stdin, stdout, stderr

Defined in header <stdio.h>

```
#define stdin  /* implementation-defined */    (1)
#define stdout /* implementation-defined */    (2)
#define stderr /* implementation-defined */    (3)
```

Three text streams are predefined. These streams are implicitly opened and unoriented at program startup.

1) Связанный со *стандартным входным* потоком, используемый для считывания обычного входного сигнала. При запуске программы поток полностью буферизуется тогда и только тогда, когда можно определить, что поток не ссылается на интерактивное устройство.
2) Связанный со *стандартным выходным* потоком, используемый для записи обычного вывода. При запуске программы поток полностью буферизуется тогда и только тогда, когда можно определить, что поток не ссылается на интерактивное устройство.
3) Связанный со *стандартным* потоком ошибок, используемый для записи диагностического вывода. При запуске программы поток не полностью буферизуется.

То, что представляет собой интерактивное устройство, определяется реализацией.

Эти макросы расширяются до выражений типа `FILE*`.

Хотя это и не предусмотрено POSIX, соглашение UNIX заключается в том, что `stdin`и `stdout`буферизуются строками, если связаны с терминалом и `stderr`небуферизованы.

Эти макросы могут быть расширены до изменяемых значений lvalues. Если какой-либо из этих `ФАЙЛОВ*` lvalue изменен, последующие операции с соответствующим потоком приводят к неопределенному или неопределенному поведению.

Пример

В этом примере показана функция, эквивалентная `printf`.

Запустите этот код

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

int my_printf(const char * restrict fmt, ...)
{
 va_list vl;
 va_start(vl, fmt);
    int ret = vfprintf(stdout, fmt, vl);
 va_end(vl);
    return ret;
}

int main(void)
{
 my_printf("Округление:\t%f%.0f%.32f\n", 1.5, 1.5, 1.3);
 my_printf("Заполнение:\t%05.2f%.2f%5.2f\n", 1.5, 1.5, 1.5);
 my_printf("Scientific:\t%E %e\n", 1.5, 1.5);
 my_printf("Hexadecimal:\t%a %A\n", 1.5, 1.5);
}
```

Possible output:

```
Rounding:       1.500000 2 1.30000000000000004440892098500626
Padding:        01.50 1.50  1.50
Scientific:     1.500000E+00 1.500000e+00
Hexadecimal:    0x1.8p+0 0X1.8P+0
```

References

- C17 standard (ISO/IEC 9899:2018):
    - 7.21.1 Introduction (p: 217-218)

        - 7.21.2 Streams (p: 217–219)
        - 7.21.2 Files (p: 219–221)
    - C11 standard (ISO/IEC 9899:2011):
        - 7.21.1 Introduction (p: 296–298)
        - 7.21.2 Streams (p: 298–299)
        - 7.21.2 Files (p: 300–302)
    - C99 standard (ISO/IEC 9899:1999):
        - 7.19.1 Introduction (p: 262–264)
        - 7.19.2 Streams (p: 264–265)
        - 7.19.2 Files (p: 266–268)
    - C89/C90 standard (ISO/IEC 9899:1990):
        - 4.9.1 Introduction
        - 4.9.2 Streams
        - 4.9.3 Files

See also

FILE	object type, capable of holding all information needed to control a C I/O stream (typedef)
C++ documentation for **stdin**, **stdout**, **stderr**	