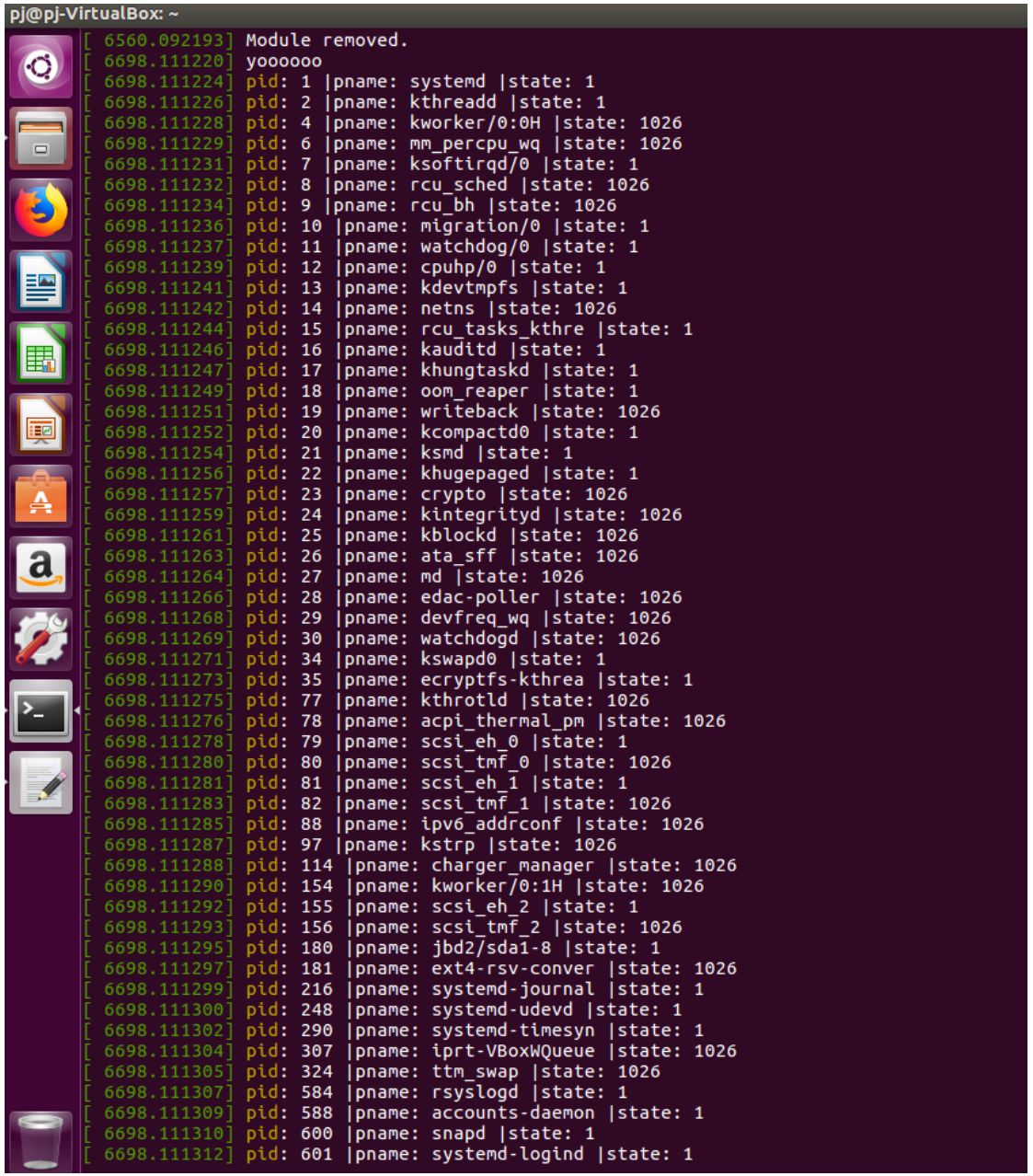
Linux Kernel Module for Listing Tasks

- In this project, you will write a kernel module that **lists all current tasks** in a Linux system.

Part I—Iterating over Tasks Linearly

- Design a kernel module that iterates through all tasks in the system using the **for_each_process()** macro(in <linux/sched/signal.h>).
- In particular, output the process id, task name (known as ***executable name***), and state of each task.
- You will probably have to read through the **task_struct** structure in **<linux/sched.h>** to obtain the names of these fields.

- Write this code in the module entry point so that its contents will appear in the **kernel log buffer**, which can be viewed using the **dmesg** command.


pj@pj-VirtualBox: ~
[6560.092193] Module removed.
[6698.111220] yooooooo
[6698.111224] pid: 1 |pname: systemd |state: 1
[6698.111226] pid: 2 |pname: kthreadd |state: 1
[6698.111228] pid: 4 |pname: kworker/0:0H |state: 1026
[6698.111229] pid: 6 |pname: mm_percpu_wq |state: 1026
[6698.111231] pid: 7 |pname: ksoftirqd/0 |state: 1
[6698.111232] pid: 8 |pname: rcu_sched |state: 1026
[6698.111234] pid: 9 |pname: rcu_bh |state: 1026
[6698.111236] pid: 10 |pname: migration/0 |state: 1
[6698.111237] pid: 11 |pname: watchdog/0 |state: 1
[6698.111239] pid: 12 |pname: cpuhp/0 |state: 1
[6698.111241] pid: 13 |pname: kdevtmpfs |state: 1
[6698.111242] pid: 14 |pname: netns |state: 1026
[6698.111244] pid: 15 |pname: rcu_tasks_kthre |state: 1
[6698.111246] pid: 16 |pname: kauditd |state: 1
[6698.111247] pid: 17 |pname: khungtaskd |state: 1
[6698.111249] pid: 18 |pname: oom_reaper |state: 1
[6698.111251] pid: 19 |pname: writeback |state: 1026
[6698.111252] pid: 20 |pname: kcompactd0 |state: 1
[6698.111254] pid: 21 |pname: ksmd |state: 1
[6698.111256] pid: 22 |pname: khugepaged |state: 1
[6698.111257] pid: 23 |pname: crypto |state: 1026
[6698.111259] pid: 24 |pname: kintegrityd |state: 1026
[6698.111261] pid: 25 |pname: kblockd |state: 1026
[6698.111263] pid: 26 |pname: ata_sff |state: 1026
[6698.111264] pid: 27 |pname: md |state: 1026
[6698.111266] pid: 28 |pname: edac-poller |state: 1026
[6698.111268] pid: 29 |pname: devfreq_wq |state: 1026
[6698.111269] pid: 30 |pname: watchdogd |state: 1026
[6698.111271] pid: 34 |pname: kswapd0 |state: 1
[6698.111273] pid: 35 |pname: ecryptfs-kthrea |state: 1
[6698.111275] pid: 77 |pname: kthrotld |state: 1026
[6698.111276] pid: 78 |pname: acpi_thermal_pm |state: 1026
[6698.111278] pid: 79 |pname: scsi_eh_0 |state: 1
[6698.111280] pid: 80 |pname: scsi_tmf_0 |state: 1026
[6698.111281] pid: 81 |pname: scsi_eh_1 |state: 1
[6698.111283] pid: 82 |pname: scsi_tmf_1 |state: 1026
[6698.111285] pid: 88 |pname: ipv6_addrconf |state: 1026
[6698.111287] pid: 97 |pname: kstrp |state: 1026
[6698.111288] pid: 114 |pname: charger_manager |state: 1026
[6698.111290] pid: 154 |pname: kworker/0:1H |state: 1026
[6698.111292] pid: 155 |pname: scsi_eh_2 |state: 1
[6698.111293] pid: 156 |pname: scsi_tmf_2 |state: 1026
[6698.111295] pid: 180 |pname: jbd2/sda1-8 |state: 1
[6698.111297] pid: 181 |pname: ext4-rsv-conver |state: 1026
[6698.111299] pid: 216 |pname: systemd-journal |state: 1
[6698.111300] pid: 248 |pname: systemd-udevd |state: 1
[6698.111302] pid: 290 |pname: systemd-timesyn |state: 1
[6698.111304] pid: 307 |pname: iprt-VBoxWQueue |state: 1026
[6698.111305] pid: 324 |pname: ttm_swap |state: 1026
[6698.111307] pid: 584 |pname: rsyslogd |state: 1
[6698.111309] pid: 588 |pname: accounts-daemon |state: 1
[6698.111310] pid: 600 |pname: snapd |state: 1
[6698.111312] pid: 601 |pname: systemd-logind |state: 1

- To verify that your code is working correctly, compare the contents of the kernel log buffer with the output of the following command, which lists all tasks in the system: ***ps –el***

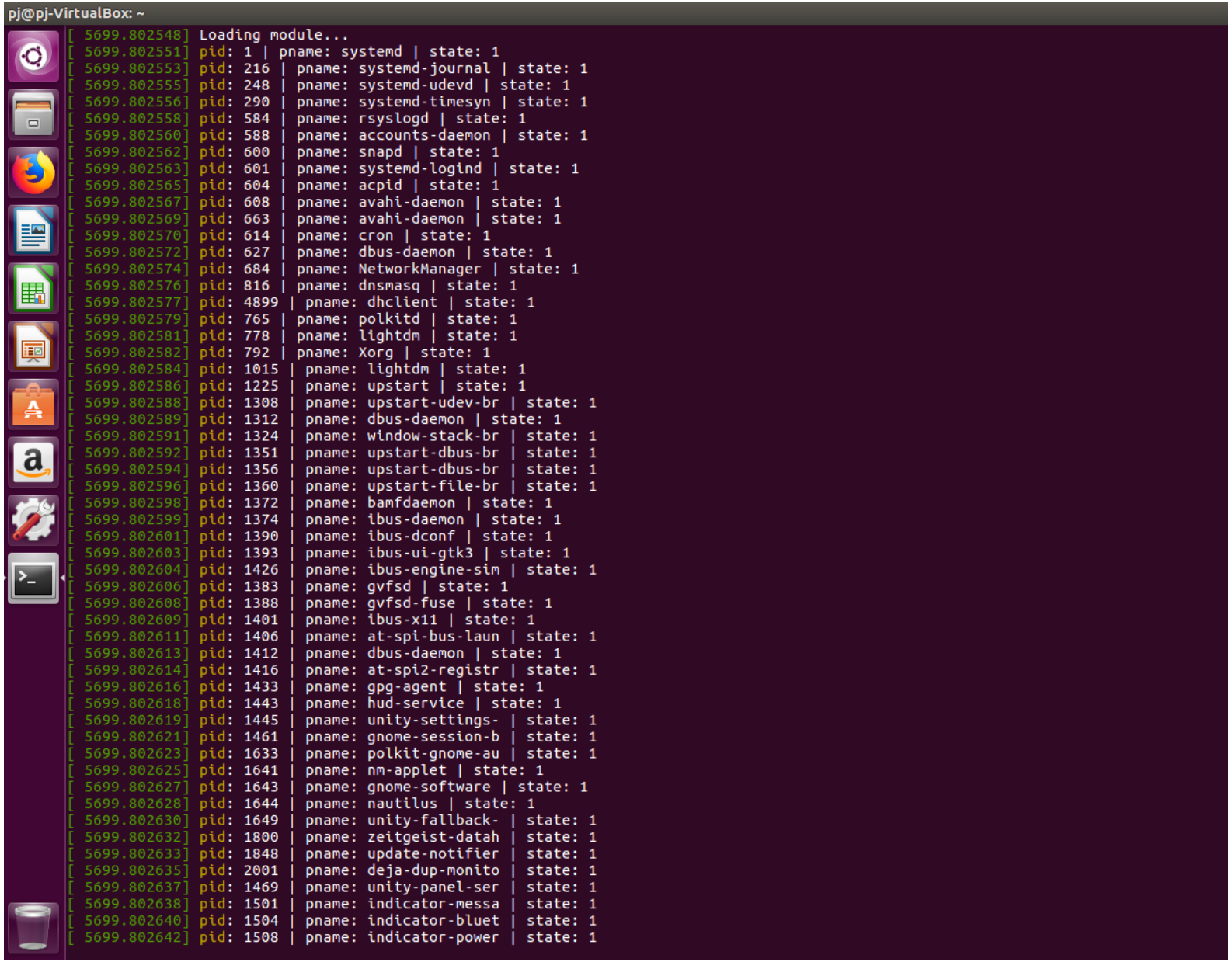
- The two values should be very similar. Because tasks are dynamic, however, it is possible that a few tasks may appear in one listing but not the other.

Part II—Iterating over Tasks with a Depth-First Search Tree

- The second portion of this project involves iterating over all tasks in the system using a **depth-first search** (DFS) tree.

trace sched.h

- Examining the task struct in <linux/sched.h>, we see two struct list head objects: **children** and **sibling**.
- These objects are pointers to a list of the task's children, as well as its siblings.
- Beginning from the init_task, design a kernel module that iterates over all tasks in the system using a DFS tree.
- Just as in the first part of this project, output the pid, name, and state of each task.
- Perform this iteration in the kernel entry module so that its output appears in the kernel log buffer.


pj@pj-VirtualBox: ~
[5699.802548] Loading module...
[5699.802551] pid: 1 | pname: systemd | state: 1
[5699.802553] pid: 216 | pname: systemd-journal | state: 1
[5699.802555] pid: 248 | pname: systemd-udevd | state: 1
[5699.802556] pid: 290 | pname: systemd-timesyn | state: 1
[5699.802558] pid: 584 | pname: rsyslogd | state: 1
[5699.802560] pid: 588 | pname: accounts-daemon | state: 1
[5699.802562] pid: 600 | pname: snapd | state: 1
[5699.802563] pid: 601 | pname: systemd-logind | state: 1
[5699.802565] pid: 604 | pname: acpid | state: 1
[5699.802567] pid: 608 | pname: avahi-daemon | state: 1
[5699.802569] pid: 663 | pname: avahi-daemon | state: 1
[5699.802570] pid: 614 | pname: cron | state: 1
[5699.802572] pid: 627 | pname: dbus-daemon | state: 1
[5699.802574] pid: 684 | pname: NetworkManager | state: 1
[5699.802576] pid: 816 | pname: dnsmasq | state: 1
[5699.802577] pid: 4899 | pname: dhclient | state: 1
[5699.802579] pid: 765 | pname: polkitd | state: 1
[5699.802581] pid: 778 | pname: lightdm | state: 1
[5699.802582] pid: 792 | pname: Xorg | state: 1
[5699.802584] pid: 1015 | pname: lightdm | state: 1
[5699.802586] pid: 1225 | pname: upstart | state: 1
[5699.802588] pid: 1308 | pname: upstart-udev-br | state: 1
[5699.802589] pid: 1312 | pname: dbus-daemon | state: 1
[5699.802591] pid: 1324 | pname: window-stack-br | state: 1
[5699.802592] pid: 1351 | pname: upstart-dbus-br | state: 1
[5699.802594] pid: 1356 | pname: upstart-dbus-br | state: 1
[5699.802596] pid: 1360 | pname: upstart-file-br | state: 1
[5699.802598] pid: 1372 | pname: bamfdaemon | state: 1
[5699.802599] pid: 1374 | pname: ibus-daemon | state: 1
[5699.802601] pid: 1390 | pname: ibus-dconf | state: 1
[5699.802603] pid: 1393 | pname: ibus-ui-gtk3 | state: 1
[5699.802604] pid: 1426 | pname: ibus-engine-sim | state: 1
[5699.802606] pid: 1383 | pname: gvfsd | state: 1
[5699.802608] pid: 1388 | pname: gvfsd-fuse | state: 1
[5699.802609] pid: 1401 | pname: ibus-x11 | state: 1
[5699.802611] pid: 1406 | pname: at-spi-bus-laun | state: 1
[5699.802613] pid: 1412 | pname: dbus-daemon | state: 1
[5699.802614] pid: 1416 | pname: at-spi2-registr | state: 1
[5699.802616] pid: 1433 | pname: gpg-agent | state: 1
[5699.802618] pid: 1443 | pname: hud-service | state: 1
[5699.802619] pid: 1445 | pname: unity-settings- | state: 1
[5699.802621] pid: 1461 | pname: gnome-session-b | state: 1
[5699.802623] pid: 1633 | pname: polkit-gnome-au | state: 1
[5699.802625] pid: 1641 | pname: nm-applet | state: 1
[5699.802627] pid: 1643 | pname: gnome-software | state: 1
[5699.802628] pid: 1644 | pname: nautilus | state: 1
[5699.802630] pid: 1649 | pname: unity-fallback- | state: 1
[5699.802632] pid: 1800 | pname: zeitgeist-datah | state: 1
[5699.802633] pid: 1848 | pname: update-notifier | state: 1
[5699.802635] pid: 2001 | pname: deja-dup-monito | state: 1
[5699.802637] pid: 1469 | pname: unity-panel-ser | state: 1
[5699.802638] pid: 1501 | pname: indicator-messa | state: 1
[5699.802640] pid: 1504 | pname: indicator-bluet | state: 1
[5699.802642] pid: 1508 | pname: indicator-power | state: 1

- To check the output of the DFS tree, use the command ***ps -eLf***

- Hw2_{studentID}.rar:
- hw2_linear.c (40%)
- hw2_dfs.c (40%)
- hw2_report(20%)
- Tell us how you implement your homework in detail and show us your results.
- 0 will be given to cheaters, so don’t copy & paste your friend’s code directly.
- Deadline:4/14(Sun.) 23:59