In [1]:

```
import os
import numpy as np
import pandas as pd
import csv
from csv import writer
from multiprocessing import Pool,Process
import image_feature
import array
from glob import glob
```

In [51]:

```
#taken from https://github.com/dchad/malware-detection
def asmImage(asmfiles):
    ext_drive = 'C:/Users/Anirban/Desktop/Machine Learning/chap43_microsoft_malware_detecti
    files = [i for i in asmfiles if '.asm' in i]
    print(files)
    process_id = os.getpid()
    ftot = len(files)
    image_feature_file = 'Image/' + str(process_id) + '-malware-asm.csv'
    print('Image Feature File:', image_feature_file)
    with open(image_feature_file,'w') as f:
        fw = writer(f)
        for ind,file in enumerate(files):
            filename = file.split(".")[0]
            size =  os.path.getsize(ext_drive + file)
            print(file)
            f = open(ext_drive + file,'rb')
            width = int(size ** 0.5)
            remender = int(size % width)
            image = array.array('B')
            image.fromfile(f,int(size - remender))
            image_reshp = np.reshape(image,(int(len(image)/width),width))
            image_reshp = np.resize(image_reshp,(800,))
            print(image_reshp.shape)
            fw.writerow(image_reshp)
            if (ind +1) % 10 == 0:
                print(process_id, ind + 1, 'of', ftot, 'files processed.')

    f.close()
```

In [33]:

```
def main():
#"""Function to perform multipricessing"""
    ext_drive = 'C:/Users/Anirban/Desktop/MAchine Learning/chap43_microsoft_malware_detecti
# multiprocessing using 11 cores
    tfiles = os.listdir(ext_drive)
    quart = int(len(tfiles)/11)
    train1 = tfiles[:quart]
    train2 = tfiles[quart:(2*quart)]
    train3 = tfiles[(2*quart):(3*quart)]
    train4 = tfiles[(3*quart):(4*quart)]
    train5 = tfiles[(4*quart):(5*quart)]
    train6 = tfiles[(5*quart):(6*quart)]
    train7 = tfiles[(6*quart):(7*quart)]
    train8 = tfiles[(7*quart):(8*quart)]
    train9 = tfiles[(8*quart):(9*quart)]
    train10 = tfiles[(9*quart):(10*quart)]
    train11 = tfiles[(10*quart):]
    print(len(tfiles), quart)
    trains = [train1, train2, train3, train4,train5,train6,train7,train8,train9,train10,tra
    p = Pool(11)
    p.map(image_feature.asmImage, trains)

if __name__=="__main__":
    main()
```

```
10868 988
```

In [3]:

```
col_names = ["pixel_" + str(i) for i in range(0,800)]
df_image_fetures = pd.concat([pd.read_csv(csv,names=col_names) for csv in glob('Image/' +"*
```

In [4]:

```
df_image_fetures.head()
```

Out[4]:

	pixel_0	pixel_1	pixel_2	pixel_3	pixel_4	pixel_5	pixel_6	pixel_7	pixel_8	pixel_9	...	pix
0	72	69	65	68	69	82	58	49	48	48	...	
1	46	122	101	110	99	58	48	48	52	48	...	
2	72	69	65	68	69	82	58	49	48	48	...	
3	72	69	65	68	69	82	58	48	48	52	...	
4	46	116	101	120	116	58	48	48	52	48	...	

5 rows × 800 columns

In [5]:

```
df_image_fetures.shape
```

Out[5]:

```
(10868, 800)
```

In [6]:

```
df_image_fetures.to_csv("Image_features.csv")
```

In [ ]:

```

```