### **Задание 1.**

Напишите регулярное выражение для проверки является ли строка IPv4 адресом.

Для тестов можно использовать файл со следующим содержимым, фильтруя вывод с помощью команды grep -E:

192.168.0.1

127.0.0.1

84.345.23.11

88.3A.56.76

224.12.76

**Ответ:**

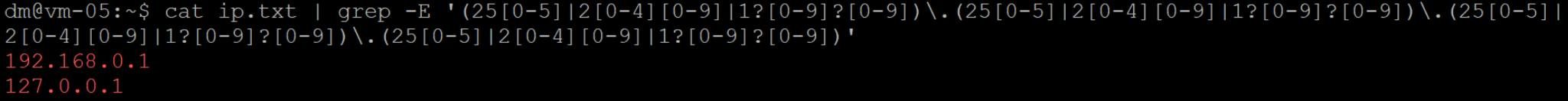
Выражение:

'(25[0-5]|2[0-4][0-9]|1?[0-9]?[0-9])\.(25[0-5]|2[0-4][0-9]|1?[0-9]?[0-9])\.(25[0-5]|2[0-4][0-9]|1?[0-9]?[0-9])\.(25[0-5]|2[0-4][0-9]|1?[0-9]?[0-9])'

команда:

cat ip.txt | grep -E '(25[0-5]|2[0-4][0-9]|1?[0-9]?[0-9])\.(25[0-5]|2[0-4][0-9]|1?[0-9]?[0-9])\.(25[0-5]|2[0-4][0-9]|1?[0-9]?[0-9])\.(25[0-5]|2[0-4][0-9]|1?[0-9]?[0-9])'

Результат:



*Пришлите получившееся выражение в качестве ответа.*

### **Задание 2.**

В Вашей конфигурации Nginx скопилось много неиспользуемых сегментов и становится сложно его читать.

Используя sed удалите все пустые строки и комментарии в конфигурации Nginx.

Попробуйте сделать это одним запуском.

Файл расположен по [ссылке](https://github.com/netology-code/snet-homeworks/blob/snet-18/5-03/nginx.conf)

*Пришлите получившуюся команду в качестве ответа*

**Решение**

Если без внесения правок в файл:

cat n2.txt | sed -E "/(^\ {0,}#+|^$)/d"

С внесением правок в файл:

sed -Ei "/(^\ {0,}#+|^$)/d" n2.txt



### **Задание 3.**

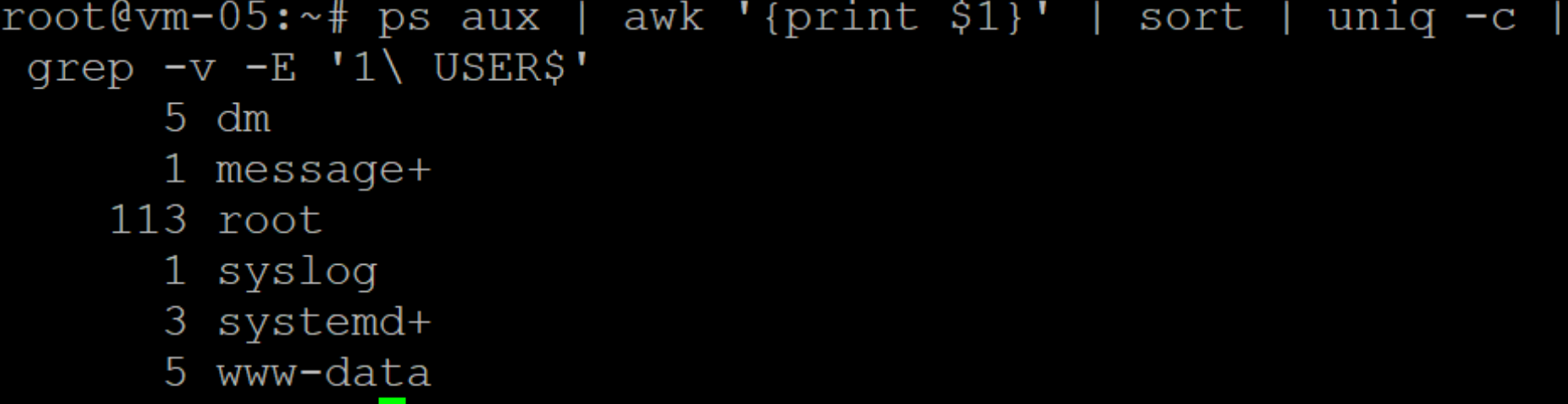
Используя awk и ps aux соберите информацию о:

* количестве процессов для каждого пользователя;

ps aux | awk '{print $1}' | sort | uniq -c | grep -v -E '1\ USER$'

* процессе с самым большим PID;

ps aux --sort pid | tail -1 | awk '{print $2}'



Или:

ps aux | awk '{print $1}' | sort | uniq -c | grep -v -E '1\ USER$' | sort -n

* (дополнительное задание со звездочкой\*) суммарном использовании памяти различными пользователями.

*Пришлите скриншоты со скриптами и демонстрацией их работы*

## **Дополнительные задания (со звездочкой\*)**

Эти задания дополнительные (не обязательные к выполнению) и никак не повлияют на получение вами зачета по этому домашнему заданию. Вы можете их выполнить, если хотите глубже и/или шире разобраться в материале.

### **Задание 4.**

Напишите bash-скрипт который собирает информацию о системе и пишет ее в лог каждые 5 секунд.

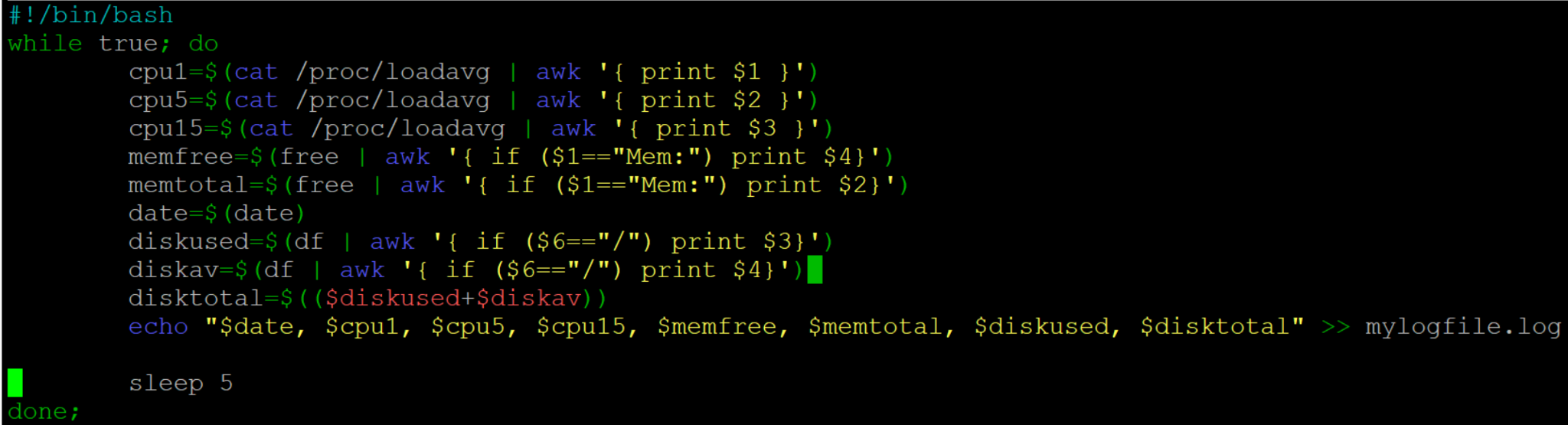
Используемые параметры:

* loadavg[1,5,15] средний показатель загрузки ЦПУ за последние 1 5 и 15 минут. *Примечание:* хранится в /proc/loadavg.
* memfree количество свободной оперативной памяти в байтах. *Примечание:* используем утилиту free.
* memtotal количество всей оперативной памяти в байтах. *Примечание:* используем утилиту free.
* diskfree свободное место на диске подключенного к /. *Примечание:* используем утилиту df.
* disktotal общий объем диска подключенного к /. *Примечание:* используем утилиту df.

***Формат записи:*** timestamp loadavg1 loadavg5 loadavg15 memfree memtotal diskfree disktotal

Успел только написать скрипт по сбору данных:

| #!/bin/bash  while true; do  cpu1=$(cat /proc/loadavg | awk '{ print $1 }')  cpu5=$(cat /proc/loadavg | awk '{ print $2 }')  cpu15=$(cat /proc/loadavg | awk '{ print $3 }')  memfree=$(free | awk '{ if ($1=="Mem:") print $4}')  memtotal=$(free | awk '{ if ($1=="Mem:") print $2}')  date=$(date)  diskused=$(df | awk '{ if ($6=="/") print $3}')  diskav=$(df | awk '{ if ($6=="/") print $4}')  disktotal=$(($diskused+$diskav))  echo "$date, $cpu1, $cpu5, $cpu15, $memfree, $memtotal, $diskused, $disktotal" >> mylogfile.log  sleep 5  done; |
| --- |



Но, не успел и не нашел как с помощью awk отфильтровать в скрипте время за последние 2 минуты от текущего. Если есть ссылка, просьба приведите.)

Пособирайте данные в течении 5-10 минут.

Анализируя этот лог с помощью awk напишите скрипт проверки состояния системы с заданными условиями:

* loadavg1 < 1 в течении последних 2 минут;
* memfree / memtotal < 60% в течении последних 3 минут;
* diskfree / disktotal < 60% в течении последних 5 минут.

Скрипт должен возвращать 0 код ответа, если все условия выполняются, и любой другой в случае невыполнения.

В консоль также необходимо выводить, какое именно из условий не выполняется.

*Пришлите получившиеся скрипты в качестве ответа.*

### **Правила приема работы**

* Код скриптов оформлен по [правилам оформления кода на Bash](https://github.com/netology-code/snet-homeworks/blob/snet-18/code-style.md)
* Код скриптов размещен в отдельном sh-файле на вашем Google диске
* В личном кабинете отправлена ссылка на ваш Google документ, в котором прикреплена ссылка на ваш sh-файл с кодом скриптов и скриншоты, демонстрирующие корректную работу скриптов
* В документе и sh-файле настроены права доступа “Просматривать могут все в Интернете, у кого есть ссылка”
* Название документа содержит название лекции и ваши фамилию и имя