

Встроенная команда unset

Краткое описание

```
сбросить [-f|v] [-n] [ИМЯ ...]
```

Описание

unset Встроенная команда используется для отмены значений и атрибутов переменных и функций оболочки. Без какой-либо опции unset пытается сначала отменить переменную, а затем функцию.

Опции

Опция	Описание
-f	обрабатывает каждый NAME из них как имя функции
-v	обрабатывает каждый NAME из них как имя переменной
-n	обрабатывает каждый NAME из них как ссылку на имя и отключает саму переменную, а не переменную, на которую она ссылается

Статус выхода

Статус	Причина
0	ошибки нет
!=0	недопустимый параметр
!=0	недопустимая комбинация параметров (-v и -f)
!=0	данное NAME доступно только для чтения

Примеры

```
РЕДАКТОР unset -v
```

```
unset -f myfunc1 myfunc2
```

Область применения

В `bash` `unset` обладает некоторыми интересными свойствами благодаря своей уникальной динамической области видимости. Если локальная переменная одновременно объявлена и отключена (путем вызова `unset` на локальном) из одной и той же области действия функции, то переменная отображается отключенной от этой области и всех дочерних областей до тех пор, пока либо не будет возвращен результат из функции, либо другая локальная переменная с тем же именем не будет объявлена под тем местом, где была отключена исходная переменная. Другими словами, переменная выглядит неустановленной для всего, пока не вернется из функции, в которой переменная была установлена (и сброшена), после чего переменные с тем же именем из более высоких областей снова будут обнаружены и доступны.

Однако, если `unset` вызывается из дочерней области относительно того, где была установлена локальная переменная, тогда переменная с тем же именем в самой внешней области становится видимой для ее области и всех дочерних элементов - как если бы переменная, которая была отключена, никогда не устанавливалась изначально. Это свойство позволяет просматривать стек вверх, поскольку имена переменных не заданы, если `unset` и локальные, которые он не устанавливает, не находятся вместе на одном уровне области видимости.

Вот демонстрация этого поведения.

```
#!/usr/bin/env bash

FUNCNEST=10

# Глубина прямой рекурсии.
# Найдите в стеке первое не-ФУНКЦИОНАЛЬНОЕ ИМЯ [1] и посчитайте, насколько мы глубоки.
Вызовите Depth() {
    # Уберите "main" с конца FUNCNAME[@], если текущая функция называется "main" и
    # Bash добавил дополнительный "main" для неинтерактивных скриптов.
    если [[ main == !(!("${FUNCNAME[1]}")|("${IMYAFUNKCIY[-1]}")) ]] && $- != * i* ]; затем
    local -a 'имена имен=("${FUNCNAME[@]:1:${#FUNCNAME[@]}-2}")'
    else
    local -a 'имена имен=("${FUNCNAME[@]:1}")'
    fi

    if (( ! ${#имена[@]} )); затем
    printf 0
    возвращает
    fi

    local n
    в то время как [[ $fnames == ${fnames[++n]} ]]; выполнить
    :
    готово

    printf -- $n
}

# Эта функция - волшебный ходок стека.
unset 2() {
    unset -v -- "$@"
}

f() {
    local a
    if (( (a=$(callDepth)) <= 4 )); затем
    (( a == 1 )) && unset a
    (( a == 2 )) && declare -g a='глобальная область уо'
    если
    вы еще
    не объявили ловушку - выполните ОТЛАДКУ
    unset 2 a # объявить -- a="5"
    сбросить a a # объявить -- a="4"
    сбросить # объявить -- a="2"
    сбросьте значение # ./unset-tests: строка 44: объявление: a: не найдено
    : # declare -- a="глобальная область действия уо"
    фи
}

a= 'глобальная область действия'
f
```

```
# vim: set fenc= utf-8 ff = unix ts = 4 sts = 4 sw = 4 ft = sh nowrap
et:
```

ВЫВОД:

```
объявить -- a="5"
объявить -- a="4"
объявить -- a="2"
./unset-tests: строка 44: объявить: a: не найдено
объявить -- a="глобальная область действия уо"
```

Некоторые вещи, на которые следует обратить внимание:

- `unset2` действительно нужна только один раз. У нас остается 5 уровней в `f 's` для оставшихся `unset` вызовов, которые удаляют внешние слои `a 's`.
- Обратите внимание, что `"a"` сбрасывается с помощью обычной команды `unset` на глубине рекурсии 1, а затем при вызове `unset` снова появляется `a` в глобальной области видимости, которая с тех пор была изменена в более низкой области видимости с помощью `declare -g`.
- Объявление глобального с помощью `declare -g` обходит все локальные и устанавливает или изменяет переменную глобальной области видимости (вне всех функций). Это не влияет на видимость глобального.
- Это не относится к отдельным элементам массива. Если два локальных массива с одинаковыми именами отображаются в разных областях, весь массив внутренней области должен быть сброшен, прежде чем какие-либо элементы внешнего массива станут видимыми. Это делает `"unset"` и `"unset2"` идентичными для отдельных элементов массива, а для массивов в целом `unset` и `unset2` ведут себя так же, как и для скалярных переменных.

Аргументы

Как и несколько других встроенных команд Bash, которые принимают имена параметров, `unset` расширяет свои аргументы.

```
~ $ (a={a..d}); сбросить 'a[2]'; объявить -p a )
объявить -a a='([0]="a" [1]="b" [3]="d")'
```

Как обычно в таких случаях, важно указывать аргументы в кавычках, чтобы избежать случайных результатов, таких как глобус.

```
~ $ (a={a..d}) b=a c=d d=1; установить -x; сбросить "${b}["{2..3}-c
\]; объявить -p a )
+ сбросить 'a[2-1]' 'a[3-1]'
+ объявить -p a
объявить -a a='([0]="a" [3]="d")'
```

Конечно, трудно уловимая косвенность все еще возможна, когда задействована арифметика, также, как показано выше, даже без дополнительных расширений.

В Bash `unset` встроенная команда оценивает индексы массива только в том случае, если сам массив установлен.

```
~ $ ( unset -v 'a[$(echo а было установлено >&2)0]' )  
~ $ ( a=(); unset -v 'a[$(echo а было установлено >&2)0]')  
была установлена
```

Соображения о переносимости

Цитирую POSIX:

Если не указаны ни `-f`, ни `-v`, наме ссылается на переменную; если переменная с таким именем не существует, не указано, должна ли функция с таким именем, если таковая имеется, быть отключена.

Поэтому рекомендуется явно указывать `-f` или `-v` при использовании `unset`. Кроме того, я предпочитаю это как вопрос стиля.

Смотрите также

- Команда `declare builtin`
- **Встроенная команда `unset`**
- Утилита POSIX `"unset"`
(http://pubs.opengroup.org/onlinepubs/9699919799/utilities/V3_chap02.html#tag_18_



Обсуждение