

본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
본이며 책 구입희망자분들께서 공개요청하셨던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1, 2 편

16 장. awk 편

16 장. awk 목 차 (책 실제페이지 : 119 페이지)

실무명령어# 16-1 절 : 특정 패턴의 문자들을 원하는 포맷으로 처리하는 강력한 유틸리티 awk 개론
실무명령어# 16-2 절 : /etc/passwd 파일에서 계정명(ID)만 출력하기
실무명령어# 16-3 절 : /etc/passwd 파일에서 각각의 ID 와 홈디렉토리만을 출력하기
실무명령어# 16-4 절 : /etc/passwd 파일을 이용한 /etc/group 파일내용 만들기
실무명령어# 16-5 절 : /etc/passwd 파일을 이용한 /etc/shadow 파일내용 만들기
실무명령어# 16-6 절 : /home 디렉토리의 각 계정별로 압축하는 명령어 만들기
실무명령어# 16-7 절 : 각각 다른 여러디렉토리 내에 동일한 파일을 vi로 수정하는 명령어 만들기
실무명령어# 16-8 절 : 아파치 웹로그파일(access_log)에서 유일방문자수 카운트하기(유일 IP 주소 카운트)

- 본 자료의 저작권은 “슈퍼유저코리아” (www.superuser.co.kr)에 있습니다 .
- 본 자료는 “슈퍼유저코리아”에서 출판한 『리눅스 실무명령어 바이블(1편,2편)』의 실제 내용중 일부분으로서 책의 본문내용을 확인하고자하시는 분들의 요청에 의해 요청횟수가 많았던 부분만을 일부 공개한 것입니다.
- 본 책자의 자세한 모든 정보는 <http://www.superuser.co.kr/linuxcommandbible/> 에서 확인하시기 바랍니다.
- 본 자료의 수정 및 편집은 허용되지않습니다.
- 본 책은 권수 : 2 권 (1 편, 2 편), 페이지수 : 총 1,859 페이지, 장(chapter): 382 장, 절: 1,900 개의 절로 구성되어있습니다.
- 저작권 및 문의전화(사무실) : TEL : 051-583-8128, FAX : 051-583-8130

실무명령어# 16-1 : 특정 패턴의 문자들을 원하는 포맷으로 처리하는 강력한 유틸리티 awk 개론

awk 는 1977 년 Bell 연구소의 Alfred v. Aho' Peter j. Wenberger 와 Brian W. Kernighan 에 의해 특정문자의 패턴을 원하는 포맷으로 처리하기 위해 개발된 셸에서 바로 사용가능한 일종의 유틸리티이며 언어입니다. 즉, 텍스트파일로 표준입력을 받아들여 목적에 맞게 출력할 수 있는 문자처리 언어입니다. 특정 문자들을 원하는 포맷에 맞게 출력하기위해 주로 셸프로그램이나 셸에서 바로 사용됩니다. 또한 awk 는 gawk 로 링크되어 사용하므로 둘은 동일한 명령어입니다.

awk 를 사용하기 위해서 꼭 알아두셔야하는 것은 awk 는 기본적으로 탭 또는 공백으로 구분되는 각각의 단어들을 하나의 변수로 처리합니다. 즉, 탭과 공백을 기본적인 구분문자로 사용한다는 의미입니다. 하지만 탭과 공백을 무시하고 특정문자(: (콜론) ; (세미콜론) , (콤마) 등)를 구분문자로 처리하고자 한다면 -F 옵션을 사용하면 됩니다. 즉, -F 옵션 다음에 원하는 특정문자를 지정하면 그 문자를 구분문자로 하여 각각의 단어들을 변수로 처리할 수 있습니다.

예를 들어 "-F:" 으로 지정하였다면 ":" (콜론) 문자를 구분문자로 하여 모든 단어들이 처리되며, 또한 "-F;" 으로 지정하였다면 ";" (세미콜론) 을 구분문자로 하여 모든 단어들이 처리됩니다.



본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
본이며 책 구입희망자분들께서 공개요청하였던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 I, II 편

그리고 awk 를 사용할 때에는 거의 대부분 다른 명령어와 함께 사용됩니다. 즉, awk 사용의
대부분은 다른 명령어의 결과를 입력으로 받아서 처리하는 용도로 사용된다는 의미입니다.
다음 사용예에서 이 의미를 확인해 보시기 바랍니다.

명령어위치 : /bin/awk

사용형식

```
awk [옵션] '스크립트' [변수=값][파일...]
```

```
awk [옵션] [-F] '{스크립트}'
```

* 참고 : awk 의 사용형식은 실제 사용예를 통해서 익히시는 것이 좋습니다.

이번 장에서는 awk 의 문법적인 설명보다는 서버관리 측면의 실무에서 awk 를 바로 활용할
수 있는 유용한 방법들을 알려드립니다.

실무명령어# 16-2 : /etc/passwd 파일에서 계정명(ID)만 출력하기

웹호스팅서버 또는 많은 사용자들이 사용하는 서버의 /etc/passwd 파일에는 수많은 사용자
의 계정정보가 저장되어 있습니다. 이때 각 사용자의 ID 만을 출력하고자 한다면 어떻게
하면 될까요? awk 를 사용하면 간단히 처리할 수 있습니다. 즉, /etc/passwd 파일의 내용을
살펴보면 ":"(콜론)으로 계정 사용자정보의 항목들이 구분되어있다는 것을 알고 계실 것입
니다.

즉, 간단히 /etc/passwd 파일을 살펴보면 아래와 같이 관리자가 원하는 중요한 정보들이 많
이 있습니다. 즉, /etc/passwd 파일에는 다음과 같은 정보들이 순서대로 기록되어 있습니
다.

| | | | | | | |
|-----|----------|-------|-------|-------|---------|---------|
| 계정명 | : 패스워드부분 | : UID | : GID | : 코멘트 | : 홈디렉토리 | : 기본사용셸 |
| \$1 | \$2 | \$3 | \$4 | \$5 | \$6 | \$7 |

이들 정보는 awk 명령어의 필드분리명령에 의해 왼쪽에서 부터 차례대로 \$1...\$7 에 대응됩
니다. 이런 사항을 이용하여 /etc/passwd 파일의 정보에서 원하는 부분만을 출력해주는 방
법에 대해서 설명합니다.

즉, 아래의 예는 awk 를 이용하여 /etc/passwd 파일에서 각 계정사용자들의 ID 만을 출력하
는 것입니다.

```
[root@file root]# cat /etc/passwd | awk -F: '{print $1}'
```

좀 더 상세히 설명하면 위와 같이 "cat /etc/passwd"명령어로 /etc/passwd 파일의 각각의
행들이 "|" (파이프명령어)에 의해 차례대로 awk 의 입력으로 들어오게 됩니다. 이때 "-F:"
에 의해 입력으로 들어온 /etc/passwd 파일의 각각의 행들이 ":"(콜론)으로 구분됩니다.



본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
본이며 책 구입희망자분들께서 공개요청하셨던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1,11 편

예를들어 /etc/passwd 의 내용중 아래와 같은 행이 있다고 가정 하겠습니다.

```
sspark:x:501:501::/home/sspark:/bin/bash
```

이 행은 "-F:"에 의해 각각의 항목들이 ":"(콜론)으로 구분되어 다음과 같은 awk 의 입력변
수로 작용합니다.

\$1 에는 sspark 이 할당되고,
\$2 에는 x 가 할당되며,
\$3 에는 501 이 할당되고,
\$4 에는 501 이 할당되며,
\$5 에는 공백이 할당되며,
\$6 에는 /home/sspark 이 할당되고
\$7 에는 /bin/bash 가 각각 할당됩니다.

이것이 바로 "-F:"의 의미입니다. 이제 awk 에서 사용된 "-F:"의 의미를 확실히 아셨을 것
이라 생각합니다.

그렇다면 '{print \$1}'는 어떤 의미일까요?

위의 의미 그대로 \$1 만을 출력(print)하라는 의미입니다.

설명이 다소 길어지기는 하였지만 아래 명령의 결과로는 /etc/passwd 파일의 각각의 행들이
":"(콜론)문자로 구분된 후 첫번째 단어인 각행의 ID 들만이 결과로 출력이 됩니다.

```
[root@file root]# cat /etc/passwd | awk -F: '{print $1}'
root
bin
daemon
gopher
ftp
nobody
rpm
vcsa
nscd
sshd
rpc
rpcuser
nfsnobody
mailnull
smmisp
pcap
apache
squid
webalizer
xfs
named
```

본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
분이며 책 구입희망자분들께서 공개요청하였던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1, 11 편

```
ntp
gdm
postgres
bible
sspark
working
locli
mysql
test01
test02
test03
jinious
public
samba
[root@file root]#
```

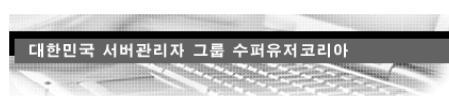
첫번째 사용예의 설명을 정확하게 이해하시기 바랍니다. 이 사용예만 정확하게 이해한다면
다음 예들은 이 예를 조금 응용한 것이므로 쉽게 이해하실 수 있을 것입니다.

실무명령어# 16-3 : /etc/passwd 파일에서 각각의 ID와 홈디렉토리만을 출력하기

다음은 앞의 예를 조금 응용하여 각각의 ID와 함께 홈디렉토리까지 같이 출력하는 예를 보
도록 하겠습니다. 실제 서버에 설정되어 있는 각각의 계정과 홈디렉토리를 확인해야할 경
우에 실테이터를 만들기 위하여 사용하는 방법입니다.

이는 각각의 ID와 홈디렉토리를 조사하고자 할 때에 주로 사용되는 방법입니다.

```
[root@file root]# cat /etc/passwd | awk -F: '{print $1"          "$6}'
root          /root
bin           /bin
daemon       /sbin
gopher       /var/gopher
ftp          /var/ftp
nobody       /
rpm          /var/lib/rpm
vcsa         /dev
nscd         /
sshd         /var/empty/sshd
rpc          /
rpcuser      /var/lib/nfs
nfsnobody    /var/lib/nfs
mailnull     /var/spool/mqueue
smmsp       /var/spool/mqueue
```



본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
분이며 책 구입희망자분들께서 공개요청하셨던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1,11 편

```
pcap          /var/arpwatch
apache        /var/www
squid         /var/spool/squid
webalizer     /var/www/html/usage
xfs           /etc/X11/fs
named        /var/named
ntp           /etc/ntp
gdm           /var/gdm
postgres     /var/lib/pgsql
bible        /home/bible
sspark       /home/sspark
working      /home/working
locli        /home/locli
mysql        /home/mysql
test01       /home/test01
test02       /home/test02
test03       /home/test03
jinious      /home/jinious
public       /home/public
samba        /home/samba
[root@file root]#
```

앞의 예와 다른 부분은 '{print \$1" "\$6}' 뿐이므로 아래 전체명령어를 설명하면
다음과 같습니다.

/etc/passwd 파일의 각행들을 ":"(콜론)으로 구분하여 첫번째인 ID 와 여섯번째인 홈디렉토
리를 결과로 출력하라는 의미입니다.

실무명령어# 16-4 : /etc/passwd 파일을 이용한 /etc/group 파일내용 만들기

이번에는 /etc/passwd 파일을 이용하여 /etc/group 파일의 내용을 만들어 보도록 하겠습니다.
/etc/group 파일의 형식은 "ID:x:GID:소속계정리스트"와 같은 형태로 되어 있습니다.

여기에서 소속계정리스트는 각계정의 GID 를 모두 조사하여 각 그룹리스트에 입력을 시켜주
어야 하지만 awk 명령어 한번으로 이를 만들기가 까다로우므로 소속그룹리스트가 없다는 가
정하에 위와같은 형식의 /etc/group 파일을 만들어 보도록 하겠습니다.

```
[root@file root]# cat /etc/passwd | awk -F: '{print $1"xW:"$4":"}'
root:x:0:
bin:x:1:
daemon:x:2:
gopher:x:30:
ftp:x:50:
nobody:x:99:
rpm:x:37:
```

본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
분이며 책 구입희망자분들께서 공개요청하였던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1,11 편

```
vcsa:x:69:  
nscd:x:28:  
sshd:x:74:  
rpc:x:32:  
rpcuser:x:29:  
nfsnobody:x:65534:  
mailnull:x:47:  
smmisp:x:51:  
pcap:x:77:  
apache:x:48:  
squid:x:23:  
webalizer:x:67:  
xfs:x:43:  
named:x:25:  
ntp:x:38:  
gdm:x:42:  
postgres:x:26:  
bible:x:500:  
sspark:x:501:  
working:x:502:  
locli:x:503:  
mysql:x:504:  
test01:x:505:  
test02:x:506:  
test03:x:507:  
jinious:x:509:  
public:x:510:  
samba:x:511:  
[root@file root]#
```

지금까지 사용했던 방법과 동일하게 "cat /etc/passwd" 파일을 출력한 다음 ":"문자로 각행
들의 항목들을 구분(-F:)한 다음 '{print \$1"W:xW:"\$4:"}'이 출력되게 됩니다. 여기서
"W"문자의 의미는 그뒤에 나오는 문자를 그대로 출력하기 위한 특수문자입니다.

따라서 \$1 은 ID 를 출력하고 \$4 는 GID 를 출력하므로 '{print \$1"W:xW:"\$4:"}'의 결과는
"ID:x:GID:"와 같은 형태로 출력이 됩니다.

따라서 위의 명령의 결과는 /etc/group 파일의 내용과 동일한 결과를 출력하게 됩니다.

실무명령어# 16-5 : /etc/passwd 파일을 이용한 /etc/shadow 파일내용 만들기

이번에는 awk 를 활용하여 실제정사용자들의 /etc/shadow 파일을 출력하는 예를 보도록 하겠
습니다.



본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
분이며 책 구입희망자분들께서 공개요청하셨던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 I, II 편

먼저 grep 명령어를 아셔야하는데 grep 명령어의 자세한 설명은 이 책의 grep 명령어편을 참
고하시고, 아래명령어의 의미를 자세히 설명해 보도록 하겠습니다.

명령어 : `grep home /etc/passwd | awk -F: '{print "grep "$1" /etc/shadow"}'`

"grep home /etc/passwd"의 의미는 /etc/passwd 파일내에서 home 이라는 문자를 가진 행들은
모두 출력하라는 것입니다. 즉, /etc/passwd 파일에서 실제정을 사용하는 계정들이라면 그
홈디렉토리가 home 이라는 것에서 착안한 것입니다. 만약 홈디렉토리를 host 로 사용하는
계정사용자라면 "grep host /etc/passwd"라고 하시면 될 것이고, user1 으로 사용한다면
"grep user1 /etc/passwd"으로 하면 될 것입니다.

따라서 "grep home /etc/passwd"의 결과로 인하여 /etc/passwd 파일의 내용중 home 을 홈디
렉토리로 시작하는 계정들은 모두 출력이 될 것입니다.

그 다음 출력되는 각 행들을 awk 의 입력으로 들어가기 위하여 "|" (파이프명령어)를 사용하
였습니다.

즉, /etc/passwd 파일의 각 행들에서 /home 을 홈디렉토리의 시작으로 가지는 모든 계정들의
행의 출력이 `awk -F: '{print "grep "$1" /etc/shadow"}'`의 입력으로 들어가게 됩니다.

그 다음 `awk -F: '{print "grep "$1" /etc/shadow"}'`이 실행이 되면 다음과 같은 결과를 얻
게 됩니다.

```
[root@file root]# grep home /etc/passwd | awk -F: '{print "grep "$1" /etc/shadow"}'  
grep bible /etc/shadow  
grep sspark /etc/shadow  
grep working /etc/shadow  
grep locli /etc/shadow  
grep mysql /etc/shadow  
grep test01 /etc/shadow  
grep test02 /etc/shadow  
grep test03 /etc/shadow  
grep jinious /etc/shadow  
grep public /etc/shadow  
grep samba /etc/shadow  
[root@file root]#
```

그 다음 위의 결과를 가지고 다음과 같이 간단한 SHELL 스크립트를 작성합니다. 아래 보시
는 바와 같이 vi 를 열어서 "#!/bin/bash"를 첫행에 추가한 다음 test.sh 이라는 파일을 만
들었습니다.

```
[root@file home]# vi test.sh
```

```
#!/bin/bash  
grep bible /etc/shadow  
grep sspark /etc/shadow  
grep working /etc/shadow
```



본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
분이며 책 구입희망자분들께서 공개요청하셨던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1,11 편

```
grep locli /etc/shadow
grep mysql /etc/shadow
grep test01 /etc/shadow
grep test02 /etc/shadow
grep test03 /etc/shadow
grep jinious /etc/shadow
grep public /etc/shadow
grep samba /etc/shadow
~
~
~
~
"test.sh" [New] 12L, 274C written
[root@file home]#
```

그런 다음 아래와 같이 실행가능한 파일을 만들기 위하여 test.sh 파일에 700 이라는 퍼미션
을 주었습니다. 그리고 test.sh 를 실행한 결과를 아래에서 보실 수 있을 것입니다.

```
[root@file home]# chmod 700 test.sh
[root@file home]#
[root@file home]# ./test.sh
bible:$1$Fy3B99j.$7Mx7eyV5MKvBJHUoJeWFG/:12352:0:99999:7:::
sspark:$1$1tmjn2RW$wMte72wqdvwenDQLwk8ax0:12358:0:99999:7:::
working:$1$6tbDsd.O$viOzGkf9mm9sQdhquiCHS/:12363:0:99999:7:::
locli:$1$RZAcGxSH$wj0fKbsvSW1dujaP8HXcE.:12363:0:99999:7:::
mysql:!!!:12363:0:99999:7:::
test01:$1$4m0iFIe0$eNQiLLb6YZXTkYsGcEPMI.:12364:0:99999:7:::
test02:$1$sbmK0i0e$ri70QcI0poemRnwKFVHLj.:12364:0:99999:7:::
test03:$1$RgUe.iqW$WGPY1N1QXD/Kx8mThpJ57/:12364:0:99999:7:::
jinious:$1$ZgCGoUrb$VqxCMYr1W1Cfb3jUGMKa21:12369:0:99999:7:::
public:$1$Nu9QKsAU$pv6PuSOwtI26BWRKxE/ix1:12364:0:99999:7:::
samba:$1$Cq5x8KrL$StXBXggG1z/OwXsefgBUOG.:12364:0:99999:7:::
[root@file home]#
```

이 결과의 의미는 다음과 같습니다.

/etc/passwd 파일의 각 계정들 중 /home 을 홈디렉토리의 시작으로하는 모든 계정들의
/etc/shadow 파일을 생성한 것입니다.

혹, 센서있는 독자분이 계시다면 이런 질문을 하실 것입니다. "이런것을 만들어서 어디에
쓸까요?"라고 질문을 하실지 모릅니다.

만약 여러분들께서 현재 사용하고 계시는 서버의 계정중 /home 을 홈디렉토리의 시작으로
하는 모든 계정을 다른 서버로 이전하고자 한다면 그 계정들의 /etc/passwd 파일의 내용과
/etc/shadow 파일의 내용, 그리고 /etc/group 파일의 내용을 이전되는 새서버에 원본과 동일
하게 복사를 해주셔야 할 것입니다. 이때 매우 유용하게 사용할 수 있습니다. 앞서 설
명드린 /etc/group 파일만드는 방법도 이와 같은 이유에서 예를 든 것입니다.



본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
분이며 책 구입희망자분들께서 공개요청하였던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1,11 편

필자와 같이 실제 두대이상의 서버를 관리하면서 다른 서버로 일부 계정을 이전해야하는 경
험을 하신 독자분이 계신다면 필자가 설명한 이 내용이 매우 유용하게 쓰일 것임을 확신합
니다.

실무명령어# 16-6 : /home 디렉토리의 각 계정별로 압축하는 명령어 만들기

다음은 특정디렉토리(예에서는 /home)내에 존재하는 각각의 계정별로 tar 로 압축하는 명
령어를 만드는 방법입니다.

즉, grep 명령어와 awk 를 활용하면 이와 같은 작업을 굉장히 쉽게 작업을 할 수 있습니다.

즉, 이번 예에서는 다음과 같은 작업을 하기위한 명령어(셸스크립트)를 생성하는 것이 목
적이됩니다.

목적 : /backup 디렉토리내에 /home 에 존재하는 각각의 계정들별로 "ID.tar.gz"라는 파일
로 tar 로 압축을 하고자 함.

이와 같은 작업을 하기위하여 필자는 아래와 같은 명령어를 사용합니다.

```
ls -l /home | grep "^d" | awk '{print "tar cvfpz " $9"W.tarW.gz" " W/homeW/"$9}'
```

이 명령어는 모두 "|" (파이프명령어)로 구분되는 3개의 명령어가 한번에 실행이 됩니다.

먼저 "ls -l /home"는 /home 에 있는 모든 파일리스트들이 출력이 될 것입니다.

그리고 "grep "^d" "는 /home 내의 디렉토리 목록만 출력이 됩니다. 이것은 각각의 계정들
은 모두 /home 내의 디렉토리를 가지고 있기 때문입니다.

그리고 이런 결과는 그 뒤에 나오는 awk 의 입력으로 들어가게 됩니다. 그뒤의 awk 명령어
는 넘어온 각각의 행들에서 'tar cvfpz ID.tar.gz /home/ID'라는 결과를 각각 만들어 출력
합니다. 만약 필자가 사용한 awk 명령어가 이해되지 않는다면 그대로 복사해서 사용하셔도
무방합니다.

단, 필자가 사용한 awk 명령어들은 외우는 것이 아니라 목적인 조건에 맞도록 적절히 생성
하여 사용하는 것이므로 외우는 것은 올바른 공부가 아닙니다. 필자와 같이 awk 명령어를
마음대로 사용하는 것이 어렵다면 그대로 복사해서 사용하셔도 되겠지만 그 의미는 정확하
게 이해하고있어야 합니다.

이와 같은 awk 명령어의 결과로 다음과 같은 결과를 얻었습니다.

```
[root@file backup]# ls -l /home | grep "^d" | awk '{print "tar cvfpz " $9"W.tarW.gz"
" W/homeW/"$9}'
tar cvfpz bible.tar.gz /home/bible
tar cvfpz jinious.tar.gz /home/jinious
tar cvfpz locli.tar.gz /home/locli
tar cvfpz lost+found.tar.gz /home/lost+found
```


본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
분이며 책 구입희망자분들께서 공개요청하였던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1,11 편

그리고 생성한 backup.sh 쉘파일에 아래와 같이 실행퍼미션(700)을 설정한 다음 /backup 위
치에서 실행(/backup.sh)을 하였습니다. 아래 실행결과를 보시는 바와같이 실행결과
의 길이가 너무 길어지므로 중간에서 생략하였습니다.

```
[root@file backup]# chmod 700 backup.sh
[root@file backup]#
[root@file backup]# ./backup.sh
tar: Removing leading `/' from member names
home/bible/
home/bible/.bash_logout
home/bible/.bash_profile
home/bible/.bashrc
home/bible/.emacs
home/bible/.gtkrc
home/bible/.bash_history
home/bible/.viminfo
home/bible/www/
home/bible/www/index.html
tar: Removing leading `/' from member names
home/jinious/
home/jinious/.bash_logout
home/jinious/.bash_profile
home/jinious/.bashrc
home/jinious/.emacs
home/jinious/.gtkrc
home/jinious/www/
tar: Removing leading `/' from member names
home/locli/
home/locli/.bash_logout
home/locli/.bash_profile
home/locli/.bashrc
home/locli/.emacs
home/locli/.gtkrc
home/locli/www/
home/locli/www/1.php
home/locli/www/phpm/
home/locli/www/phpm/css/
home/locli/www/phpm/css/phpmyadmin.css.php
home/locli/www/phpm/images/
home/locli/www/phpm/images/arrow_ltr.gif
home/locli/www/phpm/images/arrow_rtl.gif
home/locli/www/phpm/images/asc_order.png
home/locli/www/phpm/images/browse.png
home/locli/www/phpm/images/button_browse.png
home/locli/www/phpm/images/button_drop.png
....
중략
```

본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
분이며 책 구입희망자분들께서 공개요청하셨던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1,11 편

참고로 이런 실행결과 메시지들을 콘솔(모니터 화면)로 출력하지 않으려면 다음과 같이 하
시면 됩니다.

```
[root@file backup]# ./backup.sh >& /dev/null
```

이 의미는 실행과정에서 출력되는 메시지들을 모두 "/dev/null"("널터미널", 일종의 쓰레기
통)로 버리라는 의미이므로 실행과정 중 출력되는 메시지들은 나타나지 않고 정상적인 실행
만하게 됩니다.

이렇게하여 압축된 결과를 확인해 보시면 다음과 같습니다.

```
[root@file backup]# ls -l
total 2888440
-rwx----- 1 root root 415 Nov 13 23:46 backup.sh
-rw-r--r-- 1 root root 1591 Nov 13 23:47 bible.tar.gz
-rw-r--r-- 1 root root 1006 Nov 13 23:47 jinious.tar.gz
-rw-r--r-- 1 root root 1807845 Nov 13 23:47 locli.tar.gz
-rw-r--r-- 1 root root 970 Nov 13 23:47 mysql.tar.gz
-rw-r--r-- 1 root root 967 Nov 13 23:47 public.tar.gz
-rw-r--r-- 1 root root 2749180960 Nov 14 00:20 samba.tar.gz
-rw-r--r-- 1 root root 1008 Nov 14 00:20 sspark.tar.gz
-rw-r--r-- 1 root root 1040 Nov 14 00:20 test01.tar.gz
-rw-r--r-- 1 root root 1045 Nov 14 00:20 test02.tar.gz
-rw-r--r-- 1 root root 1042 Nov 14 00:20 test03.tar.gz
-rw-r--r-- 1 root root 203830279 Nov 14 00:23 working.tar.gz
[root@file backup]#
```

위와 같이 /home 내에 존재하는 각각의 계정별로 모두 압축되어 저장된 것을 확인할 수 있
을 것입니다.

물론, /home 을 통째로 압축하는 방법이 보다 간단하지만 반드시 각각의 계정별로 압축저장
을 해야하는 경우도 종종 발생합니다. 이런 경우에 요긴하게 사용하시기 바랍니다.

실무명령어# 16-7 : 각각 다른 여러디렉토리 내에 동일한 파일을 vi 로 수정하는 명령어 만 들기

좀 황당한 시도일런지는 모르겠습니다. 일반사용자에게는 그다지 유용하지 않을 수 있지만
여러분과 저 같은 서버관리자들에게는 매우 유용한 방법이라 생각하기에 설명해 드립니다.

즉, 여러 개의 서로 다른 디렉토리내에 동일한 프로그램이나 어플리케이션이 설치되어 있는
경우에는 디렉토리 이름은 다르지만 각각의 디렉토리내에 들어있는 파일들은 모두 동일할
것입니다. 따라서 서로 다른 디렉토리내에 존재하는 동일한 이름의 파일을 한꺼번에 수정하
는 vi 명령어(셸스크립트)를 만들어 실행하면 여러 번 사용해야하는 vi 명령어를 한번의 명
령어로 해결 할 수 있습니다. 물론 각각 vi 가 오픈한 후에 수정해야 할 내용은 직접해야겠



본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
분이며 책 구입희망자분들께서 공개요청하셨던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1,11 편

지요.

자, 그럼 이와 같은 상황에서 awk 를 이용하여 vi 명령어를 생성해 보도록 하겠습니다. 현
재 위치에는 여러 개의 서로 다른 이름의 디렉토리가 존재하고 그 각각의 디렉토리 내에 존
재하는 post.html 파일을 수정하기 위한 것입니다. 아래와 같이 awk 를 이용하여 “vi 디렉
토리명/post.html” 와 같은 출력을 얻을 수 있습니다.

```
[root@file home]# ls -l | grep "^d" | awk '{print "vi "$9"/post.html"}'  
vi aceace/post.html  
vi anymore/post.html  
vi bankcd/post.html  
vi belpex/post.html  
vi celvise/post.html  
vi charge/post.html  
vi clubcm/post.html  
vi daesung/post.html  
vi darun/post.html  
vi dfc21/post.html  
vi dongnam/post.html  
vi doom4/post.html  
vi dotoru/post.html  
vi ds3etc/post.html  
vi ds5ezp/post.html  
vi easy8105/post.html  
vi event/post.html  
vi feelhouse/post.html  
vi fishing/post.html  
[root@file home]#
```

이제 위의 결과로 출력된 vi 문들을 모두 복사하여 임의의 파일로 저장하시면 됩니다. 필
자의 경우에는 이 파일의 이름을 modfile.sh 라고 하였습니다. 주의하실 것은 쉘스크립트
로 실행할 것이므로 맨 첫번째 행에는 “#!/bin/bash” 를 넣어주셔야 한다는 것입니다.

```
#!/bin/bash  
  
vi aceace/post.html  
vi anymore/post.html  
vi bankcd/post.html  
vi belpex/post.html  
vi celvise/post.html  
vi charge/post.html  
vi clubcm/post.html  
vi daesung/post.html  
vi darun/post.html  
vi dfc21/post.html  
vi dongnam/post.html  
vi doom4/post.html
```

본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
분이며 책 구입희망자분들께서 공개요청하였던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1,11 편

```
vi dotoru/post.html  
vi ds3etc/post.html  
vi ds5ezp/post.html  
vi easy8105/post.html  
vi event/post.html  
vi feelhouse/post.html  
vi fishing/post.html
```

그리고 아래와 같이 저장한 파일에 실행퍼미션을 주십시오.

```
[root@file home]# chmod 755 modfile.sh
```

그런다음 이제 아래의 예와 같이 modfile.sh 를 실행하시면 됩니다.

```
[root@file home]# ./modfile.sh
```

위와같이 실행하시면 서로다른 디렉토리에 존재하는 동일한 이름의 파일들을 대상으로 vi
명령어를 차례대로 자동수행할 수 있습니다.

이 방법은 여러사용자가 함께 사용하는 호스팅서버에서 각각 다른 디렉토리내에 존재하는
동일한 파일을 vi 로 차례대로 오픈하여 수정하는 방법을 생각하던 중 시도해 보았던 방법
인데 필자의 이 작은 경험이 여러분들의 서버관리에 유용하게 쓰일 수 있을 것 같아서 설명
한 것입니다.

실무명령어# 16-8 : 아파치 웹로그파일(access_log)에서 유일방문자수 카운트하기(유일 IP 주소 카운트)

서버관리자들에게 있어서 로그파일의 의미는 아무리 강조하여도 지나치지 않을 것입니다.
그중에서도 웹로그파일은 웹방문자들의 숫자를 카운트하거나 웹페이지별 방문자를 분석하는
등의 작업을 할 수가 있는 매우 유용한 파일입니다.

이번 장에서는 서버관리자들에게 흔히 요구되는 것이 홈페이지의 유일 IP 방문자숫자를 확인
해 달라는 것입니다. 물론, 유일 IP 를 카운트하여 분석하는 작업이 경력있는 서버관리자
들에게는 별 어려움 없는 일이겠지만 초보관리자들을 가끔씩 힘겹게하는 일이 될 수도 있
습니다.

또한, webalizer 나 accesswatch 등의 웹로그 분석프로그램을 사용하여 분석한다면 이또한
매우 쉬운 일입니다. 하지만, 필자가 여기서 알려드리고자 하는 방법은 서버관리자가 간단
한 명령어로 유일 IP 를 뽑아내는 방법에 대한 것입니다.

자, 그럼 먼저 우리가 흔히 볼 수 있는 웹로그파일의 이름이 access_log 입니다.

그리고 이 파일의 내용을 간단히 보면 다음과 같은 형식을 가지고 있습니다. 즉, 각행의 첫
번째 항목이 IP 주소라는 것을 확인하시기 바랍니다.



본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
분이며 책 구입희망자분들께서 공개요청하셨던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1,11 편

```
[root@host1 logs]# ls -l
합계 92408
-rw-r--r--  1 root    root    94525796  9월 25 21:46 access_log
[root@host1 logs]#
[root@host1 logs]# tail access_log
221.150.53.43 - - [25/Sep/2003:21:41:38 +0900] "GET /images/products/F0100213a.jpg
HTTP/1.1" 200 3534
221.150.53.43 - - [25/Sep/2003:21:41:38 +0900] "GET /images/products/F0100214a.jpg
HTTP/1.1" 200 3580
221.150.53.43 - - [25/Sep/2003:21:41:38 +0900] "GET /images/products/F0100215a.jpg
HTTP/1.1" 200 3601
221.150.53.43 - - [25/Sep/2003:21:41:38 +0900] "GET /images/products/F0100216a.jpg
HTTP/1.1" 200 3412
221.150.53.43 - - [25/Sep/2003:21:41:38 +0900] "GET /images/products/F0100217a.jpg
HTTP/1.1" 200 3362
221.150.53.43 - - [25/Sep/2003:21:41:39 +0900] "GET /images/products/F0100218a.jpg
HTTP/1.1" 200 3507
221.150.53.43 - - [25/Sep/2003:21:41:39 +0900] "GET /images/products/F0100219a.jpg
HTTP/1.1" 200 3507
221.150.53.43 - - [25/Sep/2003:21:41:41 +0900] "GET
/shopping.php?page=11&keyfield=&key=&bigcode=F&midcode=F01 HTTP/1.1" 200 28235
221.150.53.43 - - [25/Sep/2003:21:41:41 +0900] "GET /images/products/F0100220a.jpg
HTTP/1.1" 200 3202
221.150.53.43 - - [25/Sep/2003:21:41:41 +0900] "GET /images/products/F0100221a.jpg
HTTP/1.1" 200 3202
[root@host1 logs]#
```

자, 그럼 이제 아래와 같은 명령어로 access_log 파일의 전체 유일 IP 를 뽑아내 보겠습니다.

```
[root@host1 logs]# cat access_log | awk '{print $1}' > imsi ; sort -u imsi > imsi3 ;
wc -l imsi3
 4249 imsi3
[root@host1 logs]#
```

위와 같이 하면 현재의 access_log 파일에 유일 IP 주소를 카운트할 수 있습니다. 결론적으
로 위의 결과에서 웹페이지를 방문한 유일 IP 주소가 4249 개이므로 유일방문자수가 4249 명
이라는 것을 알 수가 있습니다.

위의 명령어를 간단히 복사하신 다음 여러분들의 access_log 파일이 존재하는 곳에서 그대
로 해보시기 바랍니다.

위의 명령어에 대한 설명을 간단히 하면 다음과 같습니다.

```
cat access_log
access_log 파일전체를 출력합니다.
```



본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
분이며 책 구입희망자분들께서 공개요청하였던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부분이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1,11 편

```
awk '{print $1}' > imsi
```

로그파일의 맨 첫번째 항목인 IP 주소부분만을 imsi 파일에 저장합니다.

```
sort -u imsi > imsi3
```

imsi 파일에 저장된 IP 주소들을 유일한 IP 로 정렬한 후에 이 데이터를 imsi3 에 저장합니다.
즉, imsi3 파일에는 중복되지않은 유일 IP 주소리스트들이 저장됩니다.

```
wc -l imsi3
```

imsi3 에 저장된 행(행단위로 IP 가 저장되어 있음)을 카운트합니다.

좀 복잡해 보이지만 명령어를 이해한 다음 한번 사용해 보시면 금방 익숙해 질 수 있습니다.

그리고 이런 복합적인 명령어를 자주 사용하시다 보면 서버작업을 매우 간편하게 할 수도
있습니다.

위와 같이 명령어 몇개를 조합하여 유일 IP 주소를 카운트 하고 있지 않습니까.

자, 이번에는 특정 일자에 해당하는 유일 IP 주소를 카운트하는 방법입니다.

```
[root@host1 logs]# grep 20W/SepW/2003 access_log | awk '{print $1}' > imsi ; sort -u  
imsi > imsi3 ; wc -l imsi3  
75 imsi3  
[root@host1 logs]#
```

즉, 위의 결과로 보아 2003 년 9 월 20 일자에 웹페이지를 방문한 유일방문자수는 75 명이라
는 것을 알 수 있습니다.

위의 명령어를 간단히 설명하면 다음과 같습니다.

```
grep 20W/SepW/2003 access_log
```

access_log 파일에서 20/Sep/2003 이라는 문자열들을 가진 행만을 뽑아냅니다. 즉, 특정 날
짜에 해당하는 문자가 들어있는 행만을 작업대상으로 뽑아낸 것입니다.

```
awk '{print $1}' > imsi
```

20/Sep/2003 에 해당하는 행들에서 IP 주소부분만을 뽑아서 imsi 파일에 저장합니다.

```
sort -u imsi > imsi3
```

imsi 파일에 저장된 데이터를 대상으로 유일데이터 정렬을 하여 imsi3 에 저장합니다 .

```
wc -l imsi3
```

imsi3 의 행의 수를 카운트합니다.

이렇게하면 특정날짜의 유일 IP 주소를 카운트할 수있습니다.



본 자료는 슈퍼유저코리아에서 출판한 “리눅스 실무명령어 바이블 (전 2 권)”의 본문중 일부
본이며 책 구입희망자분들께서 공개요청하셨던 부분중 요구횟수가 가장 많았던 부분만을 발
췌하여 공개한 것입니다. 비록 일부본이지만 유용하게 사용하시기 바랍니다.

리눅스 실무명령어 바이블 1, 2 편

여러분들께서도 위와 같은 방법으로 테스트해보시기 바랍니다. 위에서 20W/SepW/2003 라고
된 부분은 2003 년 9 월 20 일자라는 의미입니다. 즉, access_log 파일 전체에서 2003 년 9 월
20 일자의 유일 IP 방문자수는 75 명이 된다는 의미입니다.

만약 다른날짜의 유일 IP 방문자수를 카운트하고자 한다면 다음과 같이 일자부분과 월, 년부
분을 바꾸어 주시면 됩니다. 아래는 2003 년 9 월 21 일자 유일 IP 방문자 수 입니다.

```
[root@host1 logs]# grep 21W/SepW/2003 access_log | awk '{print $1}' > imsi ; sort -u  
imsi > imsi3 ; wc -l imsi3  
67 imsi3  
[root@host1 logs]#
```

다음은 2003 년 9 월 22 일자 유일 IP 방문자 수입니다.

```
[root@host1 logs]# grep 22W/SepW/2003 access_log | awk '{print $1}' > imsi ; sort -u  
imsi > imsi3 ; wc -l imsi3  
90 imsi3  
[root@host1 logs]#
```

필자가 예로든 awk 명령어들은 모두 실제 예입니다. 따라서 그대로 따라해 보셔도 무리없이
실행이 됨을 알 수 있습니다. 단, 위의 예와 같이 날짜부분에서 월에 해당하는 문자들은 9
월이기때문에 Sep 가 된 것이지만, 만약 7 월이라면 Jul 이 되며, 8 월이라면 Aug 등으로 해
주셔야합니다. 이 부분만 정확하게 알고 계신다면 다양한 응용이 가능합니다.

- 본 자료의 저작권은 “슈퍼유저코리아” (www.superuser.co.kr)에 있습니다 .
- 본자료는 “슈퍼유저코리아”에서 출판한 『리눅스 실무명령어 바이블(1 편, 2 편)』의 실제
내용중 일부본으로서 책의 본문내용을 확인하고자하시는 분들의 요청에 의해 요청횟수가
많았던 부분만을 일부 공개한 것입니다.
- 본 책자의 자세한 모든 정보는 <http://www.superuser.co.kr/linuxcommandbible/> 에서
확인하시기 바랍니다.
- 본 자료의 수정 및 편집은 허용되지않습니다.
- 본 책은 권수 : 2 권 (1 편, 2 편), 페이지수 : 총 1,859 페이지, 장(chapter): 382 장,
절: 1,900 개의 절로 구성되어있습니다.
- 저작권 및 문의전화(사무실) : TEL : 051-583-8128, FAX : 051-583-8130

