

Студент (име и презиме):  
 Број индекса:  
 18. фебруар 2026.

Факултет инжењерских наука  
 Практикум из рачунарских алата  
 Испит

**Задаци:**

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% укупно]**

- [2%] — Излистати садржај директоријума `/usr/bin` и испис преусмерити у датотеку `програми.txt` у тренутном радном директоријуму.

1 `$ ls /usr/bin > програми.txt` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — Исписати број линија у датотеци направљеној у претходној питаљници.

1 `$ cat програми.txt | wc -l` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — Пронаћи где се налази извршна датотека наредбе `python3`.

1 `$ which python3` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — **Нерекурзивно** (само на првом нивоу хијерархије) пронаћи празне директоријуме у кућном (HOME) директоријуму.

1 `$ find ~ -maxdepth 1 -type d -empty` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — Приказати последњих 10 наредби које је корисник унео у терминал.

1 `$ history | tail -10` решење \_\_\_\_\_  
 2 или \_\_\_\_\_  
 3 `$ history 10` \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — Направити симболички линк са називом **пречица** који показује на замишљену датотеку `/var/www/html/index.html`.

1 `$ ln -s /var/www/html/index.html пречица` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — У проширеном формату излистати садржај тренутног радног директоријума и, рекурзивно, свих његових поддиректоријума.

1 `$ ls -lR` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — Пошто већ не постоји, додати нову наредбу `up` која олакшава навигацију кроз хијерархију датотечног система и када се изврши оде се у родитељски директоријум.

1 `$ alias up='cd ..'` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [4%] — Шта је потребно урадити да наредба `up` из претходног питаљнице буде увек доступна, а не само у току тренутне сесије терминала? Одговорити у највише две реченице и што прецизније.

1 Потребно је: \_\_\_\_\_  
 2 1. Наредбу `alias up='cd ..'` додати у `~/.bashrc` датотеку и сачувати.  
 3 2. Извршити наредбу `source ~/.bashrc` или се поново пријавити на систем.  
 \_\_\_\_\_ решење \_\_\_\_\_

## Задатак 2 – [55%]

- [10%] — Написати **bash** скрипту која прима барем један аргумент и за сваки од њих исписује број карактера који садржи (дужину) у формату **аргумент: број\_карактера**. Уколико корисник не унесе ниједан аргумент, обавестити га и прекинути извршавање скрипте.

решење

```

1  #!/usr/bin/env bash
2
3  if [[ $# -lt 1 ]]; then
4      echo "Грешка: Морате проследити барем један аргумент."
5      exit 1
6  fi
7
8  for argument in $@; do
9      echo "$argument: ${#argument}"
10  done
11
12
13
14
15

```

решење

- [25%] — Написати **bash** скрипту која као једини аргумент прима датум и време вашег рођења у формату **ГГГГММДДссмм**, где су:

- ГГГГ - година (од 2000 до 2010),
- ММ - месец (од 01 до 12),
- ДД - дан (од 01 до 31, независно од месеца – да не компликујемо превише),
- сс - сати (од 00 до 23),
- мм - минути (од 00 до 59),

а затим у тренутном радном директоријуму (и свим поддиректоријумима) мења временски жиг свих датотека чији сте власник Ви, односно корисник који покреће скрипту. Пре промене временског жига датотека извршити проверу унетог аргумента према горе наведеним правилима и уколико се догоди погрешан унос обавестити корисника и прекинути извршавање скрипте.

решење

```

1  #!/usr/bin/env bash
2
3  # ----- (I начин) -----
4
5  # Провера да ли је тачно један аргумент прослеђен скрипти
6  if [[ $# -ne 1 ]]; then
7      echo "Грешка: Неисправан број аргумената! \
8  Скрипта прима тачно један аргумент."
9      exit 1
10  fi
11
12  VREME=$1
13  regex="^(200[0-9]|2010)(0[1-9]|1[0-2])(0[1-9]|1[12])[0-9]|3[01])\
14  (0[0-9]|1[0-9]|2[0-3])([0-5][0-9])$"
15
16  # Провера да ли се унос поклапа са регуларним изразом
17  if ! [[ "$VREME" =~ $regex ]]; then
18      echo "Грешка: Унос '$VREME' није у исправном формату ГГГГММДДссмм \
19  или су вредности ван дозвољеног опсега."
20      exit 2
21  fi
22
23  find . -type f -user "$(whoami)" -exec touch -t "$VREME" '{}' '+'
24
25

```

```

26 # ----- (II начин) -----
27
28 # Провера да ли је тачно један аргумент прослеђен скрипти
29 if [[ $# -ne 1 ]]; then
30     echo "Грешка: Неисправан број аргумената! \
31 Скрипта прима тачно један аргумент."
32     exit 1
33 fi
34
35 VREME=$1
36
37 # Провера да ли унос има тачно 12 цифара
38 if ! [[ "$VREME" =~ ^[0-9]{12}$ ]]; then
39     echo "Грешка: Унос мора имати тачно 12 цифара."
40     exit 2
41 fi
42
43 # Издавајање делова из унетог аргумента помоћу 'cut' наредбе
44 godina_str=$(echo "$VREME" | cut -c 1-4)
45 mesec_str=$(echo "$VREME" | cut -c 5-6)
46 dan_str=$(echo "$VREME" | cut -c 7-8)
47 sati_str=$(echo "$VREME" | cut -c 9-10)
48 minuti_str=$(echo "$VREME" | cut -c 11-12)
49
50 # Претварање стрингова у бројеве помоћу аритметичке експанзије
51 # Ово исправно обрађује водеће нуле (нпр. "08" постаје 8)
52 godina=$((godina_str))
53 mesec=$((mesec_str))
54 dan=$((dan_str))
55 sati=$((sati_str))
56 minuti=$((minuti_str))
57
58 # Провера да ли су вредности унутар дозвољеног опсега
59 if (( godina < 2000 || godina > 2010 )) || \
60 (( mesec < 1 || mesec > 12 )) || \
61 (( dan < 1 || dan > 31 )) || \
62 (( sati < 0 || sati > 23 )) || \
63 (( minuti < 0 || minuti > 59 )); then
64     echo "Грешка: Унос '$VREME' није у исправном формату ГГГГММДДссмм \
65 или су вредности ван дозвољеног опсега."
66     exit 3
67 fi
68
69 find . -type f -user "$(whoami)" -exec touch -t "$VREME" '{} ' '+'
70

```

решење

[20%] — Потребно је у директоријуму /cis (сматрати да **не** постоји) направити два поддиректоријума који одговарају предметима из прве године: **pra** и **programiranje**. За оба предмета направити и одговарајућу корисничку групу на систему.

На предметима ради 5 корисника: **vlada**, **ivan**, **isidora**, **uros** и **djole**. Направити наведене кориснике на систему и подесити лозинке за сваког од њих. Уредити дозволе на систему тако да буде задовољено:

- Корисник **vlada** као управник Центра за интегрисане системе (ЦИС) је власник оба директоријума (**pra** и **programiranje**).
- Корисници **isidora** и **uros** имају право читања оба директоријума (у својству осталих корисника система) али права писања и извршавања имају само у директоријуму **programiranje**.
- Корисници **ivan** и **djole** имају право читања оба директоријума (у својству осталих корисника система) али права писања и извршавања имају само у директоријуму **pra**.

Обезбедити да све новонастале датотеке унутар директоријума /cis/pra и /cis/programiranje припадају групи направљеној за одговарајући предмет.

решење

```

1  # mkdir -p /cis/{pra, programiranje}
2
3  # groupadd pra
4  # groupadd programiranje
5
6  # useradd vlada
7  # passwd vlada
8
9  # useradd isidora
10 # usermod -a -G programiranje isidora
11 # passwd isidora
12
13 # useradd uros
14 # usermod -a -G programiranje uros
15 # passwd uros
16
17 # useradd ivan
18 # usermod -a -G pra ivan
19 # passwd ivan
20
21 # useradd djole
22 # usermod -a -G pra djole
23 # passwd djole
24
25 # chown vlada:pra /cis/pra
26 # chown vlada:programiranje /cis/programiranje
27
28 # chmod 2774 /cis/pra
29 # chmod 2774 /cis/programiranje
30
31
32
33
34
35
36
37
38
39

```

решење

### Задатак 3 – [25%]

[2%] — Иницијализовати Гит репозиторијум у тренутном радном директоријуму.

решење

```

1  $ git init

```

решење

[2%] — Подесити Гит корисничко име на Ваш надимак, али само локално тј. за тренутни репозиторијум.

решење

```

1  $ git config user.name надимак

```

решење

[2%] — Претпоставимо да постоји једна празна датотека README.md у радном директоријуму. Додати је у индекс и комитовати са поруком "Додавање прве датотеке."

решење

```

1  $ git add README.md
2  $ git commit -m "Додавање прве датотеке."

```

решење

- [2%] — Претпоставимо да сте грешком комитовали празну датотеку и да сте сада у исту додали неки текст и да су измене додате у индекс. Комитујте измене али тако да у историји комитова **не** постоји додатни комит који се односи на додавање текста у датотеку, већ да изгледа као да је и текст додат у претходном комиту. Порука комита треба да остане иста као у претходном.

1 `$ git commit --amend --no-edit` решење \_\_\_\_\_  
 решење \_\_\_\_\_

- [2%] — Потражити помоћ за `git log` наредбу.

1 `$ git log --help` решење \_\_\_\_\_  
 2 или \_\_\_\_\_  
 3 `$ git help log` решење \_\_\_\_\_

- [2%] — У скраћеном формату уз приказ гранања исписати историју свих комитова, са свих грана.

1 `$ git log --oneline --all --graph` решење \_\_\_\_\_  
 решење \_\_\_\_\_

- [2%] — Направити **локални** Гит алијас на наредбу из претходне питалице тако да исти испис добијете наредбом `git hist`.

1 `$ git config alias.hist 'log --oneline --all --graph'` решење \_\_\_\_\_  
 решење \_\_\_\_\_

- [2%] — Исписати статус репозиторијума у скраћеном формату али тако да се виде и све непраћене (*untracked*) датотеке унутар непраћених директоријума.

1 `$ git status -su` решење \_\_\_\_\_  
 решење \_\_\_\_\_

- [2%] — Означити последњи комит ознаком (тагом) `beta`, тако да се додата ознака направи као засебан Гит објекат. Порука ознаке треба бити "Бета верзија."

1 `$ git tag -a beta -m "Бета верзија."` решење \_\_\_\_\_  
 решење \_\_\_\_\_

- [2%] — Вратити стање датотеке `README.md` у радном директоријуму на стање из комита са контролном сумом `3abf45`.

1 `$ git checkout 3abf45 -- README.md` решење \_\_\_\_\_  
 2 или \_\_\_\_\_  
 3 `$ git restore --source 3abf45 README.md` решење \_\_\_\_\_  
 решење \_\_\_\_\_

- [5%] — Потребно је:

- Направити грану `testiranje` и преbacити се на исту једном наредбом у истој линији.
- Додати у индекс и комитовати нову датотеку `test.py`.
- Вратити се на грану `master` без експлицитног навођења њеног имена.
- Спојити се са граном `testiranje`.
- Обрисати грану `testiranje`.

\_\_\_\_\_ решење \_\_\_\_\_  
 1 `$ git checkout -b testiranje`  
 2 `$ git add test.py`  
 3 `$ git commit -m "Додата датотека за тестирање."`  
 4 `$ git switch -`  
 5 `$ git merge testiranje`  
 6 `$ git branch -d testiranje`  
 7 \_\_\_\_\_  
 8 \_\_\_\_\_  
 9 \_\_\_\_\_  
 10 \_\_\_\_\_ решење \_\_\_\_\_