

# BASH 1/3



Mattias Wecksten

2010



---

---

---

---

---

---

---

---

# Scripting

- A script is a number of commands stored in a text file that are supposed to be executed in sequence.
- Functionality
  - Read data
  - Process data
  - Produce results
- Automation of repetitive tasks
  - Same result every time
- Don't forget to set the executable flag (chmod 755).

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

# Shell script flavors

- sh
- bash
- csh
- tcsh
- ksh
- zsh
- sh (bourne shell) is not the easiest or best one to use, but similarly to vi the bourne shell is widely available, which makes your shell scripts quite portable.

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## Bourne shell skeleton

```
#!/bin/sh

# TO-DO: Fill in your script
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## The Hello world!

```
#!/bin/sh
# Name: Hello world!
#
# Prints "Hello world!" to the terminal.
echo "Hello world!"
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## A simple script

```
#!/bin/sh
# Name: logg
#
# 1.Copy the file logg
# 2.List all lines with WARNING
cp ./tmp/logg ./backup
cat ./tmp/logg | grep WARNING
```

Mattias Wecksten

2010



---

---

---

---

---

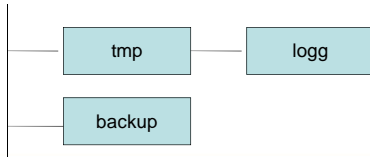
---

---

---

## A simple script

```
cp ./tmp/logg ./backup  
cat ./tmp/logg | grep WARNING
```



Mattias Wecksten

2010



---

---

---

---

---

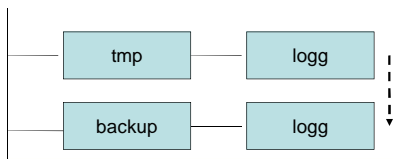
---

---

---

## A simple script

```
cp ./tmp/logg ./backup  
cat ./tmp/logg | grep WARNING
```



Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## A simple script

```
cp ./tmp/logg ./backup  
cat ./tmp/logg | grep WARNING
```

```
192342 STATUS OK  
192343 STATUS OK  
192344 WARNING - INTRUDER ALERT 192.47.66.2  
192345 STATUS OK  
192346 STATUS OK  
192347 WARNING - INTRUDER ALERT 192.47.47.2  
.....
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## A simple script

```
cp ./tmp/logg ./backup  
cat ./tmp/logg | grep WARNING
```

```
192344 WARNING - INTRUDER ALERT 192.47.66.2  
192347 WARNING - INTRUDER ALERT 192.47.47.2
```



---

---

---

---

---

---

---

---

## Variables

- A variable is a value in a register
- The value could be text or numbers
  - Simplifies if you want to replicate settings.
  - Could be used to communicate between scripts.
- The special character \$ indicates the USE of a variable.
  - Today is \$day



---

---

---

---

---

---

---

---

## Variables

- Variables are variable
- To save data give
  - The variable name
  - = (equals)
  - The value
- If the variable exists it is overwritten.
  - Otherwise created.



---

---

---

---

---

---

---

---

## Variables

- Create

```
var1="En text!"
var2=/usr/bin
var3=apple
var4=En text!
```
- Use

```
echo $var1           -> echo "En text!"
ls $var2             -> ls /usr/bin
echo "The $var3 is a fruit" -> The apple is a fruit
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## Simple backup

```
#!/bin/sh
# Name: backup
#
# Copies the file /tmp/logg to /home/backup

cp /tmp/logg /home/backup
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## Simple backup using variables

```
#!/bin/sh
# Name: var_backup
#
# Copies the file /tmp/logg to /home/backup

backup_path=/home/backup
cp /tmp/logg $backup_path
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## Interactive backup (keyboard input)

```
#!/bin/sh
# Name: int_backup
#
# Copies a named file to /home/backup

backup_path = /home/backup
echo "File to save:"
read filename
cp $filename $backup_path
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## Executing scripts

```
lygos% chmod +x backup
lygos% sh backup
lygos% sh -x backup
lygos% backup
```



Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## Arguments to the script

```
#!/bin/sh
# Name: arg_print
#
# Print the arguments 0-4

echo "The script has $# arguments."
echo $0
echo $2 - $4 !!!
echo $3 - $1
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## Arguments to the script

```
lygos.hh.se% arg_print arg1 "arg 2" arg3 arg4
The script has 4 arguments.
arg_print
arg 2 - arg4 !!!
arg3 - arg1
lygos.hh.se% _
```



---

---

---

---

---

---

---

---

## Safe remove (trash can)

```
#!/bin/sh
# Name: srm [FILE]
#
# Backup files, then rm.

backup_path=/tmp
cp $1 $backup_path
echo $1 is stored in $backup_path
rm $1
```



---

---

---

---

---

---

---

---

## Safe Remove

```
lygos% srm jobb.txt
jobb.txt is stored in /tmp
lygos% _
```



---

---

---

---

---

---

---

---

## Debugging

```
#!/bin/sh -x

echo Runs three programs...

good_program
instable_program
well_written_program

echo Finished
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## Debugging

```
lygos.hh.se% runner
Runs three programs...
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## Debugging

```
lygos.hh.se% runner
Runs three programs...
+ echo Runs three programs...
+ good_program
+ instable_program
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---



# Flow Control



for  
while  
if  
case

Mattias Wecksten 2010



---

---

---

---

---

---


---

---

# Flow Control

- Repeat
  - for
  - while
- Select
  - if
  - case

Mattias Wecksten 2010



---

---

---

---

---

---


---

---

# for

```
for var in element...  
do  
#  
done  
  
for i in 1 2 3 4 5  
do  
echo "i = $i"  
done
```

Mattias Wecksten 2010



---

---

---

---

---

---

---

---

## while

```
while [ expression ]
do
#
done

while [ "$command" != "quit" ]
do
    read command
done
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## if

```
if [ expression ]; then
# true
else
# false
fi

if [ $day = "friday" ]; then
    echo "Dress casual."
else
    echo "Dress strict!"
fi
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## if

```
if [ expression ]; then
#
elif [ expression ]; then
#
else
#
fi
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## if

```
if [ $day = "friday" ]; then
  echo "Dress casual."
elseif [ $day = "saturday" ]; then
  echo "Dress."
else
  echo "Dress strict!"
fi
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## if

```
if [ $temp -lt 18 ]; then
  echo "A bit chilly."
else
  echo "Aaaaah. Comfortable."
fi
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## case

```
case $variable in
  pattern)
    #
    ;;
  *)
    #
    ;;
esac
```

Mattias Wecksten

2010



---

---

---

---

---

---

---

---

## case

```
case $action in
  start)
    echo "Initializing..."
    my_server start
    ;;
  *)
    echo "Restarting..."
    my_server stop
    my_server start
    ;;
esac
```



---

---

---

---

---

---

---

---