

1 КОНСТРУКЦИИ

список – это любая последовательность команд, разделенная **;** или **newline**, которые всегда взаимозаменяемы.

```
if список; then список
[ elif список; then список ] ...
[ else список ]
fi
```

```
for name [ in word ... ]
do список
done
```

```
for name in word ...; { list }
```

```
foreach name (word ...)
```

```
список
```

```
end
```

```
while список; do список; done
```

```
until список; do список; done
```

```
repeat word; do список; done
```

```
repeat word sublist
```

```
case word in
```

```
[ pattern ) список ;; ] ...
```

```
esac
```

```
case word { [ pattern ) список ;; ] ... }
```

```
select name [ in word ... ]; do список; done
```

Порожденный командный процессор: (*список*)

Текущий командный процессор: { *список* }

```
function word [ ( ) ] ... { список }
```

```
word ... ( ) { list }
```

```
word ... ( ) sublist
```

```
time [ pipeline ]
```

Условие: [[*exp*]]

Другие конструкции зависят от настроек

NO_SHORT_LOOPS and

CSH_JUNKIE_LOOPS и должны избегаться при использовании в скриптах.

2 ПОДСТАНОВКИ (GLOBBING)

Также смотрите опции **GLOB**, **EXTENDED_GLOB**, **KSH_GLOB**, **NULL_GLOB**, **NOMATCH**, **SH_GLOB**, **GLOB_DOTS**. *X*, *Y*, ... – любой образец (*pattern*). **#** и **##** требуют группировки предыдущих символов; эти же и **~**, **^** требуют наличия **EXTENDED_GLOB**.

*	Любая строка	с другими символами, например	Флаги подстановки задаются в форме (#X) и
?	Любой символ	[-+[:xdigit:]]	требуют наличия опции EXTENDED_GLOB . Они могут
[...]	Любой из помещенных в скобки символов	[^...]	появляться в группах. X может быть следующими:
[[:X:]]	Классы символов, где X может быть:		
alnum	Цифры и буквы,	< <i>x-y</i> >	i Соответствие независимо от регистра
alpha	Буквы,		l Буквы в нижнем регистре соответствуют
blank	Пробелы или табуляция,		буквам в верхнем
cntrl	Управляющий символ,	~ X	I Чувствительность к регистру: отменяет
digit	Десятичные цифры,	(X Y)	действие i и I
graph	Печатный, не пробельный символ,	X~Y	b Включает обратные ссылки для скобок.
lower	Символ в нижнем регистре,	(X Y~Z)	Массивы \$match , \$mbegin , \$mend
print	Печатный символ,	X#	содержат соответствующие строки,
punct	Печатный, не буква, цифра или пробел,	X##	индексы начала/конца
space	Пробельный символ,	(X)	V Отключает обратные ссылки, отменяя b
upper	Символ в верхнем регистре,	**/	Устанавливает \$MATCH , \$MBEGIN , \$MEND для строк
xdigit	Шестнадцатеричная цифра.	***/	M Отключает m .
	Вышеприведенные шаблоны используют		alim Позволяет <i>lim</i> ошибок в соответствии
	локаль, и могут быть скомбинированы		(для отключения задайте 0)

s	Соответствие только в начале строки (используйте в <code>????param expr</code>)	(setgid) (02000)		(или kb, mb, blocks) = (или <, >) <i>n</i>
e	Соответствие только в конце строки	t	файлы с битом sticky (01000)	~
		fspec	права доступа в стиле chmod например, <code>f70?</code> или <code>f:u+w,go-w:</code>	,
		estr	вычислить строку <i>str</i> , использовать файл (\$REPLY) если статус равен 0 или установить <code>\$reply</code> для файлового массива	-
		ddev	на номер устройства <i>dev</i>	M
		l[- +]ct	количество связей <i>ct</i> или больше (+), или меньше (-) чем <i>ct</i>	T
		U	владельцем является текущий эффективный пользователь	N
		G	владельцем является текущая эффективная группа	D
		uuid	владельцем является пользователь <i>uid</i> ; может также принимать формы <code>.name.</code> , <code>!name!</code> , ... или <code>(name)</code> , <code>{name}</code> , ...	n
		ggid	владельцем является группа <i>gid</i> , также как для <i>uuid</i> .	o[nLlamcd]
		a[Mwhm][- +]n	доступ производился (менее чем, более чем) <i>n</i> дней (месяцев, недель, часов, минут) назад	o[nLlamcd]
		m[Mwhm][- +]n	тоже самое для изменений	[beg[,end]]
		c[Mwhm][- +]n	тоже самое для изменений inode	:...
		L[kKmMpP][- +]n	размер в байтах	выбираемые индексы полученного списка файлов остальное рассматривается как модификаторы истории (каждый со своим знаком :)

Модификаторы подстановки указываются в скобках после шаблона (обычно с операцией *and*):

/	каталог
.	обычный файл
@	символьная ссылка
=	сокет
p	именованный канал (FIFO)
*	обычный исполняемый файл (0100)
%	файл устройства (символьного или блочного)
%b	специальное блочное
%c	специальное символьное
r	доступный для чтения (0400)
w	доступный для записи (0200)
x	доступный для выполнения (0200)
A	доступный для чтения группой (0040)
I	доступный для записи группой (0020)
E	доступный для выполнения группой (0010)
R	доступный для чтения всеми (0200)
W	доступный для записи всеми (0200)
X	доступный для выполнения всеми (0200)
s	с установленным битом пользователя (setuid) (04000)
S	с установленным битом группы

3 Опции

†означает установки по умолчанию: эти опции появляются со словом *po* в начале списка опций; *+o* отключает однобуквенные опции (показаны в скобках)

ALL_EXPORT	Экспортировать все новые параметры процессора (-a)	AUTO_CD	Выполнять <code>cd</code> если команда является каталогом (-J)	AUTO_PARAM_KEYS	становятся именами Умное удаления после дополнения параметра
ALWAYS_LAST_PROMPT	Вернуться к приглашению после перечисления	AUTO_LIST	Перечислять неоднозначные дополнения (-9)	AUTO_PARAM_SLASH	<code>\$path<TAB></code> в <code>→ \$path/</code>
ALWAYS_TO_END	В конец слова после выполнения дополнения	AUTO_MENU	Показывать меню после второго нажатия на <code>TAB</code>	AUTO_PUSHD	При выполнении <code>cd</code> выполнять <code>pushd (-N)</code>
APPEND_HISTORY	Добавлять историю в файл	AUTO_NAME_DIRS	Параметры с путями,	AUTO_REMOVE_SLASH	Удалять слеш из имени

AUTO_RESUME	каталога после дополнения cmd может вести себя подобно %cmd (-W)	CSH_JUNKIE_LOOPS	Списки могут записываться как list; end	HASH_CMDS [†]	Хешировать команды при запуске
BAD_PATTERN [†]	Выдавать ошибки при неправильном шаблоне подстановки (+2)	CSH_JUNKIE_QUOTES	Не допускать не-маскированных переводов строк в кавычках	HASH_DIRS [†]	Хешировать каталоги при запуске команд
BANG_HIST [†]	Использовать !hist в командной строке (+K)	CSH_NULLCMD	Не использовать \$NULLCMD, \$READNULLCMD	HASH_LIST_ALL [†]	Хешировать все команды при дополнениях
BARE_GLOB_QUAL [†]	Использовать скобки как квалификаторы глоббинга	CSH_NULL_GLOB	Только одна подстановка должна соответствовать	HIST_ALLOW_CLOBBER	Допускать разрушающие перенаправления в истории
BASH_AUTO_LIST [†]	Выдавать список только после второго нажатия на TAB	DVORAK	Для корректировки использовать клавиатуру Дворака	HIST_BEEP [†]	Выдавать сигнал на неправильной ! истории
BEEP [†]	Выдавать сигнал при ошибках и т.п. (+B)	EQUALS [†]	Выполнять раскрытие =cmd	HIST_EXPIRE_DUPS_FIRST	Удалять дублирующиеся строки для уменьшения истории
BG_NICE [†]	Уменьшать приоритет фоновых заданий (-6)	ERR_EXIT	Выходить из командного процессора при возникновении ошибок (-e)	HIST_FIND_NO_DUPS	Никогда не показывать дубликаты в истории
BRACE_CCL	foo{ab} → fooa foob	EXEC [†]	Выполнять команды (+n)	HIST_IGNORE_ALL_DUPS	Никогда не сохранять дубликат существующей записи истории
BSD_ECHO	Встроенная команда echo будет работать аналогично команде в BSD	EXTENDED_GLOB	использовать #, ~ и ~ в шаблонах	HIST_IGNORE_DUPS	Не допускать смежных повторений в истории (-h)
CDABLE_VARS	cd foo аналогично cd ~foo (-T)	EXTENDED_HISTORY	Сохранять временные метки в файле истории	HIST_IGNORE_SPACE	Не сохранять строки ' cmd' (-g)
CHASE_DOTS	Анализировать ссылки при выполнении .. в каталогах	FLOW_CONTROL [†]	~S, ~Q производят контроль выполнения	HIST_NO_FUNCTIONS	Не сохранять определения функций
CHASE_LINKS	Анализировать символьные ссылки в каталогах (-w)	FUNCTION_ARGZERO [†]	Устанавливать \$0 для функций и команды source	HIST_NO_STORE	Не сохранять команды работы с историей
CHECK_JOBS [†]	Сообщать статус заданий при выполнении exit	GLOB [†]	Выполнять подстановки (+F)	HIST_REDUCE_BLANKS	Удалять из истории лишние пробелы
CLOBBER [†]	> в существующий файл требует > (+C)	GLOBAL_EXPORT [†]	typeset -x применяется глобально	HIST_SAVE_NO_DUPS	Удалять дубликаты при сохранении истории
COMPLETE_ALIASES	Дополнения используют нераскрытые алиасы	GLOBAL_RCS [†]	Использовать стартовые файлы из /etc	HIST_VERIFY	Продолжать редактирование после раскрытия !
COMPLETE_IN_WORD	Дополнять в точке курсора в слове	GLOB_ASSIGN	scalar=* выполняет подстановки справа	HUP [†]	При выходе посылать заданиям сигнал SIGHUP
CORRECT	Корректировка имени команд (-0)	GLOB_COMPLETE	Дополнять подстановки с помощью меню	IGNORE_BRACES	Не выполнять раскрытие {...} (-I)
CORRECT_ALL	Корректировка всех аргументов (-O)	GLOB_DOTS	Ведущие точки соответствуют метасимволам (wildcards) (-4)	IGNORE_EOF	Не выполнять выхода при получении первых десяти eof (-7)
CSH_JUNKIE_HISTORY	Одиночный знак !	GLOB_SUBST	Текст из параметров может подставляться	INC_APPEND_HISTORY	Сохранять историю по мере поступления
				INTERACTIVE	Командный процессор в интерактивном режиме

	(не устанавливается) (-i)		на TAB (-Y)		каталога в стеке
INTERACTIVE_COMMENTS	Интерактивно использовать комментарии (-k)	MONITOR	Разрешить контроль заданий (-m)	PUSHD_MINUS	Поменять местами минус и плюс в pushd
KSH_ARRAYS	Синтаксис работы с массивами подобный ksh	MULTIOS†	Неявно выполнять tee/cat для множества <, >	PUSHD_SILENT	Не печатать стек каталогов (-E)
KSH_AUTOLOAD	Эмулировать загрузку функций ksh	NOMATCH†	Выдавать ошибку для несуществующих раскрытий (+3)	PUSHD_TO_HOME	При запуске без аргументов, pushd переходит в домашний каталог (-D)
KSH_GLOB	Эмулировать шаблоны, *(...) ksh и т.п.	NOTIFY†	Сообщать при изменении статуса фонового задания (-5)	RC_EXPAND_PARAM	A\${array}Z → Aa1Z Aa2Z ... (-P)
KSH_OPTION_PRINT	Выдавать опции как это делает ksh	NULL_GLOB	Удалить несуществующие раскрытия из списка (-G)	RC_QUOTES	echo ''' → ' .-файлы, в противном случае просто /etc/zshenv (+f)
LIST_AMBIGUOUS	Только отображать неоднозначные дополнения	NUMERIC_GLOB_SORT	Сортировка чисел при раскрытии	RCS†	Предпочитать точное соответствие при дополнении (-S)
LIST_BEEP	Выдавать сигнал при неоднозначных дополнениях	OCTAL_ZEROES	В математических выражениях 0 задает восьмеричные числа	REC_EXACT	Не может вызвать больших повреждений
LIST_PACKED	Сжимать список дополнений	OVER_STRIKE	Редактор запускается в режиме замены	RESTRICTED	Не выдавать запрос для rm * (-H)
LIST_ROWS_FIRST	При дополнении первыми отображать строки	PATH_DIRS	Производить поиск dir/cmd в путях поиска (-Q)	RM_STAR_SILENT	Не верить первому ответу RMSTAR
LIST_TYPES	Показывать типы файлов в списке дополнений (-X)	POSIX_BUILTINS	Встроенная команда builtin является специальной	RM_STAR_WAIT	Читать/записывать историю по мере надобности
LOCAL_OPTIONS	Опции установленные в функциях являются локальными	PRINT_EIGHT_BIT	Показывать в листинге символы с установленным восьмым битом	SHARE_HISTORY	Сначала выполнять ~file, =cmd
LOCAL_TRAPS	Сбрасывать ловушки (traps) при выходе из функции			SH_FILE_EXPANSION	Запретить (, ,), < в шаблонах
LOGIN	Командный процессор в режиме login (не устанавливается) (-l)	PRINT_EXIT_VALUE	Отображать ненулевые коды возврата (-1)	SH_GLOB	Читать команды со стандартного ввода (-s)
LONG_LIST_JOBS	Всегда использовать jobs -l (-R)	PRIVILEGED	Привилегированный режим: в первую очередь безопасность! (-p)	SHIN_STDIN	Пустая команда предполагает поведение :
MAGIC_EQUAL_SUBST	Любое выражение var=expr раскрывает файлы expr	PROMPT_BANG	В приглашении ! является специальным	SH_NULLCMD	Интерпретация однобуквенных опций как в ksh
MAIL_WARNING	Выдавать предупреждение при был доступ к файлу почты (-U)	PROMPT_CR†	Выдавать CR перед приглашением (+V)	SH_OPTION_LETTERS	Короткие версии for, select, if, function
MARK_DIRS	Добавлять / к списку каталогов при раскрытии (-8)	PROMPT_PERCENT†	Выполнять раскрытие % в приглашении	SH_WORD_SPLIT	Разбивать слова подобно другим командным процессорам (-y)
MENU_COMPLETE	Прокручивать список дополнений при нажатии	PROMPT_SUBST	Раскрывать подстановки в приглашении	SINGLE_COMMAND	Считать команду и прекратить работу (-t)
		PUSHD_IGNORE_DUPS	Только одна запись для	SINGLE_LINE_ZLE	Редактор использует только одну строку (-M)
				SUN_KEYBOARD_HACK	Игнорировать несоответствующий,

UNSET [†]	завершающий знак ‘ (-L) Неустановленные параметры вызывают ошибку (+u)	XTRACE	по мере чтения (-v) При запуске печатать команды и аргументы (-x)	командного процессора (-Z)
VERBOSE	Печатать введенные строки	ZLE	Использовать строковый редактор	

4 РАСКРЫТИЕ ПАРАМЕТРОВ

<code>\$name</code>	Заменить самое короткое соответствие шаблону	Флаги: использование <code>\${(o)name}</code> и т.п.
<code>\${name}</code>	Базовая подстановка параметра	A <code>\${...:=...}</code> создает массив
<code>\${+name}</code>	1 если <code>name</code> установлен, 0 в противном случае	AA ... создает ассоциативный массив
<code>\${name:-word}</code>	<code>\$name</code> если не равен нулю, иначе <code>word</code>	@ Разбивает на слова в двойных кавычках
<code>\${name-word}</code>	<code>\$name</code> если установлен, иначе <code>word</code> (Аналогично другим с/без выражениям).	e Использовать раскрытия в результате
<code>\${name:=word}</code>	<code>\$name</code> если не равен нулю, иначе использовать <code>word</code> и установить <code>name</code> равным ему	P Заставить заново использовать <code>\$name</code> в качестве имени
<code>\${name==word}</code>	Безусловное присвоение <code>\${name:?word}</code>	o Сортировать слова в порядке возрастания
<code>\${name:+word}</code>	<code>\$name</code> если не равен нулю, иначе напечатать <code>word</code> и выйти	O Сортировать слова в порядке убывания
<code>\${name#pattern}</code>	<code>\$name</code> с самым коротким (длинным) соответствием шаблону <code>pattern</code> удаляется с головы. Параметры как в подстановке; оригинальный параметр не изменяется	i Независимость от регистра при использовании o или O
<code>\${name##pattern}</code>	Включает (отключает) RC_EXPAND_PARAM	L Все буквы в нижнем регистре
<code>\${name%pattern}</code>	Включает (отключает) SH_WORD_SPLIT	U Все буквы в верхнем регистре
<code>\${name%%pattern}</code>	Включает (отключает) GLOB_SUBST	C Сделать первую букву заглавной
<code>\${name/pattern/repl}</code>	Применяет модификатор шаблона <code>mod</code>	V Отображать специальные символы
<code>\${(S)name/pattern/repl}</code>	Выполняет оба набора изменений над значением, не выполняет дополнительного поиска, смотрите (P)	q Маскировать результаты символом \
		qq Маскировать результаты символом ’
		qqq Маскировать результаты символом "
		qqqq Маскировать результаты символами \$’...’
		Q Удалить один уровень маскирования командного процессора
		% Раскрыть маски приглашения
		%% Раскрыть как приглашение с текущими настройками
		X Выдать ошибки разбора масок, шаблонов
		c <code>\${#name}</code> подсчитывает символы
		w <code>\${#name}</code> подсчитывает слова
		W Как w, но считает пустые слова
		k В ассоциативном массиве включать ключи
		v В ассоциативном массиве включать значения
		p Использовать маски печати в аргументах
		F Соединить слова с переводами строк
		f Разбить по границе строк

z Разбить обычным способом
t Подставленное описание, не значение

Флаги с разделителями; используйте любую пару символов вместо двоеточий, также можно использовать <>, (), {}, []

l:expr::string1::string2:
 Заполнить слова слева до *expr* символов используя повторение *string1* (по умолчанию пробел), *string2* вставляется только раз

r:expr::string1::string2:
 Тоже самое только заполняя справа

j:string:
 Соединить слова используя *string* (происходит до разбиения)

s:string:
 Разбить слова по *string*

Флаги применяемые с $\${...#...}$ или $\${...%...}$

S также искать в подстроках
I:expr:
 Искать/подставлять *expr*-е совпадение

M Включить совпадение
R Включить не совпавшую часть (остаток)
B Включить индекс начала
E Включить индекс конца
N Включить длину совпавшей части

Сводная таблица правил подстановки

- 1 Вложенная подстановка, $\${\${...}}$
- 2 Индексирование параметров по имени, $\${name[i]}$
- 3 Флаг (P)
- 4 Соединение "\${...}"
- 5 Вложенный индекс, $\${\${...}[i]}$
- 6 #, %, /. модификации :
- 7 флаг (j) или соединение пробелами
- 8 разбиение (s), (f), (z) или =
- 9 Разбиение слов командного процессора (без флагов)
- 10 Флаг (e)
- 11 Заполнение (l) или (r)

Флаги в индексировании: использование $\${name[(i)index]}$ и т.п.

e Только для обратной совместимости
w Индексировать по словам скаляра
s:string:
 Разделять слова с помощью *string*

p В следующем выражении использовать печатные маски *s*

f Индексировать по строкам: то же самое как *pws:\n:*
r Обратное индексирование массива/подстроки/слова
 Для ассоциативных массивов, соответствие для значений

R Как *r*, но последнее совпадение (все для ассоциативных массивов)

k В ассоциативных массивах, ключи являются шаблонами; получить первый

K В ассоциативных массивах, ключи являются шаблонами; получить все

i Как *r*, но вернуть индекс. Для ассоциативных массивов, соответствие для ключей

I Как *i*, но последнее совпадение (все для ассоциативных массивов)

n:expr:
 Использовать *expr*-е первое/последнее совпадение

b:expr:
r, R, i, I начинают поиск с *expr*-й элемента.

5 История

Также смотрите параметры **histchars**, **HISTFILE**, **HISTSIZE**, **SAVEHIST** и опции **APPEND_HISTORY**, **CSH_JUNKIE_HISTORY**, **EXTENDED_HISTORY**, **HIST_ALLOW_CLOBBER**, **HIST_IGNORE_DUPS**, **HIST_IGNORE_SPACE**, **HIST_NO_STORE**, **HIST_VERIFY**, **BANG_HIST**, **HIST_BEEP**, **HIST_EXPIRE_DUPS_FIRST**, **HIST_FIND_NO_DUPS**, **HIST_IGNORE_ALL_DUPS**, **HIST_NO_FUNCTIONS**, **HIST_REDUCE_BLANKS**, **HIST_SAVE_NO_DUPS**, **INC_APPEND_HISTORY**, **SHARE_HISTORY**.

События:

!	начать подстановку истории, если не после пробела, новой строки, =, (!-n	строка <i>n</i> до текущей	!{...}	изолировать ссылку на историю
!!	предыдущая команда	!str	последняя строка начинавшаяся с <i>str</i>	!"	не выполнять подстановок на этой строке
!n	командная строка <i>n</i>	!?str[?]	последняя строка содержащая <i>str</i>		
		!#	текущая команда		

Слова: отделенные от события символом ‘:’

0	первое слово в строке (команда)	%	слово соответствующее ?s	x*	то же самое, что и x-\$
n	n-й аргумент команды	x-y	диапазон слов	x-	то же самое, но пропустить слово \$
^	первый аргумент команды	-y	то же самое, что и 0-y		
\$	последний аргумент команды	*	все аргументы		

Модификаторы: также с подстановками (globbing) и параметрами

h	(голова) удалить последний компонент пути	q	замаскировать слова от дальнейших подстановок	g	заменить <i>old</i> на <i>new</i> (строки) (до s) заменить каждое вхождение
r	удалить суффикс <i>.suf</i>	Q	удалить один уровень маскирования	f	повторять пока не останется вхождений
e	оставить только суффикс <i>suf</i>	x	то же самое, но разбить слова по пробелам	F:exp:r:	то же самое, но максимум <i>exp</i> изменений (как префикс) применять к каждому слову
t	(хвост) оставить только последний компонент пути	l	все буквы в нижний регистр	w	то же самое, но разделять слова с помощью <i>sep</i>
&	повторить последнюю подстановку	u	все буквы в верхний регистр	W:sep:	
p	не выполнять новую команду	s/old/new[/]			

6 ПАРАМЕТРЫ

Специальные параметры: массивы именованы в нижнем регистре, за исключением *status*; те, что помечены как † могут присваиваться:

!	PID последнего фонового задания	EGID†	эффективный GID	TTY	имя терминала командного процессора
ARGC		EUID†	эффективный UID	TTYIDLE	время ожидания терминала (сек.) или -1
#	кол-во позиционных параметров	ERRNO	номер системной ошибки	UID†	UID
\$	текущий PID	GID†	текущий GID	USERNAME†	имя пользователя
-	набор флагов командного процессора	HOST	имя машины	VENDOR	производитель компьютера
argv†		LINENO	номер вводимой строки	ZSH_NAME	имя запуска командного процессора
*†	массив позиционных параметров	LOGNAME	имя пользователя	ZSH_VERSION	ID версии zsh
@	то же самое, что и argv[@]	MACHTYPE	тип машины		
status		OLDPWD	предыдущий рабочий каталог		
?	статус последней программы	OPTARG			
pipestatus	массив статусов при конвейерной обработке	OPTIND	значение и индекс последней опции getopts		
-	последний аргумент предыдущей команды	OSTYPE	тип OS		
CRUPTYPE	тип CPU определенный во время работы	PPID	PID родительского процесса		
		PWD	текущий рабочий каталог		
		RANDOM†	произвольное целое число: присвойте для генерации		
		SECONDS†	кол-во секунд прошедших с момента запуска командного процессора		
		SHLVL	увеличивается для каждого zsh		
		signals	имена сигналов		
				Другие параметры используемые командного процессора († — путь разделенный двоеточием)	
				ARGVO	экспортируйте для изменения argv[0]
				BAUD	скорость линии (0 для игнорирования)
				cdpath, CDPATH†	поиск в каталогах для команды cd
				COUMNS	No. колонок на терминале
				DIRSTACKSIZE	максимальный размер стека каталогов
				FCEDIT	редактор по умолчанию для команды fc
				figignore, FIGIGNORE†	суффиксы, пропускаемые при дополнении
				fpath, FPATH†	пути поиска автозагружаемых функций
				histchars	три символа: 1) начало истории (!),

2) быстрая подстановка истории (^), 3) комментарий (#)	MAILCHECK как часто проверять MAIL (сек.)	REPORTTIME для длительных команд выдавать занятое время (сек.)
HISTCHARS то же самое, что и <code>histchars</code>	mailpath, MAILPATH [†] список файлов проверяемых на наличие новой почты. За каждым может указываться ?'выдаваемое сообщение'	RXPROMPT приглашение выдаваемое в правой части строки
HISTFILE где сохранять историю командного процессора	manpath, MANPATH [†] не используется командным процессором, вероятно используется командой <code>man</code>	SAVEHIST макс. число строк в файле истории
HISTSZIE максимальное кол-во команд в истории (внутреннее хранение)	module_path, MODULE_PATH [†] пути для динамических модулей; не импортируется	SPROMPT приглашение используемое при проверке ошибок
HOME значение по умолчанию для команды <code>cd</code>	NULLCMD используется для перенаправлений без команды	STTY аргументы для для команды <code>stty</code> , экспортируйте эту переменную для запуска до выполнения внешней команды
IFS разделитель слов при вводе	path, PATH [†] пути поиска команд	TERM тип терминала
KEYTIMEOUT время ожидания клавиши в последовательности	POSTEDIT сообщение после выхода из строкового редактора	TIMEFMT формат отчетов о времени выполнения
LANG основное значение региональных настроек	PROMPT, prompt приглашение, используемое редактором	TMOUТ путь к временным файлам (/tmp/zsh)
LC_ALL переопределяет LANG и другие LC_*	PS1 приглашение при продолжении редактирования	TMPPREFIX список пользователей для наблюдения за их входом/выходом (также значения <code>all</code> , <code>notme</code> , % tty, @ host)
LC_COLLATE определяет порядок символов	PROMPT2, PS2 приглашение используемое командой <code>select</code>	WATCHFMT формат отчетов <code>watch</code>
LC_CTYPE определяет классы символов для сообщений:	PROMPT3, PS3 приглашение трассировщика выполнения	WORDCHARS не-цифро-буквенные символы используемые редактором как часть слова
LC_MESSAGES не используется zsh	PROMPT4 PS4 заменить %v в приглашениях	ZBEEP последовательность для вывода вместо выдачи сигнала
LC_NUMERIC для вывода десятичной точки и разделителя чисел	psvar, PSVAR [†] команда используемая при использовании перенаправления	ZDOTDIR где искать .zshrc и т.п.
LC_TIME формат времени и даты	READNULLCMD	
LINES No. строк на терминале		
LISTMAX No. файлов для перечисления без запроса		
LOGCHECK как часто проверять <code>watch</code> (сек.)		
MAIL файл проверяемый на наличие почты		

управляющие последовательности строки приглашения: помеченные как [†] могут использовать целые числа *n*, которые должны сразу следовать за %. По умолчанию число равно 1, за исключением %_.

%% A '%'	%m [†] имя машины до <i>n</i> -й точки	%N имя скрипта, функции
%) A ')'	%S %B %U начать standout, жирный, подчеркивание	%i номер строки внутри %N
%d %/ [†] \$PWD	%s %b %u остановить соответствующий режим	%w дата в виде day-dd
%~ [†] \$PWD, но использует сокращение ~	%t %@ время в 12-часовом формате	%W дата в виде mm/dd/yy
%h %! текущий номер события истории	%T время в 24-часовом формате	%D дата в виде yy-mm-dd
%L текущее значение \$SHLVL	%* то же самое, с секундами	%D{string}
%M полное имя машины	%n \$USERNAME	использовать <code>strftime</code> для

	форматирования <i>string</i>	следовать за '('. тест истинен, если:			
%l	текущий терминал	с . ~	путь ~ содержит $\geq n$ элементов	управляющие последовательности в \$WATCHFMT:	
/?	код возврата последней команды	/ C	то же самое, но для абсолютного пути		
%_†	состояние парсера, <i>n</i> для указания максимального уровня	t	текущая минута равна <i>n</i>	%n	имя пользователя
%E	очистить до конца строки	T	текущий час равен <i>n</i>	%a	'logged on' или 'logged off'
%#	# если root, иначе %	d	текущий день равен <i>n</i>	%l	терминал пользователя
%v†	<i>n</i> -й элемент \$psvar	D	месяц равен <i>n</i> (Jan = 0)	%M	полное имя удаленного компьютера
%{...%}	строка, которая не перемещает курсор	w	день недели равен <i>n</i> (Sun = 0)	%m	имя компьютера до первой '.'
%<string< >string> % [<string] % [>string]	обрезает <i>string</i> слева или справа, <i>n</i> задает максимальную длину	?	код возврата последней команды был равен <i>n</i>	%S %U %B	начать режимы standout, жирный, подчеркивание
%c† % .†	компонент \$PWD (устаревшее)	#	запущен с uid <i>n</i>	%s %u %b	остановить соответствующий режим
%C	то же самое, но без раскрытия ~	g	запущен с gid <i>n</i>	%t %@	время в 12-часовом формате
		L	\$SHLVL $\geq n$	%T	время в 24-часовом формате
		S	\$SECONDS $\geq n$	%i	номер строки внутри %N
		v	\${#psvar} $\geq n$	%w	дата в виде day-dd
		-	по крайней мере <i>n</i> конструкций командного процессора	%W	дата в виде mm/dd/yy
		!	истина, если привилегированный командный процессор	%D	дата в виде yy-mm-dd

Коды для трехэлементных выражений в строке приглашения, формат $\%(char.true-text.false-text)$, целое число *n* может предшествовать или

Трехоперандные выражения в \$WATCHFMT, формат $\%(char.true-text.false-text)$, может быть использован с l, n, m or M (истина, если не пустое значение для соответствующего %), или a (истина для подключения (login), ложь для отключения (logout)).

7 Условия

Проверки файлов: за ними следует имя файла

Cond**истина если файл**

- a существует
- b специальный блоковый
- c специальный символьный
- d каталог
- e существует
- f обычный файл
- g с установленным битом setgid
- h символьная ссылка
- k с установленным битом закрепления (sticky bit)
- p FIFO/pipe
- r доступный для чтения
- s имеет размер > 0
- u с установленным битом setuid
- w доступный для чтения

- x исполняемый/каталог доступный для чтения
- L символьная ссылка
- O владеет UID
- G владеет GID
- S сокет
- N время доступа не позже чем
время модификации

Другие тесты с одним аргументом:

- n строка, длина > 0
- o опция, установлена
- t файловый дескриптор (fd), открыт на терминал
- z строка, нулевой длины

Проверки с двумя аргументами ([[a test b]]):

- nt файл *a* моложе *b*
- ot файл *a* старше *b*
- ef имена ссылаются на один и тот же файл
- =
- == строка соответствует шаблону
- != ... не соответствует
- < ASCII до
- > ASCII после
- eq числа равны
- ne числа не равны
- lt числовое $a < b$
- gt числовое $a > b$
- le числовое $a \leq b$
- ge числовое $a \geq b$

А также группировка (...), отрицание !, и &&, или ||; специальная обработка /dev/fd.