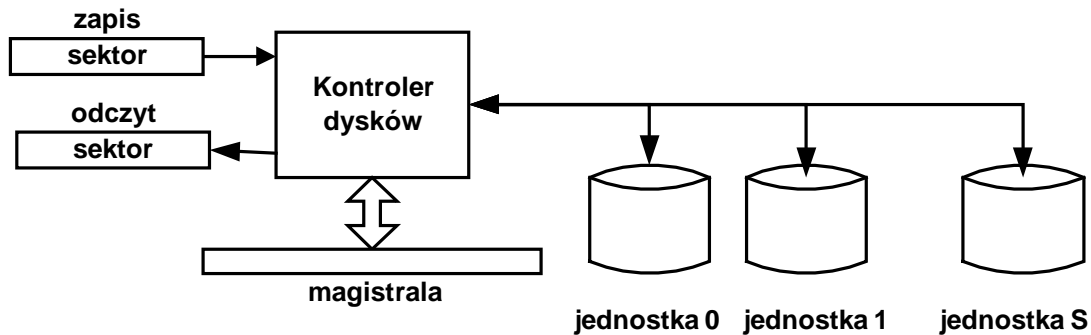


# 1. Dyski i systemy plików



Kontrolery dysków widziane są w katalogu /dev jako pliki specjalne zaczynające się od liter sd. Można je wylistować za pomocą polecenia:

```
$ls /dev/sd*
```

/dev/sda	Dysk a
/dev/sda1	Dysk a partycja 1
/dev/sda2	Dysk a partycja 1
/dev/sda3	Dysk a partycja 1
/dev/sda4	Dysk a partycja 1
/dev/sdb	Dysk b
/dev/sdb1	Dysk b partycja 1
/dev/sr0	Napęd CDROM

Przykład 1-1 Urządzenia dyskowe widziane jako pliki specjalne

- Urządzenie /dev/sda reprezentuje dysk ATA jako całość
- Urządzenia /dev/sda1 - /dev/sda4 jego części nazywane partycjami.
- Urządzenie /dev/sdb reprezentuje dysk USB a /dev/sr0 napęd CDROM.

Urządzenia dyskowe widziane są przez system Linux jako urządzenia SCSI (ang. *Small Computer System Interface*). Jest to przyjęte dla uzyskania jednolitej obsługi różnorodnych urządzeń. Urządzenia SCSI widziane są jako /dev/sg0, /dev/sg1, ....

Można się o tym przekonać wykonując polecenie:

```
$ls /dev/s*
```

```
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3 /dev/sda4  
/dev/sdb /dev/sdb1 /dev/sg0  
/dev/sg1 /dev/sg2 /dev/sr0
```

Dalsze informacje o urządzeniach SCSI można uzyskać za pomocą polecenia `lsscsi` jak poniżej.

```
$lsscsi
[0:0:0:0]    cd/dvd  HL-DT-ST DVD-RAM GH22NP20  1.02 /dev/sr0
[3:0:0:0]    disk    ATA      ST3500413AS  JC4B /dev/sda
[4:0:0:0]    disk    SanDisk  Cruzer Edge  1.27 /dev/sdb
```

Przykład 1-2 Informacja o dyskach uzyskana za pomocą polecenia `lsscsi`

Liczby w nawiasach oznaczają:

- Numer kontrolera SCSI
- Numer magistrali SCSI
- Identyfikator urządzenia SCSI
- Logiczny numer jednostki LUN (ang. *Logical Unit Number*)

Ten sposób identyfikacji typowy jest dla urządzeń SCSI

Z powyższego przykładu widać że urządzenia dyskowe widziane są także jako urządzenia SCSI. Pokazują to wyniki polecenia:

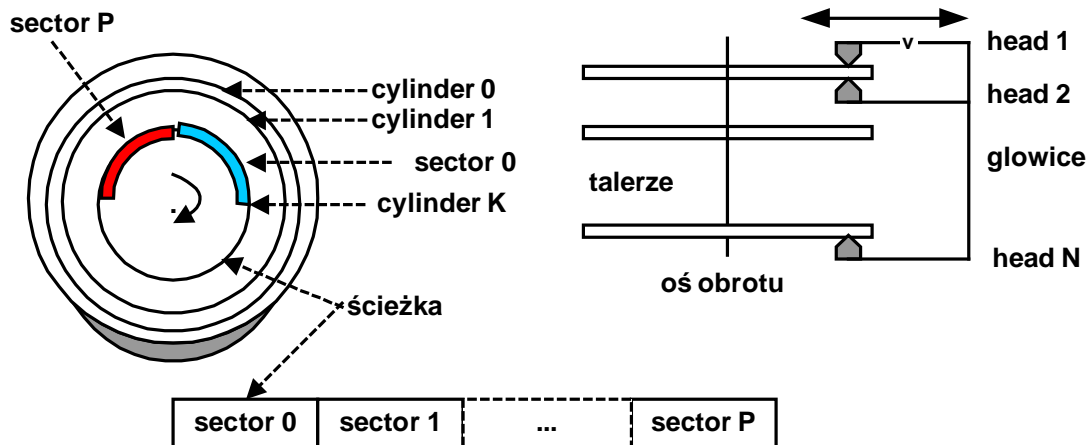
```
$lsscsi -g
[0:0:0:0] cd/dvd HL-DT-ST DVD-RAM      1.02 /dev/sr0
/dev/sg0
[3:0:0:0] disk    ATA      ST3500413AS  JC4B /dev/sda
/dev/sg1
[4:0:0:0] disk    SanDisk  Cruzer Edge  /dev/sdb
/dev/sg2
```

Przykład 1-3 Informacja o dyskach uzyskana za pomocą polecenia `lsscsi -g`

Jak widać dysk ATA widziany jest jako `/dev/sda` i `/dev/sg1`

## 1.1. Struktura dysku

Dysk składa się z obrotowych talerzy pokrytych materiałem magnetycznym. Nad talerzami przesuwa się głowice (ang. heads). Zapisy ułożone są na ścieżkach które składają się z sektorów (ang. sectors). Ścieżki które mogą być odczytane bez ruchu głowic tworzą cylinder.



Rys. 1-1 Struktura dysku

Najmniejszą porcją danych które daje się odczytać jest sektor (512 bajtów). Adres sektora składa się z trójki (C,H,S):

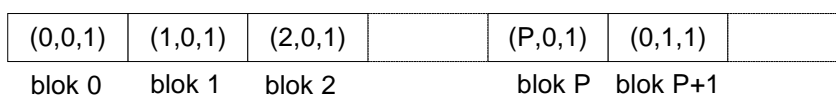
C – numer cylindra

H – numer głowicy

S - numer sektora na ścieżce

Jest to tak zwane adresowanie CHS ( ang. *Cylinder Head Sector* ).

Jeżeli przyjmie się jednolity sposób przeliczania trójwymiarowej adresacji sektora dysku na adres jednowymiarowy to dysk można uważać za jednowymiarową tablicę składającą się z sektorów do których można pisać i czytać. Przyjmuje się że sektor 0 na cylindrze 0 głowicy 1 ma adres 0. Adres 1 otrzymuje sektor 1 na cylindrze 0 głowicy 1 i tak dalej.



Rys. 1-2 Dysk widziany jako tablica bloków

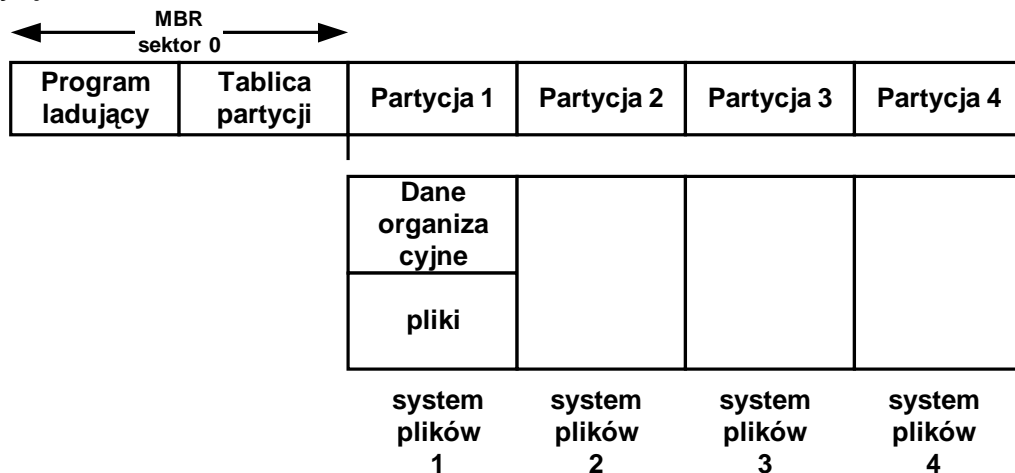
Po wyczerpaniu wszystkich sektorów ścieżki system zmienia głowicę nie zmieniając cylindra. Dopiero po wyczerpaniu wszystkich sektorów

fizycznych bieżącego cylindra, następuje jego przejście na cylinder o numerze wyższym o 1 i numeracja kontynuowana jest od pierwszego sektora fizycznego pierwszej głowicy tego cylindra, aż do wyczerpania wszystkich sektorów.

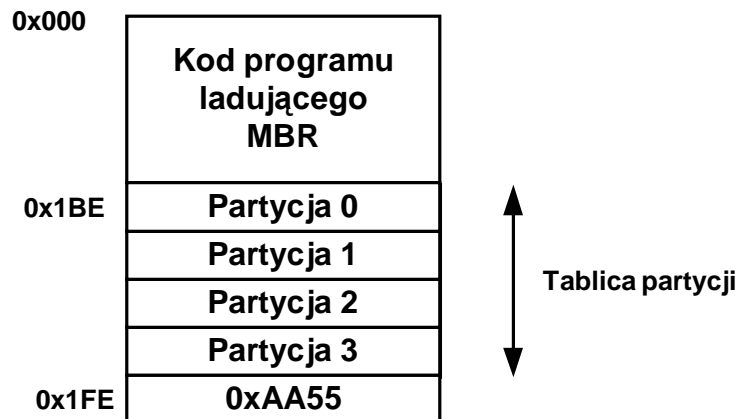
Adresowanie dysków za pomocą metody CHS nie sprawdza się gdyż nie uwzględnia ona faktu że wewnętrzne cylindry mają mniej sektorów niż zewnętrzne. Kontroler posługuje się metodą LBA (ang. Logical Block Addressing), jednak programy tłumaczą adresowanie LBA na CHS.

Posługiwanie się numerami bloków byłoby bardzo niewygodne. Stąd organizowane są one przez system operacyjny w system plików.

Dysk składa się z obszaru MBR (ang. *Master Boot Record*) zapisanego w sektorze 0 dysku oraz partycji. Położenie partycji definiuje tablica partycji zawarta w MBR.



Rys. 1-3 Podział dysku na partycje



Rys. 1-1 Struktura sektora ładującego MBR

Sektor ładujący zawiera krótki program ładujący i tablicę partycji opisującą podział dysku. Każda partycja opisana jest za pomocą następujących danych:

1	Flaga aktywności
2	Początek partycji
3	Typ partycji
4	Koniec partycji
5	Sektor początkowy partycji
6	Liczba sektorów partycji

Rys. 1-2 Zawartość tablicy partycji

Jedna partycja oznaczona jest jako aktywna. Zawiera ona program ładujący.

W Linuksie istnieje kilka programów do edycji podziału dysków:

- **parted**
- **gparted**
- **fdisk**

Przykład: program **fdisk**.

```
$sudo fdisk /dev/sda
```

```

juka@Ubol:~$ sudo fdisk /dev/sda
[sudo] password for juka:

Polecenie (m wyświetla pomoc): m
Polecenie
  a  zmiana flagi rozruchu
  b  modyfikacja etykiety dysku BSD
  c  zmiana flagi kompatybilności z DOS-em
  d  usunięcie partycji
  l  wypisanie znanych typów partycji
  m  wyświetlenie tego menu
  n  dodanie nowej partycji
  o  utworzenie nowej, pustej DOS-owej tablicy partycji
  p  wypisanie tablicy partycji
  q  zakończenie bez zapisywania zmian
  s  utworzenie nowej, pustej etykiety dysku Suna
  t  zmiana ID systemu partycji
  u  zmiana jednostek wyświetlania/wprowadzania
  v  weryfikacja tablicy partycji
  w  zapis tablicy partycji na dysk i zakończenie
  x  dodatkowe funkcje (tylko dla ekspertów)

```

Przykład 1-4 Opcje programu fdisk

Polecenie m wyświetla opcje programu. Za pomocą opcji p można wyświetlić tablicę partycji dysku /dev/sda

```

Disk /dev/sda: 500.1 GB, 500107862016 bytes
głowic: 255, sektorów/ścieżkę: 63, cylindrów: 60801, w sumie sektorów: 976773168
Jednostka = sektorów, czyli 1 * 512 = 512 bajtów
Rozmiar sektora (logiczny/fizyczny) w bajtach: 512 / 512
Rozmiar we/wy (minimalny/optimalny) w bajtach: 512 / 512
Identyfikator dysku: 0xbb44bb44

Urządzenie Rozruch   Początek      Koniec      Bloków      ID System
/dev/sda1   *              63          757753919  378876928+  7  HPFS/NTFS/exFAT
/dev/sda2              757753920    774124154    8185117+   b3  Nieznany
/dev/sda3              774125568    937709567    81792000   83  Linux
/dev/sda4              937709568    953335807    7813120    82  Linux swap / Solaris

```

Przykład 1-5 Tablica partycji dysku /dev/sda

System Linux rozróżnia trzy rodzaje partycji:

- Podstawową (ang. primary)
- Rozszerzoną (ang. extended)
- Logiczną (ang. logical)

Na dysku może być tylko jedna partycja typu rozszerzonego. Może ona zawierać dowolną liczbę partycji logicznych.

## 1.2. Linux – organizacja systemu plików

System plików zamienia urządzenie blokowe które czyta i pisze do tablicy sektorów partycji w hierarchiczną bazę danych składającą się z plików i katalogów.

Działa on w przestrzeni użytkownika – funkcja FUSE (ang. File System in User Space)

Obecnie mechanizm plików używany jest nie tylko do zapamiętywania informacji ale także do komunikacji z jądrem poprzez wirtualny system plików VFS (ang. Virtual File System). Przykładem może być katalog `/proc` czy `/sys`.

Najszerzej stosowane w Linuksie systemy plików to:

- **ext2** – przez długi czas był standardem i wzorowany był na UFS (ang. Unix File System)
- **ext3** – wprowadzone zostało kronikowanie co polepszyło niezawodność i przyspieszyło start systemu.
- **ext4** – rozszerzenie ext3 o duże pliki
- **ISO 9660** – standardowy system plików dla płyt CD-ROM
- **fat** - systemy plików używane przez Windows

System plików tworzony jest podczas partycjonowania. Wykonuje to np. polecenie:

```
$mkfs -t /ext4 /dev/sdf2
```

Tworzenie systemu plików polega na zapisie na partycji informacji organizacyjnych - superbloku. Wykonywane jest zwykle przy instalacji systemu.

### 1.2.1. Montowanie systemu plików

Dołączanie systemu plików do głównego systemu plików nazywa się montowaniem. Aby system plików zamontować potrzebne są następujące informacje:

- Nazwa urządzenia na którym system plików się znajduje
- Typ systemu plików
- Punkt montowania (miejsce w systemie plików gdzie montowany system plików będzie umieszczony)

```
$mount
/dev/sda3 on / type ext4 (rw,errors=remount-ro)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
...
```

Przykład 1-6 Uzyskiwanie informacji o zamontowanych systemach plików

Każdy wiersz zawiera:

- Nazwa urządzenia
- Słowo on
- Punkt montowania
- Słowo type
- Opcje

Składnia polecenia `mount`:

```
mount -t typ urządzenie punkt montowania
```

Zwykle nie podajemy typu bo system sam to rozpoznaje

```
$sudo mount /dev/sdb1 /home/juka/usb
$mount
...
/dev/sdb1 on /home/juka/usb type vfat
```

Przykład 1-7 Montowanie dysku USB w katalogu usb (musi wcześniej istnieć)

Odmontowanie systemu plików:

```
umount punkt_montowania lub urządzenie
```

Identyfikator UUID

Montowanie systemu plików wymaga podania identyfikatora

### 1.2.2. Identyfikator systemu plików UUID

Montowanie systemu plików wymaga podania nazwy urządzenia. Nazwa ta może zależeć od kolejności wykrywania urządzeń i się zmieniać. Aby rozwiązać ten problem wprowadzono mechanizm uniwersalnego identyfikatora UUID (ang. *Universal Unique Identifier*).

- Identyfikator UUID jest nadawany podczas inicjacji systemu plików.
- Identyfikatory UUID można uzyskać za pomocą polecenia `blkid`.



```
$blkid
/dev/sda1: TYPE="ntfs" UUID="72C0DE8EC0DE57C5" LABEL="windows"
/dev/sda2: UUID="30fcb748-ad1e-4228-af2f-951e8e7b56df" SEC_TYPE="ext2"
TYPE="ext3"
/dev/sda5: TYPE="swap" UUID="8c4e69f8-5074-42c0-8134-0b2429c4c02c"
/dev/sdb1: SEC_TYPE="msdos" UUID="4848-E35A" TYPE="vfat"
```

Przykład 1-8 Uzyskanie UUID dysków za pomocą polecenia blkid

### 1.2.3. Buforowanie dysku

Aby podnieść szybkość działania mechanizmu dostępu do dysków stosuje się buforowanie danych przed ich zapisem na dysk. Podnosi to szybkość działania ale powoduje groźbę niespójności lub utraty danych przy niekontrolowanym wyłączeniu systemu. Zapis z buforów na dysk następuje:

- Przez system operacyjny – co jakiś czas
- Odmontowanie systemu plików polecenie umount
- Polecenie sync

Buforowanie ma miejsce też przy odczycie z dysku, dane odczytywane są na zapas.

### 1.2.4. Tabela systemów plików /etc/fstab

W pliku /etc/fstab znajduje się informacja o systemach plików które mają być zamontowane podczas startu systemu. W kolejnych wierszach opisany jest sposób montowania kolejnych urządzeń.

```
$cat /etc/fstab
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda3 during installation
UUID=b0d57349-73e8-4ba7-9d65-f354ae97ba0e / ext4 errors=remount-ro 0 1
# swap was on /dev/sda4 during installation
UUID=270037b7-4691-4450-9563-96ab0c5ad38e none swap sw 0 0
/dev/fd0 /media/floppy0 auto rw,user,noauto,exec,utf8 0 0
```

Przykład 1-9 Zawartość pliku /etc/fstab.

Znaczenie pól:

UUID lub nazwa urządzenia	Identyfikator urządzenia UUID lub nazwa (np. /dev/sda3)
Punkt montowania	Miejsce w systemie plików gdzie uwidoczni się urządzenie
Typ systemu plików	Typ np. ext3, ext4, fat, ntfs
Opcje	Lista opcji montowania
Informacja o kopii bezpieczeństwa	Wartość 0
Kolejność sprawdzania spójności systemu plików	W partycji podstawowej 1 a 2 dla innych partycji, 0 gdy kontrola nie jest prowadzona.

Przykład 1-10 Znaczenie pól linii pliku /etc/fstab

### 1.2.5. Sprawdzanie zajętości systemu plików

Gdy zabraknie miejsca w danym systemie plików grozi to awaria systemu, stąd ważne jest okresowe testowanie zajętości dysków. Służy temu polecenie **df**.

```

$df
System plików      1K-bl      użyte dostępne %uż. zamont. na
udev                2008416         4   2008412    1% /dev
tmpfs               403840        1060   402780    1% /run
/dev/sda3           80375616 30873204 45396428   41% /
none                 4              0         4     0% /sys/fs/cgroup
none                 5120           0        5120    0% /run/lock
none                2019200        124   2019076    1% /run/shm
none                102400         64   102336    1% /run/user

```

Przykład 1-11 Badanie zajętości dysków za pomocą polecenia df

System plików	Nazwa urządzenia
1K-bl	Pojemność całkowita systemu plików w liczbie 1 Kb bloków
użyte	Liczba zajętych bloków
dostępne	Liczba zajętych bloków
%uż	Procent zajętych bloków
Zamont. na	Punkt montowania

Tab. 1-1 Pola generowane poleceniem df

Bardziej szczegółowe dane o tym gdzie jest najwięcej danych można uzyskać za pomocą polecenia `du`.

`du -opcje katalog`

Zwykle generowane jest bardzo dużo danych. Opcja `-d` poziom pozwala zredukować przegląd do określonego poziomu. Poniżej dano przykład wyświetlania zajętości katalogu `/usr` do głębokości 1. Wyniki podawane są domyślnie w blokach 1024 bajtów.

```
$du -d 1 /usr
2147240 /usr/lib
520     /usr/games
1807484 /usr/share
120     /usr/local
25764  /usr/sbin
446208 /usr/src
149336 /usr/bin
178952 /usr/include
154312 /usr/arm-none-eabi
6796   /usr/lib32
4916736 /usr
```

Przykład 1-12 Dane o zajętości katalogu `/usr` otrzymane za pomocą polecenia `du`

### 1.2.6. Naprawa systemu plików

System plików składa się z samych plików i rodzaju bazy danych zawierającej informacje o ich rozmieszczeniu i atrybutach. System ten musi być spójny. Np. gdy dopisujemy coś do pliku to muszą być wykonane:

- Zmiana zawartości pliku
- Aktualizacja „bazy danych” – iwęzła, bitmapy, katalogu

Aktualizacja ta musi być transakcją. Do naruszenia spójności zachodzi na ogół przez niekontrolowane wyłączenie zasilania, aktualizacja może wtedy nie być kompletna.

Narzędziem sprawdzania systemu plików jest program `fsck` (ang. *file system check*).

```
# fsck /dev/sdb
```

Jako parametr podać trzeba punkt montowania który można znaleźć w pliku `/etc/fstab`.

Polecenie nie może być wykonywane gdy system plików jest zamontowany.

### Przykład

Dysk USB `/dev/sdb1` zamontowany jest w katalogu `/media/juka/UDISK 2.0`.

Czynności które wykonujemy to:

1. Wyświetlenie zawartości katalogu `/media`: `ls /media/juka`
2. Sprawdzenie jakie urządzenia usb są widoczne: `lsusb`
3. Sprawdzenie jakie systemy plików są zamontowane: `mount`
4. Odmontowanie systemu plików USB: `umount /dev/sdb1`
5. Sprawdzenie systemu plików: `fsck`

```
$ls /media/juka
UDISK 2.0
$lsusb
Bus 001 Device 003: ID 13fe:1e00 Kingston Technology
Company Inc. Flash Drive 2 GB [ICIDU 2 GB]
$mount
/dev/sda3 on / type ext4 (rw,errors=remount-ro)
/dev/sdb1 on /media/juka/UDISK 2.0 type vfat
$umount /dev/sdb1
$sudo fsck -p /dev/sdb1
fsck z pakietu util-linux 2.20.1
fsck.fat 3.0.26 (2014-03-07)
There are differences between boot sector and its
backup.
...
Not automatically fixing this.
/dev/sdb1: 33927 files, 449413/488450 clusters
```

Przykład 1-13 Sprawdzenie dysku USB `/dev/sdb1`

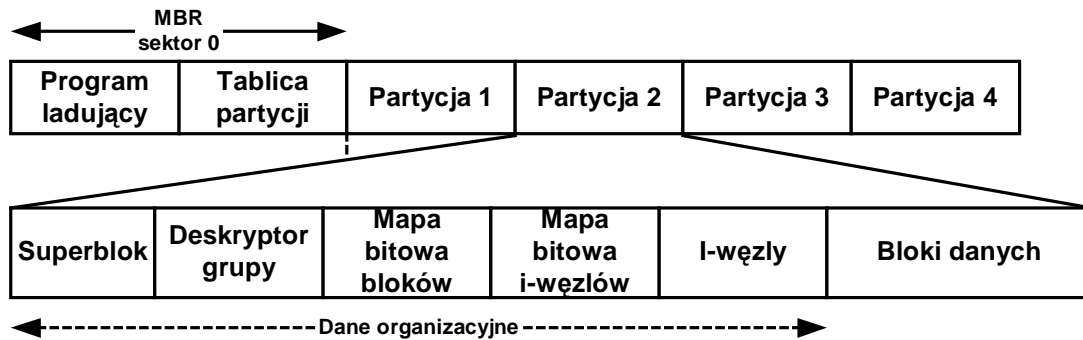
Program `fsck` posiada wiele opcji które specyfikują co robić w przypadku napotkania błędów (`-p` – napraw drobne błędy) . I-węzły nie przypisane do żadnych katalogów umieszczane są w katalogu `/lost+found`

## 1.3. Organizacja systemu plików ext2,ext3

### 1.3.1. System plików ext2

System plików składa się z dwóch komponentów:

- Plików z danymi
- Bazy danych potrzebnej do zarządzania tymi plikami: zawartość, położenie, atrybuty



Rys. 1-4 Organizacja dysku typu ext2

- Superblok – zawiera informacje o globalnych parametrach partycji: UUID, wielkość partycji, rozmiar bloku, liczba bloków, liczba i-węzłów, czas utworzenia, itd.
- Deskryptor grupy – zawiera informacje o położeniu map bitowych, liczbie wolnych bloków, liczbie i-węzłów, liczbie katalogów.
- Mapa bitowa bloków – zawiera informacje który blok jest wolny a który zajęty. Każdemu blokowi odpowiada jeden bit 1 – blok zajęty, 0 – blok wolny.
- Mapa bitowa i-węzłów – zawiera informacje który i-węzeł jest wolny a który zajęty.
- I-węzeł – zawiera informacje o atrybutach pliku i blokach wchodzących w skład pliku.

```

$sudo tune2fs -l /dev/sda3
tune2fs 1.42.9 (4-Feb-2014)
Filesystem volume name:    <none>
Last mounted on:          /
Filesystem UUID:          b0d57349-73e8-4ba7-9d65-
f354ae97ba0e
Filesystem magic number:   0xEF53
Filesystem features:       has_journal ext_attr
resize_inode dir_index filetype needs_recovery
extent flex_bg sparse_super large_file huge_file
uninit_bg dir_nlink extra_isize
Filesystem flags:          signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Filesystem OS type:       Linux
Inode count:              5120000
Block count:              20448000
Reserved block count:     1022400
Free blocks:              12348432
Free inodes:              4760762
First block:              0
Block size:               4096
Fragment size:           4096
Blocks per group:         32768
Fragments per group:     32768
Inodes per group:         8192
Inode blocks per group:   512
Flex block group size:    16
Filesystem created:       Tue Jan 27 10:02:37 2015
Last mount time:          Thu Mar 3 11:45:01 2016
Last write time:          Thu Mar 3 11:45:01 2016
Mount count:              54
Maximum mount count:      -1
Last checked:             Tue Jan 27 10:02:37 2015
Check interval:          0 (<none>)
Lifetime writes:          68 GB
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               256
...

```

Przykład 1-14 Zawartość superbloku - fragment

Podstawową jednostką informacji o pliku jest i-węzeł (ang. i-node). I-węzły są ponumerowane od 1 do maksimum. I-węzły identyfikowane są za pomocą numerów.

```
$ls -i
3419808 blkid.txt
3540501 build-przyklad1-Desktop-Debug
3407885 Dokumenty
3540549 cpp
```

Przykład 1-15 Informacja na temat numerów i-węzłów odpowiadających plikom

I-węzeł zawiera informacje potrzebne do zarządzania plikiem. Są one następujące:

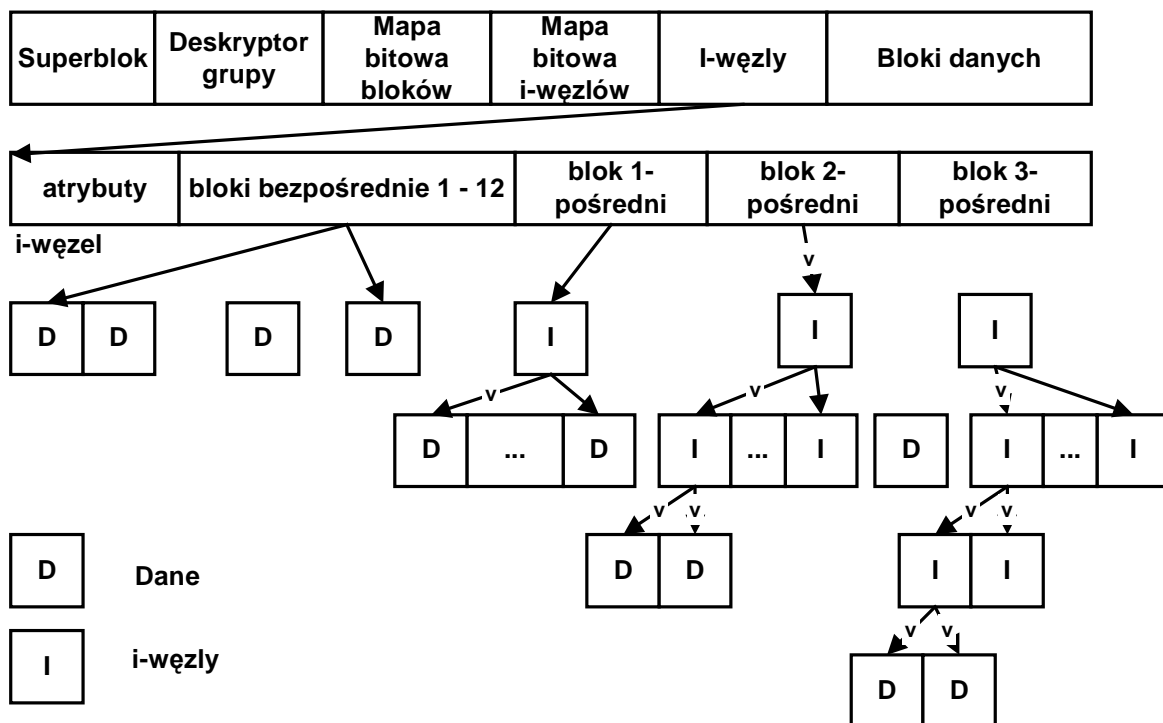
- Z jakich sektorów składa się plik
- Atrybuty pliku: właściciel, typ, uprawnienia, czasy utworzenia, modyfikacji, dostępu

```
$df -i
System plików    iwęzły użyteI    wolneI    %uż.I    zamont.
na
udev              502104         485      501619      1% /dev
tmpfs             504800         475      504325      1% /run
/dev/sda3         5120000       359273   4760727      8% /
```

Przykład 1-16 Informacja na temat liczby i-węzłów w poszczególnych systemach plików

Nazwa pola	Zawartość
<b>mode</b>	Typ pliku, prawa dostępu, bity SETUID i SETGID
<b>links</b>	Liczba dowiązań
<b>uid</b>	Identyfikator właściciela pliku
<b>gid</b>	Identyfikator grupy właściciela pliku
<b>size</b>	Wielkość pliku w bajtach
<b>addr</b>	Adres pierwszych 12 bloków bezpośrednich i trzech pośrednich (1,2,3)
<b>gen</b>	Numer generacji zwiększany przy każdym dostępie
<b>atime</b>	Czas ostatniego dostępu do pliku
<b>mtime</b>	Czas ostatniej modyfikacji pliku
<b>ctime</b>	Czas ostaniej modyfikacji parametrów i-węzła

Tab. 1-2 Zawartość i-węzła



Rys. 1-5 Organizacja dysku typu ext2

Jeśli blok ma 1024 bajty a adres bloku zajmuje 4 bajty to blok 1-pośredni może zawierać informacje o położeniu 256 bloków. Adresacja 1-pośrednia umożliwia zapamiętanie pliku o wielkości do  $256 \cdot 1024 + 12 \cdot 1024 = 268$  KB.

Przy blokach 1 kB schemat adresacji 3-pośredniej umożliwia zapisanie plików do 16 GB.

Przy blokach 8 kB schemat adresacji 3-pośredniej umożliwia zapisanie plików do 64 TB.

Informacje o pliku można uzyskać za pomocą polecenia `stat` jak pokazano poniżej.

```

$stat kat1.txt
Plik: „kat1.txt”
rozmiar: 113 bloków: 8 bloki I/O: 4096    zwykły plik
Urządzenie: 803h/2051d    inody: 3413085    dowiązań: 1
Dostęp: (0644/-rw-r--r--) Uid:(1000/juka)
Gid:(1000/juka)
Dostęp:      2016-01-21 19:17:52.927149764 +0100
Modyfikacja: 2016-01-21 19:17:52.951149764 +0100
Zmiana:      2016-01-21 19:17:52.951149764 +0100
Utworzenie:  -

```

Przykład 1-17 Uzyskanie atrybutów pliku za pomocą polecenia `stat`



### 1.3.2. System plików ext3

Gdy aktualizowany jest jakiś plik, powiedzmy że dopisujemy coś na końcu, to trzeba wykonać czynności:

- Zapisać nowy sektor
- Zmienić i-węzeł
- Zmienić bitmapę

Te czynności muszą być wykonane od początku do końca, stanowiąc pojedynczą transakcję. Jeśli by natychmiast modyfikować te trzy obszary, zajęłoby to dużo czasu ze względu na ruch głowic. Z tego powodu modyfikację wprowadza się w buforze w pamięci RAM, który to bufor co jakiś czas (np. 30 sek) zapisywany jest na dysk.

Niekontrolowane wyłączenie zasilanie może spowodować utratę spójności systemu plików. Aby zabezpieczyć się przed taką awarią wprowadzono mechanizm księgowania (ang. *journaling*). System plików ext3 wykorzystuje ten mechanizm.

Mechanizm księgowania wykorzystuje dziennik. Jest to bufor cykliczny w którym odnotowywane są wszystkie zapisy sektorów. Dziennik nie może być na tym samym systemie plików. Do prowadzenia operacji odczytu/zapisu w dzienniku stosuje się urządzenie JBD (ang. *Journaling Block Device*). Wykorzystuje się trzy operacje:

- Rejestr zapisów dziennika
- Operacje atomowe
- Transakcje

Operacjami atomowymi są zapisy sektorów, transakcję wyznacza wywołanie systemowe. Gdy przyjdzie potwierdzenie że zaktualizowano wszystkie struktury danych, wpis z dziennika może być usunięty. Aby zwiększyć szybkość działania, istnieją systemy plików księgujące tylko metadane (np. XFS). Inne systemy plików z kronikowaniem to NTFS, ReiserFS

Zalety systemu plików ext3:

- Zgodność z ext2 (dysk ex2 może być zamontowany jako ext3)
- Większa szybkość niż ext2
- Większa niezawodność względem ext2

## 1.4. Katalogi

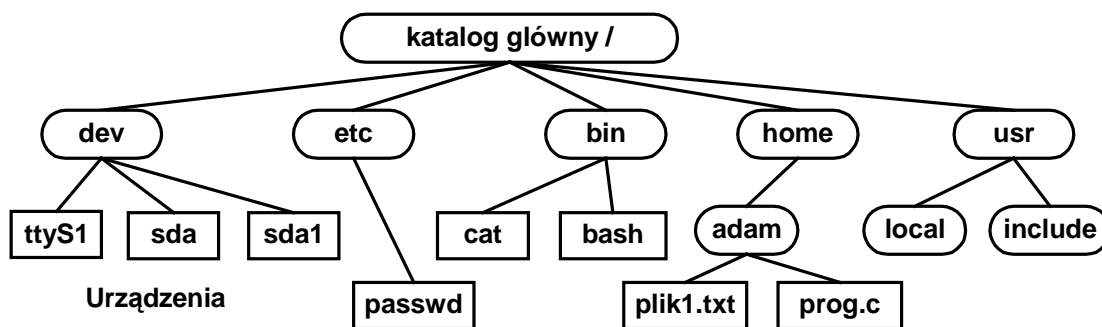
Opisany mechanizm umożliwia zapis plików ale nie wprowadza ich uporządkowania. Do tego celu stosuje się mechanizm katalogów. Celem istnienia katalogów jest ułatwienie dostępu do plików i nadanie im uporządkowanej struktury.

**Katalog jest to zbiór nazw plików.**

Własności katalogów:

- Katalog jest plikiem specjalnym.
- Katalogi mogą być zagnieżdżane.
- Katalogi mają strukturę drzewa. Każdy podkatalog ma dokładnie jeden katalog macierzysty.
- Węzły końcowe są: plikami regularnymi, plikami specjalnymi, katalogami
- Jeden katalog – katalog główny jest wyróżniony. Tworzy on wierzchołek drzewa katalogów a jego katalogiem macierzystym jest on sam.

System plików – zbiór katalogów i plików. Mają budowę hierarchiczną. Zazwyczaj system plików związany jest z partycją dyskową lub innym urządzeniem pamięciowym.



Rys. 1-3 Fragment systemu plików

Położenie pliku w systemie plików definiowane jest przy pomocy ścieżki którą należy przebyć od korzenia systemu plików / do danego pliku. Ścieżka zaczyna się od / i zawiera nazwy podkatalogów oddzielone znakiem "/" a na końcu nazwę pliku. Np.:

`/home/adam/prog.c`

Jest to tak zwana ścieżka bezwzględna – zaczyna się ona od znaku "/"

Implementacja katalogów:

- Katalogi są plikami.
- Pliki zawierają pozycje odpowiadające plikom lub katalogom.
- Każda z pozycji zawiera co najmniej: nazwę pliku i numer i-węzła. Numer i-węzła jednoznacznie identyfikuje plik w ramach systemu plików.

#### Kropka i podwójna kropka

W każdym katalogu zawarte są dwie pozycje:

- .. podwójna kropka – jest łączem do katalogu macierzystego
- . pojedyncza kropka – jest łączem do bieżącego katalogu

Prawa dostępu dotyczące katalogów

Podobnie jak do plików regularnych do katalogów stosują się prawa dostępu dla właściciela, grupy i innych. Mają jednak inne znaczenie.

Odczyt	r	Można uzyskać nazwy plików i katalogów zawartych w tym katalogu
Zapis	w	Można tworzyć i kasować pliki w podkatalogu
Wykonanie	x	Można przejść do katalogu za pomocą polecenia <code>cd</code> lub funkcji <code>chdir</code> . Aby wykonać program lub utworzyć plik trzeba mieć prawa wykonania na całej bezwzględnej ścieżce biegnącej do pliku.

Tab. 1-3 Prawa dostępu dotyczące katalogów

### **1.5. Literatura:**

- [1] Andrew S. Tannenbaum, Systemy operacyjne wydanie 3, Helion Gliwice 2010.
- [2] Brian Ward, Jak działa Linux, podręcznik administratora, Helion 2015.