In [1]:
```
import numpy as np #imported numpy library
```

In [2]:
```
import pandas as pd # imported pandas library
```

In [3]:
```
import seaborn as sns # imported seaborn library
```

In [4]:
```
import matplotlib.pyplot as plt # imported matplot library
%matplotlib inline # inline function will allow us to see the approaching jupyter
```

In [6]:
```
#to load the data set from excelfile into jupyter environment
dataset = pd.read_excel("C:/Users/subed/OneDrive/Desktop/garments_worker_productiv
```

In [7]:
```
dataset # calling dataset.it shows first five and last five rows with all columns
# dimention of dataset = 1197 rows × 15 columns
```

Out[7]:

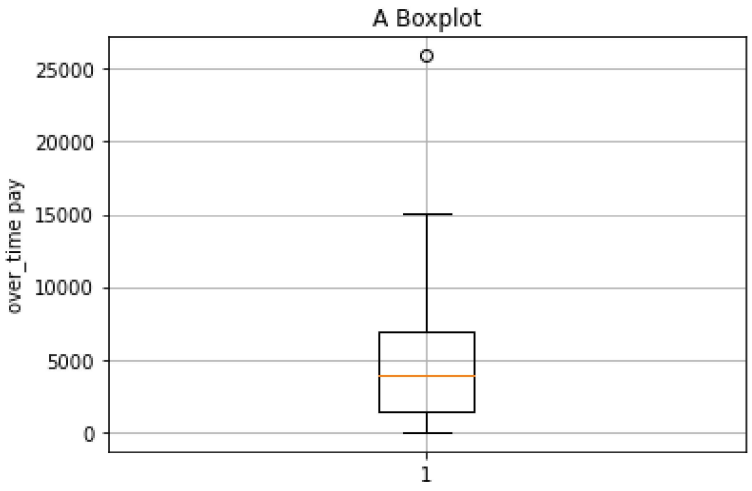
	date	quarter	department	day	team	targeted_productivity	smv	wip	over_time
0	2015-01-01	Quarter1	sweing	Thursday	8	0.80	26.16	1108.0	7080
1	2015-01-01	Quarter1	finishing	Thursday	1	0.75	3.94	NaN	960
2	2015-01-01	Quarter1	sweing	Thursday	11	0.80	11.41	968.0	3660
3	2015-01-01	Quarter1	sweing	Thursday	12	0.80	11.41	968.0	3660
4	2015-01-01	Quarter1	sweing	Thursday	6	0.80	25.90	1170.0	1920
...	...	...	...	...	...	...	...	...	...
1192	2015-03-11	Quarter2	finishing	Wednesday	10	0.75	2.90	NaN	960
1193	2015-03-11	Quarter2	finishing	Wednesday	8	0.70	3.90	NaN	960
1194	2015-03-11	Quarter2	finishing	Wednesday	7	0.65	3.90	NaN	960
1195	2015-03-11	Quarter2	finishing	Wednesday	9	0.75	2.90	NaN	1800
1196	2015-03-11	Quarter2	finishing	Wednesday	6	0.70	2.90	NaN	720

1197 rows × 15 columns

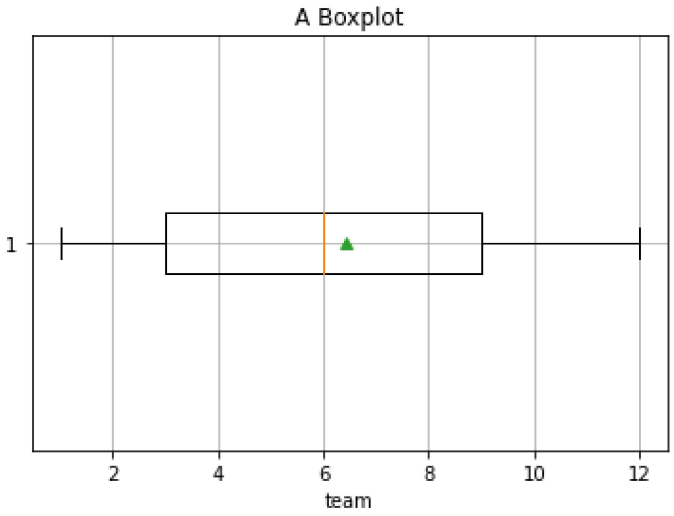
In [8]:
```
dataset.columns # this displays all the columns name in console
#smv = standard minute value
#wip= work in progress
# over_time = amount of overtime done by each team in minute
# targeted_productivity: is productivity set by authority for each team for each d
# actual_productivity : is the productivity delivered by workers in % ( from 0 to
```

Out[8]:
```
Index(['date', 'quarter', 'department', 'day', 'team', 'targeted_productivity',
       'smv', 'wip', 'over_time', 'incentive', 'idle_time', 'idle_men',
       'no_of_style_change', 'no_of_workers', 'actual_productivity'],
      dtype='object')
```

In [14]:
```
#plotting box_plot
''' Introduction of box_plot
A box plot is a graphical display of the distribution of data.
It helps to interpret data on the basis of 5 key numbers.
They are minimum, first quartile, median, third quartile and maximum with outliers
IQR: distance between first and third quartile.
Whiskers: lines beyonds the IQR ( = 1.5* IQR on either side)
Outliners: numbers beyonds the minimum and maximum points.
'''
# what inference can we get from box plot/ Applications
# About the median value of data
# About the distribution of data (either unifrom or skewed)/Symmetry of data
# About the degree of dispersion of data
# outliners values
# data skewness

plt.boxplot(dataset.over_time)    # Creating plot
plt.ylabel('over_time pay')
plt.title('A Boxplot')
plt.grid(True)
plt.show()   # show plot
# By defalult it is single horizontal box plot.
```

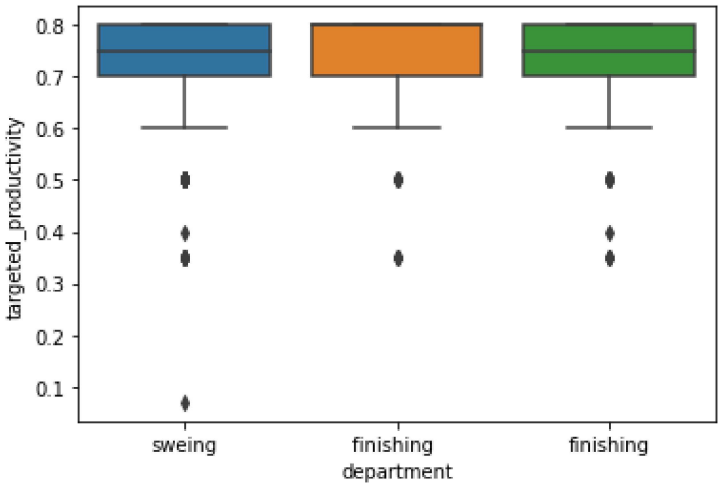


In [15]:
```
plt.boxplot(dataset['team'],vert = False,showmeans=True,) # showing single horizon
plt.xlabel('team')
plt.title('A Boxplot')
plt.grid(True)
plt.show()
# We can conclude from this data that The  members are uniformly distributed in ea
# the team members are (from 1 to 12)
```



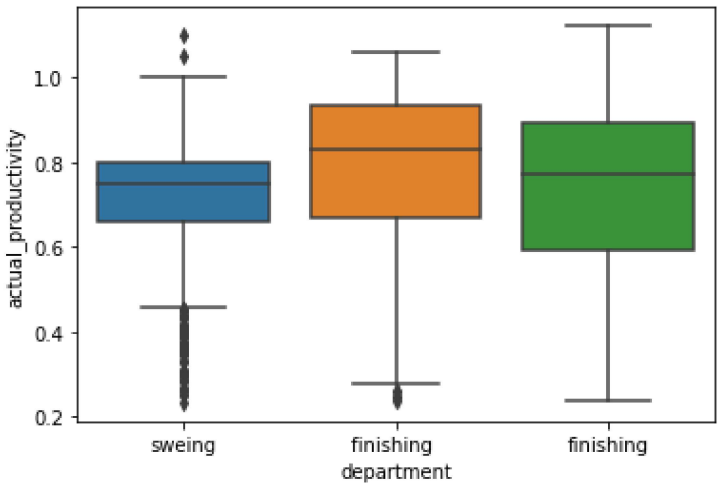
In [33]:
```
sns.boxplot(x='department',y='targeted_productivity',data = dataset)

'''we got three department, where the targeted productivity is almost equal to 0.8
with minimum value of 0.6'''
```

Out[33]:
```
'we got three department, where the targeted productivity is almost equal to 0.8
(around 80% )\nwith minimum value of 0.6'
```



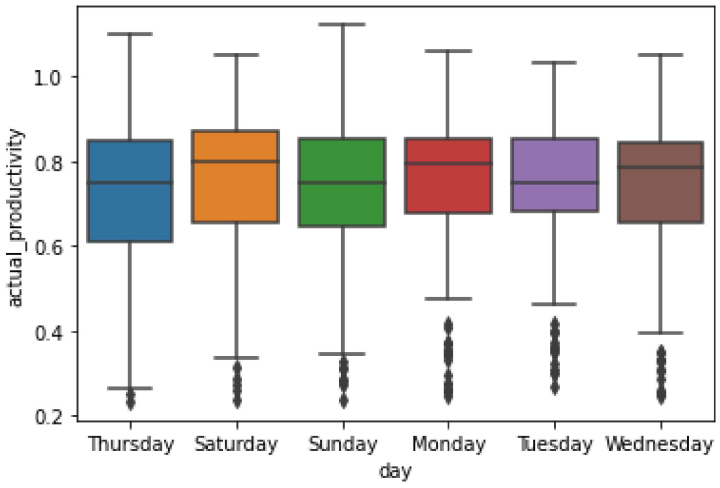
In [34]:
```
sns.boxplot(x='department', y='actual_productivity',data = dataset)
'''But what we got is the actual productivity is better than the expected one. tha
higher efficiency in result.
The finishing department has highest actual_productivity of around
'''
# By compairing the data result of we can say that the workers and working environ
```

Out[34]: <AxesSubplot:xlabel='department', ylabel='actual_productivity'>



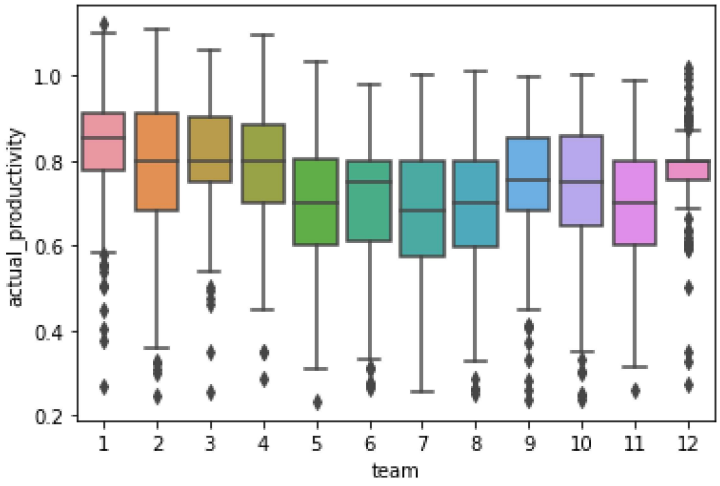
In [35]:
```
sns.boxplot(x='day', y='actual_productivity',data = dataset)
# we see that the productivity is highest in saturday than other days.
```

Out[35]: <AxesSubplot:xlabel='day', ylabel='actual_productivity'>



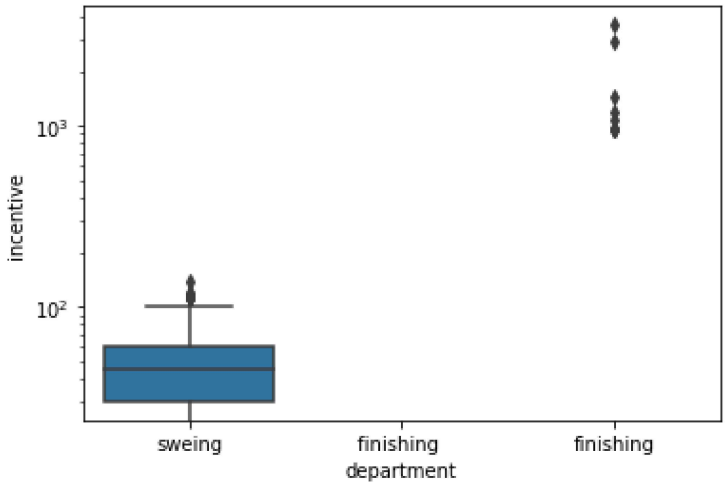
In [37]:
```
sns.boxplot(x='team', y='actual_productivity',data = dataset,)
# we can conclude that the team1 is showing more productive work and team 7 is sho
```

Out[37]: <AxesSubplot:xlabel='team', ylabel='actual_productivity'>



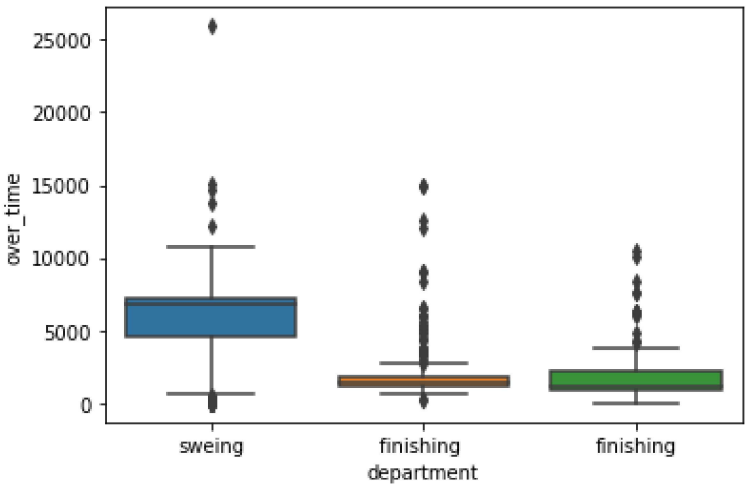
In [43]:
```
sns.boxplot(y='incentive', x='department',data = dataset)
# We see that only sweing department got incentive though the actual_productivity
```

Out[43]: []



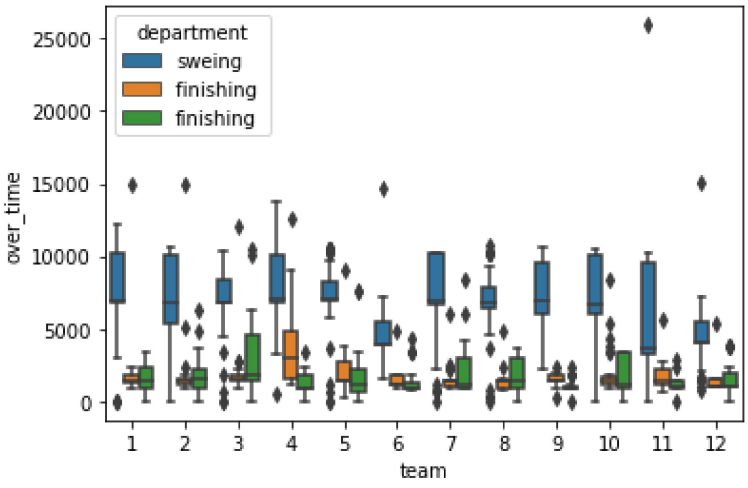
In [44]:
```
sns.boxplot(x='department', y='over_time',data = dataset)
# We see the overtime pay is also highest in sewng department.
```

Out[44]: <AxesSubplot:xlabel='department', ylabel='over_time'>



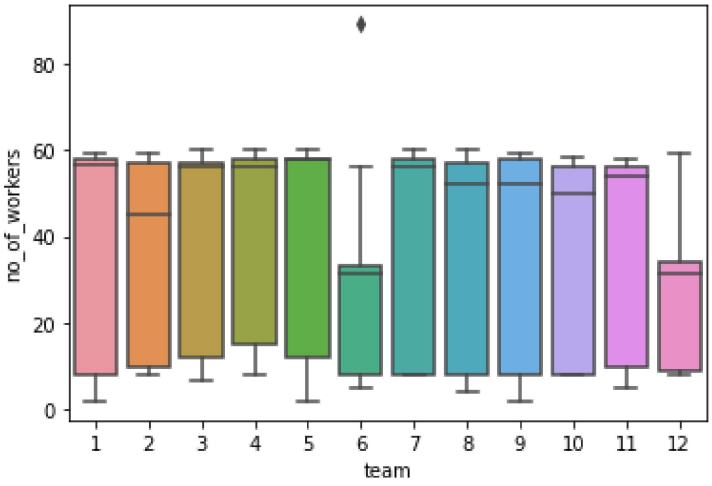
In [46]:
```
sns.boxplot(x='team', y='over_time',data = dataset, hue = 'department')
# This implies overtime is higest paid in dep 1,4,7,9 of sweing department.
```

Out[46]: <AxesSubplot:xlabel='team', ylabel='over_time'>



In [52]:
```
sns.boxplot(x='team', y='no_of_workers',data = dataset)
# team 6  and 12 has less numbers of workers in comparision of others.
```

Out[52]: <AxesSubplot:xlabel='team', ylabel='no_of_workers'>



In [ ]:
```

```