Първо контролно за курса “Операционни системи”, СИ, СУ, ФМИ, 31.03.2019 г.

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

Задача	1 (30)	2 (30)	3 (30)	4 (30)	Общо (120)
получени точки					

Забележка: За всички задачи, в имената на файловете и директориите няма специални символи. Във файловата система може да съществуват директории, до които нямате достъп като обикновен потребител.

Задача 1. Даден е текстовият файл **planets.txt**, който съдържа информация за гравитационно закръглените обекти в дадена слънчева система. На всеки ред има информация за точно един обект в следните колони, разделени с ’**;**’:

- име на обекта
- тип на обекта (един знак)
  - T - земен тип
  - G - газов гигант
  - I - леден гигант
- средно разстояние на обекта до локалната звезда
- маса на обекта (относителна величина)
- обем на обекта (относителна величина)
- плътност (g/cm^3)
- средна орбитална скорост (km/s)

Първият ред във файла е header, който описва имената на колоните.
Данните за обектите не са сортирани.

Намерете и изведете разделени с таб името и масата на обекта, който е **едновременно**:

- най-близкият до локалната звезда
- от същия тип като типа на най-далечният до локалната звезда обект

Примерен входен файл:

```
name;type;distance;mass;volume;density;speed
earth;T;1.00000011;1;1;5.52;29.7859
mars;T;1.52366231;0.107;0.151;3.94;24.1309
saturn;G;9.53707032;95;763.62;0.7;9.6724
mercury;T;0.38709893;0.055;0.056;5.43;47.8725
venus;T;0.72333199;0.815;0.857;5.24;35.0214
jupiter;G;5.20336301;318;1321.3;1.33;13.0697
neptune;I;30.06896348;17;57.747;1.76;5.4778
uranus;I;19.19126393;14.5;63.102;1.3;6.8352
```

Задача 2. Вие сте асистент по ОС. На първото упражнение казвате на студентите да си напишат данните на лист, взимате го и им правите акаунти. След упражнението обаче, забравяте да вземете листа със себе си - сещате се половин час по-късно, когато трябва да въведете имената на студентите в таблица, но за зла беда в стаята вече няма ни помен от листа (вероятно иззет от спешния отряд на GDPR-полицията)

Сещате се, че в началото на упражнението UNIX-часовникът е показвал 1551168000, а в края - 1551176100.

Напишете команда, която изкарва разделени с таб факултетните номера и имената на потребителите от специалност СИ, чиито home директории са променили статуса си (status change time) в зададения времеви интервал.

Приемете, че всички потребители от СИ имат home директории под /home/SI.
Примерен изход:

```
62198   Ivaylo Georgiev
62126   Victoria Georgieva
62009   Denitsa Dobreva
62208   Trayana Nedelcheva
```

Няколко реда от /etc/passwd за справка:

```
s62136:x:1302:503:Alexander Ignatov, SI, 2, 2:/home/KN/s62136:/bin/bash
s62171:x:1031:504:Deivid Metanov:/home/SI/s62171:/bin/bash
s62126:x:1016:504:Victoria Georgieva:/home/SI/s62126:/bin/bash
s62009:x:1170:504:Denitsa Dobreva,SI,3,3:/home/SI/s62009:/bin/bash
s62196:x:1221:504:Elena Tuparova,SI,2,1:/home/SI/s62196:/bin/bash
```

Задача 3. От всички файлове в home директорията на потребителя velin, изведете дълбочината на файл, който:

- има същия inode като този на най-скоро променения файл сред тях
- има минимална дълбочина

Пояснение
Под "дълбочина"да се разбира дълбочина в дървото на файловата система: например файлът /foo/bar/baz има дълбочина 3.

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

```
process P          process Q
  p_1                q_1
  p_2                q_2
  p_3                q_3
```

Осигурете чрез два семафора синхронизация на `P` и `Q` така, че да са изпълнени едновременно следните времеви зависимости:

(1) инструкция `p_1` да се изпълни преди `q_2`
(2) инструкция `q_2` да се изпълни преди `p_3`
(3) инструкция `q_1` да се изпълни преди `p_2`
(4) инструкция `p_2` да се изпълни преди `q_3`

Забележка: За решение с повече семафори ще получите 20 точки.

Успех! ☠

Примерни решения

Задача 1.

```
egrep "^[^;]+;$(tail -n +2 planets.csv | sort -t ';' -k 3 -n | tail -n 1
    | cut -d ';' -f 2);" planets.csv | sort -t ';' -k 3 -n | head -n 1 \
    | awk -F ';' '{print $1"\t"$4}'
```

Задача 2.

```
find /home/SI -maxdepth 1 -type d \
    -printf '%p %C@\n' \
        | awk "$2 > 1551168000 && $2 < 1551176100 { print $1 }" \
        | xargs -I {} grep -F :{}: /etc/passwd \
        | cut -d: -f1,5 | cut -c2- \
| cut -d, -f1 \
| tr : '\t'
```

Задача 3.

```
find ~velin -inum $(find ~velin -type f -printf '%T@ %i\n' 2> /dev/null \
                    | sort -n -k1 \
                    | head -1 \
                    | cut -d ' ' -f2) 2> /dev/null \
    | tr -cd '/\n' | sort | head -1 | grep -o . | wc -l
```

Задача 4. Условията (1) и (3) определят времева среща (randevouz) на процесите след първата им инструкция.

Аналогично, (2) и (4) определят randevouz на процесите след втората им инструкция.

За двате срещи използваме два семафора – `t1` и `t2`, инициализираме ги с блокиращо начално състяние:

```
semaphore t1,t2
t1.init(0)
t2.init(0)
```

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

```
process P                process Q
  p_1                      q_1
  t1.signal()              t2.signal()
  t2.wait()                t1.wait()
  p_2                      q_2
  t1.signal()              t2.signal()
  t2.wait()                t1.wait()
  p_3                      q_3
```

Инструкцията `q_2` ще се изпълни след като броячът на семафора `t1` стане положителен. Това се случва след изпълнението на ред `t1.signal()`, който следва инструкция `p_1`.

Аналогично, инструкцията `p_2` ще се изпълни след като броячът на семафора `t2` стане положителен. Това се случва след изпълнението на ред `t2.signal()`, който следва инструкция `q_1`.

По подобен начин ще се развият събитията и след вторите инструкции.

Лесно се вижда, че след първото randevouz стойностите на броячите в семафорите ще са 0 и процесите коректно ще реализират втората среща със същите семафори.