



SUBJECT: **ADVANCED ROBOTICS**

SEMESTER: OT23

ASSIGNMENT: **Activity 1.2**

WORK FORMAT: **Teams**

DOCUMENT FORMAT: **Digital - Blackboard**

MADE BY: M.Sc. GILBERTO ALFONSO MONTES RAMIREZ

AUGUST 2023

Instructions I. Report a PDF document with a front cover + the pages with contents of your work. Include the honor code in the cover and consider using IEEE format for references.

1. Managing Processes in Linux

- a. Go to slide 18 in the presentation file C1.4 Introduction to Linux. It is about Managing Processes with Linux terminal commands.
- b. Report the implementation of three out of the whole set of commands in that slide (minimum). Use Cocalc or your local Linux Terminal to show the result of each command you chose in a clever way, *e.g. short-description of what you are doing + screenshot of executing the command and its result.*

2. Use Piping with Python

- a. Code a python script to generate an array of five integers
- b. Code the Bubble sort algorithm in a python script
- c. Search how to use **Linux-piping strategy** to feed the array of integers into the Bubble sort algorithm
 - i. **You should be doing something similar to one of the commands we studied in the last class:**

echo 5 | python3 scaleNumber.py

3. Conclusions & Appendix

- a. In a **short** paragraph, describe *how these commands can help you up when programming algorithms for robotic or embedded systems?*
- b. In a **short** paragraph, describe *how different it is running python scripts directly in the programming environment you use from running those .py files with linux-piping?*

4. Always use References

- Use IEEE format

5. Report a document to Blackboard

- Use the link of this activity to report your document.
- NOTE: To report in Blackboard, convert your document to PDF format**

=====

1. MANAGING PROCESSES IN LINUX SHELL

The very first point was a quite simple task. Here three examples

Upon calling terminal, I used ps

Current executing processes

Lunch ping process

Lunch sleep process

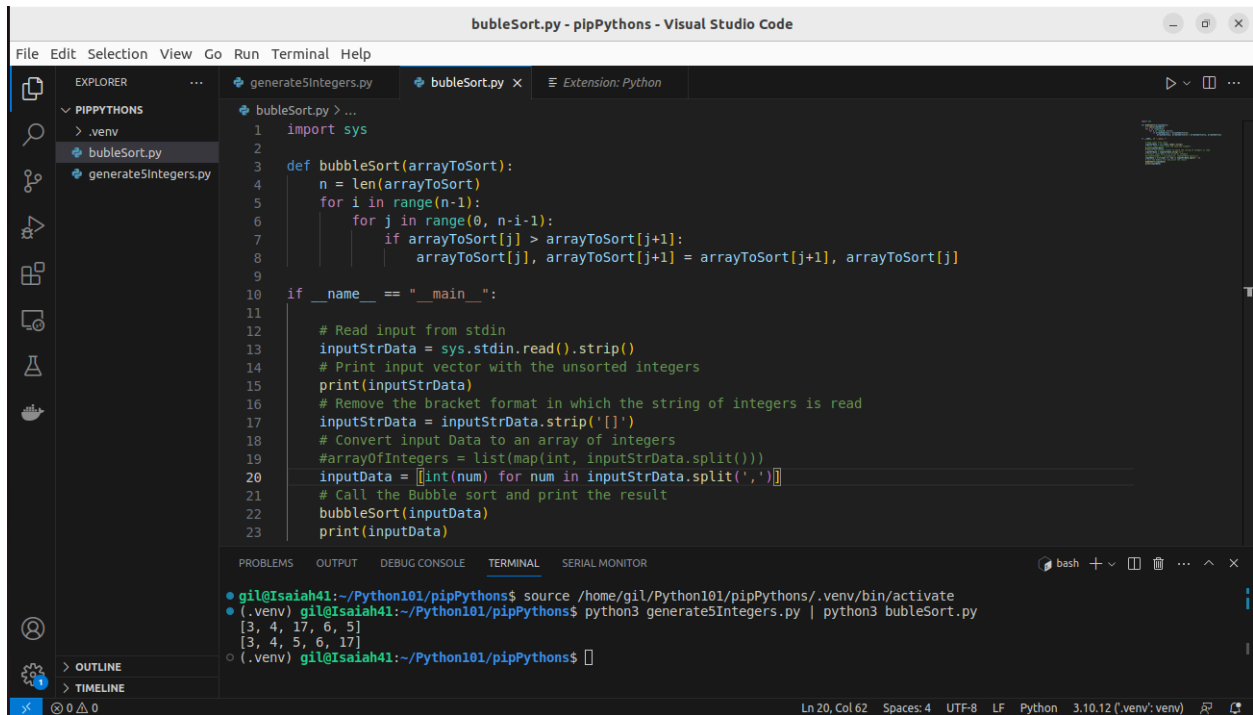
List processes again

Call a stopped job back ON.
Check how the number 1 is
related to the previous
terminal line [1]+ Stopped

jobs will list the jobs
currently running or stopped

```
gil@Isaiah41: ~  
gil@Isaiah41:~$ ps  
  PID TTY          TIME CMD  
 176894 pts/1    00:00:00 bash  
 176900 pts/1    00:00:00 ps  
gil@Isaiah41:~$ ping google.com  
PING google.com(qro01s28-in-x0e.1e100.net (2607:f8b0:4012:814::200e)) 56 data bytes  
64 bytes from qro01s28-in-x0e.1e100.net (2607:f8b0:4012:814::200e): icmp_seq=1 ttl=115 time=31.8 ms  
64 bytes from qro01s28-in-x0e.1e100.net (2607:f8b0:4012:814::200e): icmp_seq=2 ttl=115 time=34.9 ms  
^Z  
[1]+  Stopped                  ping google.com  
gil@Isaiah41:~$ sleep 500  
^Z  
[2]+  Stopped                  sleep 500  
gil@Isaiah41:~$ ps  
  PID TTY          TIME CMD  
 176894 pts/1    00:00:00 bash  
 176910 pts/1    00:00:00 ping  
 176914 pts/1    00:00:00 sleep  
 176918 pts/1    00:00:00 ps  
gil@Isaiah41:~$ fg %1  
ping google.com  
64 bytes from qro01s28-in-x0e.1e100.net (2607:f8b0:4012:814::200e): icmp_seq=3 ttl=115 time=31.3 ms  
64 bytes from qro01s28-in-x0e.1e100.net (2607:f8b0:4012:814::200e): icmp_seq=4 ttl=115 time=35.9 ms  
64 bytes from qro01s28-in-x0e.1e100.net (2607:f8b0:4012:814::200e): icmp_seq=5 ttl=115 time=33.6 ms  
^Z  
[1]+  Stopped                  ping google.com  
gil@Isaiah41:~$ fg %2  
sleep 500  
^Z  
[2]+  Stopped                  sleep 500  
gil@Isaiah41:~$ jobs  
[1]-  Stopped                  ping google.com  
[2]+  Stopped                  sleep 500  
gil@Isaiah41:~$
```

2. Piping with two Python files



The screenshot shows the Visual Studio Code interface with a file named `bubbleSort.py` open. The code implements a bubble sort algorithm. The terminal at the bottom shows the execution of the program using piping to connect the output of `generate5Integers.py` to the input of `bubbleSort.py`.

```
1 import sys
2
3 def bubbleSort(arrayToSort):
4     n = len(arrayToSort)
5     for i in range(n-1):
6         for j in range(0, n-i-1):
7             if arrayToSort[j] > arrayToSort[j+1]:
8                 arrayToSort[j], arrayToSort[j+1] = arrayToSort[j+1], arrayToSort[j]
9
10 if __name__ == "__main__":
11
12     # Read input from stdin
13     inputStrData = sys.stdin.read().strip()
14     # Print input vector with the unsorted integers
15     print(inputStrData)
16     # Remove the bracket format in which the string of integers is read
17     inputStrData = inputStrData.strip('[]')
18     # Convert input Data to an array of integers
19     #arrayOfIntegers = list(map(int, inputStrData.split()))
20     inputData = [int(num) for num in inputStrData.split(',')]
21     # Call the Bubble sort and print the result
22     bubbleSort(inputData)
23     print(inputData)
```

```
gileIsaiah41:~/Python101/pipPythons$ source /home/gil/Python101/pipPythons/.venv/bin/activate
(.venv) gileIsaiah41:~/Python101/pipPythons$ python3 generate5Integers.py | python3 bubbleSort.py
[3, 4, 17, 6, 5]
[3, 4, 5, 6, 17]
(.venv) gileIsaiah41:~/Python101/pipPythons$
```

You can check and run the python files used in this activity. They are available in Blackboard.