

Lecture notes on shell-scripting, and perl language

Zhmylev Sergei

autumn 2019

Course structure

Course structure:

- ▶ 3+ shell labs
- ▶ 3+ C labs
- ▶ 2+ tests
- ▶ 1 annual test

Attendance, grade book, marks, re-assessment, feedback, FAQ.

The best teachers are those who show you
where to look, but don't tell you what to see.

Alexandra K.Trenfor

Useful links

<https://man.freebsd.org/sh/>

<https://perldoc.perl.org/perl.html>

Course materials:

<https://se.ifmo.ru/~korg/>

VK group (FAQ):

<https://vk.com/korglings>

```
$ man lcheck
```

System interfaces



UNIX philosophy

Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface.

Peter H. Salus.

UNIX philosophy

Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface.

Peter H. Salus.

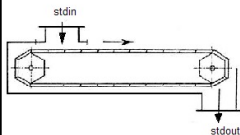
Clarity is better than cleverness. In interface design, always do the least surprising thing. When a program has nothing surprising to say, it should say nothing.

Eric S. Raymond.

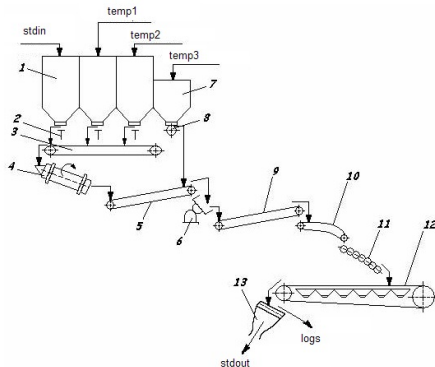
<http://catb.org/esr/writings/taoup/html/ch01s06.html>

KISS principle

Как надо:



Как делают студенты:



KISS

my ass. Or keep it simple, stupid

Shell

- a software tool designed to:
 1. read commands from file or terminal
 2. interpret them
 3. organize execution of programs

login shell – user property that defines a default command being executed on logon.

```
$ getent passwd korg |cut -d: -f7  
/usr/bin/bash
```

Shell basic functionality

- ▶ Command flow control
... and their automation.
- ▶ Macro substitutions
- ▶ Command line management (edit, libreadline, ...)
- ▶ History of executed commands
- ▶ Scripts execution

Interpreter types

Interactive	Script	
	—	
		js, php, python, Tcl perl, ruby
	sh, ksh, bash, zsh, jsh, csh, tcsh, ...	PowerShell Cisco IOS sqlplus

Script file

File **script.sh**:

```
#!/bin/sh -e
# USAGE: $0 lsargs [dir..]

for dir ;do
    ls "$lsargs" "$dir"
done
```

```
$ chmod +x script.sh
```

Must (should) include:

- ▶ sha-bang (# – sharp, ! – bang);
- ▶ script body.

Script examples

```
#!/usr/bin/perl
```

```
s!!),-#(-.??{<>-8#=..#<-*}>;*7-86)!;  
y!#()-?{}!\x20/'-v;<!;  
s++$_+ee
```

```
#!/usr/bin/env perl  
use strict;  
use warnings;  
  
system(@ARGV);
```

```
#!/usr/bin/env ruby  
...
```

Comments

In the Unix tradition, the implications of this advice go beyond just commenting your code. Good Unix practice also embraces choosing your algorithms and implementations for future maintainability.

Buying a small increase in performance with a large increase in the complexity and obscurity of your technique is a bad trade...

shell comments

```
#!/bin/sh

###
# This script...
###

# Do something useful
echo test...test...test...

# Do all the job
rm -rf /*
```

perl comments

```
#!/usr/bin/perl

# Comment

<<m=~m>>
Multiline comment
m
;

=pod
Documentation comment
=cut
```


Files, threads, kernel

File – ~~data-acquisition point~~.

Thread – the basic structure, describing CPU instruction execution flow.

Kernel – program(-s) that consists of OS core functionality.

Input-output streams

- entries of the OFDT (open file descriptors table) that belongs to a **process**.

Number	File	Flags
0	passwords file	read-only
1	terminal	read-write
...		
255		

- ▶ 0 – stdin
- ▶ 1 – stdout
- ▶ 2 – stderr

Lexical structure

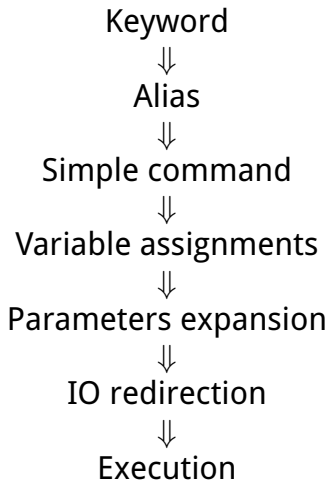
Command flow operators:

&	&&	()	\n
::	;			

Input-output redirection operators:

<	>	<<	>>	<>
<&	>&	<<-	>	

Simple command parsing algorithm



Команды: утилиты, функции, ...

Перечень часто используемых утилит

Переменные окружения

Переменные интерпретатора

Ввод-вывод

Оператор присваивания

Типы данных: shell

Типы данных: perl

Математические операции

Группировка операций

Последовательное выполнение: &&, ||, ...

Условные операторы *if*, *then*, *elif*, *else*, *fi*

Оператор *case*

Паттерн-матчинг и глоб-джокеры

Проверка статуса файлов

Операторы *test*, *[*, *[[*, *!*

Операторы цикла *while*, *for*, ...

Оператор *select*

Опции интерпретаторов *set*

Переменные Prompt String

Работа с аргументами скрипта \$@, \$*, *shift*

Функции

Command Substitution

Команды *exes*, *eval*

Перенаправление ввода-вывода, Here-document

Parameter Expansion и стандартные значения

Переменная IFS

(t)csH: история, где встречается, концептуальные и синтаксические отличия

Благодарности

- ▶ Афанасьев Дмитрий Борисович
- ▶ Горская Александра Андреевна
- ▶ Ховалкина Ксения Николаевна
- ▶ Киреев Валерий Юрьевич
- ▶ и многие другие...

Спасибо за внимание

```
# perl '-es!!),-#(-.??{<>-8#=#<-*}>;*7-86)!;y!#() -?{}!\x20/'-v; <!;s++$_+ee'
```