
Shell Scripting 4

Аритметика. Масиви. Работа с параметри в скриптовете.

Атрибути на променливи

- По подразбиране променливите в shell скриптирането съдържат текстов низ (string)
 - Може да се зададат и други типове за променливите с помощта на задаването на атрибути за променлива с declare
-

Целочислени променливи

- Ако искаме една променлива да може да получава само целочислени стойности:
 - declare -i num
 - Така променливата num вече може да съдържа само цели числа.
 - Ако зададем друга стойност, няма да се появи грешка, а стойността на num ще бъде зададена на 0.
 - За премахване на атрибута и връщане на поведението по подразбиране:
 - Declare +i num
 - За целите на аритметиката е нужно променливата да е с атрибут за число.
-

C-style аритметични изрази

- В shell се поддържат аритметични изрази, познати от програмирането.
 - За целта се използва let командата
 - `let n=30/3`
 - Изразите могат да се запишат и в двойка кръгли скоби:
 - `((++x))`
 - `((p=x / 100))`
 - Ако се нуждаем от стойността на даден израз, може да запишем и `$((...))`
 - `p=$((300 / 5))`
-

C-style аритметични изрази (2)

- Ако една променлива е зададена като цяла, може да запишем аритметичен израз за нейна стойност и в кавички:
 - `num="30%8"`
 - Ако една променлива участва в аритметичен израз, не е нужно да бъде поставена в кавички.
 - Ако желаем, може да използваме аритметични изрази в `if` и `while`.
-

Константни (read-only) променливи

- Може да създадете променливи със статут на константни (read-only), които да не могат да бъдат променени, веднъж щом са създадени.
 - Създаването на такава константа се случва по следния начин: `declare -r constant="value"`
-

Експортиране на променливи

- По подразбиране променлива създадена в даден скрипт е локална за скрипта (или локална за сесията на терминала)
- Дадена променлива може да бъде експортирана, за да може да се използва в някой подпроцес (подскрипт)
- Променливата не може да бъде експортирана към викащ скрипт или програма.
- Атрибутите на дадена променлива не могат да се експортират.
- Променлива може да се експортира с помощта на `export` командата:

- `export message="I'm exported"`

Масиви

- В shell се поддържат масиви по сходен начин с познатите до момента от програмирането масиви.
 - Масивът е структура, която съдържа множество данни.
 - Данните се съхраняват и извличат чрез индекси, започващи от 0.
 - Пример:
 - `x[0]="hello"`
 - `x[1]="world"`
 - Извличане: `${x[0]}`
 - Извличане на всички елементи: `${x[*]}`
-

Масиви (2)

- Инициализиране на масив:
 - `ar=(1 3 2 5 6 9)`
 - Брой на елементи в масив/дължина на низ:
 - `${#array[@]}`
 - Списък с индексите в масива:
 - `${!array[@]}`
 - Възможно е да има липсващи индекси - не е задължително индексите да са последователни!
 - В по-новите си версии `bash` поддържа асоциативни масиви
-

Параметри на скрипт

- Често при извикването си скриптовете се нуждаят от параметри.
 - Някои от тези параметри могат да са задължителни, други - по избор.
-

Позиционни параметри

- Задават се при извикването на скрипта, като се номерират от 1 нататък.
 - \$1, \$2, \$3... и т.н., над 9 се отбелязват с къдрави скоби: \${10}
 - \$0 съдържа името на скрипта според извикването.
 - \$@ - съдържа всички аргументи в себе си.
 - \$# - съдържа броя на подадените аргументи.
-

shift

- Премахва първия аргумент, като останалите се преномерират (2 става 1, 3 става 2 и т.н.).
 - \$# също се намаля с 1.
 - Полезно при циклична обработка на аргументите с while.
-

Getopts

- Инструмент, който позволява на скрипта да приема допълнителни опции, които започват с тиренце (пример: -x).
 - Всяка опция може да приема допълнителни аргументи (пример: -f filename)
 - Прочетете повече в Интернет:
<https://www.computerhope.com/unix/bash/getopts.htm>
 - <https://www.shellscript.sh/tips/getopts/>
-

Благодаря за вниманието

Автор:

П. Р. Петров - преподавател по професионална
подготовка по Програмиране в ПГЕЕ “К. Фотинов”, гр.
Бургас
