Контролно 2, курс 'Операционни системи', СУ, ФМИ, 18.05.2016 г.

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

Задача	1	2	3	4	Общо
получени точки					
максимум точки	30	30	30	30	120

Забележка: За отлична оценка са достатъчни 100 точки!

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.

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   0:02 [ksoftirqd/0]
```

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

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
```

Задача 4.

Опишете накратко основните процедури и структури данни, необходими за реализация на семафор.

Каква е разликата между слаб и силен семафор?

Опишете максимално несправедлива ситуация, която може да се получи в избирателна секция, ако на входа на секцията пазач – член на изборната комисия пуска гласоподавателите вътре така:

(1) във всеки момент в секцията може да има най-много двама гласоподаватели.

(2) пазачът работи като слаб семафор.

Решения

Задача 1. Symlinks

```
#!/bin/dash
if [ $# -ne 1 -o ! -d $1 ]; then
    exit 1
fi
find -L $1 -type l
```

или

```
#!/bin/dash
if [ $# -ne 1 -o ! -d $1 ]; then
    exit 1
fi

for i in $(find $1 -type l); do
    dst=$(stat -c %N $i | cut -d "'" -f 4)
    if [ ! -e $(dirname $i)/$dst ]; then
        echo $i
    fi
done
```

или

```
#!/bin/dash
if [ $# -ne 1 -o ! -d $1 ]; then
    exit 1
fi

find $1 -type l | xargs ls -l | while read foo; do
    src=$(echo $foo |tr -s ' ' |cut -d ' ' -f 9)
    dst=$(echo $foo |tr -s ' ' |cut -d ' ' -f 11)

    if [ ! -e $(dirname $src)/$dst ]; then
        echo $src
    fi
done
```

Задача 1. Hardlinks

```
#!/bin/dash
if [ $# -ne 1 -o ! -d $1 ]; then
    exit 1
fi

find $1 -type f | xargs ls -l | awk '$2 != 1 {print $9}'
```

или

```
#!/bin/dash
if [ $# -ne 1 -o ! -d $1 ]; then
    exit 1
fi

for i in $(find $1 -type f); do
    nh=$(stat -c %h $i)
    if [ $nh -ge 2 ]; then
        echo $i
    fi
done
```

Задача 2.

```
#!/bin/bash
if [ $# -ne 1 -o $(id -u) -ne 0 ]; then
    exit 1
fi

users=$(ps -e -o user | tail -n +2 | sort | uniq)

for user in $users; do
    total_rss=0

    while read line; do
        current_pid=$(echo $line|awk '{print $1}')
        current_rss=$(echo $line|awk '{print $2}')
        total_rss=$(expr $total_rss + $current_rss)
    done < <(ps -u $user -o pid,rss | tail -n +2 | sort -n -k 2)

    echo "user:" $user "total_rss:" $total_rss

    if [ $total_rss -gt $1 ]; then
        echo "greater than $1, will kill $current_pid"
        kill -s SIGTERM $current_pid
        sleep 2
        kill -s SIGKILL $current_pid
    fi
done
```

Задача 3.

```
#!/bin/bash
if [ $(id -u) -ne 0 ]; then
    exit 1
fi

cat /etc/passwd | while read line; do
    user=$(echo $line |cut -d':' -f 1)
    uid=$(echo $line |cut -d':' -f 3)
    gid=$(echo $line |cut -d':' -f 4)
    home=$(echo $line |cut -d':' -f 6)

    if [ -z $home ]; then
        echo "$user has no homedir set"
        continue
    fi

    if [ ! -d $home ]; then
        echo "$user has homedir $home which is not a directory"
        continue
    fi

    dirperm=$(ls -ld $home | awk '{print $1}')
    dirowner=$(ls -ld $home | awk '{print $3}')

    if [ $dirowner != $user ]; then
        echo "$user is not owner of $home"
        continue
    fi

    if [ $(echo $dirperm |cut -c 3) != "w" ]; then
        echo "$user (owner) cannot write in $home"
    fi
done
```

Задача 4.

Структурите данни, необходими за реализация на семафор са:

Брояч `cnt`, в който се пази броя на процесите, които могат да бъдат допуснати до ресурса, охраняван от семафора.

Контейнер `Q`, в който се пази информация кои процеси чакат достъп до ресурса.

Процедурите, необходими за реализация на семафор са:

Конструктор `Init(c0:integer)`, който задава начална стойност на брояча `cnt`. Контейнерът `Q` се инициализира да е празен.

Метод `Wait()`, който се ползва при опит за достъп до ресурса (заемане на ресурса). Броячът се намалява с единица и ако стане отрицателен, процесът викащ `Wait()` се блокира, а номерът му се вкарва в контейнера `Q`.

Метод `Signal()`, който се ползва при завършване на достъпа до ресурса (освобождаване на ресурса). Броячът се увеличава с единица и ако `Q` не е празен, един от процесите в него се вади и активира.

Семафорът е *силен*, когато контейнера `Q` е реализиран като обикновена опашка – винаги активираме процеса, блокиран най-рано.

Семафорът е *слаб*, когато контейнера `Q` не е реализиран като обикновена опашка – при изпълнение на `Signal()` активираме процес, който може да не е първи в списъка на чакащите.

Ако пазачът на входа на избирателната секция действа като слаб семафор, може да се получи следната неприятна ситуация:

Първите двама гласоподаватели влизат в секцията, пристига трети гласоподавател (неприятел на пазача) и чака. След него започват да пристигат приятели на пазача и той ги пуска с предимство. Може да се стигне дотам, че третият гласоподавател чака цял ден и гласува последен.

Подобна несправедлива ситуация при достъп до ресурс наричаме *starvation* (гладуване).