

# Материалы к лекциям по разработке скриптов, интерпретаторам семейства sh и perl

Жмылёв Сергей Александрович

Осень 2018

# Структура курса I

Взаимодействие с системой

Философия UNIX

Потоки ввода-вывода

Командный интерпретатор (shell)

Разновидности интерпретаторов

Командный файл (скрипт)

Комментарии

Команды: утилиты, функции, ...

## Структура курса II

Перечень часто используемых утилит

Переменные окружения

Переменные интерпретатора

Ввод-вывод

Оператор присваивания

Типы данных: shell

Типы данных: perl

Математические операции

## Структура курса III

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*, ...

## Структура курса IV

Оператор *select*

Опции интерпретаторов *set*

Переменные Prompt String

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

Функции

Command Substitution

Команды *exec, eval*

Перенаправление ввода-вывода,  
Here-document

# Структура курса V

Parameter Expansion и стандартные значения

Переменная IFS

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

# Организационные вопросы

Организационный объём курса:

- ▶ 3+ лабораторных по shell
- ▶ 3+ лабораторных по С
- ▶ 2+ рубежных контрольных
- ▶ 1+ письменный зачет
- ▶ ?+ курсовых работ

Посещаемость, электронный журнал, баллы, пересдачи, зачетки, отзывы, вопросы.

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

Alexandra K.Trenfor



## Полезные ссылки

<https://www.freebsd.org/cgi/man.cgi?query=sh>

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

Материалы по курсу:

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

Группа ВКонтакте (объявления,  
вопросы):

<https://vk.com/korglings>

Репозиторий с лекционными  
материалами:

<https://github.com/zhmylove/lec>

# Взаимодействие с системой



## Философия UNIX

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.

## Философия UNIX

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.

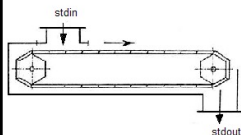
http:

[//catb.org/esr/writings/taoup/html/ch01s06.html](http://catb.org/esr/writings/taoup/html/ch01s06.html)

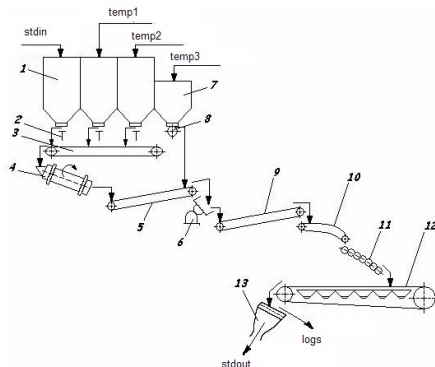
УНИВЕРСИТЕТ ИТМО

# Принцип KISS

Как надо:



Как делают студенты:



# KISS

my ass. Or keep it simple, stupid

## Потоки ввода-вывода

– это записи из таблицы дескрипторов открытых файлов для **процесса**.

Номер	Файл	Флаги
0	файл с паролями	чтение
1	терминал	запись
...		
255		

- ▶ 0 – стандартный поток ввода (stdin)
- ▶ 1 – стандартный поток вывода (stdout)
- ▶ 2 – стандартный поток ошибок (stderr)



## Командный интерпретатор (shell)

– это утилита, предназначенная для чтения команд из файла или терминального устройства, их интерпретации и организации выполнения других утилит.

login shell – атрибут пользователя, указывающий путь к командному интерпретатору, запускаемому при входе.

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

# Основные функции shell

- ▶ Управление потоком выполнения команд  
... в т.ч., их автоматизация
- ▶ Работа с «макро-подстановками»
- ▶ Управление командной строкой  
(редактирование, readline, ...)
- ▶ Поддержка истории выполненных команд
- ▶ Исполнение командных файлов



## Скриптовые

js, php, python,  
Tcl  
perl, ruby

## Интерактивные

sh, ksh, bash,  
zsh, jsh,  
csh, tcsh,

PowerShell  
Cisco IOS  
sqlplus

# Командный файл (скрипт)

## Файл **script.sh**:

```
#!/bin/sh -e  
# USAGE: $0 lsargs [dir..]  
  
for dir ;do  
    ls "$lsargs" "$dir"  
done
```

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

Может (должен) включать:

- ▶ sha-bang (# – sharp, ! – bang);
- ▶ тело скрипта.

# Примеры скриптов

```
#!/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  
...
```

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...

# Комментарии в shell

```
#!/bin/sh

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

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

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

# Комментарии в perl

```
#!/usr/bin/perl

# Comment

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

=pod
Documentation comment
=cut
```

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)ssh: 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++$+_tee'
```