AMANDA를 이용하여 백업하기

유성태

alloying (at) kldp.org

AMANDA는 "Advanced Maryland Automated Network Disk Archiver"의 약자로서 Maryland 대학에서 개발한 백업 유틸리티이다. AMANDA는 네트웍 여기저기에 흩어져 있는 컴퓨터들의 데이타를 백업하기에 적당한 솔루션이다.

고친 과정

고침 0.12002-09-17고친이 styoo

최초 작성

고침 0.22002-10-17고친이 styoo

원격 서버에서 복원하는 방법 추가

1. 서문

윈도우 머신에서 네트웍에 흩어져 있는 여러 대의 파일 서버에 있는 데이타를 백업/복원하기 위해서는 네트웍 드라이브로 연결해서 윈도우 백업 소프트웨어를 이용하거나 또는 상용 소프트웨어를 사용하는데, 상용 소프트웨어의 경우 각 파일 서버마다 모듈을 사야하므로 그 가격이 만만치 않다. 그리고, 윈도우에서 제공하는 백업 소프트웨어를 사용할 경우, 백업할 대상을 네트웍 드라이브로 연결해서 백업할 수도 있는데, 이 경우는 일부 테잎 장치를 지원하지 않는 문제가 있다. 그래서 리눅스 머신에서 그 작업을 대신할 수 없을까 해서 백업 소프트웨어를 찾아보니 AMANDA라는 것이 괜찮아 보였다. 약 한 달의 준비 기간과 시험기간을 거쳐 실전에 사용해보니 비교적 만족스럽기에 여러분에게도 추천하고자 이 글을 작성하게 되었다.

1.1. 저작권 정보

1.2. 책임의 한계

본 저자는 문서의 내용이 야기할 수 있는 어떠한 결과에 대해서도 책임을 지지 않습니다. 본 문서에서 내포하고 있는 정보들 및 예제들은 여러분이 알아서 활용하십시오. 비록 최선을 다 했으나 이 문서는 틀린 점이나 오류가 있을 수도 있습니다. 만약 여러분이 틀린 점을 발견했다면 꼭 저에게 알려 주시기 바랍니다.

1.3. 감사의 글

AMANDA라는 비교적 훌륭한 백업 소프트웨어를 개발하고 있는 AMANDA 개발 그룹에 감사드립니다.

1.4. 피드백

이 문서에 대한 발전적인 제안이나 수정사항, 문제점 등에 대한 피드백은 언제든지 환영합니다. `<alloying (at) kldp.org>`로 메일을 보내 주십시오.

2. Amanda 설치

"UNIX Backup & Recovery" 책에 "Using Amanda"라는 Amanda 사용에 대한 내용이 있으며 http://www.backupcentral.com/amanda.html (http://www.backupcentral.com/amanda.html)에서 온라인으로도 볼 수 있다. 설치는 위 자료를 참고하기 바란다.

Amanda는 백업을 운영할 사용자를 정하게 되어 있으며, 래드햇 또는 데비안은 그 사용자가 backup으로 지정되어 있다. backup이 아닌 다른 사용자로 지정하려면 패키지를 다시 빌드하여야 하는데 configure 단계에서 원하는 사용자로 바꾼 후 패키지를 빌드하면 된다. 필자는 데비안 소스 패키지를 받아다가 --with-user=backup --with-group=backup을 --with-user=styoo --with-group=styoo로 바꾸었다. 당연한 말이지만, styoo는 필자가 사용하는 리눅스 박스의 login name이다. 그 이외의 configure 옵션은 특별히 바꿀 필요가 없을 것이다. 데비안 소스 패키지를 받아서 새로이 빌드할 경우, rules 파일외에 몇 개의 파일에서 backup을 원하는 사용자로(여기서는 styoo) 바꾸어 주어야한다. 새로이 빌드하기가 번거롭다면, 사용자를 바꿀 필요없이 제공되는 패키지 그대로 설치해서 사용해도 된다. 그러나 이때는 root 또는 backup 사용자로 백업을 실행해야 할 것이다.

Amanda는 서버 s/w와 클라이언트 s/w로 구성되어 있다. 서버 s/w는 테잎 장치가 부착되어 있으며 백업 프로세스가 돌아갈 서버(이하 테잎서버라 칭함.)에 설치될 것이며, 클라이언트 s/w는 백업할 데이타를 저장하고 있는 시스템(이하 파일서버라 칭함)에 설치될 것이다. 테잎서버에도 백업할 데이타가 있다면 클라이언트 s/w를 설치해야 한다.

그 외에 반드시 필요한 소프트웨어로는 tar, mt-st가 있으며, 윈도우 시스템의 데이타를 samba를 이용하여 백업하려고 한다면 samba가 필요할 것이고 테잎 장치에 autoloader나 scsi changer와 같은 미디어 체인저가 있다면 mtx도 필요할 것이다.

Amanda는 백업할 때 dump와 tar 두 가지 명령 중에 한 가지를 사용할 수 있다. 여기서는 tar를 사용하여 백업하는 것을 기준으로 설명하므로 버전 1.12 이후의 tar가 반드시 설치되어 있어야 한다.

3. 대략적인 AMANDA의 작동에 대하여

여러가지 세세한 설정에 대해서는 6절에서 설명하고 있으니 참고하기 바라며, 여기서는 대강의 Amanda 작동 순서를 설명하겠다.

백업을 시작하기 전에 테잎을 인식할 수 있는 라벨을 amlabel 명령으로 테잎에 써 넣어야 한다. 라벨을 테잎에 쓰고 나면 amcheck 명령으로 설정에 문제가 없는지 점검을 하고 문제가 없다면 백업을 실행할 수 있는 것이다.

백업은 amdump 명령으로 실행시킨다. 실행이 되면 disklist 파일로부터 무엇을 백업할 것인지 확인한 다음, 백업해야 할 데이타 용량을 추정 계산하기 시작한다. disklist 파일은 사용자가 작성해야 하며 여기에는 백업할 대상(백업 서버의 디렉토리)의 목록이 저장되어 있다.

처음 백업하는 디렉토리라면 실제 용량에 0.5를 곱한 값을 백업할 용량으로 산정한다. 기존에 백업했던 디렉토리는 압축률이 지난 번 백업에서 결정된 것이 있으므로 그 값을 사용하여 백업할 용량을 계산한다.

용량을 추정 계산 하면서 백업할 각 대상의 백업 레벨도 결정한다. 그 대상이 기존에 백업한 적이 없다면 Full 백업(이하 전체백업이라 칭함)을 할 것이고 그렇지 않다면 Amanda가 알아서 Incremental 백업(이하 증분 백업이라 칭함)의 레벨을 결정한다. 물론 사용자가 백업 레벨을 설정할 수도 있다. 백업 레벨에 대한 것은 8.6절을 참고하기 바란다.

모든 백업 대상에 대한 용량 계산이 끝난 후 데이타를 받기 시작하며 그 데이타를 테잎서버의 로컬디스크에 저장한다. 로컬디스크에 저장이 완료된 백업 대상부터 테잎으로 쓰기 시작한다. 설정하기에 따라서는 테잎으로 바로 쓸 수도 있다. 테잎으로 바로 쓰는 것에 대해서는 8.7절을 참고하기 바란다. 어쨌든 Amanda는 데이타를 받아서 로컬디스크에 저장했다가 테잎으로 쓰는 방식을 취하고 있다는 것이다. 각 백업 대상의 데이타를 받아서 나름의 형식으로 각각 하나의 파일로 만드는데 이때 파일들을 실제로 묶어주는 것은 tar를 이용한다. 또는 tar 대신에 dump를 사용할 수도 있으나 여기서는 tar를 사용하는 것을 기준으로 설명하겠다.

amdump를 실행시킨 후 백업 진행 상태는 amstatus 명령으로 확인할 수 있다. 백업이 완료되면 Amanda 설정 파일에 지정되어 있는 이메일 주소로 완료된 백업에 대한 정보가 보내지며 그 안에는 에러 상황도 같이 표시된다.

백업의 순서를 다시 한 번 정리해 보면,

1. amlabel 명령으로 테잎에 라벨을 붙인다.
2. amcheck 명령으로 설정의 이상 유무를 점검한다.
3. amdump 명령으로 백업을 실행시킨다.
4. amstatus 명령으로 백업 진행 상황을 점검한다.

4. 테잎 장치의 설치 및 작동

테잎 장치의 물리적인 연결은 각자 알아서 하기로 하고 여기서는 일반적인 SCSI 테잎 장치의 인식 여부 및 작동 여부에 대해 설명하도록 하겠다.

SCSI 테잎 장치가 제대로 인식되었는지 확인하려면, `/proc/scsi/scsi` 파일을 읽어보기 바란다. 필자 시스템의 `/proc/scsi/scsi` 파일을 읽어보면,

```
$ cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 03 Lun: 00
  Vendor: HP       Model: C5683A           Rev: C005
  Type:   Sequential-Access                ANSI SCSI revision: 02
```

HP의 모델명 C5683A라는 테잎 장치가 인식되어 있음을 알 수 있다.

보유하고 있는 테잎 장치의 모델이 표시되면 정상적으로 인식된 것이다. 만일, 인식이 안된다면 커널에서 SCSI 관련 부분이 제대로 설정이 안된 것일 가능성이 많으니 커널을 새로이 컴파일 해야 할 것이다. 커널 컴파일과 SCSI 장치에 대한 것은 The Linux Kernel HOWTO (http://www.tldp.org/HOWTO/Kernel-HOWTO.html) 문서와 The Linux 2.4 SCSI subsystem HOWTO (http://www.tldp.org/HOWTO/SCSI-2.4-HOWTO/index.html) 문서를 각각 참고하기 바란다.

테잎 장치가 제대로 작동하는 지 확인해 보기 위해 tar 명령으로 테잎에 파일을 써 넣어 보자. 테잎 장치에 쓸 수 있는 권한은 root에게 있을 것이므로 root로 로그인해서 tar 명령을 실행해야 할 것이다.

`/home/styoo`를 백업해 보자. 테잎 장치가 하나 뿐이라면 테잎 장치는 `/dev/st0`가 될 것이다.

```
# tar cvf /dev/st0 /home/styoo
```

제대로 실행이 되는가? 제대로 된다면 백업한 목록을 한 번 확인해 보자.

```
# tar tvf /dev/st0
```

백업한 것을 /tmp 디렉토리에 복원해 보자.

```
# cd /tmp
# tar xvf /dev/st0
```

정상적으로 작동하는가? 그렇다면 테잎 장치의 설치와 작동이 정상적이므로 Amanda를 이용하여 백업할 준비가 되었다고 하겠다. 테잎 장치가 제대로 인식만 된다면 tar를 이용한 백업이나 복원에는 별다른 문제가 없을 것이다. 이 단계에서 문제가 발생한다면 주위 사람들에게 물어보거나 HOWTO 문서를 참고하여 해결하기 바란다.

5. AMANDA가 사용하는 파일들

Amanda가 작동하는데 필요한 파일이 여러개 있다. 그 파일들에 대해서 알아보자. Amanda는 필요에 따라서 여러 가지 설정을 가질 수 있다. 설정 파일의 최상위 디렉토리는 `/etc/amanda`이며 그 밑에 서브디렉토리를 만들어서 입맛에 맞도록 설정을 변경할 수 있다.

예를 들어, 프로젝트 명 SpaceWar와 EarthBurn이라는 두 개의 프로젝트를 수행하고 있다고 하자. 각 프로젝트 별로 따로 백업을 관리할 계획이라면 `/etc/amanda/SpaceWar`와 `/etc/amanda/EarthBurn`라는 두 개의 디렉토리를 만들어서 그 아래에 필요한 필요한 설정 파일을 넣어 둘 수 있다. 그리고 공통으로 사용하는 설정은 `/etc/amanda` 디렉토리에 넣어두면 된다.

테잎서버에서 사용하는 파일들은 아래와 같다. 각각의 위치는 Amanda 소스로부터 바이너리를 빌드할 때 설정하기에 따라 바뀔 수 있다.

`/etc/amanda/amanda.conf`

테잎 장치에 대한 설정, 동시에 진행될 백업 프로세스의 수등과 같은 공통적으로 사용되는 설정에 대한 정의를 포함하고 있다. 반드시 이 파일이 필요한 것은 아니다. 다만 공통되는 설정을 한 곳에 모아두는 것이 편하다는 필자의 생각에 따른 것일 뿐이다.

`/etc/amandahosts`

테잎서버에서 파일서버로 액세스가 가능하도록 테잎 서버의 호스트네임과 사용자 이름을 저장하고 있다.

`/etc/amandapass`

윈도우 파일 서버의 데이타를 samba를 이용하여 백업할 경우, 공유 이름과 접근 권한을 가진 사용자 이름 그리고 그 사용자의 암호에 대한 목록을 저장하는 파일이다.

`/etc/amanda/SpaceWar/amanda.conf`

이 절의 처음에서 예를 든, 가상의 SpaceWar라는 프로젝트에 대한 설정 파일이며 그 안에 `/etc/amanda/amanda.conf` 파일을 부르는 설정이 들어 있으며, 테잎 라벨에 대한 설정, 로그 파일의 위치 등을 포함한다.

`/etc/amanda/SpaceWar/disklist`

이 파일에 SpaceWar라는 프로젝트의 백업할 디렉토리와 그 디렉토리가 있는 파일 서버의 목록 그리고 백업 형태를 지정한다.

`/var/lib/amanda/amandates`

`/etc/amanda/*/disklist` 파일에서 정의된 백업 목록중에서 백업이 완료된 것에 대하여 그 디렉토리 이름과 백업 레벨 그리고 백업된 날짜를 저장하고 있다.

`/var/lib/amanda/gnutar-lists/`

이 디렉토리 밑에 백업된 디렉토리(또는 파일시스템)에 대한 상태가 저장된 파일들이 있으며 이를 참조로 다음 증분 백업 할 때에 포함할 파일들이 결정된다. 이 디렉토리에 있

는 파일들은 tar에 의해 생성된다.

`/var/lib/amanda/SpaceWar/curinfo/`

백업된 각 디렉토리에 대한 히스토리가 저장된다.

`/var/lib/amanda/SpaceWar/index/`

흔히 catalogue라고 하는 백업된 파일 목록이 저장된다.

`/var/lib/amanda/SpaceWar/tapelist`

사용할 테잎 목록을 저장하고 있다.

`/var/log/amanda/SpaceWar/log/`

백업 로그가 저장된다.

6. AMANDA 설정 하기

6.1. 필자의 시스템 구성

들어가기 전에, 필자가 사용하는 테잎 장치, 리눅스 서버 그리고 설정에 필요한 가정 등에 대한 얘기를 잠깐하고자 한다.

우선, 테잎 장치는 모델명 ULT3581인 IBM Ultrium Tape Autoloader이다. 1 개의 드라이브와 7개의 슬롯을 가진 제품이다. 미디어 하나의 용량은 비압축 상태로 100GB이며 50% 압축이 된다고 한다면 최대 200GB까지 하나의 미디어에 저장할 수 있다. 테잎에 데이타를 기록하는 속도도 평균 14MB 이상이 나오며 윈도우에서는 20MB 속도가 나오는 것을 확인하였다. 특정 제품 선전을 위한 것이 아님은 두 말할 나위도 없다. DAT, DLT 등에 비해 월등히 빠른 속도를 보여주고 있으며 Super DLT와는 비교를 하지 않아서 아는 바가 없다.

리눅스는 Debian 3.0이며 커널은 버전 2.4.18을 사용하고 있다. 커널은 소스로부터 따로 컴파일 하였다. 위 테잎 장치에 대한 디바이스 드라이버를 IBM에서 제공하고 있으며 RedHat 7.1/7.2에서 사용할 수 있고 RedHat에서 제공하는 2.4.2 커널 또는 kernel.org에서 제공하는 2.4.9 커널을 지원한다고 한다. 그러나 반드시 IBM에서 제공하는 디바이스 드라이버를 사용할 필요는 없다. 필자는 2.4.18의 sg 모듈과 st 모듈을 이용하고 있으며 아직 아무런 문제점을 발견하지 못하였다. IBM의 드라이버를 사용하였을 때 속도가 더 향상되는 지는 모르겠으나, 이 글을 쓰는 시점에 옛 기억을 더듬어 보면 별 차이가 없었던 것으로 기억된다.

5절에서 언급한 적이 있지만, 프로젝트명 SpaceWar에서 사용하는 파일들을 백업한다는 가정하에 설정 부분을 진행할 것이다. 그리고 아래에서 설명할 설정은 되도록이면 일반적으로 적용할 수 있도록 하겠지만 특정 부분은 위에서 언급한 필자의 시스템 구성에 맞추어서 설명하도록 하겠다.

6.2. `/etc/amanda/amanda.conf` 파일 설정

아마도 Amanda Server s/w를 설치하면 `/etc/amanda/DailySet1/amanda.conf`파일이 있을 것이다. 이 파일을 `/etc/amanda/amanda.conf`로 복사한 후 편집을 하도록 하자. `/etc/amanda/amanda.conf`에 설정할 여러가지 파라미터가 있다. 필요한 것들을 하나씩 알아보자.

mailto

백업이 이루어진 결과에 대한 정보를 메일로 받을 사용자를 지정한다.

mailto "aaa@bbb.ccc.co.kr"

dumpuser

백업을 실행할 테잎서버의 사용자를 지정한다.

dumpuser "styoo"

2절에서 언급한 대로 패키지를 Amanda 사용자를 변경하여 새로이 빌드하지 않고, 제공되는 그대로 설치하였다면 dumpuser는 backup이 될 것이다. 또는 root가 될 수도 있다.

etimeout

파일서버의 백업할 각 디렉토리의 용량을 측정하는데 할당가능한 최대 시간(단위:초).

etimeout 3600

파일서버의 백업할 디렉토리의 용량을 측정하는데 걸리는 시간이 3600초를 초과한다면, 더 이상 측정하지 않고 그 디렉토리에 대한 백업을 진행하지 않는다.

dtimeout

네트웍 불량 등의 이유로 파일서버로부터 백업할 데이타를 받지 못할 경우 대기할 최대 시간(단위:초).

dtimeout 3600

백업이 진행되기 전 또는 백업 진행 중에 어떤 이유로 파일서버로부터 데이타를 받지 못할 경우, 3600초 동안 대기하다가 그 이상 시간이 초과되면 백업을 진행하지 않는다.

tapedev

테잎 장치를 지정한다.

tapedev "/dev/nst0"

테잎 장치가 `/dev/st0`가 아니라 왜 `/dev/nst0`인지 궁금할 수도 있다. 그것은 Amanda가 non-rewinding 테잎 장치를 사용하기 때문이다. non-rewinding 테잎 장치는 "n"을 붙여서 표시한다. 따라서 `/dev/st0`이 아니라 `/dev/nst0`인 것이다. `/dev/st0`와 `/dev/nst0`의 차이는 tar 명령으로 쉽게 확인할 수 있다.

/usr/bin디렉토리를 테잎에 백업한 다음, /tmp에 복원하는 경우를 가정하자. 먼저, /dev/st0에 쓰는 경우를 보자.

```
$ tar cvf /dev/st0 --directory /usr/bin .
```

백업이 끝나면 /tmp 디렉토리로 이동해서 복원을 해 보자.

```
$ cd /tmp
$ tar xvf /dev/st0
```

테잎이 자동으로 rewind 된 후에 백업된 파일들이 복원될 것이다. 이제, /dev/nst0에 백업하는 경우를 보자.

```
$ tar cvf /dev/nst0 --directory /usr/bin .
```

백업이 끝났으면 복원을 해 보자.

```
$ cd /tmp
$ tar xvf /dev/nst0
```

파일이 제대로 복원 되는가? 아마도 복원되는 파일이 전혀 없거나 에러가 발생할 것이다. 테잎에 데이타를 써 넣은 후 테잎이 rewind 되지 않았기 때문이다. 테잎을 rewind 시킨 후 다시 복원을 시도해 보자.

```
$ mt -f /dev/nst0 rewind
$ tar xvf /dev/nst0
```

파일이 복원되는가? 아마도 잘 될 것이다. 비슷하게 /dev/st0에 데이타를 백업한 후 이어서 또 다른 데이타를 백업하면 테잎이 처음으로 rewind된 후 데이타를 덮어쓰게 된다. 그러나 /dev/nst0를 사용하면 뒤에 이어서 데이타를 써 넣는다. Amanda는 각 백업 대상을 각각 하나의 파일로 만든 후 테잎에 써넣는 방식을 취하고 있으므로 만들어진 각 백업 대상의 각 파일을 차례 차례 테잎에 써넣어야 한다. 따라서 non-rewinding 테잎 장치인 /dev/nst0를 사용하는 것이다.

changerdev

autoloader와 같은 미디어 체인저가 있으면 지정한다. 아래의 tpchanger에 따라서 설정할 필요가 없는 경우도 있다. 필자가 사용하는 IBM 3581 autoloader는 이를 지정할 필요가 없었다.

changerdev "/dev/sg0"

tpchanger

autoloader와 같은 미디어 체인저가 있으면 이를 구동할 스크립트를 지정한다. IBM 3581 autoloader의 경우 mtx 패키지에 들어 있는 mtx-changer 스크립트를 약간 수정하니 그럭저럭 구동하였다.

tpchanger "mtx-changer"

tapetype

사용할 테잎의 종류를 지정한다.

tapetype IBM-LTO

위에서 예로 든 IBM-LTO는 Amanda에서 지정된 이름이 아니라 사용자가 임의로 지정하는 것이다. IBM-LTO가 어떤 테잎인가에 대한 정의를 따로 해 주어야 한다.

columnspec

백업 프로세스가 끝난 후, Amanda는 메일로 그 정보를 보내준다. 그때 사용할 각 목록의 길이 등을 지정한다.

columnspec
"HostName=0:13,Disk=1:20,OrigKB=1:9,OutKB=1:9,DumpRate=1:7,TapeRate=1:7"

HostName=0:13의 의미는, 호스트네임의 표시에 최대 13자를 사용할 수 있고 "0"은 그 왼쪽에 공백이 없다는 뜻이다. 비슷하게 1:7은 해당 아이템에 사용할 수 있는 글자의 수는 7개 이며 그 왼쪽에 공백을 하나 넣어라는 의미이다. 그리고 각 아이템의 순서대로 메일에 표시된다.

holdingdisk

3절에서 설명했다시피 Amanda는 데이타를 로컬디스크에 저장했다가 다시 테잎으로 옮겨 쓰는 방식을 취하고 있다. 이 로컬디스크를 holdingdisk라고 부른다.

```
holdingdisk hd1 {
comment "main holding disk"
directory "/backup-image/Image"
use -10Mb
chunksize 0
}
```

위에서 "use -10Mb"의 의미는 10Mb를 제외한 나머지 공간 모두를 Amanda가 사용한다는 뜻이다. 양수 값을 지정하면 그 크기 만큼만 사용하라는 의미이다. 그리고 Amanda는 데이타를 받아서 백업이미지를 만들때 하나의 파일로 만든다. 이때 OS 차원에서 파일 크기에 제한이 있고 백업이미지가 그 제한 크기보다 크다면 문제가 발생할 것이다. 이를 방지하기 위해서 chunksize 값을 지정할 수 있다. "chunksize 0"는 이미지 크기를 OS에 맡긴다는 의미이다. "chunksize 2Gb"라고 지정하면 백업이미지를 2Gb 크기로 쪼개라는 뜻이 된다.

define tapetype

위에서 지정한 tapetype을 정의한다.

```
define tapetype IBM-LTO {
comment "just produced by tapetype program"
length 101376 mbytes
```

```
filemark 0 kbytes
speed 13500 kps
}
```

사용하는 테잎의 종류는 테잎 장치에 따라서 결정되므로 테잎 장치가 무엇이냐에 따라 위의 정의는 달라질 것이다. Amanda 홈페이지 (http://www.amanda.org)에서 대부분의 테잎 장치에 대한 tapetype을 구할 수 있다. tapetype을 알 수가 없다면 **tapetype** 명령으로 구할 수도 있으므로 걱정할 필요는 없다. **tapetype** 명령의 사용법은 **tapetype --help**를 참고하기 바란다. 또는 Amanda 사용자 메일링 리스트 (http://marc.theaimsgroup.com/?l=amanda-users)를 뒤져보면 구할 수 있을 것이다.

define dumptype global

```
define dumptype global {
program "GNUTAR"
compress server fast
index yes
record yes
}
```

- program "GNUTAR"는 백업 프로그램으로 tar를 사용한다는 의미이다.
- compress server fast는 테잎서버에서 백업 데이타를 압축하는데 빠른 압축 속도를 사용한다는 의미이다. Amanda는 gzip을 이용하여 압축을 하며 fast는 gzip의 압축률 옵션이다. 즉 가장 낮은 압축률을 의미한다. fast 대신에 best를 사용할 수도 있다. 그리고 server 대신에 client를 사용할 수도 있다. 그 의미는 구구절절이 설명하지 않아도 알 수 있을 것이다.
- index yes는 백업되는 파일 목록(카탈로그)를 저장하라는 의미이다.
- record yes는 `/var/lib/amanda/amandates` 를 백업할 때 업데이트하라는 의미이다.

6.3. `/etc/amanda/SpaceWar/amanda.conf` 파일 설정

SpaceWar라는 프로젝트와 관련된 데이타들을 따로 백업한다는 가정하에 그와 관련된 설정을 따로 지정할 수 있다. 먼저. `/etc/amanda` 밑에 SpaceWar라는 디렉토리를 하나 만들고 그 안에 amanda.conf 파일을 만들어 편집을 하자.

includefile

포함할 파일을 지정한다.

includefile "/etc/amanda/amanda.conf"

`/etc/amanda/amanda.conf` 파일에 공통으로 사용할 파라메타에 대한 정의가 들어 있으므로 당연히 포함되어야 한다.

org

이 백업 설정에 대한 적당한 제목을 지정한다.

org "SpaceWar"

SpaceWar라는 프로젝트에 대한 백업 설정이므로 SpaceWar라고 지정하였다.

inparallel

테잎서버에서 동시에 진행 할 수 있는 백업 프로세스의 수

inparallel 4

maxdumps

하나의 파일서버로 부터 동시에 백업을 진행할 수 있는 최대 개수

maxdumps 2

A라는 파일서버에 SpaceWar의 데이타가 저장되는 디렉토리가 4개 있다면 그 중에서 동시에 2개 디렉토리에 대한 백업을 진행할 수 있다. 전체적으로 동시에 진행되는 백업 프로세스의 수는 inparallel 파라메타에 의해 조절된다. maxdumps 파라메타는 dumptype에서 지정할 수도 있다.

runtapes

SpaceWar 프로젝트의 백업에 사용될 테잎의 개수

runtapes 2

LTO 미디어는 하드웨어 압축을 하지 않는 상태로 100Gb의 용량을 저장할 수 있으므로 runtapes 2는 200Gb의 데이타를 백업할 수 있다는 의미이다. 3절에서 언급했듯이 각 디렉토리에 대한 백업 용량을 대략 계산한 다음 그 용량이 runtapes 파라메타 값으로부터 계산되는 용량을 초과하면 일부가 백업되지 않으므로 runtapes의 값을 적당히 지정해야 할 것이다.

dumpcycle

한 번의 전체백업으로부터 다음 번 전체백업까지의 주기

dumpcycle 4 weeks

이 값은 적절히 알아서 잡아주면 될 것이다. 1개월에 한 번 정도 전체백업을 한다고 생각하면 4주 정도면 될 것이다. dumpcycle은 Amanda가 자동으로 백업 레벨을 결정하기 위해 필요한 하나의 파라메타이며, 필자는 백업 레벨과 스케줄을 수동으로 결정하기 때문에 dumpcycle 파라메타가 그리 큰 의미는 없었다. dumpcycle 값을 0으로 지정하면 항상 전체백업을 한다는 의미이며 dumpcycle 파라메타는 dumptype에서 지정할 수도 있다.

runspercycle

dumpcycle 동안에 실행될 amdump의 회수

runspercycle 4

dumpcycle이 4주이고 일주일에 한 번 백업을 실행한다면 dumpcycle 동안에 실행될 amdump의 회수는 4회이다. 수동으로 백업 스케줄을 조절할 계획이라면 이 파라메타도 큰 의미는 없다.

tapecycle

dumpcycle 동안에 사용될 테잎의 개수

tapecycle 4 tapes

dumpcycle 동안에 2개의 테잎을 가지고 데이타를 계속 덮어 쓰면서 사용할 계획이라고 하자. 이때 테잎이 에러가 날 경우를 생각해서 tapecycle은 4 정도로 해 두면 될 것이다.

labelstr

테잎 라벨을 구성할 문자열

labelstr "^SpaceWar-[0-9][0-9][0-9]*$"

amlabel 명령으로 테잎에 라벨을 써 넣을때 위 문자열의 구성에 맞아야 한다.

define dumptype

`/etc/amanda/amanda.conf` 파일에서 정의했던 dumptype global 외에 다른 dumptype을 정의한다.

```
define dumptype always-full {
       global
       comment "Full dump of this filesystem always"
       priority high
       dumpcycle 0
       maxdumps 4
       }
```

dumpcycle을 0으로 설정하여 항상 전체백업을 하도록 설정하고 dumptype global을 포함하도록 하였으며 maxdumps를 4로 지정하였다.

```
define dumptype incr-only {
       global
       comment "incremental dump only"
       priority high
       strategy nofull
       maxdumps 4
}
```

strategy를 nofull로 지정하여 항상 백업 레벨 1의 증분 백업을 하도록 설정하였다. 기타 dumptype에서 지정할 수 있는 파라메타에 대해서는 Amanda man page를 참고하기 바란다.

기타

```
infofile "/var/lib/amanda/SpaceWar/curinfo"
logdir   "/var/log/amanda/SpaceWar/log"
indexdir "/var/lib/amanda/SpaceWar/index"
tapelist "/var/lib/amanda/SpaceWar/tapelist"
```

6.4. `/etc/amanda/SpaceWar/disklist` 파일 설정

disklist 파일에는 백업할 데이타를 갖고 있는 파일서버의 hostname과 백업할 디렉토리가 저장되어 있다.

```
gmfs1 /SpaceWar1/Program-Sources always-full
gmfs1 /SpaceWar1/Graphic-Sources always-full
gmfs2 /SpaceWar2/Protype always-full
gmfs2 /SpaceWar2/Schedule always-full
```

첫 줄을 살펴보자. gmfs1은 파일서버의 호스트네임이고, `/SpaceWar1/Program-Sources`은 gmfs1에 있는 백업할 디렉토리이며 always-full은 위에서 설정했던 dumptype이다. 이와 같은 형태로 백업할 디렉토리를 disklist 파일에 적어주면 된다. 그 외에 spindle 값과 interface를 지정할 수 있으니 Amanda man page를 참고하기 바란다.

7. 주요 AMANDA 명령어

amlabel

테잎에 라벨을 써 넣는다.

```
amlabel "configure name" "label"
```

`/etc/amanda`밑에 만들어진 SpaceWar라는 디렉토리가 configure name이 되며 `/etc/amanda/SpaceWar/amanda.conf`에 정의된 labelstr이 label이 될 것이다.

```
$ amlabel SpaceWar SpaceWar-001
```

미디어 체인저가 있고 제대로 설정이 되어 있다고 할 경우, 첫번째 slot에 있는 테잎에 라벨을 써 넣는다면,

```
$ amlabel SpaceWar SpaceWar-001 slot 1
```

amcheck

설정에 문제가 없는 지 점검을 해 준다.

```
$ amcheck SpaceWar
```

amadmin

어떤 라벨을 가진 테잎을 재 사용할 것인지 안 할 것인지, 다음에 사용할 테잎은 어떤 라벨을 가진 것인지 등등 여러가지 작업을 할 수 있다.

SpaceWar의 백업에 사용될 테잎의 목록을 보여준다.

```
$ amadmin SpaceWar tape
```

SpaceWar-001이라는 라벨을 가진 테잎을 백업에 사용하지 않도록 설정한다.

```
$ amadmin SpadeWar no-reuse SpaceWar-001
```

SpaceWar-001이라는 라벨을 가진 테잎을 백업에 사용도록 설정한다.

```
$ amadmin SpadeWar reuse SpaceWar-001
```

amdump

실제 백업을 실행시킨다.

```
$ amdump SpaceWar
```

amstatus

백업을 실행시킨 후 진행 상황을 모니터 한다.

```
$ amstatus SpaceWar
```

amrecover, amrestore

백업된 데이타를 복원한다. root의 권한으로 실행시킨다.

```
# amrecover SpaceWar
```

amflush

백업된 데이타가 어떤 이유로 테잎에 옮겨지지 않고 로컬디스크 에 저장된 것을 테잎으로 다시 옮기고자 할 때 사용한다.

```
$ amflush SpaceWar
```

amcleanup

실패한 백업이 있을 때 이를 청소하기 위해 사용된다.

```
$ amcleanup SpaceWar
```

amdump 명령으로 백업을 실행시켰으나 어떤 이유로 백업에 실패한 경우, 반드시 amcleanup 명령으로 관련 파일을 청소해 주어야한다. 그리고 로컬디스크에 임시로 저장되어 있는 백업 데이타도 지워 주어야 한다.

위에서 열거한 명령어는 8절에서 자세히 살펴볼 것이다. 나머지 명령어에 대한 것은 Amanda man page를 참고하기 바란다.

8. AMANDA를 이용한 실전 백업

몇 가지 가정을 하고 실제 백업을 해 보자.

- Amanda 패키지는 2절에서 설명한대로 configure의 사용자 옵션을 바꾸어서 설치한다.
- 테잎 장치가 제대로 작동한다.
- 테잎 장치는 IBM 3581 autoloader를 사용한다.
- 백업을 실행할 테잎서버의 사용자는 styoo이다.
- SpaceWar라는 프로젝트의 데이타를 백업한다.
- 테잎서버의 호스트네임은 backupsvr이다.
- 프로젝트 SpaceWar의 데이타는 파일서버 gmfs1과 gmfs2에 저장된다.
- gmfs1의 OS는 윈도우 2000 Server이고, gmfs2는 리눅스이다.
- 프로젝트 SpaceWar와 관련된 각 파일서버의 디렉토리는 아래와 같다.

```
gmfs1 :   E:\SpaceWar1\Program-Sources
gmfs1 :   E:\SpaceWar1\Graphic-Sources
gmfs2 :   /SpaceWar2/Prototype
gmfs2 :   /SpaceWar2/Schedule
```

- gmfs1의 `E:\SpaceWar1` 디렉토리를 backupsvr의 `/SpaceWar1`디렉토리에 SMB 마운트한다.

파일서버와 테잎서버는 서로 호스트네임을 알 수 있도록 `/etc/hosts` 파일에 기록해 두는 것이 좋다.

8.1. 실전 백업용 /etc/amanda/amanda.conf 파일

```
mailto "styoo"

dumpuser "styoo"

etimeout 3600
dtimeout 3600

tapedev "/dev/nst0"

tpchanger "mtx-changer"

tapetype IBM-LTO

column-
spec "HostName=0:13,Disk=1:40,OrigKB=1:9,OutKB=1:9,DumpRate=1:7,TapeRate=1:7"

holdingdisk hd1 {
    comment "main holding disk"
    directory "/backup-image/Image"
    use -10Mb
    chunksize 0
    }

define tapetype IBM-LTO {
comment "just produced by tapetype program"
length 101376 mbytes
filemark 0 kbytes
speed 13500 kps
}

define dumptype global {
    program "GNUTAR"
    compress server fast
    index yes
    record yes
}
```

autoloader를 구동하는데 사용할 mtx-changer 스크립트는 mtx 패키지에 포함되어 있는 것을 사용하였으나, IBM 3581 autoloader에 적용하기에는 일부 작동에 문제가 있어 필자가 일부를 수정하였다. 수정된 스크립트는 에 있으니 10절을 참고하기 바란다.

8.2. 실전 백업용 /etc/amanda/SpaceWar/amanda.conf 파일

```
includefile "/etc/amanda/amanda.conf"

org "SpaceWar"

inparallel 4

runtapes 2

dumpcycle 4 weeks

runspercycle 4

tapecycle 4 tapes

maxdumps 4

labelstr "^SpaceWar-[0-9][0-9][0-9]*$"

infofile "/var/lib/amanda/SpaceWar/curinfo"
logdir   "/var/log/amanda/SpaceWar/log"
indexdir "/var/lib/amanda/SpaceWar/index"
tapelist "/var/lib/amanda/SpaceWar/tapelist"

define dumptype always-full {
    global
    comment "Full dump of this filesystem always"
    priority high
    dumpcycle 0
}

define dumptype incr-only {
    global
    comment "Incremental dump only"
    priority high
    strategy nofull
}
```

8.3. 실전 백업용 /etc/amanda/SpaceWar/disklist 파일

```
localhost   /SpaceWar1/Program-Sources      always-full
localhost   /SpaceWar1/Graphic-Sources      always-full
gmfs2       /SpaceWar2/Prototype            always-full
```

```
gmfs2       /SpaceWar2/Schedule             always-full
```

`/Spacewar1`의 하위 디렉토리는 윈도우 2000 서버인 gmfs1에 있으므로 samba를 이용하여 백업해야 한다. amanda는 윈도우 머신의 데이타를 백업하기 위해 smbclient를 사용하여 백업을 진행한다. 그러나 그렇게 하지 않고, 윈도우의 디스크를 테잎서버에 SMB 마운트하여 마치 로컬디스크를 백업하듯이 할 수도 있다. 여기서는 SMB 마운트하여 로컬디스크를 백업하듯이 할 것이다. samba를 이용하여 백업을 진행한다면, localhost를 gmfs1으로 바꾸고 `/etc/amandapass` 파일에 공유이름과 해당 공유를 액세스 할 수 있는 사용자 그리고 암호를 적어주어야 한다. 그 포맷은 Amanda man page를 참고하기 바란다.

8.4. 백업 시작

백업을 진행할 사용자는 styoo이며 관련된 파일에 대한 쓰기 권한이 있어야 함은 두 말할 나위도 없다. `/etc/amanda`디렉토리 및 그 하위 디렉토리와 파일에 대한 읽기쓰기 권한이 styoo에게 있어야 할 것이다. 뿐만 아니라, `/var/lib/amanda` 디렉토리 및 그 하위의 파일들 그리고 `/var/log/amanda` 디렉토리 및 그 하위의 파일들에 대한 읽기쓰기 권한이 필요하다. 또한, `/dev/nst0`, `/dev/sg0`에 대한 읽기쓰기 권한도 당연히 있어야 할 것이다.

이제, 백업을 진행해 보자. Amanda를 설치하고 위와 같이 설정을 끝마친 후 Amanda 명령어를 이용하여 백업을 진행할 때 흔히 발생할 수 있는 에러 상황도 함께 취급할 것이다.

테잎 장치에 빈 테잎을 하나 넣고 amlabel 명령으로 테잎에 라벨을 붙이자. -f 옵션을 붙여서 강제로 라벨을 써 넣도록 하자.

```
$ amlabel -f SpaceWar SpaceWar-001
rewinding, reading label, not an amanda tape
rewinding, writing label SpaceWar-001, checking label
amlabel: couldn't write tapelist: No such file or directory
```

`tapelist` 파일이 없다고 불평을 한다. 이것은 `/var/lib/amanda/SpaceWar` 디렉토리가 없기 때문이니 `/var/lib/amanda/` 디렉토리에 SpaceWar 디렉토리를 만들어 주면 된다. 이렇게 해 주면 amlabel 명령이 그 밑에 tapelist 파일을 자동으로 만든다.

테잎에 라벨을 써 넣은 후, amadmin 명령으로 사용할 테잎 목록을 살펴보자.

```
$ amadmin SpaceWar tape
The next Amanda run should go onto a new tape.
The next Amanda run should go onto a new tape.
```

amlabel 명령으로 SpaceWar-004까지 테잎에 라벨을 붙여 보자. 그리고서 `/var/lib/amanda/SpaceWar/tapelist` 파일을 보면 다음과 같이 되어 있을 것이다.

```
0 SpaceWar-004 reuse
0 SpaceWar-003 reuse
```

```
        0 SpaceWar-002 reuse
        0 SpaceWar-001 reuse
```

amadmin 명령으로 사용할 테잎 목록을 다시 한 번 살펴보자.

```
        $ amadmin SpaceWar tape
        The next Amanda run should go onto tape SpaceWar-
001 or a new tape.
        The next Amanda run should go onto tape SpaceWar-
002 or a new tape.
```

runtapes과 tapecycle의 값에 따라서 그리고 tapelist 파일의 내용에 따라서 위 명령의 결과는 달라진다. 한 번 체크해 보기 바란다.

amcheck 명령으로 설정에 이상이 없는지 확인해 보자. 데비안의 경우, amanda 명령어들이 `/usr/sbin` 디렉토리에 있는데 amcheck 명령의 실행 권한이 root와 backup에게만 있을 수도 있으니 실행 권한을 추가해 주어야 할 것이다.

```
        $ amcheck SpaceWar
        ERROR: program /usr/lib/amanda/planner: not executable
        Holding disk /backup-
image/Image: 6636356 KB disk space available, using 6626116 KB
        ERROR: log dir /var/log/amanda/SpaceWar: not writable
        NOTE: skipping tape-writable test
        Tape SpaceWar-001 label ok
        NOTE: info dir /var/lib/amanda/SpaceWar/curinfo: does not exist
        NOTE: it will be created on the next run
        NOTE: index dir /var/lib/amanda/SpaceWar/index: does not exist
        Server check took 0.010 seconds

        Amanda Backup Client Hosts Check
        --------------------------------
        ERROR: gmfs2: [access as styoo not al-
lowed from styoo@backupsvr] amandahostsauth failed
        WARNING: localhost: selfcheck request timed out.  Host down?
        Client check: 2 hosts checked in 30.018 seconds, 2 problems found
```

몇 가지 에러와 경고 메세지가 보인다. 하나씩 살펴보자.

- "ERROR: program /usr/lib/amanda/planner: not executable" 여기서 planner라는 것은 amdump 명령어가 실행된 후 amdump의 부름을 받고 실행되는 명령어이다. 이 외에도 몇 가지 다른 명령어가 있는데 데비안의 경우, 이들 명령어는 `/usr/lib/amanda` 디렉토리에 존재한다. 실행 권한이 root 또는 backup에게만 있을 수 있으니 다른 사용자에게 실행 권한이 없는 명령어를 찾아서 실행 권한을 부여해야 할 것이다.

- "ERROR: log dir /var/log/amanda/SpaceWar: not writable" 이것은 SpaceWar 디렉토리가 없기 때문이니 디렉토리를 만들어 준다. 그 밑에 표시된 curinfo와 index가 없다는 메세지도 마찬가지로 `/var/lib/amanda` 디렉토리 밑에 SpaceWar를 만들어 준다.
- "ERROR: gmfs2: [access as styoo not allowed from styoo@backupsvr] amandahostsauth failed" 이 메세지는 backupsvr의 사용자 styoo에게 파일서버 gmfs2(리눅스 머쉰임)의 amandad에 접근할 권한이 없기 때문에 발생한다. gmfs2의 `/etc/amandahosts` 파일에

```
backupsvr styoo
```

라고 한 줄을 넣어주면 될 것이다.

- "WARNING: localhost: selfcheck request timed out. Host down?" 이 메세지는 amandad(Amanda 데몬)가 다른 사용자의 권한으로 실행되었기 때문이다. `/etc/inetd.conf`을 열어보면 아래와 같은 내용이 있을 것이다.

```
amanda dgram udp wait backup /usr/sbin/tcpd /usr/lib/amanda/amandad
amandaidx stream tcp nowait backup /usr/sbin/tcpd /usr/lib/amanda/amindexd
amidx-
tape stream tcp nowait backup /usr/sbin/tcpd /usr/lib/amanda/amidxtaped
```

위에서 backup을 실제 백업을 실행하는 사용자(여기서는 styoo)로 바꾸어 준 후 inetd를 재 시동 시키면 될 것이다.

- 위 문제가 해결되면 다음과 같은 에러가 발생할 수도 있다. "ERROR: localhost: [access as styoo not allowed from styoo@localhost] open of /home/styoo/.amandahosts failed" 이것은 말 그대로 `/home/styoo/.amandahosts`을 열 수 없다는 뜻이다. `/etc/amandahosts` 파일을 `/home/styoo/.amandahosts`에 링크를 시켜주면 된다.
- 위 문제가 해결되면 다음과 같은 에러가 발생할 수도 있다. "ERROR: localhost: [access as styoo not allowed from styoo@localhost] amandahostsauth failed" 이것은 gmfs2에 접근할 수 없다는 에러와 유사한 것으로 테잎서버(backupsvr)의 `/etc/amandahosts` 파일에

```
localhost styoo
```

라고 한 줄을 넣어주면 될 것이다.

- autoloader와 같은 미디어 체인저가 없는 경우, 다음과 같은 경고 메세지가 나타날 수도 있다. "WARNING: if a tape changer is not available, runtapes must be set to 1" 이것은 autoloader와 같은 미디어 체인저가 없기 때문에 runtape 파라메타 값이 1로 세팅되어야 한다는 뜻이다. 현재는 2로 설정되어 있다. 이 경고는 무시해도 좋다.

amcheck 명령으로 이상이 없는 것을 확인하였으면, 이제 백그라운드로 amdump 명령을 실행시켜 백업을 해 보자.

```
$ amdump SpaceWar &
```

amstatus 명령으로 백업 상황을 확인해 보자.

```
$ amstatus SpaceWar

        Us-
ing /var/log/amanda/SpaceWar/amdump from 토  9월 14 14:51:50 KST 2002

        localhost:/SpaceWar1/Program-
Sources                getting estimate
        localhost:/SpaceWar1/Graphic-
Sources                getting estimate
        gmfs2:/SpaceWar2/Prototype                         get-
ting estimate
        gmfs2:/SpaceWar2/Schedule                          get-
ting estimate



        SUMMARY          part     real estimated
                          size      size
        partition       :   4
        estimated       :   0                 0k
        failed          :   0                 0k           (  0.00%)
        wait for dumping:   0                 0k           (  0.00%)
        dumping to tape :   0                 0k           (  0.00%)
        dumping         :   0        0k       0k (  0.00%) (  0.00%)
        dumped          :   0        0k       0k (  0.00%) (  0.00%)
        wait for writing:   0        0k       0k (  0.00%) (  0.00%)
        writing to tape :   0        0k       0k (  0.00%) (  0.00%)
        failed to tape  :   0        0k       0k (  0.00%) (  0.00%)
        taped           :   0        0k       0k (  0.00%) (  0.00%)
        all dumpers active
        taper idle
```

백업할 디렉토리의 용량을 계산하고 있는 중이다. 잠시 후에 다시 한 번 amstatus 명령을 실행시켜보자.

```
        $ amstatus SpaceWar
          Us-
ing /var/log/amanda/SpaceWar/amdump from 토  9월 14 15:06:33 KST 2002

          localhost:/SpaceWar1/Program-Sources     0 no estimate
          localhost:/SpaceWar1/Graphic-Sources     0 no estimate
          gmfs2:/SpaceWar2/Prototype               0 [dump larger than tape, but can-
not incremental dump new disk]
          gmfs2:/SpaceWar2/Schedule                0 14585k dump-
ing     4832k ( 33.13%) (15:06:36)

          SUMMARY          part     real estimated
                         size      size
```

```
partition       :  4
estimated       :  2              14585k
failed          :  3                  0k           (  0.00%)
wait for dumping:  0                  0k           (  0.00%)
dumping to tape :  0                  0k           (  0.00%)
dumping         :  1     4832k    14585k ( 33.13%) (  6.42%)
dumped          :  0        0k        0k (  0.00%) (  0.00%)
wait for writing:  0        0k        0k (  0.00%) (  0.00%)
writing to tape :  0        0k        0k (  0.00%) (  0.00%)
failed to tape  :  0        0k        0k (  0.00%) (  0.00%)
taped           :  0        0k        0k (  0.00%) (  0.00%)
3 dumpers idle  : start-wait
taper idle
network free kps:      270
holding space   :  6611460k ( 99.78%)
```

위의 결과를 보면, gmfs2의 디렉토리 중에서 `/SpaceWar2/Schedule`은 뭔가가 진행되고 있는데, `/SpaceWar2/Prototype`은 dump가 테잎보다 크다고 말을 하고 있다. 이것은 `/SpaceWar2/Prototype`의 용량이 크다기 보다는 각 디렉토리에 대한 백업 용량을 모두 합산한 결과 백업할 용량이 <테잎 한 개의 용량 X runtapes의 값> 보다 크다는 뜻이다. runtapes의 값을 증가 시키면 대부분 해결될 것이다. 물론, 실제로 디렉토리 하나의 백업 용량이 테잎 한 개의 용량 보다 클 수도 있다. 그러나 Amanda는 이를 문제 삼지는 않는다. 그렇다고 어떤 디렉토리에 대한 백업이미지가 테잎 하나의 크기 보다 크다고 해서 다음 테잎에 이어쓸 수는 없다. Amanda는 다음 테잎에 이어쓰기를 지원하지 않음을 명심하기 바란다. Amanda 패키지의 FAQ 문서를 참고하기 바란다.

그리고 윈도우 서버 gmfs1의 디렉토리를 SMB 마운트한 디렉토리는 "no estimate" 라고 되어 있다. 이 문제는 대개는 퍼미션 문제이다. Amanda 패키지를 빌드할 때 styoo라는 사용자로 옵션을 주었는데, 막상 실행할 때는 엉뚱한 사용자로 amdump를 실행했기 때문이다. 제공되는 패키지 그대로 설치했다면 root 또는 backup 사용자의 권한으로 백업을 실행해야 할 것이다.

웬만한 에러 상황은 체크를 했으니 백업이 잘 완료되었다고 하자. 그리고 1 개의 테잎을 백업에 사용했다고 하자. 이제 다시 한 번 amadmin 명령으로 사용할 수 있는 테잎 목록을 살펴보자.

```
        $ amadmin SpaceWar tape
The next Amanda run should go onto tape SpaceWar-002 or a new tape.
The next Amanda run should go onto tape SpaceWar-003 or a new tape.
```

다음 백업에는 SpaceWar-002 테잎부터 사용하게 될 것이다. 다음 번(두번째) 백업에도 1 개의 테잎을 사용한다면 세번째 백업에는 SpaceWar-003 테잎을 사용할 것이다. 그 다음은 SpaceWar-004, 그 다음은? 그렇다. 다시 돌아와서 SpaceWar-001을 사용하게 된다. 다섯번째 백업에 들어가기 전에 아래와 같이 amadmin 명령으로 SpaceWar-001 테잎을 백업에 사용하지 못하게 해 놓았다면, SpaceWar-002 테잎을 사용할 것이다.

```
amadmin SpaceWar no-reuse SpaceWar-001
```

테잎 순서에 대한 더 자세한 것은 amanda를 사용하면서 익히기 바란다.

백업된 데이타를 테잎에 쓰는 중에 에러가 발생한다면 다음 테잎에 데이타를 다시 써 넣으려 할 것이다. 그러나 사용할 다음 테잎이 없다면 백업 데이타는 로컬디스크에 남게되며 백업은 종료된다. 이런 경우에는 새로운 테잎을 준비하여 라벨을 붙인 다음, amflush 명령으로 로컬디스크의 데이타를 테잎에 옮겨 놓을 수 있다.

```
$ amflush SpaceWar
```

테잎으로 데이타를 옮겨 쓴 다음에는 amcleanup 명령으로 청소를 해 주어야 한다.

```
$ amcleanup SpaceWar
```

8.5. 복원하기

복원을 해 보자. 복원은 amrecover와 amrestore 두 가지를 이용할 수 있다. 여기서는 amrecover를 사용하기로 한다. 복원은 root의 권한으로 실행되어야 한다.

gmfs2의 `/SpaceWar2/Schedule` 디렉토리 밑에 있던 `200206-Prog` 이란 디렉토리와 `200206-Mail` 디렉토리를 하위 디렉토리를 포함하여 모두 복원한다고 하자. `/tmp`에 복구한다고 가정하자.

```
# cd /tmp
# amrecover SpaceWar -d /dev/nst0
```

위와 같이 명령을 주면 다음과 같은 프롬프트가 나타날 것이다.

```
amrecover>
```

또는 아래와 같은 에러메세지가 나올 수도 있다.

```
220 backupsvr AMANDA index server (2.4.2p2) ready.
500 Access not allowed: [access as styoo not al-
lowed from root@localhost] amandahostsauth failed
```

`/etc/amandahosts`에 아래와 같이 root를 등록한다.

```
localhost root
```

에러가 제거 되었으면 다시 amrecover를 실행시켜 `/SpaceWar2/Schedule`이 있는 호스트를 지정하고 복원할 최상위 디렉토리를 지정한다.

```
amrecover> sethost gmfs2
amrecover> setdisk /SpaceWar2/Schedule
```

복원할 하위 디렉토리를 지정한다.

```
amrecover> cd 200206-Prog
amrecover> add *
amrecover> cd ..
amrecover> cd 200206-Mail
amrecover> add *
```

물론, 특정 파일만 복원할 수도 있다. 해당 파일이 있는 디렉토리로 이동해서 파일을 추가해 주면 될 것이다. 복원 명령을 내린다.

```
amrecover> extract
```

필요한 테잎에 대한 정보를 알려줄 것이다. 맞으면 y라고 답한다. 그러면 테잎을 로드한 후 처음부터 읽어면서 해당 파일을 하나씩 복원할 것이다.

여기까지는 테잎서버의 테잎 장치를 이용하여 복원하는 방법이다. 그러나 이런 경우가 있을 수도 있다. 미디오 호환이 가능한 두 대의 테잎 장치를 각각 다른 리눅스 서버에 장착하여 한 대는 백업 전용으로, 다른 한 대는 복원 전용으로 사용하고자 하는 경우도 있을 것이다. 이제 백업 전용 서버에서 백업한 것을 복원 전용 서버에서 백업하는 경우를 살펴보자.

백업 전용 서버의 호스트네임은 backupsvr라 하고, 복원 전용 서버의 호스트 네임은 restoresvr이라 하자. 그리고 `/SpaceWar2/Schedule` 을 `/tmp`에 복원한다고 가정하자.

```
# cd /tmp
# amrecover SpaceWar -s backupsvr -d /dev/nst0
```

-s 옵션으로 backupsvr을 index server로 지정하였다. `/SpaceWar2/Schedule`의 백업에 대한 index는 backupsvr에 있기 때문이다. 그 뒤의 과정은 앞서의 복원 과정과 동일하다.

이것으로 복원에 대한 것도 대략 알아 보았다. 더 자세한 것은 Amanda man page를 참고하면서 그리고 실제로 사용하면서 하나씩 익히는 것이 최선의 방법인 듯 싶다.

8.6. 증분 백업

dump라는 백업 프로그램은 0에서 9까지의 백업 레벨을 지원한다. Amanda가 내부적으로 사용하는 백업 프로그램은 dump 또는 tar이다. Amanda도 0에서 9까지의 백업 레벨을 제공하며 백업 레벨은 사용자의 설정을 근거로 Amanda가 스스로 결정한다.

백업 레벨 0은 전체백업을 의미하며 백업 레벨 1은 전체 백업 이후 변경된 내용에 대한 백업을 뜻한다. 백업 레벨 2는 백업 레벨 1 이후 변경된 내용에 대한 백업이다. 나머지 백업 레벨도 비슷하게 생각하면 될 것이다.

필자는 한 달에 1회 전체 백업을 하고 주간 단위로 레벨 1의 백업을 실시한다. 9월 첫째 일요일에 전체 백업을 실시했다면 둘째 일요일에 백업할 내용은 전체 백업 이후 변경, 추가된 파일들일 것이다. 세번째 일요일에 백업할 내용도 레벨 1의 백업이므로 전체 백업 이후 변경, 추가된 파일들일 것이다. 따라서 세번째 일요일에 증분 백업을 하게되면 두번째 일요일에 증분 백업한 내용은 필요없게 되며 그 때 사용한 테잎은 재활용이 가능하게 된다.

강제로 레벨 1의 백업을 실시하기 위해서는 8.2절에서 정의한 dumptype 중에서 incr-only를 사용하면 된다.

```
localhost   /SpaceWar1/Program-Sources      incr-only
localhost   /SpaceWar1/Graphic-Sources      incr-only
gmfs2       /SpaceWar2/Prototype            incr-only
gmfs2       /SpaceWar2/Schedule             incr-only
```

이제 amcheck 명령으로 이상 유무를 확인한 후, amdump 명령으로 백업을 실행시키면 될 것이다.

8.7. 테잎에 바로 쓰기

Amanda는 백업을 진행하는 중에 로컬디스크(holdingdisk)의 공간이 부족하면 공간에 여유가 생길 때까지 일부 디렉토리에 대한 백업을 일시 중단하고 있다가, 공간에 여유가 생기면 그때 다시 백업을 진행한다. 이때 시간의 지연으로 인해 백업에 걸리는 시간이 늘어나게 된다. 이 경우는 일부 디렉토리를 테잎에 바로 쓰는 것이 유리할 수도 있다. writetotape이라는 dumptype을 정의하고 테잎에 바로 쓰기를 원하는 디렉토리의 dumptype을 writetotape으로 지정하면 될 것이다.

```
        define dumptype wtritetotape {
            global
            comment "write directly to tape"
            priority high
    dumpcycle 0
    holdingdisk no
        }
```

writetotape이 적용된 disklist 파일의 예는 아래와 같다.

```
localhost   /SpaceWar1/Program-Sources      writetotape
localhost   /SpaceWar1/Graphic-Sources      always-full
gmfs2       /SpaceWar2/Prototype            always-full
gmfs2       /SpaceWar2/Schedule             always-full
```

8.8. 필요없는 파일을 백업에서 제외하려면

백업할 필요가 없는 파일이 있을 수도 있다. 이 때는 dumptype에 exclude 또는 exclude list를 지정하여 해결할 수 있다. exclude list에 대해 알아보자. dumptype은 아래와 같은 모양이 될 것이다.

```
        define dumptype always-full {
            global
            comment "write directly to tape"
            priority high
            dumpcycle 0
    exclude list "/etc/amanda/SpaceWar/exclude-files"
        }
```

exclude-files의 내용은 다음과 같이 작성할 수 있다.

```
        AUTOEXEC.BAT
        NTDETECT.COM
        arcsetup.exe
        ntldr
        WINNT
        Documents and Settings
        System Volume Information
        arcldr.exe
        bootfont.bin
        RECYCLED
        Recycled
        *.SYS
*.sys
*.ini
```

9. 참고자료

AMANDA 홈페이지

http://www.amanda.org (http://www.amanda.org)

Using Amanda

http://www.backupcentral.com/amanda.html (http://www.backupcentral.com/amanda.html)

Amanda Users 메일링 리스트 모음

http://marc.theaimsgroup.com/?l=amanda-users
(http://marc.theaimsgroup.com/?l=amanda-users)

10. 수정된 mtx-changer 스크립트

아래의 스크립트는 IBM 3581 autoloader에 적용 할 수 있도록 수정된 것임을 다시 한 번 밝혀 둔다. 일부 수정할 부분이 더 있으나 작동에 별 이상이 없길래 그냥 내버려 두었다.

```
        #! /bin/sh
        ################################################################################
        # AMANDA Tape Changer script for use with the MTX tape changer program
        # Version 1.0 - Tue Feb 20 13:59:39 CST 2001
        #
        # Based on 'stc-changer' by Eric Berggren (eric@ee.pdx.edu)
        # Updated by Tim Skirvin (tskirvin@ks.uiuc.edu)
        #
        # Given that there's no license...let's make this the Perl Artis-
tic License.
        # Just make sure you give me and Eric credit if you modify this.
        ################################################################################

        ### USER CONFIGURATION
        # Name of the tape drive (takes place of "tapedev" op-
tion in amanda.conf)
        #  and default driver number in li-
brary (usu 0) that DRIVE_NAME points to

        DRIVE_NUM=0

        # Location of "STC" command and control device
        MTX_CMD="/usr/sbin/mtx";
        MTX_CONTROL="/dev/sg0";

        # Whether tape drive must eject tape before changer retrieves
        #  (ie, EXB-2x0). Usually okay if set while not necessary, bad if
        #  required but not set.
        DRIVE_MUST_EJECT=1

        # How long to check drive readiness (in seconds) after mount-
ing (or
        #  ejecting) a volume (on some libraries, the motion or eject com-
mand may
        #  complete before the drive has the vol-
ume fully mounted and online,
```

```
        #  or ready for retrieval, result-
ing in "Drive not ready"/"Media not
        #  ready" errors). Do an "mt status" command every 5 sec-
onds upto this
        #  time.
        DRIVE_READY_TIME_MAX=120

        #  tape "mt" command location...
        MT_CMD="/bin/mt-st"     # called via "MT_CMD -
f DRIVE_NAME rewind" &
                                 #   "MT_CMD -
f DRIVE_NAME offline" to eject
                                 # and "MT_CMD -
f DRIVE_NAME status" to get ready info

        ##############################################################################
        #
        NumDrives=-1
        NumSlots=-1
        LastSlot=-1
        LoadedTape=-1

        #
        # Usage information
        #
        usage()
        {
            echo
            echo "Usage: $Progname <command> [arg...]"
            echo "  -info          reports capability and loaded tape"
            echo "  -slot <slot>   loads specified tape into drive"
            echo "        current  reports current mounted tape"
            echo "        next     loads logi-
cally next tape (loops to top)"
            echo "        prev     loads logically previ-
ous tape (loops to bot)"
            echo "        first    loads first tape"
            echo "        last     loads last tape"
            echo "        0..99    loads tape from specified slot#"
            echo "  -eject         uloads current mounted tape"
            echo "  -
reset         resets changer (and drive); loads first tape"
            echo
            exit 5
        }

        #
        # Perform "stc" changer command (& handle the "fatal" errors)
```

```
        #  else, set 'CommandResStr' and 'CommandRawResStr' to the re-
sult string
        #  and 'CommandResCode' to the exit code
        #
        dotapecmd()
        {
            cmd=$1
            arg=$2

            CommandResStr=`$MTX_CMD $MTX_CONTROL $cmd $arg 2>&1`
            CommandRawResStr=$CommandResStr
            CommandResCode=$?

            CommandResStr=`echo $CommandResStr | head -
1 | sed 's/^[^:]*: //'`
            if [ $CommandResCode -gt 1 ]; then
                echo "0 $Progname: returned $CommandResStr"
                exit 2
            fi
        }

        #
        # Unload tape from drive (a drive command; "eject-
tape" is a changer command
        #  to actually retrieve the tape). Needed by some changers (con-
trolled by
        #  setting "DRIVE_MUST_EJECT")
        #
        ejectdrive()
        {
            # Tell drive to eject tape before changer re-
trieves; req'd by some
            #  drives (ie, EXB-2x0). Not needed by QDLT-
4x00. Do a "rewind"
            #  command first, then "offline" to eject (in-
stead of "rewoffl")
            #
            if [ "$DRIVE_MUST_EJECT" -ne 0 ]; then
                mtresstr=`$MT_CMD -f $DRIVE_NAME rewind 2>&1`
                mtrescode=$?


                if [ $mtrescode -ne 0 ]; then
                    if echo "$mtresstr" | egrep -sq 'no tape'; then
                        :;   # no tape mounted; assume okay...
                    else
                        # can't eject tape, bad; output: <tape#> reason
                        echo "0 $mtresstr"
                        exit 1
```

```
                fi
            else
                mtresstr=`$MT_CMD -f $DRIVE_NAME offline 2>&1`
                mtrescode=$?

                checkdrive 1
            fi
        fi
    }

    #
    # Check drive readiness after (un)mounting a vol-
ume (which may take a while
    #  after the volume change command completes)
    #
    checkdrive()
    {
        unmounting=$1

        if [ "$DRIVE_READY_TIME_MAX" -gt 0 ]; then

            # sleep time between checks
            pausetime=5

            # number of interations to check
            numchecks=`expr $DRIVE_READY_TIME_MAX / $pausetime`
            if [ "$numchecks" -eq 0 ]; then
                numchecks=1
            fi

            # check until success, or out of attempts...
            while [ "$numchecks" -gt 0 ]; do
                mtresstr=`$MT_CMD -f $DRIVE_NAME status 2>&1`
                mtrescode=$?

                if [ $mtrescode -eq 0 ]; then
                    # Success ?
                    return 0
                else
                    # pause, before trying again....
                    if [ "$numchecks" -gt 1 ]; then
                        sleep $pausetime

                        # if unmounting a vol-
ume, check for 'mt' command
                        #  failure; (sleep first for addi-
tional comfort)
                        if [ "$unmounting" -ne 0 ]; then
                            return 0
```

```
                        fi
                    fi
                fi
                numchecks=`expr $numchecks - 1`
            done

            # failed; output: -1 reason
            echo "-1 drive won't report ready"
            exit 1
        fi
    }

    #
    # Get changer parameters
    #
    getchangerparms()
    {
        dotapecmd status
        if [ $CommandResCode -eq 0 ] && \
            echo "$CommandResStr" | egrep -sq '^Storage Changer'; then

            NumDrives=`echo $dspec | wc -l`
     NumDrives=`echo "$CommandRawResStr" | \
     grep 'Data Transfer Element' | wc -l`
            if [ "$NumDrives" -le "$DRIVE_NUM" ]; then
                echo "$Program: Invalid drive # speci-
fied ($DRIVE_NUM > $NumDrives)"
                exit 3
            fi
     # grep 'Data Transfer Element $DRIVE_NUM' | \
     LoadedTape=`echo "$CommandRawResStr" | \
     grep 'Data Transfer Element' | \
     grep 'Storage Element [0-9]' | \
     awk '{ print $7 }' `
            if [ -z "$LoadedTape" -o "$LoadedTape" = "e" ]; then
                LoadedTape=-1
            fi
            NumSlots=`echo "$CommandRawResStr" | \
     grep 'Storage Element [0-9]\{1,\}:' | \
     grep -v 'Data Element' | \
     wc -l | sed -e 's/ //g' `
            LastSlot=`expr $NumSlots - 1`
        else
            echo \
              "$Progname: Can't get changer parameters; Re-
sult was $CommandResStr"
            exit 3
        fi
    }
```

```
#
# Display changer info
#
changerinfo()
{
    getchangerparms

    # output status string: currenttape numslots randomaccess?
    echo "$LoadedTape $NumSlots 1"
    exit 0
}

#
# Eject current mounted tape
#
ejecttape()
{
    getchangerparms
    ct=$LoadedTape

    # If no tape reported mounted, assume suc-
cess (could be bad if changer
    #  lost track of tape)
    #
    if [ $ct -lt 0 ]; then
        CommandResCode=0
    else
        ejectdrive
        dotapecmd unload
    fi

    if [ $CommandResCode -ne 0 ]; then
        # failed; output: <tape#> reason
        echo "$ct $CommandResStr"
        exit 1
    #else
        # success; output: <tape#> drive
        #echo "$ct $DRIVE_NAME"
        #exit 0
    fi
}

#
# Move specified tape into drive (operation level)
#
doloadtape()
{
    slot=$1
```

```
            if [ "$slot" -eq "$LoadedTape" ]; then
                return 0
            fi
            ejectdrive
            ejecttape
            dotapecmd load $slot
            sleep 25
            return $CommandResCode
        }

        #
        # Load next available tape into drive
        #
        loadnexttape()
        {
            curslot=$1
            direction=$2

            startslot=$curslot
            while true; do
                if doloadtape $curslot; then
                    return 0
                else
                    if echo $CommandResStr | egrep -
sq 'Slot.*reported empty'; then

                        if [ "$direction" -lt 0 ]; then
                            curslot=`expr $curslot - 1`
                            if [ "$curslot" -lt 0 ]; then
                                curslot=$LastSlot
                            fi
                        else
                            curslot=`expr $curslot + 1`
                            if [ "$curslot" -gt "$LastSlot" ]; then
                                curslot=0
                            fi
                        fi

                        # Check if we're back to where we started...
                        if [ "$curslot" = "$startslot" ]; then
                            if [ "$direction" -lt 0 ]; then
                                CommandResStr="No previous vol-
ume available"
                            else
                                CommandResStr="No subsequent vol-
ume available"
                            fi
                            return 1
                        fi
```

```
                else
                    return 1
                fi
            fi
        done
    }

    #
    # Report loadtape() status
    #
    reportstatus()
    {
        if [ $CommandResCode -eq 0 ]; then
            # success; output currenttape drivename
            echo "$LoadedTape $DRIVE_NAME"
            exit 0
        else
            # failed (empty slot?); output currenttape reason
            echo "$LoadedTape $CommandResStr"
            exit 1
        fi
    }


    #
    # Move specified tape into drive (command level)
    #
    loadtape()
    {
        slot=$1

        getchangerparms

        case "$slot" in
            current)
                if [ $LoadedTape -lt 0 ]; then
                    CommandResStr="Can't determine cur-
rent tape; drive empty ?"
                    CommandResCode=1
                fi
                ;;
            prev)
                if [ $LoadedTape -le 0 ]; then
                    loadnexttape $LastSlot -1
                else
                    loadnexttape `expr $LoadedTape - 1` -1
                fi
                ;;
            next)
```

```
                    if [ $LoadedTape -ge $LastSlot -o $LoadedTape -
lt 0 ]; then
                        loadnexttape 1 1
                    else
                        loadnexttape `expr $LoadedTape + 1` 1
                    fi
                    ;;
                first)
                    loadnexttape 1 1
                    ;;
                last)
                    loadnexttape $LastSlot -1
                    ;;
                [0-9]*)
                    doloadtape $slot
                    ;;
                *)
                    # error; no valid slot specified
                    echo "$Progname: No valid slot specified"
                    exit 1
                    ;;
            esac

            if [ $CommandResCode -eq 0 ]; then
                getchangerparms
                checkdrive
            fi
            reportstatus
        }

        #
        # Reset changer to known state
        #
        resetchanger()
        {
            ejectdrive
            dotapecmd reset
            if [ $CommandResCode -ne 0 ]; then
                # failed; output: failed? reason
                echo "-1 $CommandResStr"
                exit 2;
            else
                loadtape first
            fi
        }

        ##############################################################################
        #
        # MAIN
```

```
#
Progname=`basename $0`

if [ ! -x "$MTX_CMD" ]; then
    echo "-1 $Progname: cannot run STC command ($MTX_CMD)"
    exit 2
fi
if [ -n "$MTX_CONTROL" ]; then
    if echo "$MTX_CONTROL" | egrep -sq '^-f'; then
        :;
    else
        MTX_CONTROL="-f $MTX_CONTROL"
    fi
fi
if [ -n "$DRIVE_NUM" ]; then
    DRIVE_NUM=0
fi

if [ $# -ge 1 ]; then command=$1; else command="-usage"; fi

case "$command" in
    -info)
        changerinfo
        ;;
    -slot)
        loadtape $2
        ;;
    -eject)
        ejecttape
        ;;
    -reset)
        resetchanger
        ;;
    *)
        usage
        ;;
esac

exit 0
```