За символом % может ничего не стоять или могут следовать следующие флаги:

#

Значение преобразуется в "альтернативную форму". Для преобразования типа **o** первый символ выходного потока преобразуется в нуль (префикс 0, если до этого его не было указано). Для преобразований типа **x** и **X** к ненулевому результату добавляется `0x' (или `0X' для преобразования типа **X**). При преобразованиях типа: **a**, **A**, **e**, **E**, **f**, **F**, **g** и **G** - результат всегда будет содержать десятичную точку, даже если за ним нет следующих цифр (обычно десятичная точка присутствует в результате таких преобразований, только если за ними следуют цифры). Для преобразований **g** и **G** завершающие нули не удаляются из результата, как это обычно случается. Результат других преобразований не определен.

0

Значение добавляет нули. При преобразованиях типа: **d**, **i**, **o**, **u**, **x**, **X**, **a**, **A**, **e**, **E**, **f**, **F**, **g** и **G** - преобразуемое значение слева дополняется нулями (вместо пробелов). Если присутствуют флаги **0**, и **-**, то флаг **0** игнорируется. Если преобразование указано точно(**d**, **i**, **o**, **u**, **x** и **X**), то флаг **0** игнорируется. Поведение флага при других преобразованиях не определено.

-

Выравнивает результат преобразования по левой границе поля. (По умолчанию выравнивание выполняется справа.) Исключает преобразование типа **n ,** дополняя значение справа пробелами. **-** Отменяет флаг **0**, если оба флага были установлены.

' '

(пробел). Устанавливает перед положительными числами (или пустой строкой) знак пробела для знаковых преобразований.

+

Знак + или - всегда помещается перед числом со знаком. По умолчанию знак используется только для отрицательных чисел. **+**отменяет действие пробела, если оба флага используются.

Ширина поля

Необязательная строка из десятичных цифр (с первой цифрой, отличной от нуля) определяет минимальную ширину поля. Если преобразованное значение имеет меньшее количество знаков, чем ширина поля, то оно слева дополняется пробелами (или справа, если указан флаг левого преобразования). Вместо строки десятичных цифр можно указать `*' или `*m$' (для некоторого десятичного числа m), чтобы определить ширину поля по ширине следующего аргумента или аргумента с номером m соответственно. Аргументы должны иметь тип *int*. Отрицательная ширина поля принимается как флаг `-', устанавливающий положительную ширину поля.

Несуществующая или небольшая ширина поля не делает его усеченным; если результат преобразования больше ширины поля, то поле расширяется, чтобы вместить в себя преобразованное значение.

Точность

Необязательный параметр точности в виде знака точки (`.') сопровождается необязательной строкой десятичных цифр. Вместо строки десятичных цифр можно указать `*' или `*m$' (для некоторого десятичного числа m), чтобы определить значение точности по следующему аргументу или аргументу с номером m соответственно, которые должны иметь тип *int*. Если точность указана как `.' или отрицательна, то она обрабатывается как нулевая. Это позволяет ограничивать количество выводимых символом для преобразований типов: **d**, **i**, **o**, **u**, **x** и **X**; показывать определенное количество цифр после десятичной точки для преобразований: **a**, **A**, **e**, **E**, **f** и **F**; показывать максимальное количество значащих цифр для преобразований **g** и **G** или максимальное количество символов для печати строк при преобразованиях **s** и**S**.

Модификаторы длины

Устанавливаются для следующих `целых преобразователей': **d**, **i**, **o**, **u**, **x** или **X.**

hh

Целые преобразователи соответствуют аргументам *signed char* или *unsigned char;* а также тип преобразования **n** соответствует аргументу указателя *signed char*.

h

Целые преобразователи соответствуют аргументам *short int* или *unsigned short int*; а также тип преобразования **n** соответствует аргументу указателя *short int*.

l

Целый преобразователь соответствует аргументам *long int* или *unsigned long int*; тип преобразования **n** соответствует аргументу указателя *long int*; тип преобразования **c** соответствует аргументу *wint_t*; тип преобразования **s** соответствует агрументу указателя*wchar_t*.

ll

Целые преобразователи соответствуют аргументу *long long int* или *unsigned long long int;* тип преобразования **n** соответствует аргументу указателя *long long int .*

L

Типы преобразования: **a**, **A**, **e**, **E**, **f**, **F**, **g** или **G** соответствуют аргументу *long*double. (C99 поддерживает %LF, а SUSv2 - нет.)

q

(Только для библиотеки libc5 в BSD 4.4 и Linux. Не используйте их.) Это синоним **ll**.

j

Целые преобразователи соответствуют аргументу *intmax_t* или *uintmax_t*.

z

Целые преобразователи соответствуют аргументу *size_t* или *ssize_t*. (Linux libc5 использует для этого **Z**. Не используйте данный аргумент.)

t

Целые преобразователи соответствуют аргументу *ptrdiff_t*.

Символы, которые определены как типы преобразования. Типы преобразования и их значения:

d,**i**

Параметр *int* преобразует символы в их знаковое десятичное отображение. Точность задает минимальное количество цифр в изображении результата; если результат можно показать с помощью меньшего количества цифр, то слева добавляются незначащие нули. По умолчанию значение точности равно единице. При выводе нуля с нулевой точностью выходной поток будет пуст.

o,**u**,**x**,**X**

Параметр *unsigned int* преобразуется в беззнаковое восьмеричное выражение (**o**), беззнаковое десятичное (**u**) или беззнаковое шестнадцатеричное (**x** и **X**). Буквы **abcdef** используются для преобразования **x**, а буквы **ABCDEF** - для преобразования **X**. Точность задает минимальное количество цифр в изображении результата; если результат можно показать с помощью меньшего количества цифр, то слева добавляются незначащие нули. По умолчанию значение точности равно единице. При выводе нуля с нулевой точностью выходной поток пуст.

e,**E**

Параметр *double* округляется и преобразуется в десятичное отображение в виде [-]d**.**ddd**e***(Pmdd**,** где одна цифра перед символом десятичной точки и количество цифр после нее указывают на необходимую точность; если точность отсутствует, она считается равной 6-и; если точность равна нулю, символ десятичной точки не показывается. Преобразование **E** использует букву **E** (чаще **e**) для отображения экспоненты. Экспонента всегда содержит две последние цифры; если ее значение равно нулю, экспонента равна 00.

f,**F**

Параметр *double* округляется и преобразуется в десятичное выражение в виде [-]ddd**.**ddd, где количество цифр после десятичной точки указывает на требуемую точность. Если точность отсутствует, она принимается равной 6-и; если точность равна нулю, десятичная точка не показывается. Если десятичная точка есть, перед ней должна быть минимум одна цифра. (SUSv2 не располагает информацией о типе **F** и

указывает, что этот символ предназначен для отображения символов бесконечности и NaN. Стандарт C99 определяет `[-]inf' или `[-]infinity' для указания на бесконечность, и строка начинается с `nan' для NaN в случае, если тип преобразования равен **f ;** она начинается с `[-]INF', `[-]INFINITY' или `NAN*' в случае, если тип преобразования равен **F**).

g,**G**

Параметр *double* преобразуется в тип **f** или **e** (или **F** или **E** для преобразования **G**). Точность определяется количеством значащих цифр. Если точность отсутствует, то она определяется равной 6-и цифрам; если точность равна нулю, то она трактуется как 1. Тип **e**используется, если экпонента преобразования меньше, чем -4, больше точности или равна ей. Завершающие нули удаляются из дробной части результата; десятичная точка стоит, только если за ней следует, по крайней мере, одна цифра.

a,**A**

(для C99; в SUSv2 этого нет). Для преобразования типа **a** параметр *double* отображается в шестнадцатеричной форме (с помощью букв abcdef) вида [-]**0x**h**.**hhhh**p***(Pmd; для преобразований типа **A** используется префикс **0X**, буквы ABCDEF и экспонентный разделитель **P**. Точность равна шестнадцатеричному разряду перед десятичной точкой и количеству цифр после нее. По умолчанию точность принимается равной точному значению разряда, если оно соответствует ему по основанию 2, и в противном случае размер ее является достаточным для определения значения типа *double*. Разряд перед десятичной точкой не определен для ненормализованных чисел и не равен нулю, но не определен для нормализованных чисел.

c

Если модификатор **l** не представлен, параметр *int* преобразуется в *unsigned char* и выводится как результирующее значение. Если модификатор **l** представлен, параметр *wint_t* (расширенные символы) преобразуется в многобайтовую пследовательность для вызова в функции **wcrtomb**; первое его значение преобразуется в начальное значение и выводится в виде многобайтовой строки.

s

Если модификатор **l** не представлен: параметр *const char ** преобразуется в указатель на массив символьного типа (строковый указатель). Символы из массива выводятся до заканчивающего символа **NUL** (не включая его); если точность определена, выводится не более установленного количества символов. Если точность указана, нулевой символ не нужен; если точность не указана или она больше, чем размер массива, массив должен содержать заканчивающий символ **NUL**. Если модификатор **l** присутствует: параметр*const wchar_t ** преобразуется в указатель на массив с расширенными символами. Расширенные символы из массива преобразуются в многобайтовые символы (которые вызываются функцией **wcrtomb**; при этом первое значение преобразуется в начальное значение первого расширенного символа) и содержат завершающий нулевой символ. Результирующие многобайтовые символы выводятся

до завершающего нулевого байта (не включая его). Если точность определена, выводится количество байтов, не превышающее это значение, а оставшаяся часть многобайтовых символов не выводится. Помните, что точность определяется числом выводимых*байтов*, а не *расширенных символов* или *экранной позиции*. Массив должен содержать конечный нулевой символ, если точность не указана, и содержать количество байтов, меньшее размера массива.

p

Параметр указателя *void **, выводящийся в шестнадцатеричном виде (так же, как **%#x** или **%#lx**).

n

Количество символов, выводящихся в целом типе *int ** (или других) без преобразующих параметров.

%

Выводит символ `%' без преобразующих параметров. Полный спецификатор преобразования - `%%'.