C2110 *UNIX and programming*

Lesson 7 / Module 2

PS / 2020 Distance form of teaching: Rev2

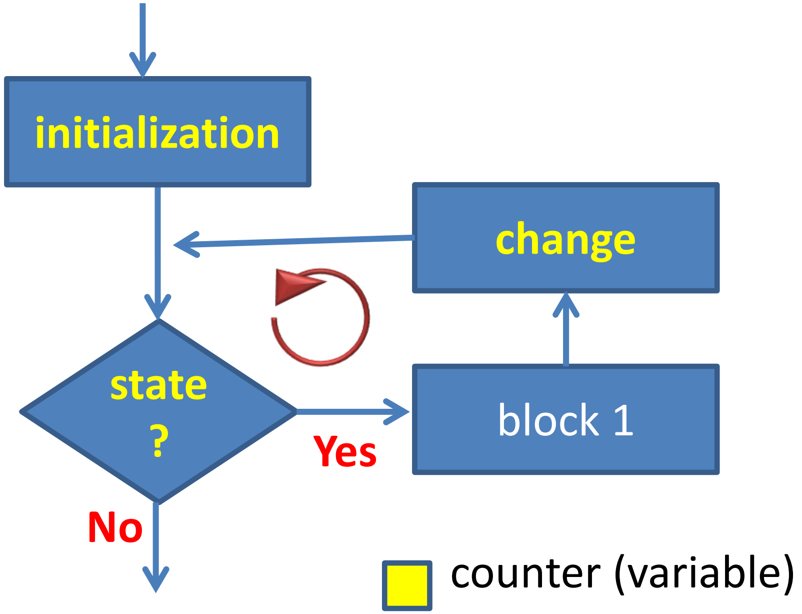
Petr Kulhanek

kulhanek@chemi.muni.cz

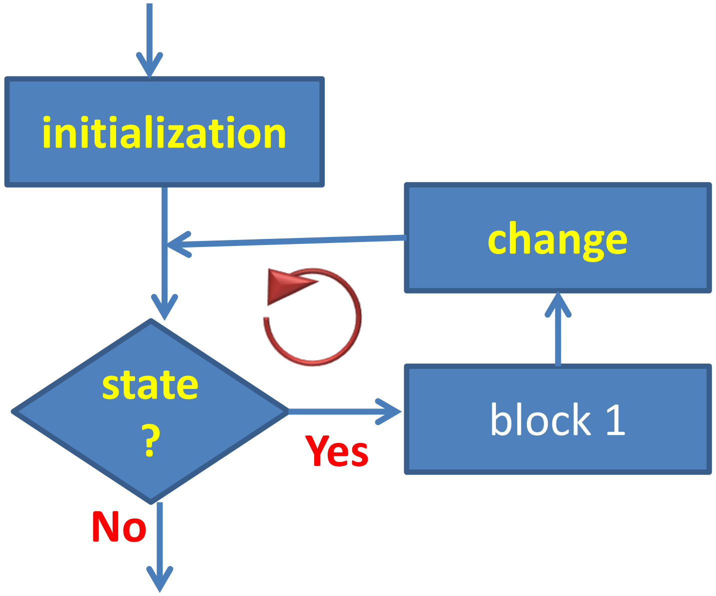
National Center for Biomolecular Research, Faculty of Science
Masaryk University, Kamenice 5, CZ-62500 Brno

Loops

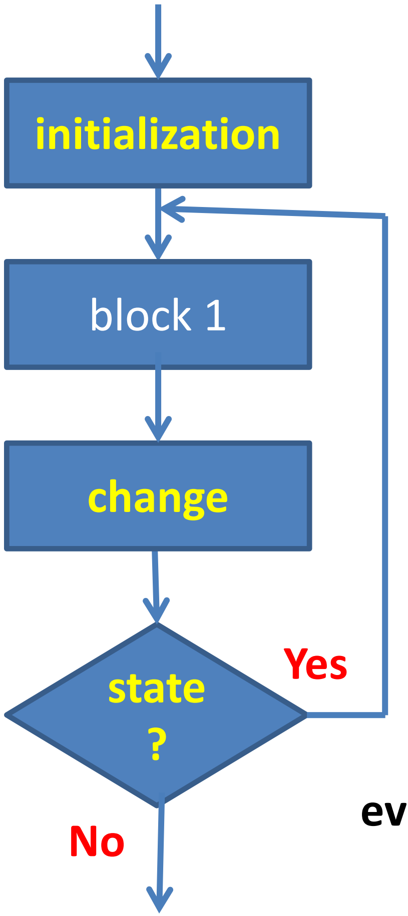
Loop block execution



Loop Using while/until ...

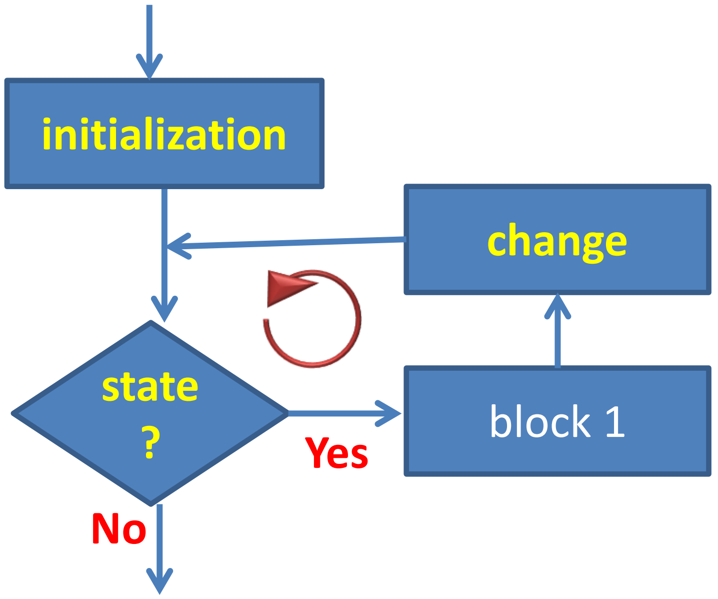


evaluates conditions **in the beginning** of the loop

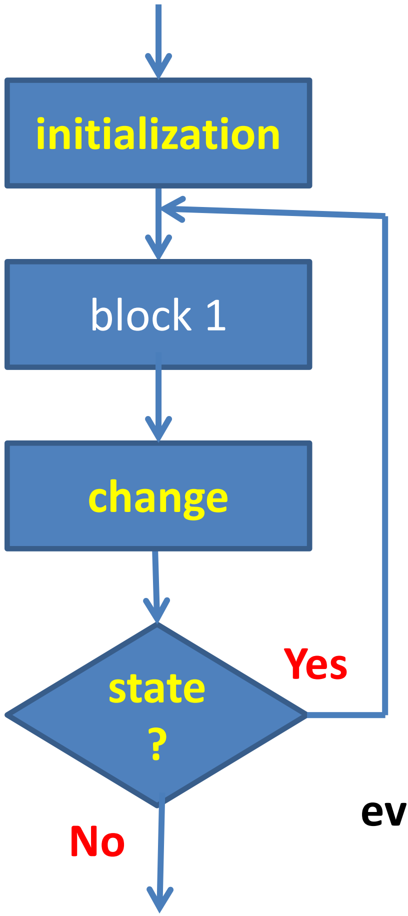


evaluates conditions **at the end** loop

Loop using while/until ...



evaluates conditions **in the beginning** of the loop

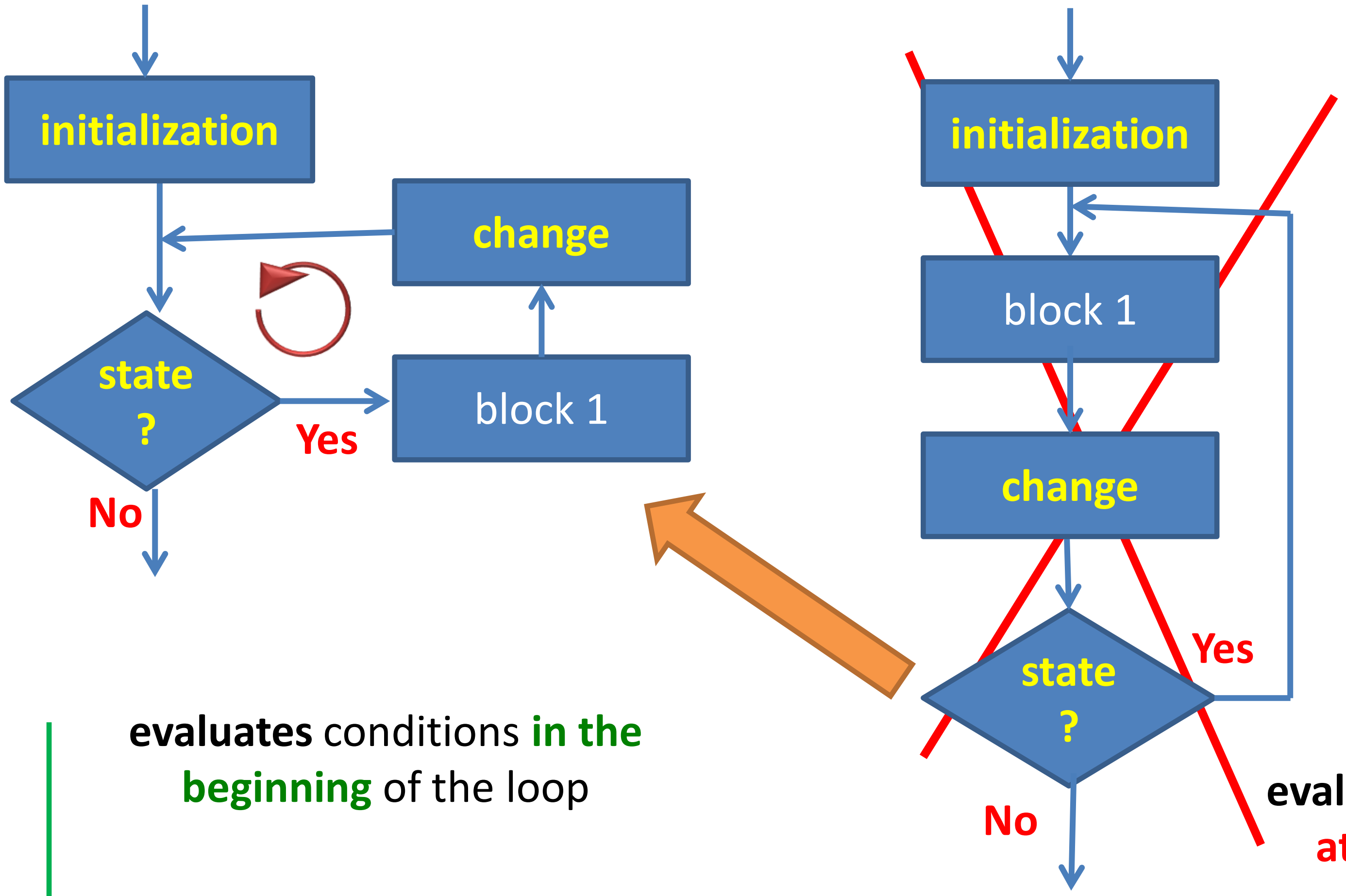


evaluates conditions **at the end** loop

complicated implementation in bash

This algorithm has no direct support in the control structures of bash, its implementation is possible, but at cost of poorer readability of resulting code.

Loop using while/until ...



evaluates conditions **in the beginning** of the loop

easy implementation in bash

evaluates conditions **at the end** loop

This algorithm has no direct support in the control structures of bash, its implementation is possible, but at cost of poorer readability of resulting code.

Loop Using while/until

A loop is a control structure that repeatedly executes a sequence of commands. Repetition and end of the loop is controlled by a condition.

```
while  command1
       do
              command2
              ...
done
```

the loop is in progress **while** command1 **returns** return value 0 (no error)

the loop is in progress **until** command1 **will not return** a return value of 0

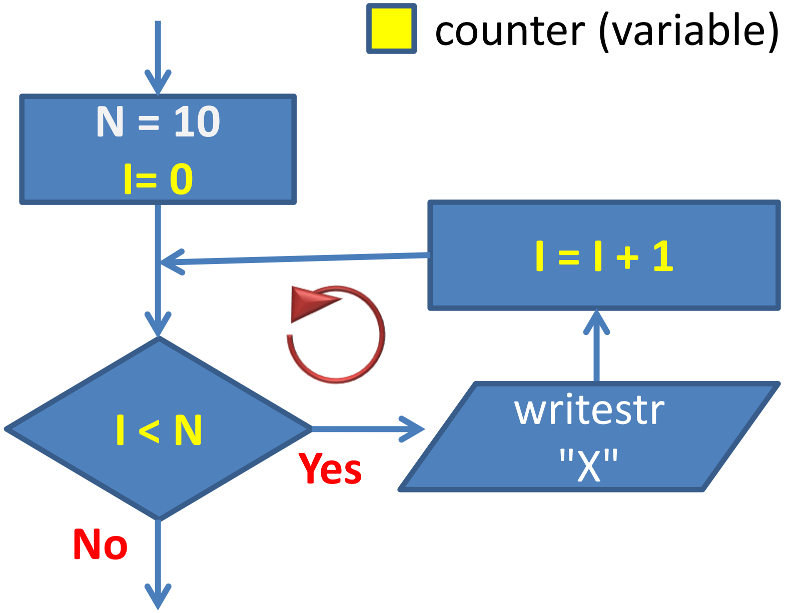
```
until  command1; do
       command2
       ...
done
```

Compact notation:

```
while  command1; do
       command2
       ...
done
```

Practical Example - Loop

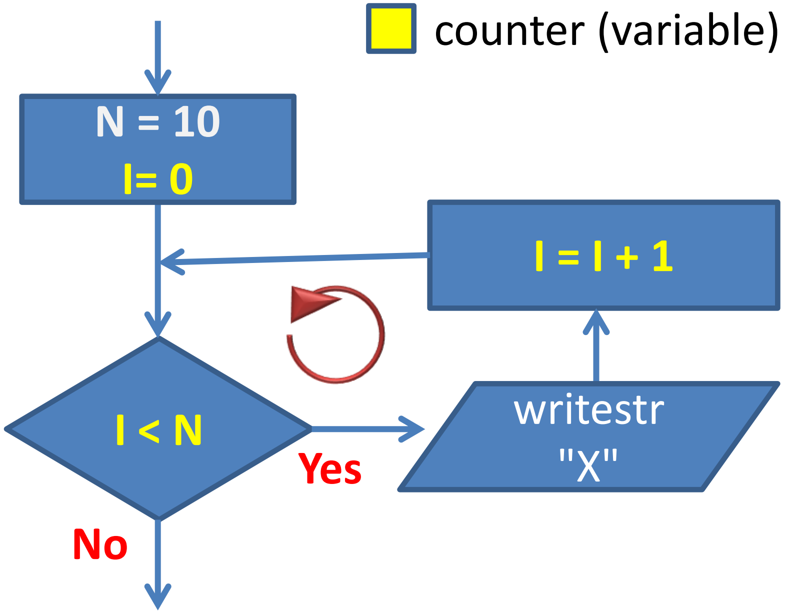
🟨 counter (variable)



```
N=10
I=0
while [[ I -lt N ]]; do
     echo "X"
     ((I = I + 1))
done
```

Practical Example - Loop

🟨 counter (variable)



$ must be used

```
N=10
I=0
while test "$I" -lt "$N"; do
     echo "X"
     ((I = I + 1))
done
```

```
N=10
I=0
while [[ I -lt N ]]; do
     echo "X"
     ((I = I + 1))
done
```

```
N=10
I=0
while [[ "$I" -lt "$N" ]]; do
     echo "X"
     ((I = I + 1))
done
```

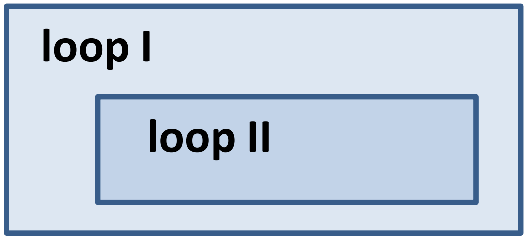
$ optional, if [[   ]] or ((   )) block is used

Exercise I

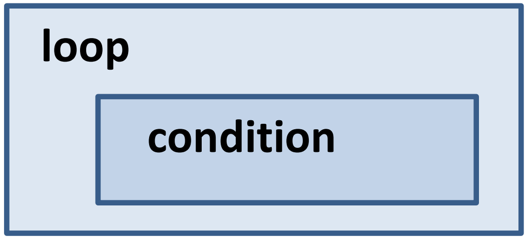
1. Write a bash script, which prints N "X" characters to the terminal. The number of characters will be entered by the user as the first argument of the script. The script prints an error message if the specified number of characters does not exceed two.

Complex Constructions - Nesting

Bash language **does not have a labels and command goto**, or their equivalents. Thus, more complex constructions can be implemented only by immersing loops and conditions into each other. The level of immersion is not limited.

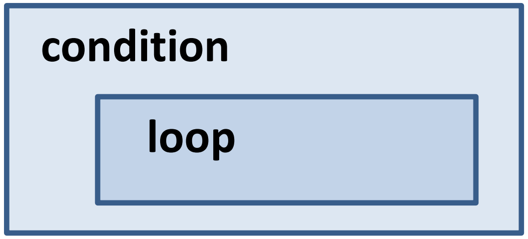

loop I

loop II

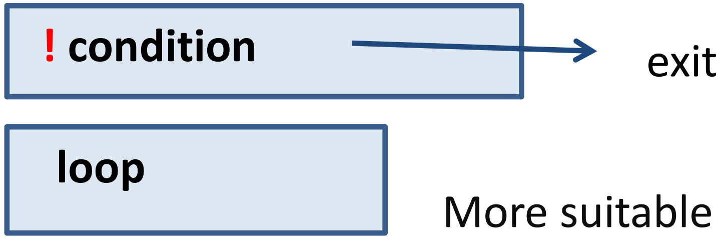

loop

condition

However, when designing an algorithm/script, we try to avoid unnecessary nesting (mostly for easier orientation in the script).


condition

loop


! condition

exit

loop

More suitable arrangement, e.g., for testing input data from users.

Nesting Loops - Example

```
N=10
I=0
while [[ I -lt N ]]; do
        J=0
        while [[ J -lt I ]]; do
                echo -n "X"
                ((J = J + 1))
        done
        echo ""
        ((I = I + 1))
done
```

outer loop counter can affect behavior of the inner loop

For nested structures, we pay attention to **indentation of text blocks** which increases **clarity and readability of the code**. Text editors have built-in support that makes indentation easier, such as in **gedit** editor, the indentation of the selected text block can be achieved with the TAB or Shift + TAB key.

Exercise II

1. Write bash scripts for the following tasks. Make user enter the dimension of the rendered shape after running the script. While working on the script, employe the algorithm created from the homework.

Task 1

Print a square of **X** characters into the terminal. The length of the side of the square is entered by the user.

```
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
```

Ignore the fact that it is not visually a square. However, number of **X** characters on the line and the number of lines must be the same.

Task 2

Print a right triangle with **X** characters into the terminal, so that one perpendicular is located at the top and the other on the left. The length of the perpendicular is entered by the user.

```
X X X X X X X X X X
X X X X X X X X X
X X X X X X X X
X X X X X X X
X X X X X X
X X X X X
X X X X
X X X
X X
X
```

Task 3

Print a right triangle with **X** characters into the terminal, so that one perpendicular is located at the bottom and the other on the left. The length of the perpendicular is entered by the user.

```
x
x x
x x x
x x x x
x x x x x
x x x x x x
x x x x x x x
x x x x x x x x
x x x x x x x x x
x x x x x x x x x x
```