



Generated Plot

The above plot is generated by the following python script:

```
1 import matplotlib.pyplot as plt
2 import os
3 import numpy as np
4
5 #run for 10 times
6 times = 10
7
8 #values of N
9 NValues = [10, 50, 100]
10
11 #input filename
12 inputFile = "buffer.txt"
13
14 #output filename
15 outputFile = "output.txt"
16
17 #compile the file at the beginning
18 a = os.system("g++ a3_6.cpp")
19
20 Vals = []
```

```

21
22 for q in NValues:
23     #opening the input file
24     fout = open(inputFile, "w")
25
26     #writing value of N to it
27     fout.write(str(q)+"\n")
28
29     #closing the file
30     fout.close()
31
32     #defining Y
33     Y = [[],[],[],[],[ ]]
34
35     for i in range(times):
36         #running the executable and getting output in output.txt
37         a = os.system("./a.out < "+ inputFile + " > " + outputFile)
38
39         #reading from output.txt
40         fin = open(outputFile, "r")
41
42         #list of strings from fin
43         L = list(filter(None, fin.read().split('\n')))
44
45         for i in range(5):
46             L[i] = float(L[i])
47             Y[i].append(L[i])
48

```

```

49     for i in range(5):
50         Y[i] = sum(Y[i])/len(Y[i])
51
52     Vals.append(np.array(Y))
53 Vals = np.array(Vals)
54 Vals = np.transpose(Vals)
55
56 # data to plot
57 n_groups = 3
58 means = Vals
59
60 # create plot
61 plt.figure(num=None, figsize=(12, 9), dpi=80, facecolor='w', edgecolor='k')
62 index = np.arange(n_groups)
63 bar_width = 0.15
64 opacity = 0.9
65
66 labels=['Non-preemptive FCFS',
67         'Non-preemptive SJF',
68         'Pre-emptive SJF',
69         'Round Robin (δ = 2)',
70         'Highest response-ratio']
71 colors=['#A0569E', '#FFC99B', '#3D4CAF', '#49D89A', '#E87F7F']
72

```

```
73 for i in range(5):
74     plt.bar(index + bar_width*i, means[i], bar_width, alpha = opacity, color=colors[i],label=labels[i])
75
76 plt.xlabel('Value of N', fontsize=14)
77 plt.ylabel('Average Turnaround Times',fontsize=14)
78 plt.title('Comparison of different scheduling techniques',fontsize=14)
79 plt.xticks(index + 2*bar_width, ('N = 10', 'N = 50', 'N = 100'), fontsize=12)
80 temp = int(max(means.flatten()))+50
81 plt.yticks(ticks=range(0,temp,50),fontsize=12)
82 plt.legend(prop={'size': 15})
83
84 plt.tight_layout()
85 plt.savefig("barChart.png")
```