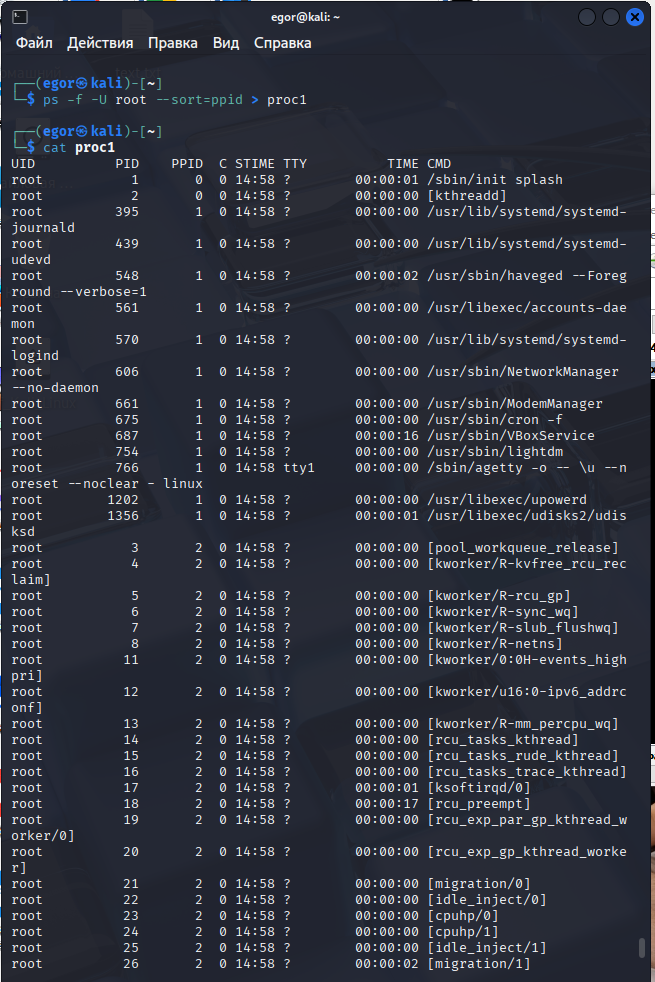
1. Создайте файл proc1. В нём должен быть список процессов пользователя root с сортировкой по идентификатору родительского процесса. Используйте команду ps и изученные утилиты.

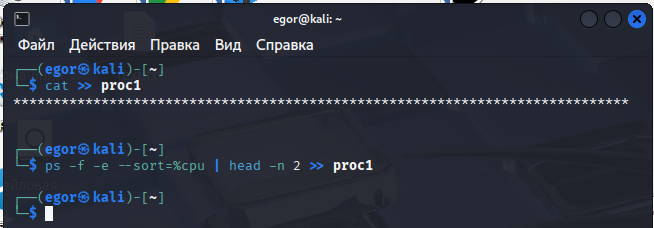
Выведем процессы пользователя root в файл proc1

ps -f -U root --sort=ppid > proc1

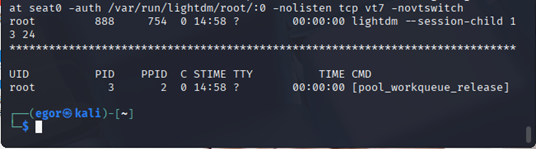


1. Добавьте к файлу proc1 сведения о процессе, который сейчас потребляет больше ресурсов центрального процессора. Используйте команду ps.

Добавим разделитель в виде \* и запишем в файл proc1 сведения о процессе, который потребляет больше всего ресурсов



Посмотрим конец файла



1. Запустите утилиту top. Изучите содержимое информационных полей утилиты. Вы должны получить информацию о степени использования ресурсов системы, времени её работы и количестве пользователей.

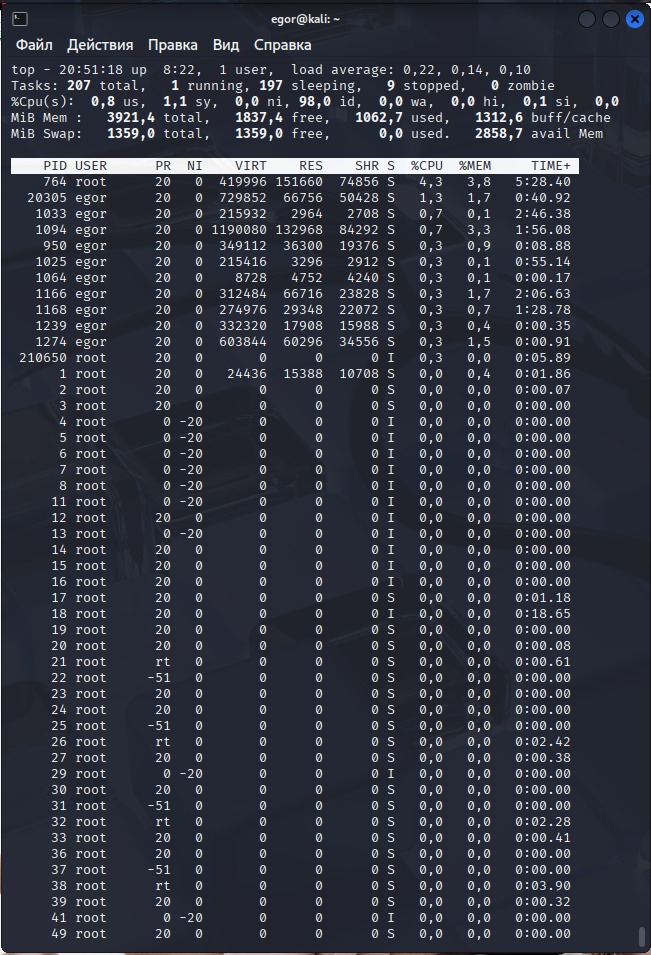
**В первой строке отображается:** название утилиты (top); текущее время ; как долго работает система после последней загрузки (up); сколько в системе активных пользователей (user); средняя загрузка (за последние 1, 5 и 15 минут) (load average).

**Во второй строке отображается:** общее количество процессов (total); количество процессов работающих в настоящее время (running); количество спящих процессов (sleeping); количество остановленных процессов (stopped); количество процессов-зомби (zombie).

**В третьей строке отображается:** потребление ресурсов CPU пользовательскими процессами (us); потребление ресурсов CPU системными процессами (sy); потребление ресурсов CPU пользовательскими процессами с измененным приоритетом (ni); работа CPU в холостом режиме (id); время простаивания CPU, в течении которого он ожидал завершения операции ввода-вывода (wa); время работы аппаратных прерываний (hi); время работы программных прерываний (si); время, когда реальный CPU не был доступен (параметр для виртуальных машин).

**В четвертой строке отображается потребление оперативной памяти (Mem):** общее количество оперативной памяти (total); количество свободной памяти (free); количество используемой памяти (used); зарезервированная память под буфер и кэш (buff/cache).

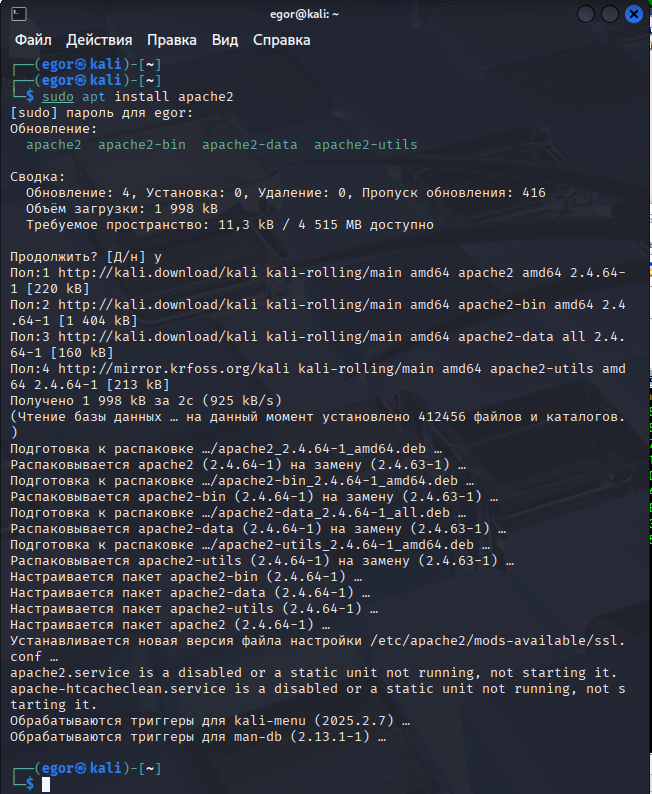
**В пятой строке отображается потребление оперативной памяти (Swap):** общее количество оперативной памяти (total); количество свободной памяти (free); количество используемой памяти (used); зарезервированная память для новых процессов (avail Mem).

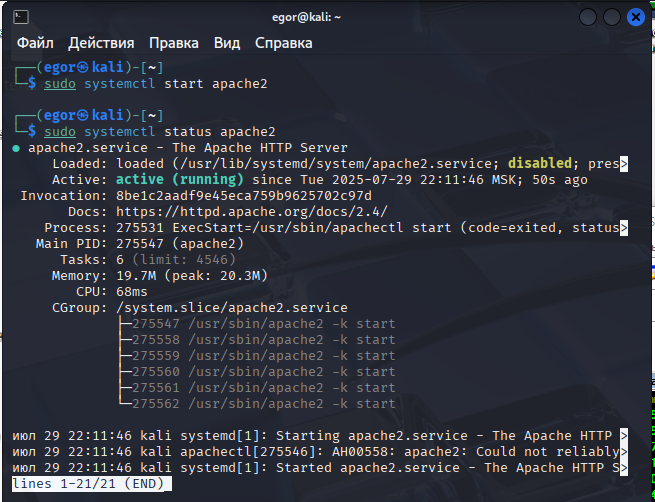


В таблице есть следующие столбцы: **PID** – идентификатор процесса; **USER** – имя пользователя, под которыми работает процесс; **PR** – приоритет процесса; **NI** – базовый приоритет (или еще говорят любезность процесса, чем она выше, тем ниже приоретет); **VIRT** – объем виртуального пространства; **RES** – память в физической оперативной памяти; **SHR** – объем памяти используемой процессом; **S** – состояние процесса (активный, спит, в ожидании и т.д); **%CPU** - использование ресурсов CPU; **%MEM** – использование памяти; **TIME+** – время работы CPU с процессом.

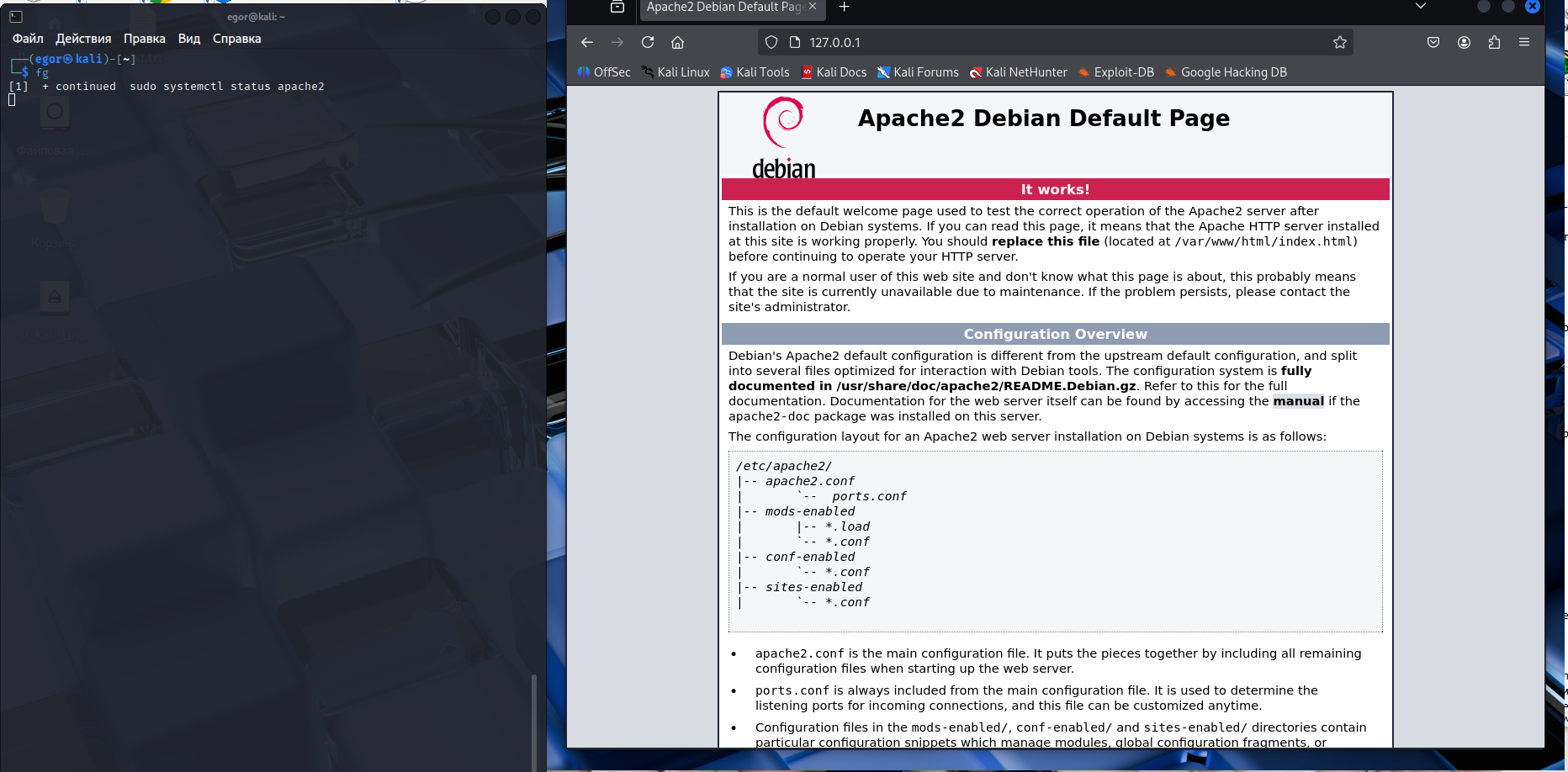
***По результатам анализа таблицы на скрине можем сделать следующий вывод:*** что в 20:51:18:

1. система уже работает 8:22;
2. в системе активный 1 пользователь;
3. среднее время загрузки за последнюю 1 минуту 0.22, за последние 5 минут 0.14 и за последние 15 минут 0.10;
4. в системе общее количество процессов 207;
5. из них работающих в данный момент 1;
6. спящих 197;
7. остановленных 9;
8. процессов-зомби 0;
9. потребление cpu пользовательскими процессами 0.8%;
10. системными процессами 1.1%;
11. процессами с измененным приоритетом 0%;
12. простаивание cpu 98%;
13. процент времени простаивания cpu ожидая завершения операций ввода/вывода 0%;
14. процент времени работы аппаратных прерываний 0%;
15. процент времени работы программных прерываний 0.1%;
16. процент времени недоступности реального cpu 0%;
17. общее количество оперативной памяти около 3.9 Гб;
18. свободно около 1.8Гб;
19. используется около 1.1 Гб;
20. зарезервированная память около 1.3 Гб;
21. общее количество памяти в файле подкачки около 1.4 Гб;
22. свободно около 1.4 ГБ;
23. используется 0 Гб;
24. зарезервировано около 2.9 Гб.
25. Установите веб-сервер apache используя следующую команду sudo apt install apache2. Запустите сервер apache командой sudo systemctl start apache2. Запустите команду sudo systemctl status apahce2 в фоновом режиме.





1. Переместите последнюю задачу с заднего плана на передний при помощи команды fg. После перемещения определите, успешно ли запущен сервер (в выводе должна быть надпись active (running)). Подтвердите успешный запуск веб-сервера переходом по адресу 127.0.0.1 в браузере.



1. Выполните команду Killall -u имя\_пользователя. Опишите, что получили.

При выполнении команды killall -u egor происходит выход из учетной записи, т.е. убиваются все процессы пользователя egor.

