

System Software Experiment 1

Lecture 7 – I/O

spring 2020

Hwansoo Han (hhan@skku.edu)

Advanced Research on Compilers and Systems, ARCS LAB

Sungkyunkwan University

<http://arcs.skku.edu/>

printf

```
int i = -123;  
char c = 'a';  
float f = 0.52841;
```

Character	Printed as...	Example
%c	Character (문자)	A
%d, %i	Decimal integer	-123
%u	Unsigned decimal integer	4294967173
%o	Unsigned octal integer	37777777605
%x, %X	Unsigned hexadecimal integer	ffffff85, FFFFFFF85
%e, %E	Floating point number with e or E	5.284100e-01, 5.284100E-01
%f	Floating point number (precision default : 6)	0.528410
%g, %G	Shorter format between %f and %e	0.52841
%s	String (문자열) (hexadecimal)	
%p	Pointer address	0x00000000ffffff85
%%	Print "%"	

printf

```
char str[] = "Blue moon!"
```

Character	Printed as...	Example
%12s	Field width 12, right-adjusted	"__Blue moon!"
%.7s	Precision 7	"Blue mo"
%-11.8s	Precision 8, Field width 11, left-adjusted	"Blue moo__"
%05d	Padded with zeroes	00123
%-#9x	0x prefix is attached to hexadecimal number	0x7b_____
%10.5f	Precision 5, Field width 10	"__0.12346"
%-12.5e	Precision 5, Field width 12, e-format	"1.23457e-01_"

scanf

- `int scanf(const char * format, void * input);`
- Treats `\t`, `\n`, `\r`, ' ' equally as whitespaces. Ignores them by default.
- `scanf("%[^\n]", str);` Reads until `\n` is met. Reads including white spaces.
"Hello, World!\n"
`scanf("%s", str);` -> "Hello\0"
`scanf("[^\n]", str);` -> "Hello, World!\0"

fopen, fclose

- FILE * fopen (const char *filename, const char * mode);
- Returns File pointer

Mode Character	Function
r	Open for read-only
w	Create for writing, if exists, overwrite.
a	Open for appending (Write from the end-of-file)
r+	Open for update
w+	Open for update (overwrite if already exists)
a+	Open for appending
rb	Open for read-only. Treat input as binary.

- int fclose(FILE * stream);
- Return 0 if succeeded.

File pointer

Input.txt

PEN PINEAPPLE
APPLE PEN

- `FILE *fp = fopen("./input.txt", "r");`

fp
↓

Input.txt

P	E	N	'	'	P	I	N	E	A	P	P	L	E	\n	A	P	P	L
---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	---	---	---	---

offset

0

4

8

12

16

Pointing location can be changed via various functions.
(`fscanf`, `fgetc`, `fgets`, `fseek`, `fread`....)

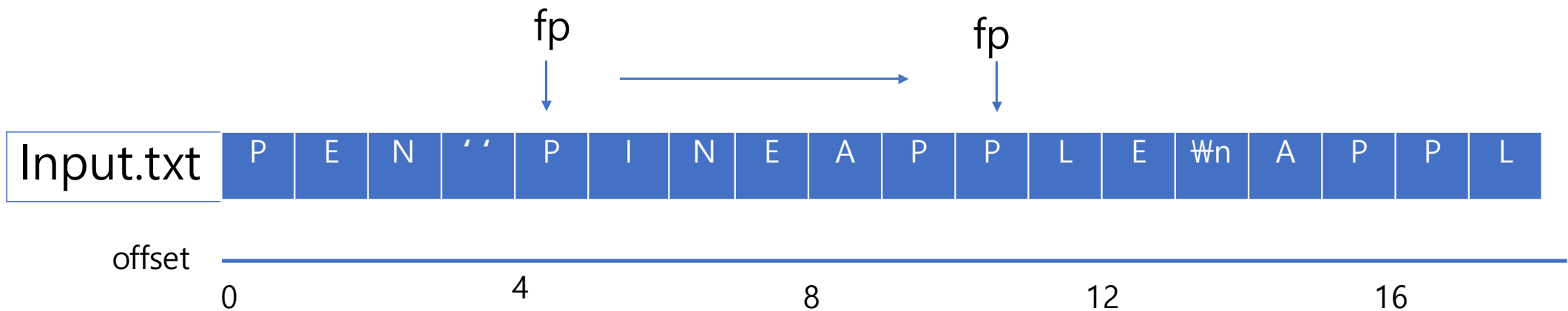
File seek: fseek

- `fseek(fp, 6, SEEK_CUR)`

SEEK_CUR : From the current offset
SEEK_SET : From the beginning of the file
SEEK_END : From the end of the file

Input.txt

PEN PINEAPPLE
APPLE PEN

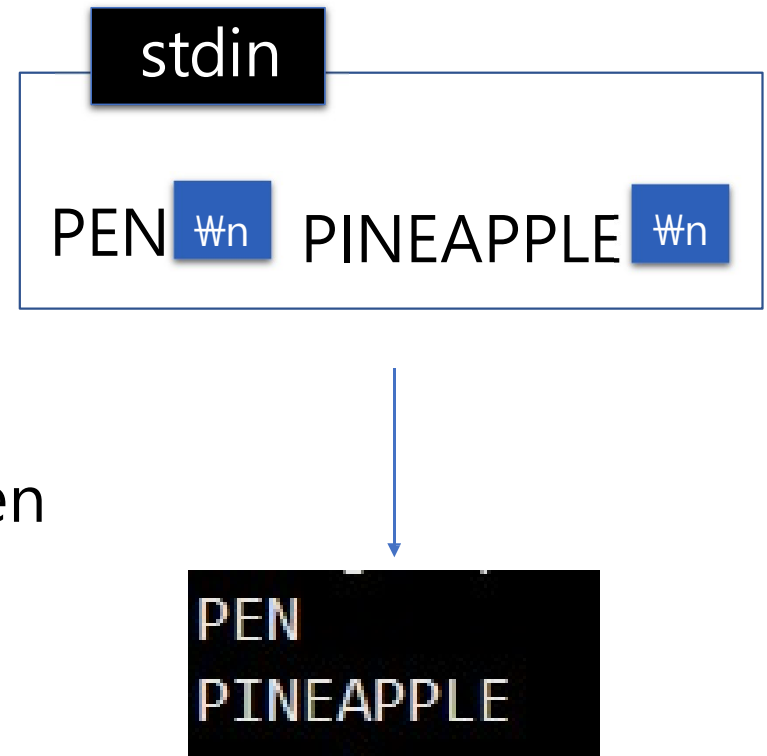


More Functions ...

- `ftell(fp)` : return current offset of `fp`.
- `rewind(fp)` : rewind `fp` back to the beginning.

stdin, stdout, stderr

- Standard stream input/output
- Defined in stdio.h
- stdin : connected to the keyboard
- stdout, stderr : connected to the screen



fgetc, fputc, fgets, fputs

- For a character

```
int fputc (int chr, FILE *stream);
```

```
int fgetc (FILE *stream);
```

- For a string

```
int fputs(const char *str, FILE *stream);
```

```
char *fgets(char *str, int size, FILE *stream);
```

No need do specify the format

Return	If failed
character (ASCII)	EOF
character (ASCII)	EOF

Return	If failed
Positive number	EOF
str	null (str not changed)

* at the end of file

fprintf, fscanf

- In stdio.h

```
int fprintf(FILE *ofp, const char *format, ...);
```

```
int fscanf(FILE *ifp, const char *format, ...);
```

Return	If failed
The number of printed bytes	-1
The number of data the function read	EOF

```
#include <stdio.h>
int main(){
    FILE * fp;
    char str[20];
    fp = fopen("test.txt", "w+");
    fprintf(fp, "PEN PINEAPPLE\nAPPLE PEN");
    rewind(fp);
    fscanf(fp, "%s", str);
    printf("%s\n", str);
    fclose(fp);
}
```

```
1 PEN PINEAPPLE
2 APPLE PEN
~
"test.txt" 2 lines --50%--
```

```
root@compute21:/home/seokha/SSE# ./fprintf
PEN
```

`fprintf(stdout, "...");` = `printf("...");`
`fscanf(stdin, ...);` = `scanf(...);`

fread, fwrite

- `size_t fread (void * ptr, size_t size, size_t count, FILE * stream);`
- `size_t fwrite (const void * ptr, size_t size, size_t count, FILE * stream);`
- Returns successfully conducted number of elements
Error if `count != return`, or EOF
- Not interrupted by whitespace characters.

fgets vs fscanf

PEN PINEAPPLE
APPLE PEN

fgets(str[0], 100, fp);

P	E	N	'	'	P	I	N	E	A	P	P	L	E	\n	
---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	--

fgets(str[1], 100, fp);

A	P	P	L	E	'	'	P	E	N	\n					
---	---	---	---	---	---	---	---	---	---	----	--	--	--	--	--

Includes whitespace characters.

Stops at \n

fscanf(fp, "%s", str);

P	E	N													
---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--

fscanf(fp, "%s", str);

P	I	N	E	A	P	P	L	E							
---	---	---	---	---	---	---	---	---	--	--	--	--	--	--	--

Omit whitespace characters

Stop at whitespace characters

fgets vs fscanf

PEN PINEAPPLE
APPLE PEN

```
fscanf(fp, "%s", str);  
fgets(str[1], 15, fp);
```

P	E	N												
'	'	P	I	N	E	A	P	P	L	E	\n			

```
PEN  
PINEAPPLE  
  
root@compute21:/home/seokha/SSE#
```

```
fgets(str[1], 15, fp);  
fscanf(fp, "%s", str);
```

P	E	N	'	'	P	I	N	E	A	P	P	L	E	\n
A	P	P	L	E										

```
PEN PINEAPPLE  
  
APPLE  
root@compute21:/home/seokha/SSE#
```

fgets vs fscanf

Wn
' 'PEN PINEAPPLEWn
APPLE PENWn

```
fscanf(fp, "%s", str[0]);
```

[illegible]

```
fscanf(fp, "%s", str[1]);
```

P	I	N	E	A	P	P	L	E						
---	---	---	---	---	---	---	---	---	--	--	--	--	--	--

```
fgets(str[2], 100, fp);
```

[illegible]

```
fscanf(fp, "%s", str[1]);
```

[illegible]

fgets vs fscanf

Wn
' 'PEN PINEAPPLEWn
APPLE PENWn

```
fgets(str[0], 15, fp);
```

[illegible]

```
fgets(str[1], 15, fp);
```

'	P	E	N	'	P	I	N	E	A	P	P	L	E	Wn
---	---	---	---	---	---	---	---	---	---	---	---	---	---	----

```
fscanf(fp, "%s", str[2]);
```

[illegible]

```
fscanf(fp, "%s", str[1]);
```

[illegible]

fgets vs fscanf

Wn
' 'PEN PINEAPPLEWn
APPLE PENWn

```
fgets(str[0], 15, fp);
```

```
fgets(str[1], 15, fp);
```

```
fscanf(fp, "%[^\\n]", str[2]);
```

```
fscanf(fp, "%s", str[1]);
```

```
fscanf(fp, "%s", str[0]);
```

```
fgets(str[1], 15, fp);
```

```
fscanf(fp, "%s", str[2]);
```

```
fgets(str[3], 15, fp);
```

[illegible]

EOF, feof

- How to recognize we have reached the end of file?
- EOF : End Of File
- fgets returns 0 when they meet the end.

```
while(fgets(str, sizeof(str), fp) != 0){  
    .....  
} // do something until the program reads the end of file.
```


EOF, feof

- `feof(FILE * stream);`
- Returns 1 if fp exceeds end of file (`!= EOF`), 0 if not.

```
while(!feof(fp)){  
    fgets(str, sizeof(str), fp);  
    printf("%s", str);  
}
```

- What happens?

✓ Make a program adding all the numbers in the file.



```
5 30 131  
2524 2215  
930 348
```

```
root@compute21:/home/seokha/SSE# ./add  
Total : 6183
```

File access

✓ Make a program double spacing a file.

Everywhere that I go, everywhere that I be
If you are not surrounding me with your energy
I don't wanna be there, don't wanna be anywhere
Any place that I can't feel you, I just wanna be near you
And yes, I'm a mess but I'm blessed
To be stuck with you
Sometimes it gets unhealthy
We can't be by ourselves
We'll always need each other, and
Yes I'm a mess but I'm blessed
To be stuck with you
I just want you to know that
If I could swear I'll go back
Make everything all better

~~~~~

Everywhere that I go, everywhere that I be  
If you are not surrounding me with your energy  
I don't wanna be there, don't wanna be anywhere  
Any place that I can't feel you, I just wanna be near you  
And yes, I'm a mess but I'm blessed  
To be stuck with you  
Sometimes it gets unhealthy  
We can't be by ourselves  
We'll always need each other, and  
Yes I'm a mess but I'm blessed  
To be stuck with you  
I just want you to know that  
If I could swear I'll go back  
Make everything all better

# Exercise

- Read the file “mytext.txt”
  - Change every occurrence of “like” to “love”  
Change every occurrence of “want” to “need”
  - Write results to file.
  - Be careful to keep everything else untouched, including new lines (\n) and spaces (‘ ’)
- Submit your screenshots of 1. Your code 2. The output text file on iCampus.  
(You don’t need to zip the files nor write a report on docx, etc. Upload just the photo files)



mytext.txt

# Summary!

```
scanf("%[^\n]", str);
```

```
FILE * file = fopen("mytext.txt", "w");  
fclose(file);
```

```
char c;
```

```
char str[100];
```

```
fputc('a', file)
```

```
c = fgetc(file)
```

```
fputs(str, file)
```

```
fgets(str, 100, file) \n까지 읽되, \n을 str에 포함
```

```
fprintf(file, "string: %s", str) fscanf(file, "%[^\n]", str) \n까지 읽되, \n 포함X
```

```
fwrite(str, sizeof(char), 100, file) fread(str, sizeof(char), 100, file)
```

```
+ fgets, fscanf 가 동작 후 파일 포인터를 어디서 멈추게 하는지
```

```
while(!feof(file))
```