

In [70]:

```
# read data : dataframe
from pandas_datareader import DataReader

df = DataReader('INTC', 'yahoo', start = '2010-01-01', end = '2020-01-01')
df
```

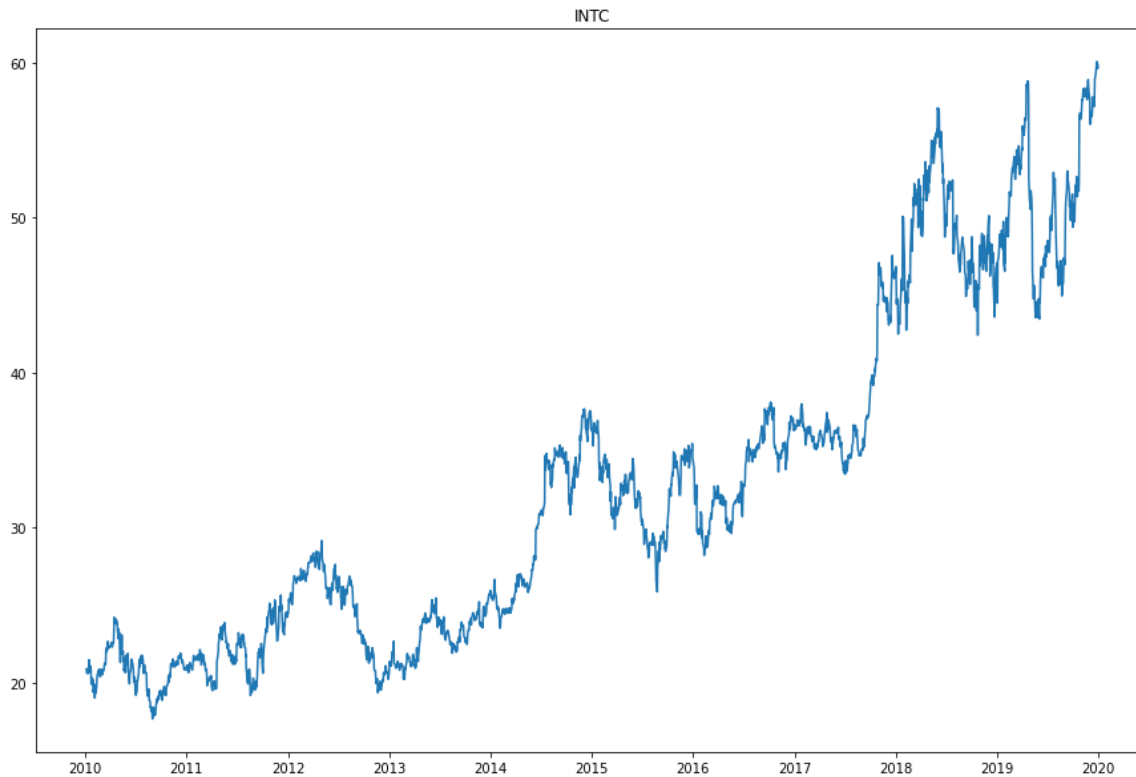
Out[70]:

	High	Low	Open	Close	Volume	Adj Close
Date						
2010-01-04	21.030001	20.730000	20.790001	20.879999	47800900.0	14.954695
2010-01-05	20.990000	20.600000	20.940001	20.870001	52357700.0	14.947535
2010-01-06	20.930000	20.670000	20.809999	20.799999	40037400.0	14.897401
2010-01-07	20.760000	20.340000	20.730000	20.600000	54041500.0	14.754156
2010-01-08	20.889999	20.400000	20.540001	20.830000	48234700.0	14.918890
...	...	...	...	...	...	...
2019-12-24	59.480000	59.160000	59.250000	59.410000	6313200.0	57.968597
2019-12-26	59.849998	59.349998	59.410000	59.820000	11480300.0	58.368649
2019-12-27	60.480000	59.900002	59.910000	60.080002	14085800.0	58.622341
2019-12-30	60.049999	59.380001	59.990002	59.619999	12750500.0	58.173500
2019-12-31	59.930000	59.360001	59.619999	59.849998	14432900.0	58.397919

2516 rows × 6 columns

In [71]:

```
from matplotlib import pyplot as plt
FIGSIZE = (15,10)
plt.figure(figsize=FIGSIZE)
plt.plot(df['Close'])
plt.title('INTC')
plt.show()
```



In [72]:

```
from sklearn.preprocessing import MinMaxScaler
import numpy as np

scaler = MinMaxScaler()
dfs = np.array(df['Close'])
dfs = scaler.fit_transform(dfs.reshape(-1, 1))
```

In [73]:

```
from keras.preprocessing.sequence import TimeseriesGenerator
dfg = TimeseriesGenerator(dfs, dfs, 60)
```

In [88]:

```
import keras as k

model = k.Sequential([
    k.layers.LSTM(128, input_shape=(dfg.length, 1)),
    k.layers.Dense(32),
    k.layers.Dense(1)
])

model.compile(optimizer='rmsProp', loss = 'MeanAbsoluteError')
```

In [89]:

```
history = model.fit(dfg, epochs = 32)
```

Epoch 1/32  
20/20 [=====] - 3s 126ms/step - loss: 0.  
0963

Epoch 2/32  
20/20 [=====] - 2s 125ms/step - loss: 0.  
0422

Epoch 3/32  
20/20 [=====] - 2s 125ms/step - loss: 0.  
0330

Epoch 4/32  
20/20 [=====] - 3s 126ms/step - loss: 0.  
0412

Epoch 5/32  
20/20 [=====] - 2s 125ms/step - loss: 0.  
0341

Epoch 6/32  
20/20 [=====] - 3s 125ms/step - loss: 0.  
0370

Epoch 7/32  
20/20 [=====] - 3s 125ms/step - loss: 0.  
0290

Epoch 8/32  
20/20 [=====] - 2s 124ms/step - loss: 0.  
0299

Epoch 9/32  
20/20 [=====] - 3s 125ms/step - loss: 0.  
0340

Epoch 10/32  
20/20 [=====] - 3s 125ms/step - loss: 0.  
0301

Epoch 11/32  
20/20 [=====] - 3s 126ms/step - loss: 0.  
0283

Epoch 12/32  
20/20 [=====] - 3s 125ms/step - loss: 0.  
0288

Epoch 13/32  
20/20 [=====] - 2s 125ms/step - loss: 0.  
0287

Epoch 14/32  
20/20 [=====] - 3s 127ms/step - loss: 0.  
0252

Epoch 15/32  
20/20 [=====] - 3s 126ms/step - loss: 0.  
0280

Epoch 16/32  
20/20 [=====] - 2s 125ms/step - loss: 0.  
0264

Epoch 17/32

20/20 [=====] - 3s 126ms/step - loss: 0.  
0237  
Epoch 18/32  
20/20 [=====] - 3s 128ms/step - loss: 0.  
0246  
Epoch 19/32  
20/20 [=====] - 3s 126ms/step - loss: 0.  
0244  
Epoch 20/32  
20/20 [=====] - 3s 126ms/step - loss: 0.  
0265  
Epoch 21/32  
20/20 [=====] - 3s 127ms/step - loss: 0.  
0192  
Epoch 22/32  
20/20 [=====] - 3s 126ms/step - loss: 0.  
0269  
Epoch 23/32  
20/20 [=====] - 3s 126ms/step - loss: 0.  
0209  
Epoch 24/32  
20/20 [=====] - 2s 125ms/step - loss: 0.  
0240  
Epoch 25/32  
20/20 [=====] - 3s 126ms/step - loss: 0.  
0238  
Epoch 26/32  
20/20 [=====] - 3s 127ms/step - loss: 0.  
0223  
Epoch 27/32  
20/20 [=====] - 3s 126ms/step - loss: 0.  
0239  
Epoch 28/32  
20/20 [=====] - 3s 126ms/step - loss: 0.  
0221  
Epoch 29/32  
20/20 [=====] - 3s 126ms/step - loss: 0.  
0211  
Epoch 30/32  
20/20 [=====] - 3s 127ms/step - loss: 0.  
0235  
Epoch 31/32  
20/20 [=====] - 3s 125ms/step - loss: 0.  
0228  
Epoch 32/32  
20/20 [=====] - 3s 125ms/step - loss: 0.  
0217

In [ ]:

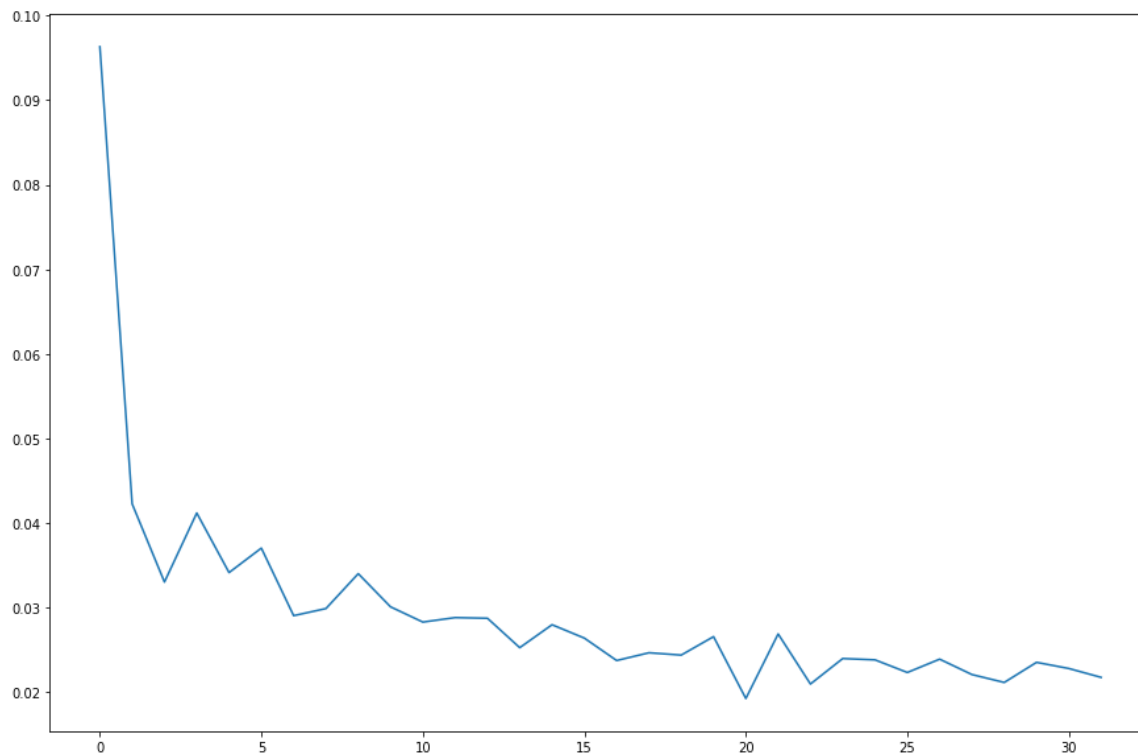
```
model.save('stock.h5')
```

In [90]:

```
plt.figure(figsize=FIGSIZE)
plt.plot(history.history['loss'])
```

Out[90]:

[<matplotlib.lines.Line2D at 0x7f613b6ebb38>]



In [91]:

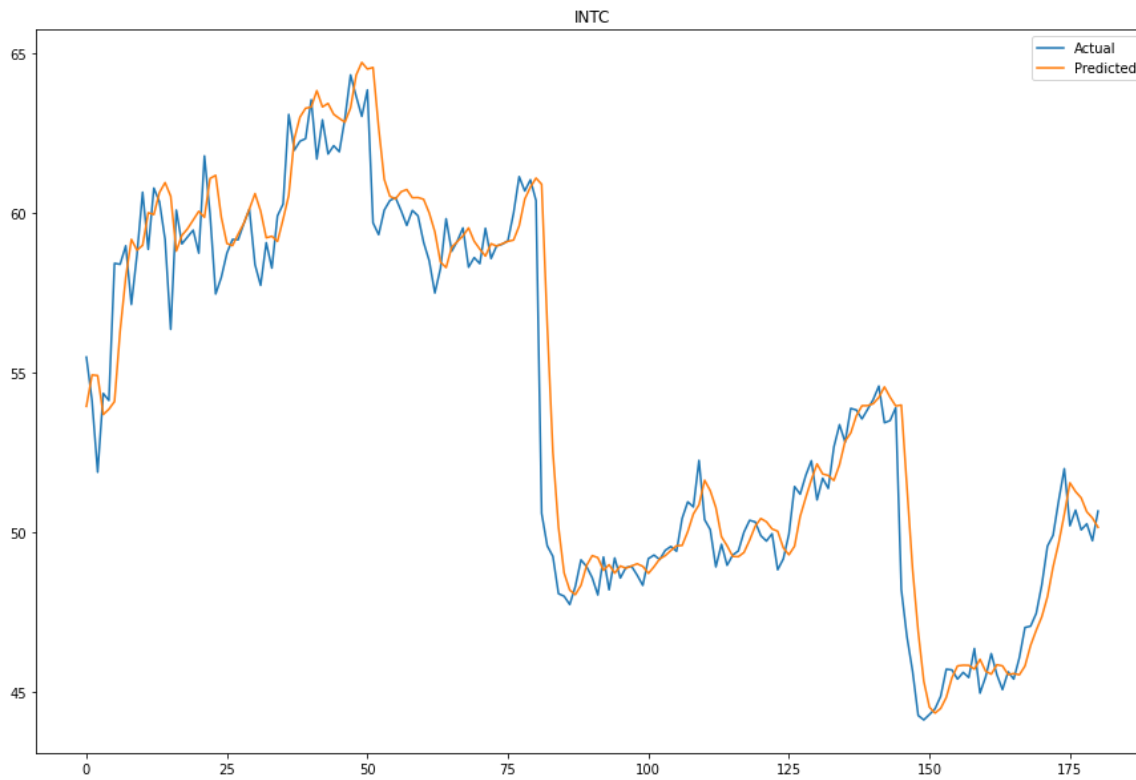
```
df = DataReader('INTC', 'yahoo', '2020-01-01')
dfx, dfy = [], []
for i in range((len(df)-60)):
    X = df['Close'][i:i+60]
    Y = df['Close'][i+60]
    X = np.array(X).reshape(-1, 1)
    X = scaler.fit_transform(X)
    X = np.expand_dims(X, axis = 0)
    res = model.predict(X)
    res = scaler.inverse_transform(res)
    dfx.append(res[0][0])
    dfy.append(Y)
```

In [92]:

```
plt.figure(figsize=FIGSIZE)
plt.plot(np.array(dfy))
plt.title('INTC')
plt.plot(np.array(dfx))
plt.legend(['Actual', 'Predicted'])
```

Out[92]:

<matplotlib.legend.Legend at 0x7f613b962d30>



In [ ]:

```
import keras as k
```

In [ ]:

```
md = k.models.load_model('stock.h5')
```



In [ ]:

```
X = df['Close'][:60]
Y = df['Close'][60]

X = np.array(X).reshape(-1, 1)
X = scaler.fit_transform(X)
X = np.expand_dims(X, axis = 0)
res = model.predict(X)
res = scaler.inverse_transform(res)
res, Y
```

WARNING:tensorflow:Model was constructed with shape (None, 64, 1) for input Tensor("lstm\_input:0", shape=(None, 64, 1), dtype=float32), but it was called on an input with incompatible shape (None, 60, 1).

Out[ ]:

```
(array([[53.022606]], dtype=float32), 55.4900016784668)
```

In [ ]:

```
md.get_weights()[0][0][:5]
```

Out[ ]:

```
array([ 0.0325065 ,  0.02529051, -0.05658942,  0.08502866, -0.100
79148],
      dtype=float32)
```

In [ ]:

```
model.get_weights()[0][0][:5]
```

Out[ ]:

```
array([ 0.0325065 ,  0.02529051, -0.05658942,  0.08502866, -0.100
79148],
      dtype=float32)
```