Ex.1

1. **cgroups**, which is abbreviated from **control groups**, is a Linux kernel feature that limits, accounts for, and isolates the resource usage (CPU, memory, disk I/O, network, etc.) of a collection of processes.
2. Differences: the resource usage of a cgroup can be configuired or customized, while that of a process can not.

   Similarities: They're hierarchical. The child inherit certain attributes from their parent.
3. **Namespaces** are a feature of the Linux kernel that partitions kernel resources such that one set of processes sees one set of resources while another set of processes sees a different set of resources so that resources can be properly distributed to different processes.

   The feature works by having the same namespace for a set of resources and processes, but those namespaces refer to distinct resources. Examples of such resources are process IDs, hostnames, user IDs, etc.

Ex.2

Q1

1. Intel Core i7-10710U
2. 16.0 GB
3. To monitor usage of CPU and RAM, I first run the `process()` function in `test.py` and then run `process_handle.py` to check the usage of CPU and RAM with the parameter `pid`. Then I run `plot.py` to plot the figure below. We can see that CPU usage is constantly around 9-10%, while the RAM usage increases in a linear manner.

   usage

Q2

First, use pandas in Python to read all files of the dataset

```python
import bz2
import glob
import pandas as pd
from tqdm import tqdm

files = glob.glob("flight_data/*.bz2")
```

Then we use `data.columns` to check what are the columns of this dataset.

1.
```python
data = pd.DataFrame(columns=["UniqueCarrier", "ArrDelay"])
for file in tqdm(files):
    one_file = pd.read_csv(file, compression='bz2', header=0,
encoding='latin-1', usecols=["UniqueCarrier","ArrDelay"])
    print(data.columns)
    data = data.append(one_file)

p1 = data[["UniqueCarrier", "ArrDelay"]]
p1 = p1.dropna()
p1 = p1[p1["ArrDelay"] > 0]
top1 = p1.groupby("UniqueCarrier").size()
top1.sort_values(ascending=False).head(1)
```

   Then we can check the output to find the most commonly late carrier, which is:

   - DL 8825137 times

2.
```python
data = pd.DataFrame(columns=['Origin','WeatherDelay'])
for file in tqdm(files):
    one_file = pd.read_csv(file, compression='bz2', header=0,
encoding='latin-1', usecols=[16,25])
    print(data.columns)
    data = data.append(one_file)

p2 = data[['Origin','WeatherDelay']]
p2 = p2.dropna()
p2 = p2[p2['WeatherDelay'] > 0]
top3 = p2.groupby('Origin').size()
top3.sort_values(ascending=False).head(3)
```

   Then we can check the output to find the 3 most commonly late origins, which are:

   - DFW 72276 times
   - ATL 58137 times
   - ORD 57754 times

3.
```
carrierlist = top1.index.to_list()
for carrier in carrierlist:
    one_carrier = p1[p1['UniqueCarrier'] == carrier]
    longest_delay = one_carrier.loc[:,'ArrDelay'].max()
    print(carrier + "'s longest delay is ")
    print(longest_delay)
    print("\n")
```

The longest delay for each carrier is:

- 9E's longest delay is 1942.0
- AA's longest delay is 1525.0
- AQ's longest delay is 1024.0
- AS's longest delay is 1139.0
- B6's longest delay is 1392.0
- CO's longest delay is 1178.0
- DH's longest delay is 1438.0
- DL's longest delay is 1189.0
- EA's longest delay is 1179.0
- EV's longest delay is 1187.0
- F9's longest delay is 920.0
- FL's longest delay is 1175.0
- HA's longest delay is 1309.0
- HP's longest delay is 1323.0
- ML (1)'s longest delay is 584.0
- MQ's longest delay is 1707.0
- NW's longest delay is 2598.0
- OH's longest delay is 1380.0
- OO's longest delay is 1435.0
- PA (1)'s longest delay is 1438.0
- PI's longest delay is 1381.0
- PS's longest delay is 1033.0
- TW's longest delay is 931.0
- TZ's longest delay is 1300.0
- UA's longest delay is 1612.0
- US's longest delay is 1073.0
- WN's longest delay is 889.0
- XE's longest delay is 939.0
- YV's longest delay is 715.0

Q3

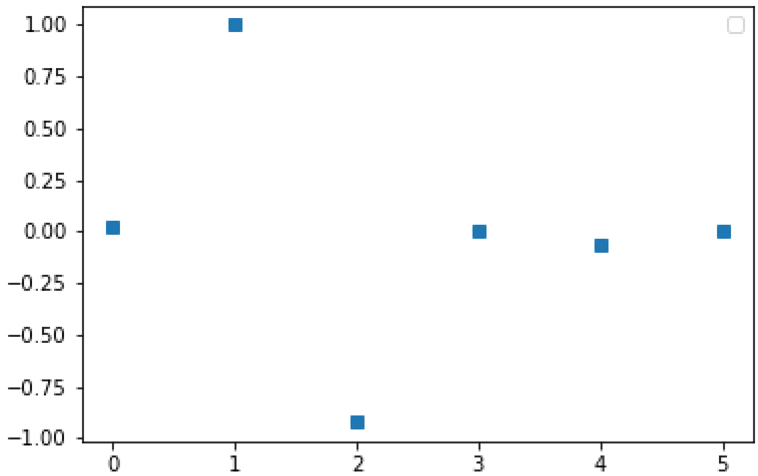
I first use a enum class to represent "UniqueCarrier", which is of type string.

```
from enum import Enum
i = 0
```

```
class Carrier(Enum):
    AA = 1
    AQ = 2
    AS = 3
    B6 = 4
    CO = 5
    DL = 6
    EV = 7
    F9 = 8
    FL = 9
    HA = 10
    MQ = 11
    NW = 12
    OH = 13
    OO = 14
    UA = 15
    US = 16
    WN = 17
    XE = 18
    YV = 19
    E9 = 20

data_2008 = pd.read_csv("./flight_data/2008.csv.bz2",
compression='bz2', header=0, encoding='latin-1')
data_2008 = data_2008.sample(140000)
data_2008 = data_2008.reset_index(drop=True)

for i in tqdm(range(140000)):
    c = data_2008.loc[i,'UniqueCarrier']
    for j in Carrier:
        if c == j.name:
            data_2008.loc[i,'enumCarrier'] = j.value
            break

# Create the dataset
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
X =
data_2008[['DayOfWeek','DepTime','CRSDepTime','ArrTime','CRSArrTime'
,'enumCarrier']].values
X = X.astype('int')
y = data_2008['DepDelay'].values
y = y.astype('int')
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=0)
```

```
# Use the linear regression model
lr = linear_model.LinearRegression().fit(X,y)
print("Training set score: {:.2f}".format(lr.score(X_train,
y_train)))
print("Test set score: {:.2f}".format(lr.score(X_test, y_test)))

# Plot the feature plot
import matplotlib.pyplot as plt
plt.plot(lr.coef_, 's')
plt.legend()
plt.show()
```

The result figure is:



Therefore, we can see that departure delay is mainly due to second features, which is **DepTime**.

Ex.3

Please see `ex3.java` for detail.