

UNIX 및 실습

# 11장 보충 – awk (1)

# awk란?

- ▶ 데이터 조작 및 보고서 생성에 사용되는 유닉스 프로그래밍 언어
  - ▶ 개발자 세 사람(Alfred **A**ho, Peter **W**einberger, Brian **K**ernighan)의 이름 첫글자로 조합
  - ▶ nawk : awk의 최신 버전
  - ▶ gawk : GNU 버전
- ▶ 명령으로 간단한 조작을 할 수 있으며, 큰 규모의 응용 프로그램 작성도 가능
- ▶ 셸 스크립트와 소규모 데이터베이스 관리에서 빼놓을 수 없는 유용한 툴
- ▶ 사용자가 지정한 패턴 검색이나 특별한 작업을 수행하기 위해 파일(또는 입력)을 줄 단위로 조사
- ▶ 특정 작업이 없는 패턴의 경우 패턴과 일치된 모든 줄이 출력되고, 특정 작업이 지정된 경우 해당 작업을 수행한 결과 출력

# awk 형식 (1)

- ▶ awk 명령과 다음표로 묶은(혹은 파일에 있는) 프로그램 명령, 입력 파일명으로 구성
  - ▶ gawk 'pattern' filename
  - ▶ gawk '{action}' filename
  - ▶ gawk 'pattern {action}' filename

## ▶ 예제 파일 (employees)

Tom Jones	4424	5/12/66	543354
Mary Adams	5346	11/4/63	28765
Sally Chang	1654	7/22/54	567534
Billy Black	1683	9/23/44	335678

### ▶ 실행 예

```
[kgu@lily ~]$ gawk '/Mary/' employees
Mary Adams 5346      11/4/63      28765
```

```
[kgu@lily ~]$ gawk '{print $1}' employees
Tom
Mary
Sally
Billy
```

```
[kgu@lily ~]$ gawk '/Sally/ {print $1, $2}' employees
Sally Chang
```

# awk 형식 (2)

## ▶ 파이프 연결

- ▶ *command* | gawk 'pattern' filename
- ▶ *command* | gawk '{action}' filename
- ▶ *command* | gawk 'pattern {action}' filename

## ▶ 실행 예

```
[kgu@lily ~]$ df | gawk '$4 > 3000000'
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/mapper/vg_lily-lv_root	51606140	14985396	33999304	31%	/
/dev/mapper/vg_lily-lv_home	182104276	28573372	144280500	17%	/home

```
[kgu@lily ~]$ last | gawk '/kgu/ {print $3}'
```

220.68.173.204  
59.3.145.29  
203.232.252.170  
220.68.173.204  
220.68.173.201  
203.232.252.170  
220.68.173.204  
203.232.252.111

# 출력 형식 지정 (1)

- ▶ awk 명령의 동작 부분은 중괄호{ }로 묶는다
  - ▶ 동작 부분이 생략되어 있으면 기본 동작 수행
  - ▶ print는 특별한 서식이 지정되지 않은 간단한 출력에 사용 - 복잡한 서식이 필요한 경우 printf와 sprintf 사용

## ▶ 예제

```
[kgu@lily ~]$ date
2013. 05. 20. (월) 08:46:06 KST
[kgu@lily ~]$ date |awk '{print "Month: " $2 "Year: " $1}'
Month: 05.
Year: 2013.
```

## ▶ Escape Sequences

<code>\b</code>	백스페이스
<code>\f</code>	Form feed
<code>\n</code>	New line
<code>\r</code>	Carriage return
<code>\t</code>	Tab
<code>\047</code>	8진수 47(ASCII 코드)
<code>\c</code>	<code>c</code> 를 문자 자체로 표현 (예: <code>\w</code> )

```
[kgu@lily ~]$ gawk '/Sally/ {print "\t\t\tHave a nice day, " $1, $2
"!}" employees
Have a nice day, Sally Chang!
```

## ▶ OFMT 변수 : 숫자 서식을 지정하는 특수 변수

```
[kgu@lily ~]$ gawk 'BEGIN {OFMT="%.2f"; print 1.23456, 1.23E-2}
1.23 0.01
```

# 출력 형식 지정 (2)

## ▶ printf

```
[kgu@lily ~]$ echo "UNIX" | gawk ' {printf "|%-15s|\n", $1}'  
|UNIX|  
[kgu@lily ~]$ echo "UNIX" | gawk ' {printf "|%15s|\n", $1}'  
|          UNIX|
```

```
[kgu@lily ~]$ gawk '{printf "The name is: %-15s ID is %8d\n", $1, $3}' employees  
The name is: Tom           ID is      4424  
The name is: Mary         ID is      5346  
The name is: Sally        ID is      1654  
The name is: Billy        ID is      1683
```

# 파일에서 awk 명령 사용

## ▶ -f 옵션으로 파일 지정

```
[kgu@lily ch11_awk]$ cat awkfile
/^Mary/{print "Hello, Mary !"}
{print $1, $2, $3}
[kgu@lily ch11_awk]$ gawk -f awkfile employees
Tom Jones 4424
Hello, Mary !
Mary Adams 5346
Sally Chang 1654
Billy Black 1683
```

# 레코드와 필드

- ▶ 레코드 구분자 : CR(Carriage Return)
- ▶ \$0 : 전체 레코드
- ▶ NR : 레코드 번호를 저장하는 내장변수

```
[kgu@lily ch11_awk]$ gawk '{print NR, $0}' employees
1 Tom Jones 4424 5/12/66 543354
2 Mary Adams 5346 11/4/63 28765
3 Sally Chang 1654 7/22/54 567534
4 Billy Black 1683 9/23/44 335678
```

- ▶ 각 레코드는 필드 구분자(보통 화이트 스페이스)로 구분된 필드들로 구성
- ▶ NF : 필드 수 기록하는 내장변수

```
[kgu@lily ch11_awk]$ gawk '{print NR, $1, $2, $5, NF}' employees
1 Tom Jones 543354 5
2 Mary Adams 28765 5
3 Sally Chang 567534 5
4 Billy Black 335678 5
```

- ▶ 필드 구분자를 변경하고자 하는 경우 -F 옵션 사용 (둘 이상의 경우 [ ] 이용)

```
[kgu@lily ch11_awk]$ gawk -F: '/kgu/ {print NR, $1, NF}' /etc/passwd
52 kgu 7
[kgu@lily ch11_awk]$ gawk -F'[ :\t]' '/kgu/ {print NR, $1, NF}' /etc/passwd
52 kgu 7
```



# 패턴과 동작

- ▶ 패턴은 정규표현식이거나 참, 거짓의 결과를 갖는 조건 표현식 혹은 이 둘의 조합

```
[kgu@lily ch11_awk]$ gawk '/Tom/' employees
Tom Jones      4424      5/12/66      543354
[kgu@lily ch11_awk]$ gawk '$3 < 4000' employees
Sally Chang    1654      7/22/54      567534
Billy Black    1683      9/23/44      335678
```

- ▶ 동작은 중괄호{ }로 묶여있으며, ;으로 구분하거나 개행문자로 각 줄에 하나씩 둘 수 있음

```
pattern { action statement; action statement; etc. }
또는
pattern {
    action statement
    action statement
}
```

# 정규표현식 (1)

## ▶ awk 정규표현식 메타문자

메타	의미	메타	의미
^	행의 시작 지시자	[ABC]	[ ]사이의 문자 집합 중 하나에 대응
\$	행의 끝 지시자	[^ABC]	[ ] 사이의 문자집합에 속하지 않는 한 문자와 대응
.	하나의 문자와 대응	[A-Z]	A부터 Z 사이의 문자집합 중 하나에 대응
*	선행 문자와 같은 문자의 0개 혹은 임의 개수에 대응	A B	A나 B 중 하나에 대응
+	선행 문자와 같은, 하나 이상의 문자에 대응	(AB)+	하나 이상의 문자셋 AB와 대응
?	선행 문자와 같은, 0개 혹은 하나의 문자에 대응	W*	문자로서의 *
		&	검색열을 저장하여 치환열에서 사용

## ▶ 지원되지 않는 메타문자

- ▶  $W < > W$  : 단어지시자
- ▶  $W( W )$  : 역 참조
- ▶  $.W\{ W \}$  : 반복

# 정규표현식 (2)

## ▶ match 연산자 : ~

- ▶ 특정 레코드나 필드 내에서 일치하는 패턴이 존재하는지 검사

```
[kgu@lily ch11_awk]$ gawk '$1 ~ /[Bb]illy/' employees
Billy Black 1683 9/23/44 335678
[kgu@lily ch11_awk]$ gawk '$1 ~ /ly$/' employees
Sally Chang 1654 7/22/54 567534
Billy Black 1683 9/23/44 335678
```

## ▶ POSIX 문자 항목 (대괄호 문자 클래스)

[:alnum:]	알파벳과 숫자	[:graph:]	공백이 아닌 문자	[:space:]	공백문자
[:alpha:]	알파벳	[:lower:]	소문자	[:upper:]	대문자
[:cntrl:]	제어문자	[:print:]	[:graph:]와 유사하나 공백 포함	[:xdigit:]	16진수 [0-9a-fA-F]
[:digit:]	숫자	[:punct:]	구두점		

# 스크립트 파일 안의 awk 명령

## ▶ 예제

```
[kgu@lily ch11_awk]$ cat info
# Script name: info
# This is a comment.
/Tom/ {print "Tom's birthday is " $4}
/Mary/ {print NR, $0}
/^\^Sally/ {print "Hi, Sally, " $1 " has a salary of $" $5 "."}
# End of info
[kgu@lily ch11_awk]$ gawk -f info employees
Tom's birthday is 5/12/66
2 Mary Adams 5346          11/4/63          28765
Hi, Sally, Sally has a salary of $567534.
```

# 비교식

- ▶ 관계연산자
  - ▶ C 프로그래밍 참조
- ▶ 조건식

```
조건식1 ? : 구문2 : 구문3  
[kgu@lily ch11_awk]$ gawk '{max= ($1 > $2) ? $1: $2; print max}' filename
```

- ▶ 산술계산 (+, -, \*, /, %, ^ (지수))
- ▶ 논리연산자 (&&, ||, !)
- ▶ 범위 패턴

# 변수

## ▶ 초기화와 강제변환

- ▶ name="Nancy"
- ▶ x++
- ▶ number = 35
- ▶ name + 0 : 문자열을 숫자로 강제 변환
- ▶ number " " : 숫자를 문자열로 강제 변환

## ▶ 대입연산자 : =, +=, -=, \*=, /=, %=, ^=

- ▶ 예 : a ^= 5 (a = a^5)

## ▶ 내장 변수

- ▶ ARGV : 명령줄 매개변수 개수
- ▶ ARGV : 명령줄 매개변수 배열
- ▶ FILENAME : 현재 입력파일 이름
- ▶ FNR : 현재 파일의 레코드 수
- ▶ FS: 필드 구분자
- ▶ NF : 현재 레코드의 필드 개수
- ▶ NR : 지금까지 레코드 개수
- ▶ OFMT : 숫자를 위한 출력 서식
- ▶ OFS : 출력 필드 구분자
- ▶ ORS : 출력 레코드 구분자
- ▶ RLENGTH : match 함수가 찾은 일치하는 문자열 길이
- ▶ RS : 입력 레코드 구분자
- ▶ RSTART : match 가 찾은 문자열 위치
- ▶ SUBSEP : 배열 요소 구분자

```
[kgu@lily ch11_awk]$ gawk -F'[ :\t]' '$8 == "CA" { $8 = "California"; print}' datebook
Betty Boop 245-836-8357 635 Cutesy Lane, Hollywood, California 91464 6/23/23 14500
Tommy Savage 408-724-0140 1222 Oxbow Court, Sunnyvale, California 94087 5/19/66 34200
```

# BEGIN 패턴과 END 패턴

## ▶ BEGIN 패턴

- ▶ 입력 파일을 처리하기 전에 수행해야 하는 동작들 지정
- ▶ 일반적으로 OFS, RS, FS 등과 같은 내장변수 값을 변경하거나 사용자 정의변수 초기화, 헤더나 제목 출력에 이용

## ▶ END 패턴

- ▶ 모든 입력 줄 처리가 끝난 후 처리
- ▶ 처리한 레코드 수나 계산 결과 출력

# 입력 리다이렉션(getline)

- ▶ 현재 처리 중인 입력 파일이 아닌, 표준입력이나 파이프, 또는 다른 파일로부터 입력을 받기 위해 사용

```
[kgu@lily ch11_awk]$ gawk 'BEGIN {"date" | getline d; print d}' datebook
2013. 05. 20. (월) 10:56:23 KST
[kgu@lily ch11_awk]$ gawk 'BEGIN {"date" | getline d; split(d, darr); print darr[3]}' datebook
20.
```

```
[kgu@lily ch11_awk]$ cat awk_file2
BEGIN {printf "What is your name? "
        getline name < "/dev/tty"
}
$1 ~ name {print "Found " name " on line ", NR "."
          print $0
}
END {print "See ya, " name "."}
[kgu@lily ch11_awk]$ gawk -f awk_file2 /etc/passwd
What is your name? kgu
Found kgu on line 52.
kgu:x:1000:1000:kgu:/home/kgu:/bin/bash
See ya, kgu.
```

;가 아닌 새로운 줄로  
명령 구분 가능



# 실습 (awk를 포함한 모든 유틸리티 사용)

## ▶ don\_data

```
Mike Harrington:(510) 548-1278:250:100:175
Christian Dobbins:(408) 538-2358:155:90:201
Susan Dalsass:(206) 654-6279:250:60:50
Archie McNichol:(206) 548-1348:250:100:175
Jody Savage:(206) 548-1278:15:188:150
Guy Quigley:(916) 343-6410:250:100:175
Dan Savage:(406) 298-7744:450:300:275
Nancy McNeil:(206) 548-1278:250:80:75
John Goldenrod:(916) 348-4278:250:100:175
Chet Main:(510) 548-5258:50:95:135
Tom Savage:(408) 926-3456:250:168:200
Elizabeth Stachelin:(916) 440-1763:175:75:300
```

Name	Phone	1st	2nd	3rd	Sum
Mike Harrington	(510) 548-1278	250	100	175	525
Christian Dobbins	(408) 538-2358	155	90	201	446
Susan Dalsass	(206) 654-6279	250	60	50	360
Archie McNichol	(206) 548-1348	250	100	175	525
Jody Savage	(206) 548-1278	15	188	150	353
Guy Quigley	(916) 343-6410	250	100	175	525
Dan Savage	(406) 298-7744	450	300	275	1025
Nancy McNeil	(206) 548-1278	250	80	75	405
John Goldenrod	(916) 348-4278	250	100	175	525
Chet Main	(510) 548-5258	50	95	135	280
Tom Savage	(408) 926-3456	250	168	200	618
Elizabeth Stachelin	(916) 440-1763	175	75	300	550

1. 첫번째 달에 100달러 이상 기부한 사람의 이름과 첫번째 달 기부 금액을 출력
2. 두번째 달에 100달러 이하 기부한 사람의 이름과 전화번호를 출력
3. 3개월간 500달러 이상 기부한 사람의 이름 출력
4. 3개월간 평균 150달러 이상 기부한 사람 이름 출력
5. 레코드 번호와 이름, 3개월간 기부한 총금액 출력
6. 표 형식으로 칸을 맞추어 출력 (printf 사용)
7. 처음에 제목을 출력하고, 각 레코드 번호, 이름, 각 달의 기부금액, 3개월 합계를 표 형식으로 칸에 맞추어 출력, 마지막 줄에는 전체 레코드 수와 기부 총액을 출력
8. 3개월간 기부한 총금액을 기준으로 정렬하여 이름, 3개월간 기부한 총금액 출력

# [실습과제]

- ▶ 실습 각 단계 화면 캡처하여 pdf 파일로 정리하여 과제 제출 ([cms.mmu.ac.kr/bear](https://cms.mmu.ac.kr/bear))
- ▶ 제출기한 : 5월 26일 자정