Automount 미니하우투

Don

don (at) sabotage.org

도운이

김명운

kmw (at) physics3.sogang.ac.kr

나상호

sabrina (at) debianusers.org

v0.3 1998-10-22

이 파일은 autofs automounter를 설명하고, 어떻게 설정하는지 그리고 몇몇 문제들을 피하는데 중점을 두었다.

고친 과정

고침 0.22001-11-24고친이 나상호

DocBook 변환

고침 0.11998-12-31고친이 김명운

최초 번역

1. 소개

1.1. Automount - 무엇 그리고 왜

Automounting은 데몬에 의해서 어떤 파일 시스템을 마운트 (그리고 언마운트)를 자동적으로 하는 과정이다. 만약 파일 시스템이 언마운트 되어있다면, 그리고 사용자가 그곳을 액세스하려고 시도할경우, 자동적으로 다시 마운트가 될것이다. 이것은 특히 큰 네트웍 환경과 여러 머신들 (특히 항상켜져있는 것들이 아닌) 서로간의 파일 시스템의 마운트에 특히 유용하다. 이것은 또한 제거할수있는 디바이스들이나, 또는 강제로 도스 파일 시스템의 아스키 변환과

같은 도스 파일 시스템의 아스키 변환을 끝내는 것 사이에 쉬운 교환과 같은 다른 용도에도 매우 유용하다.

1.2. automounting의 형태

리눅스에서 두가지 종류의 automounter들이있다;**AMD**와 **autofs**이다. AMD는 automount 데몬이고, 가정하건대 SunOS의 AMD와 같은 작용을한다. 이것은 유저 공간에서 구현되었고, 커널의 부분이 아님을 의미한다. 만약 당신이 AMD 데몬을 통하여 NFS 시스템으로부터 모든 automount 파일 시스템의 교통량의 통로인 지역호스트와 NFS 마운트를 하려고할때 커널이 automounting된것을 이해할 필요는없다.

Autofs는 커널에의해서 도움을 받는 더 새로운 시스템으로, 커널의 파일 시스템 코드들이 automount가 마운트 하는곳들이 다른편의 파일 시스템에 의존한다는 것을 알고 automount 프로그램이 이것을 받아들인다. 단지 이 미니하우투에서는 autofs만이 설명될것이다.

2. 설치

autofs는 커널 공간에서 구현되었기 때문에, 당신의 커널은 컴파일을통해 지원해야 한다. 2.0.xx에서 실험적인 옵션이지만, 매우 안정적인 것으로 보인다. 2.1.xx (그리고 2.2.xx 커널에서도) 그것은 실험적인 것이 아니다.

automount 프로그램과 그 설정 파일들 또한 필요하다; rpm을 사용하는것 (레드햇으로부터, 설치의 부분으로) 은 설치하는 가장 좋은 방법이다. automount 프로그램은 /etc/rc.d/init.d 디렉토리 아래에서 rc 스크립트에 의해서 실행 되어야한다. rpm으로 프로그램을 설치한 후, 당신은 레드햇의 제어판을 사용하여 당신의 rc?.d 디렉토리로 부터 연결에 의해서 또는 다른 배포판에서 이와 유사한 작용을하는 당신이 좋아하는 프로그램을 통하여 데몬이 실행되고 있는지를 반드시 확인할 필요가 있을것이다. rpm이 아닌것은 자신의 시스템에 맞게 적용시켜야 할것이다. 그리고 rc 스크립트가 하는일에 관해서는 너무 어렵기 때문에 알필요가 없다; 만약 당신이 이 하우투를 읽고 있다면 당신은 아마 알기를 원하지 않을것이다.

3. 설정

RPM을 설치하고 난후 요점들은 쉽게 충분히 얻을수 있을것이다. 그러나, 만약 당신이 전에 이것에대한 경험이 없다면, 이 부분은 당신이 잘 알지 못하는 부분이다.

/etc에는 두개의 파일이 있고, 하나는 `auto.master`라 불리고 또다른 하나는 `auto.misc`라고 불린다. 나의 auto.master 파일은 이와같다:

```
/auto   /etc/auto.misc  --timeout 60
```

첫 번째 내용은 마운트할곳을 나타내는것이 아니다. 마운트 할곳들의 집합 (두 번째 내용에서 나타나는)이 될것이다. 세 번째 옵션은 마운트된 파일 시스템이 사용후 60초후에 언마운트를 시도하는것을 나타낸다. 물론, 만약 사용되고 있다면, 언마운트 할수없다.

Auto.misc는 "지도 파일"이다. 여러 지도 파일들은 auto.master안에 정의 되어질수있다. 나의 auto.misc 파일은 이와같다:

```
kernel          -ro,soft,intr           ftp.kernel.org:/pub/linux
cd              -fstype=iso9660,ro      :/dev/cdrom
zip             -fstype=auto            :/dev/hdd4
floppy          -fstype=vfat            :/dev/fd0
```

첫 번째 열 ("key")은 마운트 위치를 나타낸다. 이 경우에 /auto/floppy나 또는 다른 어떤것이 될것이다. 중간 부분은 옵션들이다; 이것에 관한 자세한 내용들은 맨 페이지를 읽어보아라. 그리고 마지막 열은 파일 시스템의 출처를 나타내는 특정한 위치들을 지정한다. "커널"의 내용은 NFS 마운트 된것이라 가정되었다. 다른 모든 라인들은 지역 디바이스임을 의미한다.

4. 언마운트하는데 오랜 기다림

당신은 몇번 60초 타임아웃을 주시하고 생각할것이고, 그것은 플로피를 여는데 오랜 시간을 기다려야 한다**.** 아마 나는 단지 디스크를 **sync**하고 마운트된것을 꺼낸후 아무도 그것을 알지 못할것이다**.** 나는 정상적인 변형을 제안한다. 무엇보다도, 당신은 타임아웃을 바꿀수있다. 그러나 그것은 그리 효율적이지 못하다; 단지 15초후나 또는 아무때나 시스템에게 언마운트 할것을 말한다. automount 프로그램에게 언마운트 하도록 하는것이 실제적인 방법이다. 만약 당신이 (kill 프로그램과 함께)`SIGUSR1` 신호를 automount 프로세스에게 보낸다면 가능한한 모든것을 언마운트 할것이다. 그러나 전에 사람들이 그들의 윈도우 매니저에서 언마운트 버튼을 누른다면, 조금의 문제가있다.

automount 프로세스는 루트에 의해서 실행되고 단지 루트로부터만 신호를 받아들인다. 이 이유중 절반은 당신이 아마 automounting을 할수있어서 당신은 루트 권한 없이 마운트나 언마운트를 할수있다. 지저분한 행동인 suid-root C 프로그램을 만드는것은 쉬운일이다. 그러나, sudo를 이용해서 사용자들에게 고유한 kill 신호를 보내는것이 가능하다. 단지 문제는 sudo가 당신에게 현재 프로세스의 PID를 알수있는 "프로세스 부명령어들"을 사용하지 못하게 할것이다. 당신은 killall이라 불리는 프로그램을 가지고 있고 당신에게 이것을 할수있게 할것이다. (제안들에 관해 감사한다.)

```
ALL     ALL=NOPASSWD:/usr/bin/killall -USR1 automount
```

다른한편으로, 당신은 당신의 사용자들이 모든 프로세스에 -SIGUSR1 신호를 보내는것을 허락해야 한다. 그것은 프로그램에 여러 가지 영향을 미친다; 어떤 윈도우 매니저들은 재사용될수 있지만, 그러나 xemacs는 죽인다. 그래서 여기서는 더 이상의 killall내에서 버퍼가 더 실행되는 경우는 없기를 희망한다.

5. 질문들

5.1. 나는 **/auto/floppy**나 내가 찾고자 하는 마운트 포인트를 찾을수없다**.**

만약 automount가 거의 셋업되었고, 당신이 그것을 사용하려고 한다면, 마운트 한곳을 찾을수있고 심지어 당신이 사용하지 않을때는 볼수없다. 만약 당신이 그래픽 툴들을 이용해 디렉토리를 보려고 한다면, 당신은 매뉴얼에 설정 한데로 이름을 직접 써 넣어야 한다. 불행하게도 이용 가능한 보이지 않는 마운트한곳은 대부분 autofs의 장애로 선택 할수가없다. 만약 당신에게 이것이 정말 버그라면, 설정 파일을 수정하라. (힌트, "설정"을 위한 .c파일의 끝부분들중의 한부분)

5.2. 무엇이 마운트 되어 있는지 내가 어떻게 확인하나**?**

df명령이다. 마운트가 되어 있지 않았다면 옵션 없이 하면되고, 만약 마운트가 되어있는 상태라면 옵션을 추가하여 보아라.

5.3. 나는 윈**95** 디스크 **("vfat")**를 놓았고 그것은 단지 일정한 **FAT** 디스크로만 자동 감지한다**.**

이것은 automount의 문제가아니니다. 이글을 썼을때, "auto" 파일 시스템 형태는 전에 도스 파일 시스템으로 성공적으로 마운트되어있다면, vfat로 마운트하려고 시도하지 않을것이다. VFAT는 윈95와 윈NT에서 긴파일 이름 지원을 위해 기존의 FAT/MSDOS 파일 시스템에 밀어넣은것이다.

마운트의 저자들중 한사람에 따르면 마운트는 단지 파일 시스템의 형태를 특정화하기위해 시스템 콜 주위를 감싸기 때문에 파일 시스템의 형태를 정하는 것은 여전히 사용자들의 책임이다. 현재 고려 중인 "자발적"인 것 보다는 마운트는 파일 시스템의 여러 형태를 가지고 순차적으로 시도해 나간다. 나는 만약 누군가가 마운트 프로그램을 만든 사람을 찾으려고 한다면, 당신의 의견은 환영받을 것이다.

5.4. 내 파일 시스템인 **/grumblesmurf**가 마운트 되어 있고 **kill -SIGUSR1**은 이 디렉토리를 언마운트 하지 않는다**.**

무언가에 의해서 사용되고 있다. 루트역시 그것은 언마운트 할수 없을것이다. 만약 당신이 마운트를 한 사람이라면 (다른 사람은 전혀 사용하지 않은) 셸에서 현재 디렉토리를 확인하라. 만약 그곳에 아무것도 없다면, (특별히 디렉토리 브라우저 같은것들)소위 문앞의 안보이는 발 같은 것을 찾아보아라. 만약 당신이 찾기를 포기한다면, fuser 프로그램을 사용하여 보아라.

5.5. autofs에 대하여 누구에게 감사하여야 하는가?

내가 아니다. 나는 그것에 관해 한것이 아무것도 없다. 나는 단지 "autofs"에 관한 훌륭한 것들을 모두에게 알리고 쉽게 사용할수 있다는것을 알린것 뿐이다. AMD(힌트, 그들은 자유로운 툴의 전 버전을 높은 가격에 팔았다.) 의 원 비겹자들과 비교하여, autofs는 매우 잘 문서화되어 있고 구현이 잘 되어있어 진심으로 감사한다. 모든것은 Transmeta에게 저작권이 있고 그래서 나는 어떠한 고마운 분들을 나열할수 없음을 죄송스럽게 생각한다.

5.6. 어디에서 automounting에 관한 것들을 더 배울수있는가?

이 곳은 autofs 메뉴얼은 http://www.linuxhq.com/lg/issue24/nielsen.html.에 있다. 또한 am-utils에 관해서는 http://www.cs.columbia.edu/~ezk/am-utils사이트를 참조하라.

이 주소에 대하여 감사한다.