

Systemy Operacyjne

System plików

Dr hab. inż. Krzysztof Rzecki, prof. AGH

Na podstawie: Abraham Silberschatz, *Koncepcje systemów operacyjnych*



System plików - wprowadzenie

- System plików umożliwia na dostęp do przechowywanych:
 - programów i danych,
 - dla systemu operacyjnego i użytkowników.
- System plików składa się z dwóch części:
 - kolekcji plików do bezpośredniego przechowywania danych,
 - struktury katalogów, która organizuje pliki i przechowuje informacje o nich.
- Zwykle system plików znajduje się na urządzeniach z pamięcią nieulotną, ale w szczególnych przypadkach może to być np. pamięć RAM.



Koncepcja pliku

- Komputer może przechowywać informacje na różnych nośnikach: dysk twardy, taśma magnetyczna, dysk optyczny, pamięć USB, urządzenie NVM.
- System operacyjny dostarcza mechanizm nadający logiczny widok na przechowywane informacje - tworzy abstrakt między własnościami fizycznymi urządzenia do przechowywania, a logiczną strukturą pliku. W ten sposób pliki są mapowane przez system operacyjny na urządzenie fizyczne.
- Dla użytkownika plik to kolekcja informacji zapisana w pamięci zapasowej, która nie może być zapisana dopóki nie jest zawarta do postaci pliku.
- Pliki zwykle reprezentują programy (zarówno źródłowe, jak i binarne) oraz dane.
- Pliki mogą być w formie luźnej, jak pliki tekstowe, albo ściśle ustrukturyzowane (źródła, binaria).



Atrybuty plików

- Nazwa pliku (kwestia wielkości liter, kwestia rozszerzenia).
- Plik jest bytem niezależnym od procesu, użytkownika, a nawet systemu operacyjnego plików.
- Atrybuty plików zależą od systemu plików, zwykle są to:
 - Nazwa - symboliczna nazwa pliku, jedyna informacja o pliku w zrozumiałym dla człowieka zapisie.
 - Identyfikator - unikalne oznaczenie, zwykle liczba, która identyfikuje plik w systemie plików.
 - Typ - rozróżnienie pliku.
 - Położenie - informacja o położeniu pliku na dysku.
 - Rozmiar - bieżący rozmiar pliku (w bajtach, słowach lub blokach).
 - Zabezpieczenia - informacje o kontroli dostępu dotyczące czytania, pisania i uruchamiania (itp.).
 - Znacznik czasu - informacje o znacznikach czasu mogą obejmować utworzenie, ostatnią modyfikację oraz ostatnie użycie pliku.
 - Identyfikator użytkownika i grupy - informacje dot. właściciela pliku.
- Informacje o plikach przechowywane są w strukturze katalogów (w pliku katalogu).



Operacje na plikach

- Tworzenie pliku - dwa kroki: rezerwacja miejsca, wpis w katalogu.
- Otwieranie - sprawdzenie uprawnień, itp. i wywołanie `open()`.
- Zapis - wywołanie systemowe z danymi do zapisu oraz `open()`. Zapis odbywa się sekwencyjnie, więc system musi przechowywać informację o wskaźniku miejsca do zapisu. Przypadek nadpisywania i dopisywania.
- Czytanie - wywołanie systemowe z wskaźnikiem do pliku oraz informacją, gdzie do pamięci zawartość pliku ma zostać wczytana. Podobnie, jak przy zapisie, system potrzebuje wskaźnika odczytu (i zwykle jest to ten sam wskaźnik).
- Pozycjonowanie w pliku - wskaźnik położenia w pliku, ang. *seek*.
- Usuwanie pliku - znalezienie pliku w katalogu, zwolnienie miejsca, oznaczenie w katalogu (wymazanie wpisu). Niektóre systemy plików: kwestia twardych dowiązań.
- Obcinanie pliku - np. usuwanie zawartości, ale zachowanie atrybutów.



Typy plików

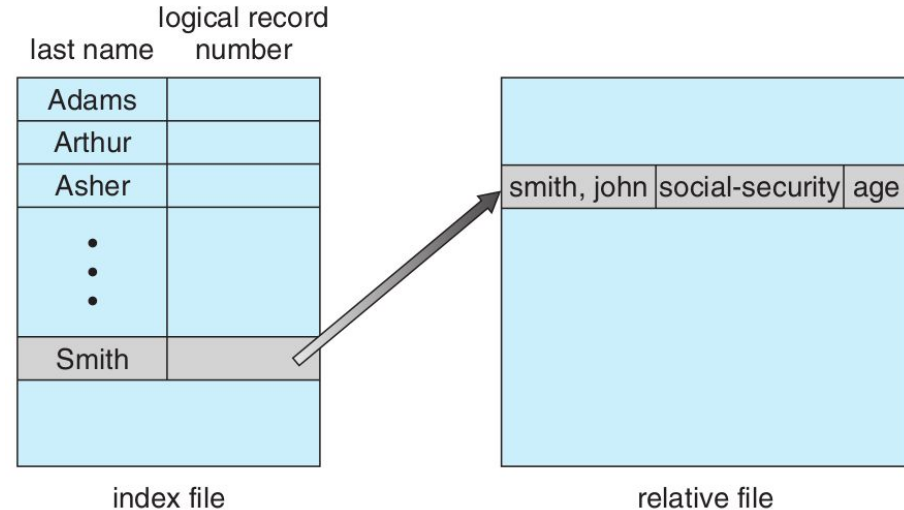
- Typ pliku rozpoznawany po rozszerzeniu nie jest tym samym, co typ pliku rozpoznawany w atrybutach pliku.

file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, perl, asm	source code in various languages
batch	bat, sh	commands to the command interpreter
markup	xml, html, tex	textual data, documents
word processor	xml, rtf, docx	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	gif, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	rar, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, mp3, mp4, avi	binary file containing audio or A/V information

Struktura katalogu

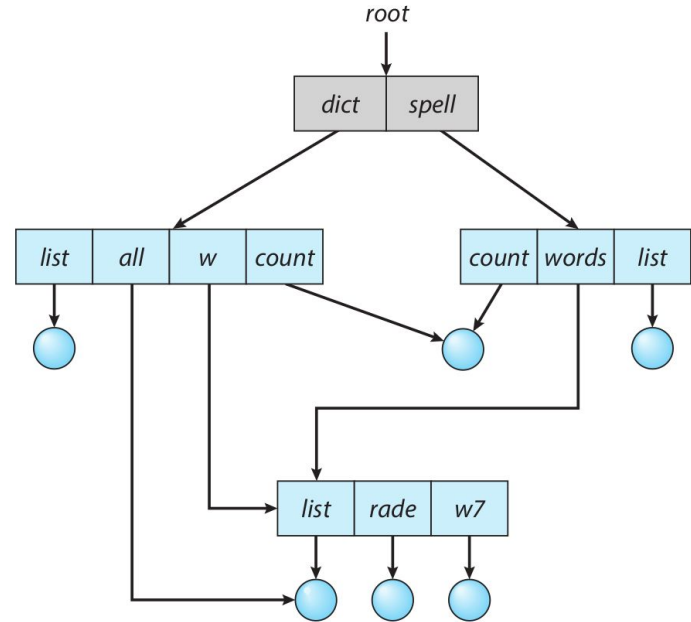
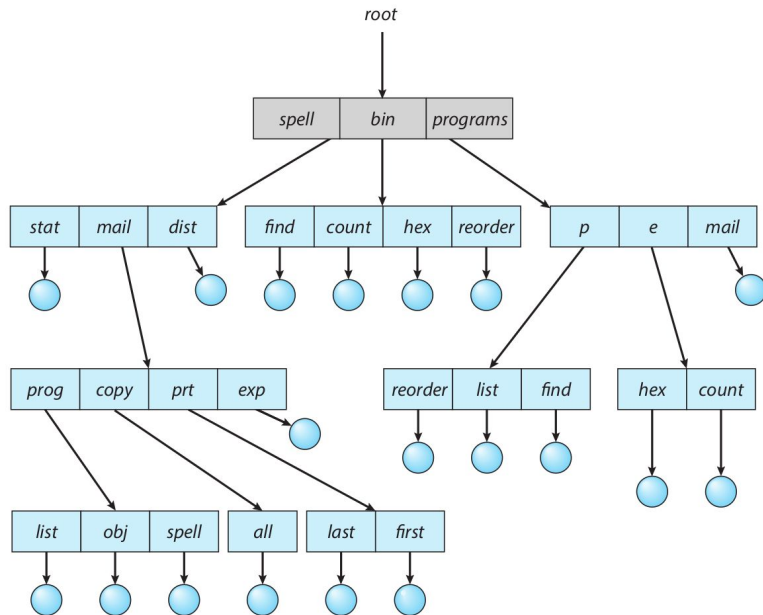
Operacje:

- Szukanie pliku.
- Tworzenie pliku.
- Usuwanie pliku.
- Listowanie zawartości katalogu.
- Zmiana nazwy pliku.
- Trawers systemu plików.



Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*

Zaawansowane struktury katalogów



Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*



Rodzaje dostępu

Podstawowe:

- **Read** - odczyt.
- **Write** - zapis.
- **Execute** - wykonywanie.
- **Append** - dopisywanie na końcu.
- **Delete** - usuwanie.
- **List** - listowanie nazwy i atrybutów pliku.
- **Attribute change** - zmiana wartości atrybutów.

A także:

- **Renaming** - zmiana nazwy.
- **Copying** - kopiowanie.
- **Editing** - edycja.

Uprawnienia do plików w Linux ext

```
krz@zinc:~/abc$ ls -al
razem 44
drwxrwxr-x  2 krz krz  4096 paź  9 18:40 .
drwx----- 98 krz krz 28672 paź  9 18:40 ..
-rw-rw-r--  1 krz krz    0 paź  9 18:40 file.txt
krz@zinc:~/abc$
```

- Katalog bieżący: ~/abc oraz .
- Katalog nadrzędny: ..

Ustawianie uprawnień:

```
$ chmod uprawnienia plik
$ chmod 644 file.txt
$ chmod a+rx,a-w directory
```

Uprawnienia, przykład:

```
drwxr-x---
0123456789
```

Pozycja 0: d (dir), l (link), b (block), c (character)

Pozycja 1, 2 i 3: uprawnienia właściciela 'u'

Pozycja 4, 5 i 6: uprawnienia grupy 'g'

Pozycja 7, 8 i 9: uprawnienia pozostałych 'o'

Pozycje 1..9: uprawnienia wszystkich 'a'

rwX - read, write, eXecute

421 - zapis numeryczny

np. r-x = 5, rw- = 6, r-- = 4.

Uprawnienia do plików w Linux ext

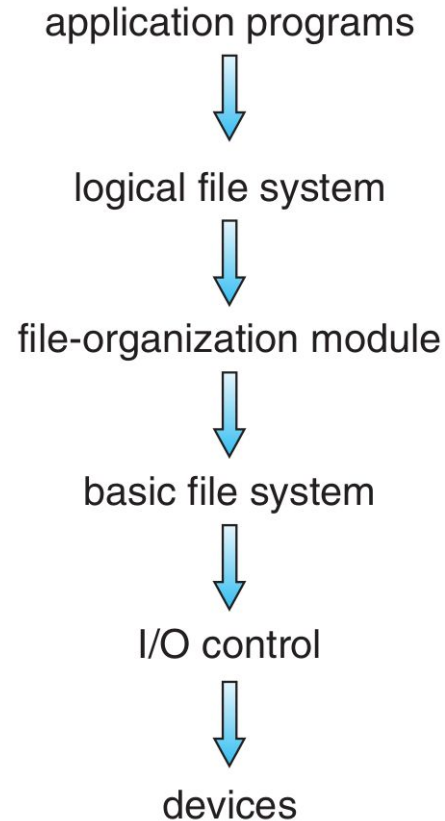
0	---	brak uprawnień	blokada
1	--x	wykonywanie	nieprzydatne
2	-w-	zapis	zbieranie sekretnych logów (plik)
3	-wx	zapis i wykonywanie	zbieranie sekretnych logów (dir)
4	r--	odczyt	stała konfiguracja
5	r-x	odczyt i uruchamianie	pliki wykonywalne, katalogi
6	rw-	odczyt i zapis	pliki edytowalne
7	rwx	odczyt, zapis i uruchamianie	skrypty i katalogi usera :-)
	??s	bit suid	programy specjalne
	??t	sticky bit	katalog specjalny

Struktura systemu plików

Założenia:

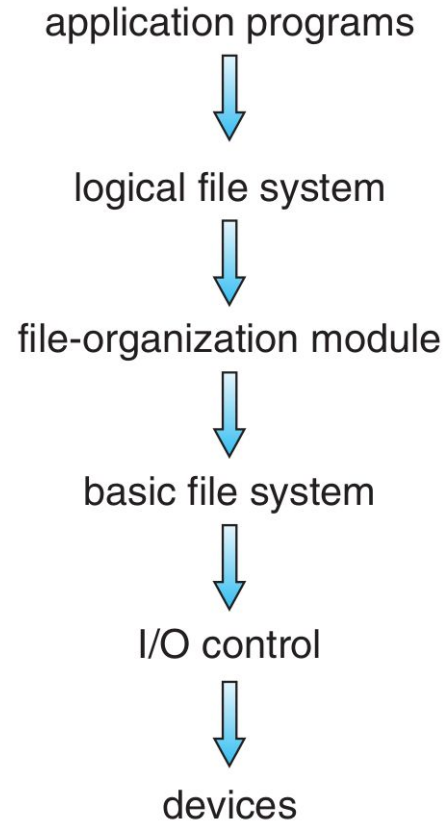
1. Dysk może być nadpisywany w miejscu; możliwy jest odczyt bloku, modyfikacja bloku zapis bloku w to samo miejsce.
2. Możliwe jest sięgnięcie w dowolny blok (sektorów) na dysku, czyli dostęp do pliku może być realizowany sekwencyjnie lub swobodnie.

Transfer danych z/na dysk zorganizowany jest w bloki wielkości zwykle 512 lub 4096 bajtów.



Struktura systemu plików

- I/O - sterownik urządzenia oraz przerwań do transferu danych.
- Podstawowy system plików - blokowy sybsystem I/O w Linux.
- Moduł organizacji plików - posiada wiedzę na temat plików oraz logicznych bloków, a także wolnego miejsca.
- Logiczny system plików - zarządza metadanymi, strukturą katalogów, wykorzystuje FCB (file control block) w np. Linux poprzez inode.



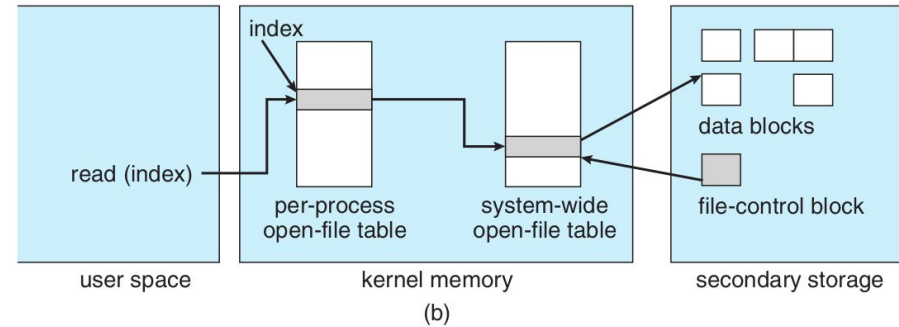
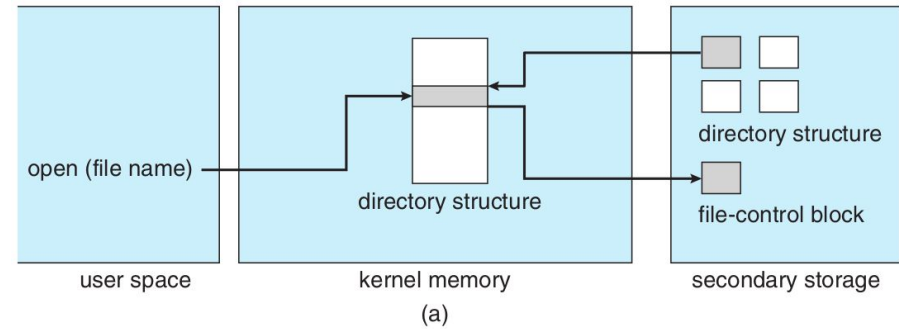


Operacje systemu plików

- Blok kontrolny ładowania (ang. *Boot control block*), Główny rekord rozruchowy (ang. *Master Boot Record*) - informacje zapisane w pierwszym sektorze dysku (CHS = 0,0,1), zajmuje jeden sektor, czyli 512 bajtów i zawiera (w przypadku Linux) program rozruchowy systemu operacyjnego (446 bajtów) oraz tablicę partycji.
Polecenie podglądu MBR: `hd -n 512 /dev/sda`
Polecenie podglądu partycji: `cat /proc/partitions`
- GPT, ang. *Globally Unique Partition Table* - nowszy typ rekordu ładowania.
- Blok kontrolny wolumenu (ang. *Volume control block*) - informacje o partycji/wolumenie, jak liczbę bloków w wolumenie, rozmiar bloków, liczba wolnych bloków, itp.
- Struktura katalogów zawierająca informacje o organizacji plików.
- Blok kontrolny pliku (ang. *File Control Block*) - zawiera wiele informacji o pliku. Posiada unikalny identyfikator.

Otwarcie/odczyt pliku

- (a) Otwarcie pliku
- (b) Odczyt pliku



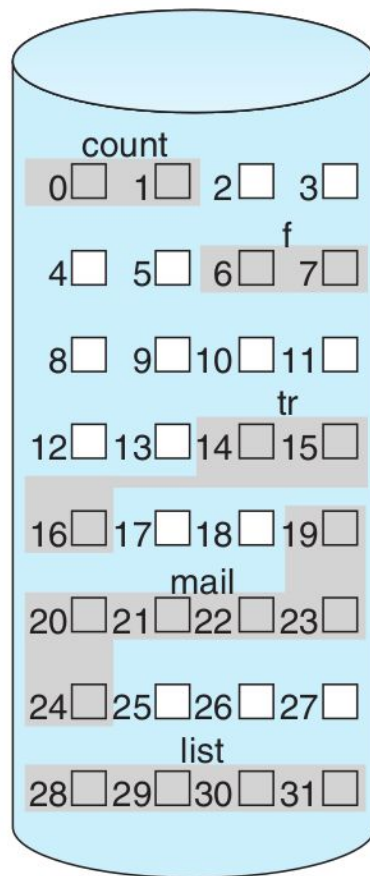


Metody alokacji plików

Na kolejnych slajdach:

- Alokacja ciągła - ang. *Contiguous Allocation*
- Alokacja połączona - ang. *Linked Allocation*
- Alokacja indeksowana - ang. *Indexed Allocation*

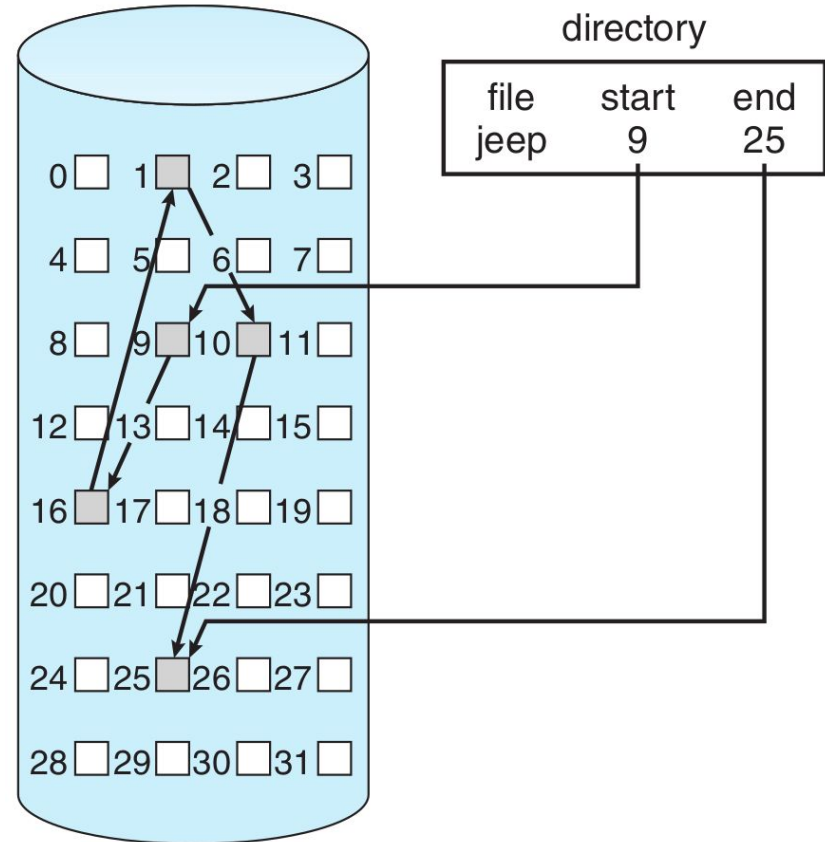
Alokacja ciągła



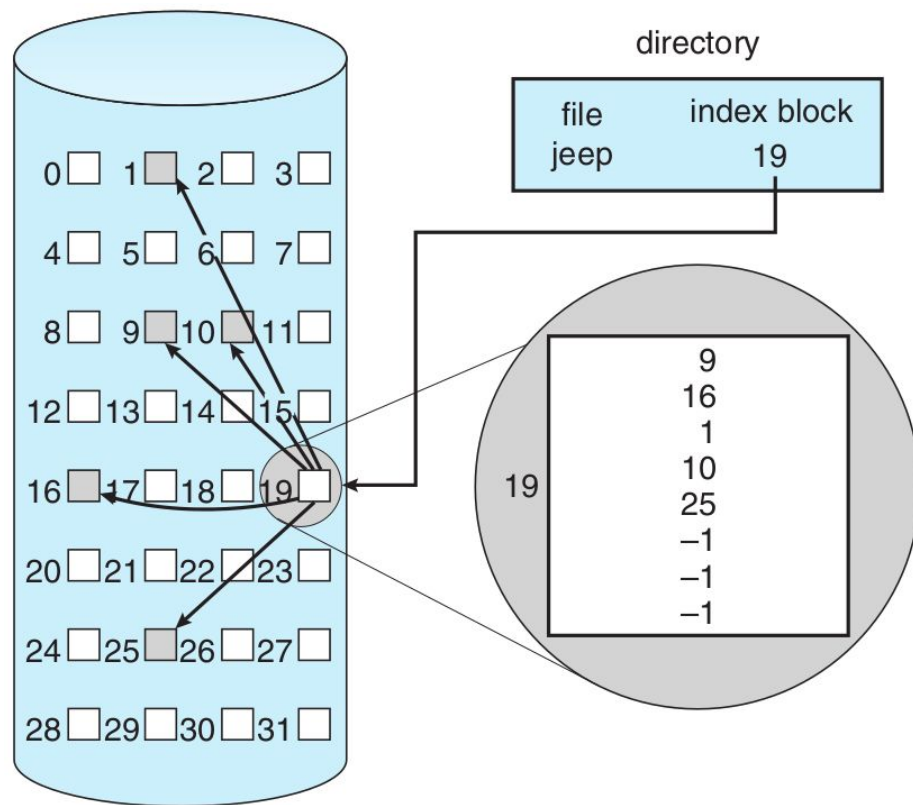
directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

Alokacja połączona



Alokacja indeksowana



Tworzenie pliku na urządzenie blokowe



- Utworzenie pustego pliku (100 MB), z którego zrobimy urządzenie blokowe:

```
$ dd if=/dev/zero of=disc0 bs=1M count=100
```

```
$ file -s disc0
```

```
disc0: data
```

```
$ ls -sh disc0
```

```
101M disc0
```


Tworzenie urządzenia blokowego z pliku

- Znajdź pierwsze wolne urządzenie blokowe:

```
$ losetup -f
```

```
np.:
```

```
/dev/loop12
```

```
$ sudo file -s `losetup -f`
```

```
/dev/loop12: empty
```

- Utworzenie z pliku urządzenia blokowego:

```
$ losetup `losetup -f` disc0
```

```
$ file -s /dev/loop12
```

```
/dev/loop12: data
```

Zakładanie partycji na urządzeniu blokowym

- Podział tak utworzonego urządzenia na 2 partycje po 50 MB:

```
$ sudo fdisk /dev/loop12
    n, [Enter], p, 3 x [Enter], +50M, Enter
    n + [Enter], p, 4 x [Enter], p, [Enter], w, [Enter]
```

- Synchronizacja z jądrem systemu operacyjnego:

```
$ sudo partprobe /dev/loop12
```

```
$ sudo file -s /dev/loop12
/dev/loop12: DOS/MBR boot sector; partition 1 : ID=0x83, start-CHS
(0x0,32,33), end-CHS (0x6,127,57), startsector 2048, 102400 sectors;
partition 2 : ID=0x83, start-CHS (0x6,127,58), end-CHS (0xc,190,50),
startsector 104448, 100352 sectors
```

```
$ sudo file -s /dev/loop12p1
/dev/loop12p1: data
```

Formatowanie (zakładanie systemu plików)



- Utworzenie systemu plików na pierwszej z partycji:

```
$ mkfs.ext4 /dev/loop12p1
```

```
$ file -s /dev/loop12p1
```

```
/dev/loop0p1: Linux rev 1.0 ext4 filesystem data,  
UUID=c22af2f0-449d-4119-a117-45edfa32e287 (extents) (64bit)  
(large files) (huge files)
```

- Narzędzia do serwisowania systemu plików:

- \$ e2fsck - sprawdzanie i/lub naprawa systemu plików
- \$ fsck.ext4 - alias do j.w.
- \$ tune2fs - zmiana konfiguracji systemu plików
- \$ resize2fs - zmiana wielkości systemu plików
- \$ debugfs - debugging, w tym przeglądanie i edycja systemu plików

Zarządzanie systemem plików

- Podłączenie pliku jako urządzenia blokowego:

```
$ losetup /dev/loop12 disc0
```

// już jest z poprzednich slajdów

- Podmontowanie systemu plików z pierwszej partycji:

```
$ mkdir mntdir
```

```
$ sudo mount /dev/loop12p1 mntdir
```

```
$ df -h mntdir
```

```
/dev/loop12p1      43M    24K    40M    1% /home/krz/mntdir
```

- Odmontowanie systemu plików:

```
$ sudo umount mntdir
```

- Odłączanie urządzenia loopback

```
$ losetup -d /dev/loop12
```

Szyfrowanie partycji urządzenia blokowego

- Podłączenie pliku jako urządzenia blokowego (jak poprzednio), czyli :

```
$ losetup `losetup -f` disc0 // założmy, że to znów loop12
```

- Zaszyfrowanie urządzenia blokowego (druga z partycji):

```
$ sudo cryptsetup luksFormat /dev/loop12p2
```

```
$ sudo file -s /dev/loop12p2
```

```
/dev/loop12p2: LUKS encrypted file, ver 2 [, , sha256] UUID:
```

```
5bbf82ca-613b-4a32-8c1d-b18b2ad6f520
```

- Podłączenie zaszyfrowanej partycji,jako urządzenie blokowe:

```
$ sudo cryptsetup luksOpen /dev/loop12p2 encrypted_partition
```

```
$ ls -l /dev/mapper/encrypted_partition
```

```
/dev/mapper/encrypted_partition -> ../dm-0
```

Formatowanie zaszyfrowanej partycji

- Podłączenie pliku jako urządzenia blokowego (jak poprzednio), czyli :

```
$ losetup `losetup -f` disc0 // założmy, że to znów loop12
```

Podłączenie zaszyfrowanej partycji jako urządzenie blokowe (jak poprzednio):

```
$ sudo cryptsetup luksOpen /dev/loop12p2 encrypted_partition
```

- Utworzenie systemu plików na zaszyfrowanej partycji:

```
$ mkfs.ext4 /dev/mapper/encrypted_partition
```

- Montowanie zaszyfrowanej partycji:

```
$ mkdir mntdir_enc
```

```
$ sudo mount /dev/mapper/encrypted_partition mntdir_enc
```

```
$ df -h mntdir
```

```
/dev/mapper/encrypted_partition 27M 24K 25M 1% /home/krz/mntdir_enc
```


Odlączenie zaszyfrowanej partycji



- Odmontowanie:
`$ sudo umount mntdir_enc`
- Odlączenie zmapowania kryptograficznego:
`$ sudo cryptsetup luksClose encrypted_partition`
- Odlączenie urządzenia blokowego:
`$ losetup -d /dev/loop12`



Zadanie dodatkowe

- Zaprojektować i napisać program, który we wskazanych miejscach urządzenia blokowego odczyta zawartość bloków, zabezpieczy i nadpisze celem utworzenia sztucznego uszkodzenia.
- Następnie, oprogramowaniem do naprawy systemu plików naprawić uszkodzenie.
- Porównać systemy plików pod względem odporności na uszkodzenia.
- Opracować wynik w formie sprawozdania.



Grand Canyon