第七章 开发环境搭建和Shell编程

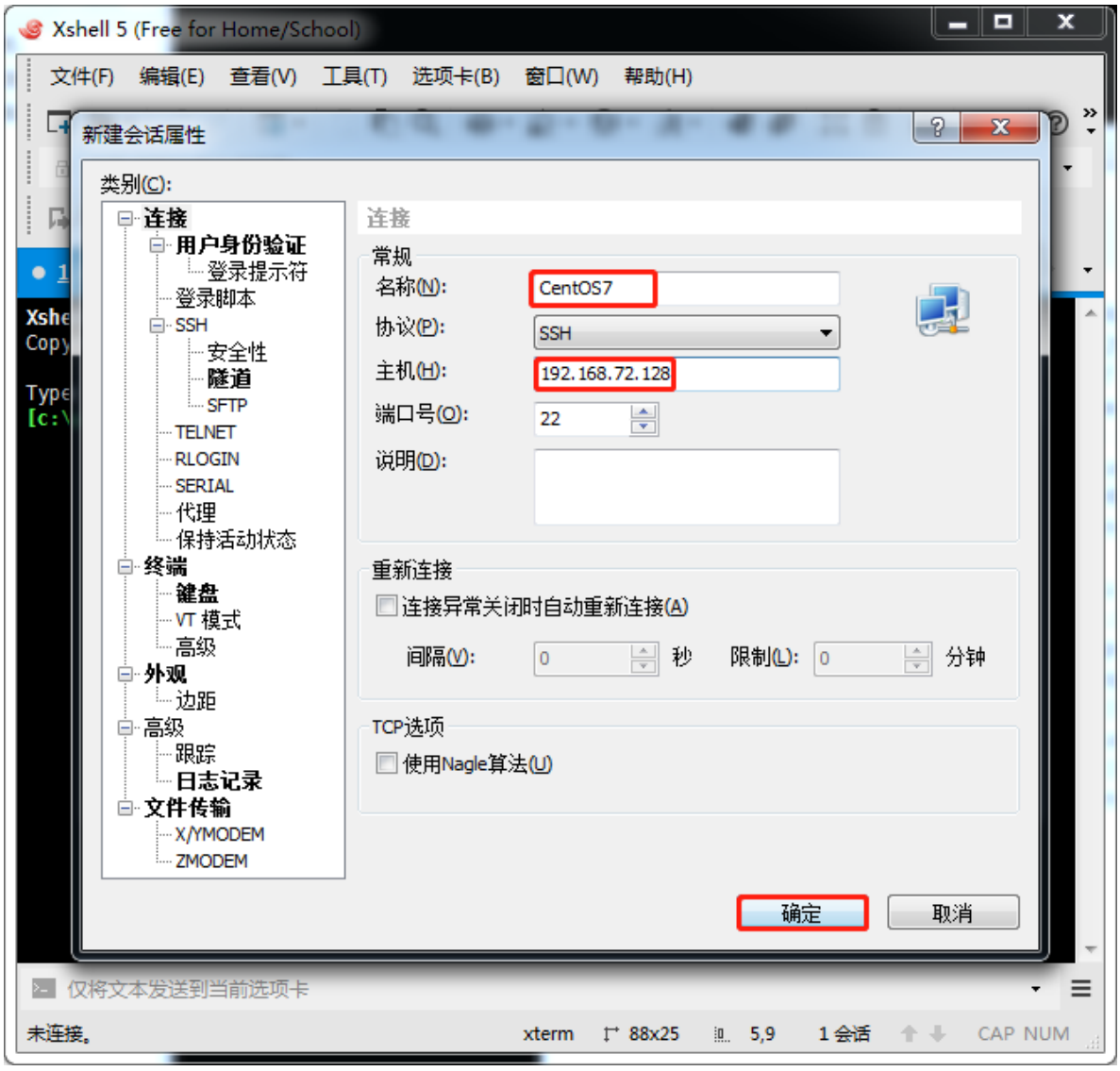
7.1 开发环境搭建（掌握）

7.1.1 Xshell和Xftp工具

（1）下载和安装方式

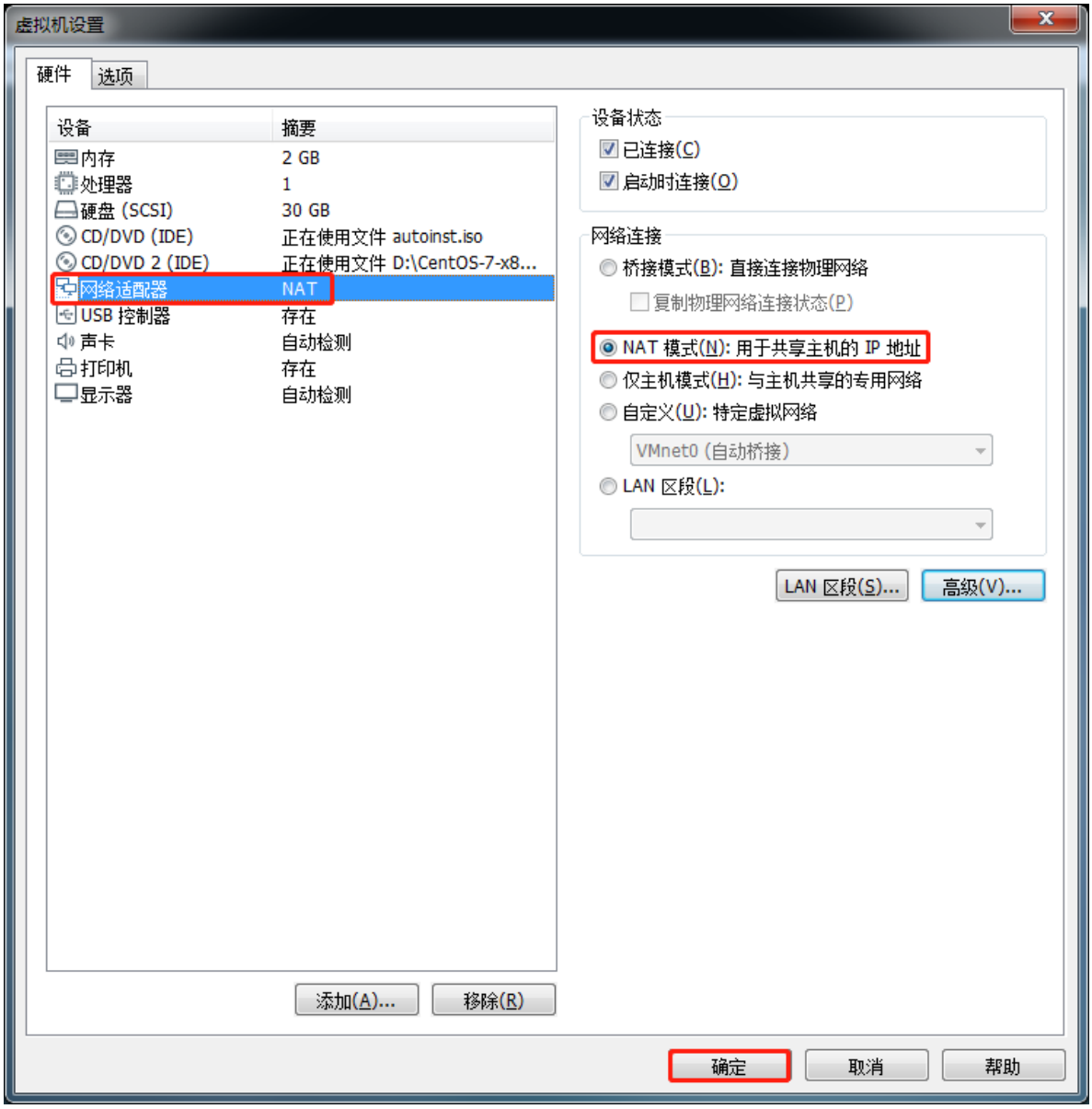
- 下载地址：https://www.netsarang.com/zh/
- 安装方式：直接一路点击下一步即可，安装过程选择免费版。

（2）使用方式



（3）网络模式设置

- 设置网络连接模式为NAT模式，如下图：



（4）修改配置文件

- 使用root用户打开/etc/sysconfig/network-scripts/ifcfg-eno16777736文件，添加内容如下：

```
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.72.128
GATEWAY=192.168.72.2
NETMASK=255.255.255.0
DNS1=114.114.114.114
```

（5）配置文件生效

使用命令使得配置文件生效：`service network restart`

7.1.2 JDK的下载和安装

（1）下载和安装方式

- 下载地址：https://www.oracle.com/java/technologies/javase-downloads.html
- 安装方式：将下载好的jdk安装包通过Xftp工具传输到CentOS系统中，使用tar命令解压即可。

（2）配置环境变量

- 使用root用户打开配置文件/etc/profile，向文件末尾追加内容如下：

```
export JAVA_HOME=/usr/javajdk
export PATH=$JAVA_HOME/bin:$PATH
```

- 保存退出后让文件生效并验证是否配置成功

```
source /etc/profile
javac -version
```

7.1.3 Tomcat的下载和安装

（1）下载和安装方式

- 下载地址：https://tomcat.apache.org/download-80.cgi
- 安装方式：将下载好的Tomcat安装包通过Xftp工具传输到CentOS系统中，使用tar命令解压即可。

（2）启动和关闭方式

```
startup.sh
shutdown.sh
```

（3）开放防火墙端口

```
/sbin/iptables -I INPUT -p tcp --dport 8080 -j ACCEPT  开启8080端口（暂时开通）
```

（4）配置环境变量

- 使用root用户打开配置文件/etc/profile，向文件末尾追加内容。

```
export CATALINA_HOME=/usr/tomcat
export PATH=$CATALINA_HOME/bin:$PATH
```

- 保存退出后让文件生效并验证是否配置成功

```
source /etc/profile
startup.sh
```

（5）发布Web项目

- 将Web项目打成war包，通过Xftp工具将war包放在tomcat/webapp目录并启动

7.1.4 Mysql的下载和安装

（1）下载Mysql的repo源

```
wget http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm
```

（2）安装rpm包

```
rpm -ivh mysql57-community-release-el7-8.noarch.rpm
```

（3）安装Mysql

```
yum install mysql-server
```

（4）启动服务

```
service mysqld start
```

（5）查看服务状态

```
systemctl status mysqld
```

（6）使用root用户登录

```
mysql -u root
```

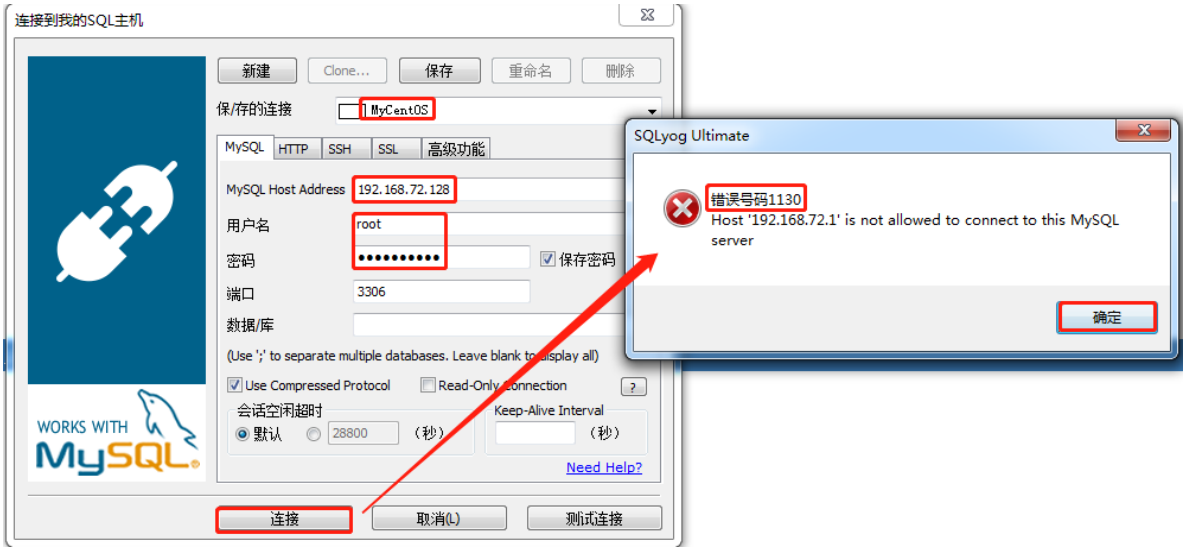
（7）修改临时密码

```
alter user 'root'@'localhost' identified by 'QiDian@666';
```

7.1.5 图形化界面访问数据库

（1）使用SQLyog工具

- 启动图形化界面工具SQLyog连接虚拟机中Mysql数据库，如下图：



（2）解决方案

- 使用root权限登录数据库后选择mysql库

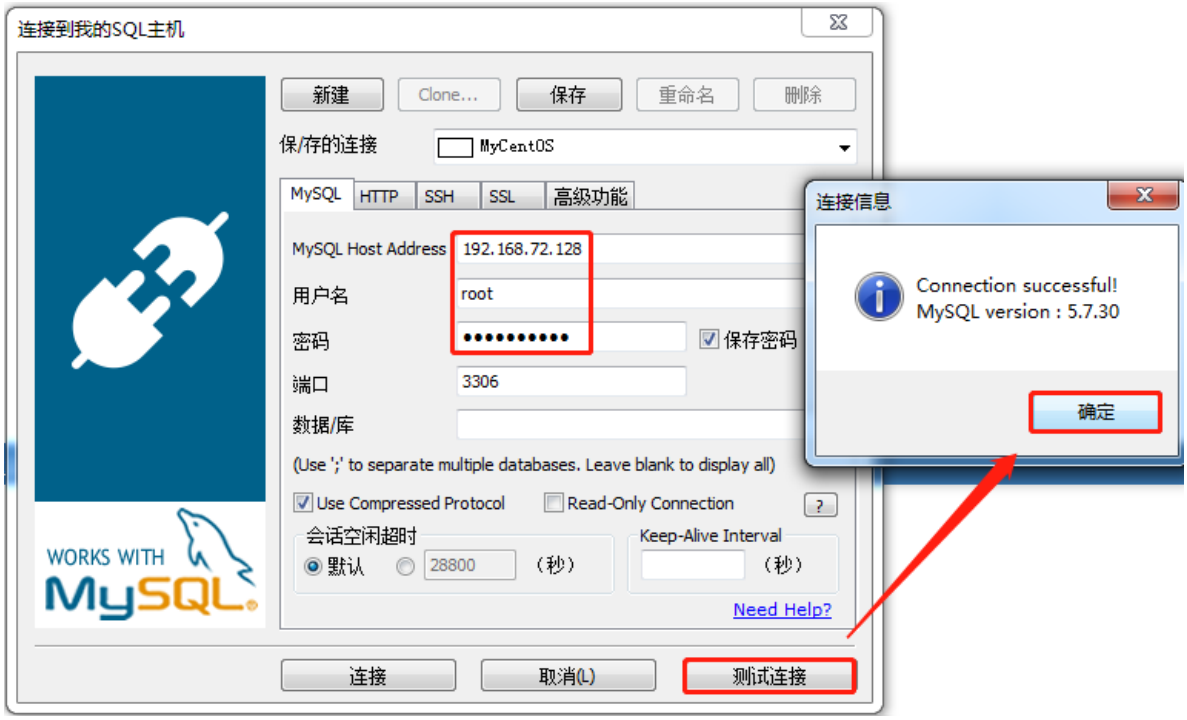
```
mysql -u root -p
use mysql;
```

- 查看mysql库中的user表的host值后修改为通配符%

```
select host from user where user='root';
update user set host='%' where user='root';
flush privileges;
```

- 查看修改结果并重新测试

```
select user,host from user;
```



7.2 Shell编程（熟悉）

7.2.1 基本概念

- Shell是一个命令行解释器，可以接收应用程序或用户命令，然后访问操作系统内核。
- Shell是一个功能相当强大的编程语言，易编写、易调试、灵活性强；

7.2.2 编写第一个程序

- 使用vi工具创建xxx.sh的文件。
- 以#!/bin/bash开头并编写代码后保存。

7.2.3 执行Shell程序的方式

更改权限：chmod u+x hello.sh

- 方式一：./文件名，此方式需要执行权限。
- 方式二：/bin/bash 文件名，此方式不需要执行权限。

/bin/bash hello.sh
bash hello.sh
sh hello.sh

7.2.4 变量的定义

（1）语法格式

- 定义变量：变量=值　　等号两侧不允许有空格
- 撤销变量：unset 变量

（2）定义规则

- 变量名称可以由字母、数字和下划线组成，但是不能以数字开头，环境变量名建议大写。
- 不能使用bash里的关键字。
- 中间不能有空格，可以有下划线。
- 在bash中，变量默认类型都是字符串类型，无法直接进行数值运算。
- 变量的值如果有空格，需要使用双引号或单引号括起来。

7.2.5 常用运算符

（1）算术运算符

运算符	说明	举例
+	加法	`expr $a + $b` 结果为 30。
-	减法	`expr $a - $b` 结果为 -10。
*	乘法	`expr $a \* $b` 结果为 200。
/	除法	`expr $b / $a` 结果为 2。
%	取余	`expr $b % $a` 结果为 0。
=	赋值	a=$b 将把变量 b 的值赋给 a。
==	相等。用于比较两个数字，相同则返回 true。	[$a == $b] 返回 false。
!=	不相等。用于比较两个数字，不相同则返回 true。	[$a != $b] 返回 true。

echo `expr $ia + $ib`

13 ic=$[$ia+$ib]
14 echo $ic

（2）关系运算符

运算符	说明	英文	举例
-eq	检测两个数是否相等，相等返回 true。	equal	[$a -eq $b] 返回 false。
-ne	检测两个数是否不相等，不相等返回 true。	not equal	[$a -ne $b] 返回 true。
-gt	检测左边的数是否大于右边的，如果是，则返回 true。	greater than	[$a -gt $b] 返回 false。
-lt	检测左边的数是否小于右边的，如果是，则返回 true。	less than	[$a -lt $b] 返回 true。
-ge	检测左边的数是否大于等于右边的，如果是，则返回 true。	Greater than or equal to	[$a -ge $b] 返回 false。
-le	检测左边的数是否小于等于右边的，如果是，则返回 true。	Less than or equal to	[$a -le $b] 返回 true。

7.2.6 流程控制语句

（1）if判断

```
if [ 条件判断式 ]
  then
    程序
fi
```

3 score=60
4 echo $score
5
6 if [$score -gt 60]
7 then
8 echo "Congrats! Passed!"
9 elif [$score -eq 60]
10 then
11 echo "60 wan sui, duoyifen langfei!"
12 else
13 echo "Failed! Boo"
14 fi

（2）case语句

```
case $变量名 in
 "值1"）
   如果变量的值等于值1，则执行程序1
   ;; break
 "值2"）
   如果变量的值等于值2，则执行程序2
   ;;
 …省略其他分支…
 *） default
   如果变量的值都不是以上的值，则执行此程序
   ;;
esac
```

```
3 # ask user input 1-4 integer
4 echo "Input 1-4:"
5 read num # read an integer and put it into num
6
7 # use case branch
8 case $num in
9   1) echo "you chose one"
10  ;;
11  2) echo "you chose two"
12  ;;
13  3) echo "you chose three"
14  ;;
15  4) echo "you chose four"
16  ;;
17  *) echo "invalid input! What do you want?"
18  ;;
19 esac
```

（3）for循环

```
for (( 初始值;循环控制条件;变量变化 ))
do
  程序
done
```

```
4 sum=0
5 for (( i=1;i<=100;i++ ))
6 do
7   sum=$[$sum+$i]
8 done
9
10 echo $sum
```

（4）while循环

```
while [ 条件判断式 ]
do
  程序
done
```

```
3 sum=0
4 i=1
5
6 while [ $i -le 100 ]
7 do
8   sum=$[$sum+$i]
9   i=$[$i+1]
10 done
11
12 echo $sum
```

7.2.7 函数

```
[ function ] funname[()]   Note: 在这里，中括号代表“可省略的内容”
{
   Action;
   [return int;]
}
funname
```

```
3 function sum()
4 {
5   s=$[$1+$2]
6   echo $s
7 }
8
9 #提示用户输入
10 read -p "input num1: " num1
11 read -p "input num2: " num2
12
13 #调用函数
14 sum $num1 $num2
```

```
[baiaohou@localhost shell]$ sh function.sh
input num1: 3
input num2: 4
7
```