

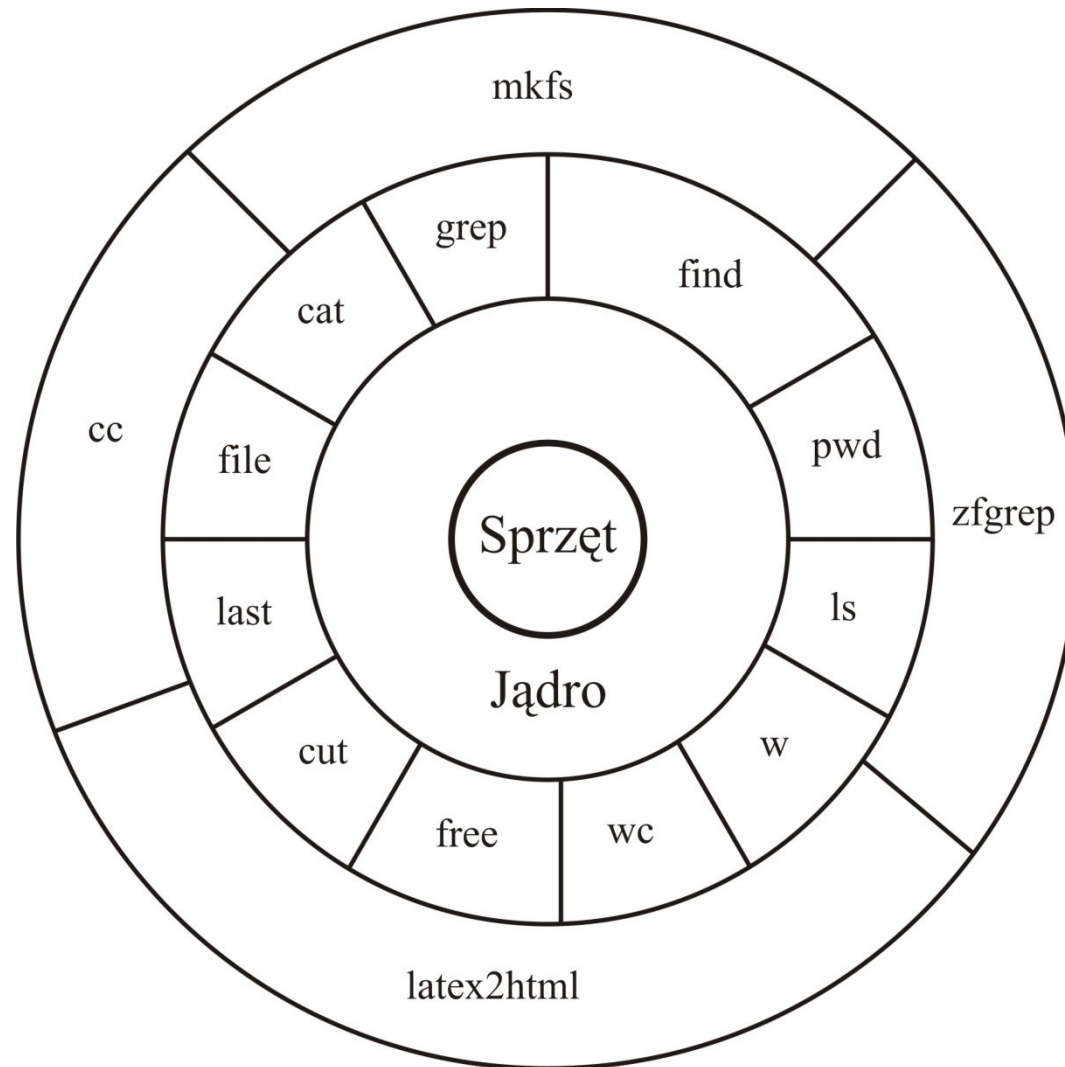


Wprowadzenie do systemu UNIX

Cz. 2:
Systemy plików

Krzysztof Boryczko

Architektura systemu



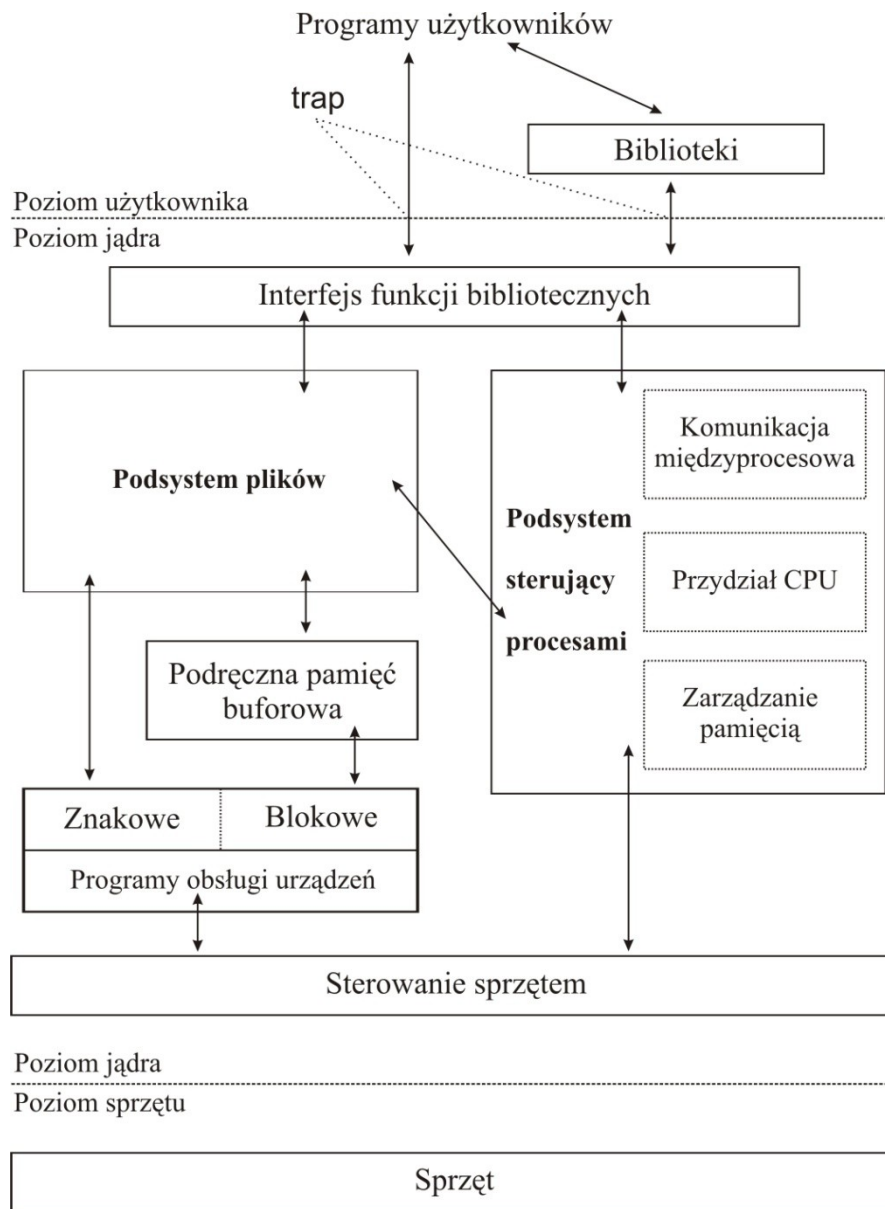
Opis schematu

- Sprzęt znajdujący się w środku systemu komputerowego dostarcza zasobów biernych.
- Bezpośredni dostęp do sprzętu ma jedynie jądro systemu operacyjnego. Stąd zmiana konfiguracji sprzętowej pociąga za sobą konieczność rekonfiguracji jądra.
- Kolejna warstwa to komendy systemowe i programy użytkowe. Nie mają one bezpośredniego dostępu do sprzętu.
- Proces może pracować w trybie użytkownika lub trybie jądra.
- Inne programy użytkowe mogą korzystać z programów i komend systemowych. Faktycznie warstw reprezentujących programy użytkowe może być więcej.

Rola systemu

- System UNIX wspiera iluzję:
 1. Systemu plików dysponującego „miejscem” do przechowywania informacji.
 2. Przestrzeni, w której wykonują się procesy użytkowników. Mówi się, że są one „żywe”.
- Stąd plik i proces stanowią podstawowe pojęcia w modelu systemu UNIX.

Schemat jądra



Funkcje systemowe

- Wraz z interfejsem bibliotecznym stanowią granicę między programami użytkownika a jądrem systemu.
- Zadaniem funkcji bibliecznych jest odwzorowanie wywołań funkcji w programach na wywołania funkcji systemowych, umożliwiających dostęp do jądra systemu.
- Funkcje biblieczne są łączone z kodem programu użytkownika po jego kompilacji, na etapie linkowania.
- Procesy komunikują się z systemem plików przy pomocy funkcji systemowych, z których najpopularniejsze to: *open*, *close*, *read*, *write*, *stat*, *chown* oraz *chmod*.
- Podstawowymi funkcjami sterowania procesami są: *fork*, *exec*, *wait*, *exit*, *brk*, *signal*.

Podsystem plików

- Zarządza plikami, przydzielając im miejsce, administruje dostępnym miejscem w systemach plików, steruje dostępem do plików i udostępnia dane użytkownikom.
- Dane udostępniane są z pliku za pomocą mechanizmu buforowania, który reguluje przepływ danych między jądrem i urządzeniami pamięci pomocniczej. Mechanizm buforowania współdziała z podprogramami obsługi blokowych urządzeń wejścia/wyjścia w celu zainicjowania transmisji danych do i z jądra. Blokowe urządzenia wejścia/wyjścia są urządzeniami o dostępie bezpośrednim.

Podsystem sterujący procesami

- Jest odpowiedzialny za:
 1. Szeregowanie procesów – realizuje politykę przydzielania CPU procesom. Ustala on kolejność wykonania procesów oraz przydziela im procesor do chwili zakończenia wykonywania lub wyczerpania kwantu czasu.
 2. Zarządzanie pamięcią operacyjną – przydzielanie pamięci dla procesów. Jeśli w danym momencie czasu nie ma dość pamięci fizycznej dla wszystkich procesów, to w zależności od stopnia deficytu jądro przesyła ich fragmenty lub całe procesy między obszarem wymiany tak, aby każdy z nich miał szansę się wykonać.
 3. Realizacja mechanizmów komunikacji międzyprocesowej – dostępne mechanizmy umożliwiają asynchroniczne sygnalizowanie pewnych zdarzeń oraz synchroniczną transmisję komunikatów między procesami.

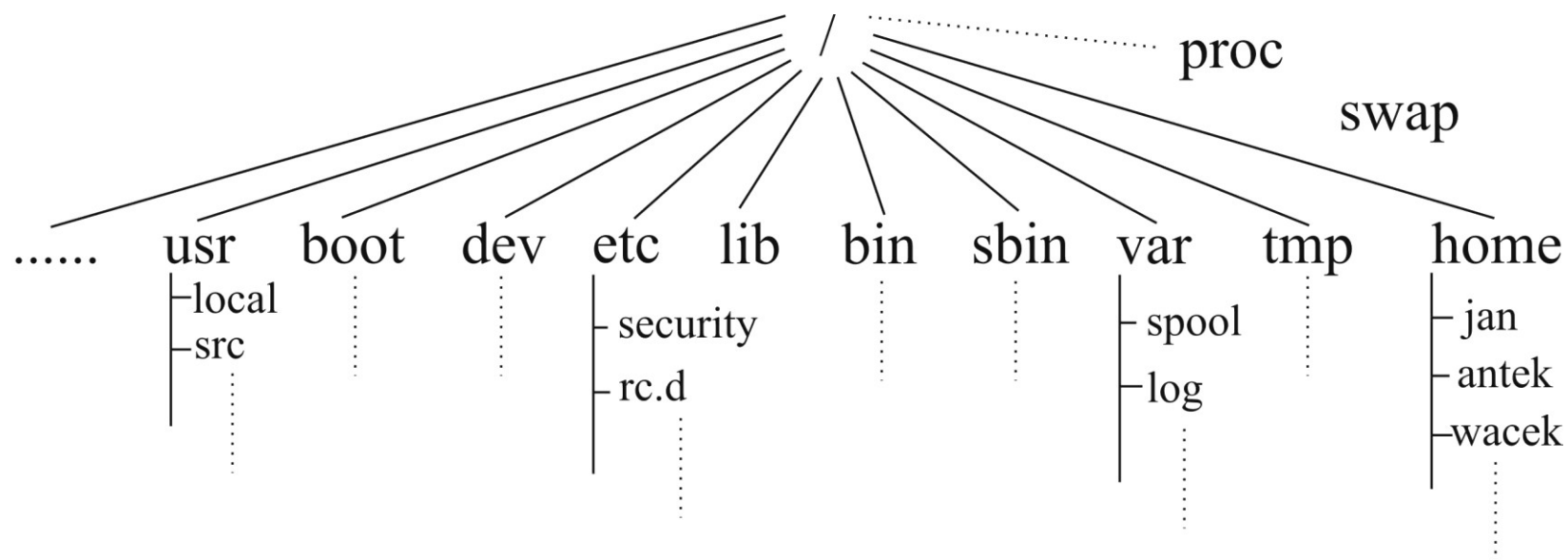
Współpraca podsystemów

- Ma miejsce w następujących sytuacjach:
 - Uruchamianie nowego procesu.
 - Konieczność wykorzystania pliku wymiany (pamięć wirtualna).

Drzewo katalogów, systemy plików



Przykład drzewa katalogów



Ścieżki dostępu (nazwy pliku)

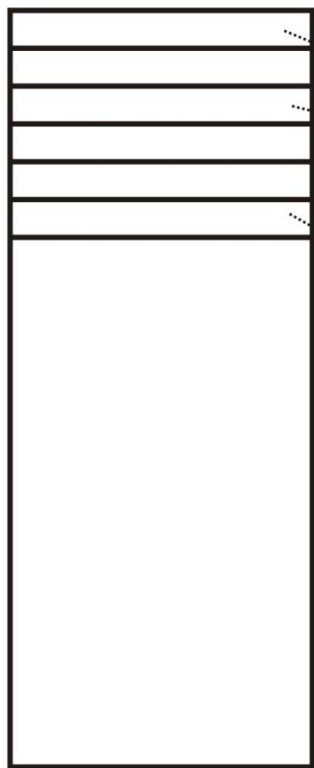
- Bezwzględna – zaczyna się od / i jeśli jest poprawna, pozwala dotrzeć do pliku niezależnie od bieżącego katalogu.
- Względna – zaczyna się od znaku innego niż / i mówi jak dotrzeć do pliku idąc od bieżącego katalogu. Stąd jej postać zależy od tegoż katalogu.

i-węzeł

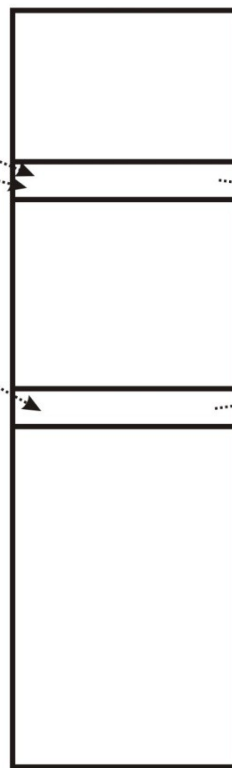
- To podstawowa struktura opisująca plik.
- Każdy plik jest opisywany przez jeden i-węzeł niezależnie od tego pod iloma nazwami plik ten występuje.
- i-węzeł zawiera informacje takie jak:
 - typ pliku,
 - identyfikator właściciela indywidualnego i grupowego,
 - prawa dostępu,
 - daty: dostępu do pliku, modyfikacji pliku, modyfikacji i-węzła,
 - tablicę alokacji pliku na dysku,
 - inne dane, zależne od rodzaju systemu plików.

Dostęp do pliku

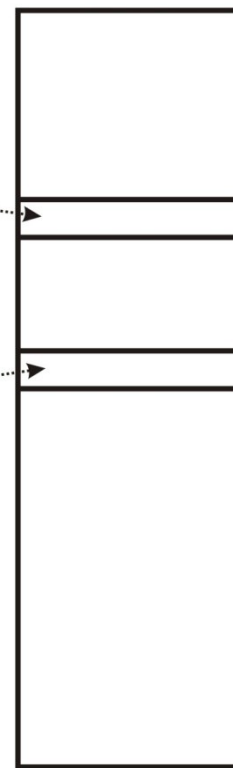
Tablica deskryptorów
plików użytkownika



Tablica
plików



Tablica
i-węzłów



Struktury pliku w jądrze

- Tablica deskryptorów plików użytkownika – jest przydzielana każdemu procesowi. Tu przechowuje się deskryptory wszystkich otwartych przez proces plików.
- Tablica plików – globalna struktura jądra. Tu pamięta się m.in. przesunięcie względem początku pliku bajtu od którego rozpocznie się kolejna operacja zapisu lub odczytu.
- Tablica i-węzłów – i-węzły są przechowywane w systemie plików, lecz podczas operacji na plikach jądro czyta je do tablicy i-węzłów przechowywanej w pamięci.

Przykładowy listing

boryczko@galaxy:~\$ ls -ali | less

total 817552

68804729	drwx--x---+	35	boryczko	kigrp	36864	2012-09-30 15:46	./
68231186	drwxr-xr-x	144	root	kigrp	4096	2012-10-04 10:55	../
2695184	-rw-----	1	boryczko	kigrp	676	2010-06-25 15:26	.Xauthority
2695185	-rw-r--r--	1	boryczko	kigrp	366	1992-03-03 11:16	.Xdefaults
2695186	-rw-----	1	boryczko	kigrp	0	1997-10-17 16:10	.addressbook
2695187	-rw-----	1	boryczko	kigrp	2285	2000-02-03 12:59	.addressbook.lu
2695188	-rw-----	1	boryczko	kigrp	414	1994-09-27 19:50	.aehistory
2695189	-rw-r--r--	1	boryczko	kigrp	4006	2012-10-06 23:27	.bash_history
85124650	drwx-----	2	boryczko	kigrp	4096	1994-09-08 20:01	.elm/
2695190	-rw-r--r--	1	boryczko	kigrp	110	1992-03-03 11:18	.filemgrrc
2695191	-rw-r-S---	1	boryczko	kigrp	0	1993-05-06 16:01	.gopherrc
2695192	-rw-r--r--	1	boryczko	kigrp	2514	2010-08-26 18:09	.history
2695193	-rw-----	1	boryczko	kigrp	43	2012-01-02 14:03	.lessht
85124651	drwxr-xr-x	3	boryczko	kigrp	4096	2008-10-01 14:23	.mc/
85124652	drwx-----	2	boryczko	kigrp	4096	2003-05-08 13:05	.netscape/
2695194	-rw-r--r--	1	boryczko	kigrp	0	1997-09-08 17:29	news_time
2695195	lrw-r--r--	1	boryczko	kigrp	167	2004-12-17 15:04	newsr->/usr/newsr
2695196	-rwxr-xr-x	1	boryczko	kigrp	795	1997-10-27 15:30	openwin-init

Typy plików

- Ogólne
 - - - plik regularny,
 - d - katalog,
 - l – dowiązanie (link) symboliczne,
 - b – plik reprezentujący urządzenie blokowe (/dev),
 - c - plik reprezentujący urządzenie znakowe (/dev),
 - s – socket (komunikacja międzyprocesowa),
 - p – pipe (komunikacja międzyprocesowa).
- Specyficzne dla implementacji
 - D – door (Solaris),

Dowiązanie symboliczne

- Nowy plik specjalnego typu, jego zawartość to ścieżka dostępu do oryginału.
- Usuwając oryginał tracimy informacje.
- Możliwy do utworzenia jeśli nie istnieje oryginał.
- Może wiązać pliki znajdujące się w różnych systemach plików.
- Może wskazywać na katalogi.

Dowiązanie twarde

- Kolejna nazwa na istniejący plik, zapisana najczęściej we wpisie. Nie ma możliwości stwierdzenia co jest oryginałem.
- Informacja dostępna do momentu usunięcia ostatniego dowiązania.
- Nie można utworzyć dowiązania twardego do nieistniejącego pliku.
- Nie można utworzyć dowiązania twardego do pliku znajdującego się w innym systemie pliku.
- Nie można utworzyć dowiązania twardego do katalogu.
- Nie wszystkie systemy plików go implementują.

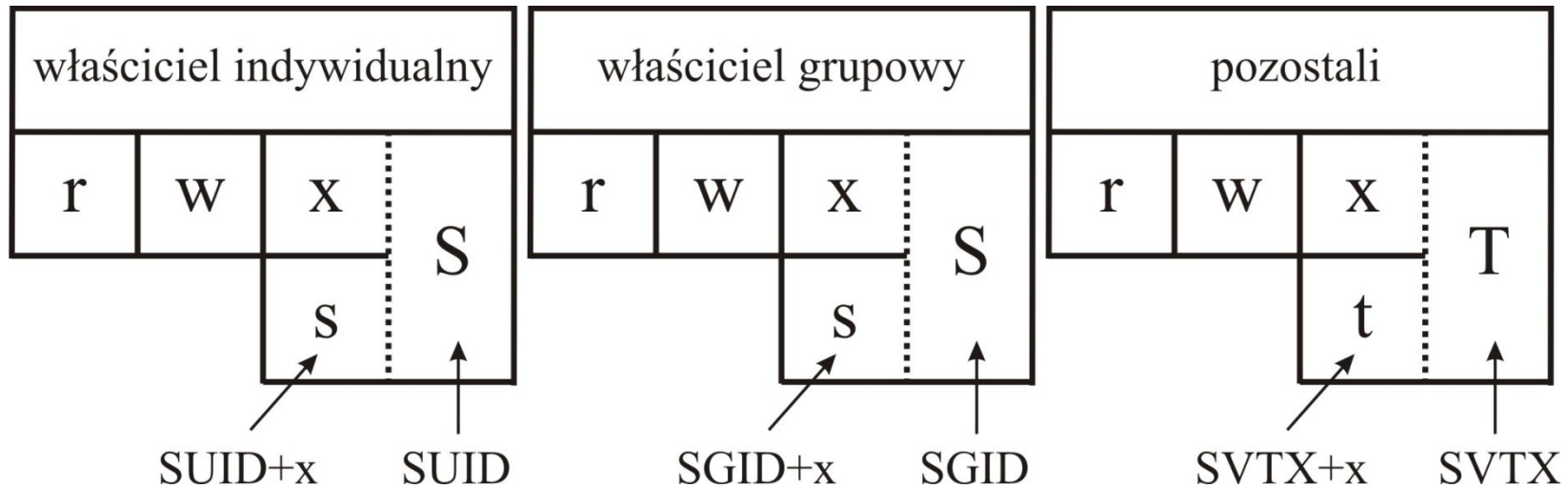
Modele praw dostępu - regulacje

- Sprawy bezpieczeństwa systemów komputerowych regulował dokument *Trusted Computer System Evaluation Criteria* (TCSEC) – tzw. *Pomarańczowa książka*. Powstał w roku 1983, zmodyfikowany w 1985.
- Od roku 2005 obowiązuje dokument *Common Criteria for Information Technology Security Evaluation*.
- Dokumenty zawierają m.in. Klasyfikację kontroli dostępu.

Modele praw dostępu

- DAC (Discretionary Access Control) – przekazuje uprawnienia do zarządzania danym obiektem w ręce jego właściciela.
- MAC (Mandatory Access Control) – polityki bezpieczeństwa w systemie są kontrolowane centralnie przez administratora bezpieczeństwa (SELinux, TrustedBSD).
- RBAC (Role Based Access Control) – polega na umożliwieniu zarządzania jedynie autoryzowanym użytkownikom. Praktycznie polega to na przypisaniu możliwości wykonania danej operacji zdefiniowanym w systemie rolom.

Prawa dostępu do plików (DAC)



Znaczenie praw dostępu – plik regularny

- r – jeśli ustawiony, umożliwia odczyt zawartości pliku (polecenia cat, more, less, vi, emacs, ...).
- w – jeśli ustawiony, umożliwia modyfikację zawartości pliku (edytorem lub >>).
- x – jeśli ustawiony, interpreter poleceń podejmie próbę uruchomienia (jako skryptu lub programu).

Znaczenie praw dostępu – katalog

- r – jeśli ustawiony, umożliwia odczyt zawartości katalogu (polecenie ls).
- w – jeśli ustawiony, umożliwia modyfikację zawartości katalogu (tworzenie nowych plików lub usuwanie istniejących).
- x – jeśli ustawiony, umożliwia utworzenie danego katalogu bieżącym (wykonanie polecenia cd z argumentem będącym nazwą katalogu zakończy się sukcesem).

Rozszerzone prawa dostępu

- SUID (s lub S na prawie wykonywania właściciela indywidualnego) – **dla pliku regularnego**, utworzony z niego proces wykonuje się z efektywnym identyfikatorem właściciela pliku, a nie użytkownika, który go uruchomił (np. passwd, chfn, chsh).
- SGID (s lub S na prawie wykonywania właściciela grupowego) – dla **pliku regularnego**, uruchomiony proces posiada efektywny identyfikator efektywny właściciela grupowego pliku, a nie podstawowej grupy użytkownika, który go uruchomił; **dla katalogu**, właścicielem grupowym każdego utworzonego w nim pliku jest właściciel grupy katalogu, a nie grupa podstawowa użytkownika, który plik utworzył.

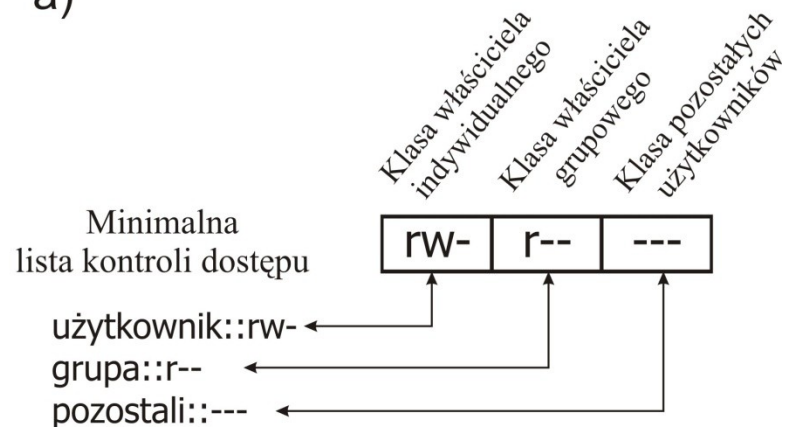
Rozszerzone prawa dostępu - cd

- SVTX (t lub T na prawie wykonania dla pozostałych użytkowników) – **dla katalogu**, plik z katalogu może usunąć tylko jego właściciel indywidualny (przykład */tmp*).
- Specyficzne dla dystrybucji – np. SVTX w linuksach dla **pliku regularnego** oznacza, iż segmenty utworzonego procesu, po zakończeniu procesu nie będą usuwane z pamięci – jeśli to możliwe.

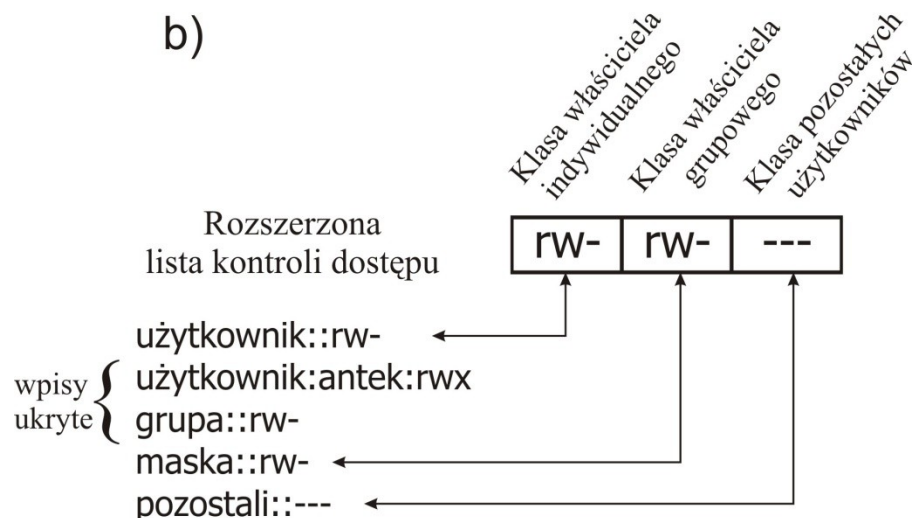
Listy kontroli dostępu

Access Control List (ACL)

a)



b)



Właściciele pliku

- W i-węzłach opisujących plik w systemie plików przechowywane są numery identyfikacyjne użytkownika będącego właścicielem indywidualnym i grupy będącej właścicielem grupowym.
- W momencie tworzenia pliku właścicielem indywidualnym staje się użytkownik go tworzący, a grupowym jego grupa podstawowa (wyjątek w katalogach z ustawionym prawem SGID).
- Zmiany właściciela indywidualnego może dokonać jedynie administrator systemu.
- Zmiany właściciela grupowego może dokonać właściciel indywidualny, pod warunkiem, że nowym właścicielem grupowym będzie grupa, do której on należy.
- Jeśli zamiast nazw właścicieli pojawią się na listingu numery ID, to jest to wynikiem błędnej konfiguracji w plikach */etc/passwd* lub */etc/group*.

Daty przechowywane w i-węźle

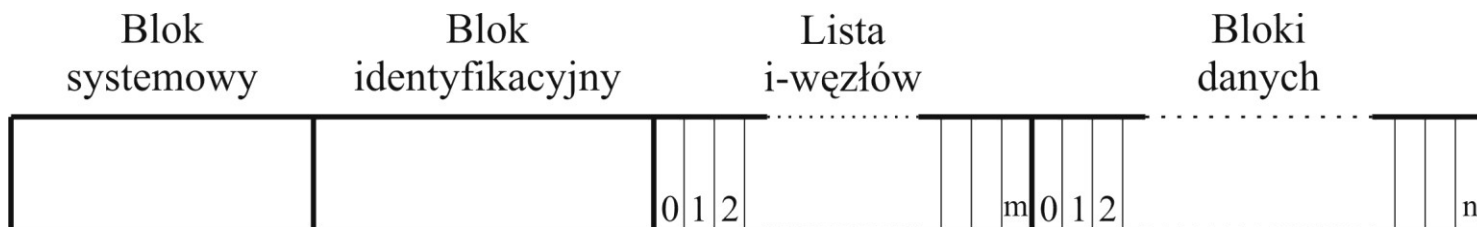
- Data ostatniej modyfikacji zawartości pliku – zmieniła się jego zawartość. W tej sytuacji zmiane ulega również data ostatniego dostępu do pliku.
- Data ostatniego dostępu do pliku – plik był czytany, ale jego zawartość nie uległa zmianie.
- Data ostatniej modyfikacji i-węzła – zmieniono wartości atrybutów opisujących plik.

Nazwa pliku

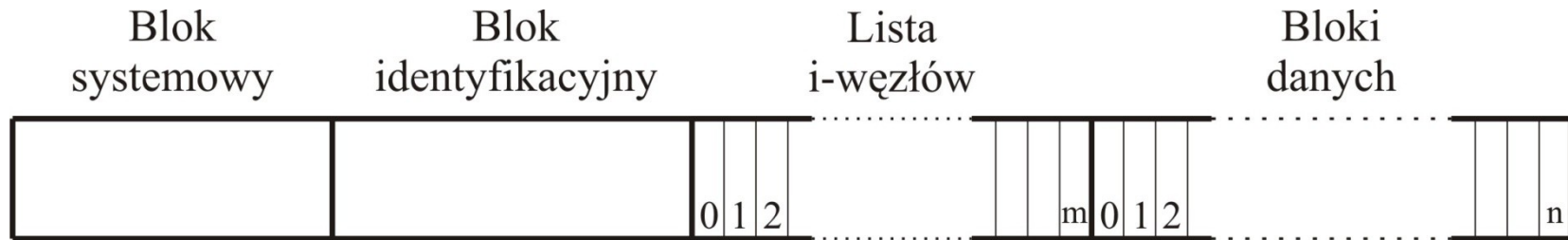
- Duże, małe litery alfabetu angielskiego. Uwaga na polskie – nie każdy system plików jest dostatecznie „spolszczony”.
- Cyfry.
- Niektóre znaki specjalne.
- Nazwy wielocłonowe łączymy znakami „-” „_” lub „.” .
Raczej nie spacja, ze względu na składnię wiersza polecenia (ls moj\ katalog, ls "moj katalog").
- Kropka jest normalnym znakiem. Nie oddziela nazwy od rozszerzenia, bo tu nie ma rozszerzenia.
- Nazwa . i .. są zarezerwowane i nie można ich używać.
- Nazwa zaczynające się od kropki powoduje, że plik jest ukryty (nie ma atrybutu). Stosowana w nazwach plików konfiguracyjnych.

Definicje systemu plików

- Format przechowywania informacji. Zawiera *metadane* (struktury zawierające opis formatu przechowywania danych) i *dane*.
- System plików składa się z ciągu bloków logicznych, zawierających 512, 1024, 2048 lub dowolną inną wielokrotność 512 bajtów.
- System plików to sterownik w jądrze systemu operacyjnego oraz grupa programów narzędziowych dla jego założenia i administrowania nim.

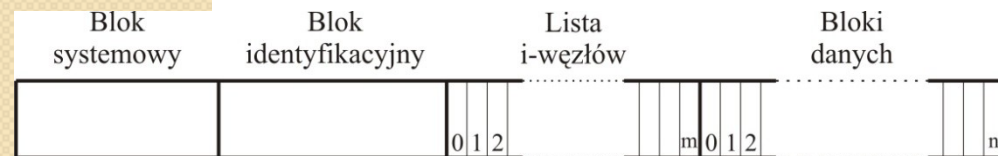


Organizacja systemu plików (s5)



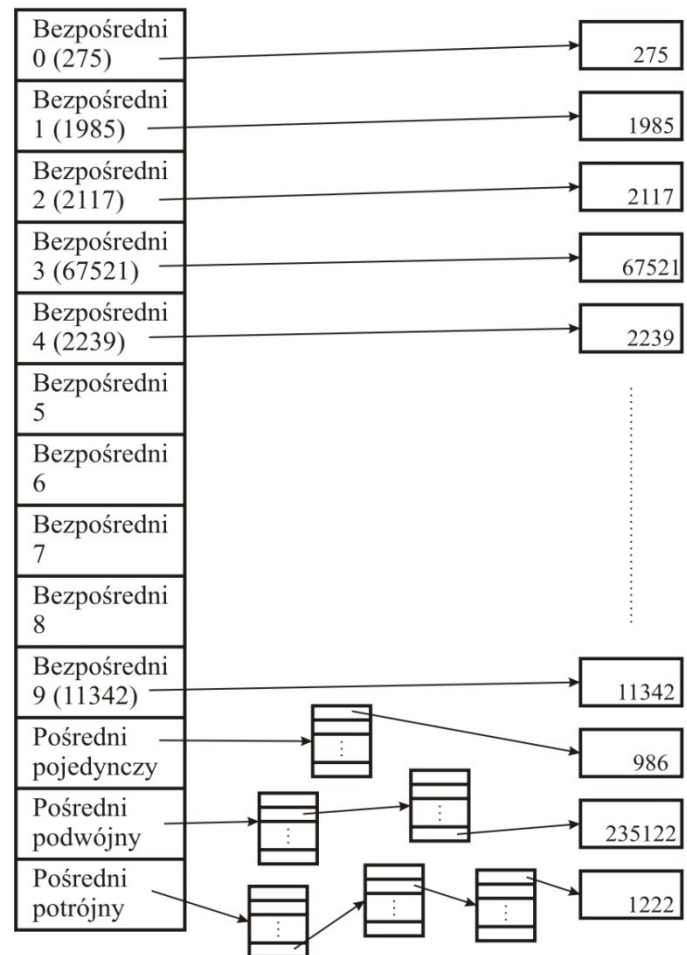
- Blok systemowy – zajmuje zazwyczaj pierwszy sektor dysku i zawiera program ładujący do inicjalizacji systemu operacyjnego.
- Blok identyfikacyjny (superblok) – zawiera informacje o rozmiarze systemu plików, liczbie możliwych do utworzenia w nim plików, lista wolnych i-węzłów, lista wolnych bloków danych Jego uszkodzenie powoduje, że system plików nie może zostać zamontowany.
- Lista i-węzłów – lista ponumerowanych struktur opisujących pliki.
- Lista bloków danych – lista ponumerowanych bloków przechowujących dane. Rozmiar ustalany w momencie zakładania systemu plików (od 1kb do 64kb). Nie wszystkie rozmiary są obsługiwane przez jądro.

Rozmieszczenie pliku w systemie plików (s5)



I-węzeł
tablica "alokacji"

Bloki danych



Idea modyfikacji

- Efektywność operacji wejścia/wyjścia systemu plików jest tym większa, im mniej ruchów wykonuje głowica przy odczycie/zapisie danych. Stąd:
- Najprościej zwiększyć rozmiar bloku danych.
- Należy dokonać grupowania i-węzłów i opisywanych przez nie dloków danych (ffs, xfs, ext, ...).
- Należy alokować ciągłe obszary bloków danych (xfs, ...).

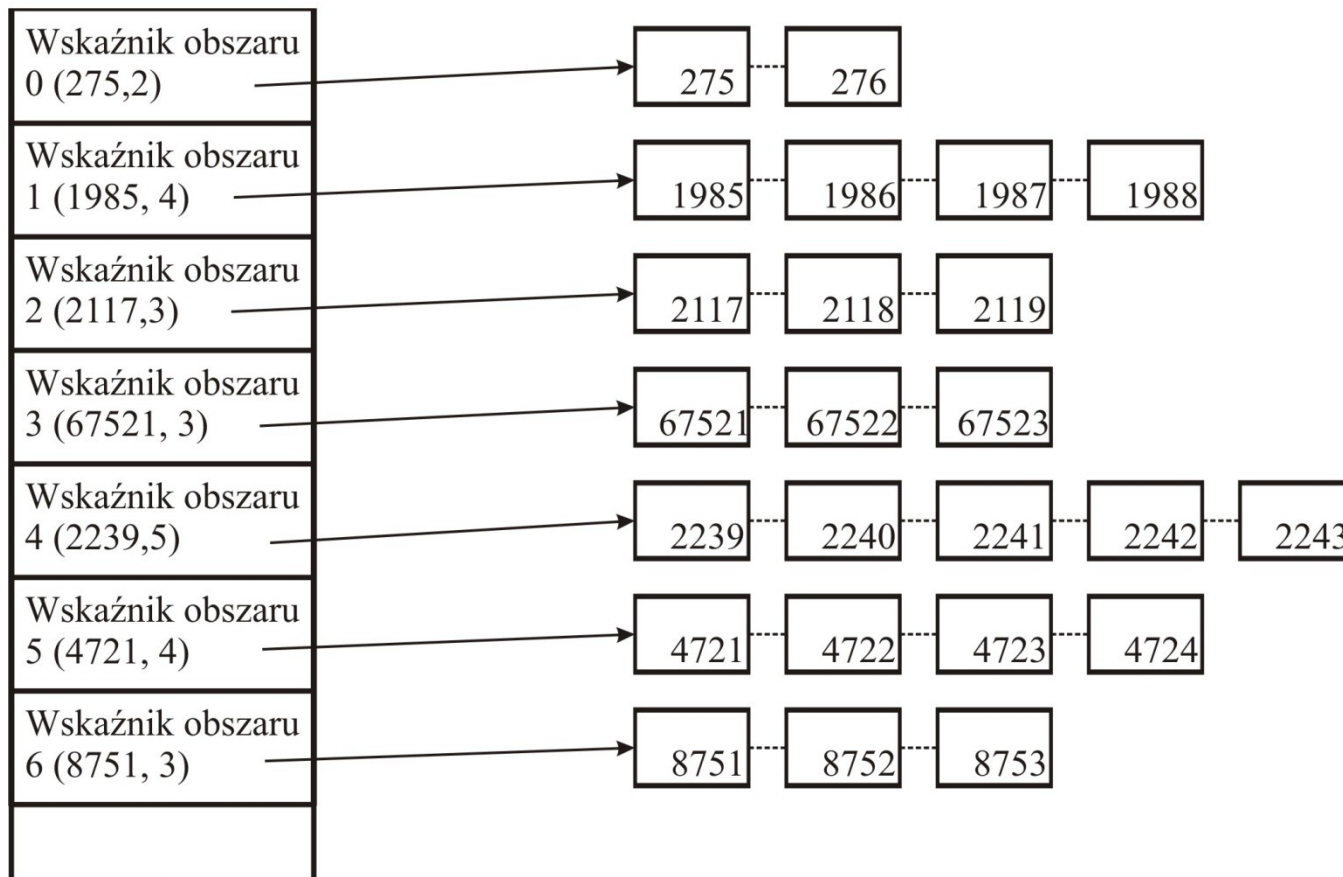
Fast File System (FFS) - BSD

Rekord startowy	Grupa 0			Grupa 1	Grupa 2	Grupa N
	Blok identyfikacyjny	i-węzły	Bloki danych				

Alokacja ciągłego obszaru bloków danych

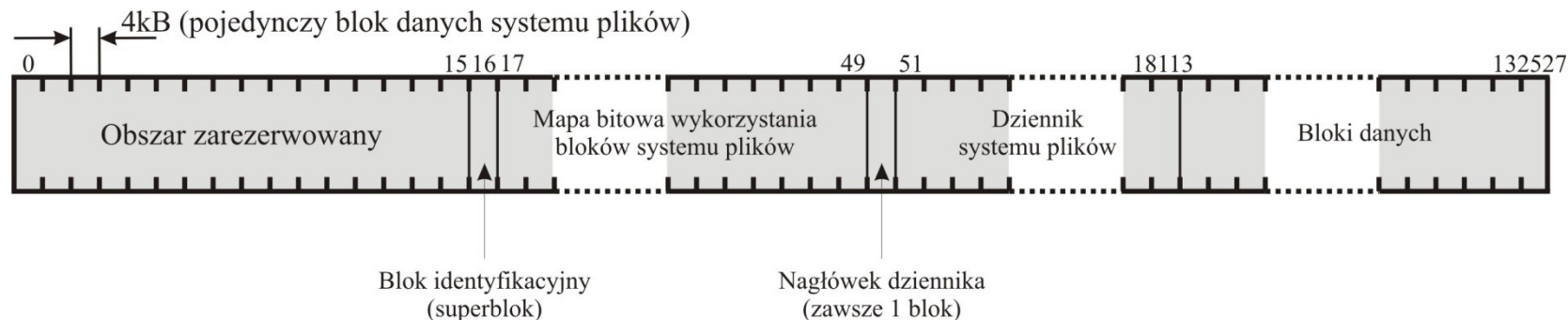
I-węzeł
lista “alokacji”

Bloki danych



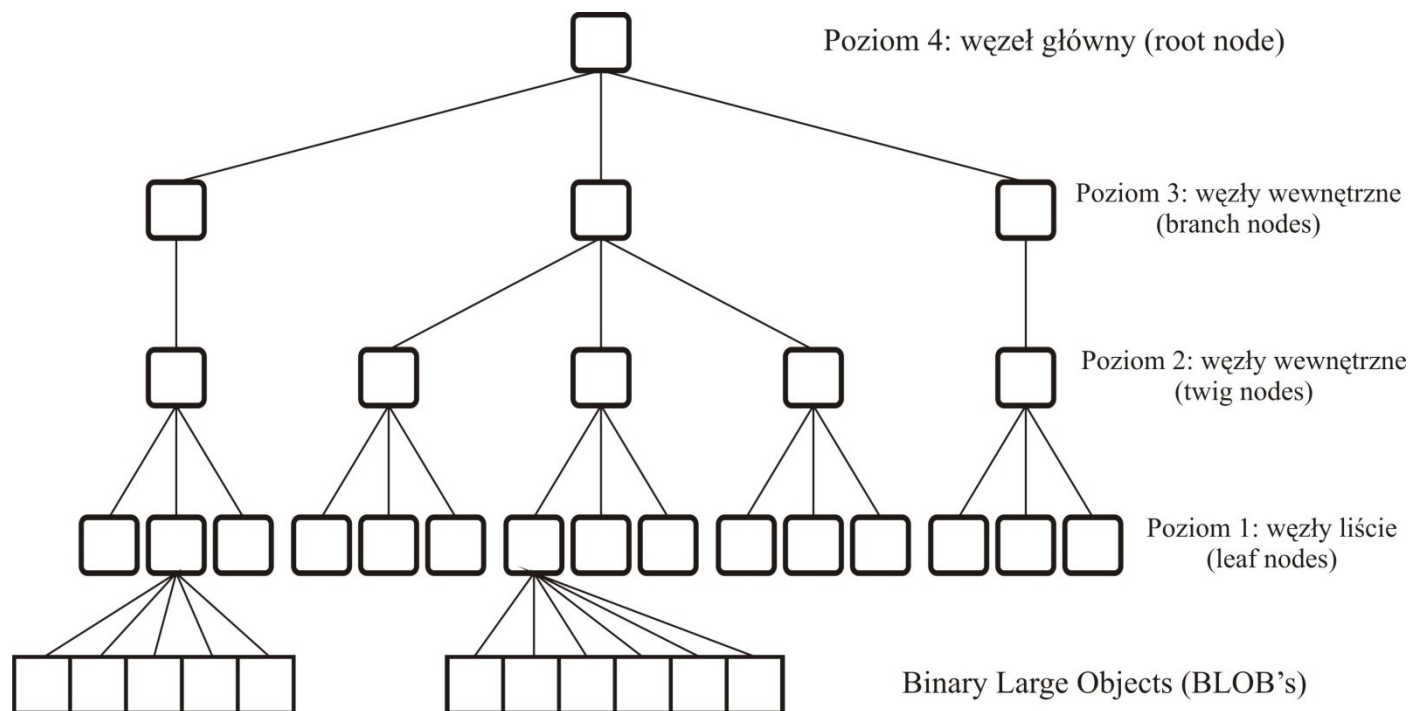
B, B+, S+ drzewa

- Idea zaproponowana przez Hansa Reiser'a w systemach plików Reiserfs 3.6 oraz Reiser4.
- System plików składa się z bloków o rozmiarze 512-65536 B (4096 B).



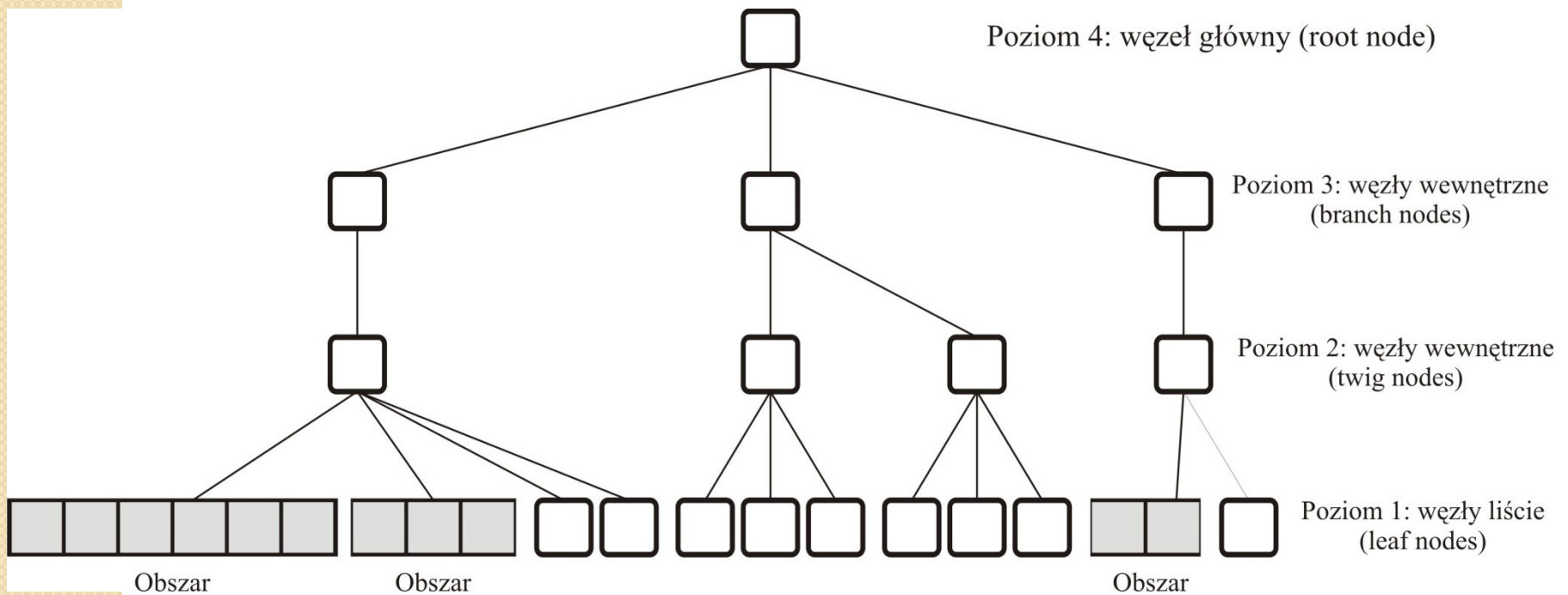
ReiserFS 3.6

- Zbalansowane B+ drzewo przechowuje informacje.
- Drzewo tworzą węzły odpowiadające blokom dyskowym.
- Informacja przechowywana jest we wpisach zajmujących jeden lub więcej węzłów.



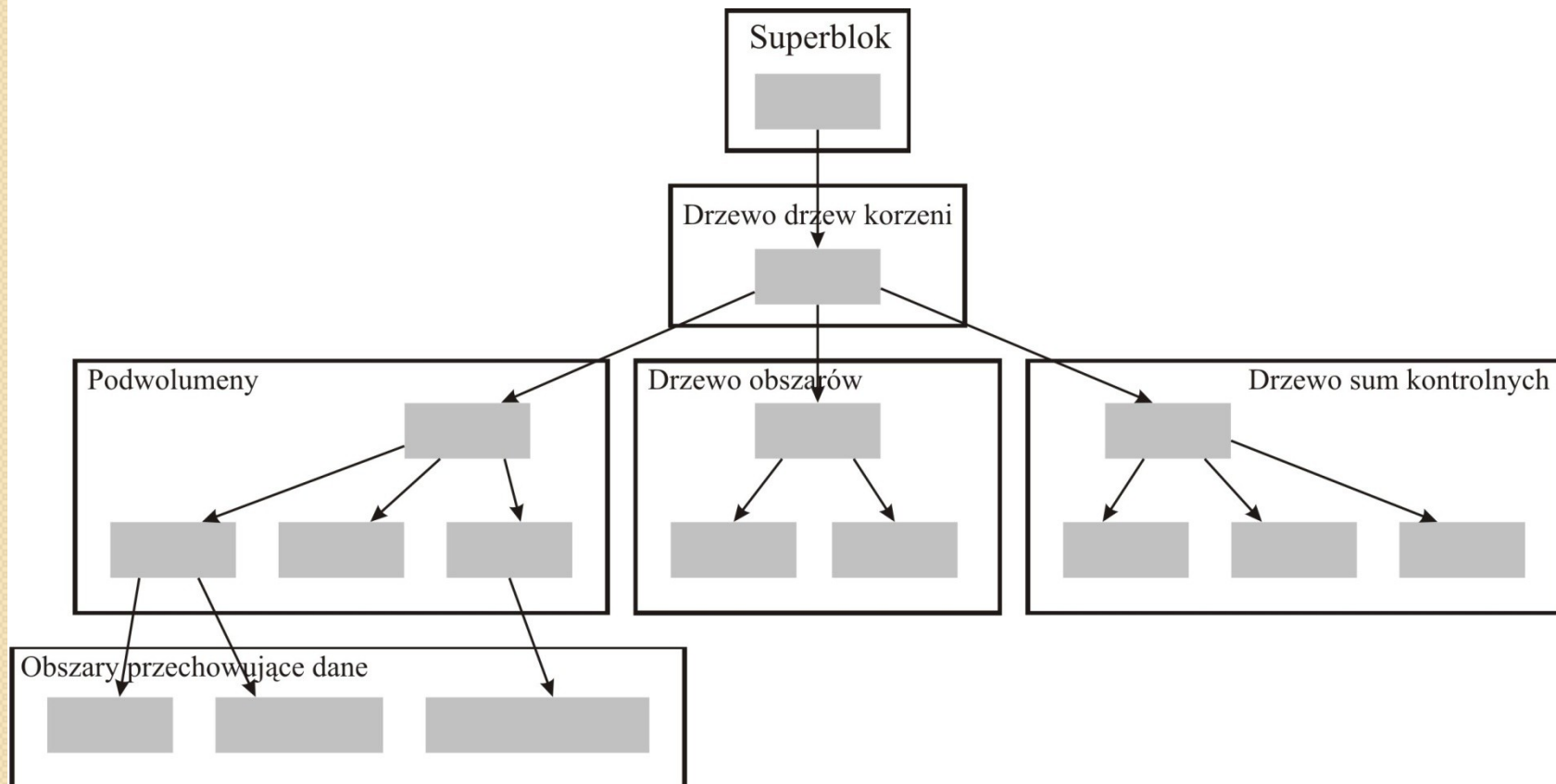
Reiser4

- Wprowadzono obszary.



Btrfs

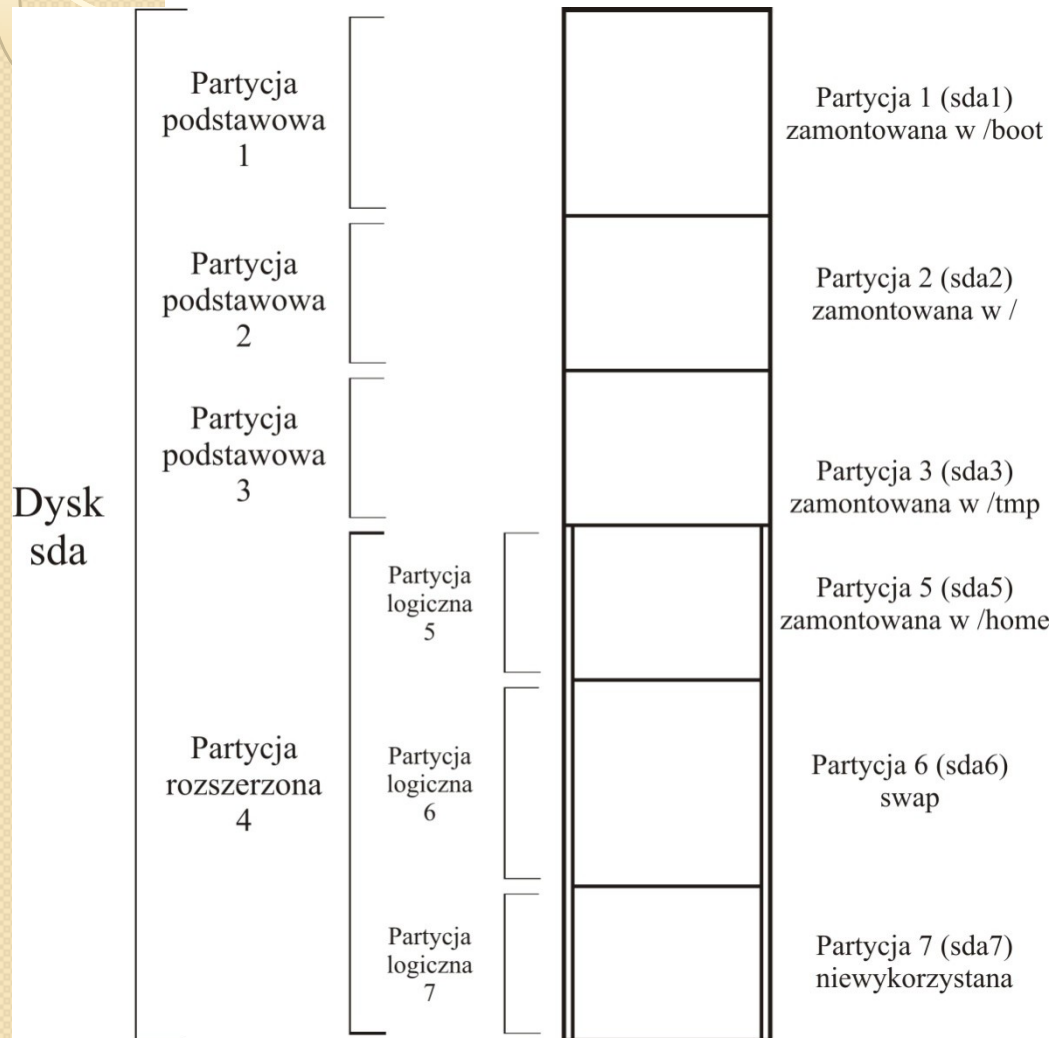
- Cała informacja przechowywana jest z wykorzystaniem B-drzew.



Zakładanie systemu plików – krok 1 utworzenie partycji

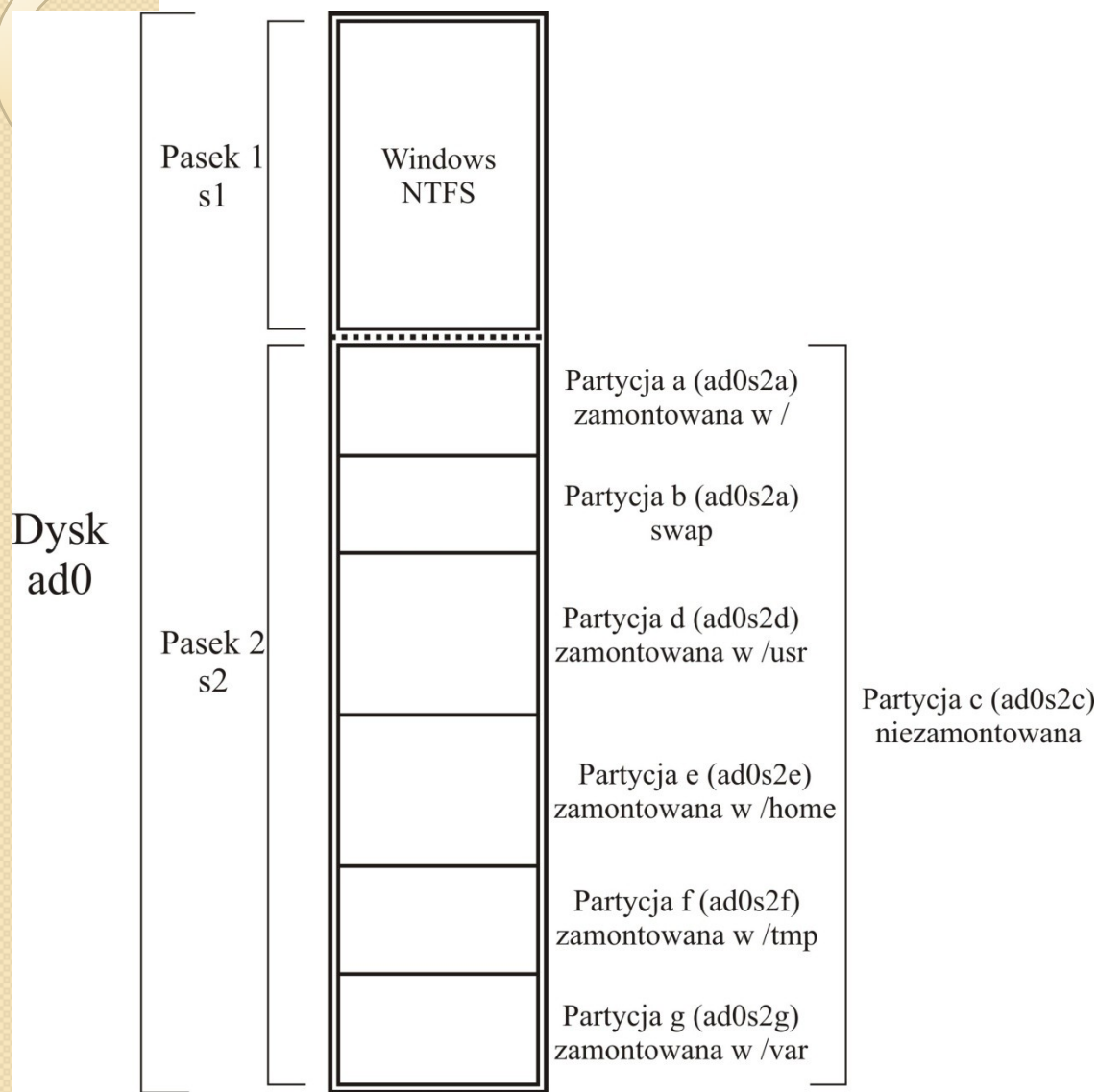
- Ustalenie dostępnych urządzeń dyskowych i stopnia ich wykorzystania.
 - W systemach linuksowych:
 - Plik */proc/partitions*
 - Program narzędziowy: *fdisk -l*
 - Dowolny inny, np. *parted*
 - Polecenie *lsblk*
 - W rodzinie BSD:
 - Program *sysinstall*, tryb *Custom*, punkt 3 *Partitions*.
- „Wykrojenie” partycji o zadanym rozmiarze.

Organizacja dysku - linux



- Nazwy urządzeń:
 - hd – dyski z interfejsem IDE do wersji 9 FC. a master na 1-szej magistrali, b slave na 1-szej magistrali, itd.
 - sd – dyski z interfejsem SCSI. Od FC 9. Wszystkie dyski „widziane” są jako SCSI. Numeracja od „a” począwszy.
 - scd – napędy SCSI CDROM.
 - fd – napęd dyskietek.
 - sd – napęd tasiemki z interfejsem SCSI.
 - md – urządzenia RAID.
- Numer kolejny partycji. Partycje 0-3 – podstawowe, 4 – rozszerzona, 5 - logiczne.

Organizacja dysku - BSD



Typ urządzenia:

- `ad` lub `ada` – dysk typu IDE, SATA.
- `da` – dysk typu SCSI .
- `acd` lub `cd` – napęd CDROM IDE, SATA.
- `fd` – napęd dyskietek.
-

Numer urządzenia danego typu zainstalowanego w systemie. Numeracja od 0.

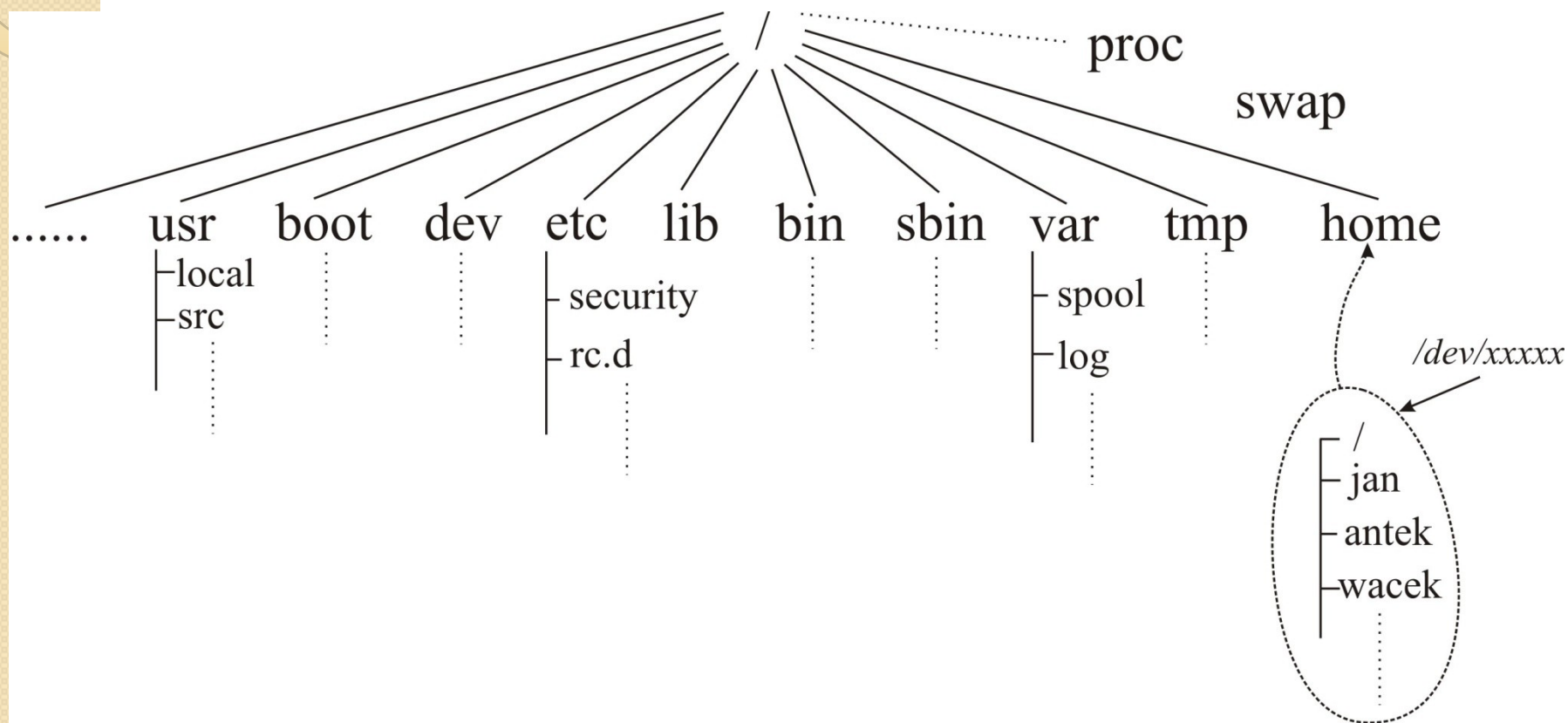
Numer paska na dysku poprzedzony literą `s`. Numeracja od 1.

Partycja w obrębie paska. Numeracja od litery „a”.

Zakładanie systemu plików – krok 2 formatowanie

- Służy do tego polecenie *mkfs*, które należy uruchomić z opcją *-t* i wartością będącą nazwą systemu plików. Musimy podać nazwę pliku reprezentującego urządzenie, na którym założony zostanie dany system plików. Np.: *mkfs -t ext4 /dev/sda5* Spowoduje to uruchomienie polecenia *mkfs.ext4* i utworzenie systemu plików z domyślnymi wartościami atrybutów.
- Dla nadania wartości specyficznym dla danego systemu plików atrybutów należy uruchomić wprost polecenie zakładające dany system plików. Np.: *mkfs.ext4 -i 8192 /dev/sda5*
- Ten drugi sposób jest wykorzystywany przy zakładaniu dedykowanych danemu zastosowaniu systemów plików.

Montowanie systemów plików



Zasady montowania

- System plików montuje się w katalogu.
- Katalog nie musi być pusty.
- Zawartość katalogu jest jedynie „zasłonięta” przez zamontowany system plików. Po jego odmontowaniu staje się dostępna.
- Montowanie czyni dostępną zawartość systemu plików. Po odmontowaniu nie jest ona tracona – staje się dostępna po ponownym zamontowaniu.

Polecenie mount

- Składnia:
mount [opcja] co gdzie
- Opcje dzielą się na:
 - ogólne – montują każdy system plików,
 - dedykowane – najczęściej służą wymuszeniu specjalnego działania wybranego typu systemu plików.
- Zazwyczaj stosuje się montowanie z domyślnym zestawem opcji (no_suid).
- Ale wprowadzenie systemu kontyngentów wymaga użycia dodatkowych opcji.

Plik /etc/fstab

- Plik wg którego następuje montowanie systemów plików podczas inicjalizacji systemu oraz użycia polecenia *mount* z wiersza polecenia.
- Plik tekstowy. Jedna linijka opisuje jeden montowany system plików. Składa się z 6-ciu kolumn oddzielonych co najmniej jednym białym znakiem.

```
#
# /etc/fstab
# Created by anaconda on Mon Jul 11 14:38:36 2011
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid( for more info
#
UUID=03b72b1f-83df-4ec1-b4b8-44af38a08a55 /          ext4  defaults    1 1
UUID=0db1bd9c-6bde-40e3-b464-bc15a4fa96ef /var        ext4  defaults    1 2
UUID=334c1b92-216e-4939-83c1-d336aceda5c3 swap        swap  defaults    0 0
tmpfs          /dev/shm          tmpfs  defaults    0 0
devpts         /dev/pts          devpts gid=5,mode=620 0 0
sysfs          /sys              sysfs  defaults    0 0
proc           /proc             proc   defaults    0 0
nfs.dydaktyka.agh.edu.pl:/ /nfs      nfs4   ro,async,fsc
```


Plik /etc/fstab - cd

- Kolumna 1 – ścieżka dostępu do pliku reprezentującego urządzenie blokowe, na którym znajduje się dany system plików **lub** etykieta systemu plików **lub** UUID (polecenie *uuidgen*).
- Kolumna 2 – ścieżka dostępu do katalogu, w którym dany system plików ma zostać zamontowany.
- Kolumna 3 – identyfikator typu systemu plików (*ext3*, *ext4*, *jfs*, *xfs*, ...).

Plik /etc/fstab - cd

- Kolumna 4 – opcje montowania w postaci listy identyfikatorów opcji oddzielonych przecinkiem (np. *defaults,usrquota,grpquota*).
- Kolumna 5 – informacja dla polecenia *dump*. Wartość równa 0 oznacza, że *dump* nie będzie wykonywał kopii zapasowej danego systemu plików. Wartość różna od 0 oznacza minimalny poziom zrzutu, przy którym dojdzie do kopiowania tego systemu plików.
- Kolumna 6 – Informacja dla polecenia *fsck*. Wartość 0 oznacza, że dany system nie musi podlegać weryfikacji spójności. Wartość różna od 0 oznacza kolejność sprawdzania spójności (istotna kolejność: główny system plików wartość 1, a pozostałe wartości wynikające z kolejności montowania).

Odmontowanie systemu plików

- Może go dokonać użytkownik *root* lub inny, którego nazwa lub UID pojawiły się w opcjach montowania lub pliku */etc/fstab*.
- Polecenie *umount* jako argumentu wymaga podania punktu montowania lub nazwy pliku reprezentującego urządzenie.
- Wykonanie polecenia *umount* zakończy się niepowodzeniem jeśli odmontowywany system plików zawiera choć jeden otwarty plik lub katalog będący bieżącym dla jednego procesu.
- Znajdowanie „przeszkadzających” procesów poleceniami *fuser* lub *lsof*.

System kontyngentów (quota)

- Funkcjonalność systemu plików – możliwy do zdefiniowania dla w obrębie systemu plików.
- Wprowadza ograniczenie na bloki danych oraz i-węzły.
- Definiowany dla użytkowników indywidualnych i grup użytkowników.
- Wprowadzone limity mogą być:
 - twarde – ustalone wartości nie mogą zostać przekroczone.
 - miękkie – możliwe do przekroczenia, ale na określony czas. Po przekroczeniu czasu, mimo nie osiągnięcia limitu twardego zasoby nie będą przydzielane.

Definiowanie systemu kontyngentów (linux)

- Opcje dla montowania z uwzględnieniem systemu quota nie należą do domyślnych stąd konieczność ich dodania w pliku */etc/fstab*.
- Przemontowanie systemu plików, np: *mount -o remount /home*
- Stworzenie plików baz danych z limitami, np.: *quotacheck -cug /home*
- Uruchomienie systemu: *quotaon -a*
- Nałożenie ograniczeń: *edquota*
- Edycja ograniczeń: *quota -v*
- Wyłączanie działania systemu: *quotaoff -a*

Definiowanie systemu kontyngentów (BSD)

- W domyślnej konfiguracji jądro nie obsługuje systemu kontyngentów. Stąd konieczność jego rekompilacji z opcją QUOTA.
- Reszta wg identycznego schematu.

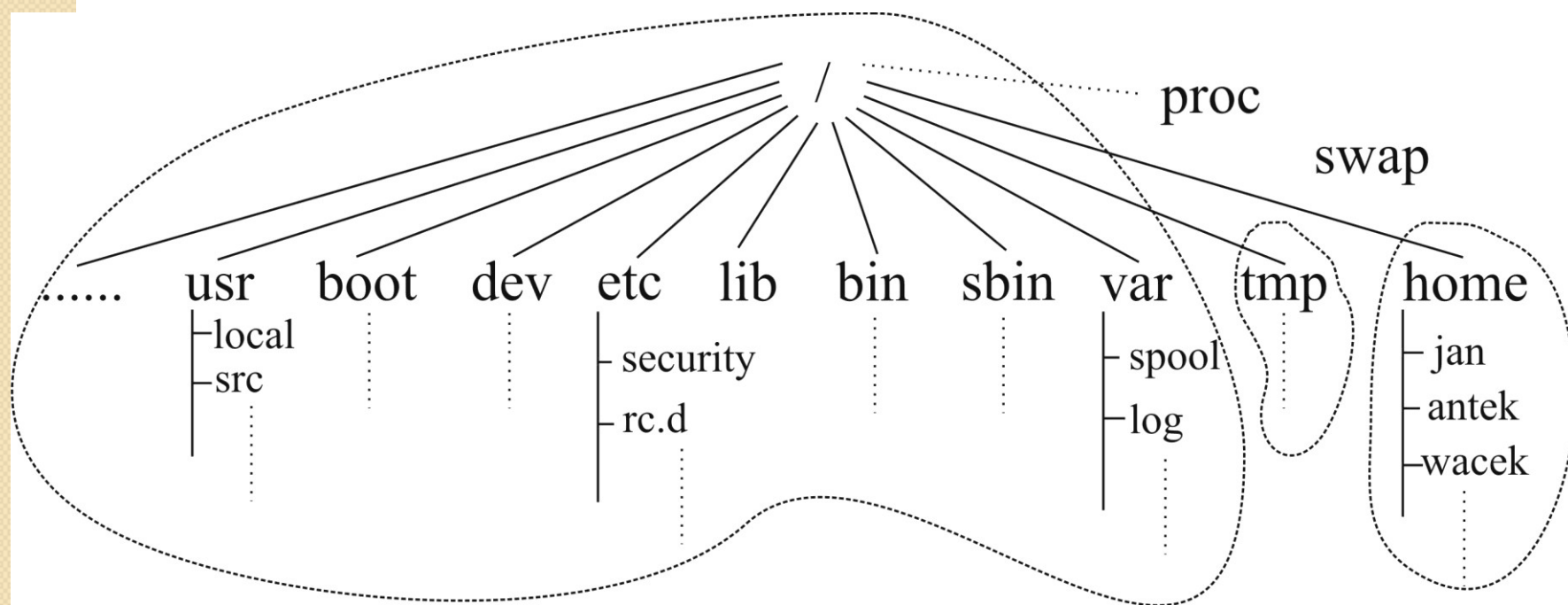
Dziennikowane systemy plików

- Dziennik służy przechowywaniu informacji o zmianach danych w systemie plików. Są one rejestrowane w dzienniku przed ich fizyczną realizacją.
- Spójność dziennikowanego systemu plików może zostać przywrócona w oparciu o zawartość dziennika.
- Mechanizmy dziennikowania mogą dotyczyć:
 - Zmian związanych z wykonaniem każdej operacji zapisu wprowadzonych jedynie do metadanych systemu plików.
 - Zmian związanych z wykonaniem każdej operacji zapisu wprowadzonych do danych i metadanych systemu plików.

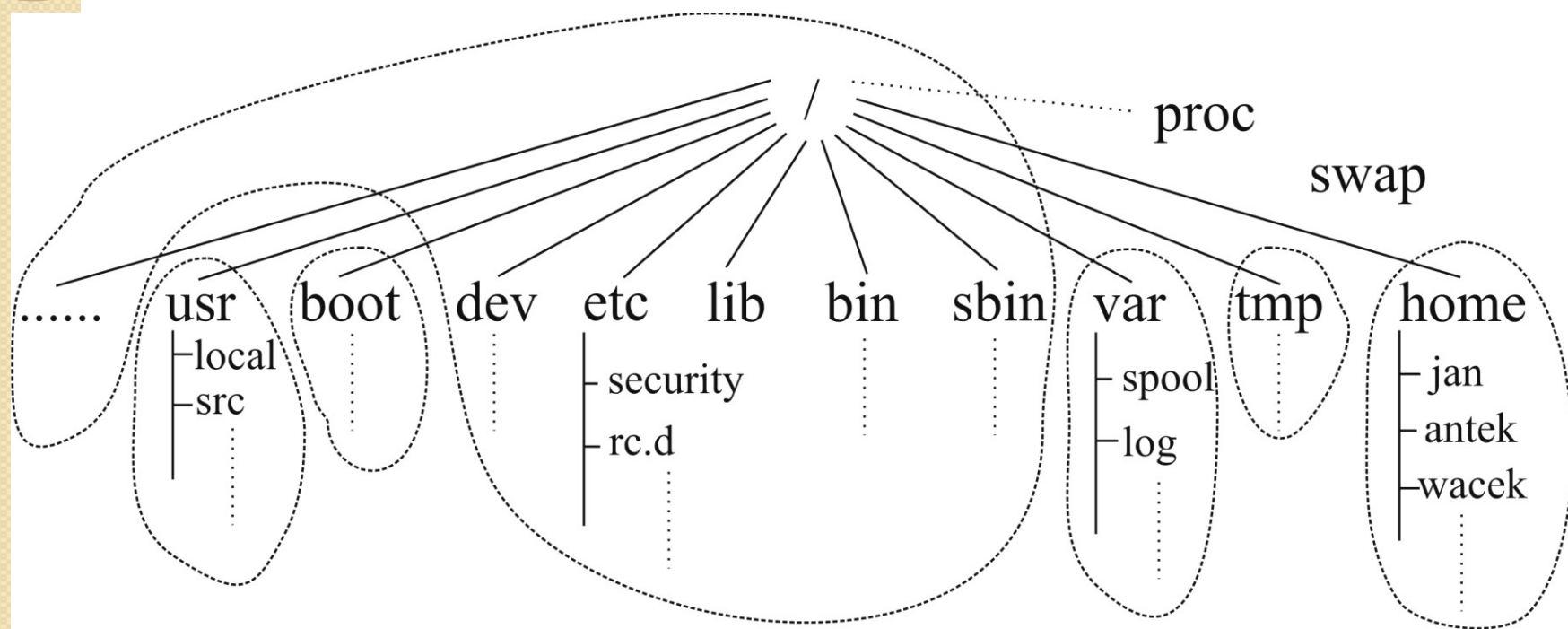
Dziennikowane systemy plików - cd

- W dzienniku mogą być zapisywane:
 - Całe transakcje i usuwane operacje już wykonane.
 - Zapisywanie operacji już wykonanych.
- Położenie dziennika:
 - Plik regularny w obrębie systemu plików.
 - Specjalny obszar systemu plików, niedostępny dla programów użytkowych.
 - Obszar poza systemem plików, przydzielony w specjalnym obszarze dysku (najlepiej innego niż zajmowany przez system plików).

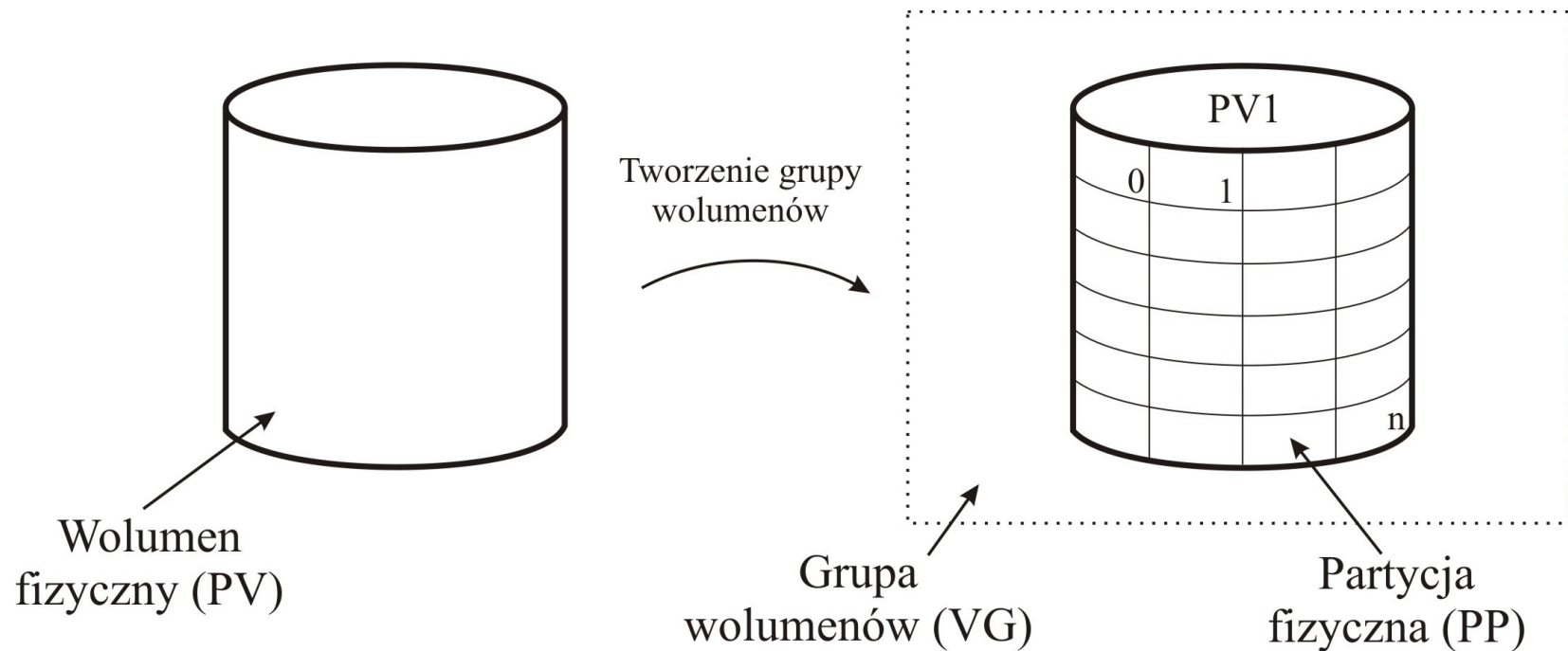
Drzewo katalogów - laptop



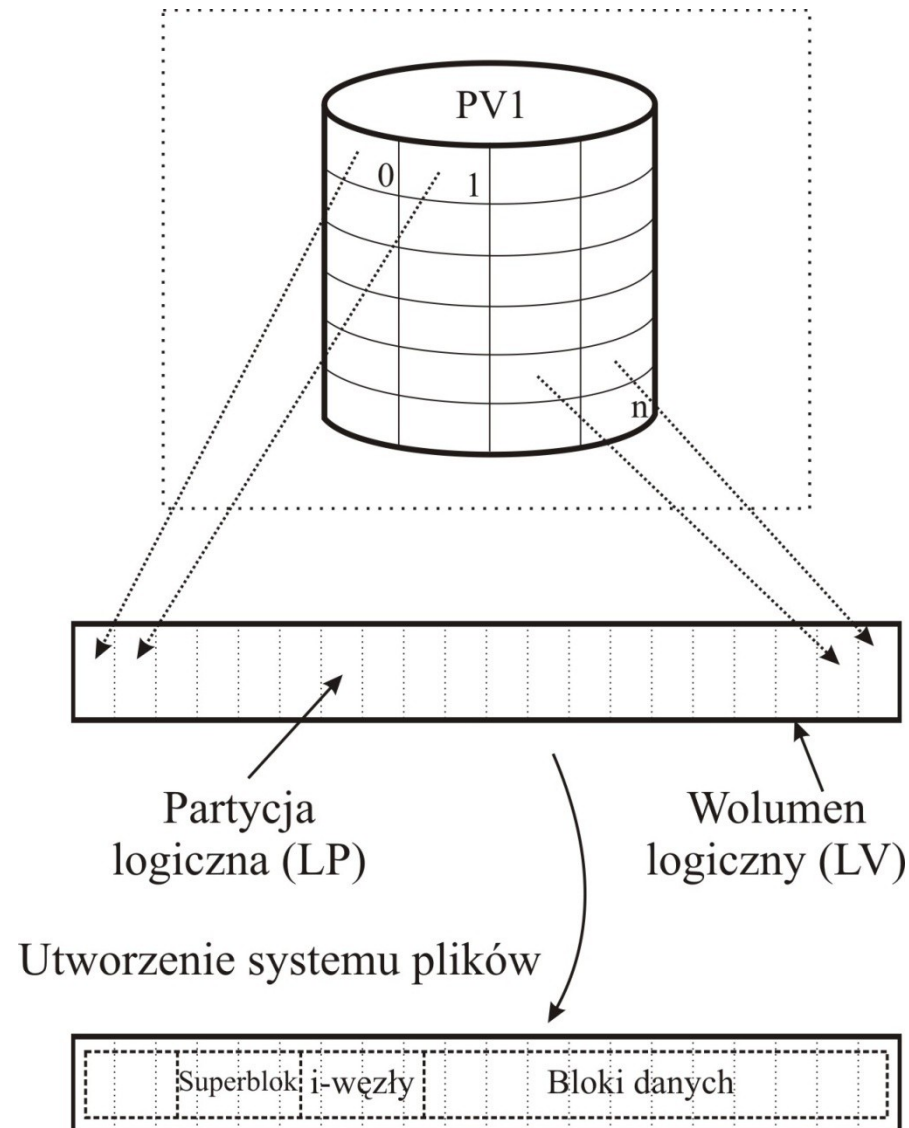
Drzewo katalogów - serwer



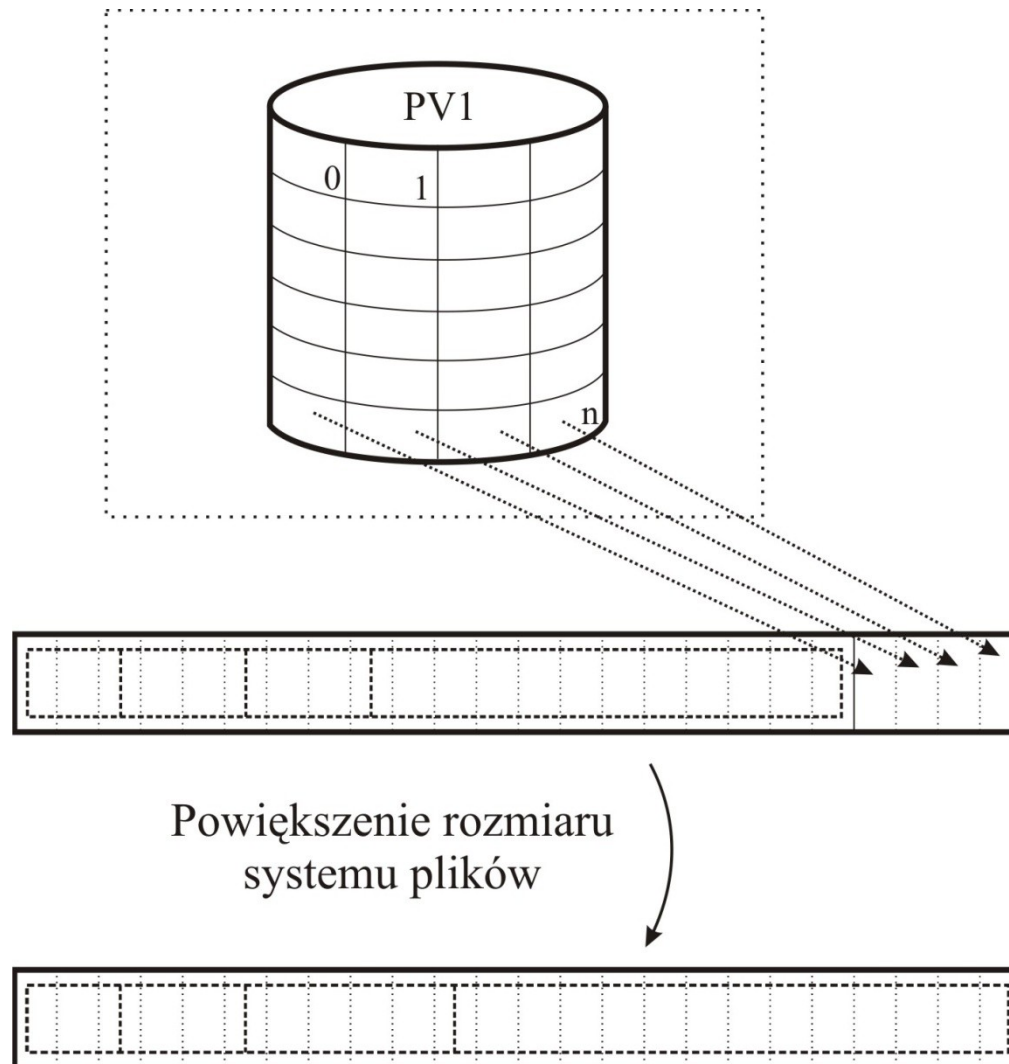
„Łączenie” partycji – LVM (1)



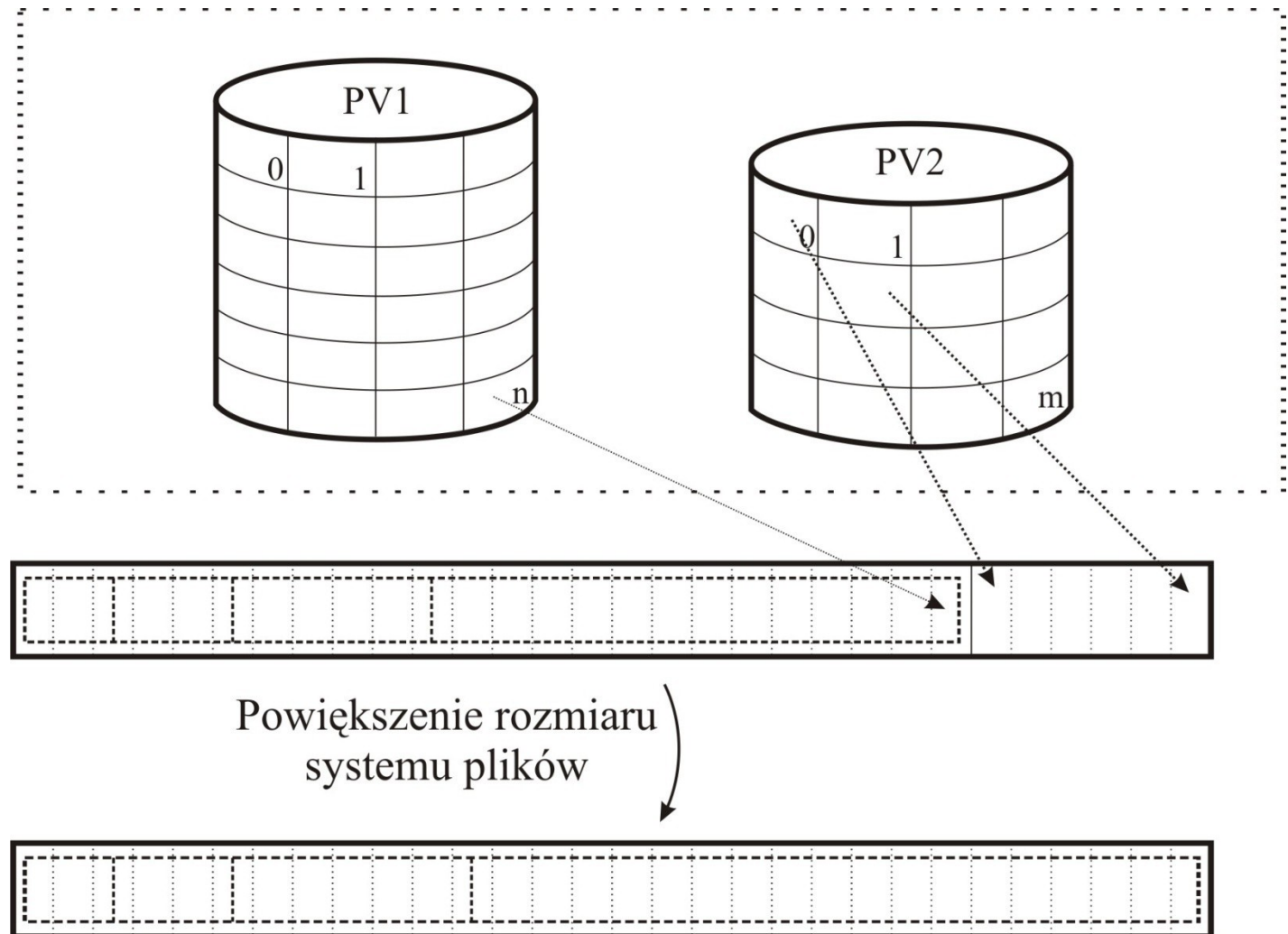
„Łączenie” partycji – LVM (2)



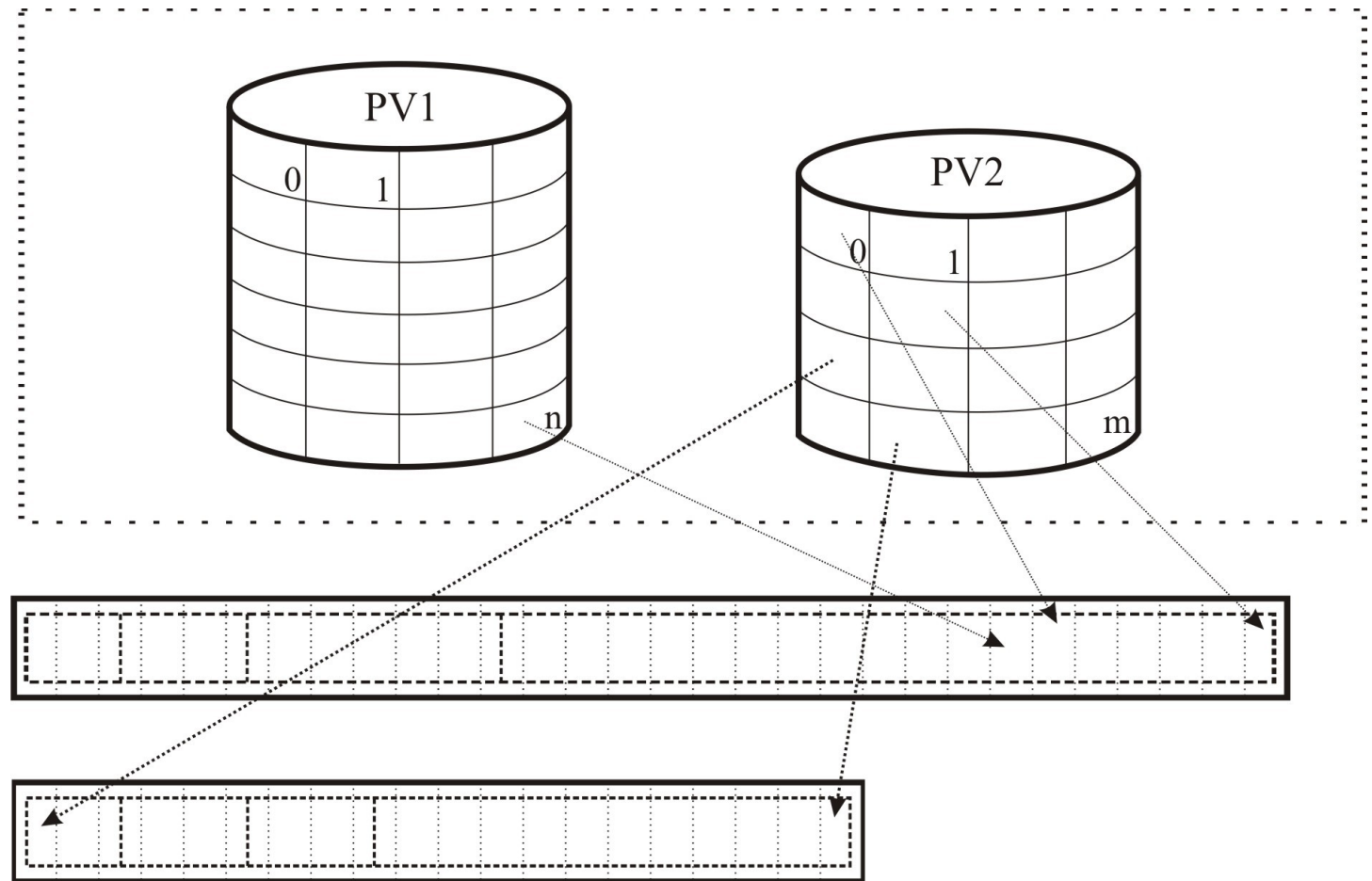
„Łączenie” partycji – LVM (3)



„Łączenie” partycji – LVM (4)



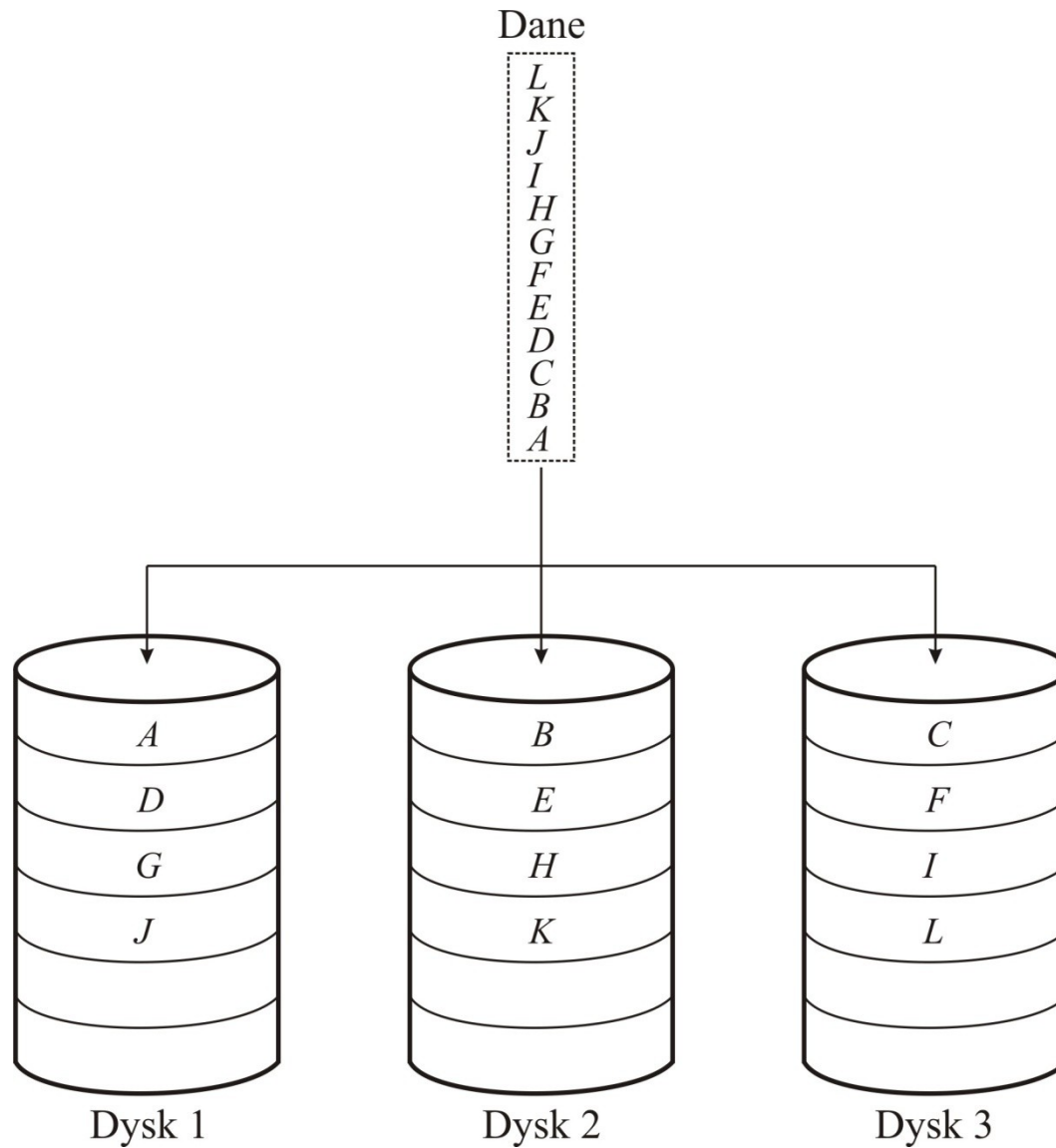
„Łączenie” partycji – LVM (5)



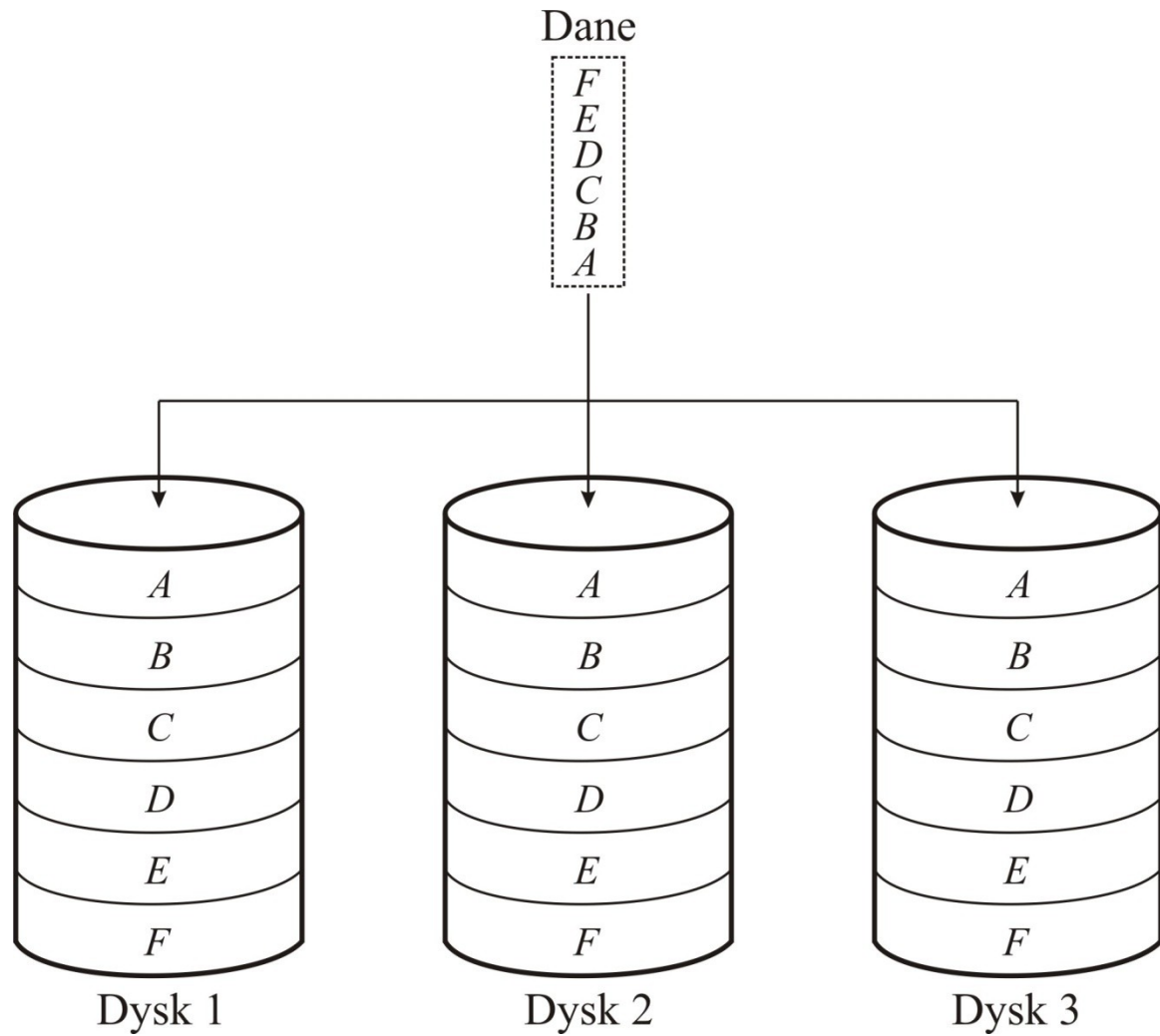
Macierze RAID

- Realizacja sprzętowa
- Implementacje programowe:
 - Linux – mdadm
 - BSD – vinum, geom
- Podstawowe poziomy: 0, 1, 5, 5+1, 1+0, 0+1

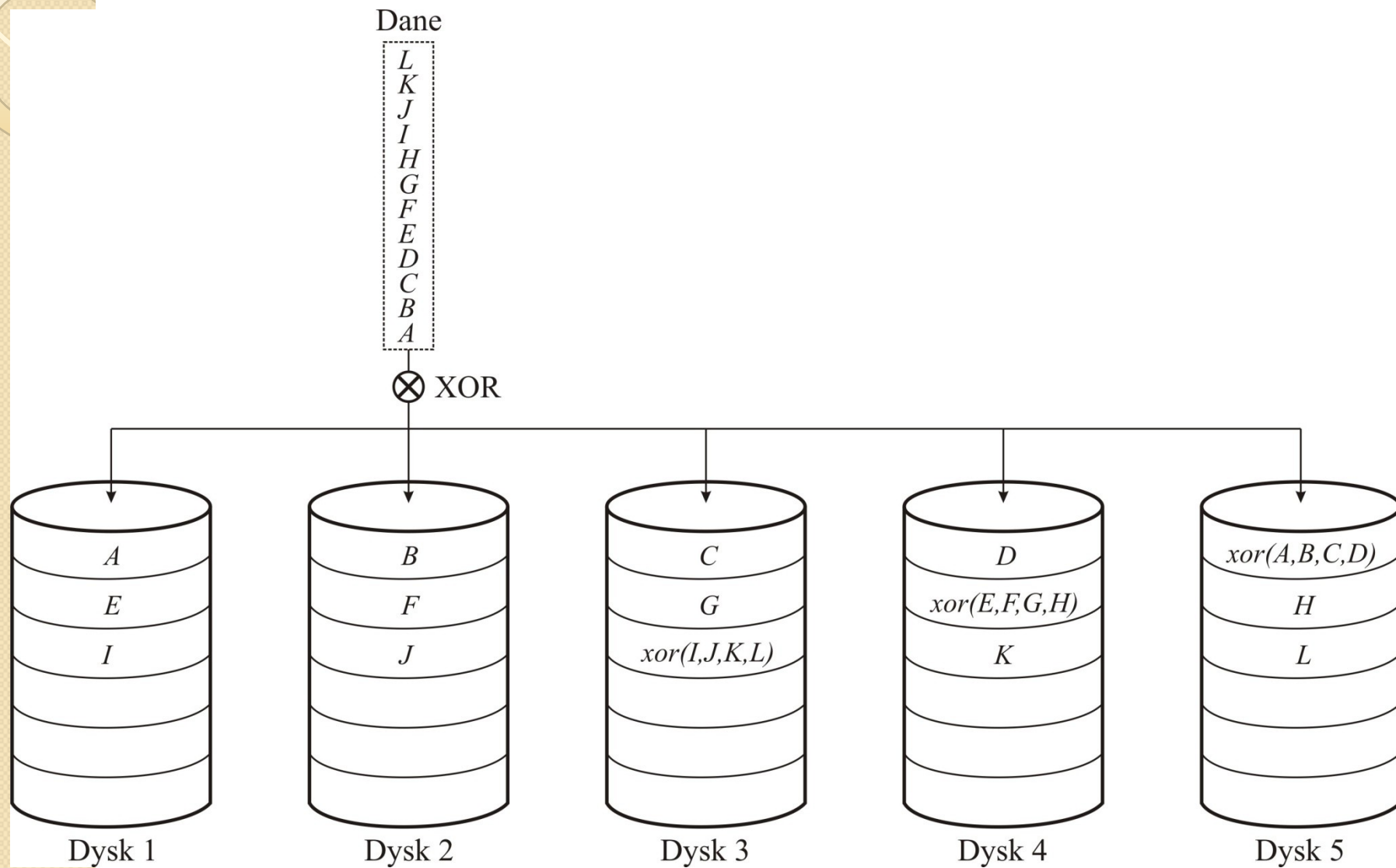
RAID0



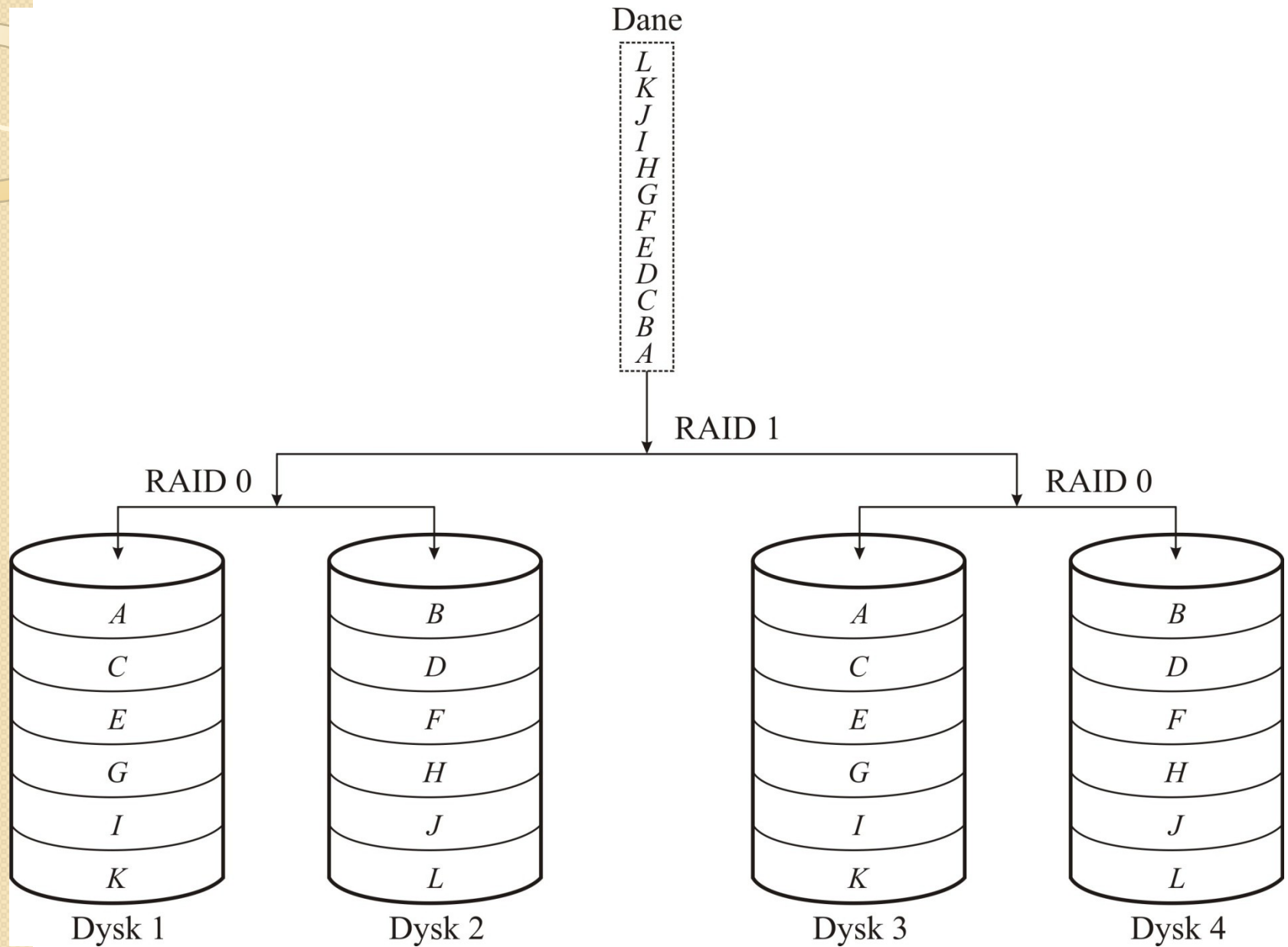
RAID 1



RAID 5



RAID 01



RAID 10

