Main Program :

This program uses two functions initFunction() and exitFunction() ;

initFunction() will be executed when the module is loaded by using following command :

`insmod ./SuryawanshiLKM.ko processID=1000`

In function initFunction( ) we compare processID which is the module argument passed to compare with current process's pid and print process data, parent process data and child process data.

exitFunction() will be executed when module is removed using command :

`rmmod Suryawanshi.ko`

static int __init initFunction(void) : This definition signifies that initFunciton() is the function to be invoked when module is loaded.

static void __exit exitFunction(void) : This definition signifies that exitFunction () is the function to be invoked when module is removed.

Using macros like module_init() and module_exit(), we can tell compiler which function is init and cleanup function respectively.

linux/sched.h

This is the header file needed to use task_struct data structure which stores the information about task or process

task_struct (line : 1511) Data Structure used to store data about executing process

pid (line :1632) : pid is the process id of type pid_t which is signed interger.

state (line: 1519) : it is used to store state of current process .

comm(line: 1707): It is used to store executable name of process.

prio (line: 1539) : It is used to store priority of process provided by scheduler. It is of type int.

static_prio(line: 1539) & normal_prio(line: 1539) : Since scheduler can change the priority of a process stored in prio, original priority can be stated in static and normal priority. Both are of type int

parent(line: 1645) : task struct of process has pointer to its parent struct. parent of process is pointed by this field .

children(line: 1649) : task_struct of process has pointer to head of doubly linked list of children process which is pointed by this field. It is if type list_head.

Sibling(line:1650) : task_struct of process has pointer list of sibling process which is pointed by this field. It is of type list_head.

for_each_process(line: 3049) macro used to iterate over list of process.

linux/list.h

list_for_each() (line : 424) : This macro is used to iterate over list of children.

List_entry() (line: 364) : sibling element of current process is passed to this macro. It will compute and get pointer which holds address of child which stores pointer to next child of process.