



OSY Home 6

Class lecturer: OLI10

Deadline: 12.12.2025, 0:00 (expired)

Submit  1 / 1  - Adam Vlček (VLC028), 12.12.2025 01:33, **1 hour, 33 minutes after deadline**

Select multiple files using CTRL or SHIFT or **drag them to this window**

Assignment

Source code

Upload



Hlubší seznámení se s principy fungování OS. Pro kombinovanou formu studia není možné využít MCU v laboratoři, bude proto využít simulátor GreenThread.

1. Vyzkoušejte si a seznámte se s příkladem připraveným v simulátoru FreeRTOS [osy-freertos-posix](#). Součástí je i zkrácená [kapesní příručka FreeRTOS](#), kde jsou vybrány jen základní funkce FreeRTOSu. V příkladu jsou použity pouze základní funkce pro vytvoření tasku `xTaskCreate`, `vTaskSuspend`, `vTaskResume` a `vTaskDelay`. Fungování by mělo být s dosavadními znalostmi pochopitelné.
2. Seznamte se simulátorem GreenThread [osy-gthreads](#). Na tomto simulátoru je možno si vyzkoušet rozdíl mezi preemptivním a kooperativním multitaskem. Změnu chování lze ovlivnit nastavením makra `GT_PREEMPTIVE`.
3. Rozšiřte si implementaci GT simulátoru o funkce `gt_suspend`, `gt_resume`, `gt_delay` a `gt_task_list`, které by měly mít stejné chování, jako odpovídající funkce ve FreeRTOS. Pro funkci `gt_task_list` bude potřeba rozšířit implementaci GT i o pojmenování vlákna. Jméno bude jako parametr funkce `gt_go`.

Přeneste příklad se třemi tasky z FreeRTOS příkladu do GT simulátoru a ověřte, zda se oba příklady chovají stejně. Přidejte si i nějaký svůj příklad, na kterém si svou implementaci nových funkcí ověříte.

Při implementaci nových funkcí bude potřeba zasahovat i do hotových funkcí, a to zejména: `gt_go`, `gt_yield`, `gt_sig_handle`, a asi `gt_init`. Ostatní funkce v souboru `gthr.c` lze považovat za servisní a není nutno do nich zasahovat.

Nepřehlédněte komentář `// manage timers here`. Právě v této funkci je potřeba řešit časování!