./

Learning Report – Linux OS Programming



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be approved By** | **Remarks/Revision Details** |
| 1 |  | Name/PS No | Name/PS No | Module Owner Name | Comments |
| 2 | 28/02/21 | 99003779 |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

Table of Contents

ACTIVITY 1……………………………………………………………………………………………………………………………………………………………………4

DEFINE BOX…………………………………………………………………………………………………………………………………………………………

LEARNING OUTCOME……………………………………………………………………………………………………………………………………

ACTIVITY 2……….…………………………………………………………………………………………………………………………………………………………8

DESIGN & LINK WITH LIBRARIES……………………………………………………………………………………………………………………………..8

LEARNING OUTCOME …………………………………………………………………………………………………….………………………………..9

ACTIVITY 3……….…………………………………………………………………………………………………………………………………………………………8

SYSTEM CALLS AND SIGNALS……………………………………………………………………………………………………………………………..8

LEARNING OUTCOME …………………………………………………………………………………………………….………………………………..9

ACTIVITY 4……….…………………………………………………………………………………………………………………………………………………………8

PROCESSES……………………………………………………………………………………………………………………………..8

LEARNING OUTCOME …………………………………………………………………………………………………….………………………………..9

ACTIVITY 5……….…………………………………………………………………………………………………………………………………………………………8

THREADS……………………………………………………………………………………………………………………………..8

LEARNING OUTCOME …………………………………………………………………………………………………….………………………………..9

ACTIVITY 6……….…………………………………………………………………………………………………………………………………………………………8

MUTEX AND SEMAPHORES ……………………………………………………………………………………………………………………………..8

LEARNING OUTCOME …………………………………………………………………………………………………….………………………………..9

ACTIVITY 7……….…………………………………………………………………………………………………………………………………………………………8

INTER PROCESS COMMUNICATION ………………………………………………………………………………………………………………..8

LEARNING OUTCOME …………………………………………………………………………………………………….………………………………..9

GITHUB……………………………………………………………………………………………………………………………………….…………………………10

[Table of Figures 3](#_Toc64289204)

[Table of Tables](#_Toc64289205) 3

## Table of Figures

[Figure 1 Structure](file:///C:\Users\Neha%20Tabassum\Desktop\LR_Linux-OS-Programming_99003779.docx#_Toc52177314) 4

[Figure 2 Static and dynamic libraries linking](file:///C:\Users\Neha%20Tabassum\Desktop\LR_Linux-OS-Programming_99003779.docx#_Toc52177315) 5

Figure 3: System Calls and signals …………………………………………………………………………………………………....6

Figure 4: Processes……………………………………………………………………………………………………………………………7

Figure 5: Threads………………………………………………………………………………………………………………………………8

Figure 6: Semaphores and Mutex…………………………….……………………………………………………………………….9

Figure 7: Inter process Communication (IPC)…………………………………………………………………………………..10

**ACTIVITY 1**

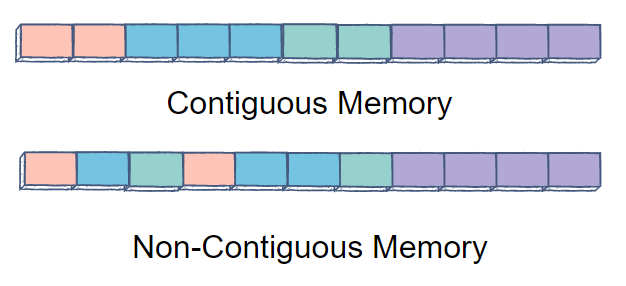
**Group Activity: Define Box**

**Group Coding Activity:**

A weighted coloured Box is associated with following attributes

-unique od, length, breadth, height, color, weight

1. Create a structure to define Box as encapsulated unit (user define type)
2. Create an array of boxes using dynamic memory allocation
3. Perform the following operations:
   1. Add a box at the end of array
   2. Display the state of all boxes
   3. Find the box with given ID
   4. Count all the boxes with specified color
   5. Find average volume of all boxes
   6. Find the difference between min and max volume
   7. Update weight of box with specified id
   8. Remove the Box with given id



**Figure 1: Structures**

**Learning OUTCOME :**

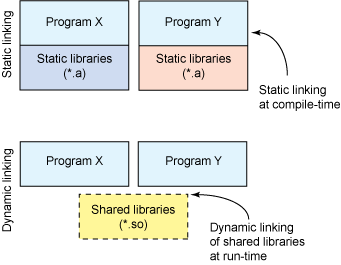
1. Understanding the togetherness of team work
2. Understanding malloc for the purpose of dynamic memory allocation.
3. Understanding function call in structure and array.
4. Understanding the concept of structure in writing the code.

**ACTIVITY 2**

**Design & Link with Libraries**

**Activity Description :**

1. To Develop functions of basic programs
2. To design the prototypes in different header files
3. To write the test code to invoke the functions
4. To write a simple makefile
5. To generate static libraries and linking with the test codes
6. To generate Dynamic libraries and linking with the test codes



**Figure 2: Static and dynamic libraries linking**

**LEARNING OUTCOME :**

1. Learnt the purpose of test files, include files and source files.
2. Learnt about how to generate static library.
3. Learnt the concept of linking of static library.
4. Learnt about how to generate dynamic library.
5. Learnt about the concept of dynamic library linking

**ACTIVITY 3**

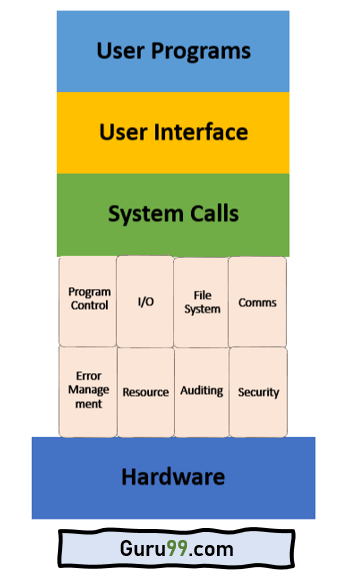
**System Calls and Signals**

**Activity Description :**

1. To copy one file contents to other using open,read,write,close system calls

2. To count no.of lines, words, characters in given file (like wc command)

3. To send specific signal to a target process (with given id, like kill command)



**Figure 3: System calls and signals**

**LEARNING OUTCOME :**

1. Learnt system call functions using close,write, read and open
2. Learnt the concept of character count, word count , line count like wc command in c programming.
3. Learnt the concept handlers in signal.

**ACTIVITY 4**

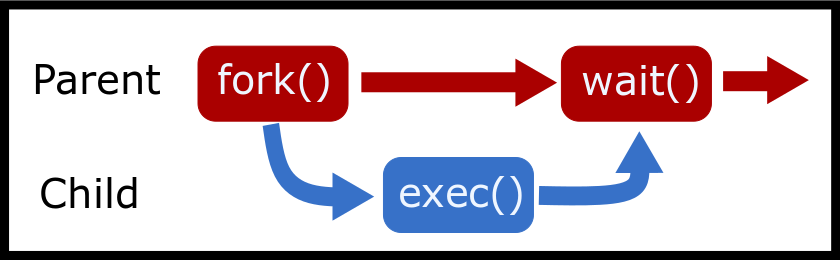
**Processes**

**Activity Description :**

1. To design our own minishell

2. To compile & link any c/c++ program within child process by launching gcc using execl/execlp.

3. To build multifile program using fork & exec as follows



**Figure 4: Processes**

**LEARNING OUTCOME :**

1. Learnt about the creation of child process with fork system.
2. Understanding of execlp system call that’s replaces current process’s data, text, heap, and stack segment with a totally new program from the disk.
3. Learnt the concept of waitpid system call which completes the child process and blocks the parent process.

**ACTIVITY 5**

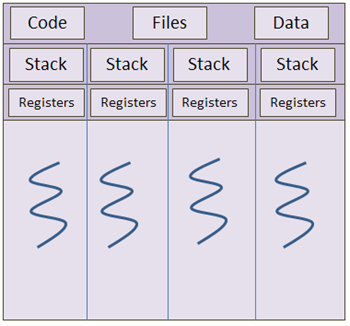
**Threads**

**Activity Description :**

1. To compute parallel sum of large array using threads.

2. To find min/max element from large array (1000 data points) using parallel computations (multthreading)

3. To print periodically, the current time



**Figure 5: Threads**

**LEARNING OUTCOME :**

1. Learnt the concept of pthread\_create function for the creation of threads.
2. Learnt the concept of pthread\_join function so that the sub thread can be executed faster than the main thread
3. Learnt the implementation of function threads for periodically printing the time.

**ACTIVITY 6**

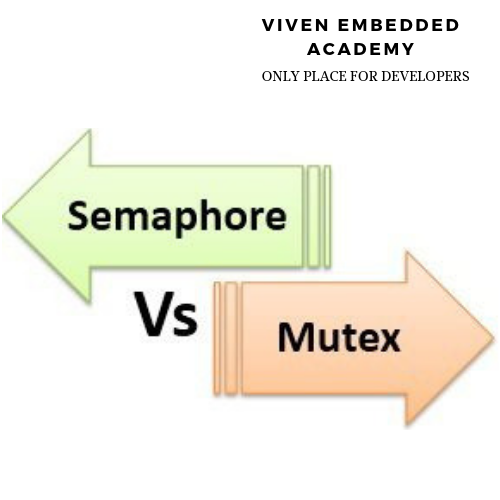
**Mutex and Semaphores**

**Activity Description :**

1. To implement producer consumer problem using Stack operations using semaphores

2.To Implement producer consumer problem using circular buffer operations using semaphores (multthreading)

3. To implement above two programs with mutex.



**Figure 6: Semaphores and mutex**

**LEARNING OUTCOME :**

1. Learnt to prevent the race around condition using mutex for the mutual exclusion of critical section.
2. Understanding of the implementation of semaphores for sysnchronization of different processes in executing the critical sections.
3. Learnt the synchronization of different processes in execution of the critical section using semaphores.
4. Learnt the ideology of deadlock and their negligence.

**ACTIVITY 7**

**Interprocess communication (IPC)**

**Activity Description :**

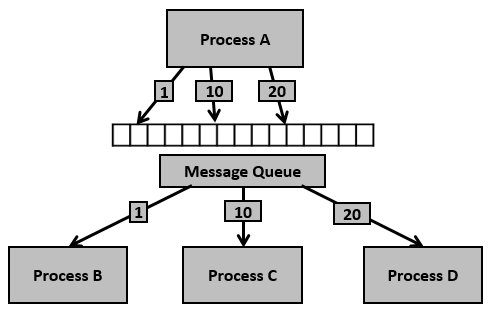
1. To implement a simple client-server scenario using message queues.

2. To retrieve file attributes using message queues.

3. To implement simple chat application between two processes using

named pipes (FIFOs)

4. To implement producer consumer problem between two processes using shared memory and named semaphores (POSIX APIs)



**Figure 7: Inter Process Communication.**

**LEARNING OUTCOME :**

1. Understood the ideology of usage and creation of named pipes for IPC.
2. Understood the concept of named semaphores between processes that are not related.
3. Understood the usage of named pipes for IPC
4. Understood the usage of unnamed shared memory for communication between two processes
5. Understood the usage of unnamed shared memory for communication between two unrelated processes

**Github :**

**Git Repositories/link**

<https://github.com/99003738/define_box.git> (Group Activity)

<https://github.com/99003726/mywork.git>