



НПГ по КТС – ГРАД ПРАВЕЦ

Курсова работа

ПО

ОПЕРАЦИОННИ СИСТЕМИ

„Командния интерфейс на Linux“

Ученик:
Здравко Анев

Преподавател:
Памела Николова

02.01.2026

Курсова работа

Описание на Linux и използваната Linux дистрибуция – Kali Linux

1. Операционна система Linux

2. Linux дистрибуции

3. Kali Linux – използваната дистрибуция

4. Характеристики и предимства на Kali Linux

ЗАДАЧА 1 - СЪЗДАВАНЕ НА ФАЙЛОВЕ И ДИРЕКТОРИИ

ЗАДАЧА 2 - СЪЗДАВАНЕ НА ФАЙЛОВЕ И ДИРЕКТОРИИ

ЗАДАЧА 3 - КОПИРАНЕ И ПРЕМЕСТВАНЕ НА ФАЙЛОВЕ

ЗАДАЧА 4 - ПОТРЕБИТЕЛИ, ГРУПИ И ПРАВА

ЗАДАЧА 5 - ОБРАБОТКА НА ТЕКСТ

ЗАДАЧА 6 - ТЪРСЕНЕ НА ФАЙЛОВЕ

ЗАДАЧА 7 - РЕГУЛЯРНИ ИЗРАЗИ

ЗАДАЧА 8 - УПРАВЛЕНИЕ НА ПРОЦЕСИТЕ

ЗАДАЧА 9 - ПАКЕТНИ МЕНИДЖЪРИ

ЗАДАЧА 10 - РАБОТА С АРХИВИ

Описание на Linux и използваната Linux дистрибуция – Kali Linux

1. Операционна система Linux

Linux е свободна и отворена операционна система, създадена като алтернатива на комерсиалните операционни системи. Ядрото Linux е разработено през 1991 г. от Линус Торвалдс и оттогава се развива с помощта на глобална общност от програмисти. Основното предимство на Linux е неговият **отворен код**, което означава, че всеки може да го използва, изучава, променя и разпространява.

Linux е базиран на **Unix архитектура**, което го прави изключително стабилен, сигурен и подходящ за многопотребителска среда. Той поддържа едновременна работа на множество потребители и процеси, като осигурява ефективно управление на хардуерните ресурси – процесор, памет, дисково пространство и мрежови интерфейси.

Основни характеристики на Linux:

- Висока **стабилност и надеждност**, особено при продължителна работа;

- Отлична **сигурност**, благодарение на системата за права и потребители;
- **Гъвкавост и модулност** – може да се използва както на сървъри и персонални компютри, така и на вградени системи;
- Богат набор от **дистрибуции**, адаптирани за различни нужди – домашна употреба, сървъри, програмиране, образование и киберсигурност.

Linux се използва масово при уеб сървъри, облачни платформи, суперкомпютри, мрежово оборудване и мобилни устройства (Android е базиран на Linux). Това го прави една от най-широко използваните операционни системи в света.

2. Linux дистрибуции

Linux не се разпространява като една единствена операционна система, а под формата на **дистрибуции**. Дистрибуцията включва Linux ядро, системни библиотеки, пакетен мениджър, графична среда и набор от предварително инсталирани приложения. Различните дистрибуции са оптимизирани за конкретни цели – например Ubuntu за крайни потребители, CentOS и Debian за сървъри, Arch Linux за напреднали потребители и Kali Linux за информационна сигурност.

3. Kali Linux – използваната дистрибуция

Kali Linux е специализирана Linux дистрибуция, предназначена за **киберсигурност, penetration testing (тестове за проникване), етично хакерство и дигитална криминалистика**. Тя се разработва и поддържа от екипа на Offensive Security и е базирана на стабилната платформа Debian.

Основната цел на Kali Linux е да предостави готова среда с предварително инсталирани инструменти за:

- анализ на мрежи;
- откриване и експлоатиране на уязвимости;
- тестване на безжични мрежи;
- анализ на пароли и криптографски алгоритми;
- дигитална криминалистика и анализ на данни.

Kali Linux включва **стотици инструменти**, групирани по категории, като:

- Information Gathering (събиране на информация);
- Vulnerability Analysis (анализ на уязвимости);
- Wireless Attacks (атаки върху Wi-Fi мрежи);
- Web Application Testing;
- Password Attacks;
- Exploitation Tools и Forensics.

4. Характеристики и предимства на Kali Linux

Kali Linux е проектиран за професионална употреба и обучение в областта на информационната сигурност. Сред основните му характеристики са:

- Поддръжка на **Live режим**, който позволява стартиране без инсталация;
- Възможност за работа на **виртуални машини** (VMware, VirtualBox)
- Отлична **хардуерна поддръжка**, включително за безжични адаптери;
- Чести актуализации и активна поддръжка;
- Подходящ както за обучение, така и за реални тестове по сигурността.

За разлика от стандартните Linux дистрибуции, Kali Linux **не е предназначен за ежедневна домашна употреба**, а за специфични задачи, свързани със защита и анализ на компютърни системи. Поради мощните си инструменти, той се използва основно от администратори, специалисти по киберсигурност и студенти в IT направления.

ЗАДАЧА 1 - СЪЗДАВАНЕ НА ФАЙЛОВЕ И ДИРЕКТОРИИ

Създайте файловете и директориите от фигурата по-долу във вашата домашна директория. След като създадете всичко изведете файловата структура в терминала и се върнете във вашата домашна директория. Изтрийте директорията 'Cluj' при четен номер или директорията "Iasi" при нечетен номер.

```

(zdravko@zdravko)~/OS_COURSE_WORK
$ mkdir -p Romania/Bucharest/{Sector1,Sector6} Romania/Cluj/{Cluj-Napoca,Turda} Romania/Constanta/{Mangalia,Medgidia} Romania/Iasi/{Pascani,Targu_Frumos}

(zdravko@zdravko)~/OS_COURSE_WORK
$ touch Romania/Bucharest/Sector1/residents.docx

(zdravko@zdravko)~/OS_COURSE_WORK
$ touch Romania/Bucharest/Sector6/budget.xls

(zdravko@zdravko)~/OS_COURSE_WORK
$ touch Romania/Iasi/Pascani/schools.docx

(zdravko@zdravko)~/OS_COURSE_WORK
$ touch Romania/Constanta/Mangalia/tourism.info

(zdravko@zdravko)~/OS_COURSE_WORK
$ touch Romania/Cluj/Cluj-Napoca/map.pdf

(zdravko@zdravko)~/OS_COURSE_WORK
$ tree Romania
Romania
├── Bucharest
│   ├── Sector1
│   │   └── residents.docx
│   └── Sector6
│       └── budget.xls
├── Cluj
│   ├── Cluj-Napoca
│   │   └── map.pdf
│   └── Turda
├── Constanta
│   ├── Mangalia
│   │   └── tourism.info
│   └── Medgidia
├── Iasi
│   ├── Pascani
│   │   └── schools.docx
│   └── Targu_Frumos
└── 13 directories, 5 files

(zdravko@zdravko)~/OS_COURSE_WORK
$ cd ~

(zdravko@zdravko)~
$ rm -r OS_COURSE_WORK/Romania/Iasi

(zdravko@zdravko)~
$ tree OS_COURSE_WORK/Romania
OS_COURSE_WORK/Romania
├── Bucharest
│   ├── Sector1
│   │   └── residents.docx
│   └── Sector6
│       └── budget.xls
├── Cluj
│   ├── Cluj-Napoca
│   │   └── map.pdf
│   └── Turda
├── Constanta
│   ├── Mangalia
│   │   └── tourism.info
│   └── Medgidia
└── 10 directories, 4 files

(zdravko@zdravko)~
$ █

```

<code>mkdir -p</code>	създава директории и поддиректории наведнъж
<code>{ }</code>	позволява създаване на няколко директории с една команда
<code>~</code>	символ за домашната директория на потребителя
<code>rm -r</code>	изтрива директория заедно с цялото ѝ съдържание
<code>tree</code>	показва структурата на директориите
<code>touch</code>	създава празни файлове.
<code>cd ~</code>	връща потребителя в домашната директория

ЗАДАЧА 2 - СЪЗДАВАНЕ НА ФАЙЛОВЕ И ДИРЕКТОРИИ

Създайте файловете и директориите от фигурата по-долу във вашата домашна директория. След като създадете всичко изведете файловата структура в терминала и се върнете във вашата домашна директория. Изтрийте директорията „Archives“ за четен номер и „Spreadsheets“ за нечетен номер.

```

(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ mkdir -p ProjectFiles/{Documents,TextFiles,Spreadsheets,Images,Archives}

(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ touch ProjectFiles/Documents/{report1.docx,report2.docx,report3.docx,summary1.docx,summary2.docx}

(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ touch ProjectFiles/TextFiles/{note1.txt,note2.txt,note3.txt,note4.txt,note5.txt}

(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ touch ProjectFiles/Spreadsheets/{data1.xlsx,data2.xlsx,data3.xlsx}

(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ touch ProjectFiles/Images/{logo.png,digram.jpg,chart.png}

(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ touch ProjectFiles/Archives/{backup1.zip,backup2.zip,backup3.zip}

(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ tree ProjectFiles
ProjectFiles
├── Archives
│   ├── backup1.zip
│   ├── backup2.zip
│   └── backup3.zip
├── Documents
│   ├── report1.docx
│   ├── report2.docx
│   ├── report3.docx
│   ├── summary1.docx
│   └── summary2.docx
├── Images
│   ├── chart.png
│   ├── digram.jpg
│   └── logo.png
├── Spreadsheets
│   ├── data1.xlsx
│   ├── data2.xlsx
│   └── data3.xlsx
└── TextFiles
    ├── note1.txt
    ├── note2.txt
    ├── note3.txt
    ├── note4.txt
    └── note5.txt

6 directories, 19 files

(zdravko@zdravko)-[~/OS_COURSE_WORK]
$

```



```

(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ cd ~

(zdravko@zdravko)-[~]
$ rm -r OS_COURSE_WORK/ProjectFiles/Spreadsheets

(zdravko@zdravko)-[~]
$ tree OS_COURSE_WORK/ProjectFiles
OS_COURSE_WORK/ProjectFiles
├── Archives
│   ├── backup1.zip
│   ├── backup2.zip
│   └── backup3.zip
├── Documents
│   ├── report1.docx
│   ├── report2.docx
│   ├── report3.docx
│   ├── summary1.docx
│   └── summary2.docx
├── Images
│   ├── chart.png
│   ├── diagram.jpg
│   └── logo.png
└── TextFiles
    ├── note1.txt
    ├── note2.txt
    ├── note3.txt
    ├── note4.txt
    └── note5.txt

5 directories, 16 files

(zdravko@zdravko)-[~]
$ 

```

ЗАДАЧА 3 - КОПИРАНЕ И ПРЕМЕСТВАНЕ НА ФАЙЛОВЕ

Вариант за нечетен номер

- Копирайте всички .txt файлове в директорията FOLDER3

- Преместете всички .pdf файлове в директорията FOLDER1
- Преместете всички файлове, които започват с името strypes, в директорията FOLDER2
- Копирайте всички .docx файлове в директорията FOLDER4
- Изтрийте всички файлове, които имат три символа като разширение в директорията FOLDER3
- Направете FOLDER1 скрита директория

```
(zdravko@zdravko)~$ cd OS_COURSE_WORK
(zdravko@zdravko)~[/OS_COURSE_WORK]$ mkdir -p MAIN_DIR/{FOLDER1,FOLDER2,FOLDER3,FOLDER4}
(zdravko@zdravko)~[/OS_COURSE_WORK]$ touch MAIN_DIR/FOLDER1/{readme.txt,report.pdf,company_data.docx}
(zdravko@zdravko)~[/OS_COURSE_WORK]$ touch MAIN_DIR/FOLDER2/{strypes_summary.txt,example.docx,finance.pdf}
(zdravko@zdravko)~[/OS_COURSE_WORK]$ touch MAIN_DIR/FOLDER4/{info.txt,strypes_plan.pdf,slides.pptx}
(zdravko@zdravko)~[/OS_COURSE_WORK]$ tree MAIN_DIR
MAIN_DIR
├── FOLDER1
│   ├── company_data.docx
│   ├── readme.txt
│   └── report.pdf
├── FOLDER2
│   ├── example.docx
│   ├── finance.pdf
│   └── strypes_summary.txt
├── FOLDER3
└── FOLDER4
    ├── info.txt
    ├── slides.pptx
    └── strypes_plan.pdf

5 directories, 9 files
(zdravko@zdravko)~[/OS_COURSE_WORK]$ cp MAIN_DIR/*.txt MAIN_DIR/FOLDER3/
```

01:

```
MAIN_DIR
├── FOLDER1
│   ├── company_data.docx
│   ├── readme.txt
│   └── report.pdf
├── FOLDER2
│   ├── example.docx
│   ├── finance.pdf
│   └── strypes_summary.txt
├── FOLDER3
│   ├── info.txt
│   ├── readme.txt
│   └── strypes_summary.txt
└── FOLDER4
    ├── info.txt
    ├── slides.pptx
    └── strypes_plan.pdf
```

02:

```
(zdravko@zdravko)~[~/OS_COURSE_WORK]
$ mv MAIN_DIR/**/*.pdf MAIN_DIR/FOLDER1/

mv: 'MAIN_DIR/FOLDER1/report.pdf' and 'MAIN_DIR/FOLDER1/report.pdf' are the same file

(zdravko@zdravko)~[~/OS_COURSE_WORK]
$ tree MAIN_DIR
MAIN_DIR
├── FOLDER1
│   ├── company_data.docx
│   ├── finance.pdf
│   ├── readme.txt
│   ├── report.pdf
│   └── strypes_plan.pdf
├── FOLDER2
│   ├── example.docx
│   └── strypes_summary.txt
├── FOLDER3
│   ├── info.txt
│   ├── readme.txt
│   └── strypes_summary.txt
└── FOLDER4
    ├── info.txt
    └── slides.pptx

5 directories, 12 files

(zdravko@zdravko)~[~/OS_COURSE_WORK]
$
```

```
(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ cp MAIN_DIR/**/*.docx MAIN_DIR/FOLDER4/
```

```
(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ tree MAIN_DIR
```

```
MAIN_DIR
├── FOLDER1
│   ├── company_data.docx
│   ├── finance.pdf
│   ├── readme.txt
│   └── report.pdf
├── FOLDER2
│   ├── example.docx
│   ├── strypes_plan.pdf
│   └── strypes_summary.txt
├── FOLDER3
│   ├── info.txt
│   └── readme.txt
└── FOLDER4
    ├── company_data.docx
    ├── example.docx
    ├── info.txt
    └── slides.pptx
```

```
5 directories, 13 files
```

03:

```
(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ rm MAIN_DIR/FOLDER3/*.???
```

```
(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ tree MAIN_DIR
```

```
MAIN_DIR
├── FOLDER1
│   ├── company_data.docx
│   ├── finance.pdf
│   ├── readme.txt
│   └── report.pdf
├── FOLDER2
│   ├── example.docx
│   ├── strypes_plan.pdf
│   └── strypes_summary.txt
├── FOLDER3
└── FOLDER4
    ├── company_data.docx
    ├── example.docx
    ├── info.txt
    └── slides.pptx
```

04: 5 directories, 11 files

```
(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ mv MAIN_DIR/FOLDER1 MAIN_DIR/.FOLDER1

(zdravko@zdravko)-[~/OS_COURSE_WORK]
$ tree MAIN_DIR
MAIN_DIR
├── FOLDER2
│   ├── example.docx
│   ├── strypes_plan.pdf
│   └── strypes_summary.txt
├── FOLDER3
└── FOLDER4
    ├── company_data.docx
    ├── example.docx
    ├── info.txt
    └── slides.pptx

4 directories, 7 files
```

05:

<code>cp MAIN_DIR/*/*.txt MAIN_DIR/FOLDER3/</code>	Копира всички файлове с разширение .txt от поддиректориите в FOLDER3.
<code>mv MAIN_DIR/*/*.pdf MAIN_DIR/FOLDER1/</code>	Премества всички файлове с разширение .pdf в директория FOLDER1.
<code>mv MAIN_DIR/*/strypes* MAIN_DIR/FOLDER2/</code>	Премества всички файлове, чиито имена започват със strypes, в директория FOLDER2.
<code>cp MAIN_DIR/*/*.docx MAIN_DIR/FOLDER4/</code>	Копира всички файлове с разширение .docx в директория FOLDER4.
<code>rm MAIN_DIR/FOLDER3/*.*.*</code>	Изтрива всички файлове в FOLDER3, които имат разширение от точно три символа.
<code>mv MAIN_DIR/FOLDER1 MAIN_DIR/.FOLDER1</code>	Преименува FOLDER1 в .FOLDER1, като по този начин я прави скрита директория.

ЗАДАЧА 4 - ПОТРЕБИТЕЛИ, ГРУПИ И ПРАВА

Вариант за нечетен номер

Задача: Управление на потребители, групи и файлови права в Linux

1. Създайте потребител с име `developer` и групи с имена `devops` и `support`.
2. Добавете потребителя `developer` към групите `devops` и `support`.
3. Копирайте файла `/etc/group` във вашата домашна директория.
4. Променете правата на копирания файл на `rwxt-x---`.
 - Използвайте както буквен, така и цифров формат (например `chmod u=rwx,g=rx,o=---` и `chmod 750`).
5. Направете файла собственост на потребителя `developer` и групата `devops`.
6. Проверете дали промените са коректни, като използвате `ls -l` и `id developer`.

Изисквания:

- Уверете се, че всички файлове и директории се създават с правилни права и

Собственост.

```

(zdravko@zdravko)~/OS_COURSE_WORK
$ sudo groupadd devops
[sudo] password for zdravko:

(zdravko@zdravko)~/OS_COURSE_WORK
$ sudo groupadd support

(zdravko@zdravko)~/OS_COURSE_WORK
$ sudo useradd -m developer

(zdravko@zdravko)~/OS_COURSE_WORK
$ sudo usermod -aG devops,support developer

(zdravko@zdravko)~/OS_COURSE_WORK
$ cp /etc/group ~/

(zdravko@zdravko)~/OS_COURSE_WORK
$ chmod u=rwx,g=rX,o= ~/group

(zdravko@zdravko)~/OS_COURSE_WORK
$ chmod 750 ~/group

(zdravko@zdravko)~/OS_COURSE_WORK
$ sudo chown developer:devops ~/group

(zdravko@zdravko)~/OS_COURSE_WORK
$ ls -l ~/group
-rwxr-x--- 1 developer devops 1465 Dec 29 11:11 /home/zdravko/group

(zdravko@zdravko)~/OS_COURSE_WORK
$ id developer
uid=1003(developer) gid=1006(developer) groups=1006(developer),1004(devops),1005(support)

```

<code>sudo groupadd devops</code>	Създава група с име devops .
<code>sudo groupadd support</code>	Създава група с име support .
<code>sudo useradd -m developer</code>	Създава потребител developer и му създава домашна директория.
<code>sudo usermod -aG devops,support developer</code>	Добавя потребителя developer към групите devops и support .

<code>cp /etc/group ~/</code>	Копира файла <code>/etc/group</code> в домашната директория на текущия потребител.
<code>chmod u=rwx,g=rx,o=--- ~/group</code>	Задава права на файла чрез буквен формат.
<code>chmod 750 ~/group</code>	Задава същите права чрез цифров формат.
<code>sudo chown developer:devops ~/group</code>	Променя собствеността на файла на потребител developer и група devops .
<code>ls -l ~/group</code>	Проверява правата, собственика и групата на файла.
<code>id developer</code>	Проверява към кои групи принадлежи потребителят developer .

ЗАДАЧА 5 - ОБРАБОТКА НА ТЕКСТ

Вариант за нечетен номер

Създайте файл с име `fruits.info`, който съдържа следните данни:

Формат: ПЛОД;КОЛИЧЕСТВО;ЦЕНА

Извършете следните операции:

1. Изведете в стандартния изход само имената на плодовете, сортирани по азбучен ред.
2. Изведете в стандартния изход цялото съдържание на файла, сортирано по количество
3. във възходящ ред.
4. Запишете името на плода с най-висока цена в нов файл с име `expensive_fruit`.
5. Изведете първите три реда от файла, сортирани по цена във възходящ ред.

```
(zdravko@zdravko) - [~/OS_COURSE_WORK]
$ cat << EOF > fruits.info
```

```
Ябълка;70;1.8
Банан;90;2.1
Круша;50;2.4
Праскова;60;3.0
Грозде;80;2.9
EOF
```

```
(zdravko@zdravko) - [~/OS_COURSE_WORK]
$ nano fruits.info
```

```
(zdravko@zdravko) - [~/OS_COURSE_WORK]
$ cut -d';' -f1 fruits.info | sort
```

```
Банан
Грозде
Круша
Праскова
Ябълка
```

```
(zdravko@zdravko) - [~/OS_COURSE_WORK]
$ sort -t';' -k2,2n fruits.info
```

```
Круша;50;2.4
Праскова;60;3.0
Ябълка;70;1.8
Грозде;80;2.9
Банан;90;2.1
```

```
(zdravko@zdravko) - [~/OS_COURSE_WORK]
$ sort -t';' -k3,3n fruits.info | tail -n 1 | cut -d';' -f1 > expensive_fruit
```

```
(zdravko@zdravko) - [~/OS_COURSE_WORK]
$ sort -t';' -k3,3n fruits.info | head -n 3
```

```
Ябълка;70;1.8
Банан;90;2.1
Круша;50;2.4
```

```
(zdravko@zdravko) - [~/OS_COURSE_WORK]
$ cat expensive_fruit
```

```
Праскова
```

```
(zdravko@zdravko) - [~/OS_COURSE_WORK]
$
```

<code>cat << EOF > fruits.info</code>	Създава файл <code>fruits.info</code> и позволява въвеждане на съдържание от стандартния вход.
<code>cut -d';' -f1 fruits.info sort</code>	Извежда само имената на плодовете и ги сортира по азбучен ред.
<code>sort -t';' -k2,2n fruits.info</code>	Извежда цялото съдържание, сортирано по количество във възходящ ред.
<code>sort -t';' -k3,3n fruits.info tail -n 1 cut -d';' -f1 > expensive_fruit</code>	Записва името на плода с най-висока цена във файл <code>expensive_fruit</code> .
<code>sort -t';' -k3,3n fruits.info head -n 3</code>	Извежда първите три реда, сортирани по цена във възходящ ред.

NOTE
<p><code>-d';'</code> задава разделител ;</p> <p><code>-k</code> определя поле за сортиране</p> <p><code>n</code> указва числово сортиране</p>

ЗАДАЧА 6 - ТЪРСЕНЕ НА ФАЙЛОВЕ

Вариант за нечетен номер

Използвайте командата `find`, за да изпълните следните задачи във вашата домашна директория и всички нейни поддиректории:

1. Изведете броя на всички директории, чиито имена съдържат думата “`dir`”.
2. Изведете броя на всички файлове, чиито имена завършват на “`.conf`”.
3. Намерете и изведете информация за най-големия `.pdf` файл.

4. Изведете броя на всички .txt файлове, които в името си съдържат думата “notes”.
5. Изведете пълния път до всички файлове с разширение .html, които са по-нови от 7 дни.
6. Изведете информация за всички файлове, които имат разширение от точно 3 символа (например .txt, .pdf, .sh).

```
(zdravko@zdravko)-[~]
$ find ~ -type d -iname '*dir*' | wc -l
74

(zdravko@zdravko)-[~]
$ find ~ -type f -name '*.conf' | wc -l
6

(zdravko@zdravko)-[~]
$ find ~ -type f -name '*.pdf' -printf '%s %p\n' | sort -n | tail -1
250870 /home/zdravko/Downloads/pycharm-2025.2.3/help/ReferenceCardForMac.pdf

(zdravko@zdravko)-[~]
$ find ~ -type f -iname '*notes*.txt' | wc -l
0

(zdravko@zdravko)-[~]
$ find ~ -type f -name '*.html' -mtime -7 | sed -n '1p;$p' | { read x 86 echo "$x" || echo "none"; }
none

(zdravko@zdravko)-[~]
$
```

```
(zdravko@zdravko)~$ find ~ -type f -regextype posix-extended -regex '.*[^\./]{3}$' -print0 \
| { IFS= read -r -d '' _ 66 find ~ -type f -regextype posix-extended -regex '.*[^\./]{3}$' -exec ls -lh {} + || echo "none"; }

-rw-rw-r-- 1 zdravko zdravko 131 Oct 16 15:30 /home/zdravko/cookies.txt
-rw-rw-r-- 1 zdravko zdravko 20 Nov 13 09:59 /home/zdravko/example.txt
-rw-rw-r-- 1 zdravko zdravko 69 Nov 13 11:18 /home/zdravko/file1.txt
-rw-rw-r-- 1 zdravko zdravko 0 Oct 15 19:12 /home/zdravko/found_flags.txt
-rw-rw-r-- 1 zdravko zdravko 124 Nov 13 11:05 /home/zdravko/fruits.txt
-rw-rw-r-- 1 zdravko zdravko 4.2K Dec 30 11:11 /home/zdravko/.mozilla/firefox/78gfb5yt.default-esr/addonStartup.json.lz4
-rw-rw-r-- 1 zdravko zdravko 912K Dec 30 11:26 /home/zdravko/.mozilla/firefox/78gfb5yt.default-esr/AlternateServices.bin
-rw-rw-r-- 1 zdravko zdravko 171 Dec 30 11:11 /home/zdravko/.mozilla/firefox/78gfb5yt.default-esr/compatibility.ini
-rw-rw-r-- 1 zdravko zdravko 1.3K Dec 30 11:25 /home/zdravko/.mozilla/firefox/78gfb5yt.default-esr/datareporting/glean/db/data.safe.bin
-rw-rw-r-- 1 zdravko zdravko 566 Oct 16 06:49 /home/zdravko/.mozilla/firefox/78gfb5yt.default-esr/pkcs11.txt
-rw-rw-r-- 1 zdravko zdravko 3.2M Dec 29 08:06 /home/zdravko/.mozilla/firefox/78gfb5yt.default-esr/security_state/data.safe.bin
-rw-rw-r-- 1 zdravko zdravko 3.9K Dec 29 08:02 /home/zdravko/.mozilla/firefox/78gfb5yt.default-esr/serviceworker.txt
-rw-rw-r-- 1 zdravko zdravko 1.2K Dec 30 10:54 /home/zdravko/.mozilla/firefox/78gfb5yt.default-esr/settings/data.safe.bin
-rw-rw-r-- 1 zdravko zdravko 33K Dec 30 11:12 /home/zdravko/.mozilla/firefox/78gfb5yt.default-esr/WebsiteSecurityServiceState.bin
-rw-rw-r-- 1 zdravko zdravko 28K Oct 15 19:29 /home/zdravko/.mozilla/firefox/78gfb5yt.default-esr/weave/logs/error-sync-1760556555997.txt
-rw-rw-r-- 1 zdravko zdravko 0 Dec 30 11:11 /home/zdravko/.mozilla/firefox-esr/Crash Reports/crash_helper_server.log
-rw-rw-r-- 1 zdravko zdravko 58 Oct 15 18:46 /home/zdravko/.mozilla/firefox/installs.ini
-rw-rw-r-- 1 zdravko zdravko 247 Oct 15 18:46 /home/zdravko/.mozilla/firefox/profiles.ini
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:35 /home/zdravko/OS_COURSE_WORK/MAIN_DIR/.FOLDER1/finance.pdf
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:35 /home/zdravko/OS_COURSE_WORK/MAIN_DIR/.FOLDER1/readme.txt
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:35 /home/zdravko/OS_COURSE_WORK/MAIN_DIR/.FOLDER1/report.pdf
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:35 /home/zdravko/OS_COURSE_WORK/MAIN_DIR/FOLDER2/strypes_plan.pdf
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:35 /home/zdravko/OS_COURSE_WORK/MAIN_DIR/FOLDER2/strypes_summary.txt
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:35 /home/zdravko/OS_COURSE_WORK/MAIN_DIR/FOLDER4/info.txt
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:09 /home/zdravko/OS_COURSE_WORK/ProjectFiles/Archives/backup1.zip
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:09 /home/zdravko/OS_COURSE_WORK/ProjectFiles/Archives/backup2.zip
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:09 /home/zdravko/OS_COURSE_WORK/ProjectFiles/Archives/backup3.zip
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:09 /home/zdravko/OS_COURSE_WORK/ProjectFiles/Images/chart.png
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:09 /home/zdravko/OS_COURSE_WORK/ProjectFiles/Images/diagram.jpg
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:09 /home/zdravko/OS_COURSE_WORK/ProjectFiles/Images/logo.png
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:08 /home/zdravko/OS_COURSE_WORK/ProjectFiles/TextFiles/note1.txt
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:08 /home/zdravko/OS_COURSE_WORK/ProjectFiles/TextFiles/note2.txt
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:08 /home/zdravko/OS_COURSE_WORK/ProjectFiles/TextFiles/note3.txt
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:08 /home/zdravko/OS_COURSE_WORK/ProjectFiles/TextFiles/note4.txt
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 08:08 /home/zdravko/OS_COURSE_WORK/ProjectFiles/TextFiles/note5.txt
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 07:43 /home/zdravko/OS_COURSE_WORK/Romania/Bucharest/Sector6/budget.xls
-rw-rw-r-- 1 zdravko zdravko 0 Dec 29 07:44 /home/zdravko/OS_COURSE_WORK/Romania/Cluj/Cluj-Napoca/map.pdf
-rw-rw-r-- 1 zdravko zdravko 427 Oct 16 11:34 /home/zdravko/PyCharmMiscProject/.idea/inspectionProfiles/Project_Default.xml
-rw-rw-r-- 1 zdravko zdravko 320 Oct 16 11:27 /home/zdravko/PyCharmMiscProject/.idea/misc.xml
-rw-rw-r-- 1 zdravko zdravko 276 Oct 16 11:27 /home/zdravko/PyCharmMiscProject/.idea/modules.xml
-rw-rw-r-- 1 zdravko zdravko 5.8K Oct 16 19:28 /home/zdravko/PyCharmMiscProject/.idea/workspace.xml
-rw-rw-r-- 1 zdravko zdravko 1.5K Oct 16 11:08 /home/zdravko/PyCharmMiscProject/.venv/bin/activate.csh
-rw-rw-r-- 1 zdravko zdravko 1.7K Oct 16 11:08 /home/zdravko/PyCharmMiscProject/.venv/bin/activate.ps1
-rw-rw-r-- 1 zdravko zdravko 100K Oct 16 11:52 /home/zdravko/PyCharmMiscProject/.venv/bin/_pycache_/readelf.cpython-313.pyc
-rw-rw-r-- 1 zdravko zdravko 7 Oct 16 11:52 /home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/bcrypt-5.0.0.dist-info/top_level.txt
-rw-rw-r-- 1 zdravko zdravko 75 Oct 16 11:52 /home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/cffi-2.0.0.dist-info/entry_points.txt
-rw-rw-r-- 1 zdravko zdravko 19 Oct 16 11:52 /home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/cffi-2.0.0.dist-info/top_level.txt
-rw-rw-r-- 1 zdravko zdravko 2.4K Oct 16 11:52 /home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/mako/ext/_pycache_/autohandler.cpython-313.pyc
-rw-rw-r-- 1 zdravko zdravko 2.7K Oct 16 11:52 /home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/mako/ext/_pycache_/babelplugin.cpython-313.pyc
-rw-rw-r-- 1 zdravko zdravko 3.9K Oct 16 11:52 /home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/mako/ext/_pycache_/beaker_cache.cpython-313.pyc
-rw-rw-r-- 1 zdravko zdravko 5.4K Oct 16 11:52 /home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/mako/ext/_pycache_/extract.cpython-313.pyc
-rw-rw-r-- 1 zdravko zdravko 185 Oct 16 11:52 /home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/mako/ext/_pycache_/init_.cpython-313.pyc
-rw-rw-r-- 1 zdravko zdravko 2.7K Oct 16 11:52 /home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/mako/ext/_pycache_/linguaplugin.cpython-313.pyc
-rw-rw-r-- 1 zdravko zdravko 670 Oct 16 11:52 /home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/mako/ext/_pycache_/preprocessors.cpython-313.pyc
-rw-rw-r-- 1 zdravko zdravko 6.0K Oct 16 11:52 /home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/mako/ext/_pycache_/pygmentplugin.cpython-313.pyc
-rw-rw-r-- 1 zdravko zdravko 2.6K Oct 16 11:52 /home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/mako/ext/_pycache_/turbogears.cpython-313.pyc
```

```
find ~ -type d -iname '*dir*' | wc -l
```

find ~ търси в домашната директория и поддиректориите. -type d ограничава търсенето само до директории. -iname прави търсенето нечувствително към главни/малки букви. *dir* означава, че името съдържа „dir“. wc -l брои намерените редове.

```
find ~ -type f -name '*.conf' | wc -l
```

-type f търси само файлове. -name '*.conf' намира файлове, завършващи на .conf. wc -l брои броя им.

```
find ~ -type f -name '*.pdf' -printf '%s %p\n' | sort -n | tail -1
```

-printf '%s %p\n' извежда размера в байтове и пътя. sort -n сортира по числова стойност (размер). tail -1 показва най-големия файл.

<pre>find ~ -type f -iname '*notes*.txt' wc -l</pre>	<p>Търси файлове с разширение .txt, които съдържат думата „notes“ в името. wc -l брой резултатите.</p>
<pre>find ~ -type f -name '*.html' -mtime -7 sed -n '1p;\$p' { read x && echo "\$x" echo "none"; }</pre>	<p>търси в домашната директория (~) само файлове (-type f) с разширение .html (-name '*.html'), които са променени през последните 7 дни (-mtime -7).</p> <p>Частта след тръбите използва sed и read, за да провери дали има поне един резултат, като при наличие извежда намерените пътища, а ако няма нито един файл – показва съобщението none.</p>
<pre>find ~ -type f -regextype posix-extended -regex '.*\.[^./]{3}\$' -print0 \ { read -r -d '' _ && find ~ -type f -regextype posix-extended -regex '.*\.[^./]{3}\$' -exec ls -lh {} + echo "none"; }</pre>	<p>Командата find ~ -type f -regextype posix-extended -regex '.*\.[^./]{3}\$' -print0 търси в домашната директория (~) само файлове (-type f), като използва разширени POSIX регулярни изрази (-regextype posix-extended), за да намери файлове с разширение от точно 3 символа (-regex '.*\.[^./]{3}\$'), и ги извежда в защитен формат с NULL разделител (-print0).</p> <p>Частта след тръбата () проверява дали е намерен поне един файл (read -r -d ""), и ако има — показва подробна информация за тях (ls -lh), а ако няма — извежда съобщението none</p>

ЗАДАЧА 7 - РЕГУЛЯРНИ ИЗРАЗИ

Вариант за нечетен номер

Използвайте командата grep, за да извършите следните операции с файла

/usr/share/dict/words:

1. Изведете броя на думите, които започват с ‘p’ и завършват на ‘s’.
2. Изведете броя на думите, които съдържат точно седем малки латински букви.
3. Изведете броя на думите, които не съдържат точно седем малки латински букви.
4. Изведете броя на думите, които не започват с ‘d’ или ‘t’.
5. Запишете във файла filtered_list всички думи, които започват с главна латинска буква и съдържат специален символ (например „-“ или „““).

```

(zdravko@zdravko)-[~]
$ grep -E '^p.*s$' /usr/share/dict/words | wc -l
3364

(zdravko@zdravko)-[~]
$ grep -E '^[a-z]{7}$' /usr/share/dict/words | wc -l
9978

(zdravko@zdravko)-[~]
$ grep -Ev '^[a-z]{7}$' /usr/share/dict/words | wc -l
94356

(zdravko@zdravko)-[~]
$ grep -Ev '[dt]' /usr/share/dict/words | wc -l
94804

(zdravko@zdravko)-[~]
$ grep -E '[A-Z].*[-'\'' ]' /usr/share/dict/words > filtered_list

(zdravko@zdravko)-[~]
$ cat filtered_list
Aachen's
Aaliyah's
Aaron's
AA's
Abbasid's
Abbas's
Abbott's
Abby's
ABC's
Abdul's
Abelard's
Abel's
Abelson's
Aberdeen's
Abernathy's
Abe's
Abidjan's
Abigail's
Abilene's
ABM's
Abner's
Abraham's
Abram's
Abrams's
AB's
Absalom's
Abuja's
Abyssinian's
Abyssinia's
Acadia's
Acapulco's
Accenture's
Accra's

```

```

grep          -E          '^p.*s$'
/usr/share/dict/words | wc -l

```

^p означава „започва с p“, .* – произволни символи, s\$ – „завършва на s“.

wc -l брой намерените думи.

<code>grep -E '^ [a-z]{7}\$' /usr/share/dict/words wc -l</code>	[a-z]{7} означава точно 7 малки латински букви, а ^ и \$ гарантират, че думата няма други символи.
<code>grep -Ev '^ [a-z]{7}\$' /usr/share/dict/words wc -l</code>	-v обръща резултата – избира всички думи, които не отговарят на шаблона за 7 малки букви.
<code>grep -Ev '^ [dt]' /usr/share/dict/words wc -l</code>	^[dt] означава „започва с d или t“, а -v избира всички останали думи.
<code>grep -E '^ [A-Z].*[-'\"]' /usr/share/dict/words > filtered_list</code>	^[A-Z] – думата започва с главна буква .* – произволни символи [-"] – съдържа тире - или апостроф ' > записва резултата във файла filtered_list.

ЗАДАЧА 8 - УПРАВЛЕНИЕ НА ПРОЦЕСИТЕ

Вариант за нечетен номер

1. Създайте четири процеса sleep във фонов режим — с аргументи 100, 150, 200 и 250 секунди.
2. Изведете списък с всички фонов процеси и състоянието им.
3. Превключете втория процес на foreground режим и след това го прекратете.
4. Стартирайте процеса xterm с приоритет NI = 10.
5. Докато е активен, променете приоритета му на NI = 2.
6. Стартирайте процеса хеуес с приоритет NI = -6 (като root).
7. Покажете първите 5 процеса, използващи най-много оперативна памет.
8. Създайте нов потребител ivan, влезте от негово име и стартирайте три процеса sleep с аргумент 90 секунди.
9. Като root, намерете и прекратете само един от процесите на ivan, останалите оставете да работят.
10. Изведете броя на всички нишки в системата.
11. Изведете името и PID на процеса, който има най-нисък приоритет.
12. Стартирайте браузъра firefox и изведете броя на процесите, които той е създал.

13. Изпратете сигнал SIGSTOP към всички процеси, чиито имена започват с буквата х.
14. Изпратете сигнал SIGCONT за възстановяване на същите процеси.
15. Напишете команда за изключване на системата.

```
(zdravko@zdravko)-[~]  
$ sleep 100 &  
sleep 150 &  
sleep 200 &  
sleep 250 &  
  
[1] 26555  
[2] 26556  
[3] 26557  
[4] 26558
```

```
(zdravko@zdravko)-[~]  
$ jobs -l  
  
[1] 26555 running sleep 100  
[2] 26556 running sleep 150  
[3] - 26557 running sleep 200  
[4] + 26558 running sleep 250
```

```
(zdravko@zdravko)-[~]  
$ >fg %2  
[2] running sleep 150  
^C
```

```
(zdravko@zdravko)-[~]  
$ nice -n 10 xterm &  
  
[1] 4341  
  
nice: 'xterm': No such file or directory  
[1] + exit 127 nice -n 10 xterm
```

```
(zdravko@zdravko)-[~]  
$ pgrep -n xterm  
  
(zdravko@zdravko)-[~]  
$ renice -n 2 -p $(pgrep -n xterm)
```

```
(zdravko@zdravko)-[~]  
$ sudo nice -n -6 xeyes  
  
[1] 4341
```

```
(zdravko@zdravko)-[~]  
$ ps -eo pid,comm,%mem,rss --sort=-rss | head -n 6
```

PID	COMMAND	%MEM	RSS
2161	firefox-esr	4.0	572576
2459	Isolated Web Co	3.5	505848
2490	Isolated Web Co	2.8	401348
2594	Isolated Web Co	1.6	233488
2480	Isolated Web Co	1.3	189396

```
(zdravko@zdravko)-[~]  
$ sudo useradd -m ivan
```

```
(zdravko@zdravko)-[~]  
$ sudo passwd ivan
```

New password:

Retype new password:

passwd: password updated successfully

```
(zdravko@zdravko)-[~]  
$ su -ivan
```

su: invalid option -- 'i'

Try 'su --help' for more information.

```
(zdravko@zdravko)-[~]  
$ su - ivan
```

Password:

\$ sleep 90 &

\$ sleep 90 &

\$ sleep 90 &

```
(zdravko@zdravko)-[~]
$ ps -u ivan -o pid,comm,args

    PID COMMAND          COMMAND
  14876 sh                -sh
  15279 sleep             sleep 90
  15280 sleep             sleep 90
  15281 sleep             sleep 90
```

```
(zdravko@zdravko)-[~]
$ sudo kill 15280

[sudo] password for zdravko:
```

```
(zdravko@zdravko)-[~]
$ ps -u ivan -o pid,comm,args

    PID COMMAND          COMMAND
  14876 sh                -sh
  15279 sleep             sleep 90
  15280 sleep             [sleep] <defunct>
  15281 sleep             sleep 90
```

```
(zdravko@zdravko)-[~]
$ firefox &
```

```
[1] 17945
```

```
(zdravko@zdravko)-[~]
$
```

```
[1] + done          firefox
```

```
(zdravko@zdravko)-[~]
$ pgrep -c firefox
```

```
1
```

```
(zdravko@zdravko)-[~]
$ pkill -STOP '^x'
pkill -CONT '^x'
```

```
(zdravko@zdravko)-[~]
$ ps -eLo tid | tail -n +2 | wc -l

1062
```

```
(zdravko@zdravko)-[~]
$ ps -eo pid,ni,comm --sort=-ni | head -n 2

    PID  NI COMMAND
    138  19 khugepaged
```

```
sleep 100 &
sleep 150 &
sleep 200 &
sleep 250 &
```

& стартира процеса във фонов режим (background).

<code>jobs -l</code>	jobs показва фоновите задачи от текущия shell; -l добавя PID
<code>fg %2</code>	fg %2 вкарва втората задача на foreground. Ctrl+C праща SIGINT и прекратява процеса.
<code>nice -n 10 xterm &</code>	nice -n 10 стартира процес с по-нисък приоритет (по-високо NI).
<code>pgrep -n xterm</code> <code>renice -n 2 -p \$(pgrep -n xterm)</code>	renice променя NI на работещ процес по PID.
<code>sudo nice -n -6 xeyes</code>	Отрицателен NI е по-висок приоритет, затова се изисква root (sudo).
<code>ps -eo pid,comm,%mem,rss --sort=-rss head -n 6</code>	ps -eo избира колони; --sort=-rss сортира по RSS (реална RAM) низходящо; head -n 6 включва и заглавния ред + 5 процеса.
<code>sudo useradd -m ivan</code> <code>sudo passwd ivan</code> <code>su - ivan</code>	useradd -m прави home директория. su - ivan сменя потребителя със зареждане на профила му.
<code>ps -u ivan -o pid,comm,args</code> <code>sudo kill 1234</code>	ps -u ivan показва процеси на потребителя. kill PID прекратява само конкретния процес.
<code>ps -eLo tid tail -n +2 wc -l</code>	ps -eL изброява нишки; tid е ID на нишка. tail -n +2 маха заглавния ред, wc -l брой.
<code>ps -eo pid,ni,comm --sort=-ni head -n 2</code>	Сортира по NI низходящо и показва първия процес (след заглавието).
<code>firefox &</code> <code>pgrep -c firefox</code>	pgrep -c брой процесите с име firefox

<code>pkill -STOP '^x'</code>	pkill праща сигнал по име; ^x означава „започва с x“
<code>pkill -CONT '^x'</code>	SIGCONT възобновява спрени процеси
<code>sudo shutdown -h now</code>	shutdown -h now изключва машината веднага (нужен е root).

ЗАДАЧА 9 - ПАКЕТНИ МЕНИДЖЪРИ

Задание: Инсталиране и работа с Geany

1.1 Инсталирайте чрез командния интерфейс приложението Geany във вашата Linux дистрибуция.

Приложете списък с използваните команди и направете екранна снимка на стартираната програма Geany.

1.2 Създайте чрез Geany нов файл на избран от вас език за програмиране (например Python, C или C++).

Въведете кратка програма, която извежда съобщение на екрана.

Направете екранна снимка на прозореца на Geany с отворения файл и резултата от изпълнението.

1.3 Деинсталирайте програмата Geany чрез командния интерфейс.

Приложете списък с използваните команди.

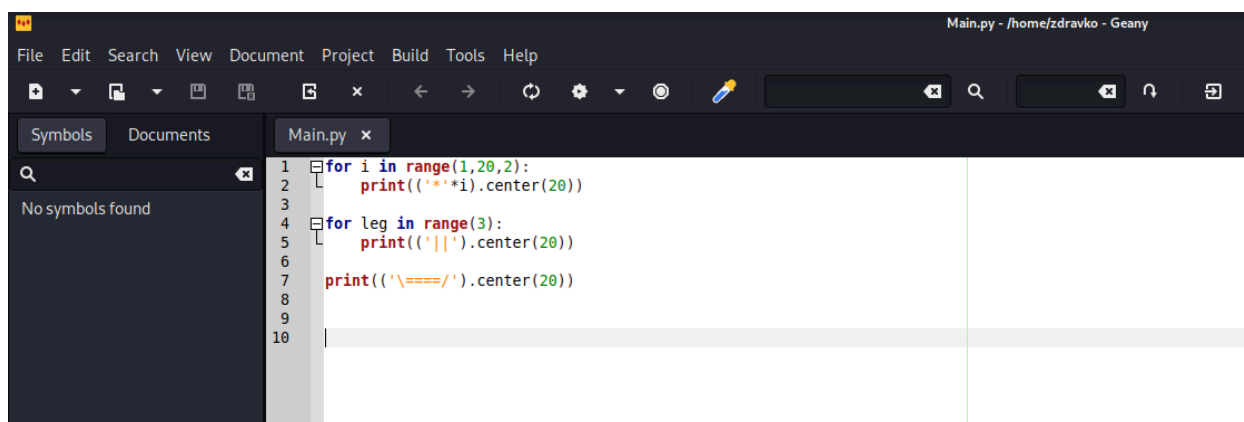
```

/home/zdravko/Main.py:7: SyntaxWarning: invalid escape sequence '\='
print(('\\==/') .center(20))

  *
 ***
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
  ||
  ||
  ||
  \\==/

(program exited with code: 0)
Press return to continue

```



<code>sudo apt update</code>	Обновяване на списъка с пакети
<code>sudo apt install geany</code>	Инсталиране на Geany
<code>geany &</code>	Стартиране на Geany
File → New → Save as hello.py	Създаване на Python файл (в Geany)
Build → Execute (F5)	Изпълнение на програмата (в Geany)

<code>sudo apt remove geany</code>	Деинсталиране на Geany
<code>sudo apt autoremove</code>	Премахване на ненужни зависимости
<code>sudo apt purge geany</code>	Пълно премахване с конфигурации (по избор)

ЗАДАЧА 10 - РАБОТА С АРХИВИ

Вариант за нечетен номер

1.1 Създайте архив с име `home_backup.tar` от всички файлове и директории във вашата

домашна директория.

1.2 Разархивирайте архива `home_backup.tar` в поддиректория с име `restore`.

1.3 Добавете всички файлове с разширение `.sh` (скриптове) от текущата директория към

архива `scripts_backup.tar`.

1.4 Изведете списък на всички елементи, съдържащи се в архива `home_backup.tar`, заедно с техните размери.

1.5 Изведете списък само на тези файлове от архива `home_backup.tar`, които съдържат

думата „`config`“ в името си.

```
(zdravko@zdravko)-[~]
$ sudo tar -cf home_backup.tar ~

tar: Removing leading `/' from member names
tar: /home/zdravko/home_backup.tar: archive cannot contain itself; not dumped
tar: /home/zdravko/.cache/JetBrains/PyCharm2025.2/.port: socket ignored
tar: /home/zdravko/.ssh/agent/s.qo0h2f1PJD.agent.jX3vui4KPU: socket ignored
tar: /home/zdravko/.ssh/agent/s.qo0h2f1PJD.agent.p5wJBHQ28m: socket ignored
tar: /home/zdravko/.ssh/agent/s.qo0h2f1PJD.agent.CizLhzm6WN: socket ignored

(zdravko@zdravko)-[~]
$ ls
cookies.txt  deep_find_flags.sh  Documents  example.txt  file1.txt  filtered_list  found_flags.txt  group
data         Desktop            Downloads  fg           file.txt   forensic_tmp   fruits.txt       home_backup.tar
```

```

(zdravko@zdravko)-[~]
$ tar -xvf home_backup.tar -C restore
tar: home/zdravko/PyCharmMiscProject/Dockerfile: time stamp 2049-01-01 04:00:00 is 725713518.084114796 s in the future
tar: home/zdravko/PyCharmMiscProject/chall.c: time stamp 2049-01-01 04:00:00 is 725713516.915842294 s in the future
tar: home/zdravko/PyCharmMiscProject/chall.c: time stamp 2049-01-01 04:00:00 is 725713516.915756993 s in the future
tar: home/zdravko/.cache/.fr-MjCmms/chall.c: time stamp 2049-01-01 04:00:00 is 725713513.067985862 s in the future
tar: home/zdravko/.cache/.fr-Chk9TV/Dockerfile: time stamp 2049-01-01 04:00:00 is 725713513.052945198 s in the future
tar: home/zdravko/.cache/.fr-mVORWF/chall.c: time stamp 2049-01-01 04:00:00 is 725713513.052298258 s in the future
tar: home/zdravko/PyCharmProjects/PythonProject/Dockerfile: time stamp 2049-01-01 04:00:00 is 725713513.011737228 s in the future
tar: home/zdravko/PyCharmProjects/PythonProject/chall: time stamp 2049-01-01 04:00:00 is 725713512.213451041 s in the future
tar: home/zdravko/PyCharmProjects/PythonProject/chall.c: time stamp 2049-01-01 04:00:00 is 725713512.211449764 s in the future
tar: home/zdravko/Downloads/HARD/pwn/Dockerfile: time stamp 2049-01-01 04:00:00 is 725713511.577678547 s in the future
tar: home/zdravko/Downloads/HARD/pwn/chall: time stamp 2049-01-01 04:00:00 is 725713501.452272563 s in the future
tar: home/zdravko/Downloads/HARD/pwn/compile.sh: time stamp 2049-01-01 04:00:00 is 725713501.451048233 s in the future
tar: home/zdravko/Downloads/HARD/pwn/chall.c: time stamp 2049-01-01 04:00:00 is 725713501.450999321 s in the future
tar: home/zdravko/Downloads/mini_heap/Dockerfile: time stamp 2049-01-01 04:00:00 is 725713501.321936118 s in the future
tar: home/zdravko/Downloads/mini_heap/chall: time stamp 2049-01-01 04:00:00 is 725713501.321854213 s in the future
tar: home/zdravko/Downloads/mini_heap/chall.c: time stamp 2049-01-01 04:00:00 is 725713501.321810381 s in the future

(zdravko@zdravko)-[~]
$ cd restore

(zdravko@zdravko)-[~/restore]
$ ls
home

(zdravko@zdravko)-[~/restore]
$ cd home

(zdravko@zdravko)-[~/restore/home]
$ ls
zdravko

(zdravko@zdravko)-[~/restore/home]
$ cd zdravko

(zdravko@zdravko)-[~/restore/home/zdravko]
$ ls
cookies.txt  deep_find_flags.sh  Documents  example.txt  file1.txt  filtered_list  found_flags.txt  group  Music  OOP_UPR  Pictures  PyCharmMiscProject  Romania  vasev
data         Desktop      Downloads  fg            file.txt  forensic_tmp  fruits.txt      Main.py  nef.txt  OS_COURSE_WORK  Public  PyCharmProjects  Templates  Videos

```

```

(zdravko@zdravko)-[~/restore/home/zdravko]
$ tar -cf scripts_backup.tar *.sh

(zdravko@zdravko)-[~/restore/home/zdravko]
$ cat scripts_backup.tar
deep_find_flags.sh000077500017500000000467215073770506014455 0ustar  zdravkozdravko#I/usr/bin/env bash
set -e

ROOT="$1"
OUT="found_flags.txt"
TMPDIR="forensic_tmp"
: > "$OUT"
rm -rf "$TMPDIR"
mkdir -p "$TMPDIR"

echo "[*] Scanning text & binary (strings) for common patterns..."
# 1. Search plain text and binary (via strings)
rg -i --hidden --no-ignore -n --glob 'inode_modules' 'flag\{.*?\}|FLAG\{.*?\}|FLAG:.*|CTF\{.*?\}' "$ROOT" >> "$OUT" || true

# 2. Run strings on non-text files and search
find "$ROOT" -type f ! -name '*.txt' -print0 | while IFS= read -r -d '' f; do
    # quick skip small text files already matched
    filetype=$(file -b --mime-encoding "$f" 2>/dev/null || echo "binary")
    if [[ "$filetype" != "us-ascii" && "$filetype" != "utf-8" ]]; then
        strings "$f" | rg -i 'flag\{.*?\}|FLAG\{.*?\}|FLAG:.*|CTF\{.*?\}' -n --hidden --no-ignore && \
            printf '%s: (from strings)\n' "$f" >> "$OUT"
    fi
done

# 3. Extract common archives (zip, 7z, tar, rar) into tmp and scan extracted
echo "[*] Extracting archives and scanning inside..."
find "$ROOT" -type F \( -iname '*.zip' -o -iname '*.7z' -o -iname '*.tar' -o -iname '*.tar.gz' -o -iname '*.tgz' -o -iname '*.rar' \) -print0 | while IFS= read -r -d '' arc; do
    mkdir -p "$TMPDIR/extracted"
    7z x -y -o"$TMPDIR/extracted" "$arc" >/dev/null 2>&1 || unzip -qq -d "$TMPDIR/extracted" "$arc" >/dev/null 2>&1 || tar -xpf "$arc" -C "$TMPDIR/extracted" >/dev/null 2>&1 || true
    rg -i -n 'flag\{.*?\}|FLAG\{.*?\}|FLAG:.*|CTF\{.*?\}' "$TMPDIR/extracted" >> "$OUT" || true
    rm -rf "$TMPDIR/extracted"/*
done

# 4. Images: check metadata and simple stego tools if installed
echo "[*] Checking image metadata..."
find "$ROOT" -type f \( -iname '*.png' -o -iname '*.jpg' -o -iname '*.jpeg' -o -iname '*.bmp' \) -print0 | while IFS= read -r -d '' img; do
    exiftool "$img" 2>/dev/null | rg -i 'flag|ctf|comment|description' -n && echo "$img (exif)" >> "$OUT"
    # zsteg if available (png LSB)
    if command -v zsteg >/dev/null 2>&1; then
        zsteg "$img" 2>/dev/null | rg -i 'flag|ctf' -n && echo "$img (zsteg)" >> "$OUT"
    fi
done

# 5. Common encodings: search for base64 / hex that might decode to flags
echo "[*] Searching for base64-like strings that decode to flags..."
rg -o --hidden --no-ignore -n '([A-Za-z0-9+/]{40,}={0,2})' "$ROOT" | cut -d: -f1,2,3 | while IFS= read -r line; do
    b64=$(echo "$line" | awk -F: {print $3})
    echo "$b64" | base64 -d 2>/dev/null | rg -i 'flag\{.*?\}|FLAG\{.*?\}' -n && echo "base64-found-from:$line" >> "$OUT" || true
done

echo "[*] Done. Results (if any) in $OUT"

```



```
(zdravko@zdravko) ~/restore/home/zdravko/
$ tar -tvf home_backup.tar
drwx----- zdravko/zdravko 0 2026-01-02 16:54 home/zdravko/
-rw----- zdravko/zdravko 0 2025-10-15 18:46 home/zdravko/.ICEauthority
drwxr-xr-x zdravko/zdravko 0 2025-10-15 18:46 home/zdravko/Templates/
-rw-rw-r-- zdravko/zdravko 9860270080 2026-01-02 16:52 home/zdravko/home_backup.tar
drwxrwxr-x zdravko/zdravko 0 2025-12-29 07:42 home/zdravko/Romania/
drwxrwxr-x zdravko/zdravko 0 2025-12-29 07:38 home/zdravko/Romania/Bucharest/
drwxrwxr-x zdravko/zdravko 0 2025-12-29 07:38 home/zdravko/Romania/Bucharest/Sector1/
drwxrwxr-x zdravko/zdravko 0 2025-11-29 09:06 home/zdravko/Romania/Bucharest/Sector6 ~/
drwxrwxr-x zdravko/zdravko 0 2025-11-29 09:06 home/zdravko/Romania/Bucharest/Sector6 ~/Romania/
drwxrwxr-x zdravko/zdravko 0 2025-11-29 09:06 home/zdravko/Romania/Bucharest/Sector6 ~/Romania/Cluj/
drwxrwxr-x zdravko/zdravko 0 2025-11-29 09:06 home/zdravko/Romania/Bucharest/Sector6 ~/Romania/Cluj/Cluj-Napoca ~/
drwxrwxr-x zdravko/zdravko 0 2025-11-29 09:06 home/zdravko/Romania/Bucharest/Sector6 ~/Romania/Cluj/Cluj-Napoca ~/Romania/
drwxrwxr-x zdravko/zdravko 0 2025-11-29 09:06 home/zdravko/Romania/Bucharest/Sector6 ~/Romania/Cluj/Cluj-Napoca ~/Romania/Constanta/
drwxrwxr-x zdravko/zdravko 0 2025-11-29 09:06 home/zdravko/Romania/Bucharest/Sector6 ~/Romania/Cluj/Cluj-Napoca ~/Romania/Constanta/Mangalia ~/
drwxrwxr-x zdravko/zdravko 0 2025-11-29 09:06 home/zdravko/Romania/Bucharest/Sector6 ~/Romania/Cluj/Cluj-Napoca ~/Romania/Constanta/Mangalia ~/Romania/
drwxrwxr-x zdravko/zdravko 0 2025-11-29 09:06 home/zdravko/Romania/Bucharest/Sector6 ~/Romania/Cluj/Cluj-Napoca ~/Romania/Constanta/Mangalia ~/Romania/Iasi/
drwxrwxr-x zdravko/zdravko 0 2025-11-29 09:06 home/zdravko/Romania/Bucharest/Sector6 ~/Romania/Cluj/Cluj-Napoca ~/Romania/Constanta/Mangalia ~/Romania/Iasi/Pascani/
drwxrwxr-x zdravko/zdravko 0 2025-11-29 09:06 home/zdravko/Romania/Bucharest/Sector6 ~/Romania/Cluj/Cluj-Napoca ~/Romania/Constanta/Mangalia ~/Romania/Iasi/Targu Frumos/
```

```
(zdravko@zdravko) ~/restore/home/zdravko/
$ tar -tf home_backup.tar | grep config
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/bin/numpy-config
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/scapy/contrib/automotive/scanner/configuration.py
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/scapy/contrib/automotive/scanner/__pycache__/configuration.cpython-313.pyc
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/scapy/__pycache__/config.cpython-313.pyc
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/scapy/config.py
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/mako/testing/__pycache__/config.cpython-313.pyc
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/mako/testing/config.py
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/mako/testing/config.py
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/numpy/typing/tests/data/reveal/ufunc_config.pyi
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/numpy/typing/tests/data/pass/__pycache__/ufunc_config.cpython-313.pyc
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/numpy/typing/tests/data/pass/ufunc_config.py
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/numpy/typing/tests/data/fail/ufunc_config.pyi
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/numpy/__pycache__/configtool.cpython-313.pyc
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/numpy/__pycache__/__config__.cpython-313.pyc
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/numpy/tests/test_numpy_config.py
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/numpy/tests/__pycache__/test_numpy_config.cpython-313.pyc
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/numpy/tests/test_configtool.cpython-313.pyc
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/numpy/_core/_ufunc_config.py
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/numpy/_core/lib/npypkg-config/
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/numpy/_core/lib/npypkg-config/mlib.ini
home/zdravko/restore/home/zdravko/PyCharmMiscProject/.venv/lib/python3.13/site-packages/numpy/_core/lib/npypkg-config/npymath.ini
```

<code>tar -cf home_backup.tar ~</code>	Създава архив <code>home_backup.tar</code> , който съдържа всички файлове и директории от домашната директория.
<code>mkdir restore</code>	Създава поддиректория с име <code>restore</code> , в която ще се разархивира архивът.
<code>tar -xf home_backup.tar -C restore</code>	Разархивира съдържанието на <code>home_backup.tar</code> в директорията <code>restore</code> .
<code>tar -cf scripts_backup.tar *.sh</code>	Архивира всички файлове с разширение <code>.sh</code> от текущата директория.
<code>tar -rf scripts_backup.tar *.sh</code>	Добавя <code>.sh</code> файловете към вече съществуващия архив <code>scripts_backup.tar</code> .

<pre>tar -tvf home_backup.tar</pre>	Показва списък на всички файлове в архива заедно с размери и допълнителна информация.
<pre>tar -tf home_backup.tar grep config</pre>	Извежда само тези файлове от архива, които съдържат думата „config“ в името си.