|  |  |  |
| --- | --- | --- |
| **PPT第1页**：  下面我们介绍一些机器人控制的常用操作，和一些linux系统中常用工具。 |  |  |
| **PPT第2页：**  上节课我们介绍了如何使用键盘来控制机器人运行，我们发现机器人的运行实际上是比较缓慢的，如何能加速我们的机器人呢 |  |
| **PPT第3页：**  启动键盘控制，在这里我们看到可以使用去，q，w，e键来为机器人进行加速，Q/Z用来控制机器人的最大速度，W/X,E/C分别用来控制线速度和角速度。 |  |
| **PPT第4，5页：**  同学们可以自己操作一下，看一下机器人运动的变化 |  |
| **PPT第6页：**  接下来我们尝试一下查看机器人的摄像头所对应的画面  此时摄像头正好面对窗户 | 终端输入rosrun image\_view image\_view image:=/camera/rgb/image\_raw |
| **PPT第7页**  我们可以启动键盘控制，看一下机器人在运动过程中摄像头画面的变化情况 | 操作演示$ roslaunch robot\_sim\_demo robot\_keyboard\_teleop.py |
| **PPT第8页：**  学会了控制机器人之后我们来了解一下，Xbot机器人的启动涉及到了多少个节点。可以看到gazebo的前端和后端，robot state publisher用来发布机器人当前的状态，spawner用来启动我们的机器人模型。cmd\_vel\_mux,这是一个速度选择器的节点。 | 终端输入rosnode list |
| **PPT第9页：**  我们来看一下这个cmd\_vel\_mux节点的详细信息。 | 终端输入 rosnode info /cmd\_vel\_mux |
| **PpT第10页**  说完了机器人的一些基本操作我们介绍一下ubuntu中输入法的设置和github的使用。首先介绍下Fcitx输入法的安装 |  |
| **PPT第11页**  安装后重启，在语言设置中选择Fcitx并添加拼音和英文键盘，在全局配置栏中设置输入切换快捷键。 |  |
| **PPT第12页**  这一部分我们介绍一下Github中下载与上传操作，gitHub是一个面向[开源](https://baike.baidu.com/item/%E5%BC%80%E6%BA%90/20720669)及私有[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6/12053)项目的托管平台，因为只支持git 作为唯一的版本库格式进行托管，故名gitHub |  |
| **PPT第13页**  使用ros系统，需要用到许多数据包，有些时候你需要使用的ROS数据包并没有Debian包的形式，这时你需要从数据源安装它。代码开发人员通常使用的有三种主流的版本控制系统：SVN，Git和Mercurial。  下面介绍一下如何使用Git来下载你需要的代码资源。 首先要说明的是在github上只能clone一个完整的项目，为了保证一个项目的完整性，github不允许仅clone单个件或文件夹。 |  |
| **PPT第14页**  **登录github https://github.com在搜索框中通过关键字搜索你需要的项目** | 操作打开<https://github.com>，搜索相关项目  Simulation Tools In ROS |  |
| **PPT第15页**  **复制当前项目地址** |  |  |
| **PPT第16页**  **创建一个你自己的工作空间，并在终端中克隆该项目** | 操作，在终端输入  cd ~/catkin\_ws/src  git clone https://github.com/ros-simulation/gazebo\_ros\_pkgs.git |  |
| **PPT第17页**  **接下来我们介绍如何上传本地文件到Github仓库，首先需要创建Github账号和绑定SSH**  **创建账号的过程在此不进行详细叙述了**  **安装git**  **配置本机git的两个重要信息，user.name和user.email,终端输入如下命令即可设置** | 操作演示  $ sudo apt-get install git  $ git config --global user.name "Your Name"  $ git config --global user.email "email@example.com" |
| **PPT第18页**  **(4)通过命令 git config --list,查看是否设置成功**  **(5)查看home目录下是否有.ssh目录，一般情况是没有的，需要我们敲命令生成这个目录，在终端输入** | 操作演示  $ ssh-keygen -t rsa -C youremail@example.com |
| **PPT第19页**  **(6)进入home目录下的.ssh目录你会看到两个文件id\_rsa和id\_rsa.pub,id\_rsa是私钥，id\_rsa.pub是公钥。**  **(7)将id\_rsa.pub文件中的内容拷贝一下** |  |
| **PPT第20页**  **(8)进入你自己的github，进入Settings->SSH and GPG keys->New SSH key**  **在Key那栏下面将(7)拷贝的东西粘贴进去就可以了，最后点击 Add SSH key按钮添加。** |  |
| **PPT第21页**  **绑定完SSH，我们现在需要正式上传文件**  **(1)进入到要上传文件的文件夹，右键“在终端打开”**  **(2)打开终端输入 $ git init**  **目的是初始化本地仓库(文件夹)**  **(3)打开终端输入 $git add .**  **添加本地仓库（当前文件夹）的所有文件** | 操作演示  $ git init  $git add. |
| **PPT第22页**  (4)终端输入 $ git commit -m "first commit"  参数-m可以说明本次提交的描述信息  (5)终端输入 $ git remote rm origin  先清空当前远程oringin | 操作演示  $ git commit –m “first commit”  $ git remote rm origin |
| **PPT第23页**  (6)终端输入 git remote add origin https://github.com/你的账号名/你新建的仓库名.git  (7)终端输入git push -u origin master上传 | 操作演示  $ git remote add origin <https://github.com/你的账号名/你新建的仓库名.git>  $ git push -u origin master |  |
| **PPT第24页**  上传成功如下图 |  |  |
| **PPT第25页**  最后我们提供了一些Linux基本指令，希望可以帮助大家进行操作，好了今天的课程就到这里 |  |