

天津医科大学理论课教案首页

(共 7 页、第 1 页)

课程名称：Linux 系统概论

课程内容/章节：Perl 语言简介 / 第 17 章

教师姓名：伊现富

职称：讲师

教学日期：2016 年 4 月 27 日 8:00-10:00

授课对象：生物医学工程与技术学院 2014 级生信班（本）

听课人数：30

授课方式：理论讲授

学时数：2

教材版本：Unix 入门经典，第 1 版

教学目的与要求（分掌握、熟悉、了解、自学四个层次）：

- 掌握 Perl 的语法结构，Perl 中的变量类型，Perl 的操作符，Perl 的判断语句和循环语句。
- 熟悉 Perl 的内置变量，Perl 的基本函数，检修和调试 Perl 脚本的方法。
- 了解 Perl 的特性、中心思想和优缺点。
- 自学其他函数的使用，Perl 脚本的编写。

授课内容及学时分配：

- (10') 引言与导入：对 Perl 进行简介，包括其特性、应用领域、中心思想和优缺点等，讲解 Perl 脚本的语法结构和运行方法。
- (20') 变量：介绍 Perl 的变量类型及常见内置变量，讲解标量、数组和散列的使用。
- (10') 操作符：介绍 Perl 中的操作符，包括数字、字符串、逻辑、文件测试和匹配操作符。
- (10') 基本函数：介绍 print、chomp、join、split、open、close 和 my 等基本函数的使用。
- (20') 判断语句：讲解 if、unless 和 given-when 等判断语句的语法和使用。
- (20') 循环语句：讲解 foreach、for、while 和 until 等循环语句的语法和使用。
- (5') 检修脚本：介绍检修和调试 Perl 脚本的方法。
- (5') 总结与答疑：总结授课内容中的知识点与技能，解答学生疑问。

教学重点、难点及解决策略：

- 重点：Perl 的变量类型，Perl 的操作符，基本函数的使用。
- 难点：Perl 的变量类型，Perl 的判断语句和循环语句。
- 解决策略：通过类比比较和实例分析帮助学生理解记忆，结合逻辑流程和语法结构讲解判断语句和循环语句。

专业外语词汇或术语：

标量型变量 (scalar variable)

关联数组 (associative array)

数组 (array)

散列 (hash)

辅助教学情况：

- 多媒体：Perl 的语法结构，各种变量的使用，判断语句和循环语句的逻辑流程。
- 板书：Perl 的变量类型，判断语句和循环语句的语法结构。
- 演示：基本函数的使用，Perl 脚本。

复习思考题：

- Perl 中的变量主要有哪三大类？
- chomp, join, split 的作用是什么？
- 列举 Perl 中的各种操作符。
- Perl 中的判断和循环语句及其语法。

参考资料：

- (美) Dan E. Krane & Michael L. Raymer 著，孙啸，陆祖宏，谢建明等译。生物信息学概论，清华大学出版社，2004。
- Randal L. Schwartz, Brian d foy & Tom Phoenix 著，盛春 译。Perl 语言入门（第六版），东南大学出版社，2012。
- Cynthia Gibas & Per Jambeck 著，孙超，郭庆民，刘相国，吴斌 译。生物信息学中的计算机技术，中国电力出版社，2002。
- 陶士珩 主编。生物信息学，科学出版社，2007。
- 维基百科等网络资源。

主任签字：

年 月 日

教务处制

一、引言与导入 (10 分钟)

1. 简介

- 高级、通用、直译式、动态的程序语言
- Practical Extraction and Report Language, 实用摘录与报表语言
- Pathologically Eclectic Rubbish Lister, 病态折中式垃圾列表器
- 拉里·沃尔 (Larry Wall), 1987 年 12 月 18 日
- Perl: 程序语言本身; perl: 实际编译并运行程序的解释器; PERL: 错误的写法

2. 特性

- 具有动态语言的强大灵活的特性
- 借用了 C、sed、awk、shell 等语言的特性, 提供了许多冗余语法
- 使用了语言学的思维 (泛型变量、动态数组、Hash 表等)
- 程序员可以忽略内部数据存储、类型、内存越界等细节
- 内部集成了正则表达式的功能
- 巨大的第三方代码库 CPAN (Comprehensive Perl Archive Network, Perl 综合典藏网)

3. 应用领域: 脚本语言中的瑞士军刀, 网络编程、图形编程、系统管理、生物信息……

4. 中心思想: TMTOWTDI

5. 优缺点

- 优点: 容易使用 (但学习并不简单), 几乎不受限制, 速度通常很快
- 缺点: 灵活、随意和过度的冗余语法, write-only, 解释器耗费资源

6. 语法结构与运行方法

• 语法结构

```
#!/usr/bin/perl

use strict;
use warnings;

print "Hello World!\n";
```

• 运行方法

```
# Step 1: 编写脚本
vim hello.pl
# Step 2: 修改权限
chmod 755 hello.pl
# Step 3: 运行脚本
./hello.pl
```

二、【重点、难点】变量 (20 分钟) (以队列、字典进行类比, 同时进行实例分析)

1. 简介: 变量“无类型”, 主要包括标量、数组和散列三大类

2. 标量: 只包含一个元素的变量; 以\$开头

- 字符串, 双引号: \$name = "Paul";
- 数字, 无引号: \$age = 29;
- 字符串, 单引号: \$where_to_find_him = 'http://www.weinstein.org';

3. 数组: 含有任意数量元素的变量, 以其存储顺序作为索引; 以@开头

• 声明与解引用

```
# 字符串
@authors = ("Paul", "Joe", "Jeremy", "Harley");
$authors[4] = "Tom";

# 数字
@list = (1, 2, 3, 4);
# 解引用: $array[index]
$authors[4] # Tom
$list[0] # 1
```

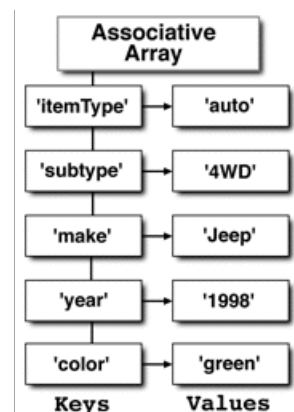
- 数组操作: shift, unshift, pop, push



4. 散列：像字典一样，把不同的变量按照它们的逻辑关系组织起来，并以作为“键”的变量进行索引；以% 开头

```
# 创建散列
%person = (
    name => 'Paul',
    age => '29',
    url => 'http://www.weinstein.org',
)

# 提取键值：$hash{key}
$person{"age"} # 29
```



5. 内置变量

Perl 提供了大量的预定义变量。下面列举了常用的一些预定义变量：

\$_	在执行输入和模式搜索操作时使用的默认空格变量
\$.	文件中最后处理的当前行号
\$@	由最近一个 eval() 运算符提供的 Perl 语法报错信息
\$!	获取当前错误信息值，常用于 die 命令
\$0	含有正在执行的程序名
\$\$	正在执行本脚本的 Perl 进程号
\$PERL_VERSION / \$^V	Perl 解释器的版本、子版本和修订版本信息
@ARGV	含有命令行参数
ARGV	一个特殊的文件句柄，用于遍历 @ ARGV 中出现的所有文件名
@INC	库文件的搜索路径
@_	在子例程中， @_ 变量含有传给该子例程的变量内容
%ENV	关联数组型变量 %ENV 含有当前环境信息
%SIG	关联数组型变量 %SIG 含有指向信号内容的句柄

- 三、【重点】操作符（10 分钟）（和 shell 脚本中的操作符进行比较，同时进行实例分析）
主要包括：数字操作符，字符串操作符，逻辑操作符，文件测试操作符，匹配操作符。

操作符	含义
+	加法
-	减法
*	乘法
/	除法
**	乘幂，乘方
%	取模，取余
<	小于
>	大于
==	等于
<=	小于等于
>=	大于等于
!=	不等于
<=>	比较。a<=>b：a 等于 b 时返回 0，a 大于 b 时返回 1，a 小于 b 时返回-1

操作符	含义
-r	可读
-w	可写
-x	可执行
-e	存在
-z	存在但没有内容
-s	存在且有内容
-f	普通文件
-d	目录
-l	符号链接
-T	看起来像文本文件
-B	看起来像二进制文件
-M	最后被修改后至今的天数
-A	最后被访问后至今的天数
-C	最后 inode 变更后至今的天数

操作符	含义
.	连接。"string1" . "string2"
x	重复。"string" x number
lt	小于
gt	大于
eq	等于
le	小于等于
ge	大于等于
ne	不等于
cmp	比较。类似于数字比较的<=>

操作符	含义
&&	逻辑 AND，与
	逻辑 OR，或
!	逻辑 NOT，非
?	条件操作符

操作符	含义
=~	绑定操作符，匹配
!~	绑定操作符，不匹配
~~	智能匹配操作符

四、【重点】基本函数 (10 分钟) (实例分析、操作演示)

主要介绍 `print`、`chomp`、`join`、`split`、`open`、`close`、`my` 等函数的作用及其用法。

向标准输出打印文本

```
print "Hello Again\n";
```

向一个具有文件句柄的文件打印文本

```
print FILE "Hello Again\n";
```

打印变量的值

```
print "How are on this day, the " . $date . "?\n";
```

删除变量末尾的 (多个) 换行符, 返回删除的换行符的个数

```
chomp $name;
```

```
chomp @authors;
```

joining a number of strings together with a colon delimiter

```
$fields = join ':', $data_field1, $data_field2, $data_field3;
```

splitting a string into substrings

```
($field1, $field2) = split '://', 'Hello:World', 2;
```

splitting a scalar and creating an array

```
@fields = split '://', $raw_data;
```

open the file and slurp its contents into an array

and then close the file

```
open(FILE, "/etc/passwd");
```

```
@filedata = <FILE>;
```

```
close(FILE);
```

```
open my $IN, '<', $file_in or die "$0 : failed to open
```

```
input file '$file_in' : $!\n";
```

```
while(<$IN){
```

```
    chomp;
```

```
    actions;
```

```
}
```

```
close $IN or warn "$0 : failed to close input file
```

```
'$file_in' : $!\n";
```

Global variable \$name is given a name

```
$name = "Paul";
```

Enter our loop

```
foreach (@filedata) {
```

```
    # declare a new variable for just the loop
```

```
    my $current_file;
```

```
    # create a local version of name to temporarily
```

```
    # assign values within the loop to
```

```
    local $name;
```

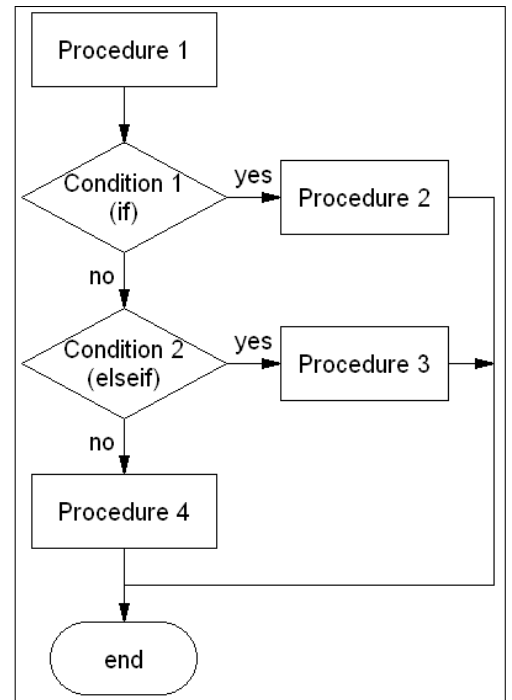
```
    ...
```

```
}
```

五、【难点】判断语句 (20 分钟) (结合逻辑流程讲解语法结构, 并进行实例分析)

```
1. if
# if 区块
if ($hour > 22) {
    print "should sleep...\n";
}
# if 语句
print "hello" if $guest >= 1;

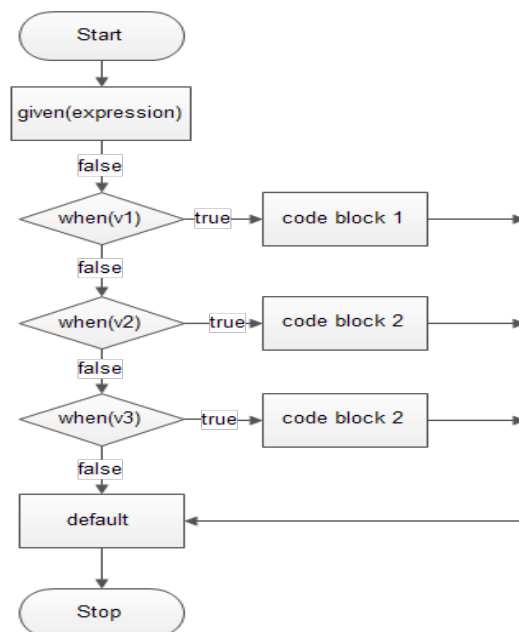
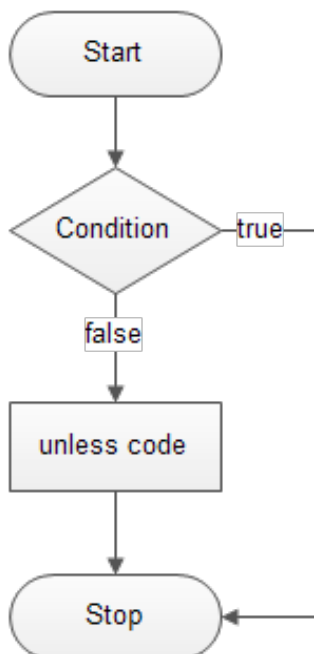
# if-elsif-else
if ($name eq "Paul") {
    print "Hi Paul\n";
}
elsif ($name eq "Joe") {
    print "Hi Joe\n";
}
elsif ($name eq "Jeremy") {
    print "Hi Jeremy\n";
}
else {
    print "Sorry, have we meet before?";
}
```



```
2. unless
# unless 区块
unless ($credit > 100) {
    print "You can not graduate! \n";
}
# unless 语句
print "eat\n" unless $food==0;

# given-when
given ($foo) {
    say "a" when "a";
    when (/b/) {say "b";}
    default {say "not match";}
}
```

3. given-when



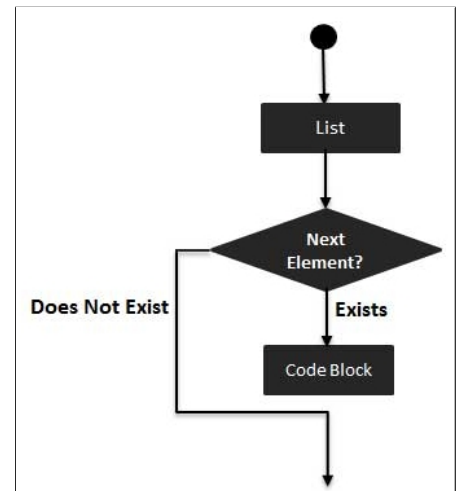
六、【难点】循环语句 (20 分钟) (结合逻辑流程讲解语法结构, 并进行实例分析)

1. foreach

```
@group = 1..10;

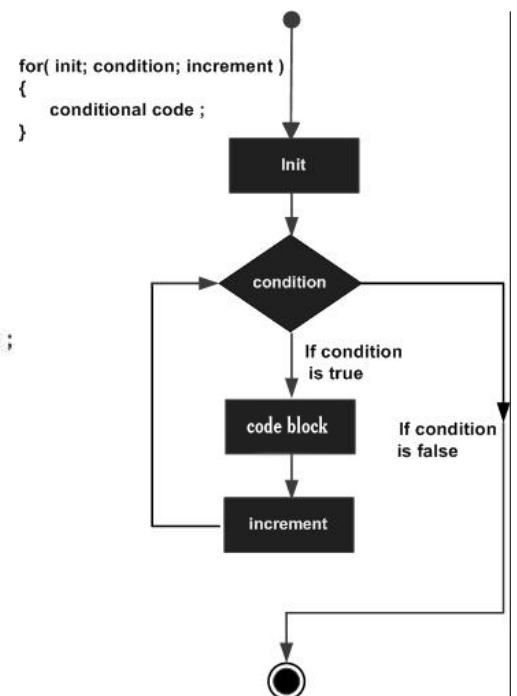
# foreach 循环
foreach my $element (@group) {
    print "$element\n";
}

# 等价的 for 循环
for (@group) {
    print "$_\n";
}
print "$_\n" for @group;
```



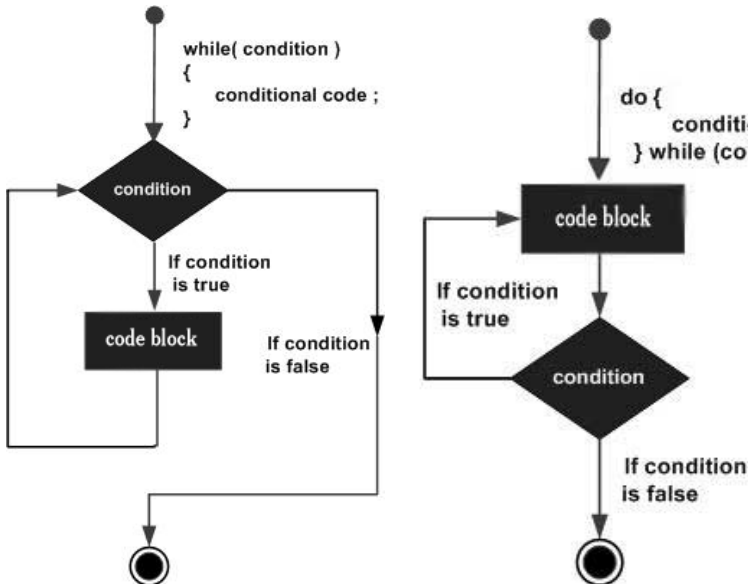
2. for

```
# 从1数到10
for ($i = 1; $i <= 10; $i++) {
    print "I can count to $i!\n";
}
```



3. while

```
while( condition )
{
    conditional code ;
}
```



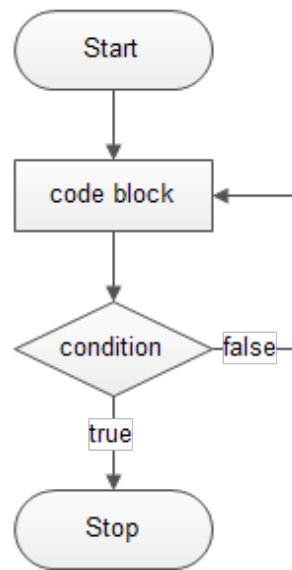
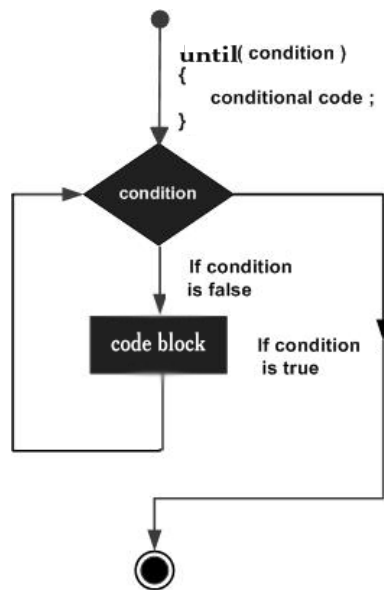
```
$i = 0;
# while
while ($i < 10) {
    print "$i\n";
    $i++;
}
```

```
# do-while
do {
    print "$i\n";
    $i = $i + 1;
} while ($i < 10);
```

4. until

```
$i = 0;
# until
until ($i == 10) {
    print "$i\n";
    $i++;
}
```

```
# do-until
do {
    print "$i\n";
    $i++;
} until ($i == 10);
```



七、 检修脚本 (5 分钟)

```

# 不洁模式
#!/usr/bin/perl -T
# 打开警告
use warnings;
# 严格模式，语法更加规范
use strict;
    
```

```

# 检查语法
perl -c script.pl
# 格式化脚本
perltidy script.pl
# 调试脚本
perl -d script.pl
    
```

八、 总结与答疑 (5 分钟)

1. 知识点

- Perl 语言简介：中心思想，优缺点，语法结构
- 变量：标量，数组，散列，内置变量
- 操作符：数字、字符串、逻辑、文件测试、匹配操作符
- 基本函数：print, chomp, join, split, open, close, my
- 判断语句：if, unless, given-when
- 循环语句：foreach, for, while, until
- 检修脚本：检查语法，格式化脚本，调试脚本

2. 技能

- 掌握 Perl 语言的基本语法
- 使用 Perl 编写简单的应用脚本