

Wstęp do Informatyki - Wykład 5

Przechowywanie danych

Przechowywanie danych

- Przechowywanie danych można rozumieć na dwóch poziomach:
 - sprzętowym
 - logicznym
- Na poziomie sprzętowym mówimy o urządzeniach do przechowywania danych - nośnikach danych.
- Na poziomie logicznym zajmujemy się formatami danych oraz algorytmami ich zapisu i odczytu.

Poziom sprzętowy

- Klasyczny dysk twardy (HDD) - pamięć masowa
- Wykorzystuje nośnik magnetyczny.
- Film przedstawiający pracę dysku twardego:
<https://upload.wikimedia.org/wikipedia/commons/4/44/HardDisk2.ogv>
- Nośniki magnetyczne umożliwiają odzyskiwanie danych po nadpisaniu.

Poziom sprzętowy

- Pamięć typu Flash
- Wykorzystuje stronicowanie - strony to ustalone z góry zbiory komórek pamięci
- Każda komórka pamięci jest oparta na technologii MLC (Multi level cell).
Oznacza to, że w przeciwieństwie do SLC (Single level cell), w jednej komórce można zapisać dane na dwóch bitach, zamiast jednego.
- Pozwala to na gęstszy zapis, który jednak jest bardziej narażony na błędy zapisu i odczytu.

Poziom sprzętowy

- Nośniki CD-ROM, DVD, Blu-Ray
- Fizyczne wyżłobienia w wierzchniej warstwie płyty, odczytywane przez czytnik optyczny.
- Istnieją dyski DVD i BD-R, których deklarowana trwałość wynosi 100,150, a nawet 1000 lat.

Poziom sprzętowy

Taśma magnetyczna. Cartridge produkcji firmy IBM:



Poziom logiczny

- Na poziomie logicznym mówimy m.in. o **partycjach, formatowaniu** oraz o **systemach plików**
- Partycja jest wydzielonym obszarem dysku twardego.
- Formatowanie jest procesem budowania systemu plików na partycji.

Poziom logiczny

- System plików to zestaw formatów i algorytmów, służący do zapisywania i odczytywania plików w pamięci masowej.
- Nowoczesne systemy plików są wyposażone między innymi w mechanizmy odzyskiwania i naprawy plików.
- Najpopularniejsze systemy plików używane w dzisiejszych komputerach:
 - NTFS (systemy z rodziny Windows od Win NT)
 - FAT32 (pamięci typu Flash)
 - EXT3, EXT4 (systemy Unixowe)

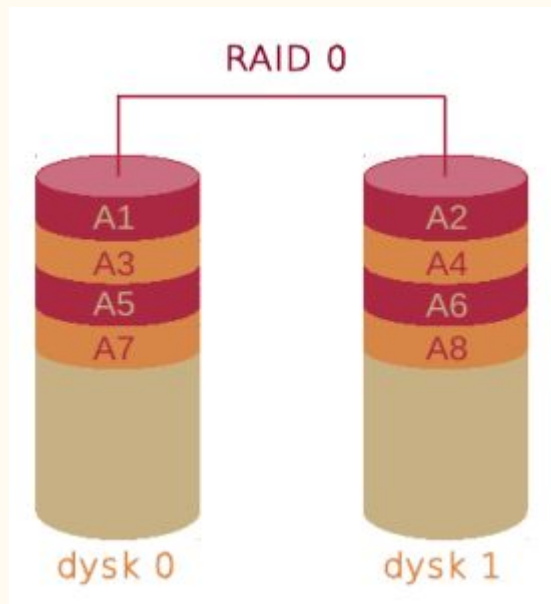
Kopie zapasowe

- Aby zminimalizować prawdopodobieństwo utraty danych stosuje się kopie zapasowe.
- Są one wykonywane na żądanie.
- Dane w kopii zapasowej nie są później zmieniane.

Kopie zapasowe vs RAID

- RAID (**R**edundant **A**rray of **I**ndependent **D**isks) to wykorzystanie w zestawie komputerowym więcej niż jednego dysku twardego do przechowywania tych samych danych.
- Jest to technika, która chroni przed awariami dysków.
- **Nie jest** to kopia zapasowa.

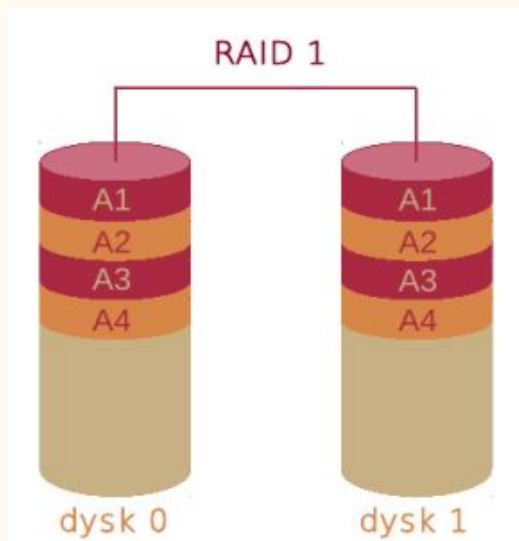
RAID 0



- Wykorzystanie dwóch takich samych dysków.
- Dane są przeplatane pomiędzy dyskami.
- Brak odporności na awarie dysków.
- Szybkość zapisu/odczytu.

źródło: opracowanie data center kylos.pl

RAID 1



- Dane są zapisywane równoległe na dwóch dyskach.
- Odporność na awarię jednego z nich.

źródło: opracowanie data center kylos.pl

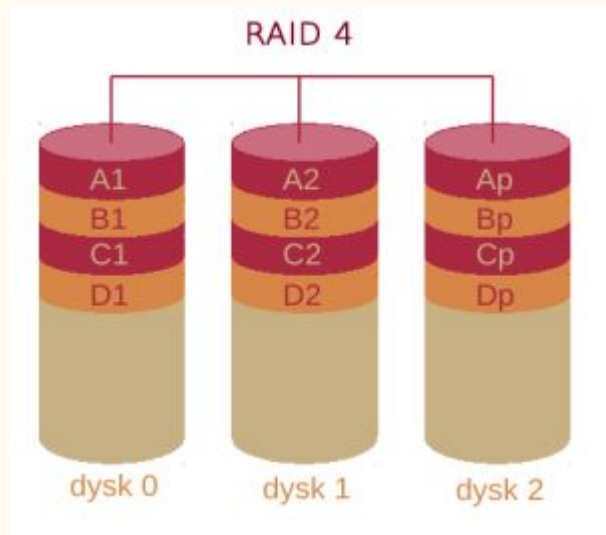
RAID 2

- Jeden z dysków w macierzy zarezerwowany na informacje dotyczące korekcji błędów metodą kodów Hamminga.
- Odporność na awarię jednego spośród pozostałych dysków.
- Obecnie nieużywany.

RAID 3

- Zasada działania bardzo podobna do RAID 2.
- Różnica - w RAID 3 dane dzielone są na poziomie bajtów, a nie bitów.
- Obecnie nieużywany.

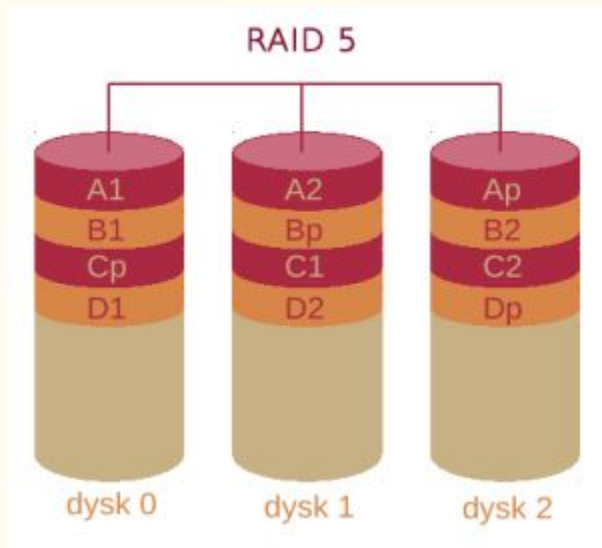
RAID 4



- Podobnie jak RAID 2 i RAID 3, wymaga minimum 3 dysków.
- Dane zapisywane są równoległe na dwóch dyskach.
- Trzeci dysk zawiera informacje o tzw. parzystości.
- Możliwość odtworzenia danych w przypadku awarii dowolnego z dysków.
- Obecnie wyparty przez RAID 5 i RAID 6

źródło: opracowanie data center kylos.pl

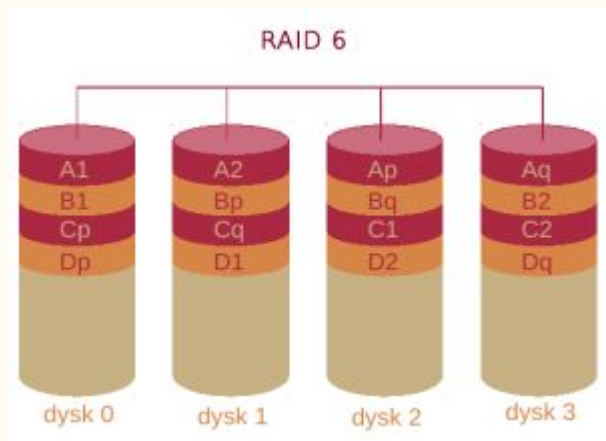
RAID 5



- Podobnie jak w RAID 4, ale informacja o parzystości zapisywana jest na różnych dyskach.
- Odporność na awarię dowolnego dysku.

źródło: opracowanie data center kylos.pl

RAID 6



- Podobnie jak w RAID 5, ale informacja o parzystości zapisywana jest podwójnie na różnych dyskach.
- Odporność na awarię dwóch dysków, ale utrata pojemności i szybkości.

źródło: opracowanie data center kylos.pl

Poziom logiczny - XML

- Na poziomie logicznym podstawowym pojęciem związanym z przechowywaniem danych jest plik.
- Występuje wiele formatów plików, każdy z nich jest specjalizowany do przechowywania innych danych.
- Jednym z formatów służących do przechowywania danych jest XML.

Poziom logiczny - XML

- XML (eXtensible Markup Language) to język opisu danych (nie jest to język programowania!)
- Podobnie jak HTML jest językiem typu markup, to znaczy wykorzystuje zestaw znaczników, tzw. tagów.
- Dzięki tagom definiuje się węzły, które posiadają zawartość oraz właściwości.

Poziom logiczny - XML

```
1 <rozkład>
2   <miejscowosc rozmiar="542000">Poznań</miejscowosc>
3   <linia typ="tramwaj">
4     <numer>5</numer>
5     <poczatek>Górczyn</poczatek>
6     <koniec>Miłostowo</koniec>
7   </linia>
8   <linia typ="autobus">
9     <numer>105</numer>
10    <poczatek>Rondo Rataje</poczatek>
11    <koniec>Piątkowo</koniec>
12  </linia>
13 </rozkład>
```

źródło: J. Kobusiński: "XML w przykładach"

Elementy XML

- Plik XML składa się z:
 - Węzłów (np. `<linia>`)
 - Atrybutów (np. `typ="tramwaj"`)
 - Encji - znaków specjalnych (np. `>`; `<`;))
 - PCDATA - dane tekstowe parsowalne, tj. encje są zamieniane na symbole (np. `>` staje się `>`)
 - CDATA - dane tekstowe nieparsowalne

XML - DTD

- Aby ustalić konkretny format danych w XML stosuje się deklaracje DTD.
- DTD opisuje, jaki węzeł powinien być korzeniem dokumentu XML oraz jakie inne węzły powinien zawierać.

XML - DTD

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
```

źródło: https://www.w3schools.com/xml/xml_dtd_intro.asp

XML - DTD

- !DOCTYPE note: korzeniem dokumentu jest węzeł “note”
- !ELEMENT note: definiuje zawartość węzła note: "to,from,heading,body"
- !ELEMENT to: definiuje element to jako "#PCDATA"
- !ELEMENT from: definiuje element to jako "#PCDATA"
- !ELEMENT heading: definiuje element to jako "#PCDATA"
- !ELEMENT body: definiuje element to jako "#PCDATA"

XML - DTD

Deklarowanie zawartości elementu:

```
<!ELEMENT element-name (element-content)>
```

Na przykład:

```
<!ELEMENT note (to,from,heading,body)>
```

XML - DTD

Powtarzanie zawartości 1 lub więcej razy:

```
<!ELEMENT element-name (element-content+)>
```

Na przykład:

```
<!ELEMENT rozklad (miejscowosc,linia+)>
```

XML - DTD

Powtarzanie zawartości 0 lub więcej razy:

```
<!ELEMENT element-name (element-content*)>
```

Na przykład:

```
<!ELEMENT rozklad (miejscowosc,linia*)>
```

dopuszcza, że w mieście nie ma żadnych linii.

XML - DTD

Deklarowanie atrybutów:

```
<!ATTLIST element-name attribute-name attribute-type attribute-value>
```

Na przykład w DTD:

```
<!ATTLIST payment type CDATA "check">
```

w XML:

```
<payment type="check" />
```

Dziękuję za uwagę!