FACULTY OF ENGINEERING AND APPLIED SCIENCE

SOFE 3950U Operating Systems

LABORATORY REPORT

Professor: **Amin Avan**

Experiment:

LAB REPORT #: **4**

LAB CRN NUMBER and group #: CRNT: **74025** | Group: **#6**

LAB GROUP MEMBERS				
#	Surname	Name	ID	Signature
1	**Radadiya**	**Rohan**	**100704614**	**R.R**
2	**Sibi**	**Sabesan**	**100750081**	**S.S.**
3	**Hussain**	**Syed Airaj**	**100789134**	**A.H**

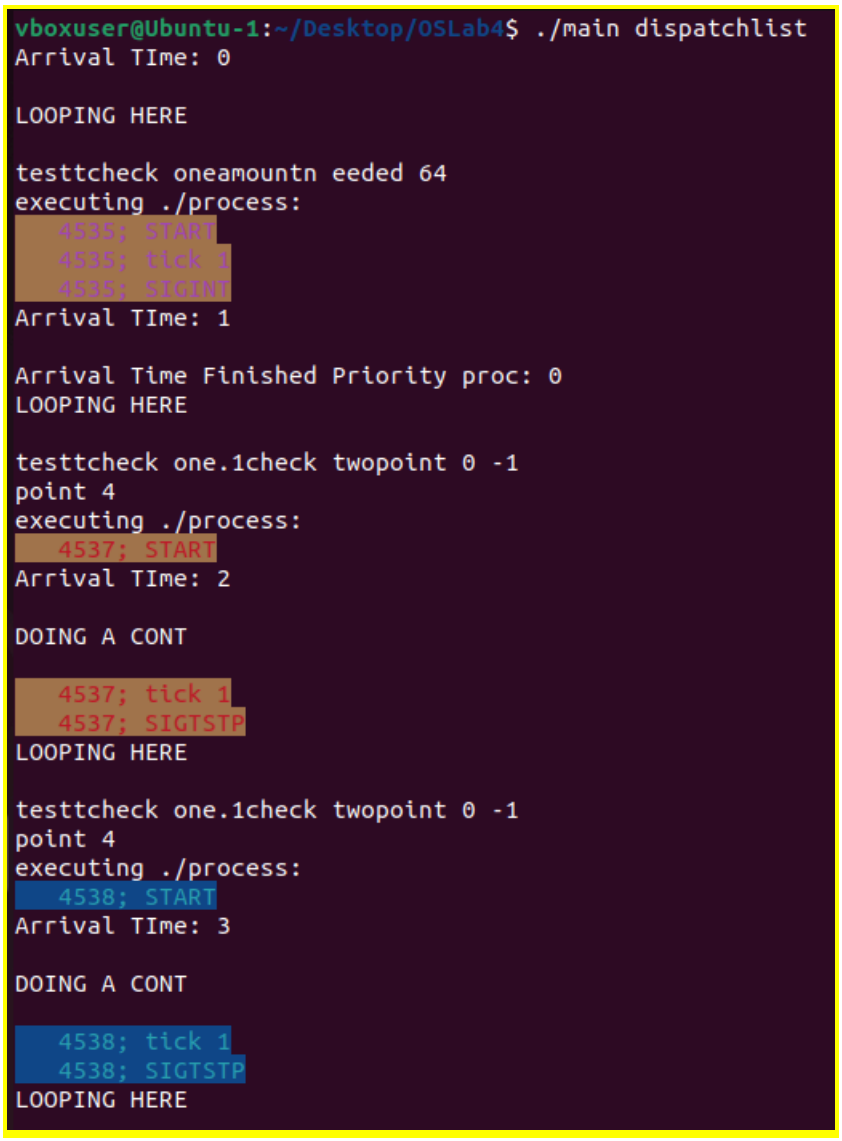
GITHUB LINK: https://github.com/rohanradadiya/OSLAB4

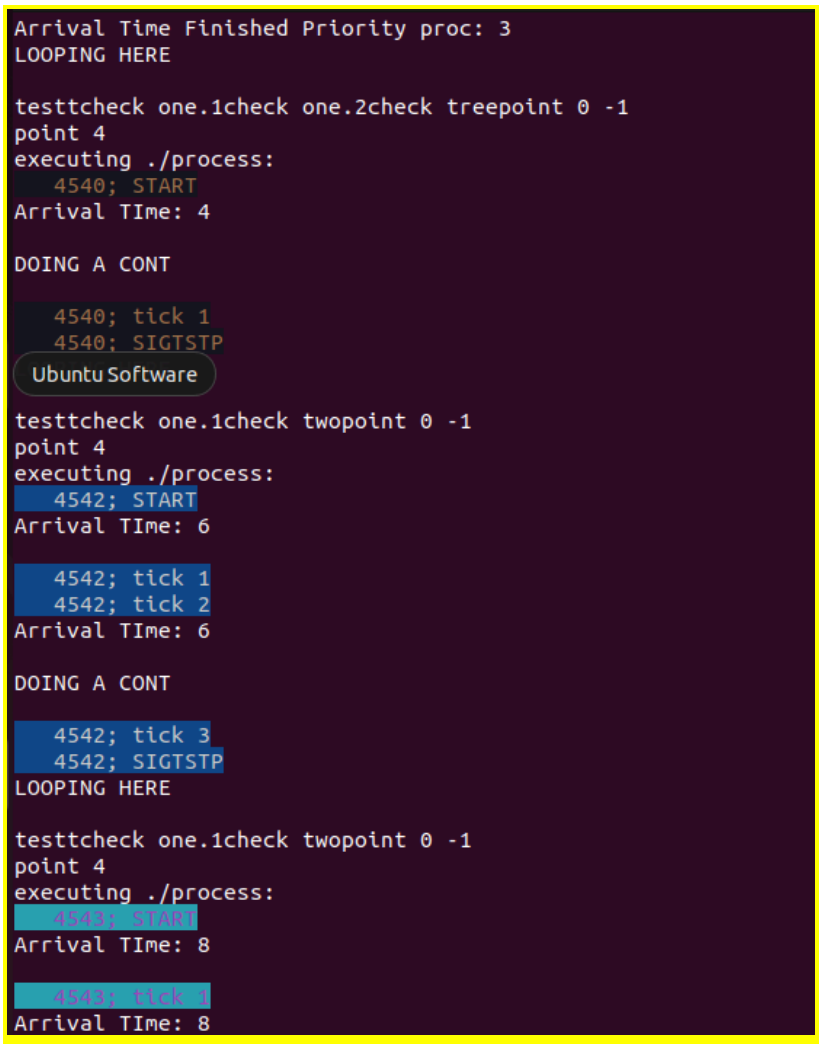
Introduction:

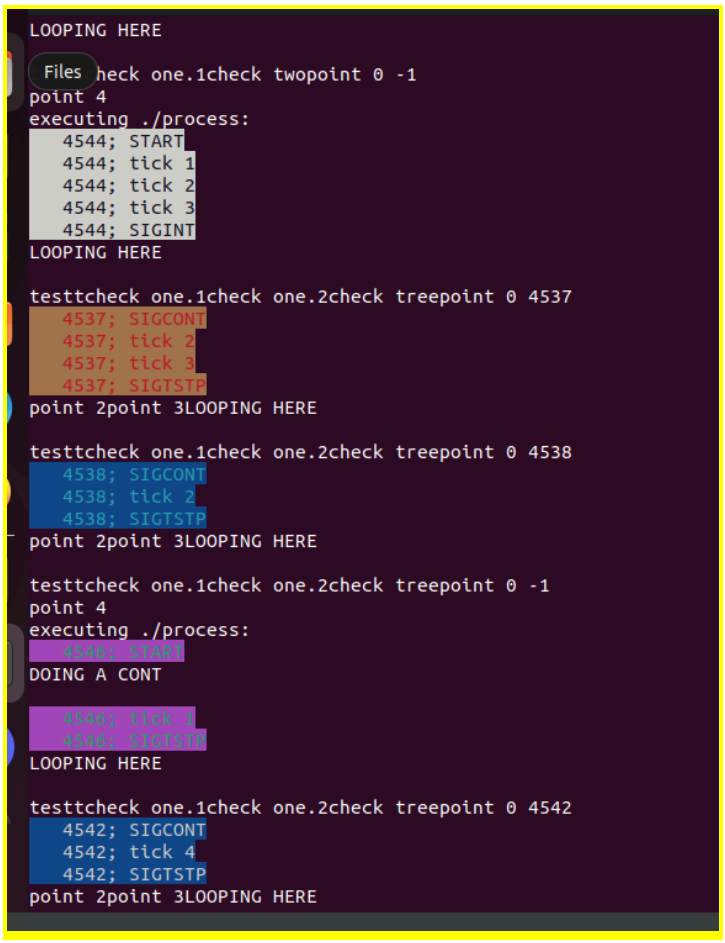
To start off, the laboratory “Scheduling” is very helpful in understanding the functions of a HOST dispatcher. The HOST is an acronym for “Hypothetical Operating System Testbed and serves the core purpose of acting as a crucial laboratory environment for evaluating not only the performance, but also the effectiveness of process scheduling algorithms within a multiprogramming system. The processes are handled on a FCFO (First Come First Serve) basis, which helps ensure the completion of processes. It was also understood that resource management is a critical functionality of HOST, and overall, in terms of processes, the HOST serves as submission, resource allocation, termination, and execution.

Overall, the lab was very helpful in understanding these concepts and being able to implement them in a practical project.

Screenshots:




Arrival Time Finished Priority proc: 3
LOOPING HERE
testtcheck one.1check one.2check treepoint 0 -1
point 4
executing ./process:
4540; START
Arrival TIme: 4
DOING A CONT
4540; tick 1
4540; SIGTSTP
Ubuntu Software
testtcheck one.1check twopoint 0 -1
point 4
executing ./process:
4542; START
Arrival TIme: 6
4542; tick 1
4542; tick 2
Arrival TIme: 6
DOING A CONT
4542; tick 3
4542; SIGTSTP
LOOPING HERE
testtcheck one.1check twopoint 0 -1
point 4
executing ./process:
4543; START
Arrival TIme: 8
4543; tick 1
Arrival TIme: 8



Discussion:

a. ***Describe and discuss what memory allocation algorithms you could have used and justify your final design choice.***

In terms of what memory allocation algorithms could have been used, there were a variety of options. For instance, the most suitable ones would have indeed been the “buddy system” or the “segregated free lists” approach. This is because in terms of the buddy system, the algorithm is very efficient and the implementation of the algorithm is fairly simple. Also, the memory blocks tend to be in sizes of powers of two, and this is very beneficial as it allows for very efficient merging and splitting. In terms of the segregated free lists approach, because the HOST system has a limited amount of available memory, keeping free lists for various block

sizes would tend to help with efficient memory allocation, increasing the overall performance and efficiency at the same time. Overall, I would choose one of these memory allocation algorithms for this task.

b. Describe and discuss the structures used by the dispatcher for queuing, dispatching and allocating memory and other resources.

In terms of the structures utilized by the dispatcher for queuing, dispatching and allocating memory and other resources, there are various different types. To start off, there are queuing structures which have subcategories known as submission queues and priority queues. There are two different types of submission queues that the dispatcher does the job of maintaining, specifically for the real-time processes and the user priority processes. Both of these queues are brought from the job dispatch list.

The other type of queue is the priority queue, which contains processes which are queued based on the specific priority levels, and are usually organized from higher to lower priority. Furthermore, the dispatcher utilizes a scheduling algorithm for this, and can cause a process degradation if they are not completed under a certain amount of time. Moving on, the four-level priority dispatcher operates based on four levels being: real-time processes, and user processes based on their specific priority levels, where the dispatcher makes sure that the higher priority processes are executed before the lower priority ones. In terms of memory partitioning, the memory allocation for processes is managed utilizing a certain variable partition allocation scheme, and each process is allocated a certain block of memory for its duration. The memory management unit (MNU) is also another structure that relates to the overall topic, where memory allocation must be managed efficiently within the certain elated constraints. Lastly, it is

important to ensure that resources are available only to the processes which actually require them, so that each process has the dedicated resources that they need.

c. Describe and justify the overall structure of your program, describing the various modules and major functions (descriptions of the function 'interfaces' are expected)

The overall structure of the program is fairly simple, with the main files containing the code being dispatchlist, main.c, Makefile, queue.h, sigtrap.c, and a file named process.txt for organization purposes, not containing any real code within it. The dispatchlist file contains the numerical values for the dispatcher, while the main.c file contains the functional code needed to run the HOST dispatcher program. Specifically, the main.c file contains the functional code to read from the dispatch list, the code required to find free memory and returning the location of memory, and various other methods and interfaces. Moving on, the “Makefile” contains the code to generate and debug binary files, and even removing .o and .exe files using rhe @echo command. The queue.h file contains the functional code to push and pop queues, and lastly, the sigtrap.c file contains the required code required in order to report the system signals applied to the necessary processes. Overall, all of the comments in the files can be utilized to better understand the code and the functional methods as well.

d. Discuss why such a multilevel dispatching scheme would be used, comparing it with schemes used by "real" operating systems. Outline shortcomings in such a scheme, suggesting possible improvements. Include the memory and resource allocation schemes in your discussions.

To begin, such a multilevel dispatching scheme would be utilized for many reasons. For example, the flexibility and priority-based scheduling are major advantages for this type of dispatching scheme. Because multilevel dispatching allows to prioritize processes based on the

order of highest priority to lowest priority, it is very useful to use that method. Also, by using this type of organizing scheme, it makes it more efficient for the dispatcher to be able to understand the new conditions and what is really required. Moving on, this also increases the efficiency of resource utilization and allocation because the higher priority processes are achieving what they need and this prevents events such as starvation and increases the amount of throughput. Other schemes such as the round robin scheduling and priority-based scheduling can be utilized because of the fact that the round robin scheduling algorithm may not prioritize the processes based on their urgency and needs, and priority-based scheduling gives priorities to the different processes.

On another note, some possible improvements can be to increase the priority of the processes that are currently waiting, so that minimizes the risk of a situation of starvation, and another possible improvement can be to implement resource allocation algorithms so that there are enough resources available and the processes that require the certain resources get what they need. A feedback mechanism can also greatly help in improving the accuracy of priority adjustments in the overall scenario.

Conclusion:

All in all, the lab project was very useful in understanding the various aspects of the HOST dispatcher program and numerous other functionalities that were taught in the course. Using the practical method and being able to create a working program is a crucial aspect of learning this topic, and the laboratory project helped in doing so. There are also many other places of improvement which can be done in terms of being able to get the necessary processes

to be able to execute with the resources that they require. The provided information, discussion, and screenshots in the report go hand in hand with the provided source code files as well