CS60038: Advances in Operating Systems Design

Student 1: Sourabh Soumyakanta Das (20CS30051)
Student 2: Shiladitya De (20CS30061) **Assignment:** 1

Problem : Building the Linux Kernel

You need to make the following configurations using **`menuconfig`**, and then build and install the compiled kernel over the Ubuntu **20.04 LTS** Desktop version. You need to download the kernel source from https://www.kernel.org/ only. Do not use any other third-party download servers.

1. Remove NUMA memory allocation, scheduler, and emulation
2. Remove Kyber I/O Scheduler
3. Include multipath TCP (MPTCP)

§1 Remove NUMA memory allocation, scheduler and emulation

NUMA was removed by unselecting it from `processor type and features`

§1.1 Before removing NUMA

We can see various options of NUMA which are set in `config` file:

```
ssd@ssd: $ cat config | grep 'NUMA'
CONFIG_ARCH_SUPPORTS_NUMA_BALANCING=y
CONFIG_NUMA_BALANCING=y
CONFIG_NUMA_BALANCING_DEFAULT_ENABLED=y
CONFIG_X86_NUMACHIP=y
CONFIG_NUMA=y
CONFIG_AMD_NUMA=y
CONFIG_X86_64_ACPI_NUMA=y
CONFIG_NUMA_EMU=y
CONFIG_USE_PERCPU_NUMA_NODE_ID=y
CONFIG_ACPI_NUMA=y
CONFIG_NUMA_KEEP_MEMINFO=y
```

Even we can see the NUMA related info using `numactl --hardware` command:

```
ssd@ssd: $ numactl --hardware
available: 1 nodes (0)
node 0 cpus: 0 1 2 3
node 0 size: 3475 MB
node 0 free: 1202 MB
node distances:
node   0
  0:  10
ssd@ssd: $
```

§1.2 After removing NUMA

Now it can be seen that *CONFIG_NUMA* is not set:

```
ssd@ssd:~/Desktop$ cat config | grep 'NUMA'
CONFIG_ARCH_SUPPORTS_NUMA_BALANCING=y
# CONFIG_NUMA is not set
```

On using the command `numactl --hardware`, it can be found that it is NUMA is not available on the system:

```
ssd@ssd:~/Desktop$ numactl --hardware
No NUMA available on this system
```

§2 Remove Kyber I/O Schedular

Kyber I/O was removed by unselecting the options from `IO Schedulers`.

§2.1 Before removing Kyber I/O

We can see that *CONFIG_MQ_IOSCHED_KYBER* is set in config:

```
ssd@ssd: $ cat config | grep 'KYBER'
CONFIG_MQ_IOSCHED_KYBER=y
```

Even we can find the `kyber_iosched.ko` file in `/lib/modules/5.10.191/kernel/block/`:

```
ssd@ssd:/lib/modules/5.10.191/kernel/block$ ls | grep kyber
kyber-iosched.ko
```

§2.2 After removing Kyber I/O scheduler

We can now see that *CONFIG_MQ_IOSCHED_KYBER* is not set in config:

```
ssd@ssd:~$ cat config | grep 'KYBER'
# CONFIG_MQ_IOSCHED_KYBER is not set
```

We can see that the `kyber_iosched.ko` is not there in the directory:

```
ssd@ssd:/lib/modules/5.10.191/kernel$ cd block
bash: cd: block: No such file or directory
ssd@ssd:/lib/modules/5.10.191/kernel$
```

§3 Include multipath TCP

MPTCP is included by selecting in the `Networking support > Networking options`

§3.1 Before adding MPTCP

We can see that *CONFIG_MPTCP* is not set in config:

```
ssd@ssd:~/Desktop$ cat config | grep 'MPTCP'
# CONFIG_MPTCP is not set
```

When we ran `dmesg | grep MPTCP` we could not find anything:

```
ssd@ssd:~/Desktop$ dmesg | grep MPTCP
ssd@ssd:~/Desktop$
```

§3.2 After adding MPTCP

We can see that *CONFIG_MPTCP* is now set in config:

```
ssd@ssd: $ cat config | grep 'MPTCP'
CONFIG_MPTCP=y
CONFIG_MPTCP_IPV6=y
```

Now on running `dmesg | grep MPTCP`, we can see about MPTCP:

```
ssd@ssd: $ dmesg | grep MPTCP
[    0.315632] MPTCP token hash table entries: 4096 (order: 4, 98304 bytes, linear)
ssd@ssd: $
```