Първо контролно за курса 'Операционни системи', СУ, ФМИ, 01.04.2017 г.

Име:_____ ФН:_____ Спец.:__ Курс:__ Гр.:__

Задача	1	2	3	4	Общо
получени точки					
максимум точки	30	30	30	35	125

Забележка: За отлична оценка са достатъчни 100 точки!

Задача 1. Напишете серия от команди, извеждащи на екрана само броя на всички обекти във файловата система, чиито собственик е текущият потребител.
Забележка: Във файловата система със сигурност съществуват директории, до които нямате достъп.

Задача 2. Намерете имената на топ 5 файловете в текущата директория с най-много hardlinks.

Задача 3. Напишете серия от команди, които вземат от файла `/etc/passwd` първите имена на студентите от специалност СИ, чиито фамилии завършват на "а". Изведете колко пъти се среща най-често срещаното име и кое е то.
Примерно съдържание на файла:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
s61934:x:1177:504:Mariq Cholakova:/home/SI/s61934:/bin/bash
s61916:x:1178:504:Simeon Aleksandrov:/home/SI/s61916:/bin/bash
s61884:x:1179:504:Ruslan Kapelev:/home/SI/s61884:/bin/bash
s61895:x:1180:504:Zorka Shindova:/home/SI/s61895:/bin/bash
s61899:x:1182:504:Zorka Ivanova, SI, 2kurs, 5gr:/home/SI/s61899:/bin/bash
s81374:x:1117:503:Ivan Kamburov, KN, 2kurs, 7gr:/home/KN/s81374:/bin/bash
s81382:x:1118:503:Teodora Cirkova:/home/KN/s81382:/bin/bash
```

Примерен изход:

```
2 Zorka
```

Задача 4. Всеки от процесите `P`, `Q` и `R` изпълнява поредица от три инструкции:

```
process P          process Q          process R
  p_1                q_1                r_1
  p_2                q_2                r_2
  p_3                q_3                r_3
```

Осигурете чрез семафори синхронизация на `P`, `Q` и `R` така, че инструкция `p_1` да се изпълни преди `q_2` и `r_2`.

Забележка: Решения на задачата с повече от един семафор носят не повече от 20 точки.

Примерни решения

Задача 1.

```
find / -user $(id -un) 2>/dev/null | wc -l
```

Задача 2.

```
ls -l |tail -n +2 |tr -s ' ' |sort -t ' ' -k 2 -rn |head -n 5 | cut -d ' ' -f 9-
```

или

```
find . -maxdepth 1 -type f -printf "%n %f\n" | sort -rn | head -n 5 | cut -d " " -f 2-
```

Задача 3.

```
egrep -i "\b[a-z]* [a-z]*\b.*SI" /etc/passwd \
    | cut -d ":" -f 5 | cut -d " " -f1 | sort | uniq -c | sort -rn | head -n 1
```

или

```
fgrep SI /etc/passwd | cut -d ':' -f 5 | cut -d ',' -f 1 \
    | egrep 'a$' | cut -d ' ' -f 1 | sort | uniq -c |sort -rn | head -n 1
```

или

```
egrep ':[A-Za-z]+ [A-Za-z]*a(:|,).*\/home\/SI' /etc/passwd \
    | cut -d ':' -f 5 | cut -d ' ' -f 1 |sort |uniq -c | sort -rn | head -n 1
```

Задача 4. За синхронизация използваме семафор `t`, инициализираме го с блокиращо начално състяние:

```
semaphore t
t.init(0)
```

Добавяме в кода на процесите `P`, `Q` и `R` синхронизиращи инструкции:

```
process P                process Q                process R
  p_1                      q_1                      r_1
  t.signal()               t.wait()                 t.wait()
  p_2                      t.signal()               t.signal()
  p_3                      q_2                      r_2
                           q_3                      r_3
```

Всяка от инструкциите `q_2` и `r_2` може да се изпълни след като съответният процес премине бариерата `t.wait()`.

Това се случва за пръв път след изпълнението на ред `t.signal()` в процеса `P`, който следва инструкция `p_1`. Така изпълнението на `p_1` преди `q_2` и `r_2` е гарантирано.

Да допуснем, че процесът `Q` преминава през инструкцията си `t.wait()` преди процеса `R`. Веднага след това той изпълнява `t.signal()`, което ще позвали и на `R` да премине през своята инструкция `t.wait()`. Така ще се осигури изпълнението и на двете инструкции `q_2` и `r_2`.

Аналогична е ситуацията, когато `R` преминава през `t.wait()` преди процеса `Q`.