Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированные системы управления

Лабораторная работа №3

по Linux-server

«Процессы и управление ими в ОС Linux»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент |  |  |  | Кретов И.О. |
| Группа С-АС-22-1 |  |  |  |  |
| Руководитель |  |  |  | Кургасов В.В. |

Липецк 2023 г.

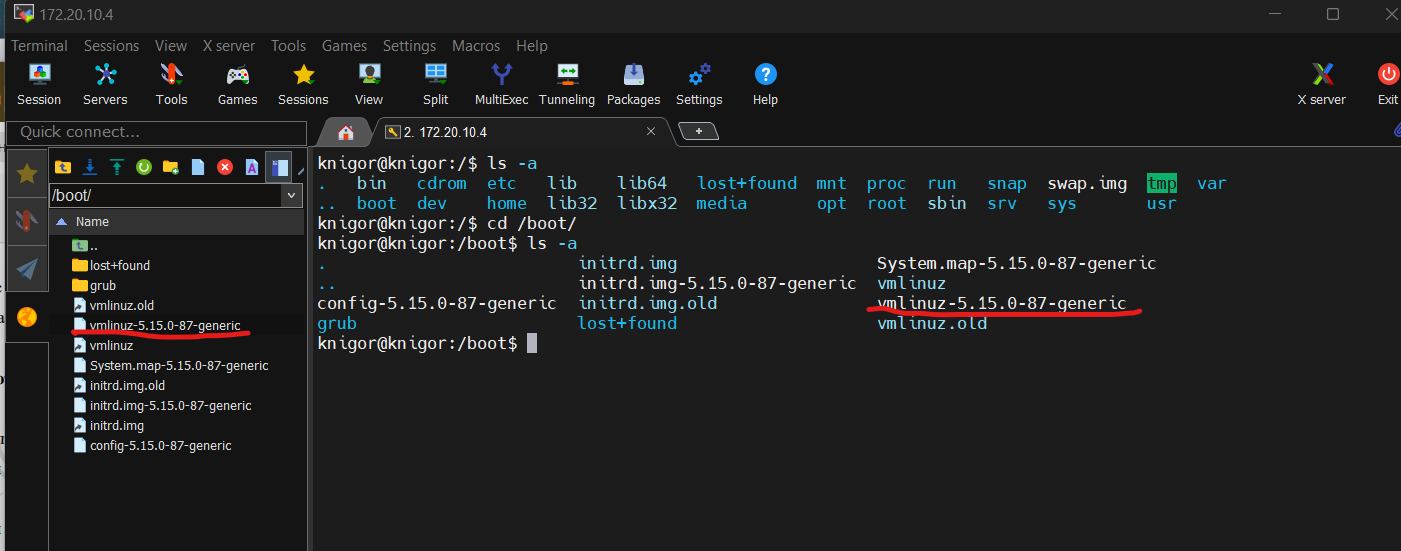
# Цель работы:

Ознакомиться на практике с понятием процесса в операционной системе. Приобрести опыт и навыки управления процессами в операционной системе.

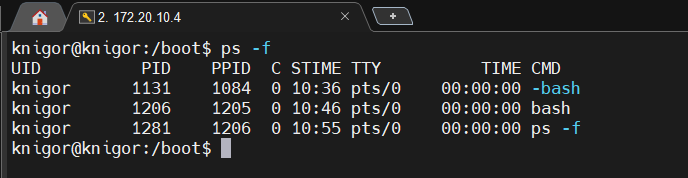
1. **Первая часть.**

Файл с образом ядра находится в папке /boot/

Перейдя в директорию напишем команду ls -a, просмотрим файлы и по названию узнаем версию ядра 5.15.0.87



Прописав команду ps -f мы можем посмотреть процессы



Рассмотрим общий синтаксис команды:

**$ ps опции**

**$ ps опции | grep параметр**

Во втором варианте мы используем утилиту grep для того, чтобы отобрать нужные нам процессы по определенному критерию. Теперь рассмотрим опции утилиты. Они делятся на два типа - те, которые идут с дефисом Unix и те, которые используются без дефиса - BSD. Лучше пользоваться только опциями Unix, но мы рассмотрим и одни и другие. Заметьте, что при использовании опций BSD, вывод утилиты будет организован в BSD стиле.

* **-A, -e, (a)** - выбрать все процессы;
* **-a** - выбрать все процессы, кроме фоновых;
* **-d, (g)** - выбрать все процессы, даже фоновые, кроме процессов сессий;
* **-N** - выбрать все процессы кроме указанных;
* **-С** - выбирать процессы по имени команды;
* **-G** - выбрать процессы по ID группы;
* **-p, (p)** - выбрать процессы PID;
* **--ppid** - выбрать процессы по PID родительского процесса;
* **-s** - выбрать процессы по ID сессии;
* **-t, (t)**- выбрать процессы по tty;
* **-u, (U)** - выбрать процессы пользователя.

Опции форматирования:

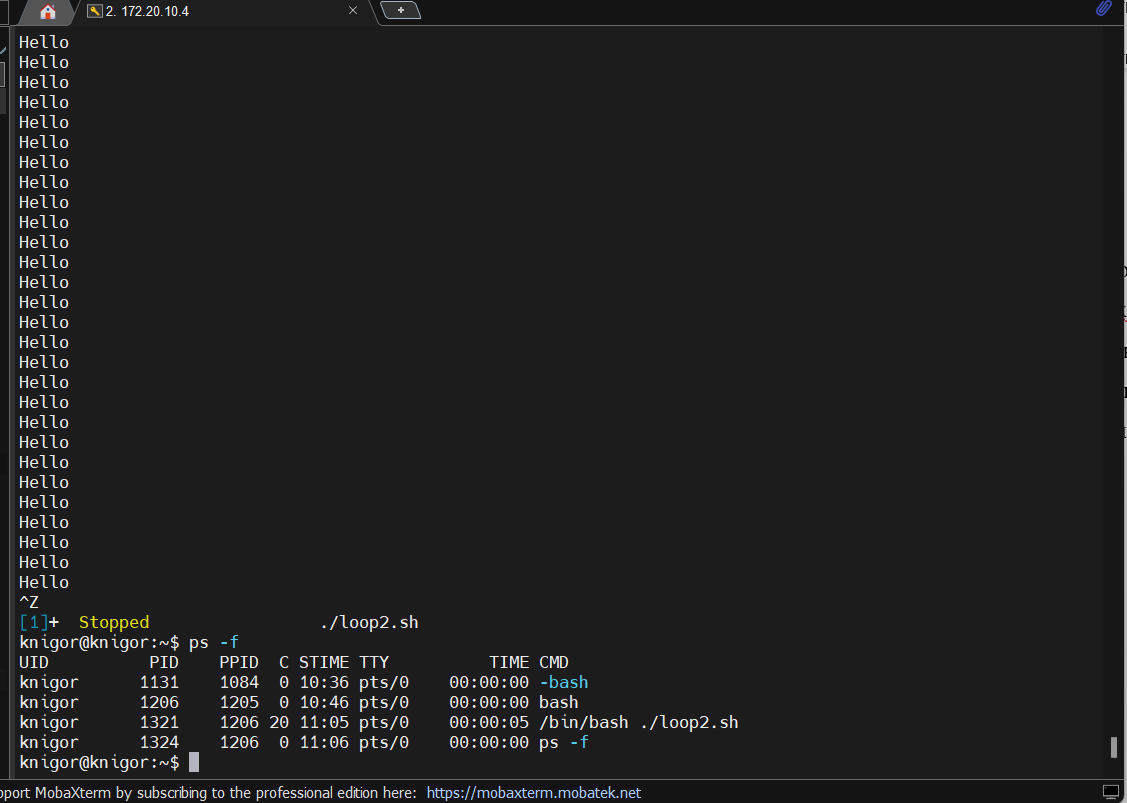
* **-с** - отображать информацию планировщика;
* **-f** - вывести максимум доступных данных, например, количество потоков;
* **-F** - аналогично -f, только выводит ещё больше данных;
* **-l** - длинный формат вывода;
* **-j, (j)** - вывести процессы в стиле Jobs, минимум информации;
* **-M, (Z)** - добавить информацию о безопасности;
* **-o, (o)** - позволяет определить свой формат вывода;
* **--sort, (k)** - выполнять сортировку по указанной колонке;
* **-L, (H)**- отображать потоки процессов в колонках LWP и NLWP;
* **-m, (m)** - вывести потоки после процесса;
* **-V, (V)** - вывести информацию о версии;
* **-H** - отображать дерево процессов;

При использовании опции**-f** команда выдает такие колонки:

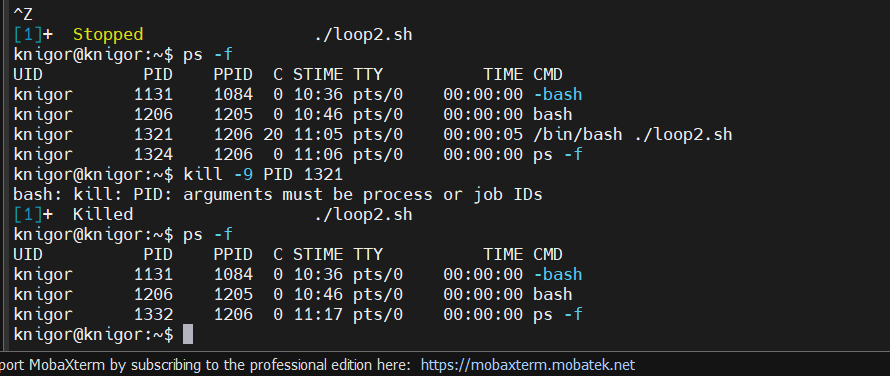
* **UID** - пользователь, от имени которого запущен процесс;
* **PID** - идентификатор процесса;
* **PPID** - идентификатор родительского процесса;
* **C** - процент времени CPU, используемого процессом;
* **STIME** - время запуска процесса;
* **TTY** - терминал, из которого запущен процесс;
* **TIME** - общее время процессора, затраченное на выполнение процессора;
* **CMD**- команда запуска процессора;
* **LWP** - показывает потоки процессора;
* **PRI** - приоритет процесса.

Далее с помощью редактора vim написать два сценария loop и loop2

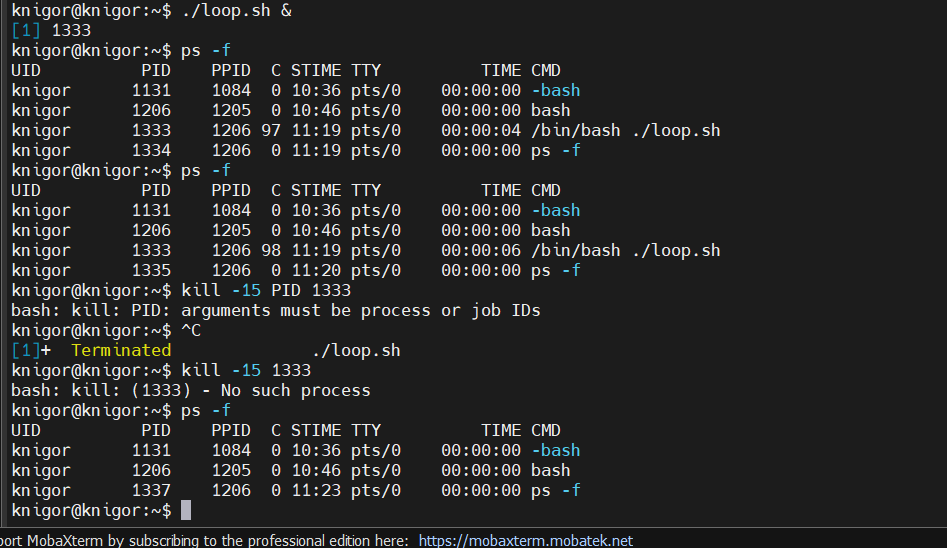
Создаем два файла vim loop.sh заполняем его, аналогично делаем с loop2, только в loop2 у нас будет выводится Hello в цикле, с помощью chmox +x loop2.sh дадим команду на запуск и запустим скрипт через команду ./loop2.sh, После запуска ждем 1 секунду и останавливаем выполнение скрипта подав сигнал STOP ctr + Z и пишем команду ps -f, смотрим на выведенную информацию



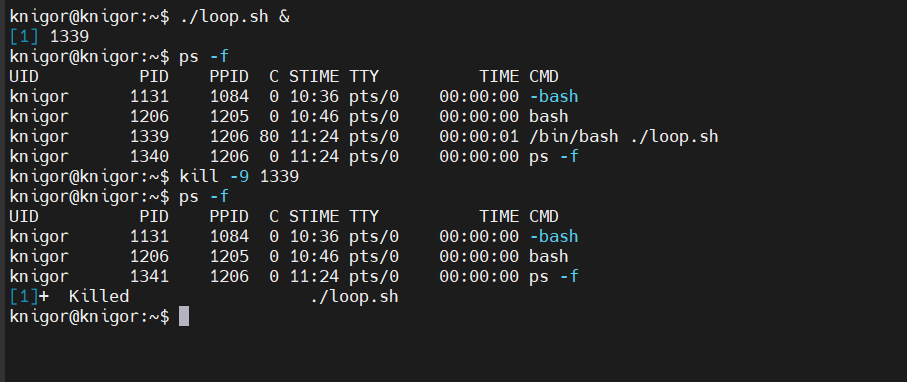
Далее убиваем процесс kill – 9 PID. После убийства процесса мы видим изменения на скриншоте ниже



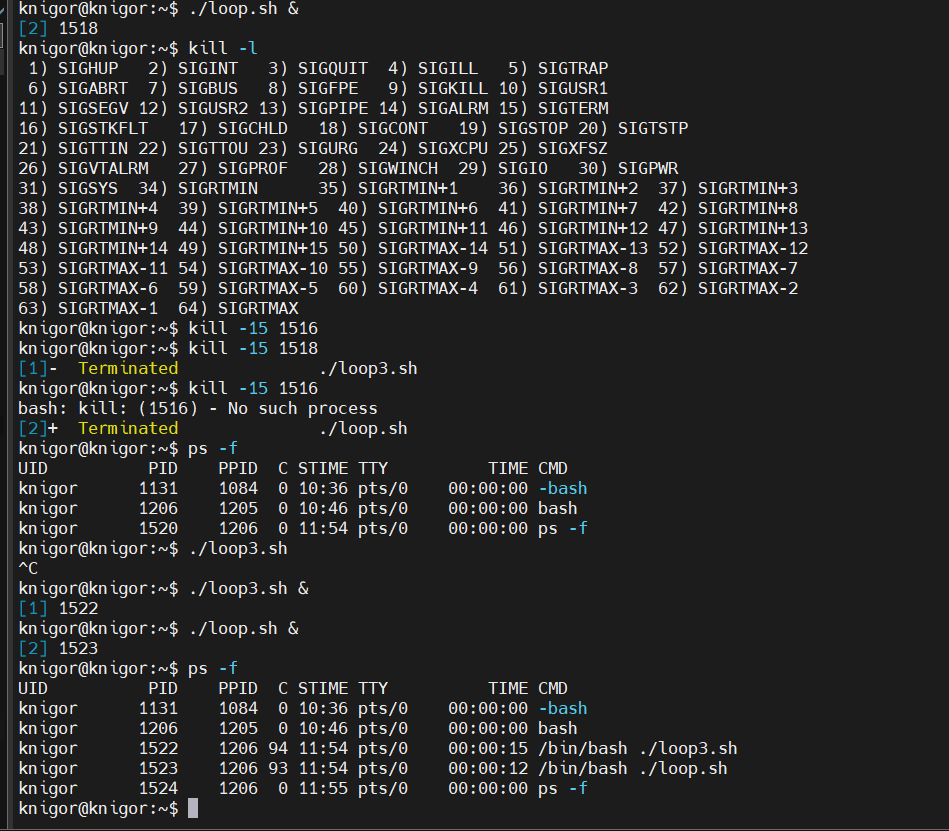
Далее нужно в фоновом режиме запустить loop.sh , сделать это можно с помощью & пропишем два раза ps -f, а затем kill -15 сигнал SIGTERM - немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы;



Далее еще раз запустим в фоне, а затем убьем процесс kill -9 PID, ниже мы видим результат выполнения работы

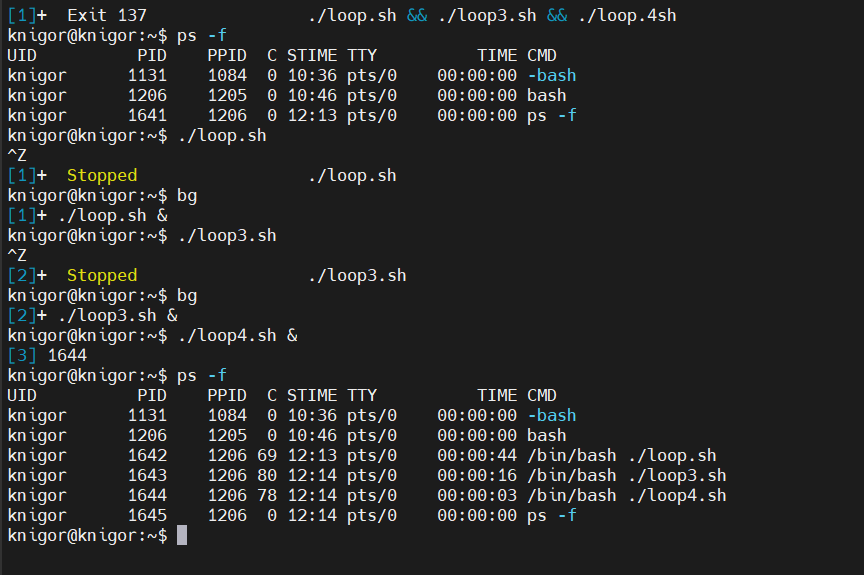


Ниже приведет результат работы пунктов 12 и 13



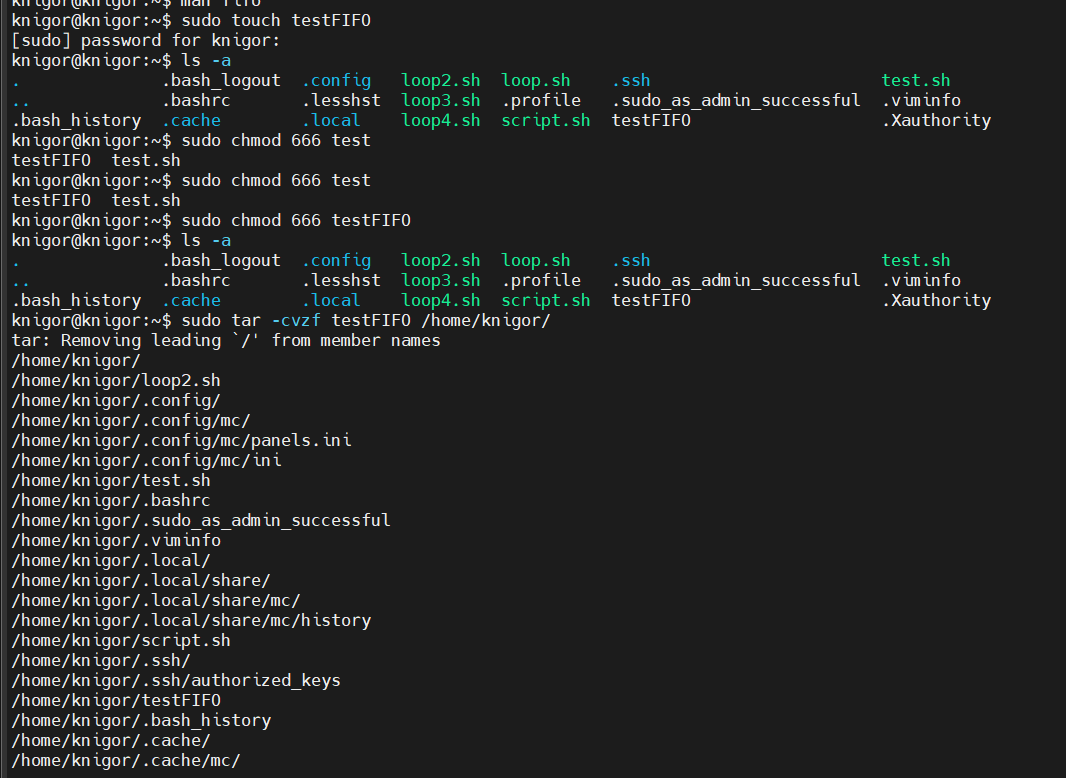
1. **Вторая часть**

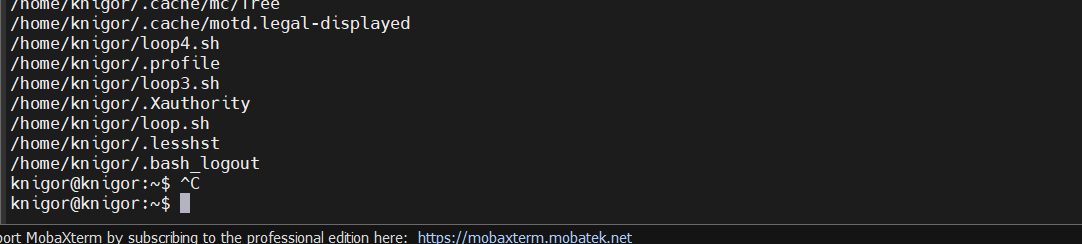
Запустим три задачи, две в интерактивном, одну в фоновом, Переводим одну из задач интерактивной в фоновую, потом сделать наоборот. Результат выполнения работы представлен ниже.



Мы запускаем ./loop.sh потом с помощью сигнала ctr + Z останавливаем его, потом через bg возобновляем, делаем так же с ./loop3.sh и ./loop4.sh & запускаем в фоне, ps -f выводим результат.

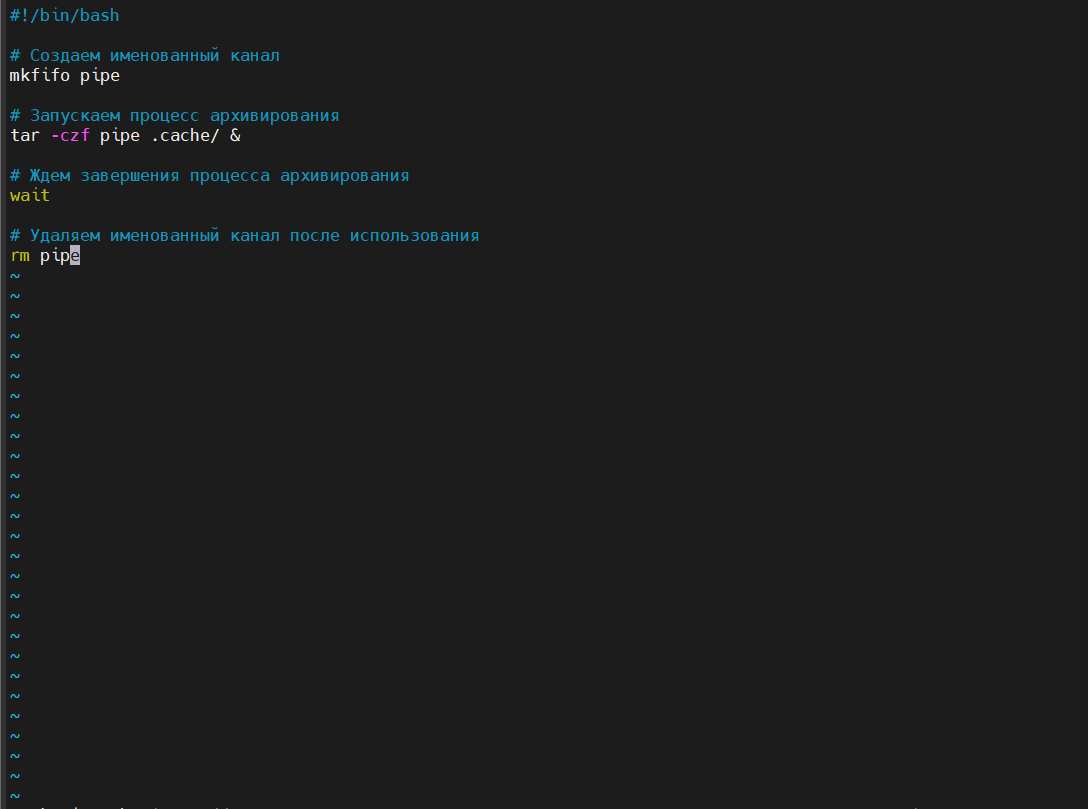
Чтобы создать именованный канал в Linux Ubuntu, вам нужно использовать инструмент FIFO (First In, First Out). FIFO – это особый вид файла, который работает как канал. Когда процесс записывает данные в FIFO, следующий процесс, который читает из FIFO, получает эти данные.



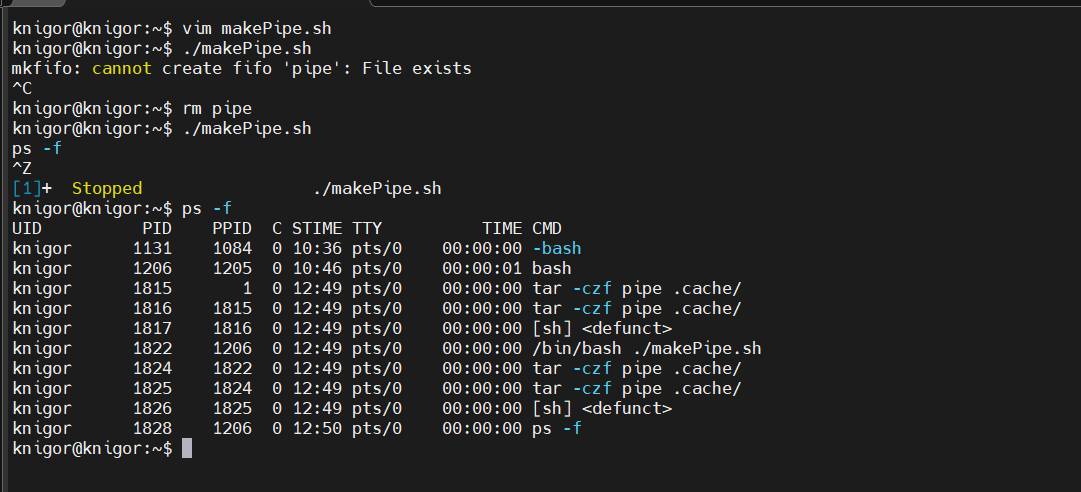


Далее выполним пункт b, я напишу скрипт

Содержание скрипта

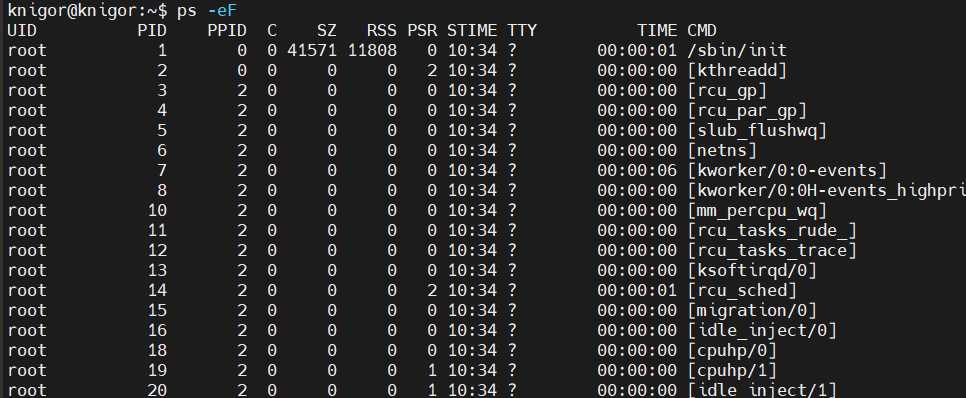


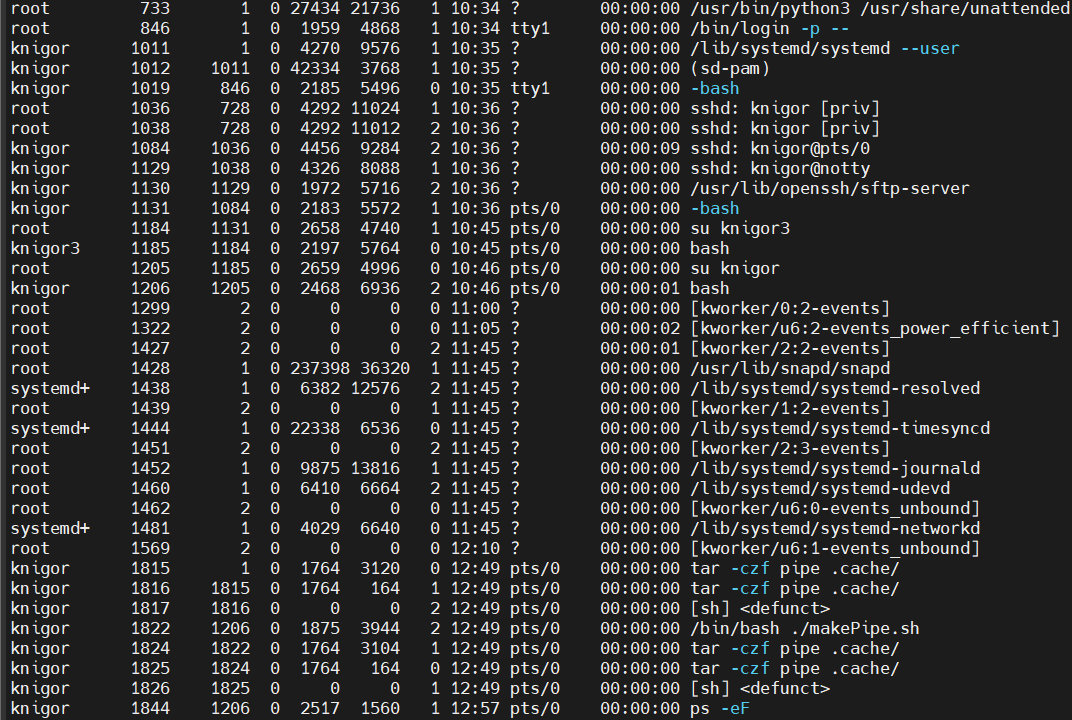
Ниже представлены результаты выполнения



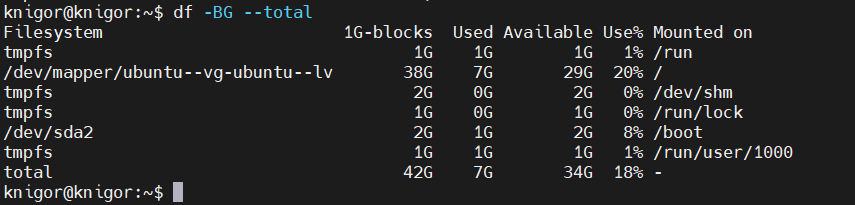
1. **Третья часть**

Вариант №4 С помощью команды ps -ef мы выводим полный список процессов, у всех пользователей.

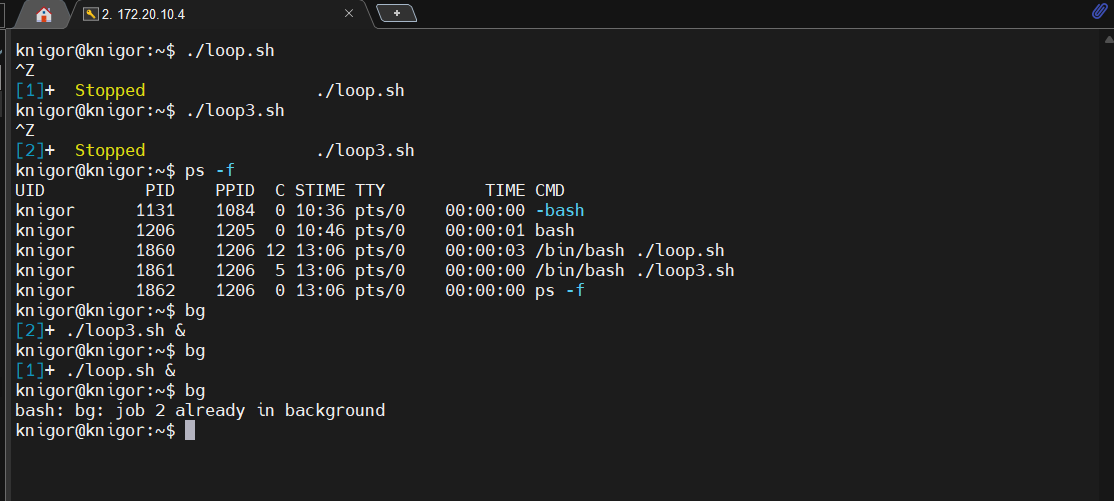




С помощью команды df -BG –total мы сможем посмотреть размер образа, сколько было выделено на операционную систему

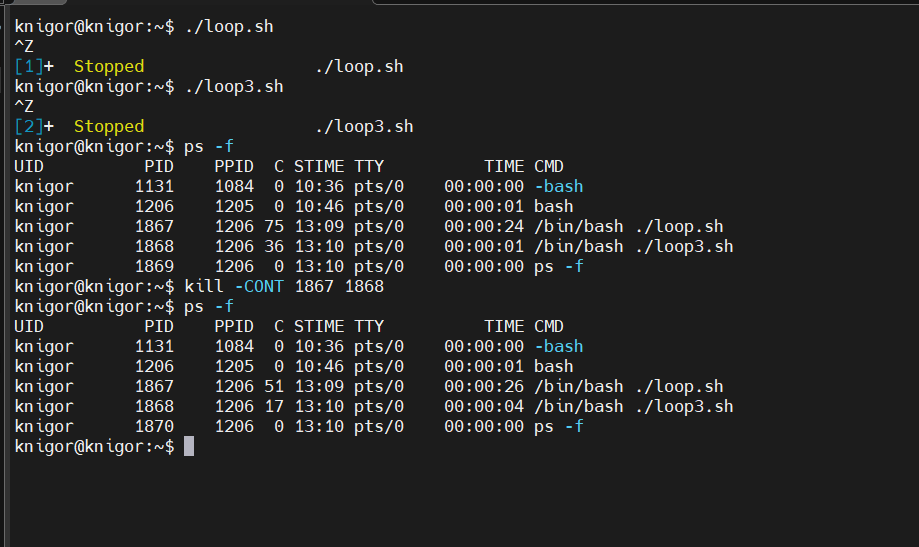


С помощью команды SIGTSTP это ctr + Z приостановим процесс текущего пользователя, через несколько секунд запустим снова



Выше мы видим, как после запуска первого скрипта, я остановил скрипт, аналогично сделал для второго скрипта, затем проверил состояние через команду ps -f, а затем через команду bg возобновил два скрипта

Так же это можно сделать используя команду остановы kill -TSTP [pid] и для запуска kill -CONT [pid]

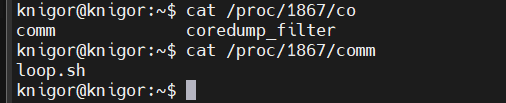


Далее по заданию нужно определить пользователя который запустил последний процесс в системе, узнать его имя, имя процесса и идентификатор, для этого будем использовать команду ps -eF, чтобы вывести полную информацию



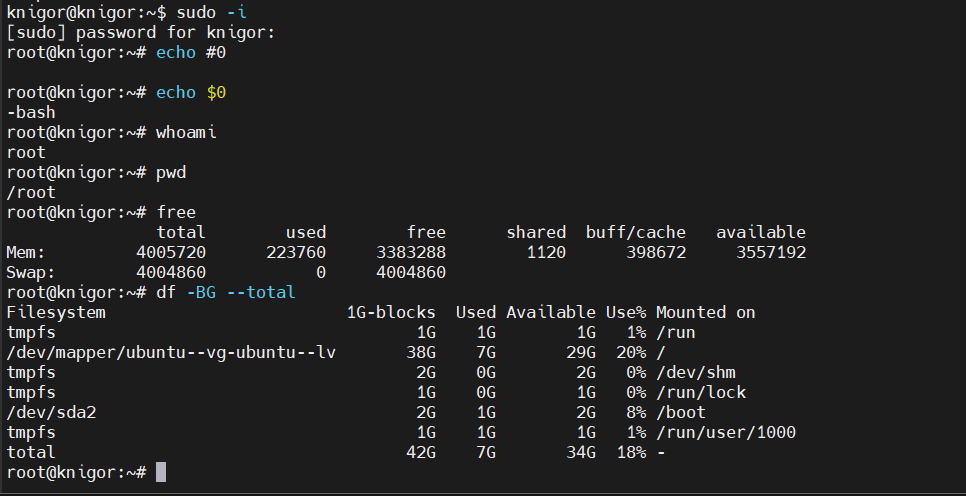


На скриншотах мы видим информацию по последним процессам пользователя knigor PID – это идентификатор процесса, чтобы, чтобы посмотреть имя процесса, нужно знать его PID, это есть, значит через команду cat /proc/1867/comm мы узнаем. Результат скриншотом ниже

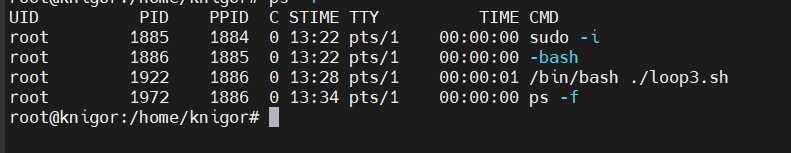


1. **Четвертая часть**

Для начала зайдем под супер-пользователя root, а затем начнем выполнять пункты из задания, ниже будут представлены скриншоты выполнения команд.

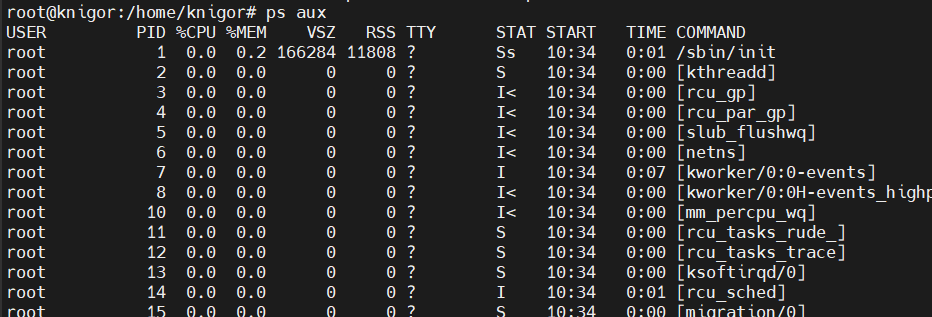


Прописав команду ps -f мы ответим на первые три подпункта в пункте 3. Результат выполнения на скриншоте ниже

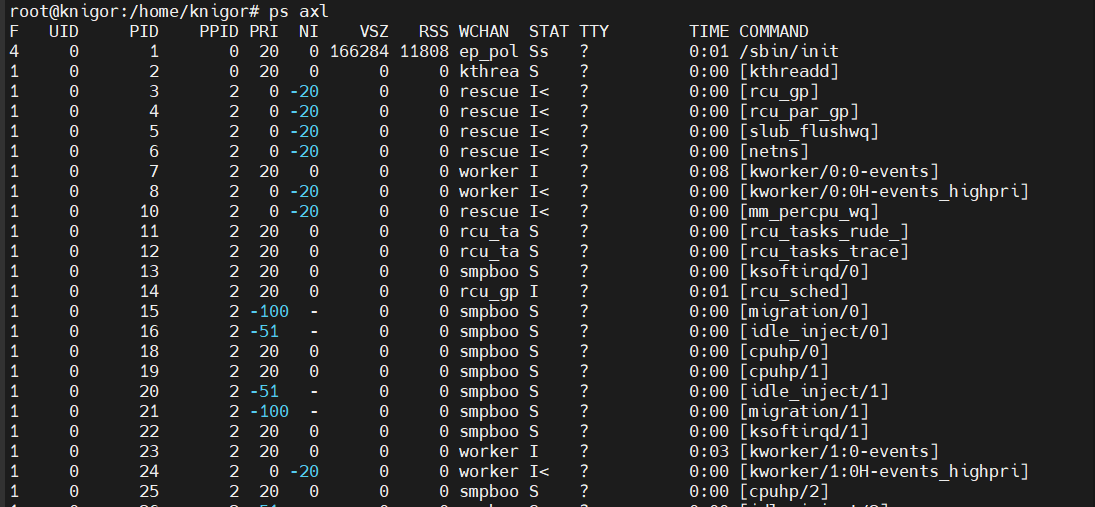


Командой ps -eF отобразим все процессы у всех пользователей

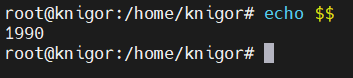
Команда ps aux чтобы отобразить команды управления процессами



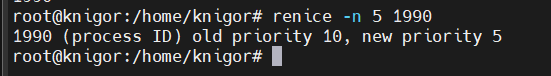
Для того чтобы определить текущее положение команды nice выполним команду, ps axl



Чтобы определить PID интерпретатора нужно написать команду echo $$



Ставим новый приоритет



Получаем информацию о процессах bash

