TC的技术含量其实不高，但是涉及的东西很多很杂，网上零零散散的都有，在这里我想总的整理一下，算是栽颗小树吧…**…**

**注：如果你只想自己装个环境写代码的话，请不要看以下写的，真的是无比之烦，建议你看官网的这个或其他博客：**

**http://simspark.sourceforge.net/wiki/index.php/Installation\_on\_Linux；**

**接下来，我将按以下流程讲解一下RoboCup仿真3D比赛环境的搭建方法：**

***1. 硬件要求：***

***（1）电脑硬件要求；***

***（2）交换机硬件要求；***

***（3）网线硬件要求；***

***2. 各个电脑任务分配；***

***3. 网络拓扑结构与IP配置（主要针对配置A、B两组）；***

***4. 系统安装与更新；***

***5. 软件安装与系统配置（最麻烦的地方）；***

***6. 测试与使用。***

***以下是几个主要参考资料和当时最新版软件，如果你要找更新的版本，请自行到相应网站查找：***

***比赛环境搭建说明2013英文版： http://vdisk.weibo.com/s/A9EtgmXv\_PvAg/1414220729***

***simspark-0.2.4 : http://vdisk.weibo.com/s/A9EtgmXv\_Pvfp/1414220781***

***rcssserver3d-0.6.8.1 : http://vdisk.weibo.com/s/A9EtgmXv\_PvOT/1414221074***

***rcssserver3d-0.6.8.1 补丁（没这个补丁问题也不大） : http://vdisk.weibo.com/s/A9EtgmXv\_PvOH/1414221055***

***ode-0.11.1-tbb：http://vdisk.weibo.com/s/A9EtgmXv\_Ptpj/1414236805***

***roboviz：http://vdisk.weibo.com/s/A9EtgmXv\_Ptp6/1414236776***

***proxy：http://vdisk.weibo.com/s/A9EtgmXv\_Ptp8/1414236797***

***rclm2：http://vdisk.weibo.com/s/A9EtgmXv\_Ptp4/1414236789***

***restore.py: http://vdisk.weibo.com/s/A9EtgmXv\_OXft/1414335116***

***shFor3D.tar.gz: http://vdisk.weibo.com/s/A9EtgmXv\_Pg4s/1414380754***

1. 硬件要求：

（1） 电脑：

总共要5台（i5/i7 x 1; 中档以上显卡 x 1; i3 配置以上 x 2; 配置随意 x 1）

因为比赛分为A、B两个小组，一组需要5台，所以这次比赛有10台。这里只以一组举例。

（这次组织方放血了，给的配置都超预期了，另外，那个“中档以上显卡的电脑”最好用英伟达的，显卡驱动比较好装）

（2）交换机：

要一台Gibibit的，也就是千兆交换机，虽说交换机基本都是千兆的，但之前的安徽省比赛就只给了个百兆的……

分A、B两组比赛时，需要两台。

（3）网线：

一般的网线就行。

以上信息，基本就是那个"2013年英文版“的翻译。

2. 各个电脑任务分配：

正所谓“好记性不如烂笔头”，比赛场上有10台电脑，很难搞清谁是谁，所以最好每个都贴个标签。

（1）cpu最强的那台运行rcssserver3d，给它贴个标签：servera;

（名字这么看，“server” + “a”，"server"好理解，”a"表示A组，因为我用A组举例的嘛！）

（2）“中档以上显卡”的那台运行roboviz，给它贴个标签: monitora

（3）cpu次强的两台分别运行两方球队，分别标记：clienta1，clienta2

（4）配置最差的那台，作为NFS和NIS，标记:robocup3dfa；

（为什么是"robocup3d"+"f"+"a"呢？因为这台要装fedora。别看这台硬件配置无所谓，但它却是这个比赛环境的核心！）

3. 网络拓扑结构与IP配置（主要针对配置A、B两组-----10台电脑）：

这步其实就是在说10台电脑接网线的问题。

如果只有一组电脑的话，那倒容易，5台电脑都接在同一个交换机上，并给每台电脑分配同一网域的不同IP即可。

可麻烦就在于有两组电脑，而且正式比赛时还要求各自独立！！！ 我心都碎成渣渣了……

记得常用的两个命令：

查看IP：ifconfig

查看链接通断： ping

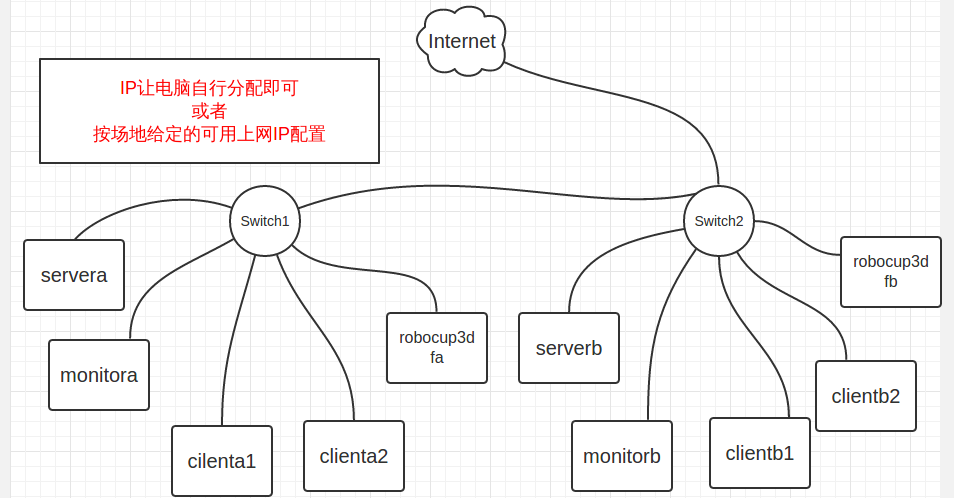
以下将分为三个阶段分别给出以及IP配置结构图：

（1）用于软件更新与软件安装阶段：

因为我们还要系统更新和软件安装，所以还要连上Internet。

这时候的IP地址基本是当地网络自行分配的，也可能要自己设定，但方式不重要，重要的是能上网就行。

那个Internet是连到switch2还是连到switch1上都无所谓咯，这里就连在switch2上了。



（2）用于代码上传和代码测试阶段：

这个时候就要自己给电脑分配IP地址了，以下是我给的IP配置，你可以根据自己的习惯，给自己弄个熟悉的。

monitora: 　　192.168.2.11　　　　monitorb:　　192.168.2.21

servera:   　　192.168.2.12　　　　serverb:　　  192.168.2.22

clienta1:   　　192.168.2.13　　　　clientb1:　　  192.168.2.23

clienta2:   　　192.168.2.14　　　　clientb2:　　  192.168.2.24

robocup3dfa:  192.168.2.19　　　　robocup3dfb:  192.168.2.29

所有的子网掩码(Netmask):255.255.255.0

所有的网关(gateway): 192.168.2.1（明白人应该知道这个是瞎掰的）

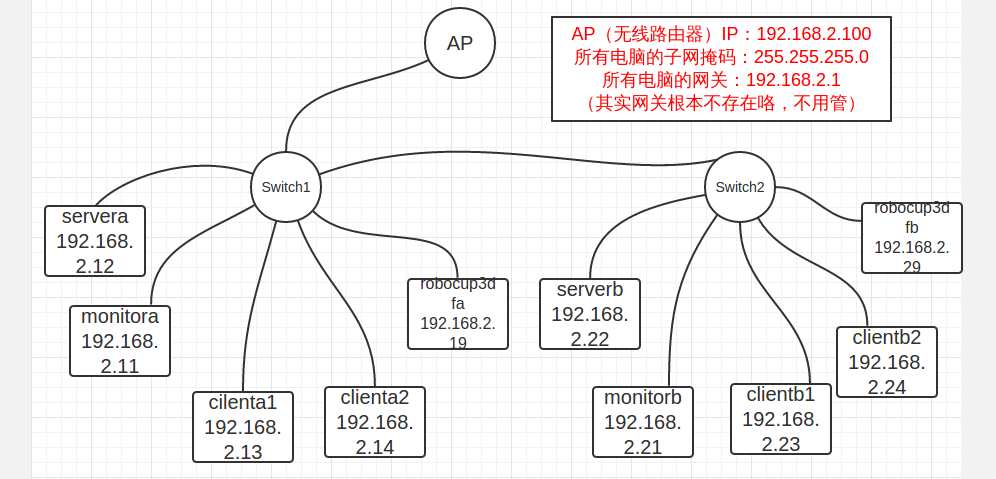
所有的dns: 不用设置（或8.8.8.8）

"AP"指的是无线路由器，它是用来让大家远程上传代码的，给它的IP是“192.168.2.100”，“100”是随便选的，只要别和上面的一样，随你怎么配。

"AP"是连在switch1还是switch2上都无所谓，这里我连在switch1上了。

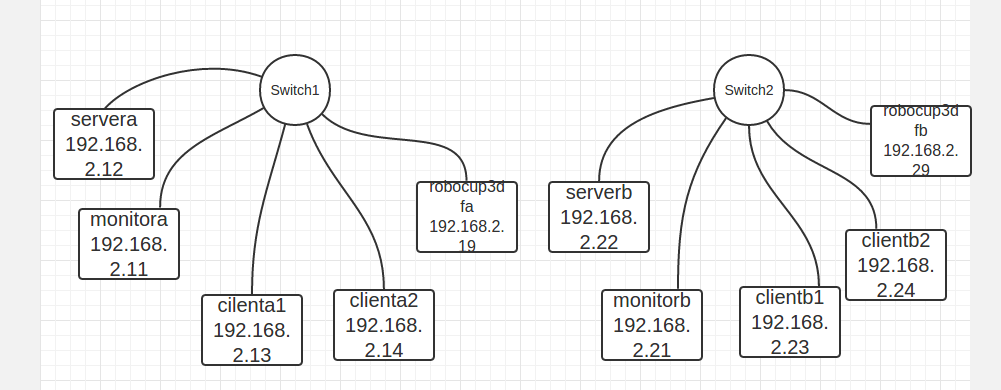
(这里要说明一下为什么是“192.168.2.xxx“而不是"192.168.1.xxx"，因为无线路由器经常分配192.168.1.xxx"，为了避免冲突，就用".2"了，当然".3"、“.4"也可以咯。

如果你对网络不熟悉的话，这句话的进一步理解还得到实践中去体会。)



（3）用于正式比赛：

这就让A、B组的机器独立了，不会相互干扰了。



4. 系统安装与更新：

继续只用A组举例：

本次比赛使用两种linux操作系统：

Ubuntu 12.04 64bit：monitora，servera，clienta1，clienta2

Fedora 20 64bit：robocup3dfa

注意：

1. 在系统安装过程中，会提示设置初始用户的用户名，Ubuntu还会提示设置主机名。

2. 为了方便记忆考虑，推荐就按以上名字设置：如monitora装Ubuntu时，相应的初始用户名和主机名就都设为monitora。

3. 建议所有电脑用同一个密码，以后用起来方便。

系统更新：

Ubuntu:

切换root身份：  
sudo -i  
  
更新：  
apt-get update -y

Fedora:

切换root身份  
su -  
  
先装一个自动搜索最快镜像的插件  
yum install yum-plugin-fastestmirror.noarch -y  
  
更新：  
yum update -y

5. 软件安装与系统配置（最麻烦的地方）：

首先，我们回顾一下“2. 各个电脑任务分配”里说的东西：

servera：运行rcssserver3d；

monitora：运行roboviz；

clienta1和clienta2 :运行球队程序；

robocup3dfa：做NFS和NIS;

安装软件当然要根据它们的不同作用来安装咯！

为了以后配置考虑，我们先给所有的Ubuntu系统配置root密码:

passwd root

好了，那么就先从

**（1）servera开始吧：**

安装ssh服务：

apt-get install openssh-server openssh-client -y

安装vim编辑器（如果需要的话）：

apt-get install vim -y

安装编译器：

apt-get install g++ -y

安装各种其他依赖库和相关工具：

apt-get install subversion cmake libfreetype6-dev libode-dev libsdl-dev ruby ruby-dev libdevil-dev libboost-dev libboost-thread-dev libboost-regex-dev libboost-system-dev libtbb-dev build-essential automake libtool git libboost-program-options-dev libboost-thread-dev aptitude -y  
  
编译和安装多线程ode（建议不要装，装过之后server很容易崩溃，而且没它也不影响各个球队表现(多亏KylinSky我才知道这个情况的!)。只是标准要求要用，不过这次比赛就没装它）:  
cd ode-0.11.1-tbb/  
 ./autogen.sh  
 ./configure --enable-shared --enable-malloc LIBS=-lX11  
 make -j && make install  
  
编译安装simspark(我这里就不说明补丁了，说了反而添堵，自己看里面的README，很容易自己加上去)：

下载simspark文件

$ svn co https://svn.code.sf.net/p/simspark/svn/trunk simspark

Configure the package, generate build files and install the package:

**Simspark:**

$ cd simspark/spark

$ mkdir build

$ cd build

$ cmake ..

$ make

$ sudo make install

$ sudo ldconfig

**Note: type 'make uninstall' to uninstall the package**

**Rcssserver3D:**

[ set to the rcssserver3d directory (rcssserver3d) ]

$ mkdir build

$ cd build

$ cmake ..

$ make

$ sudo make install

$ sudo ldconfig

**Note: type 'make uninstall' to uninstall the package**

**Make sure the linker can find your shared libraries:**

$ sudo gedit /etc/ld.so.conf

[ add the line '/usr/local/lib' if it isn't already there, save and close ]

$ sudo ldconfig

**Run the simulation：**

$rcssserver3d

现在就可以运行球场平台了，试试吧！

**（2）monitora:**

安装ssh服务：

apt-get install openssh-server openssh-client -y

安装vim编辑器（如果需要的话）：

apt-get install vim -y

安装编译器：

apt-get install g++ -y

安装各种其他依赖库和相关工具：

apt-get install subversion cmake libfreetype6-dev libode-dev libsdl-dev ruby ruby-dev libdevil-dev libboost-dev libboost-thread-dev libboost-regex-dev libboost-system-dev libtbb-dev build-essential automake libtool git libboost-program-options-dev libboost-thread-dev aptitude -y

(其实以上和servera的安装一样，其实还多装了一些，图个省心，就这样了)  
  
安装java，为了运行roboviz：  
 apt-get install openjdk-8-jdk openjdk-8-jre -y  
  
配置roboviz：

如果给的软件内部没有bin文件夹，则  
cd roboviz-dev  
./build-linux64.sh  
如果存在bin文件夹，则可直接进行配置  
设置roboviz的server的IP：  
其实现在你就可以打开roboviz了，只是一片黑。  
这是理所当然的，要知道monitora上是没有rcssserver3d的，rcssserver3d在servera上。  
修改文件：roboviz/bin/linux-amd64/resources/config.txt和roboviz/config.txt  
将其中： Server Host          : localhost  
改为： Server Host          : 192.168.2.12  
  
安装rclm2（全名RoboCupLeagueManager），它的作用估计也能猜出个大概了吧，先装着，回头还要说它：  
./configure  
make  
 make install

现在，你可以在servera上运行rcssserver3d，而在monitora上运行roboviz了，这样负荷就从一台机器转移到两台机器上去了（当然，还有交换机）。

**（3）clienta1和clienta2：**

安装ssh服务：

apt-get install openssh-server openssh-client -y

安装vim编辑器（如果需要的话）：

apt-get install vim -y

安装编译器：

apt-get install g++ -y

安装各种其他依赖库和相关工具：

apt-get install subversion cmake libfreetype6-dev libode-dev libsdl-dev ruby ruby-dev libdevil-dev libboost-dev libboost-thread-dev libboost-regex-dev libboost-system-dev libtbb-dev build-essential automake libtool git libboost-program-options-dev libboost-thread-dev aptitude -y

安装java，为了运行proxy：

apt-get install openjdk-8-jdk openjdk-8-jre -y

（和monitora一样，只是java的作用不一样）

ubuntu都OK啦！

只要Ubuntu们OK了，基本的比赛环境也就可以了。

现在可以就可以按照以下方式开始比赛了：

servera 运行 rcssserver3d

monitora 运行 roboviz

clienta1 运行 ./start.sh 192.168.2.12

clienta2 运行 ./start.sh 192.168.2.12

“原来这样就可以啦！”

嗯……确实可以这么说，不过，还有几个问题，你要是现在不给解决了，后头有的是苦吃的：

问题1：你要让各个球队怎么上传代码？U盘拷贝不是不行，只是到时那么多人，不但有上传，还有测试的，测试不行还有再上传的……有你受的了；

问题2：不要忘记哦！每有一队上传代码，你不但要给clienta1传一份，还得要给clienta2也要传一份，人一多，事儿一多，就有你混乱的了；

问题3：一个组基本有6到7支队伍，打单循环赛，你要怎么让每两队之间打一场？是让一支队伍先和剩下所有队伍打吗，然后再让一支队伍和剩下所有球队打吗？

         虽说是机器人，咱也不能这么累着人家啊。比赛场次安排也得有个规范的。

**（4）robocup3dfa：**

TC的价值如何体现，就是要看这台机器配置的怎么样了，不多说了，开始吧：

以下有很多配置，我是用自己写的shell脚本来完成的，每个脚本的功能都很简单：[shFor3D.tar.gz](http://vdisk.weibo.com/s/A9EtgmXv_Pg4s/1414380754)

 （**robocup3dfa上的所有命令均已root身份运行，除非有特别说明处！！！！！！**）

先是些基本软件工具：

安装vim编辑器（如果需要的话）

yum install vim -y;

安装syslog(当时很惊讶，fedora20居然没有自带这个服务)

yum install rsyslog.x86\_64 -y

开启syslog服务：

systemctl enable rsyslog

systemctl start rsyslog

在下载下来的shFor3D.tar.gz，有我写的一个用于远程测试的c源程序，很简单，两个文件共115行（包括空行），先解压编译，已备后用（更多说明见“其他重要信息”）：

解压sim3dStart

tar -axf sim3dStart.tar.gz

编译sim3dStart服务器和客户端

cd sim3dStart/

gcc sim3dStart-server.c -o sim3dStart-server

gcc sim3dStart-client.c -o sim3dStart-client

I. 安装和配置NFS服务器：

啥是NFS（NetworkFileSystem)服务器呢，简单点说，就是个远程U盘，可以通过网络“插”在多台电脑上。

更进一步可参考：[《鸟个的linux私房菜（服务器）》----NFS服务器](http://vbird.dic.ksu.edu.tw/linux_server/0330nfs.php)

安装NFS需要的两个软件：

yum install rpcbind.x86\_64 -y;

yum install nfs-utils.x86\_64 -y;

shFor3D里有个teams文件，用来罗列队名的，很重要！shFor3D里的脚本基本都用到它（一行一个队名，以下是今年的示例,其中ccwss1和ccwss2为测试用户）：  
ccwss1  
ccwss2  
AHJZU\_3D  
Apollo3D  
CIT3D  
Dreamwing3D  
HS  
HfutEngine3D  
Icarus  
Kylinsky3D  
Miracle3D  
NBUT\_kylin  
SEU\_Jolly  
Strive3D  
YuShan3D  
  
创建NFS开放的目录（其实我要开放的是各个参赛队伍的主目录，当然也要同时创建这些用户咯），

./UsersInit.sh

运行过之后会产生一个usersPasswd.txt的文件，名面有对应各个用户的用户名与密码

（其实这个脚本还有另一种用法，参见“其他重要信息”）  
  
如果有用户信息错误，以下命令可以删除所有用UsersInit.sh创建的用户：  
./UserDel.sh  
  
现在每个用户已被创建，并且主目录被建立在/robocup3D下，比如ccwss1的主目录为：/robocup3D/ccwss1  
  
设置NFS的配置文件，即标出要开放那些目录： /etc/exports  
在文件中添加以下内容：  
/robocup3D/ccwss1       127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/ccwss2       127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/AHJZU\_3D     127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/Apollo3D     127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/CIT3D        127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/Dreamwing3D  127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/HS           127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/HfutEngine3D 127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/Icarus       127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/Kylinsky3D   127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/Miracle3D    127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/NBUT\_kylin   127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/SEU\_Jolly    127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/Strive3D     127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
/robocup3D/YuShan3D     127.0.0.1(rw,no\_root\_squash)    192.168.2.0/24(rw,no\_root\_squash)  
  
开放防火墙（也是个脚本，每次重启时都要运行一下）：  
./iptables\_rule.sh  
  
开启NFS服务：  
systemctl enable nfs-server.service  
systemctl start nfs-server.service

到这里，robocup3dfa上NFS服务器就算配置完毕了！

现在我们要暂且回到ubuntu，去安装NFS客户端（每台Ubuntu都要按装）：

安装nfs客户端（aptitude和apt-get的功能一样，因为这是从别人blog上抄的，尊重原版，我也就这样写了）

aptitude -y install nfs-common

好啦！你现在已经完成NFS的全套配置了。

你在Ubuntu机器上来这样测试一下(注意，此时要先按“3. 网络拓扑结构与IP配置”中“（2）用于代码上传和代码测试阶段”的方式来设置fedora和ubuntu的IP)：

showmount -e 192.168.2.19

成功的话有类似下列输出：

Export list for localhost:

/robocup3D/ccwss1 127.0.0.1(rw,no\_root\_squash) 192.168.2.0/24(rw,no\_root\_squash)

/robocup3D/ccwss2 127.0.0.1(rw,no\_root\_squash) 192.168.2.0/24(rw,no\_root\_squash)

/robocup3D/AHJZU\_3D 127.0.0.1(rw,no\_root\_squash) 192.168.2.0/24(rw,no\_root\_squash)

........................................（以下省略）........................................

基本就是和你刚刚配置的在fedora上的配置文件/etc/exports一样

2012年和2013年我都做过国赛的TC助手，这两界把NFS搭建完毕之后，差不多就算整个环境搭建完成了。

现在我们不妨测试一下(注意，此时要先按“3.网络拓扑结构与IP配置”中“（2）用于代码上传和代码测试阶段”的方式来设置fedora和Ubuntu的IP)，

在clienta1上操作：  
新建两个文件夹：

mkdir -p myccwss1 myccwss2  
  
挂载NFS的两个开放目录到刚刚新建的文件夹中：

mount -t nfs 192.168.2.19:/robocup3D/ccwss1 myccwss1

mount -t nfs 192.168.2.19:/robocup3D/ccwss2 myccwss2  
  
在clienta2上操作：  
（同clienta1）

这时你可以在clienta1(或clienta2)上复制两份可执行文件分别放到myccwss1和myccwss2里。  
假设你是在clienta1上进行复制的，此时你会惊奇的发现clienta2里的myccwss1和myccwss2也有了刚刚复制的文件。  
这就是NFS，一台电脑的硬盘（robocup3dfa)被多台电脑分享了。  
  
还是按刚刚的方式：

servera 运行 rcssserver3d

monitora 运行 roboviz

clienta1 运行 ./start.sh 192.168.2.12

clienta2 运行 ./start.sh 192.168.2.12

开始比赛吧！

不过，本着“不作死就不会死”的精神，我们从Ubuntu回到fedora上继续（如果你刚刚进行了测试，请还原IP到能上网的状态）：

II. 安装和配置NIS服务器：

啥是NIS（NetworkInfomationSystem)服务器呢？说不好，总之是用来统一管理用户等一系列信息用的。

它主要有两个程序组成：ypserv和yppasswd。

更进一步可参考：[《鸟个的linux私房菜（服务器）》----NIS服务器](http://vbird.dic.ksu.edu.tw/linux_server/0430nis.php)

安装NIS服务器程序（ypserv），同时这也是robocup3dfa上最后要安装的软件，安装完成后请按“3. 网络拓扑结构与IP配置”中“（2）用于代码上传和代码测试阶段”的方式来设置robocup3dfa的IP：

yum install ypserv.x86\_64 -y

设置NIS(ypserv)配置文件，设置域名与固定ypserv端口：/etc/sysconfig/network

(在文件最后追加)  
NISDOMAIN=robocup3dfa   
YPSERV\_ARGS="-p 1011"  
  
设置NIS(ypserv)配置文件，设置NIS服务网域：/etc/ypserv.conf  
(在文件最后追加）  
127.0.0.0/255.255.255.0 :       \*:      \*:      none  
192.168.2.0/255.255.255.0:      \*:      \*:      none  
  
设置NIS配置文件，设置IP与域名对应关系：/etc/hosts  
(在文件最后追加)

　　192.168.2.11　　　　monitora

　　192.168.2.12　　　　servera

　　192.168.2.13　　　　clienta1

　　192.168.2.14　　　　clienta2

　　192.168.2.19　　　　robocup3dfa

设置NIS(yppasswd)配置文件，固定端口：/etc/sysconfig/yppasswdd

(在文件最后追加)  
YPPASSWDD\_ARGS="--port 1012"  
  
设置主机名（重启后无效）：  
hostname robocup3dfa  
设置主机名（重启后有效）：/etc/hostname  
（在文件最后追加）  
robocup3dfa  
  
开启ypserv和yppasswd服务：  
systemctl enable ypserv  
systemctl start ypserv  
systemctl enable yppasswdd  
systemctl start yppasswdd  
  
初始NIS数据库：  
/usr/lib64/yp/ypinit -m  
（之后会弹出确认相关信息，基本都是OK，第一个确认是按“ctrl+D", 第二个确认是输入“y"回车，自己仔细阅读看一下吧）

现在我们要暂且回到ubuntu，去安装NIS客户端（每台Ubuntu都要安装）：

安装NIS客户端程序（注意，安装过程中不要走神，中途会弹出窗口让你设置NIS服务器域名，当然，我们得输入：robocup3dfa）

(同时这也是所有Ubuntu上最后要安装的软件，安装完成后请按“3.网络拓扑结构与IP配置”中“（2）用于代码上传和代码测试阶段”的方式来设置Ubuntu们的IP：

aptitude -y install nis

设置NIS客户端配置文件,指定NIS服务器IP： /etc/yp.conf  
（在文件最后追加）  
ypserver 192.168.1.19  
  
设置NIS客户端配置文件，给本机加入NIS服务切换规则：  
（就修改红字标注的那些行，其实这几行中"nis"的先后顺序是有蹊跷的，关于“hosts“那行，你若把“nis"放句尾也行，只是在你用sudo，或切换用户，或ping 主机名时……自己可以试试，反正客户端有好几台）  
passwd: compat nis   
group: compat nis  
shadow: compat nis  
hosts: files nis mdns4\_minimal [NOTFOUND=return] dns mdns4  
  
测试NIS客户端:  
yptest  
成功的话，会产生如下输出：

*Test 1: domainname*

*Configured domainname is "robocup3dfa"*

*Test 2: ypbind*

*Used NIS server: robocup3dfa*

*Test 3: yp\_match*

*WARNING: No such key in map (Map passwd.byname, key nobody)*

*....(中间省略)....*

*Test 6: yp\_master*

*robocup3dfa*

*....(中间省略)....*

*Test 8: yp\_maplist*

*passwd.byname*

*protocols.byname*

*hosts.byaddr*

*hosts.byname*

*....(中间省略)....*

*Test 9: yp\_all*

*ccwss1 robocup3d:$1$U9Gccb60$K5lDQ.mGBw9x4oNEkM0Lz/:2001:2000::/robocup3D/ccwss1:/bin/bash*

*....(中间省略)....*

*1 tests failed*  
  
总过有9个”test"有一个failed，不过没有关系，NIS照样可以用。

给每个用户统一设置挂载点：

创建挂载的入口目录（全都在/robocup3D/\*\*\*）：

./mkdir\_home.sh

挂载(每次开机重启时，都得要执行):

./mount\_nfs.sh（注意事项，參见“其他重要信息”）

我们在Ubuntu上来测试NIS吧，

a,用户登录测试：

在clienta1电脑上（密码在之前用UsersInit.sh生成的usersPasswd.txt文件里找）：

su - ccwss1

密码：

b,修改密码测试：

yppasswd

旧密码：

新密码：

再输一次：

OK啦！这次真的是全部OK了，可是，可是……作死的血液又在沸腾了.

如果有人不小心传错文件了，或有故意恶搞，塞个几十G的文件过来，那还不得把服务器撑爆吗？

所以，我就给每个用户添加了硬盘容量使用限制，主要用的软件叫quota，不需要安装，fedora自带，那么我们就愉快的开始吧！

（事实证明quota就是多余的，大家都是中华民族的好儿女，没人恶搞。最最重要的一点就是：那破无线路由器，唉……，组织方给的所有硬件都很到位，就这无线路由器最煞风景了，据说是那天临时从一个办公室抢来的。）

回到robocup3dfa上

让根目录支持quota：

mount -o remount,usrquota,grpquota /

让根目录长期支持quota，修改/etc/fstab  
将这行：  
UUID=5ab7082c-41f4-482e-8a0c-fd90d92884b6 　　/               ext4    errors=remount-ro　　0       1  
改为（你的电脑上应该会有些不一样，但是只要记住是在倒数第三项上追加红色文字就行了）：  
UUID=5ab7082c-41f4-482e-8a0c-fd90d92884b6 　　/               ext4    errors=remount-ro,usrquota,grpquota 　　0       1  
  
开启quota：  
quotaoff -vaug  
quotacheck -avugm  
quotaon -vaug  
  
给每个用户设置500M容量限制（应该没有500M，大概480M多一点的样子）：  
./quota.sh

好了，你可以到clienta1上测试一下：

切换ccwss1用户：

su - ccsww1

创建一个400M的文件并查看：

dd if=/dev/zero of=bigfile1 bs=1M count=400

ls -lh

再创建一个400M的文件并查看（这时你看到bigfile2应该只有180几M）

dd if=/dev/zero of=bigfile1 bs=1M count=400

ls -lh

测试完成后，就把两个bigfile删了吧。

6. 测试与使用：

首先，咱们确认一下，我们已经将每台机器的IP都设置成了192.168.2.xxx了，现在我们再接上无线路由器（AP），并设置其IP。

请按“3. 网络拓扑结构与IP配置”中“（2）用于代码上传和代码测试阶段”的方式来设置IP和网线接法。

复制一份先前编译好的sim3dStart-server到servera上（还记得这个吗）

servera 运行 sim3dStart-server

monitor 运行 roboviz  
此时roboviz会是一片漆黑；  
  
接下来，将要模拟各个队伍的代码上传与测试(以下操作均在笔记本上完成)：  
  
首先连上无线路由器。  
  
(这时我们不必要用root身份）  
远程上传可执行文件：  
scp -rp myBinary ccwss1@192.168.2.13:/robocup3D/ccwss1  
  
远程登录ccwss1：  
ssh ccwss1@192.168.2.13  
（此时的远程登录机为clienta1，登录后，你会发现用scp传上来的文件）  
  
远程启动rcssserver3d：  
restartSim3d  
这时roboviz就会显示球场了；  
  
运行球队代码：  
./start.sh 192.168.2.12  
如果clienta1上运行了proxy，则只需运行(proxy的更多说明见“其他重要信息”)：  
./start.sh  
或  
./start.sh 127.0.0.1  
  
  
这时你可以再用另一个用户登录，上传并运行代码来进行比赛测试（注意有一些不同）：

远程上传代码：  
scp -rp myBinary ccwss2@192.168.2.14:/robocup3D/ccwss2  
  
远程登录ccwss2：  
ssh ccwss2@192.168.2.14  
（此时的远程登录机为clienta2，，登录后，你会发现用scp传上来的文件）  
  
运行球队代码：  
./start.sh 192.168.2.12  
如果clienta2上运行了proxy，则只需运行(proxy的更多说明见“其他重要信息”)：  
./start.sh  
或  
./start.sh 127.0.0.1  
  
如果其他球队要再测试的话，只要运行restartSim3d就能重启rcssserver3d了。

最后冲刺了！马上就可以完成啦！

**不做特别说明，以下我们均已root身份执行命令！！！**

接下来我们要操作monitora这台电脑

各台Ubuntu的root密码之前设了没？就在“4. 软件安装与系统配置”一开始的地方

设置免密码登录：

ssh-keygen

然后一路回车；

ssh-copy-id root@clienta1  
输入clienta1的root密码

ssh-copy-id root@clienta2  
输入clienta2的root密码

ssh-copy-id root@servera  
输入servera的root密码  
  
此时你在monitora上登录任意上面任意一台电脑都不用密码了，可以像下面这样试试：  
ssh root@clienta1

配置rclm2：

修改/usr/local/share/types/3Dspark/config文件（具体路径可能不一样，主要是types/3Dspark/config这个文件）：

NET\_PREFIX=10.10.0 ==>> NET\_PREFIX=192.168.2

SERVER=${NET\_PREFIX}.1 ==>> SERVER=${NET\_PREFIX}.2

CLIENT1=root@${NET\_PREFIX}.2 ==>> CLIENT1=root@${NET\_PREFIX}.3

CLIENT2=root@${NET\_PREFIX}.3 ==>> CLIENT2=root@${NET\_PREFIX}.4

这时候你应该可以猜出我上面在干什么吧；  
  
初始化rclm2：  
rclm2 groupA  
cd groupA  
./script/init 3Dspark  
  
创建文件teamsOfA，列出小组中各参赛队用户主目录（以下以今年的A组为例）：  
/robocup3D/AHJZU\_3D  
/robocup3D/Apollo3D  
/robocup3D/Dreamwing3D  
/robocup3D/Icarus  
/robocup3D/Kylinsky3D  
/robocup3D/Miracle3D  
  
  
编制A组赛程安排：  
./script/schedule/init teamsOfA  
这时会产生一个文件var/schedule。打开看一下，差不多是这个样子：  
/robocup3D/AHJZU\_3D vs /robocup3D/Apollo3D  
/robocup3D/Apollo3D vs /robocup3D/Dreamwing3D  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*以下省略＊＊＊＊＊＊＊＊＊  
  
如果用rclm2来管理比赛的话，比赛场次顺序就是按照以上那个文件来的。  
  
让rclm2启用roboviz，修改script/game/start文件：  
将  
pushd ~/Desktop/roboviz/  
改为你自己的roboviz路径，我修改的路径为：  
pushd /home/monitora/Desktop/roboviz/bin/linux-amd64/  
  
终于到最后一步了！当每个球队都上传了代码后，运行以下命令，正式比赛：  
./script/start

其他重要信息：

1. 关于restartSim3d:

（1）一个bug:

sim3dStart-server有一个bug，就是当在servera上同时启动sim3dStart-server和rcssserver3d时，用于远程重启rcssserver3d的restartSim3d会运行失败。

（2）restartSim3d脚本：

restartSim3d是个shell脚本，就一行命令：

sim3dStart-client servera

猜猜看，如果换成在B组，这个命令会产生什么结果吗？

如果我们要用在B组，我们应当把servera改成serverb（或者是IP：192.168.2.22也行）

2. UsersInit.sh的用法：

像这样运行程序：

./UsersInit.sh

程序会创建teams里的所有用户，并随机产生6位密码，然后保存至usersPasswd.txt文件里。这里有个问题。

我常常想将用户和密码信息提前1到2个星期通知个参赛队，但是要知道我这时还没有搭建环境呢！

到了在会场再次运行UsersInit.sh时，它又会创建新的密码。这可麻烦了。那么提前告知大家用户名与密码有什么用呢？

鉴于这点，这个脚本还有个用法

./UsersInit.sh oldUsersPasswd.txt

其中oldUsersPasswd.txt可以是你之前运行./UsersInit.sh时得到的usersPasswd.txt(然后把它改名为oldUsersPasswd.txt以示区别)，

这种用法，程序会根据oldUsersPasswd.txt里的用户名与对应的密码来进行设置，也就是说旧密码有效。

3. 电脑重启注意事项：

电脑重启的顺序很重要，否则……试试看吧，试过之后印象更深刻。

先打开：fedora；

并且要确认：网络配置是否为192.168.2.19

并且要打开防火墙：./iptables\_rule.sh

此时才能打开Ubuntu，而且还要确实IP配置和执行挂载命令 ./mount\_nfs.sh

4. proxy：

为了和国际接轨，这个是今年国赛新加的东西，需要运行在clienta1和clienta2上，按正统规则来说，这个东西是属于标准的，但它与今年13支参赛队中的6支都不兼容（具体表现为所有的球员都不能正常走路），所以我也给取消不用了，不过既然是标准，我就讲讲用法吧。

（操作servera电脑）  
切换root身份：

su -

设置rcssserver3d的Agent的端口，从3100改为3110： /usr/local/share/simspark/spark.rb（路径可能不同）

$agentPort = 3100 ==>> $agentPort = 3110

删除临时配置文件：

rm -rf ~/.simspark  
  
(在clienta1和clienta2电脑上操作）  
解压并运行proxy：  
tar -axf proxy.tar.gz  
./startProxy.sh 192.168.2.12 3110 3100  
  
此时，本来运行球队的命令：  
./start.sh 192.168.2.12  
就可以改为：  
./start.sh  
或  
./start.sh 127.0.0.l  
  
如果要用rclm2的话，还要修改rclm2的相关配置文件内容（见下）

5. rclm2的使用说明：

rclm2主要做以下5件事儿：

（1）复制并修改rcssserver3d的配置文件到servera的root主目录里；

（2）编制单循环赛的比赛场次安排；

（3）按顺序启动rcssserver3d, roboviz, 球队1的代码， 球队2的代码

（4）rcssserver3d崩溃时，记录并重置比赛时间和比分；

（5）压缩log文件；

第（1）件事儿挺重要的，它会在./script/start第一次运行后完成，不管你用不用rclm2，最好还是运行一次这个程序。

第（2）件事儿也最好让rclm2帮你做，毕竟它编制出的比赛场次比较合理。

第（3）件事儿嘛，是正式比赛时重复最多的步骤，其实当你习惯了用ssh手动启动时，不用rclm2也无所谓。

第（4）件事儿也挺重要的，这主要是因为我们的rcssserver3d经常会崩溃（crash）（参加过比赛的老队员一般知道），比如在比赛到了153秒的时候，roboviz会突然显示

　　　　　　"disconnected"，于是比赛要重开，那么rclm2会自动恢复比赛时间和比分。如果你不想用rclm2的话，手动也能完成这个工作：

打开root主目录下的naosoccersim.rb文件（之前在./script/start后产生的文件），记得还得是root身份哦：  
cd

vim naosoccersim.rb  
  
其中大约在118，119行的样子，  
是这样两行：  
obj.setTime(0)  
obj.setScores(0,0)  
改为：  
obj.setTime(153)  
obj.setScores(2,1)  
重开rcssserver3d和roboviz，这时时间是从153开始了，比分一开始就是2：1.  
  
如何找到rcssservver3d崩溃时的时间？（最好先看看“6. 不用rclm2，而用手动启动rcssserver3d的注意事项”）  
root主目录里有个sparkmonitor的log文件，用vim打开它，并跳转到最后一行，行首就标注有time，那个就是崩溃时的时间。  
至于比分嘛，这个大家都会记得的。

第（5）件事儿基本没人关注，它会把每次比赛的log文件从servera里传到monitora里，并压缩保存进groupA/archives文件夹里。如果要把restore.py整合进rclm2里面，这个还是有用的。

6. 不用rclm2，而用手动启动比赛时的注意事项：

用rclm2在./script/start第一次配置之后，若要改用手动启动rcssseerver3d和球队代码的话，

在启动rcssserver3d时一定要以root身份，并且得在root主目录下运行rcssserver3d命令，否则之前的用rclm2做的相关配置就会失效，总重要的是不会产生sparkmonitor这个log文件。

7. restore脚本：

“都是rcssserver3d崩溃惹的祸“，如果就在你快要进球时，rcssserver3d崩溃了怎么办？以前的国赛，只能自认倒霉。现在，这个脚本可以回复rcssserver3d崩溃前的样子，

但要用sparkmonitor这个log文件。

在servera机器上：

在rcssserver3d崩溃后做的第一件事，保存sparkmonitor：

cp sparkmonitor sp.backup

重开比赛，并按之前的左右场顺序上双方球员，待所有球员准备完毕后，恢复状态：

./restore sp.backup

你可以自己测试一下，先让比赛比到一半，然后杀掉rcssserver3d，再按以上步骤恢复球场状态。

这个说的是手动的情况，如果要用rclm2，你得要修改script/game/start这个脚本，今年A组没用rclm2，B组用了，至于怎么修改的，我忘备份了，不好意思……

如果你使用了proxy，那么还得在多修改一些地方：

修改script/game/start，大概在第57和第62行：

将

ssh ${CLIENT1} "su - $NAME1 -c ‘./start.sh $SERVER &> log/start.log‘" &

ssh ${CLIENT2} "su - $NAME2 -c ‘./start.sh $SERVER &> log/start.log‘" &

修改为：

ssh ${CLIENT1} "su - $NAME1 -c ‘./start.sh 127.0.0.1 &> log/start.log‘" &

ssh ${CLIENT2} "su - $NAME2 -c ‘./start.sh 127.0.0.1 &> log/start.log‘" &

8. mount\_nfs.sh使用说明：

这个脚本很短，就13行（注释就占了6行），不妨打开看一下。

注意这样一行：

mount -t nfs robocup3dfa:$HomeDir/$username $HomeDir/$username

你应该能看出来，如果把这个脚本用到B组而不做修改，肯定会出问题的（可能到正式比赛时才会显现），用到B组时，我们要修改为：

mount -t nfs robocup3dfb:$HomeDir/$username $HomeDir/$username

一点其他建议：

组织一场比赛不但得有TC，还要有OC（OrganizationCommittee)。可国内比赛不分这个，以上是TC分内的事儿，但还有OC的事儿也得你自己来，

什么“统计比分”啦、“发布通知”啦、“联系各方”啦，“编写规则”啦，“打印奖状”啦……总之有你烦的了！一定不要觉得费时间，耐下心思，简单培训几个会场的志愿者（如果没有志愿者………………），都是些很简单的活儿，虽然一开始可能有点不熟悉，不过很快就可以上手的。