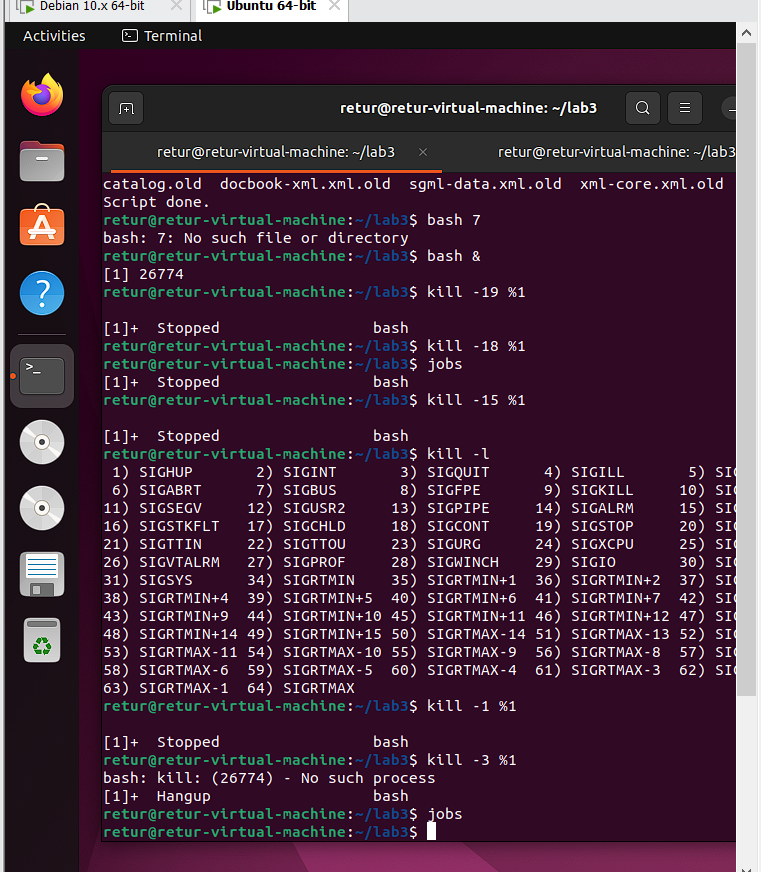


exec ls -R /etc *выводит содержимое etc рекурсивно*



*TERM - игнорируется  
INT - очищает командную строку*

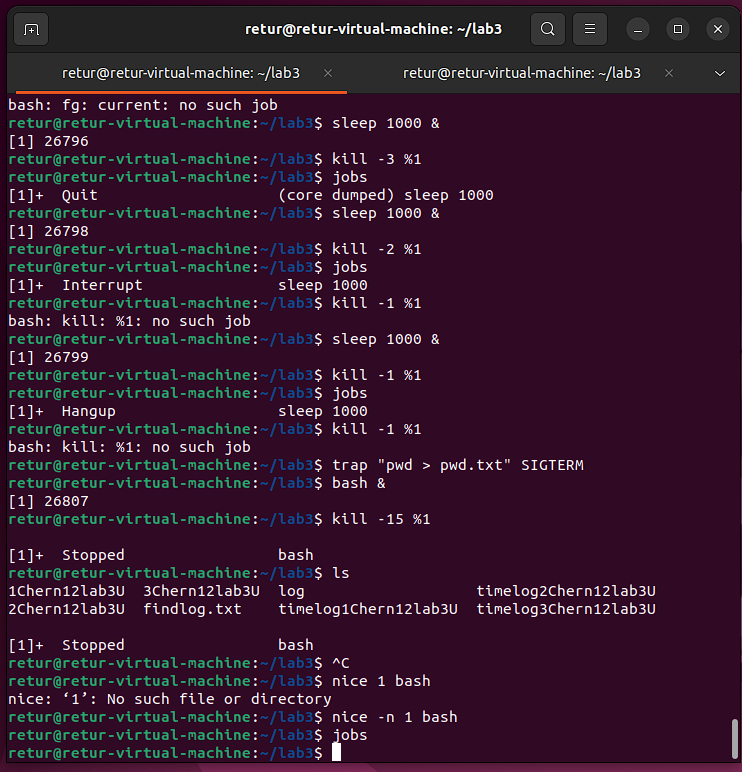
*HUP - завершает сеанс*

1. *Ничего не произойдёт*
2. *На все*

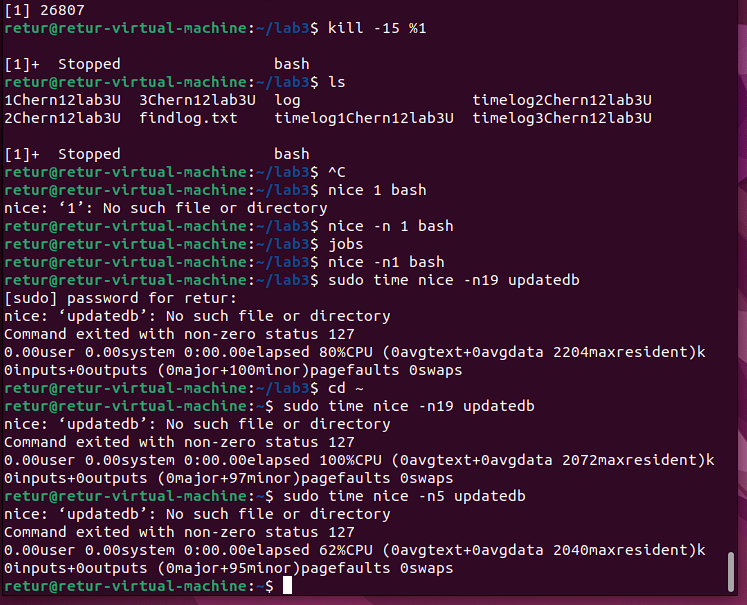
*7 trap “pwd > pwd.txt” SIGTERM*

*8 нет*

*9ничего*



10



*11. at now + 5 minute*

*ps >> ps.txt*

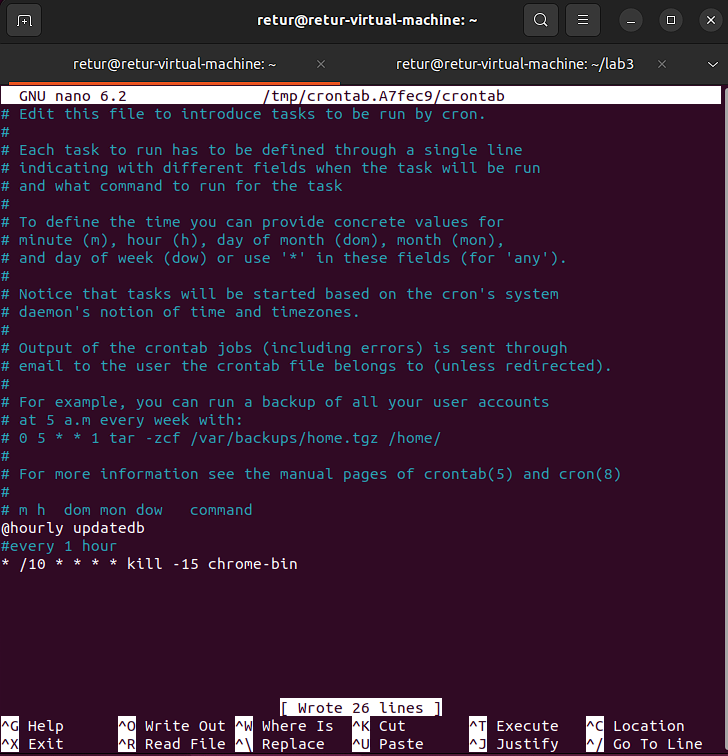
*12. at 4pm*

*killall -9 firefox-bin*

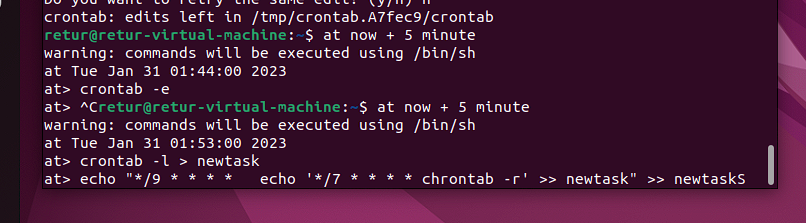
*13, 14. 0 0 \* \* \* updatedb*

*0 \* \* \* \* updatedb - каждый час*

*0,5,10,15,20,25,30,35,40,45,50,55 \* \* \* \* updatedb - каждые 5 минут*



15



Контрольные вопросы

1) *Объясните понятия процесса и ресурса. Какое их значение в организации вычислительного процесса в ОС Linux?*

Процесс – набор выполняемых заданий, инструкций. Ресурс – это то, что выделяется процессу для выполнения данных задач. Процесс представлен в памяти в виде дескриптора процесса, который включает: исполняемый текст программы и управляющая информация. Концепция процессов обеспечивает свойство многозадачности ОС, т.е. когда в активном состоянии могут находиться несколько задач, между которыми разделяется процессорное время.

2) *Какая информация содержится в описателях процессов? Как просмотреть их содержание в процессе работы с системой?*

Описатель процесса – это дескриптор. Дескриптор включает в себя исполняемый текст программы задачи, управляющую информацию (контекст процесса), необходимую для выделения и использования выделенных процессу ресурсов: приоритет, необходимое адресное пространство, вектор прерывания (слово состояния программы), величину кванта процессорного времени, признаки состояния процесса и пр.

Посмотреть: ps -l, ps.

3) *Какими способами можно организовать выполнение программ в фоновом режиме?*

Для запуска фонового (параллельного с другими потомками) процесса в командную строку необходимо и достаточно последним символом добавить знак &

4) *Какие особенности выполнения программ в фоновом режиме? Как избежать вывода фоновых сообщений на экран и прерывания выполнения фоновых программ при прекращении сеанса работы с системой?*

Фоновые процессы обладают некоторыми недостатками:

- не допускают ввода с клавиатуры;

- обеспечивают вывод на экран, но при этом нарушают целостность вывода диалогового процесса.

Общепринятый прием исключения влияния фонового вывода на интерактивную работу:

$<командная\_строка> > имя\_файла.out &

- <командная\_строка> планирует задание для фонового режима;

- перенаправляет вывод вместо экрана в указанный файл головного каталога пользователя.

особенности работы с фоновым режимом:

- выполняемая в фоновом режиме программа (команда), требующая стандартного ввода, должна читать его из файла с использованием перенаправленного ввода;

- программа, выполняемая в фоновом режиме, не может быть прервана <Ctrl\*C>, так как она отсоединяется от клавиатуры и может быть прекращена только с помощью команды kill или выходом из системы;

- выход из системы exit надо выполнять два раза: для завершения фонового процесса и завершения основного процесса shell.

5) *Как пользователь может повлиять на распределение ресурсов между активными процессами?*

С помощью команды nice.

Если при выполнении задания образуются несколько порожденных процессов, то все они имеют одинаковый приоритет равный родительскому. В этом случае все процессы получают ресурсы равными долями (простой режим разделения времени). При необходимости выделения наиболее важных родительских процессов порожденным второстепенным можно понизить приоритет с помощью команды.

6) *Как можно прервать выполнение активных процессов? Какая информация для этого необходима и откуда она извлекается?*

Завершение процессов – одна из функций управления процессами. Прекратить выполнение любого процесса можно с помощью команды:

kill [-опции] PID1 [PID2......] — передает сигнал процессу PID.

7) *Перечислите базовые средства взаимодействия процессов в Linux.*

Каналы, сигналы, разделяемая память, очереди сообщений, сокеты

8) *Поясните особенности работы с каналами в Linux.*

Канал - это средство связи стандартного вывода одного процесса со стандартным вводом другого. Эта особенность широко используется даже в командной строке UNIX (в shell-е). Когда процесс создает канал, ядро устанавливает два файловых дескриптора для пользования этим каналом. Один такой дескриптор используется, чтобы открыть путь ввода в канал (запись), в то время как другой применяется для получения данных из канала (чтение)

9) *Почему отложенные вызовы не обрабатываются непосредственно обработчиком прерывания таймера?*

Потому что функции отложенных вызовов выполняются в системном контексте, а не в контексте прерывания