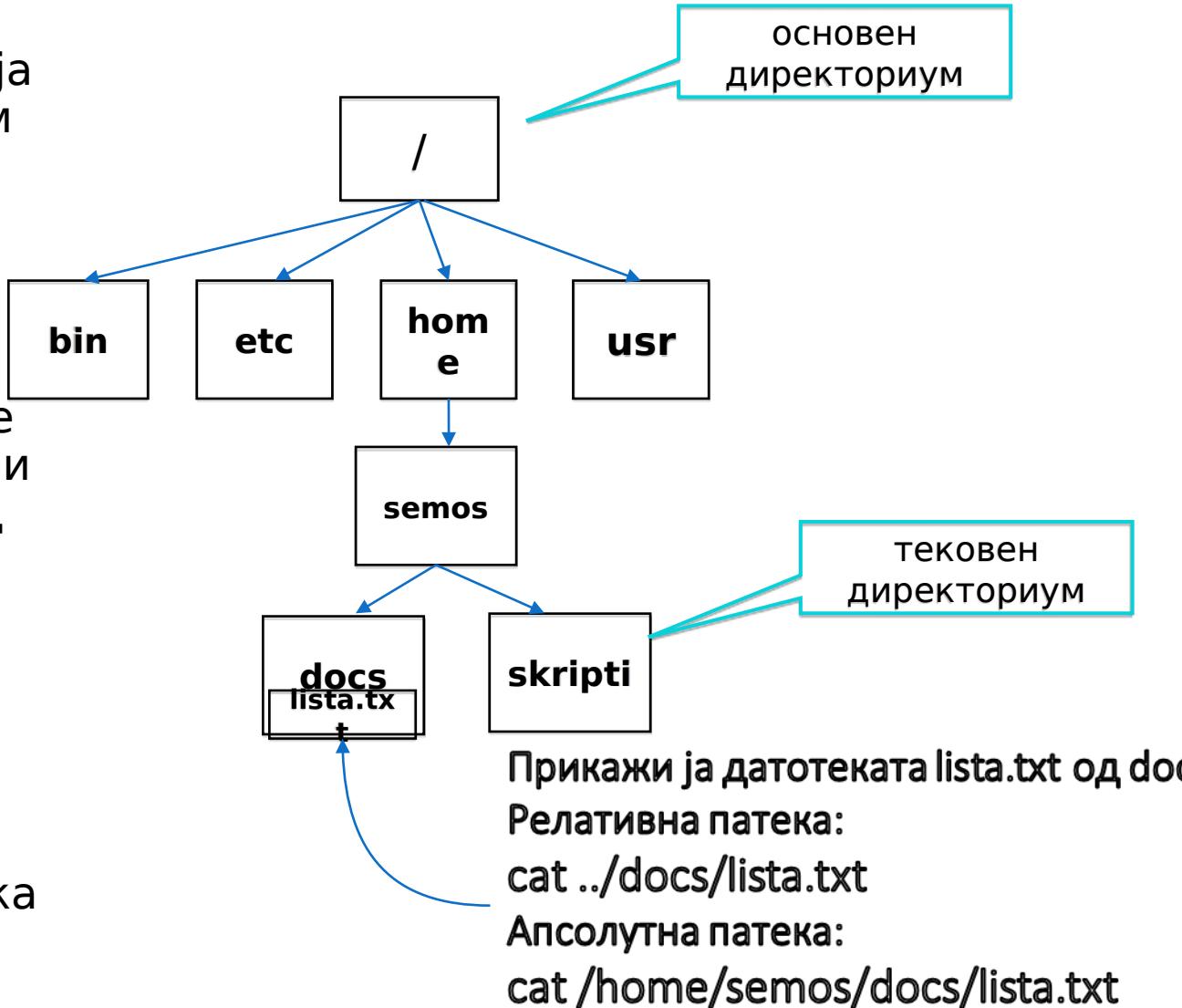
/.../sh	Bourne shell
/.../csh	C shell
/.../tcsh	TC shell
/.../ksh	Korn shell
/.../bash	Bourne Again SHell

# Filesystem структура



# Релативна и апсолутна патека

- **Патека** е единствена локација до датотека или директориум во датотечен систем на оперативен систем која содржи листа од имиња на директориуми на патеката
- **Апсолутна патека** е дефинирана како наведување на локацијата на датотека или директориум започнувајќи од основниот директориум (/).
- **Релативна патека** е дефинирана како патека поврзана со тековниот работен директориум
- **Поими:**
- **директориум** – фолдер, папка
- **датотека** - фајл



# Навигација низ директориуми

`cd directory`

- тековен директориум
- .. - претходен директориум

**Пример:** Премини во твојот директориум

`cd ~`

`cd`

**Пример:** Премини во претходниот директориум

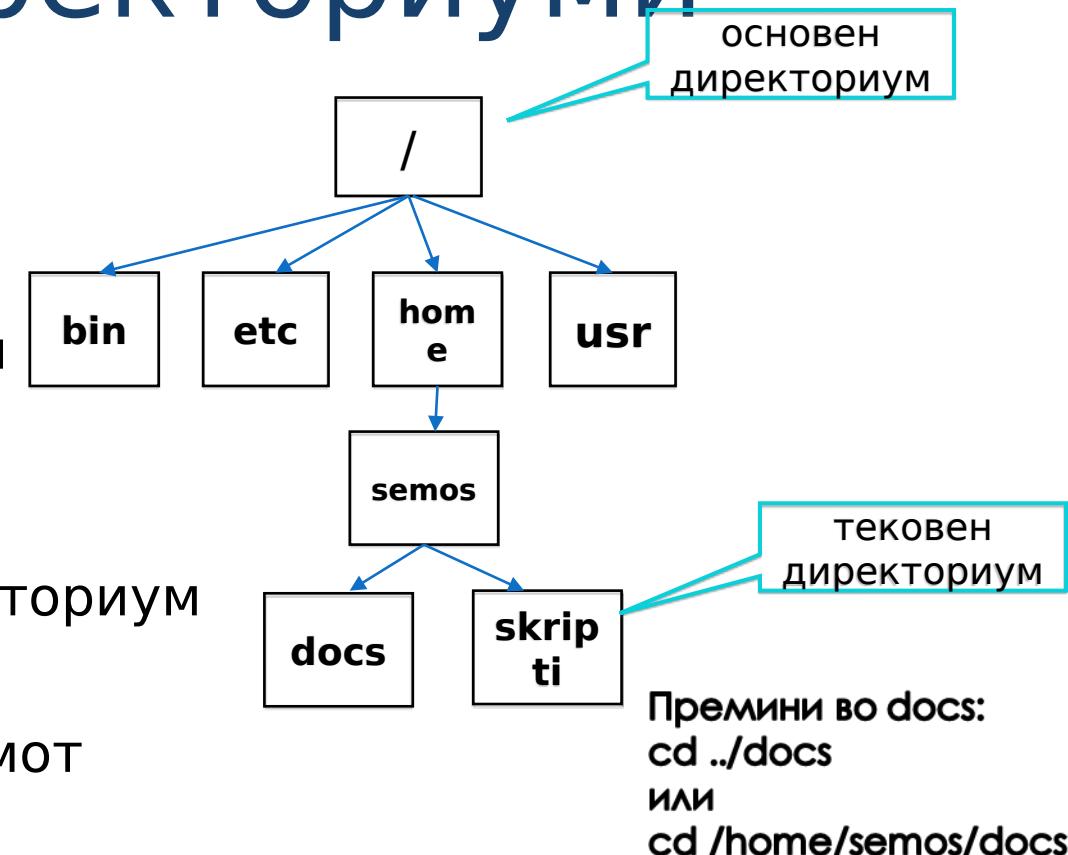
`cd ..`

**Пример:** Премини во / (root) директориумот

`cd /`

Командите за работа со директориуми и датотеки може да се задаваат со релативна и со апсолутна патека

`pwd` - прикажува кој е работниот директориум



# Команда ls

- **ls** се користи за листање на содржина на директориум:
  - **ls -l** - long listing
  - **ls -a** листање и на скриени датотеки кои започнуваат со “.”
  - **ls -F** на имињата на директориумите им се додава знакот /. На извршните датотеки им се додава \*
  - **ls -ls** сортирај по големина
  - **ls -lt** сортирај по време

```
test@test-VirtualBox:~$ ls
Documents Downloads Music p1 Pictures Public Templates Videos
test@test-VirtualBox:~$ ls -l
total 32
drwxr-xr-x 2 test test 4096 noe 16 14:12 Documents
drwxr-xr-x 2 test test 4096 noe 16 14:12 Downloads
drwxr-xr-x 2 test test 4096 noe 16 14:12 Music
drwxrwxr-x 2 test test 4096 noe 19 16:34 p1
drwxr-xr-x 2 test test 4096 noe 19 14:22 Pictures
drwxr-xr-x 2 test test 4096 noe 16 14:12 Public
drwxr-xr-x 2 test test 4096 noe 16 14:12 Templates
drwxr-xr-x 2 test test 4096 noe 16 14:12 Videos
test@test-VirtualBox:~$
```

# Креирање и манипулација на директориуми и датотеки

Директориум се креира со:

**`mkdir`** *directory*

На пример:

**`mkdir examples`**

креира директориум под тековниот, а:

**`mkdir ../new`**

креира директориум на исто ниво со тековниот.

# Креирање и манипулација на директориуми и датотеки

Командата **cp** се користи за копирање на датотеки:

**cp [-option] source destination**

На пример:

**cp manual doc**

ќе ја ископира датотеката **manual** во директориумот **doc**, ако тој директориум постои:

**cp manual manual-copy**

ќе ја ископира датотеката **manual** во друга датотека со име **manual-copy**.

# Креирање и манипулација на директориуми и датотеки

**rm** *file*

се користи за бришење на датотеки.

За бришење на директориуми се користи:

**rmdir** *directory*

За тоа да биде можно, директориумот треба да биде празен.

За бришење на директориум рекурзивно:

**rm -r** *directory*

# Креирање и манипулација на директориуми и датотеки

**mv** преименува датотеки и директориуми:

**mv [-option]** *file1 file2*

**mv [-option]** *directory1 directory2*

**mv [-option]** *file directory*

Ако новата локација веќе постои, таа ќе биде пребришана.

**Примери:**

**mv ex.c c\_prog/lab1.c**

**mv plan ..../january/plan1**

# Креирање и манипулација на директориуми и датотеки

## Прикажување содржина на датотека на екран - **cat, head, tail**

**cat** прикажува содржина на датотека на екран:

**Пример:**

```
cat ex.c
```

**head** прикажува први n линии од датотека на екран:

**Пример:**

```
head 5 ex.c
```

**tail** прикажува последни n линии од датотека на екран:

**Пример:**

```
tail 5 ex.c
```

## Промена на тековно време на датотека

Со **touch** се врши поставување на тековно системско време и датум на постоечка датотека. Ако зададената датотека не постои, тогаш креира празна датотека (со големина 0)

**touch [-option] file**

**Пример:**

```
touch ex.txt
```