**SPRAWOZDANIE**

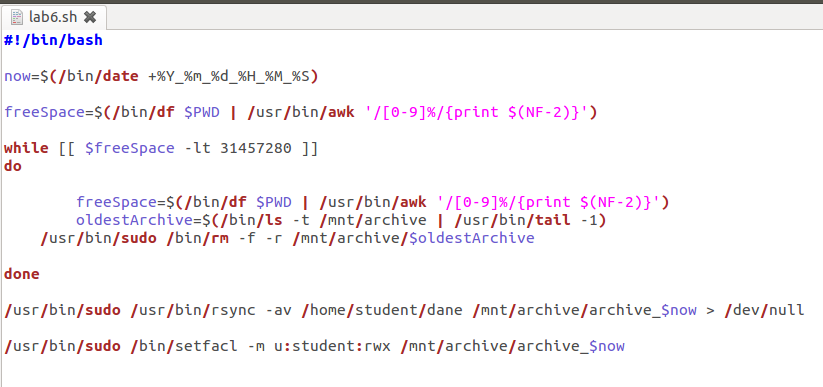
**BEZPIECZEŃSTWO SYSTEMÓW INFORMATYCZNYCH**

|  |  |  |
| --- | --- | --- |
| **Osoba wykonująca** | **Grupa** | **Data** |
| Michał J. Sidor | 5.5/9 | 14.01.2018r. |
| **Uczelnia** | **Wydział** | **Kierunek** |
| Politechnika Lubelska | Elektrotechniki i Informatyki | Informatyka I. stopnia, stacjonarne |
| **Temat** | | |
| **LABORATORIUM NR 6 i 7.**  BEZPIECZEŃSTWO DANYCH | | |
|

**Zadanie 6.1**

P1. W sprawozdaniu umieść treść skryptu, wpis z cron-a oraz zrzut ekranu prezentujący efekt działania skryptu. Odpowiedz na pytanie: Dlaczego skrypt wywoływany przez cron może wykonać kopię katalogu użytkownika, skoro użytkownik nie jest zalogowany?

Treść skryptu:



#!/bin/bash

now=$(/bin/date +%Y\_%m\_%d\_%H\_%M\_%S)

freeSpace=$(/bin/df $PWD | /usr/bin/awk '/[0-9]%/{print $(NF-2)}')

while [[ $freeSpace -lt 31457280 ]]

do

freeSpace=$(/bin/df $PWD | /usr/bin/awk '/[0-9]%/{print $(NF-2)}')

oldestArchive=$(/bin/ls -t /mnt/archive | /usr/bin/tail -1)

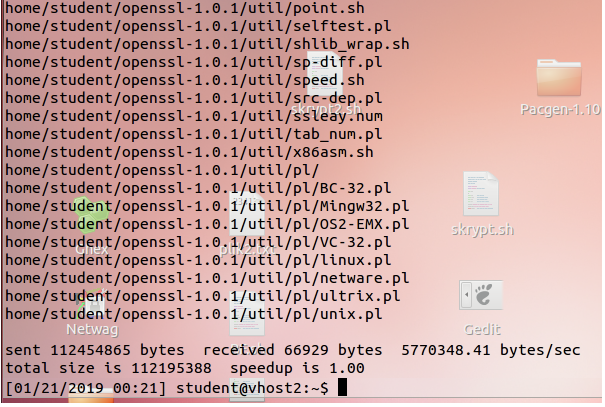
/usr/bin/sudo /bin/rm -f -r /mnt/archive/$oldestArchive

done

/usr/bin/sudo /usr/bin/rsync -av /home/student/dane /mnt/archive/archive\_$now > /dev/null

/usr/bin/sudo /bin/setfacl -m u:student:rwx /mnt/archive/archive\_$now

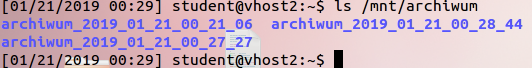
Działanie skryptu:



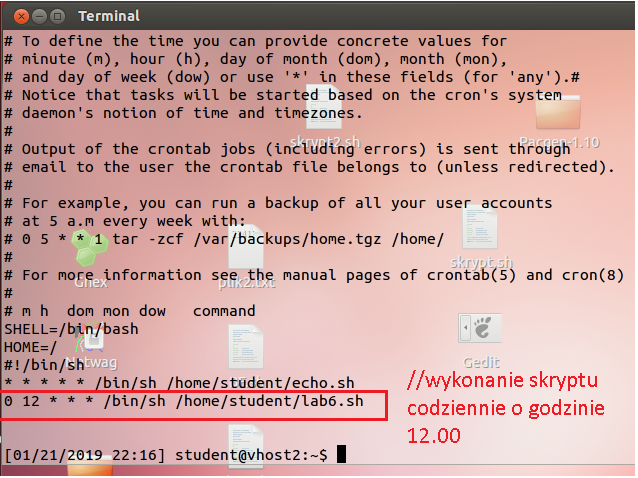
Utworzone archiwum:



Kolejne archiwa:



Wpis crontab:

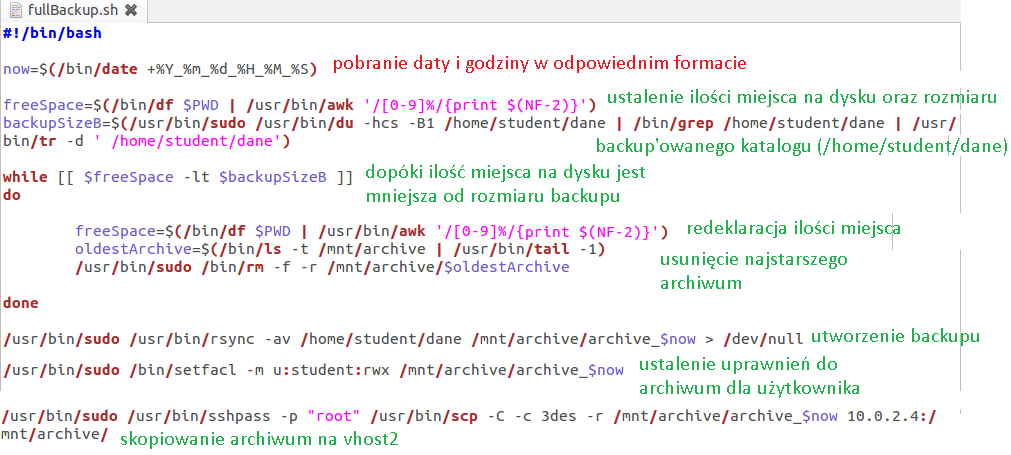


Cron może działać i wykonywać skrypty niezależnie od zalogowania lub nie jakichkolwiek użytkowników. Działa on przez wykorzystanie swojego “daemon’a” (działającego w oddzielnej powłoce systemowej), który sprawdza crontaby poszczególnych użytkowników i wykonuje zawarte w nich polecenia. Daemon Crona uzyskuje uprawnienia dostępu do danych konkretnego użytkownika na podstawie konkretnego crontabu (crontaby, którymi się posługujemy - wywoływane poleceniem crontab -e - są crontabami dla poszczególnych użytkowników - w przeciwieństwie do crontaba ogólnego/systemowego zawartego w katalogu /etc/crontab).

**Zadanie 6.4**

P2. W sprawozdaniu umieść treść niezbędnych skryptów, zrzuty ekranu prezentujące działanie opracowanego rozwiązania, porównanie czasów wykonania archiwum z niniejszego zadania oraz z zadania 1. Wyjaśnij dlaczego niniejsza kopia jest kopia przyrostową.

Skrypt wykonujący pełną kopię zapasową wraz z kopiowaniem jej na vhost2 (z zastosowaniem kompresji oraz szyfrowania 3DES) z komentarzem (zielony i czerwony):



#!/bin/bash

now=$(/bin/date +%Y\_%m\_%d\_%H\_%M\_%S)

freeSpace=$(/bin/df $PWD | /usr/bin/awk '/[0-9]%/{print $(NF-2)}')

backupSizeB=$(/usr/bin/sudo /usr/bin/du -hcs -B1 /home/student/dane | /bin/grep /home/student/dane | /usr/bin/tr -d ' /home/student/dane')

while [[ $freeSpace -lt $backupSizeB ]]

do

freeSpace=$(/bin/df $PWD | /usr/bin/awk '/[0-9]%/{print $(NF-2)}')

oldestArchive=$(/bin/ls -t /mnt/archive | /usr/bin/tail -1)

/usr/bin/sudo /bin/rm -f -r /mnt/archive/$oldestArchive

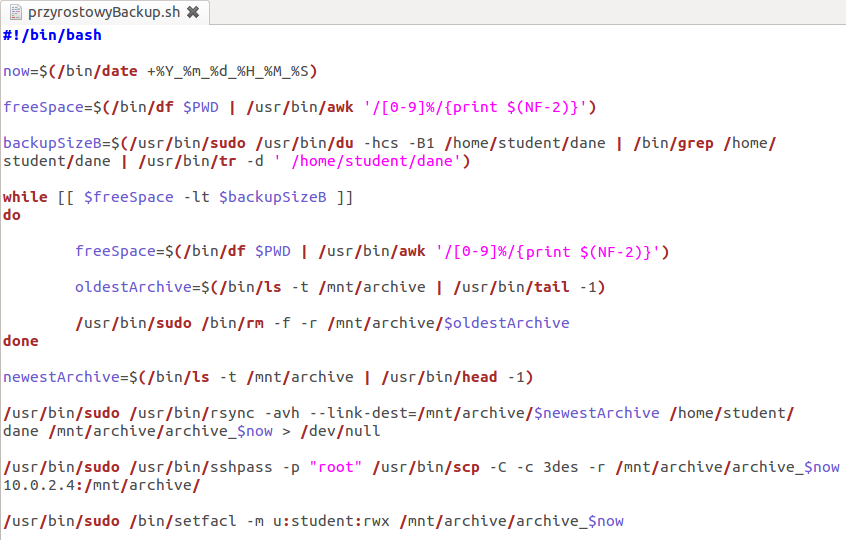
done

/usr/bin/sudo /usr/bin/rsync -av /home/student/dane /mnt/archive/archive\_$now > /dev/null

/usr/bin/sudo /usr/bin/sshpass -p "root" /usr/bin/scp -C -c 3des -r /mnt/archive/archive\_$now 10.0.2.4:/mnt/archive/

/usr/bin/sudo /bin/setfacl -m u:student:rwx /mnt/archive/archive\_$now

Skrypt wykonujący kopię przyrostową wraz kopiowaniem jej na vhost2 (z zastosowaniem kompresji oraz szyfrowania 3DES):

****

#!/bin/bash

now=$(/bin/date +%Y\_%m\_%d\_%H\_%M\_%S)

freeSpace=$(/bin/df $PWD | /usr/bin/awk '/[0-9]%/{print $(NF-2)}')

backupSizeB=$(/usr/bin/sudo /usr/bin/du -hcs -B1 /home/student/dane | /bin/grep /home/student/dane | /usr/bin/tr -d ' /home/student/dane')

while [[ $freeSpace -lt $backupSizeB ]]

do

freeSpace=$(/bin/df $PWD | /usr/bin/awk '/[0-9]%/{print $(NF-2)}')

oldestArchive=$(/bin/ls -t /mnt/archive | /usr/bin/tail -1)

/usr/bin/sudo /bin/rm -f -r /mnt/archive/$oldestArchive

done

newestArchive=$(/bin/ls -t /mnt/archive | /usr/bin/head -1)

/usr/bin/sudo /usr/bin/rsync -avh --link-dest=/mnt/archive/$newestArchive /home/student/dane /mnt/archive/archive\_$now > /dev/null

/usr/bin/sudo /usr/bin/sshpass -p "root" /usr/bin/scp -C -c 3des -r /mnt/archive/archive\_$now 10.0.2.4:/mnt/archive/

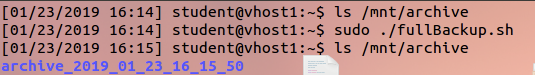
/usr/bin/sudo /bin/setfacl -m u:student:rwx /mnt/archive/archive\_$now

Wpis z crontab’a uruchamiający skrypt tworzący pełną kopię w każdy piątek o 16.30 (5 dzień tygodnia) oraz skrypt tworzący kopię przyrostową w dni robocze poza piątkiem (1-4 = poniedziałek do czwartku) o 16.30.

****

Działanie skryptów:

1. fullBackup.sh (wykonanie pełnej kopii zapasowej)





2. przyrostowyBackup.sh (wykonanie przyrostowej kopii zapasowej):





Możemy przewidywać, że wykonanie pełnych kopii zajmie dłużej niż w ćwiczeniu pierwszym, ponieważ w tym przypadku, oprócz samego utworzenia archiwum, odbywa się jeszcze przesył jego kopii na drugą maszynę wirtualną, podczas którego dane zostają szyfrowane algorytmem 3DES oraz kompresowane, co niewątpliwie czyni cały proces bardziej złożonym obliczeniowo, a więc - jego wykonanie zajmie dłużej.

*Wyniki:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Rodzaj backup’u** | **Pełny (zad.4)** | **Przyrostowy (zad.4)** | **Pełny (zad.1)** |
| **Czas wykonania (user+sys) [s]** | 0,060 | 0,044 | 0,028 |
| 0,076 | 0,044 | 0,036 |
| 0,040 | 0,040 | 0,020 |
| 0,056 | 0,048 | 0,028 |
| 0,048 | 0,040 | 0,028 |
| 0,040 | 0,040 | 0,028 |
| 0,064 | 0,056 | 0,032 |
| 0,048 | 0,052 | 0,020 |
| 0,040 | 0,056 | 0,020 |
| 0,040 | 0,056 | 0,020 |
| **Czas średni** | **0,0512** | **0,0476** | **0,026** |

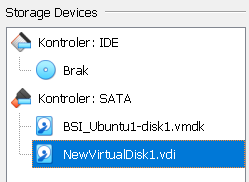
Zgodnie z oczekiwaniami - wykonanie kopii z przesłaniem jej duplikatu na drugą maszyną wirtualną (z zastosowaniem szyfrowania oraz kompresji podczas przesyłu) zajęło znacznie więcej czasu, niż utworzenie kopii lokalnie. Utworzenie kopii przyrostowej odbyło się zauważalnie, choć nie - znacząco - szybciej niż wykonanie kopii pełnej.

Kopia przyrostowa różni się od kopii pełnej tym, że nie są kopiowane wszystkie pliki z danego katalogu, a jedynie te, które zostały zmodyfikowane lub dodane od czasu utworzenia ostatniej kopii zapasowej.

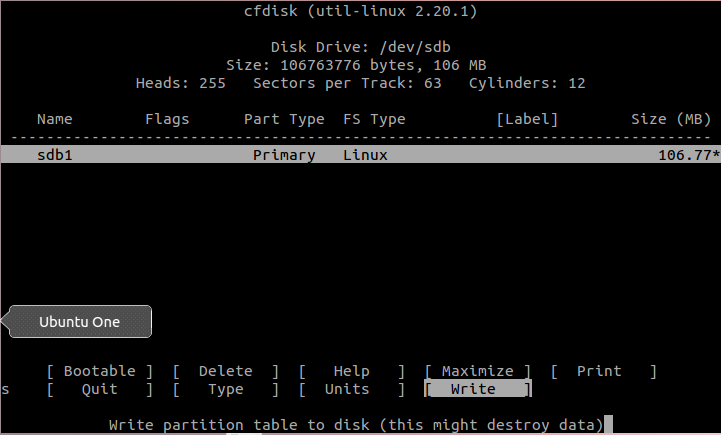
**Zadanie 7.2**

P.3 Udokumentuj wykonanie zadania.

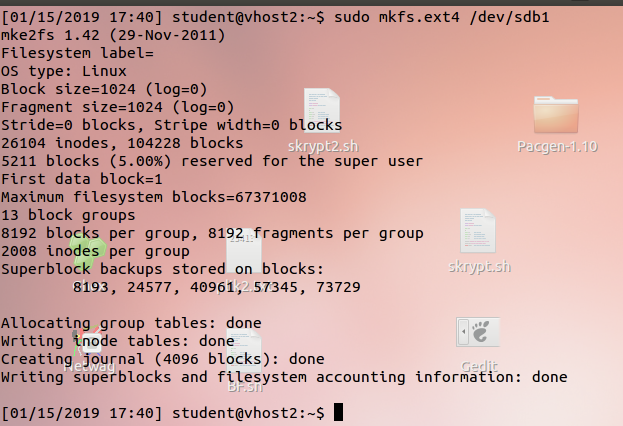
1. Dodanie dysku do systemu



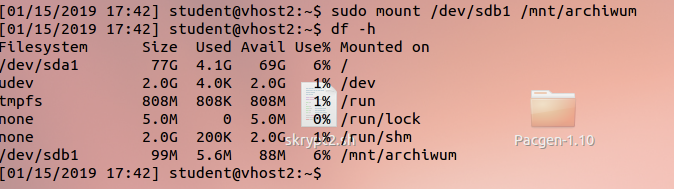
b. Tworzenie partycji na nowym dysku



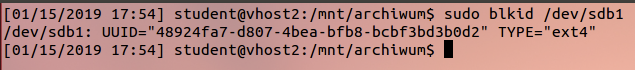
c. Ustawienie systemu plików na nowej partycji na .ext4

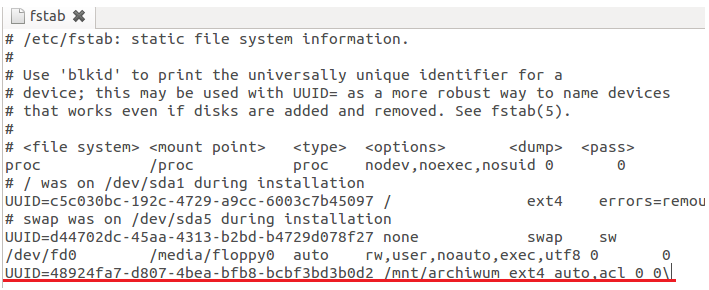


d. Zamontowanie partycji w /mnt/archiwum



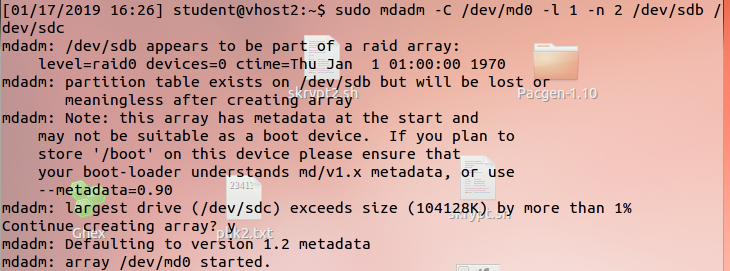
e. Dodanie wpisu do pliku fstab (zapewnienie automatycznego montowania)



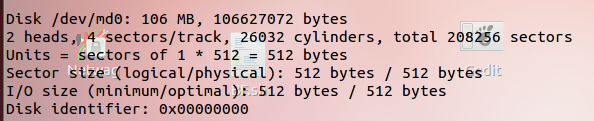


2.

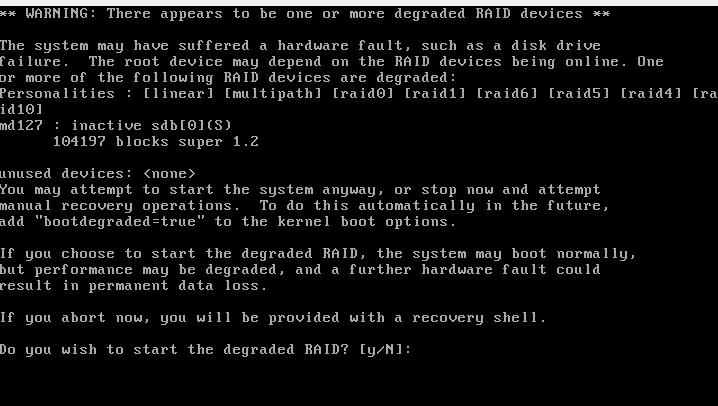
1. Utworzenie macierzy RAID1



Wynik polecenia fdisk -l dla utworzonej macierzy



3. Komunikat o awarii macierzy przy uruchamianiu systemu

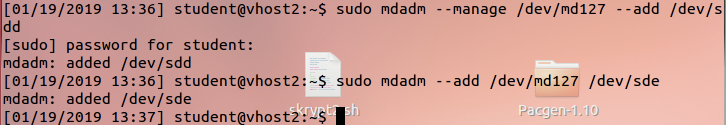


4. Dodanie nowego dysku do zdegenerowanej macierzy 

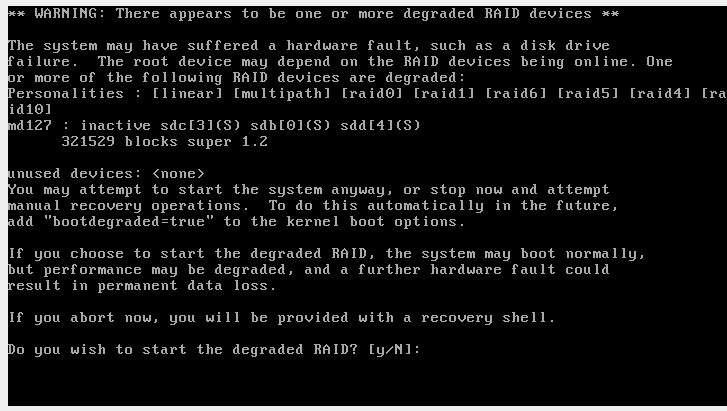
Po naprawieniu macierzy - komunikat o jej uszkodzeniu znika.

5.

a. Migracja macierzy RAID1 do RAID5

b. Dodanie dwóch dodatkowych dysków do macierzy 

c. Zwiększenie rozmiaru woluminu 

6. Po usunięciu jednego z dysków macierzy - ulega ona degeneracji. Możemy jednak wciąż uzyskać dostęp do danych, które były na niej przechowywane. 

P.3 Odpowiedz na pytanie co się stanie (w poszczególnych wersjach RAID [0,1,5,6]) jeśli usuniemy z macierzy jeden dysk. Co się stanie jeśli usuniemy kolejny (drugi) dysk.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **RAID0** | **RAID1** | **RAID5** | **RAID6** |
| **jeden dysk** | utrata wszystkich danych | dane nie zostają utracone, są one dostępne z poziomu dysku zapasowego | zachowujemy dostęp do wszystkich danych, nawet podczas wymiany uszkodzonego dysku | zachowujemy dostęp do wszystkich danych, nawet podczas wymiany uszkodzonych dysków |
| **dwa dyski** | utrata wszystkich danych | utrata wszystkich danych | utrata wszystkich danych |

Powyższe wyniki dla poszczególnych wersji macierzy RAID są ściśle związane z zastosowanymi w nich technologiami.

W macierzy RAID0 przy awarii jednego lub większej ilości dysków utracimy całą jej zawartość, ponieważ zapis danych jest rozkładany naprzemiennie pomiędzy dyski wchodzące w jej skład, a więc - utrata dostępu do części danych uniemożliwia odczyt ich całości.

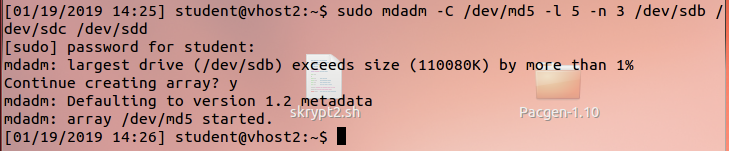
W macierzy RAID1 jest zastosowana technologia tzw. mirroring’u, co oznacza, że wszystkie dane zapisywane są dwukrotnie - raz na każdym z dysków tworzących macierz. Umożliwia to więc w razie awarii jednego z nich dostęp do całości danych z pozostałego sprawnego dysku, lecz w przypadku awarii obydwu dysków - wszystkie dane zostają utracone.

W macierzy RAID5 - zapisane na dyskach sumy kontrolne umożliwiają odzyskanie danych nawet w przypadku awarii jednego z dysków, lecz nie w przypadku awarii wielu dysków jednocześnie.

W macierzy RAID6 - dzięki zastosowaniu podwójnych sum kontrolnych - jesteśmy w stanie odzyskać wszystkie dane w przypadku awarii jednego lub nawet dwóch dysków w macierzy.

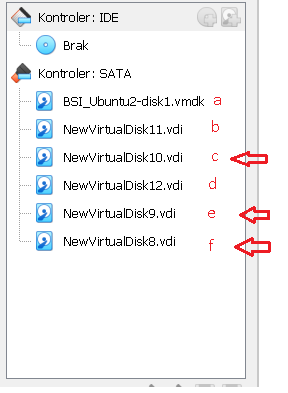
**Zadanie 7.3**

Utworzenie macierzy RAID5



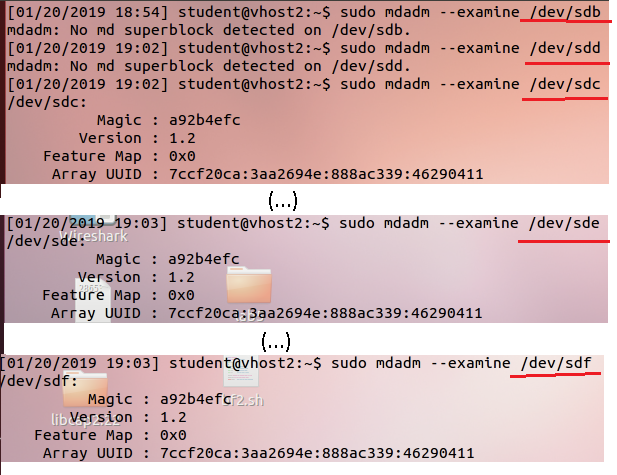
P.4 W sprawozdaniu udokumentuj uruchomienie macierzy na drugim systemie. Na co należy zwrócić uwagę, by operacja zakończyła się sukcesem?

Należy zwrócić uwagę na to, że poprzez zmianę kolejności podłączenia dysków - oznaczenia (/dev/sd**a, ...**/sd**b**) również zmieniły swoją kolejność. Wcześniejsze dyski b, c i d, uwzględniając zmienioną kolejność podłączenia, powinny mieć teraz oznaczenia c,e i f. Macierz została utworzona na dyskach 8,9 i 10, po przełączeniu dysków do drugiego systemu - macierz powinna być uruchamiana na tych samych dyskach. Zmianę ilustruje poniższy zrzut ekranu.

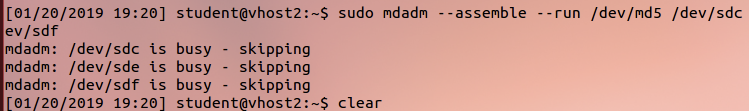


Aby upewnić się, które dyski wchodziły w skład macierzy możemy, oprócz podejrzenia ich kolejności w ustawieniach VBox, skorzystać z polecenia mdadm --examine /dev/sdX, gdzie X to litera z ciągu {a,b,c,…}. Polecenie zwróci informacje nt. dysków, które wchodziły uprzednio w skład macierzy oraz informację o braku ‘superblock’ dla tych, które nie były do niej podłączone.

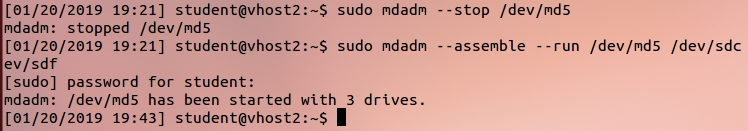
Wywołanie poleceń dla poszczególnych dysków potwierdza uprzednie założenie - dyski, z których składała się macierz (uprzednio - b, c, d) mają teraz oznaczenia c, e, f.



Po próbie uruchomienia ponownego macierzy - mdadm zwraca komunikat informujący, że dyski są zajęte:

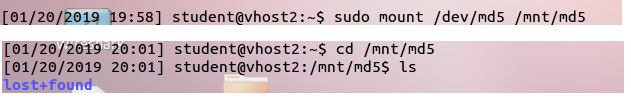


Musimy więc zatrzymać działanie macierzy md5:



Po zatrzymaniu md5, próba ponownego uruchomienia macierzy kończy się sukcesem.

Po zamontowaniu macierzy możemy ją otworzyć:



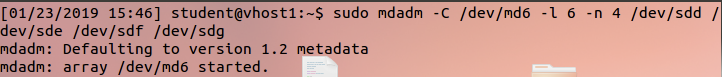
Zgodnie z przewidywaniami dla macierzy typu RAID5 - po usunięciu więcej niż jednego dysku - dane przechowywane w macierzy zostały utracone.

Zadanie 7.4

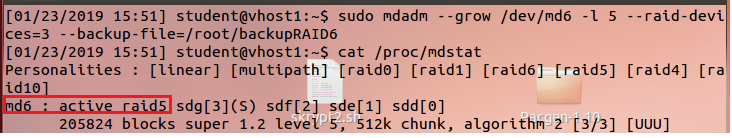
Utwórz macierz RAID6. Przeprowadź jej migrację do RAID5. Przedstaw proces tworzenia i migracji macierzy.

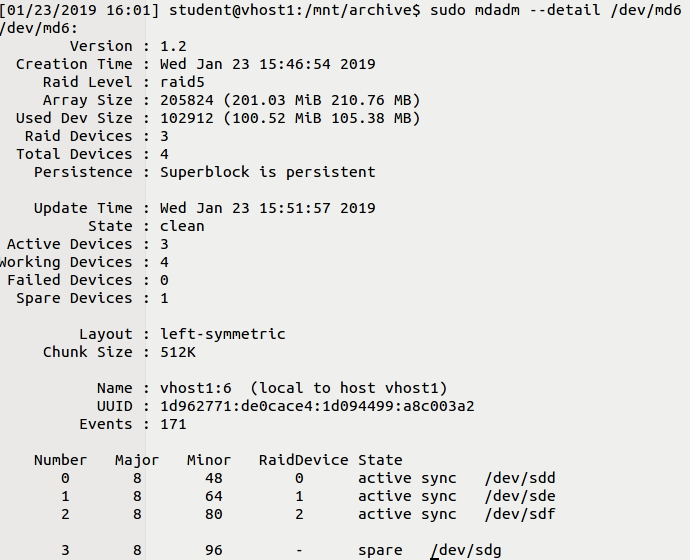
Pierwszym krokiem jest dodanie przynajmniej 4 dysków do maszyny wirtualnej (minimalna ilość nośników do utworzenia macierzy RAID6).

Utworzenie macierzy RAID6:



Migracja do RAID5 (wraz z utworzeniem kopii zapasowej przechowywanych danych):





Zarówno zawartość pliku mdstat, jak i polecenie mdadm --detail potwierdzają powodzenie migracji macierzy - /dev/md6 jest teraz macierzą typu RAID5. Czwarty dysk (/dev/sdg) został teraz dyskiem zapasowym, ponieważ - w przeciwieństwie do macierzy typu RAID6 - RAID5 wymaga trzech, nie - czterech, dysków do działania.