**Homework 2: Device Drivers and Kernel Modules (Due December 18, 2019, Wednesday , 9 AM)**

We wish to create a memo book of measurements including a unit converter as a character device driver in Linux. Three types of unit conversions will be possible:

- temperature measurement units (between Celcius, Fahrenheit and Kelvin),
- distance measurement units (between kilometers and miles) and
- weight measurement units (between kilograms and pounds).

Each memo book will be able to store measurements of the same type. Thus, each memo book will be implemented as a separate device; i.e. memo0 for storing and converting temperature measurements, memo1 for storing and converting distance measurements and memo2 for storing and converting weight measurements.

All devices will be single open devices, i.e. only one process will be able to access a device at a time. You can refer to the *Linux Device Drivers* book for an example on how to do this.

The device nodes will be named /dev/memo0, /dev/memo1 and /dev/memo2. Each of the three devices contains a global and persistent memory area made up of a *mode* field and a *linked list*. The mode determines the current setting for the measurement unit.

Mode will be:

- 0: Celcius, 1: Fahrenheit, 2: Kelvin
- 0: kilometers, 1: miles
- 0: kilograms, 1: pounds

Each entry in the linked list will be a structure consisting of a *date field* and a *measurement value* field.

All measurements are stored in their corresponding devices as metric units (Celcius, kilometers and kilograms). Before any read and write operations, the current mode of the device has to be set using the appropriate IOCTL command given below.

Any text **written** to a device will be added to its linked list in increasing order of measurement dates and times. Assume that the contents of the buffer to write will be passed on as a character string in the given format, e.g. as “20.01.2001 20:01 768”, where 768 is the value of the measurement in the unit given by the current mode of the device.

A **read** operation on the device will return, in a character buffer, all the entries in the corresponding list in increasing order of their measurement dates and times. The entries returned will be formatted as given above. (Note that the entries will be concatenated into one character buffer.)

During a write or a read operation, if the current mode of the device is not one of the metric units (Celcius, kilometers, kilograms), the measurement data need to be converted into the metric units before being stored or returned.

The module **cleanup** function will free the memory spaces allocated to the linked lists of the devices.

There are three **IOCTL** commands for the driver:

MEMO_CLEAR: This command does not take any arguments. All entries on the device will be deleted.

MEMO_SET_MODE: This command will take an integer as argument and will set the current mode.

MEMO_GET_MODE: This command will take an integer pointer (buffer) as argument and will write the current mode into this buffer.

IMPORTANT NOTES AND REMINDERS

- The data structures used in this project are different than those used in the example SCULL. Any unnecessary pieces of code carried over from SCULL will be penalized.
- There is NO partial read or write.
- The measurements lists are global, i.e., all processes access and use the same lists.
- All entries on the lists are lost when the module is removed.

EXAMPLE: A simple user-space program for this device can be outlined as follows:

```
fd = open("/dev/memo0", ...);
ioctl(fd, MEMO_SET_MODE, 2);
write(fd, buffer1, ...);
ioctl(fd, MEMO_GET_MODE, &buffer3);
if (buffer3 != 1)
      ioctl(fd, MEMO_SET_MODE, 1);
read(fd, buffer2, ...);
ioctl(fd, MEMO_CLEAR);
read(fd, buffer2, ...);
close(fd);
```