207SE Lab 1 – Introduction to operating systems

# Your task

1. In **one paragraph** describewhat you feel the future of operating systems will be. You might include diagrams to support your predictions and describe how the future operating systems will differ from the current one.

Operating system has been the tool to manage and exploit computing resources. The current generation of operating systems has all been about translating interaction to a smaller devices and touch screens such as smart phones, fridges. Operating systems help those devices to translate human commands into machine code to understand what the human is trying to make the machine do. This has been possible with the technology advances in the past 10 years and it is really booming. Money is being spent on smarter operating systems such as AI which is the future of operating systems I believe. I believe that in the future we do not have to physically do anything to command a machine to do something, we just speak to them and they will be able to understand what we want them to do.

.

# Evidence

The paragraph the future of operating systems, and commented source code with output examples for **either** activity b1 or b2.

## *Activity b1*

W rite a sim ple program (or adapt the C++ program in Appendix A) to parse a string to determ ine if it fulfils **one of the following gramm ar order rules**. Below are rules related to the basic gramm ar. For exam ple in the g ramm ar, an <agent word> is followed b y an <action word> and the <action word> is follow by a <object word>.

1. #vocabularies
2. agent=["bot", "mike"]
3. direction=["forward", "backward", "left", "right"]  #saves words as there types
4. objec=["nut", "plum", "cat", "cup"]
5. action=["pick", "put", "lift", "drop", "go"]
6. pronoun=["i", "you", "we"]
7. colour=["red", "blue"]
9. sentance= input("Input sentance structure: ")   #take input from user
10. sentance = sentance.lower() #takes the input and makes everything lowercase
11. words = sentance.split()    #splits string
13. #Grammer Rules
14. correct1=["age", "act", "obj"]
15. correct2=["age", "act", "col", "obj"]
16. correct3=["age", "act", "dir"]
17. correct4=["pro", "act", "obj"]
18. correct5=["pro", "act", "dir"]
20. Nlist=[]            #create a **new** list
22. **for** i in range(len(words)): #**for** loop to go through each word in input string
23. **if** words[i] in agent:   #match a word with its type
24. Nlist.append("age") #append age (meaning agent) to the **new** list
25. **continue**
26. elif words[i] in direction:
27. Nlist.append("dir")
28. **continue**
29. elif words[i] in objec:
30. Nlist.append("obj")
31. **continue**
32. elif words[i] in action:
33. Nlist.append("act")
34. **continue**
35. elif words[i] in pronoun:
36. Nlist.append("pro")
37. **continue**
38. elif words[i] in colour:
39. Nlist.append("col")
40. **continue**        #output will be a list like ["age", "act", "obj"]

43. **if** Nlist == correct1:       #match Nlist with a grammer rule, **if** it matches then print correct grammer, **if** not then print Incorrect grammer
44. print (" Correct Grammer! ")
45. elif Nlist == correct2:
46. print (" Correct Grammer! ")
47. elif Nlist == correct3:
48. print (" Correct Grammer! ")
49. elif Nlist == correct4:
50. print (" Correct Grammer! ")
51. elif Nlist == correct5:
52. print (" Correct Grammer! ")
53. **else**:
54. print ("Incorrect Grammer!")

|  |  |
| --- | --- |
| **Examples of grammatically correct and incorrect inputs** |  |
| bot pick plum  W e drop cup |  |
| I go nut  bot lift left |  |
| bot pick plum plum  drop cup |  |
| nut go left  left left left |  |

