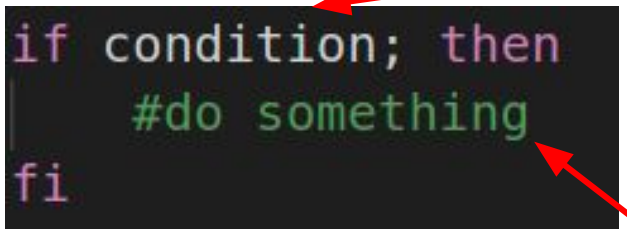
Shell Scripting 2

Условни конструкци

Условна конструкция - if

- Действа подобно на познатия if от C#
- Синтаксис:



Условие, което се оценява на вярно или невярно

Кодът тук се изпълнява само, ако условието е вярно.

A else?

- Shell разполага и с else, за да поддържаме възможност за изпълнение на код при невярно условие:

```
if condition; then
    #do something
else
    #do something else!
fi
```

Return codes

- При преключване, изпълнимите файлове връщат код на изпълнението към ОС.
- Този код показва дали програмата е завършила успешно или с грешка.
- Кодовете са от 0 до 255, като 0 показва успех.
- Може да върнете код чрез exit командата:
  - exit 0

Условни изрази

- Ограждат се в двойка квадратни скоби:

Израз	Значение
[[$var]]	Проверява дали var не е празна.
[[$var = "hello"]]	Проверява дали var има стойност "hello"
[[$var="hello"]]	Присвоява стойността, връща винаги true!
[[-e $filename]]	Проверява дали съществува файл с име съвпадащо със стойността на filename

Забележете как в единия случай около = има интервали, а в другия не!

Сравнение на числа

- Формат: [[arg1 OP arg2]], където OP е:
  - -eq: проверка за равенство
  - -lt: проверка за по-малко
  - -gt: роверка за по-голямо
  - -le: проверка за по-малко или равно
  - -ge: проверка за по-голямо или равно
  - -ne: проверка за различие
- Не бива да се ползват традиционните оператори =, <, > и т.н.

Вложени if-ове

- If-овете могат да се влагат, както и да се разглежда поредица от свързани проверки.
- Пример: Вложен if

```
if [[ ! -d $bindir ]]; then
    # if not: create bin directory
    if mkdir "$bindir"; then
        echo "created ${bindir}"
    else
        echo "Could not create ${bindir}."
        exit 1
    fi
fi
```

elif

- Проверява всяко едно от условията до срещане на вярното
- Ако всички са неверни изпълнява else

```
if [[ $1 = "cat" ]]; then
    echo "meow"
elif [[ $1 = "dog" ]]; then
    echo "woof"
elif [[ $1 = "cow" ]]; then
    echo "mooo"
else
    echo "unknown animal"
fi
```

По-сложни условия

- Понякога е нужно да правим по-сложни проверки, например за няколко условия едновременно.
- Може да реализирате проверка с and (и) по следния начин:
  - [[condition1 && condition2]]
- Може да реализирате проверка с or (или) по следния начин:
  - [[condition1 || condition2]]
- Може да направите условие базирано на отрицание с !:
  - [[! condition]]

echo

- Командата echo извежда на стандартния изход параметрите, които са подадени, като след тях поставя нов ред.
- Ако желаете да няма нов ред след тях използвайте -n след командата

printf

- Командата printf може да извежда форматиран низ, като не задава след него нов ред по подразбиране.
- https://www.geeksforgeeks.org/printf-command-in-linux-with-examples/

Потоци от данни

- Входен (stdin) - означение: 0, /dev/stdin
- Изходен (stdout) - означение: 1, /dev/stdout
- За грешки (stderr) - означение: 2, /dev/stderr
- /dev/null - унищожава всякаква информация насочена към него

Приемане на входни данни от друго място

- Може да приемате входни данни от друго място, а не от stdin, чрез оператор <:
  Пример: grep pesho < names.txt

Изпращане на изходни данни към друго място

- Може да изпращате изходни данни към друго място, а не към stdout, чрез оператор >:
  Пример: ls ~ > filelist.txt
- > операторът презаписва файла, ако вече съществува или го създава, в противен случай.
- >> операторът записва към края на файла, ако той съществува

Насочване към конкретен поток

- Насочване на конкретен поток към дадено място с помощта на N> :
  - команда 2> $HOME/error.log
- Насочване към конкретен поток с помощта на >&N
  - >&2 (насочва изхода към stderr, може да се запише и 1>&2)
  - 2>&1 насочва stderr към stdout
- Насочване и на stdout, и на stderr към един файл:
  - команда > logfile 2>&1, където logfile е път/име на файл, в който да се запазват данните

Pipes

- Можете да приложите изхода от една команда като входни данни за друг с помощта на pipe, това става чрез оператор |
- Пример:
  - ls | grep file.txt

Благодаря за вниманието

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