C2110 *UNIX and programming*

Lesson 10 / Module 1

PS / 2020 Distance form of teaching: Roar3

Petr Kulhanek

kulhanek@chemi.muni.cz

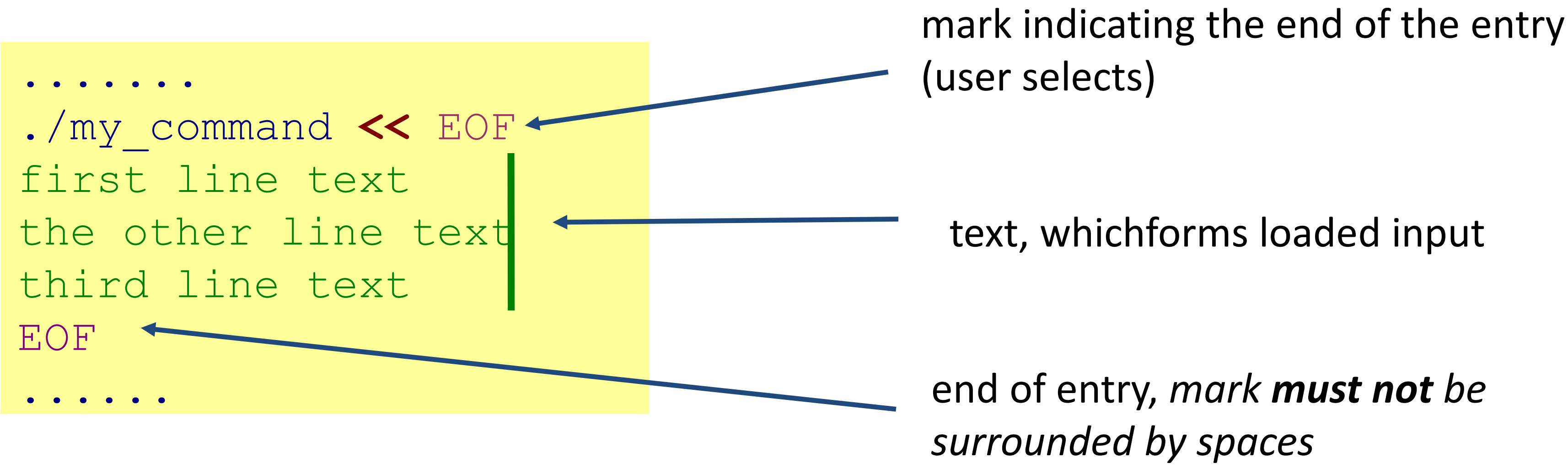
National Center for Biomolecular Research, Faculty of Science
Masaryk University, Kamenice 5, CZ-62500 Brno

Bash

- Redirecting input from a script

Redirecting Input from a Script

Standard input redirection of program my_command from file script.



```
.......
./my_command << EOF
first line text
the other line text
third line text
EOF
......
```

mark indicating the end of the entry (user selects)

text, whichforms loaded input

end of entry, *mark **must not** be surrounded by spaces*

This method of redirection is especially useful in scripts, but it also works on the command line. The advantage is the expansion of variables in the read text.

Examples

```
#!/bin/bash

for((I=1;I<=10;I++)); do
     NAME=`printf "%02d.txt" $I`
cat << EOF > $NAME
Toto je soubor cislo: $I
EOF
done
```

The result of commands preceded by back quotation marks `` is saved into the variable NAME.

The highlighted text is sent to **standard input** of cat command, the variables are expanded before sending the input, then the cat command saves it to $NAME file.

```
#!/bin/bash

PHASE=1.2

gnuplot << EOF
plot sin(x+$PHASE)
EOF
```

In this way, you can programmatically create scripts for gnuplot.

Exercise 1

1. Write a script that creates ten files. The file name will be in the format XX.txt where XX is the file number. If the file number is less than ten, use the 0 character for the first digit in the name. Each file will contain the following text (X is the file number):

```
Automatically created text file
File number is: X
```

2. Write a script that asks the user for the name of the image file in png format and then renders the sin(x) function into it.