

第一部分 Linux 实践作业

1、自己有电脑的同学，安装一下 Linux 系统，可以参考实验室理论组的靳国强师兄写的安装教程（<http://lug.ustc.edu.cn/www/activities/lug200711/linuxinstall.pdf>）。如果安装的是 Ubuntu 系统，最好把“源”设置成学校的镜像，设置方法如下：

(1)、用我们提供的例子（<ftp://202.38.75.52/incoming/sources.list>）替换/etc/apt/sources.list文件。（方法：a、下载（终端下可以用 wget 或 lftp 下载，具体方法自己可以 man）到本地，假设保存为~/sources.list；b、执行 `sudo mv ~/sources.list /etc/apt/sources.list`）

(2)、执行：`sudo apt-get update`
`sudo apt-get upgrade`（可选）

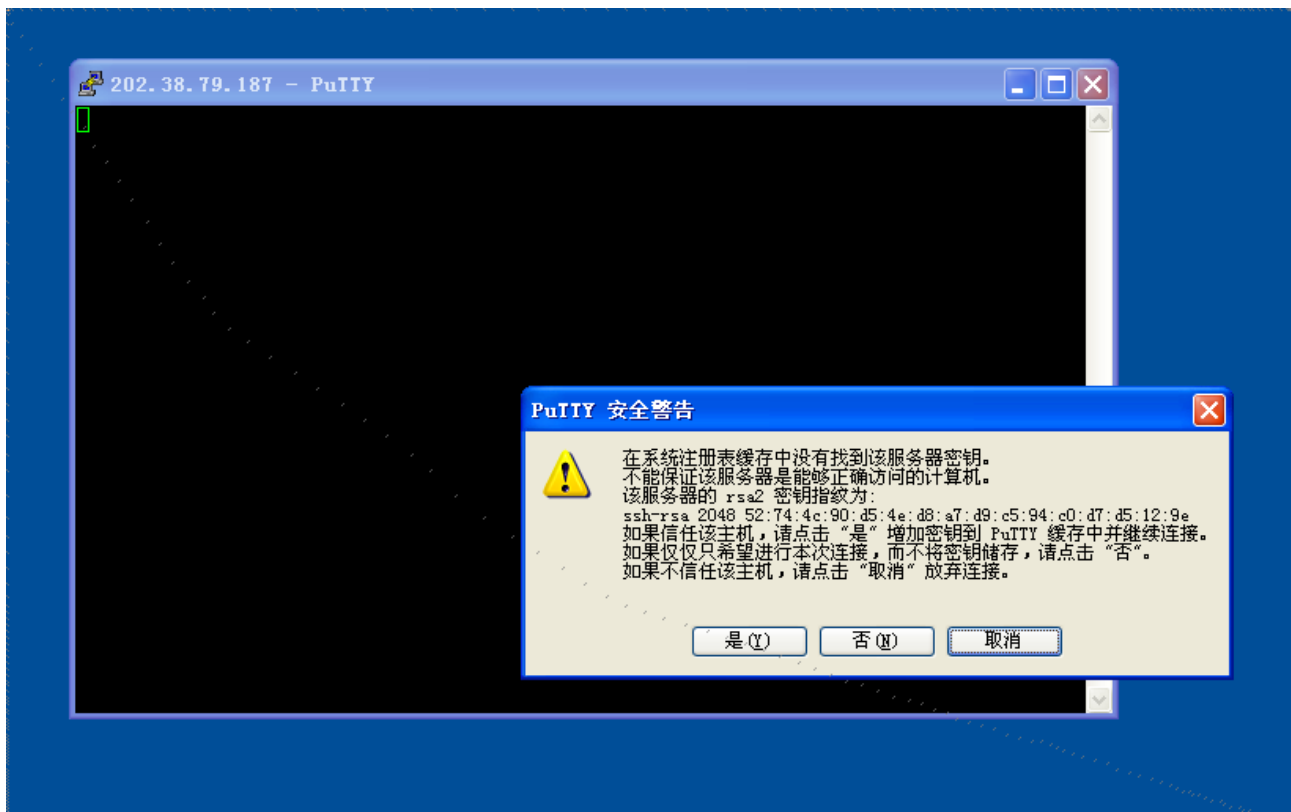
2、暂时还没有电脑的同学，可以利用平时机房上机的时间登入到我们实验室的服务器上先练习使用。登入的方法如下：

(1)、下载并安装 [putty](#)。

(2)、打开 putty，填入主机 IP 地址：202.38.79.187，如下图所示：



(3)、点击“打开”按钮，出现如下画面，这是第一次登入的警告信息，点击“是”按钮。



(4)、出现登入画面后，填写用户名 public:



(5)、按回车键，输入密码: public (注意: 屏幕上不显示*号)，再按回车键，登入成功，画面如下:

```
public@song: ~
login as: public
public@202.38.79.187's password:
Linux song 2.6.18-4-686 #1 SMP Wed May 9 23:03:12 UTC 2007 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Dec 12 16:05:12 2007 from baj-desktop.local
public@song:~$
```

(6)、这时可以练习一些命令的使用，如：

```
public@song: ~
public@song:~$ ls
Welcome
public@song:~$ cat Welcome
# Welcome to WrightEagle's public server! #
public@song:~$ date
Wed Dec 12 16:26:33 CST 2007
public@song:~$ uptime
 16:26:48 up 72 days, 23:52,  2 users,  load average: 0.02, 0.01, 0.00
public@song:~$ whoami
public
public@song:~$ id
uid=1009(public) gid=1009(public) groups=1009(public)
public@song:~$ free -mt
              total          used          free      shared    buffers     cached
Mem:           1011             899             111           0          231          470
-/+ buffers/cache:           198             813
Swap:           1906              0           1906
Total:          2917             899           2017
public@song:~$
```

3、熟悉常用命令（《[GNU/Linux 入门简介](#)》里面有介绍，也可以查看其它书籍（图书馆有很多））。从培训的主页（<http://home.ustc.edu.cn/~shike15/2D/index.html>）下载 [rcssbase](#), [rcssserver](#), [rcssmonitor](#) 源码包，根据《[Install System and Server](#)》的提示安装好 Server 和

Monitor（提示：安装 Base/Server 时需要 boost 库，安装 Monitor 时需要 X11 开发包。Ubuntu 下可以用命令：

```
sudo apt-get install libboost-dev
sudo apt-get install libboost-filesystem1.34.1
sudo apt-get install libboost-filesystem-dev
sudo apt-get install libx11-dev
```

完成安装）。下载 [UVA](#) 的底层源码，练习如何上球队开始比赛。

4、以上作业做完的同学，请尽快把你的执行过程写成一个或几个 Shell 脚本，用 tar 打包发给我们（baj@mail.ustc.edu.cn）。

第二部分 UVA 球队开发实验

有用的帮助：

建议参考：合肥工业大学研究小组资料 —— 第四章 UvA Trilearn 头文件与类。

几个变量：

WM 是指向 WorldModel 类的指针，用来调用 WorldModel 类成员函数。
ACT 是指向 ActHandler 类的指针，用来调用 ActHandler 类的成员函数，主要是用来发命令。
SS 使用来指向 ServerSettings 类的指针，用来获得 Server 的参数。
PS 是用来指向 PlayerSettings 类的指针，用来获得球员自身的参数。
formations 是指向 Formations 类的指针，主要用来获得与阵形相关的东西。
VecPosition 是一个类，主要是一个二维向量的抽象，可以用来表示速度，位置等。

一些函数：

```
SoccerCommand BasicPlayer::intercept ( bool isGoalie );
//发出截球命令
bool ActHandler::putCommandInQueue( SoccerCommand command );
//把命令放入发送队列准备发送
ObjectT WorldModel::getFastestInSetTo( ObjectSetT set,
                                       ObjectT obj,
                                       int *iCyclesToIntercept );
// set 这一类物体 到达 obj 这一类物体最快的是谁。
SoccerCommand BasicPlayer::moveToPos( VecPosition posTo,
                                       AngDeg angWhenToTurn,
                                       double dDistBack,
                                       bool bMoveBack,
                                       int iCycles );
//直接移动到某一点。
SoccerCommand BasicPlayer::turnNeckToObject( ObjectT o,
                                              SoccerCommand soc );
//在执行动作 soc 时也执行转脖子的动作。
SoccerCommand BasicPlayer::turnBodyToObject( ObjectT o );
//把身体转向某一物体。
```

```
SoccerCommand BasicPlayer::kickTo( VecPosition posTarget,  
                                   double dEndSpeed );  
//把球提到 posTarget 位置, 并且到 posTarget 时的速度为 dEndSpeed。
```

书面作业

分析 UVA 球队 src 目录下 PlayerTeams.cpp 文件中 SoccerCommand Player::deMeer5()函数的决策流程。可以画出决策流程图, 也可以用文字描述。

建议:

以上帮助是我们认为比较重要的函数和一些变量。在写流程时不要求对每一个函数和变量了解特别清楚。只要写清楚其大致功能即可, 最重要的是写清楚各个决策分支的功能, 即每个 if 和 else 分支下主要的功能及这些功能的逻辑嵌套关系。不要对每一行程序过分的细究。在阅读代码时尽可能根据变量名和函数名猜想其所代表的含义, 只要能在逻辑上连贯起来就行。如果猜不出来再去查找相应的函数, 先看函数的注释, 不推荐直接看代码。在读代码时要结合程序本身的注释去理解。

上面有的书面作业, 建议在培训前完成, 如果没时间可以在培训后完成。如果完成了, 随时可以发到我的邮箱里。

下面将有几个小实验, 毕竟 UVA 是一个比较大的软件工程同时, 而且离培训还有很长一段长时间, 要想能够开发出比较好的球队, 阅读代码是必不可少的。因此建议在培训前要阅读代码。如何阅读我之前已经给出一些建议, 大家可以作为一个参考。以下每个实验都给出一个实现范例, 大家可以仿照开发。

实验一、技能训练-----带球

此次实验主要是进行单个球员个人技能训练-----带球动作训练。

(1) 选一个球员, 让它拿到球之后一直带球, 直到对方球门里。(用三种不同的方式, DRIBBLE FAST 快速带球, DRIBBLE SLOW, 较慢的带球, DRIBBLE WITH BALL, 近身带球)。

(2) 折线带球, 以 Z 字型的路线往前带球, (提示比如, 向左 45 度和向右 45 度交替的执行, 角度可以任意选, 最好能带进球门里, 可以有很多种实现方法)。

(3) 带球转身, 即在带球时加上转身动作(让球绕着身体转), 转身的角度可以任意选。

(4) 把上面三种动作组合起来, 形成漂亮的带球路径。方式自己选定, 尽可能的使动作连续。动作要多样化。(选做)

有用的函数:

```
SoccerCommand BasicPlayer::dribble( AngDeg ang, DribbleT dribbleT );  
SoccerCommand BasicPlayer::intercept( bool isGoalie );
```

建议：

以上的动作最好用函数封装起来，以便后面的实验可以重用。

实验二、个人技术-----射门

(1) 定点射门：

把球放到任意一个可以射门位置，球员可以自己拿到球之后射门。

(2) 带球射门：

让球员带球，可选用实验的任意一种方式，在到达一定区域后射门。比如进了禁区，或离球门有一定的距离，条件可以自己选定。

(3) 点球大战：

上一个守门员和一个球员进行点球。这是实验主要看进球数，没有上限。（建议：这个实验要做好，取决于射门算得是否准确。在做射门时要尽可能多的考虑场上的情况，选择的射门点尽可能多样化。）

有用函数：

```
VecPosition BasicPlayer::getShootPositionOnLine( VecPosition p1,
                                                  VecPosition p2,
                                                  AngDeg *angLargest );

VecPosition BasicPlayer::getThroughPassShootingPoint( ObjectT objTeam,
                                                       VecPosition posEnd,
                                                       AngDeg *angMax );

SoccerCommand BasicPlayer::kickTo( VecPosition posTarget,
                                    double dEndSpeed );
```

这些函数的功能可以自己在源码中查到，然后看它注释。也可以在合工大的那份资料中查找。

开发要点：

主要考虑射门所要的角度以及以及射门时对手的干扰，比如守门员的位置，以及对手后卫的位置。

实验三、团队合作-----传球

(1) 两个球员之间传球。只要能够实现球从一个球员传到另外一个球员，就行。条件自己定。

(2) 实现两个球员运动中传球。即两个球员之间通过传球配合传球，向对方推进。加上

射门的动作，以及带球动作。

- (3) 两个球员之间通过配合过人，即实现二过一的战术。（比如，两个前锋过一个后卫）。
- (4) 两个球员之间配合射门。要实现这个战术，要把前面所有动作的有机的结合起来才能实现比较好。
- (5) 实现三过二的配合技术。这个实验难度相对比较大，需要仔细分析实际的情况，涉及球员之间配合，队员之间的协调。难度比二过一要大得多。最好在二过一做好了之后，再考虑这一个实验。

有用函数：

```
SoccerCommand BasicPlayer::directPass( VecPosition pos, PassT passType );
SoccerCommand BasicPlayer::leadingPass(ObjectT o, double dDist,
                                         DirectionT dir = DIR_NORTH);
SoccerCommand BasicPlayer::throughPass ( ObjectT o,
                                         VecPosition posEnd,
                                         AngDeg *angMax = NULL);
VecPosition WorldModel::getGlobalPosition( ObjectT o );
ObjectT WorldModel::getClosestInSetTo( ObjectSetT objectSet,
                                       ObjectT o ,
                                       double *dDist = NULL,
                                       double dConfThr = -1.0);
```

建议：

这个实验综合性较大，队友的信息是否准确非常重要，最好加入主动信息采集部分，考虑如何有效的获得信息。即在视觉部分可以考虑如何转脖子，如何选取视觉角，以便能经常看到需要配合的队友。

注意：

以上实验做完后把已修改好的部分代码发给我们，希望尽可能早的做完，做完后可以随时提交至 wangwk6@mail.ustc.edu.cn