《ROS机器人开发技术》

课程讲稿

|  |  |  |
| --- | --- | --- |
| 课程名称 | ： | ROS机器人开发技术 |
| 教师姓名 | ： | XXX |
| 提交时间 | ： | 2018年7月X日 |

中国大学MOOC制

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程  名称 | 《ROS机器人开发技术》 | | 章 | | 第一章 | | | 课程类型 |
| 节 | | 第一节 | | | 授课(√ )  实训( ) |
| 名称 | ROS的介绍与安装1 | | | | | | |
| 教师 |  | 课时 | | | | 1课时 | | |
| 参考  资料 | 1. 中国大学mooc《ROS入门课程》讲义， 2. 机器人操作系统简析，[美] Jason M. O'Kane 著，肖军浩 译 | | | | | | | |
| 教学  目的  要求 | 了解ROS的常用工具，组件以及IDE  学会使用vim编辑文本，进行简单操作 | | | | | | | |
| 教学  重点  难点 |  | | | PPT页面 | | | 时间分配 | |
| 教学重点 | | | | | | | |
| 无 | | |  | | |  | |
| 教学难点   |  |  |  | | --- | --- | --- | | 无 |  |  | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 教学方法 | | |
| 本授课以课堂讲授为主，与课堂演示方式相结合 | | |
| 教学内容 | 操作演示 | 知识点 |
| **PPT第1页：**  欢迎大家来到睿思学院，我是你们的老师XXX，我们这门课程的名字叫做《机器人操作系统入门》，也就是ROS的入门，相信大家对ROS也多多少少有一些了解，就算之前没有接触过不了解也没关系，相信大家学完这门课后，会对ROS有一个系统的认知。这节课我们了解一些关于ros的常见工具。 |  |  |
| **PPT第2-3页**  下面我们大致了解一下ROS的常用工具 |  |  |
| PPT第4页  Gazebo是一个机器人仿真工具，模拟器，也是一个独立的开源机器人仿真平台。Gazebo不仅开源，也是是兼容ROS最 好的仿真工具。 |  |  |
| PPT 第5页 RViz(the Robit Visualization tool)机器人可视化工具，可视化的作用是直观的，它极大的方便了监控和调试等操作。  我们可以看到一个机器人模型在3D场景中 | 输入 roslaunch robot\_sim\_demo robot\_spawn.launch 命令：$ rviz 直接打开 |  |
| PPT 第6-8页  rqt是一个基于qt开发的可视化工具，拥有扩展性好、灵活易用、跨平台等特点，主要作用和 RViz一致都是可视化，但是和RViz相比，rqt要高级一个层次 | 输入 roslaunch robot\_sim\_demo robot\_spawn.launch分别对应命令：rqt\_graph rqt\_plot  rqt\_console |  |
| PPT第9页  我们接着了解一些ros的相关组件，以slam跟navigation为例 |  |  |
| PPT第10页  机器人研究的问题包含许许多多的领域，我们常见的几个研究的问题包括：建图(Mapping)、定位(Localization)和路径规划（Path Planning） ，如果机器人带有机械臂，那么运动规划 （Motion Planning） 也是重要的一个环节。而同步定位与建图（SLAM） 问题位于定位和建图的交集部分。（如图所示）  SLAM需要机器人在未知的环境中逐步建立起地图，然后根据地区确定自身位置，从而进一步定位。 |  |  |
| PPT第11页  Navigation是机器人最基本的功能之一，ROS为我们提供了一整套Navigation的解决方案，包括全局与局部的路径规划、代价地图、异常行为恢复、地图服务器等等，这些开源工具包极 大地减少了我们开发的工作量，任何一套移动机器人硬件平台经过这套方案就可以快速部署实现。 |  |  |
| PPT 第12-13页  我们主要来了解下Navigation Stack  看一下它所包含的package |  |  |
| PPT 第14页  接下来介绍Linux下的文本编辑器  Nano跟vim |  |  |
| PPT第15-17页  先介绍nano  Nano简单易用，适合刚接触Linux的新人。  终端输入nano 就可以直接进入nano的主界面  可以看到第一行反白介绍了nano的版本号以及被编辑的文件名称。接下来就是可编辑区域，这里由于newfile文件是一个新建文件，内容为空，所以编辑区域没有任何文本文字。再看最后的两行，这是nano下可以使用的一些快捷键，给出其中几个对应的操作：  ctrl+G： 取得在线帮助  ctrl+O： 保存文件，如果你有权限就可保存修改的文件  ctrl+R： 从其他文件读入数据，可将某个文件的内容贴在本文件中  ctrl+Y： 显示前一页  ctrl+K： 剪切当前行的内容  ctrl+C：‘ 显示光标所在的位置  ctrl+X： 退出nano，如有有修改过文件，则会提示是否要保存修改的内容  ctrl+J： 调整文本格式  ctrl+W： 查找命令，按下后会跳转到末行的反白位置，输入要查找的内容回车即可。  ctrl+V：显示下一页  在这里我们只要掌握如何新建文件，如何编辑并保存退出就行。  下面稍微演示一下： |  |  |
| PPT第18页：  在终端输入nano 加想要新建的文件名  这里我新建了hello.txt文件（不要忘了扩展名）  输入hello.ros!  ctrl+O保存文件  然后 ctrl+X退出。 |  |  |
| PPT第19-20页  接下来学习一个很关键的东西vim  所有的 Unix /linux系统都会内建 vi 文书编辑器，其他的文书编辑器则不一定会存在。但是目前我们使用比较多的是 vim 编辑器。vim 具有程序编辑的能力，可以主动的以字体颜色辨别语法的正确性，方便程序设计。  vi/vim的区别简单点来说，它们都是多模式编辑器，不同的是vim 是vi的升级版本，它不仅兼容vi的所有指令，而且还有一些新的特性在里面，例如语法加亮。  选择vim的最与众不同的两个理由：   1. 支持中文   2、规则表达式很完整 |  |  |
| PPT第21页  先了解一下vim的几种模式：  一般模式，编辑模式，命令行模式  如图所示，三种模式之间的转化关系 |  | Vim的学习了解 |
| PPT 第22-23页  vim终归只是一个编写代码或编辑文档的工具，所以只要掌握一些足够我们使用的功能即可。  首先我们来学习如何启动和退出vim。启动模式主要两大类  ，这里我们就用直接终端输入vim启动即可  这里有七种退出模式，这里介绍一种很简单的模式  先启动vim |  | Vim的学习了解 |
| PPT第24页  进入编辑模式 |  | Vim的学习了解Vim的学习了解 |
| PPT第25页 退出之前都要按“Esc”回到一般模式 |
| PPT第26页  输入“ZZ”返回终端界面 |  | Vim的学习了解 |
| PPT 第27 页  下面以一个简单的例子来演示vim的使用。  第一步打开终端，新建一个TXT文档 | 命令：Vin vimlearn.txt | Vim的学习了解 |
| PPT第28页  进入一般模式  输入i，看到左下角出现插入  进入编辑模式 | 命令：i | Vim的学习了解 |
| PPT第29页  进入编辑模式  按下“Esc”，退出编辑  再输入：wq  保存并退出 | Hello ros！ 按下：“Esc” 命令 ：wq（包含：） | Vim的学习了解 |
| PPT第30页  打开文档 | 在终端输入命令：Vin vimlearn.txt | Vim的学习了解 |
| PPT第31页  下方有一堆选项 这里我们选择只读模式 | 输入 O | Vim的学习了解 |
| PPT第32页  看到我们编辑的文档内容：hello ros！ |  | Vim的学习了解 |
| PPT第33页  下面我们进行模拟练习  主要对针对vim的使用，这里我们尽量选择与我们学习机器人相关的知识。  以tut-软件包中任意一个launch文件为样本，使用vim对其操作。（上重德智能GitHub的网站下载）  主要有编写，保存退出，复制整个launch文件到另一个文档。  (建议学生下载其他的文档进行操作，对文件内容进行逐行逐段的输入，感受更为清楚。) | [重德智能GitHub](https://github.com/DroidAITech/)：<https://github.com/DroidAITech/> |  |
| PPT 第34页  这里我们以navigation\_demo.launch文件的复制演示 | 输入命令：vim navigation\_demo.launch |  |
| PPT第35页  我们再打开launch文件的文档， 比对vim打开下的内容，左右比对，发现没出错 |  |  |
| PPT第36页  想想我们上面讲过的操作，要想进入命令模式，首先得进行“esc操作”。不过此时我们只打开文档，尚未进行编辑 | 输入命令：“：r!cat x.txt” |  |
| PPT第37页  进行拷贝整个文档操作，我们将这个launch文件的内容拷贝到x.txt文档中去。  在此之前，先建立个x.txt 文档，这个上面已经教过了  看主界面左下方，出现的”多了21行”，说明我们确实往新的x文档里粘贴了内容。 |  |  |
| PPT第38页  我们看下结果，确实两个文档内容一样，说明我们的操作时成功的。  后面给出了更为详细的vim学习教程的连接，有需要的可以自己查阅。 |  |  |
| PPT第39页  这里再介绍几个常见的ROS 开发用的IDE。  QT、Eclipse、VScode、roboware  另外附上下载链接以及简单的安装教程 |  |  |
| PPT 第40页  Qt Creator是跨平台的 Qt IDE， Qt Creator 是 Qt 被 Nokia 收购后推出的一款新的轻量级集成开发环境（IDE）。此 IDE 能够跨平台运行，支持的系统包括 Linux（32 位及 64 位）、Mac OS X 以及 Windows。根据官方描述，Qt Creator 的设计目标是使开发人员能够利用 Qt 这个应用程序框架更加快速及轻易的完成开发任务。Qt Creator 包括项目生成向导、高级的 C++ 代码编辑器、浏览文件及类的工具、集成了 Qt Designer、Qt Assistant、Qt Linguist、图形化的 GDB 调试前端，集成 qmake 构建工具等。 |  |  |
| PPT 第41页  Visual Studio Code (简称 VS Code / VSC) 是一款免费开源的现代化轻量级代码编辑器，支持几乎所有主流的开发语言的语法高亮、智能代码补全、自定义热键、括号匹配、代码片段、代码对比 Diff、GIT 等特性，支持插件扩展，并针对网页开发和云端应用开发做了优化。软件跨平台支持 Win、Mac以及 Linux。它的特点免费、开源、多平台，以及集成git，代码调试，插件丰富等有点，被大家逐渐熟知和认可  在这里再介绍vscode的安装 |  |  |
| PPT 第42-43页   Eclipse 是一个开放源代码的、基于Java的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括Java开发工具（Java Development Kit，JDK）。虽然大多数用户很乐于将 Eclipse 当作 Java集成开发环境来使用，但 Eclipse 的目标却不仅限于此。Eclipse 还包括插件开发环境（Plug-in Development Environment，PDE），这个组件主要针对希望扩展 Eclipse 的软件开发人员，因为它允许他们构建与 Eclipse 环境无缝集成的工具。由于 Eclipse 中的每样东西都是插件，对于给 Eclipse 提供插件，以及给用户提供一致和统一的集成开发环境而言，所有工具开发人员都具有同等的发挥场所。 |  |  |
| PPT 第44页  **RoboWare Studio**是**ROS**的一款**IDE**，基于微软开源的**VSCode**开发，功能比较强大，使得开发更加快速、简单。 |  |  |
| PPT第45-46页  这里我们了解一下ros-tut的软件包  具体操作：先进去重德智能的GitHub网页  然后点击如图所示的的选项 |  |  |
| PPT第47页  如图所示，ros-tut的软件包可以大致了解到了 |  |  |
| PPT：第48-49页  最后我们了解下IP跟端口的基础网络知识，这部分大家可以自己通过连接了解 |  |  |
| PPT：第50页  我们了解一下关于保存log  Log 保存位置  在上面的运行结果中，保存日志的位置是“/home/xxx/.ros/log /”，但实际上它被记录在设置ROS\_ HOME环境变量的地方。如果ROS\_HOME环境变量未设置，则默认值为“~/.ros/log /” |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |