System plików

System plików

- przez "system plików" rozumiemy tutaj *strukturę katalogów* + *pliki*
- system plików przechowuje się w *pamięci pomocniczej*
  (dyski magnetyczne, optyczne; taśmy magnetyczne; dyski elektroniczne)
- niezależnie od *rodzaju* pamięci pomocniczej system plików wygląda "tak samo" dla użytkownika
- zastosowanie struktury katalogów: "organizacja" plików –nazwa katalogu powinna sugerować jego zawartość

hierarchia pamięci:

- rej. procesora (np. AX)
- pamięć podręczna p. operac.
- pamięć operacyjna
- pamięć podręczna dysku
- dysk magnetyczny
- dysk optyczny
- taśmy magnetyczne

} pamięć pomocnicza

System plików
pliki

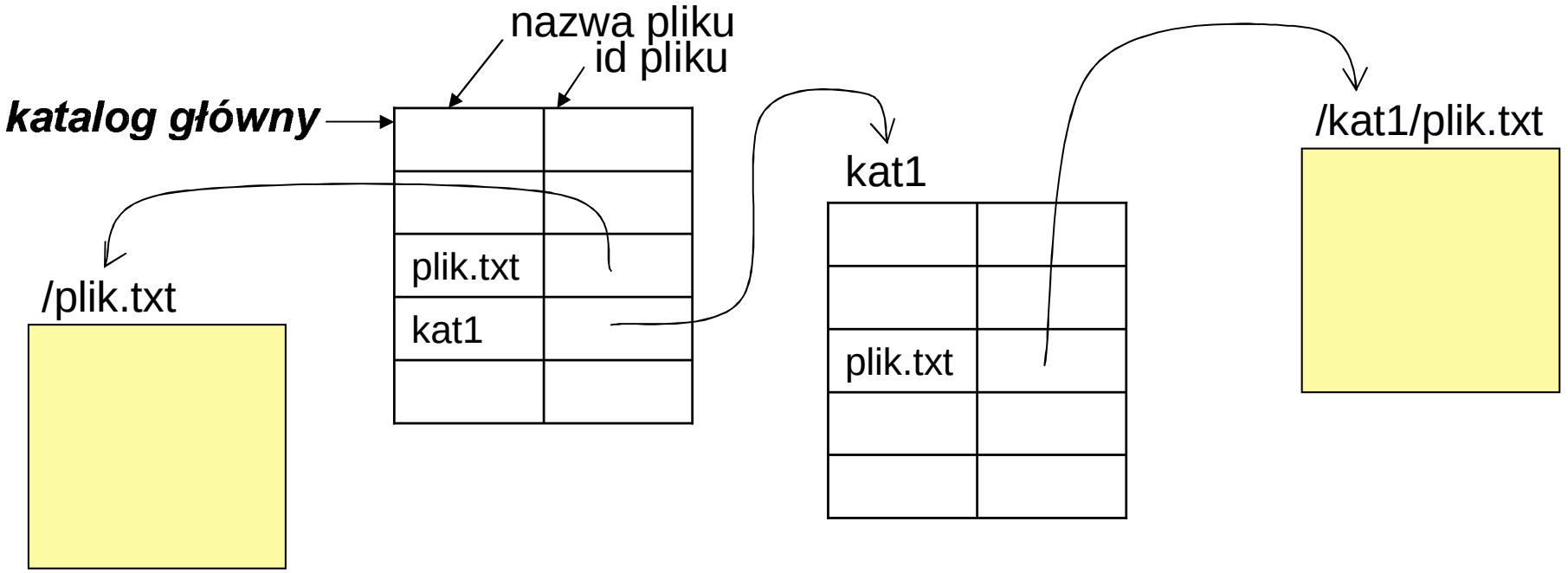
- atrybuty pliku:
  - (to co można odczytać poleceniem "ls –l" w Unix-ie)
    nazwa, typ, rozmiar, ochrona [prawa], czas utworzenia/ ostatniej modyfikacji/ ostatniego dostępu
  - "położenie" zawartości pliku na dysku
- operacje na plikach:
  - tworzenie pliku (*Unix*: d=open(..., O_CREAT);)
  - zapis i odczyt (*Unix*: write(d,...); read(d,...);)
  - zmiana bieżącej pozycji pliku (*Unix*: lseek(d, ...);)
  - usuwanie pliku (*Unix*: to jest operacja na katalogu !; unlink("ścieżka");)
  - skracanie pliku do długości=0 (*Unix*: d=open(..., O_TRUNC);)
  - dopisywanie na końcu pliku (*Unix*: d=open(..., O_APPEND);)
  - zmiana nazwy pliku (*Unix*: to jest operacja na katalogu !)
  - blokowanie fragmentów plików (*Unix*: tzw "mandatory locking"; można założyć "shared lock" lub "exclusive lock" na fragment pliku; w Unix-ie jest też mechanizm "advisory locking")
  - otwieranie/ zamykanie pliku – otrzymujemy/ zwalniamy deskryptor przy pomocy którego można wykonywać niektóre operacje na pliku (*Unix*: d=open("ścieżka"); ...; close(d);)

System plików
pliki

- struktura pliku:
  - DOS, Win, Unix: plik jest ciągiem bajtów które mogą być czytane w dowolnej kolejności (czyli właściwie nie ma struktury ...)
  - Palm OS (?): plik przypomina tabele bazy danych (wiersze i kolumny z etykietkami)
- metody dostępu do pliku:
  - dostęp sekwencyjny
    (czytamy kolejne bajty, bieżąca pozycja przesuwa się w kierunku końca pliku)
  - dostęp swobodny
    (czytamy bajty pliku w dowolnej kolejności)

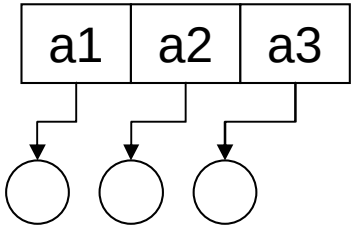
System plików
katalogi

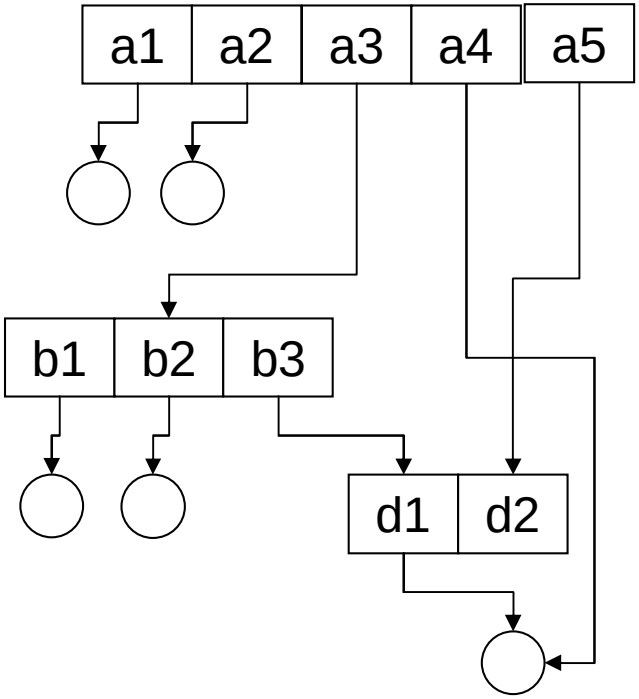
- struktura katalogów pozwala "przetłumaczyć" ścieżkę do pliku na atrybuty pliku ...
- katalog to tablica której elementami są (nazwa pliku, id pliku)
  - "id pliku" jednoznacznie identyfikuje plik
  - DOS/FAT: id pliku = nr pierwszego bloku pliku
  - Unix: id pliku = nr i-węzła
- katalogi implementuje się tak jak pliki (są to pliki zawierające wspomnianą wyżej tablicę ...)

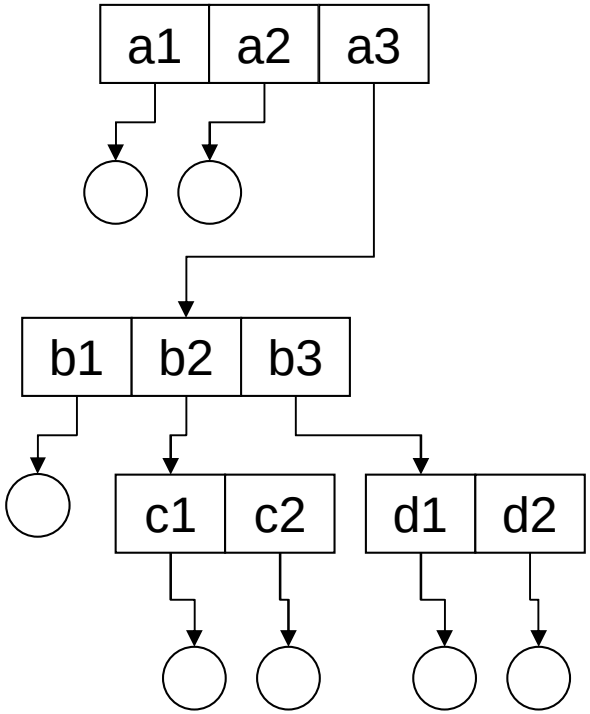


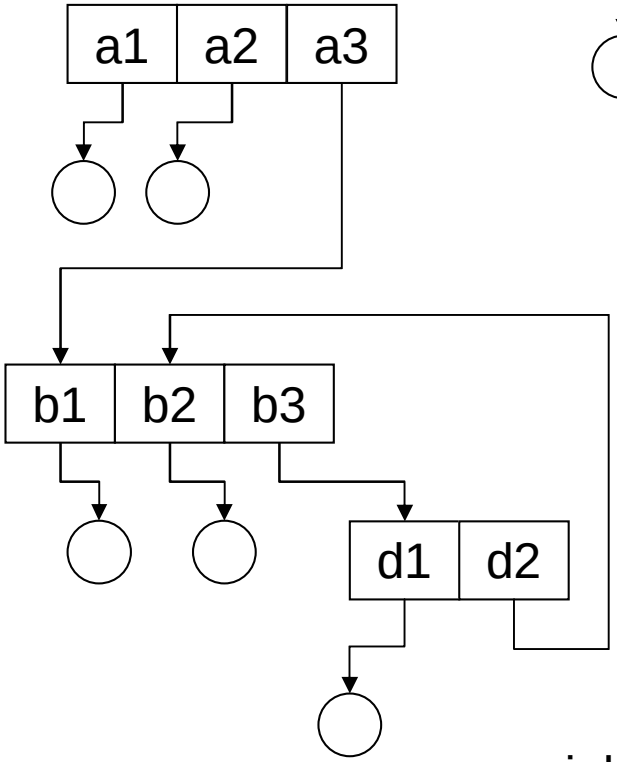
System plików
katalogi

- rodzaje struktur katalogów:
  - jednopoziomowy
  - drzewiasty
  - acykliczny
  - ogólny





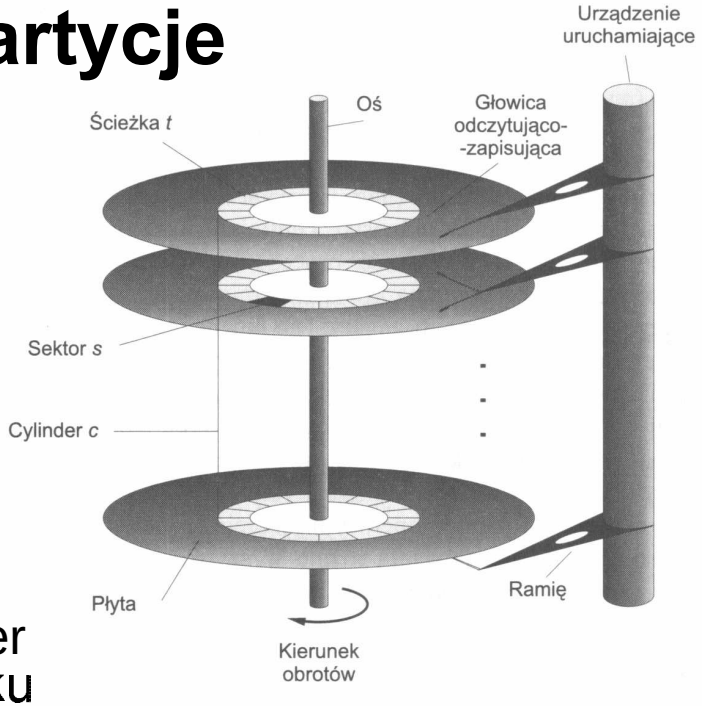




... jakie to struktury katalogów ???

System plików
podział dysku na partycje

- dysk fizyczny dzielimy na *partycje,* które są "dyskami logicznymi"
- partycja może zawierać system plików użytkownika lub nie (np. istnieją partycje obsługujące pamięć wirtualną i/lub wymianę)
- zazwyczaj każda partycja zajmuje pewien zakres cylindrów dysku fizycznego (np. partycja 1-0..100, 2-101..200, 3-201..500)
- rozruch komputera:
  - **główny rekord ładujący** (MBR = ang. Master Boot Record); znajduje się w 1 sektorze dysku fizycznego (głowica 0, cylinder 0, sektor 1)
  - **tablica partycji** w MBR opisująca partycje
  - każda partycja ma swój **rekord ładujący** (BR) w pierwszym sektorze partycji
  - **program rozruchowy** znajduje się w MBR; ładowany do pamięci i uruchamiany podczas włączania komputera; znajduje tzw *partycję aktywną* i uruchamia program rozruchowy w jej BR, który ładuje system operacyjny zainstalowany w tej partycji ...

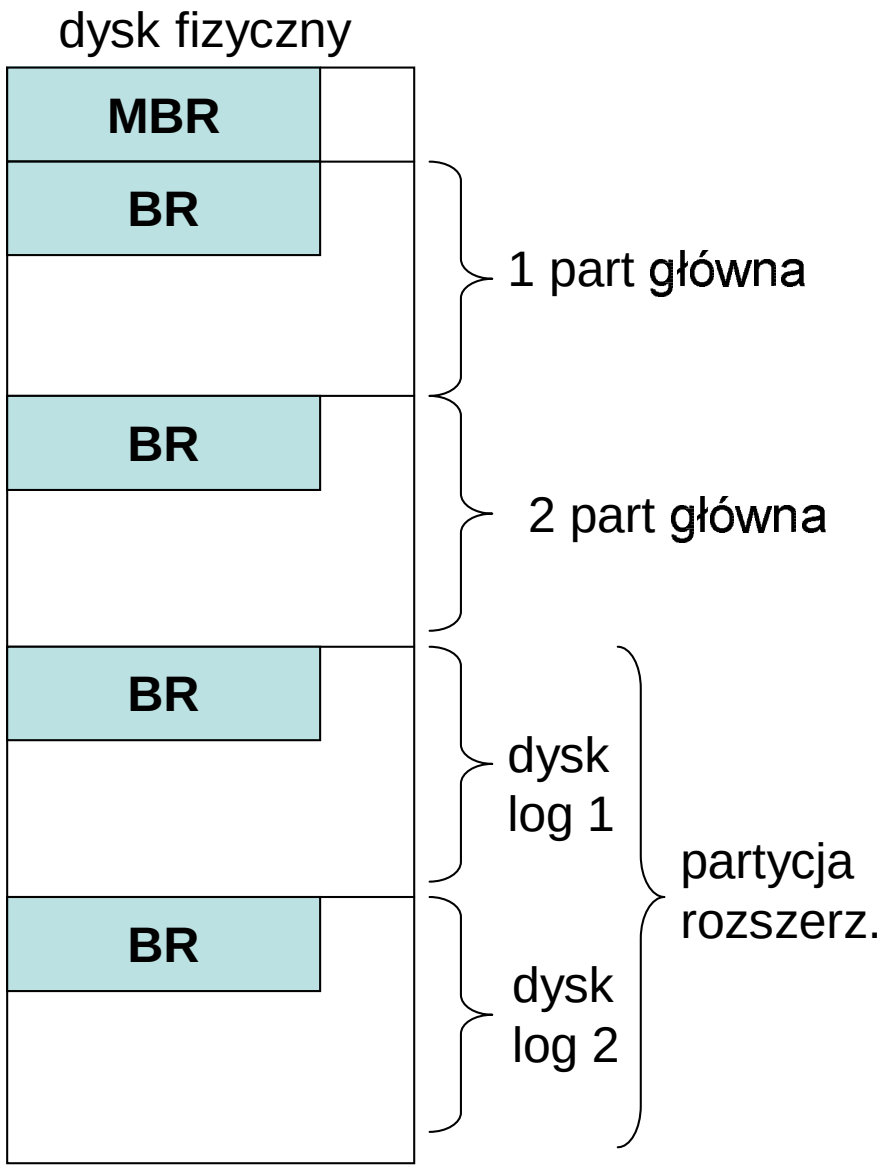


Rys. 2.5 Mechanizm dysku z ruchomymi głowicami

w każdej partycji możemy mieć inny s.operacyjny ze specyficznym dla tego s.o. programem rozruchowym w BR tej partycji);
patrz tez: LILO Linuxa

System plików
podział dysku na partycje

- partycje na komputerach PC ...
  - partycje główne
  - jedna partycja rozszerzona z dyskami logicznymi
  - liczba part. głównych + part. rozsz. (jeśli jest) <= 4
  - nieograniczona liczba dysk. log.
- *Unix*: operacje na partycjach
  - tworzenie/usuwanie partycji:
    ```
    fdisk
    cfdisk
    ```
  - tworzenie systemu plików:
    ```
    mkfs –t ext2 /dev/hda1
    ```
    ext2 – typ systemu plików;
    /dev/hda1 - 1 partycja 1 dysku



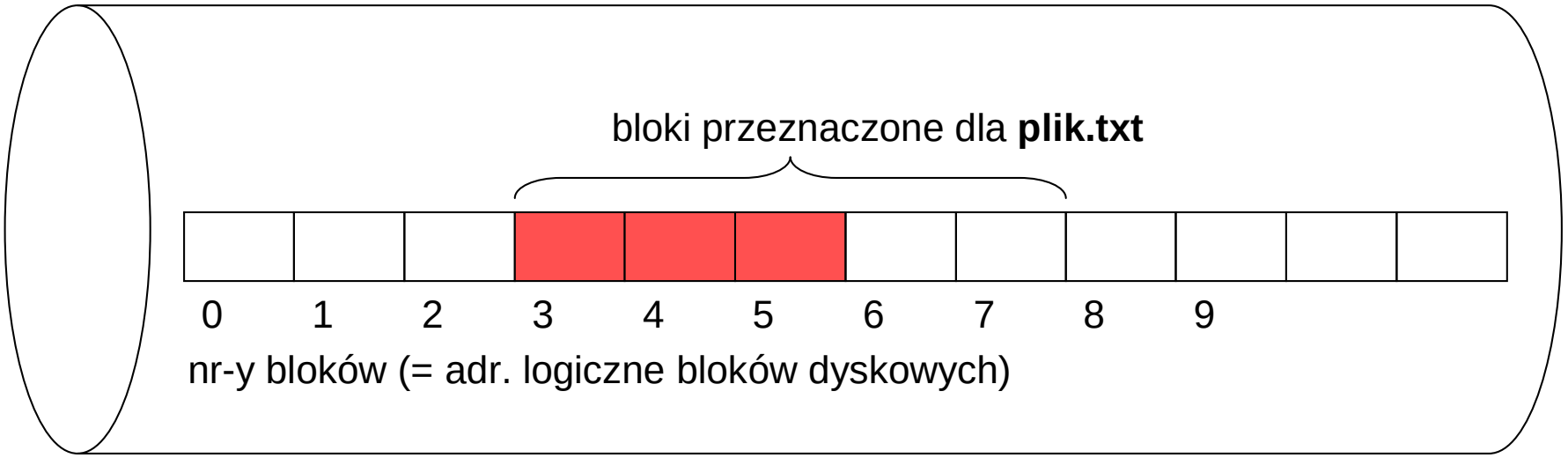
System plików (implementacja)

System plików (implementacja)

- system plików rezyduje w pamięci pomocniczej, której zadaniem jest *trwałe* przechowywanie *dużych ilości danych* ...
- cechy dysku magnetycznego:
  - kopiowanie danych między dyskiem i pamięcią operacyjną wykonuje się w jednostkach zwanych ***blokami dyskowymi*** lub ***sektorami*** (rozmiar sektora = zazwyczaj 512 B)
  - nie można przeczytać z dysku "mniej niż jeden blok" !
  - bloki dyskowe mają "adresy fizyczne" (nr głowicy, nr cylindra, nr sektora) oraz "adresy logiczne" (kolejny nr bloku)
  - można czytać/pisać bloki w dowolnej kolejności
- w blokach przechowuje się:
  - zawartość plików
  - inne struktury danych wymagane przez s. plików (np. katalogi)
- poszerzone znaczenie pojęcia "system plików":
  - katalogi + pliki (z pkt widzenia zwykłego użytkownika ...)
  - partycja zawierająca struktury danych implementujące katalogi + pliki
  - część s.o. obsługująca te struktury danych

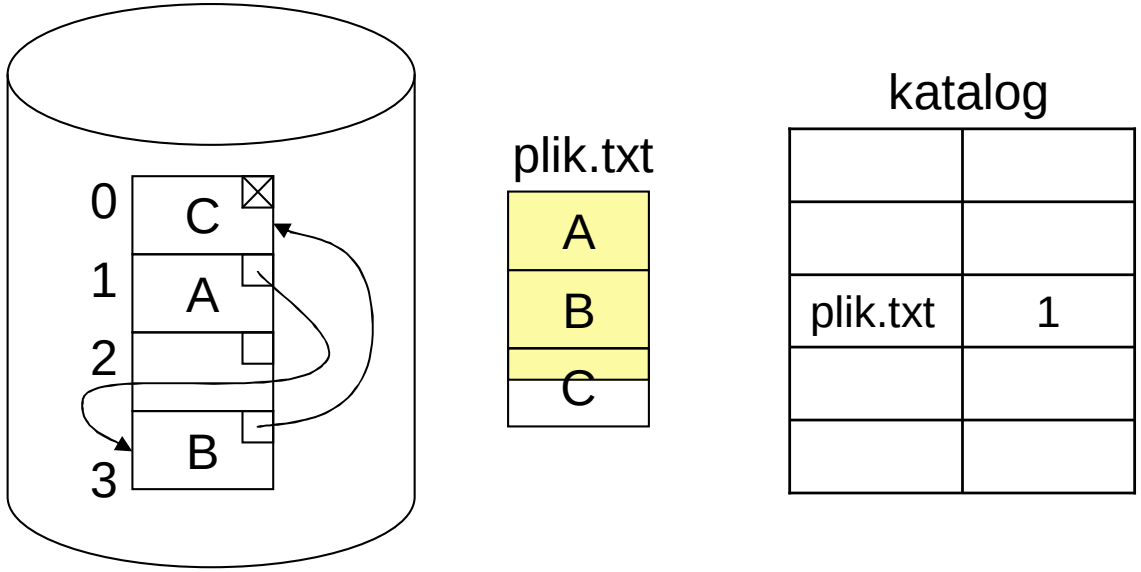
System plików (implementacja)
metody przydziału miejsca na dysku

- przydział ciągły:
  - każdy plik musi zajmować ciąg kolejnych bloków dyskowych
  - zalety:
    - bardzo szybki dostęp sekwencyjny
  - wady:
    - trzeba znać maksymalną długość pliku przy jego tworzeniu i tyle miejsca rezerwować (fragm. wewn. !)
    - po usunięciu pliku zostają "dziury" które mogą być za małe dla nowych plików (fragm. zewn. !)



System plików (implementacja)
metody przydziału miejsca na dysku

- przydział listowy:
  - w każdym bloku przechowuję adres następnego bloku
  - zalety:
    - rozwiązuje problem fragm. zewn. (czyli "niewykorzystanych dziur")
  - wady:
    - efektywny jest TYLKO dostęp sekwencyjny; dostęp swobodny jest strasznie wolny ! (przypuśćmy że chcemy przeczytać 1000-ny blok naszego pliku ... skąd wziąć jego adres ???)

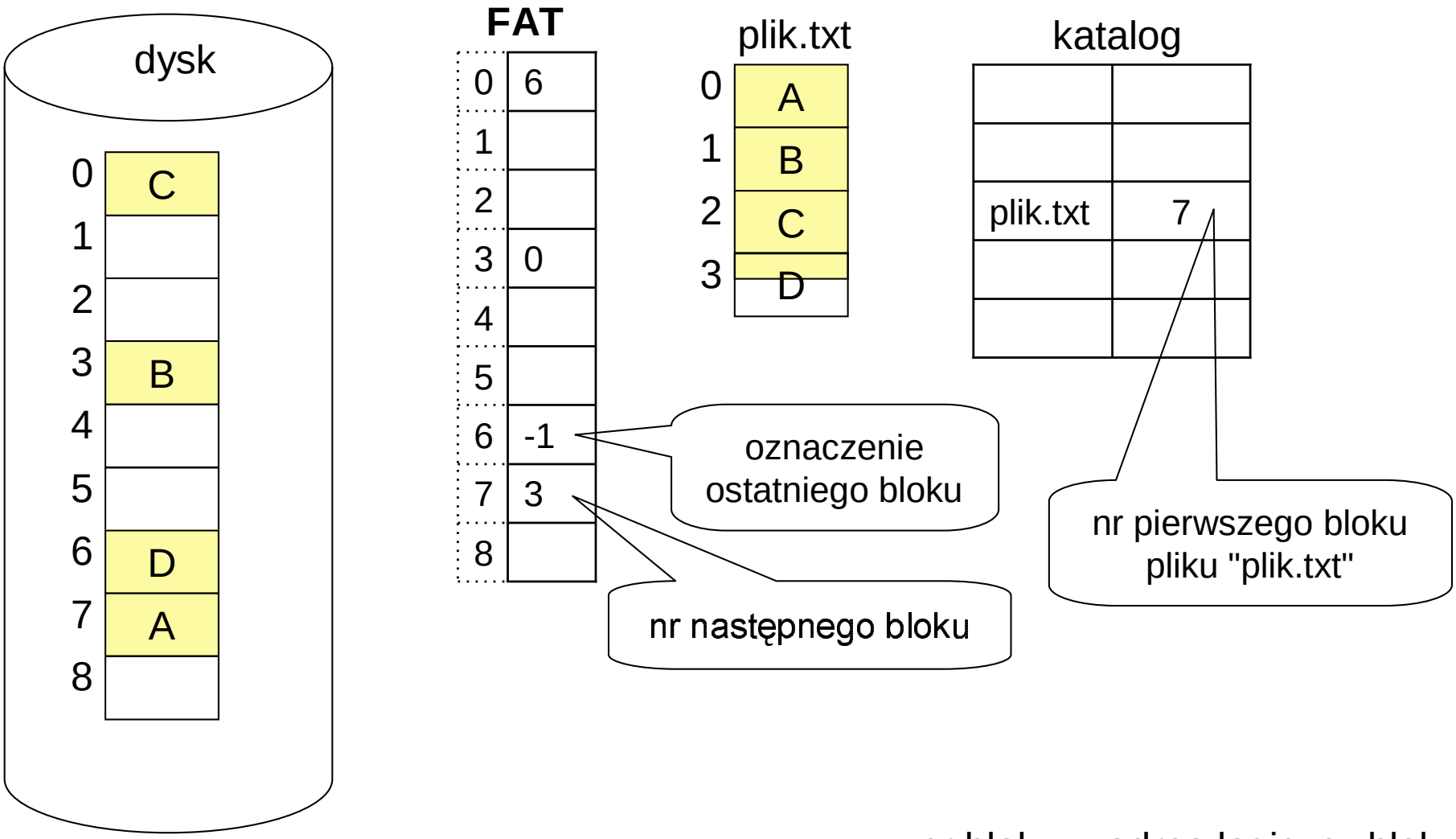


System plików (implementacja)
metody przydziału miejsca na dysku

- przydział listowy "ulepszony" = **FAT**
  - FAT [ang. File Allocation Table] = tablica alokacji plików
  - tablica ta umożliwia zamianę nr bloku w pliku na nr bloku na dysku bez odczytywania wcześniejszych bloków dyskowych
    (patrz: następny slajd)
  - tablica FAT jest na tyle mała że może być przechowywana w pamięci operacyjnej (i oczywiście także na dysku)
  - z uwagi na ograniczoną liczbę elementów tablicy FAT i duże pojemności dysków magnetycznych zamiast blokami posługujemy się tzw *klastrami* które składają się z większej liczby bloków
    (liczba bloków w klastrze jest ustalona i zapisana w BR partycji)
  - często używa się nazwy "system plików FAT??" ...
  - FAT12, FAT16, FAT32
    - przykładowo FAT16 oznacza że element tablicy FAT jest zapisany na 16 bitach, czyli FAT ma 2^16 elementów; tak więc maksymalny rozmiar dysku który potrafi obsłużyć FAT16 to 2^16 * "rozmiar klastra";
      np. jeśli klaster= 1 sektor= 2^9 B, wtedy 2^16*2^9 =2^25 =32 MB
  - *zalety*: teraz JEST możliwy efektywny dostęp swobodny ...

System plików (implementacja)
metody przydziału miejsca na dysku

- przydział listowy "ulepszony" = **FAT**



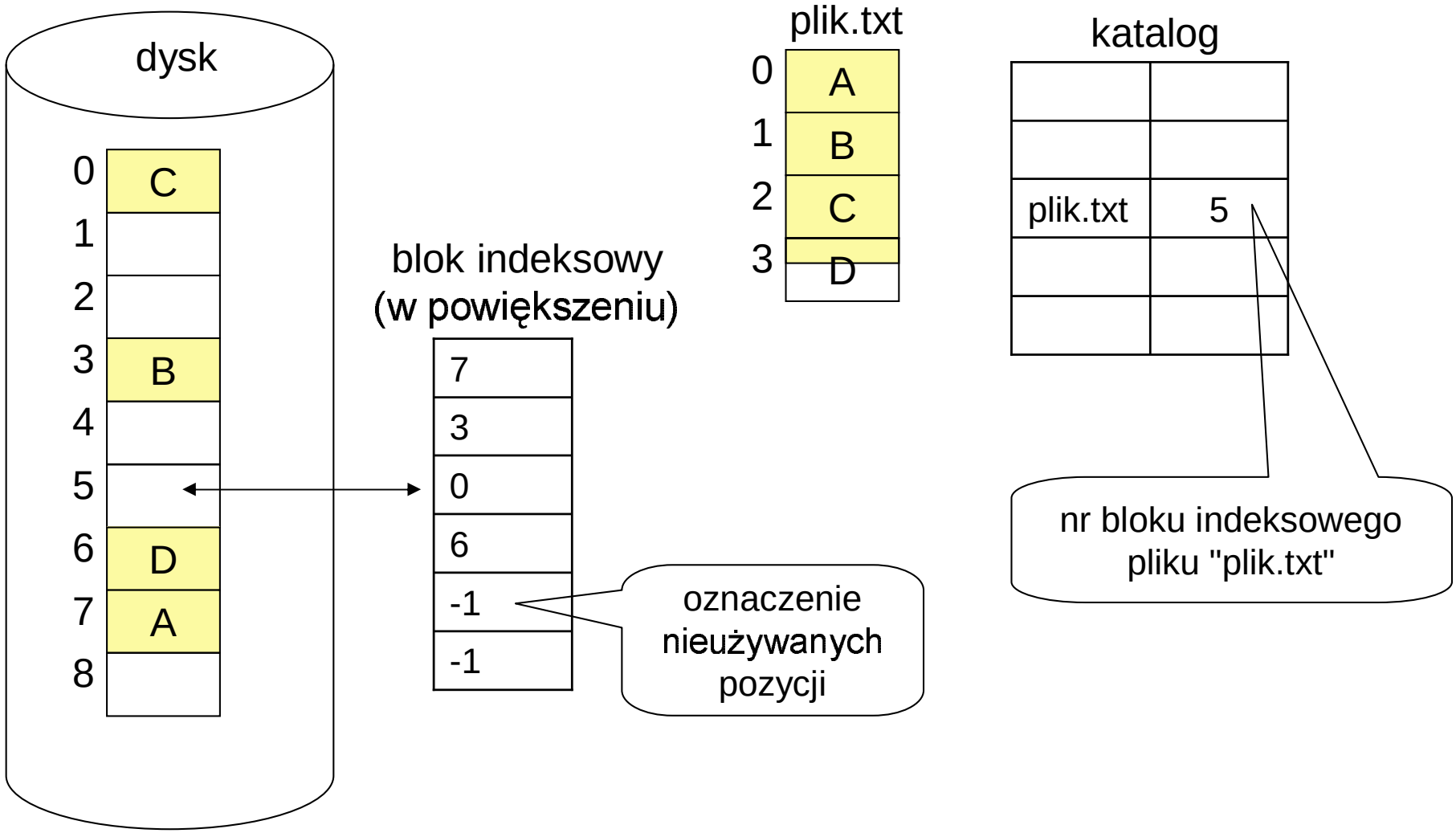
nr bloku = adres logiczny bloku

System plików (implementacja)
metody przydziału miejsca na dysku

- przydział indeksowy
  - dla każdego pliku poświęca się jeden(?) blok dyskowy na tzw *blok indeksowy*, przechowujący nr (na dysku) kolejnych bloków pliku (patrz następny slajd)
  - pozwala na szybki dostęp swobodny oraz rozwiązuje problem fragm. zewn. (podobnie jak "przydział listowy+FAT")
  - *wada*: nie wiemy ile bloków indeksowych przypisać do pliku bo to zależy od długości jaką plik osiągnie;
    rozwiązanie (w Unix-ie): stosuje się hierarchię bloków indeksowych zamiast ustalonej liczby tych bloków

System plików (implementacja)
metody przydziału miejsca na dysku

- przydział indeksowy



System plików (implementacja)
metody przydziału miejsca na dysku

- przydział indeksowy z hierarchią bloków indeksowych (Unix)
  - w Unix-ie każdy plik jest reprezentowany przez tzw **i-węzeł**, który zawiera atrybuty pliku – w szczególności położenie bloków pliku ...
  - w każdej pozycji katalogu przechowuje się "nazwę pliku" i "nr i-węzła"
  - i-węzeł zawiera także: nr-y (adr. log.) bloków danych, nr-y bloków indeksowych: "jedno-pośredni", "dwu-pośredni", "trój-pośredni"