

1. 了解 Linux 系统：阅读《鸟哥的 Linux 私房菜》自学前三部分内容，或利用互联网进行学习，简答以下

问题；（3 分）

(1) 列举三个你常用的 Linux 命令，并说明他们的功能。

(2) 一句话简要介绍 Vim 的功能，如何在 Vim 中进行插入和删除，如何保存并退出 Vim？

(3) 列举两种常用的 Linux 压缩和解压缩命令。

解答：

(1)

ls 显示当前文件夹路径下面的文件和文件夹

pwd 显示当前文件夹路径

cd 修改当前文件夹路径

(2)

Vim 是一个可以在终端使用的文本编辑器

插入：进入 vim 后，按下 i，即可在光标之前插入文本

其他有关 vim 插入命令汇总如下：

a 在光标之后追加

A 在一行的结尾处追加

I 在一行的开头处插入

o 在光标所在位置的下一行打开新行插入

o 在光标所在位置的上一行打开新行插入

删除：进入 vim 后，按下 x，即可删除光标处的文本

或者按下 dd，可以删除一行

其他有关 vim 删除文本 的命令汇总如下：

- x   删除光标下的字符 ("dl" 的缩写)
- X   删除光标前的字符 ("dh" 的缩写)
- D   从当前位置删除到行尾 ("d\$" 的缩写)
- dw   从当前位置删除到下一个单词开头
- db   从当前位置删除到前一个单词的开头
- diw   删除光标上的单词 (不包括空白字符)
- daw   删除光标上的单词 (包括空白字符)
- dG   删除到文件末
- dgG   删除到文件首

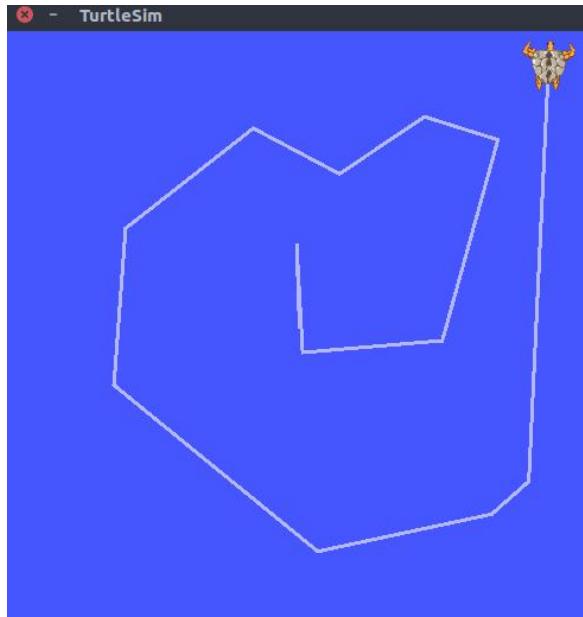
保存并推出 vim： 按下 wq 保存并推出

2. 了解 ROS：观看 ROS 免费公开课或前往 ROS 官网学习官方教程，安装好 ROS，提供运行小海龟跑的截图；（3 分）

运行步骤：

新开三个终端，分别输入：roscore， rosrun turtlesim turtlesim\_node， rosrun turtlesim turtle\_teleop\_key

运行截图如下：



3. 学习机器人姿态描述入门材料，完成坐标转换推导；（3分）

设机器人的世界坐标为  $x_a, y_a$ ，其相对于世界坐标系的方向为  $\theta_a$ （右手坐标系）。假设机器人旁边有一物体

在世界坐标系下的位姿为  $(x_b, y_b, \theta_b)$ ，请问：

(1) 该物体相对于机器人的位置和朝向是什么，即该物体在当前机器人坐标系下的位姿是多少？

(2) 机器人此时朝它的正前方（机器人坐标系 X 轴）行进了  $d$  距离，然后又转了  $\theta_d$  角，请问物体此时在这一时刻机器人坐标系下的位姿是多少？

(3) 在第（2）问的情况下，机器人发射一道激光，发现有一物体在它  $x_c, y_c, \theta_c$  处，即  $(x_c, y_c, \theta_c)$  为物体在机器人坐标系下的位姿，请问该物体在世界坐标系下的位姿为多少？

解答：

(1)

$$T_{wa} = \begin{bmatrix} \cos\theta_a & -\sin\theta_a & x_a \\ \sin\theta_a & \cos\theta_a & y_a \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_{wb} = \begin{bmatrix} \cos\theta_b & -\sin\theta_b & x_b \\ \sin\theta_b & \cos\theta_b & y_b \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_{ab} = T_{wa}^{-1} T_{wb}$$

$$= \begin{bmatrix} \cos\theta_a & \sin\theta_a & -(x_a \cos\theta_a + y_a \sin\theta_a) \\ -\sin\theta_a & \cos\theta_a & x_a \sin\theta_a + y_a \cos\theta_a \\ 0 & 0 & 1 \end{bmatrix} \cdot T_{wb}$$

$$= \begin{bmatrix} \cos(\theta_b - \theta_a) & -\sin(\theta_b - \theta_a) & x_b \cos\theta_a + y_b \sin\theta_a - (x_a \cos\theta_a + y_a \sin\theta_a) \\ \sin(\theta_b - \theta_a) & \cos(\theta_b - \theta_a) & -x_b \sin\theta_a + y_b \cos\theta_a + (x_a \sin\theta_a + y_a \cos\theta_a) \\ 0 & 0 & 1 \end{bmatrix}$$

Fall into the pit, a gain in your wit.  
吃一堑，长一智。

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Mo	Tu	We	Th	Fr	Sa	Su
Memo No. _____						
Date / /						
二、该物体在当前机器人的位姿是：						
$(x_{ab}, y_{ab}, \theta_{ab}) = (T_{ab}(0,2), T_{ab}(1,2), \theta_b - \theta_a)$						

(2)

$$T_{wa'} = \begin{bmatrix} \cos(\theta_a + \theta_d) & -\sin(\theta_a + \theta_d) & x_a + d \\ \sin(\theta_a + \theta_d) & \cos(\theta_a + \theta_d) & y_a \\ 0 & 0 & 1 \end{bmatrix}$$

同理，计算：

$$T_{ab}' = T_{wa'} T_{wb}$$

$$= \begin{bmatrix} \cos(\theta_b - \theta_a - \theta_d) & -\sin(\theta_b - \theta_a - \theta_d) & x_b \cos(\theta_a + \theta_d) + y_b \sin(\theta_a + \theta_d) \\ \sin(\theta_b - \theta_a - \theta_d) & \cos(\theta_b - \theta_a - \theta_d) & -(x_a + d) \cos(\theta_a + \theta_d) - y_a \sin(\theta_a + \theta_d) \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \sin(\theta_b - \theta_a - \theta_d) \cos(\theta_b - \theta_a - \theta_d) - x_b \sin(\theta_a + \theta_d) + y_b \cos(\theta_a + \theta_d) \\ \sin(\theta_b - \theta_a - \theta_d) \sin(\theta_b - \theta_a - \theta_d) + (x_a + d) \sin(\theta_a + \theta_d) + y_a \cos(\theta_a + \theta_d) \end{bmatrix}$$

该物体在机器人的位姿为

$$(x_{ab}', y_{ab}', \theta_{ab}') = (T_{ab}'(0, 2), T_{ab}'(1, 2), \theta_b - \theta_a - \theta_d)$$

Happiness is a station between too much and too little.

幸福是太多和太少之间的一站。

(3)

$$T_{wa'} \cdot \begin{bmatrix} \cos \theta_c & -\sin \theta_c & x_c \\ \sin \theta_c & \cos \theta_c & y_c \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta_a + \theta_d + \theta_c) & -\sin(\theta_a + \theta_d + \theta_c) & x_c \cos(\theta_a + \theta_d) - y_c \sin(\theta_a + \theta_d) + x_a + d \\ \sin(\theta_a + \theta_d + \theta_c) & \cos(\theta_a + \theta_d + \theta_c) & x_c \sin(\theta_a + \theta_d) + y_c \cos(\theta_a + \theta_d) + y_a \\ 0 & 0 & 1 \end{bmatrix}$$

∴ 位姿为

$$(T(0, 2), T(1, 2), \theta_a + \theta_d + \theta_c)$$

4. 完成基础数学坐标转换的代码作业。 (3 分)

代码如下：

```
int main(int argc, char** argv)
{
    // 机器人 B 在坐标系 O 中的坐标：
    Eigen::Vector3d B(3, 4, M_PI);

    // 坐标系 B 到坐标 O 的转换矩阵：
    Eigen::Matrix3d TOB;
    TOB << cos(B(2)), -sin(B(2)), B(0),
        sin(B(2)), cos(B(2)), B(1),
        0, 0, 1;

    // 坐标系 O 到坐标 B 的转换矩阵：
    Eigen::Matrix3d TBO = TOB.inverse();

    // 机器人 A 在坐标系 O 中的坐标：
    Eigen::Vector3d A(1, 3, -M_PI / 2);

    // 求机器人 A 在机器人 B 中的坐标：
    Eigen::Vector3d BA;

    // TODO 参照 PPT
    // start your code here (5~10 lines)
    Eigen::Matrix3d TOA;
    TOA << cos(A(2)), -sin(A(2)), A(0),
        sin(A(2)), cos(A(2)), A(1),
        0, 0, 1;

    Eigen::Matrix3d TAB = TBO * TOA;
    BA[0] = TAB(0, 2);
    BA[1] = TAB(1, 2);
    BA[2] = atan2(TAB(1, 0), TAB(0, 0));

    // end your code here
    cout
        << "The right answer is BA: 2 1 1.5708" << endl;
    cout << "Your answer is BA: " << BA.transpose() << endl;
    return 0;
}
```

代码输出结果如下：

```
sunm@sunm-Legion:~/work/code/shenlan-2d/HW1/basicTransformStudy/build$ ./basicTransformStudy
The right answer is BA: 2 1 1.5708
Your answer is BA:      2      1 1.5708
```