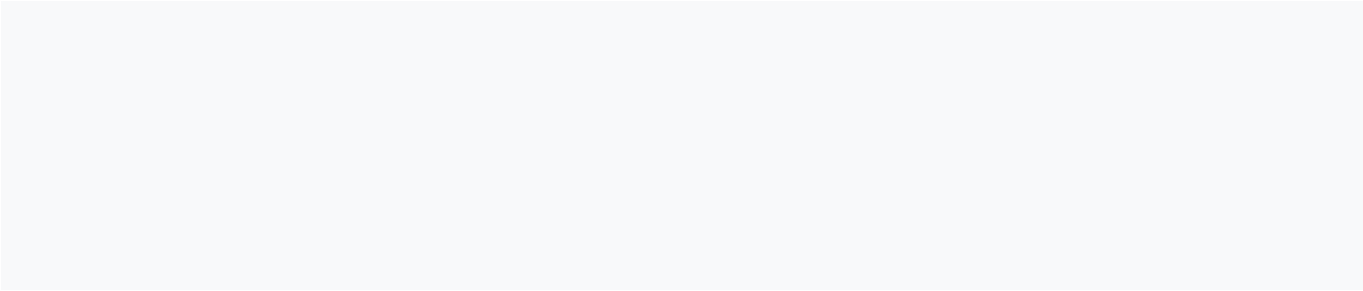


(C) : Duet Debug : Deep Dive
: printf / scanf -

형식 문자열 · 인자 처리 방식 · 스트림 버퍼링 이해



:	
:	
:	10 104
:	10 115
:	2026. 07. 14.



- 1.
- 2.
 - 2-1. C
 - 2-2. printf
 - 2-3. scanf
 - 2-4. printf scanf
 - 2-5.
- 3.
 - 3-1. 1.....
 - 1. printf
 - 2. scanf
 - 3. stdin
 - 4. getchar()
 - 5. scanf fgets
 - 6. stdout
 - 3-2. 2.....
- 4.
- 5.
 - 5-1.
 - 5-2.
 - 5-3. AI

1.

C scanf printf .

printf ,

.

scanf %c , %s . “

” ,

printf scanf , , stdin-stdout .

2.

2-1. C

C `<stdio.h>` , FILE
stdin, stdout, stderr
FILE . printf scanf , stdout stdin
FILE .
stdout
. stderr .

2-2. printf

printf format , %
%d 10 , %f , %s (\0)
printf format . , . %
d ,
printf ,
rintf(user_input) % , printf("%s", user_input) p
.

2-3. scanf

scanf , & %d
, %c . %d %c
scanf & %s &
scanf . 0 , EOF
%s , char str[10]
%9s .

2-4. printf scanf

	printf	scanf
	→ → stdout	stdin → →
	,	%s, ,

2-5.

%[][.][]
, f , c , s , p
printf . - , + , 0 0
%08.2f 8 0
.

```

        . h hh          , l ll long long long
        . %d int, %lld long long
scanf
        . %s
        . 20 %19s
        . , ,

```

3.

3-1. 1

C printf scanf , (format string), (stdin), (stdout)

1. printf

```

printf %d, %f, %c, %s
, printf , , , printf
, % printf
printf

```

2. scanf

```

scanf("%d", &num)
scanf 1 , 0 scanf
scanf , printf
scanf

```

3. stdin

```

%c
%c , ASCII 10 . ASCII 10 (\n)
Enter 123\n , %d 123 %c
" " \n stdin

```

4. getchar()

```

while(getchar()!='\n');

```

%C

stdin

5. scanf fgets

scanf("%s") fgets()

scanf("%s") "Hello World" "Hello"

fgets() "Hello World"

fgets()

6. stdout

printf("Hello");

fflush(stdout) "Hello" , "Done!"

fflush(stdout) "HelloDone!"

stdout , fflush(stdout)

printf scanf , printf , scanf

stdin , %d %c

stdout , fflush(stdout)

C

3-2. 2

```
#include <stdio.h>
```

```
#ifdef _WIN32
```

```
#include <windows.h>
```

```
#define SLEEP(ms) Sleep(ms)
```

```
#else
```

```
#include <unistd.h>
```

```
#define SLEEP(ms) usleep((ms) * 1000)
```

```
#endif
```

```
int main(void)
```

```
{
```

```
    int num;
```

```
    char ch;
```

```

char str[100];

/* ===== */
printf("==== Experiment 1 : printf Format String =====\n");

printf("%d -> %d\n", 100);
printf("%f -> %.2f\n", 3.14159);
printf("%c -> %c\n", 'A');
printf("%s -> %s\n", "Hello");

/* ===== */
printf("\n==== Experiment 2 : scanf =====\n");

printf("Enter an integer : ");

int ret = scanf("%d", &num);

printf("scanf return value = %d\n", ret);
printf("Stored integer = %d\n", num);

/* ===== */
printf("\n==== Experiment 3 : stdin Buffer =====\n");

printf("Reading one character using %c...\n");

scanf("%c", &ch);

printf("ASCII Code : %d\n", ch);

if(ch=='\n')
    printf("Result : scanf read the ENTER key (newline).\n");
else
    printf("Result : scanf read '%c'.\n", ch);

/*      */
while(getchar()!='\n');

/* ===== */
printf("\n==== Experiment 4 : Correct Character Input =====\n");

printf("Enter one character : ");

scanf("%c",&ch);

printf("You entered : %c\n", ch);

while(getchar()!='\n');

/* ===== */
printf("\n==== Experiment 5 : fgets =====\n");

```

```

printf("Enter a sentence : ");

fgets(str,sizeof(str),stdin);

printf("Result : %s",str);

/* ===== */
printf("\n===== Experiment 6 : stdout Buffer =====\n");

printf("Printing \"Hello\"...");
fflush(stdout);

SLEEP(3000);

printf("Done!\n");

printf("If fflush(stdout) is removed,");
printf(" some environments display the output only after the newline.\n");

return 0;
}

printf scanf ,stdin stdout .

printf %d,%f,%c,%s . printf , , ,
printf ,

scanf("%d", &num) . 1 ,
0 scanf scanf

%d %c . %c , ASCII
10 (\n) . Enter (stdin) ,%d

while(getchar()!='\n'); %c

fgets() .
scanf("%s") fgets() ,

"Hello" . fflush(stdout)
"Hello" , "Done!" . stdout ,
fflush(stdout)

printf scanf , stdin stdout

```

4.

```
printf scanf
(format string) printf
scanf
%d %c
getchar()
stdout fflush(stdout)
GNU C Library scanf scanf __vscanf_internal()
C
```

5.

5-1.

1. [printf, fprintf, sprintf, snprintf, printf_s, fprintf_s, sprintf_s, snprintf_s - cppreference.com](#)
2. [scanf, fscanf, sscanf, scanf_s, fscanf_s, sscanf_s - cppreference.com](#)
3. [What Is Buffering? | How Does Buffering Work? | Akamai](#)
4. [stdio.c source code \[glibc/libio/stdio.c\] - Codebrowser](#)
5. [stdio.h source code \[glibc/libio/stdio.h\] - Codebrowser](#)
6. [fscanf.c source code \[glibc/stdio-common/fscanf.c\] - Codebrowser](#)
7. [fprintf.c source code \[glibc/stdio-common/fprintf.c\] - Codebrowser](#)

5-2.

1. [printf, fprintf, sprintf, snprintf, printf_s, fprintf_s, sprintf_s, snprintf_s - cppreference.com](#)
2. [scanf, fscanf, sscanf, scanf_s, fscanf_s, sscanf_s - cppreference.com](#)
3. [fscanf.c source code \[glibc/stdio-common/fscanf.c\] - Codebrowser](#)
4. [fprintf.c source code \[glibc/stdio-common/fprintf.c\] - Codebrowser](#)

5-3. AI

AI C .

printf scanf , (format string), (variable arguments), (stdin, stdout)

AI . GNU C Library(glibc) scanf

__vfscanf_internal() .

AI . printf , scanf , stdin

, stdout fflush(stdout) .

AI

. GNU C Library , Linux (man page), C AI ,

.

AI , , , .