

Lab 02

실습 주제: 줄단위 텍스트 출력

2021-1 시스템 프로그래밍(창병모 교수님)
TA: 오현주

줄단위 텍스트 출력

- 1) 명령줄 인수로 받은 텍스트 파일을 검사하여 사용자가 원하는 줄을 출력하는 프로그램을 작성하시오. 이 프로그램은 시작하면 프롬프트를 내주며 사용자가 원하는 줄 번호를 입력받아 해당 줄을 줄번호와 함께 출력한다.

줄 번호는 다음과 같은 다양한 형태로 입력할 수 있다.

n 한 줄 번호

n, ..., m 줄 번호 리스트


n-m 줄의 범위

* 모든 줄

※힌트※

먼저 파일 내의 각 줄의 시작 위치와 길이를 저장하는 테이블을 만들고 이를 이용해서 해당 줄을 찾아서 읽은 후에 출력한다.

줄단위 텍스트 출력 - 힌트1

연속된 바이트의 나열  File

30	75	102	156	
\n	\n	\n	\n	...

- table의 index → 줄의 번호
- table의 첫번째 column → 줄의 시작 위치
- table의 두번째 column → 줄의 길이

0	0	29
1	30	44
2	75	26
3	102	53
4	156	

index가 줄의 번호

줄의 시작 위치

줄의 길이

줄단위 텍스트 출력 - 힌트1

- read 시스템 호출

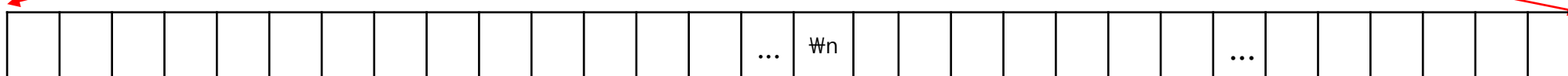
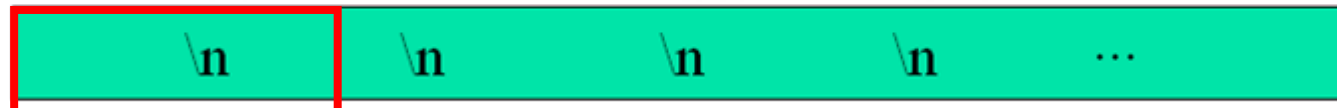
File 버퍼의 크기만큼 읽어들이м

30

75

102

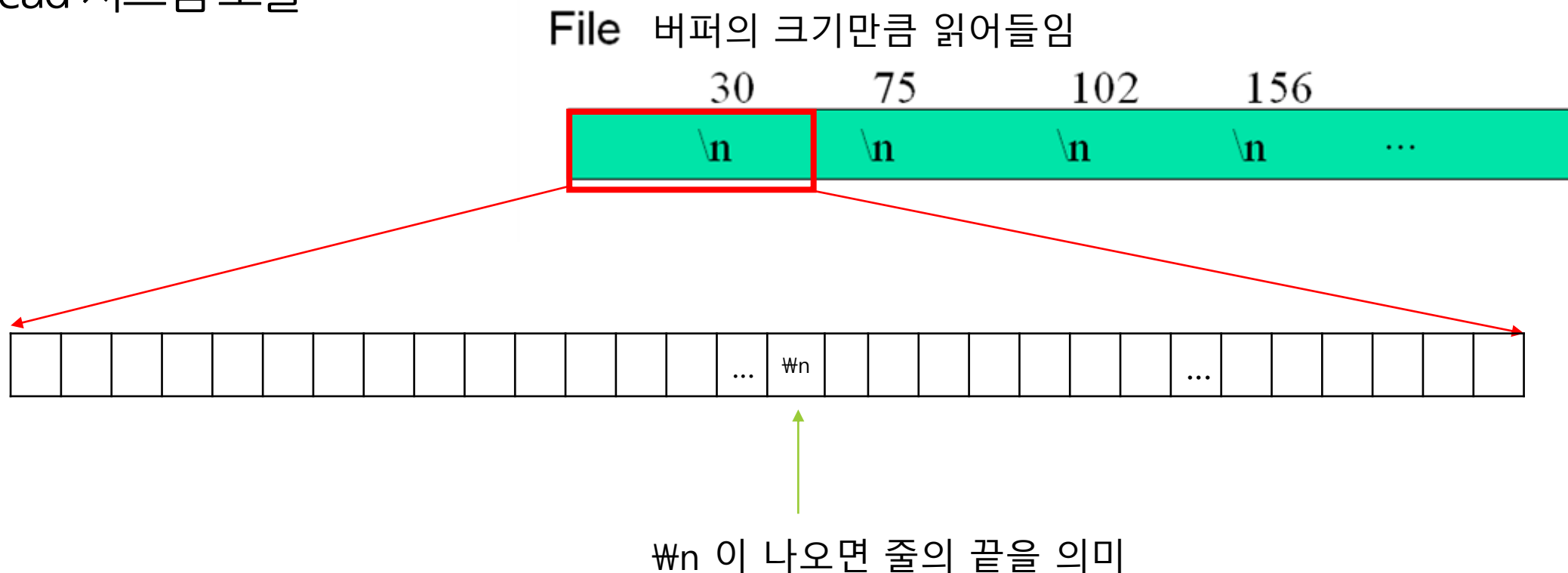
156



char 를 하나씩 비교

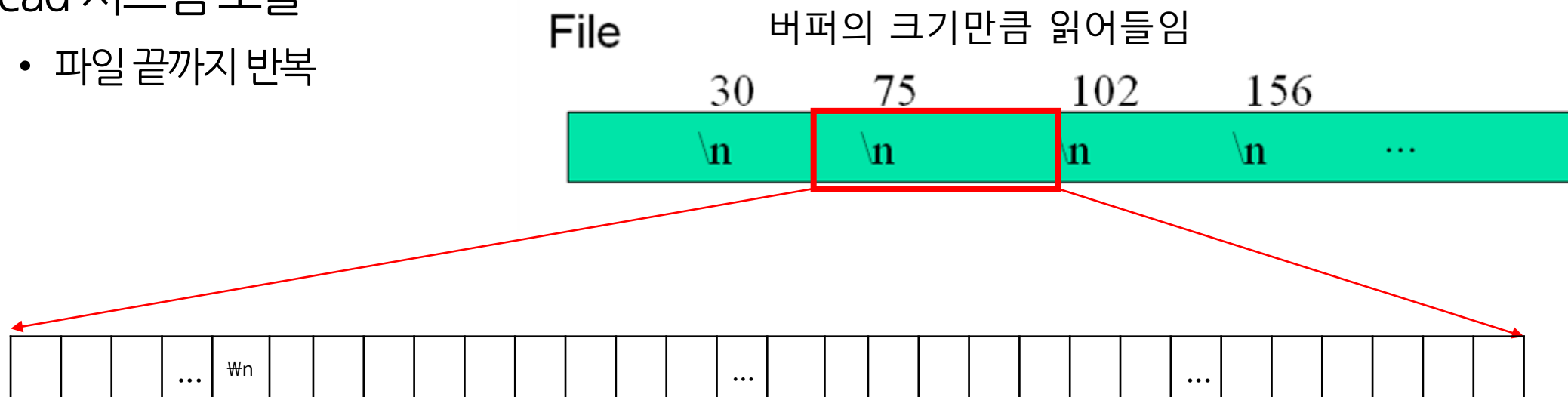
줄단위 텍스트 출력 - 힌트1

- read 시스템 호출

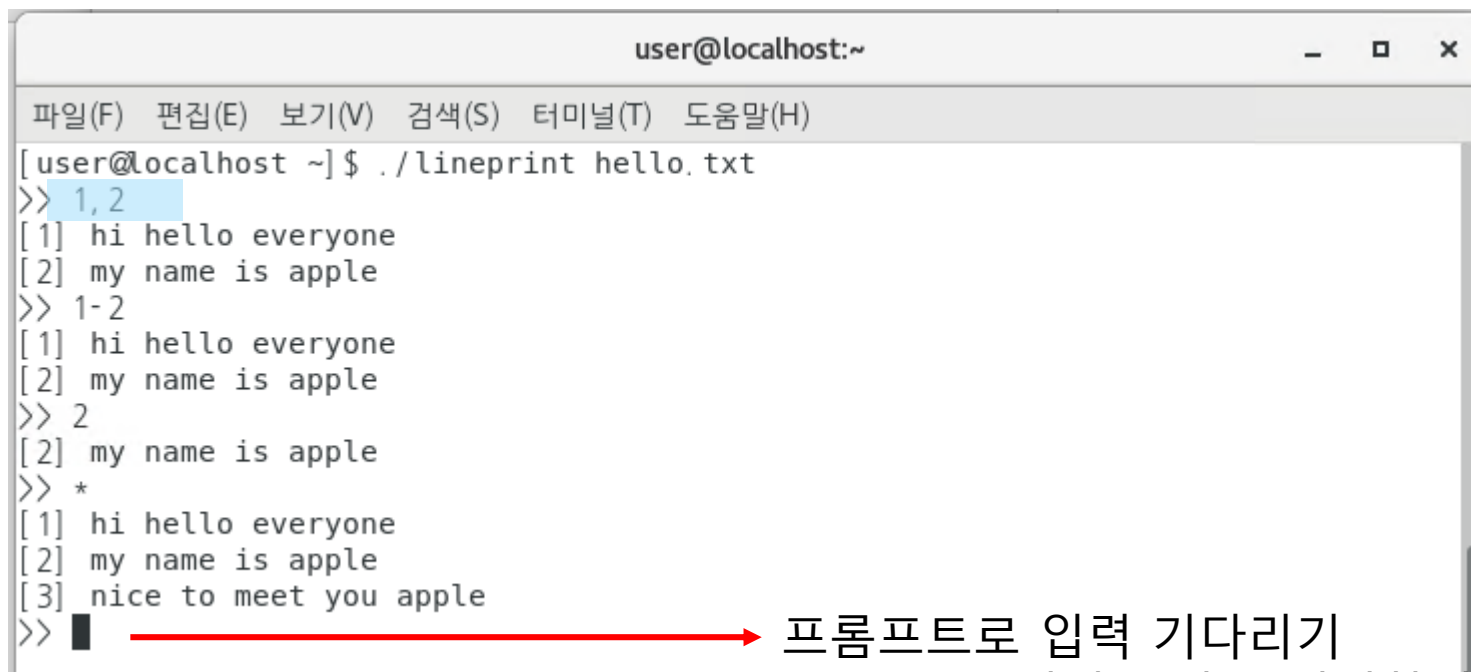


줄단위 텍스트 출력 - 힌트1

- read 시스템 호출
 - 파일 끝까지 반복



줄단위 텍스트 출력



```
user@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[user@localhost ~]$ ./lineprint hello.txt  
>> 1,2  
[1] hi hello everyone  
[2] my name is apple  
>> 1-2  
[1] hi hello everyone  
[2] my name is apple  
>> 2  
[2] my name is apple  
>> *  
[1] hi hello everyone  
[2] my name is apple  
[3] nice to meet you apple  
>> █
```

프롬프트로 입력 기다리기
- q(quit을 의미)문자를 입력하기 전까지 프롬프트로 계속 입력 받기

줄단위 텍스트 출력 - 힌트 2

```
user@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[user@localhost ~]$ ./lineprint hello.txt  
>> 1,2  
[1] hi hello everyone  
[2] my name is apple  
>> 1-2  
[1] hi hello everyone  
[2] my name is apple  
>> 2  
[2] my name is apple  
>> *  
[1] hi hello everyone  
[2] my name is apple  
[3] nice to meet you apple  
>>
```

프롬프트로 입력받는 부분

1	,	2	NULL	...			
---	---	---	------	-----	--	--	--

하나씩 읽어들이어서

- 한 줄 번호
- 줄 번호 리스트
- 줄의 범위
- 모든 줄

의 4가지 케이스 중 무엇인지 판별

줄단위 텍스트 출력 - 힌트 2

The screenshot shows a code editor with two tabs: 'hello.txt' and 'lineprint.c'. The 'hello.txt' tab is active and contains the following text:

```
hi hello everyone1
my name is apple1
nice to meet you apple1
hi hello everyone3
my name is apple3
nice to meet you apple3
hi hello everyone4
my name is apple4
nice to meet you apple4
hi hello everyone5
my name is apple5
nice to meet you apple5
hi hello everyone6
my name is apple6
nice to meet you apple6
hi hello everyone7
my name is apple7
nice to meet you apple7
```

The 'lineprint.c' tab is also visible. Below the code editor, a terminal window titled 'user@localhost:~' shows the execution of the program:

```
[user@localhost ~]$ ./lineprint hello.txt
>> 10-11
[10] hi hello everyone5
[11] my name is apple5
>> 2-9
[2] my name is apple1
[3] nice to meet you apple1
[4] hi hello everyone3
[5] my name is apple3
[6] nice to meet you apple3
[7] hi hello everyone4
[8] my name is apple4
[9] nice to meet you apple4
>> 1, 10, 12
[1] hi hello everyone1
[10] hi hello everyone5
[12] nice to meet you apple5
>> *
[1] hi hello everyone1
[2] my name is apple1
[3] nice to meet you apple1
[4] hi hello everyone3
[5] my name is apple3
[6] nice to meet you apple3
```

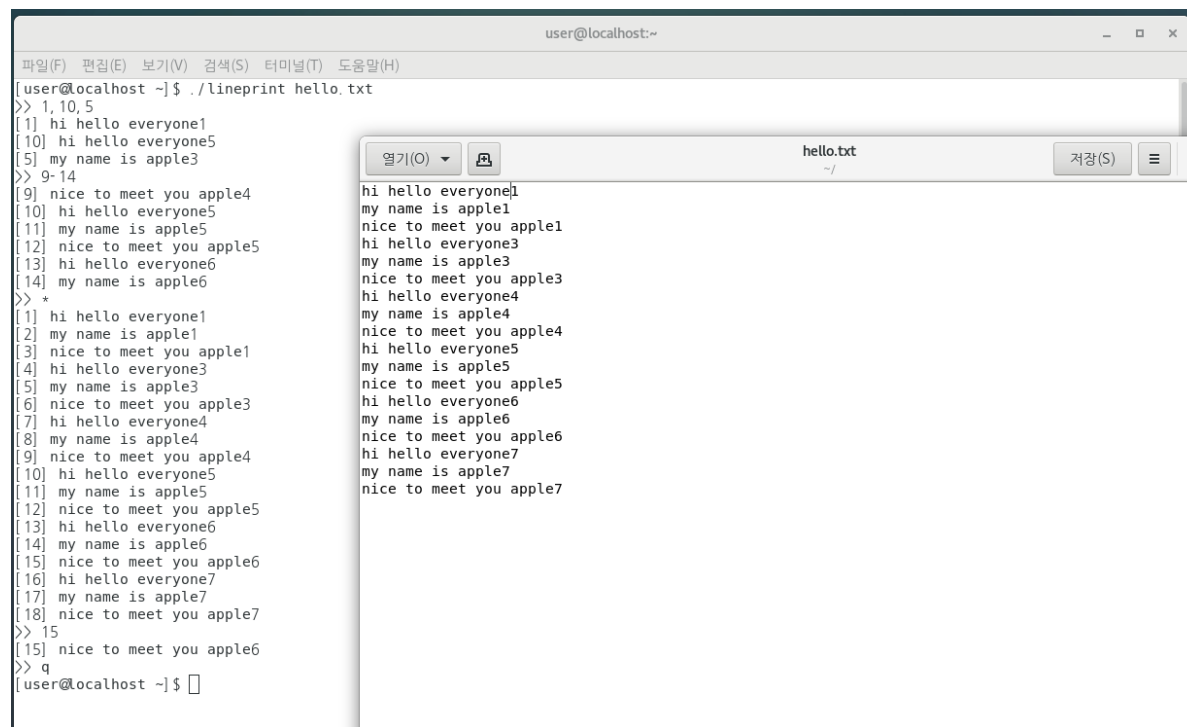
프롬프트로 입력받는 부분

1	0	-	1	1	NULL	...	
---	---	---	---	---	------	-----	--

어떻게 프롬프트로 받은 두 자리 수 이상의 수를 줄번호로 인식할지 고려해서 프로그램 작성하기

줄단위 텍스트 출력

- 반드시 파일 관련 시스템 호출을 이용하여 프로그래밍을 해주시기 바랍니다.
 - 파일 열기, 읽기 등의 파일 관련 모든 활동들은 시스템 호출만 사용해야함
- q를 입력해서 프로그램 종료하기 전까지, 프롬프트로 계속 입력을 받아주시기 바랍니다.



The image shows a terminal window and a text editor window. The terminal window, titled 'user@localhost:~', shows the execution of a program that reads from a file 'hello.txt' and prints its contents line by line. The program uses system calls to open the file, read it line by line, and print each line. The text editor window, titled 'hello.txt', shows the contents of the file being read, which consists of several lines of text including greetings and names.

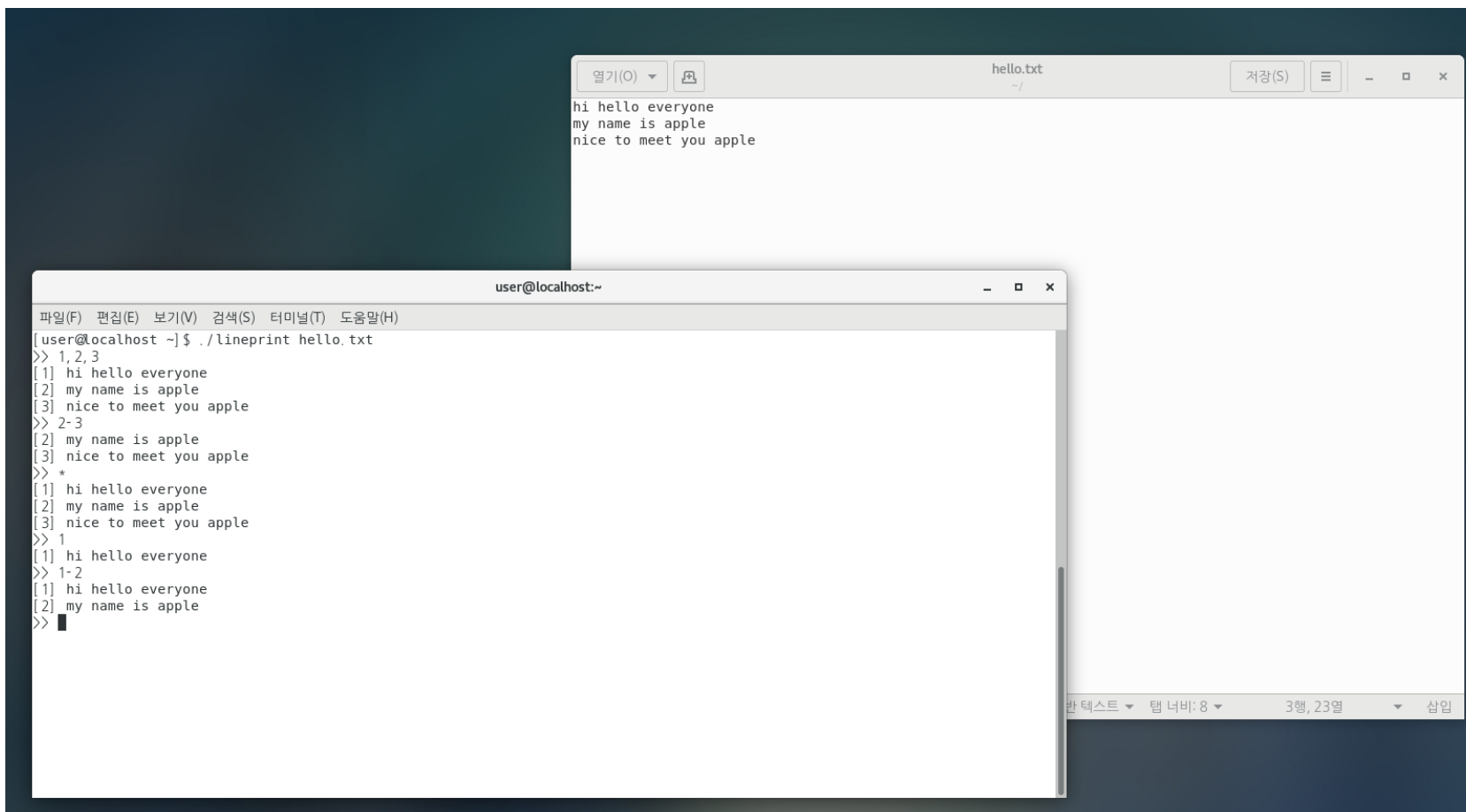
```
[user@localhost ~]$ ./lineprint hello.txt
>> 1,10,5
[1] hi hello everyone1
[10] hi hello everyone5
[5] my name is apple3
>> 9-14
[9] nice to meet you apple4
[10] hi hello everyone5
[11] my name is apple5
[12] nice to meet you apple5
[13] hi hello everyone6
[14] my name is apple6
>> *
[1] hi hello everyone1
[2] my name is apple1
[3] nice to meet you apple1
[4] hi hello everyone3
[5] my name is apple3
[6] nice to meet you apple3
[7] hi hello everyone4
[8] my name is apple4
[9] nice to meet you apple4
[10] hi hello everyone5
[11] my name is apple5
[12] nice to meet you apple5
[13] hi hello everyone6
[14] my name is apple6
[15] nice to meet you apple6
[16] hi hello everyone7
[17] my name is apple7
[18] nice to meet you apple7
>> 15
[15] nice to meet you apple6
>> q
[user@localhost ~]$
```

캡처 예시 1

```
user@localhost:~$ gcc -o lineprint lineprint.c
lineprint.c: In function 'main':
lineprint.c:42:11: warning: comparison between pointer and integer [enabled by default]
    while(c!=NULL){
           ^
lineprint.c:83:10: warning: comparison between pointer and integer [enabled by default]
    while(c!=NULL){
           ^
lineprint.c:138:2: warning: incompatible implicit declaration of built-in function 'exit' [enabled by default]
    exit(0);
    ^
user@localhost ~]$ ./lineprint hello.txt
>> 2-3
[2] my name is apple
[3] nice to meet you apple
>> 3
[3] nice to meet you apple
>> 1
[1] hi hello everyone
>> *
[1] hi hello everyone
[2] my name is apple
[3] nice to meet you apple
>> 2,3
[2] my name is apple
[3] nice to meet you apple
>> ^C
user@localhost ~]$
```

- 캡처에 보여져야하는 내용**
- 명령줄 인수로 넣어준 파일 내용(ex. hello.txt)
 - 프롬프트로 총 4가지를 입력받은 결과를 캡처해야함
 - 1) 한 줄 번호
 - 2) 줄 번호 리스트
 - 3) 줄의 범위
 - 4) 모든 줄

캡처 예시 2



```
user@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[user@localhost ~]$ ./lineprint hello.txt  
>> 1,2,3  
[1] hi hello everyone  
[2] my name is apple  
[3] nice to meet you apple  
>> 2-3  
[2] my name is apple  
[3] nice to meet you apple  
>> *  
[1] hi hello everyone  
[2] my name is apple  
[3] nice to meet you apple  
>> 1  
[1] hi hello everyone  
>> 1-2  
[1] hi hello everyone  
[2] my name is apple  
>> █
```

보고서 구성 및 제출 방법

- 보고서 내용
 - 매주 실습 문제가 적힌 보고서 양식이 제공될 예정입니다.
 - 각 문제별로
 - 캡처 결과
 - 문제 해결 방법: 어떻게 문제를 해결했는지, 프로그램 설계 내용 등
 - 소스코드: 텍스트 형식으로 넣기(캡처형식으로 넣으면 안됩니다.), 소스코드가 있는 경우에만 넣으시면 됩니다.
- 넣어주시기 바랍니다.
- 작성한 보고서는 스노우보드 Lab02에 업로드해 주시기 바랍니다.
- 제출 기한 넘어서 제출한 경우
 - 하루 delay 될 때마다 1점씩 감점