Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Отчет по Лабораторной работе №3

по курсу “ОС Linux”

Студент \_\_\_\_\_\_\_\_\_\_\_ Красиков И.А.

подпись, дата

Группа ПИ-21-1

Руководитель \_\_\_\_\_\_\_\_\_\_\_ Кургасов В.В.

подпись, дата

Липецк 2023 г.

[**Цель работы** 3](#_Toc1)

[**Ход работы** 3](#_Toc2)

[**Часть I** 3](#_Toc3)

[**Часть II** 8](#_Toc4)

[**Часть III** 9](#_Toc5)

[**Часть IV** 11](#_Toc6)

[**Вывод** 14](#_Toc7)

[**Ответы на контрольные вопросы** 14](#_Toc8)

# **Цель работы**

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

# **Ход работы**

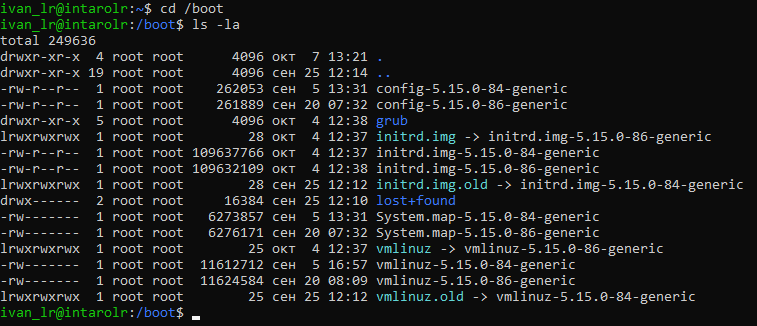
## **Часть I**

1) Войти в систему под пользовательской учетной записью (не root)



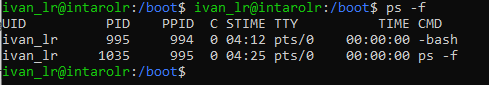
2) Найти файл с образом ядра. Выяснить по имени файла номер версии Linux.

Файл с версией ядра можно найти в каталоге /boot, файл называется vmlinuz.



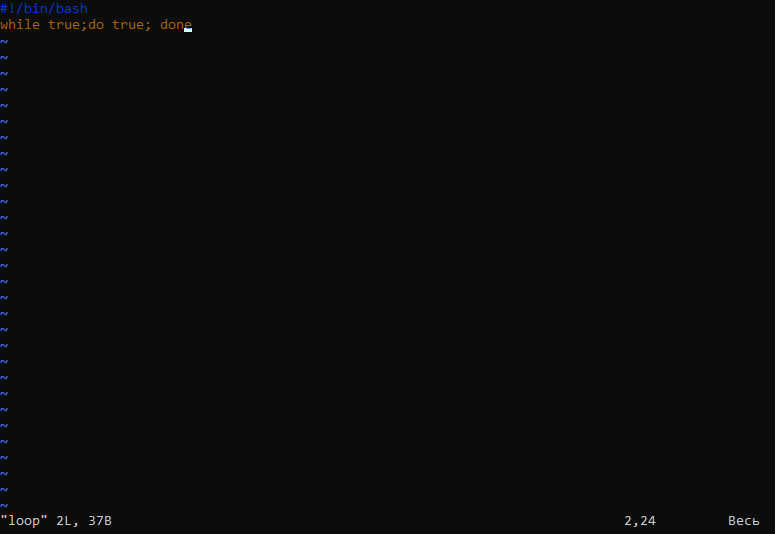
3) Посмотреть процессы ps -f. Прокомментировать, изучив предварительно справку командой man ps.

Команда ps отображает список процессов в ОС Linux. Опция –f выводит максимум доступных данные, например, количество потоков.



В процессах мы можем увидеть bash и команду ps -f.

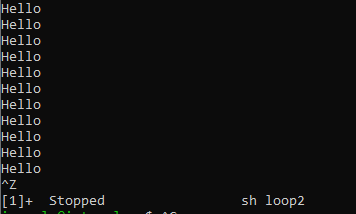
4) Написать с помощью редактора vi сценария loop и loop2.



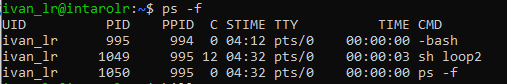
5) Запустить loop2 на переднем плане: sh loop2



6) Остановить, послав сигнал STOP.

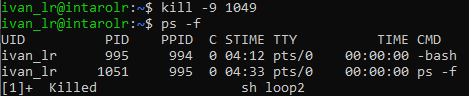


7) Посмотреть результат через команду ps –f.



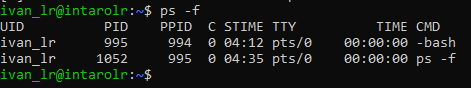
Мы видим что процесс до сих пор находиться в запущенных процессах.

8) Убить процесс loop2, послав сигнал kill -9 PID. Прокомментировать.

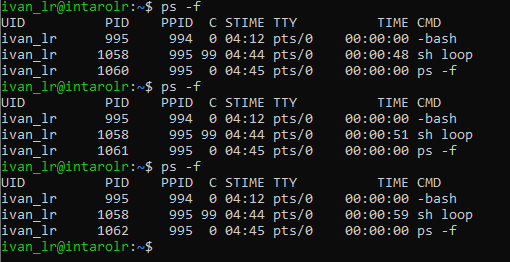


Мы видим что процесс отображается в ps –f, но с пометкой Killed.

Но если мы запустим ps –f, мы уже не увидим этот процесс.

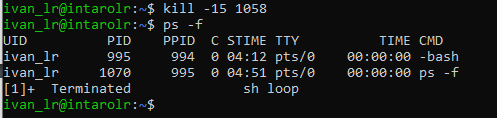


9) Запустить в фоне процесс loop: sh loop&. Не останавливая, посмотреть несколько раз: ps –f.



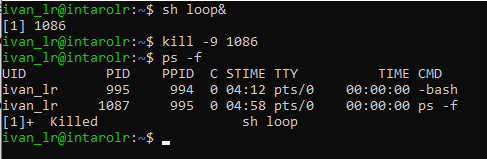
Мы можем увидеть, что процесс запущен в фоновом режиме, и меняется время, которое показывает сколько времени запущен процесс.

10) Завершить процесс loop командой kill -15 PID. Прокомментировать.

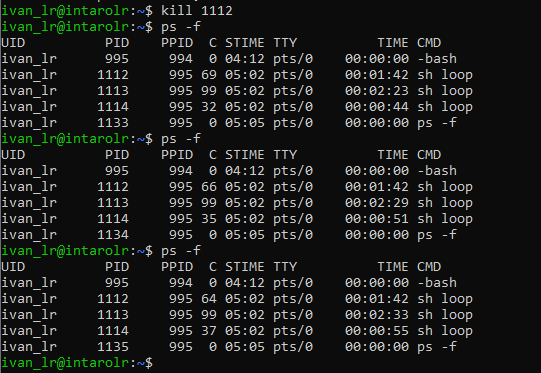
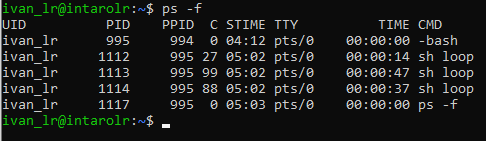
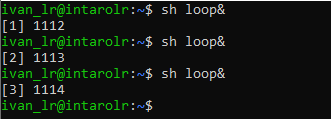


В данном случае отображается не Killed, а Terminated, он отличается тем, что это является программным завершение процесса, а не уничтожением процесса.

11) Третий раза запустить в фоне. Не останавливая, убить командой kill -9 PID.

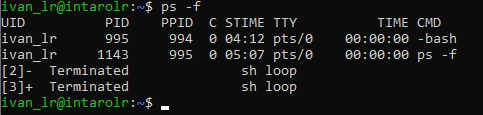


13) Запустить несколько процессов в фоне. Останавливать их и снова запускать. Записать результаты просмотра командой ps –f.



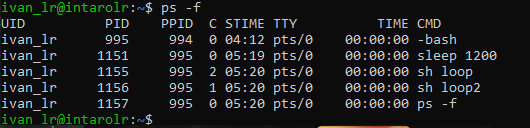
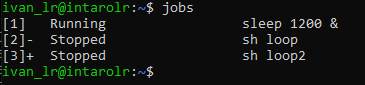
Введя команду kill PID, мы можем увидеть что процесс не завершается полностью, а только останавливается, это видно по времени работы процесса.

С помощью команды killall loop, можно завершить все процессы.



## **Часть II**

1) Запустить в консоли на выполнение три задачи: две в интерактивном режиме, одну – в фоновом.



2) Провести эксперименты по переводу из интерактивного режима в фоновый и наоборот.

Для этого можно воспользоваться командой bg

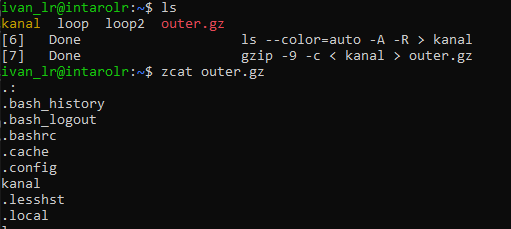
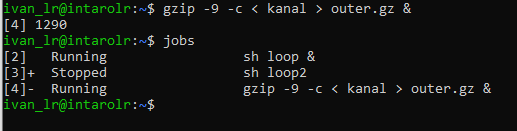


Также задачу можно перевести из фонового режима в интерактивный с помощью команды fg



3) Создать именованный канал для архивирования и осуществить передачу в канал списка домашнего каталога вместе с подкаталогами.

Для этого используется команда mkfifo



## **Часть III**

**Вариант 3**

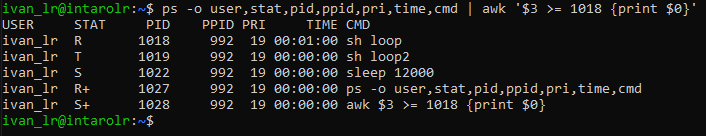
1) Сгенерировать следующую информацию о m (m > 2) процессах системы , имеющих значения идентификатора больше заданного n: флаг – сведения о процессе, статус, PID, PPID, приоритет, использованное время и имя программы.

2) Завершить выполнение двух процессов, владельцем которых является текущий пользователь, первый процесс завершить с помощью сигнала SIGKILL, задав его имя, второй – с помощью сигнала SIGINT, задав его номер.

3) Через символ “ : ” вывести идентификаторы процессов для которых родителем является командный интерпретатор.

**Выполнение заданий**

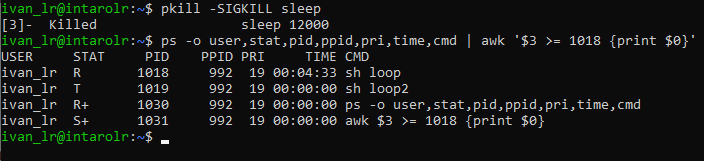
1 пункт:



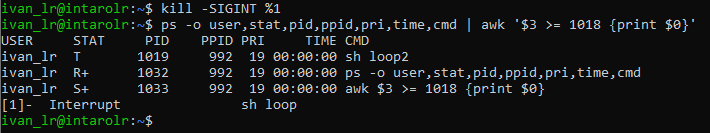
Для этого мы использовали команду ps с опцией –o, для которой можно задать какие данные будут отображаться, также используем awk чтобы задать с какого идентификатора будет отображение.

2 пункт:

По имени можно убить процесс с помощью команды pkill.

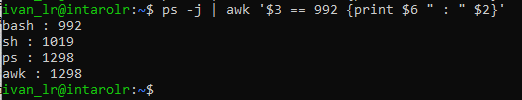


Чтобы удалить по номеру процесс нужно ввести номер процесса через знак %, перед этим узнать номер через команду jobs.



3 пункт:

Для того чтобы вывести процессы для которых родителем является командный интерпретатор нужно использовать awk и проверять PPID, который ровняется 992, потому что bash имеет PID 992.



## **Часть IV**

1) Открыть окно интерпретатора команд.



2) Вывести общую информацию о системе:

a) Вывести информацию о текущем интерпретаторе команд



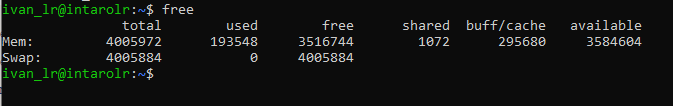
b) Вывести информацию о текущем пользователе



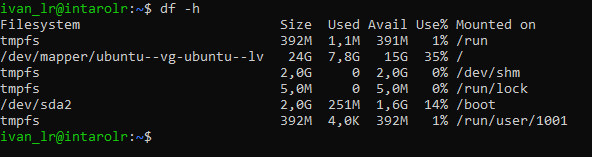
c) Вывести информацию о текущем каталоге



d) Вывести информацию о оперативной памяти и файле подкачки

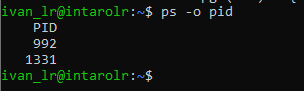


e) Вывести информацию о дисковой памяти

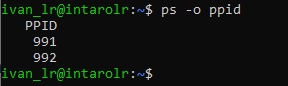


3) Выполнить команды получения информации о процессах:

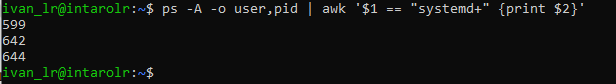
a) Получить идентификатор текущего процесса (PID)



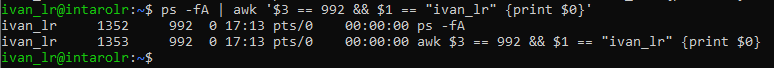
b) Получить идентификатор родительского процесса (PPID)



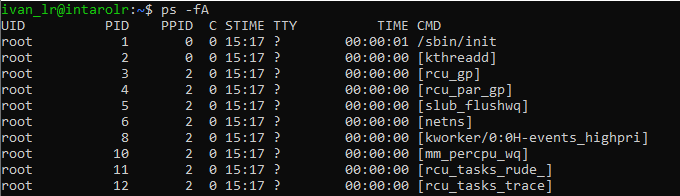
c) Получить идентификатор инициализации системы



d) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд

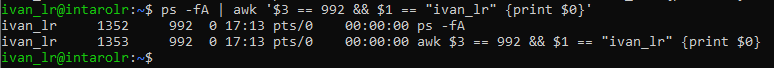


e) Отобразить все процессы

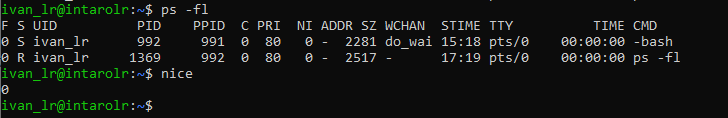


4) Выполнить команды управления процессами:

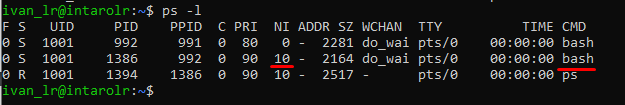
a) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе.



b) Определить текущее значение nice по умолчанию



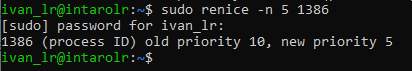
c) Запустить интерпретатора bash с понижением приоритета nice –n 10 bash



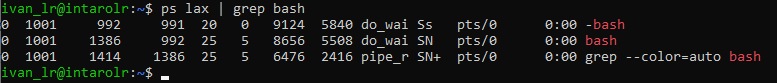
d) Определить PID запущенного интерпретатора



e) Установить приоритет запущенного интерпретатора равным 5 renice –n 5 <PID процесса>



f) Получить информацию о процессах bash



# **Вывод**

В процессе выполнения лабораторная работы, я ознакомился на практике с понятием процесса в операционной системе. Приобрел опыт и навыки управления процессами в операционной системе Linux.

# **Ответы на контрольные вопросы**

1) Перечислите состояния задачи в ОС Linux.

Ответ: В ОС Ubuntu задачи могут находиться в 4 состояниях: Running – выполняемый, Sleeping – в ожидании, T – приостановлен, Zombie – зомби.

2) Как создаются задачи в ОС Ubuntu?

Ответ: Задачи в Linux создаются следующим путем: процесс /bin/bash клонирует себя системным вызовом fork(), при этом создается клон /bin/bash с новым PID (идентификатор процесса) и PPID - равный PID родителя. Клон выполняет системный вызов exec с указанием на исполняемый файл и заменяет свой код - кодом исполняемого файла (родительский процесс при этом ждет завершения потомка - wait)

3) Назовите классы поток ОС Linux.

Ответ:

* Потоки реального времени, обслуживаемые по алгоритму FIFO.
* Потоки реального времени, обслуживаемые в порядке циклической очереди.10
* Потоки разделения времени.

4) Как используется приоритет планирования при запуске задачи?

Ответ: При планировании задач приоритет используется при вычислении «добродетели» процесса. Цель алгоритма планирования состоит в том, чтобы обеспечить грубое пропорциональное соответствие качества обслуживания приоритету, то есть чем выше приоритет, тем меньше должно быть время отклика и тем большая доля процессорного времени достанется процессу

5) Объясните, что произойдет, если запустить программу в фоновом режиме без подавления потока вывода.

Ответ: Процесс будет выполняться в фоновом режиме и завершиться независимо.

6) Объясните разницу между действием сочетаний клавиш Ctrl^Z и Ctrl^С.

Ответ: **Ctrl** + C используется для завершения процесса с помощью сигнала SIGINT , другими словами , это вежливый убивать . **Ctrl** + **Z** используется для приостановки процесса путем отправки ему сигнала SIGTSTP , который подобен сигналу сна, который может быть отменен, и процесс может быть возобновлен снова.

7) Опишите, что значит каждое поле вывода команды jobs.

Ответ: Первое поле – это номер процесса, второе поле – это состояние процесса, третье поле – это название процесса.

8) Назовите главное отличие утилиты top от ps.

Ответ: В отличие от ps команда top отображает состояние процессов и их активность в реальном режиме времени.

9) В чем отличие результата выполнения команд top и htop?

Ответ: В htop, мы можем взаимодействовать с процессами в режиме реального времени.

10) Какую комбинацию клавиш нужно использовать для принудительного завершения задания, запущенного в интерактивном режиме?

Ответ: ctrl+d

11) Какую комбинацию клавиш нужно использовать для приостановки задания, запущенного в интерактивном режиме?

Ответ: ctrl+z

12) Какая команда позволяет послать сигнал конкретному процессу?

Ответ: kill

13) Какая команда позволяет поменять поправку к приоритету уже запущенного процесса?

Ответ: renice

14) Какая команда позволяет запустить задание с пониженным приоритетом?

Ответ: nice

15) Какая команда позволяет запустить задание с защитой от прерывания при выходе из системы пользователя?

Ответ: nohup

16) Какой процесс всегда присутствует в системе и является предком всех процессов?

Ответ: systemd

17) Каким образом можно запустить задание в фоновом режиме?

Ответ: Указав в конце команды запуска символ &

18) Каким образом задание, запущенное в фоновом режиме, можно перевести в интерактивный режим?

Ответ: С помощью команды fg

19) Каким образом приостановленное задание можно перевести в интерактивный режим?

Ответ: Можно также использовать команду fg

20) Что произойдет с заданием, выполняющимся в фоновом режиме, если оно попытается обратиться к терминалу?

Ответ: Отобразиться ошибка

21) Сколько терминалов может быть открыто в одной системе? Как перемещаться между терминалами (какие комбинации клавиш необходимо использовать)?

Ответ: 6 терминалов, Ctrl+Alt+(F1-F6)

22) В чем отличие идентификаторов PID и PPID? При каких условиях возможна ситуация, когда PPID равен нулю или отсутствует?

Ответ: PPID – это PID родителя процесса, PPID равен нулю или отсутствует в ситуациях, когда нет родителя процесса.

23) Поясните, от чего зависит максимальное значение PID.

Ответ: В Linux максимальный возможный идентификатор процесса по умолчанию ограничен 32 тысячами

24) В каком случае, при создании нового процесса, его идентификатор (PID) будет меньше, чем у процесса, запущенного ранее?

Ответ: При переполнении диапазона PID.