

# Linux 入门基础

---

Linux 是一个开源的操作系统，它的内核是由 Linus Torvalds 于 1991 年创建的。

很多公司和组织都基于 Linux 操作系统，例如 Google、Red Hat、Ubuntu 等，开发自己的 Linux 发行版。

目前ROS系统主要运行在Ubuntu操作系统上，而Ubuntu操作系统是基于Debian Linux 发行版的。

所以，如果想要深入学习和实践机器人的话，最好在自己的电脑上安装好Ubuntu操作系统，使用[WSL](#)或者虚拟机[VirtualBox](#) 或者 [VMware](#)。

更重要的是，目前多数的人工智能环境都是基于Ubuntu操作系统的，所以学习和实践机器人的话，最好在Ubuntu环境下进行。

## 终端

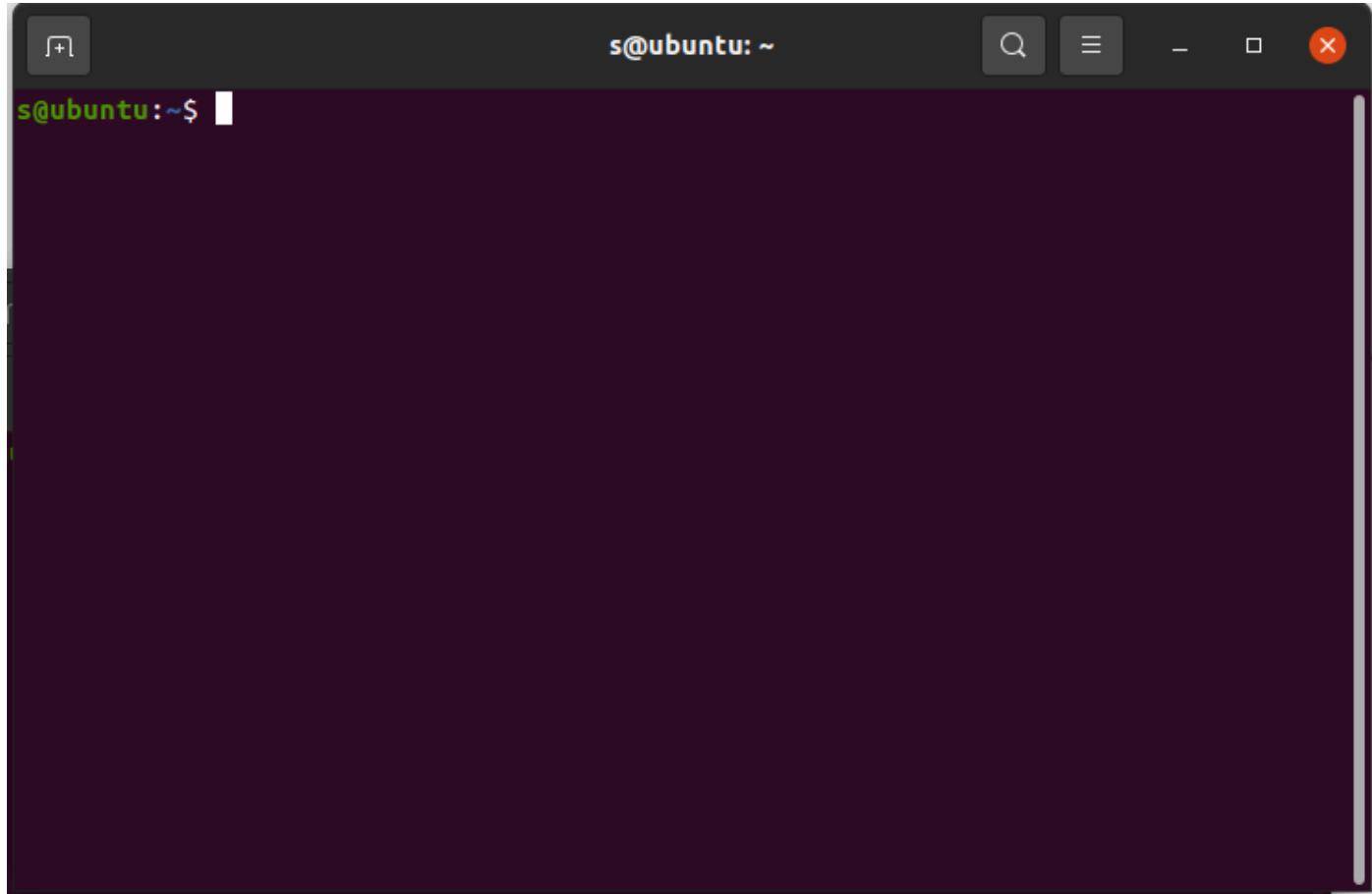
---

首先，我们要认识一个东西：终端。

终端是一个文本界面，用户可以在其中输入命令，操作系统会根据用户输入的命令来执行相应的操作。

在Ubuntu操作系统中，终端是一个非常重要的工具，用户可以在终端中执行各种命令，例如安装软件、配置系统、运行程序等。

同时按下ctrl键,alt键和t键，会召唤出一个终端界面：



在这个图中，s是用户名，@分隔符后面的部分是主机名 ubuntu，:是分隔符号，~代表着当前用户的主目录也就是当前终端所在的路径。

你可以在这个黑框框里输入一些指令来执行相应的操作，例如输入ls命令来列出当前目录下的文件和文件夹。

或者输入cd命令来切换当前目录，例如输入

```
cd ~/
```

来切换到当前用户的主目录，也就是打开终端默认的路径。

```
s@ubuntu:~$ ls
bt                  frames.gv          note.txt        src
catkin_ws           frames.pdf         Pictures        Templates
codes               linux_exp          Public         tf_tree_log
Desktop             logger_output      README.md      unitree_sdk2-main
Documents           'mr600&petbot.pdf'  ros_ws         Videos
Downloads           Multi_ws          Multi_ws.zip  实验日志
exploration_ws     Multi_ws.zip      Music          slam_gmapping.zip
explore ye          Music            slam_method
s@ubuntu:~$
```

# Linux 基础指令小实验（跟着一步一步做）

## 实验安全须知（非常重要）

在本实验中你将接触到会真实修改文件和目录的指令，请务必认真阅读：

- **rm**、**mv** 会直接删除或移动文件
- **rm -rf** 极其危险，一旦路径写错，数据无法恢复
- 本实验只允许在 **~**（你的主目录）下操作
- 严禁在 **/**、**/home**、**/usr** 等系统目录执行 **rm -rf**
- 因误操作导致系统损坏，需要自行承担维修或赔偿责任

不懂可以随时提问

---

## 一、实验目标

通过一个完整的小实验，掌握以下内容：

- Linux 终端中的 **路径概念**
- 常见文件/目录操作指令 **ls** **mkdir** **touch** **cp** **mv** **rm** **find** **cat**
- 使用 **gedit** 创建和编辑文件

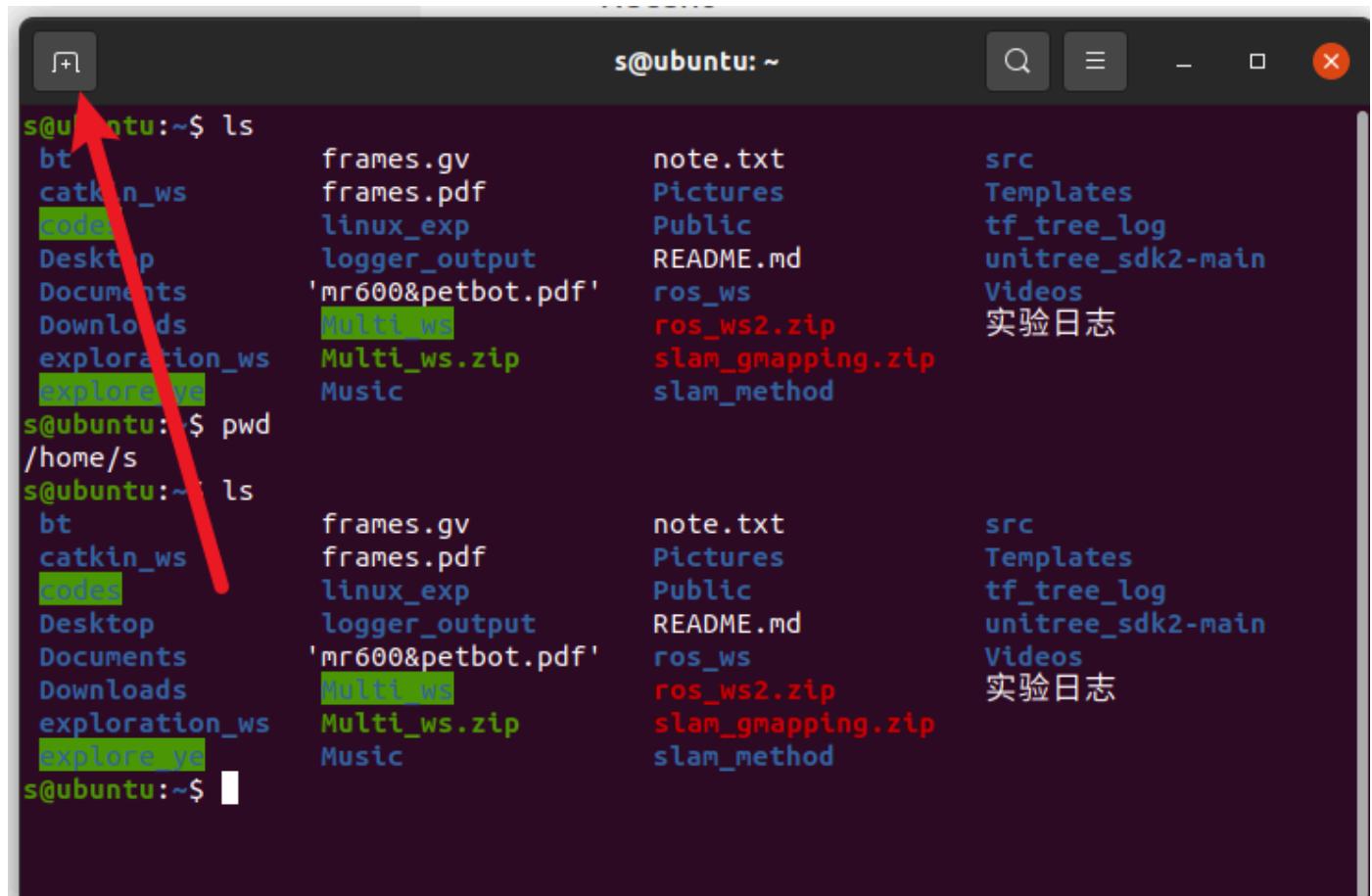
- 使用不同方式执行 Python 程序，理解：
    - 相对路径
    - 绝对路径
    - ~ (home 目录)
  - 了解基本系统与网络指令：ping、top
  - 学会执行一个完整自动化脚本
- 

## 提示

**复制粘贴：**如果想在终端进行复制使用 **ctrl + shift + c**，如果想使用粘贴可以使用 **ctrl + shift + v**。如果在终端中按下 **ctrl + v** 将会有不显示的字符，需要你按两下回退键进行删除。

**关闭程序：**在终端中关闭程序使用 **ctrl + c**，强行终止程序使用 **ctrl + z**。

**新增一个终端** 点击加号可以增加一个终端：



```
s@ubuntu:~$ ls
bt frames.gv note.txt src
catkin_ws frames.pdf Pictures Templates
codes linux_exp Public tf_tree_log
Desktop logger_output README.md unitree_sdk2-main
Documents 'mr600&petbot.pdf'
Downloads Multi_ws ros_ws Videos 实验日志
exploration_ws Multi_ws.zip slam_gmapping.zip
explore ye Music slam_method

s@ubuntu:~$ pwd
/home/s
s@ubuntu:~$ ls
bt frames.gv note.txt src
catkin_ws frames.pdf Pictures Templates
codes linux_exp Public tf_tree_log
Desktop logger_output README.md unitree_sdk2-main
Documents 'mr600&petbot.pdf'
Downloads Multi_ws ros_ws Videos 实验日志
exploration_ws Multi_ws.zip slam_gmapping.zip
explore ye Music slam_method

s@ubuntu:~$
```

## 二、实验一：手动操作

### 1 查看当前位置

确认你当前在自己的 **用户目录 (~)** 下。

```
pwd  
ls
```

```
s@ubuntu:~$ pwd  
/home/s  
s@ubuntu:~$ ls  
bt frames.gv note.txt src  
catkin_ws frames.pdf Pictures Templates  
codes linux_exp Public tf_tree_log  
Desktop logger_output README.md unitree_sdk2-main  
Documents 'mr600&petbot.pdf'  
Downloads Multi_ws ros_ws Videos  
exploration_ws Multi_ws.zip slam_gmapping.zip  
explore_ye Music slam_method
```

## 2 创建实验工作区

```
mkdir linux_exp  
cd linux_exp  
ls
```

可以观察到\$符号前面的路径改变了。

```
s@ubuntu:~$ cd linux_exp  
s@ubuntu:~/linux_exp$ ls  
s@ubuntu:~/linux_exp$
```

## 3 创建文件和目录

在windows中我们会使用右键来创建文件/文件夹，而在终端里，我们也有对应指令：

```
mkdir src  
touch note.txt  
ls
```

```
s@ubuntu:~/linux_exp$ mkdir src  
s@ubuntu:~/linux_exp$ touch note.txt  
s@ubuntu:~/linux_exp$ ls  
note.txt src  
s@ubuntu:~/linux_exp$
```

## 4 使用 gedit 创建 Python 文件

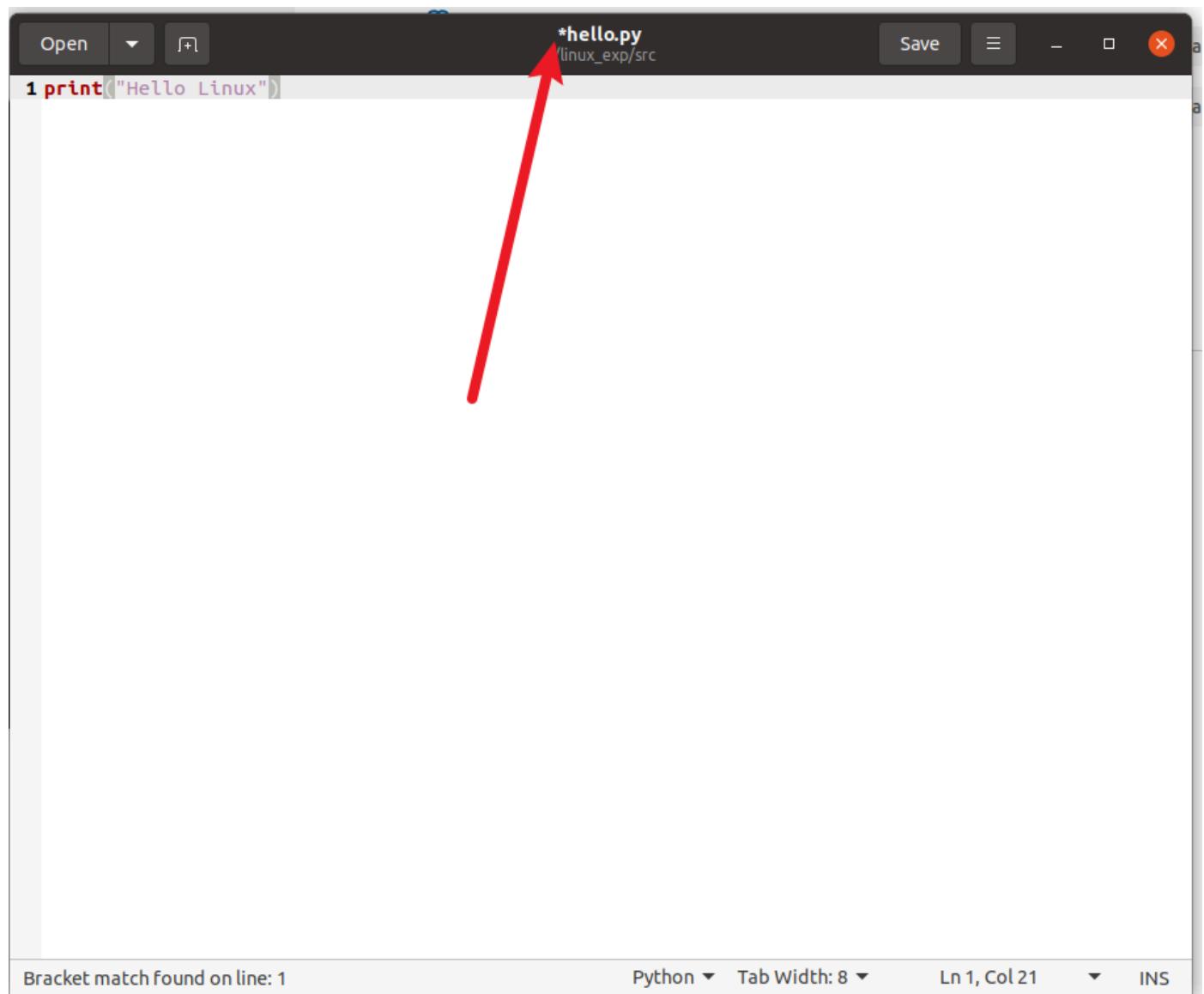
gedit就是一个文本编辑器：

```
gedit src/hello.py
```

在打开的编辑器中输入并保存：

```
print("Hello Linux")
```

效果如下图所示：



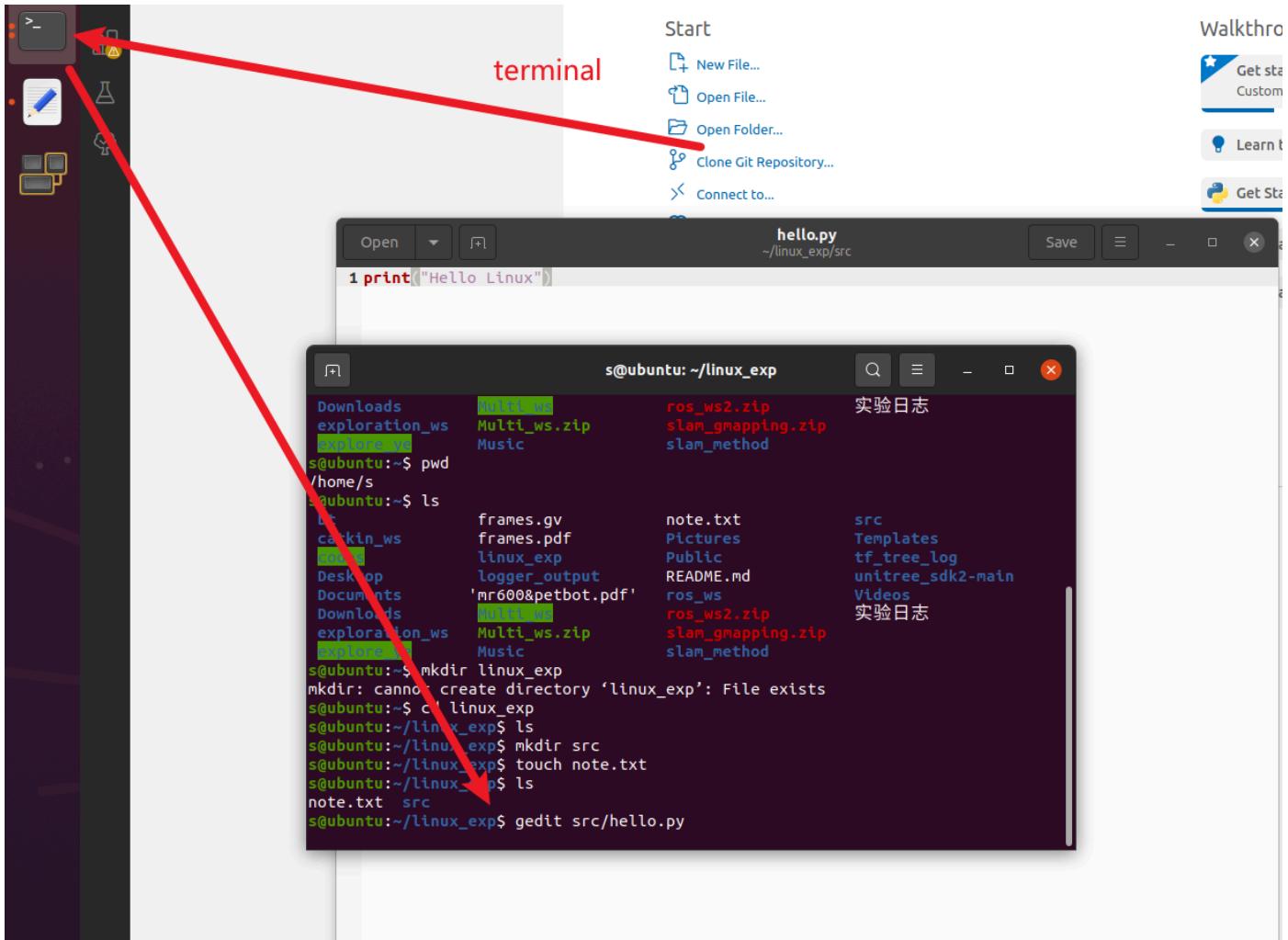
注意需要 **ctrl + s** 保存一下，这里的\*号代表有修改未保存。

保存后，效果如图：

The screenshot shows a code editor window with the following details:

- Top Bar:** Includes "Open" and "Save" buttons, a tab labeled "hello.py ~/linux\_exp/src", and standard window control buttons (minimize, maximize, close).
- Code Area:** Displays the single line of Python code: `1 print("Hello Linux")`.
- Bottom Bar:** Shows "Python" dropdown, "Tab Width: 8" dropdown, "Ln 1, Col 21" status, and "INS" mode indicator.

点击左边的终端图标。



你可以发现，这个gedit指令下面并没有出现新的输入指令的地方。这是因为gedit的文本编辑器正在运行中。你可以在终端中同时按下`ctrl`和`c`进行关闭。

```
s@ubuntu:~/linux_exp$ gedit src/hello.py
^C
s@ubuntu:~/linux_exp$
```

## 5 使用 find 查找文件

我们可以在文件夹中查找我们想要的文件

```
find ~ -name "hello.py"
```

```
s@ubuntu:~/linux_exp$ find ~ -name "hello.py"
/home/s/linux_exp/src/hello.py
```

你可以观察这个文件的路径。它是以/开头的，意味着这是一个绝对路径。

## 6 查看文件内容

```
cat src/hello.py
```

```
s@ubuntu:~/linux_exp$ cat src/hello.py
print("Hello Linux")
s@ubuntu:~/linux_exp$
```

## 7 复制、移动、删除文件（谨慎）

### 复制

```
cp src/hello.py hello_copy.py
ls
```

### 移动/修改名称

```
mv hello_copy.py hello_moved.py
ls
```

### 删除

```
rm hello_moved.py
ls
```

你可以观察这几个指令的对文件产生的效果

注意 不要使用 `rm -rf /` 或乱删目录

## 三、实验二：执行 Python，理解路径

确保你仍在 `~/linux_exp` 目录下：

```
pwd
```

```
s@ubuntu:~/linux_exp$ pwd  
/home/s/linux_exp  
s@ubuntu:~/linux_exp$
```

### 方式一：相对路径

```
python3 src/hello.py
```

### 方式二：home 路径

```
python3 ~/linux_exp/src/hello.py
```

### 方式三：绝对路径

```
python3 /home/<用户名>/linux_exp/src/hello.py
```

请将 <用户名> 替换为当前终端的用户名，即@前面的字符串

### 思考

- 为什么三种方式都能运行？
- 如果你 `cd ~` 之后，哪些还能用？

## 四、实验三：系统与网络指令（观察即可）

### 查看进程（按 q 退出）

```
top
```

```
s@ubuntu: ~/linux_exp
top - 14:01:41 up 29 min,  1 user,  load average: 0.01, 0.06, 0.17
Tasks: 351 total,   1 running, 349 sleeping,   0 stopped,   1 zombie
%Cpu(s): 0.0 us, 2.2 sy, 0.0 ni, 97.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3780.0 total, 194.6 free, 2025.5 used, 1559.9 buff/cache
MiB Swap: 2048.0 total, 2025.0 free,    23.0 used. 1375.6 avail Mem

PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
 841 root      20   0 316268   7540   6300 S  5.9  0.2 0:05.92 vmtoolsd
 2534 s        20   0 347060 133184  81780 S  5.9  3.4 0:13.71 Xorg
 5967 s        20   0 15216   3896  3132 R  5.9  0.1 0:00.03 top
    1 root      20   0 169624 12228   7676 S  0.0  0.3 0:05.47 systemd
    2 root      20   0     0     0     0 S  0.0  0.0 0:00.06 kthreadd
    3 root      0 -20     0     0     0 I  0.0  0.0 0:00.00 rcu_gp
    4 root      0 -20     0     0     0 I  0.0  0.0 0:00.00 rcu_par+
    5 root      0 -20     0     0     0 I  0.0  0.0 0:00.00 slub_fl+
    6 root      0 -20     0     0     0 I  0.0  0.0 0:00.00 netns
    8 root      0 -20     0     0     0 I  0.0  0.0 0:00.00 kworker+
   10 root     0 -20     0     0     0 I  0.0  0.0 0:00.00 mm_perc+
   11 root     20   0     0     0     0 S  0.0  0.0 0:00.00 rcu_tas+
   12 root     20   0     0     0     0 S  0.0  0.0 0:00.00 rcu_tas+
   13 root     20   0     0     0     0 S  0.0  0.0 0:00.03 ksoftir+
   14 root     20   0     0     0     0 I  0.0  0.0 0:01.29 rcu_sch+
   15 root     rt   0     0     0     0 S  0.0  0.0 0:00.01 migrati+
   16 root    -51   0     0     0     0 S  0.0  0.0 0:00.00 idle_in+
```

## 测试网络 (Ctrl + C 停止)

有时候我们会使用这个指令测试主机是否能上网。

```
ping baidu.com
```

```
s@ubuntu:~/linux_exp$ ping baidu.com
PING baidu.com (111.63.65.247) 56(84) bytes of data.
64 bytes from 111.63.65.247 (111.63.65.247): icmp_seq=1 ttl=128 time=44.7 ms
64 bytes from 111.63.65.247 (111.63.65.247): icmp_seq=2 ttl=128 time=44.1 ms
64 bytes from 111.63.65.247 (111.63.65.247): icmp_seq=3 ttl=128 time=44.8 ms
64 bytes from 111.63.65.247 (111.63.65.247): icmp_seq=4 ttl=128 time=44.1 ms
64 bytes from 111.63.65.247 (111.63.65.247): icmp_seq=5 ttl=128 time=42.8 ms
^C
--- baidu.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 42.768/44.077/44.766/0.721 ms
s@ubuntu:~/linux_exp$
```

## 五、实验四：一键自动化脚本（重点）

我们可以写一个脚本，自动从互联网上获取今天深圳的天气以及执行上面我们写的Python代码。

## 1 创建脚本文件

```
cd ~/linux_exp  
gedit run_linux_exp.sh
```

写入以下内容并保存：

```
#!/bin/bash

echo "==== Linux 基础实验脚本：自动执行Python 文件 ==="
python3 ~/linux_exp/src/hello.py

echo "==== Linux 基础实验脚本：获取当天天气 ==="
# 定义要查询的城市（可修改为你的城市，如北京、上海、Guangzhou，支持中英文）
CITY="深圳"

# 输出提示信息
echo "====="
echo "      今日天气查询（来自 wttr.in）"
echo "====="

# 从wttr.in获取天气信息并格式化输出
curl -s "wttr.in/${CITY}?format=3" # 极简输出（城市：天气 温度）
# 脚本结束提示
echo -e "\n====="
echo "          查询完成"
echo "=====
```

```
run_linux_exp.sh
~/linux_exp
Open ▾ + Save ⌂ ⌂ ⌂
1 #!/bin/bash
2
3 echo "==== Linux 基础实验脚本：自动执行Python 文件 ==="
4 python3 ~/linux_exp/src/hello.py
5
6 echo "==== Linux 基础实验脚本：获取当天气 ==="
7 # 定义要查询的城市（可修改为你的城市，如北京、上海、Guangzhou，支持中英文）
8 CITY="深圳"
9
10 # 输出提示信息
11 echo "====="
12 echo "      今日天气查询（来自 wttr.in）"
13 echo "====="
14
15 # 从wttr.in获取天气信息并格式化输出
16 curl -s "wttr.in/${CITY}?format=3" # 极简输出（城市：天气 温度）
17 # 脚本结束提示
18 echo -e "\n====="
19 echo "      查询完成"
20 echo "=====
```

sh ▾ Tab Width: 8 ▾ Ln 20, Col 46 ▾ INS

## 2 添加执行权限

```
chmod +x run_linux_exp.sh
```

## 3 执行脚本

```
./run_linux_exp.sh
```

```
s@ubuntu:~/linux_exp$ chmod +x run_linux_exp.sh
s@ubuntu:~/linux_exp$ ./run_linux_exp.sh
== Linux 基础实验脚本：自动执行Python 文件 ==
Hello Linux
== Linux 基础实验脚本：获取当天天气 ==
=====
    今日天气查询（来自 wttr.in）
=====
深圳: 🌡 +23 °C
=====
        查询完成
=====
s@ubuntu:~/linux_exp$
```

我们可以看到脚本是特别厉害的工具，它通过linux指令就能完成非常多的任務。

---

## 六、实验总结

你在本实验中实际使用并理解了：

### ❖ 三种路径写法

- **绝对路径** /home/<用户名>/linux\_exp/src/hello.py
- **相对路径** src/hello.py
- **home 路径** ~/linux\_exp/src/hello.py

👉 命令是否成功，取决于：你在哪 + 你怎么写路径

---

## 七、检查清单（自检）

- 我知道 **pwd** 在干什么
- 我不会随便用 **rm -rf**
- 我能看懂脚本里每一行命令
- 我理解为什么同一个 Python 文件可以用多种方式执行

## 八、linux指令相关学习资源

- [linux-command-manual](#)
- [geeksforgeeks - Linux Commands](#)
- [The Linux command line for beginners](#)