./

Learning Report – Linux And OS programming

Course Code: <CODE>



Name: Thummala Satyasindhu

Ps No:99003534

Module: Linux and os programming

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
| 1 | 1/03/2021 | T. Satyasindhu |  |  |  |
| 2 | 03/03/21 | T. Satyasindhu |  |  |  |
| 3 | 05/03/21 | T. Satyasindhu |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

# 

Table of Contents

[Activity 1 3](#_Toc65877231)

[Activity 2 4](#_Toc65877232)

[Activity 3 5](#_Toc65877233)

# Activity 1

Introduction to Linux OS Architecture and design

**Type of Activity:** Individual

**Goal of Activity:** Static library and Dynamic library. Creating user defined libraries and linking user defined functions as library both statically and dynamically.

**Topics Covered:** Linux OS Architecture, Linux OS commands, GCC & Build Process

Utilities, Static & Dynamic Libraries, Makefile creation

**Learning Outcomes:** Performed different functions along with test code & Make file, Link the static & shared libraries with test code

**Challenges**: Linking static, dynamic Makefile

**References:** [**https://www.geeksforgeeks.org/static-vs-dynamic-libraries/**](https://www.geeksforgeeks.org/static-vs-dynamic-libraries/)

[**https://medium.com/@dkwok94/the-linking-process-exposed-static-vs-dynamic- libraries-977e92139b5f**](https://medium.com/@dkwok94/the-linking-process-exposed-static-vs-dynamic-%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20libraries-977e92139b5f)

# 

# Activity 2

System Calls and Signals, Processes and Threads

**Type of Activity:** Individual

**Goal of Activity:**

* Built programs using System calls, Signals and Processes
* Print current time periodically.
* Finding min/max element from large array using parallel computations.
* Compute parallel sum of large array

**Topics Covered:** system calls, interrupts, kernel, scheduling

**Learning Outcomes:**

* We could be able to write the program to copy one file contents to other using open, read, write, close system calls.
* A program to send specific signal to a target process.
* To design the minishell and program to compile & link any program within child process by launching gcc using execl and program to build multifile program using fork & exec.
* We could be able to write a program to print current time periodically.

**Challenges**:

* Understanding the fork () and how child and parent working
* And also coding the processes and threads

**References:**

<https://www.geeksforgeeks.org/input-output-system-calls-c-create-open-close-read-write/>

[**https://www.tutorialspoint.com/signals\_and\_systems/index.html**](https://www.tutorialspoint.com/signals_and_systems/index.html)

[**https://www.tutorialspoint.com/operating\_system/os\_multi\_threading.htm**](https://www.tutorialspoint.com/operating_system/os_multi_threading.htm)**l**

# 

# Activity 3

Mutex, Semaphores, Deadlock, Race condition, Pipe, Message Queue, Shared memory

**Type of Activity:** Individual

**Goal of Activity:** To understand and implement the working of mutex, semaphores, rare conditions, deadlock

**Topics Covered:** Mutex, Semaphores- Named and unnamed, Race condition, Deadlock

Pipes, Shared memory, Message queue

**Learning Outcomes** Implemented the working of mutex and semaphores of Linux OS

**Challenges**: Understanding the concepts and implementation of mutex, semaphores, rare condition

**References:** <https://www.geeksforgeeks.org/mutex-vs-semaphore/>

<https://stackoverflow.com/questions/23938652/mutexs-with-pipes-in-c>

<https://www.guru99.com/deadlock-in-operating-system.html>