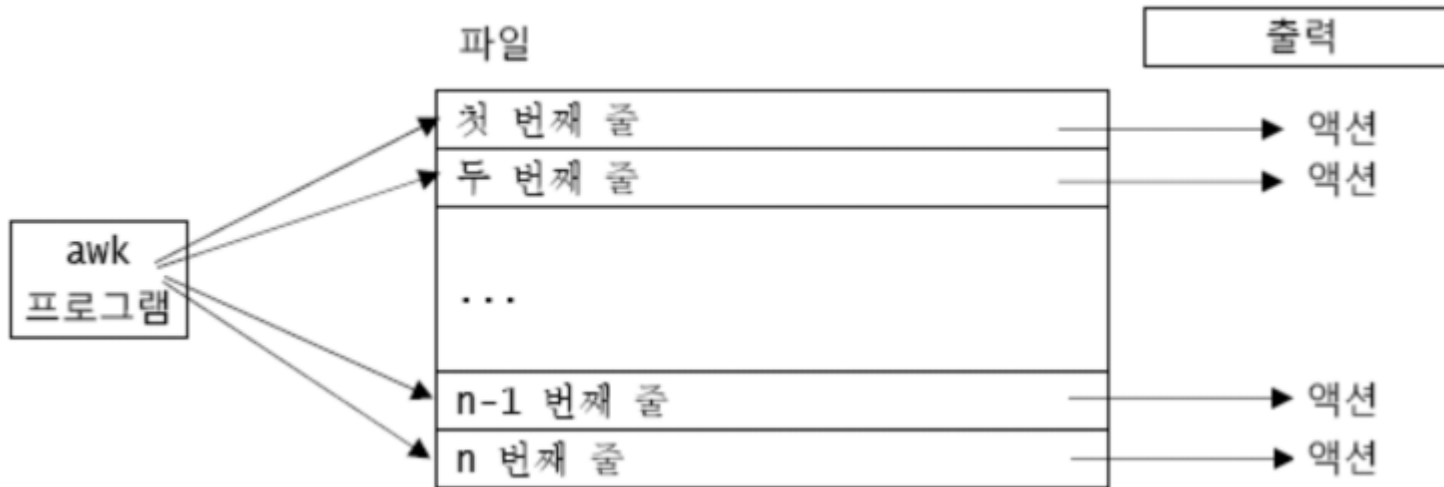


AWK

- 텍스트 파일을 대상으로 하여 각 줄을 필드들로 구분하고 이들을 awk 프로그램이 지시하는 대로 처리한다.
- 조건 { 액션 }
- 대상 파일의 각 줄을 스캔하여 조건을 만족하는 줄에 액션 수행



* AWK 프로그램 예제

스노우보드 you.txt 파일 다운로드해서 사용

* AWK 예제(1)

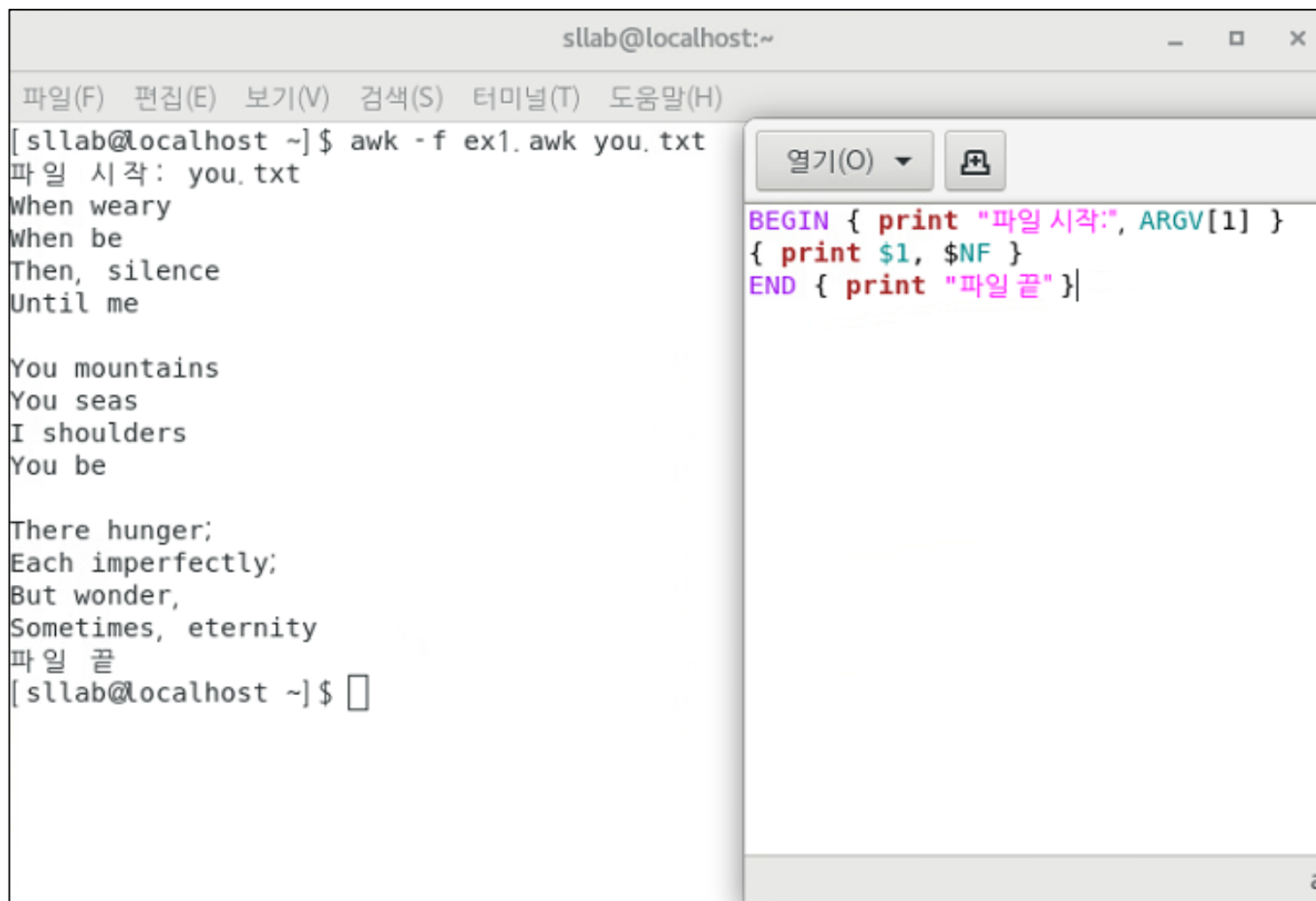
- ex1.awk 파일 작성
- 대상이 되는 텍스트 파일의 각 줄의 첫 번째 필드와 마지막 필드를 출력한다. 또한 파일 시작을 만나면 "파일 시작:"이라는 메시지와 파일 이름을 출력하고 파일 끝을 만나면 "파일 끝"이라는 메시지를 출력한다.

```
BEGIN { print "파일 시작:", ARGV[1] }  
{ print $1, $NF }  
END { print "파일 끝 " }
```

- ex1.awk 실행

```
$ awk -f ex1.awk you.txt
```

* AWK 예제(1)



```
sllab@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[sllab@localhost ~]$ awk -f ex1.awk you.txt  
파일 시작: you.txt  
When weary  
When be  
Then, silence  
Until me  
  
You mountains  
You seas  
I shoulders  
You be  
  
There hunger;  
Each imperfectly;  
But wonder,  
Sometimes, eternity  
파일 끝  
[sllab@localhost ~]$
```

```
BEGIN { print "파일 시작:", ARGV[1] }  
{ print $1, $NF }  
END { print "파일 끝" }
```

* AWK 예제(2)

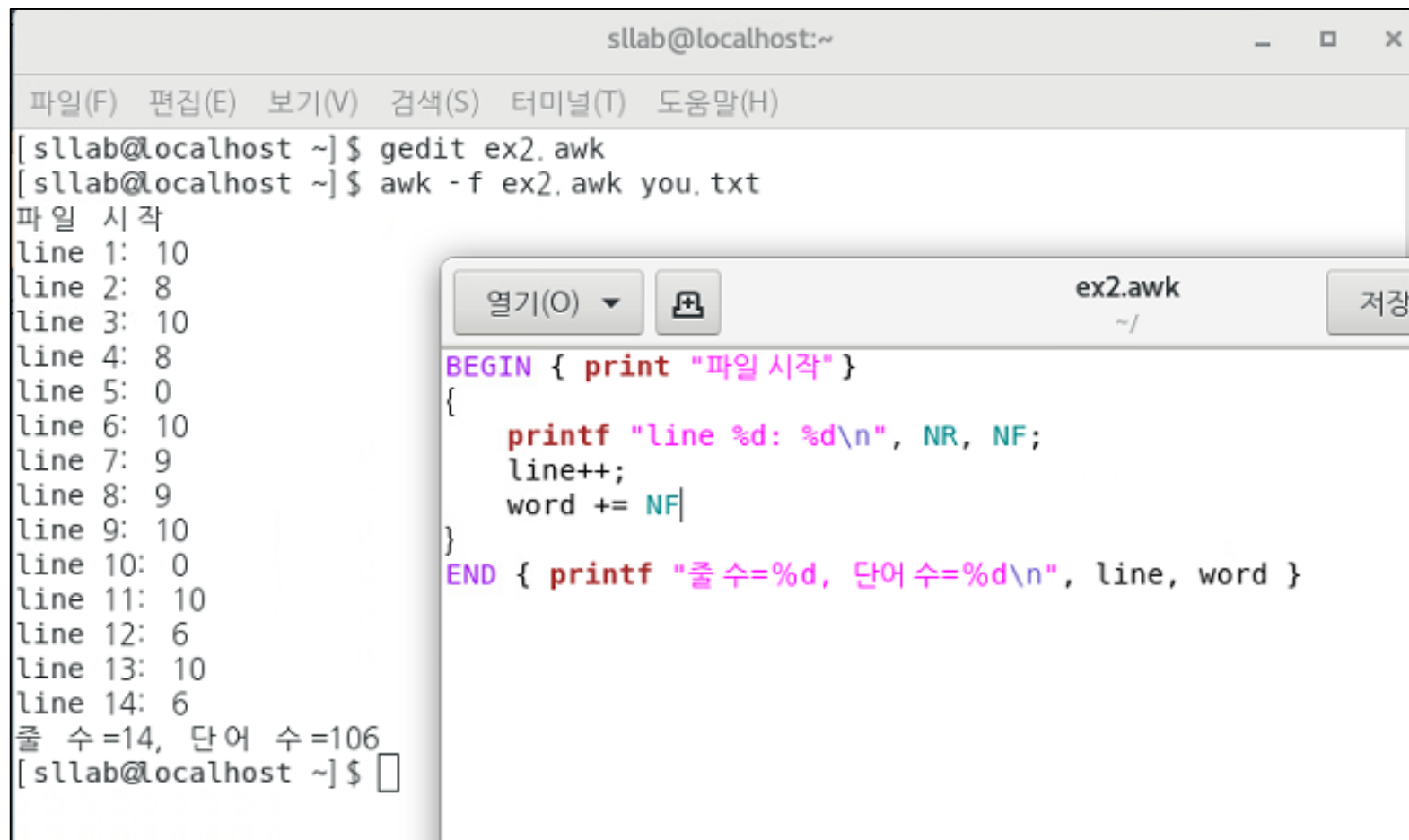
- ex2.awk 파일 작성
- 대상이 되는 텍스트 파일의 각 줄의 단어 수를 세서 출력하고 파일 끝에 도달하면 파일의 전체 줄 수와 단어 수를 출력한다. 여기서 사용한 변수 line과 word는 자동으로 정수형 변수로 선언되고 0으로 초기화된다.

```
BEGIN { print "파일 시작" }  
{  
    printf "line %d: %d\n", NR, NF;  
    line++;  
    word += NF  
}  
END { printf "줄 수 = %d, 단어 수 = %d\n", line, word}
```

- ex2.awk 실행

```
$ awk -f ex2.awk you.txt
```

* AWK 예제(2)



The screenshot shows a terminal window titled 'sllab@localhost:~' and a gedit editor window titled 'ex2.awk'.

Terminal Window:

```
[sllab@localhost ~]$ gedit ex2.awk
[sllab@localhost ~]$ awk -f ex2.awk you.txt
파일 시작
line 1: 10
line 2: 8
line 3: 10
line 4: 8
line 5: 0
line 6: 10
line 7: 9
line 8: 9
line 9: 10
line 10: 0
line 11: 10
line 12: 6
line 13: 10
line 14: 6
줄 수=14, 단어 수=106
[sllab@localhost ~]$
```

gedit Editor (ex2.awk):

```
BEGIN { print "파일 시작" }
{
    printf "line %d: %d\n", NR, NF;
    line++;
    word += NF
}
END { printf "줄 수=%d, 단어 수=%d\n", line, word }
```

* AWK 예제(3)

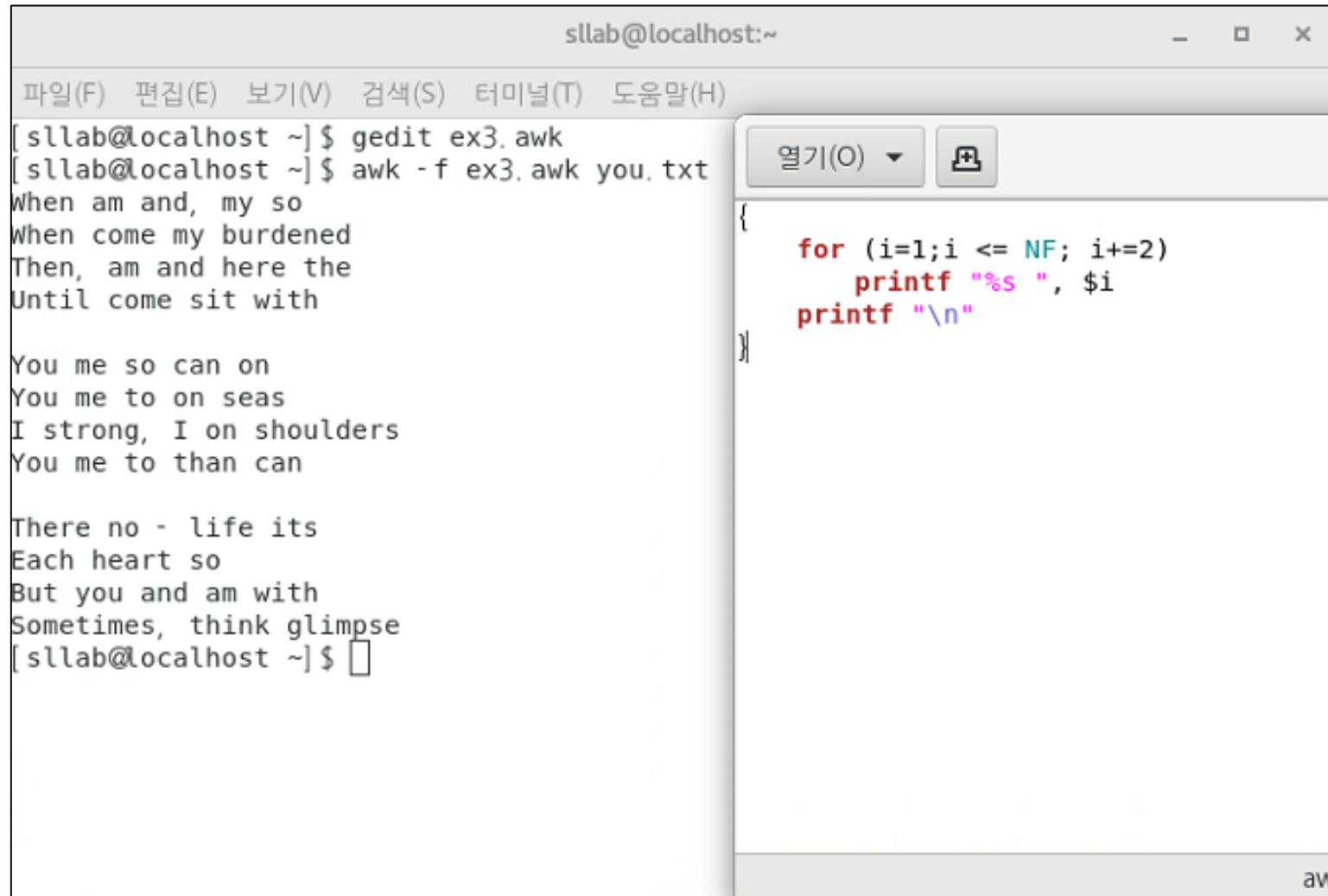
- ex3.awk 파일 작성
- 대상이 되는 텍스트 파일의 각 줄의 홀수 번째 필드(단어)들만 출력한다.

```
{  
    for (i=1; i<= NF; i += 2)  
        printf "%s ", $i  
    printf "\n"  
}
```

- ex3.awk 실행

```
$ awk -f ex3.awk you.txt
```

* AWK 예제(3)



```
sllab@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[sllab@localhost ~]$ gedit ex3.awk  
[sllab@localhost ~]$ awk -f ex3.awk you.txt  
When am and, my so  
When come my burdened  
Then, am and here the  
Until come sit with  
  
You me so can on  
You me to on seas  
I strong, I on shoulders  
You me to than can  
  
There no - life its  
Each heart so  
But you and am with  
Sometimes, think glimpse  
[sllab@localhost ~]$  
  
열기(O) [icon]  
{  
    for (i=1;i <= NF; i+=2)  
        printf "%s ", $i  
    printf "\n"  
}
```


* AWK 예제(4)

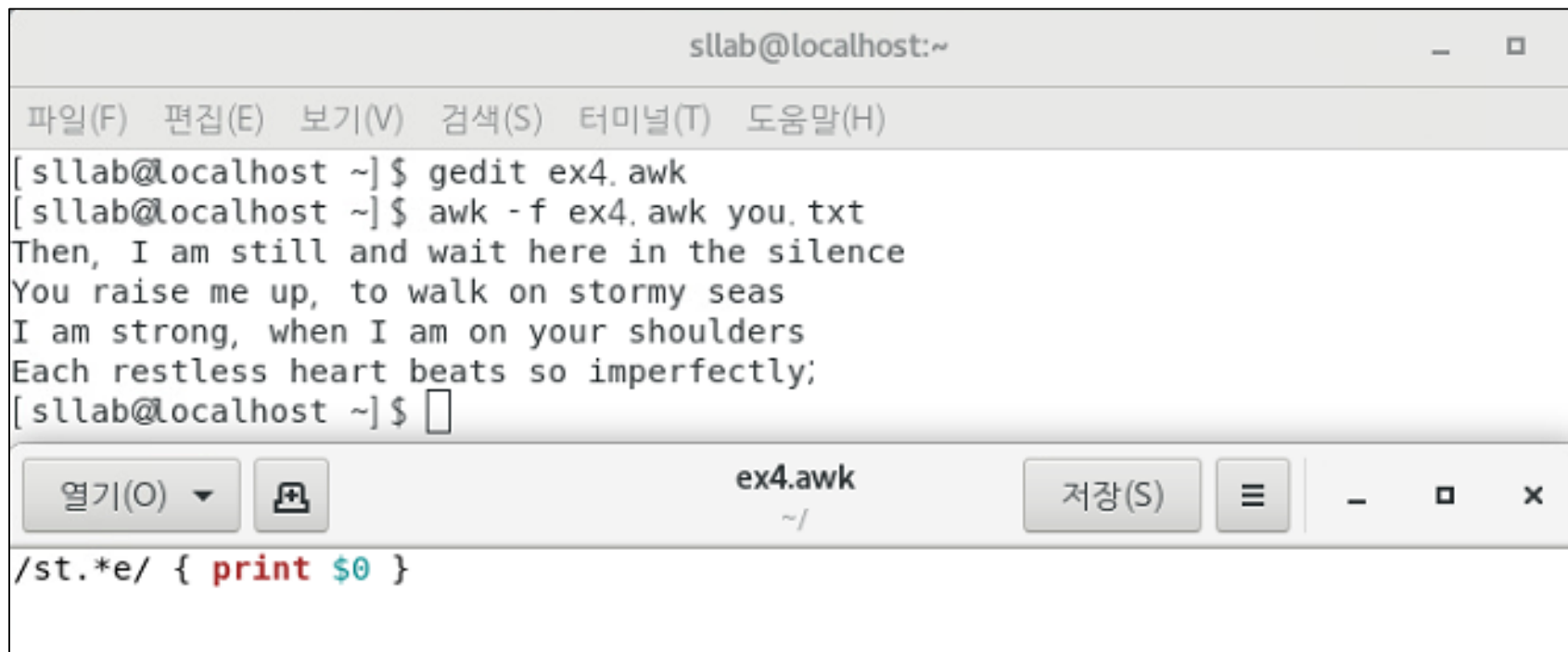
- ex4.awk 파일 작성
- 대상이 되는 텍스트 파일에서 문자열 패턴을 사용하여 st로 시작하여 e로 끝나는 문자열을 포함하는 모든 줄을 찾아 출력한다.

```
/st.*e/ {print $0 }
```

- ex4.awk 실행

```
$ awk -f ex4.awk you.txt
```

* AWK 예제(4)



The screenshot shows a terminal window titled 'sllab@localhost:~'. The window has a menu bar with '파일(F)', '편집(E)', '보기(V)', '검색(S)', '터미널(T)', and '도움말(H)'. The terminal content shows the following commands and output:

```
[sllab@localhost ~]$ gedit ex4.awk
[sllab@localhost ~]$ awk -f ex4.awk you.txt
Then, I am still and wait here in the silence
You raise me up, to walk on stormy seas
I am strong, when I am on your shoulders
Each restless heart beats so imperfectly;
[sllab@localhost ~]$
```

Below the terminal window, there is a window title bar for 'ex4.awk' with a file icon, a '열기(O)' button, a '저장(S)' button, and window control buttons. The content of the file is shown below the title bar:

```
/st.*e/ { print $0 }
```

* AWK 예제(5)

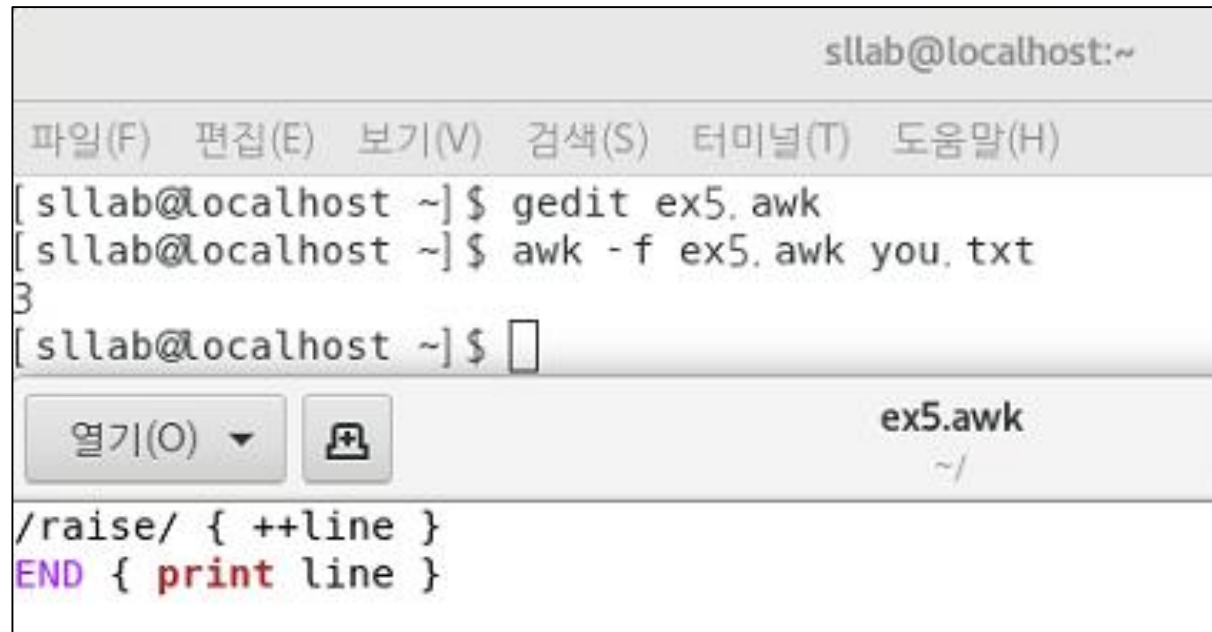
- ex5.awk 파일 작성
- raise라는 문자열을 포함하는 줄의 수를 계산하여 파일 끝을 만나면 출력한다. 여기서 사용한 변수 line은 자동으로 정수형 변수로 선언되고 0으로 초기화된다.

```
/raise/ { ++line }  
END { print line }
```

- ex5.awk 실행

```
$ awk -f ex5.awk you.txt
```

* AWK 예제(5)



The screenshot shows a terminal window titled 'sllab@localhost:~'. The menu bar includes '파일(F)', '편집(E)', '보기(V)', '검색(S)', '터미널(T)', and '도움말(H)'. The terminal history shows the user running 'gedit ex5.awk' and then 'awk -f ex5.awk you.txt'. The output of the command is '3'. Below the terminal, a snippet of the 'ex5.awk' script is shown: '/raise/ { ++line }' and 'END { print line }'.

```
sllab@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[sllab@localhost ~]$ gedit ex5.awk  
[sllab@localhost ~]$ awk -f ex5.awk you.txt  
3  
[sllab@localhost ~]$  
ex5.awk  
~/  
/raise/ { ++line }  
END { print line }
```

* AWK 예제(6)

- ex6.awk 파일 작성
- 대상이 되는 텍스트 파일에 대해서 각 단어 별 출현 빈도수를 계산하여 출력한다.
- 각 단어 별 출현 빈도수를 저장하기 위해서는 배열이 필요하다.
- 단어를 인덱스로 사용하는 연관 배열(associative array)를 사용한다.
- 파일 시작을 만나면 필드 구분자 (FS : Field Separator) 를 알파벳(a-zA-Z)이 아닌(^) 문자들로 설정한다.
- 파일의 각 줄의 각 필드에 대해 for 루프를 돌며 해당 필드를 tolower() 함수를 이용해 소문자로 변환하고 이를 연관 배열 words의 인덱스로 사용하여 해당 단어의 출현 횟수를 1씩 증가시킨다.
- 파일 끝을 만나면 연관 배열 words의 모든 인덱스(소문자로 변환된 단어) i에 대해서 for 루프를 돌며 인덱스와 그 출현 횟수를 출력한다.

* AWK 예제(6)

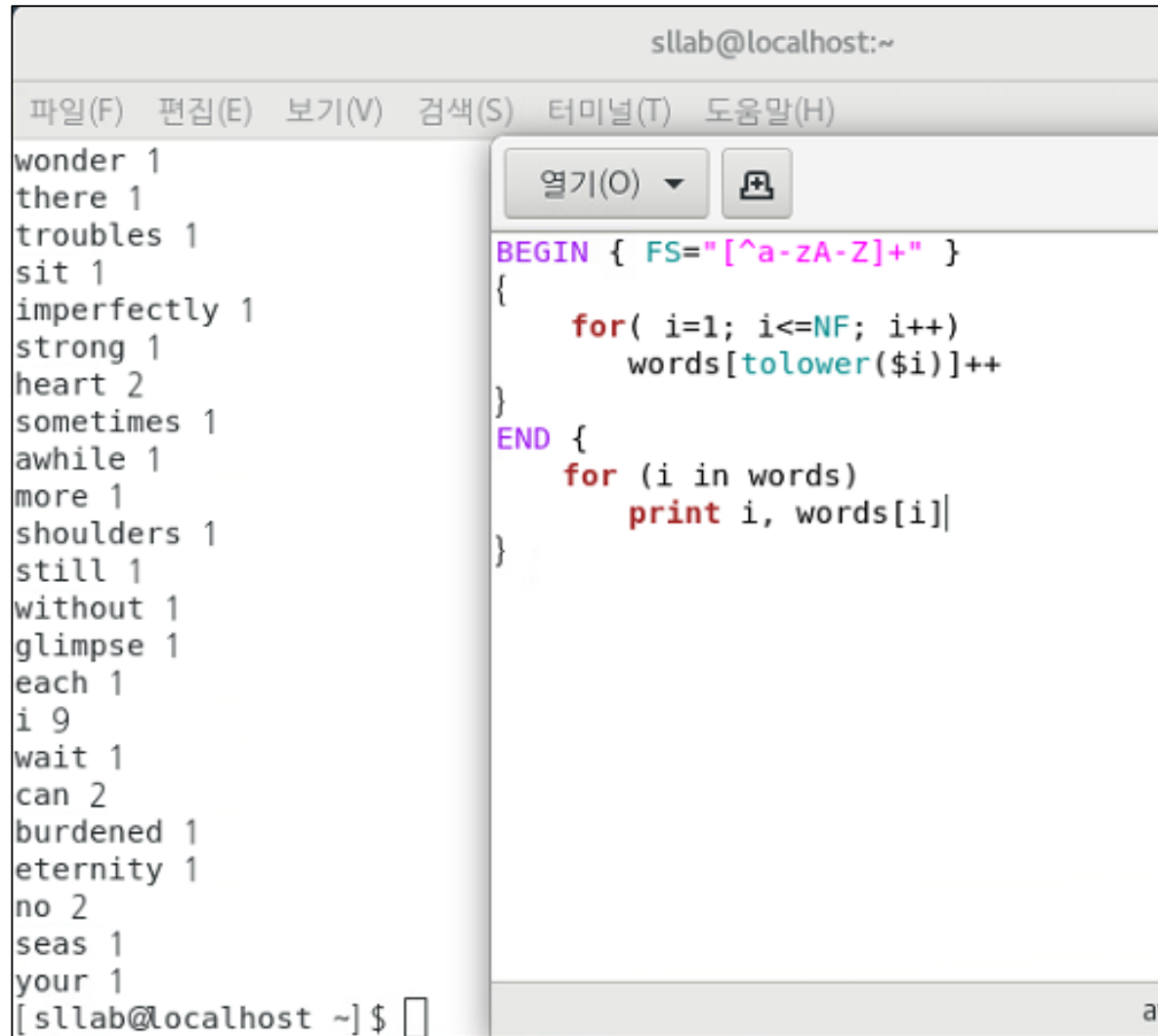
- ex6.awk 파일 작성
- 대상이 되는 텍스트 파일에 대해서 각 단어 별 출현 빈도수를 계산하여 출력한다.

```
BEGIN { FS="[^a-zA-Z]+" }  
{  
    for (i=1; i<=NF; i++)  
        words[tolower($i)]++  
}  
END {  
    for ( i in words)  
        print i, words[i]  
}
```

- ex6.awk 실행

```
$ awk -f ex6.awk you.txt
```

* AWK 예제(6)



```

sllab@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
열기(O) [icon]

wonder 1
there 1
troubles 1
sit 1
imperfectly 1
strong 1
heart 2
sometimes 1
awhile 1
more 1
shoulders 1
still 1
without 1
glimpse 1
each 1
i 9
wait 1
can 2
burdened 1
eternity 1
no 2
seas 1
your 1
[sllab@localhost ~]$

BEGIN { FS="[a-zA-Z]+" }
{
    for( i=1; i<=NF; i++)
        words[tolower($i)]++
}
END {
    for (i in words)
        print i, words[i]
}
```