

Wykład 9

Struktury pamięci masowej

Struktura dysku

Dane przechowywane są na powierzchniach

Maksimum dwie powierzchnie na talerzu.

Conajmniej jeden talerz

Dane znajdują się na koncentrycznych ścieżkach.

Ścieżki podzielone są na sektory

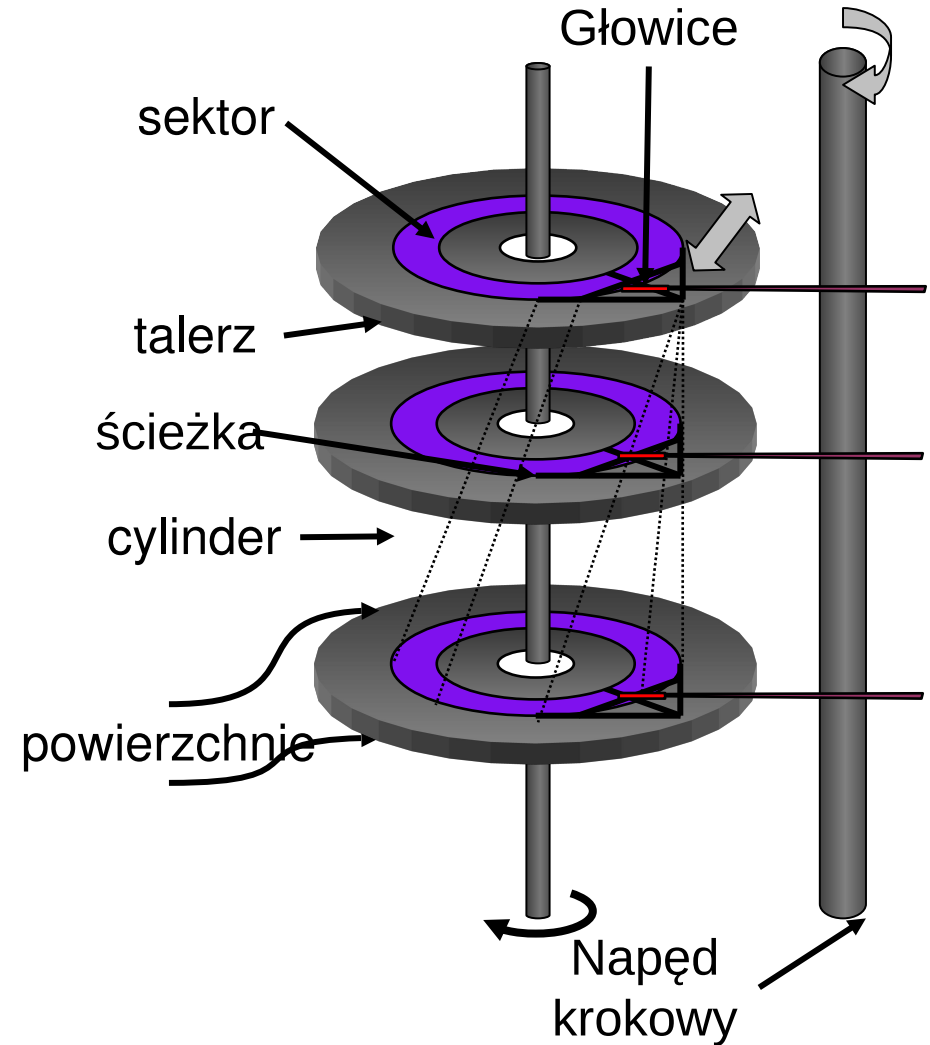
Odpowiednie ścieżki na wszystkich powierzchniach tworzą cylinder

Dane są zapisywane i odczytywane przez głowice.

Głowice przesuwane są przez napęd krokowy

Wszystkie głowice przesuwane są jednocześnie

Dysk z wymiennym lub niewymiennym nośnikiem



Przykład

	IBM 360KB floppy	WD 18GB HD
Cylinders	40	10601
Tracks per cylinder	2	12
Sectors per track	9	281 (średnio !!!)
Sectors per disk	720	35742000
Bytes per sector	512	512
Capacity	360 KB	18.3 GB
Seek time (minimum)	6 ms	0.8 ms
Seek time (average)	77 ms	6.9 ms
Rotation time	200 ms	8.33 ms
Spinup time	250 ms	20 sec
Sector transfer time	22 ms	17 μ sec

Adresowanie dysku

Dysk może mieć miliony sektorów. Jak je zaadresować.

Metoda 1: Cylinder/Głowica/Sektor.

Liczba sektorów na ścieżce może się zmieniać.

Metoda 2: Numerowanie sekwencyjne

Dysk odwzorowuje globalny numer sektora z zakresu [0-max] na numer cylindra, ścieżki oraz sektora w cylindrze.

Odwzorowanie może się zmieniać !!!

Przemieszczanie błędnych sektorów

Optymalizacja wydajności.

Nowoczesne dyski wykorzystują metodę 2.

Dzięki temu szczegóły geometrii są ukryte przed systemem operacyjnym

Ale zawsze opłaca się w jednym żądaniu odczytywać grupę kolejnych sektorów

Układ sektorów na dysku

Wewnętrzne ścieżki są krótsze niż zewnętrzne.

Dwa rozwiązania:

(a) dłuższe bity

(b) więcej sektorów na ścieżkach zewnętrznych

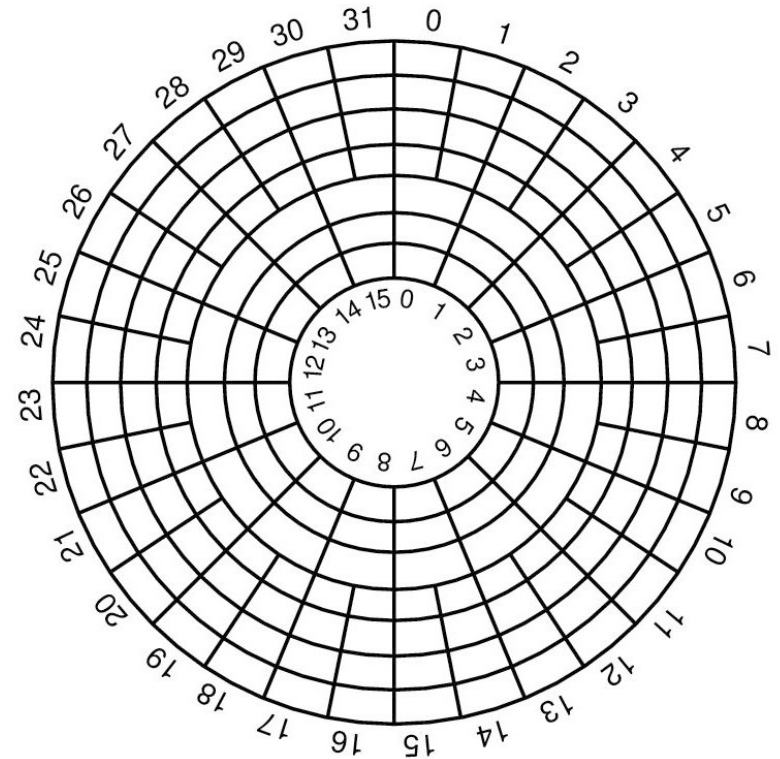
Stacja dysków elastycznych wykorzystuje rozwiązanie (a)

Nowoczesne dyski wykorzystują rozwiązanie (b).

Dysk podzielony jest na kilka stref (8-20).

W każdej ze stref na jedną ścieżkę przypada stała liczba sektorów

W rozwiązaniu (b) odczyt sektorów ze ścieżek zewnętrznych następuje szybciej.



“Zapasowe” sektory

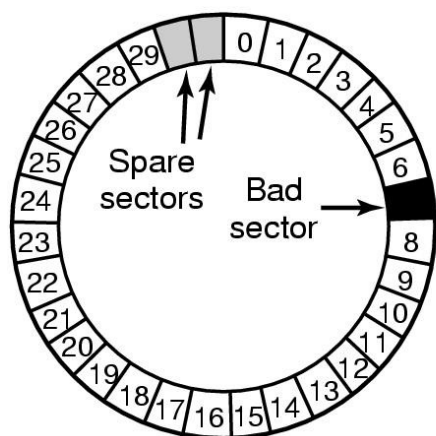
Ścieżka może zawierać dodatkowe sektory.

Gdy dany sektor zaczyna sprawiać problemy (dużo błędów poprawianych przez ECC), przenoszony jest do sektora dodatkowego.

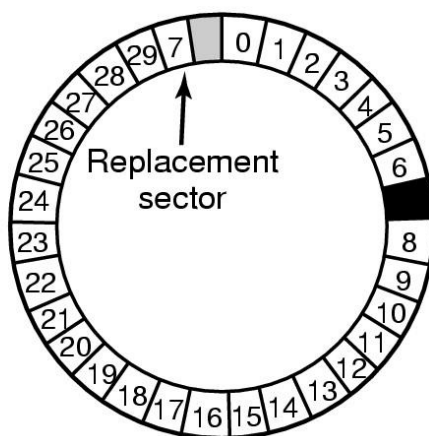
Pogarsza to wydajność.

Aby temu zapobiec, można na nowo posortować sektory na ścieżce, ale zajmuje to dużo czasu

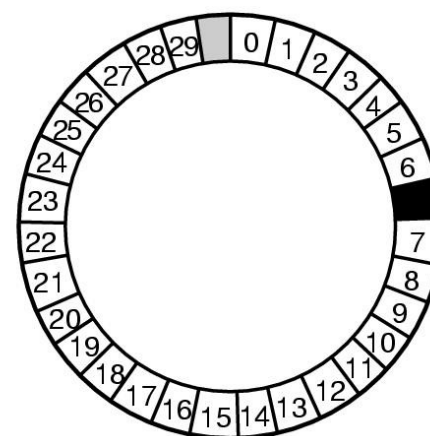
Mechanizm ten jest często ukryty przed systemem operacyjnym



(a)

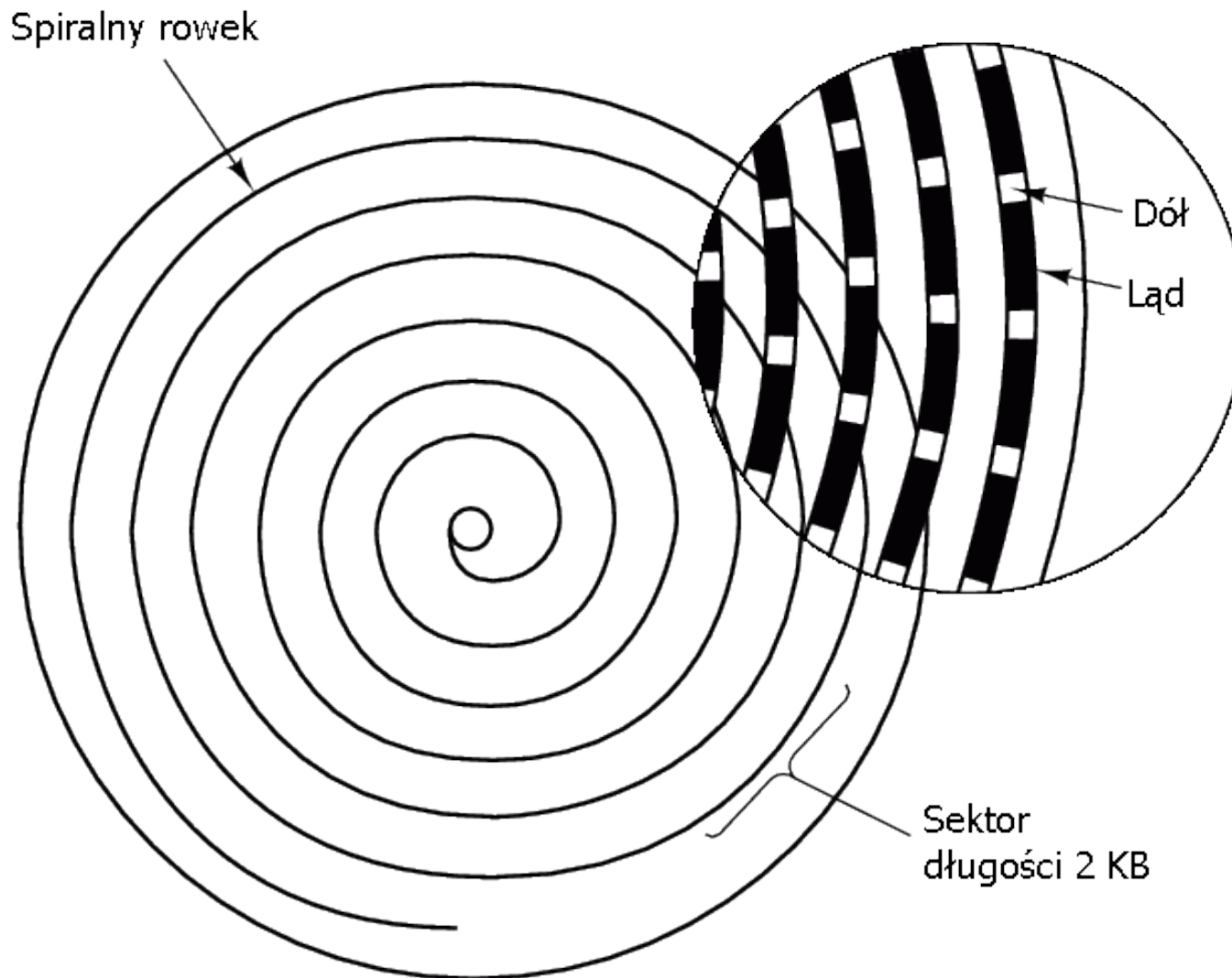


(b)



(c)

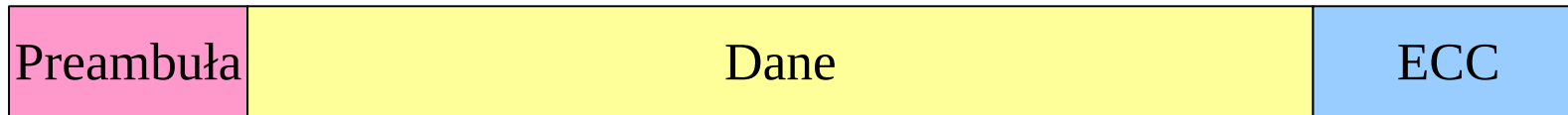
Napędy optyczne - CD-ROM, DVD-ROM, ...



W napędach optycznych nie mamy koncentrycznych ścieżek, a spiralę.

Sektor ma 2KB

Struktura sektora dyskowego



Preambuła – zawiera numer sektora, i cylindra

Dane (długość 256, 512 lub 1024 bajty)

ECC (ang. error correcting code) – nadmiarowe informacje pozwalające na wykrycie (prawie) wszystkich błędów i poprawienie niektórych z nich

Czas wykonania operacji

Czas operacji = Czas dostępu + Czas transferu

Czas dostępu, zależy od dwóch składników.

Czas wyszukiwania (ang. seek time) czas niezbędny na ustawienie się głowicy dysku nad właściwym cylindrem.

Oczekiwanie aż głowica znajdzie się nad odpowiednim sektorem.

Zależy od prędkości obrotowej dysku, średnio $\frac{1}{2}$ czasu wykonania obrotu przez talerz.

System operacyjny może próbować optymalizować czas wyszukiwania.

Czas transferu też zależy od prędkości obrotowej dysku.

Przykład

Czas wyszukiwania = 7ms; Czas oczekiwania 4ms => Czas dostępu 11ms

Czas transferu 22us (trzy rzędy wielkości mniej !!!)

Wniosek: staraj się przysyłać wiele kolejnych sektorów w jednym żądaniu.

Planowanie żądań do dysku

Cel: Minimalizuj czas spędzany na przesuwaniu głowicy i oczekiwaniu.

Na oczekiwanie mamy niewielki wpływ.

Umieszczaj bloki składające się na dany plik w “pobliżu siebie”.

Używaj algorytmów planowania do ustalania kolejności obsługi żądań.

Istnieje wiele algorytmów planowania.

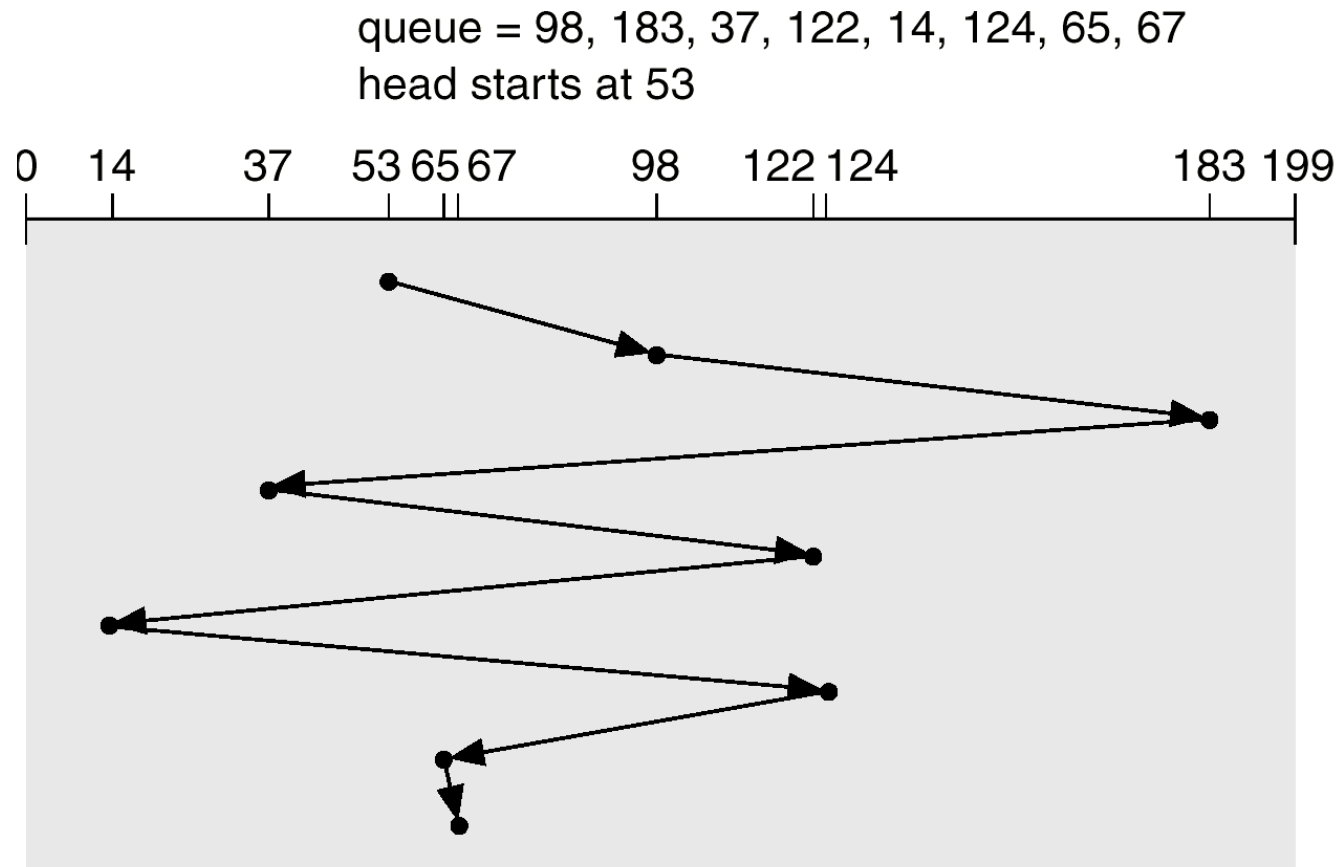
Przykład:

Głowica nad blokiem 53.

Zakładamy kolejność żądań: 98, 183, 37, 122, 14, 124, 65, 67

Zakładamy, że kontroler przetwarza jedno żądanie - w nowych dyskach nie jest to do końca prawdą.

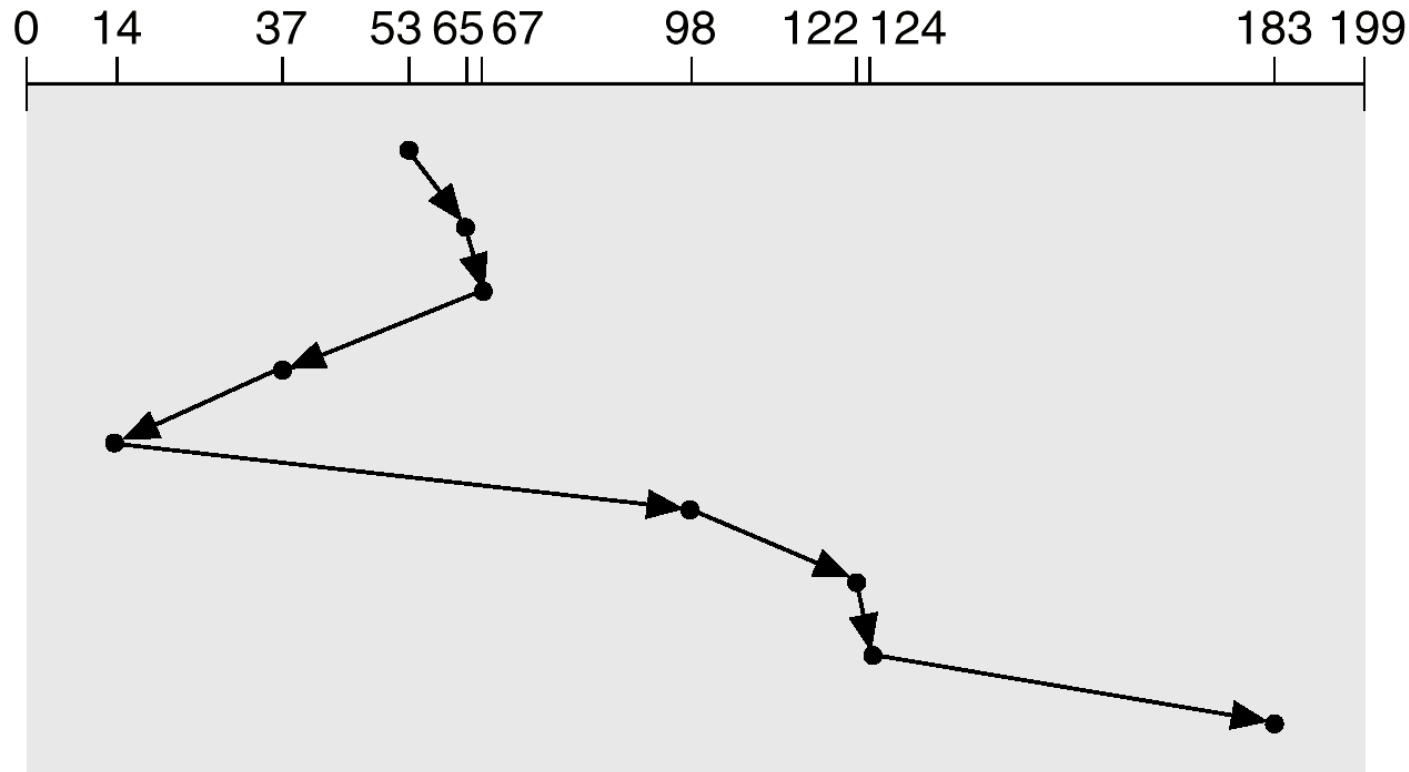
Algorytm FCFS (ang. First come, First served)



Obsługuj żądania w kolejności zgłoszenia.

Algorytm SSTF (ang. Shortest Seek Time First)

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



Wybierz żądanie o najkrótszym czasie oczekiwania.

Możliwość zagłódnienia

Algorytm SCAN

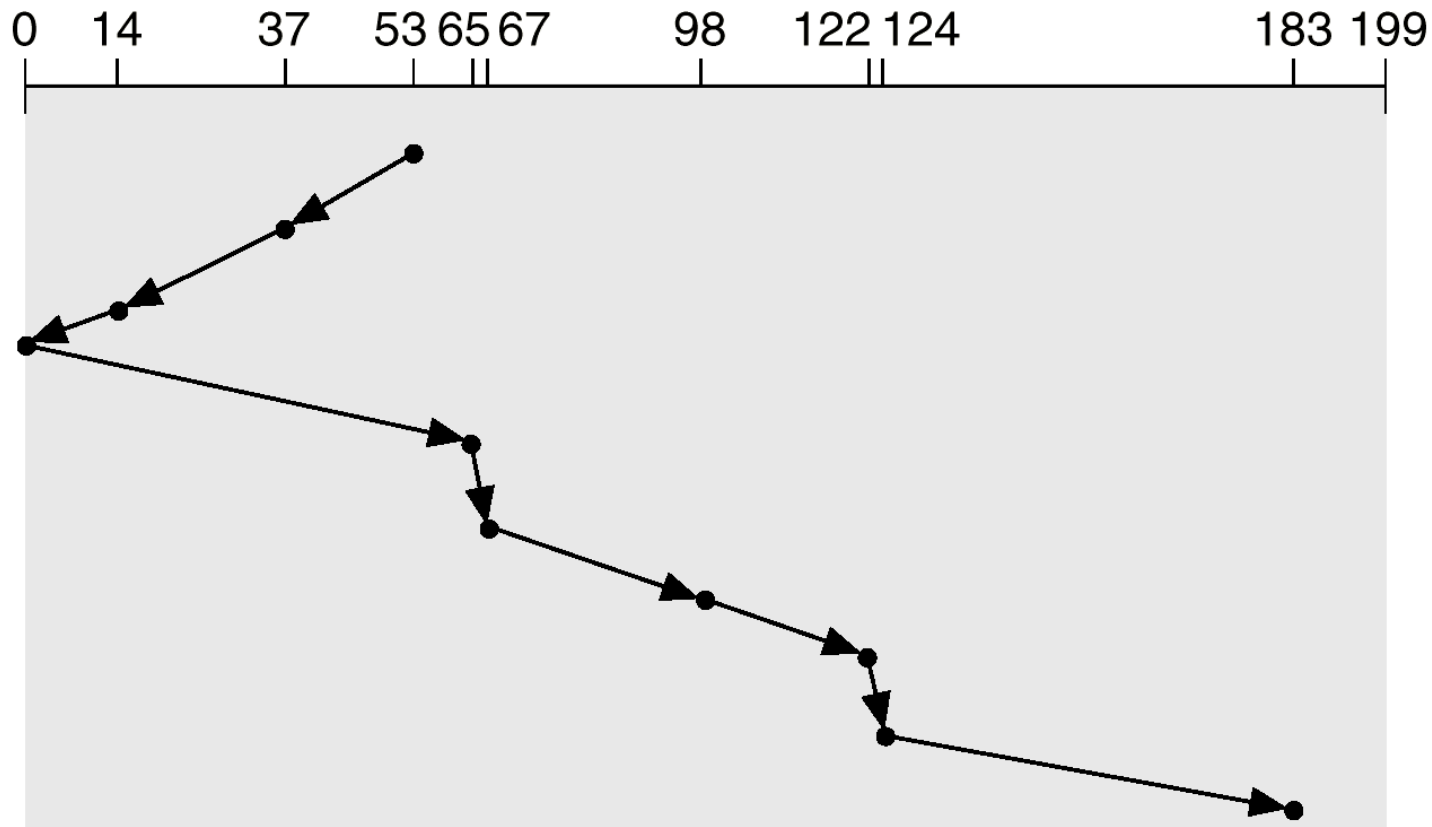
Zwany również algorytmem windy (ang. Elevator).

Ramię dysku porusza się od jednego końca dysku do drugiego, obsługując po drodze żądania.

Po dojściu do końca dysku kierunek poruszania się ramienia ulega zmianie.

Algorytm SCAN: Przykład

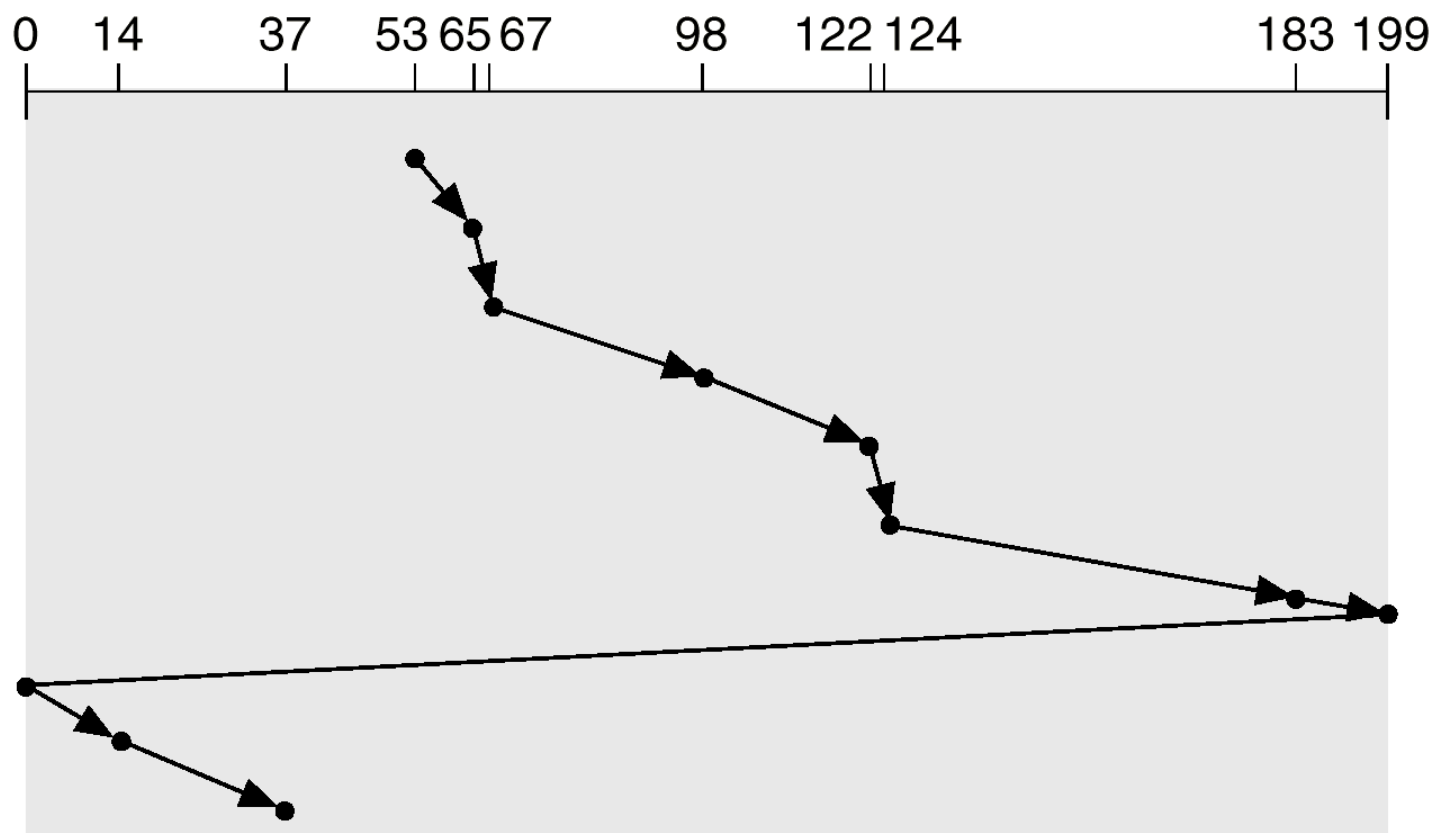
queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



Algorytm C-SCAN

Ulepszona wersja algorytmu SCAN. Gdy głowica dotrze do końca dysku, natychmiast jest przesuwana z powrotem na początek, bez obsługiwanie kolejnych żądań.

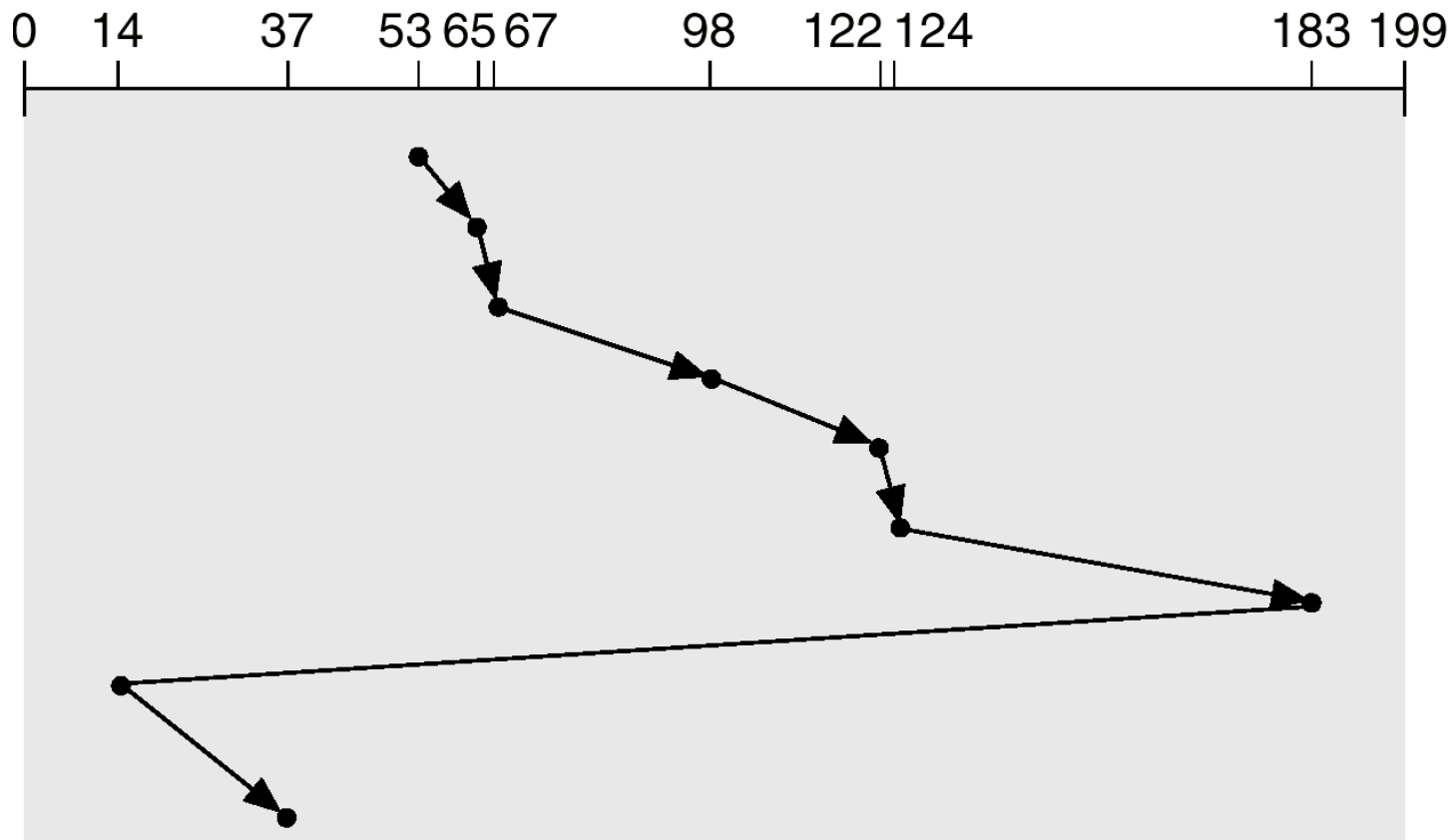
queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



Algorytm C-LOOK

Ramię przesuwa się, nie do końca a tylko tak daleko aby obsłużyć ostatnie żądanie.

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



Jaki algorytm wybrać

SSTF łatwy do zaimplementowania i ma dobrą wydajność, gdy obciążenie dysku jest niewielkie.

Algorytmy typu SCAN mają lepszą wydajność przy dużym obciążeniu dysku

- Brak możliwości zagłódzenia

- Wersje LOOK unikają niepotrzebnego przesuwania głowicy

Przejście od jednego końca dysku do drugiego jest stosunkowo mało kosztowne.

- Dlatego C-LOOK jest lepszy niż C-LOOK, bo zapewnia małą wariancję czasu dostępu.

Konkluzja:

- SSTF dla słabo obciążonych systemów

- C-LOOK dla bardzo obciążonych systemów .

Ponadto: Wraz ze wzrostem wyrafinowania elektroniki dysku (duże pamięci cache, integracja z pamięciami FLAS) pojawia się tendencja do implementacji części lub całości algorytmów szeregowania przez kontroler.

- Przykład: Natywne kolejkovanie komend (NCQ - ang. native command queueing) w standardzie Serial ATA.

Jak zbudować lepszy dysk

Wydajność procesorów wzrasta wykładniczo.

W przypadku dysków nie jest to prawdą.

Czas dostępu jest ograniczony przez mechanikę dysku

Niezawodność dysków nie jest wysoka.

Nic dziwnego, dysk ma miliony sektorów.

Rozwiązanie: wykorzystaj kilka dysków

Dane przesyłane równolegle z wielu dysków

Dane zapamiętane w paskach (ang. Stripe).

Bit parzystości na dodatkowym dysku.

Technika nosi nazwę RAID (ang. redundant array of independent drives).

Macierze RAID

Redundant Array of Inexpensive Discs Nadmiarowa Macierz Tanich Dysków (Patterson i wsp., 1988) - ma działać lepiej niż SLED (Single Large Expensive Disc)

Idea: Zbiór (macierz) dysków dzięki specjalnemu kontrolerowi widoczny dla reszty systemu jako jeden “virtualny” dysk.

Virtualny dysk dzielony jest na paski (ang. strips), które są rozpraszane na dyskach tworzących macierz.

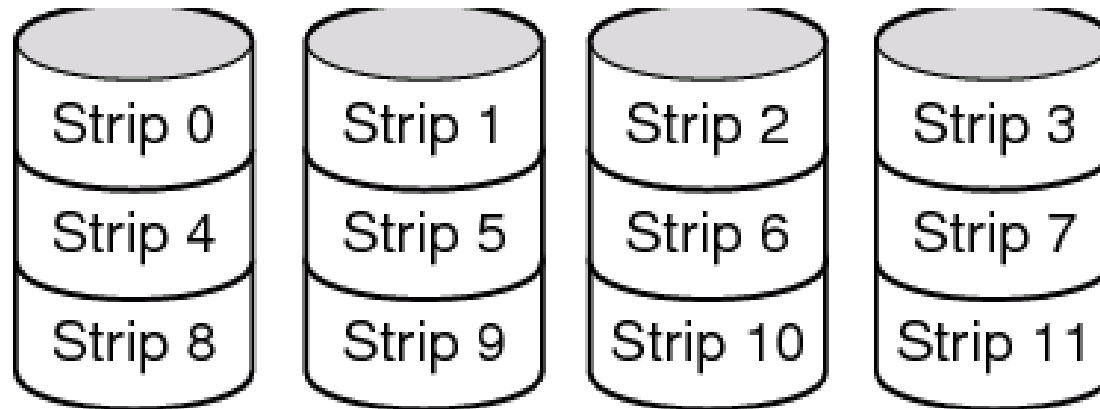
Pasek ma rozmiar k-sektorów.

Redundancja pozwala na zmniejszenie ryzyka awarii.

Ponieważ dyski w macierzy pracują równolegle, żądania mogą być realizowane szybciej.

Komercyjnie dostępne poziomy 0,1, 0+1, 4,5

RAID Level 0



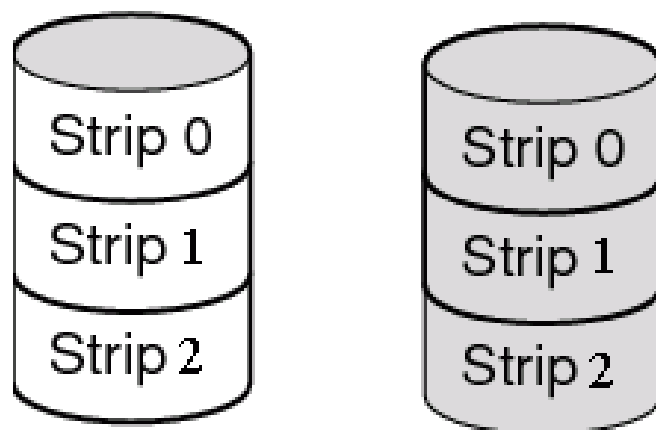
Nie jest to prawdziwa macierz RAID - nie ma redundancji. Paski są rozpraszane równomiernie po wszystkich dyskach - pojemność dysku się nie marnuje.

Dwa żądania mają szansę dotyczyć dwóch niezależnych dysków i być obsłużone równolegle - krótszy czas oczekiwania na żądanie.

Duże żądanie może dotyczyć kilku kolejnych pasków (np. strip2, strip3, strip4) wtedy czas oczekiwania jest taki sam, ale czas transmisji się znacznie skraca - dyski pracują równolegle.

Awaryjność **wzrasta** (i to znacznie - patrz wykład ze statystyki)

RAID Level 1



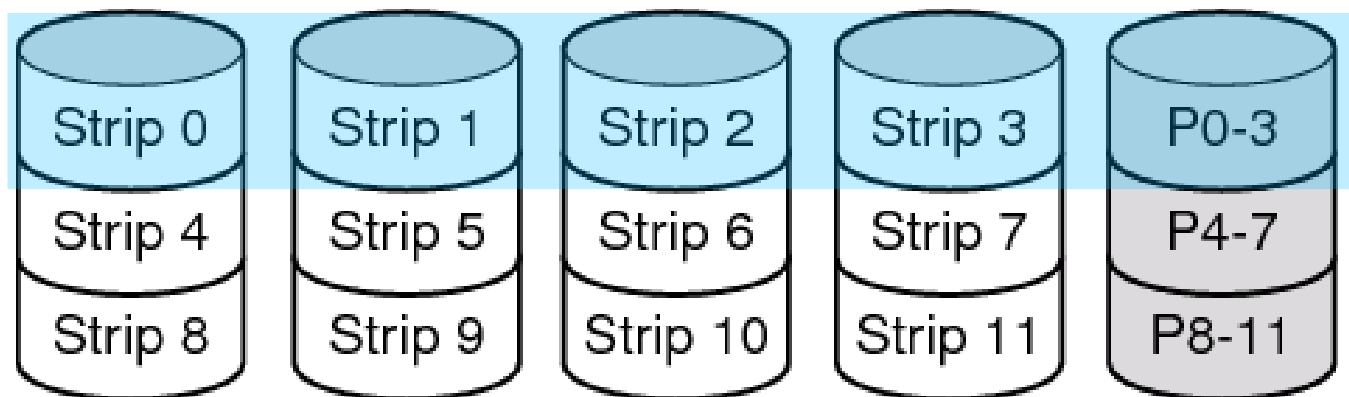
Jeden dysk jest kopią drugiego - odtworzenie stanu macierzy po awarii jest szybkie.

Mamy zapewnioną redundancję, ale kosztem dwukrotnego zmniejszenia pojemności.

Dla żądań odczytu największa możliwa wydajność ze wszystkich konfiguracji RAID (dwa żądania **zawsze** mogą być obsługane równolegle).

Żądania zapisu nie są przyspieszane.

RAID Level 4



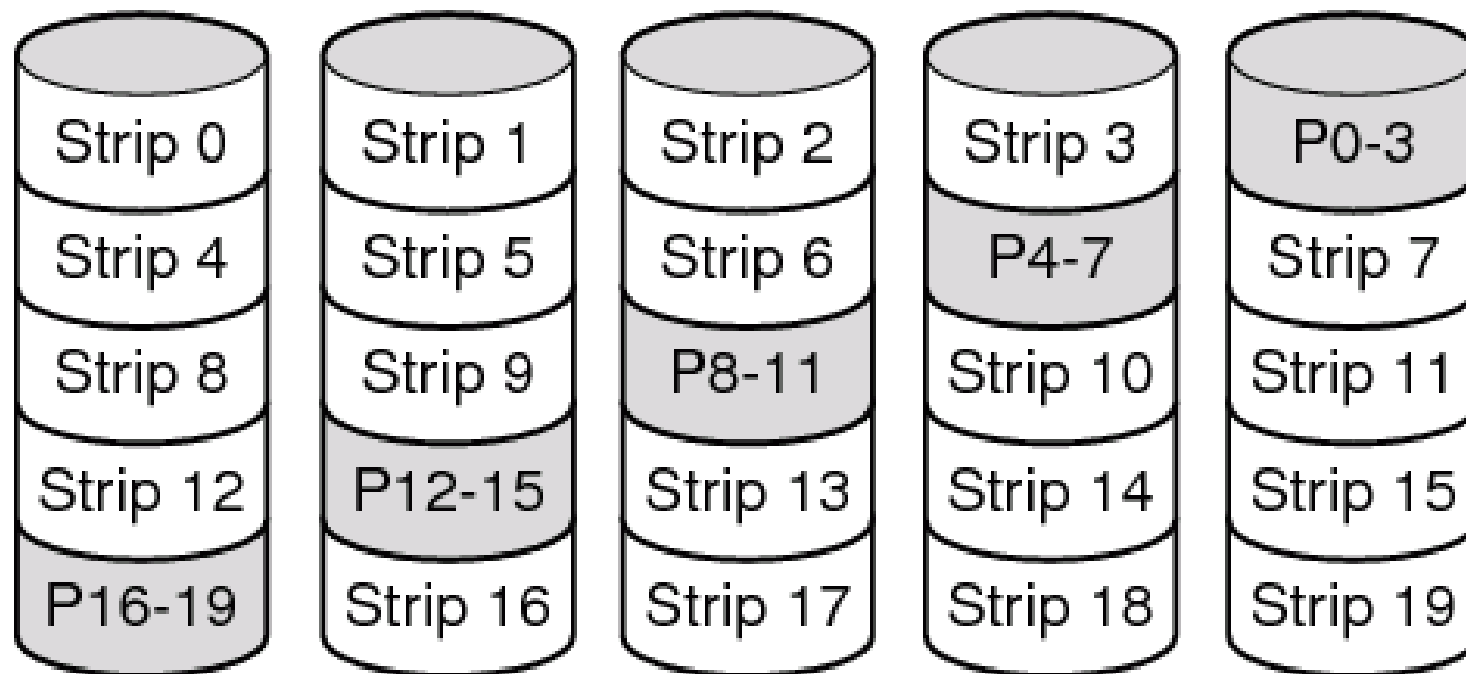
Jeden z dysków przechowuje bity parzystości - w razie awarii któregoś z dysków brakujące dane można odtworzyć na podstawie bitów parzystości i zawartości pozostałych dysków. Odbudowa po awarii macierzy jest czasochłonna.

Bardziej efektywne wykorzystanie pojemności dysków.

Każdy Zapis wymaga ponownego obliczenia bitów parzystości, dzięki czemu dysk przechowujący te bity staje się wąskim gardłem przy zapisach.

Ponadto zapis może wymagać przeprowadzenia **odczytów** z innych dysków w celu obliczenia parzystości - potencjalnie dwa odczyty i dwa zapisy jeżeli dane zajmują część paska - dwukrotne spowolnienie

RAID Level 5



Podobnie, jak RAID Level 4, ale paski z parzystością są równomiernie rozproszone po wszystkich dyskach macierzy, co usuwa wąskie gardło przy zapisach.

RAID 0+1 oraz RAID 1+0

RAID 0+1: dwie macierze RAID 0 (z przeplotem) tworzą macierz RAID 1. Minimum 4 dyski. Lepsza wydajność niż RAID 5, ale pojemność macierzy 50%.

RAID 1+0 (RAID 10). dwie macierze RAID 1 (mirror) tworzą macierz RAID 0 (z przeplotem). Minimum 4 dyski. Pewne zalety w porównaniu z RAID 0+1: np. może przetrwać awarię dwóch dysków.

Schematy RAID 0+1 oraz 1+0 często implementowane programowo (np. większość “kontrolerów RAID” zintegrowanych z płytami głównymi).

Napędy taśmowe

Napęd taśmowy wykorzystuje nośnik wymienny.

Jeżeli chodzi o koszt jednego bitu, napęd taśma jest znacznie tańsza od dysku.

Prędkość transmisji danych jest porównywalna z dyskiem.

Czas dostępu jest bardzo duży, rzędy minut

Wymagane jest przewinięcie taśmy.

Z powyższych powodów napędy taśmowe wykorzystuje się przede wszystkim do archiwizacji danych zapisanych na dysku i tworzenia kopii zapasowych.

W przypadku awarii dysku, kopia zapasowa pozwala na odtworzenie danych

Dostęp danych w dyskowych

System operacyjny umożliwia programom użytkownika, dostęp do dysku na dwa sposoby.

Sposób 1: Dostęp za pomocą systemu plików.

- Organizacja systemu plików (np. drzewo katalogów)

- Prawa dostępu do plików.

- Operacje otwarcia pliku, odczytu, zapisu i zamknięcia.

- Alokacja sektorów dysku dla pliku

Sposób 2: Dostęp surowy (ang. raw):

- Dysk traktowany jest jako tablica sektorów.

- Dostęp surowy często wymaga specjalnych przywilejów.

- Pominięcie warstwy systemu plików pozwala na zwiększenie wydajności.

- Przykład 1: Baza danych Oracle.

- Przykład 2: Program fsck w systemach Unix (naprawiający system plików)

Awaryjność

Napędy z nośnikami wymiennymi są bardziej narażone na awarie.

Nośniki optyczne są mniej awaryjne niż nośniki magnetyczne.

Mniej wrażliwe na zakłócenia wewnętrzne

W napędzie z nośnikami niewymiennym głowica nie kontaktuje się z nośnikiem

Talerz znajdują się w hermetycznie zamkniętej obudowie.

Głowica porusza się na poduszce powietrznej.

Kontakt głowicy z nośnikiem (np. w wyniku wstrząsu, upadku) bardzo często prowadzi do awarii.

Awaria napędu oznacza utratę danych.

W napędzie z nośnikiem wymiennym, po awarii napędu bardzo często nie występuje utrata danych.