НПГ по КТС – ГРАД ПРАВЕЦ

Курсова работа

по

ОПЕРАЦИОННИ СИСТЕМИ

„Командния интерфейс на Linux“

Ученик: Росен Росенов Иванов ; Преподавател: Памела Николова, Йордан Владов

02.01.2026

Теоретична част

1. Кратко описание на линукс

Linux представлява свободна и отворена операционна система, базирана на Linux ядрото, разработено за първи път от Линус Торвалдс през 1991 г. Тя е многозадачна, многопотребителска и работи на широк спектър от хардуерни платформи – от персонални компютри и сървъри до вградени системи и мобилни устройства.

Основно предимство на Linux е неговият отворен код, което означава, че всеки потребител има право да разглежда, променя и разпространява изходния код. Това води до бързо развитие, висока сигурност и стабилност на системата. Благодарение на голямата общност от разработчици, Linux се използва широко в сървърни среди, облачни платформи, научни изследвания и киберсигурност.

Linux използва йерархична файлова система, като всички файлове и директории са организирани под основната директория `/`. Управлението на процесите, паметта и хардуерните ресурси се извършва от ядрото, което осигурява ефективна и стабилна работа на системата.

Операционната система предлага мощен команден ред (Terminal), чрез който потребителите могат да изпълняват команди, да управляват файлове, потребители и услуги. Това прави Linux предпочитан избор за системни администратори и напреднали потребители.

2. Описание на използвана линукс дистрибуция (МакОС Terminal)

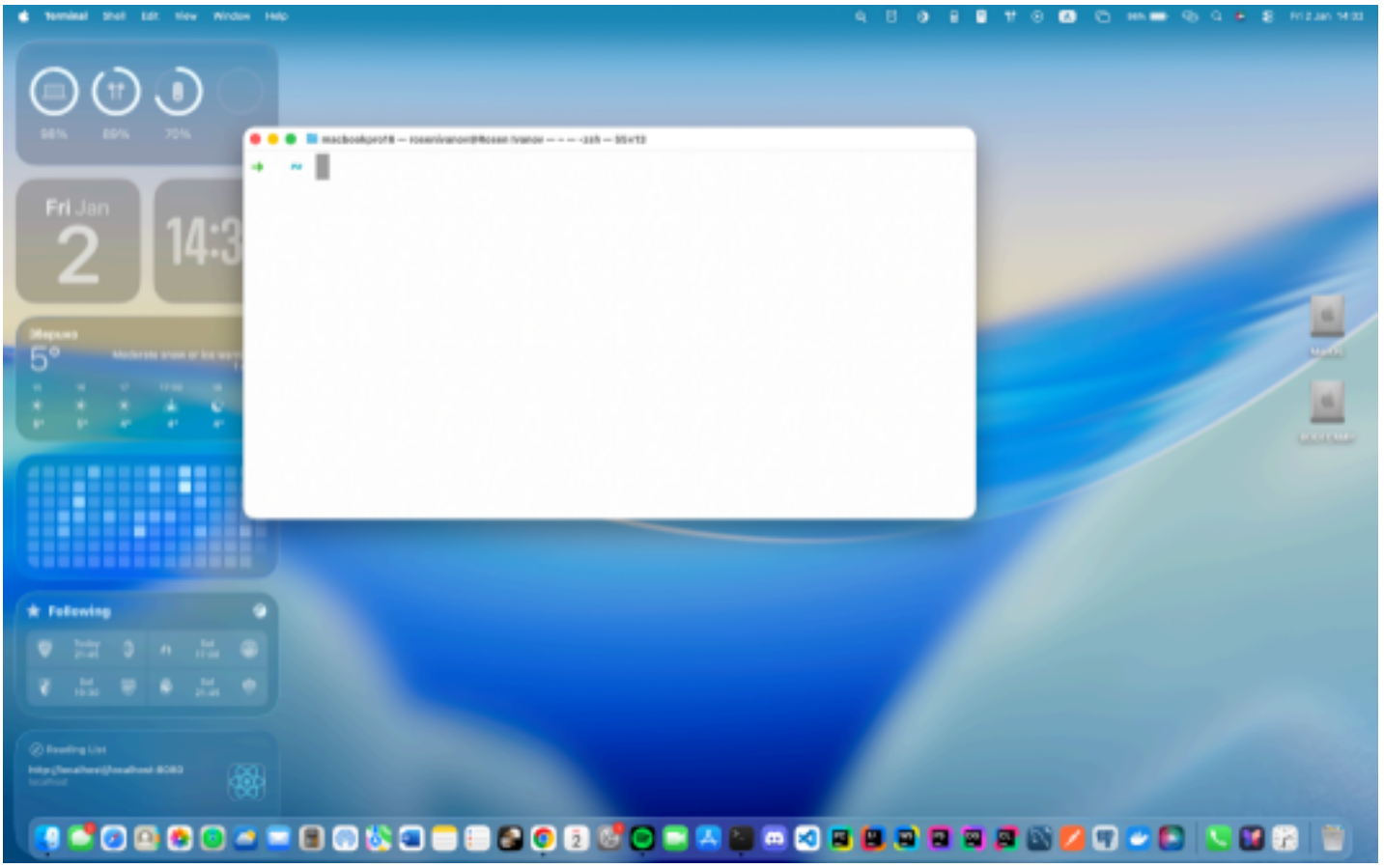
В настоящата курсова работа практическите задачи са изпълнени чрез Terminal приложението на операционната система macOS. macOS е Unix-базирана операционна система, което я прави функционално сходна с Linux по отношение на работа с команден ред и основни системни принципи. Това позволява използването на голяма част от стандартните Unix и Linux команди.

Терминалът на macOS предоставя възможност за работа с файловата система, управление на процеси и настройване на права за достъп по начин, сходен с този в Linux среда. Команди като **`ls`**, **`cd`**, **`pwd`**, **`mkdir`**, **`rm`**, **`cp`**, **`mv`** и **`chmod`** функционират по аналогичен начин, което позволява ефективно изпълнение на практическите упражнения, свързани с Linux команден ред.

Използването на macOS Terminal е избрано поради наличната хардуерна среда и възможността за работа в Unix-подобна система без необходимост от инсталиране на допълнителна Linux дистрибуция или виртуална машина. Това улеснява процеса на обучение
и позволява фокусиране върху усвояването на основните команди и концепции.

Макар macOS да не представлява Linux дистрибуция, сходството в архитектурата и командната среда позволява прилагането на наученото в реална Linux система. По този начин използваната среда е подходяща за обучение и изграждане на базови умения за работа с Linux и Unix-подобни операционни системи.

*** Как изглежда командния интерфейс и показно на кое е използваното приложение (Terminal):



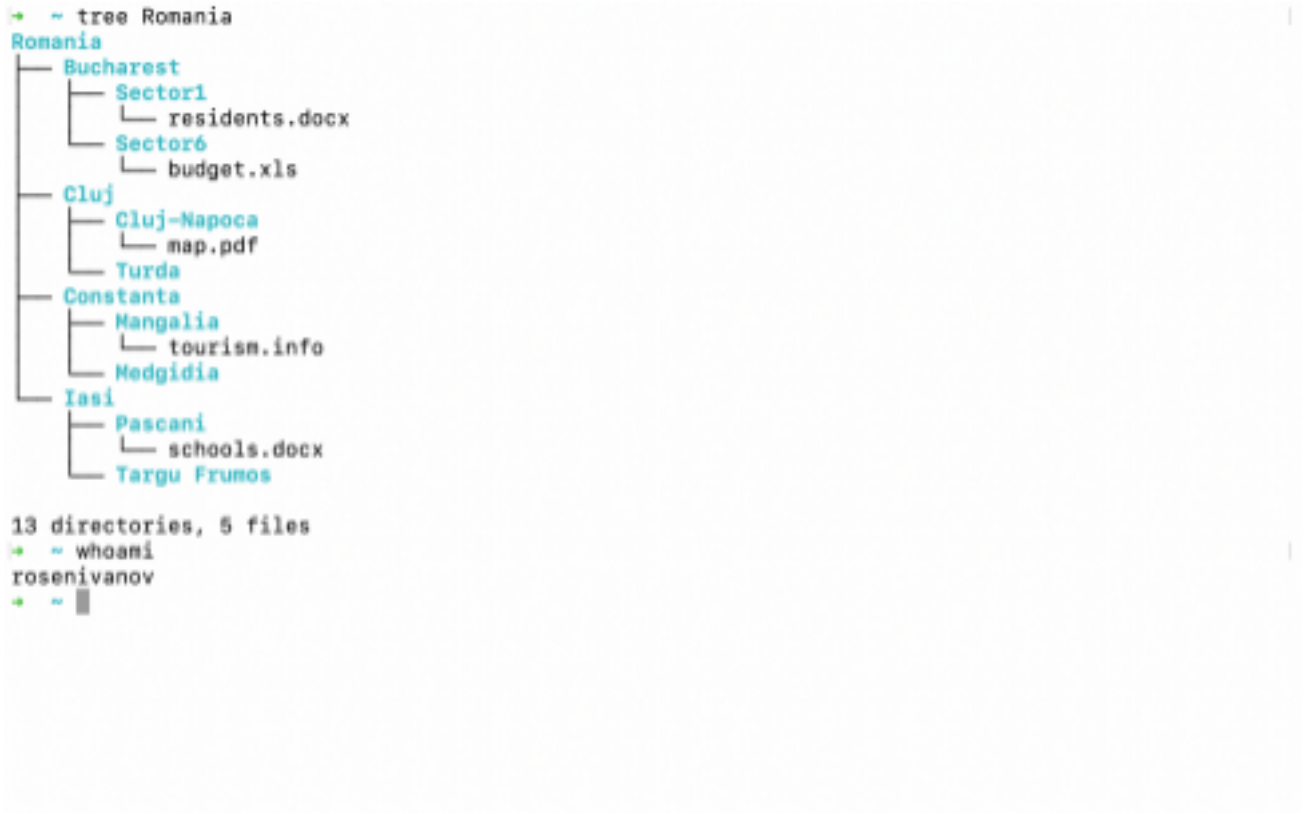
Задача 1 - СЪЗДАВАНЕ НА ФАЙЛОВЕ И ДИРЕКТОРИИ

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 илидиректорията “Iasi” при нечетен номер. (22324 = 2 + 2 + 3 + 2 + 4 = 13; 1 + 3 = 4; Четните съм аз)

1. ➜ ~ mkdir Romania
2. ➜ ~ cd Romania
3. ➜ Romania mkdir Bucharest
4. ➜ Romania mkdir Cluj
5. ➜ Romania mkdir Constanta
6. ➜ Romania mkdir Iasi

7. ➜ Romania cd Bucharest
8. ➜ Bucharest mkdir Sector1
9. ➜ Bucharest mkdir Sector6
10. ➜ Bucharest cd Sector1
11. ➜ Sector1 touch residents.docx
12. ➜ Sector1 cd ..
13. ➜ Bucharest cd Sector6 14.
➜ Sector6 touch budget.xls
15. ➜ Sector6 cd ..
16. ➜ Bucharest cd ..
17. ➜ Romania cd Cluj
18. ➜ Cluj mkdir Cluj-Napoca
19. ➜ Cluj mkdir Turda
20. ➜ Cluj cd Cluj-Napoca 21. ➜
Cluj-Napoca touch map.pdf 22.
➜ Cluj-Napoca cd ..
23. ➜ Cluj cd ..
24. ➜ Romania cd Constanta 25. ➜
Constanta mkdir Mangalia 26. ➜
Constanta mkdir Medgidia 27. ➜
Constanta cd Mangalia 28. ➜
Mangalia touch tourism.info 29.
➜ Mangalia cd ..
30. ➜ Constanta cd ..
31. ➜ Romania cd Iasi
32. ➜ Iasi mkdir Pascani
33. ➜ Iasi mkdir “Targu Frumos”
34. ➜ Iasi cd Pascani
35. ➜ Pascani touch schools.docx
36. ➜ Pascani cd

37. ➜ ~ tree Romania



38. ➜ ~ rm -rf Romania/Cluj

Командите `mkdir`, `cd` и `touch` се използват за създаване на директории, навигация между тях и създаване на файлове. Командата `tree` визуализира цялата файлово-директориална структура по ясен начин. `rm -rf` служи за изтриване на избрана директория заедно със съдържанието ѝ. Всички команди демонстрират базова работа с файловата система.

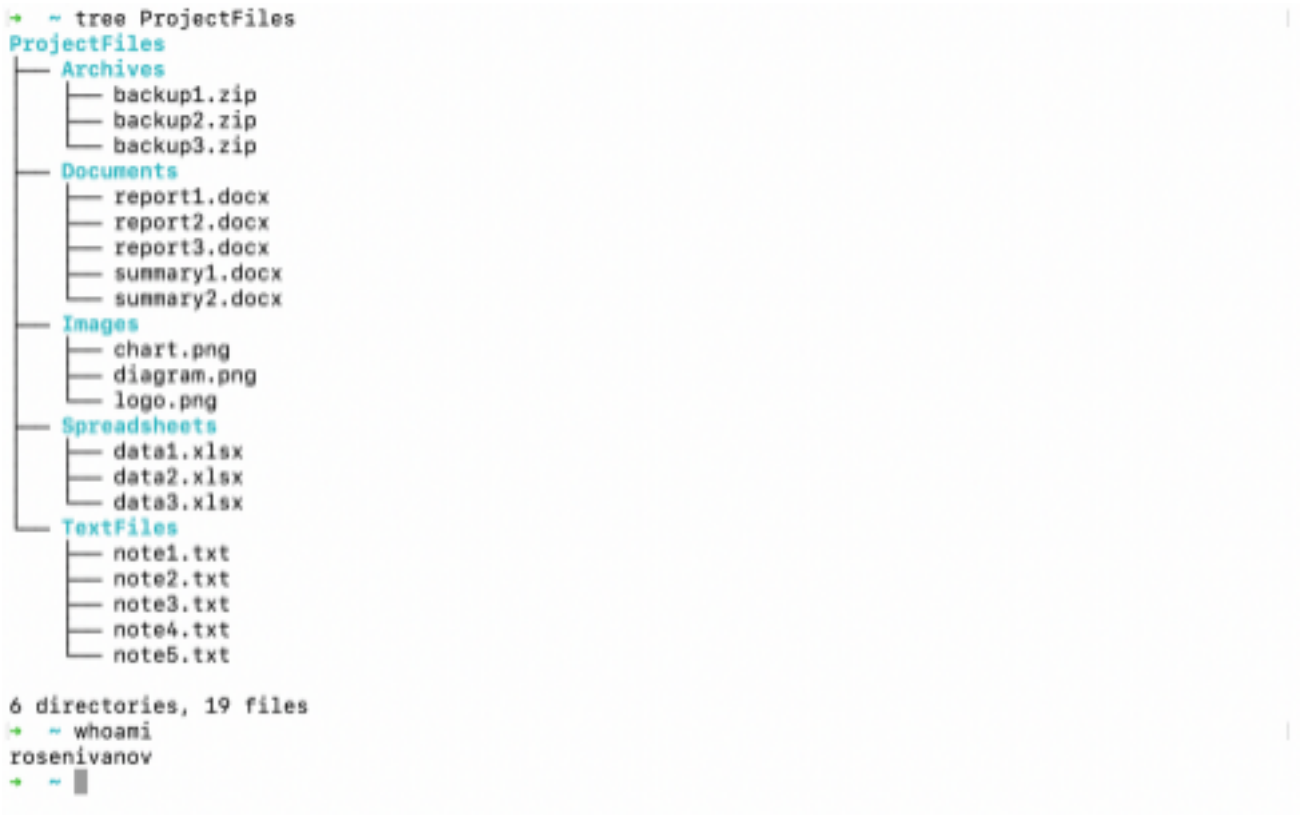
ЗАДАЧА 2 - СЪЗДАВАНЕ НА ФАЙЛОВЕ И ДИРЕКТОРИИ

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.

1. ➜ ~ mkdir ProjectFiles

```
2. ➜ ~ cd ProjectFiles
3. ➜ ProjectFiles mkdir Documents
4. ➜ ProjectFiles mkdir TextFiles
5. ➜ ProjectFiles mkdir Spreadsheets
6. ➜ ProjectFiles mkdir Images
7. ➜ ProjectFiles mkdir Archives
8. ➜ ProjectFiles cd Documents
9. ➜ Documents touch report1.docx 10.
➜ Documents touch report2.docx 11. ➜
Documents touch report3.docx 12. ➜
Documents touch summary1.docx 13. ➜
Documents touch summary2.docx 14. ➜
Documents cd ..
15. ➜ ProjectFiles cd TextFiles
16. ➜ TextFiles touch note1.txt
17. ➜ TextFiles touch note2.txt
18. ➜ TextFiles touch note3.txt
19. ➜ TextFiles touch note4.txt
20. ➜ TextFiles touch note5.txt
21. ➜ TextFiles cd ..
22. ➜ ProjectFiles cd Spreadsheets 23.
➜ Spreadsheets touch data1.xlsx 24.
➜ Spreadsheets touch data2.xlsx 25.
➜ Spreadsheets touch data3.xlsx 26.
➜ Spreadsheets cd ..
27. ➜ ProjectFiles cd Images
28. ➜ Images touch logo.png
29. ➜ Images touch diagram.png
30. ➜ Images touch chart.png
31. ➜ Images cd ..
32. ➜ ProjectFiles cd Archives
33. ➜ Archives touch backup1.zip
```

```
34. ➜ Archives touch backup2.zip
35. ➜ Archives touch backup3.zip
36. ➜ Archives cd
37. ➜ ~ tree ProjectFiles
38. ➜ ~ rm -rf ProjectFiles/Archives/
```



Отново се използват `mkdir`, `cd` и `touch` за изграждане на зададена структура от папки и файлове. Командата `tree` показва резултата от създадената структура. `rm -rf` се използва за изтриване на конкретна директория според условието. Целта е упражняване на структуриране на проектни файлове.

ЗАДАЧА 3 - КОПИРАНЕ И ПРЕМЕСТВАНЕ НА ФАЙЛОВЕ

Създайте файловете и директориите от фигурата по-долу във вашата домашна директория. Вариант за четен номер. Копирайте всички .txt файлове в директорията DIR3. Преместете всички .pdf файлове в директорията DIR1. Преместете всички файлове, които започват с името strypes, в директорията DIR2. Копирайте всички .docx файлове в директорията DIR4. Изтрийте всички файлове, които имат три символа като разширение в директорията DIR3 (например .txt, .pdf, .doc). Направете DIR1 скрита директория 1. ➜ ~ cd DIR

```
2. ➜ DIR cp DIR*/*.txt DIR3
```

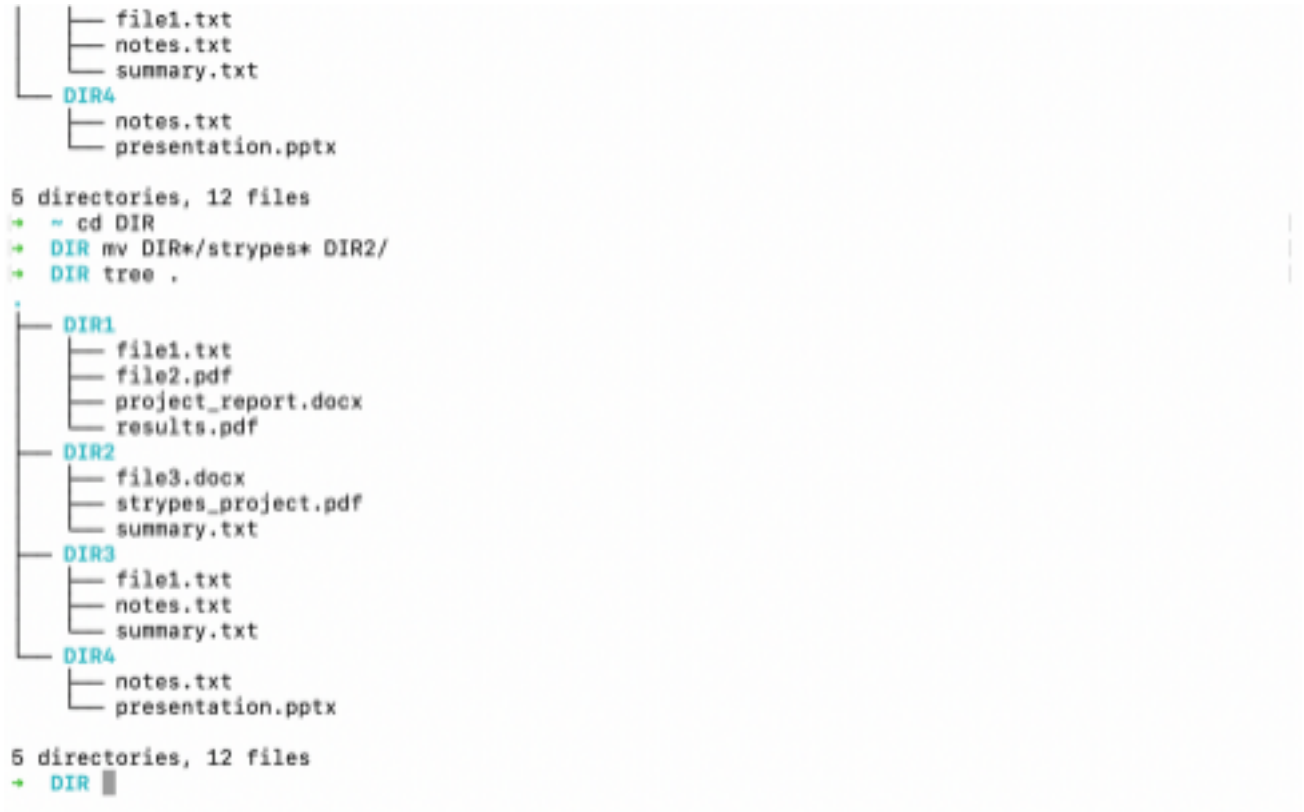
3. ➜ DIR cd
4. ➜ ~ tree DIR

```
├── DIR3
└── DIR4
    ├── notes.txt
    ├── presentation.pptx
    └── strypes_project.pdf

5 directories, 9 files
➜  ~ cd DIR
➜  DIR cp DIR*/*.txt DIR3
➜  DIR cd
➜  ~ tree DIR
DIR
├── DIR1
│   ├── file1.txt
│   ├── file2.pdf
│   └── project_report.docx
├── DIR2
│   ├── file3.docx
│   ├── results.pdf
│   └── summary.txt
├── DIR3
│   ├── file1.txt
│   ├── notes.txt
│   └── summary.txt
└── DIR4
    ├── notes.txt
    ├── presentation.pptx
    └── strypes_project.pdf

5 directories, 12 files
➜  ~
```

1. ➜ ~ cd DIR
2. ➜ DIR mv DIR*/*.pdf DIR1/
   mv: DIR1/file2.pdf and DIR1/file2.pdf are identical
3. ➜ DIR cd
4. ➜ ~ tree DIR

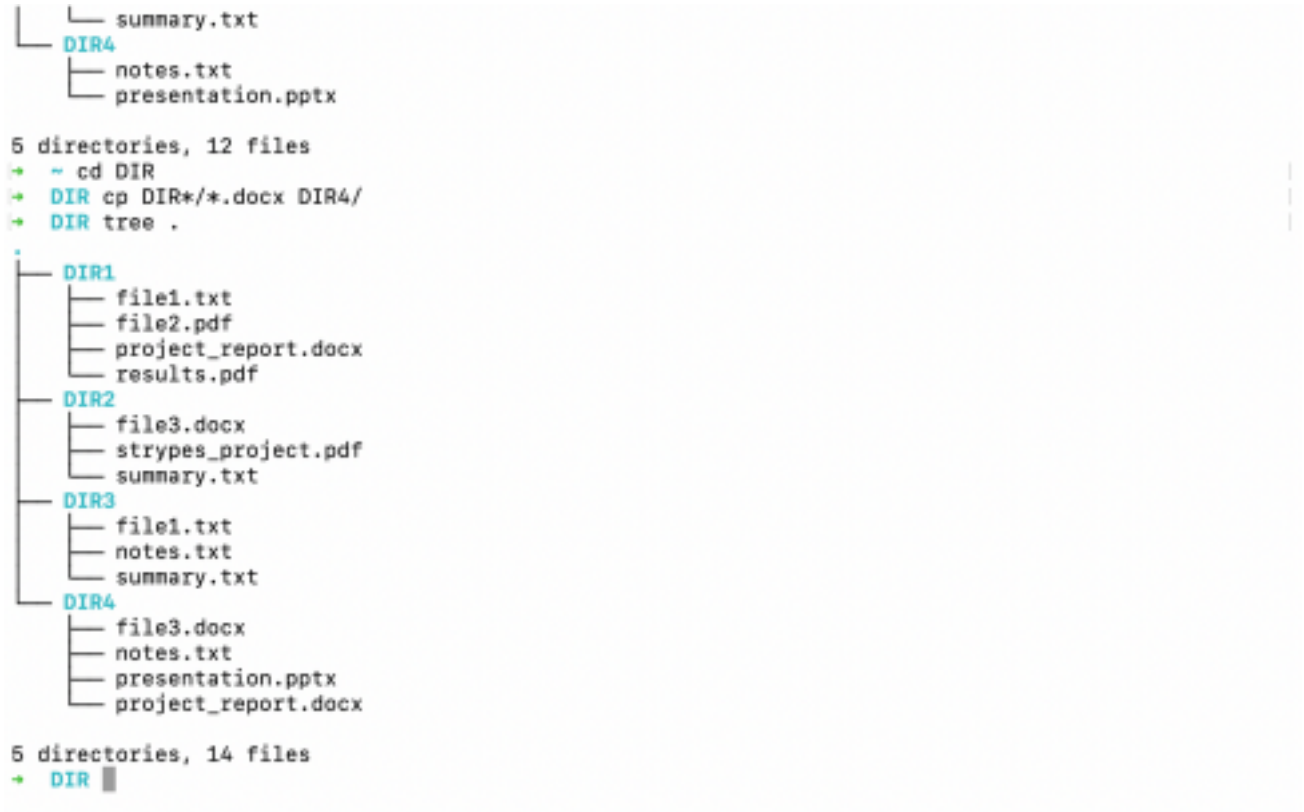
1. ➜ ~ cd DIR

2. ➜ DIR mv DIR*/strypes* DIR2/
3. ➜ DIR tree .



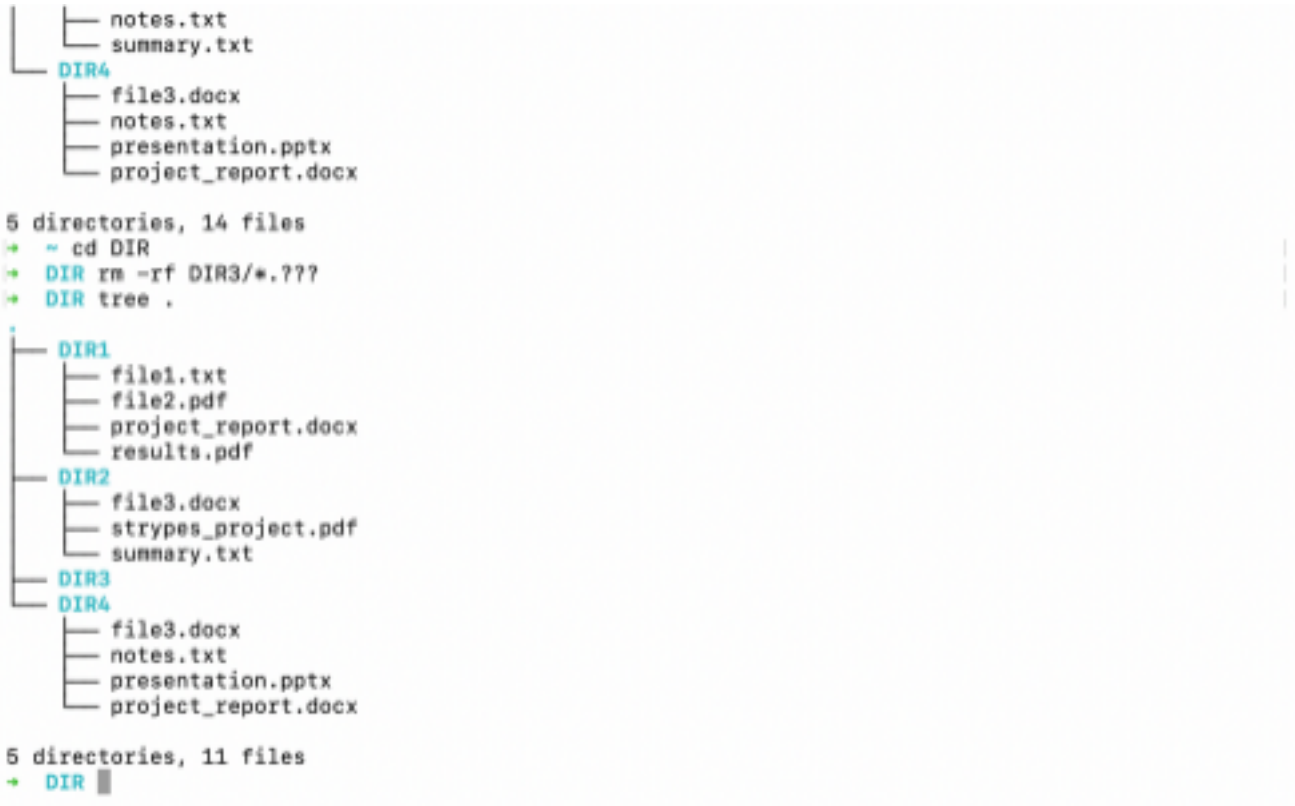
1. ➜ ~ cd DIR
2. ➜ DIR cp DIR*/*.docx DIR4/

3. ➜ DIR tree .



1. ➜ ~ cd DIR
2. ➜ DIR rm -rf DIR3/*.???

3. ➜ DIR tree .



1. ➜ DIR mv DIR1/ .DIR1/
2. ➜ DIR tree .

Командите `cp` и `mv` се използват за копиране и преместване на файлове според разширения и имена. Глобалните символи (`*.txt, strypes*`) позволяват работа с множество файлове наведнъж. `rm` изтрива файлове по зададен шаблон, а `mv DIR1 .DIR1` прави директорията скрита. Задачата упражнява управление на файлове и шаблони.

ЗАДАЧА 4 - ПОТРЕБИТЕЛИ, ГРУПИ И ПРАВА

Вариант за четен номер

Задача: Управление на потребители, групи и файлови права в Linux 1. Създайте потребител с име engineer и групи с имена engteam и admteam. 2. Добавете потребителя engineer към групите engteam и admteam.

3. Копирайте файла /etc/passwd във вашата домашна директория.

4. Променете правата на копирания файл на rw-r-----.

o Използвайте както буквен, така и цифров формат (например chmod u=rw,g=r,o=--- и chmod 640).

5. Направете файла собственост на потребителя engineer и групата engteam.

6. Проверете дали промените са приложени успешно с командите ls -l и groups engineer.

Задача 1 и 2:

```
1.➜ ~ sudo dscl . -create /Groups/engteam  2.➜ ~ sudo dscl .
-create /Groups/admteam

3.➜ ~ sudo dscl . -create /Users/engineer 4.➜ ~ sudo dscl .
-append /Groups/engteam GroupMembership engineer
5.➜ ~ sudo dscl . -append /Groups/admteam GroupMembership
  engineer
6.➜ ~ dscl . -read /Groups/engteam
  GroupMembership
7. GroupMembership: engineer
8.➜ ~ dscl . -read /Groups/admteam
  GroupMembership
9. GroupMembership: engineer
```

```
➜  ~ sudo dscl . -create /Groups/engteam
➜  ~ sudo dscl . -create /Groups/admteam
➜  ~ sudo dscl . -create /Users/engineer
➜  ~ sudo dscl . -append /Groups/engteam GroupMembership engineer
➜  ~ sudo dscl . -append /Groups/admteam GroupMembership engineer
➜  ~ dscl . -read /Groups/engteam GroupMembership
GroupMembership: engineer
➜  ~ dscl . -read /Groups/admteam GroupMembership
GroupMembership: engineer
➜  ~ whoami
rosenivanov
➜  ~
```

Задача 3:

1. ➜ ~ cp /etc/passwd ~

Задача 4:

1. ➜ ~ chmod 640 passwd
2. ➜ ~ ls passwd
3. passwd
4. ➜ ~ ls -la passwd
5. -rw-r----- 1 rosenivanov staff 9344 Jan 2 16:03 passwd

1. ➜ ~ chmod u=rw,g=r,o=--- passwd
2. ➜ ~ ls -la passwd
3. -rw-r----- 1 rosenivanov staff 9344 Jan 2 16:03 passwd

```
➜  ~ chmod 640 passwd
➜  ~ ls passwd
passwd
➜  ~ ls -la passwd
-rw-r-----  1 rosenivanov  staff  9344 Jan  2 16:03 passwd
➜  ~ clear
```

```
➜  ~ chmod u=rw,g=r,o=--- passwd
➜  ~ ls -la passwd
-rw-r-----  1 rosenivanov  staff  9344 Jan  2 16:03 passwd
➜  ~
```

1. ➜ ~ sudo chown engineer:engteam passwd 2. ➜ ~ ls -la passwd
3. -rw-r----- 1 engineer engteam 9344 Jan 2 16:03 passwd

Командите dscl, useradd, groupadd и usermod се използват за създаване и управление на потребители и групи. chmod променя правата за достъп до файл, а chown задава собственик и група. ls -l и groups служат за проверка на направените промени. Задачата демонстрира управление на сигурността в системата.

ЗАДАЧА 5 - ОБРАБОТКА НА ТЕКСТ

Вариант за четен номер

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

Формат: ПРОДУКТ;КОЛИЧЕСТВО;ЦЕНА

Извършете следните операции:

1. Изведете в стандартния изход само имената на продуктите, сортирани по азбучен ред.

2. Изведете съдържанието на файла сортирано по количество (възходящ ред). 3. Запишете името на продукта с най-ниска цена в нов файл с име cheapest_item.

4. Изведете последните три реда от файла, сортирани по цена във възходящ ред.

```
1. ➜ ~ cut -d';' -f1 products.info | sort
2. Брашно
3. Захар
4. Олио
5. Ориз
6. Сол
```

```
7. ➜ ~ sort -t';' -k2,2n products.info
8. Захар;40;2.0
9. Олио;50;3.4
10. Брашно;60;1.2
11. Ориз;80;2.5
12. Сол;100;0.6
13. ➜ ~ sort -t';' -k3,3n products.info |
    head -n 1 | cut -d';' -f1 >
    cheapest_item
14. ➜ ~ cat cheapest_item
15. Сол
16. ➜ ~ tail -n 3 products.info | sort
    -t';' -k3,3n
17. Сол;100;0.6
18. Захар;40;2.0
19. Олио;50;3.4
20. ➜ ~ whoami
21. rosenivanov
```

```
➜  ~ cut -d';' -f1 products.info | sort

Брашно
Захар
Олио
Ориз
Сол
➜  ~ sort -t';' -k2,2n products.info

Захар;40;2.0
Олио;50;3.4
Брашно;60;1.2
Ориз;80;2.5
Сол;100;0.6
➜  ~ sort -t';' -k3,3n products.info | head -n 1 | cut -d';' -f1 > cheapest_item

➜  ~ cat cheapest_item
Сол
➜  ~ tail -n 3 products.info | sort -t';' -k3,3n

Сол;100;0.6
Захар;40;2.0
Олио;50;3.4
➜  ~ whoami
rosenivanov
➜  ~
```

Командите `cut`, `sort`, `head` и `tail` се използват за извличане, сортиране и филтриране на текстови данни. Пренасочването `>` записва резултат в нов файл. Така се обработват структурирани данни без нужда от програмен език. Задачата показва силата на текстовите инструменти в Linux.

ЗАДАЧА 6 - ТЪРСЕНЕ НА ФАЙЛОВЕ

Задача: Работа с командата find

Вариант за четен номер

Използвайте командата find, за да извършите следните операции във вашата домашна директория и всички нейни поддиректории:

1. Изведете броя на всички файлове, чиито имена съдържат думата "project". 2. Изведете броя на всички директории, чиито имена започват с "data". 3. Намерете и изведете информация за най-големия .txt файл.
4. Изведете броя на всички .sh файлове, които съдържат думата "backup" в името си.
5. Намерете и изведете пълния път до всички файлове, които имат разширение .log и са били модифицирани през последните 2 дни.
6. Изведете информация за 3-те най-големи файла с разширения .pdf или .docx.

```
1. ➜ ~ mkdir zad6-test
2. ➜ ~ cd zad6-test
3. ➜ zad6-test touch project-file.txt
4. ➜ zad6-test touch other-file.txt 5.
➜ zad6-test find . -type f -iname
'*project*' | wc -l
6. 1
7. ➜ zad6-test touch projects.txt 8. ➜
zad6-test find . -type f -iname
'*project*' | wc -l
9. 2
10. ➜ zad6-test
```

```
1.➜ zad6-test mkdir data-science
2.➜ zad6-test mkdir data
3.➜ zad6-test mkdir other-for-test
4.➜ zad6-test mkdir test-again
```

```
5.➜ zad6-test find . -type d -iname 'data*'
   | wc -l
6. 2
1. ➜ zad6-test find . -type f -name '*.txt'
-exec ls -lh {} + | sort -k5 -h | tail -n 1
2. -rw-r--r-- 1 rosenivanov staff 69B Jan 3
01:11 ./other-file.txt
3. ➜ zad6-test cat other-file.txt
4. asgasg
5. asg
6. asg
7. asg
8. asg
9. asg
10. asg
11. asg
12. sd
13. gdf
14. hg
15. fdh
16. dfh
17. dfh
18. df
19. h
20. dfh
21. df

1. ➜ zad6-test touch backup.sh
2. ➜ zad6-test touch backup-file.sh 3.
➜ zad6-test touch other-sh-file.sh 4.
➜  zad6-test  touch  other-sh-file2.sh
5.➜ zad6-test find . -type f -iname
```

```
'*backup*.sh' | wc -l
6. 2
```

```
1. ➜ zad6-test touch file1.log
2. ➜ zad6-test touch file2.log
3. ➜ zad6-test touch file3.log
4. ➜ zad6-test find . -type f -name '*.log'
   -mtime -2
5. ./file3.log
6. ./file2.log
7. ./file1.log
```

```
1. ➜ zad6-test touch some.pdf
2. ➜ zad6-test touch some.docx
3. ➜ zad6-test nano some.pdf
4. ➜ zad6-test nano some.docx
5. ➜ zad6-test find . -type f \( -name '*.pdf'
   -o -name '*.docx' \) -exec ls -lh {} + |
   sort -k5 -h | tail -n 3
6. -rw-r--r-- 1 rosenivanov staff 48B Jan 3
   01:18 ./some.docx
7. -rw-r--r-- 1 rosenivanov staff 54B Jan 3
   01:18 ./some.pdf
8. ➜ zad6-test wc some.pdf & wc some.docx
9. [1] 8688
10. 14 14 54 some.pdf 11. [1] + 8688
done wc some.pdf 12. 13 12 48 some.docx
```

Командата `find` се използва за търсене на файлове и директории по име, тип, размер и време на
промяна. `wc -l` брои намерените резултати, а

`ls -lh` показва детайлна информация за файловете. Комбинирането с `sort` позволява намиране на
най-големите файлове. Целта е ефективно търсене в файловата система.

ЗАДАЧА 7 - РЕГУЛЯРНИ ИЗРАЗИ

Вариант за четен номер

Използвайте командата grep, за да извършите следните операции с файла /usr/share/dict/words:

1. Изведете броя на думите, които започват с ‘c’ и завършват на ‘t’.
2. Изведете броя на думите, които съдържат точно шест малки латински букви.
3. Изведете броя на думите, които не съдържат точно шест малки латински букви. 4. Изведете броя на думите, които не започват с ‘m’ или ‘n’.
5. Запишете във файла selected_words всички думи, които започват с главна латинска буква и завършват с цифра.

```
1. ➜ ~ grep -E '^c.*t$' /usr/share/dict/words
   | wc -l
2. 1169
3. ➜ ~ grep -E '^[a-z]{6}$'
   /usr/share/dict/words | wc -l
4. 15073
5. ➜ ~ grep -Ev '^[a-z]{6}$'
   /usr/share/dict/words | wc -l
6. 220903
7. ➜ ~ grep -E '^[^mn]' /usr/share/dict/words
   | wc -l
8. 219161
9. ➜ ~ grep -E '^[A-Z].*[0-9]$'
   /usr/share/dict/words > selected_words
10. ➜ ~ cat selected_words
11. ➜ ~ Няма нищо в selected_words, защото
```

```
нямаме такива думи във файла : (
```

Командата `grep` с регулярни изрази търси думи по зададени шаблони. Опциите `-E` и `-v` позволяват разширени изрази и отрицателно търсене. `wc -l` брои съвпаденията, а пренасочването записва резултати във файл. Задачата упражнява работа с текст и регулярни изрази.

ЗАДАЧА 8 - УПРАВЛЕНИЕ НА ПРОЦЕСИТЕ

Вариант за четен номер

1. Създайте три процеса sleep във фонов режим — с аргументи 200, 250 и 300 секунди.
2. Покажете всички процеси, стартирани от текущия шел.
3. Спрете временно процеса, който е стартиран първи.
4. Върнете го обратно във фонов режим и продължете изпълнението му. 5. Стартирайте процеса gedit с приоритет NI = 8.
6. Променете приоритета на процеса gedit на NI = 0, докато е в изпълнение.
7. Стартирайте процеса xeyes с по-висок приоритет (NI = -4) – използвайте root достъп.
8. Покажете всички процеси, сортирани по приоритет.
9. Създайте нов потребител maria и влезте от нейно име.
10. От името на maria стартирайте два процеса sleep (по 120 секунди). 11. Като root, прекратете всички процеси на потребителя maria.
12. Изведете общия брой процеси, създадени от вашия текущ потребител. 13. Изведете PID и името на процеса, който използва най-много CPU време. 14. Стартирайте процеса gnome-calculator и покажете броя на нишките, които той създава.
15. Изпратете сигнал SIGTERM към всички процеси, чиито имена завършват на d.

```
1. ➜ ~ sleep 200 &
2. [1] 8911
3. ➜ ~ sleep 250 &
4. [2] 8918
5. ➜ ~ sleep 300 & // край зад1
6. [3] 8925
7. ➜ ~ jobs -l // зад2
```

```
8. [1] 8911 running sleep 200 9. [2]
- 8918 running sleep 250
10. [3] + 8925 running sleep 300
11. ➜ ~ kill -STOP %1 // зад 3

12. [1] + 8911 suspended (signal) sleep 200
13. ➜ ~ kill -CONT %1 // зад 4
14. ➜ ~ jobs -l
15. [1] + 8911 running sleep 200 16.
[2] 8918 running sleep 250 17. [3] -
8925 running sleep 300 18. ➜ ~ nice
-n 8 gedit & // зад 5 19. [1] 13742
20. ➜ ~ sudo renice 0 -p 13742 // зад 6
21. ➜ ~ sudo nice -n -4 xeyes & / 7 22.
[4]
15938
23. ➜ ~ ps -eo pid,pri,comm | sort -k2 -n
  //8
24. ➜ ~ sudo su maria / 9
25. maria@Rosen Ivanov macbookpro16 %
26. ➜ ~ sudo pkill -u maria / 10 27.
maria@Rosen Ivanov macbookpro16 % 28.
 [2] + terminated sleep 120
29. maria@Rosen Ivanov macbookpro16 %
30. [1] + terminated sleep 120
31. ➜ ~ ps -u maria | wc -l // 11
32. 2
33.
34. ➜ ~ ps -u $(whoami) | wc -l // 12
35. 479
36. ~ ps -eo pid,comm,%cpu | sort -k3 -nr |
  head -n 1 // 13
```

```
37. 8031 /Applications/Di 14.5
38. ➜ ~ open -a Calculator & // 14 39. ➜ ~
ps -M $(pgrep Calculator) 40. USER PID TT
%CPU STAT PRI STIME UTIME COMMAND
41. rosenivanov 18081 ?? 0.0 S 46T
  0:00.15 0:01.14 /System/Ap
42. 18081 0.0 S 46T 0:00.01 0:00.01
43. 18081 0.0 S 4T 0:00.00 0:00.00
44. 18081 0.0 S 4T 0:00.00 0:00.00
45. ➜ ~ pkill -f '[^ ]*d$' / 15
```

Командите `sleep`, `jobs`, `kill`, `nice` и `renice` се използват за създаване и контрол на процеси. `ps` и `top` извеждат информация за текущите процеси и техния приоритет. `pkill` прекратява процеси по потребител или име. Задачата демонстрира управление на процеси и системни ресурси.

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++).

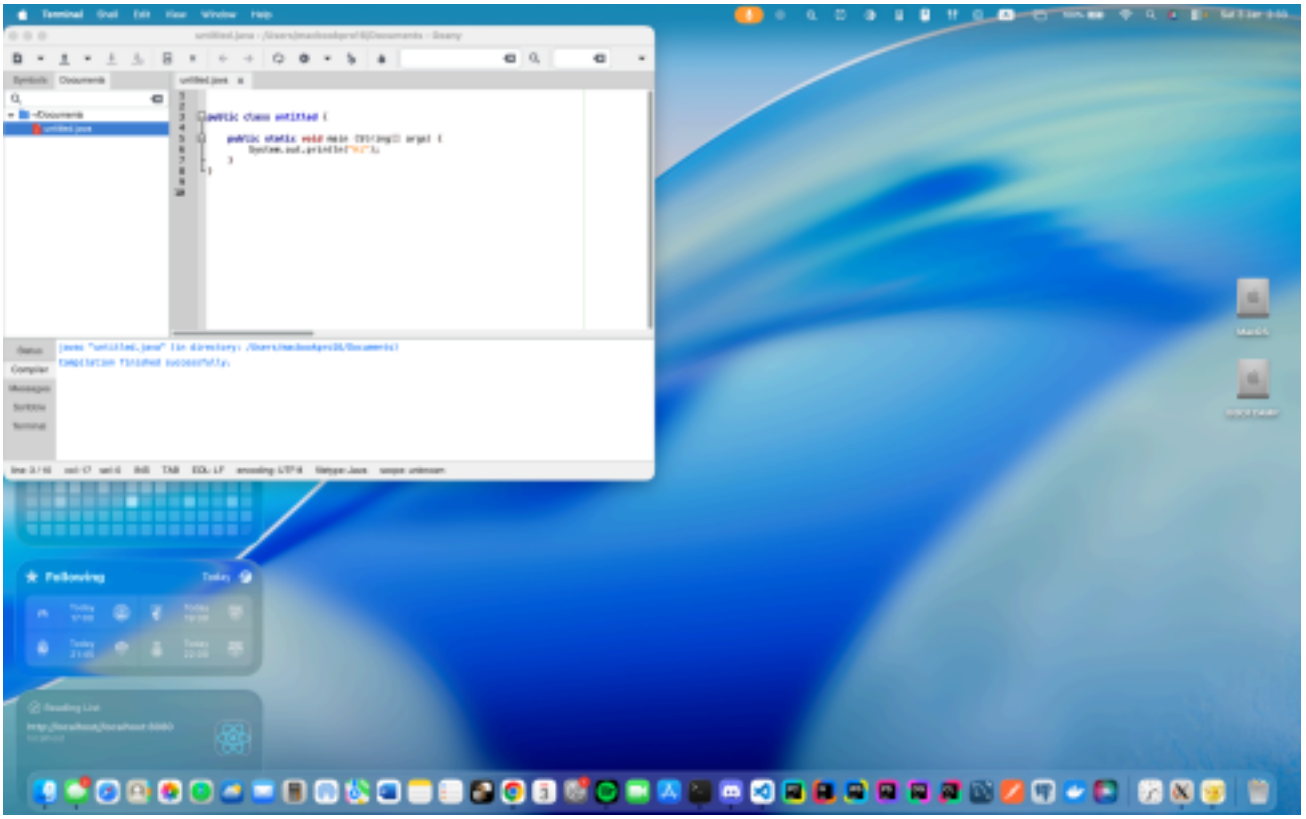
Въведете кратка програма, която извежда съобщение на екрана.

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.

```
1. ➜ ~ brew install geany
2. ➜ ~ open -a geany
```



```
1. ➜ ~ brew remove geany
```

Командата `brew install` инсталира приложението Geany чрез пакетния мениджър. `open -a` стартира програмата от терминала. `brew remove` деинсталира пакета от системата.
Задачата показва управление на софтуер чрез команден интерфейс.

ЗАДАЧА 10 - РАБОТА С АРХИВИ

Вариант за четен номер

1.1 Създайте архив с име documents_backup.tar от всички файлове и директории, намиращи се в директорията Documents на вашата домашна директория. 1.2 Разархивирайте архива documents_backup.tar в директорията /tmp. 1.3 Добавете всички файлове с разширение .pdf от текущата директория към архива files_archive.tar.

1.4 Изведете списък на всички файлове и поддиректории, съдържащи се в архива documents_backup.tar.

1.5 Изведете само имената на файловете от архива documents_backup.tar, които съдържат думата „report“ или „data“ в името си.

```
1. ➜ ~ tar -cvf documents_backup.tar
   ~/Documents // 1
2. ➜ ~ tar -xvf documents_backup.tar -C
   /tmp
3. ➜ ~ tar -cvf files_archive.tar *.pdf
4. ➜ ~ tar -tvf documents_backup.tar 5.
➜ ~ tar -tf documents_backup.tar | grep
-E 'report|data'
```

Командата `tar` се използва за създаване, разархивиране и преглед на архиви. Опциите `-c`, `-x`, `-t` и `-v` определят съответно създаване, извличане и показване на съдържание. `grep` филтрира файлове по име в архива. Задачата упражнява архивиране и работа с резервни копия.