Input/Output in C

Course: Introduction to Programming and Data Structures

Laltu Sardar

Institute for Advancing Intelligence (IAI),
TCG Centres for Research and Education in Science and Technology (TCG Crest)

tcg crest
Inventing Harmonious Future

September 6, 2024

1. Input/output in C
   - Input Functions in C
     - `scanf()`
     - `fscanf()`
     - `sscanf()`
     - `getchar()`
     - `fgetc()`
     - `fgets()`
     - `getch()`
     - `ungetc()`
   - Output functions in C
     - `printf()`
     - `fprintf()`
     - `sprintf()`
     - `putchar()`
     - `putc()`
     - `puts()`
     - `fputs()`
   - Behavior w.r.t. White-spaces

tcg crest
Inventing Harmonious Future

Input functions in C

Input/Output Functions in C

- `scanf()` - Reads formatted input from the terminal.
- `fscanf()` - Reads formatted input from a file stream.
- `sscanf()` - Reads formatted input from a string.
- `getchar()` - Reads a single character from stdin.
- `fgetc()` - Reads a single character from a file stream.
- `fgets()` - Reads a string or line from stdin or a file stream.
- `getch()` - Reads a single character from the keyboard without echoing.
- `ungetc()` - Pushes a character back onto an input stream.

tcg crest
Inventing Harmonious Future

scanf() Function

- Reads formatted input from the standard input (stdin) i.e. terminal/console.
- Syntax: `scanf(const char *format, ...);`
- Format specifiers define the type of data to read (e.g., %d for integers, %s for strings).
- It requires the address of the variable where input should be stored.
- **Returns** the number of inputs successfully read.

tcg crest
Inventing Harmonious Future

Example: scanf() Function

```c
#include <stdio.h>

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    printf("You entered: %d\n", num);
    return 0;
}
```

fscanf() Function

- Reads formatted input from a file or stream.
- Syntax: `fscanf(FILE *stream, const char *format, ...);`
- Similar to `scanf()`, but the input source is a file.
- The first argument is a pointer to the file stream.
- **Returns** the number of inputs successfully read.

tcg crest
Inventing Harmonious Future

Example: fscanf() Function

```c
#include <stdio.h>

int main() {
    FILE *file = fopen("input.txt", "r");
    int num;
    fscanf(file, "%d", &num);
    printf("Read from file: %d\n", num);
    fclose(file);
    return 0;
}
```

sscanf() Function

- Reads formatted input from a string.
- Syntax: `sscanf(const char *str, const char *format, ...);`
- Similar to `scanf()`, but the input source is a string.
- The first argument is the string from which the input is read.
- **Returns** the number of fields that were successfully converted and assigned

tcg crest
Inventing Harmonious Future

Example: sscanf() Function

```c
#include <stdio.h>

int main() {
    char input[] = "123 456";
    int a, b;
    sscanf(input, "%d %d", &a, &b);
    printf("Read values: %d and %d\n", a, b);
    return 0;
}
```

tcg crest
Inventing Harmonious Future

getchar() Function

- Reads a single character from stdin.
- Syntax: `int getchar(void);`
- Waits for user input until a character is entered.
- **Returns** the ASCII value of the character.

tcg crest
Inventing Harmonious Future

Example: getchar() Function

```c
#include <stdio.h>

int main() {
    char ch;
    printf("Enter a character: ");
    ch = getchar();
    printf("You entered: %c\n", ch);
    return 0;
}
```

fgetc() Function

- Reads a single character from a file stream.
- Syntax: `int fgetc(FILE *stream);`
- Similar to `getchar()`, but reads from a file.
- **Returns** the character read, or `EOF` on error or end of file.

Example: fgetc() Function

```
1  #include <stdio.h>
2
3  int main() {
4      FILE *file = fopen("input.txt", "r");
5      char ch;
6      ch = fgetc(file);
7      printf("First character: %c\n", ch);
8      fclose(file);
9      return 0;
10 }
```

fgets() Function

- Reads a line or string from a file or stdin.
- Syntax: `char *fgets(char *str, int n, FILE *stream);`
- Reads at most `n-1` characters and stores them in `str`.
- Stops on newline or when the limit is reached.
- **Returns** a pointer to the string buffer if successful. A NULL return value indicates an error or an end-of-file condition.

tcg crest
Inventing Harmonious Future

Example: fgets() Function

```c
#include <stdio.h>

int main() {
    char str[100];
    printf("Enter a string: ");
    fgets(str, 100, stdin);
    printf("You entered: %s\n", str);
    return 0;
}
```

getch() Function (Non-standard)

- Reads a single character from the keyboard without echoing.
- Available in some environments like Windows, but not standard in C.
- Syntax: `int getch(void);`
- **Returns** the ASCII value of the character read as an integer.

tcg crest
Inventing Harmonious Future

Example: getch() Function

```c
#include <conio.h> // For Windows

int main() {
    char ch;
    printf("Press any key to continue...\n");
    ch = getch();
    printf("You pressed: %c\n", ch);
    return 0;
}
```

ungetc() Function

- Pushes a character back onto an input stream.
- Syntax: `int ungetc(int ch, FILE *stream);`
- The pushed character will be read by the next read operation.
- Can only push back one character.
- **Returns** the integer argument c converted to an unsigned char, or EOF if c cannot be pushed back.

tcg crest
Inventing Harmonious Future

Example: ungetc() Function

```c
#include <stdio.h>

int main() {
    char ch;
    ch = getchar();
    ungetc(ch, stdin);
    printf("Pushed back: %c\n", getchar());
    return 0;
}
```

Output functions in C

Output Functions in C

- `printf()` - Prints formatted output to the terminal.
- `fprintf()` - Prints formatted output to a file stream.
- `sprintf()` - Prints formatted output to a string.
- `putchar()` - Prints a single character to the terminal.
- `fputc()` - Prints a single character to a file stream.
- `puts()` - Prints a string followed by a newline.
- `fputs()` - Prints a string to a file stream.

tcg crest
Inventing Harmonious Future

printf() Function

- Prints formatted output to the standard output (stdout).
- Syntax: `printf(const char *format, ...);`
- Format specifiers define the type of output (e.g., %d for integers, %s for strings).
- **Returns** the number of characters printed.

tcg crest
Inventing Harmonious Future

Example: printf() Function

```c
#include <stdio.h>

int main() {
    int num = 42;
    printf("The number is: %d\n", num);
    return 0;
}
```

tcg crest
Inventing Harmonious Future

fprintf() Function

- Prints formatted output to a file stream.
- Syntax: `fprintf(FILE *stream, const char *format, ...);`
- Similar to `printf()`, but the output is written to a file.
- The first argument is a pointer to the file stream.
- **Returns** the number of characters printed.

tcg crest
Inventing Harmonious Future

Example: fprintf() Function

```c
#include <stdio.h>

int main() {
    FILE *file = fopen("output.txt", "w");
    int num = 42;
    fprintf(file, "The number is: %d\n", num);
    fclose(file);
    return 0;
}
```

tcg crest
Inventing Harmonious Future

sprintf() Function

- Prints formatted output to a string.
- Syntax: `sprintf(char *str, const char *format, ...);`
- Similar to `printf()`, but the output is stored in a string.
- The first argument is the buffer where the output will be stored.
- **Returns** the number of characters printed.

tcg crest
Inventing Harmonious Future

Example: sprintf() Function

```c
#include <stdio.h>

int main() {
    char buffer[50];
    int num = 42;
    sprintf(buffer, "The number is: %d", num);
    printf("%s\n", buffer);  // Output the string
    return 0;
}
```

putchar() Function

- Prints a single character to the terminal (stdout).
- Syntax: `int putchar(int ch);`
- The argument is the character to be printed.
- **Returns** the character printed as an unsigned char, or EOF on error.

Example: putchar() Function

```c
#include <stdio.h>

int main() {
    char ch = 'A';
    putchar(ch);  // Prints: A
    putchar('\n');  // Newline
    return 0;
}
```

tcg crest
Inventing Harmonious Future

fputc() Function

- Prints a single character to a file stream.
- Syntax: `int fputc(int ch, FILE *stream);`
- The first argument is the character to print, and the second is the file stream.
- **Returns** the character printed, or EOF on error.

tcg crest
Inventing Harmonious Future

Example: fputc() Function

```c
#include <stdio.h>

int main() {
    FILE *file = fopen("output.txt", "w");
    char ch = 'A';
    fputc(ch, file);  // Writes 'A' to the file
    fclose(file);
    return 0;
}
```

puts() Function

- Prints a string followed by a newline to stdout.
- Syntax: `int puts(const char *str);`
- The argument is the string to print.
- Automatically appends a newline character at the end.
- **Returns** a non-negative value on success, or EOF on error.

tcg crest
Inventing Harmonious Future

Example: puts() Function

```c
#include <stdio.h>

int main() {
    puts("Hello, World!");  // Prints: Hello, World!
    return 0;
}
```

fputs() Function

- Prints a string to a file stream.
- Syntax: `int fputs(const char *str, FILE *stream);`
- Similar to `puts()`, but doesn't append a newline.
- The first argument is the string, and the second is the file stream.
- **Returns** a non-negative value on success, or EOF on error.

tcg crest
Inventing Harmonious Future

Example: fputs() Function

```c
#include <stdio.h>

int main() {
    FILE *file = fopen("output.txt", "w");
    fputs("Hello, File!\n", file);  // Writes "Hello, File
        !" to the file
    fclose(file);
    return 0;
}
```

Behavior w.r.t. White-spaces

Whitespace Handling	First Function	Second Function
scanf vs fscanf	■ Skips leading whitespace. ■ Stops on the first whitespace unless using %s, which stops at spaces or newlines.	■ Similar to `scanf`: skips leading whitespace. ■ Stops at whitespace for %s.
printf vs fprintf	■ Whitespace in the format string is printed exactly as written.	■ Whitespace is printed as it appears in the format string.

tcg crest
Inventing Harmonious Future

Comparison

Whitespace Handling	First Function	Second Function
gets vs fgets	■ Reads until a newline or EOF is encountered. ■ Retains all whitespaces within the input.	■ Reads whitespaces, including spaces, tabs, and newlines, until a newline or `n-1` characters are read.
puts vs fputs	■ Prints the string as is, including all whitespaces. ■ Automatically appends a newline at the end.	■ Prints the string exactly as provided, including whitespaces. ■ Does not append a newline.

Comparison

Whitespace Handling	First Function	Second Function
getchar vs fgetc	■ Reads any character, including whitespaces. ■ Whitespace characters are treated as valid input.	■ Reads any character, including whitespaces from a file. ■ Whitespace characters are valid input.
putchar vs fputc	■ Prints any character, including whitespace.	■ Prints any character, including whitespace, to the file stream.

tcg crest
Inventing Harmonious Future

Thank You

for your attention.

Questions?

tcg crest

Inventing Harmonious Future

Laltu Sardar

laltu {dot} sardar [at]tcgcrest(.)org

https://laltu-sardar.github.io