

# Introduction to RoboCup Simulation 2D League

Ke Shi

Multi-Agent Systems Lab.  
Department of Computer Science and Technology  
University of Science and Technology of China



Sep. 29, 2008

- 1 RoboCup
- 2 RoboCup Simulation 2D League
- 3 SoccerServer For Simulation 2D League
- 4 Linux System

# 简介

- RoboCup是Robot World Cup的缩写，是一个国际性的研究和教育组织。它通过提供一个标准问题来促进人工智能和智能机器人的研究。这个领域可以集成并检验很大范围内的技术，同时也可被用作综合的面向工程应用的教育。

# 简介

- RoboCup是Robot World Cup的缩写，是一个国际性的研究和教育组织。它通过提供一个标准问题来促进人工智能和智能机器人的研究。这个领域可以集成并检验很大范围内的技术，同时也可被用作综合的面向工程应用的教育。
- 为了这个目的，RoboCup 联盟选择了足球比赛作为一个基本领域，并组织了国际上级别最高、规模最大、影响最广泛的机器人足球赛事和学术会议：机器人足球世界杯及学术会议，即RoboCup。

# 简介

- RoboCup是Robot World Cup的缩写，是一个国际性的研究和教育组织。它通过提供一个标准问题来促进人工智能和智能机器人的研究。这个领域可以集成并检验很大范围内的技术，同时也可被用作综合的面向工程应用的教育。
- 为了这个目的，RoboCup 联盟选择了足球比赛作为一个基本领域，并组织了国际上级别最高、规模最大、影响最广泛的机器人足球赛事和学术会议：机器人足球世界杯及学术会议，即RoboCup。
- 为了能让一个机器人球队真正能够进行足球比赛，必须集成各种各样的技术，包括自治智能体的设计准则、多智能体合作、策略获取、实时推理、机器人学以及感知信息融合等。

## RoboCup的3个领域

- RoboCup是足球赛：由硬件或仿真机器人进行的足球赛，比赛规则与人类正规的足球赛类似。硬件机器人足球队的研发涉及计算机、自动控制、传感与感知融合、无线通讯、精密机械和仿生材料等众多学科的前沿研究与综合集成。仿真机器人足球赛在标准软件平台上进行，平台设计充分体现了控制、通讯、传感和人体机能等方面的实际限制，使仿真球队程序易于转化为硬件球队的控制软件。  
目前包括的比赛有：仿真组，小型组，中型组，标准平台组和类人组等。

## RoboCup的3个领域

- RoboCup救援比赛：是RoboCup 一个新的研究领域，目标是在如日本神户大地震这种大灾难中高效、准确地找到受难者，并实施救援行动。选择这个领域一方面是因为这是一个意义重大的社会问题，另一方面它在很多方面都和足球比赛有相似的特点，如动态环境，不完全、有噪声的信息。同时它还有些足球比赛没有的特点，如后勤学、紧急团队间的合作等。

## RoboCup的3个领域

- RoboCup救援比赛：是RoboCup 一个新的研究领域，目标是在如日本神户大地震这种大灾难中高效、准确地找到受难者，并实施救援行动。选择这个领域一方面是因为这是一个意义重大的社会问题，另一方面它在很多方面都和足球比赛有相似的特点，如动态环境，不完全、有噪声的信息。同时它还有些足球比赛没有的特点，如后勤学、紧急团队间的合作等。
- RoboCup初级比赛：是RoboCup 教育、推广的重要组成部分，是面向中小学生的的一种新型教育方式，使得他们不需要具有很高深的知识就可以轻松进入机器人学领域，通过对电子装置，硬件和软件的直接动手实践来提高自身能力，同时更有助于他们学习团队合作。



# RoboCup与FIRA

- 技术标准不同：FIRA允许采用集中控制方式；而RoboCup要求每个队员必须是自主的，因而球队中的每个队员是一个独立的“主体”，球队是一个“多主体”系统。

# RoboCup与FIRA

- 技术标准不同：FIRA允许采用集中控制方式；而RoboCup要求每个队员必须是自主的，因而球队中的每个队员是一个独立的“主体”，球队是一个“多主体”系统。
- 研究重点不同：FIRA 重点研究“灵巧机器人”，特别是动态环境中的运动控制；而RoboCup着重研究动态环境中的多智能体合作、实时规划和Agent体系结构设计。也可以认为，FIRA主要面向机器人学，而RoboCup主要面向人工智能。

# RoboCup与FIRA

- 技术标准不同：FIRA允许采用集中控制方式；而RoboCup要求每个队员必须是自主的，因而球队中的每个队员是一个独立的“主体”，球队是一个“多主体”系统。
- 研究重点不同：FIRA 重点研究“灵巧机器人”，特别是动态环境中的运动控制；而RoboCup着重研究动态环境中的多智能体合作、实时规划和Agent体系结构设计。也可以认为，FIRA主要面向机器人学，而RoboCup主要面向人工智能。
- 主要参加者不同：FIRA的主要成员来自东亚和南美等第三世界国家；而RoboCup的主要成员为日美和欧洲各发达国家。

# RoboCup与FIRA

- 技术标准不同：FIRA允许采用集中控制方式；而RoboCup要求每个队员必须是自主的，因而球队中的每个队员是一个独立的“主体”，球队是一个“多主体”系统。
- 研究重点不同：FIRA 重点研究“灵巧机器人”，特别是动态环境中的运动控制；而RoboCup着重研究动态环境中的多智能体合作、实时规划和Agent体系结构设计。也可以认为，FIRA主要面向机器人学，而RoboCup主要面向人工智能。
- 主要参加者不同：FIRA的主要成员来自东亚和南美等第三世界国家；而RoboCup的主要成员为日美和欧洲各发达国家。
- 规模不同：相比每年的RoboCup世界杯，FIRA世界杯赛的规模要小得多。

# RoboCup的历史

- RoboCup 2008, Suzhou, China
- RoboCup 2007, Atlanta, USA
- RoboCup 2006, Bremen, Germany
- RoboCup 2005, Osaka, Japan
- RoboCup 2004, Lisbon, Portugal
- RoboCup 2003, Padua, Italy
- RoboCup 2002, Fukuoka, Japan
- RoboCup 2001, Seattle, USA
- RoboCup 2000, Melbourne, Australia
- RoboCup 1999, Stockholm, Sweden
- RoboCup 1998, Paris, France
- RoboCup 1997, Nagoya, Japan

# RoboCup在中国

- RoboCup在我国起步稍晚，1999年举办首届全国赛，当时只有清华大学和中国科学技术大学参赛，如今已经发展到500余支队伍。其中包括中科院自动化所，清华大学，北京大学，浙江大学，上海交通大学，国防科技大学，南开大学和西安交通大学等。

# RoboCup在中国

- RoboCup在我国起步稍晚，1999年举办首届全国赛，当时只有清华大学和中国科学技术大学参赛，如今已经发展到500余支队伍。其中包括中科院自动化所，清华大学，北京大学，浙江大学，上海交通大学，国防科技大学，南开大学和西安交通大学等。
- 在合肥，中国科学技术大学，合肥工业大学，安徽大学和安徽建筑工页学院均组建了RoboCup机器人足球队，参加每年的RoboCup比赛。

# RoboCup在中国

- RoboCup在我国起步稍晚，1999年举办首届全国赛，当时只有清华大学和中国科学技术大学参赛，如今已经发展到500余支队伍。其中包括中科院自动化所，清华大学，北京大学，浙江大学，上海交通大学，国防科技大学，南开大学和西安交通大学等。
- 在合肥，中国科学技术大学，合肥工业大学，安徽大学和安徽建筑工页学院均组建了RoboCup机器人足球队，参加每年的RoboCup比赛。
- 2008中国机器人大赛暨RoboCup China Open 2008将于2008年12月5日至12月7日在广东中山举行，将有来自美国，日本，伊朗，新加坡和巴西等国的国外队伍参赛。



# RoboCup在中国的历史

- 2007, 山东济南
- 2006, 江苏苏州
- 2005, 江苏常州
- 2004, 广东广州
- 2003, 北京
- 2002, 上海
- 2001, 云南昆明
- 2000, 安徽合肥
- 1999, 重庆

# RoboCup的最终目标

By mid-21st century, a team of fully autonomous humanoid robot soccer players shall win the soccer game, comply with the official rule of the FIFA, against the winner of the most recent World Cup!

<http://www.robocup.org>

# 网上资源

- 中国自动化学会机器人竞赛工作委员会：  
<http://www.rcccaa.org/>

## 网上资源

- 中国自动化学会机器人竞赛工作委员会：<http://www.rcccaa.org/>
- RoboCup仿真2D资料页：<http://wrighteagle.org/2D/>

## 网上资源

- 中国自动化学会机器人竞赛工作委员会：<http://www.rcccaa.org/>
- RoboCup仿真2D资料页：<http://wrighteagle.org/2D/>
- RoboCup 2008机器人世界杯：<http://robocup-cn.org/>

## 网上资源

- 中国自动化学会机器人竞赛工作委员会：<http://www.rcccaa.org/>
- RoboCup仿真2D资料页：<http://wrighteagle.org/2D/>
- RoboCup 2008机器人世界杯：<http://robocup-cn.org/>
- 2008中国机器人大赛：<http://www.robochina.com.cn/>

- 1 RoboCup
- 2 RoboCup Simulation 2D League
- 3 SoccerServer For Simulation 2D League
- 4 Linux System

# 简介

- 仿真机器人足球赛利用计算机模拟机器人进行足球比赛，由RoboCup 仿真平台开发小组提供一个标准比赛软件平台，平台设计充分体现了控制、通讯、传感和人体机能等方面的实际限制，使仿真球队程序易于转化为硬件球队的控制软件。



# 简介

- 仿真机器人足球赛利用计算机模拟机器人进行足球比赛，由RoboCup 仿真平台开发小组提供一个标准比赛软件平台，平台设计充分体现了控制、通讯、传感和人体机能等方面的实际限制，使仿真球队程序易于转化为硬件球队的控制软件。
- 由于避免了现实物理环境和当前机器人制造技术的限制，仿真机器人足球主要把研究重点放在球队的高级功能的研究上，包括动态不确定环境中的多智能体合作、实时推理-规划-决策、机器学习和策略获取等当前人工智能的热点问题。

# 简介

- 仿真机器人足球赛利用计算机模拟机器人进行足球比赛，由RoboCup 仿真平台开发小组提供一个标准比赛软件平台，平台设计充分体现了控制、通讯、传感和人体机能等方面的实际限制，使仿真球队程序易于转化为硬件球队的控制软件。
- 由于避免了现实物理环境和当前机器人制造技术的限制，仿真机器人足球主要把研究重点放在球队的高级功能的研究上，包括动态不确定环境中的多智能体合作、实时推理-规划-决策、机器学习和策略获取等当前人工智能的热点问题。
- 正是由于摆脱了硬件限制，仿真组比赛比较容易实现，成为RoboCup比赛中历史最老、参赛队最多的一个项目，研究步伐也快于其他项目。

# 人工智能新的标准问题

- 计算机国际象棋是一个典型的标准问题，各种搜索算法可以在这个领域中评价和发展。计算机国际象棋作为一个标准问题的成功，主要原因之一是清楚的定义了对研究进展的评价，即可以用系统的棋力来评价。同样，RoboCup 也是可以通过实际的比赛对研究进展作出客观清楚的评价。

# 人工智能新的标准问题

- 计算机国际象棋是一个典型的标准问题，各种搜索算法可以在这个领域中评价和发展。计算机国际象棋作为一个标准问题的成功，主要原因之一是清楚的定义了对研究进展的评价，即可以用系统的棋力来评价。同样，RoboCup 也是可以通过实际的比赛对研究进展作出客观清楚的评价。
- 机器人足球赛与国际象棋人机对抗赛所体现出的研究背景区别主要有以下几个方面：(1)静态环境和动态环境；(2)行动效果和环境的可预测与不可预测；(3)个体与团体。以上三方面的差别使得每个足球机器人必须是一个能自主行动且与队友合作的智能体，而一支球队则是一个内部协调的多智能体系统。相反，“深蓝”本质上只是一个能在几分钟内分析600 亿个棋局的搜索程序。

# 学习的挑战

学习的挑战，目的是寻求全面综合的学习方案，它能用于需要适应环境的多智能体系统的学习，并能用标准任务来评价所提供方法的优缺点。

- 单个智能体的离线技术学习，如拦截球、踢球

# 学习的挑战

学习的挑战，目的是寻求全面综合的学习方案，它能用于需要适应环境的多智能体系统的学习，并能用标准任务来评价所提供方法的优缺点。

- 单个智能体的离线技术学习，如拦截球、踢球
- 智能体团队的离线合作学习，如传接球

# 学习的挑战

学习的挑战，目的是寻求全面综合的学习方案，它能用于需要适应环境的多智能体系统的学习，并能用标准任务来评价所提供方法的优缺点。

- 单个智能体的离线技术学习，如拦截球、踢球
- 智能体团队的离线合作学习，如传接球
- 在线技术和合作学习，如适应性跑位

# 学习的挑战

学习的挑战，目的是寻求全面综合的学习方案，它能用于需要适应环境的多智能体系统的学习，并能用标准任务来评价所提供方法的优缺点。

- 单个智能体的离线技术学习，如拦截球、踢球
- 智能体团队的离线合作学习，如传接球
- 在线技术和合作学习，如适应性跑位
- 在线对抗学习，如对预测的对手行为采取相应对策



## 团队合作的挑战

团队合作地挑战中，其基本体系结构问题就是要建立这样一种体系结构，它可以支持团队行为的规划以及更重要的团队规划的执行。这种规划和规划执行可以通过一个两层的体系结构来完成，但是整个系统必须能够实时运作。

- 对多智能体对抗比赛中意外事件的规划

## 团队合作的挑战

团队合作地挑战中，其基本体系结构问题就是要建立这样一种体系结构，它可以支持团队行为的规划以及更重要的团队规划的执行。这种规划和规划执行可以通过一个两层的体系结构来完成，但是整个系统必须能够实时运作。

- 对多智能体对抗比赛中意外事件的规划
- 规划的分解和合并

## 团队合作的挑战

团队合作地挑战中，其基本体系结构问题就是要建立这样一种体系结构，它可以支持团队行为的规划以及更重要的团队规划的执行。这种规划和规划执行可以通过一个两层的体系结构来完成，但是整个系统必须能够实时运作。

- 对多智能体对抗比赛中意外事件的规划
- 规划的分解和合并
- 团队规划的执行

# 对手建模的挑战

智能体建模是为其它智能体的目标、规划、知识、能力或情感建模和推理，是多智能体交互的关键问题。RoboCup对手建模挑战研究在动态多智能体领域内对一组对手建模。

- 在线追踪

# 对手建模的挑战

智能体建模是为其它智能体的目标、规划、知识、能力或情感建模和推理，是多智能体交互的关键问题。RoboCup对手建模挑战研究在动态多智能体领域内对一组对手建模。

- 在线追踪
- 在线策略识别

# 对手建模的挑战

智能体建模是为其它智能体的目标、规划、知识、能力或情感建模和推理，是多智能体交互的关键问题。RoboCup对手建模挑战研究在动态多智能体领域内对一组对手建模。

- 在线追踪
- 在线策略识别
- 离线审阅

- 1 RoboCup
- 2 RoboCup Simulation 2D League
- 3 SoccerServer For Simulation 2D League
- 4 Linux System

# 简介

- RoboCup仿真2D比赛是在一个标准的计算机环境内进行的，比赛规则基本上与国际足球联合会的比赛规则一致。比赛的方式是由RoboCup委员会提供标准的机器人足球仿真软件平台，称作SoccerServer。



# 简介

- RoboCup仿真2D比赛是在一个标准的计算机环境内进行的，比赛规则基本上与国际足球联合会的比赛规则一致。比赛的方式是由RoboCup委员会提供标准的机器人足球仿真软件平台，称作SoccerServer。
- RoboCup仿真2D平台支持多个虚拟球员在一个动态、不确定的多主体环境中实时地进行合作对抗。

# 简介

- RoboCup仿真2D比赛是在一个标准的计算机环境内进行的，比赛规则基本上与国际足球联合会的比赛规则一致。比赛的方式是由RoboCup委员会提供标准的机器人足球仿真软件平台，称作SoccerServer。
- RoboCup仿真2D平台支持多个虚拟球员在一个动态、不确定的多主体环境中实时地进行合作对抗。
- SoccerServer的第一个版本诞生于1995年，今年世界杯的比赛版本已发展到12.1。版本改动的原则就是使仿真平台更加接近于真实足球，与RoboCup的总目标一致。

## 结构特点

- RoboCup仿真2D比赛采用的是CS模式结构：SoccerServer作为服务器端，提供了一个虚拟场地并且模拟包括球和球员在内的所有物体移动；每个客户端程序相当于一个球员的大脑，控制场上该球员的移动。

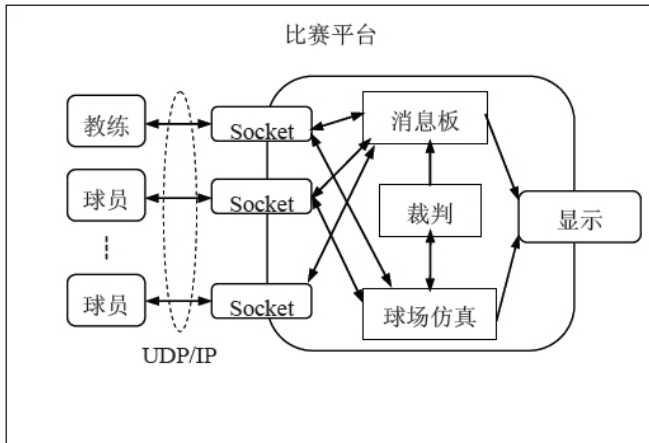
## 结构特点

- RoboCup仿真2D比赛采用的是CS模式结构：SoccerServer作为服务器端，提供了一个虚拟场地并且模拟包括球和球员在内的所有物体移动；每个客户端程序相当于一个球员的大脑，控制场上该球员的移动。
- 服务器端和客户端之间都是通过UDP/IP协议进行信息交互的：客户端程序可以发送指令去控制相应的场上球员，而服务器端按照规则给每个客户端发送它所能获得的信息。

## 结构特点

- RoboCup仿真2D比赛采用的是CS模式结构：SoccerServer作为服务器端，提供了一个虚拟场地并且模拟包括球和球员在内的所有物体移动；每个客户端程序相当于一个球员的大脑，控制场上该球员的移动。
- 服务器端和客户端之间都是通过UDP/IP协议进行信息交互的：客户端程序可以发送指令去控制相应的场上球员，而服务器端按照规则给每个客户端发送它所能获得的信息。
- 当一场比赛开始时，双方12个独立的球员程序连接到比赛平台上场比赛。每个队的目标就是将球踢进对方球门同时阻止球进入自己的球门。

## SoccerServer的结构图



## 各模块的介绍

- 球场仿真模块计算球场上对象的运动，检测他们之间的碰撞。球场上的对象包括每队各11名队员、球、球门、标记及标志线等。

## 各模块的介绍

- 球场仿真模块计算球场上对象的运动，检测他们之间的碰撞。球场上的对象包括每队各11名队员、球、球门、标记及标志线等。
- 裁判模块根据规则来控制比赛的进程。由于仿真比赛环境具有动态、实时、不确定、多主体对抗等特点，比赛不可能按照事先的设计按部就班地进行，还需要在比赛中有一个“智能”裁判。



## 各模块的介绍

- 球场仿真模块计算球场上对象的运动，检测他们之间的碰撞。球场上的对象包括每队各11名队员、球、球门、标记及标志线等。
- 裁判模块根据规则来控制比赛的进程。由于仿真比赛环境具有动态、实时、不确定、多主体对抗等特点，比赛不可能按照事先的设计按部就班地进行，还需要在比赛中有一个“智能”裁判。
- 消息板模块负责客户端之间的通讯。每SoccerServer采用离散化模式运行，即所有程序运行都是以仿真周期为单位。在每个仿真周期结束前，SoccerServer收集所有球员程序的行为请求，直到每个周期末才统一执行并更新场上信息。另外，在每个周期的开始SoccerServer根据各个球员的状态发送相应的已更新的场上信息，体现了球员感知信息和行动的异步性。

## 球员客户端

- 一个球员客户端就是球员的大脑：从服务器端接收到感知信息，并且发送命令到服务器端。

## 球员客户端

- 一个球员客户端就是球员的大脑：从服务器端接收到感知信息，并且发送命令到服务器端。
- 每个球员程序都是独立的进程，通过一个特定的端口和比赛平台连接。当一个球员程序和比赛平台建立好连接后，所有通讯信息都通过这个端口传输。一个球队最多可以连接12名队员包，括11名球员和一个场上教练。

## 球员客户端

- 一个球员客户端就是球员的大脑：从服务器端接收到感知信息，并且发送命令到服务器端。
- 每个球员程序都是独立的进程，通过一个特定的端口和比赛平台连接。当一个球员程序和比赛平台建立好连接后，所有通讯信息都通过这个端口传输。一个球队最多可以连接12名队员包，括11名球员和一个场上教练。
- 正式比赛中每个客户端程序只能控制一个场上队员。因此，要组成一支球队就需要同样数量的程序分别控制每个队员。球员之间的通讯必须经过比赛平台按照say和hear命令协议执行，而且通讯环境具有单信道、窄带宽等特点。

## SoccerServer对客户端的限制

- 每个队员都有一定的视野范围，每次只能获得局部信息，即包括可视范围内的对象信息，而且是有随机噪音的。

## SoccerServer对客户端的限制

- 每个队员都有一定的视野范围，每次只能获得局部信息，即包括可视范围内的对象信息，而且是有随机噪音的。
- 每个球员都有自己的体力值，随跑动衰减，每周期可以自动恢复一些，这样就限制球员要注意调整跑动速度，合理分配体力，也更符合现实。

## SoccerServer对客户端的限制

- 每个队员都有一定的视野范围，每次只能获得局部信息，即包括可视范围内的对象信息，而且是有随机噪音的。
- 每个球员都有自己的体力值，随跑动衰减，每周期可以自动恢复一些，这样就限制球员要注意调整跑动速度，合理分配体力，也更符合现实。
- SoccerServer还引入了风及噪声的干扰及对行为参数的干扰，使比赛更趋于真实，正如现实比赛很难无风和噪声的干扰，快速跑动中的队员不太可能急转弯等等，但这同时也增大了准确建模的难度。

# SoccerServer的特点总结

- 分布式多智能体团队合作和对抗



# SoccerServer的特点总结

- 分布式多智能体团队合作和对抗
- 动态、实时、不确定环境

## SoccerServer的特点总结

- 分布式多智能体团队合作和对抗
- 动态、实时、不确定环境
- 感知和行为异步

# SoccerServer的特点总结

- 分布式多智能体团队合作和对抗
- 动态、实时、不确定环境
- 感知和行为异步
- 球员能力受限

# SoccerServer的特点总结

- 分布式多智能体团队合作和对抗
- 动态、实时、不确定环境
- 感知和行为异步
- 球员能力受限
- 视觉受限

# SoccerServer的特点总结

- 分布式多智能体团队合作和对抗
- 动态、实时、不确定环境
- 感知和行为异步
- 球员能力受限
- 视觉受限
- 通讯受限

# SoccerServer的特点总结

- 分布式多智能体团队合作和对抗
- 动态、实时、不确定环境
- 感知和行为异步
- 球员能力受限
- 视觉受限
- 通讯受限
- 多噪声源

# SoccerServer的特点总结

- 分布式多智能体团队合作和对抗
- 动态、实时、不确定环境
- 感知和行为异步
- 球员能力受限
- 视觉受限
- 通讯受限
- 多噪声源
- 连接不可靠

# 自动裁判

- 中场开球



# 自动裁判

- 中场开球
- 进球

# 自动裁判

- 中场开球
- 进球
- 守门员发球

# 自动裁判

- 中场开球
- 进球
- 守门员发球
- 出界

# 自动裁判

- 中场开球
- 进球
- 守门员发球
- 出界
- 越位

# 自动裁判

- 中场开球
- 进球
- 守门员发球
- 出界
- 越位
- 回传球

# 自动裁判

- 中场开球
- 进球
- 守门员发球
- 出界
- 越位
- 回传球
- 发球违例

# 自动裁判

- 中场开球
- 进球
- 守门员发球
- 出界
- 越位
- 回传球
- 发球违例
- 加时赛和点球大战

# 人为干预

- 故意包围球



# 人为干预

- 故意包围球
- 故意用过多的球员堵球门

# 人为干预

- 故意包围球
- 故意用过多的球员堵球门
- 故意阻挡其他球员的移动

# 人为干预

- 故意包围球
- 故意用过多的球员堵球门
- 故意阻挡其他球员的移动
- 用大量消息堵塞比赛平台的通讯信道

# 人为干预

- 故意包围球
- 故意用过多的球员堵球门
- 故意阻挡其他球员的移动
- 用大量消息堵塞比赛平台的通讯信道
- 守门员滥发catch 指令

# 人为干预

- 故意包围球
- 故意用过多的球员堵球门
- 故意阻挡其他球员的移动
- 用大量消息堵塞比赛平台的通讯信道
- 守门员滥发catch 指令
- 其他不适当的行为

## 仿真2D源码包

- rcsssbase: 供其他各种仿真机器人足球程序包使用的基本代码。

## 仿真2D源码包

- rcsssbase: 供其他各种仿真机器人足球程序包使用的基本代码。
- rcsssserver: 是该源码包的主要部分，执行实际的仿真工作。

## 仿真2D源码包

- rcsssbase: 供其他各种仿真机器人足球程序包使用的基本代码。
- rcsssserver: 是该源码包的主要部分, 执行实际的仿真工作。
- rcssmonitor: 通过连接rcsssserver或rcsslogplayer来显示现场比赛或比赛录像。



## 仿真2D源码包

- rcsssbase: 供其他各种仿真机器人足球程序包使用的基本代码。
- rcsssserver: 是该源码包的主要部分, 执行实际的仿真工作。
- rcssmonitor: 通过连接rcsssserver或rcsslogplayer来显示现场比赛或比赛录像。
- rcsslogplayer: 可以重放rcsssserver录制的比赛录像。

# 安装仿真2D平台

- `tar xzvf rcss*.tar.gz`

## 安裝仿真2D平台

- `tar xzvf rcss*.tar.gz`
- `./configure`

## 安裝仿真2D平台

- `tar xzvf rcss*.tar.gz`
- `./configure`
- `make`

## 安裝仿真2D平台

- `tar xzvf rcss*.tar.gz`
- `./configure`
- `make`
- `make install`

## 安裝仿真2D平台

- `tar xzvf rcss*.tar.gz`
- `./configure`
- `make`
- `make install`
- `ldconfig`

# 使用仿真2D平台

- rcsoccersim

# 使用仿真2D平台

- rcsoccersim
- rcssserver



# 使用仿真2D平台

- rcsoccersim
- rcssserver
- rcssmonitor

# 使用仿真2D平台

- rcsoccersim
- rcssserver
- rcssmonitor
- rcsslogplayer

# 使用仿真2D平台

- rcsoccersim
- rcssserver
- rcssmonitor
- rcsslogplayer
- Ctrl + C

- 1 RoboCup
- 2 RoboCup Simulation 2D League
- 3 SoccerServer For Simulation 2D League
- 4 Linux System

# 特点

Linux系统由内核和其应用程序组成，不同的厂商可以发布内核与其它应用程序包的搭配，这样的组合称为“Linux发行版”。不同的发行版没有本质区别，差别仅在于内核的版本，软件包的数量和种类以及提供的服务。

- 真正意义上的多任务系统。

# 特点

Linux系统由内核和其应用程序组成，不同的厂商可以发布内核与其它应用程序包的搭配，这样的组合称为“Linux发行版”。不同的发行版没有本质区别，差别仅在于内核的版本，软件包的数量和种类以及提供的服务。

- 真正意义上的多任务系统。
- 支持数十种文件系统格式。

# 特点

Linux系统由内核和其应用程序组成，不同的厂商可以发布内核与其它应用程序包的搭配，这样的组合称为“Linux发行版”。不同的发行版没有本质区别，差别仅在于内核的版本，软件包的数量和种类以及提供的服务。

- 真正意义上的多任务系统。
- 支持数十种文件系统格式。
- 开放源码，用户可以对自已的系统进行修改。

# 特点

Linux系统由内核和其应用程序组成，不同的厂商可以发布内核与其它应用程序包的搭配，这样的组合称为“Linux发行版”。不同的发行版没有本质区别，差别仅在于内核的版本，软件包的数量和种类以及提供的服务。

- 真正意义上的多任务系统。
- 支持数十种文件系统格式。
- 开放源码，用户可以对自已的系统进行修改。
- 优良的安全特性。



## 特点

Linux系统由内核和其应用程序组成，不同的厂商可以发布内核与其它应用程序包的搭配，这样的组合称为“Linux发行版”。不同的发行版没有本质区别，差别仅在于内核的版本，软件包的数量和种类以及提供的服务。

- 真正意义上的多任务系统。
- 支持数十种文件系统格式。
- 开放源码，用户可以对自已的系统进行修改。
- 优良的安全特性。
- 。 。 。

# 特殊的目录

- /bin - 包含常用的用户命令

## 特殊的目录

- /bin - 包含常用的用户命令
- /boot - 存放Linux内核和引导程序

## 特殊的目录

- /bin - 包含常用的用户命令
- /boot - 存放Linux内核和引导程序
- /dev - 包含一些设备文件

## 特殊的目录

- /bin - 包含常用的用户命令
- /boot - 存放Linux内核和引导程序
- /dev - 包含一些设备文件
- /etc - 存放一些管理配置文件

## 特殊的目录

- /bin - 包含常用的用户命令
- /boot - 存放Linux内核和引导程序
- /dev - 包含一些设备文件
- /etc - 存放一些管理配置文件
- /home - 存放普通用户的目录

## 特殊的目录

- /bin - 包含常用的用户命令
- /boot - 存放Linux内核和引导程序
- /dev - 包含一些设备文件
- /etc - 存放一些管理配置文件
- /home - 存放普通用户的目录
- /root - 超级用户的目录

## 特殊的目录

- /bin - 包含常用的用户命令
- /boot - 存放Linux内核和引导程序
- /dev - 包含一些设备文件
- /etc - 存放一些管理配置文件
- /home - 存放普通用户的目录
- /root - 超级用户的目录
- /proc - 包含系统运行时的内核信息



## 特殊的目录

- /bin - 包含常用的用户命令
- /boot - 存放Linux内核和引导程序
- /dev - 包含一些设备文件
- /etc - 存放一些管理配置文件
- /home - 存放普通用户的目录
- /root - 超级用户的目录
- /proc - 包含系统运行时的内核信息
- /usr - 包含“用户级”的程序和文档

## 特殊的目录

- /bin - 包含常用的用户命令
- /boot - 存放Linux内核和引导程序
- /dev - 包含一些设备文件
- /etc - 存放一些管理配置文件
- /home - 存放普通用户的目录
- /root - 超级用户的目录
- /proc - 包含系统运行时的内核信息
- /usr - 包含“用户级”的程序和文档
- /var - 经常发生变动的文件

## 文件的权限

- Linux对不同用户访问同一文件的权限做了不同的规定，权限可以分为三种：可读，可写和可执行。

## 文件的权限

- Linux对不同用户访问同一文件的权限做了不同的规定，权限可以分为三种：可读，可写和可执行。
- 在Linux系统中，用户是按组分类的，一个用户属于一个或多个组，文件所有者以外的用户又可以分为文件所有者的同组用户和其它用户。

# 文件的权限

- Linux对不同用户访问同一文件的权限做了不同的规定，权限可以分为三种：可读，可写和可执行。
- 在Linux系统中，用户是按组分类的，一个用户属于一个或多个组，文件所有者以外的用户又可以分为文件所有者的同组用户和其它用户。
- 例如，/bin/l`s` 文件的权限为：-rwxr-xr-x

# Shell

- 类Unix系统与用户交互的命令行界面称之为Shell。

# Shell

- 类Unix系统与用户交互的命令行界面称之为Shell。
- 学习Shell的理由：

# Shell

- 类Unix系统与用户交互的命令行界面称之为Shell。
- 学习Shell的理由：
  - 学习一种登入类Unix系统并进行操作的通用技术。



# Shell

- 类Unix系统与用户交互的命令行界面称之为Shell。
- 学习Shell的理由：
  - 学习一种登入类Unix系统并进行操作的通用技术。
  - 利用Shell的各种高级特性提高工作效率。

# Shell

- 类Unix系统与用户交互的命令行界面称之为Shell。
- 学习Shell的理由：
  - 学习一种登入类Unix系统并进行操作的通用技术。
  - 利用Shell的各种高级特性提高工作效率。
  - 编制Shell脚本，完成各种复杂任务。

# 常用命令

- `uname` - 查看系统信息

# 常用命令

- `uname` - 查看系统信息
- `passwd` - 修改密码

# 常用命令

- `uname` - 查看系统信息
- `passwd` - 修改密码
- `ps` - 查看运行的进程

# 常用命令

- `uname` - 查看系统信息
- `passwd` - 修改密码
- `ps` - 查看运行的进程
- `killall` - 根据进程名杀死进程

## 常用命令

- `uname` - 查看系统信息
- `passwd` - 修改密码
- `ps` - 查看运行的进程
- `killall` - 根据进程名杀死进程
- `find` - 查找文件

## 常用命令

- `uname` - 查看系统信息
- `passwd` - 修改密码
- `ps` - 查看运行的进程
- `killall` - 根据进程名杀死进程
- `find` - 查找文件
- `grep` - 查找字符串



## 常用命令

- `uname` - 查看系统信息
- `passwd` - 修改密码
- `ps` - 查看运行的进程
- `killall` - 根据进程名杀死进程
- `find` - 查找文件
- `grep` - 查找字符串
- `tar` - 打包或者解包

Thank you for your attention!  
Q & A