

Part 01. 리눅스 개발 환경

Chapter 07. 쉘 프로그래밍

진행 순서

Chapter 07_01	awk 개요
Chapter 07_02	awk 실습
Chapter 07_03	sed 개요
Chapter 07_04	sed 실습

Chapter 07_01 awk 개요

AWK(오크)는 유닉스에서 처음 개발된 일반 스크립트 언어이다.

AWK의 기본 기능은 텍스트 형태로 되어있는 입력 데이터를 행과 단어 별로 처리해 출력하는 것이다.

AWK라는 이름은 이 스크립트 언어를 만든 **Aho, Weinberger, Kernighan** 세 명의 성의 앞 글자를 따서 붙여졌다.

명령의 수행 결과나 파일의 데이터 내용을 한 줄씩 읽어 들여,
한 줄의 내용을 단어 단위로 끊어서 읽어 들이고 이를 조작 및 연산에 활용할 수 있다.

실행

awk '패턴 {동작} 패턴 {동작} ... 패턴 {동작}' 파일명

command | awk '패턴 {동작} 패턴 {동작} ... 패턴 {동작}'

awk -f awk파일명 파일명

BEGIN { 동작 } # 입력을 읽기 전에 주어진 '동작'을 먼저 실행한다.

END { 동작 } # 위와 비슷하다. 입력을 모두 훑고 마지막에 주어진 '동작'을 실행한다.

/패턴/ # '패턴'에 일치하는 줄을 출력한다.

{ 동작 } # 매 줄을 읽을 때마다 '동작'을 실행한다.

print는 텍스트를 출력한다.

주요 키워드

FILENAME – 입력된 파일의 이름

NR – 현재 행의 번호

NF – 현재 행의 단어의 개수

FS – 입력 단어 분리자

OFS – 출력 단어 분리자

Chapter 07_02 awk 실습

```
1, Lee, 02-123-4567, 010-1234-4567, Seoul
2, Kim, 031-321-2222, 010-234-5678, Incheon
3, Won, 053-100-5555, 010-2356-7788, Daegu
```

information.txt

```
#!/bin/sh
/bin/awk \
'BEGIN { FS = ","; print "address book" } \
  { idx = $1 } { name = $2 } { phone = $3 } { mobile = $4 } { city = $5 } \
  { n += 1 } \
  { print "< (" idx ") " name ">" } \
  { print " Phone Number: " phone } \
  { print " Mobile Number: " mobile } \
  { print " City: " city } \
END { print "Total " n " address." }' information.txt
```

awk_test.sh

```
[root@localhost]# ./awk_test.sh
address book
< (1) Lee >
  Phone Number: 02-123-4567
  Mobile Number: 010-1234-4567
  City: Seoul
< (2) Kim >
  Phone Number: 031-321-2222
  Mobile Number: 010-234-5678
  City: Incheon
< (3) Won >
  Phone Number: 053-100-5555
  Mobile Number: 010-2356-7788
  City: Daegu
Total 3 address.
```

Chapter 07_02 awk 실습

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
...
```

/etc/passwd

```
#!/bin/sh
/bin/awk \
'BEGIN { FS = ":" } \
      { print $1 " home at " $6 }' \
/etc/passwd
```

users_home.sh

```
[root@localhost ch7]# ./users_home.sh
root home at /root
bin home at /bin
daemon home at /sbin
...
```

```
[root@localhost ]# awk -F: '{ print $1 " home at " $6 }' /etc/passwd
root home at /root
bin home at /bin
daemon home at /sbin
...
```

Chapter 07_02 awk 실습

```
1, Lee, 02-123-4567, 010-1234-4567, Seoul  
2, Kim, 031-321-2222, 010-234-5678, Incheon  
3, Won, 053-100-5555, 010-2356-7788, Daegu
```

information.txt

```
[root@localhost]# awk -F, '{print $2 "\t" $4}' information.txt  
Lee    010-1234-4567  
Kim    010-234-5678  
Won    010-2356-7788
```

상기와 같이 **awk**를 이용하여 패턴 일치가 성공하면, 특정 필드만 출력하도록 할 수 있습니다.

```
[root@localhost]# awk '{++cnt} END {print "Count = ", cnt}' information.txt  
Count = 3
```

상기와 같이 **awk**를 이용하여 카운팅 등 연산이 가능합니다.

Chapter 07_03 sed 개요

sed(stream editor)는 유닉스에서 텍스트를 분해하거나 변환하기 위한 프로그램이다.
sed는 벨 연구소의 리 E. 맥마흔이 1973년부터 1974년까지 개발하였고,
현재 유닉스 등의 여러가지 운영 체제에서 사용 가능하다.

대상이 되는 파일의 내용을 원하는 형태로 변경 가능

사용법

```
sed 's/regexp/replacement/g' inputFileName > outputFileName
```

inputFileName을 읽어 전체에서 **regexp** 패턴을 찾아 **replacement**로 치환한 후 **outputFileName**으로 쓴다.

필터

```
generateData | sed 's/x/y/g'
```

generateData 수행 후 데이터를 만든 다음 **x**를 **y**로 치환한다.

```
# echo xyz xyz | sed 's/x/y/g'
```

```
yyz yyz
```

파일

```
sed -f subst.sed inputFileName > outputFileName
```

키워드

i: 문자열 삽입

a: 새로운 행을 추가

d: 행 삭제

s: 특정 문자열을 다른 문자열로 변환

c: 특정 행을 다른 행으로 변환

p: 출력

Chapter 07_04 sed 실습

```
1,Lee,02-123-4567,010-1234-4567,Seoul  
2,Kim,031-321-2222,010-234-5678,Inchon  
3,Won,053-100-5555,010-2356-7788,Daegu
```

information.txt

```
[root@localhost]# sed 's/Won/Woo/g' information.txt  
1,Lee,02-123-4567,010-1234-4567,Seoul  
2,Kim,031-321-2222,010-234-5678,Inchon  
3,Woo,053-100-5555,010-2356-7788,Daegu
```

전체에서 문자열 “Won”을 “Woo”로 변경

```
[root@localhost ch7]# sed 's/031/02/1' information.txt  
1,Lee,02-123-4567,010-1234-4567,Seoul  
2,Kim,02-321-2222,010-234-5678,Inchon  
3,Won,053-100-5555,010-2356-7788,Daegu
```

문자열 “031”이 발견되는 1st(첫 번째) 패턴을 “02”로 변경

Chapter 07_04 sed 실습

```
1, Lee, 02-123-4567, 010-1234-4567, Seoul
2, Kim, 031-321-2222, 010-234-5678, Incheon
3, Won, 053-100-5555, 010-2356-7788, Daegu
```

information.txt

```
[root@localhost]# sed '2d' information.txt
1, Lee, 02-123-4567, 010-1234-4567, Seoul
3, Won, 053-100-5555, 010-2356-7788, Daegu
```

2번째 줄 삭제

```
[root@localhost ch7]# sed '/Won/d' information.txt
1, Lee, 02-123-4567, 010-1234-4567, Seoul
2, Kim, 031-321-2222, 010-234-5678, Incheon
```

문자열 “Won”이 발견되는 라인 삭제

```
[root@localhost ch7]# sed '2i\2, New, 00-000-0000, 000-000-0000, Korea' information.txt
1, Lee, 02-123-4567, 010-1234-4567, Seoul
2, New, 00-000-0000, 000-000-0000, Korea
2, Kim, 031-321-2222, 010-234-5678, Incheon
3, Won, 053-100-5555, 010-2356-7788, Daegu
```

2번째 줄에 새로운 라인 삽입

Chapter 07_04 sed 실습

/proc/meminfo 파일 정보를 읽어 메모리 사용률을 보여주는 스크립트

```
[root@localhost]# cat /proc/meminfo
MemTotal:      841104 kB
MemFree:       372984 kB
MemAvailable:  532328 kB
Buffers:       3268 kB
Cached:        260832 kB
...
```

```
#!/bin/sh
total=`cat /proc/meminfo | grep MemTotal | /bin/awk '{print $2}'`
free=`cat /proc/meminfo | grep MemFree | /bin/awk '{print $2}'`
use=`expr $total - $free`
echo "Mem Usage: $((use * 100 / total)) %"
```

mem_usage.sh

```
[root@localhost]# ./mem_usage.sh
Mem Usage: 55 %
```