

Keuzevak Linux - Week 6

Paul Sohier Kevin van der Vlist

Versie 1.0

29 november 2011

Inhoud

- 1 Wat en waarom
- 2 Variabelen
- 3 Systeemvariabelen
- 4 if
- 5 if/else
- 6 if/elseif/else
- 7 case/switch
- 8 while
- 9 until
- 10 functies
- 11 functievariabelen
- 12 Expressies

Shell scripting - wat en waarom

- reeks opdrachten

Shell scripting - wat en waarom

- reeks opdrachten
- controle structuren

Shell scripting - wat en waarom

- reeks opdrachten
- controle structuren
- automatiseren taken

Shell scripting - wat en waarom

- reeks opdrachten
- controle structuren
- automatiseren taken
- backups, gebruikers aanmaken, ...

Shell scripting - begin

- shebang: `#!/bin/bash`

Shell scripting - begin

- shebang: `#!/bin/bash`
- opdrachten of controle structuren

Shell scripting - begin

- shebang: `#!/bin/bash`
- opdrachten of controle structuren
- eindresultaat

Shell scripting - variabelen

- loosely typed

Shell scripting - variabelen

- loosely typed
- refereren met `$naam`

Shell scripting - variabelen

```
1 | #!/bin/bash
2 | var=2
3 | var2=5
4 | drie=" drie"
5 | echo $var2
6 | echo "$var + $drie = $var2"
```

```
1 | 3
2 | 2 + 3 = 5
```

Shell scripting - systeemvariabelen

- ingesteld door omgeving

Shell scripting - systeemvariabelen

- ingesteld door omgeving
- `export EDITOR=emacs`

Shell scripting - systeemvariabelen

- ingesteld door omgeving
- `export EDITOR=emacs`
- simpel te gebruiken

Shell scripting - systeemvariabelen

- \$PATH
- \$TERM
- \$EDITOR
- \$DISPLAY
- ...

Shell scripting - if

```
1 | if [ "Pasen" == "Pinksteren" ];  
2 | then  
3 |     echo "sint-juttemis"  
4 | fi
```

Shell scripting - if/else

```
1 | if [ "appel" == "banaan" ];  
2 | then  
3 |     echo "onbereikbaar"  
4 | else  
5 |     echo "appel en banaan zijn ongelijk"  
6 | fi
```

Shell scripting - if/elseif/else

```
1 if [ "geluid" == "hard" ];
2 then
3     echo "geluid is hard"
4 elseif [ "geluid" == "zacht" ];
5 then
6     echo "geluid is zacht"
7 else
8     echo "geluid is niet hard en niet zacht"
9 fi
```

Shell scripting - case/switch

```
1 functie="student"
2 case "$functie" in
3     "student")
4         echo "student" ;;
5     "administratie")
6         echo "administratie" ;;
7     "docent")
8         echo "docent" ;;
9     *)
10        echo "geen match" ;;
11 esac
```

Shell scripting - while

```
1 i=0
2 while [ "$i" -lt "5" ]; do
3     let i++
4     echo -n $i:
5 done
6 1:2:3:4:5:
```

Shell scripting - until

```
1 | i=10
2 | until [ "$i" -lt "5" ]; do
3 |     let i--
4 |     echo -n $i:
5 | done
6 | 9:8:7:6:5:4:
```

Shell scripting - functies

```
1 | function hallofunc {  
2 |  
3 | }
```

```
1 | hallofunc () {  
2 |  
3 | }
```

Shell scripting - functievariabelen

```
1 | function hallofunc {  
2 |     echo hallo $1  
3 | }  
4 |  
5 | hallofunc kevin
```

```
1 | hallofunc() {  
2 |     echo hallo $1  
3 | }  
4 | hallofunc kevin
```


Shell scripting - string expressies

Expressie

Waar als. . .

-z string

de lengte van de string 0 is

-n string

de lengte van string niet 0 is

string1 == string2

de twee strings gelijk zijn

string1 != string2

de twee strings ongelijk zijn

string

de string niet NULL is

```
1| if [ -z "$string" ]; then echo "lengte is 0"; fi |
```

Shell scripting - integer expressies

expressie	waar als...
-----------	-------------

<code>int1 -eq int2</code>	<code>1 == 2</code>
----------------------------	---------------------

<code>int1 -ne int2</code>	<code>1 != 2</code>
----------------------------	---------------------

<code>int1 -gt int2</code>	<code>1 > 2</code>
----------------------------	-----------------------

<code>int1 -ge int2</code>	<code>1 ≥ 2</code>
----------------------------	--------------------

<code>int1 -lt int2</code>	<code>1 < 2</code>
----------------------------	-----------------------

<code>int1 -le int2</code>	<code>1 ≤ 2</code>
----------------------------	--------------------

```
1| if [ "1" -lt "2" ]; then echo "1 < 2"; fi |
```

Shell scripting - file expressies

Expressie	Waar als...
-e file	file bestaat
-r file	file bestaat en is readable
-w file	file bestaat en is writeable
-x file	file bestaat en is executable
-f file	file bestaat en is een gewoon bestand
-d file	file bestaat en is een directory
-h file	file bestaat en is een symbolic link
-s file	file bestaat en de grote is meer als 0

```
1| if [ -e "file" ]; then echo "file bestaat"; fi |
```