



11010010  
|||||  
00010011



POCATOM

# Права доступа. Процессы

Карапетьянц Николай

000100111101000  
110101011001000  
  
0001001111010001101010110  
010000001001111

2022



# СОДЕРЖАНИЕ

2022

- Расширенные атрибуты файлов
- ACL
- Процессы



# РАСШИРЕННЫЕ АТТРИБУТЫ



**lsattr** – выводит атрибуты расширенной файловой системы

- **-R** рекурсивный вывод атрибутов каталогов и их содержимого
- **-a** вывод всех файлов в каталоге, включая скрытые
- **-d** перечислить файлы и каталоги без их содержимого
- **-v** вывод номера версии/поколения файла

**Поддерживают:**

- Ext2
- Ext3
- Ext4
- XFS
- BTRFS
- и другие

# РАСШИРЕННЫЕ АТТРИБУТЫ



**chattr** – меняет атрибуты  
расширенной файловой системы

**-R** рекурсивная обработка каталога

**-V** подробный вывод

**-f** игнорирование сообщений об  
ошибках

## Атрибуты:

- **a** : файл может быть открыт только в режиме добавления
- **A** : не добавлять время перезаписи
- **c** : автоматически сжимать при записи на диск
- **C** : отключить копирование при записи
- **D** : синхронно записывать на диск все изменения в каталоге на диск
- **i** : сделать неизменяемым;
- **S** : сразу записывать изменения на диск
- **u** : при удалении файла содержимое может быть восстановлено

# СПИСОК КОНТРОЛЯ ДОСТУПА(ACL)



## Предоставляет дополнительный механизм разрешений для файловых систем

Поддерживают:

- EXT2
- EXT3
- EXT4
- XFS
- BTRFS
- и другие

По умолчанию включен не во всех дистрибутивах

Формат:

- user:admin:rwX
- group:kvm:r-x
- other::r—
- default:user::rw-
- default:group::rw-
- mask:r-x

# ПРАВА ДОСТУПА(ACL)



## Приоритет:

- Права владельца
- Правило типа user
- Правило типа group
  - Наличие маски
  - Отсутствие маски
- Правило типа other

Маска ACL – *механизм принудительного ограничения действий всех правил*

Правила по умолчанию (атрибут **-d**):

- Наследуются файлами в каталоге, но не на сам каталог
- Не применяются к уже существующим

# ПРОЦЕССЫ



**Процесс** – совокупность программного кода и данных, загруженных в память ЭВМ.

Состояния процесса:

- работающий процесс – выполнение кода
- спящий процесс – ожидание события
- Остановленный процесс – остановленные процессы с помощью сигналов
- процесс-зомби – процесс завершил работу, но запись все еще находится в таблице процессов

PID – идентификатор процесса (от 0 до 65535)

PPID – идентификатор родительского процесса

Демоны – серверные службы

# ИДЕНТИФИКАТОРЫ



**Real UID** – идентификатор пользователя запустившего процесс

**Effective UID** – эффективный идентификатор пользователя от чьего имени запустили процесс

**Setuid (u+s)** – бит разрешения позволяет запускать исполняемый файл от имени владельца

**Setgid (g+s)** – бит разрешения позволяет запускать исполняемы от имени группы, которая владеет файлом

**Sicky Bit (+t)** – бит разрешения позволяет удалять в директории файлы только их владельцем



# PROCFS

## /proc

**ProcFS** – специальная файловая система, которая позволяет получить доступ к информации из ядра о системных процессах

**echo \$\$** - вывод идентификатора текущего процесса

**/proc/PID/cmdline** – аргументы командной строки

**/proc/PID/environ** – переменные окружения для данного процесса

**/proc/PID/status** – статус процесса

**/proc/PID/fd** – директория с открытыми файловыми дескрипторами

**/proc/cpuinfo** – информация о процессоре

**/proc/meminfo** – информация о памяти

**/proc/version** – содержит информацию о версии ядра,

# КОМАНДЫ



**ps** – отображает информацию о текущих активных процессах:

- e :все процессы
- a :все, кроме фоновых процессов
- d :все, кроме процессов сессий
- u : процессы пользователя
- x :все процессы без управляющего терминала
- f :древовидное представление
- o :вывод в пользовательском формате

**pstree** – вывод дерева процессов

**top** – вывод информации о процессах в реальном времени

**pgrep** – поиск процесса

**w** – выводит информацию о активных пользователей в системе и его процессах

**uptime** – выводит информацию о времени работы операционной системы

**kill** – остановка процесса

# ФОНОВЫЕ ПРОЦЕССЫ



**Задания** – фоновые процессы запущенные пользователем.

**Сессия** – коллекция групп заданий

**Управляющий терминал** – устройство для взаимодействия с процессами в рамках сессии

## Команды

**jobs** – отображение статуса заданий в текущем сеансе

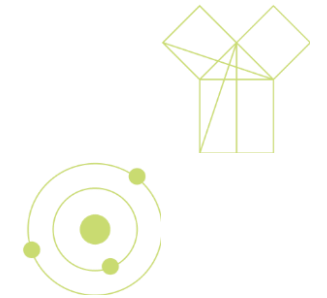
**fg** – запустить задание в основном режиме

**bg** – запустить задание в фоновом режиме

**command &** – запустить команду в фоновом режиме

**command > /dev/null 2>&1 &** – запустить команду в фоновом режиме с перенаправлением потока вывода и ошибок в null устройство

# УПРАВЛЯЮЩИЕ СИГНАЛЫ



**SIGINT(2)** – завершение процесса (Ctrl+C)

**SIGQUIT(3)** – завершение процесса с созданием дампа памяти (Ctrl+/,)

**SIGTERM (15)** – безопасно и немедленно завершает процесс

**SIGKILL(9)** – немедленно и жестко завершает процесс

**Ctrl+C** – немедленное прекращение **работы команды**

**Ctrl+Z** – перевод выполняемой команды в фоновый режим

**Ctrl+D** – завершение оболочки

**Ctrl+S** – остановить вывод на экран

**Ctrl+Q** – возобновить вывод на экран

# ССЫЛКИ



- ACL Ubuntu: [https://help.ubuntu.ru/wiki/access\\_control\\_list#access\\_control\\_list - списки контроля доступа](https://help.ubuntu.ru/wiki/access_control_list#access_control_list_-_списки_контроля_доступа)
- Команда ps: <https://andreyex.ru/operacionnaya-sistema-linux/komanda-ps-v-linux/>
- Управление процессами <https://losst.pro/upravlenie-protsessami-v-linux>
- Процессы: <https://www.opennet.ru/docs/RUS/linuxadmin/lesson06.html>
- Как управлять процессами: <https://andreyex.ru/operacionnaya-sistema-linux/kak-upravlyat-processami-v-linux/>
- Задания: <https://andreyex.ru/linux/kak-ispolzovat-komandu-jobs-v-linux/>
- Запуск процесса в фоне: <https://losst.pro/kak-zapustit-protsess-v-fone-linux>