./

Learning Report-Linux OS and Programming



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
| 1 | 02/03/2021 | Ragul Krishna |  |  |  |
| 2 | 04/03/2021 | Ragul Krishna |  |  |  |
| 3 | 06/03/2021 | Ragul Krishna |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

Contents

[Activity 1: Static and dynamic linking (03.03.2021) 4](#_Toc65877192)

[Activity 2: Operations on System calls, Signals and Processes (03.03.2021) 5](#_Toc65877193)

[Activity 3: Operations on Shell commands and Threads (04.03.2021) 6](#_Toc65877194)

[Activity 4: Inter Process Communication (05.03.2021) 7](#_Toc65877195)

[Activity 5: Operations on Message Queues and Pipes (05.03.2021) 8](#_Toc65877196)

# Activity 1: Static and dynamic linking (03.03.2021)

**Type of Activity**: Individual

**Goal of Activity**: To work on Linux os commands and makefile libraries.

**Topics covered:** Linux OS Architecture, GCC & Build Process, Utilities, Static & Dynamic Libraries.

**Learning Outcomes:** Was given activity based on the topics tutored like creating multiple functions with many operations like string compare, concatenation and creating makefiles depending on the conditions.

**Challenges:** While doing dynamic and static makefiles.

**GitHub Link:**

* <https://github.com/99003554/linux/tree/main/activity1>

**Learning Resources:**

* <https://www3.ntu.edu.sg/home/ehchua/programming/cpp/gcc_make.html>
* <https://embetronicx.com/tutorials/unit_testing/unit-testing-in-c-testing-with-unity/>

**References:**

* https://web.microsoftstream.com/video/9a2b1eba-61a3-4547-8292-374b2eeb5265?channelId=04fdad23-021c-4e64-bb7c-06b2469801f9
* <https://web.microsoftstream.com/video/5cc492de-e71c-4c15-98ff-53727580a5b6?channelId=04fdad23-021c-4e64-bb7c-06b2469801f9>

# Activity 2: Operations on System calls, Signals and Processes (03.03.2021)

**Type of Activity**: Individual

**Goal of Activity**: To work on Linux OS commands, signals, threads, Process and do the activity based on that topics

**Topics covered:** Linux OS Architecture, Process, Threads. Signals, Process life cycle.

**Learning Outcomes:** Was given activity based on the topics tutored like threads, process , signals and worked on different kinds of examples. In the signal example was given the task of terminating the pid of a process

**Challenges:** While creating threads and working on that.

**GitHub Link:**

* <https://github.com/99003554/linux/tree/main/activity2>

**Learning Resources:**

* <https://www.geeksforgeeks.org/input-output-system-calls-c-create-open-close-read-write/>
* <https://www.cs.uregina.ca/Links/class-info/330/SystemCall_IO/SystemCall_IO.html#FileIO>

**References:**

* <https://linuxhint.com/linux-exec-system-call/>

# Activity 3: Operations on Shell commands and Threads (04.03.2021)

**Type of Activity**: Individual

**Goal of Activity**: To understand the concept of threads and shell commands.

**Topics covered:** Zombie, Orphan and Daemon Process, Threads and Unit testing

**Learning Outcomes:** Understand thread concepts and its types, working of CPU during Context Switching and the concepts of Zombie process, Orphan process and Demon process. Implementation of test files

**Challenges:** Working on all the processes.

**Learning Resources:**

* <https://www.geeksforgeeks.org/zombie-and-orphan-processes-in-c/>
* <https://www.tutorialspoint.com/zombie-vs-orphan-vs-daemon-processes>

**References:**

* <https://www.thegeekstuff.com/2012/03/linux-threads-intro/>
* <https://www.cs.cmu.edu/afs/cs/academic/class/15492-f07/www/pthreads.html>

# Activity 4: Inter Process Communication (05.03.2021)

**Type of Activity**: Individual

**Goal of Activity**: To understand the concepts of Inter Process Communication.

**Topics covered:** Semaphores and Mutex.

**Learning Outcomes:** To prevent race around conditions and scheduling issues using Semaphores and Mutual Exclusion in critical sections.

**Challenges:** Working on IPC and its different examples.

**GitHub Link:**

* <https://github.com/99003554/linux/tree/main/5-3-21>

**Learning Resources:**

* <https://www.geeksforgeeks.org/inter-process-communication-ipc/>
* <https://www.guru99.com/inter-process-communication-ipc.html>

**References:**

* <https://opensource.com/article/19/4/interprocess-communication-linux-storage>
* <https://www.tutorialspoint.com/mutex-vs-semaphore>

# Activity 5: Operations on Message Queues and Pipes (05.03.2021)

**Type of Activity**: Individual

**Goal of Activity**: Working on pipes, queues and shared memory

**Topics covered:** Pipes, Shared memory and Message Queues.

**Learning Outcomes:** The concepts of pipes and working of inline inputs, working of shared memory and its types along with commands and Message Queues.

**Challenges:** Using shared memory concepts.

**GitHub Link:**

* <https://github.com/99003554/linux/tree/main/activity3>

**Learning Resources:**

* <https://www.geeksforgeeks.org/pipe-system-call/>
* <https://www.tutorialspoint.com/inter_process_communication/inter_process_communication_pipes.htm>

**References:**

* <https://www.tutorialspoint.com/inter_process_communication/inter_process_communication_shared_memory.htm>
* <https://www.tutorialspoint.com/inter_process_communication/inter_process_communication_message_queues.htm>