Assignment-8

Linux System and its Applications

Systems and Storage Laboratory
Department of Computer Science and Engineering
Chung-Ang University

Assignment-8: Checking CPU utilization

- **First, printk() the CPU id in `pxt4_file_write_iter()`**
  1. Build pxt4 module and mount the testing device with pxt4
     - **Free space** in the device must be larger than **3 * (RAM size)**
  2. Run Fio test with the following condition (next page)
     - Refer to [Practical Class 7-a]
  3. Unmount the device and remove pxt4 module
  4. Use `dmesg` command to check the result

- **Then use ds_monitoring to check how many times `pxt4_file_write_iter()` have been called from each CPU**
  - Do the same above 1 ~ 4

Assignment-8: Checking CPU utilization

- **Test configuration should satisfy the below conditions**
    - Buffered Sequential Write
    - Block Size = 4K
    - Numjobs = machine CPU cores
    - Total size = 3 times larger than your memory size
  - Example Fio Script

```
; -- start job file --
[global]
name=<job name>
directory=<pxt4 filesystem mount point (e.g. /mnt/test)>
rw=write
bs=4K
direct=0
numjobs=<number of CPU cores>
verify=meta

[fio-test]
size=<(RAM size) * 3 / (numjobs)>
group_reporting
; -- end job file --
```

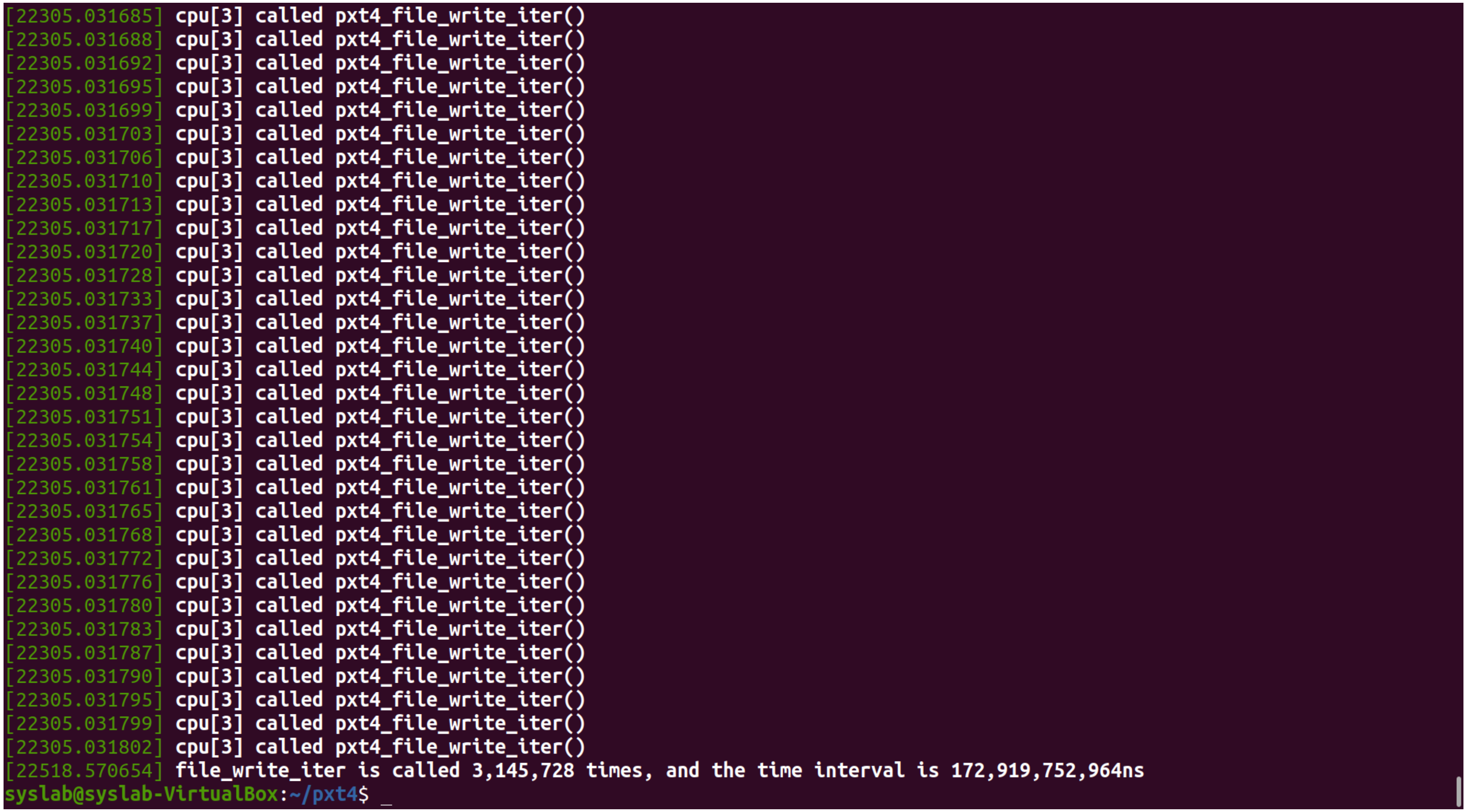
Assignment-8: Checking CPU utilization

- **What to handout**
  - Take a screenshot of
    1. First dmesg result from printk
       - ✓ You don't have to include whole lines from pxt4.
       - ✓ Just last lines are okay.
    2. The ds_monitoring result that contains
       - ✓ CPU id
       - ✓ Function call counts
       - ✓ Overall function call rate percentage

- **Submit within pdf format**
  - Make sure to include your name and student id

Example screenshot 1

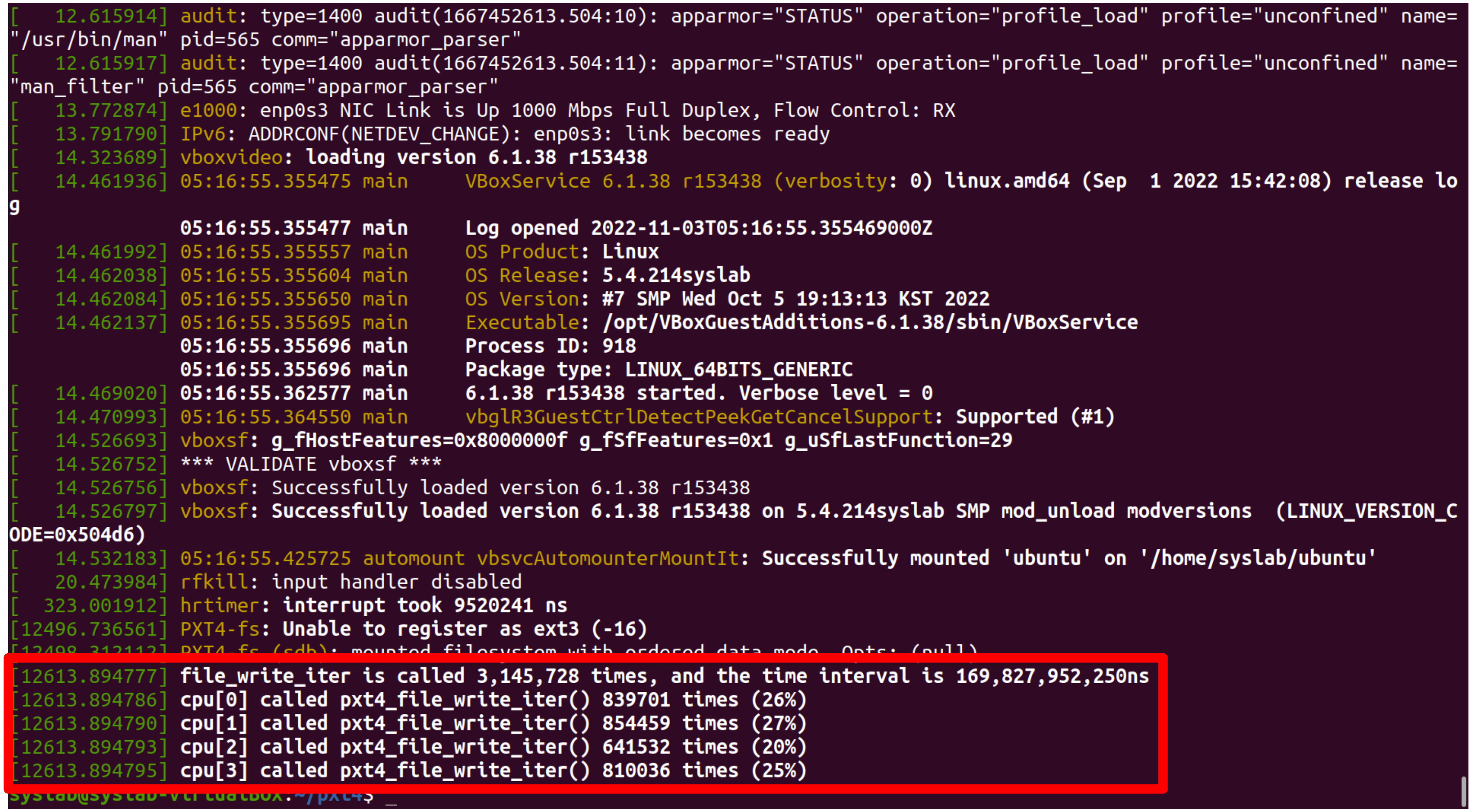
❖ A lot of printk()s in dmesg, difficult to track



```
[22305.031685] cpu[3] called pxt4_file_write_iter()
[22305.031688] cpu[3] called pxt4_file_write_iter()
[22305.031692] cpu[3] called pxt4_file_write_iter()
[22305.031695] cpu[3] called pxt4_file_write_iter()
[22305.031699] cpu[3] called pxt4_file_write_iter()
[22305.031703] cpu[3] called pxt4_file_write_iter()
[22305.031706] cpu[3] called pxt4_file_write_iter()
[22305.031710] cpu[3] called pxt4_file_write_iter()
[22305.031713] cpu[3] called pxt4_file_write_iter()
[22305.031717] cpu[3] called pxt4_file_write_iter()
[22305.031720] cpu[3] called pxt4_file_write_iter()
[22305.031728] cpu[3] called pxt4_file_write_iter()
[22305.031733] cpu[3] called pxt4_file_write_iter()
[22305.031737] cpu[3] called pxt4_file_write_iter()
[22305.031740] cpu[3] called pxt4_file_write_iter()
[22305.031744] cpu[3] called pxt4_file_write_iter()
[22305.031748] cpu[3] called pxt4_file_write_iter()
[22305.031751] cpu[3] called pxt4_file_write_iter()
[22305.031754] cpu[3] called pxt4_file_write_iter()
[22305.031758] cpu[3] called pxt4_file_write_iter()
[22305.031761] cpu[3] called pxt4_file_write_iter()
[22305.031765] cpu[3] called pxt4_file_write_iter()
[22305.031768] cpu[3] called pxt4_file_write_iter()
[22305.031772] cpu[3] called pxt4_file_write_iter()
[22305.031776] cpu[3] called pxt4_file_write_iter()
[22305.031780] cpu[3] called pxt4_file_write_iter()
[22305.031783] cpu[3] called pxt4_file_write_iter()
[22305.031787] cpu[3] called pxt4_file_write_iter()
[22305.031790] cpu[3] called pxt4_file_write_iter()
[22305.031795] cpu[3] called pxt4_file_write_iter()
[22305.031799] cpu[3] called pxt4_file_write_iter()
[22305.031802] cpu[3] called pxt4_file_write_iter()
[22518.570654] file_write_iter is called 3,145,728 times, and the time interval is 172,919,752,964ns
syslab@syslab-VirtualBox:~/pxt4$ _
```

Example screenshot 2

❖ Using ds_monitoring



```
[   12.615914] audit: type=1400 audit(1667452613.504:10): apparmor="STATUS" operation="profile_load" profile="unconfined" name=
"/usr/bin/man" pid=565 comm="apparmor_parser"
[   12.615917] audit: type=1400 audit(1667452613.504:11): apparmor="STATUS" operation="profile_load" profile="unconfined" name=
"man_filter" pid=565 comm="apparmor_parser"
[   13.772874] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[   13.791790] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[   14.323689] vboxvideo: loading version 6.1.38 r153438
[   14.461936] 05:16:55.355475 main     VBoxService 6.1.38 r153438 (verbosity: 0) linux.amd64 (Sep  1 2022 15:42:08) release lo
g
               05:16:55.355477 main     Log opened 2022-11-03T05:16:55.355469000Z
[   14.461992] 05:16:55.355557 main     OS Product: Linux
[   14.462038] 05:16:55.355604 main     OS Release: 5.4.214syslab
[   14.462084] 05:16:55.355650 main     OS Version: #7 SMP Wed Oct 5 19:13:13 KST 2022
[   14.462137] 05:16:55.355695 main     Executable: /opt/VBoxGuestAdditions-6.1.38/sbin/VBoxService
               05:16:55.355696 main     Process ID: 918
               05:16:55.355696 main     Package type: LINUX_64BITS_GENERIC
[   14.469020] 05:16:55.362577 main     6.1.38 r153438 started. Verbose level = 0
[   14.470993] 05:16:55.364550 main     vbglR3GuestCtrlDetectPeekGetCancelSupport: Supported (#1)
[   14.526693] vboxsf: g_fHostFeatures=0x8000000f g_fSfFeatures=0x1 g_uSfLastFunction=29
[   14.526752] *** VALIDATE vboxsf ***
[   14.526756] vboxsf: Successfully loaded version 6.1.38 r153438
[   14.526797] vboxsf: Successfully loaded version 6.1.38 r153438 on 5.4.214syslab SMP mod_unload modversions  (LINUX_VERSION_C
ODE=0x504d6)
[   14.532183] 05:16:55.425725 automount vbsvcAutomounterMountIt: Successfully mounted 'ubuntu' on '/home/syslab/ubuntu'
[   20.473984] rfkill: input handler disabled
[  323.001912] hrtimer: interrupt took 9520241 ns
[12496.736561] PXT4-fs: Unable to register as ext3 (-16)
[12498.312112] PXT4-fs (sdb): mounted filesystem with ordered data mode. Opts: (null)
[12613.894777] file_write_iter is called 3,145,728 times, and the time interval is 169,827,952,250ns
[12613.894786] cpu[0] called pxt4_file_write_iter() 839701 times (26%)
[12613.894790] cpu[1] called pxt4_file_write_iter() 854459 times (27%)
[12613.894793] cpu[2] called pxt4_file_write_iter() 641532 times (20%)
[12613.894795] cpu[3] called pxt4_file_write_iter() 810036 times (25%)
syslab@syslab-VirtualBox:~/pxt4$ _
```

SySLAB Systems and Storage Laboratory

Tips

- Get the currently running task's `task_struct` with macro "`current`":

```
static __always_inline struct task_struct *get_current(void)
{
    return this_cpu_read_stable(current_task);
}

#define current get_current()
```

From `arch/x86/include/asm/current.h`

- Get current task's CPU id with:

```
current->cpu
```

  - Cf> `struct task_struct`

```
struct task_struct {
    ...
    /* Current CPU: */
    unsigned int            cpu;
    ...
}
```