./

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………………………………………………………………………………………………………………………………………………………………4

LEARNING OUTCOME………………………………………………………………………………………………………………………………………….4

ACTIVITY 2……….…………………………………………………………………………………………………………………………………………………………5

DESIGN AND LINK WITH LIBRARIES………………………………………………………………………………………………………………………….5

LEARNING OUTCOME …………………………………………………………………………………………………….………………………………..5

ACTIVITY 3……….…………………………………………………………………………………………………………………………………………………………6

SYSTEM CALLS AND SIGNALS…………………………………………………………………………………………………………………………………..6

LEARNING OUTCOME …………………………………………………………………………………………………….………………………………..6

ACTIVITY 4……….………………………………………………………………………………………………………………………………………………………..7

PROCESSES………………………………………………………………………………………………………………………………………………………………7

LEARNING OUTCOME …………………………………………………………………………………………………….………………………………..7

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

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

LEARNING OUTCOME …………………………………………………………………………………………………….………………………………..8

ACTIVITY 6……….…………………………………………………………………………………………………………………………………………………………9

MUTEX AND SEMAPHORES……………………………………………………………………………………………………………………………………..9

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

ACTIVITY 7……….………………………………………………………………………………………………………………………………………………………10

INTER PROCESS COMMUNICATION ……………………………………………………………………………………………………………………….10

LEARNING OUTCOME …………………………………………………………………………………………………….……………………………….10

GITHUB……………………………………………………………………………………………………………………………………….…………………………11

[Table of Figures 3](#_Toc64289204)

## Table of Figures

[Figure 1 Structure and array](#_Toc52177314) 4

[Figure 2 Static and dynamic libraries linking](#_Toc52177315) 5

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

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

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

Figure 6: Semaphores……………………………………………………………………………………………………………………….9

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

**ACTIVITY 1**

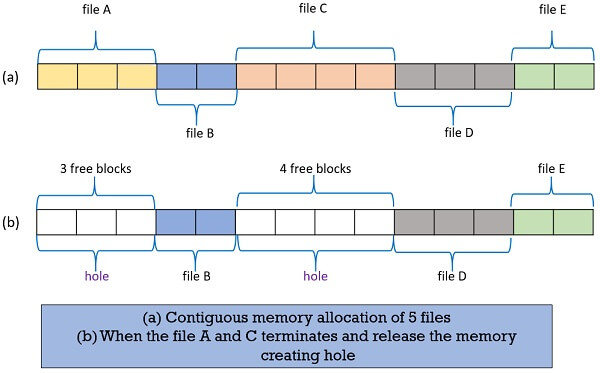
**Group Activity: Define Box**

**Group Coding Activity:**

A weighted coloured Box is associated with following attributes

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

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



**Figure 1: Structure and array**

**LEARNING OUTCOME :**

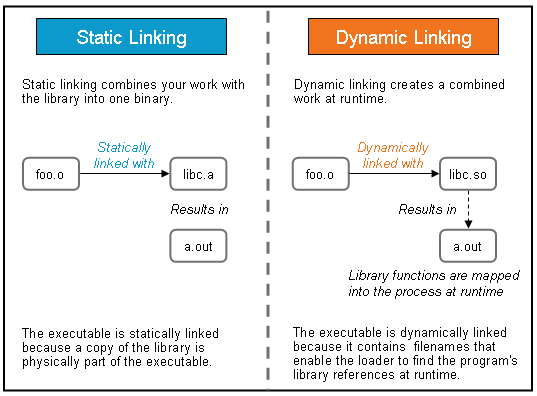
* To implement team work for the completion of the project
* To write code with understanding of structure
* To understand the concept of dynamic memory allocation using malloc
* To understand the concept of function calling using structure and array.

**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 :**

* Understanding of source file, include files and test files
* Understanding of generation of static libraries
* Understanding of the concept of static library linking
* Understanding of generation of dynamic libraries
* Understanding of the concept of dynamic library linking

**ACTIVITY 3**

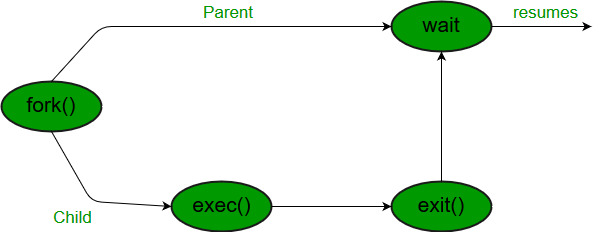
**System Calls and Signals**

**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 :**

* Understanding the concept of system call functions using open, read, write and close
* Understanding the concept of wc command sing the c programming
* Understanding the concept of handlers for signal

**ACTIVITY 4**

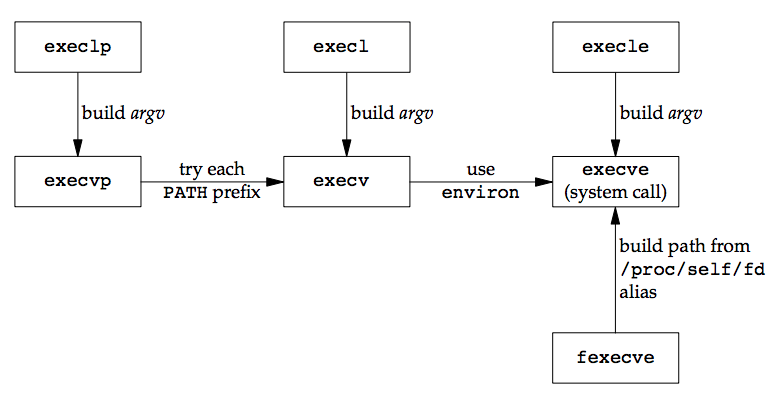
**Processes**

**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 :**

* 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.
* Understanding of fork system call for the creation of child process.
* Understanding of waitpid system call which blocks the parent process until the child process completes.

**ACTIVITY 5**

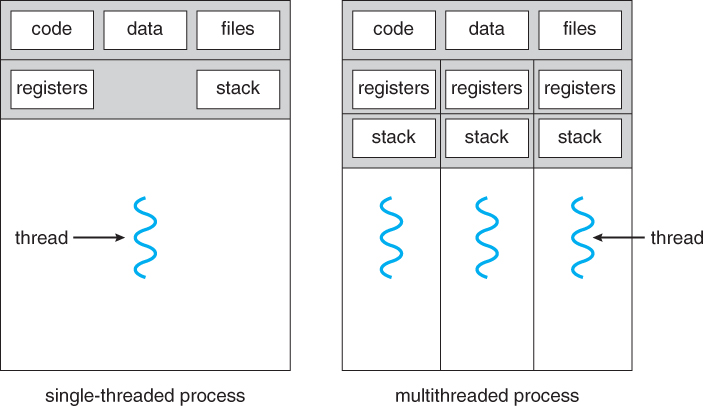
**Threads**

**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 (multithreading)

3. To print periodically, the current time



**Figure 5: Threads**

**LEARNING OUTCOME :**

* Understanding of creation of the threads using pthread\_create function
* Understanding of the concept of pthread\_join function in order to make the execution of the sub threads faster than the main threads.
* To implement the function of threads for the printing of the time periodically.

**ACTIVITY 6**

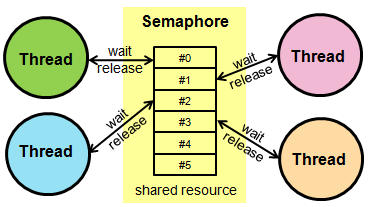
**Mutex and Semaphores**

**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**

**LEARNING OUTCOME :**

* Understanding the implementation of the mutex to prevent the race condition for the mutual exclusion of critical section.
* Understanding of the implementation of semaphores for sysnchronization of different processes in executing the critical sections.
* Understood the concept of deadlock and their avoidance.

**ACTIVITY 7**

**Inter process communication (IPC)**

**Activity Description :**

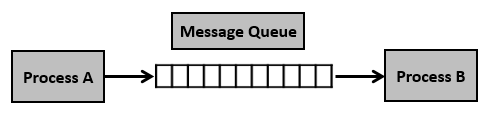
1. To implement a simple client-server scenario using message queues.

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 (IPC)**

**LEARNING OUTCOME :**

* Learnt the concept of creation and usage of named pipes for inter process communication.
* Learnt the usage of named semaphores between unrelated processes.
* Learnt the usage of named pipes for interprocess communication.
* Understsood the usage of unnamed shared memory for communication between two processes
* 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/99003779/C_Assignment.git>

<https://github.com/99003779/Linux_os_99003779.git>