

```
-- SI-2016-z1

Задача 1.

Напишете shell скрипт, който по подаден един позиционен параметър, ако този параметър е директория, намира всички symlink-ове в нея и под-директориите ѝ с несъществуващ destination.

-- SI-2016-z2

Задача 2.

Напишете shell скрипт, който приема един позиционен параметър - число. Ако скриптът се изпълнява като root, да извежда обобщена информация за общото количество активна памет (RSS - resident set size, non-swapped physical memory that a task has used) на процесите на всеки потребител. Ако за някой потребител обобщеното число надвишава подадения параметър, да изпраща подходящи сигнали за прекратяване на процеса с най-много активна памет на потребителя.

Забележка: Приемаме, че изхода в колоната RSS е число в същата мерна единица, като числото, подадено като аргумент. Примерен формат:

USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  15816  1884 ?        Ss   May12   0:03 init [2]
root         2  0.0  0.0      0     0 ?        S    May12   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    May12   0:00 [ksoftirqd/0]

Алтернативно може да ползвате изхода от ps -e -o uid,pid,rss

-- SI-2016-z3

Задача 3.

Напишете shell скрипт който, ако се изпълнява от root, проверява кои потребители на системата нямат homedir или не могат да пишат в него.

Примерен формат:

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin

-- SI-2017-z1

зад. 1:
-----

Напишете скрипт, който получава задължителен първи позиционен параметър - директория и незадължителен втори - число. Скриптът трябва да проверява подадената директория и нейните под-директории и да извежда имената на:
- при подаден на скрипта втори параметър - всички файлове с брой hardlink-ове поне равен на параметъра
- при липса на втори параметър - всички symlink-ове с несъществуващ destination (счупени simlink-ове)

Заб.: За удобство приемаме, че ако има подаден втори параметър, то той е число.

-- SI-2017-z2

зад. 2:
-----

Напишете скрипт, който приема три задължителни позиционни параметра - директория SRC, директория DST (която не трябва да съдържа файлове) и низ (ABC). Ако скриптът се изпълнява от root потребителя, то той трябва да намира всички файлове в директорията SRC и нейните под-директории, които имат в името си като под-низ ABC, и да ги мести в директорията DST, запазвайки директорийната структура.

Заб.: За удобство приемаме, че не искаме да запазваме метаданните owner, group и permission на директорийната структура, т.е., не ни интересуват тези параметри на новите директории, които скриптът би генерирал в DST.

Пример:
- в SRC (/src) има следните файлове:
    /src/foof.txt
    /src/1/bar.txt
    /src/1/foo.txt
    /src/2/1/foobar.txt
    /src/2/3/barf.txt
- DST (/dst) е празна директория
- задания низ е foo

Резултат:
- SRC:
    /src/1/bar.txt
    /dst/2/1/
    /src/2/3/barf.txt
- DST:
    /dst/foof.txt
    /dst/1/foo.txt
    /dst/2/1/foobar.txt

-- SI-2017-z3

зад. 3:
-----

Напишете скрипт, който ако се изпълнява от root потребителя:

- извежда обобщена информация за броя и общото количество активна памет (RSS - resident set size, non-swaped physical memory that a task has used) на текущите процеси на всеки потребител

- ако процесът с най-голяма активна памет на даден потребител използва два пъти повече памет от средното за потребителя, то скриптът да прекратява изпълнението му по подходящ начин.

-- SI-2017-z4

Задача 1:

Напишете shell script, който получава задължителен първи позиционен параметър - директория и незадължителен втори - име на файл. Скриптът трябва да намира в подадената деиректория:

- броя на symlink-овете, чийто destination не съществува
- Името на симлинка "->" името на destination, ако съществува

Ако е подаден като аргумент файл - да се добавят в него, а ако не е - да се извеждат на екрана.

Примерен изход:
lbaz -> /foo/bar/baz
lqux -> ../ ../qux
...
lqux -> /foo/quux
Broken symlinks: 34

-- SI-2017-z5

Задача 2:

Напишете скрипт, който приема задължителен позиционен аргумент - име на потребител (F00). Ако скриптът се изпълнява от root:
а) да извежда имената на потребителите, които имат повече на брой процеси от F00, ако има такива;
б) да извежда средното време (в секунди), за което са работили процесите на всички потребители на системата (TIME);
в) ако съществуват процеси на F00, които са работили над два пъти от средното време, скриптът да прекратява изпълнението им по подходящ начин

-- SI-2017-z6

Задача 3:
```

Напишете скрипт, който извежда името на потребителския акаунт, в чиято поле директория има най-скоро променен обикновен файл и кой е този файл.
Напишете скрипта с подходящите проверки, така че да бъде валиден инструмент.

-- SI-2018-z3

Нека съществува програма за моментна комуникация (Instant messaging), която записва логове на разговорите в следния формат:

- има определена директория за логове (LOGDIR)
- в нея има директорийна структура от следния вид:

LOGDIR/протокол/акаунт/приятел/

като на всяко ниво може да има няколко екземпляра от съответния вид,
т.е. няколко директории *протокол*, във всяка от тях може да има няколко директории *акаунт*, и във всяка от тях -- няколко директории *приятел*.

- във всяка от директориите *приятел* може да има файлове с имена от вида уууу-мм-дд-hh-мм-ss.txt (година-месец-ден и т.н., спрямо това кога е започнал даден разговор

- всеки такъв файл представлява лог на даден разговор със съответния приятел, като всяка разменена реплика между вас е на отделен ред

- даден идентификатор *приятел* може да се среща няколко пъти в структурата (напр. през различни ваши акаунти сте водили разговори със същия приятел)

Напишете скрипт, който приема задължителен позиционен аргумент - име на лог директория (LOGDIR). Скриптът трябва да извежда десетимата приятели, с които имате най-много редове комуникация глобално (без значение протокол и акаунт), и колко реда имате с всеки от тях.