

Operating System

PROJECT REPORT



NAME: VISHWAS M
SRN: PES2UG20CS390
SEC:F
DATE:29/04/2022

ABSTRACT:

The kernel is a computer program at the core of a computer's operating system and generally has complete control over everything in the system.^[1] It is the portion of the operating system code that is always resident in memory,^[2] and facilitates interactions between hardware and software components. A full kernel controls all hardware resources (e.g. I/O, memory, cryptography) via device drivers, arbitrates conflicts between processes concerning such resources, and optimizes the utilization of common resources e.g. CPU & cache usage, file systems, and network sockets. On most systems, the kernel is one of the first programs loaded on startup (after the bootloader). It handles the rest of startup as well as memory, peripherals, and input/output (I/O) requests from software, translating them into data-processing instructions for the central processing unit.

PROJECT TITLE:

Write a kernel module that lists all current tasks in a Linux system beginning from the init task. Refer to Chapter 2 in the text book for creating Linux kernel modules. Output the task name (known as executable name), state and process id of each task in a tree structure.

CODE:

```
#include <linux/init_task.h>
#include <linux/kernel.h>
#include <linux/module.h>
//#include <linux/sched.h>

void dfs(struct task_struct *task)
{
    struct task_struct *task_next;
    struct list_head *list;

    list_for_each(list, &task->children) {
        task_next = list_entry(list, struct task_struct, sibling);

        printk(KERN_INFO "pid: %d | pname: %s | state: %ld\n", task_next->pid, task_next->comm,
task_next->state);

        dfs(task_next);
    }
}

int tasks_lister_dfs_init(void)
{
    printk(KERN_INFO "Loading module...\n");

    dfs(&init_task);

    printk(KERN_INFO "Module loaded.\n");
    return 0;
}
```

```

void tasks_lister_dfs_exit(void)
{
    printk(KERN_INFO "Module removed.\n");
}

module_init(tasks_lister_dfs_init);
module_exit(tasks_lister_dfs_exit);

MODULE_LICENSE("GPL 2.0");
MODULE_DESCRIPTION("List tasks using DFS");
MODULE_AUTHOR("ViSHWAS M");

```

MAKEFILE:

```

obj-m += tasks_lister_dfs.o

KERNELDIR ?= /lib/modules/$(shell uname -r)/build
PWD := $(shell pwd)

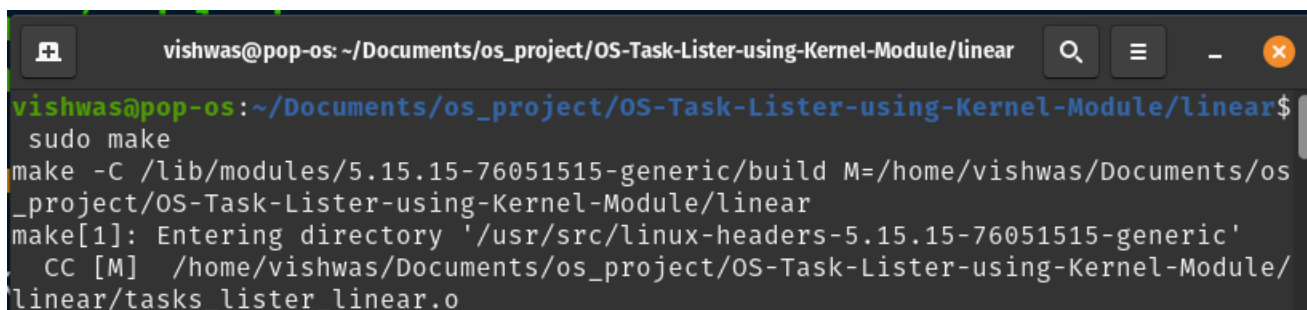
all:
    $(MAKE) -C $(KERNELDIR) M=$(PWD)

clean:
    $(MAKE) -C $(KERNELDIR) M=$(PWD) clean

```

STEPS TO RUN THE KERNEL MODULE:

- 1) OPEN THE TERMINAL**
- 2) MAKE SURE THE MAIN FILE CONTAINING THE CODE SNIPPET AND THE MAKEFILE ARE IN THE SAME DIRECTORY.**
- 3) USING “ cd ” COMMAND GO TO THE DIRECTORY WHERE THE FILES ARE PRESENT.**
- 4) WITH THE HELP OF THE COMMAND “make” WE ARE GOING TO EXTRACT ALL THE KERNEL MODULES**

A terminal window screenshot showing the execution of the 'make' command. The terminal title bar indicates the user is 'vishwas' on a 'pop-os' machine, in the directory '~/Documents/os_project/OS-Task-Lister-using-Kernel-Module/linear'. The command prompt shows 'vishwas@pop-os:~/Documents/os_project/OS-Task-Lister-using-Kernel-Module/linear\$'. The user has entered 'sudo make'. The output shows 'make -C /lib/modules/5.15.15-76051515-generic/build M=/home/vishwas/Documents/os_project/OS-Task-Lister-using-Kernel-Module/linear', followed by 'make[1]: Entering directory \'/usr/src/linux-headers-5.15.15-76051515-generic\'', and finally 'CC [M] /home/vishwas/Documents/os_project/OS-Task-Lister-using-Kernel-Module/linear/tasks_lister_linear.o'.

- 5) THEN WE HAVE TO EXEXUTE THE COMMAND “ insmod <filename>.ko”.THIS COMMAND INSTALLS ALL THE MODULES MENTIONED IN THE CODE.**
- 6) TO SEE THE REQUIRE OUTPUT OF THE PROGRAM WE HAVE TO RUN THE CODE “ dmesg ”. HERE WE CAN SEE ALL THE REQUIRED OUTPUT i.e, THE PROCESS NAME, PROCESS ID AND PROCESS STATE.**

```
vishwas@pop-os: ~/Documents/os_project/OS-Task-Lister-using-Kernel-Module/linear
10.055487] systemd-journald[314]: File /var/log/journal/ed2f8e2dd157a27cafa0
9aa620814c6/user-1000.journal corrupted or uncleanly shut down, renaming and re
lacing.
18.765261] rfkill: input handler disabled
132.934993] rfkill: input handler enabled
141.288651] rfkill: input handler disabled
1760.984664] Loading module...
1760.984753] pid: 1 | pname: systemd | state: 1
1760.984756] pid: 2 | pname: kthreadd | state: 1
1760.984757] pid: 3 | pname: rcu_gp | state: 1026
1760.984759] pid: 4 | pname: rcu_par_gp | state: 1026
1760.984760] pid: 6 | pname: kworker/0:0H | state: 1026
1760.984761] pid: 7 | pname: kworker/u2:0 | state: 1026
1760.984762] pid: 8 | pname: mm_percpu_wq | state: 1026
1760.984763] pid: 9 | pname: rcu_tasks_rude_ | state: 1
1760.984764] pid: 10 | pname: rcu_tasks_trace | state: 1
1760.984765] pid: 11 | pname: ksoftirqd/0 | state: 1
1760.984766] pid: 12 | pname: rcu_sched | state: 1026
1760.984767] pid: 13 | pname: migration/0 | state: 1
1760.984768] pid: 14 | pname: idle_inject/0 | state: 1
1760.984769] pid: 15 | pname: kworker/0:1 | state: 1026
1760.984771] pid: 16 | pname: cpuhp/0 | state: 1
1760.984772] pid: 17 | pname: kdevtmpfs | state: 1
1760.984773] pid: 18 | pname: netns | state: 1026
```

```
vishwas@pop-os: ~/Documents/os_project/OS-Task-Lister-using-Kernel-Module/linear
1760.985060] pid: 2783 | pname: pop-upgrade | state: 1
1760.985062] pid: 3173 | pname: io.elementary.a | state: 1
1760.985062] pid: 3213 | pname: GeckoMain | state: 1
1760.985063] pid: 3265 | pname: Socket Process | state: 1
1760.985065] pid: 3306 | pname: Privileged Cont | state: 1
1760.985078] pid: 3359 | pname: WebExtensions | state: 1
1760.985079] pid: 3425 | pname: Isolated Web Co | state: 1
1760.985080] pid: 3430 | pname: Isolated Web Co | state: 1
1760.985081] pid: 3477 | pname: Web Content | state: 1
1760.985082] pid: 3571 | pname: Web Content | state: 1
1760.985084] pid: 3602 | pname: Web Content | state: 1
1760.985085] pid: 3644 | pname: kworker/u2:1 | state: 1026
1760.985087] pid: 3649 | pname: nautilus | state: 1
1760.985088] pid: 3768 | pname: kworker/0:3 | state: 1026
1760.985089] pid: 3785 | pname: gnome-terminal- | state: 1
1760.985090] pid: 5225 | pname: kworker/u2:4 | state: 1026
1760.985092] pid: 5983 | pname: kworker/u2:2 | state: 1026
1760.985093] pid: 5984 | pname: bash | state: 1
1760.985094] pid: 5992 | pname: kworker/0:0 | state: 1026
1760.985095] pid: 6225 | pname: gedit | state: 1
1760.985096] pid: 6561 | pname: sudo | state: 1
1760.985096] pid: 6562 | pname: insmod | state: 0
1760.985097] Module loaded.
```

7) TO SEE THE TREE STRUCTURE OF THE PROGRAM WE HAVE TO TYPE THE COMMAND “pstree”.

```
vishwas@pop-os: ~/Documents/os_project/OS-Task-Lister-using-Kernel-Module/linear
|
|_gsd-rfkill—2*[{gsd-rfkill}]
|_gsd-screensaver—2*[{gsd-screensaver}]
|_gsd-sharing—3*[{gsd-sharing}]
|_gsd-smartcard—3*[{gsd-smartcard}]
|_gsd-sound—3*[{gsd-sound}]
|_gsd-wacom—3*[{gsd-wacom}]
|_gsd-xsettings—3*[{gsd-xsettings}]
|_gvfs-afc-volume—3*[{gvfs-afc-volume}]
|_gvfs-goa-volume—2*[{gvfs-goa-volume}]
|_gvfs-gphoto2-vo—2*[{gvfs-gphoto2-vo}]
|_gvfs-mtp-volume—2*[{gvfs-mtp-volume}]
|_gvfs-udisks2-vo—3*[{gvfs-udisks2-vo}]
|_gvfsd—gvfsd-trash—2*[{gvfsd-trash}]
|_gvfsd—2*[{gvfsd}]
|_gvfsd-fuse—5*[{gvfsd-fuse}]
|_gvfsd-metadata—2*[{gvfsd-metadata}]
|_ibus-portal—2*[{ibus-portal}]
|_ibus-x11—3*[{ibus-x11}]
|_nautilus—3*[{nautilus}]
|_pipewire—{pipewire}
|_pipewire-media—{pipewire-media-}
|_pop-system-upda—{pop-system-upda}
|_pop-upgrade—{pop-upgrade}
|_pulseaudio—3*[{pulseaudio}]
|_tracker-miner-f—5*[{tracker-miner-f}]
|_xdg-permission—2*[{xdg-permission-}]
systemd-journal
systemd-logind
systemd-resolve
systemd-timesyn—{systemd-timesyn}
systemd-udev
toucheegg—3*[{toucheegg}]
udisksd—4*[{udisksd}]
upowerd—2*[{upowerd}]
wpa_supplicant
vishwas@pop-os:~/Documents/os_project/OS-Task-Lister-using-Kernel-Module/linear$
```

```
vishwas@pop-os:~/Documents/os_project/OS-Task-Lister-using-Kernel-Module/linear$ pstree
systemd—ModemManager—2*[{ModemManager}]
|_NetworkManager—2*[{NetworkManager}]
|_accounts-daemon—2*[{accounts-daemon}]
|_acpid
|_apache2—2*[{apache2}—26*[{apache2}]]
|_avahi-daemon—avahi-daemon
|_colord—2*[{colord}]
|_cron
|_cups-browsed—2*[{cups-browsed}]
|_cupsd—2*[{dbus}]
|_dbus-daemon
|_fwupd—4*[{fwupd}]
|_gdm3—gdm-session-wor—gdm-x-session—Xorg—{Xorg}
|_gdm3—2*[{gdm3}]
|_gdm3—gdm-session-wor—2*[{gdm-session-wor}]
|_gdm3—gdm-x-session—gnome-session-b—2*[{gnome-+}
|_gdm3—gdm-x-session—2*[{gdm-x-session}]
|_gnome-keyring-d—3*[{gnome-keyring-d}]
|_named—4*[{named}]
|_networkd-dispat
|_packagekitd—2*[{packagekitd}]
|_polkitd—2*[{polkitd}]
|_pop-system-upda—{pop-system-upda}
|_pop-upgrade—2*[{pop-upgrade}]
|_python3
|_rsyslogd—3*[{rsyslogd}]
|_rtkit-daemon—2*[{rtkit-daemon}]
|_switcheroo-cont—2*[{switcheroo-cont}]
|_system76-power
|_system76-schedu—2*[{system76-schedu}]
|_systemd—(sd-pam)
|_systemd—GeckoMain—Isolated Web Co—14*[{Isolated Web Co}]
|_systemd—GeckoMain—Privileged Cont—14*[{Privileged Cont}]
|_systemd—GeckoMain—Socket Process—4*[{Socket Process}]
|_systemd—GeckoMain—WebExtensions—14*[{WebExtensions}]
|_systemd—GeckoMain—88*[{GeckoMain}]
```

CONCLUSION:

APART FROM LISTING ALL THE PROCESSES RUNNING ON THE OPERATING SYSTEMS,WE CAN USE THESE KERNEL PROGRAMMING IN:

- 1. Process Management**
- 2.Memory Management**
- 3.Device Management**
- 4.Interrupt Handling**
- 5.Input Output Communication**