

Студент (име и презиме):
 Број индекса:
 2. март 2026.

Факултет инжењерских наука
 Практикум из рачунарских алата
 Испит

Задачи:

У свим задацима важи претпоставка да се наредбе/програми/скрипте извршавају у **bash** окружењу. Наредбе које извршавају корисници обавезно започињати са **\$**, а које извршава суперкорисник са **#**.

Задатак 1 – [20%]

- [2%] — Исписати корисничко име корисника пријављеног на систем. решење _____
- 1 `$ whoami` решење _____
- [2%] — Исписати путању до кућног (HOME) директоријума пријављеног корисника. решење _____
- 1 `$ echo $HOME или $ echo ~` решење _____
- [2%] — Једном наредбом хијерархијски пронаћи све датотеке и (под)директоријуме у тренутном радном директоријуму који имају подешен **sticky bit**. решење _____
- 1 `$ find . -perm -1000` решење _____
- [2%] — На командној линији потражити помоћ за наредбу **wc**. решење _____
- 1 `$ wc --help` решење _____
- [2%] — У једној линији исписати број карактера у реченици: Астал да блиста, бејби! решење _____
- 1 `$ echo Астал да блиста, бејби! | wc -m` решење _____
- [2%] — Користећи наредбу **sed** у текстуалној датотеци **испит.txt** заменити сва појављивања речи команда речју наредба. решење _____
- 1 `$ sed -i 's/команда/наредба/g' испит.txt` решење _____
- [2%] — Упоредити садржај датотеке **build_25** са садржајем датотеке **build_26** (претпоставити да се обе налазе у тренутном радном директоријуму). решење _____
- 1 `$ diff build_25 build_26` решење _____
- [2%] — Једном наредбом, у једној линији направити директоријум **Мој_рођак_са_села** у тренутном радном директоријуму, тако да садржи и поддиректоријуме **Сезона1** и **Сезона2**. решење _____
- 1 `$ mkdir -p Мој_рођак_са_села/Сезона{1,2}` решење _____
- [2%] — Преместити све датотеке из тренутног директоријума чији назив почиње са **test** у директоријум **/tmp/old**, али само ако су датотеке у тренутном директоријуму новије од оних у **/tmp/old** или ако не постоје у **/tmp/old** (режим ажурирања). решење _____
- 1 `$ mv -u test* /tmp/old` решење _____
- [2%] — Исписати садржај датотеке **студенти.txt** тако да све линије буду нумерисане. решење _____
- 1 `$ cat -n студенти.txt` решење _____

Задатак 2 – [60%]

[15%] — Написати `bash` скрипту за сигурно копирање датотеке. Скрипта прима тачно два аргумента: први је изворна путања до датотеке коју треба копирати, а други је одредишни директоријум у који треба на сигуран начин копирати датотеку. Потребно је одрадити следеће:

- Проверити да ли су прослеђена тачно два аргумента. Ако нису, обавестити корисника и прекинути извршавање скрипте са статусом грешке.
- Проверити да ли је први аргумент постојећа датотека, а други аргумент постојећи директоријум. Ако барем једно није задовољено, обавестити корисника и прекинути извршавање скрипте са статусом грешке.
- Ако у одредишном директоријуму већ постоји датотека са истим именом као датотека коју треба копирати, потребно ју је преименовати додавањем екстензије `.old` и о томе обавестити корисника.
- Копирати изворну датотеку у одредишни директоријум и обавестити корисника о успешном копирању.

решење

```

1  #!/usr/bin/env bash
2
3  if [ $# -ne 2 ]; then
4      echo "Неисправан унос!"
5      echo "Употреба: $0 <изворна_датотека> <одредишни_директоријум>"
6      exit 1
7  fi
8
9  src="$1"
10 dest_dir="$2"
11 filename=$(basename "$src")
12 dest_file="$dest_dir/$filename"
13
14 if [ ! -f "$src" ]; then
15     echo "Грешка: Изворна датотека $src не постоји."
16     exit 2
17 fi
18
19 if [ ! -d "$dest_dir" ]; then
20     echo "Грешка: Одредиште $dest_dir није директоријум."
21     exit 3
22 fi
23
24 if [ -f "$dest_file" ]; then
25     mv "$dest_file" "${dest_file}.old"
26     echo "Постојећа датотека $dest_file је преименована у ${filename}.old."
27 fi
28
29 cp "$src" "$dest_file"
30 echo "Датотека је успешно копирана."
31
32
33
34
35
36
37
38
39
40
41
42
43
44

```

решење

[25%] — Написати `bash` скрипту која има улогу разврставања текстуалних датотека и скрипти из директоријума `./tmp` у зависности од старости датотеке. Скрипта не прима аргументе већ након покретања треба редом да прочита два корисничка уноса: први је врста датотеке и исправни уноси су `text` или `script`, а други је старост и исправни уноси су `new` или `old`. Пре сваког уноса упутити корисника шта се од њега тражи, а након уноса проверити исправност истог и уколико није исправан, обавестити корисника и прекинути извршавање скрипте са статусом грешке. Након исправног корисничког уноса скрипта разврстава датотеке на следећи начин:

- Ако је први унос `text`, а други `new` потребно је све датотеке са екстензијом `.txt` које су новије од 10 дана пребацити у директоријум `./text/new` (ако директоријум `text` и/или његов поддиректоријум `new` не постоје потребно их је направити).
- Ако је први унос `script`, а други `new` потребно је све датотеке са екстензијом `.py` и `.sh` које су новије од 10 дана пребацити у директоријум `./script/new` (ако директоријум `script` и/или његов поддиректоријум `new` не постоје потребно их је направити).
- Ако је први унос `text`, а други `old` потребно је све датотеке са екстензијом `.txt` које су старије од 10 дана пребацити у директоријум `./text/old` (ако директоријум `text` и/или његов поддиректоријум `old` не постоје потребно их је направити).
- Ако је први унос `script`, а други `old` потребно је све датотеке са екстензијом `.py` и `.sh` које су старије од 10 дана пребацити у директоријум `./script/old` (ако директоријум `script` и/или његов поддиректоријум `old` не постоје потребно их је направити).

решење

```

1  #!/usr/bin/env bash
2
3  move_text_new() {
4      dest="./text/new"
5      mkdir -p "$dest"
6      find ./tmp -type f -name '*.txt' -mtime -10 -exec mv {} "$dest" \;
7  }
8
9  move_text_old() {
10     dest="./text/old"
11     mkdir -p "$dest"
12     find ./tmp -type f -name '*.txt' -mtime +10 -exec mv {} "$dest" \;
13 }
14
15 move_script_new() {
16     dest="./script/new"
17     mkdir -p "$dest"
18     find ./tmp -type f \( -name '*.py' -o -name '*.sh' \) -mtime -10 \
19 -exec mv {} "$dest" \;
20 }
21
22 move_script_old() {
23     dest="./script/old"
24     mkdir -p "$dest"
25     find ./tmp -type f \( -name '*.py' -o -name '*.sh' \) -mtime +10 \
26 -exec mv {} "$dest" \;
27 }
28
29 echo "Унесите врсту датотеке (text или script):"
30 read type
31 if [[ "$type" != "text" && "$type" != "script" ]]; then
32     echo "Грешка: неисправан унос. Дозвољено је само: text или script."
33     exit 1
34 fi
35
36 echo "Унесите старост (new или old):"
37 read age
38 if [[ "$age" != "new" && "$age" != "old" ]]; then
39     echo "Грешка: неисправан унос. Дозвољено је само: new или old."
40     exit 1

```

```

41  fi
42
43  move_`${type}`_`${age}`
44
45
46
47
48
49
50
51
52
53
54

```

решење

[20%] — У постојећем директоријуму /dom налазе се три поддиректоријума: `sobe`, `kancelarija` и `hol`. Потребно је направити следеће кориснике и подесити им лозинке: `slavica`, `slobo`, `marta` и `bozidar`. Направити две групе: `osoblje` у којој су корисници `slavica` и `slobo` и групу `stanari` у којој су `marta` и `bozidar`. Уредити дозволе на систему тако да буде задовољено:

- Корисник `slavica` је управница дома па је због тога треба подесити за власника сва три директоријума (`sobe`, `kancelarija` и `hol`).
- Власничка група за директоријум `sobe` је `stanari`.
- Власничка група за директоријум `hol` је `osoblje`.
- Над директоријумом `kancelarija` само `slavica` као власник директоријума има сва права, док остали немају никаква права.
- Над директоријумом `hol` сва права додати особљу, а осталима одузети сва права осим читања. Обезбедити да власничка група свих новонаправљених датотека у директоријуму `hol` одговара власничкој групи директоријума `hol`.
- Над директоријумом `sobe` сва права додати станарима, а осталима одузети сва права. Унутар директоријума `sobe` онемогућити станаре да обришу датотеке ако нису власници истих.

решење

```

1  # groupadd osoblje
2  # groupadd stanari
3
4  # useradd slavica
5  # usermod -aG osoblje slavica
6  # passwd slavica
7
8  # useradd slobo
9  # usermod -aG osoblje slobo
10 # passwd slobo
11
12 # useradd marta
13 # usermod -aG stanari marta
14 # passwd marta
15
16 # useradd bozidar
17 # usermod -aG stanari bozidar
18 # passwd bozidar
19
20 # chown slavica:slavica /dom/kancelarija
21 # chown slavica:osoblje /dom/hol
22 # chown slavica:stanari /dom/sobe
23
24 # chmod 700 /dom/kancelarija
25 # chmod 2774 /dom/hol
26 # chmod 1770 /dom/sobe
27

```

28
29
30
31
32

решење

Задатак 3 – [20%]

[2%] — Додати удаљени репозиторијум под називом `origin` и URL-ом `git@github.com:original/repo.git`.

1 `$ git remote add origin git@github.com:original/repo.git`
решење

[2%] — Излистати све удаљене репозиторијуме повезане са локалним репозиторијумом.

1 `$ git remote`
или
2 `$ git remote -v`
3
решење

[2%] — Комитовати измене са произвољном поруком без експлицитног додавања у индекс. Сматрати да су датотеке у којима се налазе измене већ праћене.

1 `$ git commit -a -m "Моја бесмислена порука."`
решење

[2%] — Датотеку `info.txt` вратити из индекса у радни директоријум.

1 `$ git restore --staged info.txt`
или
2 `$ git checkout info.txt`
3
решење

[2%] — Излистати комитове у којима се мењала датотека `README.md`.

1 `$ git log -- README.md`
решење

[2%] — Исписати разлику стања репозиторијума у односу на претходни комит.

1 `$ git diff`
решење

[2%] — Приказати све детаље комита са контролном сумом `fb5979`.

1 `$ git show fb5979`
решење

[2%] — Клонирати удаљени репозиторијум са адресе `git@github.com:elektrotehnika/pra`.

1 `$ git clone git@github.com:elektrotehnika/pra`
решење

[2%] — Обрисати грану `развој`.

1 `$ git branch -d развој`
решење

[2%] — Како гласи садржај `.gitignore` датотеке ако желимо занемарити све датотеке унутар директоријума `build/`?

1 `build/`
решење