

std::ungetc

Defined in header <cstdio>

```
int ungetc( int ch, std::FILE *stream );
```

If `ch` does not equal EOF, pushes the character `ch` (reinterpreted as `unsigned char`) into the input buffer associated with the stream `stream` in such a manner that subsequent read operation from `stream` will retrieve that character. The external device associated with the stream is not modified.

Stream repositioning operations `std::fseek`, `std::fsetpos`, and `std::rewind` discard the effects of `ungetc`.

If `ungetc` is called more than once without an intervening read or repositioning, it may fail (in other words, a pushback buffer of size 1 is guaranteed, but any larger buffer is implementation-defined). If multiple successful `ungetc` were performed, read operations retrieve the pushed-back characters in reverse order of `ungetc`.

If `ch` equals EOF, the operation fails and the stream is not affected.

A successful call to `ungetc` clears the end of file status flag `std::feof`.

A successful call to `ungetc` on a binary stream decrements the stream position indicator by one (the behavior is indeterminate if the stream position indicator was zero).

A successful call to `ungetc` on a text stream modifies the stream position indicator in unspecified manner but guarantees that after all pushed-back characters are retrieved with a read operation, the stream position indicator is equal to its value before `ungetc`.

Parameters

ch - character to be pushed into the input stream buffer
stream - file stream to put the character back to

Return value

On success `ch` is returned.

On failure EOF is returned and the given stream remains unchanged.

Notes

The size of the pushback buffer varies in practice from 4k (Linux, MacOS) to as little as 4 (Solaris) or the guaranteed minimum 1 (HPUX, AIX).

The apparent size of the pushback buffer may be larger if the character that is pushed back equals the character existing at that location in the external character sequence (the implementation may simply decrement the read file position indicator and avoid maintaining a pushback buffer).

Example

demonstrates the use of `std::ungetc` in its original purpose: implementing `std::scanf`

Run this code

```
#include <cctype>
#include <cstdio>

void demo_scanf(const char* fmt, std::FILE* s)
{
    while (*fmt != '\0') {
        if (*fmt == '%') {
            switch (*++fmt) {
                case 'u': {
                    int c{};
                    while (std::isspace(c=std::getc(s))) {}
                    unsigned int num{};
                    while (std::isdigit(c)) {
                        num = num*10 + c-'0';
                        c = std::getc(s);
                    }
                    std::printf("%u scanned %u\n", num);
                }
            }
        }
    }
}
```

```
        std::ungetc(c, s);
        break;
    }
    case 'c': {
        int c = std::getc(s);
        std::printf("%%c scanned '%c'\n", c);
        break;
    }
}
} else {
    ++fmt;
}
}
}

int main()
{
    if (std::FILE* f = std::fopen("input.txt", "w+")) {
        std::fputs("123x", f);
        std::rewind(f);
        demo_scanf("%u%c", f);
        std::fclose(f);
    }
}
```

Output:

```
%u scanned 123
%c scanned 'x'
```

See also

fgetc gets a character from a file stream
getc (function)

C documentation for ungetc

Retrieved from "https://en.cppreference.com/mwiki/index.php?title=cpp/io/c/ungetc&oldid=132645"