

Assignment 6 – device part – platform device driver for platform serial (UART) device(s)

- we must register our platform device instances, using `platform_device_register()` - refer to sample codes provided under `platform_samples/` - we must allocate and initialize appropriate instances of `platform_device{}` objects before calling `platform_device_register()` - since platform devices (serial UART controllers) require explicit resource (I/O address space addresses and IRQ no.) initialization and registration, we must find the available instances of our devices and their respective resources (I/O space addresses and IRQs), with the help of hardware reference manual and/or firmware menu (BIOS, in our case)
- as shown in the `7_rtc.c`, we must add static code for our platform devices' registrations – ideally, we must add our methods and related `platform_device{}` objects to a separate source file and add it to `<KSRC>` - provide a `Kconfig` and `Makefile` for this source file – we must allow static / disable configuration options for this `Kconfig` menu item - we must not allow module configuration option for this `Kconfig` menu item - note that this source file does not provide a device driver – this is a source file that registers our `platform_device{}` object instances corresponding to our serial (UART) device instances !!!

Assignment 6 – device part – platform device driver for platform serial (UART) device(s)

- this code must be integrated into the kernel such that it will execute before the device driver is loaded – refer to 7_rtc.c for such a technique – in addition, this code must be added to `<ksrc>/arch/x86/custom/` folder - In this case, custom is our folder that will contain our source files, Makefile and Kconfig – you have to repeat the procedure followed in assignment3, with appropriate changes needed for this case
- we must also disable the existing device driver of the serial UART device, using kernel configuration menu – you must use the techniques you have learned to find the device driver that is currently managing the serial UART devices
hints – `procfs`, `sysfs`, `.config` and kernel configuration menu !!!
- once integrated, reconfigure the kernel, recompile the kernel, reboot and test that our device is integrated correctly - use `sysfs` for verifying our devices and their attributes

h

Assignment 6 – device part – platform device driver for platform serial (UART) device(s)

- read class notes relevant to platform frame-work for platform devices and platform drivers
- read class note relevant to interrupt handling / ISRs and serial / UART driver / serial_class.c
- read the interrupt internals related slides/pdf
- read chapter 7 of LKD/3 for interrupt handling theory
- read chapter 9 of LDD/3 for hw controller access macros
- read data-sheet for serial/UART controller
- read UART primer slides/pdf
- refer to sample codes under platform_samples/
- refer to <ksrc>/include/linux/platform_device.h
- refer to <ksrc>/drivers/base/platform.c
- a gentle reminder - use your common-sense, when needed !!