



# Scripting in KSH & BASH

...FOR DUMMIES :D

# Type of lines in a script

- ▶ First line of the script (it defines the interpreter):

```
#!/usr/bin/ksh      (#!/usr/bin/bash or for bash scripts)
```

- ▶ Comment lines begin with #:

```
# This is a comment
```

- ▶ Regular code:

```
if [[ $NAME != 'TEST' ]]; then  
    echo 'This is a test'  
fi
```

# Variables management

## ► Definition of variables:

```
ALIAS_HOST=SERVER_HBA1
ALIAS_ARRAY=ARRAY_PORT1
COUNTER=0
ZONENAME=          # Empty
```

## ► Output of variables:

```
$ echo ${ALIAS_HOST}
SERVER_HBA1
$ print ${COUNTER}
0
```

## ► Modification of string variables:

```
$ ZONENAME=${ALIAS_HOST}_${ALIAS_ARRAY}
$ echo ${ZONENAME}
SERVER_HBA1_ARRAY_PORT1
```

## ► Arithmetic operations on integer variables:

```
$ (( COUNTER = COUNTER + 1 ))
$ echo ${COUNTER}
1
```

# Getting output from execution

- ▶ We can assign a variable the value returned from a command:

```
$ ps -ef | grep -c horcmd_  
6  
  
$ HORCM_INSTANCES="$(ps -ef | grep -c horcmd_)"  
$ echo ${HORCM_INSTANCES}  
6
```

- ▶ This can be seen in older scripts using this syntax:

```
$ HORCM_INSTANCES=`ps -ef | grep -c horcmd_`
```

# Branches (1...)

- ▶ If condition is true, then...:

```
if [[ ${ALIAS_HOST} == 'SERVER_HBA1' ]]; then
    print 'The value of the alias is the expected one'
fi
```

- ▶ If condition is true, then...; if not, then...:

```
if [[ ${ALIAS_HOST} == 'SERVER_HBA1' ]]; then
    print 'The value of the alias is the expected one'
else
    print 'The value of the alias is NOT the expected one :('
fi
```

# Branches (2...)

- If condition1 is true, then...; if condition2 is true, then...; if not, then...:

```
if [[ ${ALIAS_HOST} == 'SERVER1_HBA1' ]]; then
    print 'The value of the host alias is the expected one'
elif [[ ${ALIAS_HOST} == 'SERVER2_HBA1' &&
        ${ALIAS_ARRAY} == 'ARRAY1_PORT1' ]]; then
    print 'The values are the expected ones'
else
    print 'NONE of the expected values are the expected ones :('
fi
```

# Branches (3)

- ▶ Pattern matching of variables:

```
case ${ALIAS_HOST} in
    'SERVER_HBA1')
        print 'The value of the host alias is the expected one'
        ;;
    'SERVER_HBA2')
        print 'The value of the host alias is SERVER_HBA2'
        ;;
    *)
        print 'The value of the host alias is wrong'
        ;;
esac
```

# Loops (1...)

- ▶ Iterate “while” condition is met:

```
while (( "${COUNTER}" <= 10 )); do
    print "Value of counter is ${COUNTER}"
    (( COUNTER = COUNTER + 1 ))
done
```

- ▶ Iterate “until” condition is met (opposite of previous):

```
until (( "${COUNTER}" > 10 )); do
    print 'We have not yet met the desired value...'
    (( COUNTER = COUNTER + 1 ))
done
```



# Loops (2)

- Iterate using a range of known values:

```
for i in 1 2 3 4 5; do
    print "Sleeping ${i} seconds"
    sleep ${i}
done
```

- Iterate using the output of a command as index:

```
print 'The list of instances in this host are:'
for INSTANCE in $(ls /etc/horcmd_* | cut -d'_' -f2); do
    print "${INSTANCE}"
done
```

# Functions

- ▶ Ability to group a common functionality for future reuse:

```
# This script will have ${PARAMETER} as its exit value
function exitValue {
    PARAMETER=$1          # First parameter of the function

    print 'This line will be printed'
    return ${PARAMETER}
}
```

```
$ exitValue 7
```

```
This line will be printed
```

```
$ echo $?
```

```
7
```

# Data redirection

- ▶ We can redirect the output from a command execution to a file:

```
$ symcfg list >/tmp/alma/symcfg_list.out 2>/tmp/alma/symcfg_list.err
```

> will redirect the “standard” output (overwriting the old content in the file)

2> will redirect the “error” output (it’s normally output to the screen as well)

>> will redirect the “standard” output (appending content to the file)

- ▶ We can keep the screen output and save to a file as well using the `tee` command:

```
$ symcfg list | tee /tmp/alma/symcfg_list.out
```

# Useful commands - grep

- ▶ Will show the lines that match the pattern given as argument:

```
$ ps -ef | grep horcmd_  
$ grep ${INSTANCE_NAME} /etc/horcm60.conf
```

- ▶ `-i` will ignore the case
- ▶ `-v` will show the lines that don't match the pattern
- ▶ `-c` will show the number of lines match the pattern
- ▶ `-B NUM` will show the matching line and the previous (Before) `NUM` ones
- ▶ `-A NUM` will show the matching line and the following (After) `NUM` ones

# Useful commands - sed

- ▶ Will perform “live” editing on the given file/input:

- ▶ `-e` will define the “regular expression” that will operate on the file

```
$ sed -e 's/OLD/NEW/g' /etc/horcm60.conf >/tmp/alma/horcm60.conf.new
```

- ▶ `s` will do a substitution
  - ▶ `g` will do the substitution on the whole file (default is to do it only in first occurrence)

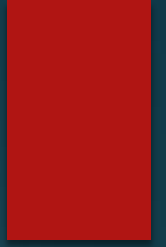
# Useful commands - awk

- ▶ Swiss-knife-like command, but we'll use it mainly for printing selected columns of text:

```
$ raidcom get ldev -ldev_list undefined -fx -I${INSTANCE} |  
awk '$1 == "LDEV" &&  
$6 ~ "6[[:xdigit:]][:xdigit:]][:xdigit:]]"'  
{print $3}'
```

- ▶ If the first column (`$1`) matches the string "LDEV"...
- ▶ ...and the sixth column (`$6`) matches a 4 digit number with the first one being 6 and the rest hexadecimal numbers, then...
- ▶ print the third column (`$3`)

# Examples





# Questions?

► Thank you!!!

