МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №8**

**Дисциплина: Операционные системы**

**Тема: «ПРОЦЕССЫ. РАБОТА С ПРОЦЕССАМИ»**

Работу выполнил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ахвердян Г.А

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль): Математическое и программное

обеспечение компьютерных технологий

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. А. Нигодин

Краснодар

2025

**Практические задания:**

**Задание 1.**

1) Выведите на экран все потоки процесса syslog-ng, используя команду ps.

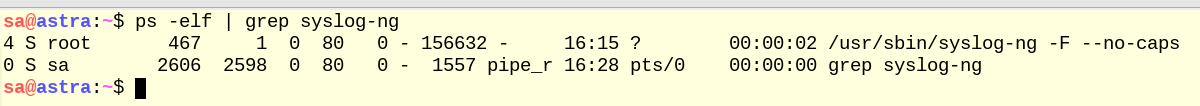
****

Рисунок 1 – Потоки процесса

2) Попробуйте выполните то же задание, используя утилиту htop.

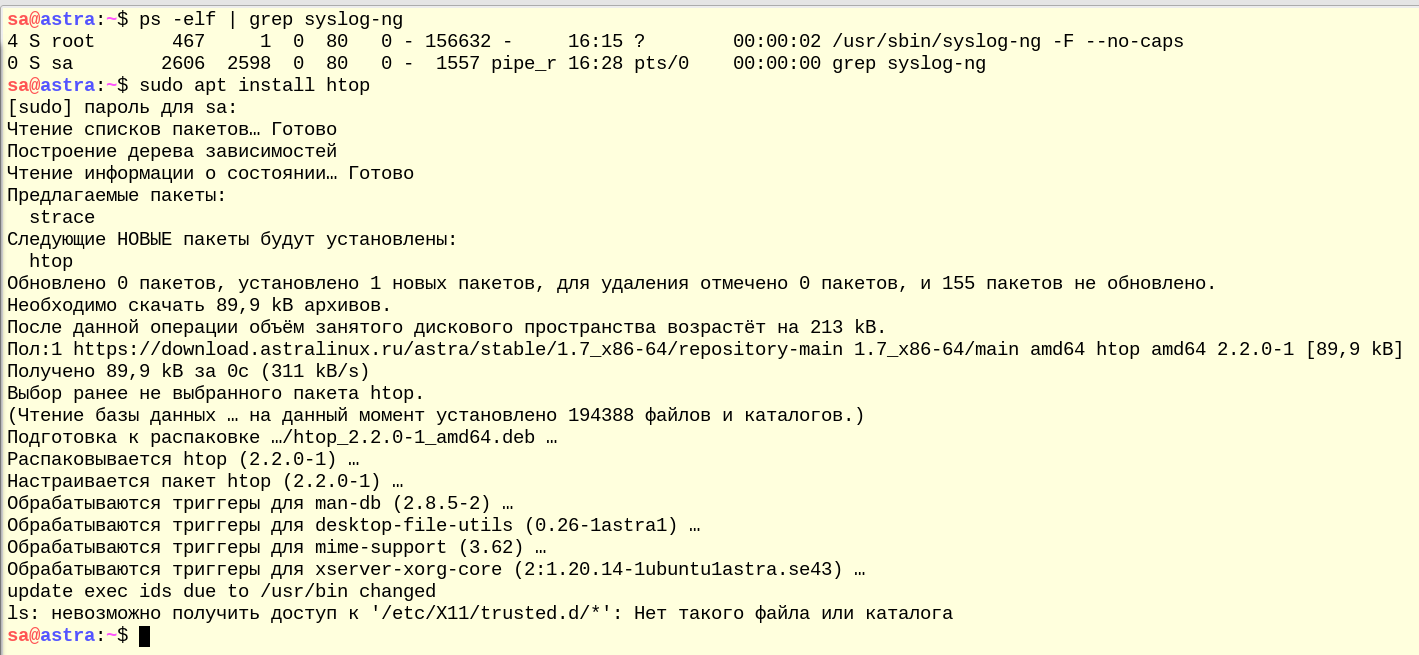


Рисунок 2 – Загрузка htop

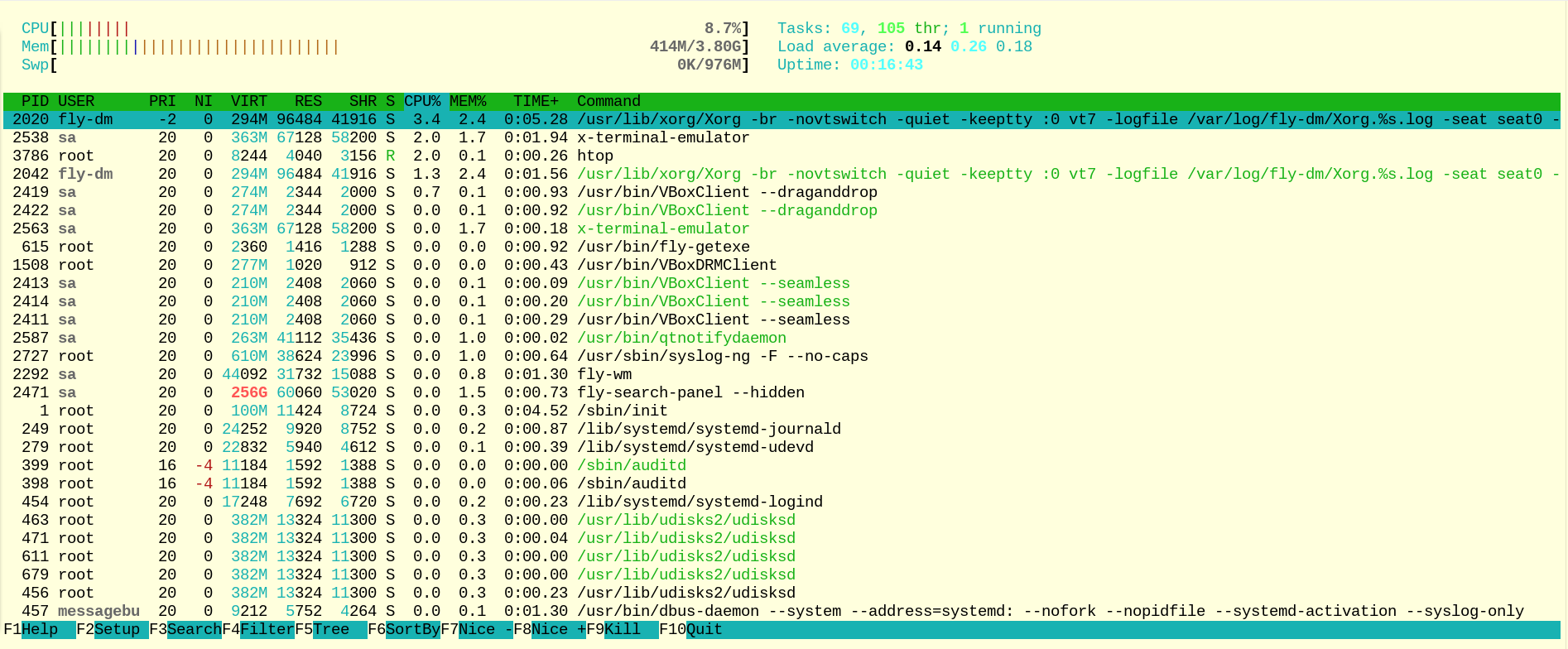
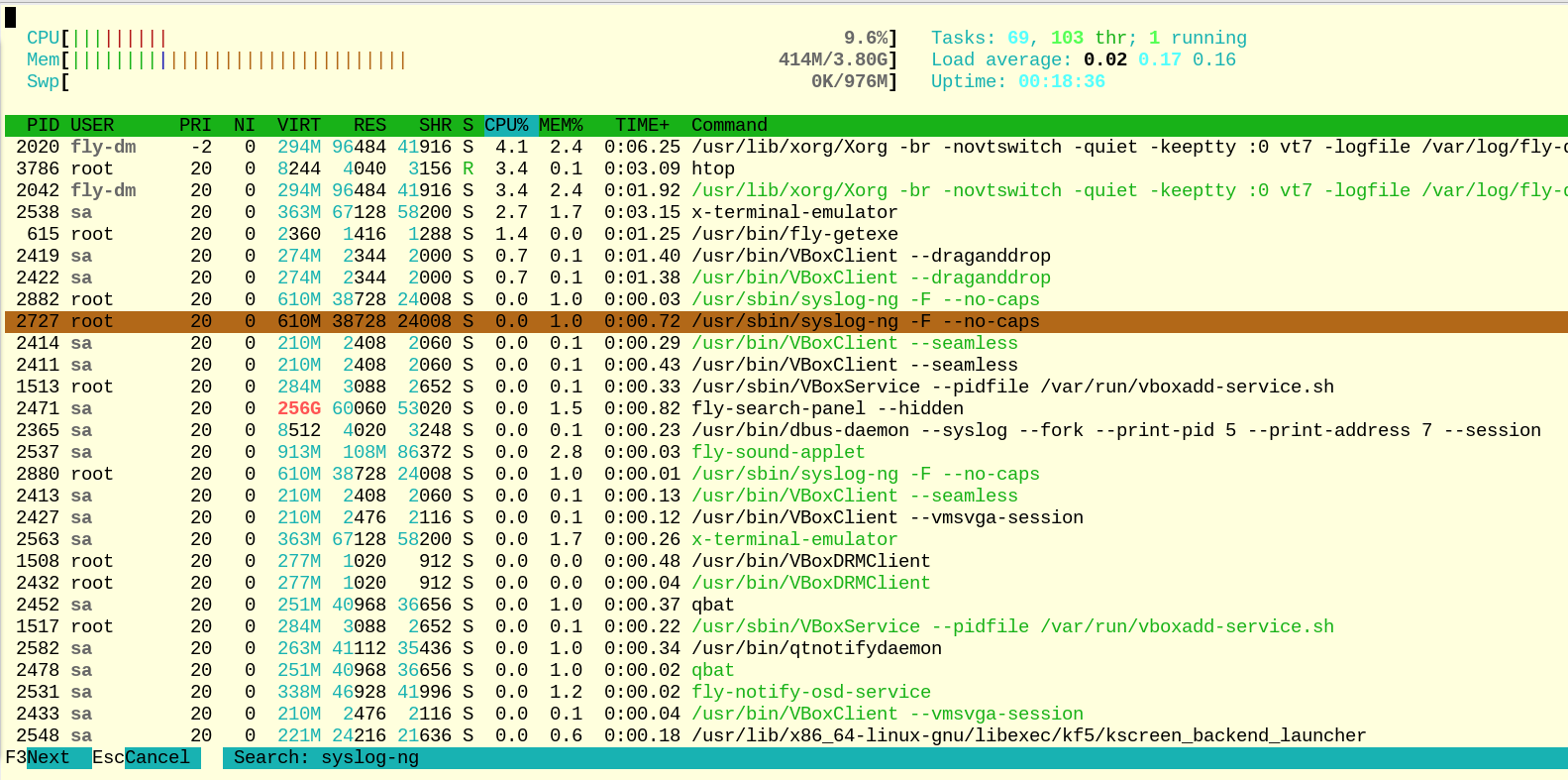


Рисунок 3 – Потоки процесса через утилиту htop

  
Рисунок 4 – Потоки процесса через утилиту htop

**Задание 2.**

1) Выведите на экран информацию, в том числе PID, %MEM, EUID, RUID, SUID и Сommand о топ-15 процессов по использованию памяти, используя команду ps.

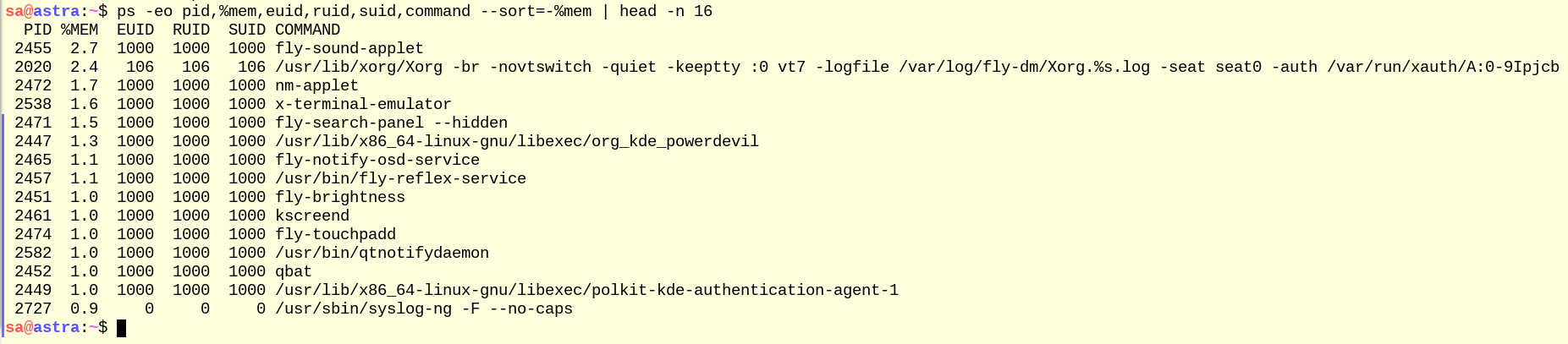


Рисунок 5 – Топ-15 процессов по использованию памяти

2) Попробуйте выполнить то же задание, используя утилиту htop.

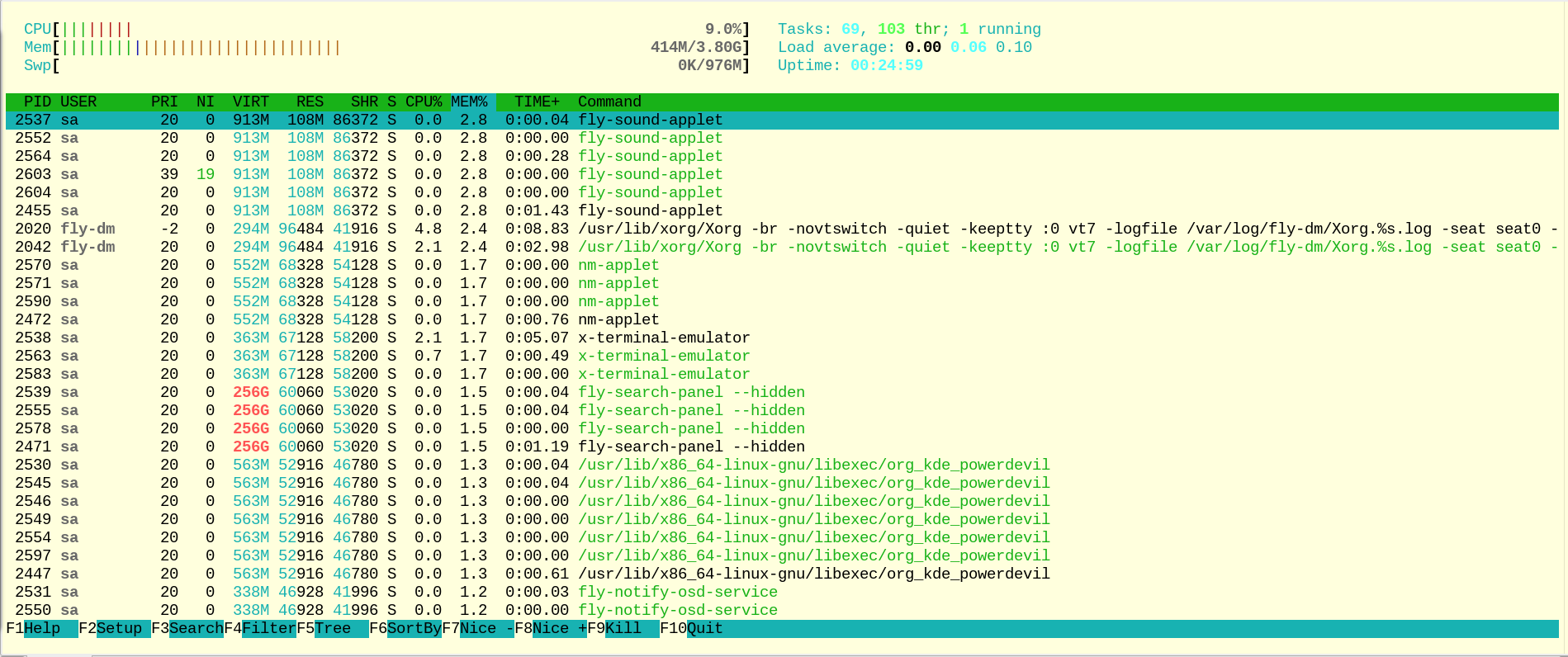


Рисунок 6 – Топ процессов по использованию памяти в htop

**Задание 3.**

1) Создайте файл в домашнем каталоге test\_nano и откройте его текстовым редактором nano.

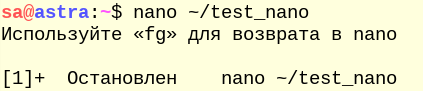


Рисунок 7 – Создание файла

2) В этом же терминале, используя командную строку, найдите этот процесс и повысьте его приоритет до 15.

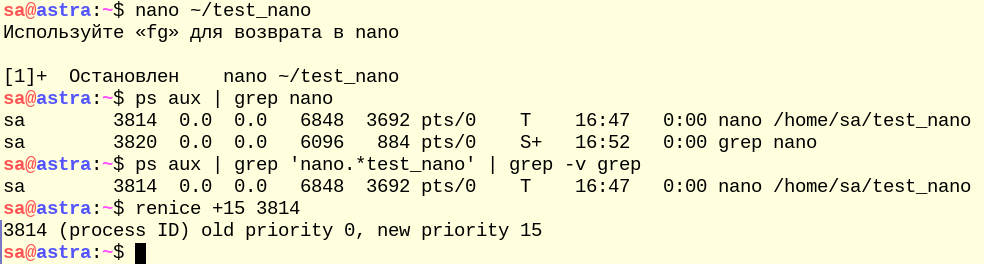


Рисунок 8 – Повышение приоритета

3) Верните из фона процесс nano и закройте его.



Рисунок 9 – Возвращение из фона

4) Откройте файл test\_nano текстовым редактором nano, задав ему приоритет 25 при создании.

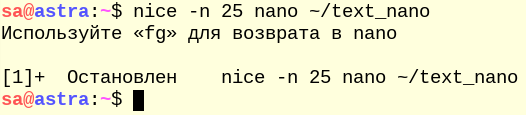


Рисунок 10 – Задание приоритета

5) Завершите процесс nano, отправив ему сигнал SIGKILL.

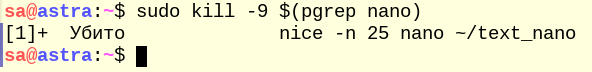


Рисунок 11 – Завершение процесса

**Задание 4.**

1) Тут вам необходимо будет управлять приоритетами процессов. Для наглядности работы по управлению приоритетом процессов используйте скрипт, который создаст нагрузку на CPU.   
Создайте файлы ~/scripts/resource-demanding1.sh и ~/scripts/resource-demanding2.sh со следующим содержимым (два файла нам нужны, чтобы впоследствии различать процессы не только по PID, но и по команде запуска):



Рисунок 12 – Создание директории и создание файла

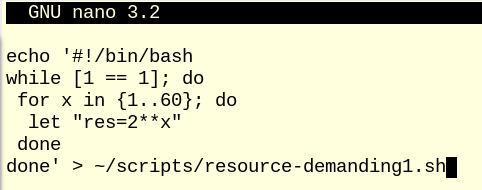
****

Рисунок 13 – Скрипт

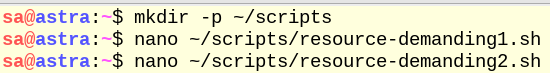
****

Рисунок 14 – Создание второго файла

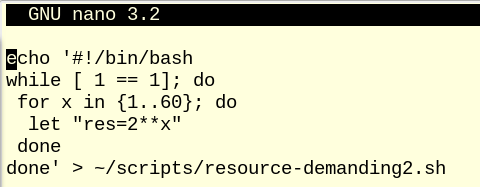
****

Рисунок 15 – Скрипт второго файла

• Запустите утилиту htop в новом терминале и отсортируйте процессы по использованию CPU.

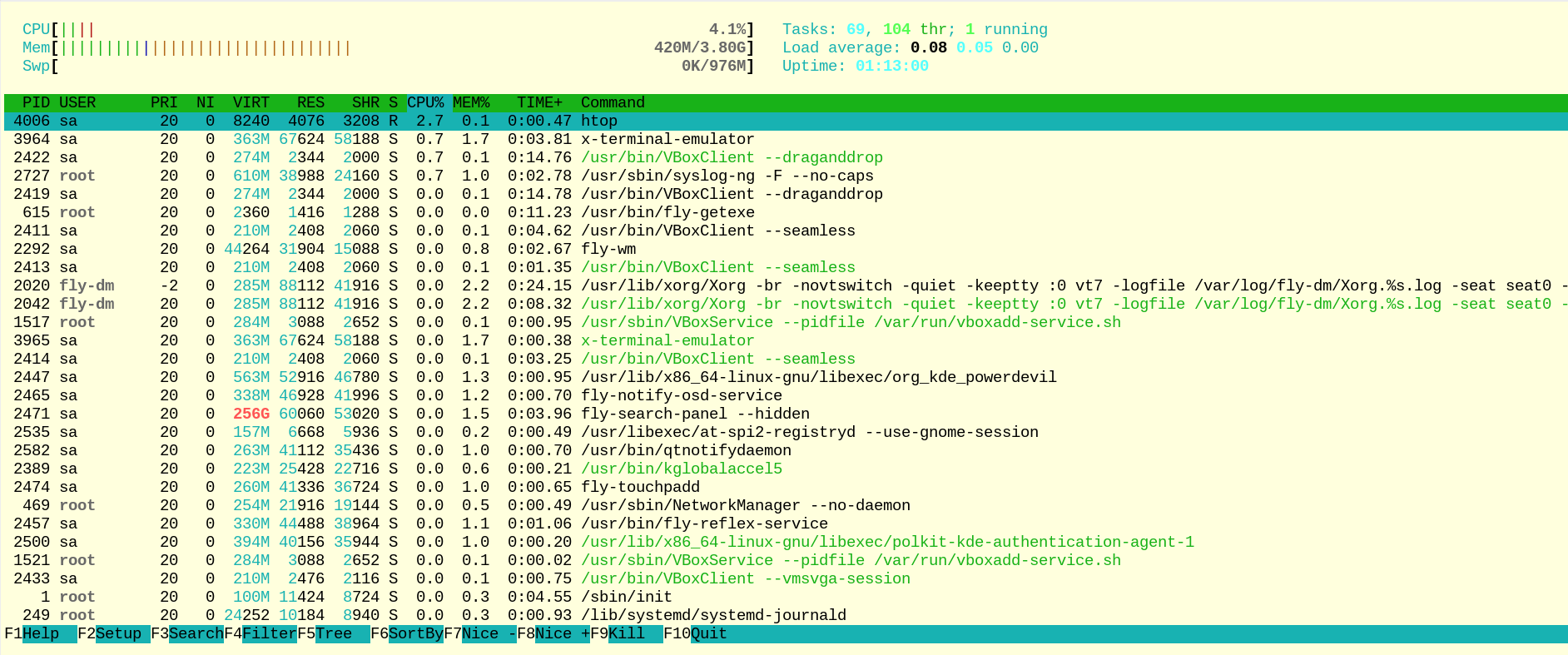
****

Рисунок 16 – Процессы по использования CPU

• Сделайте файлы исполнимыми для владельца и запустите их с помощью команды taskset в фоновом режиме на первом ядре процессора. Сколько процентов первого ядра процессора используют созданные процессы?

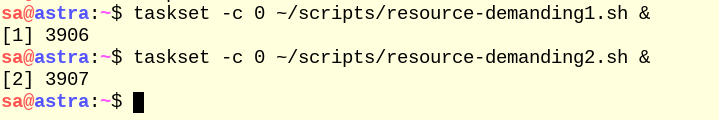


Рисунок 17 – Кол-во процентов первого ядра процессора использующее созданные процессы

• Измените приоритет первого процесса, понизив его до 25, 30, 35 и 39 последовательно. Посмотрите, как менялось при этом использование процессора.

- Изменяя приоритет, практически ничего не меняется

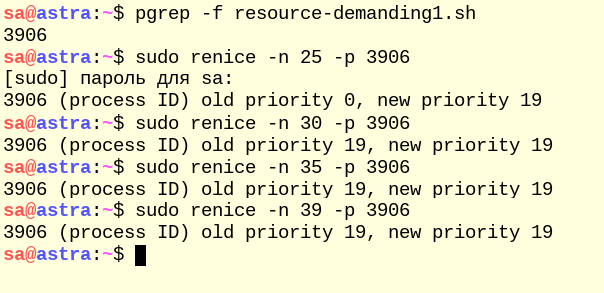


Рисунок 18 – Изменение приоритета

• Повысьте приоритет первого процесса до 15, 10, 5, 0 последовательно. Посмотрите, как менялось при этом использование процессора?

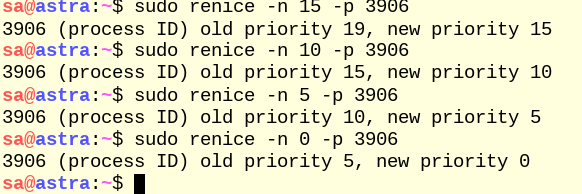


Рисунок 19 – Повышение приоритета

• Задайте приоритет второму процессу так, чтобы он потреблял не менее 75% ресурсов первого ядра CPU, не изменяя при этом приоритет первого процесса (0).

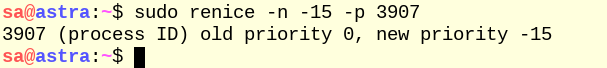


Рисунок 20 – Потребленее не менее 75%

• Завершите процессы, отправив первому сигнал SIGINT, а второму SIGKILL.

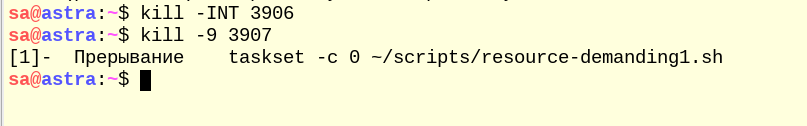


Рисунок 21 – Завершение процессов

**Задание 5.**

1) Задайте ограничение для текущей оболочки на количество запущенных потоков. Задайте ограничение, превышающее текущее количество запущенных потоков пользователем на 3, запустите несколько раз новый процесс bash, убедитесь, что лимит действует, и закройте терминал с ошибкой.

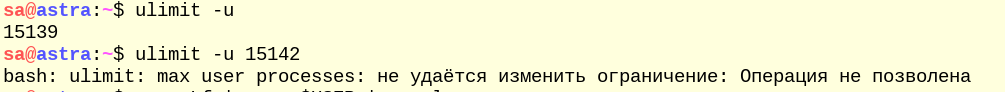


Рисунок 22 – Задание ограничения

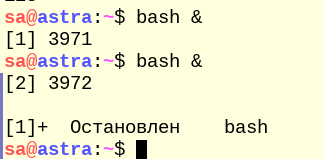


Рисунок 23 – Действие лимита

**Задание 6.**

1) Получите информацию из псевдофайловой системы /proc:  
О процессе syslog-ng:

Строка запуска процесса

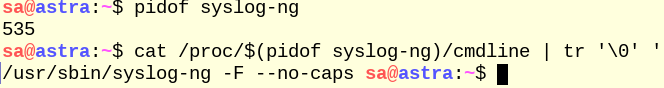


Рисунок 24 – Строка запуска процесса

Окружение процесса

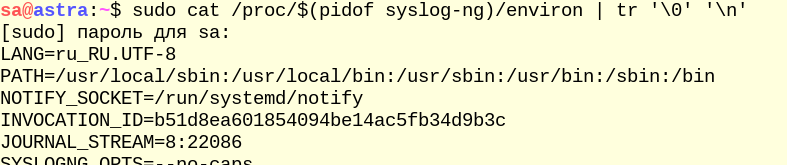


Рисунок 25 – Окружение процесса

Сведения об объемах данных, прочитанных и записанных процессом в хранилище информации

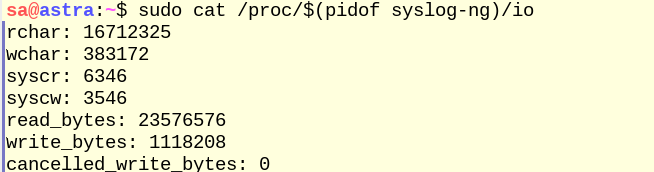


Рисунок 26 – Сведения об объемах данных, прочитанных и записанных процессом в хранилище информации

Лимиты процесса

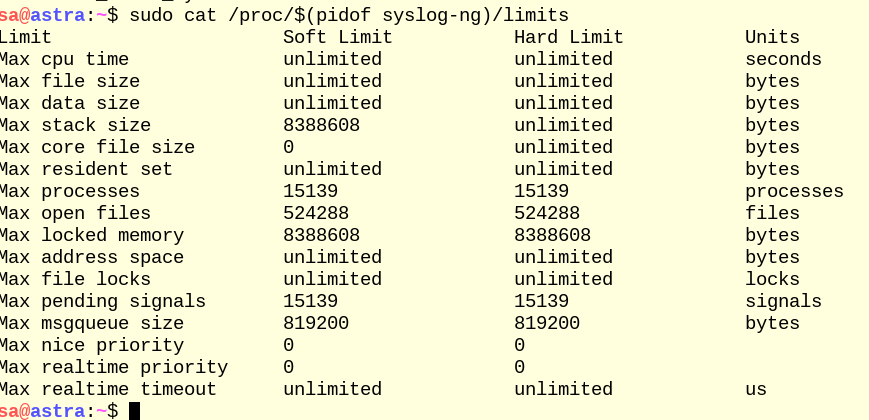


Рисунок 27 – Лимиты процесса

Информация о статусе процесса

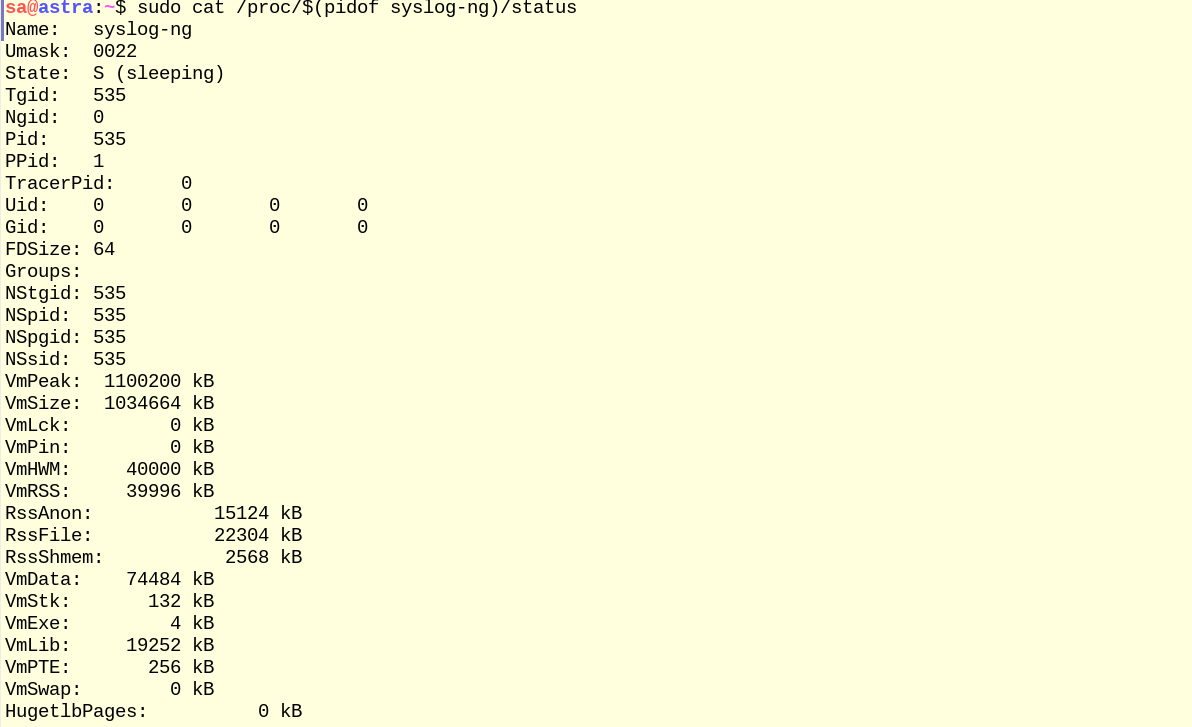


Рисунок 28 – Информация о статусе процесса

Об ОС:

Список параметров, которые были переданы ядру при загрузке



Рисунок 29 – Список параметров, которые были переданы ядру при загрузке

Сведения о всех установленных процессорах

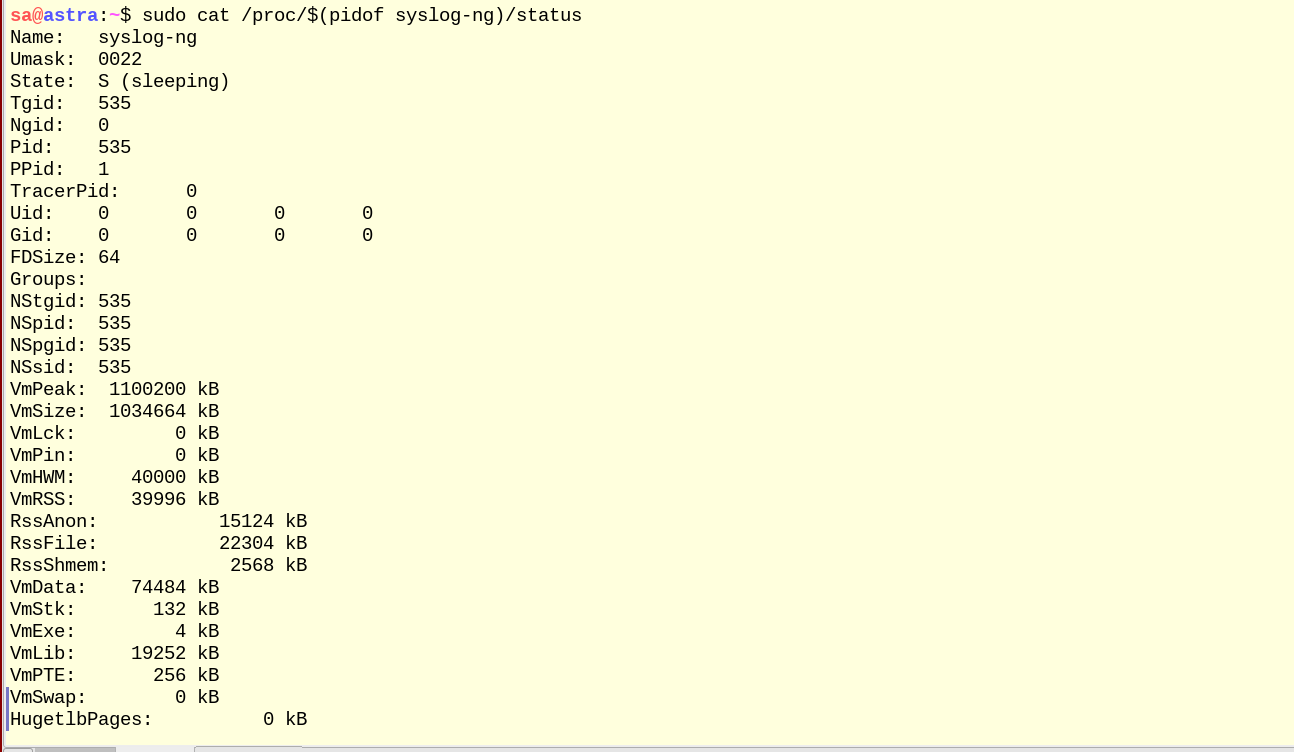


Рисунок 30 – Сведения о всех установленных процессорах

Информация о состоянии памяти

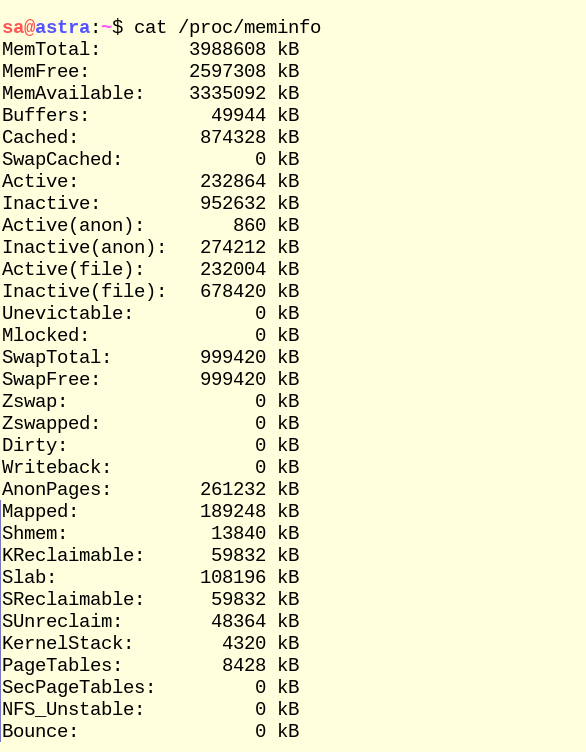


Рисунок 31 – Информация о состоянии памяти

Перечень устройств в системе

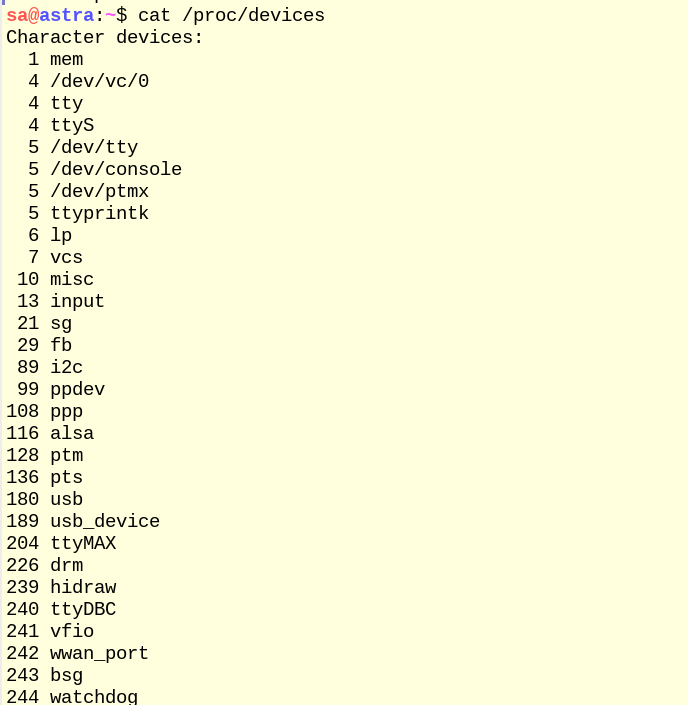


Рисунок 32 – Перечень устройств в системе

Перечень файловых систем, поддерживаемых ядром ОС

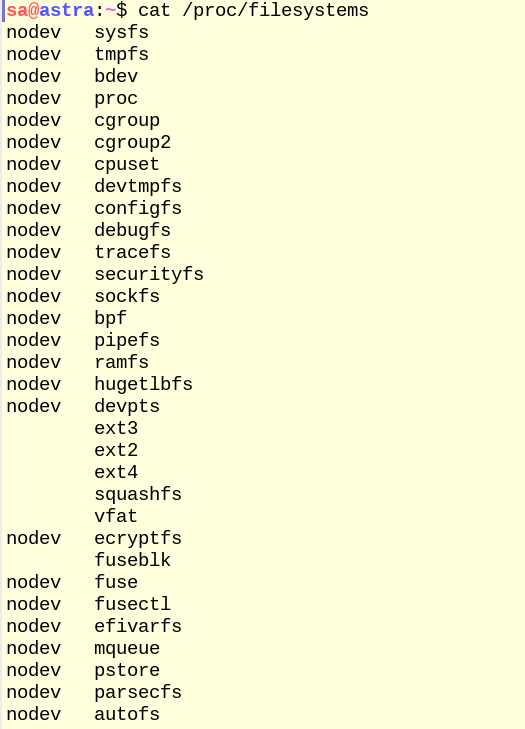


Рисунок 33 – Перечень файловых систем, поддерживаемых ядром ОС

Перечень смонтированных файловых систем

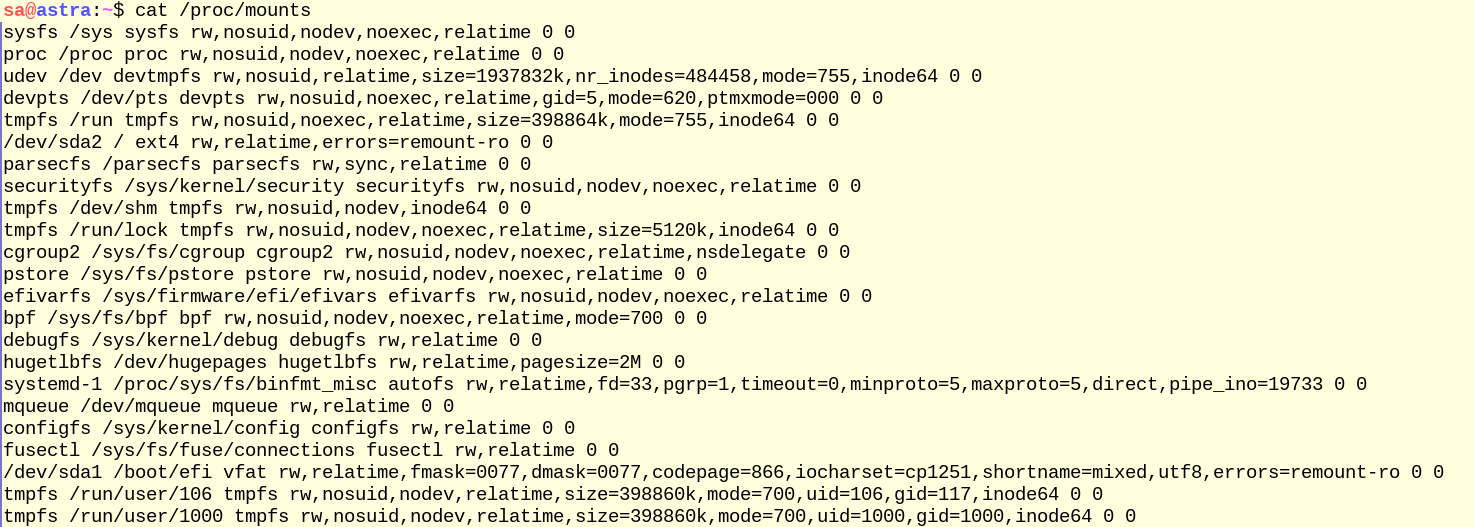


Рисунок 34 – Перечень смонтированных файловых систем

Список разделов подкачки

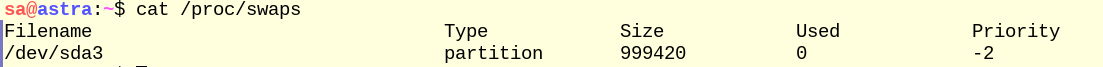


Рисунок 35 – Список разделов подкачки

**Вопросы**

**1)** Дополните фразу «проблема иерархической структуры процессов в современных Linux дистрибутивах заключается в»- Сложности управления зависимостями между процессами

**2)** Сколько состояний есть у процесса?- 4: 1.Выполнение, 2.Ожидание, 3.Приостановка, 4.завершение

**3**) Чем больше значение параметра приоритета процесса, тем…

- Ниже его приоритет

**4)** Перечислите все команды, позволяющие управлять процессами в фоновом режиме работы. **-** & — запуск в фоне  
- bg — возобновление остановленного процесса в фоне  
- fg — возврат процесса на передний план  
- jobs — список фоновых задач текущей сессии  
- nohup — запуск с игнорированием SIGHUP  
- disown — удаление задачи из списка jobs

**5)** Соотнесите команду для назначения политики планирования и тип политики:   
A. SCHED\_OTHER - 2. sudo chrt -o -p 0 {PID}  
B. SCHED\_RR - 3. sudo chrt -r -p {priority} {PID}  
C. SCHED\_FIFO - 1. sudo chrt -f {priority} {PID}

**6)** Какая команда выведет процессы и потоки?

- ps –eLf

**7)** Какое назначение у сигналов SIGINT, SIGKILL, SIGCHILD, SIGSTOP, SIGSTP, SIGCONT, SIGTERM?- SIGINT – прерывание процесса  
- SIGKILL – принудительное завершение процесса  
- SIGCHILD – уведомление родительского процесса о завершении дочернего  
- SIGSTOP – приостановка процесса  
- SIGSTP – приостановка процесса  
- SIGCONT – возобновление выполнения приостановленного процесса  
- SIGTERM – запрос о завершении процесса

**8)** Какая команда выполнит поиск по имени команды, использующейся при запуске процесса?

- ps aux | grep <имя\_процесса>

**9)** Что содержит в себе файл /proc/cmdline/?  
- Строку запуска процесса

**10)** Что делает команда ulimit -u ?  
- Устанавливает максимальное количество процессов для текущего пользователя

**Вывод**

В этой лабораторной я познакомился с особенностями жизненного цикла процессов в Linux, их рождением, жизнью и смертью. Узнал, что такое сигналы, для чего они нужны и как их использовать. Я также научился продвинутым приемам извлечения информации о процессах и управления ими.