SEMINAR 2
BASH SCRIPTS

RECAP: RUNNING THE INTERPRETER

#!/USR/BIN/ENV BASH

/USR/BIN/SH

SYMLINK TO BOURNE SHELL

COMPATIBLE SHELL

TYPES OF QUOTES

- PLAIN TEXT: `'i love CAOS'`
- EXECUTE COMMAND: `` `uname -o` ``
- SUBSTITUTE VARIABLES: `"echo $my_var"`

VARIABLE SYNTAX

```
var1=123
var2=
var_class = 'CAOS'
var3="i love $var_class!"

echo '$var1' "$var2" "$var3" "$another_var"
```

LACK OF SPACES IS IMPORTANT!

```
os_name = `uname -o` # Wrong, won't work
```

COMMAND OUTPUT

```
os_name=$(uname -o)
arch=`uname -m`

echo "Running $os_name on $arch"
```

MAGIC VARIABLES

- $? RETURN CODE OF LAST COMMAND
- $0...$9 ARGUMENTS
- $# NUMBER OF ARGUMENTRS
- $@ ARGUMENT LIST
- $* STRING WITH LIST OF ARGUMENTS

FUNCTION DEFINITIONS

THREE WAYS OF DECLARING FUNCTIONS:

```
very_important_function() {
  # function definition
}

function very_important_function() {
  # function definition
}

function very_important_function {
  # function definition
}
```

ONLY FIRST IS COMPATIBLE WITH SH

FUNCTION ARGUMENTS ARE ACCESSIBLE THROUGH MAGIC VARIABLES

CALLING FUNCTIONS

CALLING SYNTAX:

```
value=$(very_important_function hello world)
very_important_function hello world
```

VARIABLES INSIDE FUNCTIONS ARE GLOBAL
BASH AND ZSH SUPPORT LOCAL KEYWORD

JUST LIKE WITH COMMANDS FUNCTION OUTPUT CAN BE PIPED

RETURN VALUE 0
MEANS SUCCESS
OR TRUTH

CONDITIONAL EXECUTION

- CMD1 && CMD2 EXECUTES CMD2 AFTER CMD1 FINISHED SUCCESSFULLY
- CMD1 || CMD2 EXECUTES CMD2 IF CMD1 HAS FAILED

TRUE AND FALSE ARE MAGIC PROGRAMS

- TRUE RETURNS 0
- FALSE RETURNS 1

IF STATEMENTS

```
if cmd
then
  # executed if cmd exited with zero
else
  # executed otherwise
fi
```

[COMMAND (AKA TEST)

```
if [ $x -eq $y ]
then
  # Do something when $x is equal to $y
elif [ $x -lt $y ]
then
  # Do something when $x is less than $y
fi
```

WHILE LOOP

```
while cmd
do
   # do something while return code is 0
done
```

FOR LOOP

```
for item in <list>
do
  # Do something with an item
done
```

IN BASH AND ZSH WE CAN USE FAMILIAR SYNTAX:

```
for (( i=0; i<10; i++ ))
do
  echo $i
done
```

STRING AS LIST

```
for item in $(echo "i love      CAOS")
do
  echo "$item"
done
```

IFS SPECIFIES SEPARATOR

```
IFS=':'
for item in $(echo 'i:love:CAOS')
do
  echo $item
done
```

SH CAN EXECUTE ARITHMETIC EXPRESSIONS INSIDE $(( ))

HOWEVER

ONLY INTEGERS ARE SUPPORTED

FOR FLOATING-POINT ARITHMETIC

USE BC -L

TEXT PROCESSING UTILITIES

- GREP FOR SEARCH/FILTRATION
- SED AND AWK FOR EDITING

DEMO

SEARCHING FOR FUNCTIONS IN MANPATH