Министерство цифрового развития, связи и массовых коммуникаций

Российской Федерации Ордена Трудового Красного Знамени

федеральное государственное бюджетное образовательное

учреждение высшего образования

Московский технический университет связи и информатики

Кафедра «Системного программирования»

Лабораторная работа №4. Часть 1.

по дисциплине

«Операционные системы»

Выполнила: студентка группы БСТ2104

Первухина А. А.

Проверила: Алексанян Диана Ашотовна

Москва 2023

**Содержание**

[**Задание 1. Управление процессами** 3](#_Toc150444289)

[**Задание 1.1. Мониторинг производительности системы** 3](#_Toc150444290)

[**Задание 1.2. Создание скрипта для получения статистики** 12](#_Toc150444291)

[**Задание 1.3. Создание автозапускаемой службы** 14](#_Toc150444292)

[**Дополнительное задание** 16](#_Toc150444293)

[**Задание 2. Реализация обмена данными между процессами** 18](#_Toc150444294)

[**Задание 2.1. Разделяемая память** 19](#_Toc150444295)

[**Задание 2.2. Сокеты** 20](#_Toc150444296)

[**Задание 2.3. Каналы** 22](#_Toc150444297)

[**Вывод** 24](#_Toc150444298)

**Цели работы:**

**Задание 1.1. Мониторинг производительности системы**

1. Выполните проверку системы при помощи команды top. Отсортируйте процессы по: объему используемой памяти; времени работы; идентификатору; проценту использования времени процессора.

2. Запустите новый процесс в системе, найдите его в выводе команды top. Измените приоритет запущенного процесса командой nice и убедитесь, что изменения отражены в выводе top.

3. С помощью команд atop и atopsar выполните сортировку процессов минимум по трем параметрам общей нагрузки.

4. При помощи команды mpstat и опции P выведите информацию по: определенному процессору, всем процессорам.

5. Выполните проверку системы при помощи команды pidstat. Выведите по четыре отчета статистики использования процессора для каждой активной задачи в системе с интервалом в три секунды. В отчете приведите ответ в виде снимков экрана по каждому пункту задания.

**Задание 1.2. Создание скрипта для получения статистики**

Напишите скрипт, осуществляющий сбор статистики использования процессоров и статистики работы процессов средствами mpstat и pidstat, выводящий данные в текстовый файл с именем yyyy-mm-dd\_h:m:s\_syslog.

В отчет включите текст скрипта, содержимое файла лога и вывод команды ls -l для директории сохранения логов скрипта.

**Задание 1.3. Создание автозапускаемой службы**

С помощью утилиты systemd зарегистрируйте свою службу по сбору статистики, запускающую скрипт из задания 2 при старте системы. Запустите службу и проверьте ее работу.

В отчете приведите снимок экрана с выводом команды ls -l для директории сохранения логов скрипта, статистику выполнения службы и текст файла .service.

**Дополнительное задание**

Создайте таймер для запуска созданной службы каждую минуту, запустите таймер и проверьте его статус. После нескольких (5-7) минут работы, остановите таймер. Отразите в отчете результат работы таймера (снимок экрана со статистикой работы таймера и вывод команды ls –l для директории сохранения логов скрипта).

**Выполнение**

**Пункт 1.** Выполните проверку системы при помощи команды top. Отсортируйте процессы по: объему используемой памяти; времени работы; идентификатору; проценту использования времени процессора.

Воспользуемся командой **top -o %MEM**

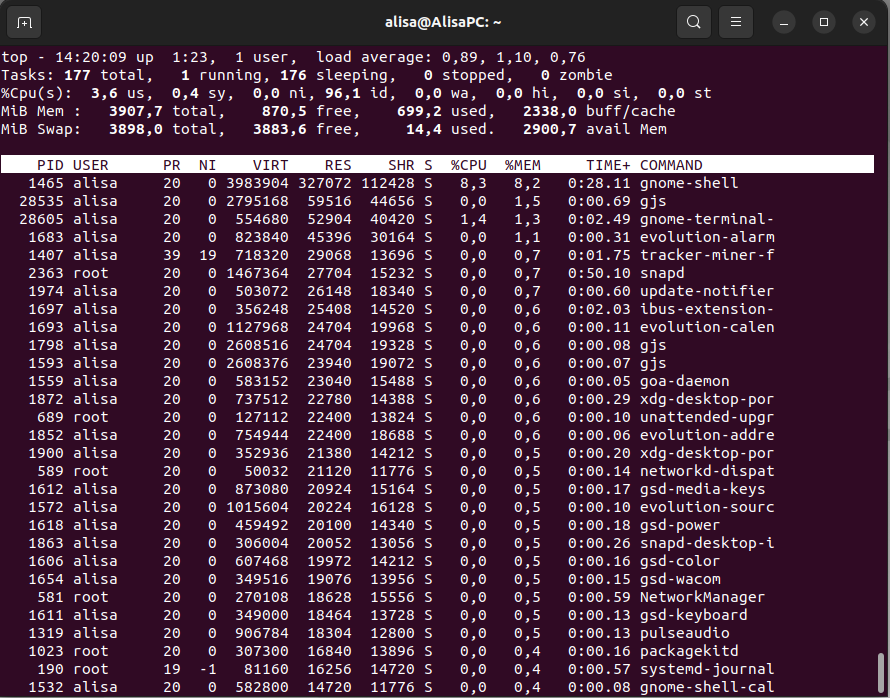


Рис. 1 — Вывод команды top с сортировкой M.

Воспользуемся командой **top -o TIME+**

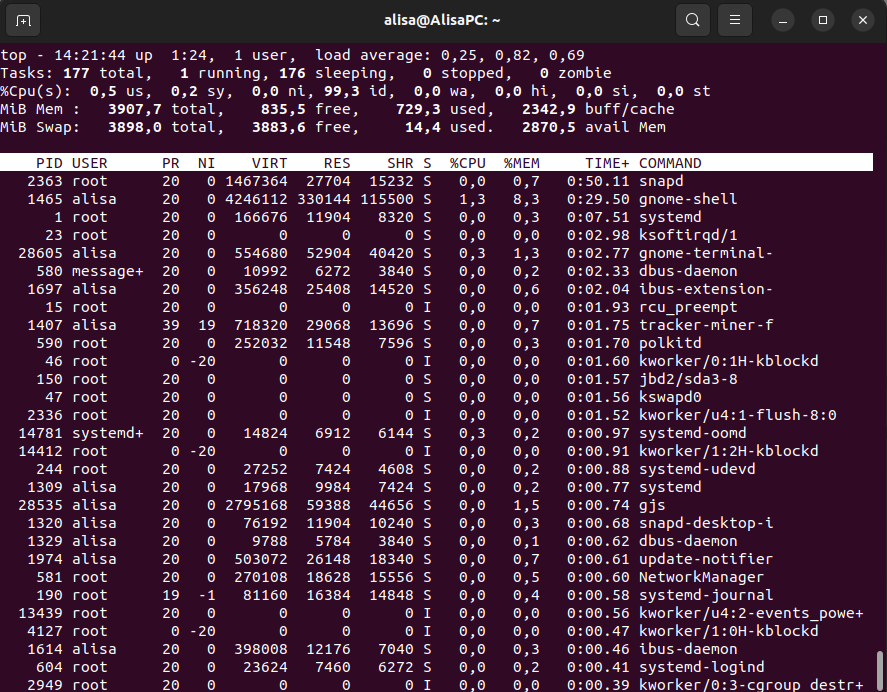
****

Рис. 2 — Вывод команды top с сортировкой T.

Воспользуемся командой **top -o PID** для сортировки по объему используемой памяти.

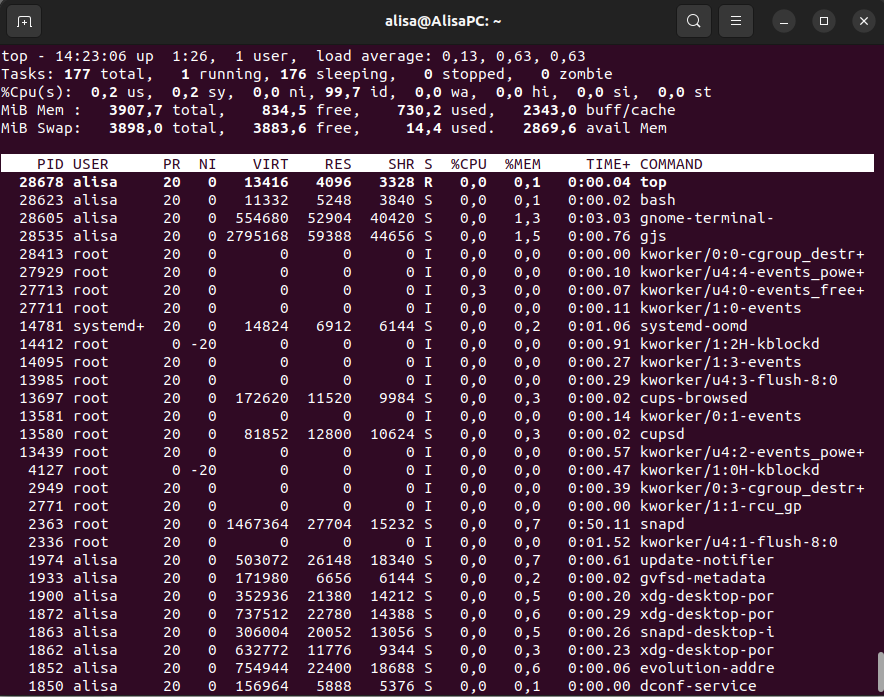


Рис. 3 — Вывод команды top с сортировкой P.

Воспользуемся командой **top -o %CPU**

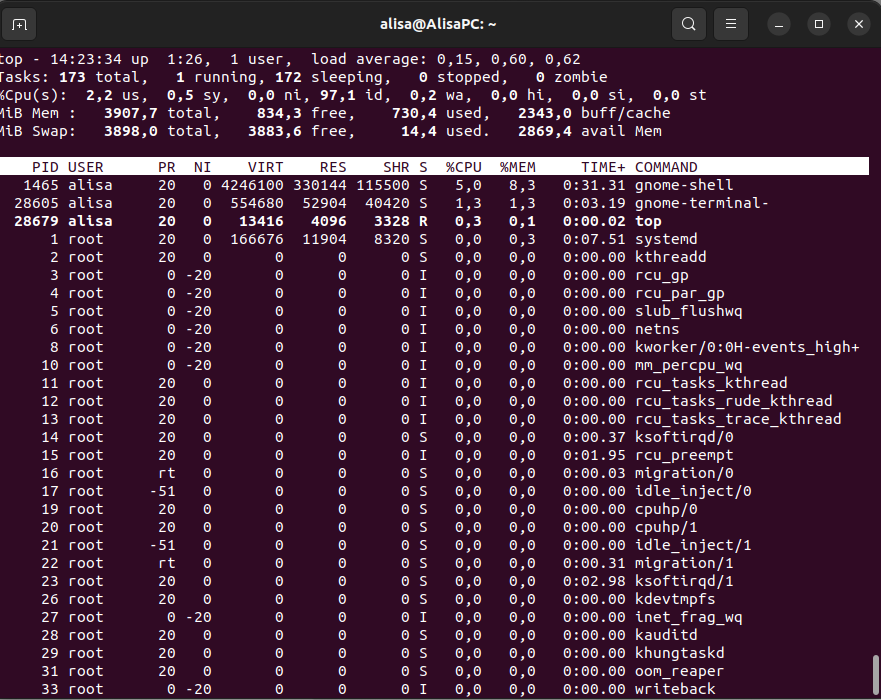
****

Рис. 4 — Вывод команды top с сортировкой N.

**Пункт 2.** Запустите новый процесс в системе, найдите его в выводе команды top. Измените приоритет запущенного процесса командой nice и убедитесь, что изменения отражены в выводе top

Открыла firefox, его PID 29857

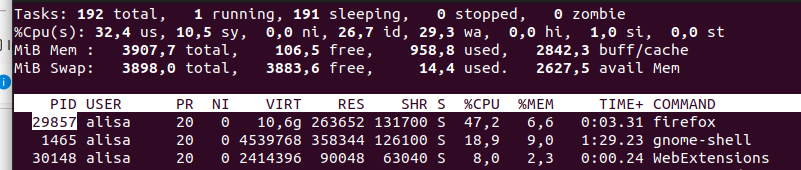
****

Рис. 5 – команда top

Воспользуемся командой sudo renice -n 10 -p 29857

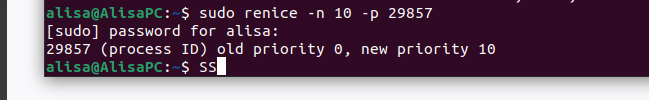
****

Рис. 6 – команда изменения приоритета

Результат при выводе команды top

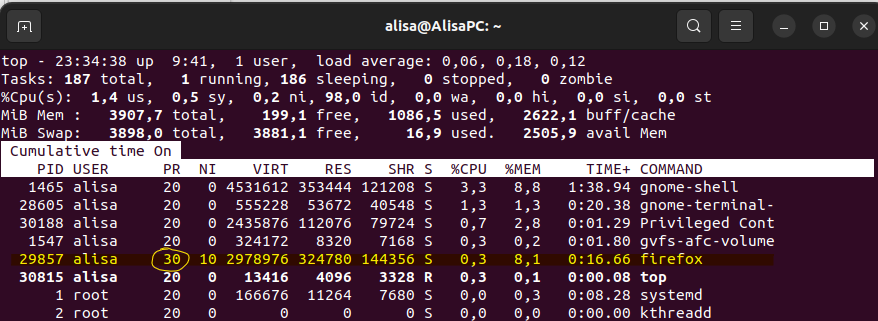
****

Рис. 7 – команда top

**Пункт 3.** С помощью команд atop и atopsar выполните сортировку процессов минимум по трем параметрам общей нагрузки.

Сортировка процессов по объёму использования памяти

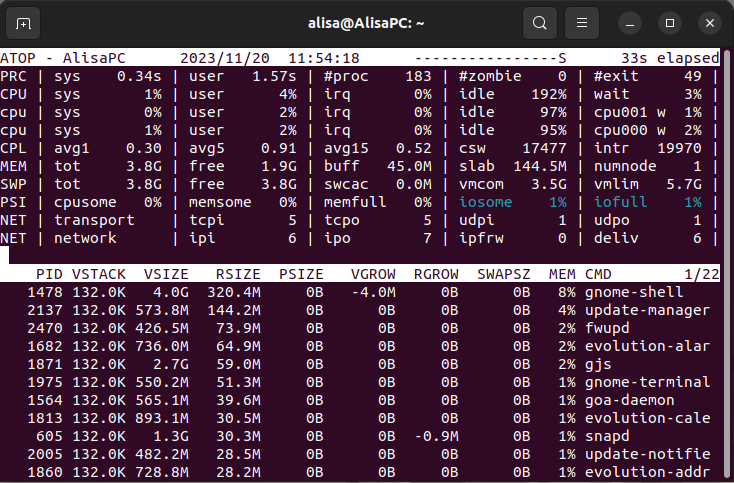
****

Рис. 8 – Вывод команды atop с ключом m

**Пункт 4.** Запросить список команд, работающих в фоновом режиме, в формате вывода командной строки

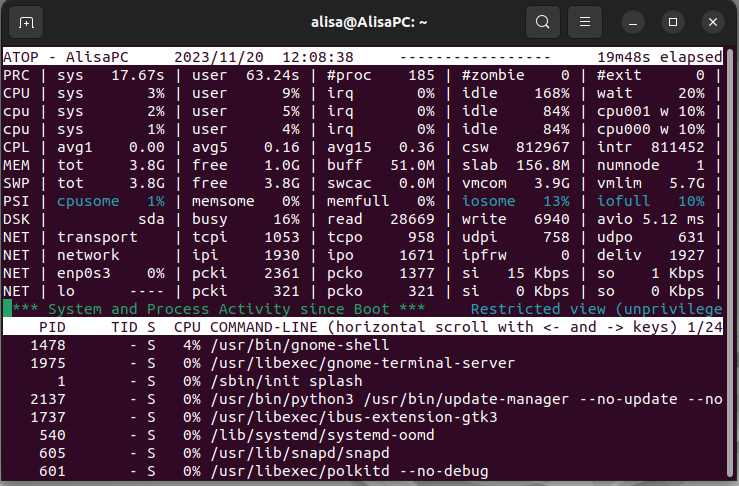
****

Рис. 8 – Вывод команды atop с ключом с

Сортировка по времени работы процесса

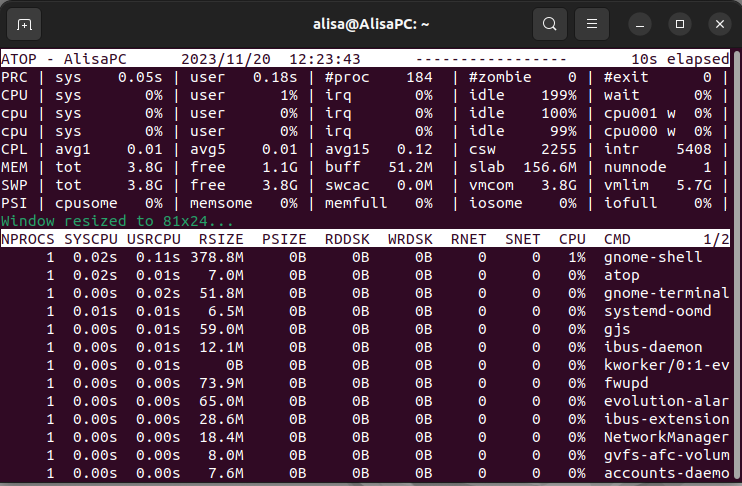
****

Рис. 8 – Вывод команды atop с ключом T

Анализ логов нагрузки на RAM 20 ноября

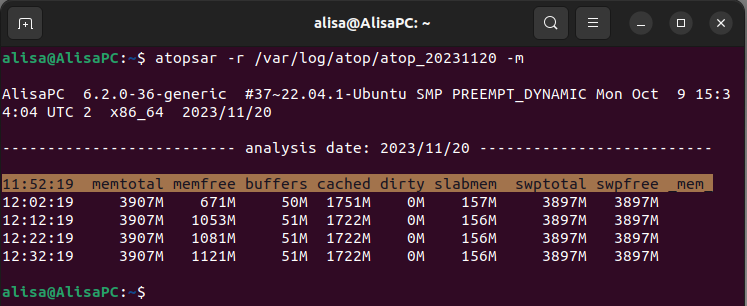
****

Рис. 9 – Вывод команды atopsar

Анализ логов нагрузки на сетевые интерфейсы

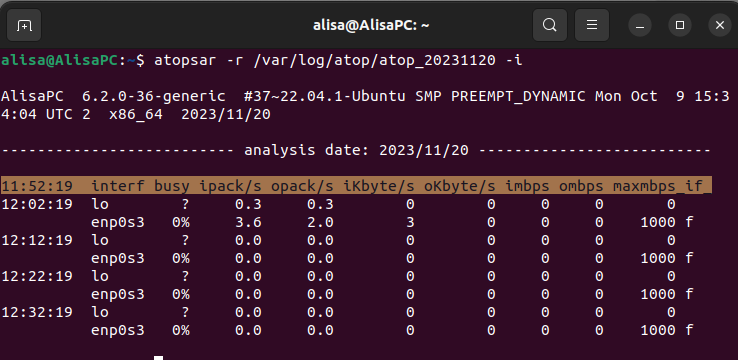


Рис. 10 – Вывод команды atopsar

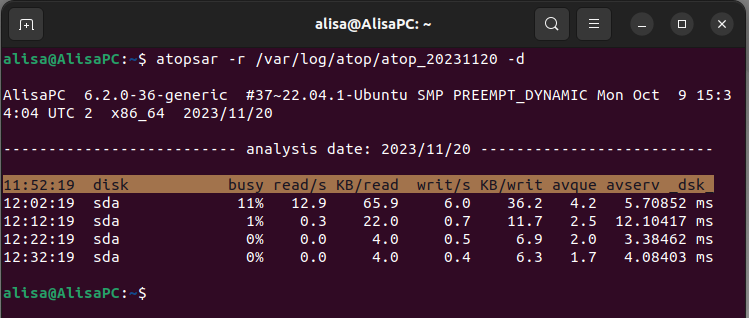


Рис. 11 – Вывод команды atopsar

Вывод информации по всем процессорам и определённому процессору

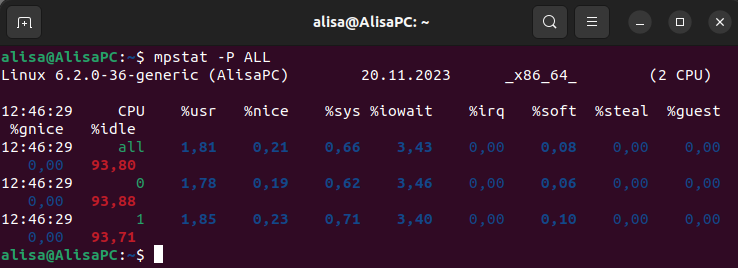


Рис. 12 – Вывод команды mpstat

**Пункт 5.** Вывод четырёх отчётов по статистике использования процессора для каждой активной задачи в системе с интервалом в 3 секунды

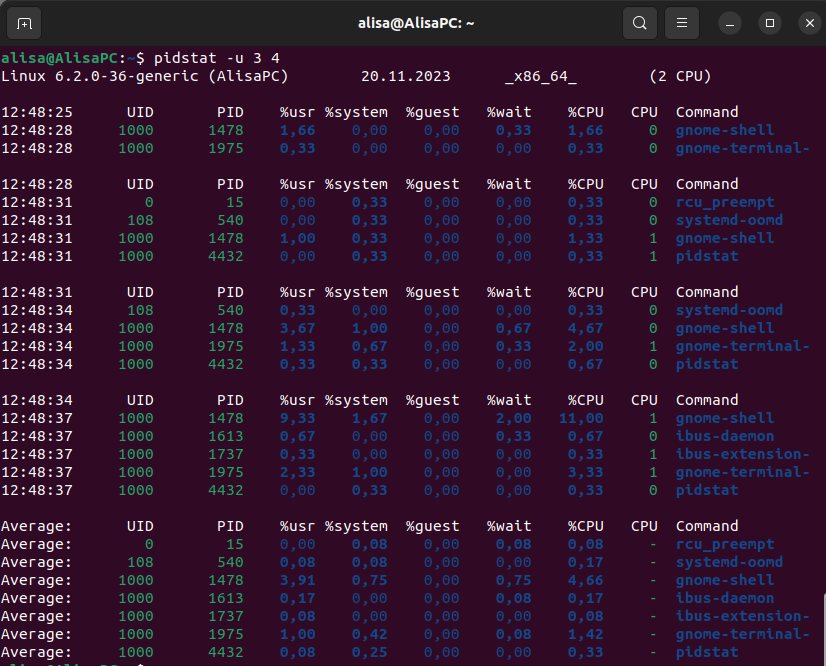


Рис. 13 – Вывод команды mpstat

# **Задание 1.2. Создание скрипта для получения статистики**

Напишите скрипт, осуществляющий сбор статистики использования процессоров и статистики работы процессов средствами **mpstat** и **pidstat**, выводящий данные в текстовый файл с именем yyyy-mm-dd\_h:m:s\_syslog.

Результаты выполнения представлены на рисунках 14-15:

Текст скрипта “collect\_logs.sh”

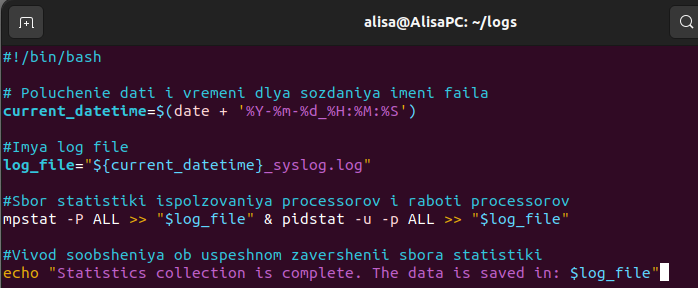


Рис. 14 – текст скрипта

Создание и запуск скрипта, вывод содержимого директории сохранения логов и содержимого файла лога

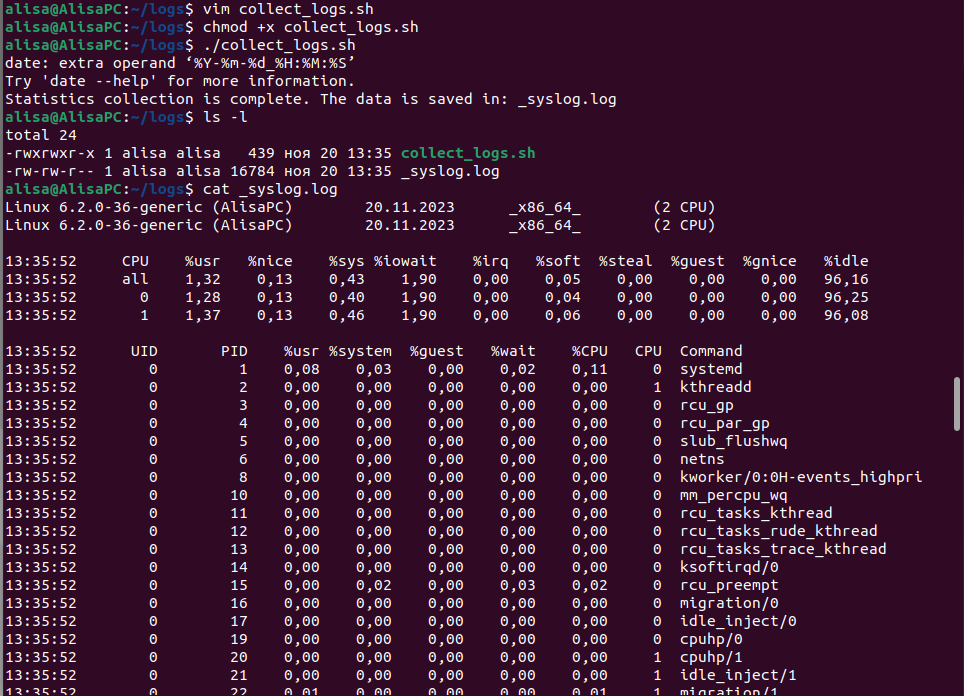
****

Рис. 15 – запуск скрипта

# **Задание 1.3. Создание автозапускаемой службы**

С помощью утилиты **systemd** зарегистрируйте свою службу по сбору статистики, запускающую скрипт из задания 1.2 при старте системы. Запустите службу и проверьте её работу.

Результаты выполнения представлены на рисунках 16

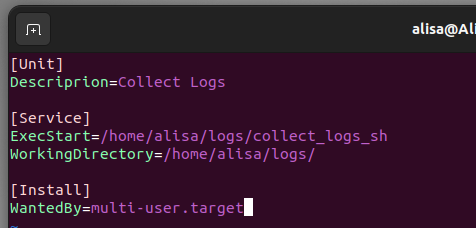
****

Рис. 16 – текст службы

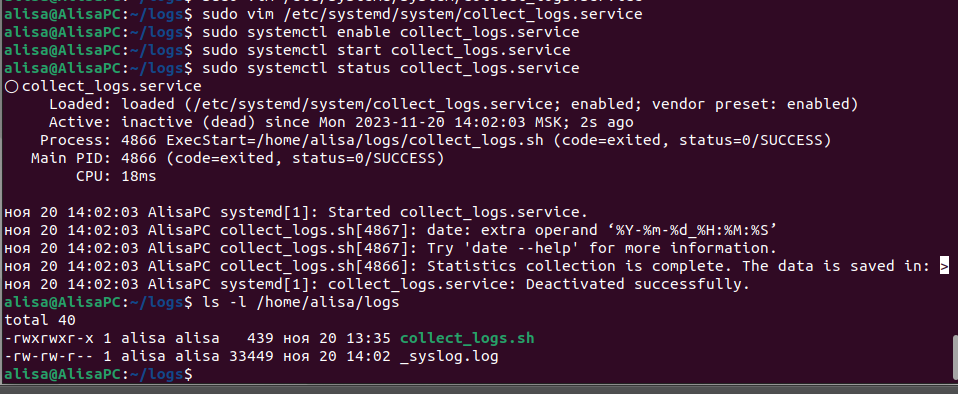
****

Рис. 17 – запуск службы

# **Дополнительное задание**

Создайте таймер для запуска созданной службы каждую минуту, запустите таймер и проверьте его статус. После нескольких (5-7) минут работы, остановите таймер.

Результаты выполнения представлены на рисунках

Рисунок 18 – Создание, запуск и проверка работы таймеры для запуска созданной службы по сбору статистики каждую минуту

Рисунок 19 – Текст файла “collect\_logs.timer”

Рисунок 20 – Вывод содержимого директории сохранения логов

# **Вывод**

В результате проделанной работы были изучены основные команды и инструменты для управления процессами, приобретены практически навыки работы со средствами мониторинга производительности в Linux и автоматизации запуска процессов, а также были изучены средства коммуникации процессов в ОС GNU/LINUX и получены практически навыки по использованию механизмов межпроцессорного взаимодействия.