

Bash scripting



BASH

THE BOURNE-AGAIN SHELL

Sadržaj

- Uvod
- Kontrole toka
- Prosleđivanje parametara skripti
- Funkcije
- Regularni izrazi
- Crontab
- Finalna skripta

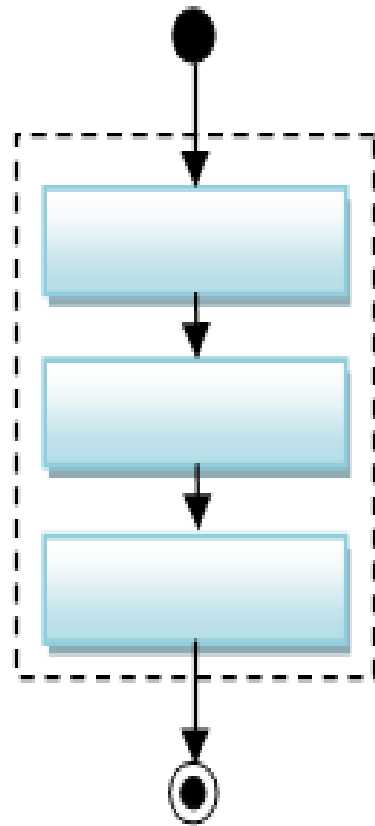
Uvod

- `#!/bin/bash`
- Demo: inic skripta
 - Kreira direktorijum i par fajlova u tekucem direktorijumu

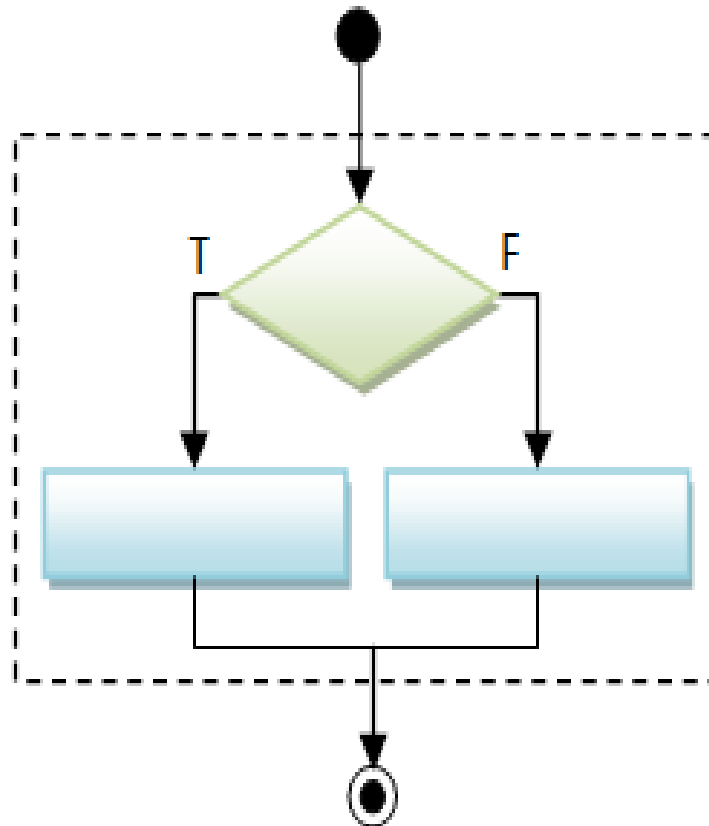
Uvod

- pipes |
- Demo: infoGather skripta
 - Pretpostavka: imamo root pristup nekom Linux serveru
 - Skripta će prikupiti neke zanimljive informacije
- ssh hacker@server 'bash -s' < [infoGather.sh](#)

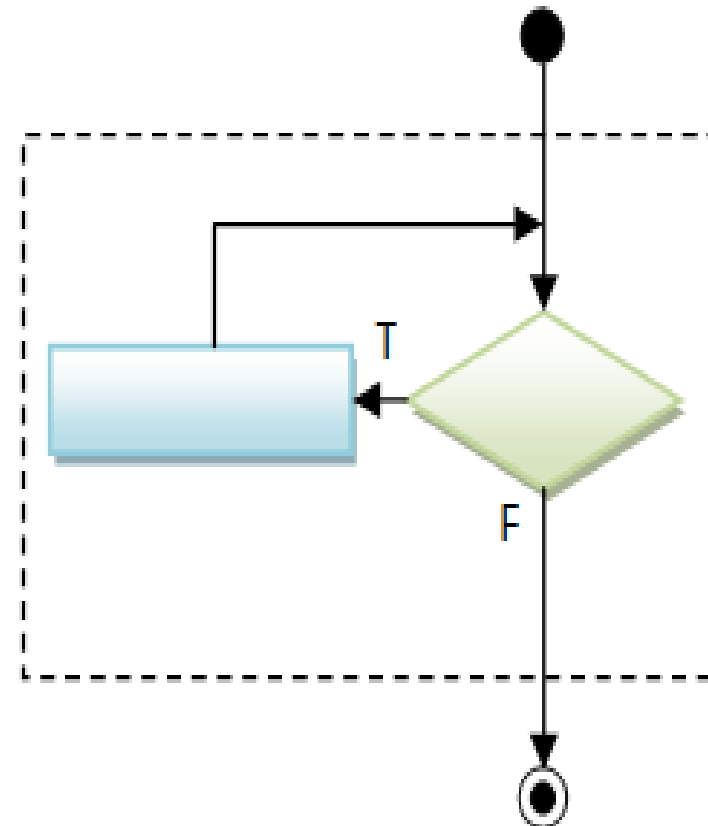
Kontrola toka



Sequential



Conditional (Decision)



Loop (Iteration)

IF

```
[ $var = 5 ]; echo $?  
test $var = 5; echo $?
```

- Rezultat obrnut od očekivanog
- Za numeričko poređenje: -gt -lt -eq -geq -leq
- -a za and, -o za or

```
[ -f f1.txt ]; echo $?
```

- -f file, -d dir, -z zero, -r read, -x exec

```
if [ $var -eq 1 ]  
then  
    cat f1.txt  
else  
    cat f2.txt  
fi
```

- Demo: skripta rootCheck
 - Skripta koja proverava da li korisnik koji ju je pozvao ima root privilegije

FOR

```
for i in $(cat f1.txt)
do command
done
```

- Demo: skripta pingList
 - Pozvaćemo komandu ping za spisak domena iz fajla domains.txt

- Mogucnosti za for petlju:

```
for i in $(cat f1.txt)
for i in {1..5}
for i in {0..10..2}
for (( c=1; c<=5; c++ ))
for (( ; ; ))
```

- IFS - Internal Field Separator
- default value = `<space><tab><newline>`
- ako želimo da nešto drugo bude delimiter

```
oldIFS=$IFS
IFS=:
```

WHILE

```
v=1  
while [ $v -lt 5 ]  
do  
    echo $v; let v++;  
done
```

Prosleđivanje parametara

- \$1, \$2 - parametri prosleđeni skripti
- \$0 - komanda za pokretanje
- \$# - broj prosleđenih parametara
- \$@ - dohvata sve parametre (zna koji su odvojeni)
- \$* - svi parametri spojeni u string

- Demo: nmapRead skripta
 - interaktivno uzimanje i prosleđivanje parametara Nmap-u
 - `nmap -sT 147.91.79.140- 147.91.79.143 -p 80 -oG res.log`
- Demo: scriptStarter skripta
 - pokreće skripte koje se nalaze u prosleđenom direktorijumu

Opcioni parametri

```
while getopts a:b opt; do
  case $opt in
    a) param = $OPTARG;;
    b) echo "op -b"
        b=1;;
    *) echo "ERROR"
        exit -1;;
  esac
done
```

Funkcije

```
function function_name {  
    <commands>  
}
```

```
function_name() {  
    <commands>  
}
```

- Argumenti se šalju kroz \$1, \$2 ...
- Povratna vrednost se šalje kroz \$?

Regularni izrazi (Regex)

- . - jedan karakter, * - nula ili nekoliko karaktera, ? - prethodni karakter 0 ili 1
- {3} - tačno tri karaktera
- [a-d]{4} - četiri karaktera, svaki iz skupa {a,b,c,d}
- Sed - Stream Editor
- Razlika Sed vs Awk?

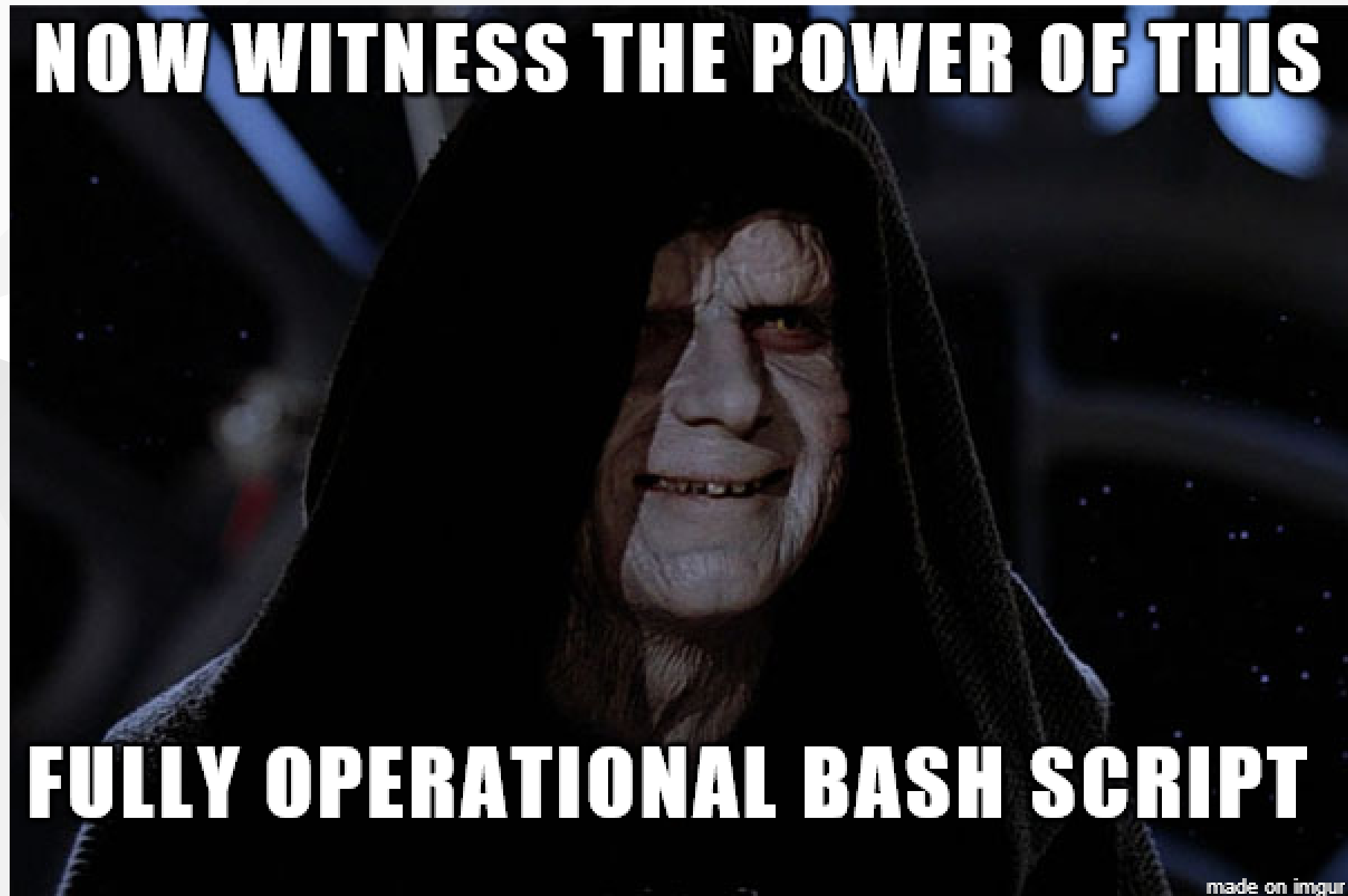
Crontab

- Zakazivanje izvršavanja skripti
- crontab -l
- crontab -e

The diagram illustrates the components of a cron job entry. On the left, a sample entry is shown: `# * * * * * command to execute`. The asterisks are annotated with vertical lines and labels:
 - The first asterisk is labeled "min (0 - 59)".
 - The second asterisk is labeled "hour (0 - 23)".
 - The third asterisk is labeled "day of month (1 - 31)".
 - The fourth asterisk is labeled "month (1 - 12)".
 - The fifth asterisk is labeled "day of week (0 - 7)".
 On the right, the text "CRON job" is displayed in large blue letters, with a clock icon integrated into the letter "O".

Demo: specScan skripta

- Skriptu može da pokrene samo korisnik sa root privilegijama
- Skripta se pokreće sa: `./specScan website.com -o -r report.txt -s`
- `-o` je opcioni parametar za skeniranje OS na serveru na kom je prosleđen sajt
- `-s` je opcioni parametar za pokretanje pajton programa Sublister koji pronalazi podomene



Bash vs Python

- Bash skripte ce biti krace nego kratki Pajton programi
- Bash skripte ce biti duze od duzih Pajton programa
- Brzina, isti princip
- Pipes
- Malo grananja u programu
- Automatizacija
- Nisu potrebne strukture podataka
- EH mora znati oba!