ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ»

В Г. СМОЛЕНСКЕ

Кафедра вычислительной техники

Дисциплина: ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Лабораторная работа № 6.

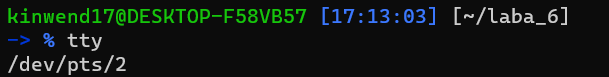
ПЛАНИРОВАНИЕ ЗАДАНИЙ И УПРАВЛЕНИЕ ПРИОРИТЕТАМИ ПРОЦЕССОВ

|  |  |  |
| --- | --- | --- |
|  | Студент: | Старостенков А.А. |
|  | Группа: | ВМ-22 (маг.) |
|  | Вариант: | 15 |
|  | Преподаватель: | Федулов Я.А. |

СМОЛЕНСК

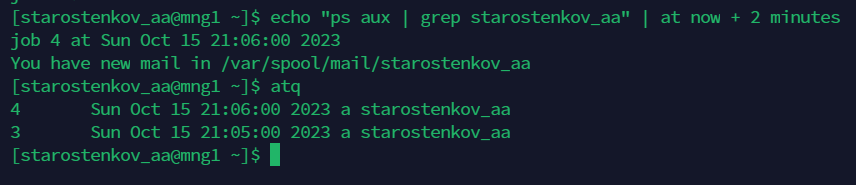
2023

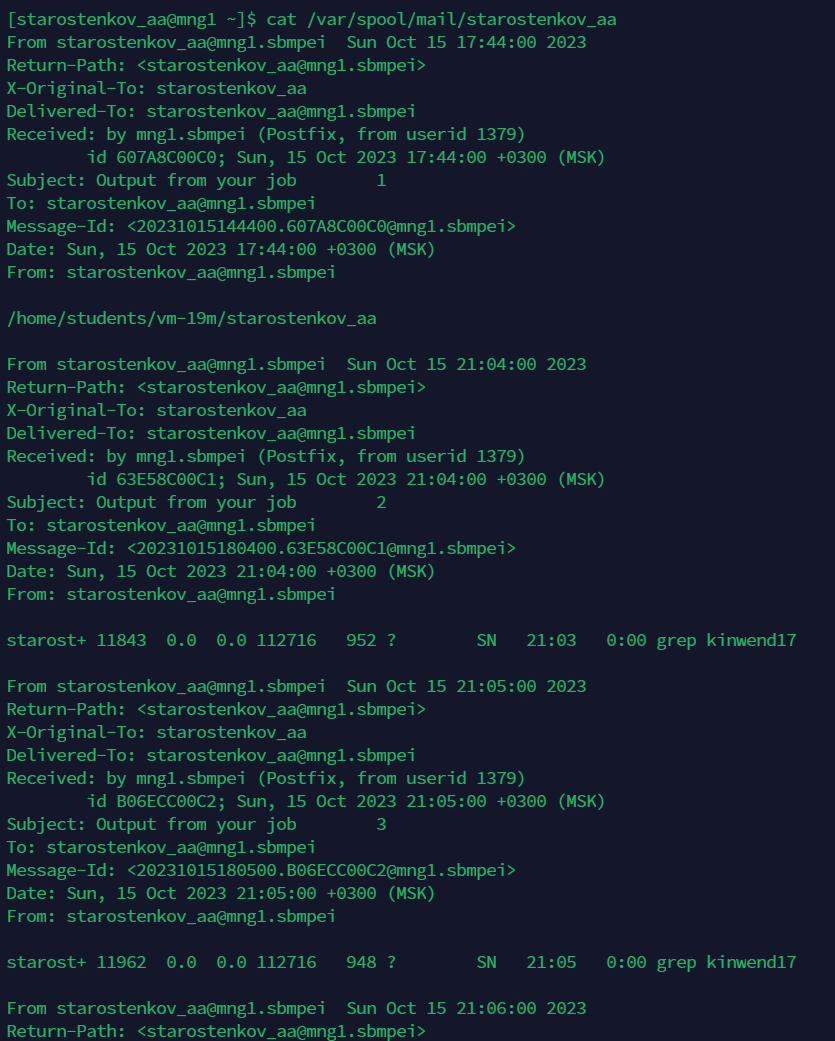
1. Определите имя файла текущего терминала командой tty (при работе в текстовом режиме имя файла основного терминала /dev/tty1).

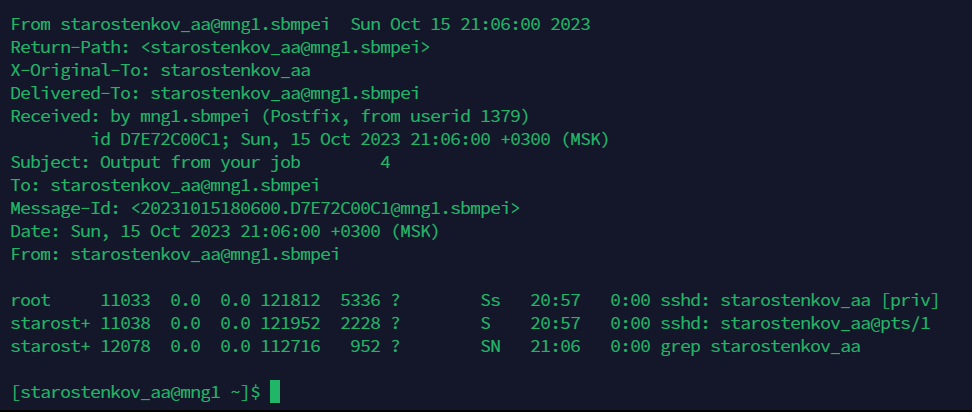


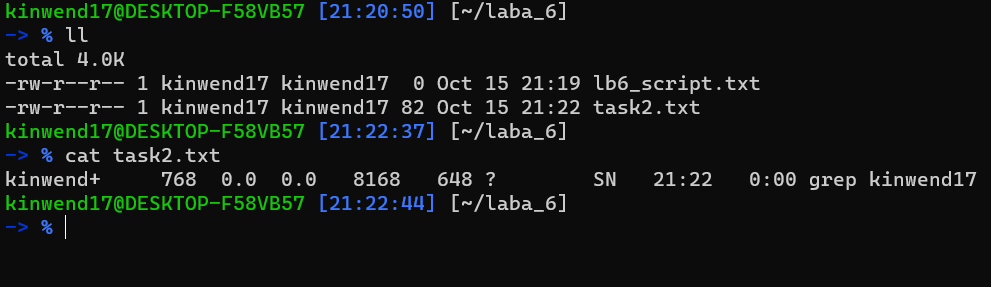
1. Создайте отложенное на 2 минуты задание, выводящее подробный список всех процессов пользователя student (в моём случае kinwend17) на экран.
2. Дождитесь вывода списка процессов на экран.

Сразу стоит оговориться: отложенная команда выполняется, но в терминале локальной машины не выводится сообщение о выполнении. Команда «at» использует почтовый сервер для вывода уведомления о выполнении. Для примера, сделаем выполнение команды на вычислительном кластере СФ МЭИ. Наше письмо (уведомление) будет храниться в файле /var/spool/mail/starostenkov\_aa. Для локальной машины будет дополнять команды перенаправлением вывода в файл, чтобы смотреть результаты работы.

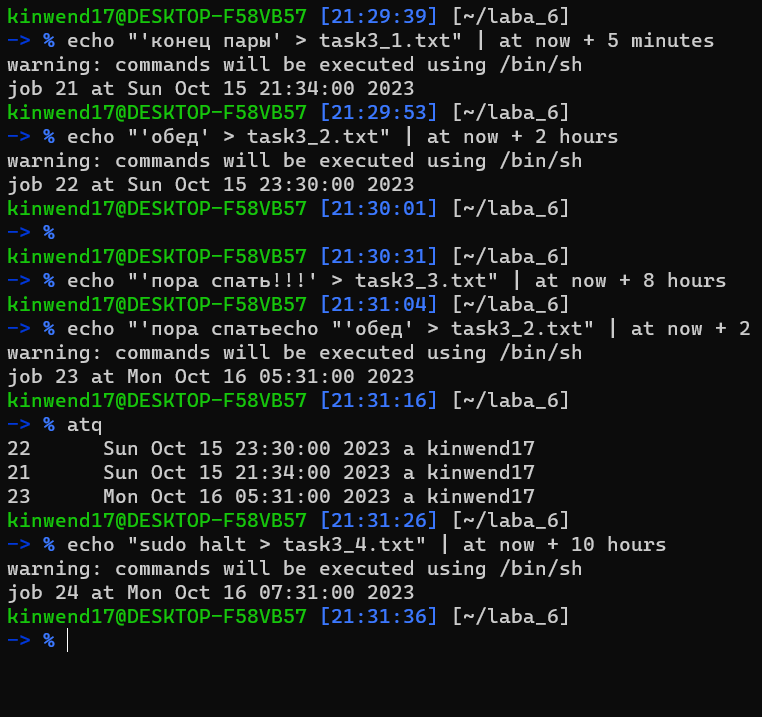




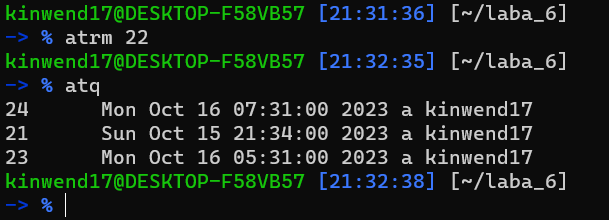




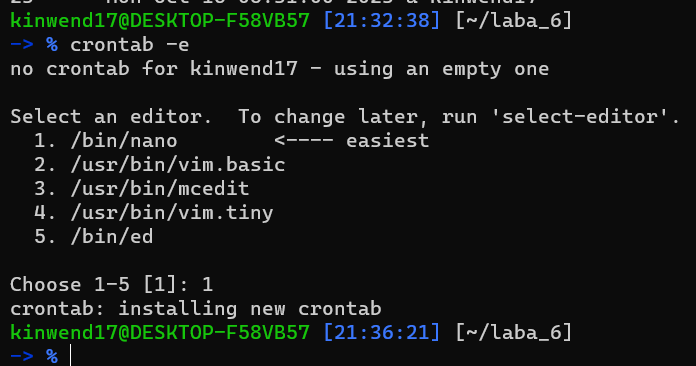
1. Установите отложенные задания, выводящие сообщения: «конец пары», «обед», «пора спать!!!» (с одновременным выключением компьютера командой halt — только для суперпользователя root).

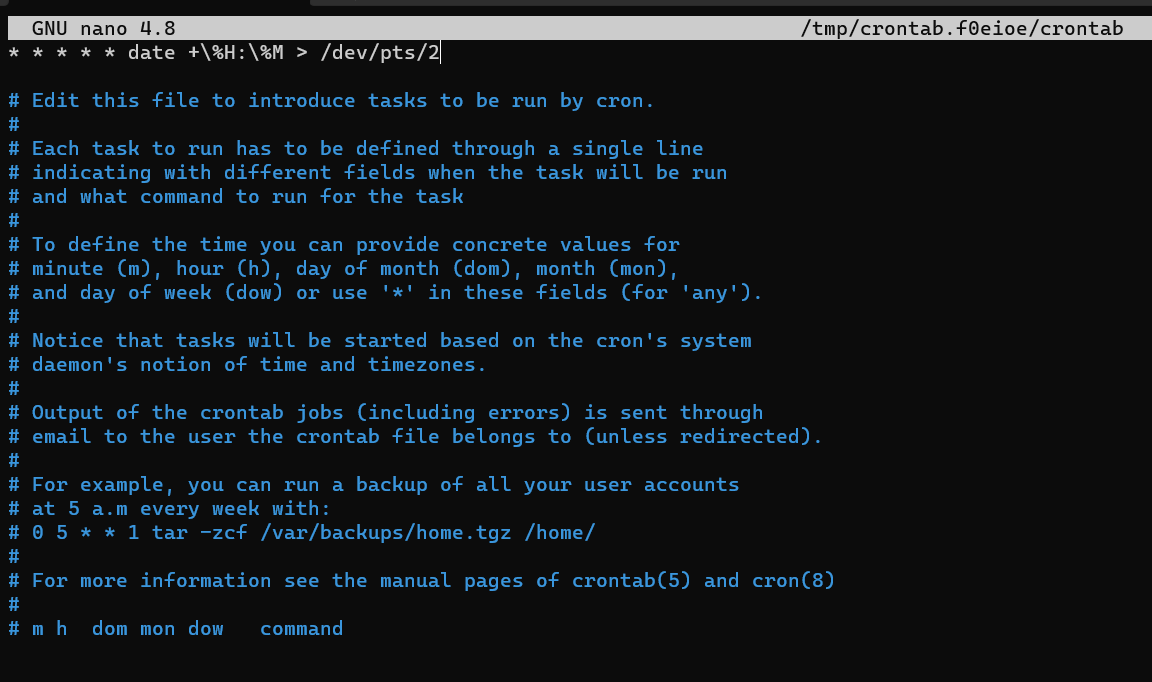


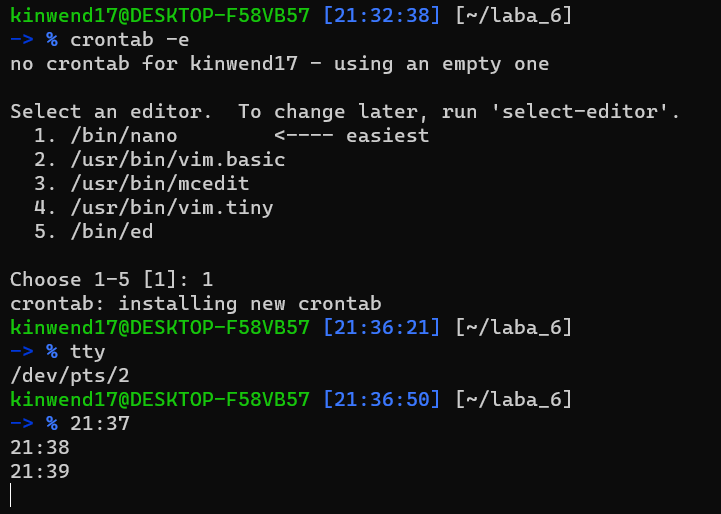
1. Удалите задание, выводящее на экран сообщение «обед».



1. Создайте задание, выводящее на экран текущее время в формате ЧЧ:ММ каждую минуту.





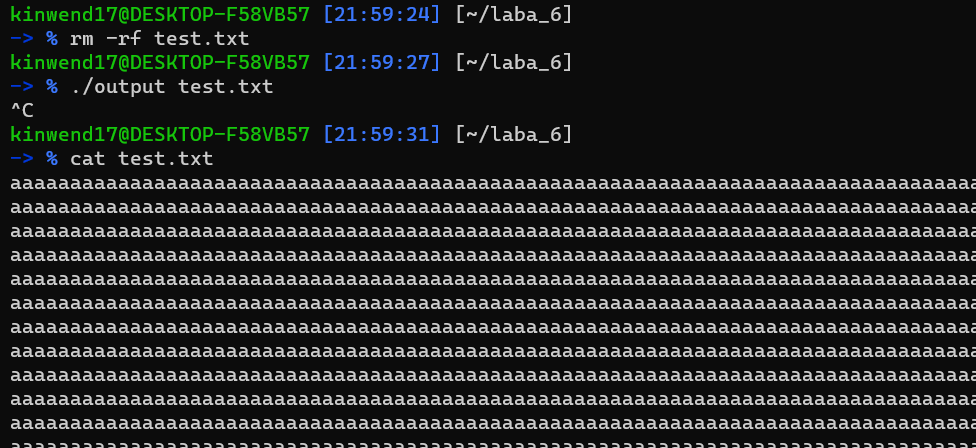


1. Напишите и скомпилируйте программу output, которая в бесконечном цикле выводит символ «a» в файл, имя которого указано в первом аргументе командной строки.

Программу напишем на языке С, код будем писать с помощью VS Code, скомпилируем с помощью gcc.

Gcc outpuc.c -o output





1. Запустите одновременно два фоновых процесса и через некоторое время принудительно завершите их.

Запуск:

./output a.txt & ./output b.txt &

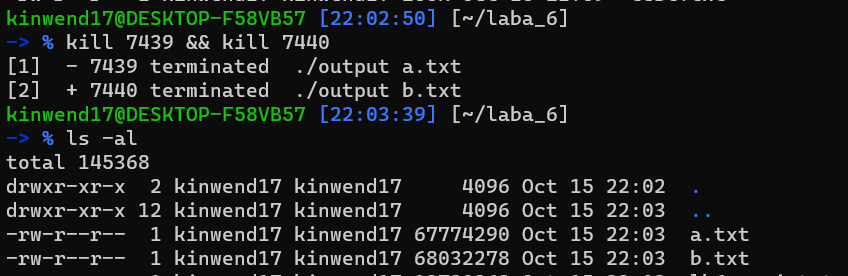
Завершение:

kill PID1 & kill PID2 &

где PID1 и PID2 — идентификаторы двух запущенных процессов.



1. Сравните размер файлов a.txt и b.txt.

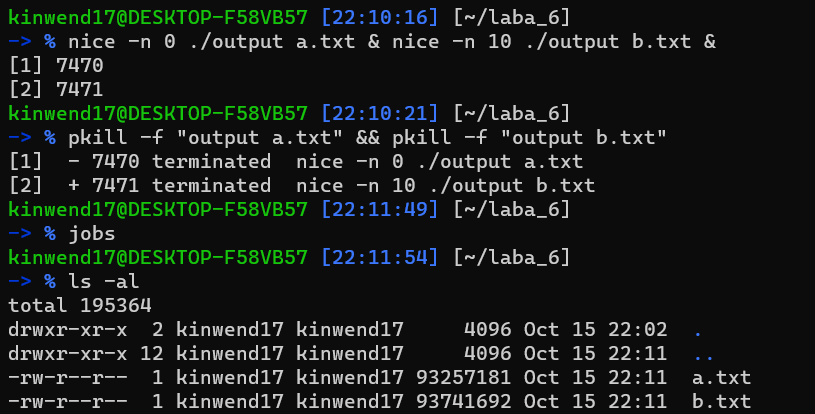


Файл a.txt имеет размер 67 774 290 байт, а файл b.txt имеет размер 68 032 278 байт. Файл b.txt больше по размеру на 257 988 байт.

Размеры файлов a.txt и b.txt отличаются, потому что мы выполнили два фоновых процесса, каждый из которых запускал программу output, которая бесконечно добавляла символ «a» в соответствующий файл. В процессе выполнения, один из процессов мог завершиться быстрее другого, что привело к разнице в размере файлов. Это нормальное поведение, так как бесконечные процессы завершаются непредсказуемо, и файлы могут закрыться с разными данными.

Итак, файл b.txt больше по размеру, потому что один из фоновых процессов, запущенных для output b.txt, успел записать больше символов «a» в файл, чем процесс для output a.txt.

1. Запустите одновременно два фоновых процесса с разными приоритетами (0 и +10 соответственно) и через некоторое время принудительно завершите их.



Размер файлов a.txt и b.txt, как видно из ваших данных, следующий:

* a.txt: 93,257,181 байт.
* b.txt: 93,741,692 байта.

Размер b.txt больше, чем a.txt.

Скорее всего, процесс с более высоким приоритетом (nice -n 10) успел записать больше символов «a» в файл b.txt, чем процесс с нулевым приоритетом (nice -n 0) в файл a.txt. Это объясняет разницу в размерах файлов a.txt и b.txt.

**КОНТРОЛЬНЫЕ ВОПРОСЫ:**

1. Что осуществляет планировщик cron?

Планировщик cron в Linux осуществляет периодическое выполнение задач (команд, скриптов) в определенное время или с определенной периодичностью. Это позволяет автоматизировать выполнение регулярных заданий, таких как резервное копирование, обновление данных и многие другие операции.

1. Какие основные команды для работы с планировщиком заданий в Linux

Основные команды для работы с планировщиком заданий в Linux:

* crontab: Команда для управления cron-заданиями пользователя. Можно создавать, редактировать и удалять задания.
* cron: Демон-планировщик, который управляет выполнением cron-заданий.
* at: Команда для выполнения отложенных заданий один раз в указанный момент времени.
* nice: Команда для изменения приоритета выполнения процессов.

1. Что происходит с выводимой отложенным процессом информацией?

Выводимая информация отложенным процессом в случае использования at отправляется по электронной почте на почтовый ящик пользователя.

1. Для чего нужна команда at? Какой у ней формат?

Команда at в Linux используется для планирования выполнения заданий в определенный момент времени или через определенное время. Её формат обычно такой: **at [опции] время**. Опции позволяют уточнить, когда и как выполнять задание. Затем вводится команда, которую нужно выполнить.

1. С помощью каких команд осуществляется управление приоритетами процессов в Linux?

Для управления приоритетами процессов в Linux используются следующие команды:

* nice: устанавливает приоритет выполнения процесса.
* renice: изменяет приоритет существующего процесса.

1. Каково влияние приоритета на производительность процесса?

Приоритет процесса влияет на то, насколько «жадно» процесс использует процессорное время. Процессам с более высоким приоритетом предоставляется больше процессорного времени, и они выполняются быстрее, в то время как процессы с низким приоритетом могут выполняться медленнее. Установка правильного приоритета позволяет более эффективно управлять ресурсами процессора в системе.