

Tutorial 1 : System Calls

July 30, 2019

Objective :

- This tutorial is intended to provide the way to use the most common system calls in order to make input-output operations on files, as well as operations to handle files and directories in Linux

Recommended Systems/Software Requirements:

- Any flavour of Linux

References:

1. *Unix concepts and applications*, Fourth Edition, Sumitabha Das, TMH.

It is expected that students brush up the topics on “File Handling in C” before attempting the tutorials. A web link for the topic has been provided in the helpful resources section.

- Tut 1 : Learn how to use the system calls for opening, reading and writing to files - *open*, *read*, *write* and *lseek* system calls
 1. Login to the system, open the *Terminal* and use the **man** command to read the manual pages of *open*, *read*, *write* and *lseek* system calls:
 2. Download the file **open_read_write_with_linux_sys_calls** provided in the helpful resources section and compile using *gcc*.

This C program intends to read 100 characters from a file and to print these 100 characters on the terminal. It uses *open* system call to open a file. The *open* call returns a file descriptor *fd* which is then used in the successive *read* and *write* system calls to access the file.
- Tut 2: Write lines of text using system calls
 1. Download the file **write_lines_of_text_sys_call** provided in the helpful resources section of Lab2 and compile using *gcc*.
- Tut 3: Simulate the *ls* command
 1. Download the file **simulating "ls" command** provided in the helpful resources section and compile using *gcc*.

This C program performs simulation of the *ls* command in Linux which lists all the folders and sub-folders of its present working directory.