

# \*x tipo sistemų programų kūrimo aplinka

Saulius Gražulis

2009 ruduo

Vilnius University, Faculty of Mathematic and Informatics  
Institute of Informatics



This set of slides may be copied and used as specified in the  
[Attribution-ShareAlike 4.0 International](#) license



*Suppose you went back to Ada Lovelace and asked her the difference between a script and a program. She'd probably look at you funny, then say something like: Well, a script is what you give the actors, but a program is what you give the audience. That Ada was one sharp lady...*

*Larry Wall*

<http://www.perl.com/pub/a/2007/12/06/soto-11.html>

2009-10-12

# sh interpretatorius

sh – dar viena programavimo kalba

- Komandų interpretatorius sh taip pat yra dar viena \*x sistemos komanda, taigi:
  - sh> sh < script.sh > result.txt
  - sh> sh script.sh > result.txt
  - sh> sh script.sh script-arg1 script-arg2 > result.txt
- Iš failo script.sh interpretatorius skaito savo komandas ir jas vykdo eilutė po eilutės
- likę argumentai (script-arg1 script-arg2) tampa šios skripto *poziciniaisiais parametrais*, jie pasiekiami script.sh kode kaip \$1, \$2 ir t.t.

```
#!/bin/sh

set -ue

FILE=""

for arg in $*
do
    case "$arg" in
        -a) echo "This is 'a' option" ;;
        -b) echo "This is 'b' option" ;;
        *) FILE=$arg;; # process file args
    esac
done
```

- Paprasčiausias skriptas – tai tiesiog komandų seka faile:
  - simple.sh:

```
#!/bin/sh
echo "I am a simple 'sh' script"
```

- Pirmas komandos argumentas matomas kaip \$1:
  - onearg.sh:

```
#!/bin/sh  
echo "$0: my first argument is '$1'"
```

- O \$0? Tai tiesiog paties skripto vardas :)

- Kintamiesiems galime priskirti reikšmes (simbolių eilutes) ir toliau jas naudoti:
  - vars.sh:

```
#!/bin/sh
ARG="$1"
STR="my_first_argument"
echo $0: ${STR} is "\${ARG}\ "
```

- Kintamuosius verta naudoti, kad svarbių reikšmių apibrėžimai būtų tik vienoje vietoje:
  - spot.sh:

```
#!/bin/sh
FILE="$1"
echo $0: file "`${FILE}`" contents is:
cat ${FILE}
```

Ir šiaip su prasmingais kintamųjų vardais kodas lengviau skaitomas...



- if-then-else:
  - ifthenelse.sh:

```
#!/bin/sh
if [ "$1" = "null" ]
then
    echo "$0: my argument is 'null'"
fi

if [ $# -eq 0 ]; then
    echo $0: I have no arguments
fi
```

- while ciklai:
  - args.sh:

```
#!/bin/sh

set -ue

n=1

while [ $# -gt 0 ]
do
    echo "$0 arg_$n is '$1'"
    n=$(( $n + 1 ))
    shift
done
```

- for ciklai:
  - for.sh:

```
#!/bin/sh
for i in alpha beta gamma
do
    echo -n "$i_"
done
echo ""

for i in $(seq 1 5) $(seq 6 2 22)
do
    echo -n "$i_"
done
echo ""
```