实验1

Part1

为了熟悉基础操作，首先进行了以下操作：

1. 用 `ls` 命令查看当前目录下的文件
2. 用 `cd` 命令进入到 `/Desktop` 目录
3. 用 `mkdir` 命令在 `/Desktop` 目录下创建了一个名为 `/exercise` 的文件夹，用于存放实验文件。并再新建一个文件夹 `/e1`，存放实验1的文件。进入 `/e1` 文件夹。并再新建目录 `/paht1`

4. 用 `pwd` 命令，查看当前目录的绝对路径。
5. 用 `touch` 命令创建了一个名为 `file1` 的文件。
6. 用 `vim` 命令，进入 `file1` 文件，发现本地未安装 `vim`。
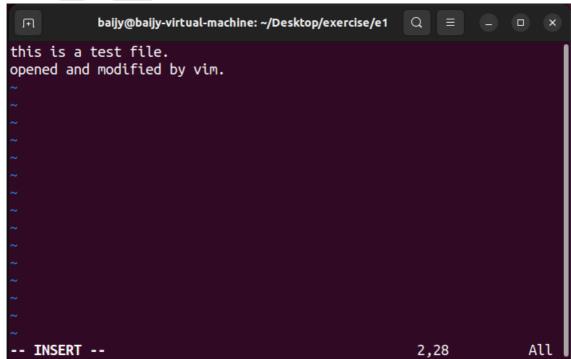7. 用 `nano` 命令，进入 `file1` 文件，编辑了一些文字。

，随后保存并退出。

8. 用 `sodu apt install vim` 命令，安装了 `vim`。

```
baijy@baijy-virtual-machine:~/Desktop/exercise/e1$ sudo apt install v
im
[sudo] password for baijy:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  vim-runtime
Suggested packages:
  ctags vim-doc vim-scripts
The following NEW packages will be installed:
  vim vim-runtime
0 upgraded, 2 newly installed, 0 to remove and 108 not upgraded.
Need to get 8,570 kB of archives.
After this operation, 37.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

，重新用 `vim` 打开 `file1` 文件，增加一行并保存退出。

```
baijy@baijy-virtual-machine: ~/Desktop/exercise/e1

this is a test file.
opened and modified by vim.
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                            2,28              All
```

9. 用 `cp` 命令，将 `file1` 文件复制到此文件并命名为 `file2`.
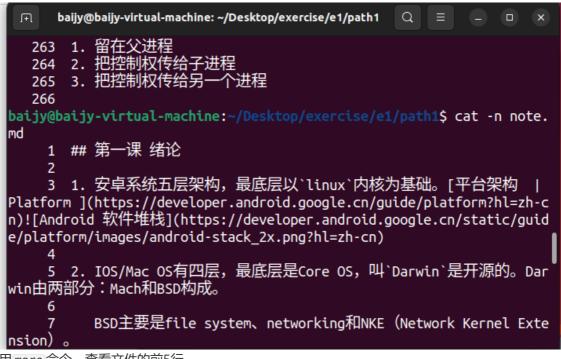
10. 用 `rm` 命令，删除 `file2` 文件。

```
baijy@baijy-virtual-machine:~/Desktop/exercise/e1$ cp file1 file2
baijy@baijy-virtual-machine:~/Desktop/exercise/e1$ ls
file1  file2
baijy@baijy-virtual-machine:~/Desktop/exercise/e1$ rm file2
baijy@baijy-virtual-machine:~/Desktop/exercise/e1$ ls
file1
```

11. 用 `mv` 命令，将 `file1` 文件移动到 `/path1` 目录下,并用 `mv` 进行重命名为 `file`。

```
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ mv ../file1
.
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ ls
file1
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ mv file1 fil
e
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ ls
file
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$
```

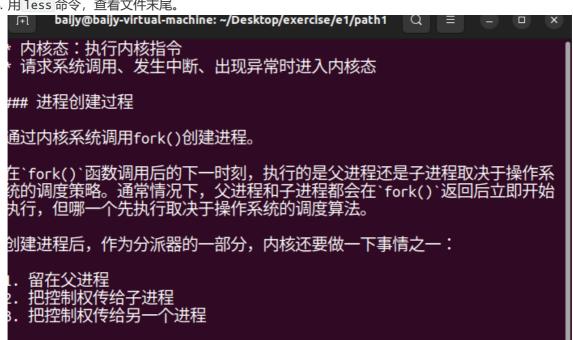为了测试文件查看命令，将课堂笔记复制到虚拟机中并打开查看。

12. 用 cat 命令直接查看整个文件，并用 -n 参数查看行号。

```
   263   1. 留在父进程
   264   2. 把控制权传给子进程
   265   3. 把控制权传给另一个进程
   266
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ cat -n note.md
     1   ## 第一课 绪论
     2
     3   1. 安卓系统五层架构，最底层以`linux`内核为基础。[平台架构   |
Platform ](https://developer.android.google.cn/guide/platform?hl=zh-cn)![Android 软件堆栈](https://developer.android.google.cn/static/guide/platform/images/android-stack_2x.png?hl=zh-cn)
     4
     5   2. IOS/Mac OS有四层，最底层是Core OS，叫`Darwin`是开源的。Darwin由两部分：Mach和BSD构成。
     6
     7       BSD主要是file system、networking和NKE（Network Kernel Extension）。
```

13. 用 more 命令，查看文件的前5行。

```
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ more -n 5 note.md
## 第一课 绪论

1. 安卓系统五层架构，最底层以`linux`内核为基础。[平台架构   | Platform ](https://developer.android.google.cn/guide/platform?hl=zh-cn)![Android 软件堆栈](https://developer.android.google.cn/static/guide/platfo
--More--(1%)
```

14. 用 less 命令，查看文件末尾。

```
* 内核态：执行内核指令
* 请求系统调用、发生中断、出现异常时进入内核态

### 进程创建过程

通过内核系统调用fork()创建进程。

在`fork()`函数调用后的下一时刻，执行的是父进程还是子进程取决于操作系统的调度策略。通常情况下，父进程和子进程都会在`fork()`返回后立即开始执行，但哪一个先执行取决于操作系统的调度算法。

创建进程后，作为分派器的一部分，内核还要做一下事情之一：

1. 留在父进程
2. 把控制权传给子进程
3. 把控制权传给另一个进程

(END)
```

15. 用 grep 命令，查找文件中的 fork 关键字。

```
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ grep fork no
te.md
通过内核系统调用fork()创建进程。
在`fork()`函数调用后的下一时刻，执行的是父进程还是子进程取决于操作系
统的调度策略。通常情况下，父进程和子进程都会在`fork()`返回后立即开始
执行，但哪一个先执行取决于操作系统的调度算法。
```

16. 用 man man 查看 man 命令的帮助文档。

```
baijy@baijy-virtual-machine: ~/Desktop/exercise/e1/path1

MAN(1)                    Manual pager utils                    MAN(1)

NAME
       man - an interface to the system reference manuals

SYNOPSIS
       man [man options] [[section] page ...] ...
       man -k [apropos options] regexp ...
       man -K [man options] [section] term ...
       man -f [whatis options] page ...
       man -l [man options] file ...
       man -w|-W [man options] page ...

DESCRIPTION
       man  is the system's manual pager.  Each page argument given
       to man is normally the name of a program, utility  or  func-
       tion.   The  manual  page associated with each of these argu-
 Manual page man(1) line 1 (press h for help or q to quit)
```

17. 用 rmdir 命令，删除目录。

```
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ mkdir new
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ ls
file  new   note.md
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ rmdir new
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ ls
file  note.md
```

18. 用 ps 命令，查看当前进程状态。

```
file   note.md
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$  ps
    PID TTY          TIME CMD
   4766 pts/0    00:00:00 bash
  18701 pts/0    00:00:00 ps
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ top
```

19. 用 `top` 命令，查看系统资源使用情况。

```
 baijy@baijy-virtual-machine: ~/Desktop/exercise/e1/path1

top - 19:07:31 up  1:03,  2 users,  load average: 0.32, 0.39, 0.63
Tasks: 292 total,   1 running, 291 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.0 us,  0.8 sy,  0.0 ni, 98.0 id,  0.0 wa,  0.0 hi,  0.2 s
MiB Mem :   3870.6 total,    639.1 free,   1545.5 used,   1686.0 buff
MiB Swap:   2140.0 total,   2140.0 free,      0.0 used.   2003.1 avai

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM
 1286 baijy     20   0 4342840 271616 134316 S   1.3   6.9
 1164 mysql     20   0 1784040 397840  35840 S   1.0  10.0
 1607 baijy     20   0  392124  12364   7296 S   0.3   0.3
 1741 baijy     20   0  146060  40356  29412 S   0.3   1.0
 5616 root      20   0       0      0      0 I   0.3   0.0
18720 baijy     20   0   16300   4224   3456 R   0.3   0.1
    1 root      20   0  166800  11900   8188 S   0.0   0.3
    2 root      20   0       0      0      0 S   0.0   0.0
    3 root       0 -20       0      0      0 I   0.0   0.0
```

20. 用 `history` 命令查看历史命令。通过加参数，可以查看最近5行。

```
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ history 5
  462  history 453
  463  history -n 10
  464  history
  465  history 20
  466  history 5
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$
```

21. 用 `!行号`，可以执行具体的历史命令。

22. 用 `df` 命令，查看磁盘使用情况。

```
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ df
Filesystem     1K-blocks      Used Available Use% Mounted on
tmpfs             396352      2204    394148   1% /run
/dev/sda3       40585548  23221044  15487252  60% /
tmpfs            1981756         0   1981756   0% /dev/shm
tmpfs               5120         4      5116   1% /run/lock
/dev/sda2         524252      6220    518032   2% /boot/efi
tmpfs             396348       116    396232   1% /run/user/1000
/dev/sr1         4919592   4919592         0 100% /media/baijy/Ubuntu
22.04.3 LTS amd64
/dev/sr0          159128    159128         0 100% /media/baijy/CDROM
```

Part2

写简单的 `helloworld` 脚本.

```bash
#!/bin/bash
echo "Hello World"
```

给予可执行权限后运行。

```
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ ./try.sh
hello world
```

随后进行了一些别的shell语句尝试：

```
#!/bin/bash
echo "hello world"
date
who
x=10
y=$[$x*2]
echo "x is $x, y=x*2,y is $y"
echo "the \$HOME is $HOME"
```

```
~
~
~
~
~
~
~
~
"try.sh" 9L, 113B                                      6,8          All
```

输出如下：

```
baijy@baijy-virtual-machine:~/Desktop/exercise/e1/path1$ ./try.sh
hello world
2024年 03月 05日 星期二 19:33:12 CST
baijy    tty2         2024-03-05 18:05 (tty2)
baijy    tty3         2024-03-05 18:07
x is 10, y=x*2,y is 20
the $HOME is /home/baijy
```

随后进行了一些别的shell语句尝试：