

Konfiguracja autoloader'a do pracy z Bacula w systemie operacyjnym GNU/Linux

2010-09-23 autor: **gani**

Artykuł prezentuje przygotowanie urządzenia autoloader'a do pracy z Bacula w systemie GNU/Linux oraz podstawową konfigurację Bacula dla tego typu urządzenia.

Wstęp

Techniczna ewolucja taśmy magnetycznej doprowadziła nas do momentu, w którym do zarządzania magazynem taśm zaprzęgnięto roboty. Robot przy pomocy promienia laserowego rozpoznaje taśmę, którą chce użyć. Następnie zabiera odpowiednią taśmę, podjeżdża do napędu taśmowego i z obłądną precyzją załadowuje do niego taśmę. Wraca na swe pierwotne miejsce i oczekuje na rozkaz wyładowania taśmy i umieszczenia jej na powrót w magazynie.

Może brzmi to jak fragment z literatury fantastyczno-naukowej, niemniej jednak nie ma w tym ani grama fikcji. Dokładnie takie operacje, jak powyżej opisane, zachodzą w urządzeniu autoloader'a.

Co to jest autoloader

Autoloader jest to urządzenie wyposażone w napęd taśm magnetycznych, zmieniarkę taśm (robot) oraz magazyn przechowujący taśmy (to tak w skrócie). W odróżnieniu od bibliotek taśmowych, autoloader nie ma możliwości rozbudowy, czy to o dodatkowe napędy czy też magazynki. Jest to więc tańsze rozwiązanie zautomatyzowanych urządzeń ze zmieniarką, lecz niech słowo „tańsze” nie wprowadza w błąd, gdyż mimo tego są to stosunkowo drogie urządzenia, których cena może sięgać rzędu kilkudziesięciu tysięcy złotych.

Proszę wybaczyć mi to wyjaśnienie, gdyż jestem świadom, że osoby chcące skonfigurować autoloader do pracy z Bacula zapewne wiedzą jak on działa, a już na pewno wiedzą, czym autoloader jest. Jestem jednak świadom, że i tutaj mogą zdarzać się pomyłki, których chciałbym uniknąć.

Użyta konfiguracja

W tym artykule użyłem autoloader'a Exabyte 110L podłączonego do kontrolera Adaptec AHA-2940U2W poprzez interfejs SCSI. Całość działa pod kontrolą systemu operacyjnego GNU/Linux.

Przygotowania

Przed przystąpieniem do konfiguracji w systemie operacyjnym należy upewnić się, że urządzenie autoloader'a jest poprawnie skonfigurowane w BIOS'ie kontrolera.

Pierwszą czynnością jaką trzeba wykonać, aby skonfigurować urządzenie do pracy z Bacula, jest zapoznanie się z nazwami plików urządzeń, jakie zostały przydzielone urządzeniom. Posłużyłem się tu liczbą mnogą, ponieważ w systemie operacyjnym autoloader jest widoczny w postaci dwóch urządzeń: napędu taśm i zmieniarki. Informacje o nazwach plików urządzeń można uzyskać np. poprzez wywołanie komendy `lsscsi` w następujący sposób:

```
# lsscsi -g
[0:0:0:0]    mediumx  EXABYTE   Exabyte 110L      2.8    /dev/sch0  /dev/sg2
[0:0:1:0]    tape     IBM       ULTRIUM-TD1     16E0   /dev/st0   /dev/sg3
[1:0:0:0]    disk     ATA       ST3250410AS     3.AA   /dev/sda   /dev/sg0
[3:0:0:0]    cd/dvd   TSSTcorp  CDDVDW SH-S223F  SB03   /dev/sr0   /dev/sg1
```

gdzie użyty przełącznik -g powoduje wyświetlenie również ogólnych nazwy plików urządzeń SCSI (tzw. nazwy urządzeń „generic”) podłączonych do dowolnej magistrali SCSI.

Kolumny powyższego listingu to kolejno:

- identyfikacja urządzenia na magistrali SCSI w postaci: [numer magistrali, kanał magistrali, SCSI ID, LUN],
- typ urządzenia. Typ „mediumx” oznacza urządzenie zmieniaarki, a „tape” to typ urządzenia napędu taśm magnetycznych,
- nazwa producenta czyli tzw. „vendor” np. „EXABYTE”, „IBM” itd,
- model urządzenia np. „Exabyte 110L” czy „ULTRIUM-TD1”
- numer rewizji sprzętu,
- nazwa pliku urządzenia (np. sd* - dysk, sr* - napęd optyczny, st* – napęd taśmowy, sch* – urządzenie zmieniaarki) właściwa dla każdego typu. Zazwyczaj nazwa ta pochodzi od nazwy modułu jądra obsługującego dany typ urządzenia.
- nazwa ogólna. Każdemu urządzeniu SCSI odpowiada ogólna nazwa urządzenia. Za jej nadanie odpowiedzialny jest moduł jądra systemu o nazwie sg (SCSI generic).

Z powyższego listingu warto zapamiętać dwie rzeczy: ogólną nazwę pliku urządzenia zmieniaarki (w tym przykładzie jest to /dev/sg2) oraz nazwę pliku urządzenia napędu taśm magnetycznych (tutaj jest to /dev/st0). To właśnie nimi można posłużyć się podczas podawania nazw plików urządzeń w pliku konfiguracyjnym demona magazynowania Bacula (plik bacula-sd.conf).

Kończąc podrozdział przygotowań, wspomnę jeszcze o subtelnej lecz znaczącej uwadze odnośnie pliku urządzenia napędu taśm. W katalogu /dev dla każdego pliku /dev/st* powinien istnieć plik /dev/nst* np. /dev/st0 i /dev/nst0. Różnica pomiędzy nimi jest taka, że w przypadku użycia w pliku konfiguracyjnym /dev/st* po każdym zwolnieniu przez Bacula napędu taśmowego, znajdująca się w nim taśma zostanie przewinięta na początek. Takie użycie generuje wiele zbędnych przejść taśmy, a co za tym idzie powoduje szybsze zużycie taśmy. Aby tego uniknąć potrzeba posłużyć się plikiem /dev/nst* zamiast /dev/st*. Nst (skrót od ang. Non-rewind SCSI Tape device) gwarantuje, że taśma pozostanie na swej pozycji po każdej operacji Bacula na taśmie.

UDEV

Pamiętając już nazwy plików urządzeń autoloader’a (dla przypomnienia tutaj jest to /dev/sg2 oraz /dev/st0) można przystąpić do konfigurowania Bacula do pracy z autoloader’em. Minusem użycia takich nazw plików urządzeń w konfiguracji Bacula jest to, że mogą one ulec zmianie np. po podłączeniu w przyszłości innych urządzeń SCSI lub po manipulacji SCSI ID urządzeń. Tak więc fakt, że teraz plik urządzenia zmieniaarki to /dev/sg2 wcale nie oznacza, że będzie on taki zawsze, lecz może ulec zmianie na np. /dev/sg3.

Jeśli używany system operacyjny wykorzystuje do alokacji plików urządzeń system plików UDEV to warto zdefiniować reguły UDEV, które podczas każdego startu systemu operacyjnego rozpoznają urządzenia autoloader’a i stworzą do nich dowiązania symboliczne o stałych zdefiniowanych nazwach.

W celu określenia w/w reguł potrzebne będzie zdobycie dodatkowych informacji o urządzeniach autoloader'a, aby na ich podstawie móc stworzyć warunki do jednoznacznego rozpoznania sprzętu.

Dodatkowe informacje o sprzęcie można uzyskać przy pomocy programu „udevinfo”.

Dla urządzenia zmieniariki będzie to komenda:

```
# udevinfo -a -p $(udevinfo -q path -n /dev/sg2)
...
looking at parent device ,/devices/pci0000:00/0000:00:08.0/0000:01:06.0/host0/
target0:0:0:0/0:0:0:0':
  KERNELS=="0:0:0:0"
  SUBSYSTEMS=="scsi"
  DRIVERS=="ch"
  ATTRS{device_blocked}=="0"
  ATTRS{type}=="8"
  ATTRS{scsi_level}=="3"
  ATTRS{vendor}=="EXABYTE "
  ATTRS{model}=="Exabyte 110L "
  ATTRS{rev}=="2.8 "
  ATTRS{state}=="running"
  ATTRS{timeout}=="0"
  ATTRS{iocounterbits}=="32"
  ATTRS{iorequest_cnt}=="0x17"
  ATTRS{iodone_cnt}=="0x17"
  ATTRS{ioerr_cnt}=="0x1"
  ATTRS{modalias}=="scsi:t-0x08"
  ATTRS{evt_media_change}=="0"
  ATTRS{queue_depth}=="2"
  ATTRS{queue_type}=="none"
...
```

a dla napędu taśmowego będzie to:

```
# udevinfo -a -p $(udevinfo -q path -n /dev/nst0)
...
looking at parent device ,/devices/pci0000:00/0000:00:08.0/0000:01:06.0/host0/
target0:0:1/0:0:1:0':
  KERNELS=="0:0:1:0"
  SUBSYSTEMS=="scsi"
  DRIVERS=="st"
  ATTRS{device_blocked}=="0"
  ATTRS{type}=="1"
  ATTRS{scsi_level}=="4"
  ATTRS{vendor}=="IBM "
  ATTRS{model}=="ULTRIUM-TD1 "
  ATTRS{rev}=="16E0"
  ATTRS{state}=="running"
  ATTRS{timeout}=="900"
  ATTRS{iocounterbits}=="32"
  ATTRS{iorequest_cnt}=="0x14"
  ATTRS{iodone_cnt}=="0x13"
  ATTRS{ioerr_cnt}=="0x3"
  ATTRS{modalias}=="scsi:t-0x01"
  ATTRS{evt_media_change}=="0"
  ATTRS{queue_depth}=="2"
  ATTRS{queue_type}=="none"
...
```

Pominałem wyświetlenie pełnych długich listingów w/w komend pozostając na pokazaniu jedynie ich fragmentów, które przydadzą się do napisania reguł UDEV. Trzy kropki w listingach (...) oznaczają, że jest to ich wycinek.

Otrzymane powyżej dane to szereg informacji, na bazie których można już zdefiniować warunki do rozpoznania autoloader'a przez UDEV.

Katalog zawierający reguły UDEV'a u mnie znajduje się w lokalizacji:

```
/etc/udev/rules.d/
```

W tym katalogu tworzę plik o nazwie 55-bacula.rules i umieszczam w nim warunki dla autoloader'a. W tym celu posłużę się dwoma parametrami: vendor i model. Mój plik 55-bacula.rules wygląda następująco:

```
# cat /etc/udev/rules.d/55-bacula.rules
KERNEL=="sg[0-9]",          ATTRS{vendor}=="EXABYTE*",ATTRS{model}=="Exabyte 110L*",
\
    SYMLINK+="changer-exabyte-110l"

KERNEL=="nst[0-9]",        ATTRS{vendor}=="IBM*",ATTRS{model}=="ULTRIUM-TD1*", \
    SYMLINK+="tapedrive-ibm-lt01"
```

Takie wpisy powodują to, że po wykryciu w systemie operacyjnym mojego autoloader'a i przydzieleniu mu plików urządzeń, zostaną utworzone dowiązania symboliczne wskazujące na pliki urządzeń autoloader'a. Dla urządzenia zmieniarke będzie to dowiązanie symboliczne o nazwie „changer-exabyte-110l” a dla napędu taśm magnetycznych będzie to dowiązanie o nazwie „tapedrive-ibm-lt01”. Nie ma już więc znaczenia czy urządzenie zmieniarke zostanie obsłużone przez plik urządzenia /dev/sg2 czy /dev/sg5, ponieważ dowiązanie symboliczne /dev/changer-exabyte-110l zawsze będzie wskazywało na aktualny plik urządzenia zmieniarke. W taki sam sposób zadziała reguła dla napędu taśmowego.

W celu bardziej precyzyjnego określenia warunków reguł UDEV można rozszerzyć wpisy w pliku 55-bacula.rules o dodatkowe parametry uzyskane przy pomocy programu udevinfo.

W dalszej części artykułu będę posługiwał się nazwami urządzeń utworzonymi przy pomocy reguł UDEV.

Konfiguracja demona magazynowania (Storage Daemon)

Do pliku bacula-sd.conf dopisuję dwa zasoby „Autochanger” oraz „Device”. Pierwszy posłuży do zdefiniowania dyrektyw dla zmieniarke taśm, drugi natomiast będzie zawierał dyrektywy dla napędu taśm autoloader'a. Obydwa zasoby wyglądają następująco:

```
Autochanger {
    Name = Exabyte110L_Dev
    Device = Drive-1
    Changer Command = „/usr/local/bacula/etc/scripts/mtx-changer %c %o %S %a %d”
    Changer Device = /dev/changer-exabyte-110l
}

Device {
    Name = Drive-1
    Drive Index = 0
    Media Type = LTO1
    Archive Device = /dev/tapedrive-ibm-lt01
}
```

```
Automatic Mount = yes
Always Open = yes;
Removable Media = yes
Random Access = no
AutoChanger = yes
Alert Command = „sh -c ‚tapeinfo -f %c |grep TapeAlert|cat’”
}
```

Opis dyrektyw użytych w tej konfiguracji można znaleźć w oficjalnej dokumentacji Bacula:

Zasób „Autochanger” - http://bacula.org/5.0.x-manuals/en/main/main/Autochanger_Resource.html

Zasób „Device” - http://bacula.org/5.0.x-manuals/en/main/main/Storage_Daemon_Configuratio.html#SECTION00203000000000000000

Konfiguracja zarządcy (Director)

Poniżej znajduje się konfiguracja zasobu „Storage” w pliku konfiguracyjnym bacula-dir.conf.

```
Storage {
  Name = Exabyte110L
  Address = darkstar
  SD Port = 9103
  Password = „xxxxxxxxxx”
  Media Type = LTO1
  Device = Exabyte110L_Dev
  Autochanger = yes
}
```

Opis dyrektyw zasobu „Storage” znajduje się pod adresem:

http://bacula.org/5.0.x-manuals/en/main/main/Configuring_Director.html#SECTION00181400000000000000

MTX-CHANGER

Bacula do zarządzania urządzeniem autoloader’a wykorzystuje skrypt powłoki o nazwie mtx-changer. Skrypt ten odpowiada za takie operacje jak np.:

- załadowanie/wyładowanie taśmy z/do napędu taśm magnetycznych,
- wskazanie, czy w napędzie znajduje się taśma i jeśli tak, to z jakiego numeru slotu (magazynka) ona pochodzi,
- wyświetlenie numerów magazynków, w których znajdują się taśmy oraz jeśli zmieniarka jest wyposażona w czytnik kodów kreskowych (ang. barcode reader) to zwrócenie również nazw zapisanych kodem kreskowym na taśmach,
- wyświetlenie całkowitej ilości magazynków w urządzeniu autoloader’a.

Być może uzyskanie tych informacji wydaje się banalne, ponieważ wszystkie te dane można pobrać poprzez wydanie jednej komendy programu mtx. Gdy jednak czytnik zajrzy do pliku mtx-changer to zauważy zapewne, że użyto tam właśnie programu mtx oraz mt. Niby nic nadzwyczajnego, lecz proszę pamiętać, że tego skryptu używa Bacula podczas pracy z urządzeniem zmieniarke i jest to świetne miejsce do dopasowania komend skryptu mtx-changer do własnych potrzeb. Wspominam o tym, ponieważ niektóre autoloader’y bywają „grymaśne” jeśli chodzi o współpracę z programem mtx i to

również uwzględnili programiści Bacula tworząc plik konfiguracyjny o nazwie mtx-changer.conf (wykorzystywany przez skrypt mtx-changer), w którym poprzez zmianę wartości odpowiednich zmiennych skryptu można skorygować najczęstsze „grymasy” zmieniarzek taśm spotykane np. podczas załadowania lub wyładowania taśmy z napędu.

Użyty na potrzeby tego artykułu autoloader Exabyte110L wymaga przestawienia zmiennej pliku konfiguracyjnego mtx-changer.conf o nazwie offline w następujący sposób:

```
offline=1
```

Ta zmiana powoduje, że przed każdym wyładowaniem taśmy z napędu zostanie wykonana komenda:

```
mt -f {plik napędu taśm} offline
```

która zwalnia taśmę z napędu i pozwala na wyładowanie jej przez zmieniarkę.

Test napędu taśm oraz zmieniarcki

Jednym z narzędzi dostarczanych wraz z Bacula jest program btape. Przy jego pomocy można przetestować w sposób praktyczny zarówno swoją konfigurację autoloader'a i jego napędu taśmowego określoną w pliku konfiguracyjnym bacula-sd.conf jak i zdolność obydwu do współpracy z Bacula. Jest to dobry początek do tego, aby sprawdzić czy urządzenie będzie współpracować z Bacula. Co więcej, w przypadku nie zaliczenia testu program btape potrafi podpowiedzieć to, jakich dodatkowych dyrektyw można użyć w konfiguracji urządzeń w Bacula, aby skorygować błędy wykryte w trakcie testu.

Do wykonania testu potrzeba będzie jedna taśma załadowana do napędu przed uruchomieniem narzędzia btape. Po wydaniu komendy btape, np.:

```
# /usr/local/bacula/sbin/btape -c /usr/local/bacula/etc/bacula-sd.conf /dev/
tapedrive-ibm-lt01
Tape block granularity is 1024 bytes.
btape: butil.c:284 Using device: „/dev/tapedrive-ibm-lt01” for writing.
23-wrz 12:58 btape JobId 0: 3301 Issuing autochanger „loaded? drive 0” command.
23-wrz 12:58 btape JobId 0: 3302 Autochanger „loaded? drive 0”, result is Slot 1.
btape: btape.c:476 open device „Drive-1” (/dev/tapedrive-ibm-lt01): OK
*
```

zostanie udostępniona mini-konsola. Pełną listę opcji dostępnych w tej konsoli można uzyskać poprzez wydanie komendy „help”.

Do uruchomienia testu napędu taśm, w konsoli btape wpisuję komendę „test”:

```
* test
=== Write, rewind, and re-read test ===
I'm going to write 10000 records and an EOF
then write 10000 records and an EOF, then rewind,
and re-read the data to verify that it is correct.
This is an *essential* feature ...
btape: btape.c:1148 Wrote 10000 blocks of 64412 bytes.
btape: btape.c:608 Wrote 1 EOF to „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1164 Wrote 10000 blocks of 64412 bytes.
btape: btape.c:608 Wrote 1 EOF to „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1206 Rewind OK.
10000 blocks re-read correctly.
```

```
Got EOF on tape.
10000 blocks re-read correctly.
=== Test Succeeded. End Write, rewind, and re-read test ===
```

```
btape: btape.c:1274 Block position test
btape: btape.c:1286 Rewind OK.
Reposition to file:block 0:4
Block 5 re-read correctly.
Reposition to file:block 0:200
Block 201 re-read correctly.
Reposition to file:block 0:9999
Block 10000 re-read correctly.
Reposition to file:block 1:0
Block 10001 re-read correctly.
Reposition to file:block 1:600
Block 10601 re-read correctly.
Reposition to file:block 1:9999
Block 20000 re-read correctly.
=== Test Succeeded. End Write, rewind, and re-read test ===
```

```
=== Append files test ===
```

This test is essential to Bacula.

```
I'm going to write one record  in file 0,
                        two records in file 1,
                        and three records in file 2
```

```
btape: btape.c:578 Rewound „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:608 Wrote 1 EOF to „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:608 Wrote 1 EOF to „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:608 Wrote 1 EOF to „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:476 open device „Drive-1” (/dev/tapedrive-ibm-lt01): OK
btape: btape.c:578 Rewound „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1418 Now moving to end of medium.
btape: btape.c:629 Moved to end of medium.
We should be in file 3. I am at file 3. This is correct!
```

Now the important part, I am going to attempt to append to the tape.

```
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:608 Wrote 1 EOF to „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:578 Rewound „Drive-1” (/dev/tapedrive-ibm-lt01)
Done appending, there should be no I/O errors
```

Doing Bacula scan of blocks:

```
1 block of 64448 bytes in file 1
End of File mark.
2 blocks of 64448 bytes in file 2
End of File mark.
3 blocks of 64448 bytes in file 3
End of File mark.
1 block of 64448 bytes in file 4
End of File mark.
Total files=4, blocks=7, bytes = 451,136
End scanning the tape.
We should be in file 4. I am at file 4. This is correct!
```

The above Bacula scan should have output identical to what follows.
Please double check it ...

```
=== Sample correct output ===
1 block of 64448 bytes in file 1
End of File mark.
2 blocks of 64448 bytes in file 2
End of File mark.
3 blocks of 64448 bytes in file 3
End of File mark.
1 block of 64448 bytes in file 4
End of File mark.
Total files=4, blocks=7, bytes = 451,136
=== End sample correct output ===
```

If the above scan output is not identical to the sample output, you MUST correct the problem or Bacula will not be able to write multiple Jobs to the tape.

```
=== Write, backup, and re-read test ===
```

I'm going to write three records and an EOF then backup over the EOF and re-read the last record. Bacula does this after writing the last block on the tape to verify that the block was written correctly.

This is not an *essential* feature ...

```
btape: btape.c:578 Rewound „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:813 Wrote first record of 64412 bytes.
btape: btape.c:824 Wrote second record of 64412 bytes.
btape: btape.c:835 Wrote third record of 64412 bytes.
btape: btape.c:608 Wrote 1 EOF to „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:851 Backspaced over EOF OK.
btape: btape.c:856 Backspace record OK.
btape: btape.c:874
Block re-read correct. Test succeeded!
=== End Write, backup, and re-read test ===
```

```
=== Forward space files test ===
```

This test is essential to Bacula.

I'm going to write five files then test forward spacing

```
btape: btape.c:578 Rewound „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
```

```

btape: btape.c:608 Wrote 1 EOF to „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:608 Wrote 1 EOF to „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:608 Wrote 1 EOF to „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:608 Wrote 1 EOF to „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1905 Wrote one record of 64412 bytes.
btape: btape.c:1907 Wrote block to device.
btape: btape.c:608 Wrote 1 EOF to „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:578 Rewound „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1632 Now forward spacing 1 file.
We should be in file 1. I am at file 1. This is correct!
btape: btape.c:1644 Now forward spacing 2 files.
We should be in file 3. I am at file 3. This is correct!
btape: btape.c:578 Rewound „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1657 Now forward spacing 4 files.
We should be in file 4. I am at file 4. This is correct!

btape: btape.c:1675 Now forward spacing 1 more file.
We should be in file 5. I am at file 5. This is correct!

=== End Forward space files test ===

```

Powyższy listing pokazuje wyniki testu napędu, który to test przeszedł bez błędów. Można więc stwierdzić, że napęd będzie współpracował z Bacula.

Na koniec powyższego testu program btape wykrył, że napęd jest częścią autoloader'a i zaproponował dla niego przeprowadzenie testu.

```

Ah, I see you have an autochanger configured.
To test the autochanger you must have a blank tape
that I can write on in Slot 1.
Do you wish to continue with the Autochanger test? (y/n): y

=== Autochanger test ===
3301 Issuing autochanger „loaded” command.
Slot 1 loaded. I am going to unload it.
3302 Issuing autochanger „unload 1 0” command.
unload status=OK 0
3303 Issuing autochanger „load 1 0” command.
3303 Autochanger „load 1 0” status is OK.
btape: btape.c:476 open device „Drive-1” (/dev/tapedrive-ibm-lt01): OK
btape: btape.c:1562 Rewound „Drive-1” (/dev/tapedrive-ibm-lt01)
btape: btape.c:1569 Wrote EOF to „Drive-1” (/dev/tapedrive-ibm-lt01)

The test autochanger worked!!

```

Ten test również został zaliczony. Urządzenie autoloader'a jest gotowe do użycia.

Etykietowanie taśm

Poniżej zaprezentowane zostało przykładowe etykietowanie taśm w konsoli bconsole przy użyciu nazw zapisanych kodem kreskowym (ang. Barcode). Zaetykietowane zostaną taśmy ze slotów 1,2,3,4,5,6 czemu odpowiada parametr slots=1-6.

```
*label storage=Exabyte110L pool=ExabytePool slots=1-6 barcodes
Automatically selected Catalog: MojaBazaBaculi
Using Catalog „MojaBazaBaculi”
Connecting to Storage daemon Exabyte110L at darkstar:9103 ...
3306 Issuing autochanger „slots” command.
Device „Exabyte110L_Dev” has 10 slots.
Connecting to Storage daemon Exabyte110L at darkstar:9103 ...
3306 Issuing autochanger „list” command.
The following Volumes will be labeled:
Slot  Volume
=====
  1  TAPE01L1
  2  TAPE02L1
  3  TAPE03L1
  4  TAPE04L1
  5  TAPE05L1
  6  TAPE06L1

Do you want to label these Volumes? (yes|no): yes
Connecting to Storage daemon Exabyte110L at darkstar:9103 ...
Sending label command for Volume „TAPE01L1” Slot 1 ...
3301 Issuing autochanger „loaded? drive 0” command.
3302 Autochanger „loaded? drive 0”, result: nothing loaded.
3304 Issuing autochanger „load slot 1, drive 0” command.
3305 Autochanger „load slot 1, drive 0”, status is OK.
3000 OK label. VolBytes=64512 DVD=0 Volume=“TAPE01L1” Device=“Drive-1” (/dev/nst0)
Catalog record for Volume „TAPE01L1”, Slot 1  successfully created.
Sending label command for Volume „TAPE02L1” Slot 2 ...
3307 Issuing autochanger „unload slot 1, drive 0” command.
3304 Issuing autochanger „load slot 2, drive 0” command.
3305 Autochanger „load slot 2, drive 0”, status is OK.
3000 OK label. VolBytes=64512 DVD=0 Volume=“TAPE02L1” Device=“Drive-1” (/dev/nst0)
Catalog record for Volume „TAPE02L1”, Slot 2  successfully created.
Sending label command for Volume „TAPE03L1” Slot 3 ...
3307 Issuing autochanger „unload slot 2, drive 0” command.
3304 Issuing autochanger „load slot 3, drive 0” command.
3305 Autochanger „load slot 3, drive 0”, status is OK.
3000 OK label. VolBytes=64512 DVD=0 Volume=“TAPE03L1” Device=“Drive-1” (/dev/nst0)
Catalog record for Volume „TAPE03L1”, Slot 3  successfully created.
Sending label command for Volume „TAPE04L1” Slot 4 ...
3307 Issuing autochanger „unload slot 3, drive 0” command.
3304 Issuing autochanger „load slot 4, drive 0” command.
3305 Autochanger „load slot 4, drive 0”, status is OK.
3000 OK label. VolBytes=64512 DVD=0 Volume=“TAPE04L1” Device=“Drive-1” (/dev/nst0)
Catalog record for Volume „TAPE04L1”, Slot 4  successfully created.
Sending label command for Volume „TAPE05L1” Slot 5 ...
3307 Issuing autochanger „unload slot 4, drive 0” command.
3304 Issuing autochanger „load slot 5, drive 0” command.
3305 Autochanger „load slot 5, drive 0”, status is OK.
3000 OK label. VolBytes=64512 DVD=0 Volume=“TAPE05L1” Device=“Drive-1” (/dev/nst0)
Catalog record for Volume „TAPE05L1”, Slot 5  successfully created.
Sending label command for Volume „TAPE06L1” Slot 6 ...
3307 Issuing autochanger „unload slot 5, drive 0” command.
3304 Issuing autochanger „load slot 6, drive 0” command.
3305 Autochanger „load slot 6, drive 0”, status is OK.
3000 OK label. VolBytes=64512 DVD=0 Volume=“TAPE06L1” Device=“Drive-1” (/dev/nst0)
Catalog record for Volume „TAPE06L1”, Slot 6  successfully created.
```