

Bash 4 - грубый обзор



Внимание: Поскольку Bash 4 существует уже довольно давно (скоро выйдет 4.3), я считаю его "стандартным". Эта страница больше не поддерживается и оставлена здесь, чтобы ваши ссылки работали. Смотрите страницу изменений в Bash, чтобы узнать о новых материалах.

Помимо множества исправлений, начиная с версии Bash 3.2, Bash 4 принесет некоторые интересные новые функции для пользователей оболочки и разработчиков сценариев. См. Также Изменения в Bash для небольшого общего обзора с более подробной информацией.

Здесь представлены не все изменения и новости, а только самые большие или самые интересные. Изменения в completion и компоненте readline не рассматриваются.

Хотя, если вы знакомы с этими частями Bash (и Bash 4), не стесняйтесь написать здесь главу.

Полный список исправлений и изменений находится в файле ИЗМЕНЕНИЙ или НОВОСТЕЙ вашего дистрибутива Bash 4.

Текущая доступная **стабильная** версия -
4.4.18 (03 февраля 2018):

- <ftp://ftp.cwru.edu/pub/bash/bash-4.4.18.tar.gz>
(<ftp://ftp.cwru.edu/pub/bash/bash-4.4.18.tar.gz>)
- <ftp://ftp.gnu.org/pub/gnu/bash/bash-4.4.18.tar.gz>
(<ftp://ftp.gnu.org/pub/gnu/bash/bash-4.4.18.tar.gz>)

Новые или измененные команды и ключевые слова

Новое ключевое слово "coproc"

В Bash 4 представлены концепции сопроцессов, хорошо известные особенности других оболочек. Основная концепция проста: она запускает любую команду в фоновом режиме и настраивает массив, который заполняется доступными файлами,

представляющими `filedescriptors` запущенного процесса.

Другими словами: он позволяет запускать процесс в фоновом режиме и взаимодействовать с его входными и выходными потоками данных.

Смотрите ключевое слово `coproc`

Новый "mapfile" встроенный

`mapfile` Встроенный компонент способен отображать строки файла непосредственно в массив. Это позволяет избежать необходимости самостоятельно заполнять массив с помощью цикла. Это позволяет вам определить диапазон строк для чтения и, при необходимости, вызвать обратный вызов, например, для отображения индикатора выполнения.

Смотрите: встроенная команда `mapfile`

Изменения в ключевом слове "case"

`case` Конструкция понимает два новых терминатора списка действий:

`;&` Терминатор заставляет выполнение продолжить со следующим списком действий (вместо завершения `case` конструкции).

`;;&` Терминатор заставляет `case` конструкцию тестировать следующий заданный шаблон вместо завершения всего выполнения.

Смотрите Описание случая

Изменения во встроенном "объявлении"

Теперь `-p` опция выводит все атрибуты и значения объявленных переменных (или функций, при использовании с `-f`). Выходные данные полностью можно повторно использовать в качестве входных данных.

Новая опция `-l` объявляет переменную таким образом, что содержимое преобразуется в нижний регистр при присваивании. Для верхнего регистра то же самое относится к `-u`. Опция `-s` приводит к тому, что содержимое должно быть заглавным перед назначением.

`declare -A` объявляет ассоциативные массивы (см. Ниже).

Изменения во встроенном "read"

`read` Встроенная команда имеет несколько интересных новых функций.

`-t` Параметр для указания значения тайм-аута был слегка изменен. Теперь он принимает дробные значения и специальное значение 0 (ноль). Когда `-t 0` указано, `read` немедленно возвращается со статусом выхода, указывающим, ожидают ли данные или нет. Однако, когда задается тайм-аут `read` и время ожидания

встроенного тайм-аута истекает, любые частичные данные, полученные до тайм-аута, сохраняются в данной переменной, а не теряются. Когда истекает время ожидания, `read` завершается с кодом, превышающим 128.

Была введена новая опция, `-i`, для предварительной загрузки входного буфера некоторым текстом (при использовании `Readline`, с `-e`). Пользователь может изменить текст или нажать `return`, чтобы принять его.

Смотрите встроенную команду `read`

Изменения во встроенной "справке"

Сама сборка не сильно изменилась, но отображаемые данные теперь более структурированы. Тексты справки в лучшем формате, их гораздо легче читать.

Есть две новые опции: `-d` отображает краткое изложение текста справки, `-m` отображает формат, подобный `manpage`.

Изменения во встроенном "ulimit"

Помимо использования размера блока 512 байт везде в режиме POSIX, `ulimit` поддерживает два новых ограничения: `-b` для максимального размера буфера сокета и `-T` для максимального количества потоков.

Расширения

Расширение скобки

Расширение в виде фигурных скобок было настроено так, чтобы при запросе ряда чисел результаты расширения отображались с начальными нулями.

См. [Расширение скобки](#)

Расширение параметров

Добавлены методы для изменения случая по времени расширения.

Во время расширения вы можете изменить синтаксис, добавив операторы к имени параметра.

См. [Изменение регистра при расширении параметров](#)

Расширение подстроки

При использовании расширения подстроки для позиционных параметров начальный индекс 0 теперь приводит к добавлению \$ 0 к списку (если используются позиционные параметры). Раньше это расширение начиналось с 1 доллара:

```
# это должно отображать 0 долларов на Bash v4, 1 доллар на Bash v3
echo $ {@: 0:1}
```

Глобализация

Появилась новая опция оболочки `globstar`. При включении Bash будет выполнять рекурсивную глобализацию `**` – это означает, что он сопоставляет все каталоги и файлы с текущей позиции в файловой системе, а не только с текущим уровнем.

Новая опция оболочки `dirspell` позволяет исправлять орфографию имен каталогов во время глобализации.

См. раздел [Расширение имени пути \(глобализация\)](#)

Ассоциативные массивы

Помимо классического метода целых индексированных массивов, Bash 4 поддерживает ассоциативные массивы.

Ассоциативный массив - это массив, индексированный произвольной строкой, что-то вроде

```
объявить -ДОЦ

ASSOC[First]="первый элемент"
ASSOC[Привет]= "второй элемент"
ASSOC [Питер Пэн]= "Странный парень"
```

Смотрите [Массивы](#)

Перенаправление

Существует новый `&>>` оператор перенаправления, который добавляет стандартный вывод и стандартную ошибку в именованный файл. Это то же самое, что и старая добрая `>>FILE 2>&1` нотация.

Синтаксический анализатор теперь понимает `|&` как синоним `2>&1 |`, который перенаправляет стандартную ошибку для команды через канал.

См. [Перенаправление](#)

Интересные новые переменные оболочки

Переменная	Описание
------------	----------

Переменная	Описание
BASHPID	содержит PID текущей оболочки (это отличается от того, что \$\$ делает!)
PROMPT_DIRTRIM	задает максимальное количество не сокращенных элементов пути в приглашении
FUNCNEST	контролируйте максимальное количество рекурсий функций оболочки

Смотрите Специальные параметры и переменные оболочки

Интересные новые опции оболочки

Упомянутые параметры оболочки **отключены по умолчанию**, если не указано иное.

Вариант	Описание
checkjobs	проверяйте и сообщайте о любых запущенных заданиях при выходе из оболочки
compat*	установите режимы совместимости для более старых версий оболочки (влияет на сопоставление регулярных выражений в <code>[[...]]</code>)
dirspell	позволяет исправлять орфографию имен каталогов во время глобализации
globstar	включает рекурсивную глобализацию с <code>**</code>
lastpipe	(4.2) для выполнения последней команды в конвейере в текущей среде



Смотрите список опций оболочки

Разное

- Если команда не найдена, оболочка пытается выполнить названную функцию оболочки `command_not_found_handle`, предоставляя слова команды в качестве аргументов функции. Это можно использовать для отображения сообщений, дружественных пользователю, или для выполнения поиска по различным командам
- Поведение режима `set -e (errexit)` было изменено, теперь он работает более интуитивно (и лучше документирован на странице руководства).
- Цель вывода для функции `xtrace (set -x / set +x)` настраивается **начиная с версии Bash 4.1** (ранее она была исправлена `stderr`): переменная с именем `BASH_XTRACEFD` может быть установлена на `filedescriptor`, который должен получать выходные данные
- Bash 4.1 может записывать историю в системный журнал (только для включения во время компиляции в `config-top.h`)



Обсуждение

 bash4.txt  Последнее редактирование: 2018/02/03 07:52 автор narutowindy

Этот сайт поддерживается Performing Databases - вашими экспертами по администрированию баз данных

Bash Hackers Wiki



За исключением случаев, когда указано иное, контент на этой вики лицензируется по следующей лицензии:
Лицензия GNU Free Documentation 1.3