实验报告

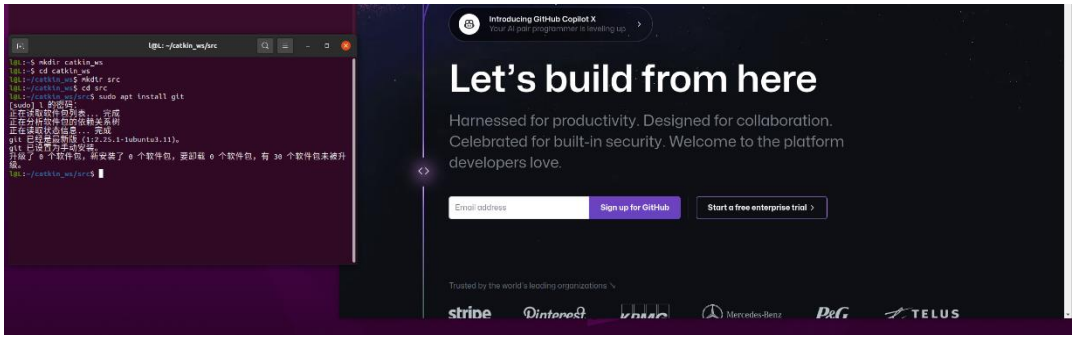
姓名：廖嘉辉
时间：6.26-7.9

【实验目的】学习 ROS

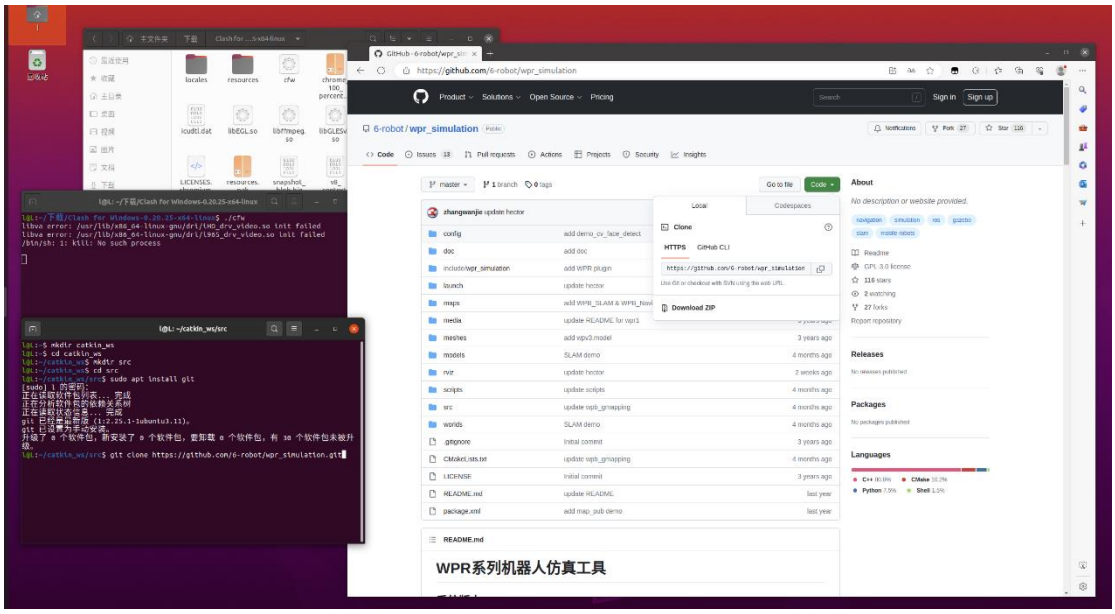
【实验过程】

1. 学习从 GitHub 下载使用软件包

先通过“mkdir 文件夹名称”创建文件夹，“cd 文件夹名”进入文件夹，建立 catkin_ws/src 文件夹作为工作空间

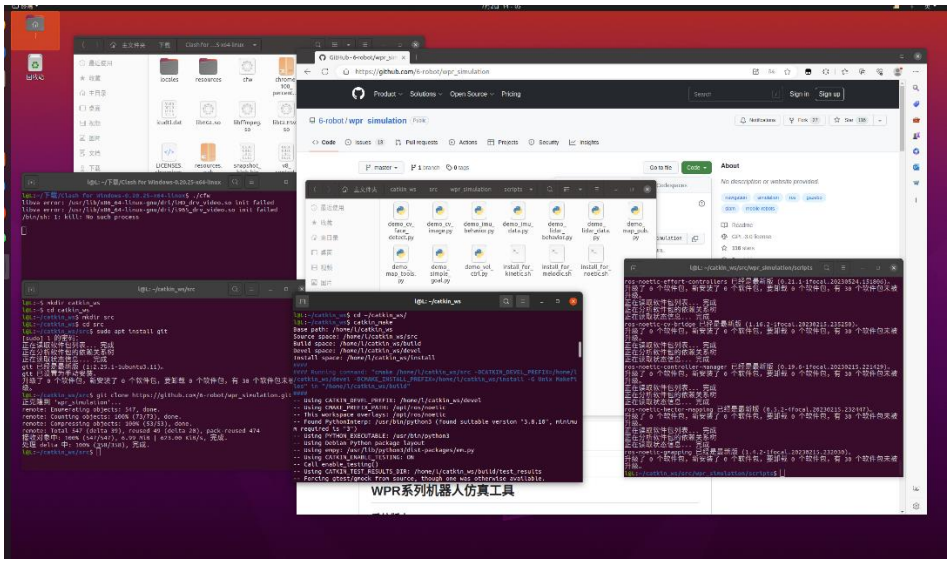


再“sudo apt install git”安装更新 git，在 GitHub 找到 ROS 软件包后，在 code/local 下复制网址，后在终端中用“git clone 网址”安装软件包。

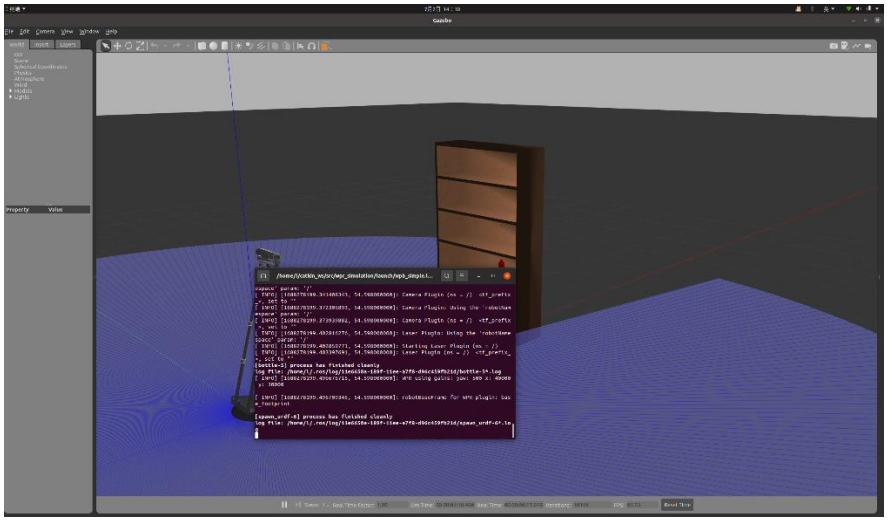


在终端打开新软件包下的 scripts，输入“./install_for_noetic.sh”安装编译需要的依赖库“cd ~”回到主目录，在

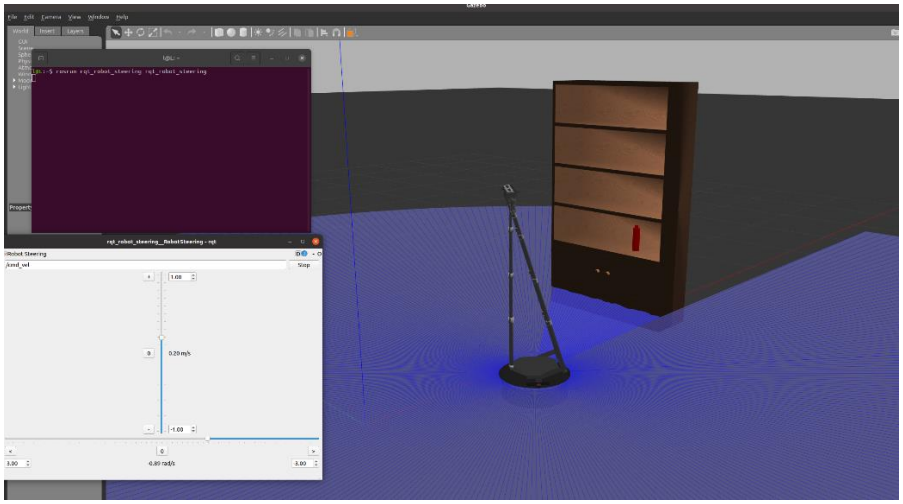
~/catkin_ws 目录下运行“catkin_make”进行编译



使用“source ~/catkin_ws/devel/setup.bash”载入工作空间环境设置，再用“roslaunch wpr_simulation wpb_simple.launch”运行软件包



“rosrun rqt_robot_steering rqt_robot_steering”打开运动控制软件，可对机器人运动进行控制



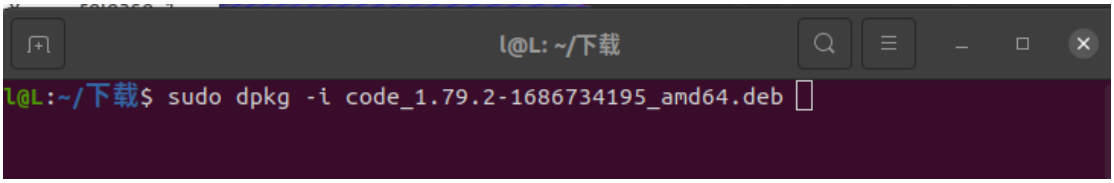
“gedit ~/.bashrc”打开终端初始化脚本，在末尾加上设置工作空间环境参数的 source 指令“source ~/catkin_ws/devel/setup.bash”，即可直接运行 ROS 程序

```
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112   if [ -f /usr/share/bash-completion/bash_completion ]; then
113     . /usr/share/bash-completion/bash_completion
114   elif [ -f /etc/bash_completion ]; then
115     . /etc/bash_completion
116   fi
117 fi
118 source /opt/ros/noetic/setup.bash
119 source ~/catkin_ws/devel/setup.bash
```

sh ▾ 制表符宽度：8 ▾ 第 119 行，第 36 列 ▾ 插入

2. VScode 安装与设置

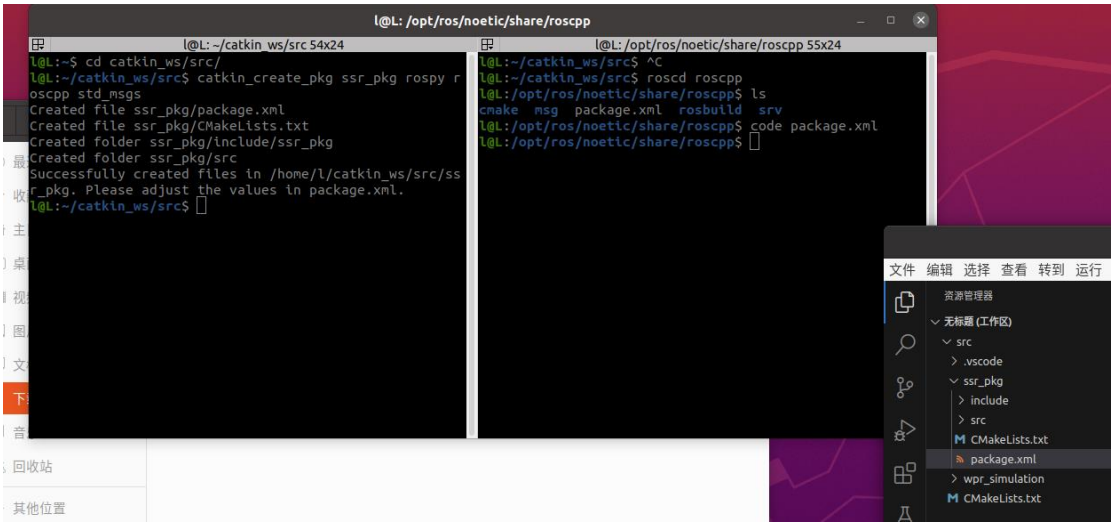
在 VScode 官方网站下载软件包后，用“sudo dpkg -i 软件包名”安装



在扩展中下载 ROS，Cmake Tools 等插件，

3. 创建 Node 节点

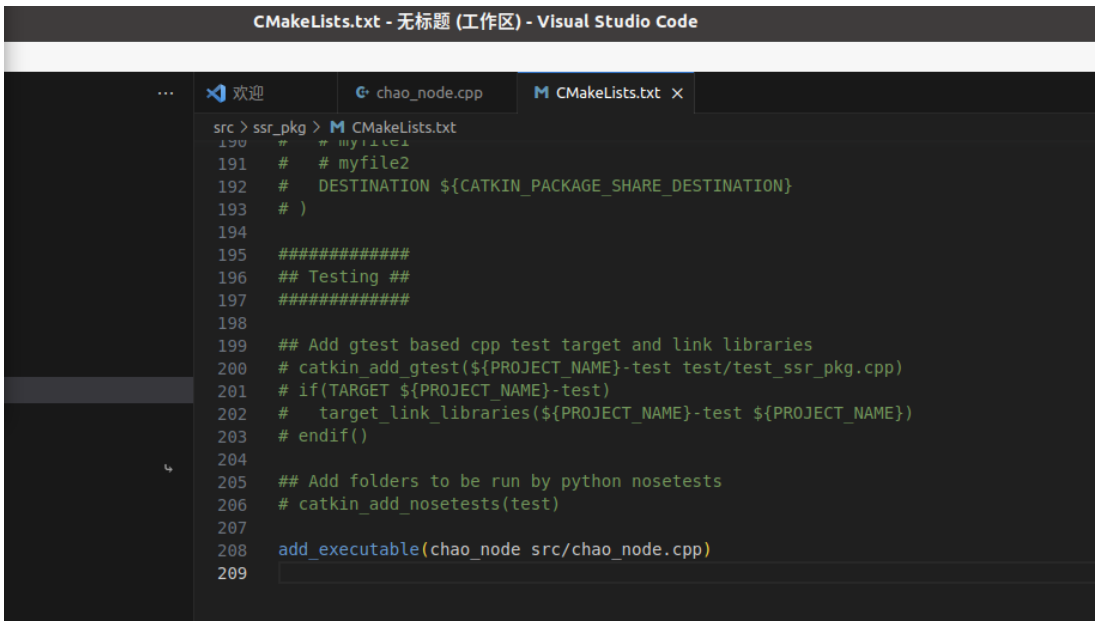
创建 package 软件包



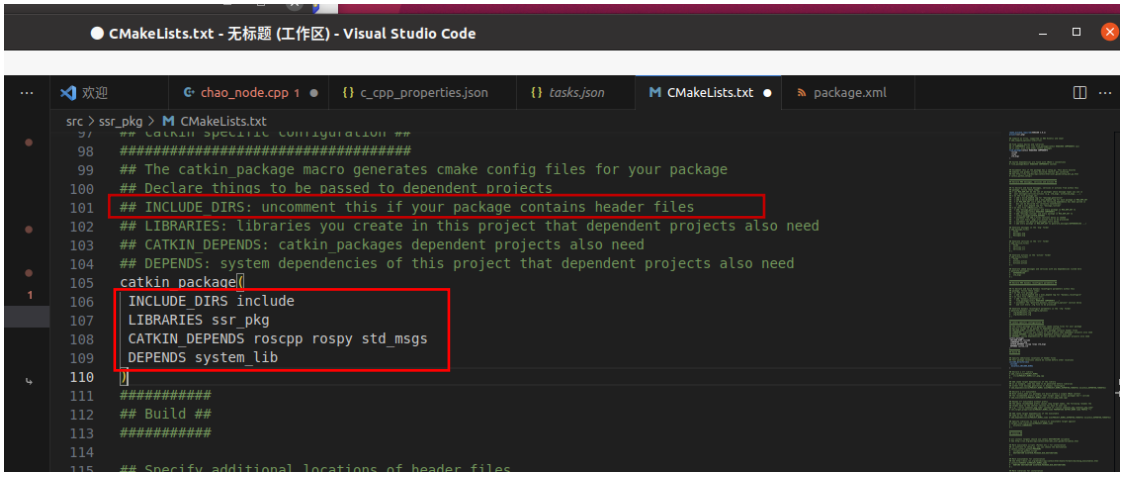
“catkin_create 包名 依赖项”

“roscd”在终端中进入指定软件包的文件地址

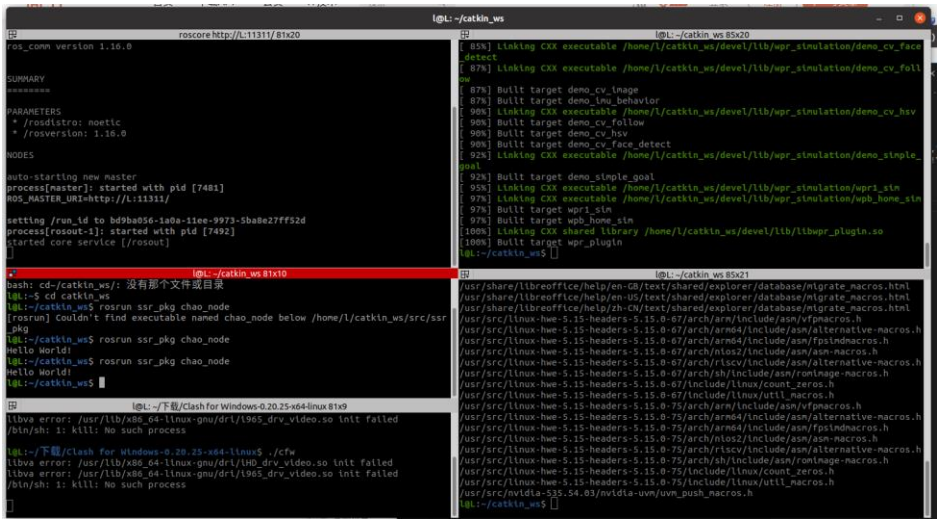
为新建节点添加编译规则



括号内第一项为可执行文件名字，第二项为指定从哪个代码文件进行编译
写完代码后，编译时报错，显示“ros/ros.h:没有哪个文件或目录”
原因：catkin_package()括号内的内容被注释

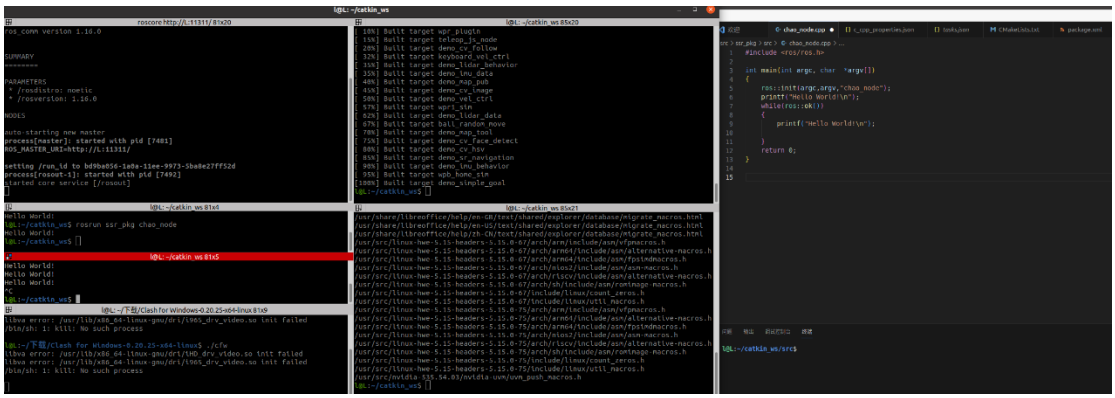


去注释后可在终端中用“catkin_make”完成编译
用“rosrun 包名 节点名”运行节点



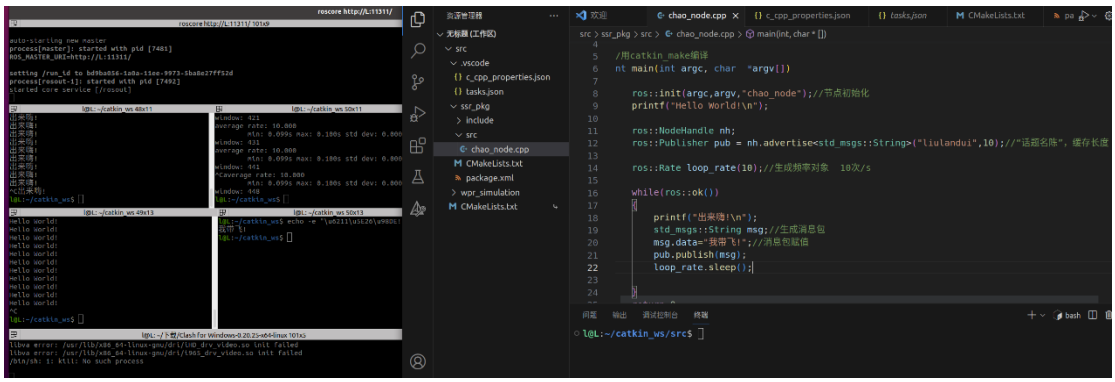
完善 ROS 节点，将 ros::init 函数所在库文件链接进来

```
209
210   target_link_libraries(chao_node
211     ${catkin_LIBRARIES}
212   )
```



ros::ok()让终端能够相应外部信号

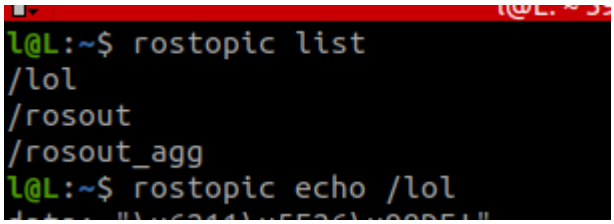
4.编写 publisher 发布者节点



在 ros index 中查找 std_msgs 消息类型

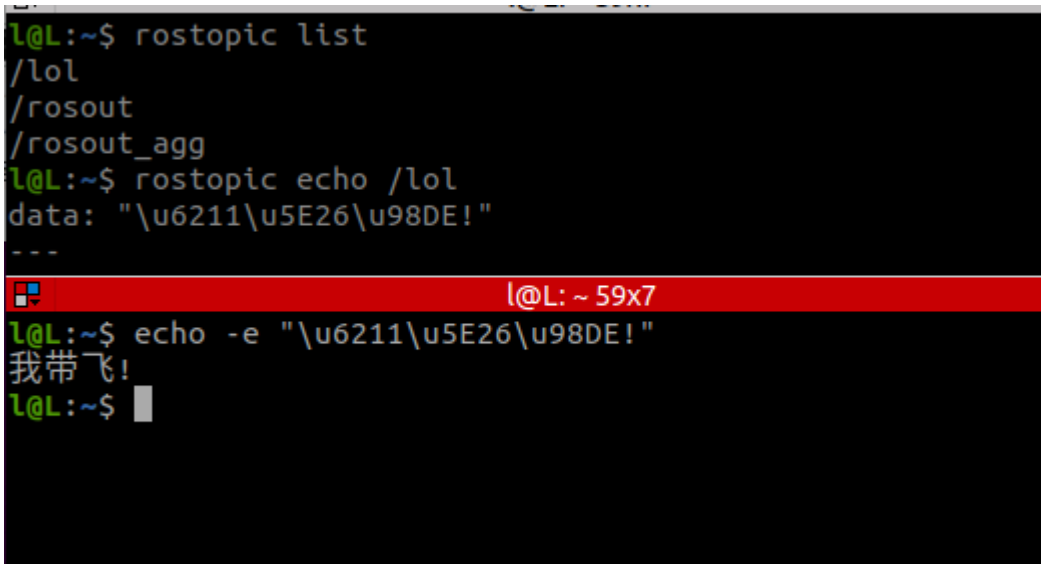
其中 string 类型为 string data，data 为变量名，消息赋值时直接给 data 赋值即可

用“rostopic list”可查看当前 ROS 系统中活跃的话题有哪些

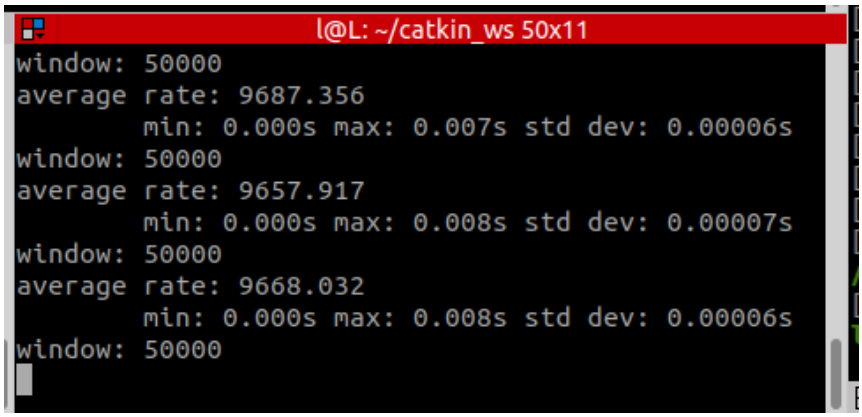


用“rostopic echo 话题名称”查看话题内容

其中 data 为赋值的字符串变量，后面为 unicode 字符，用“echo -e 编码”可查看编码内容



用“rostopic hz 话题名称”可查看消息发送频率



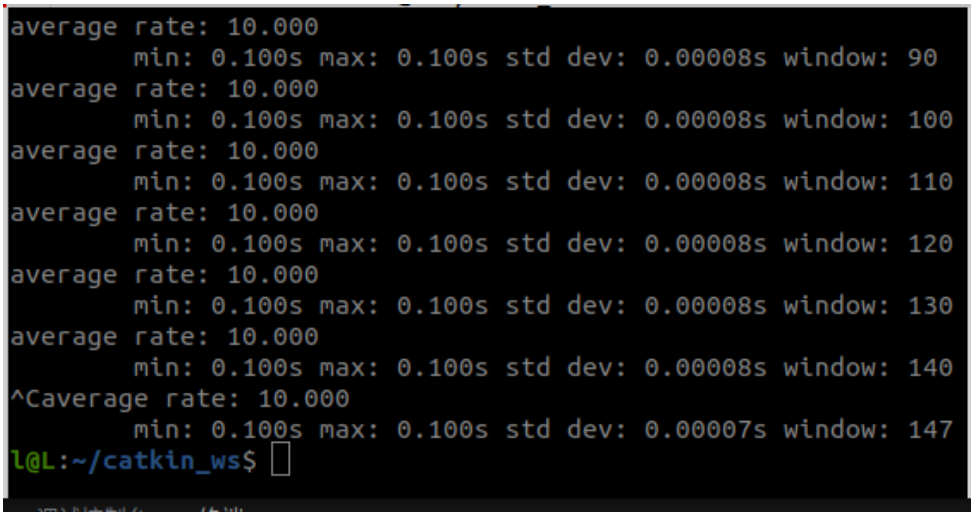
调用.sleep 函数会在 while 循环内做一个短时间阻塞，以此控制循环的执行频率

```
ros::NodeHandle nh;
ros::Publisher pub = nh.advertise<std_msgs::String>("liulandui",10);//“话题名陈”，缓存长度

ros::Rate loop_rate(10);//生成频率对象  10次/s

while(ros::ok())
{
    printf("出来嗨!\n");
    std_msgs::String msg;//生成消息包
    msg.data="我带飞!";//消息包赋值
    pub.publish(msg);
    loop_rate.sleep();

}
```



【实验结果】

1. 能够从 github 下载使用 ROS 软件包
2. 学会 ROS 节点的编译与运行
3. 了解 publisher 发布者部分操作