## Управление процессами в Linux

### Цель работы

Получить представление о процессах, как о способе управления ресурсами в Линукс. Научиться получать и анализировать информацию о процессах и управлять состоянием выполняющихся процессов.

### Задания к работе

1. Войти в систему с собственной учетной записью

Изображение выглядит как текст

Автоматически созданное описание

1. Получить справку о команде ps

Изображение выглядит как текст

Автоматически созданное описание

1. Командой ps вывести краткую информацию о выполняющихся процессах в текущем терминале и определить PID текущей оболочки

Изображение выглядит как текст, устройство, счетчик

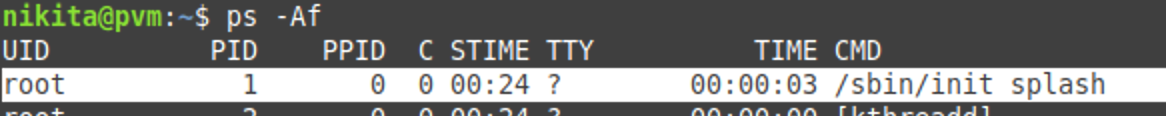
Автоматически созданное описание

1. Получить подробную информацию о загруженных процессах и выяснить, какой из них использует максимальный объем памяти, а какой - максимально загружает процессор

Изображение выглядит как текст

Автоматически созданное описание

1. Из таблицы, полученной в п.4 выяснить, какой PID имеет процесс init и от чьего имени он запущен



1. Открыть новый сеанс с собственной учетной записью в и запустить в нем текстовый редактор nano

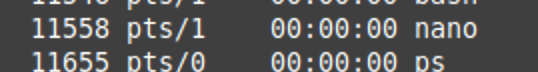
Изображение выглядит как текст

Автоматически созданное описание

1. Вернуться в исходный сеанс и снова просмотреть список процессов. Определить PID MC, запущенного от вашего имени

Изображение выглядит как текст

Автоматически созданное описание



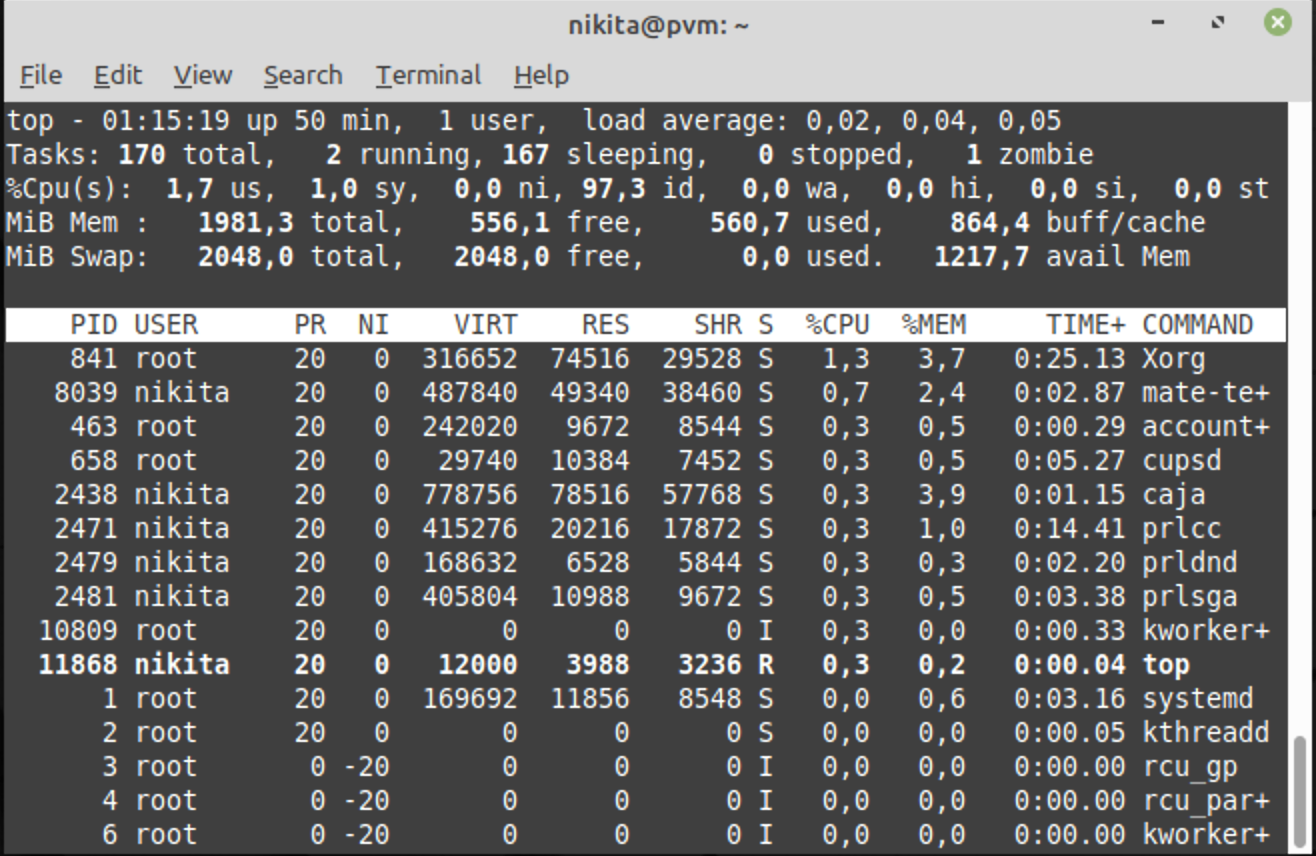
1. Командой kill снять все процессы nano



Изображение выглядит как текст, черный

Автоматически созданное описание

1. В tty1 выполнить команду top. Сравнить ее возможности с возможностями ps



1. Используя top или ps определить, какие процессы порождены (поле PPID) процессом init (PID=1)

Изображение выглядит как текст

Автоматически созданное описание

1. Запустить поиск всех файлов .html от каталога /. Приостановить этот процесс (Ctrl+Z).

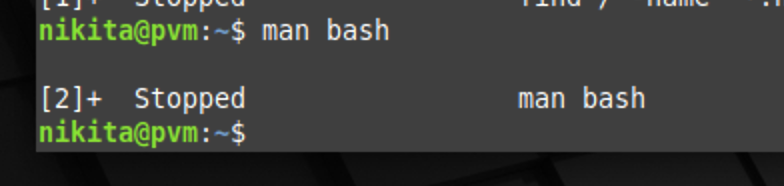
Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

1. Запустить команду man bash и приостановить ее выполнение



1. Командой jobs определить номера задач, запущенных в предыдущих пунктах.

Изображение выглядит как текст

Автоматически созданное описание

1. Командой fg продолжить выполнение man bash.

Изображение выглядит как текст

Автоматически созданное описание

1. Принудительно (kill) завершить команду find.

Изображение выглядит как текст

Автоматически созданное описание

### Контрольные вопросы

1. В системе зарегистрированы и работают пользователи user1 и user2. Может ли user1 завершить работу процесса, запущенного пользователем user2?
2. Может ли user1 понизить приоритет процесса, запущенного user2?
3. Может ли user1 повысить приоритет собственного процесса?

На все вопросы ответ один – если user имеет доступ к sudo, то да. Но сам по себе, конечно не может.

### Дополнительные задания

1. Установите программу htop и познакомьтесь с ней. Используя ее в процессе повседневной работы ответьте на несколько вопросов:

(htop стоял на основной ОС, поэтому использую его)

* 1. Какие программы потребляют больше всего оперативной памяти?

Изображение выглядит как текст, внутренний

Автоматически созданное описание

Больше всего – запущенная виртуальная система

* 1. Насколько часто центральный процессор бывает загружен полностью и при выполнении каких задач?

Изображение выглядит как текст

Автоматически созданное описание

Больше и чаще всего – iTerm’ом, так как на данный момент это активная задача

* 1. Сколько памяти потребляет система в режиме простоя?

В районе 2-3 гб

* 1. Требуется ли системе апгрейд и каких компонентов в первую очередь?

Для комфортной работы с виртуалками, лучше установить дополнительную оперативную память (в сумме на 16-32 гб).

Процессор сам по себе двухъядерный, но благодаря технологии виртуализации ядра – считается как четырехядерный. Со своими задачами справляется, хотя если нацеливать систему на работу с множеством виртуальных ОС, то кончено его тоже можно поменять.