파일 입출력 관련 명령어

ITLEC
by 마하나임

cut

형식 cut [옵션] “파일이름”

사용 예 #cut –c 2-4 data

옵션 -c 문자수로 열을 계산해서 출력한다

-s <구분자> 필드 구분자를 사용한다.

-d <구분자> 파일 내의 필드로 따져서 추출한다.

-f filed

cat data

hong 28 011-222-2222 seoul
park 34 017-333-3333 kyunggi
im 23 019-444-4444 chungnam
son 49 016-555-5555 us
gil 19 018-666-6666 korea
jang 21 011-7777-7777 japan
lee 16 016-8888-8888 china
sa 45 017-9999-9999 canada
hwang 32 015-555-5555 kwangju

cut -c 2-4 data

ong
ark
m 2
on
il
ang
ee
a 4
wan

cut -d " " -f 1,3 data

hong 011-222-2222
park 017-333-3333
im 019-444-4444
son 016-555-5555
gil 018-666-6666
jang 011-7777-7777
lee 016-8888-8888
sa 017-9999-9999
hwang 015-555-5555

tail -5 /etc/passwd
listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:Nobody:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x Nobody:/:
user1:x:100:1::/drive/user1:/bin/ksh

문제1) /etc/passwd 파일에서 첫 번째 필드와, 세 번째부터 다섯 번째 필드를 : 구분자를 기준으로 나누어서 마지막 5줄
만 출력하라.

cut -d : -f 1,3-5 /etc/passwd | tail -5
listen:37:4:Network Admin
nobody:60001:60001:Nobody
noaccess:60002:60002:No Access User
nobody4:65534:65534:SunOS 4.x Nobody
user1:100:1:

paste

형식 paste [옵션] "파일이름" "파일이름"
사용 예 #paste –d : exam1 exam2
옵션 -s 한 파일의 내용을 한 줄로 보여준 후 다른 파일의 내용을 한 줄로 덧붙인다.
-d 출력되는 내용의 구분자를 지정한다.

cat exam1
red
blue
white

cat exam2
yellow
green
gray
black

paste exam1 exam2
red yellow
blue green
white gray

black

paste -d : exam1 exam2

red:yellow

blue:green

white:gray

:black

paste -s -d "|" exam1 exam2

red|blue|white|

yellow|green|gray|black

diff

형식 diff [옵션] “파일이름” “파일이름”

사용 예 # diff exam1 exam2

옵션 -b space를 무시하고 비교한다.

cat exam1

red

blue

white

cat exam3

red

green

blue

white

diff exam1 exam3

1a2

> green

grep

형식 grep [옵션][패턴] “파일이름”

사용 예 # grep seoul data

옵션 -v ‘패턴’을 포함하지 않는 행을 출력

-i 대소문자를 구분하지 않는다.

-n 줄 번호를 함께 출력한다.

-l 파일명을 출력한다.

-c 일치하는 라인의 개수.

특정 문자열 찾기:

grep copying help

- help 파일에서 copying 이라는 문자열을 포함하는 각줄을 보여준다.

정규식을 사용한 예:

grep -n '[dD]on₩'t' tasks

- tasks 파일에서 don't 나 Don't 문자열을 포함하는 각 줄을 그 줄 번호와 함께(-n 옵션) 보여준다.

- 이 예제에서 사용된 패턴은 [,] 같은 쉘에서 특별한 의미로 쓰이는 문자들을 포함하며, 하나 이상의 정규식이 사용되었기에 따옴표로 묶어준 경우이다. 큰 따옴표를 사용한다면, 작은 따옴표도 하나의 패턴임을 지시한다. 즉, on't 문자열을 on₩'t 로 표시할 필요가 없다.

파이프를 이용한 예:

ls -l | grep '^d.....x'

- 현재 디렉토리 내용 중에 다른 사용자에게 실행 권한이 부여된 하위 디렉토리가 어떤 것이 있는지를 알아보는 경우이다.

This lists all the directories in the current directory for which other users have execute permission.

- 정규식에서 ^ 문자는 그 줄의 처음을 뜻한다. 즉, 윗 경우는 각 줄의 첫칸에 d 로 시작해서, 10 번째 칸에 있는 문자가 x 인 임의의 문자열을 찾는 경우이다.

결과를 방향 전환해서 파일로 저장하는 경우:

grep Smith /etc/passwd > smurffs

- passwd 파일에서 Smith 문자열을 포함하는 줄을 찾아, 그 결과를 smurffs 파일에 기록 하는 경우이다. 이것은 결국 현재 시스템에서 사용하고 있는 사람 중에 그 username 이 Smith 이거나, 실재 이름이 Smith 인 모든 사람을 찾아 볼 수 있다.

원하는 파일 삭제하는 예:

rm -f $(grep -l "패턴")

 # grep 011 data

hong 28 011-222-2222 seoul

jang 21 011-7777-7777 japan

 # grep -n 011 data ← n 옵션은 라인번호를 보여 준다.

1:hong 28 011-222-2222 seoul

6:jang 21 011-7777-7777 japan

예)

 # cat exam1

red

blue

white

 # cat exam2
yellow
green
gray
black
 # cat exam3
red
green
blue
white

 # grep gray *
exam2:gray
 # grep -n gray *
exam2:3:gray

 # grep GRAY *
 # grep -in GRAY * ← -i 옵션은 대소문자를 구분 하지 않게 해준다.
exam2:3:gray
 # grep -l telnet /etc/* ← -l 옵션은 찾고자 하는 패턴을 포함하는 파일명만 출력시켜준다.
/etc/inetd.conf

sort

형식 sort [옵션] "파일이름"
사용 예 # sort –k2 –r data
옵션 **-f 대소문자를 구분하지 않는다.**
-r 내림차순으로 정렬
-b space를 무시한다.
-k 필드 번호를 나타낸다.
-t <구분자> 필드 구분자로 <구분자>를 사용한다.
-n 숫자 순서로 정렬
sort를 옵션 없이 사용할 경우에는 첫 번째 필드를 기준으로 오름차순으로 정렬된다.
 # cat data
hong 28 011-222-2222 seoul
park 34 017-333-3333 kyunggi
im 23 019-444-4444 chungnam
son 49 016-555-5555 us
gil 19 018-666-6666 korea
jang 21 011-7777-7777 japan
lee 16 016-8888-8888 china

sa 45 017-9999-9999 canada
hwang 32 015-555-5555 kwangju

 # sort data
gil 19 018-666-6666 korea
hong 28 011-222-2222 seoul
hwang 32 015-555-5555 kwangju
im 23 019-444-4444 chungnam
jang 21 011-7777-7777 japan
lee 16 016-8888-8888 china
park 34 017-333-3333 kyunggi
sa 45 017-9999-9999 canada
son 49 016-555-5555 us

-r 옵션은 내림차순으로 정렬시키는 옵션이다.
 # sort -r data
son 49 016-555-5555 us
sa 45 017-9999-9999 canada
park 34 017-333-3333 kyunggi
lee 16 016-8888-8888 china
jang 21 011-7777-7777 japan
im 23 019-444-4444 chungnam
hwang 32 015-555-5555 kwangju
hong 28 011-222-2222 seoul
gil 19 018-666-6666 korea
정렬의 기준이 되는 필드를 지정하고자 한다면 -k 옵션을 사용한다.
 # sort -k3 data
hong 28 011-222-2222 seoul
jang 21 011-7777-7777 japan
hwang 32 015-555-5555 kwangju
son 49 016-555-5555 us
lee 16 016-8888-8888 china
park 34 017-333-3333 kyunggi
sa 45 017-9999-9999 canada
gil 19 018-666-6666 korea
im 23 019-444-4444 chungnam

문제1) 나이가 많은 연장자부터 순서대로 정렬하여 다른 파일에 저장하여 보자.
sort -k2 -r data > data2
cat data2
son 49 016-555-5555 us
sa 45 017-9999-9999 canada
park 34 017-333-3333 kyunggi

hwang 32 015-555-5555 kwangju
hong 28 011-222-2222 seoul
im 23 019-444-4444 chungnam
jang 21 011-7777-7777 japan
gil 19 018-666-6666 korea
lee 16 016-8888-8888 china

-t 옵션은 필드 구분자를 지정하는 역할을 한다.

 # sort data

gil 19 018-666-6666 korea
hong 28 011-222-2222 seoul
hwang 32 015-555-5555 kwangju
im 23 019-444-4444 chungnam
jang 21 011-7777-7777 japan
lee 16 016-8888-8888 china
park 34 017-333-3333 kyunggi
sa 45 017-9999-9999 canada
son 49 016-555-5555 us

 # sort -t " " -k2 data

lee 16 016-8888-8888 china
gil 19 018-666-6666 korea
jang 21 011-7777-7777 japan
im 23 019-444-4444 chungnam
hong 28 011-222-2222 seoul
hwang 32 015-555-5555 kwangju
park 34 017-333-3333 kyunggi
sa 45 017-9999-9999 canada
son 49 016-555-5555 us

문제2) /etc/passwd 파일을 : 구분자를 이용하여 분류하여 세 번째 필드인 uid를 기준으로 숫자순서대로 내림차순으로 정렬한 후 위에서 5줄 까지만 출력하여라.

 # sort -t : -k3 -r /etc/passwd | head -5

nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
nobody4:x:65534:65534:SunOS 4.x Nobody:/:
noaccess:x:60002:60002:No Access User:/:
nobody:x:60001:60001:Nobody:/:

 # sort -t : -k3 -n -r /etc/passwd | head -5 ← 여기서 –n 옵션은 숫자순서로 정렬 한다는 의미이다.

nobody4:x:65534:65534:SunOS 4.x Nobody:/:
noaccess:x:60002:60002:No Access User:/:

nobody:x:60001:60001:Nobody:/:
user1:x:100:1::/drive/user1:/bin/ksh
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
/etc/passwd 파일을 ':' 기호를 구분자로 하여(-t :), 세 번째 필드인 uid를 기준으로(-k3), 숫자순서대로(-n) 내림차순으로(-r) 정렬한 후, 위에서 5라인 까지만(head)출력 하였다.

sed

형식 sed [옵션] "파일이름"
사용 예 # sed 's/01/82-1/g' data
옵션 p 행을 출력한다(-n 옵션과 함께 사용할 경우, 선택된 행만 출력한다.)
d 선택한 행을 삭제한다.
-f 파일 안의 내용을 실행한다.
's/가/나/g' '가' 문자열을 '나' 문자열로 대체한다.
-e 다중편집을 한다.
-q sed를 종료한다.

(1) 개념

sed 는 Stream Editor 의 약자로서 파일의 수정을 주 목적으로 한다.
이는 파일을 순방향으로 읽는 동안 연산을 수행하며 텍스트 화일에서의 반복 수정에 용이하다.

(2) 기능

주어진 텍스트 패턴을 갖는 모든 행을 delete
특정 행에서 어떤 텍스트 패턴을 다른 패턴으로 바꿈
하나의 파일을 다른 곳의 파일로 복사
입력화일의 특정부분을 출력화일로 보냄

sed '/011/p' data ← 패턴에 따라 내용이 나오나 같은 내용이 두번 반복되어 나오는 것을 볼수 있다.
hong 28 011-222-2222 seoul
hong 28 011-222-2222 seoul
park 34 017-333-3333 kyunggi
im 23 019-444-4444 chungnam
son 49 016-555-5555 us
gil 19 018-666-6666 korea
jang 21 011-7777-7777 japan
jang 21 011-7777-7777 japan
lee 16 016-8888-8888 china
sa 45 017-9999-9999 canada
hwang 32 015-555-5555 kwangju

sed -n '/011/p' data ← 패턴에 해당하는 부분만 보려면 –n 옵션으로 사용한다.
hong 28 011-222-2222 seoul
jang 21 011-7777-7777 japan

 # sed '1,3d' data ← 1번 라인부터 3번 라인까지 삭제하고 출력

son 49 016-555-5555 us
gil 19 018-666-6666 korea
jang 21 011-7777-7777 japan
lee 16 016-8888-8888 china
sa 45 017-9999-9999 canada
hwang 32 015-555-5555 kwangju

문제1) data 파일의 내용을 5번째 라인부터 끝까지 삭제하고 출력하자.

sed '5,$d' data
hong 28 011-222-2222 seoul
park 34 017-333-3333 kyunggi
im 23 019-444-4444 chungnam
son 49 016-555-5555 us

 # sed '4q' data ← q 옵션은 종료옵션이다. 4q는 4라인까지 출력한 후 종료하라는 의미이다.

hong 28 011-222-2222 seoul
park 34 017-333-3333 kyunggi
im 23 019-444-4444 chungnam
son 49 016-555-5555 us

문제2) seoul 이라는 특정 문자열이 포함된 라인을 제외하고 data 파일을 출력하자.

cat data
hong 28 011-222-2222 seoul
park 34 017-333-3333 kyunggi
im 23 019-444-4444 chungnam
son 49 016-555-5555 us
gil 19 018-666-6666 korea
jang 21 011-7777-7777 japan
lee 16 016-8888-8888 china
sa 45 017-9999-9999 canada
hwang 32 015-555-5555 kwangju
sed '/seoul/d' data
park 34 017-333-3333 kyunggi
im 23 019-444-4444 chungnam
son 49 016-555-5555 us
gil 19 018-666-6666 korea
jang 21 011-7777-7777 japan
lee 16 016-8888-8888 china

sa 45 017-9999-9999 canada
hwang 32 015-555-5555 kwangju

sed 명령어로 파일 내의 특정한 문자열을 다른 분자열로 대체할 수 있다.
cat data
hong 28 011-222-2222 seoul
park 34 017-333-3333 kyunggi
im 23 019-444-4444 chungnam
son 49 016-555-5555 us
gil 19 018-666-6666 korea
jang 21 011-7777-7777 japan
lee 16 016-8888-8888 china
sa 45 017-9999-9999 canada
hwang 32 015-555-5555 kwangju

‘s/대상문자/바꿀문자/g’
‘n,ms/대상문자/바꿀문자/g’

sed **'s/japan/bosung/g'** data ← 이 부분은 vi editer에서도 적용 되는 옵션이다. ‘s
hong 28 011-222-2222 seoul
park 34 017-333-3333 kyunggi
im 23 019-444-4444 chungnam
son 49 016-555-5555 us
gil 19 018-666-6666 korea
jang 21 011-7777-7777 bosung
lee 16 016-8888-8888 china
sa 45 017-9999-9999 canada
hwang 32 015-555-5555 kwangju

문제) data 파일 목록에서 japan 라인을 제외하고, 이름과 전화번호로 구성된 내용을 위에서 5번째 라인부터 출력하여라.
#sed ‘/japan/d’ data | cut –d “ “ –f 1,3 | tail + 5

만약 변경된 라인만 출력하려 할 때는 –n 옵션과 끝 부분을 ‘gp’로 하면 된다.
 # sed –n '5,$s/china/busan/gp' data
lee 16 016-8888-8888 busan

-e 다음의 command는 편집 대본이다. 여기서는 여러 개의 편집 대본을 하나의 명령행에 지정가능하다
cat data
hong 28 011-222-2222 seoul
park 34 017-333-3333 kyunggi
im 23 019-444-4444 chungnam

son 49 016-555-5555 us
gil 19 018-666-6666 korea
jang 21 011-7777-7777 japan
lee 16 016-8888-8888 china
sa 45 017-9999-9999 canada
hwang 32 015-555-5555 kwangju

sed -e 's/us/babo/g' -e '/china/d' data

hong 28 011-222-2222 seoul
park 34 017-333-3333 kyunggi
im 23 019-444-4444 chungnam
son 49 016-555-5555 babo
gil 19 018-666-6666 korea
jang 21 011-7777-7777 japan
sa 45 017-9999-9999 canada
hwang 32 015-555-5555 kwangju

awk

1. awk 의 기본 개념

1) awk 란?

; awk 란 이름은 이 유틸리티를 작성한 A.V.Aho, P.J. Weinberger, B. Kernigham 의 머리글자를 따온 것
① awk 는 일종의 프로그래밍 언어지만 일반적인 언어라기 보다는 주로 패턴의 검색과 조작을 주목적으로 만들어진 것이다.
② 파일의 각 라인에서 필드(field)를 인식할 수 있는 패턴 매칭 기능을 가지고 이들 필드를 자유자재로 조작 가능한 유틸리티를 작성하고자 만든 것이다.

2) awk 의 응용분야

데이터 프로세싱, 리포트 작성, 간단한 데이터베이스 구축, 등

3) awk 를 이용한 작업

① 프로그래머가 자신이 작성한 프로그램의 입력 화일이 특정한 형식에 들어 맞게 이루어져 있는지 검사.
② 출력화일을 처리하여 리포트를 만들어 냄.
③ 다른 프로그램의 입력 형식에 맞게 변환하는 작업에 이용.

2. awk 프로그램의 구조 및 실행

(1) awk 프로그램의 구조

1) awk ' pattern {action}

pattern {action}

.
.
.
' filenames <----------입력화일(예제 : students)

2) awk -f parttern-action-file filenames <----- 입력화일
awk 실행 action 을 가진 프로그램 file

패턴

BEGIN	특정 명령을 실행하기 전에 먼저 실행시킨다.
END	특정 명령을 실행한 후 제시되는 문장을 실행시킨다.
/정규표현식/	정규표현식의 패턴을 포함하는 라인에서 문장을 실행시킨다.
패턴1 && 패턴2	패턴1과 패턴2를 동시에 만족시킬 때 문장을 실행시킨다.
패턴1 || 패턴2	패턴1이나 패턴2 중 하나만 만족시켜도 문장을 실행시킨다.
! 패턴	패턴과 일치하지 않을 경우 문장을 실행시킨다.

for (x=1,x<10,x+ -)

연산자

산술 연산자	+	더하기
	-	빼기
	*	곱하기
	/	나누기
	A % B	A를 B로 나눈 후 나머지 값
	+ +	어떤 값에서 1을 증가
	+ -	어떤 값에서 1을 감소
대입연산자	A = B	A = B
	A + = B	A = A+ B
	A -= B	A = A-B
	A *= B	A = A*B
	A /= B	A = A/B
	A %= B	A = A%B
논리연산자	||	or
	&&	and
	!	not
비교연산자	A > B	A가 B보다 크다.
	A >= B	A가 B보다 크거나 같다.
	A < B	A가 B보다 작다
	A <=B	A가 B보다 작거나 같다.
	==	A와 B가 같다.

!=	A와 B가 다르다.

==========

awk의 내부변수

==========

FILENAME	현재 처리되고 있는 파일 이름
FS	필드 구분자
RS	레코드 구분자
NF	현재 레코드에서의 필드 수
NR	현재 파일에서 전체 레코드 수
OFS	출력시의 필드 구분자
ORS	출력시의 레코드 구분자

==========

 # cat data

hong 28se 011-222-2222 seoul
park 34se 017-333-3333 kyunggi
im 23se 019-444-4444 chungnam
son 49se 016-555-5555 us
gil 19se 018-666-6666 korea
jang 21se 011-7777-7777 japan
lee 16se 016-8888-8888 china
sa 45se 017-9999-9999 canada
hwang 32se 015-555-5555 kwangju

 # awk '{print $1,$3}' data ← $1, $3 필드를 출력하라는 말이다. 만약 전체 필드를 출력하려면 $0 인자를 사용한다.

hong 011-222-2222
park 017-333-3333
im 019-444-4444
son 016-555-5555
gil 018-666-6666
jang 011-7777-7777
lee 016-8888-8888
sa 017-9999-9999
hwang 015-555-5555

문제1) 25세 이상인 사람의 이름과 나이를 출력하라.

 # awk '$2 > 25 {print $1,$2}' data

hong 28se
park 34se
son 49se
sa 45se
hwang 32se

하지만 이보다 정확한 데이터 추출을 위해서는 se 라는 문자를 제거해줄 필요가 있다.

```
# sed 's/se / /g' data | awk '$2 > 25 {print $1,$2}'
```

hong 28
park 34
son 49
sa 45
hwang 32

```
# sed 's/se / /g' data | awk '$2 > 25 {print $1"씨는 ",$2"세 입니다."}'
```

hong씨는 28세 입니다.
park씨는 34세 입니다.
son씨는 49세 입니다.
sa씨는 45세 입니다.
hwang씨는 32세 입니다.

```
# cat > sedfile1
s/se / /g
# cat > awkfile1
$2 > 25 {print $1"씨는 ",$2"세 입니다"}

# sed -f sedfile1 data | awk -f awkfile1
```

hong씨는 28세 입니다
park씨는 34세 입니다
son씨는 49세 입니다
sa씨는 45세 입니다
hwang씨는 32세 입니다

문제2) 나이의 평균을 구하는 프로그램을 구현해보자.

```
# cat sedfile1
s/se / /g
# cat > awkfile2
BEGIN {
        sum = 0;
        line = 0;
}
{
        sum + = $2;
        line + + ;
}
END {
average = sum / line;
print "나이의 평균 : "average"세";
}
# sed -f sedfile1 data | awk -f awkfile2
나이의 평균 : 29.6667세
```

(1) 예제 입력 파일 소개

① 입력화일의 이름은 students

② 이 파일의 각 라인은 3 개의 필드로 구성(학생 성명, 학과명, 나이)

③ 각 필드는 공백에 의해서 분리(공백을 필드 분리자로 간주함.)

< awk 는 각 라인에서 필드를 추출해 내는 데 필드 분리자(field separator)를 사용, 필드 분리자는 보통 하나 이상의 공백 문자이다.>

1) 입력화일 예제 students

```
$ cat students
John,P Physics 20
Rick,L Mechanical 21
Jack,T electrical 23
Larry,M Chemical 22
Phil,R Electrical 21
Mike,T mechanical 22
Paul,R Chemical 23
```

John,T Chemical 23
Tony,N Chemical 22
James,R Electrical 21

예 1) 식(expression)에 맞는 field 프린트하기

$ awk '$3 > 22 {print $1}' students
Jack,T
Paul,R
John,T

예 2) if 문을 사용하여 조건에 맞는 line 분리하기(각 파일에 저장)

step 1 : if 문을 사용하는 프로그램을 awkprog1 이라는 파일로 만든다.

$ cat awkprog1
{ if ($1 ~ /^J/) printf "%s₩n", $0 > "Jfile"
if ($1 ~ /^P/) printf "%s₩n", $0 > "Pfile"}

step 2 : students 입력화일에 awkpog1 프로그램 화일을 적용한다.

$ awk -f awkprog1 students

step 3 : 결과 보기

$ cat Jfile
John,P Physics 20
Jack,T electrical 23
John,T Chemical 23
James,R Electrical 21

$ cat Pfile
Phil,R Electrical 21
Paul,R Chemical 23

예 3) 평균값 구하기

<프로그램 awkprog2, awkprog3>

$ cat awkprog2
{sum + = $3}
END {printf "The average of the ages is %.2f₩n", sum/NR}

```
$ cat awkprog3
{sum + = $3
+ + no}
END {printf "The average of the ages is %.2f₩n", sum/no}
```

<결 과>

```
$ awk -f awkprog3 students
The average of the ages is 21.80
```

예 4) while 과 do 문을 이용하여 평균값 구하기

<프로그램 awkprog4>

```
$ cat awkprog4
{if (NF > 0) {
sum = 0
n = 1
while (n <= NF) {
sum = sum +  $n
n = n+ 1
}
printf "Average is %d₩n", sum/NF
}
else
print}
```

<예 제>

```
% awk -f awkprog4 test
Average is 17
Average is 3
Average is 25
Average is 0
```