

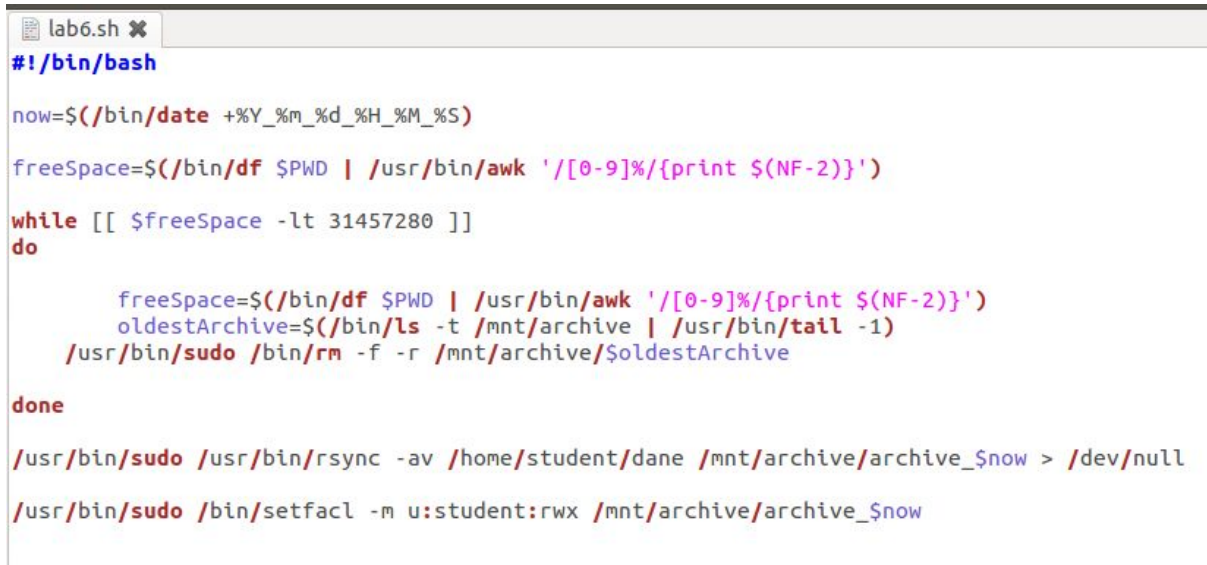
SPRAWOZDANIE
BEZPIECZEŃSTWO SYSTEMÓW
INFORMATYCZNYCH

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:



```
lab6.sh ✕
#!/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
```

```
#!/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:

```
home/student/openssl-1.0.1/util/point.sh
home/student/openssl-1.0.1/util/selftest.pl
home/student/openssl-1.0.1/util/shlib_wrap.sh
home/student/openssl-1.0.1/util/sp-diff.pl
home/student/openssl-1.0.1/util/speed.sh
home/student/openssl-1.0.1/util/src-dep.pl
home/student/openssl-1.0.1/util/ssleay.num
home/student/openssl-1.0.1/util/tab_num.pl
home/student/openssl-1.0.1/util/x86asm.sh
home/student/openssl-1.0.1/util/pl/
home/student/openssl-1.0.1/util/pl/BC-32.pl
home/student/openssl-1.0.1/util/pl/Mingw32.pl
home/student/openssl-1.0.1/util/pl/OS2-EMX.pl
home/student/openssl-1.0.1/util/pl/VC-32.pl
home/student/openssl-1.0.1/util/pl/linux.pl
home/student/openssl-1.0.1/util/pl/netware.pl
home/student/openssl-1.0.1/util/pl/ultrix.pl
home/student/openssl-1.0.1/util/pl/unix.pl

sent 112454865 bytes  received 66929 bytes  5770348.41 bytes/sec
total size is 112195388  speedup is 1.00
[01/21/2019 00:21] student@vhost2:~$
```

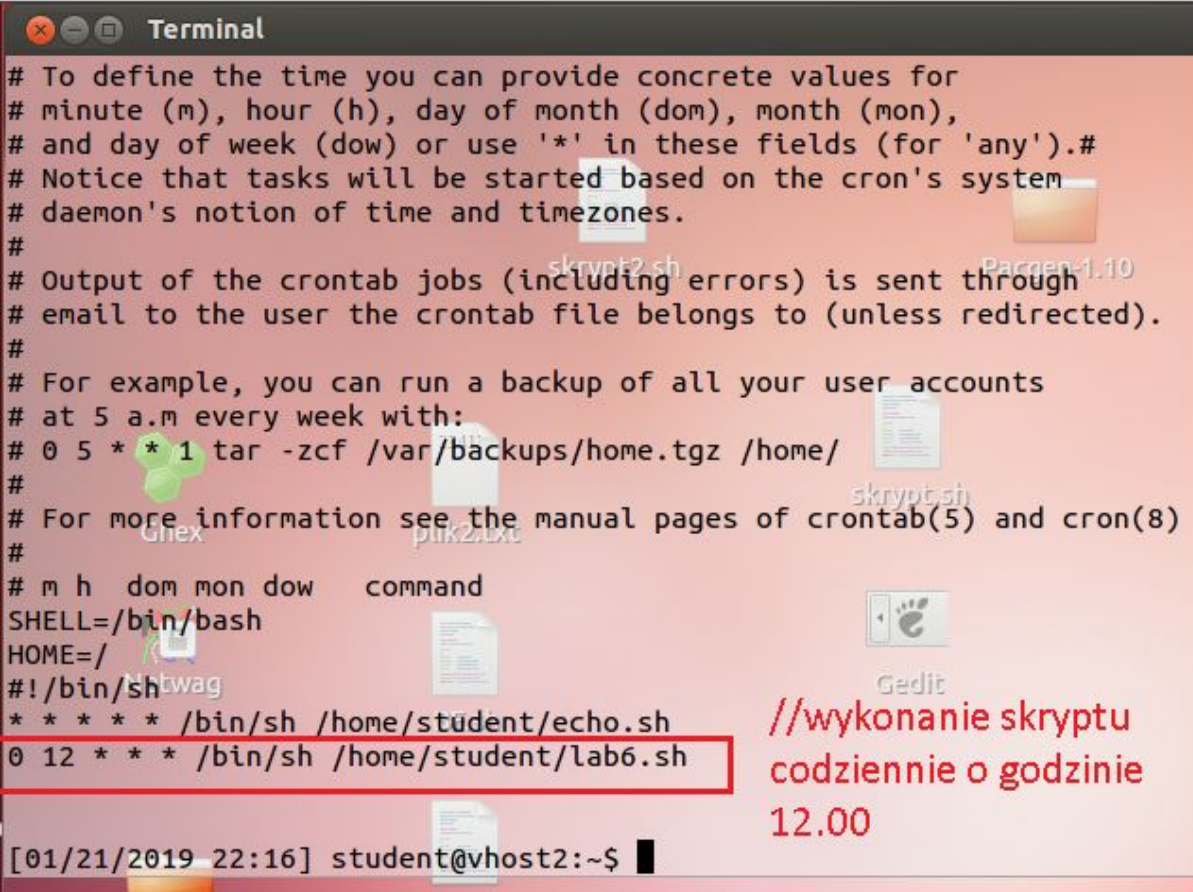
Utworzone archiwum:

```
[01/21/2019 00:21] student@vhost2:~$ ls /mnt/archiwum
archiwum_2019_01_21_00_21_06
[01/21/2019 00:23] student@vhost2:~$
```

Kolejne archiwa:

```
[01/21/2019 00:29] student@vhost2:~$ ls /mnt/archiwum
archiwum_2019_01_21_00_21_06  archiwum_2019_01_21_00_28_44
archiwum_2019_01_21_00_27_27
[01/21/2019 00:29] student@vhost2:~$
```

Wpis crontab:



```
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
SHELL=/bin/bash
HOME=/
#!/bin/sh
* * * * * /bin/sh /home/student/echo.sh
0 12 * * * /bin/sh /home/student/lab6.sh

[01/21/2019 22:16] student@vhost2:~$
```

//wykonanie skryptu
codziennie o godzinie
12.00

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):

```
fullBackup.sh ✕
#!/bin/bash

now=$(/bin/date +%Y_%m_%d_%H_%M_%S)  pobranie daty i godziny w odpowiednim formacie

freeSpace=$(/bin/df $PWD | /usr/bin/awk '/[0-9]%/{'print $(NF-2)}')  ustalenie ilości miejsca na dysku oraz rozmiaru
backupSizeB=$(/usr/bin/sudo /usr/bin/du -hcs -B1 /home/student/dane | /bin/grep /home/student/dane | /usr/
bin/tr -d ' ' /home/student/dane')  backup'owanego katalogu (/home/student/dane)

while [[ $freeSpace -lt $backupSizeB ]]  dopóki ilość miejsca na dysku jest
do  mniejsza od rozmiaru backupu

    freeSpace=$(/bin/df $PWD | /usr/bin/awk '/[0-9]%/{'print $(NF-2)}')  redeklaracja ilości miejsca
    oldestArchive=$(/bin/ls -t /mnt/archive | /usr/bin/tail -1)
    /usr/bin/sudo /bin/rm -f -r /mnt/archive/$oldestArchive  usunięcie najstarszego
                                                             archiwum
done

/usr/bin/sudo /usr/bin/rsync -av /home/student/dane /mnt/archive/archive_$now > /dev/null  utworzenie backupu
/usr/bin/sudo /bin/setfacl -m u:student:rwX /mnt/archive/archive_$now  ustalenie uprawnień do
                                                             archiwum dla użytkownika

/usr/bin/sudo /usr/bin/sshpas -p "root" /usr/bin/scp -C -c 3des -r /mnt/archive/archive_$now 10.0.2.4:/
mnt/archive/  skopiowanie archiwum na vhost2
```

```
#!/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/sshpas -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):

```
przyrostowyBackup.sh ✕
#!/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/sshpas -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
```

```
#!/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/sshpas -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.

```
30 16 * * 5 /bin/sh /home/student/fullBackup.sh  
30 16 * * 1-4 /bin/sh /home/student/przyrostowyBackup.sh
```

Działanie skryptów:

1. fullBackup.sh (wykonanie pełnej kopii zapasowej)

```
[01/23/2019 16:14] student@vhost1:~$ ls /mnt/archive  
[01/23/2019 16:14] student@vhost1:~$ sudo ./fullBackup.sh  
[01/23/2019 16:15] student@vhost1:~$ ls /mnt/archive  
archive_2019_01_23_16_15_50
```

```
[01/23/2019 16:14] student@vhost2:~$ ls /mnt/archive  
[01/23/2019 16:14] student@vhost2:~$ ls /mnt/archive  
archive 2019 01 23 16 15 50
```

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

```
[01/23/2019 16:15] student@vhost1:~$ sudo ./przyrostowyBackup.sh  
[01/23/2019 16:16] student@vhost1:~$ ls /mnt/archive  
archive_2019_01_23_16_15_50 archive_2019_01_23_16_16_13
```

```
[01/23/2019 16:15] student@vhost2:~$ ls /mnt/archive  
archive_2019_01_23_16_15_50 archive_2019_01_23_16_16_13
```

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.

a. Dodanie dysku do systemu



b. Tworzenie partycji na nowym dysku

```
cfdisk (util-linux 2.20.1)

Disk Drive: /dev/sdb
Size: 106763776 bytes, 106 MB
Heads: 255 Sectors per Track: 63 Cylinders: 12

Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
sdb1      Primary    Linux      106.77*

Ubuntu One

[ Bootable ] [ Delete ] [ Help ] [ Maximize ] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]

Write partition table to disk (this might destroy data)
```

c. Ustawienie systemu plików na nowej partycji na .ext4

```
[01/15/2019 17:40] student@vhost2:~$ sudo mkfs.ext4 /dev/sdb1
mke2fs 1.42 (29-Nov-2011)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
26104 inodes, 104228 blocks
5211 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
13 block groups
8192 blocks per group, 8192 fragments per group
2008 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

[01/15/2019 17:40] student@vhost2:~$
```

d. Zamontowanie partycji w /mnt/archiwum

```
[01/15/2019 17:42] student@vhost2:~$ sudo mount /dev/sdb1 /mnt/archiwum
[01/15/2019 17:42] student@vhost2:~$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	77G	4.1G	69G	6%	/
udev	2.0G	4.0K	2.0G	1%	/dev
tmpfs	808M	808K	808M	1%	/run
none	5.0M	0	5.0M	0%	/run/lock
none	2.0G	200K	2.0G	1%	/run/shm
/dev/sdb1	99M	5.6M	88M	6%	/mnt/archiwum

```
[01/15/2019 17:42] student@vhost2:~$
```

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

```
[01/15/2019 17:54] student@vhost2:/mnt/archiwum$ sudo blkid /dev/sdb1
/dev/sdb1: UUID="48924fa7-d807-4bea-bfb8-bcbf3bd3b0d2" TYPE="ext4"
[01/15/2019 17:54] student@vhost2:/mnt/archiwum$
```

```
fstab ✕
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc nodev,noexec,nosuid 0 0
# / was on /dev/sda1 during installation
UUID=c5c030bc-192c-4729-a9cc-6003c7b45097 / ext4 errors=remou
# swap was on /dev/sda5 during installation
UUID=d44702dc-45aa-4313-b2bd-b4729d078f27 none swap sw
/dev/fd0 /media/floppy0 auto rw,user,noauto,exec,utf8 0 0
UUID=48924fa7-d807-4bea-bfb8-bcbf3bd3b0d2 /mnt/archiwum ext4 auto,acl 0 0|
```

2.

a. Utworzenie macierzy RAID1

```
[01/17/2019 16:26] student@vhost2:~$ sudo mdadm -C /dev/md0 -l 1 -n 2 /dev/sdb /
dev/sdc
mdadm: /dev/sdb appears to be part of a raid array:
  level=raid0 devices=0 ctime=Thu Jan 1 01:00:00 1970
mdadm: partition table exists on /dev/sdb but will be lost or
  meaningless after creating array
mdadm: Note: this array has metadata at the start and
  may not be suitable as a boot device. If you plan to
  store '/boot' on this device please ensure that
  your boot-loader understands md/v1.x metadata, or use
  --metadata=0.90
mdadm: largest drive (/dev/sdc) exceeds size (104128K) by more than 1%
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

Wynik polecenia fdisk -l dla utworzonej macierzy

```
Disk /dev/md0: 106 MB, 106627072 bytes
2 heads, 4 sectors/track, 26032 cylinders, total 208256 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```


3. Komunikat o awarii macierzy przy uruchamianiu systemu

```
** WARNING: There appears to be one or more degraded RAID devices **

The system may have suffered a hardware fault, such as a disk drive
failure. The root device may depend on the RAID devices being online. One
or more of the following RAID devices are degraded:
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md127 : inactive sdb[0](S)
          104197 blocks super 1.2

unused devices: <none>
You may attempt to start the system anyway, or stop now and attempt
manual recovery operations. To do this automatically in the future,
add "bootdegraded=true" to the kernel boot options.

If you choose to start the degraded RAID, the system may boot normally,
but performance may be degraded, and a further hardware fault could
result in permanent data loss.

If you abort now, you will be provided with a recovery shell.

Do you wish to start the degraded RAID? [y/N]:
```

4. Dodanie nowego dysku do zdegenerowanej macierzy

```
[01/17/2019 16:58] student@vhost2:~$ sudo mdadm --manage /dev/md127 --add /dev/sdc
mdadm: added /dev/sdc
```

Po naprawieniu macierzy - komunikat o jej uszkodzeniu znika.

5.

a. Migracja macierzy RAID1 do RAID5

```
[01/19/2019 13:17] student@vhost2:~$ sudo mdadm /dev/md127 --grow -l 5
mdadm: level of /dev/md127 changed to raid5
```

b. Dodanie dwóch dodatkowych dysków do macierzy

```
[01/19/2019 13:36] student@vhost2:~$ sudo mdadm --manage /dev/md127 --add /dev/sdd
[sudo] password for student:
mdadm: added /dev/sdd
[01/19/2019 13:36] student@vhost2:~$ sudo mdadm --add /dev/md127 /dev/sde
mdadm: added /dev/sde
[01/19/2019 13:37] student@vhost2:~$
```

c. Zwiększenie rozmiaru woluminu

```
[01/19/2019 13:42] student@vhost2:~$ sudo mdadm --grow /dev/md127 --size=max
mdadm: component size of /dev/md127 has been set to 104192K
```

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

```

** WARNING: There appears to be one or more degraded RAID devices **

The system may have suffered a hardware fault, such as a disk drive
failure. The root device may depend on the RAID devices being online. One
or more of the following RAID devices are degraded:
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md127 : inactive sdc[3](S) sdb[0](S) sdd[4](S)
          321529 blocks super 1.2

unused devices: <none>
You may attempt to start the system anyway, or stop now and attempt
manual recovery operations. To do this automatically in the future,
add "bootdegraded=true" to the kernel boot options.

If you choose to start the degraded RAID, the system may boot normally,
but performance may be degraded, and a further hardware fault could
result in permanent data loss.

If you abort now, you will be provided with a recovery shell.

Do you wish to start the degraded RAID? [y/N]:

```

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.

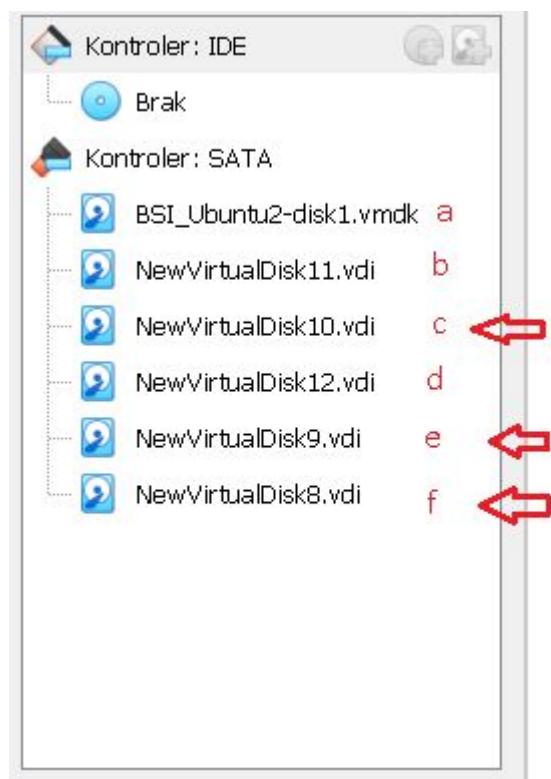
Zadanie 7.3

Utworzenie macierzy RAID5

```
[01/19/2019 14:25] student@vhost2:~$ sudo mdadm -C /dev/md5 -l 5 -n 3 /dev/sdb /dev/sdc /dev/sdd
[sudo] password for student:
mdadm: largest drive (/dev/sdb) exceeds size (110080K) by more than 1%
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md5 started.
[01/19/2019 14:26] student@vhost2:~$
```

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/sda, .../sdb) 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.

```
[01/20/2019 18:54] student@vhost2:~$ sudo mdadm --examine /dev/sdb
mdadm: No md superblock detected on /dev/sdb.
[01/20/2019 19:02] student@vhost2:~$ sudo mdadm --examine /dev/sdd
mdadm: No md superblock detected on /dev/sdd.
[01/20/2019 19:02] student@vhost2:~$ sudo mdadm --examine /dev/sdc
/dev/sdc:
    Magic : a92b4efc
    Version : 1.2
    Feature Map : 0x0
    Array UUID : 7ccf20ca:3aa2694e:888ac339:46290411
    (...)
[01/20/2019 19:03] student@vhost2:~$ sudo mdadm --examine /dev/sde
/dev/sde:
    Magic : a92b4efc
    Version : 1.2
    Feature Map : 0x0
    Array UUID : 7ccf20ca:3aa2694e:888ac339:46290411
    (...)
[01/20/2019 19:03] student@vhost2:~$ sudo mdadm --examine /dev/sdf
/dev/sdf:
    Magic : a92b4efc
    Version : 1.2
    Feature Map : 0x0
    Array UUID : 7ccf20ca:3aa2694e:888ac339:46290411
```

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

```
[01/20/2019 19:20] student@vhost2:~$ sudo mdadm --assemble --run /dev/md5 /dev/sdc
/dev/sdf
mdadm: /dev/sdc is busy - skipping
mdadm: /dev/sde is busy - skipping
mdadm: /dev/sdf is busy - skipping
[01/20/2019 19:20] student@vhost2:~$ clear
```

Musimy więc zatrzymać działanie macierzy md5:

```
[01/20/2019 19:21] student@vhost2:~$ sudo mdadm --stop /dev/md5
mdadm: stopped /dev/md5
[01/20/2019 19:21] student@vhost2:~$ sudo mdadm --assemble --run /dev/md5 /dev/sdc
/dev/sdf
[sudo] password for student:
mdadm: /dev/md5 has been started with 3 drives.
[01/20/2019 19:43] student@vhost2:~$ █
```

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

Po zamontowaniu macierzy możemy ją otworzyć:

```
[01/20/2019 19:58] student@vhost2:~$ sudo mount /dev/md5 /mnt/md5
[01/20/2019 20:01] student@vhost2:~$ cd /mnt/md5
[01/20/2019 20:01] student@vhost2:/mnt/md5$ ls
lost+found
```

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:

```
[01/23/2019 15:46] student@vhost1:~$ sudo mdadm -C /dev/md6 -l 6 -n 4 /dev/sdd /dev/sde /dev/sdf /dev/sdg
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md6 started.
```

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

```
[01/23/2019 15:51] student@vhost1:~$ sudo mdadm --grow /dev/md6 -l 5 --raid-devices=3 --backup-file=/root/backupRAID6
[01/23/2019 15:51] student@vhost1:~$ cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md6 : active raid5 sdg[3](S) sdf[2] sde[1] sdd[0]
      205824 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]
```

```
[01/23/2019 16:01] student@vhost1:/mnt/archive$ sudo mdadm --detail /dev/md6
/dev/md6:
    Version : 1.2
  Creation Time : Wed Jan 23 15:46:54 2019
    Raid Level : raid5
    Array Size : 205824 (201.03 MiB 210.76 MB)
  Used Dev Size : 102912 (100.52 MiB 105.38 MB)
    Raid Devices : 3
  Total Devices : 4
 Persistence : Superblock is persistent

 Update Time : Wed Jan 23 15:51:57 2019
   State : clean
 Active Devices : 3
Working Devices : 4
 Failed Devices : 0
  Spare Devices : 1


 Layout : left-symmetric
 Chunk Size : 512K

   Name : vhost1:6 (local to host vhost1)
  UUID : 1d962771:de0cace4:1d094499:a8c003a2
 Events : 171

Number   Major   Minor   RaidDevice State
   0         8       48         0   active sync  /dev/sdd
   1         8       64         1   active sync  /dev/sde
   2         8       80         2   active sync  /dev/sdf

   3         8       96         -   spare      /dev/sdg
```

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.