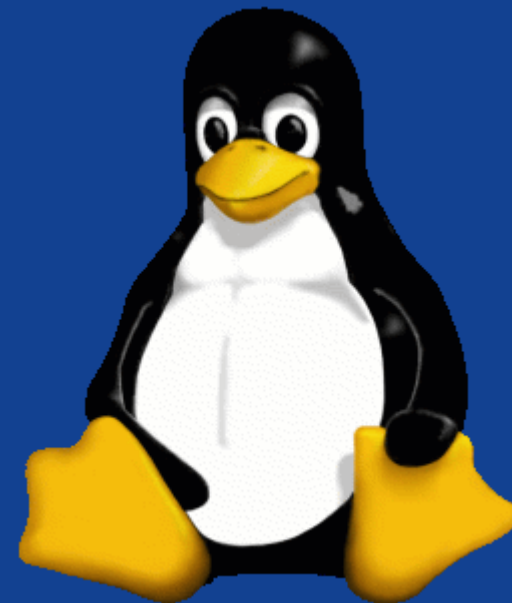


# Linux II

19-10-2020

Kornel Tercjak

Mateusz Zbytnewski



# Agenda

1. Boot process – proces uruchamiania
2. Zarządzanie użytkownikami
3. Uprawnienia
4. Dowiązania do plików (inode number)
5. Strumienie
6. Sygnały
7. SSH i SCP
8. Aliasy

## Linux – proces uruchomienia

Stare:

1. BIOS
2. Boot Loader
3. Kernel
4. Init
5. Runlevel scripts

Nowe:

1. BIOS / UEFI
2. Boot Loader
3. Kernel
4. systemd

### BIOS:

1. Test POST
2. Inicjalizacja sprzętu
3. Uruchomienie kodu z MBR (stage 1)
4. Uruchomienie bootloadera (stage 2)
5. Załadowanie systemu operacyjnego

### UEFI:

1. Test POST
2. Inicjalizacja sprzętu
3. Wczytanie listy aplikacji EFI
4. Uruchomienie aplikacji EFI (bootloader, kernel)
5. (gdy aplikacja EFI to bootloader) Załadowanie systemu operacyjnego

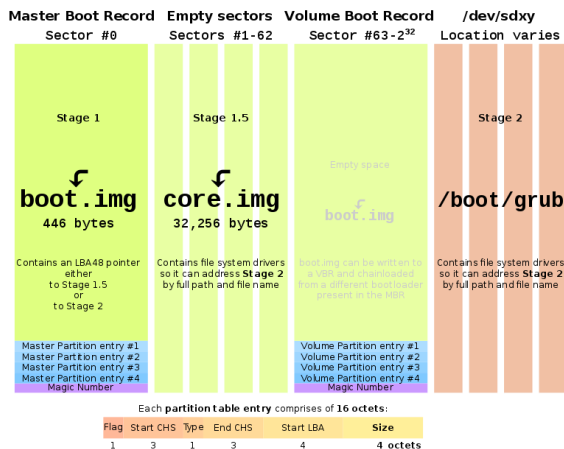
## MBR (Master Boot Record)

- Pierwszy sektor dysku twardego o rozmiarze 512 bajtów.
- Pierwsze 446 bajty to boot loader, następne 64 bajty są zarezerwowane na tablicę partycji, a ostatnie 2 bajty to sygnatura rozruchu (0x55 0xAA).

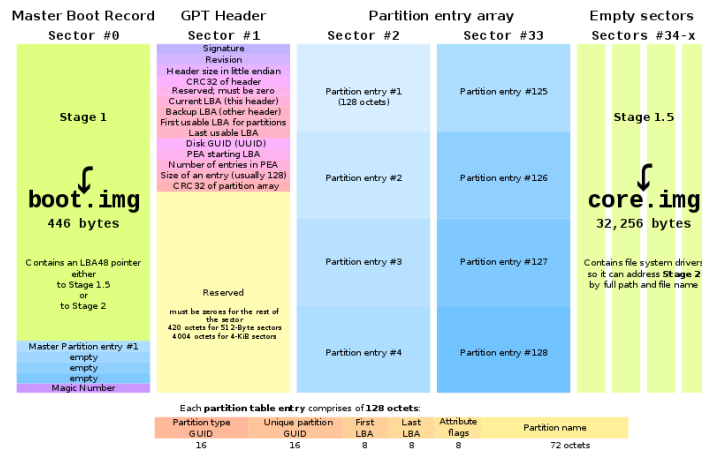
MBR nie może załadować bezpośrednio kernela, ponieważ nie wie jaki system plików jest używany dlatego też wymagany jest bootloader właściwy dla systemu plików.

# GRUB (GRand Unified Boot loader)

## MBR + GRUB



## GPT + GRUB



Konfiguracja:  
- /boot/grub2/grub.cfg

## Kernel - serce systemu

1. Bootloader / UEFI uruchamia obraz kernela i initramfs
2. Ładuje podstawowe sterowniki z initramfs
3. Startuje proces init

## Inicjalizacja systemu

### Initproces

Uruchamia skrypty w zależności od poziomu pracy (runlevel):

0 – zatrzymanie systemu

1 – single-user mode

2, 3 – multi-user

4, 5 – w zależności od distro

6 - reboot

### Systemd:

Uruchamia usługi (services) w zależności od celu (target):

poweroff.target

rescue.target

multi-user.target

graphical.target

reboot.target



## Administracja - cechy

- login,
- UID,
- GID,
- katalog domowy,
- powłoka.

## /etc/passwd

```
Mon 21 Oct 15:41:48 gardon@gardon-pc linux-2 $ cat /etc/passwd | grep gardon
gardon:x:1000:1000:gardon:/home/gardon:/usr/bin/zsh
```

- nazwa użytkownika,
- x oznacza że hasło jest szyfrowane i jest przechowywane w pliku */etc/shadow*,
- user ID,
- group ID,
- nazwa grupy użytkownika,
- katalog domowy użytkownika,
- powłoka startująca użytkownika.

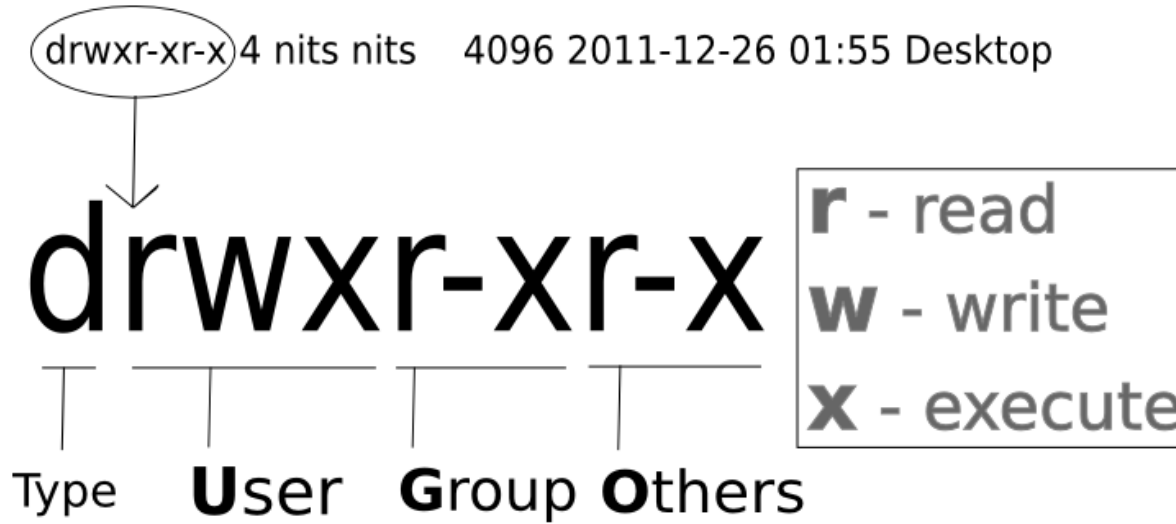
## Administracja użytkownikami - zarządzanie

- useradd - dodawanie użytkowników,
- passwd - zmiana hasła,
- usermod - modyfikowanie użytkowników,
- /etc/passwd - plik z danymi użytkowników,
- /etc/shadow - plik z hasłami użytkowników

## Administracja grupami - zarządzanie

- groupadd - dodawanie grup,
- gpasswd - administracja grupami,
- /etc/group - plik z danymi grup,

## Uprawienie podstawowe



*File permissions in Linux*

## Uprawienie podstawowe - przykład

- - - plik regularny
- d - katalog
- l – dowiązanie symboliczne
- c – urządzenie strumieniowe
- b – urządzenie blokowe
- s – gniazdo sieciowe

```
[ccsscm@wrlinb146 bts-sc-ccs]$ ll /dev
total 0
crw-rw---- 1 root root 10, 235 Sep 3 01:22 autofs
drwxr-xr-x 2 root root 820 Sep 3 01:20 block
drwxr-xr-x 2 root root 120 Sep 3 01:20 bsg
drwxr-xr-x 3 root root 60 Sep 3 01:20 bus
crw-r--r-- 1 root root 255, 171 Sep 3 01:22 casr
crw-r--r-- 1 root root 255, 173 Sep 3 01:22 ccsm
lrwxrwxrwx 1 root root 3 Sep 3 01:20 cdrom1 -> sr0
lrwxrwxrwx 1 root root 3 Sep 3 01:20 cdrwl -> sr0
crw-r--r-- 1 root root 255, 178 Sep 3 01:22 cdt
crw-r--r-- 1 root root 255, 172 Sep 3 01:22 cecc
crw-r--r-- 1 root root 255, 176 Sep 3 01:22 cevt
drwxr-xr-x 2 root root 5080 Sep 3 01:22 char
crw----- 1 root root 5, 1 Sep 3 01:20 console
lrwxrwxrwx 1 root root 11 Sep 3 01:20 core -> /proc/kcore
drw-r--r-- 2 root root 200 Sep 3 01:22 cpqhealth
drwxr-xr-x 50 root root 1020 Sep 3 01:20 cpu
crw-rw---- 1 root root 10, 62 Sep 3 01:20 cpu_dma_latency
crw-r--r-- 1 root root 255, 180 Sep 3 01:22 crom
drwxr-xr-x 5 root root 100 Sep 3 01:20 disk
brw-rw---- 1 root disk 253, 0 Sep 3 01:20 dm-0
```

## Dowiązania (soft/hard link)

Jest to specjalny typ plików, który wskazuje na dowolny plik lub katalog

- symboliczne (soft link):

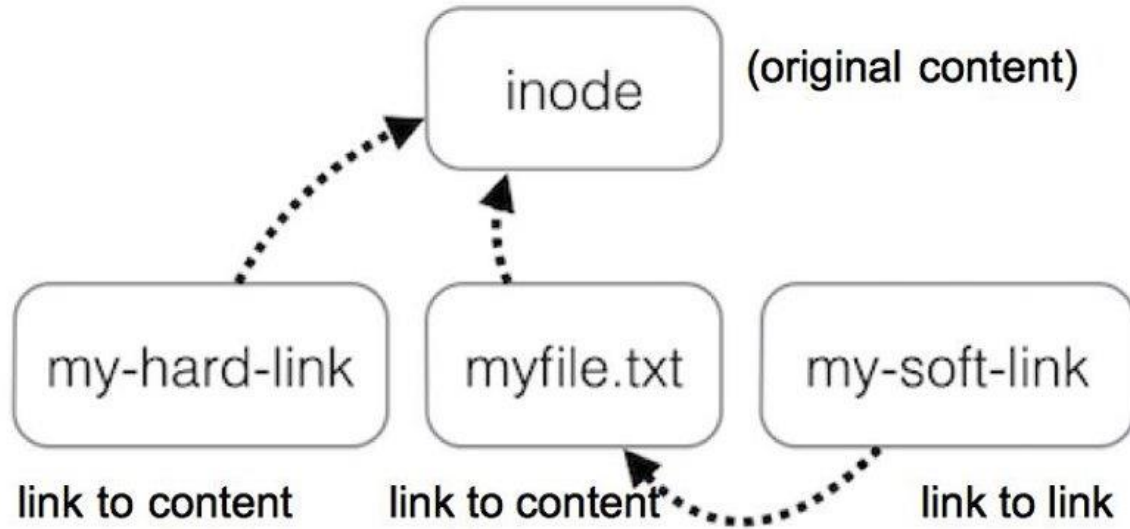
```
$ ln -s file.txt soft-file.txt
```

- twarde (hard link):

```
$ ln file.txt hard-file.txt
```

```
Mon 21 Oct 14:50:38 gardon@gardon-pc linux-2 $ ls -li
total 0
5381321 -rw-r--r-- 2 gardon gardon 0 10-21 14:50 file.txt
5381321 -rw-r--r-- 2 gardon gardon 0 10-21 14:50 hard-link.txt
5381323 lrwxrwxrwx 1 gardon gardon 8 10-21 14:50 soft-link.txt -> file.txt
```

## Dowiązania (soft/hard link)





# Inode Number

Jest to numer indeksu reprezentujący zbiór informacji (metadata) na temat pliku lub katalogu:

- rodzaj pliku: zwykły plik, katalog,
- uprawnienia: rwx,
- liczba twardych dowiązań,
- ID użytkownika (właściciela pliku),
- ID grupy,
- rozmiar pliku,
- czas dostępu, modyfikacji,
- dostępy specjalne dla użytkowników i grup,
- i inne...

## Strumienie

- > przekieruj standardowe wyjście,
- >> przekieruj i dopisz do standardowego wyjścia,
- 2> przekieruj strumień błędów,
- 2>> przekieruj i dopisz strumień błędów,
- &> przekieruj oba wyjściowe strumienie,
- < standardowy strumień wejścia,
- | (pipe) – przekaż standardowy strumień wyjścia do kolejnego polecenia.

## Sygnały (SIGNALS)

- Są podstawowym sposobem komunikacji w systemie Linux.
- Jest to mechanizm asynchroniczny (niewymagający potwierdzenia) i uproszczony.
- Na podstawie samego numeru sygnału program podejmuje określone działania.

## Sygnały (SIGNALS) - rodzaje

- SIGINT (2) - przerwanie procesu,
- SIGKILL (9) - zakończenie procesu z wywłaszczeniem,
- SIGTERM (15) - programowe zakończenie procesu,
- SIGCONT (18) - wznowienie procesu,
- SIGSTOP (19) - wstrzymanie procesu,

**\$ kill -l # wyświetla listę dostępnych sygnałów**

## Sygnały (SIGNALS) - wywołanie

- kill,
- pkill,
- ctrl + c - SIGINT,
- ctrl + z - SIGTSTP,
- ctrl +\ - SIGQUIT.

## Sygnały (SIGNALS) - przechwytywanie

Czy sygnały można przechwycić ?

## Sygnały (SIGNALS)

- Sygnały są kolejgowane.
- Nie ma możliwości pominięcia nowego sygnału, ale nie ma też możliwości zliczenia kolejnych sygnałów tego samego typu.
- Nie można również odróżnić kolejności w jakiej różne sygnały się pojawiły.

Służy do szyfrowanego łączenia się zdalnie z innymi serwerami.

Standardowe użycie ssh:

```
$ ssh <użytkownik>@<nazwa_serwera>
```

Z podaniem ścieżki do klucza i portu :

```
$ ssh -i <ścieżka_do_klucza> <użytkownik>@<nazwa_serwera> -  
p <port>
```



## SCP - Secure copy

Wykorzystuje protokół SSH do przesyłania plików pomiędzy zdalnymi maszynami.

Kopiowanie pliku ze zdalnego serwera do lokalnego

```
$ scp <uzytkownik>@<nazwa_serwera>:foobar.txt  
</ścieżka/do/katalogu/docelowego>
```

Kopiowanie pliku z lokalnego serwera do zdalnego

```
$ scp foobar.txt <uzytkownik>@<nazwa_serwera>:  
</ścieżka/do/katalogu/docelowego>
```

# Alias

## Utworzenie alias'u

```
$ alias ll='ls -l'
```

## Wyświetlenie wszystkich dostępnych alias'ów:

```
$ alias
```

```
$ alias -p ll
```

**NOKIA**