

Systemy Tolerujące Błędy

Projekt

Temat E: Symulacja działania macierzy dyskowych w modelu RAID0, RAID1, RAID3. Obliczanie bitów parzystości (dla RAID3). Symulowane dyski o pojemności co najmniej 64 x 32 bity.

Zespół:

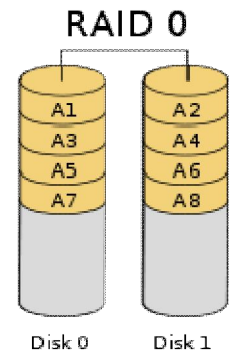
Jacek Donchor
Grzegorz Tkacz
Paweł Wacnik

1. Opis teoretyczny działania macierzy dyskowych RAID0, RAID1 i RAID3.

RAID (*Redundant Array of Independent Disks*, Nadmiarowa macierz niezależnych dysków) polega na współpracy dwóch lub więcej dysków twardych w taki sposób, aby zapewnić dodatkowe możliwości, nieosiągalne przy użyciu jednego dysku.

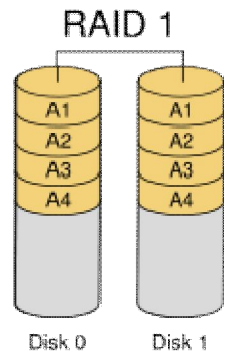
RAID 0

Polega na połączeniu ze sobą dwóch lub więcej dysków fizycznych tak, aby były widziane jako jeden dysk logiczny. Powstała w ten sposób przestrzeń ma rozmiar taki jak $N \times \text{rozmiar najmniejszego z dysków}$. Dane są przeplecione pomiędzy dyskami. Dzięki temu uzyskujemy znaczne przyspieszenie operacji zapisu i odczytu ze względu na zrównoleglenie tych operacji na wszystkie dyski w macierzy. RAID0 charakteryzuje się brakiem odporności na awarie dysków tzn. gdy jeden dysk zostanie uszkodzony to dane zawarte w całej macierzy są tracone.



RAID 1 (lustrzany)

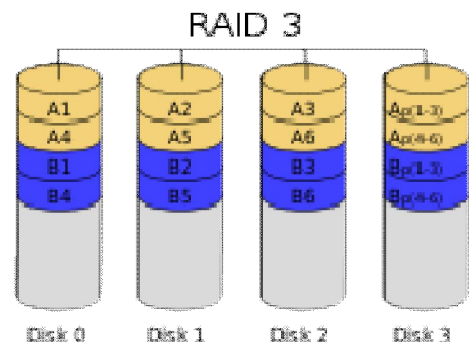
Polega na replikacji pracy dwóch lub więcej dysków fizycznych. Powstała przestrzeń ma rozmiar najmniejszego nośnika. RAID 1 jest zwany również lustrzanym (ang. mirroring). Charakteryzuje się odpornością na awarię $N - 1$ dysków przy N -dyskowej macierzy.



RAID 3

Dane składowane są na $N-1$ dyskach. Ostatni dysk służy do przechowywania sum kontrolnych. Działa jak striping (RAID 0), ale w macierzy jest dodatkowy dysk, na którym zapisywane są kody parzystości obliczane przez specjalny układ. Ta macierz jest odporna na awarie jednego dysku.

Kontrola parzystości polega na wykonaniu operacji *xor* dla pasków na dyskach 0 $N-1$ i zapisy wyniku tej operacji na dysku N . Gdy zostanie uszkodzony któryś z dysków dane są odzyskiwane poprzez wykonanie operacji *xor* na dla pasków na pozostałych dyskach i zapisu ich do dysku wstawionego na miejsce uszkodzonego dysku.



2. Opis implementacji z fragmentami kodu źródłowego.

Projekt został zrealizowany w języku C# w Visual Studio C# 2010 Express dla .NET Framework 3.5. Jest to aplikacja okienkowa.

Aplikacja składa się z 8 klas. Klasą „główną” jest *RaiDSyM.cs* dziedzicząca po klasie *Form.cs*. W tej klasie znajdują się metody odpowiedzialne za :

- odświeżanie okna aplikacji - *RaiDSyM_Paint(object sender, PaintEventArgs e)*
- częstotliwość odświeżania - *timer_Tick(object sender, EventArgs e)*
 - zmienna *sleepTime*
- wybór typu RAID - *btnPoziomRaid_Click(object sender, EventArgs e)*
- wstrzykiwanie błędów - *btnWstrzyknij1_Click(object sender, EventArgs e)*

Klasa *Przeptyw.cs* pobiera dane z pliku *Wejscie.xml* odpowiednio sformatowane w klasie *Load.cs*. Metoda *Run()* odpowiada za działanie aplikacji. Jest to metoda włączona jako wątek , który jest usypiany na czas zadeklarowany w zmiennej *sleepTime* w klasie *RaiDSyM.cs*, w tym czasie jest wykonywana klasa *RaiDSyM.cs*. W niej wywoływane są błędy i odpowiednie komunikaty.

Klasa *Load.cs* pobiera dane z pliku *wejscie.xml* (część plik po prawej) i wstawia je do listy paskiWejsciove przechowującej tablice typu byte. Lista ta jest przekazywana do klasy *Przeptyw.cs* w której elementy tej

```
<?xml version="1.0" encoding="utf-8" ?>
<root>
  <pasek>00000000000000000000000000000000</pasek>
  <pasek>00000000000000000000000000000001</pasek>
```

klasy są „podpinane” do odpowiednich pasków. W tej klasie jest również metoda *zapis(String komunikat)* odpowiedzialna za tworzenie raportu z działania aplikacji. Zapisuje ona do pliku *wyjście.txt* komunikaty o: zapisie danych do odpowiedniego paska i dysku, wystąpieniu błędu, naprawie błędu itp.

```
public void zapis(String komunikat)
{
    StreamWriter SW;
    SW = File.AppendText("Wyjście.txt");
    String text = string.Format("{0} {1}", DateTime.Now, komunikat);
    SW.WriteLine(text);
    SW.Close();
}
```

Przykładowy raport w pliku *wyjście.txt*

```
2010-11-13 17:49:47 Program Włączono
2010-11-13 17:49:48 Do dysku 0 zapisano pasek 0, zawartość: 00000000000000000000000000000000
2010-11-13 17:49:50 Do dysku 1 zapisano pasek 0, zawartość: 00000000000000000000000000000000
2010-11-13 17:49:51 Do dysku 0 zapisano pasek 1, zawartość: 00000000000000000000000000000001
2010-11-13 17:49:53 Do dysku 1 zapisano pasek 1, zawartość: 00000000000000000000000000000001
2010-11-13 17:49:54 Błąd został naprawiony
2010-11-13 17:49:55 Wystąpił błąd w Dysku 0, w pasku 1, Błąd zostanie niebawem naprawiony poprzez skopiowania paska 1 z Dysku 1 do Dysku 0
2010-11-13 17:49:56 Do dysku 0 zapisano pasek 2, zawartość: 00000000000000000000000000000010
```

Klasa *Dysk.cs* odpowiada za rysowania dysków w zależności od wybranego typu RAID.
Klasa *Pasek.cs* odpowiada za rysowanie pasków. Wykorzystywana jest przez sparametryzowany konstruktor:

```

public Pasek(int x,int y, int nrPasku, Color kolor)
{
    this.kolor=kolor;
    this.x = x;
    this.y = y;
    this.nrPasku = nrPasku;
}

```

Parametry tego konstruktora określają pozycje na ekranie (x,y), nr pasku, i kolor tego pasku. Paski mają zazwyczaj czarny kolor, przypadku wystąpienia błędu przyjmują kolor czerwony. W klasie *Przeplwy* tworzona jest lista typu *Pasek* która dodaje paski w metodzie *Run()*. W klasie *RaiDSyM* w metodzie *RaiDSyM_Paint(object sender, PaintEventArgs e)* lista z paskami jest „rysowana”.

Dla RAID3 wykorzystywana jest klasa *KontrolaParzystości.cs* w której metoda *liczBityParzystości(byte[] pasekZDysk1, byte[] pasekZDysk2, byte[] pasekZDysk3)* typu tablica bajtów. Służy ona do liczenia zawartości pasków zapisywanych w dysku 3. Gdy wystąpi błąd którymś dysku to również wykorzystywana jest ta metoda do odzyskania danych w tym dysku poprzez wprowadzenie do parametrów tej funkcji zawartości pasków z pozostałych 3 dysków.

Klasa *Blad.cs* odpowiada za wstrzykiwania błędów do układu poprzez:

- zaznaczenie na czerwono wybranego paska
- zapis do raportu o szczegółach błędu
- wyświetlenie na ekranie *MessageBox* z odpowiednim komunikatem

3. Przykłady działania ze zrzutami ekranu.

- RAID0

The screenshot shows a software interface for simulating RAID configurations. On the left, there's a control panel titled "Błędy" (Errors) with input fields for "Nr Dysku" (Disk Number) set to 1 and "Nr Paska" (Sector Number) set to 11. A button labeled "Wstrzyknij Błąd" (Inject Error) is visible. Below this, a text box states: "Dla Raid0 każdy wstrzyknięty błąd przekracza zakres tolerancji układu." (For Raid0, every injected error exceeds the system's tolerance range).

In the center, there are two 4x4 grids representing disks. The first grid, labeled "Dysk 0", contains data sectors 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20. The second grid, labeled "Dysk 1", contains data sectors 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, and 21. The sector containing the value 11 in Disk 1 is highlighted in red, indicating an error.

A Windows-style message box is overlaid on the interface. Its title is "Błąd przekraczający zakres tolerancji symulowanego układu" (Error exceeding the tolerance range of the simulated system). The message text reads: "Wystąpił błąd w Dysku 1, w pasku 11, Zapis został wstrzymany, ponieważ ten błąd uniemożliwia odczyt zapisanych danych w Raid0 (dalszy zapis nie ma sensu)" (An error occurred in Disk 1, in sector 11, The write was blocked, because this error makes it impossible to read the saved data in Raid0 (further write is not sense)). There is an "OK" button at the bottom right of the message box.

- RAID1

-błąd tolerowalny przez układ

Błędy

☒ błąd tolerowalny przez układ

☐ błąd przekraczający zakres tolerancji symulowanego układu

Nr Dysku:

Nr Paska:

Wstrzyknij Błąd

				10							
0	1	2	3					0	1	2	3
4	5	6									
8	9	10	11								
12	13	14	15								
16	17	18	19								
20	21	22	23								
24	25										

Błąd naprawialny przez układ, nie przekraczający zakresu tolerancji symulowan...

Wystąpił błąd w Dysku 0, w pasku 10. Błąd zostanie niebawem naprawiony poprzez skopiowania paska 10 z Dysku 1 do Dysku 0

OK

- błąd przekraczający zakres tolerancji symulowanego układu

Błędy

☐ błąd tolerowalny przez układ

☒ błąd przekraczający zakres tolerancji symulowanego układu

Nr Dysku:

Nr Paska:

Wstrzyknij Błąd

Dla Raid1 można wstrzykiwać błędy tolerowane przez układ, lub błędy przekraczające zakres tolerancji

Błąd przekraczający zakres symulowanego układu

Wystąpił błąd przekraczający zakres tolerancji układu, tzn. Dwa paski o tym samym numerze zostały uszkodzone i nie można odzyskać danych z tych pasków dalszy zapis został wstrzymany

OK

28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	

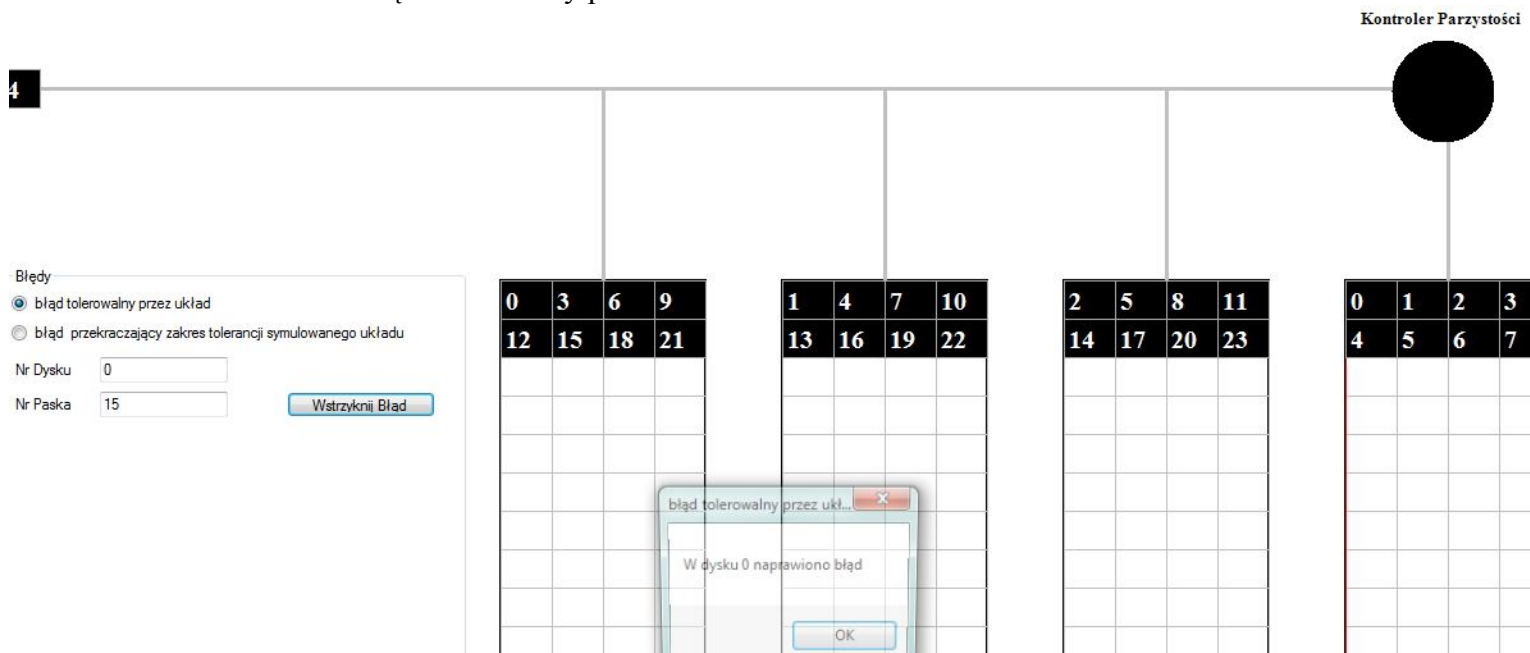
Dysk 0

28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	

Dysk 1

- **RAID3**

- błąd tolerowalny przez układ



- błąd przekraczający zakres tolerancji symulowanego układu

